Guide utilisateur de Talend
Open Studio for ESB

7.3.1

Dernière mise à jour : 2020-02-23
# Table des matières

**Copyleft** ........................................................................................................................................................................4

**Qu’est-ce que le Studio Talend ?** ................................................................................................................................. 6

**Projets et Business Models** ............................................................................................................................................ 7
  * Travailler avec les projets ............................................................................................................................................. 7
  * Travailler avec les Business Models .......................................................................................................................... 14

**Services de données** ........................................................................................................................................................... 27
  * Services de données : concepts et principes .................................................................................................................. 27
  * Conception de Jobs et de Routes .................................................................................................................................... 29
  * Conception d’un Service .................................................................................................................................................. 129
  * Gestion des Jobs, des Routes et des Services ............................................................................................................... 167
  * Mapping de flux de données ........................................................................................................................................... 215

**Gestion des métadonnées dans l’intégration de données** ................................................................................................. 274
  * Objectifs ........................................................................................................................................................................... 274
  * Centraliser des métadonnées de base de données ...................................................................................................... 274
  * Centraliser des métadonnées JDBC ................................................................................................................................ 283
  * Centraliser des métadonnées SAS .................................................................................................................................. 289
  * Centraliser des métadonnées File Delimited ................................................................................................................ 292
  * Configurer un schéma File Positional ........................................................................................................................... 299
  * Centraliser des métadonnées File RegEx ........................................................................................................................ 305
  * Centraliser des métadonnées d’un fichier XML ........................................................................................................... 309
  * Centraliser des métadonnées d’un fichier Excel .......................................................................................................... 313
  * Centraliser les métadonnées d’un fichier LDIF ............................................................................................................. 337
  * Centraliser les métadonnées d’un fichier JSON ........................................................................................................... 342
  * Centraliser les métadonnées d’une connexion LDAP .................................................................................................. 360
  * Centraliser les métadonnées Azure Storage .................................................................................................................. 366
  * Centraliser les métadonnées Google Drive ..................................................................................................................... 370
  * Centraliser les métadonnées Marketo ............................................................................................................................. 372
  * Centraliser les métadonnées d’une connexion Salesforce ........................................................................................... 376
  * Centraliser les métadonnées Snowflake ........................................................................................................................ 380
  * Configurer un schéma générique ..................................................................................................................................... 385
  * Centraliser les métadonnées MDM .................................................................................................................................. 391
  * Centraliser les métadonnées Web Service ...................................................................................................................... 410
  * Centraliser une connexion FTP .......................................................................................................................................... 424
  * Utilisation du mapper hiérarchique ................................................................................................................................. 428
  * Exporter une métadonnée en tant que contexte et réutiliser ses paramètres de contexte pour configurer une connexion ............................................................................................................................... 428
  * Comment utiliser les métadonnées centralisées dans un Job ......................................................................................... 438

**Utilisation des routines** ......................................................................................................................................................... 440
  * Gestion des routines ....................................................................................................................................................... 440
Annexes...........................................................................................................................................................466
Personnaliser le Studio Talend et ses préférences.......................................................................................... 466
SQL Templates....................................................................................................................................................520
Règles d’écriture des SQLTemplates................................................................................................................531
Copyleft

Convient à la version 7.3.1. Annule et remplace toute version antérieure de ce guide.

Le contenu de ce document est correct à la date de publication.

Cependant, des mises à jour peuvent être disponibles dans la version en ligne, sur Talend Help Center.

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

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

Mentions légales

Talend et Talend ESB sont des marques déposées de Talend, Inc.

Talend, Talend Integration Factory, Talend Service Factory et Talend ESB sont des marques déposées de Talend, Inc.


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

Accord de licence


Qu’est-ce que le Studio Talend ?

Talend vous fournit un ensemble de Studios, certains open source, certains nécessitant une souscription, que vous pouvez utiliser pour créer vos projets et gérer des données de tous types et de toutes tailles.

Projets et Business Models

Travailler avec les projets

Créer un projet

Un projet est la plus grande structure physique de stockage des différents types d’éléments. Une fois votre Studio Talend démarré et avant de commencer à construire un Business Model, un Job d’intégration de données, une Route ou une autre tâche, vous devez créer ou importer un projet.

Créer un projet lors du premier démarrage du Studio

Procédure

1. Démarrez le Studio Talend et connectez-vous au référentiel local.
2. Dans la fenêtre de connexion, sélectionnez l’option Create a new project et saisissez un nom de projet dans le champ.

Réservez : Gardez à l'esprit que :
• Le nom d’un projet est sensible à la casse.
• Le nom d’un projet doit commencer par une lettre anglaise et peut contenir uniquement des lettres, des chiffres, des traits d’union (-) et des tirets bas (_).
• Le caractère trait d’union (-) est considéré comme un tiret bas (_).


Créer un nouveau projet après le premier démarrage du Studio

Pourquoi et quand exécuter cette tâche

Pour créer un nouveau projet local après le premier démarrage du Studio, procédez comme suit:
**Procédure**

1. Dans la fenêtre de connexion, sélectionnez l’option **Create a new project** et saisissez un nom de projet dans le champ.

2. Cliquez sur **Create** pour créer le projet. Le nouveau projet s’affiche dans la liste des projets existants.

3. Sélectionnez dans la liste le projet et cliquez sur **Finish** pour l’ouvrir dans le Studio.

Vous pouvez passer d’un projet à l’autre, dans la barre de menu du Studio, en utilisant le menu **File > Switch Project or Workspace**.

**Importer un projet Démo**

Talend fournit plusieurs projets démo que vous pouvez importer dans votre Studio Talend. Les projets démo disponibles dépendent du produit que vous utilisez, et peuvent inclure des Jobs, des Services de données prêt(e)s à l’emploi, vous permettant de comprendre les fonctionnalités des différents composants Talend.
Vous pouvez importer le projet démo depuis la page de connexion à votre Studio en tant que projet séparé, ou depuis la perspective **Integration** dans votre projet courant.

**Importer un projet démo en tant que projet séparé**

**Procédure**

1. Lancez votre Studio Talend et, dans la fenêtre de connexion, sélectionnez **Import a demo project** puis cliquez sur **Select**.
2. Dans la boîte de dialogue qui s’ouvre, sélectionnez le projet démo à importer et cliquez sur **Finish**.

**Remarque:** Les projets démo disponibles dans la boîte de dialogue varient selon le produit que vous utilisez.

3. Dans la boîte de dialogue qui s’ouvre, saisissez un nom pour le projet démo que vous souhaitez importer et cliquez sur **Finish**.
   Une barre s’affiche, indiquant la progression de l’opération.
4. Dans la fenêtre de connexion, sélectionnez dans la liste le projet démo importé et cliquez sur **Finish** pour ouvrir ce projet dans le Studio.
   Toutes les données du projet démo sont importées dans le Studio, dans différents dossiers du référentiel, notamment les fichiers d’entrée et les métadonnées de connexion nécessaires à l’exécution des exemples de la démo.

**Importer le projet démo dans votre projet courant**

**Procédure**

1. Lancez votre Studio et, dans la perspective **Integration**, cliquez sur l’icône de la barre d’outils.
2. Dans la boîte de dialogue qui s’ouvre, sélectionnez le projet démo à importer et cliquez sur **Finish**.
   Une barre s’affiche, indiquant la progression de l’opération, puis un message de confirmation s’ouvre.
3. Cliquez sur **OK**.

**Importer un projet**

Dans le Studio Talend, vous pouvez importer un ou plusieurs projets que vous avez déjà créés dans une version antérieure du Studio.

**Pourquoi et quand exécuter cette tâche**

**Importer un projet**

**Procédure**

1. Dans la fenêtre de connexion du Studio, sélectionnez **Import an existing project** puis cliquez sur **Select** pour ouvrir l’assistant **Import**.
2. Saisissez un nom pour votre nouveau projet dans le champ **Project Name**.

**Avertissement:** Assurez-vous que le nom de projet saisi est unique, et gardez à l’esprit que :

- Le nom d’un projet est sensible à la casse.
- Le nom d’un projet doit commencer par une lettre anglaise et peut contenir uniquement des lettres, des chiffres, des traits d’union (-) et des tirets bas (_).
- Le caractère trait d’union (-) est considéré comme un tiret bas (_).

3. Cliquez sur **Select root directory** ou **Select archive file** selon la source de laquelle vous souhaitez importer.


5. Cliquez sur **Finish** pour valider l’opération et retourner à la fenêtre de connexion.
Résultats

Lorsque l’import des projets est terminé, le nom des projets importés sont affichés dans la liste Project de la fenêtre de bienvenue.

Vous pouvez donc sélectionner le projet importé que vous souhaitez ouvrir dans le Studio Talend et cliquer sur Finish pour lancer le Studio.


Importer plusieurs projets

Procédure

1. Dans la fenêtre de connexion du Studio, sélectionnez Import an existing project puis cliquez sur Select pour ouvrir l’assistant Import.
2. Cliquez sur Import several projects.
3. Cliquez sur Select root directory ou Select archive file selon la source de laquelle vous souhaitez importer.
   Par défaut, le Workspace sélectionné est celui qui correspond à la version de votre Studio.
   Parcourez votre système jusqu’au répertoire Workspace du Studio contenant les projets à importer.
5. Cochez la case **Copy projects into workspace** pour faire une copie des projets importés dans le nouveau Workspace.
   Cette option est disponible lorsque vous importez plusieurs projets depuis un répertoire racine.

   **Remarque:** Si vous souhaitez supprimer les dossiers d’origine du projet du répertoire Studio Talend depuis lequel vous importez, décochez cette case. Il est recommandé de conserver une version de sauvegarde.

6. Cochez la case **Hide projects that already exist in the workspace** pour masquer les projets existants dans la liste **Projects**. Cette option est disponible uniquement lorsque vous importez plusieurs projets.

7. Dans la liste **Projects**, sélectionnez les projets à importer et cliquez sur **Finish** pour valider.

**Résultats**

Lorsque l’import des projets est terminé, le nom des projets importés sont affichés dans la liste **Project** de la fenêtre de bienvenue.
Vous pouvez donc sélectionner le projet importé que vous souhaitez ouvrir dans le Studio Talend et cliquer sur Finish pour lancer le Studio.

**Remarque:** Une fenêtre d’initialisation du générateur de code peut apparaître lors du lancement de l’application. Attendez jusqu’à ce que l’initialisation soit terminée.

**Exporter un projet**

Le Studio Talend vous permet d’exporter les projets créés ou importés dans l’instance courante du Studio Talend.

**Procédure**

2. Cochez les cases correspondant aux projets que vous souhaitez exporter. Vous pouvez, en fonction de vos besoins, sélectionner une partie d’un projet à partir du bouton Filter Types... (pour les utilisateurs expérimentés).

3. Dans le champ To archive file, naviguez jusqu’au fichier d’archive dans lequel vous souhaitez exporter les projets sélectionnés ou saisissez son nom.

4. Dans la zone Options, sélectionnez le format de compression et le type de structure que vous souhaitez utiliser.

5. Cliquez sur Finish pour valider les modifications.

Résultats
Le fichier archive contenant les projets exportés est créé dans le répertoire spécifié.

Travailler avec les Business Models

Qu’est-ce qu’un Business Model ?

Les Business Models Talend permettent à toutes les parties prenantes d’un projet d’intégration de données de représenter graphiquement leurs besoins sans avoir à se soucier de leur implémentation technique. Grâce aux Business Models ainsi élaborés, le service informatique de l’entreprise peut
ensuite mieux comprendre ces besoins et les traduire en processus techniques. Un Business Model intègre généralement les systèmes et les processus déjà en place dans l’entreprise, ainsi que ceux dont elle aura besoin à l’avenir.

Les Business Models permettent généralement d’identifier et de résoudre rapidement les goulots d’étranglement et autres points faibles du projet à mettre en place, ainsi que de limiter les dépassements de budget, voire de réduire l’investissement initial. Puis, pendant et après la mise en place du projet, les Business Models peuvent être revus et corrigés, si besoin est. Puis, pendant et après la mise en place du projet, les Business Models peuvent être revus et corrigés, si besoin est.


Dans l’espace de modélisation graphique de la perspective Integration du Studio Talend, vous disposez de nombreux outils vous permettant de :

• modéliser vos besoins métier,
• créer des éléments dans le référentiel de métadonnées et les attribuer à vos objets de Business Model,
• définir les propriétés d’apparence de vos objets de Business Model.

**Création d’un Business Model**

**Procédure**

1. Dans la vue Repository de la perspective Integration, cliquez-droit sur le nœud Business Models et sélectionnez Create Business Model.

L’assistant de création vous guide à travers les étapes de création d’un nouveau Business Model.

2. Saisissez les propriétés du Business Model comme dans le tableau suivant :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nom du nouveau Business Model. Un message s’affiche si vous saisissez des caractères interdits.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Objectif du Business Model ou toute information utile concernant son utilisation.</td>
</tr>
</tbody>
</table>
Champ | Description
--- | ---
Description | Description du Business Model.
Author | Champ en lecture seule affichant par défaut le nom de l’utilisateur courant.
Version | Champ en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l’aide des boutons M et m.
Status | Liste dans laquelle sélectionner le statut du Business Model en cours de création.
Path | Liste dans laquelle sélectionner le dossier dans lequel créer le Business Model.

3. **Le Modeler** s’ouvre dans l’espace de modélisation graphique vide. Vous pouvez créer autant de modèles que vous le souhaitez et tous les ouvrir.

La représentation graphique de votre Business Model est affichée dans l’éditeur graphique. Vous pouvez cliquer sur le panneau approprié pour passer d’un Business Model à un autre. Si plusieurs onglets sont ouverts, assurez-vous que la représentation graphique affichée est correcte.

Le **Modeler** se compose des panneaux suivants :

- L’espace de modélisation graphique de la perspective **Integration**,
- une **Palette** de formes et lignes spécifiques aux Business Models,
- le panneau **Business Model** affichant les informations spécifiques concernant tout le modèle ou une partie.

**Modélisation d’un Business Model**

Si plusieurs onglets sont ouverts dans votre éditeur graphique, assurez-vous que la représentation graphique affichée est correcte, en cliquant sur l’onglet approprié.
En effet, la vue **Business Model** ainsi que les éléments des Menus affichent des informations relatives au modèle actif seulement.

Déposez les éléments de la **Palette**, puis connectez-les à l’aide de liens.

Cette **Palette** est composée de représentations graphiques des **objets** pouvant constituer un Business Model.

Le terme **objets** couvre tous les systèmes stratégiques, les éléments intermédiaires de transformation, de routage, les étapes décisionnelles, jusqu’au type de terminaux du flux de données en sortie. Chaque objet tient un rôle spécifique dans votre Business Model selon la description, la définition et les attributions (assignment) que vous lui attribuez.

Tous les objets sont représentés par des **formes** dans la **Palette** et toutes ces formes peuvent être inclues dans une représentation graphique du process.

Notez que, si les formes ne s’affichent pas dans la **Palette**, il faut cliquer sur l’icône du répertoire **business** pour dérouler la bibliothèques de formes.

**Formes (Shapes)**

 Sélectionnez la forme correspondant à l’**objet** que vous souhaitez inclure dans votre Business Model. Puis cliquez dessus dans la **Palette** et déposez-la dans l’éditeur graphique.

Alternativement, la barre d’accès rapide aux formes apparaît lorsque vous conservez votre curseur immobile quelques instants sur l’espace de modélisation.
Par exemple, si votre Business Model inclut une étape de décision, sélectionnez la forme losange dans la Palette.

**Remarque:** Passez le pointeur de votre souris au-dessus de la barre d’accès rapide, pour faire apparaître les bulles contextuelles de chacune de formes.

Puis cliquez une fois dans l’espace de modélisation pour la faire apparaître graphiquement.

La forme déposée est entourée d’un cadre noir dont les points d’angle vous permettent de la redimensionner selon vos besoins. En outre, une zone de saisie bleue vous permet d’apposer un libellé sur la forme en sélection.

![Forme Losange](image)

Donnez un nom parlant qui vous permettra ensuite d’identifier rapidement le rôle de cette forme dans le process. Deux flèches opposées apparaissent sur cette forme, elles vous permettent de créer des connexions vers d’autres formes.

Vous pouvez ainsi rapidement définir une séquence et des dépendances entre les éléments formant votre process. Vous pouvez ainsi rapidement définir une séquence et des dépendances entre les éléments formant votre process.

Pour plus d’informations, consultez **Connexions** à la page 19.

Les formes disponibles comprennent :

<table>
<thead>
<tr>
<th>Libellé</th>
<th>Détails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td>Représente une condition <em>if</em> dans le flux de données. Permet de prendre des mesures contextuelles.</td>
</tr>
<tr>
<td>Action</td>
<td>Tout type d’action, notamment, transformation, traduction ou format par exemple.</td>
</tr>
<tr>
<td>Terminal</td>
<td>Tout type de terminal de sortie.</td>
</tr>
<tr>
<td>Data</td>
<td>Tout type de données numériques ou alphanumériques.</td>
</tr>
<tr>
<td>Document</td>
<td>Ajoute un objet document qui peut servir en entrée ou en sortie pour fournir les données à traiter.</td>
</tr>
<tr>
<td>Input</td>
<td>Ajoute un objet d’entrée permettant à l’utilisateur de saisir ou de fournir manuellement les données à traiter.</td>
</tr>
<tr>
<td>List</td>
<td>Répertorie les données extraites sous la forme d’une liste. Cette dernière peut être paramétrée pour ne contenir que les données d’une certaine nature.</td>
</tr>
<tr>
<td>Database</td>
<td>Ajoute un objet base de données qui peut servir en entrée ou en sortie du traitement de données.</td>
</tr>
<tr>
<td>Actor</td>
<td>Représente schématiquement tous les acteurs du décisionnel à l’implémentation technique.</td>
</tr>
<tr>
<td>Ellipse</td>
<td>Ajoute une forme elliptique.</td>
</tr>
</tbody>
</table>
Connexions

Pourquoi et quand exécuter cette tâche

Une des phases essentielles de la conception d’un Business Model consiste à mettre en place les relations entre les formes source et cible.

Plusieurs solutions sont offertes :

- Relationship
- Directional Relationship
- Bidirectional Relationship

Sélectionnez l’outil Relationship de la Palette. Puis dans l’éditeur graphique, tirez un lien d’une forme vers une autre pour dessiner la connexion.

Ou si vous n’avez pas encore placé la forme cible de la connexion, vous pouvez créer la relation et la cible en une fois :

**Procédure**

1. Placez simplement votre souris au dessus de la forme source pour faire apparaître les doubles flèches.
2. Sélectionnez la flèche correspondant à la connexion dont vous avez besoin.
3. Faites glisser l’une des flèches vers une zone vide de l’espace de modélisation et relâchez la souris pour afficher le menu contextuel.
4. Sélectionnez la connexion appropriée dans la liste : Vous pouvez choisir entre Create Relationship To, Create Directional Relationship To et Create Bidirectional Relationship To.
5. Puis sélectionnez l’élément cible de la connexion parmi les éléments proposés.
Vous pouvez créer une connexion vers un élément déjà en place dans votre modèle. Sélectionnez *Existing Element* dans le menu contextuel puis choisissez l'élément à connecter.

La relation s'établit automatiquement entre les deux formes sélectionnées.

La nature de cette connexion peut être décrite à l'aide d'éléments du référentiel de métadonnées (*Repository*) et sa mise en forme peut être paramétrée dans le panneau *Properties*, consultez *Propriétés* à la page 22.
Lors de la création d'une connexion, une zone de saisie vous permet d'ajouter un libellé à la connexion que vous venez de créer. Choisissez un nom significatif afin de vous aider à identifier rapidement le type de relation créée.

**Remarque:** Vous pouvez également ajouter des notes et des commentaires à votre process pour permettre une reprise du modèle par la suite.

Pour plus d'informations, consultez Commenter et réarranger un Business Model à la page 21.

**Commenter et réarranger un Business Model**

Les outils suivants de la Palette vous permettent de personnaliser votre Business Model :

<table>
<thead>
<tr>
<th>Libellé</th>
<th>Détails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>Sélectionnez et déplacez les formes et relations dans l'éditeur graphique du Modeler.</td>
</tr>
<tr>
<td>Zoom</td>
<td>Zoomez sur une partie de la représentation graphique Pour en voir le détail. Pour effectuer un zoom arrière, appuyez sur la touche Maj et cliquez sur l'éditeur graphique.</td>
</tr>
<tr>
<td>Note/Text/Note attachment</td>
<td>Permet d'ajouter des commentaires, des notes afin de conserver toute information utile concernant tout ou partie du process.</td>
</tr>
</tbody>
</table>

**Ajouter une note ou du texte libre**

Pour ajouter une note, sélectionnez l'outil Note dans la Palette, à droite de l'éditeur graphique.

Alternativement, cliquez-droit sur la représentation graphique ou sur la forme que vous souhaitez annoter et sélectionnez Add Note. Ou sélectionnez l'outil Note dans la barre d'accès rapide.

Une note de type Post-it apparaît sur l'éditeur graphique. Si la note est liée à un élément particulier, un lien est automatiquement créé.

Saisissez votre commentaire dans la zone de saisie ou, si cette dernière ne s'affiche pas, saisissez directement sur la note.

Pour ajouter une note attachée à une forme, cliquez sur la flèche vers le bas, à côté de l'outil Note dans la Palette puis sélectionnez Note attachment. Lorsque la flèche noire apparaît, faites-la glisser vers une zone vide de l'éditeur. Relâchez pour afficher le menu contextuel du lien vous demandant de créer une nouvelle note.

Vous pouvez également sélectionner la fonction Add Text pour saisir librement du texte directement dans l'éditeur. Vous pouvez accéder à cette fonctionnalité dans le menu déroulant Note de la Palette ou via le raccourci situé à côté de la fonction Add Note dans la barre d'accès rapide.

**Réarranger une vue métier**

Vous pouvez adapter le "look and feel" de votre Business Model via le menu contextuel.
Placez votre curseur dans l’éditeur, cliquez-droit pour afficher le menu, et sélectionnez **Arrange all**. Les formes se déplacent automatiquement pour une lisibilité optimale du Business Model.

Vous pouvez également sélectionner manuellement tout ou partie du process.

Pour ce faire, cliquez-droit n’importe où dans l’éditeur, et choisissez **Select** dans le menu contextuel.

Vous pouvez sélectionner :

- **All** : toutes les formes et connexions du processus,
- **All shapes** : toutes les formes présentes dans l’éditeur de modélisation,
- **All connectors** : toutes les connexions reliant les formes.

À partir de ce menu, vous pouvez également effectuer un zoom avant ou arrière sur une partie du modèle et ainsi changer la vue affichée.

**Propriétés**

Les informations de propriétés affichées dans l’onglet **Business Model** correspondent à la sélection en cours dans l’éditeur. Les propriétés peuvent concerner l’ensemble du processus ou seulement une partie du processus selon la sélection. Si rien n’est sélectionné dans l’éditeur, l’onglet **Business Model** fournit des informations générales sur lui-même.

La vue **Business Model** comporte différents types d’informations concernant : les informations de création du Business Model dans l’onglet **Main** le format dans l’onglet **Appearance** les règles et la grille dans l’onglet **Rulers and Grid** les attributions dans l’onglet **Assignment**.

Les informations de création du Business Model dans l’onglet **Main** Pour plus d’informations concernant l’onglet **Main**, consultez **Afficher les onglets/vues de configuration des Jobs** à la page 493.
Onglet Appearance

À partir de l’onglet Appearance (Formats), vous pouvez modifier les couleurs de remplissage et de bordures, changer l’apparence des formes et des liens afin de personnaliser votre Business Model et le rendre plus lisible.

L’onglet Business Model inclut les options de format suivants :

- changer la couleur de la forme
- colorer les bordures
- ajouter du texte
- ajouter un effet dégradé,
- ajouter un effet ombré

Vous pouvez également déplacer et gérer les différentes formes utilisées dans la vue de votre Business Model à l’aide des outils de modification. Cliquez-droit sur une des formes, pour accéder aux outils appropriés.

Onglet Rulers and Grid

Procédure

1. Pour afficher l’onglet Rulers & Grid dans l’éditeur graphique, cliquez sur dans la Palette, puis cliquez sur une zone vide de l’espace de modélisation pour désélectionner toute sélection courante.

2. Cliquez sur l’onglet Rulers & Grid pour accéder aux paramètres de configuration des règles et de la grille.

3. Dans la zone Display, cochez la case Show Ruler pour afficher la Règle (Ruler), ou Show Grid pour afficher la Grille (Grid) ou les deux. Grid in front place la grille devant toutes les autres formes du process.

4. Dans la zone Measurement, définissez l’unité de mesure de la règle Centimeters (centimètres), Inches (pouces) ou Pixels.

5. Dans la zone Grid Line, cliquez sur le bouton Color pour définir la couleur des lignes de la grille et sélectionnez le type de lignes dans la liste Style.

6. Cochez la case Snap To Grid pour aligner les formes en fonction de la grille ou cochez la case Snap To Shapes pour aligner les formes en fonction des formes déjà déposées dans le Business Model.

Vous pouvez également restaurer les valeurs par défaut en cliquant sur le bouton Restore Defaults.
Onglet Assignment

L’onglet **Assignment** affiche, sous forme de tableau, les détails des attributs du **Repository** (référentiel) affectés à la forme ou à la connexion sélectionnée.

Pour afficher les informations d’attribution, sélectionnez une forme ou une connexion dans la vue active, puis cliquez l’onglet **Assignment** de la vue **Business Model**.

Sinon, passez simplement la souris au dessus de la forme à laquelle vous avez affecté des éléments pour faire apparaître la liste des attributions.

Vous pouvez modifier certaines informations ou lier un commentaire. En outre, si vous mettez à jour des données dans le **Repository**, les informations d’attribution sont automatiquement mises à jour.
Pour plus d’informations concernant les attributions, voir Attribution d’éléments du Repository à un Business Model à la page 25.

### Attribution d’éléments du Repository à un Business Model

L’onglet **Assignment** de la vue **Business Models** répertorie les éléments sélectionnés dans le **Repository** et qui ont été affectés à une forme du Business Model.

Vous pouvez définir ou décrire un objet de votre Business Model en lui affectant (Assignment) différents types d’informations, par exemple en lui affectant des éléments métadonnées.

Vous pouvez mentionner la nature des métadonnées affectées ou traitées, facilitant ainsi la phase de développement technique à venir.

Pour attribuer un élément, glissez l’élément du **Repository** vers une des formes composant votre Business Model courant.

Le tableau d’attribution, **Assignment**, situé sous l’espace de modélisation, est mis à jour automatiquement à l’aide des données d’attribution de l’objet en sélection.

Les types d’éléments du référentiel que vous pouvez attribuer sont les suivants :

<table>
<thead>
<tr>
<th>Elément</th>
<th>Détails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Designs</td>
<td>Si la représentation graphique d’un Job est disponible, elle peut être réutilisée comme métadonnée du Business Model actif.</td>
</tr>
<tr>
<td>Métadonnées</td>
<td>Toute donnée descriptive conservée dans le référentiel peut être affectée à un objet de la vue. Il peut s’agir par exemple des coordonnées de connexion à une base de données.</td>
</tr>
<tr>
<td>Propriétés</td>
<td>Vous pouvez réutiliser comme métadonnées tout Business Model conservé dans le référentiel de ce projet.</td>
</tr>
<tr>
<td>Documentation</td>
<td>Documents de tout type et tout format. Il peut s’agir de documentation technique, de spécifications en format texte ou d’une simple description de vos bases de données.</td>
</tr>
<tr>
<td>Elément</td>
<td>Détails</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Routines (Code)</td>
<td>Si vous avez développé des routines, d'automatisation de tâches, par exemple, vous pouvez les attribuer à votre Business Model. Le répertoire Routine est conservé sous le nœud Code de la vue Repository.</td>
</tr>
</tbody>
</table>

Pour plus d'informations concernant les éléments du Repository, consultez Qu’est-ce qu’un Job ? à la page 29
Services de données

Services de données : concepts et principes

Architecture fonctionnelle de Talend Data Integration

L'architecture fonctionnelle du Studio Talend est un modèle architectural qui identifie les fonctions, les interactions et les besoins informatiques correspondants du Studio Talend. L’architecture d’ensemble a été décrite en isolant les fonctionnalités spécifiques en blocs fonctionnels.

La figure ci-dessous illustre les principaux blocs fonctionnels utilisés pour la prise en charge des tâches d’intégration de données de votre entreprise.

Remarque:
Différents blocs fonctionnels sont définis :

Différents blocs fonctionnels sont définis :

- Le bloc Clients inclut un ou plusieurs Studio Talend et navigateur(s) Web pouvant être sur la même machine ou sur des machines différentes.

Dans le Studio, vous pouvez mener à bien des processus d’intégration de données sans tenir compte du volume des données et de la complexité des processus. Le Studio Talend vous permet de travailler sur les projets pour lesquels vous avez les droits d’accès.

Depuis le navigateur Web, vous pouvez vous connecter à un Talend Administration Center distant, via un protocole HTTP sécurisé.

- Le bloc Talend Servers comprend le module Talend Administration Center (serveur d’applications) basé Web connecté :

  - à deux référentiel partagés : un basé sur un serveur SVN ou Git et l’autre sur un référentiel d’artefacts,
  - à des serveur(s) d’exécution Talend.

Talend Administration Center permet la gestion et l’administration de tous les projets. Les métadonnées d’administration (comptes d’utilisateurs, droits d’accès et autorisations des projets, par exemple) sont stockées dans la base de données d’Administration. Les métadonnées d’administration (comptes d’utilisateurs, droits d’accès et autorisations des projets, par exemple) sont stockées sur le serveur de base de données et les métadonnées des projets (Jobs, Business Models et Routines par exemple) sont stockées sur le serveur SVN ou Git.

Pour plus d’informations, consultez le Guide utilisateur de Talend Administration Center.

- Le bloc Repositories inclut le serveur SVN ou Git et le référentiel d’artefacts. Le serveur SVN ou Git est utilisé pour centraliser toutes les métadonnées des projets telles que les Jobs, Business Models, Routines, Routes, Services, partagés entre les différents utilisateurs finaux et accessibles depuis le Studio Talend afin de les développer et à partir de Talend Administration Center pour les publier, déployer et moniterer.

Le référentiel d’artefacts est utilisé pour stocker :

  - les mises à jour du logiciel (Software Updates) disponibles au téléchargement,
  - les Jobs publiés depuis le Studio Talend prêts à être déployés et exécutés.
• Le bloc **Talend Execution Servers** représente un ou plusieurs serveurs d’exécution, déployé(s) dans votre système d’information. Les Jobs de **Talend** sont déployés sur le Job Serveur à travers le Job Conductor de l’Administration Center, afin d’être exécutés à l’heure, la date ou au moment programmé.

Pour des informations plus détaillées sur les serveurs distants d’exécution, consultez le Guide utilisateur de Talend Administration Center.

• Le bloc **Databases** représente les bases de données d’Administration, d’Audit et de Monitoring. La base de données d’Administration est utilisée pour gérer les comptes d’utilisateurs, droits d’accès et autorisations des projets notamment. La base de données d’Audit est utilisée pour évaluer les différents aspects des Jobs implémentés dans un Projet réalisé via le Studio Talend. Cette opération permet de fournir de solides facteurs quantitatifs et qualitatifs au support décisionnel orienté processus. Les bases de données Monitoring incluent les bases de données Talend Activity Monitoring Console et Service Activity Monitoring.

Talend Activity Monitoring Console vous permet de montrer l’exécution des processus techniques. Elle fournit des fonctionnalités de monitoring détaillé pouvant être utilisées afin de consolider les informations de log collectées, comprendre les interactions sous-jacentes des flux de données, empêcher les erreurs générées de manière inattendue et aider à prendre des décisions concernant la gestion du système.

Service Activity Monitoring vous permet de montrer les appels de services. Il fournit des fonctionnalités de monitoring et des informations consolidées concernant les événements dont l’utilisateur final peut comprendre les requêtes et les réponses sous-jacentes, montrer les erreurs générées de manière inattendue et aider à prendre des décisions concernant la gestion du système.

**Architecture fonctionnelle de Talend ESB**

L’architecture fonctionnelle du Studio Talend est un modèle architectural qui identifie les fonctions, les interactions et les besoins informatiques correspondants du Studio Talend. L’architecture globale isole les différentes fonctionnalités et les schématisé sous forme de blocs fonctionnels.

La figure ci-dessous illustre les principaux blocs fonctionnels utilisés pour la prise en charge des tâches d’intégration de votre d’entreprise.
Les trois différents types de blocs fonctionnels sont définis comme suit :

- Le bloc **Client** comprend un Studio Talend dans lequel vous pouvez effectuer des processus d'intégration de données ou de services de données, des routes de médiation et des services. Pour plus d'informations concernant chaque élément du Repository, consultez les chapitres respectifs du présent guide utilisateur.

- Le bloc **Talend Execution Servers** représente un ou plusieurs conteneurs de Runtime (conteneurs d'exécution) Talend déployés dans votre système d'information. Le Talend Runtime vous permet de déployer et d'exécuter les Jobs, Routes et Services créés dans le Studio. Pour plus d'informations concernant le déploiement d'éléments dans le Talend Runtime via le Studio, consultez Exporter un Service pour un déploiement à la page 164. Pour plus d'informations concernant le Talend Runtime, consultez le Guide Talend ESB Runtime Configuration Guide (en anglais).

   Un seul Talend Runtime peut être utilisé pour tous les Services d'Infrastructure, les fournisseurs et consommateurs ESB. Il est habituel d'avoir des Services d'infrastructure ESB sur un conteneur dédié et de déployer les participants (fournisseurs et consommateurs) sur différents conteneurs. Si vous avez plusieurs conteneurs de Runtime Talend dans lesquels déployer vos Routes et Services, vous pouvez utiliser la répartition de charge afin d'équilibrer leurs exécutions selon vos besoins. Toutes les instances de Runtime Talend communiquent entre elles via le Service Locator pour identifier l'instance la plus à même de déployer et d'exécuter vos Routes et Services.

- Le bloc **Monitoring Database** représente la base de données de Service Activity Monitoring qui monte les appels de services.

   Service Activity Monitoring permet aux utilisateurs finaux de moniter les appels de services. Il permet de moniter les informations d'un événement consolidé, que l'utilisateur final peut utiliser pour comprendre les requêtes et les réponses sous-jacentes composant un événement, de moniter les erreurs pouvant être générées de manière inattendue et venir en support des systèmes décisionnels. Pour plus d'informations concernant le SAM, consultez le chapitre correspondant dans le Guide Talend ESB Runtime Configuration Guide (en anglais).


### Conception de Jobs et de Routes

#### Qu'est-ce qu'un Job ?

**Remarque:**

Les Jobs de services de données sont appelés Jobs dans la documentation suivante.

Un Job constitue la couche d'exécution ou l'implémentation technique d'un Business Model. Il est la représentation graphique fonctionnelle d'un ou plusieurs composants connectés, permettant de définir et d'exécuter des processus de gestion de flux de données. Il traduit les besoins métier en code, en routines ou en programmes, puis se charge d'exécuter ces derniers. En d'autres termes, le Job permet de mettre en place votre flux de données.

Les Jobs que vous créez peuvent gérer les différentes sources et cibles dont vous avez besoin lors de vos processus d'intégration de données ou de services de données, ou lors de tout autre processus lié.

Avec le Studio Talend, vous pouvez:
• modéliser des actions d’intégration de données ou des services de données grâce à la bibliothèque de composants techniques,
• changer les paramètres par défaut des composants et même créer de nouveaux composants ou familles de composants qui correspondent au mieux à vos besoins,
• paramétrer les connexions et les relations entre composants afin de définir la nature des actions et leur déroulement,
• accéder au code généré pour le programme ou les composants afin de le modifier ou de le documerter,
• créer et ajouter des éléments au Référentiel à des fins de réutilisation et de partage (avec d’autres projets, d’autres processus ou, à moyen terme, avec d’autres utilisateurs).

Avertissement:

Prise en main d’un Job simple

Cette section présente un exemple pas à pas qui montre comment créer un Job, ajouter des composants, les configurer et exécuter le Job. Ce Job se nomme A_Basic_Job. Il lit un fichier texte, affiche son contenu dans la console Run et écrit les données dans un autre fichier texte.

Créer un Job

Le Studio Talend vous permet de créer un Job en déposant différents composants techniques de la Palette dans l’espace de modélisation graphique, puis en connectant ces composants les uns aux autres.

Pourquoi et quand exécuter cette tâche

Pour créer le Job d’exemple décrit dans cette section, procédez comme suit :

Procédure

1. Dans la vue Repository de la perspective Integration, cliquez-droit sur le nœud Job Designs et sélectionnez Create job dans le menu contextuel.
L’assistant de création **New Job** vous guide dans les étapes de définition des propriétés de votre nouveau Job.

2. Renseignez les propriétés du Job comme montré dans la capture d’écran.

Les champs correspondent aux propriétés suivantes :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Saisissez un nom pour le nouveau Job. Un message s’affiche si vous saisissez des caractères interdits.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Saisissez toute information que vous jugerez utile concernant l’utilisation du Job.</td>
</tr>
<tr>
<td>Champ</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Description</td>
<td>Si nécessaire, saisissez une description contenant toute information permettant de décrire ce que fait le Job et comment il le fait.</td>
</tr>
<tr>
<td>Author</td>
<td>Champ en lecture seule affichant par défaut le nom de l’utilisateur courant.</td>
</tr>
<tr>
<td>Locker</td>
<td>Champ en lecture seule affichant par défaut l’identifiant de l’utilisateur ayant verrouillé le Job courant. Ce champ est vide lorsque vous créez un Job, et n’a des données que lorsque vous éditez les propriétés d’un Job existant.</td>
</tr>
<tr>
<td>Version</td>
<td>Champ en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l’aide des boutons M et m. Pour plus d’informations, consultez Gestion des versions d’un Job ou d’une Route à la page 190.</td>
</tr>
<tr>
<td>Status</td>
<td>Sélectionnez dans la liste le statut du Job que vous créez.</td>
</tr>
<tr>
<td>Path</td>
<td>Sélectionnez le dossier dans lequel vous souhaitez créer le Job.</td>
</tr>
</tbody>
</table>

3. Un onglet de modélisation vide s’ouvre, indiquant simplement le nom du nouveau Job.

Résultats

Le Job que vous avez créé s’affiche sous le nœud Job Designs, dans la vue Repository.

Vous pouvez ouvrir autant de Jobs créés que vous le souhaitez, simplement en double-cliquant sur le nom du Job dans le Repository.

Ajouter des composants au Job

Une fois le Job créé, il faut ajouter des composants dans l’espace de modélisation graphique, un tFileInputDelimited, un tLogRow et un tFileOutputDelimited dans cet exemple.

Il y a différents moyens d’ajouter un composant dans l’espace modélisation graphique. Vous pouvez :

• trouver votre composant dans la Palette en saisissant des mots-clés dans le champ de recherche de la Palette et le déposer dans l’espace de modélisation graphique,
• ajouter un composant en saisissant directement des mots-clés dans l’espace de modélisation graphique,
• ajouter un composant de sortie en le glissant depuis un composant d’entrée déjà dans l’espace de modélisation graphique.
• glisser-déposer un élément de métadonnées centralisé depuis le nœud Metadata dans l’espace de modélisation graphique et sélectionner le composant qui vous intéresse dans la boîte de dialogue Components.

Cette section décrit les trois premières méthodes. Pour plus d’informations concernant la manière de déposer un composant depuis le nœud Metadata, consultez le Centraliser des métadonnées de base de données à la page 274.

Déposer le premier composant depuis la Palette

Pourquoi et quand exécuter cette tâche

Le premier composant de cet exemple est ajouté depuis la Palette. Ce composant définit la première tâche exécutée par le Job. Dans cet exemple, il faut ajouter un composant tFileInputDelimited afin de lire un fichier texte.

Pour déposer un composant depuis la Palette, procédez comme suit :

Procédure

1. Saisissez les mots-clés dans le champ de recherche de la Palette et appuyez sur Entrée afin de valider votre recherche.

Les mots-clés peuvent être le nom complet ou partiel du composant ou une phrase décrivant sa fonctionnalité, si vous ne connaissez pas son nom, par exemple, tfileinputde, fileinput, ou read file row by row (ou lit un fichier ligne par ligne). La Palette affiche uniquement les familles dans lesquelles le composant peut être trouvé. Si la Palette n’est pas affichée dans le Studio, consultez Modifier la disposition et les paramètres de la Palette à la page 489.

Remarque:

• Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case Also search from Help when performing a component searching dans la fenêtre des Preferences > Palette Settings. Pour plus d’informations, consultez Préférences de la Palette (Talend > Palette Settings) à la page 511.
• Les composants n’ont pas toujours de lettre comme préfixe dans leur nom. La bonne pratique recommande de spécifier simplement le corps principal lors de la recherche d’un composant par son nom.
2. Sélectionnez le composant à utiliser et, dans l’espace de modélisation graphique, cliquez à l’endroit où vous souhaitez l’ajouter.

Résultats

Notez que vous pouvez procéder de la même manière afin de déposer une note dans votre Job.

Chaque composant ajouté au Job s’affiche dans un rectangle bleu montrant que c’est un sous-job.

Ajouter le deuxième composant en saisissant son nom dans l’espace de modélisation graphique

Pourquoi et quand exécuter cette tâche

Le second composant du Job est ajouté en saisissant son nom directement dans l’espace de modélisation graphique au lieu de le déposer depuis la Palette ou le noeud Metadata.


Pour ajouter un composant directement dans l’espace de modélisation graphique, procédez comme suit :
**Procédure**

1. Dans l’espace de modélisation graphique, cliquez à l’emplacement où vous souhaitez ajouter le composant et saisissez votre (vos) mot(s)-clé(s), à savoir le nom complet ou partiel du composant, ou une phrase décrivant sa fonctionnalité si vous ne connaissez pas son nom. Dans cet exemple, commencez à saisir `log`.

**Remarque:**
- Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case *Also search from Help when performing a component searching* dans la fenêtre des *Preferences > Palette Settings*. Pour plus d’informations, consultez *Préférences de la Palette (Talend > Palette Settings)* à la page 511.
- Les composants n’ont pas toujours de lettre comme préfixe dans leur nom. La bonne pratique recommande de spécifier simplement le corps principal lors de la recherche d’un composant par son nom.

Une liste s’affiche sous le champ de recherche et affiche tous les composants correspondant à votre recherche, dans l’ordre alphabétique.

2. Double-cliquez sur le composant que vous souhaitez utiliser, le *tLogRow* dans cet exemple, afin de l’ajouter dans l’espace de modélisation graphique.

**Ajouter un composant de sortie en le glissant depuis un composant d’entrée**

**Pourquoi et quand exécuter cette tâche**

Ajoutez le troisième composant, un *tFileOutputDelimited*, pour écrire les données lues depuis le fichier source dans un autre fichier texte. Ajoutez-le en glissant depuis le composant *tLogRow*, ici composant d’entrée pour le composant à ajouter.

**Procédure**

1. Cliquez sur le *tLogRow* pour afficher l’icône `o` placée à côté.
2. Glissez-déposez l’icône `o` à l’emplacement où vous souhaitez ajouter un nouveau composant.

Un champ textuel et une liste de composants s’affichent. La liste des composants affiche tous les composants pouvant être connectés au composant d’entrée.
3. Pour réduire les résultats de la recherche, saisissez dans le champ le nom du composant à ajouter, ses premières lettres ou une phrase décrivant sa fonctionnalité si vous ne connaissez pas son nom, puis double-cliquez sur le composant de votre choix, le `tFileOutputDelimited` dans cet exemple, dans la liste de composants, pour l’ajouter à l’espace de modélisation graphique.

**Remarque:**
- Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case *Also search from Help when performing a component searching* dans la fenêtre des *Preferences > Palette Settings*. Pour plus d’informations, consultez *Préférences de la Palette (Talend > Palette Settings)* à la page 511.
- Les composants n’ont pas toujours de lettre comme préfixe dans leur nom. La bonne pratique recommande de spécifier simplement le corps principal lors de la recherche d’un composant par son nom.

Le nouveau composant est automatiquement connecté au composant d’entrée, le `tLogRow`, à l’aide d’un lien `Row > Main`. 
**Connecter les composants**

Maintenant que les composants sont ajoutés dans l'espace de modélisation graphique, ils doivent être connectés entre eux. Lorsque des composants sont connectés, ils forment un sous-job. Les Jobs sont composés d'un ou plusieurs sous-jobs effectuant différentes opérations.

Dans cet exemple, comme le `tLogRow` et le `tFileOutputDelimited` sont déjà connectés, connectez le `tFileInputDelimited` au `tLogRow`.

Afin de relier les composants, utilisez l’une des deux méthodes :

**Méthode par clics**

**Procédure**

1. Cliquez-droit sur le composant source, le `tFileInputDelimited` dans cet exemple.
2. Dans le menu contextuel qui s’ouvre, sélectionnez le type de connexion que vous souhaitez utiliser afin de relier les composants, *Row > Main* dans cet exemple.
3. Cliquez sur le composant cible pour créer le lien, `tLogRow` dans cet exemple.

![Diagramme de connexion](image)

Notez qu’un cercle noir s’affiche si le composant cible n’est pas compatible avec le lien.

Selon la nature et le rôle des composants que vous souhaitez connecter, différents types de liens sont disponibles. Seules les connexions autorisées sont listées dans le menu contextuel.

**Méthode par glisser-déposer**

**Procédure**

1. Cliquez sur le composant d’entrée, le `tFileInputDelimited` dans cet exemple.
2. Lorsque l’icône O s’affiche, cliquez dessus et glisser votre curseur sur le composant de destination, le `tLogRow` dans cet exemple.

Un lien *Row > Main* est automatiquement créé entre ces deux composants.
Cette méthode requiert moins d’opérations mais elle ne fonctionne uniquement avec les connexions de type **Row**: Main, Lookup, Output, Filter et Reject, selon la nature et le rôle des composants que vous connectez.

### Configurer les composants

Maintenant que les composants sont reliés entre eux, leurs propriétés doivent être définies.

Pour plus de détails concernant les propriétés des composants, consultez [Définir les propriétés des composants](#) à la page 55.

### Configurer le composant **tFileInputDelimited**

#### Procédure

1. Double-cliquez sur le composant **tFileInputDelimited** pour ouvrir sa vue **Basic settings**.

2. Cliquez sur le bouton […+] à côté du champ **File Name/Stream**.
3. Parcourez votre système ou saisissez le chemin vers le fichier d’entrée, `customers.txt` dans cet exemple.
5. Cliquez sur le bouton […+] correspondant au champ **Edit schema**.
6. Dans l’éditeur de schéma qui s’ouvre, cliquez trois fois sur le bouton [+•] afin d’ajouter trois colonnes.
7. Nommez ces colonnes **id**, **CustomerName** et **CustomerAddress** respectivement puis cliquez sur **OK** afin de fermer l’éditeur.
8. Dans la boîte de dialogue qui s’ouvre, cliquez sur **OK** afin d’accepter la propagation des changements.
Cela vous permet de copier le schéma créé au composant suivant, le **tLogRow** dans cet exemple.

### Configurer le composant tLogRow

**Procédure**

1. Double-cliquez sur le **tLogRow** pour ouvrir sa vue **Basic settings**.
2. Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.
   Ainsi, le contenu du fichier **customers.txt** est affiché sous forme de tableau afin d’être plus facilement lisible.

### Configurer le composant tFileOutputDelimited

**Procédure**

1. Double-cliquez sur le composant **tFileOutputDelimited** pour ouvrir sa vue **Basic settings**.
2. Cliquez sur le bouton [...] à côté du champ File Name.
3. Parcourez votre système ou saisissez le chemin d’accès à votre fichier de sortie, customers.csv dans cet exemple.
4. Cochez la case Include Header.
5. Si nécessaire, cliquez sur le bouton Sync columns afin de récupérer le schéma du composant d’entrée.

**Exécuter le Job**

**Pourquoi et quand exécuter cette tâche**

Maintenant que les composants sont configurés, le Job peut être exécuté.

Pour ce faire, procédez comme suit :

**Procédure**

1. Appuyez sur les touches Ctrl+S pour enregistrer le Job.
2. Cliquez sur la vue Run, puis cliquez sur Run pour exécuter le Job.

**Résultats**

Le fichier est lu ligne par ligne, les champs extraits sont affichés dans la console Run et écrits dans le fichier de sortie spécifié.
Qu’est-ce qu’une Route ?

Une Route est une règle définissant comment les messages sont déplacés d’un service (ou endpoint) à un autre. C’est un processus graphique, contenant deux composant ou plus reliés ensemble, qui vous permet de configurer et de tester facilement vos règles de routage et de médiation.

Les Routes que vous créez peuvent couvrir toutes les sources et cibles nécessaires à vos processus de routage.

Lorsque vous créez un Job dans le Studio Talend, vous pouvez :

- mettre en place des règles de routage ou de médiation à l'aide d'une bibliothèque de composants techniques basés sur les Enterprise Integration Patterns standards,
- changer les paramètres par défaut des composants et même créer de nouveaux composants ou familles de composants qui correspondent au mieux à vos besoins,
- créer et ajouter des éléments au référentiel afin de les réutiliser et de les partager (dans d’autres projets ou Routes ou avec d’autres utilisateurs).

Avertissement:

Pour pouvoir exécuter les Jobs que vous avez créés dans le Studio Talend, vous devez installer une JVM Oracle 1.8 ou une version ultérieure (les JVM IBM ne sont pas supportées). Vous pouvez la télécharger sur http://www.oracle.com/technetwork/java/javase/downloads/index.html.
Présentation d'une Route simple

Cette section fournit un exemple complet vous permettant de créer une Route simple, d’y ajouter des composants et de l’exécuter.

Cette Route sera nommée A_Basic_Route et va démarrer un échange de messages, configurer le corps des messages et afficher les messages de log dans la console d’exécution.

Créer une Route

Le Studio Talend vous permet de créer des Routes en déposant différents composants de médiation de la Palette dans l’espace de modélisation graphique et en les connectant.

Les Routes ainsi créées sont stockées dans un référentiel central. Vous pouvez créer différents dossiers afin de mieux les classer.

Pourquoi et quand exécuter cette tâche

Pour créer la Route d’exemple décrite dans cette section, procédez comme suit :

Procédure

1. Dans la vue Repository de la perspective Integration, cliquez-droit sur le nœud Routes et sélectionnez Create Route dans le menu contextuel.

L’assistant New Route s’ouvre pour vous permettre de définir les propriétés principale de la nouvelle Route.
2. Renseignez les propriétés de la Route comme dans la capture d’écran précédente.
Les champs correspondent aux propriétés suivantes :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nom de la nouvelle Route. Un message s’affiche si vous saisissez des caractères interdits.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Objectif de la Route ou toute information utile concernant l’utilisation de la Route.</td>
</tr>
<tr>
<td>Description</td>
<td>Description de la Route contenant toute informations permettant de décrire ce que fait la Route et de quelle manière.</td>
</tr>
<tr>
<td>Author</td>
<td>Champ en lecture seule affichant par défaut le nom de l’utilisateur courant.</td>
</tr>
<tr>
<td>Locker</td>
<td>Champ en lecture seule qui affiche par défaut l’identifiant de l’utilisateur ayant verrouillé la Route. Ce champ est vide lorsque vous créez une Route et contient des données uniquement lorsque vous modifiez les propriétés d’une Route existante.</td>
</tr>
<tr>
<td>Version</td>
<td>Champ en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l’aide des boutons M et m. Pour plus d’informations, consultez Gestion des versions d’un Job ou d’une Route à la page 190.</td>
</tr>
<tr>
<td>Status</td>
<td>Liste dans laquelle sélectionner le statut de la Route que vous créez.</td>
</tr>
<tr>
<td>Path</td>
<td>Liste dans laquelle sélectionner le dossier dans lequel la Route sera créée.</td>
</tr>
</tbody>
</table>

3. Un espace de modélisation graphique vide s’ouvre et affiche le nom de la Route comme libellé d’un onglet.
Résultats
La Route créée s’affiche à présent sous le nœud Routes dans la vue Repository.
Vous pouvez ouvrir une ou plusieurs Routes créée(s) en double-cliquant sur celle(s)-ci dans la vue Repository.

Ajouter des composants à la Route
Une fois la Route créée, vous devez ajouter des composants dans l’espace de modélisation graphique, un cTimer, un cSetBody et un cLog dans cet exemple.
Il y a différents moyens d’ajouter un composant dans l’espace modélisation graphique. Vous pouvez :
• trouver votre composant dans la Palette en saisissant des mots-clés dans le champ de recherche de la Palette et le déposer dans l’espace de modélisation graphique, Les mots-clés peuvent être le nom complet ou partiel du composant ou une phrase décrivant sa fonctionnalité, si vous ne connaissez pas son nom. Si la langue de votre Studio est le français, vous pouvez saisir vos mots-clés en français.
• ajouter un composant en saisissant directement son nom complet ou partiel dans l’espace de modélisation graphique.
• ajouter un composant de sortie en le glissant depuis un composant d’entrée déjà dans l’espace de modélisation graphique.

Déposer le premier composant de la Route depuis la Palette

Pourquoi et quand exécuter cette tâche
Le premier composant de cet exemple est ajouté à partir de la Palette, c’est un cTimer qui permet de déclencher l’échange de messages.
Pour déposer un composant depuis la Palette, procédez comme suit :

Procédure
1. Saisissez les mots-clés dans le champ de recherche de la Palette et appuyez sur Entrée afin de valider votre recherche.
Les mots-clés peuvent être des noms partiels ou complets de composants, ou une phrase décrivant sa fonctionnalité, si vous ne connaissez pas son nom. Par exemple, **timer** (minuteur en français), ou **generate message exchange** (génère des échanges de messages). La Palette affiche uniquement les familles dans lesquelles le composant peut être trouvé. Si la Palette n’est pas affichée dans le Studio, consultez Modifiez la disposition et les paramètres de la Palette à la page 489.

**Remarque:**

- Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case **Also search from Help when performing a component searching** dans la fenêtre des **Preferences > Palette Settings**. Pour plus d’informations, consultez Préférences de la Palette (Talend > Palette Settings) à la page 511.
- Les composants n’ont pas toujours de lettre comme préfixe dans leur nom. La bonne pratique recommande de spécifier simplement le corps principal lors de la recherche d’un composant par son nom.

2. Sélectionnez le composant à utiliser et, dans l’espace de modélisation graphique, cliquez à l’endroit où vous souhaitez l’ajouter.

**Résultats**

Notez que vous pouvez également déposer une note dans votre Route, de la même manière que pour les composants.
Ajouter le deuxième composant de la Route en saisissant son nom dans l'espace de modélisation graphique

Pourquoi et quand exécuter cette tâche

Le deuxième composant de la Route est ajouté par saisie directe dans l'espace de modélisation graphique et non par glisser-déposer depuis la Palette.


Pour ajouter un composant directement dans l’espace de modélisation graphique, procédez comme suit :

Procédure

1. Dans l’espace de modélisation graphique, cliquez à l’emplacement où vous souhaitez ajouter le composant et saisissez votre (vos) mot(s)-clé(s), à savoir le nom complet ou partiel du composant, ou une phrase décrivant sa fonctionnalité si vous ne connaissez pas son nom. Dans cet exemple, commencez à saisir cset.

Remarque:

- Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case Also search from Help when performing a component searching dans la fenêtre des Preferences > Palette Settings. Pour plus d’informations, consultez Préférences de la Palette (Talend > Palette Settings) à la page 511.
- Les composants n’ont pas toujours de lettre comme préfixe dans leur nom. La bonne pratique recommande de spécifier simplement le corps principal lors de la recherche d’un composant par son nom.

Une liste s’affiche sous le champ de recherche et affiche tous les composants correspondant à votre recherche, dans l’ordre alphabétique.
2. Double-cliquez sur le composant de votre choix pour l’ajouter dans l’espace de modélisation graphique. Par exemple, cliquez sur le cSetBody.

**Ajouter un composant de sortie à la Route en glissant depuis le composant d’entrée**

**Pourquoi et quand exécuter cette tâche**

Vous allez ajouter le troisième composant, un cLog, pour enregistrer les échanges de messages. Vous allez ajouter le composant en glissant depuis le composant cSetBody, qui est le composant d’entrée de celui que vous ajoutez.

**Procédure**

1. Cliquez sur le composant cSetBody pour afficher l’icône o placée dessus.
2. Glissez-déposez l’icône o à l’emplacement où vous souhaitez ajouter un nouveau composant.
   Un champ textuel et une liste de composants s’affichent. La liste des composants affiche tous les composants pouvant être connectés au composant d’entrée.

3. Pour réduire la recherche, saisissez le nom ou une partie du nom du composant, dans le champ de texte, ou un groupe de mots décrivant sa fonctionnalité si vous ne connaissez pas son nom. Dans la liste, double-cliquez sur le composant de votre choix, le cLog dans cet exemple, pour l’ajouter.
à l’espace de modélisation graphique. Le nouveau composant est automatiquement connecté au composant d’entrée cSetBody, à l’aide d’un lien Row > Route.

**Remarque:**
- Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case Also search from Help when performing a component searching dans la fenêtre des Preferences > Palette Settings. Pour plus d’informations, consultez Préférences de la Palette (Talend > Palette Settings) à la page 511.
- Les composants n’ont pas toujours de lettre comme préfixe dans leur nom. La bonne pratique recommande de spécifier simplement le corps principal lors de la recherche d’un composant par son nom.

![Diagramme des composants cSetBody, cLog et cTimer](image)

**Relier les composants de la Route**

Maintenant que les composants sont ajoutés dans l’espace de modélisation graphique, ils doivent être connectés entre eux.

Dans cet exemple, les composants cSetBody et cLog sont déjà reliés, vous devez seulement relier le cTimer au cSetBody.

Afin de relier les composants, utilisez l’une des deux méthodes :

Vous pouvez également déposer des composants au milieu d’une connexion de type Row. Pour plus d’informations, consultez Ajouter un composant entre deux composants reliés à la page 51.

Pour plus d’informations concernant l’utilisation des différents types de connexions, consultez Utilisation des connexions dans une Route à la page 86.

**Créer le lien Route par clic-droit et clic**

**Procédure**

1. Cliquez-droit sur le composant source, le cTimer dans cet exemple.
2. Dans le menu contextuel qui s’ouvre, sélectionnez le type de lien à utiliser pour relier les composants, Row > Route dans cet exemple.
3. Cliquez sur le composant cible pour créer le lien, ici le cSetBody.

![Diagramme des connexions](image)

Notez qu’un cercle noir s’affiche si le composant cible n’est pas compatible avec le lien.
Selon la nature et le rôle des composants que vous souhaitez connecter, différents types de liens sont disponibles. Seules les connexions autorisées sont listées dans le menu contextuel.

**Créer le lien Route par glisser-déposer**

**Procédure**

1. Cliquez sur le composant d’entrée, le **cTimer** dans cet exemple.
2. Lorsque l’icône **O** s’affiche, cliquez dessus et glissez le curseur jusqu’au composant de destination, le **cSetBody** dans cet exemple.

   Un lien **Row > Route** est automatiquement créé entre les deux composants.

Cette méthode comporte moins d’étapes mais ne fonctionne qu’avec les liens **Row > Route**.

**Configurer les composants**

**Pourquoi et quand exécuter cette tâche**

Maintenant que les composants sont reliés entre eux, leurs propriétés doivent être définies.

Renommez les composants afin de mieux identifier leur rôle, par exemple en **Starter** pour le **cTimer**, en **Set_message_body** pour le **cSetBody** et en **LogMessages** pour le **cLog**.

**Remarque:**

Il est recommandé de nommer chaque composant d’une manière unique afin de mieux leur rôle au sein de la Route. Cela est particulièrement utile pour la génération de code de certains composants, par exemple pour le **cSOAP**. Pour plus d’informations, consultez la documentation concernant les **Talend Mediation Components** sur Talend Help Center.

**Procédure**

1. Double-cliquez sur le composant **cTimer** pour ouvrir sa vue **Basic settings**.
2. Dans le champ Repeat, saisissez 1 pour générer une fois l’échange de messages. Laissez les paramètres par défaut pour les autres options.

3. Double-cliquez sur le composant cSetBody pour ouvrir sa vue Basic settings.

4. Sélectionnez Constant dans la liste Language et saisissez "Hello World!" dans le champ Expression, en tant que corps du message.

5. Laissez les valeurs par défaut des paramètres du composant cLog afin d’enregistrer les échanges de messages.

Exécuter la Route

Pourquoi et quand exécuter cette tâche

Une fois les composants configurés, la Route peut être exécutée.

Pour ce faire, procédez comme suit :
**Procédure**

1. Appuyez sur les touches `Ctrl+S` pour sauvegarder la Route.
2. Cliquez sur l’onglet **Run** et cliquez sur le bouton **Run** pour exécuter la Route.

**Résultats**

Les logs de l’échange de messages sont affichés dans la console.

**Utilisation des composants**

Les sections suivantes donnent des informations détaillées sur des sujets variés relatifs à la gestion des composants dans des Jobs et des Routes :

- Ajouter un composant entre deux composants reliés à la page 51
- Définir les propriétés des composants à la page 55
- Chercher un Job utilisant un composant particulier à la page 64
- Configurer les valeurs par défaut dans le schéma d’un composant dans un Job à la page 66
- Utiliser des composants Camel dans une Route à la page 68

**Ajouter un composant entre deux composants reliés**

Lorsque vous créez un Job ou une Route, vous pouvez insérer un composant entre deux composants reliés à l’aide d’un lien **Row** ou **Route**, à condition que le nouveau composant puisse servir de composant intermédiaire entre les deux autres.

Les exemples ci-dessous montrent différentes manières d’insérer un **tMap** entre un **tFileInputDelimited** et un **LogRow** reliés à l’aide d’un lien **Row > Main**.

**Glisser-déposer les composants sur le lien depuis la Palette**

**Procédure**

1. Depuis la **Palette**, localisez et sélectionnez le **tMap**.
2. Glissez-déposez le composant sur le lien **Row**.
S'il vous est demandé de nommer la connexion de sortie du nouveau composant ajouté, ce qui est le cas avec un composant tMap, saisissez un nom et cliquez sur OK pour fermer la boîte de dialogue.

**Remarque:**
Il vous sera peut-être demandé de récupérer le schéma du composant cible. Dans ce cas, cliquez sur OK pour accepter ou cliquez sur No pour refuser.

Le composant est inséré au milieu du lien qui est maintenant divisé en deux liens.

**Ajouter le composant en saisissant son nom sur le lien**

**Procédure**

1. Cliquez sur le lien qui relie les deux composants existants pour le sélectionner.

2. Saisissez le nom du nouveau composant que vous souhaitez ajouter, le tMap dans cet exemple, puis double-cliquez sur le composant dans la liste de suggestions pour l'ajouter au lien.
3. S’il vous est demandé de nommer la connexion de sortie du nouveau composant ajouté, ce qui est le cas avec un composant tMap, saisissez un nom et cliquez sur OK pour fermer la boîte de dialogue.

**Remarque:**
Il vous sera peut-être demandé de récupérer le schéma du composant cible. Dans ce cas, cliquez sur OK pour accepter ou cliquez sur No pour refuser.

Le composant est inséré au milieu du lien qui est maintenant divisé en deux liens.

**Ajouter le composant dans l’espace de modélisation graphique et déplacer le lien existant**

**Procédure**

1. Ajoutez le nouveau composant, le tMap dans cet exemple, dans l’espace de modélisation graphique en faisant un glisser-déposer depuis la Palette ou en saisissant son nom dans l’espace de modélisation graphique.
2. Sélectionnez le lien et déplacez le curseur de votre souris vers la fin du lien jusqu'à ce que celui-ci prenne la forme d'un symbole +.


4. Reliez le composant tMap au tLogRow à l'aide d'un lien Row > Main.
Définir les propriétés des composants

Les propriétés de chacun des composants dans un Job ou une Route permettent de paramétrer l’exécution technique du Job actif.

Les propriétés de chaque composant sont définies dans la vue **Component**, dans l’onglet **Basic settings** pour les propriétés de base et dans l’onglet **Advanced settings** pour les propriétés avancées. La vue **Component** regroupe aussi les vues **View** et **Documentation** qui contiennent des informations secondaires sur le composant sélectionné.

**Onglet Basic settings**

**Pourquoi et quand exécuter cette tâche**

L’onglet **Basic settings** fait partie de la vue **Component**, qui est située en bas de la fenêtre de la perspective **Integration** du Studio Talend.

Les paramètres de base des composants sont différents en fonction de ces exigences fonctionnelles dans le Job ou la Route.

**Remarque:**

Certains composants sont paramétrés à l’aide de code ou de fonctions. Veillez à bien utiliser le code Java dans les propriétés en Java.
Pour les composants **File** (Fichier) et **Database** (Base de données) dans un Job, vous pouvez centraliser leurs propriétés dans les métadonnées stockées dans le dossier **Metadata du Repository**. Dans l'onglet **Basic settings**, vous pouvez paramétrer vous même les propriétés en sélectionnant l’option **Built-In** dans la liste déroulante **Property Type** ou vous pouvez utiliser les propriétés stockées dans le dossier **Metadata Manager du Repository** en sélectionnant l’option **Repository** dans la liste déroulante **Property Type**. Cette dernière option vous permet de gagner du temps si vous utilisez souvent ces paramètres.

Sélectionnez **Repository** dans le champ **Property type** et désignez les métadonnées comportant les paramètres appropriés. Pour plus d’informations, consultez Centraliser des métadonnées de base de données à la page 274.

Vous avez aussi la possibilité de cliquer sur la métadonnée souhaitée sous le nœud Metadata de la vue Repository et de la glisser jusqu’à votre composant déjà présent dans l’espace de modélisation, ainsi ces propriétés seront renseignées automatiquement.

Si vous avez sélectionné le mode **Built-in** et que vous avez paramétré manuellement les propriétés de votre composant, vous avez encore la possibilité de sauvegarder ces propriétés sous forme de métadonnée dans le Repository. Pour cela :

**Procédure**

1. Cliquez sur l’icône de disquette. L’assistant de création de métadonnée correspondant à votre composant s’ouvre.
2. Suivez les étapes de l’assistant. Pour plus d’informations concernant la création de métadonnées, consultez Centraliser des métadonnées de base de données à la page 274.
3. La métadonnée apparaît alors sous le nœud Metadata du Repository.

**Résultats**

Pour tous les composants traitant des flux de données (la plupart des composants), vous pouvez définir un schéma **Talend** afin de décrire et sélectionner les données à traiter. Comme pour les propriétés (**Properties**), ce schéma peut être en local (**Built-in**) ou stocké dans le Repository dans les métadonnées que vous avez créées. Vous trouverez une description détaillée d’un schéma dans la section suivante.

**Paramétrer un schéma built-in dans un Job**

Un schéma est dit ponctuel, et donc défini comme **built-in** dans les paramètres de propriétés lorsqu’il ne sert généralement qu’une fois dans un Job.

**Procédure**

1. Sélectionnez **Built-in** dans la liste **Property Type** dans la vue **Basic settings**.
Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](https://docs.oracle.com/en/java/javase/11/docs/api) (en anglais).

Les types de données **Talend** les plus utilisés sont les suivants :

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données **Object**.

- **List** : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément `xsd:list`.

- **Document** : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

Dans les propriétés de sortie, vous devez définir le schéma de sortie. Pour récupérer le schéma défini dans le schéma d'entrée, cliquez sur **Sync columns** dans la vue **Basic settings**.

**Avertissement:**

Lorsque vous créez une table de base de données, il est recommandé de spécifier une longueur dans le champ **Length** pour toutes les colonnes de type String, Integer ou Long et de renseigner le champ **Precision** pour les colonnes de type Double, Float ou BigDecimal dans le schéma du composant utilisé. Sinon, des erreurs inattendues peuvent survenir.

**Paramétrer un schéma du Repository dans un Job**

Si vous utilisez fréquemment des connexions à des bases de données ou des fichiers spécifiques lorsque vous créez vos Jobs, vous pouvez éviter d'avoir à définir toujours les mêmes propriétés en créant des fichiers de métadonnées que vous pouvez centraliser dans le répertoire **Metadata** du référentiel (**Repository**) de la perspective **Integration**.

**Procédure**

1. Pour rappeler un fichier de métadonnées dans votre Job :
   - Sélectionnez **Repository** dans la liste déroulante **Schema type** et sélectionnez le fichier approprié dans la liste Repository.
• Ou bien, cliquez sur la métadonnée sous le nœud **Metadata** du Repository et glissez-la jusqu’au composant que vous avez préalablement déposé dans l’espace de modélisation.

2. Cliquez sur **Edit Schema** pour vérifier que les données sont appropriées.

Vous pouvez apporter des modifications au schéma **Repository** que vous utilisez pour votre Job. Cependant, notez que le schéma devient alors ponctuel, ses propriétés changent automatiquement en **built-in**, il sera attaché au Job courant.

**Remarque:**
Vous ne pouvez pas modifier le schéma stocké dans le référentiel à partir de cette fenêtre. Pour éditer ce schéma stocké, cliquez-droit sur le nœud **Metadata** et sélectionnez l’option d’édition correspondante (**Edit connection** ou **Edit file**) dans le menu contextuel.

**Utiliser partiellement un schéma du Repository dans un Job**
Lorsque vous utilisez un schéma du référentiel, si vous ne souhaitez pas utiliser toutes les colonnes prédéfinies, vous pouvez sélectionner des colonnes précises, sans modifier le schéma et le passer en mode Built-In.

La procédure suivante décrit l'utilisation partielle d'un schéma du référentiel pour un composant de base de données, d'entrée. Les étapes peuvent varier selon le composant que vous utilisez.
Procédure

1. Cliquez sur le bouton […] à côté du champ **Edit schema**, dans l’onglet **Basic settings**. La boîte de dialogue **Edit parameter using repository** s’ouvre. Par défaut, l’option **View schema** est sélectionnée.

2. Cliquez sur **OK**. La boîte de dialogue **Schema** s’ouvre et affiche toutes les colonnes du schéma. La case **Used Column** devant chaque colonne indique si la colonne est utilisée.


4. Cliquez sur **OK**. Un message s’ouvre et vous demande de cliquer sur le bouton **Guess Query**.

**Remarque:**
L’opération Guess Query est nécessaire uniquement pour les métadonnées de base de données.
5. Cliquez sur **OK** pour fermer la boîte de dialogue. La fenêtre **Propagate** s'ouvre. Cliquez sur **Yes** pour propager les modifications.

6. Dans la zone **Basic settings**, cliquez sur **Guess Query**. Le nom des colonnes sélectionnées s’affiche dans la zone **Query** comme attendu.
Afficher un champ de manière dynamique (Ctrl+Espace)

Pourquoi et quand exécuter cette tâche
Dans tous les champs de la vue Properties de vos composants, vous pouvez accéder à la liste des variables globales et de contexte et afficher de manière dynamique les valeurs contenues dans chaque champ à l’aide du raccourci clavier Ctrl+Barre d’espace.

Procédure
1. Placez le curseur de votre souris dans n’importe quel champ de l’onglet Component.
2. Appuyez sur Ctrl+Barre d’espace pour accéder à la liste des variables.

Les paramètres affichés peuvent être : des messages d’erreur, le nombre de lignes traitées, etc. La liste varie en fonction des composants sélectionnés ou du contexte sur lequel vous travaillez.
Pour plus d’informations, consultez Utiliser les contextes et les variables à la page 88.

Onglet Advanced settings
Certains composants, notamment les composants File et Database dans les Jobs, offrent de nombreuses fonctions avancées.
Le contenu de l’onglet **Advanced settings** change en fonction du composant sélectionné.

Généralement, cet onglet regroupe les paramètres qui ne sont pas requis pour une utilisation de base ou habituelle du composant, mais ils sont requis pour les utilisations avancées.

**Mesurer les flux de données**

L’onglet **Advanced settings** propose aussi les fonctionnalités de statistiques avec l’option **tStatCatcher**. Pour plus d’informations concernant les fonctionnalités Stats & Log, consultez Automatiser l’utilisation des statistiques & logs à la page 116.

**Onglet Dynamic settings des composants dans un Job**

**Pourquoi et quand exécuter cette tâche**

Les vues **Basic settings** et **Advanced settings** de tous les composants comportent divers cases à cocher et listes déroulantes permettant de paramétrer chaque composant. D’une manière générale, les valeurs de ces types de paramètres ne peuvent être éditées qu’au moment de la création.

L’onglet **Dynamic settings**, dans la vue **Component**, vous permet de personnaliser ces paramètres sous la forme de code ou de variable.

Cette fonctionnalité vous permet par exemple de configurer ces paramètres comme des variables qui deviendront donc dépendant des contextes, alors qu’ils ne sont pas censés être par défaut.

Un autre avantage de cette fonctionnalité réside dans le fait que vous êtes maintenant capable de changer les paramètres de contexte au moment de l’exécution. Cela est très utile lorsque vous exportez votre Job script afin de le déployer sur un serveur d’exécution par exemple.

**Pour personnaliser ces types de paramètres, notamment les variables de contexte, suivez les étapes qui suivent :**

**Procédure**

1. Sélectionnez le composant dont les onglets Basic et Advanced settings contiennent les paramètres que vous souhaitez définir comme variable.
2. Cliquez sur l’onglet **Dynamic settings**.
3. Cliquez sur le bouton [+] pour créer une nouvelle ligne de paramètres dans le tableau.
4. Cliquez dans la cellule **Name** du paramètre pour afficher la liste des paramètres disponibles. Par exemple : **Print operations**

**Remarque:** Dans la cellule Code, vous pouvez entrer une variable de contexte mais aussi un bout de code Java.

**Résultats**

Les listes ou cases à cocher correspondantes deviennent donc indisponibles et sont surlignées en jaune dans les onglets Basic settings ou Advanced settings.

![tContextLoad_1](image)

**Remarque:** Si vous souhaitez configurer un paramètre comme variable de contexte, assurez-vous d’avoir bien créé la variable dans la vue Contexts. Pour plus d’informations concernant la définition des variables de contexte, consultez Définir les variables de contexte dans la vue Contexts à la page 89.

Pour comprendre comment définir un paramètre dynamique, consultez Définir des groupes de contextes.

**Onglet View**

L’onglet View de la fenêtre [Component] vous permet de changer le format d’affichage par défaut d’un composant dans l’éditeur graphique.

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label format</td>
<td>Libellé libre qui s’affiche au niveau de l’éditeur graphique. Des variables peuvent être utilisées pour retrouver et afficher des valeurs d’autres champs. L’aide contextuelle de ce champ fournit généralement la variable correspondante où la valeur du champ est stockée.</td>
</tr>
<tr>
<td>Hint format</td>
<td>Bulle contextuelle apparaissant lorsque vous passez la souris au dessus du composant.</td>
</tr>
<tr>
<td>Show hint</td>
<td>Cochez cette case pour activer la fonctionnalité d’indication contextuelle.</td>
</tr>
</tbody>
</table>

Vous pouvez personnaliser les textes des **Label** et **Hint** à l’aide des balises HTML suivantes :

- **Gras** : `<b> LibelléOuBulle </b>`
- **Italique** : `<i> LibelléOuBulle </i>`
- **Retour chariot** : `LibelléOuBulle `<br> ContinueLigneSuiv`
- **Couleur** : `<Font color='#RGBcolor'> LibelléOuBulle </Font>`
Pour changer vos préférences de l'onglet View, cliquez sur Window > Preferences > Talend > Designer.

**Onglet Documentation**

N'hésitez pas à ajouter tout commentaire ou morceau de texte que vous jugerez utile dans le champ Comment.

Dans l'onglet **Documentation**, vous pouvez ajouter votre texte dans le champ **Comment**. Ensuite, cochez la case **Show Information**, et une icône d'information s'affiche à côté du composant correspondant dans l'espace de modélisation graphique.

Vous pouvez également ajouter le contenu de votre commentaire, ou documentation, dans l'aide contextuelle d'un composant en utilisant la variable associée (\_COMMENT\_). Lorsque vous placez votre curseur sur cette icône, le texte saisi dans le champ **Comment** s'affiche dans une bulle d'aide.

Pour une utilisation avancée de la fonction Documentation, il est préférable d'utiliser la partie **Documentation** du référentiel, qui vous permet de conserver et réutiliser tout type de documentation de façon centralisée.

**Chercher un Job utilisant un composant particulier**

**Pourquoi et quand exécuter cette tâche**

**Remarque:**

Vous devez ouvrir au moins un Job dans votre Studio afin d'afficher la Palette à droite de l'espace de modélisation graphique et lancer la recherche.

À partir de la Palette, vous avez la possibilité de rechercher tous les Jobs utilisant le composant sélectionné. Pour cela :

![Diagramme de composant tFileInputDelimited_1](image.png)
Procédure

1. Dans la Palette, cliquez-droit sur le composant que vous cherchez et sélectionnez l’option **Find Component in Jobs**.

Une barre de progression s’affiche pour vous indiquer l’évolution de la recherche en pourcentage, puis la boîte de dialogue **Find a Job** s’affiche à l’écran et donne la liste de tous les Jobs utilisant ledit composant.
2. Dans la liste des Jobs, sélectionnez celui qui vous intéresse puis cliquez sur OK pour l’ouvrir dans l’espace de modélisation graphique.

Configurer les valeurs par défaut dans le schéma d’un composant dans un Job

Pourquoi et quand exécuter cette tâche

Vous pouvez configurer des valeurs par défaut dans le schéma de certains composants pour remplacer des valeurs nulles récupérées de la source de données.

Remarque:
Pour le moment, seuls les tFileInputDelimited, tFileInputExcel et tFixedFlowInput supportent les valeurs par défaut dans le schéma.


id;firstName;lastName;company;city;phone
1;Michael;Jackson;IBM;Roma;2323
2;Elisa;Black;Microsoft;London;4499
3;Michael;Dujardin;;8872
4;Marie;Dolvina;;6655
5;Jean;Perfide;;3344
6;Emilie;Taldor;Oracle;Madrid;2266
7;Anne-Laure;Paldufier;Apple;;4422
Pour configurer des valeurs par défaut :

**Procédure**

1. Double-cliquez sur le composant d’entrée `tFileInputDelimited` afin d’afficher sa vue `Basic settings`.

Dans cet exemple, les métadonnées du composant d’entrée sont stockées dans le Repository. Pour plus d’informations concernant la création de métadonnées dans le Repository, consultez Centraliser des métadonnées de base de données à la page 274.

2. Cliquez sur le bouton `[...]` à côté du champ `Edit schema` et sélectionnez l’option `Change to built-in property` dans la boîte de dialogue afin d’ouvrir l’éditeur du schéma.

3. Saisissez `Talend` entre guillemets dans le champ `Default` pour la colonne `company`. Saisissez `Paris` entre guillemets dans le champ `Default` pour la colonne `city` et cliquez sur `OK` pour fermer l’éditeur de schéma.
4. Configurez le composant de sortie tLogRow pour afficher les résultats d'exécution comme vous le souhaitez et exécutez le Job.

Dans le flux de données de sortie, les informations manquantes sont complétées selon les valeurs configurées par défaut.

Utiliser des composants Camel dans une Route

De nombreux composants Camel sont supportés mais non inclus directement dans la Palette, comme camel-mina, camel-http4, par exemple. Ces composants Camel requièrent le cMessagingEndpoint pour fonctionner et leurs bibliothèques doivent être ajoutées à la Route. Toutes les versions des composants Camel publiées avant le Studio Talend sont supportées.
Avertissement:
Le processus d'utilisation d'un composant Camel externe dans une Route est différent de l'utilisation d'un composant, externe ou personnalisé, dans un Job. Pour plus d'informations concernant l'utilisation de composants externes ou utilisateur dans un Job, consultez Définir le dossier de composants utilisateur (Talend > Components) à la page 506.

La capture d'écran suivante illustre l'utilisation du composant camel-mina.

Pour ajouter la bibliothèque camel-mina à la Route, vous pouvez utiliser le même composant cMessagingEndpoint et ajouter mina à la liste Dependencies, dans la vue Advanced settings.

Vous pouvez également utiliser un composant cConfig et ajouter la bibliothèque camel-mina à la liste Dependencies du composant cConfig.

**Utiliser les composants tPrejob et tPostjob**

Les composants *tPrejob* et *tPostjob* sont conçus pour déclencher l’exécution de tâches avant et après l’exécution d’un Job plus simple à utiliser. Ces composants sont différents des autres car ils ne traitent en effet pas les données et ne sont pas configurables. L’une des propriétés-clé de ces composants est l’assurance qu’ils seront toujours exécutés même si le Job contenant les données principales échoue. Ils sont donc très utiles pour démarrer et arrêter des actions pour un Job donné.

**Remarque:**

Puisque les composants *tPrejob* et *tPostjob* n’ont pas été créés pour le traitement de données, ils ne peuvent être exécutés en multi thread. Ils sont conçus pour rendre votre Job plus clair.

Pour utiliser les composants *tPrejob* et *tPostjob*, déposez-les simplement dans l’espace de modélisation graphique comme vous le ferez avec d’autres composants et connectez le *tPrejob* à un composant ou sous-Job qui doit effectuer une tâche avant un Job, et le *tPostjob* à un composant ou sous-Job qui doit effectuer une tâche après un Job, à l’aide d’une connexion *Trigger*. Un carré orange affiche les parties pré et post-job qui sont des types différents de sous-Jobs.
Les tâches nécessitant l’utilisation d’un composant `tPrejob` sont par exemple :

- le chargement d’informations de contexte requises pour l’exécution du sous-Job,
- l’ouverture d’une connexion à une base de données,
- la vérification de l’existence d’un fichier.

Les tâches nécessitant l’utilisation d’un composant `tPrejob` sont par exemple :

- la suppression des fichiers temporaires créés durant l’exécution du Job principal,
- la fermeture de la connexion à une base de données ou à un service extérieur,
- toute tâche devant être exécutée même si le Job ou le sous-Job précédent a échoué.

Pour un exemple d’utilisation des composants `tPrejob` et `tPostjob`, consultez Orchestration (Intégration).

**Télécharger/charger des composants de la communauté**

Le Studio Talend vous permet d’accéder à une liste de composants communautaires dans **Talend Exchange** qui sont compatibles avec votre version du Studio Talend. Vous pouvez télécharger ces composants pour les réutiliser dans les Jobs de votre Studio. Depuis le Studio Talend, vous pouvez également charger les composants que vous avez créés vers **Talend Exchange** afin de les partager avec les autres utilisateurs de la Communauté.

**Avertissement:** Assurez-vous que le paramètre `-Dtalend.disable.internet` n’est pas dans le fichier `.ini` du Studio ou défini comme `false`.

Un clic sur le lien **Exchange** dans la barre d’outils du Studio Talend ouvre la vue **Exchange** dans l’espace de travail, dans laquelle vous pouvez voir les listes de :

- composants compatibles sur **Talend Exchange** que vous pouvez télécharger et installer,
- composants que vous avez téléchargés et installés dans une version précédente du Studio Talend, mais pas encore dans votre version actuelle du Studio,
- composants que vous avez créés et chargés sur **Talend Exchange**, afin de les partager avec les autres utilisateurs de la Communauté **Talend**.
Notez que l’approche expliquée dans cette section doit être utilisée uniquement pour les composants susmentionnés.

**Remarque:**

- Avant de pouvoir télécharger des composants de ou charger vos propres composants vers la Communauté, vous devez vous connecter à **Talend Exchange** depuis votre Studio. Si vous ne vous connectez pas à **Talend Exchange** au lancement du Studio, vous pouvez toujours vous y connecter depuis la page de préférences **Talend Exchange**. Pour plus d’informations, consultez Préférence Exchange (Talend > Exchange) à la page 509.
- Les composants disponibles au téléchargement dans la vue ne sont pas validés par Talend. Il est possible que vous soyez confronté(e) à des erreurs de chargement lors de l’installation de certains composants à partir de Talend Exchange, que le nom d’un composant soit différent dans la Palette de celui dans la vue **Exchange**, ou que vous ne trouviez pas certains composants dans votre Palette après une installation apparentemment réussie.

**Installer des composants de la communauté à partir de Talend Exchange**

**Pourquoi et quand exécuter cette tâche**

Pour installer des composants de la communauté à partir de **Talend Exchange** dans la Palette de votre Studio Talend :

**Procédure**

1. Cliquez sur le lien **Exchange** dans la barre d’outils du Studio Talend pour ouvrir la vue **Exchange** dans l’espace de travail.

2. Dans la vue **Available Extensions**, dans le champ à droite, si nécessaire, saisissez le nom complet d’un composant ou une partie du nom dans le champ et cliquez sur le bouton Refresh afin de trouver rapidement le composant qui vous intéresse.

3. Cliquez sur le lien **view/download** pour afficher la page de téléchargement du composant.

Available Extensions

<table>
<thead>
<tr>
<th>Extension Name</th>
<th>Version</th>
<th>Rating</th>
<th>Author</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook Application Insights Component</td>
<td>0.1</td>
<td>★★★☆☆☆</td>
<td>saburo</td>
<td>view/download</td>
</tr>
<tr>
<td>tFileOutputDelimitedExport</td>
<td>1.0</td>
<td>★★★☆☆☆</td>
<td>Alezis</td>
<td>view/download</td>
</tr>
<tr>
<td>tScriptRules</td>
<td>1.0</td>
<td>★★★☆☆☆</td>
<td>walkerca</td>
<td>view/download</td>
</tr>
<tr>
<td>tWorldBank components Demo</td>
<td>0.2</td>
<td>★★★☆☆☆</td>
<td>saburo</td>
<td>view/download</td>
</tr>
<tr>
<td>pUpdateMailOffer</td>
<td>V02</td>
<td>★★★☆☆☆</td>
<td>PayZen</td>
<td>view/download</td>
</tr>
<tr>
<td>pCreateMailOffer</td>
<td>V02</td>
<td>★★★☆☆☆</td>
<td>PayZen</td>
<td>view/download</td>
</tr>
<tr>
<td>tDBOutput</td>
<td>1.1</td>
<td>★★★☆☆☆</td>
<td>BiSi</td>
<td>view/download</td>
</tr>
<tr>
<td>tDBInput</td>
<td>1.1</td>
<td>★★★☆☆☆</td>
<td>BiSi</td>
<td>view/download</td>
</tr>
<tr>
<td>pCreatePayment</td>
<td>V06</td>
<td>★★★☆☆☆</td>
<td>PayZen</td>
<td>view/download</td>
</tr>
</tbody>
</table>
4. Lisez les informations relatives au composant, notamment sa description et les commentaires des utilisateurs, ou écrivez votre propre commentaire, et/ou notez le composant, si vous le souhaitez. Pour plus d'informations concernant les remarques et les notes des composants de la Communauté, consultez Commenter et noter un composant de la Communauté à la page 75. Si nécessaire, cliquez sur la flèche pointant vers la gauche afin de retourner à la liste des composants.

5. Cliquez sur le bouton Install à droite de la page de téléchargement des composants pour commencer le téléchargement puis l'installation.

Un indicateur de progression apparaît pour vous montrer l’avancement du téléchargement et de l’installation. Lorsque l’installation est terminée avec succès, la vue Downloaded Extensions s’ouvre et affiche le statut du composant, Installed.
Réinstaller ou mettre à jour des composants de la Communauté

Pourquoi et quand exécuter cette tâche

Dans la vue Exchange, vous pouvez réinstaller des composants déjà téléchargés et installés dans une version précédente de votre Studio Talend ou installer la version mise à jour du Studio Talend ou des composants dans votre Studio.

**Remarque:**

Par défaut, lorsque vous êtes connecté(e) à Talend Exchange, une boîte de dialogue s’ouvre pour vous notifier lorsqu’une mise à jour d’un composant de la communauté installé est disponible. Si vous vérifiez souvent les mises à jour des composants communautaires et que vous ne souhaitez pas que cette boîte de dialogue s’ouvre à nouveau, vous pouvez la désactiver dans les préférences de Talend Exchange. Pour plus d’informations, consultez Préférence Exchange (Talend > Exchange) à la page 509.

Pour réinstaller un composant de la communauté que vous avez déjà téléchargé ou mettre à jour un composant installé, procédez comme suit :

**Procédure**

1. Dans la vue Exchange, cliquez sur Downloaded Extensions pour afficher une liste des composants déjà téléchargés depuis Talend Exchange.

   Dans la vue Downloaded Extensions, les composants que vous avez installés dans votre version précédente du Studio Talend mais pas encore dans votre Studio actuel affichent un lien Install dans la colonne Install/Update. Les composants ayant des mises à jour disponibles dans Talend Exchange possèdent un lien Update.

2. Cliquez sur le lien Install ou Update du composant qui vous intéresse afin de commencer l’installation.

   Un indicateur de progression apparaît pour vous montrer l’avancement du téléchargement et de l’installation. Lorsque l’installation est terminée avec succès, la vue Downloaded Extensions s’ouvre et affiche le statut du composant, Installed.
Commenter et noter un composant de la Communauté

Pourquoi et quand exécuter cette tâche

Pour commenter et noter un composant de la Communauté :

Procédure

1. Dans la vue Available Extensions, cliquez sur le lien view/download du composant que vous souhaitez commenter ou noter, afin d’ouvrir la page de téléchargement des composants communautaires.

2. Dans la page de téléchargement des composants, cliquez sur le lien write a review pour ouvrir la boîte de dialogue Review the component.


   Après validation par le modérateur de Talend Exchange, votre commentaire est publié sur Talend Exchange et affiché dans la zone User Review de la page de téléchargement des composants.

Charger dans un composant créé Talend Exchange

Pourquoi et quand exécuter cette tâche

Vous pouvez créer vos propres composants afin de les utiliser dans vos Jobs, dans le Studio Talend et les charger dans Talend Exchange pour les partager avec les autres utilisateurs de la Communauté Talend. Pour plus d’informations concernant la création et le déploiement de vos propres composants
À propos des services de données dans le Studio Talend, consultez Définir le dossier de composants utilisateur (Talend > Components) à la page 506.

Pour charger dans Talend Exchange un composant que vous avez créé, procédez comme suit :

**Procédure**

1. Dans la vue Exchange, cliquez sur My Extensions pour ouvrir la vue My Extensions.

2. Cliquez sur le lien Add New Extension dans la partie supérieure droite de la vue pour ouvrir la page de téléchargement du composant.

3. Renseignez les informations requises, notamment le titre du composant, la version initiale, les informations de compatibilité avec le Studio et la description du composant. Renseignez le chemin d’accès au paquetage source dans le champ File puis cliquez sur le bouton Upload Extension.

   Après réussite du chargement, le composant s’affiche dans la vue My Extensions, dans laquelle vous pouvez mettre à jour, modifier et supprimer tout composant chargé dans Talend Exchange.
Gérer les composants chargés dans Talend Exchange

Dans la vue **Exchange**, vous pouvez gérer les composants que vous avez chargés dans **Talend Exchange**, notamment mettre à jour leur version, modifier leurs informations et supprimer des composants depuis **Talend Exchange**.

Pour mettre à jour la version d’un composant, procédez comme suit :

1. Depuis la vue **My Extensions**, cliquez sur l’icône 🔄 dans la colonne **Operation** du composant que vous souhaitez mettre à jour afin d’ouvrir la page de mise à jour du composant.

2. Renseignez la version initiale et la compatibilité des Studios, renseignez le chemin d’accès au paquetage source, dans le champ **File**, puis cliquez sur le bouton **Update Extension**.

   Lorsque le chargement du composant mis à jour est réussi, le composant est remplacé par sa nouvelle version dans **Talend Exchange** et la vue **My Extension** affiche la nouvelle version du composant et sa date de mise à jour.

Pour modifier les informations d’un composant chargé dans **Talend Exchange**, procédez comme suit :

2. Renseignez les informations de compatibilité du Studio ainsi que la description du composant, puis cliquez sur le bouton Modify Extension pour mettre à jour les informations du composant dans Talend Exchange.

Pour supprimer un composant que vous avez chargé dans Talend Exchange, cliquez sur l’icône du composant, dans la vue My Extensions. Le composant est supprimé de Talend Exchange et ne s’affiche plus dans la liste des composants de la vue My Extensions.

Utilisation des connexions dans un Job

Dans le Studio Talend, un Job ou un sous-Job se compose d’un groupe de composants logiquement reliés les uns aux autres par des connexions. Utilisez ces connexions afin de définir la manière dont sont coordonnés les composants. Cette section décrit les différents types de connexions et leurs paramètres associés.

Types de connexions

Il y a de nombreux types de connexions qui définissent les données à traiter, la sortie des données, ou bien la séquence logique du Job.

Cliquez-droit sur un composant dans l’espace de modélisation graphique afin d’afficher un menu contextuel listant toutes les connexions disponibles pour le composant sélectionné.

Les sections suivantes décrivent tous les types de connexions disponibles.

Connexion de type Row

La connexion de type Row manipule les données elles-mêmes. Les connexions de type Row sont Main, Lookup, Reject, Output, Uniques/Duplicates ou Combine selon la nature du flux de données traité.
Principal

La connexion Row de type Main est la connexion la plus courante. Elle transmet les flux de données d’un composant à l’autre, en faisant un boucle sur chacune des lignes pour lire ou extraire les données selon la définition des propriétés du composant.

Les données transférées à travers les connexions de type Row sont caractérisées par une définition du schéma qui décrit la structure des données dans le fichier d’entrée.

**Remarque:**

Vous ne pouvez pas connecter deux composants d’entrée à l’aide d’une connexion de type **Row > Main**. Une seule connexion entrante de type **Row** est possible par composant. Vous ne pouvez pas lier deux fois le même composant cible à l’aide d’un lien **Row > Main**. La seconde connexion **Row** se transforme en connexion **Lookup** (de référence) automatiquement.

Cliquez-droit sur le composant d’entrée et sélectionnez **Row > Main** dans la liste des connexions.

Sinon, cliquez sur le composant pour le sélectionner, cliquez-droit dessus, ou cliquez sur l’icône O s’affichant sur le côté du composant et déplacez votre curseur vers le composant de destination. Une connexion de type **Row > Main** sera automatiquement créée.

Pour en savoir plus sur les cas spécifiques de connexions **Row** multiples, consultez [Job à entrées/sorties multiples](#) à la page 81.

**Lookup**

La connexion de type Lookup est une connexion Row reliant un composant d’un flux secondaire à un composant d’un flux principal (ce composant doit être capable de recevoir plus d’un flux d’entrée). Cette connexion est uniquement disponible dans le cas d’utilisation de flux multiples.
Vous pouvez transformer une connexion de type **Lookup** en connexion de type Main row, et inversement vous pouvez changer une connexion Lookup en Main row. Pour ce faire, cliquez-droit et sélectionnez **Set this connection as Main**.

Pour plus d’informations, consultez **Job à entrées/sorties multiples** à la page 81.

**Filter**

La connexion de type **Filter** relie le composant spécifique tFilterRow à un composant de sortie. Cette connexion Row regroupe les données répondant aux critères du filtre. Ce composant particulier permet aussi une connexion de rejet (**Reject**) pour traiter le flux de données ne répondant pas aux critères.

**Rejects**

La connexion de type **Rejects** est une connexion reliant un composant de traitement à un composant de sortie. Cette connexion Row regroupe les données NE répondant PAS aux critères du filtre ou qui ne sont pas compatibles avec la sortie attendue. Cette connexion vous permet d’isoler les données ne pouvant pas être traitées pour différentes raisons (type de données inapproprié, valeur Null non définie, etc.). Pour certains composants, cette connexion est activée lorsque l’option **Die on error** est désactivée.

**ErrorReject**

La connexion ErrorReject est une connexion Row reliant un composant tMap à un composant de sortie. Cette connexion est activée lorsque vous décochez la case **Die on error** dans l’éditeur du tMap et elle rassemble les données impossibles à traiter (type de données inapproprié, valeur nulle non définie, format de date erroné, etc.).

Pour plus d’informations, consultez **Gestion des erreurs** à la page 239.

**Output**:

La connexion de type **Output** est une connexion Row, traitant de la donnée, et qui relie un composant tMap à un ou plusieurs composants de sortie. Les sorties d’un Job pouvant être multiples, une boîte de dialogue s’affiche pour vous demander de nommer chacune des connexions.
Remarque:
Dans la liste des connexions output suggérées, vous retrouvez ainsi les connexions supprimées. Cette fonctionnalité vous permet de ne pas avoir à ressaisir toutes les informations de propriétés.

Pour plus d’informations, consultez Job à entrées/sorties multiples à la page 81.

Uniques/Duplicates

Ces connexions relient un composant tUniqRow à des composants de sortie.

La connexion de type Uniques regroupe les premières lignes rencontrées dans un flux d’entrée. Ce flux de données uniques est ensuite dirigé vers le composant de sortie approprié ou vers un autre sous-job de traitement.

La connexion de type Duplicates regroupe les doublons possibles des premières lignes rencontrées. Ce flux de rejet est dirigé vers le composant adéquat, pour être analysé par exemple.

Job à entrées/sorties multiples

Certains composants permettent de manipuler des données de sources multiples et/ou vers des sorties multiples. Le plus souvent se sont des composants de traitement de données, notamment le tMap.

Si vous avez besoin d’effectuer une jointure ou des transformations sur un flux, il est préférable d’utiliser le composant tMap, qui est conçu pour ce type de besoin.

Pour plus d’informations concernant le mapping et la transformation de données, consultez Interfaces de mapping à la page 215.

Combine

La connexion de type Combine relie un composant CombinedSQL à un autre.

Lorsque vous cliquez-droit sur le composant CombinedSQL pour le relier à un autre, sélectionnez Row > Combine.

Connexion de type Iterate

La connexion de type Iterate sert à faire une boucle sur plusieurs fichiers d'un répertoire donné, sur les lignes d'un fichier ou sur les entrées d'une base de données.

Un composant ne peut être la cible que d’une seule connexion d’itération, Iterate. La connexion Iterate est généralement utilisée sur le composant de début (Start) d’un flux (dans un sous-job).

Certains composants ne peuvent fonctionner avec d'autres composants qu'à l'aide d'une connexion d'itération, c'est le cas du composant tFilelist. Pour plus d’informations concernant la configuration d’une connexion Iterate, consultez Paramètres d’une connexion Iterate à la page 84.

Remarque: À la différence des autres types de connexions, le nom de la connexion Iterate est en lecture seule.

Avertissement: Notez que globalMap n’est pas sécurisé. Soyez prudent(e) lorsque vous utilisez globalMap.put("key","value") et globalMap.get("key") pour créer vos propres variables globales et que vous récupérez leurs valeurs dans vos Jobs, notamment après un lien Iterate, avec l’option d’exécution parallèle activée.
**Connexion de type Trigger**

Les connexions de déclenchement, Trigger, aident à définir le processus de traitement. Les connexions de type Trigger ne servent pas au transfert de données.

La connexion utilisée met en place une relation de dépendance entre un Job principal et ses sous-jobs. De cette manière, l’un ou l’autre des Jobs est déclenché en fonction de la nature de cette connexion de déclenchement.

Les connexions de déclenchement (Trigger) sont réparties en deux catégories :

- déclenchements de sous-job : **On Subjob Ok**, **On Subjob Error** et **Run if**,
- déclenchements de composant : **On Component Ok**, **On Component Error** et **Run if**.

**OnSubjobOK** (précédemment **Then Run**): Cette connexion est utilisée pour déclencher le sous-job qui suit à condition que le sous-job principal se soit exécuté sans erreur. Cette connexion s'utilise uniquement avec le composant de début (Start) de votre Job.

Ces connexions sont utilisées pour orchestrer et vérifier plus facilement les sous-jobs composant le Job ou traiter les erreurs qui n'étaient pas prévues.

**OnSubjobError**: Cette connexion est utilisé pour déclencher le sous-job qui suit lorsque le premier sous-job (principal) ne s’est pas déroulé correctement. Ce sous-job "on error" vous permet d’identifier les éventuels goulots d’étranglement ou de corriger l’erreur si cela est possible.

**OnComponentOK** et **OnComponentError** sont des déclencheurs de composants. Ils peuvent déclencher n’importe quel composant source dans un sous-job.

**OnComponentOK** ne déclenche l’exécution du composant cible qu’une fois l’exécution du composant source terminée correctement. Il peut servir à déclencher un sous-job de notification par exemple.
**OnComponentError** déclenche le sous-job ou composant cible lorsqu’une erreur a été rencontrée dans le processus initial.

**Run if** déclenche un sous-job ou un composant si les conditions définies sont réunies. Pour plus d’informations concernant la configuratoin d’une connexion **Run if**, consultez [Paramètres d’une connexion Run if](#) à la page 85.

Pour plus d’informations concernant la configuration d’une connexion **Trigger**, consultez [Paramètres d’une connexion Trigger](#) à la page 85.

**Connexion de type Link**

La connexion **Link** n’est utilisée qu’avec les composants ELT. Ces connexions transmettent les informations de schémas au composant de mapping ELT afin d’utiliser ces informations dans la construction des requêtes de base de données spécifiques.

La connexion **Link** ne transmet donc pas de données en tant que telles, mais simplement les métadonnées des tables faisant l’objet de la requête.

Pour sélectionner une connexion **Link**, cliquez-droit sur un composant ELT, puis cliquez sur **Link > New Output**.

**Avertissement:**

Le nom que vous affectez à la connexion **Link** doit impérativement reprendre le nom de la table à requêter.

En effet, le nom de la connexion étant utilisé dans la formulation de la requête SQL généré par le composant ELT tMap, un même nom ne devrait jamais être utilisé deux fois.

**Définir les paramètres des connexions**

Vous pouvez afficher les propriétés d’une connexion en la sélectionnant et en cliquant sur l’onglet **Component**, ou bien en cliquant-droit sur la connexion et en sélectionnant **Settings** dans le menu contextuel. Cette section résume la configuration des propriétés des connexions.

**Paramètres d’une connexion Row**

**Pourquoi et quand exécuter cette tâche**

L’onglet **Basic settings** de la vue **Component** de la connexion affiche le schéma du flux de données géré par la connexion. Vous pouvez modifier le schéma en cliquant sur le bouton **Edit schema**. Pour plus d’informations, consultez [Paramétrer un schéma built-in dans un Job](#) à la page 56.
L’onglet **Advanced settings** vous permet de moniter le flux de données d’une connexion dans un Job, sans avoir à utiliser un composant *tFlowMeter*.

Pour moniter les données dans la connexion, renseignez les paramètres suivants dans l’onglet **Advanced settings** :

**Procédure**

1. Cochez la case *Monitor this connection*.
2. Dans la liste **Mode**, sélectionnez *Absolute* pour enregistrer dans le log le nombre de lignes passant dans la connexion, ou *Relative* pour évaluer le ratio (%) du nombre de lignes passé dans cette connexion par rapport à une connexion de référence. Si vous sélectionnez *Relative*, vous devez également sélectionner une connexion de référence dans la liste **Connections List**.
3. Cliquez sur le bouton [+] pour ajouter une ligne au tableau **Thresholds** et définir une plage pour le nombre lignes à enregistrer dans le log.

**Paramètres d’une connexion Iterate**

Lorsque vous configurez une connexion Iterate, vous activez les itérations parallèles. Pour plus d’informations, consultez *Lancer des itérations parallèles pour lire des données* à la page 214.
Services de données

Paramètres d’une connexion Trigger
Paramètres d’une connexion Run if

Pourquoi et quand exécuter cette tâche
Dans la vue Basic settings de la connexion Run if, vous pouvez configurer la condition du sous-job en Java.

Vous pouvez utiliser des variables dans votre condition. Le raccourci clavier Ctrl+Barre d’espace vous donne accès à toutes les variables globales et de contexte. Pour plus d’informations, consultez Utiliser des variables dans un Job ou une Route à la page 102.

Avertissement:
Lorsque vous ajoutez un commentaire après la condition, assurez-vous de l’entourer de /* et */, même si le commentaire ne se compose que d’une seule ligne.

Dans l’exemple suivant, un message est déclenché si le fichier d’entrée contient 0 ligne de données.

Procédure
1. Créez un Job et déposez les trois composants suivants dans l’espace de modélisation graphique : un tFileInputDelimited, un tLogRow, et un tMsgBox.
2. Reliez les composants comme suit :
   - Cliquez-droit sur le composant tFileInputDelimited, sélectionnez Row > Main dans le menu contextuel et cliquez sur le composant tLogRow.
   - Cliquez-droit sur le composant tFileInputDelimited, sélectionnez Trigger > Run if dans le menu contextuel et cliquez sur le tMsgBox.
3. Configurez le tFileInputDelimited afin qu’il lise un fichier contenant zéro ligne de données.
4. Sélectionnez la connexion Run if entre le tFileInputDelimited et le tMsgBox et cliquez sur sa vue Component. Dans le champ Condition, dans l'onglet Basic settings, appuyez sur les touches Ctrl +Espace pour accéder à la liste des variables et sélectionnez la variable NB_LINE du composant tFileInputDelimited. Modifiez la condition comme suit :

```java
((Integer)globalMap.get("tFileInputDelimited_1_NB_LINE"))==0
```

5. Cliquez sur la vue Component du composant tMsgBox et saisissez un message, "No data is read from the file" par exemple, dans le champ Message.

6. Sauvegardez et exécutez le Job. Vous devriez voir le message défini dans le composant tMsgBox.

**Utilisation des connexions dans une Route**

Dans le Studio Talend, une Route se compose d’un groupe de composants logiquement reliés les uns aux autres par des connexions. Cette section décrit les types de connexions disponibles pour les Routes ainsi que les paramètres correspondants.

**Types de connexions dans les Routes**

Il y a de nombreux types de connexions qui définissent comment router vos messages au sein d’une Route Camel.

Cliquez-droit sur un composant dans l’espace de modélisation graphique afin d’afficher un menu contextuel listant toutes les connexions disponibles pour le composant sélectionné.

Les sections suivantes décrivent tous les types de connexions disponibles.

**Connexions de type Row dans les Routes**

Une connexion Row gère les messages à router. Les connexions Row sont try, catch, finally ou route selon le composant sélectionné.

**route**

Ce type de connexion est le plus courant. Il passe des messages d’un endpoint à un autre, ou d’un endpoint à un processeur et d’un processeur à un autre endpoint.

**try**

Ce lien relie spécifiquement un composant cTry à un composant récepteur, afin de gérer les erreurs dans une partie de votre Route.

Pour isoler la partie de votre Route susceptible de générer une erreur, vous pouvez la mettre dans un bloc Try via le composant cTry et son lien try. Une fois isolée, si la Route génère une erreur, l’erreur est envoyée au gestionnaire d’erreurs, sauf si un bloc Catch se trouve juste après le bloc Try. Pour plus d’informations, consultez la section catch ci-dessous.

**catch**

Le lien catch peut être utilisé uniquement avec un composant cTry et si un lien try a déjà été utilisé pour isoler une partie d’une Route susceptible de générer une erreur.

Le lien catch capture les erreurs générées par la Route mise dans un bloc Try et vous permet de les gérer, si nécessaire et de continuer la Route, si possible.

**finally**

Le lien finally peut être utilisé uniquement avec un composant cTry et si un lien try a déjà été utilisé pour isoler une partie d’une Route susceptible de générer une erreur.
Le lien **finally** vous permet d’exécuter des instructions finales sans prendre en compte les problèmes pouvant survenir dans les blocs et/ou Catch, pour fermer une connexion à une base de données, par exemple.

**Connexion de type Trigger**

Les connexions Trigger définissent des canaux spécifiques dans lesquels router des messages selon certaines conditions spécifiques.

Les messages sont filtrés et routés vers les Routes spécifiques selon les conditions définies.

**when**

Le lien **when** relie spécifiquement un composant **cMessageRouter** à des composants récepteurs, pour filtrer et router des messages dans une ou plusieurs Routes de sortie, selon des conditions définies.

Ces conditions peuvent être définies dans les paramètres **Connection** de chaque lien **when** créé dans votre Route. Les messages ne répondant pas aux conditions définies peuvent être récupérés via un lien **otherwise**. Pour plus d’informations, consultez la section **otherwise** ci-dessous.

**otherwise**

Le lien **otherwise** peut être utilisé uniquement avec un composant **cMessageRouter** et si au moins un lien **when** a déjà été utilisé pour filtrer et router des messages.

Le lien **otherwise** récupère tous les messages ne répondant pas aux conditions définies dans les liens **when**.

**Remarque:**

Il n’est pas recommandé d’effectuer de gestion de message après un lien **when** ou **otherwise**. Utilisez toujours un endpoint Mock/Direct pour les remplacer et créez une nouvelle Route pour gérer les messages.

**Définir les paramètres d’une connexion dans une Route**

Vous pouvez afficher les propriétés d’une connexion en la sélectionnant et en cliquant sur l’onglet **Component**, ou bien en cliquant-droit sur la connexion et en sélectionnant **Settings** dans le menu contextuel. Cette section résume la configuration des propriétés des connexions.

**Paramètres d’une connexion Catch**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez configurer un lien **Catch** pour capturer l’exception retournée par le lien **try** du composant **cTry**.

**Procédure**

1. Sélectionnez le lien **Catch** de votre Route pour afficher sa vue **Basic settings** dans l’onglet **Components**.
2. Dans le champ **Exceptions**, saisissez le nom de la classe de l’exception susceptible de se produire durant le routage **try**. Si vous saisissez **java.lang.Exception.class**, la mère de toutes les exceptions, toutes les exceptions seront capturées. Vous pouvez essayer de personnaliser l’exception générée en spécifiant une classe d’exception plus précise.
Résultats

Lors de l’exécution de votre Route, si une exception se produit dans le routage suivant un lien `try`, elle sera capturée par le lien `catch` qui correspond le mieux à l’exception. Pour plus d’informations, consultez Connexion de type Row à la page 78.

Paramètres d’une connexion When

Pourquoi et quand exécuter cette tâche

Vous pouvez configurer un lien When pour filtrer les messages selon une condition et router ces messages filtrés :

Procédure

1. Sélectionnez le lien When dans votre Route afin d’afficher sa vue Basic settings dans l’onglet Components.
2. Dans la liste Type, sélectionnez le type de condition que vous souhaitez utiliser dans votre Route.
3. Dans le champ Condition, saisissez la condition que vous souhaitez utiliser pour filtrer vos messages.

Résultats

Lors de l’exécution de votre Route, les messages répondant à la condition définie dans le lien when seront passés au composant suivant et les messages ne correspondant pas peuvent être perdus ou récupérés par un lien otherwise. Pour plus d’informations, consultez Paramètres d’une connexion Catch à la page 87. Pour un scénario utilisant ces liens, consultez Composants Talend Talend Help Center (https://help.talend.com).

Utiliser les contextes et les variables

Les variables représentent des valeurs qui changent tout au long de l’exécution d’un programme.
Une variable globale est une variable système à laquelle vous pouvez accéder par tout module ou toute fonction. Elle conserve sa valeur même après la fin de l’exécution du programme ou de la fonction.

**Avertissement:** Notez que globalMap n’est pas sécurisé. Soyez prudent(e) lorsque vous utilisez `globalMap.put("key", "value")` et `globalMap.get("key")` pour créer vos propres variables globales et que vous récupérez leurs valeurs dans vos Jobs, notamment après un lien *Iterate*, avec l’option d’exécution parallèle activée.

Une variable de contexte est une variable définie par l’utilisateur pour un contexte particulier. Il est possible que vous souhaitiez gérer différemment les types d’exécution de votre Job ou Route selon les conditions d’utilisation, connues comme contextes (*Prod* et *Test* dans l’exemple ci-dessous). Par exemple, vous pouvez intégrer dans votre cycle de validation plusieurs phases de test avant que votre Job ou Route soit mis en production.

Un contexte est caractérisé par des paramètres. Ces derniers sont généralement des variables sensibles au contexte qui seront par la suite ajoutées à la liste des variables disponibles à l’aide du raccourci clavier *Ctrl+Espace* dans les propriétés du composant dans la vue *Component*.

Le Studio Talend vous offre la possibilité de créer plusieurs contextes d’ensembles de données. De plus, vous pouvez créer des contextes ponctuels à partir de l’onglet Contexts d’un Job ou d’une Route, ou vous pouvez centraliser les contextes sous le nœud *Contexts* de la vue *Repository*, afin de les réutiliser dans des Jobs ou des Routes.

Dans un Job, vous pouvez définir les valeurs de vos variables de contexte lors de leur création, ou charger dynamiquement vos paramètres de contexte, soit explicitement à l’aide du composant *tContextLoad*, soit implicitement à l’aide de la fonctionnalité de chargement implicite de contextes (*Implicit Context Load*), lorsque les Jobs sont exécutés.

Cette section décrit comment créer des contextes et des variables ainsi que définir les valeurs des paramètres de contexte.

Pour un exemple de chargement dynamique des paramètres de contexte à l’aide du composant *tContextLoad*, consultez *Contextes*.

Pour un exemple de chargement dynamique de paramètres de contexte à l’aide de la fonctionnalité de chargement implicite des contextes (*Implicit Context Load*), consultez Exemples de Jobs d’intégration de données.

**Définir les variables de contexte pour un Job ou une Route**

Vous pouvez définir les variables de contexte pour un Job ou une Route par deux moyens :

- Dans la vue *Contexts* du Job ou de la Route.
- Avec la touche *F5* à partir de la vue *Component* d’un composant.

**Définir les variables de contexte dans la vue Contexts**

La vue *Contexts* se situe parmi les onglets de configuration sous l’espace de modélisation.

La vue *Contexts* montre toutes les variables qui ont été définies dans le Job courant ou la Route courante, ainsi que les variables de contexte importées dans le Job courant.
À partir de cette vue, vous pouvez gérer vos variables built-in (local) :

- Créer et gérer les contextes built-in.
- Créer, éditer et supprimer les variables built-in.
- Réorganiser les variables de contexte.
- Ajouter les variables de contexte built-in dans le Repository.
- Importer les variables à partir d’une source contexte du Repository.
- Éditer les variables de contexte stockées dans le Repository et mettre à jour les changements dans le Repository.
- Supprimer les variables stockées dans le Repository importées du Job courant ou de la Route courante.

L’exemple suivant montre comment définir deux contextes appelés Prod et Test ainsi qu’une série de variables (host, port, database, username, password et table_name) sous ces deux contextes pour un Job.

Définir les contextes

**Procédure**

1. Ouvrez le Job dans l’espace de modélisation et rendez-vous dans la vue **Contexts**.
   Si la vue **Contexts** ne s’affiche pas, sélectionnez **Window > Show view > TalendContexts** pour ouvrir la vue **Contexts** dans la perspective **Integration**.
2. Cliquez sur le bouton [+] dans le coin supérieur droit.
   La boîte de dialogue **Configure Contexts** s’ouvre et un contexte nommé **Default** est créé par défaut.
3. Sélectionnez le contexte par défaut et cliquez sur **Edit** pour le renommer, **Prod** dans cet exemple. Cliquez sur **OK**.
4. Dans la boîte de dialogue ouverte, cliquez sur **New...** et saisissez **Test** dans la boîte de dialogue **New Context**, puis cliquez sur **OK**.
5. Cochez la case à côté du contexte que vous souhaitez configurer comme contexte par défaut.
   Vous pouvez également configurer le contexte par défaut en sélectionnant le nom du contexte dans la liste **Default context environment**, dans la vue **Contexts**.
   Si nécessaire, déplacez un contexte vers le haut ou vers le bas en le sélectionnant et en cliquant sur le bouton **Up** ou **Down**.
Exemple
Dans cet exemple, définissez Test comme contexte par défaut et déplacez-le vers le haut.

6. Cliquez sur OK pour valider votre définition de contexte et fermer la boîte de dialogue. Les nouveaux contextes s’affichent dans la table des variables de contexte, dans la vue Contexts.
7. Répétez les étapes ci-dessus pour ajouter autant de contextes que nécessaire.
   Si vous ne souhaitez pas définir les valeur de chaque nouveau contexte de zéro, vous pouvez créer le premier contexte et définir ses valeurs, car, lorsque vous créez un nouveau contexte, tous les paramètres du contexte par défaut sont copiés dans le nouveau contexte. Vous pouvez ensuite modifier les valeurs du nouveau contexte selon vos besoins.

Définir les variables

Procédure
1. Cliquez sur le bouton [+] au bas de la vue Contexts pour ajouter des lignes à la table.

2. Cliquez dans le champ Name et saisissez le nom de la variable que vous créez.
   Nommez la première variable host dans cet exemple.
3. Dans la liste Type, sélectionnez le type de la variable.
4. Si nécessaire, cliquez dans le champ Comment et saisissez un commentaire décrivant la variable.
5. Cliquez dans le champ Value et saisissez la valeur de la variable sous chaque contexte.
Selon le type de variable, le champ **Value** différera légèrement lorsque vous cliquerez dedans et fonctionnera de manière différente :

<table>
<thead>
<tr>
<th>Type</th>
<th>Champ de valeur</th>
<th>Valeur par défaut</th>
</tr>
</thead>
<tbody>
<tr>
<td>String (type par défaut)</td>
<td>Champ de texte modifiable</td>
<td>Null</td>
</tr>
<tr>
<td>Boolean</td>
<td>Liste déroulante avec deux options : true et false</td>
<td></td>
</tr>
<tr>
<td>Character, Double, Integer, Long, Short, Object, BigDecimal</td>
<td>Champ de texte éditable</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Champ de texte modifiable, avec un bouton pour ouvrir la boîte de dialogue <em>Select Date &amp; Time</em>.</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Champ de texte éditable, avec un bouton pour ouvrir pour la boîte de dialogue <em>Open</em> pour la sélection de fichiers.</td>
<td></td>
</tr>
<tr>
<td>Directory</td>
<td>Champ de texte modifiable, avec un bouton pour ouvrir la boîte de dialogue <em>Browse for Folder</em> pour la sélection de dossiers.</td>
<td></td>
</tr>
<tr>
<td>List of Value</td>
<td>Champ de texte éditable, avec un bouton pour ouvrir la boîte de dialogue <em>Configure Values</em> pour la création et la configuration de listes.</td>
<td>Vide</td>
</tr>
<tr>
<td>Password</td>
<td>Champ de texte éditable. Le texte saisi est crypté.</td>
<td></td>
</tr>
</tbody>
</table>

**Avertissement:** Il est recommandé que d’entourer les valeurs des variables de type String par des guillemets doubles afin d’éviter d’éventuelles erreurs lors de l’exécution du Job.

6. Si nécessaire, cochez la case à côté de la variable de votre choix et saisissez, dans le champ **Prompt**, le message que vous souhaitez afficher.

   Cela vous permet de voir un prompt pour la valeur de la variable et de le modifier lors de l’exécution. Vous pouvez afficher ou cacher une colonne **Prompt** de la table en cliquant sur le triangle noir pointant vers la droite ou la gauche, à côté du nom du contexte correspondant.

7. Répétez les étapes ci-dessus pour définir toutes les variables pour les différents contextes.

   - port, de type **String**,
   - database, de type **String**,
   - username, de type **String**,
   - password, de type **Password**,
   - table_name, de type **String**.

**Résultats**

Toutes les variables créées et leurs valeurs sous les différents contextes sont affichées dans la table et sont prêtes à être utilisées dans votre Job. Vous pouvez continuer de modifier les variables dans cette vue si nécessaire.

Vous pouvez également ajouter une variable de contexte built-in dans le référentiel afin de pouvoir la réutiliser dans d’autres Jobs. Pour plus d’informations, consultez **Ajouter une variable de contexte built-in dans le Repository** à la page 99.

Consultez également:
Définir les variables à partir de la vue Component

Pourquoi et quand exécuter cette tâche

La manière la plus rapide de créer une seule variable de contexte est d’appuyer sur la touche F5 à partir de la vue Component. L’exemple suivant démontre comment créer une variable de contexte lors de la configuration d’un chemin de fichier pour un composant dans un Job.

Procédure

1. Dans le champ Component correspondant, placez votre curseur dans le champ que vous souhaitez paramétrer.
2. Appuyez sur F5 pour afficher la boîte de dialogue New Context Parameter :

![New Context Parameter](image)

3. Donnez un nom à cette nouvelle variable dans le champ Name, renseignez le champ Comment si nécessaire et choisissez le Type.

**Remarque**: Le nom de la variable doit respecter certaines règles de saisie et ne doit contenir aucun caractère interdit, notamment les espaces.

4. Saisissez un message Prompt à afficher pour confirmer l’utilisation de cette variable lors de l’exécution du Job courant (généralement utilisé comme test), puis cochez la case Prompt for value pour afficher le message Prompt et le champ de valeur editable lors de l’exécution.

5. Si vous avez déjà renseigné la valeur du champ de propriétés correspondant, cette valeur est automatiquement reprise dans le champ Default value. Sinon, saisissez la valeur par défaut que vous souhaitez utiliser.

7. Cliquez sur l'onglet de la vue **Contexts**. Le nom de la variable doit respecter certaines règles de saisie et ne doit contenir aucun caractère interdit, notamment les espaces.

**Résultats**
Pour plus d'informations concernant la création ou l'édition d'un contexte, consultez. La variable créée est automatiquement stockée dans tous les contextes existants, mais vous pourrez par la suite changer la valeur de manière indépendante dans chaque contexte. Pour plus d'informations concernant la création ou l'édition d'un contexte, consultez Définir les contextes à la page 90.

**Centraliser les variables de contexte dans le Repository**
Vous avez la possibilité de centraliser tous les contextes dans le Repository si vous avez besoin de les réutiliser dans plusieurs Jobs ou Routes.

Vous pouvez stocker les variables de contexte dans le référentiel de plusieurs façons :

- Créer un groupe de contexte à l'aide de l'assistant **Create / Edit a context group**. Consultez Créer un groupe de contexte et définir les variables de contexte à la page 94 pour plus de détails.
- Ajouter une variable de contexte built-in dans un groupe de contexte nouveau ou déjà existant dans le référentiel. Consultez Ajouter une variable de contexte built-in dans le Repository à la page 99 pour plus de détails.
- Sauvegarder un contexte à partir de métadonnées. Consultez Créer un contexte à partir d'une métadonnée à la page 100 pour plus d'informations.

**Créer un groupe de contexte et définir les variables de contexte**
L'exemple suivant montre comment utiliser l'assistant **Create/Edit a context group** pour créer un groupe de contexte appelé **TalendDB** contenant deux contextes (appelés **Prod** et **Test**) et pour définir un série de variables (**host**, **port**, **database**, **username**, **password** et **table_name**) sous ces deux contextes, dans le Repository, pour réutilisation dans des Jobs de gestion de base de données.

Une fois que vous avez créé et adapté tous les contextes souhaités, cliquez sur **Finish** pour valider.
Le groupe de contexte s’affiche ainsi sous le nœud **Contexts** dans la vue **Repository**. Vous pouvez continuer d’éditer le groupe de contexte, les contextes et les variables de contexte dans l’assistant en cliquant-droit sur le nœud **Contexts** et en sélectionnant **Edit context group** dans le menu contextuel.

**Procédure**

1. Cliquez-droit sur le nœud **Contexts** du **Repository** et sélectionnez **Create new context group** dans le menu contextuel.
   Un assistant en deux étapes s'ouvre pour vous aider à définir les différents contextes et leurs paramètres.

2. Dans la première étape, saisissez un nom pour le groupe de contexte à créer (**TalendDB** dans cet exemple) et ajoutez toutes les informations générales, telles que la description, si elles sont requises. Les informations que vous fournissez dans le champ **Description** apparaissent dans une info-bulle lorsque vous bougez le curseur de votre souris au-dessus du groupe de contexte dans le Repository.
3. Cliquez sur **Next** pour passer à la seconde étape qui vous permet de définir les différents contextes et variables dont vous avez besoin.

Un contexte appelé **Default** a été créé et défini par le système comme étant le contexte par défaut.

4. Cliquez sur le bouton [+] dans le coin supérieur droit de l’assistant pour définir les contextes. La boîte de dialogue **Configure Contexts** s’ouvre.
5. Sélectionnez le contexte Default, cliquez sur le bouton Edit... et saisissez Prod dans la boîte de dialogue Rename Context qui s’ouvre pour renommer le contexte Default en Prod. Cliquez sur OK pour fermer la boîte de dialogue.

6. Cliquez sur le bouton New... et saisissez Test dans la boîte de dialogue New Context. Cliquez sur OK pour fermer la boîte de dialogue.

7. Cochez la case à côté du contexte que vous souhaitez configurer comme contexte par défaut. Vous pouvez également définir le contexte par défaut en sélectionnant le nom du contexte dans la liste Default context environment dans l’assistant.

   Si nécessaire, déplacez un contexte vers le haut ou vers le bas en le sélectionnant et en cliquant sur le bouton Up ou Down.

   Dans cet exemple, définissez Test comme contexte par défaut et déplacez-le vers le haut.
8. Cliquez sur OK pour valider le contexte que vous avez défini et fermer la boîte de dialogue Configure Contexts.

Les contextes nouvellement créés sont montrés dans la table des variables de contexte de l’assistant.

Définir les variables de contexte

**Procédure**

1. Cliquez sur le bouton [+ ] en bas de l’assistant pour ajouter une ligne de paramètre dans la table.
2. Cliquez dans le champ **Name** et saisissez le nom de la variable que vous créez, *host* dans cet exemple.

3. Dans la liste **Type**, sélectionnez le type de la variable correspondant au champ de composant dans lequel cette variable sera utilisée. Dans cet exemple, il s’agit du type **String** pour la variable *host*.

4. Si nécessaire, cliquez dans le champ **Comment** et saisissez un commentaire pour décrire la variable.

5. Cliquez dans le champ **Value** et saisissez la valeur de la variable sous chaque contexte.
   Selon le type de variable, le champ **Value** différera légèrement lorsque vous cliquerez dedans et fonctionnera de manière différente :

<table>
<thead>
<tr>
<th>Type</th>
<th>Champ de valeur</th>
<th>Valeur par défaut</th>
</tr>
</thead>
<tbody>
<tr>
<td>String (type par défaut)</td>
<td>Champ de texte modifiable</td>
<td>Null</td>
</tr>
<tr>
<td>Boolean</td>
<td>Liste déroulante avec deux options : true et false</td>
<td></td>
</tr>
<tr>
<td>Character, Double, Integer, Long, Short, Object, BigDecimal</td>
<td>Champ de texte éditable</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Champ de texte éditable, avec un bouton pour ouvrir la boîte de dialogue Select Date &amp; Time.</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Champ de texte éditable, avec un bouton pour ouvrir pour la boîte de dialogue Open pour la sélection de fichiers.</td>
<td></td>
</tr>
<tr>
<td>Directory</td>
<td>Champ de texte éditable, avec un bouton pour ouvrir pour la boîte de dialogue Browse for Folder pour la sélection de dossiers.</td>
<td></td>
</tr>
<tr>
<td>List of Value</td>
<td>Champ de texte éditable, avec un bouton pour ouvrir la boîte de dialogue Configure Values pour la création et la configuration de listes.</td>
<td>(Vide)</td>
</tr>
<tr>
<td>Password</td>
<td>Champ de texte éditable. Le texte saisi est crypté.</td>
<td></td>
</tr>
</tbody>
</table>

**Avertissement**: Il est recommandé que d’entourer les valeurs des variables de type String par des guillemets doubles afin d’éviter d’éventuelles erreurs lors de l’exécution du Job.

6. Si nécessaire, cochez la case à côté de la variable de votre choix et saisissez, dans le champ **Prompt**, le message que vous souhaitez afficher. Cela vous permet de voir un prompt pour la valeur de la variable et de le modifier lors de l’exécution.

Vous pouvez afficher ou cacher une colonne **Prompt** de la table en cliquant sur le triangle noir pointant vers la droite ou la gauche, à côté du nom du contexte correspondant.

7. Répétez les étapes décrites ci-dessus pour définir toutes les variables dans cet exemple.
   - *port*, de type **String**,
   - *database*, de type **String**,
   - *username*, de type **String**,
   - *password*, de type **Password**,
   - *table_name*, de type **String**
Toutes les variables créées et leurs valeurs sous les différents contextes sont affichées dans la table et sont prêtes à être utilisées dans votre Job.
Vous pouvez continuer de modifier les variables si nécessaire. Via les préférences de la configuration, vous pouvez activer ou désactiver la propagation des modifications des variables à vos Jobs. Pour plus d’informations, consultez Préférences des performances (Talend > Performance) à la page 512.

Ajouter une variable de contexte built-in dans le Repository

Pourquoi et quand exécuter cette tâche

Vous pouvez sauvegarder une variable de contexte built-in définie dans un Job ou une Route dans un nouveau groupe de contexte ou un groupe de contexte déjà existant, sous réserve que la variable de contexte n’existe pas déjà dans le groupe.

Procédure

1. Dans la vue Context d’un Job ou d’une Route, cliquez-droit sur la variable de contexte que vous souhaitez ajouter dans le Repository et sélectionnez Add to repository context dans le menu contextuel pour ouvrir la boîte de dialogue Repository Content.

2. Dans la boîte de dialogue, effectuez l’une des actions suivantes :
   - pour ajouter votre variable de contexte à un nouveau groupe de contexte, sélectionnez Create new context group et saisissez un nom pour le nouveau groupe de contexte dans le champ Group Name, puis cliquez sur OK.
• pour ajouter une variable de contexte à un groupe de contexte déjà existant, sélectionnez le groupe de contexte et cliquez sur OK.

**Avertissement:** Lorsque vous ajoutez une variable de contexte built-in à un groupe de contexte déjà existant, assurez-vous que la variable n’existe pas déjà dans le groupe de contexte.

Dans cet exemple, ajoutez la variable de contexte `password` définie dans un Job à un nouveau groupe de contexte appelé `DB_login`.

La variable de contexte est ajoutée au groupe de contexte de votre choix dans le **Repository**, avec les contextes built-in définis.

** Créer un contexte à partir d’une métadonnée **

Lorsque vous créez ou éditez une connexion de métadonnées (à l’aide d’un assistant de métadonnées File ou DB), vous avez la possibilité de sauvegarder les paramètres de connexion en tant que variables de contexte dans un groupe de contexte nouvellement créé sous le nœud **Contexts** du Repository.

Pour ce faire, complétez vos détails de connexion et cliquez sur le bouton **Export as context** dans la seconde étape de l’assistant.

Pour plus d’informations concernant cette fonctionnalité, consultez Exporter une métadonnée en tant que contexte et réutiliser ses paramètres de contexte pour configurer une connexion à la page 428.

**Appliquer des variables de contexte du référentiel dans un Job ou dans une Route**

Une fois le groupe de contexte créé et stocké dans le **Repository**, vous pouvez l’appliquer à un Job par deux moyens :

• Déposez un groupe de contexte. Ainsi, le groupe sera appliqué comme un tout. Consultez Déposer un groupe de contexte dans un Job ou dans une Route à la page 101 pour plus de détails.
• Utilisez le bouton ‣. De cette manière, les variables d’un groupe de contexte pourront être appliquées séparément. Consultez Appliquer des variables de contexte dans un Job ou une Route à l’aide du bouton contexte à la page 101 pour plus de détails.

Déposer un groupe de contexte dans un Job ou dans une Route

Pourquoi et quand exécuter cette tâche

Pour déposer un groupe de contexte dans un Job ou une Route, procédez comme suit :

Procédure

1. Double-cliquez sur le Job ou la Route auquel ajouter un groupe de contexte.
2. Une fois le Job ouvert ou la Route ouverte, glissez le groupe de contexte de votre choix, soit dans l’espace de modélisation graphique, soit dans la vue Contexts sous l’espace de travail.

La vue Contexts montre tous les contextes et toutes les variables du groupe. Vous pouvez :
• éditer les contextes en cliquant sur le bouton [+ ] dans le coin supérieur droit de la vue Contexts.
• supprimer tout le groupe ou n’importe quelle variable en sélectionnant le nom de groupe ou la variable et en cliquant sur le bouton X.
• sauvegarder n’importe quelle variable de contexte importée comme variable built-in en cliquant-droit dessus et en sélectionnant Add to built-in dans le menu contextuel.
• double-cliquer sur n’importe quelle variable de contexte pour ouvrir le groupe de contexte dans l’assistant Create / Edit a context group et mettre à jour les changements dans le Repository.

Appliquer des variables de contexte dans un Job ou une Route à l’aide du bouton contexte

Pourquoi et quand exécuter cette tâche

afin d’appliquer les variables de contexte dans un Job ou une Route, procédez comme suit :
**Procédure**

1. Double-cliquez sur le Job auquel ou la Route à laquelle ajouter une variable de contexte.
2. Une fois le Job ouvert ou la Route ouverte dans l’espace de modélisation graphique, cliquez sur la vue **Contexts** sous l’espace de travail afin de l’ouvrir.
3. Au bas de la vue **Contexts**, cliquez sur le bouton afin d’ouvrir l’assistant pour sélectionner les variables de contexte à appliquer.
4. Dans l’assistant, sélectionnez les variables de contexte que vous souhaitez appliquer, ou effacez celles dont vous n’avez pas besoin.

   **Remarque:** Les variables de contexte appliquées sont automatiquement cochées et ne peuvent être supprimées.

5. Cliquez sur **OK** pour appliquer les variables de contexte dans le Job ou la Route.
   La vue **Contexts** montre le groupe de contexte et les variables de contexte sélectionnées. Vous pouvez modifier les contextes en cliquant sur le bouton [+] dans le coin supérieur droit de la vue **Contexts**. Vous pouvez également supprimer tout le groupe ou n’importe quelle variable en sélectionnant le nom du groupe ou la variable et en cliquant sur le bouton **X**, mais vous ne pouvez pas éditer les variables stockées dans le Repository dans cette vue.

**Utiliser des variables dans un Job ou une Route**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez utiliser une variable globale déjà existante, une variable de contexte définie dans votre Job ou Route ou une variable de contexte stockée dans le Repository appliquée à votre Job dans tous les champs de propriétés de vos composants.
Procédure

1. Dans la vue Component correspondante, placez votre curseur de souris dans le champ que vous souhaitez paramétrer et appuyez sur Ctrl+Espace pour afficher la liste complète de toutes les variables globales et de contexte définies ou appliquées dans votre Job ou Route.
La liste s’allonge au fur et à mesure que vous définissez des variables (variables de contexte).

2. Double-cliquez sur la variable de votre choix pour la renseigner dans le champ.

Exécuter un Job ou une Route dans un contexte défini

Vous pouvez sélectionner le contexte dans lequel vous souhaitez exécuter votre Job ou votre Route.

Procédure

2. Dans la zone Context, sélectionnez le contexte approprié parmi ceux que vous avez créés.

Si vous n’avez pas créé de contexte, seul le contexte par défaut, Default, est proposé dans la liste.
Toutes les variables de contexte que vous avez créées pour le contexte en sélection, ainsi que leurs valeurs associées apparaissent dans un tableau.
Pour rendre permanent une modification de valeur de variable, vous devez la changer dans la vue Context si votre variable est de type built-in ou sélectionnez un autre groupe de contexte du Repository.
Gestion des Jobs : Utilisation avancée

Les sections ci-dessous donnent des informations précises concernant différentes situations de configuration avancée des Jobs d'intégration de données, comme la gestion de flux multiples d’entrée et de sortie, l’utilisation de requêtes SQL, l’utilisation de composants externes au Job, la planification d’une tâche pour exécuter votre Job.

Créer une requête à l’aide de SQLBuilder

SQLBuilder vous permet de créer des requêtes SQL et de contrôler les modifications et écarts entre tables de base de données et tables de métadonnées. Cet éditeur est disponible pour DBInput et DBSQLRow.

Vous pouvez créer une requête avec SQLbuilder que votre schéma de table soit stocké dans le Repository ou intégrées en “built-in” directement dans les propriétés du composant.

Renseignez les informations de connexion DB et sélectionnez l’entrée de référentiel appropriée si vous l’avez définie.

Supprimez la déclaration de requête présentée par défaut dans le champ Query, dans l’onglet Basic settings de la vue Component. Puis ouvrez l’éditeur de requête SQL Builder en cliquant sur le bouton [...].

L’éditeur de requêtes SQL Builder est composé de quatre vues :

- Current Schema : schéma actuel
- Database structure : arborescence de la base de données
- Query editor : éditeur de requêtes SQL
• Query execution view : résultat des requêtes de l’éditeur
• Schema view : vue du schéma

La structure de la base de données indique les tables pour lesquelles un schéma a été défini, soit dans une entrée du Repository, soit directement dans votre connexion built-in.

La vue Schema, en bas à droite de la fenêtre, fournit une description des colonnes.

Comparer les structures de base de données

Dans la vue Database Structure s’affichent toutes les tables stockées dans l’entrée de métadonnées DB Connection du Repository ou, dans le cas d’un schéma de type built-in, les tables de la base de données elle-même.

Remarque:

La connexion à la base de données, dans l’usage d’un schéma en built-in ou dans le cas d’une opération de rafraîchissement du schéma de Repository, peut prendre du temps.

Cliquez sur l’icône de rafraîchissement pour afficher les différences entre les tables de métadonnées d’une DB connection et les tables de la base de données elle-même.

L’icône de la colonne Diff indique la présence de différences ou d’écarts entre les tables. Développez l’arborescence d’une table pour identifier la colonne exacte présentant des différences.

Le surlignage rouge indique que le contenu de la colonne comporte des différences ou que cette colonne n’est pas présente dans la table équivalente de la base de données elle-même.

Le surlignage bleu indique que la colonne n’est pas est présente dans la table stockée dans Repository > Metadata.
Créer une requête

Pourquoi et quand exécuter cette tâche

L’éditeur de requête **SQL Builder** est un système multi-onglets vous permettant de construire autant de requêtes que vous le souhaitez.

Pour créer une nouvelle requête, procédez comme suit :

**Procédure**

1. Cliquez-droit sur la table ou sur une colonne de la table et sélectionnez **Generate Select Statement** dans le menu contextuel.

2. Cliquez sur l’onglet vide présenté par défaut et saisissez votre requête directement, ou accédez à la liste d’auto-complétion par **Ctrl+Espace**. L’outil d’aide à la complétion vous propose une liste de déclarations SQL courantes (Select, From, And, Groupby, etc.) ainsi que les colonnes de la table qui peuvent faire l’objet d’une requête.

Alternativement, le **Designer** graphique de requête vous permet de manipuler facilement des tables et de générer en temps réel la requête correspondante dans l’onglet **Edit**.

3. Cliquez sur l’onglet **Designer** pour passer du mode **Edit** manuel au mode graphique.

**Remarque:**

Certaines déclarations SQL ne peuvent être interprétées par défaut.

4. Si vous avez sélectionné une table, toutes les colonnes sont sélectionnées par défaut. Décochez la case correspondant aux colonnes que vous souhaitez exclure de la sélection.

5. Ajoutez des tables grâce à un simple clic-droit. Dans l’onglet **Designer**, ajoutez d’autres tables en sélectionnant **Add tables** dans le menu contextuel puis en choisissant la table à ajouter dans la liste.

   Si des jointures existent déjà entre ces tables, elles apparaissent automatiquement dans l’éditeur.

   Vous pouvez également créer facilement une jointure entre deux tables. Cliquez-droit sur les colonnes de la première table à lier et sélectionnez **Equal** dans le menu contextuel.
Les résultats de la requête active s’affichent dans la vue Results.
7. Si nécessaire, vous pouvez cocher la case context mode pour conserver l’instruction de requête originale et la personnaliser dans la zone Query du composant. Par exemple, si un paramètre de contexte est utilisé dans l’instruction de requête, vous ne pouvez pas l’exécuter en cliquant sur le bouton dans la barre d’outils.
8. Cliquez sur OK. L’instruction de requête est automatiquement chargée dans la zone Query du composant.

**Stocker une requête dans le Repository**

Pour pouvoir récupérer et réutiliser les requêtes que vous avez créées, il est recommandé de les conserver dans le Repository.

Dans l’éditeur SQL Builder, cliquez sur l’icône dans la barre d’outils pour lier la requête à la connexion de base de données et au schéma, s’ils sont également conservés dans le Repository.

La requête peut ensuite être retrouvée à partir de la vue Database structure, dans la partie gauche de l’éditeur de requêtes.
Utiliser la fonctionnalité Use Output Stream

La fonctionnalité **Use Output Stream** vous permet de traiter les données au niveau des octets via une classe `java.io.OutputStream()` écrivant les données à l'aide d'un flux binaire sans charger les données en mémoire. Lors du traitement des données en un format linéaire, par exemple, lorsque toutes les données sont de format `String`, cette fonctionnalité vous permet d'améliorer les performances globales en sortie.

La fonctionnalité **Use Output Stream** se trouve dans la vue **Basic settings** d'un grand nombre de composants, comme le `tFileOutputDelimited`.

Pour utiliser cette fonctionnalité, cochez la case **Use Output Stream** de la vue **Basic settings** d'un composant contenant cette option. Dans le champ **Output Stream** ainsi activé, définissez votre flux de sortie à l'aide d'une commande.

**Remarque:**
Avant d'utiliser la fonctionnalité Use Output Stream, vous devez ouvrir un flux.
Pour un exemple détaillé de ce prérequis et de l'utilisation de la fonctionnalité **Use Output Stream**, consultez Exemples de Jobs d'intégration de données.

Gestion des Jobs : Autres fonctions

Les sections suivantes fournissent des informations détaillées concernant divers sujets liés à la gestion de Jobs d'intégration de données, comme :

- **Utiliser les dossiers** à la page 108
- **Partager une connexion à une base de données** à la page 109
- **Ajouter une Note au Job** à la page 112
- **Afficher les onglets Code ou Outline de votre Job** à la page 113
- **Gérer l'affichage d'un sous-Job** à la page 114
- **Paramétrage des options dans la vue Job** à la page 116

**Utiliser les dossiers**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez organiser vos Jobs à l'aide de dossiers.

Pour créer un dossier, procédez comme suit :

1. **Créer un dossier** :
   - Choisissez le dossier dans la fenêtre de l'Explorateur de fichiers.
   - Cliquez sur le bouton **Create**.
   - Entrez le nom du dossier.

2. **Ajouter des Jobs au dossier** :
   - Ouvrez le Job que vous souhaitez ajouter au dossier.
   - Cliquez sur le bouton **Add**.
   - Sélectionnez le Job désiré.

3. **Organiser les Jobs dans le dossier** :
   - Vous pouvez organiser les Jobs dans le dossier en utilisant les options de menu.
   - Utilisez les options de menu pour organiser les Jobs dans le dossier.
Procédure

1. Dans la vue Repository de la perspective Integration, cliquez-droit sur Job Designs et sélectionnez Create Folder dans le menu contextuel.
La boîte de dialogue New Folder s’ouvre.

![New Folder dialog box]

2. Dans le champ Label, saisissez un nom pour le dossier et cliquez sur Finish pour confirmer vos modifications et fermer la boîte de dialogue.
Le dossier créé s’affiche sous le nœud Job Designs, dans la vue Repository.

Résultats

Remarque:
Si vous avez déjà créé les Jobs que vous souhaitez déplacer dans ce nouveau dossier, glissez-les simplement dans le dossier.

Partager une connexion à une base de données

Pourquoi et quand exécuter cette tâche
Si vous avez plusieurs Jobs utilisant une connexion à la même base de données, vous pouvez factoriser cette connexion en utilisant l’option Use or register a shared DB Connection afin que la connexion soit partagée entre le Job père et le Job fils.

Cette option a été ajoutée à tous les composants de connexion aux bases de données afin de réduire le nombre de connexions à ouvrir et fermer.

Avertissement: L’option Use or register a shared DB Connection de tous les composants de connexion aux bases de données est incompatible avec les options Use dynamic job et Use an independent process to run subjob du composant tRunJob. Utiliser une connexion partagée à une base de données avec un tRunJob ayant l’une de ces options activée fera échouer votre Job.

Considérons, par exemple, deux Jobs liés (un Job parent et un Job fils) devant se connecter à votre bases de données MySQL distante.

Pour un scénario complet, consultez MySQL.

Pour utiliser dans deux Jobs une connexion partagée à une base de données, procédez comme suit:
Services de données

**Procédure**

1. Ajoutez un composant `tMysqlConnection` (si vous travaillez avec une base de données MySQL) au Job père ainsi qu'au Job fils, si ces derniers n'utilisent pas de composant de connexion à une base de données.

2. Connectez chaque composant `tMysqlConnection` au composant approprié dans chacun de vos Jobs à l'aide d'un lien `Trigger > OnSubjobOk`.

3. Dans la vue **Basic settings** du composant `tMysqlConnection` qui va s'exécuter en premier, saisissez les informations de connexion à la base de données, si la connexion à la base de données n’est pas stockée dans le **Repository**.

4. Cochez la case **Use or register a shared DB Connection** et saisissez un nom pour la connexion dans le champ **Shared DB Connection Name**.

   Vous pouvez réutiliser cette connexion dans votre Job fils.

5. Dans la vue **Basic settings** de l’autre composant `tMysqlConnection`, dans l’autre Job, cochez la case **Use or register a shared DB Connection**, puis saisissez dans le champ **Shared DB Connection Name** le même nom que dans le Job père.

**Remarque:**

Parmi les différents Jobs partageant une même connexion à une base de données, vous devez configurer les informations de connexion uniquement dans le premier Job ouvrant la connexion à la base de données.
Gérer les icônes d’avertissement/d’erreur sur les composants

Lorsque les propriétés d’un composant ne sont pas correctement définies et contiennent une ou plusieurs erreur(s) empêchant le code du Job de se compiler correctement, des icônes d’erreur s’afficheront automatiquement à côté de l’icône du composant dans l’espace de modélisation graphique, et à côté du nom du Job dans la vue Repository.

Icônes d’erreur et d’avertissement sur les composants

Lorsqu’un composant n’est pas correctement configuré, ou si une connexion vers un autre composant manque, un cercle rouge contenant une croix blanche (icône d’erreur), ou un triangle avec un point d’exclamation (icône d’avertissement) est placé sur l’icône du composant.

Placez votre souris sur le composant pour faire apparaître les messages d’erreur et d’avertissement. Cette aide contextuelle vous informe sur les données manquantes ou le statut du composant.

Icône d’erreur sur un Job

Lorsque les paramètres d’un composant contiennent une ou plusieurs erreurs pouvant empêcher le code du Job de se compiler correctement, une croix blanche sur fond rouge s’affiche sur l’icône du Job à côté de son nom dans le Repository.

L’icône d’erreur s’affiche aussi à côté du nom du Job dans l’onglet de l’espace de modélisation.

La compilation ou la génération de code a uniquement lieu dans les cas suivants :

- l’ouverture d’un Job,
- le passage à l’onglet Code viewer,
- l’exécution du Job (en cliquant sur Run),
- l’enregistrement du Job.
L'icône d’erreur s'affiche lorsque vous effectuez l’une de ces actions.

De plus, à l’exécution du Job, la boîte de dialogue [Find Errors in Jobs] affichera la source et une courte description de chaque erreur rencontrée.

![Image](image.png)

Cliquez sur Cancel pour arrêter l’exécution ou sur Continue pour continuer l’exécution de votre Job.

**Ajouter une Note au Job**

Sélectionnez l’élément Note dans la famille Misc de la Palette puis déposez l’élément Note dans l’éditeur pour ajouter un commentaire sur un composant particulier ou sur l’ensemble de votre Job.

![Image](image.png)

Vous pouvez modifier la mise en forme de vos notes. Pour cela, sélectionnez la note puis cliquez sur l’onglet Basic settings de la vue Component.

![Image](image.png)

La case Opacity permet d’afficher ou non la couleur de fond de la note. Elle est cochée par défaut lorsque vous ajoutez une note. Si vous décochez cette case, le fond de la note devient transparent.

Les options Fonts and Colors permettent de modifier le style, la taille, la couleur, etc. de la police, ainsi que la couleur du fond et des bordures de votre note.
Les options **Adjust horizontal** et **Adjust vertical** permettent de définir l’alignement vertical et horizontal du texte dans la note.

Le champ **Text** contient le texte apparaissant dans la note.

**Afficher les onglets Code ou Outline de votre Job**

Ce panneau est situé sous la vue **Repository**. Il fournit des informations précises concernant le Job ou le Business Model ouvert dans l’espace de modélisation graphique.

Cette zone d’information est composée de deux onglets, **Outline** et **Code Viewer**, qui fournissent des informations concernant le diagramme affiché dans l’espace de modélisation(Job ou Business Model).

**INVISIBLE dans l’interface**

L’onglet **Outline** offre un aperçu rapide du Business Model ou du Job ouvert dans l’espace de modélisation graphique mais aussi une arborescence de tous les éléments utilisés dans le Job ou le Business Model. L’espace de modélisation graphique, comme les autres fenêtres, peut être redimensionné pour s’adapter à vos besoins. Ainsi, la vue **Outline** fournit un moyen pratique de vérifier où vous vous situez dans l’espace de modélisation graphique.

Comme l’espace de modélisation graphique, et comme toute zone de toute fenêtre peut être réduit(e) ou agrandi(e) selon vos besoins, la vue Outline est utile pour savoir où vous êtes dans l’espace de modélisation graphique.

Cliquez sur la zone bleue et maintenez enfoncé le clic de votre souris. La partie du diagramme visible dans l’espace de modélisation est mise en évidence à l’aide d’un rectangle bleu.

Cliquez sur la zone bleue de l’aperçu et déplacez-la pour afficher une autre partie du processus.
La vue **Outline** peut également afficher un arborescence des composants utilisés dans le diagramme. Développez le nœud d'un composant pour afficher la liste des variables disponibles pour ce composant.

Pour passer d'un aperçu à un autre, cliquez sur les icônes correspondantes en haut à droite.

**Aperçu du code**

L'onglet **Code viewer** donne un aperçu des lignes de code générées pour le composant sélectionné, sous la vue active du Job Design, ainsi que le menu d'exécution, comprenant les éléments Start, Body et End correspondant aux phases d'exécution de ce composant.

**Remarque:**

En effet, aucun code n'est généré à partir des Business Models.

A l'aide de l'aperçu de code en couleur, vous pouvez distinguer les diverses parties d'un code d'un composant sélectionné dans l'espace de modélisation. Cet aperçu est une vue partielle de l'onglet principal Code, situé en bas de l'éditeur graphique, qui lui, affiche le code généré pour l'ensemble du Job.

**Gérer l'affichage d'un sous-Job**

Un sous-Job est représenté par un rectangle bleu regroupant tous les composants constituant ce sous-Job. Chaque composant peut être considéré comme un sous-Job lorsqu'il n'est pas connecté à un autre composant.

Ce surlignage bleu vous permet de distinguer plus facilement un sous-Job d'un autre.

**Remarque:** Un Job peut être composé d'un seul sous-Job. Les rectangles oranges correspondent aux pré-Jobs et aux post-Jobs, qui sont différents des sous-Jobs. Pour plus d'informations concernant les pré-Jobs et les post-Jobs, consultez Utiliser les composants tPrejob et tPostjob à la page 70.

**Personnaliser les sous-jobs**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez modifier la couleur du sous-job et de son titre. Pour cela, sélectionnez votre sous-job et cliquez sur la vue **Component**.
Dans la vue **Basic settings**, cochez la case **Show subjob title** si vous souhaitez ajouter un titre au sous-job et dans le champ Title, saisissez le titre.

Pour modifier la couleur du titre ou du sous-job :

**Procédure**
1. Dans la vue **Basic settings**, cliquez sur le bouton **Title color/Subjob color** pour ouvrir la boîte de dialogue **Colors**.
2. Sélectionnez la couleur que vous souhaitez. Par défaut, le titre est bleu et le sous-job bleu transparent.

**Réduire l’affichage des sous-jobs**

Si votre Job est composé de nombreux sous-jobs, vous pouvez les réduire dans un souci de lisibilité. Des signes moins [−] et plus [+] en haut à droite de vos sous-jobs vous permettent de réduire et de restaurer l’affichage de vos sous-jobs.

Cliquez sur le signe moins [−] pour réduire le sous-job. Quand il est réduit, seul le premier composant du sous-job est affiché.

Cliquez sur le signe [+] pour le restaurer.

**Supprimer le surlignage d’un sous-job**

Si vous ne souhaitez surligner vos sous-jobs, vous pouvez enlever le surlignage de tous vos sous-jobs ou d’un sous-job précis.

Pour enlever le surlignage de tous vos sous-jobs, cliquez sur l’icône **Toggle Subjobs** dans la barre d’outils du Studio Talend.
Pour enlever le surlignage d’un sous-job, cliquez-droit sur ce sous-job et sélectionnez l’option **Hide subjob** dans le menu contextuel.

![Image de la fonction Hide subjob](image)

**Paramétrage des options dans la vue Job**

Dans la vue **Job**, située un bas de l’éditeur, vous pouvez définir des fonctions optionnelles du Job. Cette vue est composée de deux onglets : **Stats & Logs** et **Extra**.

L’onglet **Stats & Logs** vous permet d’utiliser de manière automatique les fonctions **Stats & Logs** et la fonction d’automatisation des paramètres de Contexte. Pour plus d’informations, consultez Automatiser l’utilisation des statistiques & logs à la page 116.

L’onglet **Extra** liste les différentes options disponibles pour automatiser certaines fonctions, notamment l’utilisation des paramètres de contexte dans la zone **Implicit Context Loading**. Pour plus d’informations, consultez Fonctions de l’onglet **Extra** à la page 118.

**Automatiser l’utilisation des statistiques & logs**

**Pourquoi et quand exécuter cette tâche**


L’onglet **Stats & Logs** est situé dans la vue **Job** sous l’espace de modélisation et vous permet d’éviter de surcharger vos Jobs avec des composants.

**Remarque:**

Cette configuration a les mêmes propriétés générales de log que les composants de Log.

Pour paramétrer les propriétés **Stats & Logs** :
Procédure

1. Cliquez sur la vue Job.
2. Sélectionnez l’onglet Stats & Logs pour afficher la vue de configuration.

3. Renseignez les informations en fonction de votre composant de sortie (console, fichier ou base de données).
4. Vérifiez que l’option Catch correspond bien à vos besoins.

Résultats

Remarque:
Vous pouvez enregistrer automatiquement ces configurations dans vos paramètres de projets en cliquant sur le bouton Save to project settings. Sinon, vous pouvez accéder à ces paramètres via File > Edit project settings > Job settings > Stats & Logs, ou via le bouton de la barre d’outils.

Lorsque vous utilisez les options Stats & Logs pour l’un de vos Jobs, vous avez la possibilité d’appliquer ces mêmes options à ses sous-jobs.

Pour cela, cliquez sur le bouton Apply to sub jobs dans l’onglet Stats & Logs de la vue Job.
Fonctions de l'onglet Extra

L'onglet Extra vous offre des options de paramétrage des données de contexte.

- Cochez la case Multithread execution pour lancer deux Jobs au même moment.
- Vous pouvez cocher l’option Implicit tContextLoad pour éviter d’utiliser le composant tContextLoad dans votre Job et ainsi automatiser l’utilisation des paramètres de contexte.

File ou Database et configurez manuellement l’accès aux fichiers ou à la base de données.

Configurez les messages (erreur/warning/info) lorsque des comportements inattendus liés aux paramètres de contexte surviennent.

Pour un exemple de chargement dynamique de paramètres de contexte à l’aide de la fonctionnalité de chargement implicite des contextes (Implicit Context Load), consultez Exemples de Jobs d’intégration de données.

- Si vous avez renseigné manuellement l’option Implicit tContextLoad, vous avez la possibilité d’enregistrer ces paramètres dans les paramètres de votre projet en cliquant sur le bouton Save to project settings, et, ainsi, réutiliser ces paramètres pour d’autres composants et dans d’autres Jobs.
- Vous avez la possibilité de récupérer les paramètres de contexte déjà définis dans les paramètres du projet (Project Settings) en cochant la case Use Project Settings.

En cochant cette case, l’option Implicit tContextLoad sera activée et tous les champs seront renseignés automatiquement.

Pour plus d’informations concernant les paramètres de contexte, consultez Définir les paramètres de contexte à la page 484.

- Vous avez aussi la possibilité de rafraîchir les paramètres de contexte récupérés des paramètres du projet en cliquant sur le bouton Reload from project settings.

Gestion des Routes : Sujets avancés

Les sections ci-dessous donnent des informations détaillées concernant les différentes configurations avancées d’une Route de médiation, notamment l’utilisation des Route resources, l’utilisation des Beans et l’utilisation de la configuration de Spring pour enrichir votre Route.

Utilisation des Route Resources

Le Route designer vous permet de créer des ressources externes pouvant être utilisées dans les Routes, par exemple, un fichier XQuery pour camel-saxon, un fichier WSDL pour camel-cxf. Cette section vous explique comment créer une ressource de Route et comment l’utiliser dans une Route.

Créer une ressource de Route

Pourquoi et quand exécuter cette tâche

Pour créer une nouvelle ressource de Route :

Procédure

1. Dans le Repository de la perspective Integration, cliquez-droit sur le nœud Resources et sélectionnez Create Resource dans le menu contextuel.
L’assistant **New Resource** s’ouvre et vous permet de configurer les propriétés principales de la nouvelle ressource de Route.

2. Renseignez les propriétés de la ressource de Route comme dans le tableau suivant :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source File</strong></td>
<td>Chemin d’accès au fichier local ou distant pour créer la ressource à partir d’un fichier existant.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Nom de la nouvelle ressource de Route. Un message s’affiche si vous saisissez des caractères interdits.</td>
</tr>
<tr>
<td><strong>Purpose</strong></td>
<td>Objectif de la ressource ou toute autre information utile concernant l’utilisation de la ressource.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Description de la ressource.</td>
</tr>
<tr>
<td><strong>Author</strong></td>
<td>Champ en lecture seule affichant par défaut le nom de l’utilisateur courant.</td>
</tr>
<tr>
<td><strong>Locker</strong></td>
<td>Champ en lecture seule affichant par défaut l’identifiant de l’utilisateur ayant verrouillé la ressource. Ce champ est vide lorsque vous créez une ressource et contient des informations uniquement lorsque vous éditez les propriétés d’une ressource existante.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>Champ en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l’aide des boutons <strong>M</strong> et <strong>m</strong>. Pour plus d’informations, consultez Gestion des versions d’un Job ou d’une Route à la page 190.</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Liste dans laquelle sélectionner le statut de la ressource que vous créez.</td>
</tr>
<tr>
<td>Champ</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>Path</td>
<td>Liste dans laquelle sélectionner le dossier dans lequel la ressource sera créée.</td>
</tr>
</tbody>
</table>

3. Cliquez sur **Finish** pour créer la ressource.
   Si vous créez la ressource à partir de rien, une ressource vide s'ouvre dans l'éditeur, dans l'espace de travail.

4. Modifiez le code de votre ressource et sauvegardez-la.

**Gérer les versions des ressources de Route**

Lorsque vous créez votre ressource de Route dans la perspective **Integration** du Studio Talend, sa version par défaut est 0.1, où 0 définit la version majeure et 1 la version mineure.

Vous pouvez créer différentes versions de la même ressource de Route. Pour ce faire :

1. Fermez votre ressource de Route si elle est ouverte dans l'espace de travail. Sinon, ses propriétés seront en lecture seule et vous ne pourrez pas les modifier.

2. Dans la vue **Repository**, cliquez-droit sur la ressource de Route et sélectionnez **Edit properties** dans la liste déroulante.
La boîte de dialogue **Edit properties** s’ouvre.

3. À côté du champ **Version**, cliquez sur le bouton **M** pour incrémenter la version majeure et le bouton **m** pour incrémenter la version mineure.
4. Cliquez sur le bouton **Finish** pour valider la modification.

**Remarque:**
Par défaut, lorsque vous ouvrez une ressource de Route, vous ouvrez sa version la plus récente.

Pour modifier la version de votre ressource de Route, vous pouvez également :

1. Fermez votre ressource de Route si elle est ouverte dans l’espace de travail. Sinon, ses propriétés seront en lecture seule et vous ne pourrez pas les modifier.
2. Dans la vue **Repository**, cliquez-droit sur votre ressource de Route et sélectionnez **Open another version** dans le menu contextuel.
3. Dans la boîte de dialogue, cochez la case **Create** et cliquez sur le bouton **M** pour incrémenter la version majeure et sur le bouton **m** pour incrémenter la version mineure.

4. Cliquez sur **Finish** pour valider les modifications et ouvrir cette nouvelle version de la ressource de Route.

Vous pouvez également sauvegarder votre ressource de Route et incrémenter sa version en cliquant sur **File > Save as...**

**Remarque:**
Cette option n'écrase pas votre ressource de Route courante. Elle sauvegarde votre ressource de Route comme une autre ressource de Route avec une autre version.

Vous pouvez accéder à une liste des différentes versions de la ressource de Route. Pour cela :

1. Cliquez-droit sur la vue **Repository** et sélectionnez **Open another version** dans le menu contextuel.
2. Sélectionnez la version de la ressource de Route que vous souhaitez éditer et cliquez sur Finish pour l'ouvrir dans l'espace de travail.

**Utiliser une ressource de Route**

Une fois créée, vous pouvez utiliser la ressource de Route dans les composants de Mediation cSOAP et cMessagingEndpoint et dans la configuration Spring.

Pour gérer les ressources de Route dans une Route, double-cliquez sur la Route pour l'ouvrir et cliquez sur l'onglet Dependencies pour afficher l'éditeur de Manifest.
Pour ajouter une ressource, dans la zone **Resources**, cliquez sur **Add...** et sélectionnez dans l’arborescence la ressource que vous souhaitez ajouter. Cliquez sur **OK**.

Une fois ajoutée, vous pouvez choisir parmi les différentes versions de la ressource en cliquant sur le champ **Version** dans la table **Resources** et sélectionnez la version que vous souhaitez utiliser dans la liste déroulante.
Services de données

- Pour supprimer une ressource de Route d’une Route, sélectionnez-la dans la liste et cliquez sur **Remove**. Il est impossible de supprimer une ressource de Route de type Built-In.
- Pour copier le chemin d’accès à une ressource de Route, sélectionnez-le dans la table et cliquez sur **Copy Path** afin de pouvoir le réutiliser dans la Route.

Cliquez sur OK pour valider vos modifications.

**Remarque:**
Les ressources de Route utilisées dans les Routes ne peuvent être supprimées du référentiel.

Lors de l’export d’une Route, les ressources appelées dans la Route sont exportées avec la Route dans un fichier KAR ESB Runtime. Pour plus d’informations concernant l’export de Routes, consultez Construction de Routes à la page 177.

**Utilisation des Beans**

Le Studio Talend vous offre la possibilité de créer des Beans Java pouvant être appelés simplement par les composants de Médiation dans des Routes. Ainsi, vous pouvez réutiliser vos bibliothèques, vos composants, vos applets et d’autres classes.

Les Beans sont stockés sous le nœud **Code** das le **Repository**.

**Remarque:**
Vous pouvez également définir des Beans dans l’onglet de Spring et les utiliser dans la Route. Pour plus d’informations concernant l’onglet Spring, consultez Utiliser la configuration de Spring à la page 128.

**Créer un Bean**

**Pourquoi et quand exécuter cette tâche**

Pour créer un bean :

**Procédure**

1. Dans la vue **Repository** de la perspective **Integration**, développez le dossier **Code**.
2. Cliquez-droit sur le nœud **Beans**.
3. Sélectionnez **Create Bean** dans le menu contextuel.
L’assistant **New Bean** s’ouvre et vous permet de renseigner les propriétés principales du nouveau Bean.

4. Renseignez les propriétés du Bean comme suit :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nom du nouveau Bean. Un message s’affiche si vous saisissez des caractères interdits.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Objectif du Bean ou toute information utile concernant l’utilisation du Bean.</td>
</tr>
<tr>
<td>Description</td>
<td>Description du Bean.</td>
</tr>
<tr>
<td>Author</td>
<td>Champ en lecture seule affichant par défaut le nom de l’utilisateur courant.</td>
</tr>
<tr>
<td>Locker</td>
<td>Champ en lecture seule affichant par défaut l’utilisateur ayant verrouillé le Bean courant. Ce champ est vide lorsque vous créez un Bean et affiche des données uniquement lorsque vous éditez les propriétés d’un Bean existant.</td>
</tr>
<tr>
<td>Version</td>
<td>Champ en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l’aide des boutons M et m. Pour plus d’informations, consultez Gestion des versions d’un Job ou d’une Route à la page 190.</td>
</tr>
<tr>
<td>Champ</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Status</td>
<td>Liste dans laquelle sélectionner le statut du Bean que vous créez.</td>
</tr>
<tr>
<td>Path</td>
<td>Liste dans laquelle sélectionner le dossier dans lequel le Bean sera créé.</td>
</tr>
</tbody>
</table>

5. Cliquez sur Finish pour créer le Bean. Un Bean d’exemple s’ouvre dans l’éditeur.

6. Saisissez le code de votre Bean.

**Utiliser un Bean**

Une fois créé, vous pouvez utiliser les Beans dans les composants de Mediation. Pour ce faire, allez dans la vue Component du composant correspondant pour configurer ses propriétés :

Pour certains composants (par exemple cBean, cDynamicRouter, etc.), vous pouvez appeler directement les Beans :

1. Dans le champ Beans class de ces composants, saisissez beans.BEAN_NAME.class. Le Bean correspondant est appelé.
2. Si le Bean appelé a plus d’une méthode, vous pouvez spécifier la méthode que vous souhaitez appeler en cochant la case Specify the method et en saisissant son nom dans le champ qui s’affiche.

Pour certains composants (par exemple cAggregate, cMulticast, etc.), vous pouvez appeler les Beans en tant que stratégie d’agrégation :

```java
package beans;

/**
 * helloExample: not return value, only print "hello + message.
 * @param [talenTypes] String
 * @param [Category] User Defined
 * @param [param] string("world") input: The string need to be printed.
 * @param [example] helloExample("world") # hello world !.
 */

public class new_beans {

    public static void helloExample(String message) {
        if (message == null) {
            message = "World"; //SNON-NLS-1$";
            System.out.println("Hello " + message + "!"); //SNON-NLS-1$ //SNON-NLS-2$
```
1. Cochez la case *Use aggregation strategy*.
2. Dans le champ *Strategy*, saisissez le nom du Bean sans préfixe ou extension.

Pour plus d’informations concernant la création de Routes, consultez Créer une Route à la page 42.

Pour plus d’informations concernant les propriétés et l’utilisation des composants de médiation, consultez la documentation des *Composants Mediation Talend ESB* sur Talend Help Center.

**Utiliser la configuration de Spring**

Spring est un framework de développement d’applications pour la version Enterprise de Java. Apache Camel est conçu pour fonctionner en harmonie avec le framework Spring. Le Route Designer du Studio Talend vous permet d’ajouter un contexte Spring à une Route pour un objectif de configuration. Vous pouvez définir des beans et des ressources en Spring XML DSL et les utiliser dans des Routes. Cela permet aux développeurs de combiner des codes Java et Spring dans la configuration des Routes. Cela s’avère utile lorsqu’il n’y a pas de composant explicite disponible dans la Palette.


Afin d’ajouter un contexte Spring à une Route, vous devez d’abord créer la Route. Pour plus d’informations concernant la création d’une Route, consultez Créer une Route à la page 42.

L’onglet *Spring* se situe dans la partie inférieure de l’espace de modélisation graphique. Lorsque vous créez une Route dans l’espace de modélisation graphique, une configuration par défaut de Spring s’ouvre dans la vue *Spring*.

Saisissez le code que vous souhaitez implémenter dans cette vue. La capture d’écran ci-dessus affiche un exemple d’utilisation d’un bean pour appeler une ressource de Route précédemment définie, nommée RouteWithSpring :

```xml
<bean id="properties" class="org.apache.camel.component.properties.PropertiesComponent">
    <property name="location" value="classpath:RouteWithSpring.properties"/>
</bean>
```

Pour plus d’informations concernant la création de ressources de Route, consultez Utilisation des Route Resources à la page 118.

Le code ci-dessous donne un autre exemple définissant un routage de message depuis un endpoint vm:a vers log:vm :

```xml
<camel:camelContext>
    <camel:route>
        <camel:from uri="vm:a" />
        <camel:to uri="log:vm" />
    </camel:route>
</camel:camelContext>
```

Lorsque vous exportez une Route, le fichier XML Spring est également exporté dans le fichier KAR. Cliquez sur le bouton Restore Default afin de retourner à la configuration par défaut de Spring.

Pour un cas réel d’utilisation de la configuration de Spring dans une Route, consultez le scénario fourni par cSEDA dans SEDA (Mediation).

Avertissement :
Lorsque la configuration de Spring est incorrecte, les Routes peuvent ne pas démarrer.

**Conception d’un Service**

**Qu’est-ce qu’un Service ?**

La perspective Integration du Studio Talend combine l’intégration de données aux services Web et permet la création graphique d’un Service incluant un fichier WSDL et un ou plusieurs Jobs de services de données couvrant toutes les sources et les cibles nécessaires à la publication du service Web. L’éditeur WSDL permet de créer ou d’éditer graphiquement des fichiers WSDL, automatisant la plupart des tâches de ces processus.

Lorsque vous créez un Service dans la perspective Integration du Studio Talend, vous pouvez :

- Créer de nouveaux fichiers WSDL ou importer des fichiers WSDL afin de visualiser leur structure.
- Créer, configurer et supprimer des objets WSDL.
- Accéder à la vue structurée à tout moment pour éditer ou documenter les objets WSDL dans le Service.
- Associer des Services à des Jobs de services de données.
- Créer et ajouter des éléments au référentiel afin de les réutiliser et de les partager (dans d’autres projets ou Services ou avec d’autres utilisateurs).
Conception de Services simples

**Avertissement:** Jusqu’à ce qu’un Service soit créé, l’espace de modélisation graphique est indisponible.

Un Service consiste en un fichier WSDL et un ou plusieurs Jobs de services de données. Les Jobs de services de données couvrent toutes les sources et cibles nécessaires aux processus d’intégration de données et les combine à des services Web. Les propriétés de chaque objet WSDL et de chaque composant d’un Job de service de données doivent être configurées séparément, afin de fonctionner correctement.

Pour plus d’informations, consultez Modifier un fichier WSDL à la page 135 et attribuer un Job de services de données à une opération de service à la page 160.

**Créer un Service**

La perspective **Integration** du Studio Talend vous permet de créer un Service à partir d’un fichier WSDL existant ou de créer un nouveau fichier WSDL à partir de rien, à l’aide de l’éditeur WSDL.

Vous pouvez également créer différents dossiers afin de mieux classifier ces Services.

Pour créer un Service :

1. Ouvrez votre Studio Talend en suivant la procédure détaillée dans le Guide de prise en main.
2. Dans la vue **Repository**, cliquez-droit sur le nœud **Services** et sélectionnez **Create Service** dans le menu contextuel.

3. La boîte de dialogue s’affiche pour vous permettre de configurer les propriétés principales du nouveau Service.
Saisissez les propriétés du Service comme suit :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nom du nouveau Service. Un message s'affiche si vous saisissez des caractères interdits.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Objectif du Service ou toute information utile concernant l'utilisation du Service.</td>
</tr>
<tr>
<td>Description</td>
<td>Description du Service.</td>
</tr>
<tr>
<td>Author</td>
<td>Champ en lecture seule affichant par défaut le nom de l'utilisateur courant.</td>
</tr>
<tr>
<td>Locker</td>
<td>Champ en lecture seule affichant par défaut l'identifiant de l'utilisateur ayant verrouillé le Service courant. Ce champ est vide lorsque vous créez un Service et contient de données uniquement lorsque vous modifiez les propriétés d'un Service existant.</td>
</tr>
<tr>
<td>Version</td>
<td>Champ en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l'aide des boutons M et m. Pour plus d'informations, consultez Gestion des versions d'un Job ou d'une Route à la page 190.</td>
</tr>
<tr>
<td>Status</td>
<td>Liste dans laquelle sélectionner le statut du Service que vous créez.</td>
</tr>
<tr>
<td>Path</td>
<td>Liste dans laquelle sélectionner le dossier où le Service sera créé.</td>
</tr>
</tbody>
</table>

4. Cliquez sur **Next** pour ouvrir la vue **Assign WSDL** dans l'assistant. Par défaut, l'option *Create new WSDL* est sélectionnée, ce qui vous permet de créer un nouveau fichier WSDL.
5. Pour importer un fichier WSDL existant, sélectionnez l’option **Import existing WSDL**. Cliquez sur **Browse...** à côté du champ **WSDL file** afin de parcourir votre système jusqu’à un fichier WSDL existant. Cochez la case **Import WSDL Schemas on finish** si vous souhaitez récupérer et stocker le schéma du fichier WSDL dans le dossier **Metadata** du référentiel.

6. Cliquez sur **Finish** pour fermer l’assistant. Le Service s’ouvre dans l’espace de modélisation graphique avec un squelette WSDL simple.
Le Service que vous avez créé s'affiche sous le nœud Services dans la vue Repository. Le PortType et les objets d'opérations sont affichées dans l'arborescence. L'icône exclamative signifie que ce service Web n'est pas encore utilisé.

7. Vous pouvez également attribuer un fichier WSDL existant à un Service après création en cliquant-droit sur le Service dans la vue Repository et en sélectionnant Assign WSDL dans le menu contextuel, pour afficher l'assistant Assign WSDL.

8. Double-cliquez sur le Service que vous avez créé dans la vue Repository, ou cliquez-droit sur le Service et sélectionnez Open WSDL Editor dans le menu contextuel pour ouvrir le Service dans l'espace de modélisation graphique.
9. Modifiez le fichier WSDL dans l’éditeur WSDL. Pour plus d’informations, consultez Modifier un fichier WSDL à la page 135.

10. Créez ou assignez un ou plusieurs Job(s) de services de données au Service à implémenter au service Web. Pour plus d’informations, consultez attribuer un Job de services de données à une opération de service à la page 160.

**Remarque:** Vous pouvez ouvrir un ou plusieurs Service(s) créé(s) en double-cliquant simplement sur le libellé du Service dans la vue Repository.

Pour créer différents dossiers pour vos Services, procédez comme suit :

1. Dans la vue Repository, cliquez-droit sur Services et sélectionnez Create folder dans le menu contextuel.

La boîte de dialogue New Folder s’ouvre.

2. Dans le champ Label, saisissez un nom pour le dossier et cliquez sur Finish pour confirmer vos modifications et fermer la boîte de dialogue.

Le dossier créé s’affiche sous le nœud Services dans la vue Repository.

**Remarque:** Si vous souhaitez déplacer dans ce dossier des Services déjà créés, glissez-les et déposez-les simplement dans le dossier.
Pour un scénario expliquant comment créer un service de données dans le cadre d’une utilisation réelle, consultez *Exemples de Routes et de services de données* sur Talend Help Center (https://help.talend.com).

Vous pouvez à présent :

- Modifier le fichier WSDL généré. Pour plus d’informations, consultez Modifier un fichier WSDL à la page 135.
- Exporter votre Service pour déploiement dans Talend Runtime. Pour plus d’informations, consultez Exporter un Service pour un déploiement à la page 164.

**Modifier un fichier WSDL**

Le Studio Talend vous permet de parcourir et de modifier graphiquement votre fichier WSDL dans l’éditeur WSDL. Chaque type d’objet WSDL clé est affiché dans un tableau (par exemple, service, binding, PortType). Chaque tableau contient une ou plusieurs ligne(s) représentant la structure de l’objet. Les objets service, binding et PortType sont liés. Une ligne s’affiche afin d’indiquer une référence (ou association) entre ces objets.

La capture d’écran ci-dessus montre un squelette WSDL simple contenant:

- **un service**, utilisé pour agréger un ensemble de ports liés, spécifiant les adresses pour les bindings, donc définir un seul endpoint de communication.
- **un binding**, mentionnant les spécifications du protocole concret et du format de données pour les opérations et les messages définis par un PortType particulier.
- **un PortType, port type**, un ensemble d’opérations abstraites dont chacune réfère à un message d’entrée et à un message de sortie.

L’éditeur WSDL vous permet de modifier un fichier WSDL dans la vue **Properties**.

La vue **Properties** est située dans la partie inférieure de l’éditeur du Studio Talend. Elle affiche la liste des attributs et des valeurs modifiables des attributs d’un objet WSDL sélectionné et contient les onglets suivants à modifier :

- L’onglet **General** affiche une liste des attributs d’un objet.
- L’onglet **Documentation** contient les informations devant être lues par l’utilisateur.
- L’onglet **Extensions** est utilisé pour ajouter des extensions des composants.

**Ajouter un service**

**Pourquoi et quand exécuter cette tâche**

L’objet de service est une collection de ports liés et spécifie l’emplacement du Service.
Pour créer un nouveau Service, procédez comme suit :

**Procédure**

1. Cliquez-droit dans la zone blanche de l'espace de modélisation graphique pour afficher le menu contextuel et sélectionnez **Add Service**.

Un nouvel objet de service est ajouté dans l'espace de modélisation avec un port.

2. Cliquez sur l'objet de service pour afficher sa vue **Properties**.

3. Dans la vue **Properties**, cliquez sur l'onglet **General**. Dans le champ **Name**, saisissez le nom du Service. Le nom du Service lui confère un nom unique parmi tous les Services définis dans le fichier WSDL.
4. Saisissez toute information concernant le Service que vous souhaitez que l'utilisateur lise, dans l'onglet **Documentation**.

5. Pour gérer les extensions, cliquez sur l'onglet **Extensions**. Vous pouvez ajouter, trier ou supprimer des extensions.

**Résultats**

Vous pouvez ajouter des ports à votre Service. Un port définit un endpoint individuel en spécifiant une seule adresse pour un binding. Pour plus d’informations, consultez Ajouter un port à un Service à la page 137.

**Ajouter un port à un Service**

**Pourquoi et quand exécuter cette tâche**

Un port définit un endpoint individuel en spécifiant une seule adresse pour un binding. Le port contient un attribut **Binding** qui référence un binding et un élément d'adresse fournissant une spécification de l'endpoint.

Les Services sont utilisés pour grouper des ensemble de ports liés. Les ports dans un Service ont les relations suivantes :

- Aucun des ports ne communique avec les autres (par exemple, la sortie de l’un des ports n’est pas l’entrée d’un autre).
- Si un Service a plusieurs ports partageant le même PortType, mais emploie différents bindings ou adresses, les ports sont alternatifs. Chaque port fournit un comportement sémantiquement équivalent (tandis que respectant les limites des formats de transport et de message imposées par chaque binding).
- Vous pouvez déterminer le PortType d’un Service en examinant ses ports. Avec cette information, un utilisateur peut déterminer si une machine donnée supporte toutes les opérations nécessaire pour compléter une tâche donnée.

Pour ajouter un port à un Service, procédez comme suit :
Procédure

1. Dans l’espace de modélisation graphique, cliquez-droit sur le Service auquel vous souhaitez ajouter un port et sélectionnez Add Port dans le menu contextuel.

2. L’assistant Port Wizard s’ouvre. Saisissez un nom, dans le champ Name. Le nom du port doit fournir un nom unique parmi les ports définis dans le Service.

3. Sélectionnez un Binding pour le port. Pour plus d’informations concernant la configuration d’un binding, consultez Configurer un binding à la page 141.

4. Sélectionnez un protocole pour le port dans la liste Protocol et saisissez l’adresse du port dans le champ Address qui apparaît.
5. Cliquez sur Finish afin de valider la création. Le port est ajouté au Service et sa vue Properties s’affiche.
6. Pour sélectionner le nouveau port, cliquez sur le port dans l’objet de service. Vous pouvez à tout moment modifier l’information saisie concernant ce port dans l’onglet **General** de la vue **Properties**.

7. Saisissez toute information concernant le port que l’utilisateur doit lire dans l’onglet **Documentation** de la vue **Properties**.

8. Pour gérer les extensions, cliquez sur l’onglet **Extensions**. Vous pouvez ajouter, trier ou supprimer des extensions.
Résultats
Vous pouvez créer un nouveau binding pour votre port ou réutilisez un port existant. Un binding définit le format du message ainsi que les détails du protocole pour les opérations et les messages définis par un PortType particulier. Pour plus d'informations, consultez Configurer un binding à la page 141.

Configurer un binding
Un binding est un objet WSDL clé fournissant une spécification concrète concernant la transmission de messages de et vers un service Web. Un binding référence exactement un PortType. La structure d’un binding est très proche de celle du PortType. Le binding contient des élément d’extensibilité (par exemple, SOAP, HTTP et MIME) spécifiant des détails du protocole. Chaque port dans un Service référence exactement un binding.

Vous pouvez créer un binding en cliquant-droit sur une zone blanche dans l’espace de modélisation graphique et en sélectionnant Add Binding dans le menu contextuel. Vous pouvez créer un nouveau binding ou en réutiliser un existant.

Créer un nouveau binding

Procédure
1. Cliquez-droit sur le Port et sélectionnez Set Binding > New Binding.

2. La boîte de dialogue New Binding s’ouvre. Saisissez le nom du binding dans le champ Name. Le nom doit être unique parmi tous les bindings définis dans le fichier WSDL. Cliquez sur OK.
Le nouveau binding est créé dans l’espace de modélisation et connecté au port.

3. Cliquez sur le binding créé, dans l’espace de modélisation graphique afin d’afficher sa vue Properties.

4. Pour spécifier le nom du binding, saisissez-le dans le champ Name ou cliquez sur l’ampoule à côté du champ Name afin d’invoquer la refactorisation.
   
   En cliquant sur l’ampoule, une boîte de dialogue Save All Modified Resources s’ouvre et vous demande de sauvegarder toutes les ressources modifiées, avant de passer à la suite.

   Cliquez sur OK. L’assistant Rename wizard s’ouvre.
Saisissez un nouveau nom dans le champ **New name**. Cochez la case **Update references** afin de propager cette modification au fichier WSDL. Cliquez sur **Preview** pour obtenir un aperçu de la source originale et de la source refactorisée.

Cliquez sur **OK** pour valider le changement et fermer la boîte de dialogue.

**5.** Cliquez sur le champ **PortType** pour configurer le PortType du binding. Vous pouvez sélectionner le PortType dans la liste contenant les PortType de votre fichier courant. Sélectionnez **New...** pour ouvrir l’assistant **New PortType** et créer un nouveau PortType, ou sélectionnez **Browse...** pour ouvrir l’assistant **Specify Port Type**. Pour plus d’informations, consultez **Configurer un PortType** à la page 148.
6. Cliquez sur le bouton **Generate Binding Content...** pour afficher l’assistant **Binding Wizard** et spécifier les détails du binding.

Le champ **Name** affiche le nom du binding à spécifier. Le champ **PortType** affiche le PortType référencé par le binding. Vous pouvez également sélectionner le PortType dans la liste contenant les types de ports de votre fichier. Sélectionnez les options de binding que vous souhaitez utiliser, dans la liste **Protocol**. Les options sont **SOAP** et **HTTP**. Utilisez le protocole **SOAP** lorsque vous voulez échanger des informations structurées et typées. Utilisez le protocole **HTTP** lorsque vous souhaitez que votre application client effectue une requête sur les informations ou les mette à jour.

Si vous sélectionnez **SOAP**, vous pouvez ensuite sélectionner le style d’encodage que vous souhaitez utiliser :
• **document literal.** Messages de style Document, encodage littéral. Utilisez ce style de binding lorsque vous souhaitez envoyer des messages SOAP pouvant être validés par un validateur XML. Tous les types de données dans un corps de message SOAP sont définis dans un schéma, les parties WSDL doivent donc pointer vers des éléments de schéma.

• **rpc literal.** Messages de style RPC, encodage littéral. Utilisez ce style de binding lorsque vous souhaitez spécifier le nom des méthodes d’opérations dans vos messages SOAP afin qu’un serveur puisse répartir les méthodes spécifiées. Les types de données doivent être définis, les parties WSDL doivent donc pointer vers des types XSD.

• **rpc encoded.** Messages de style RPC, encodage SOAP. Utilisez ce style de binding lorsque vous souhaitez encoder des graphiques de données dans vos messages SOAP afin qu’un serveur puisse désérialiser les données de l’objet. Les types de données doivent être définis, les parties WSDL doivent donc pointer vers des types XSD.

Si vous sélectionnez HTTP, vous pouvez choisir de créer un getter ou un setter HTTP.
• **HTTP GET.** Une requête GET récupère les données d’un serveur Web à partir de la valeur d’une URL et un ensemble d’en-têtes HTTP. Utilisez cette méthode lorsque vous souhaitez récupérer des informations spécifiées dans la requête.

• **HTTP POST.** Une requête POST envoie des données supplémentaires au serveur, spécifié après l’URL, ainsi que les en-têtes. Utilisez cette méthode lorsque vous souhaitez envoyer des données incluses dans le corps de la requête.

Cochez la case **Overwrite existing binding information** si vous modifiez un binding existant.

**Remarque:** Les modifications apportées aux éléments des services ne sont pas automatiquement propagées aux bindings. Vous devez régénérer le contenu du binding et cocher la case **Overwrite existing binding information** dans l’assistant du binding afin de refléter les modifications dans le WSDL.

Cliquez sur **Finish** pour valider vos spécifications et fermer l’assistant.

7. Saisissez toute information concernant le binding que les utilisateurs doivent lire, dans l’onglet **Documentation.**
8. Pour gérer les extensions, cliquez sur l’onglet **Extensions**. Vous pouvez ajouter, trier ou supprimer des extensions.

**Réutiliser un binding existant**

**Pourquoi et quand exécuter cette tâche**

Pour réutiliser un binding existant pour votre port, procédez comme suit :

**Procédure**

1. Dans l’espace de modélisation graphique, cliquez-droit sur votre port et cliquez sur **Set Binding > Existing Binding**.

2. L’assistant **Specify Binding** s’ouvre. Dans le champ **Name**, saisissez une chaîne de caractères de recherche afin de filtrer la liste ou laissez le champ vide afin de tout afficher.
3. Vous pouvez modifier la liste en sélectionnant l’une des options suivantes :
   - **Workspace** : liste les bindings disponibles dans votre répertoire workspace.
   - **Enclosing Project** : liste les bindings disponibles dans le projet contenant votre fichier.
   - **Current Resource** : liste les types de bindings disponibles dans votre fichier.
   - **Working Sets** : liste les bindings disponibles dans un ensemble spécifié de fichiers. Pour spécifier un ensemble de travail, cliquez sur Choose. Vous pouvez sélectionner des fichiers d’un ensemble de travail existant ou créer un nouvel ensemble. Une fois les fichiers sélectionnés, cliquez sur OK.

4. Les bindings sont listés dans la zone **Components**. Le champ **Declaration Location** montre l’emplacement du binding sélectionné. Sélectionnez votre binding et cliquez sur OK. Le binding est connecté au port dans l’espace de modélisation graphique.

5. Pour plus d’informations concernant la modification d’un binding dans la vue **Properties**, consultez Créer un nouveau binding à la page 141.

**Configurer un PortType**

Un PortType est un ensemble nommé d’opérations abstraites et des messages abstraits impliqués. Chaque opération se réfère à un message d’entrée et à des messages de sortie. Un PortType est référencé par un objet de binding. Chaque binding référence exactement un PortType. Puisque chaque port fait référence à exactement un binding, chaque port possède exactement un PortType.

Vous pouvez créer un PortType en cliquant-droit sur toute zone vide dans l’espace de modélisation graphique et en sélectionnant **Add PortType**. Vous pouvez également ajouter un PortType via un binding. Cela vous permet de créer un PortType et de modifier le binding afin de référencer le nouveau PortType en une action.

Vous pouvez créer un nouveau PortType pour votre binding, ou en réutiliser un existant.
Créer un nouveau PortType pour un binding

**Pourquoi et quand exécuter cette tâche**

Si vous souhaitez créer un nouveau PortType pour votre binding, procédez comme suit :

**Procédure**

1. Dans l'espace de modélisation graphique, cliquez-droit sur un objet de binding et cliquez sur **Set PortType > New PortType**.

2. L’assistant **New PortType** s’ouvre. Saisissez le nom du PortType dans le champ **Name**. Le nom doit être unique parmi tous les PortType défini dans le fichier WSDL. Cliquez sur **OK**.

Le nouveau PortType est créé et connecté au binding dans l’espace de modélisation graphique.
3. Cliquez sur l'objet du PortType que vous avez créé dans l'espace de modélisation graphique pour afficher sa vue **Properties**.

4. L'onglet **General** vous permet de définir le nom du PortType. Pour spécifier le nom du PortType, saisissez-le dans le champ **Name** ou cliquez sur l'ampoule à côté du champ **Name** pour ouvrir l'assistant **Rename**.

   En cliquant sur l'ampoule, une boîte de dialogue **Save All Modified Resources** s'ouvre et vous demande de sauvegarder toutes les ressources modifiées, avant de passer à la suite.

   Cliquez sur **OK**. L'assistant **Rename wizard** s'ouvre.
Saisissez un nouveau nom dans le champ **New name**. Cochez la case **Update references** afin de propager cette modification au fichier WSDL. Cliquez sur **Preview** pour obtenir un aperçu de la source originale et de la source refactorisée. Cliquez sur **OK** pour valider le renommage et fermer l’assistant.

5. Saisissez toute information concernant le PortType devant être lue par l’utilisateur, dans l’onglet **Documentation**.
6. Pour gérer les extensions, cliquez sur l’onglet **Extensions**. Vous pouvez ajouter, trier ou supprimer des extensions.

**Résultats**

Une fois qu’un PortType est créé, vous pouvez y ajouter des opérations afin qu’il puisse envoyer et recevoir des messages. Pour plus d’informations, consultez [Créer un nouveau binding](#) à la page 141.

**Réutiliser un PortType existant**

**Pourquoi et quand exécuter cette tâche**

Pour réutiliser un PortType pour votre binding, procédez comme suit :

**Procédure**

1. Dans la vue de création, cliquez-droit sur un objet de binding et cliquez sur **Set PortType > Existing PortType**.

2. L’assistant **Specify Port Type** s’ouvre. Dans le champ **Name**, saisissez une chaîne de caractères de recherche afin de filtrer la liste ou laissez le champ vide afin de tout afficher.
3. Vous pouvez modifier la liste en sélectionnant l’une des options suivantes :
   • **Workspace** : liste les ports disponibles dans votre répertoire workspace.
   • **Enclosing Project** : liste les ports disponibles dans le projet contenant votre fichier.
   • **Current Resource** : liste les PortType disponibles dans votre fichier.
   • **Working Sets** : liste les ports disponibles dans un ensemble spécifié de fichiers. Pour spécifier un ensemble de travail, cliquez sur **Choose**. Vous pouvez sélectionner des fichiers d’un ensemble de travail existant ou créer un nouvel ensemble. Une fois les fichiers sélectionnés, cliquez sur **OK**.

4. Les PortType existants sont listés dans la zone **Components**. Le champ **Declaration Location** montre l’emplacement du PortType. Sélectionnez votre port et cliquez sur **OK**.

5. Pour plus d’informations concernant les modifications du PortType dans la vue **Properties**, consultez Créer un nouveau PortType pour un binding à la page 149.

**Ajouter une opération**

**Pourquoi et quand exécuter cette tâche**

Un élément d’opération nomme l’opération et liste les entrées et les sorties attendues. L’élément d’opération peut également contenir un sous-élément Fault décrivant toute erreur de données pouvant être retournée par l’opération.

A l’aide d’une opération, vous pouvez déclarer quatre transmissions primitives qu’un endpoint peut supporter :
   • **One-way** : l’endpoint reçoit un message.
   • **Request-response** : l’endpoint reçoit un message et envoie un message corrélé.
   • **Solicit-response** : l’endpoint envoie un message et reçoit un message corrélé.
   • **Notification** : l’endpoint envoie un message.

Vous pouvez ajouter une opération à un PortType ou à un binding de port.

**Remarque** : Les modifications apportées aux éléments des services ne sont pas automatiquement propagées aux bindings. Vous devez régénérer le contenu du binding et cocher la case **Overwrite existing binding information** dans l’assistant du binding afin de refléter les modifications dans le WSDL. Pour plus d’informations, consultez Configurer un binding à la page 141.

Pour ajouter une opération à un PortType, procédez comme suit :

**Procédure**

1. Dans l’espace de modélisation graphique, cliquez-droit sur votre PortType. Cliquez sur **Add Operation**.

2. La vue **Properties** de la nouvelle opération s’ouvre. Dans l’onglet **General**, saisissez le nom de l’opération dans le champ **Name**.


5. Cliquez-droit sur l’opération pour y ajouter des objets d’entrée, de sortie, ou Fault. Selon les entrées et les sorties de l’opération, elle peut être classifiée comme suit :
   - **One way operation** : entrée
   - **Request response operation** : entrée, sortie
   - **Solicit response operation** : sortie, entrée
   - **Notification operation** : sortie

**Résultats**

Notez que si l’opération n’est définie dans aucun binding, une icône d’erreur s’affiche automatiquement sur le nœud correspondant à l’opération dans le Repository.

**Ajouter un message**

Les messages représentent une définition abstraite des données transmises. Un message consiste en des parties logiques, chacune étant associée à une définition dans un type système. Les messages
WSDL sont des objets clés pouvant être référencés par les éléments d'entrée, de sortie et Fault d'une opération (dans un PortType).

**Remarque:** Les modifications apportées aux éléments des services ne sont pas automatiquement propagées aux bindings. Vous devez regénérer le contenu du binding et cocher la case *Overwrite existing binding information* dans l'assistant du binding afin de refléter les modifications dans le WSDL. Pour plus d'informations, consultez Configurer un binding à la page 141.

**Créer un nouveau message**

**Pourquoi et quand exécuter cette tâche**

Pour créer un nouveau message pour votre élément d'entrée, de sortie ou Fault, procédez comme suit :

**Procédure**

1. Dans l'espace de modélisation graphique, cliquez-droit sur votre élément d'entrée, de sortie ou Fault et cliquez sur *Set Message > New Message*.

2. L'assistant *New Message* s'ouvre. Saisissez le nom du message dans le champ *Name*. Ce nom doit être unique parmi tous les messages définis dans le fichier WSDL. Cliquez sur *OK*.

3. Le nouveau message apparaît dans l'espace de modélisation graphique dans le PortType.
Réutiliser un message existant

Pourquoi et quand exécuter cette tâche

Si vous souhaitez réutiliser un message existant pour votre élément d’entrée, de sortie ou Fault, procédez comme suit :

Procédure

1. Dans l’espace de modélisation graphique, cliquez-droit sur votre élément d’entrée, de sortie ou Fault et cliquez sur Set Message > Existing Message.

2. L’assistant Specify Message s’ouvre. Dans le champ Name, saisissez une chaîne de caractères de recherche afin de filtrer la liste ou laissez le champ vide pour visualiser tous les messages disponibles, dans la zone Components.
3. Vous pouvez modifier la liste en sélectionnant l’une des options suivantes :
   • **Workspace** : liste les messages disponibles dans votre répertoire workspace.
   • **Enclosing Project** : liste les messages disponibles dans le projet contenant votre fichier.
   • **Current Resource** : liste les types de messages disponibles dans votre fichier.
   • **Working Sets** : liste les messages disponibles dans un ensemble spécifié de fichiers. Pour spécifier un ensemble de travail, cliquez sur **Choose**. Vous pouvez sélectionner des fichiers d’un ensemble de travail existant ou créer un nouvel ensemble. Une fois les fichiers sélectionnés, cliquez sur **OK**.

4. Les messages existants sont listés dans la zone **Components**. Le champ **Declaration Location** montre l’emplacement du message. Sélectionnez votre message et cliquez sur **OK**.

**Résultats**

Vous pouvez ajouter des parties à votre message. Les parties sont un mécanisme flexible pour décrire le contenu abstrait logique d’un message. Pour plus d’informations, consultez **Ajouter une partie à un message** à la page 158.
Ajouter une partie à un message

Pourquoi et quand exécuter cette tâche

Tous les messages contiennent une ou plusieurs partie(s). Les parties sont un mécanisme flexible pour décrire le contenu abstrait logique d’un message. La définition du message associe chaque partie à un type, à l’aide d’un attribut message-typing.

Remarque: Les modifications apportées aux éléments des services ne sont pas automatiquement propagées aux bindings. Vous devez regénérer le contenu du binding et cocher la case Overwrite existing binding information dans l’assistant du binding afin de refléter les modifications dans le WSDL. Pour plus d’informations, consultez Configurer un binding à la page 141.

Pour ajouter une partie à un message, procédez comme suit :

Procédure

1. Dans l’espace de modélisation graphique, cliquez-droit sur un objet d’entrée, de sortie ou Fault puis cliquez sur Add Part.

La partie est ajoutée à la référence du message d’entrée, de sortie ou Fault.

2. Sélectionnez la partie en cliquant dessus dans l’objet du PortType et cliquez sur l’onglet General dans la vue Properties.
3. Dans le champ Name, saisissez le nom de la partie.
4. Votre partie peut référer à un Type ou un Element. Sélectionnez l’option adéquate pour Reference Kind.
5. Si vous avez sélectionné Type, vous pouvez choisir un type de données de schéma XML dans la liste Type. La liste des types provient de tout schéma XML référencé disponible.
6. Si vous avez sélectionné Element, vous pouvez choisir un élément de schéma XML dans la liste Element. La liste des éléments provient de tout schéma XML référencé disponible.

Créer un nouveau type pour votre fichier WSDL

Les types décrivent tous les types de données utilisés entre le client et le serveur. WSDL n’est pas exclusivement lié à un système de typage spécifique, mais utilise les spécifications de schéma XML W3C par défaut.

WSDL permet d’étendre les types de données via des éléments d’extension. Un élément d’extensibilité peut apparaître sous les types d’éléments pour identifier le système de définition de saisie utilisé et pour fournir un élément de conteneur XML pour les définitions de type.

Pour ajouter un type XSD ou un élément à votre fichier WSDL, dans l’espace de modélisation graphique, cliquez sur la flèche à droite de l’objet du type de port.

L’éditeur de schéma XML s’ouvre.

attribuer un Job de services de données à une opération de service

Pourquoi et quand exécuter cette tâche

Après création du fichier WSDL, vous devez associer chacune des opérations du fichier WSDL à un Job fournisseur de service de données pour implémenter le service Web. Vous pouvez créer un nouveau Job de services de données ou en attribuer un existant.

Notez que si l’opération n’est définie dans aucun binding, une icône d’erreur s’affiche automatiquement sur le nœud correspondant à l’opération dans le Repository. Il est alors impossible d’associer un Job à ce binding.

Pour cela :

Procédure

   
   Si l’opération n’est définie dans aucun binding, l’option Assign Job n’est pas disponible dans le menu contextuel.

2. L’assistant Assign Job s’ouvre et vous demande de choisir une option : Create a new Job and Assign it to this Service Operation (créer un nouveau Job et l’attribuer à une opération de service) ou Assign an existing Job to this Service Operation (attribuer un Job existant à l’opération de service).
Pour créer un nouveau Job, sélectionnez l’option **Create a new Job and Assign it to this Service Operation** puis cliquez sur **Next**.

L’assistant **New Job** s’ouvre. Par défaut, un nom du Job est déjà renseigné dans le champ **Name**, c’est le nom de l’opération. Modifiez-le et saisissez les propriétés du Job selon vos besoins. Pour plus d’informations concernant la définition des propriétés d’un Job, consultez [Créer un Job](#) à la page 30.

**Avertissement:** Notez que le nom du service et le nom du Job ne peuvent être les mêmes dans le Studio.

Cliquez sur **Finish** afin de valider la création.
Une esquisse de Job est ouverte dans l’espace de modélisation graphique avec un composant **tESBProviderRequest** et un **tESBProviderResponse** déjà sélectionnés et configurés. Le type de propriété, dans la liste **Property Type** du **tESBProviderResponse** est défini comme **Repository** et le nom de l’opération est récupéré par défaut. Le **tESBProviderRequest** écoute toutes les requêtes envoyées au service Web spécifié et le **tESBProviderResponse** va renvoyer la réponse correspondant à la requête. Ces deux composants sont dans la catégorie **Web Services** de la famille **ESB** de la Palette. Vous pouvez créer votre Job fournisseur de services de données selon vos besoins pour les processus d’intégration de données et exécuter le Job pour publier le service Web.

1. Pour attribuer un Job de services de données à l’opération, sélectionnez l’option **Assign an existing Job to this Service Operation** et cliquez sur **Next**.

L’assistant **Assign Job** s’ouvre. Vous pouvez rechercher une chaîne de caractères pour filtrer la liste des Jobs ou laissez vide le champ afin de voir tous les Jobs disponibles. Sélectionnez le Job souhaité dans l’arborescence puis cliquez sur **Finish** pour l’attribuer à l’opération.

Pour plus d’informations concernant la conception d’un Job, consultez [Qu’est-ce qu’un Job ?](#) à la page 29.
3. Une fois le Job assigné à l’opération de service, vous pouvez l’ouvrir dans l’espace de modélisation graphique en double-cliquant sur l’opération. Vous pouvez également cliquer-droit sur l’opération et sélectionner **Edit Job** dans le menu contextuel. Notez que si l’opération n’est définie dans aucun binding, ces opérations ne fonctionnent pas, même si un Job a été assigné au binding.

**Monitorer les messages de log d’un Job de services de données**

Lorsque vous exécutez un Job de services de données dans le Studio, l’avancement de l’exécution s’affiche dans la console, dans l’onglet **Basic Run** ou **Debug Run** de la vue **Run**. Le log est fourni par l’utilitaire de log d’Apache CXF embarqué avec le Service Builder. Par défaut, le niveau de message est **INFO**. Il comprend tout message d’erreur, ainsi que les messages de début et de fin. Il montre également la sortie du Job si vous avez utilisé un composant **tLogRow** dans le Job.

Le niveau de log **INFO** est fixe et ne peut être changé dans le Studio Talend lorsqu’il vous utilisez le composant **tLogRow**. Cependant, vous pouvez utiliser le composant **tLogCatcher** qui, lui, offre plus de possibilités. Sinon, lorsque vous déployez un Service dans Talend Runtime, vous pouvez décider le niveau des messages de log que le Talend Runtime affichera. Pour plus d’informations concernant le système de log de Talend Runtime, consultez le Guide Talend Open Studio for ESB Talend ESB Container Administration Guide (en anglais).

La capture d’écran suivante montre un exemple de log avec des messages d’erreur.
Pour plus d’informations concernant l’exécution d’un Job, consultez Exécuter le Job à la page 40.

Exportuer un Service pour un déploiement

La perspective **Integration** du Studio Talend vous permet d’exporter un Service vers un fichier .kar pouvant être déployé lors de l’exécution. Pour cela :

**Procédure**

1. Dans la vue **Repository**, cliquez-droit sur le Service que vous souhaitez exporter et sélectionnez **Export Service** dans le menu contextuel.

2. L’assistant **Export service** s’ouvre. Cliquez sur **Browse** afin de parcourir votre système jusqu’à l’emplacement où stocker le Service exporté.
3. **Remarque:** Le Studio Talend fournit par défaut un modèle de script Maven pour cette option. Vous pouvez personnaliser ce modèle selon vos besoins.

4. Cliquez sur **Finish** pour terminer l'opération d'export et fermer l'assistant.

**Résultats**

Un fichier `.kar` est créé pour le Service à l'emplacement défini. Il peut être déployé dans votre Talend Runtime.

Notez que le fichier archive du Service contient toutes les bibliothèques dépendantes requises pour démarrer le Service dans le Runtime. Mais le `.jar JDBC` est une exception. Si une source de données pour la même base de données est configurée dans le conteneur du Runtime, avec le `.jar JDBC`, une erreur LinkageError se produira lors de l'exécution du Service. Dans ce cas, il est recommandé d'utiliser la source de données pour se connecter aux bases de données.

Lors de l'export d'un Service, tous les contextes utilisés dans le Service sont également inclus dans le fichier archive. Lorsque vous déployez le fichier `.kar` dans le Runtime, le contexte par défaut est configuré dans le service à utiliser.

Pour modifier le contexte par défaut après déploiement du Service dans le Runtime, suivez les étapes ci-dessous :

1. Allez dans le sous-répertoire `<TalendRuntimePath>\container\etc` du répertoire d'installation de Talend Runtime.
2. Créez un fichier de configuration nommé `<yourservicename>.cfg` et configurez le contexte dans ce le fichier suivant : `context=<contextName>`
3. Redémarrez le Runtime.
Gestion des Services : Autres fonctions

Les sections ci-dessous donnent des informations détaillées concernant différents sujets liés à la gestion d’un Service.

Importer des schémas WSDL

L’option Import WSDL Schema du Service vous permet de récupérer et de stocker le schéma du fichier WSDL dans le dossier Metadata de la vue Repository.

Pour ce faire, cliquez-droit sur le Service dans le Repository et sélectionnez Import WSDL Schema dans le menu contextuel.

Le schéma du fichier WSDL est importé dans le dossier Metadata du Repository, sous le nœud File XML.
Pour plus d’informations concernant l’utilisation du Metadata manager, consultez *Qu’est-ce qu’un Job ?* à la page 29.

**Gestion des Jobs, des Routes et des Services**

**Activation/Désactivation d’un composant ou d’un sous-job ou d’une Route**

Vous pouvez activer ou désactiver un sous-job ou une Route directement connecté(e) au composant sélectionné. Vous pouvez également activer ou désactiver uniquement un composant ainsi que tous les sous-jobs ou dans une Route complète liés à un composant de début. Le composant de début est le composant déclencheur du Job. Il a un arrière plan vert.

Lorsqu’un composant ou sous-job ou une Route est désactivé, vous ne pouvez pas créer ou modifier de liens à partir ou vers ce dernier. De plus, à l’exécution, aucun code n’est généré pour le composant, le sous-job ou la Route désactivé(e).
Activer ou désactiver un composant

Pourquoi et quand exécuter cette tâche
Pour activer ou désactiver un composant, procédez comme suit :

Procédure
1. Cliquez-droit sur le composant que vous souhaitez activer ou désactiver, le tFixedFlowInput par exemple.
2. Sélectionnez l’option correspondant à l’action que vous souhaitez effectuer :
   • Activate tFixedFlowInput_1 si vous souhaitez l’activer.
   • Deactivate tFixedFlowInput_1 si vous souhaitez le désactiver.

Activer ou désactiver un sous-job

Pourquoi et quand exécuter cette tâche
Pour activer ou désactiver un sous-job, procédez comme suit :

Procédure
2. Sélectionnez l’option correspondant à l’action que vous souhaitez effectuer :
   • Activate current Subjob si vous souhaitez l’activer.
   • Deactivate current Subjob si vous souhaitez le désactiver.

Activer ou désactiver tous les sous-jobs liés

Pourquoi et quand exécuter cette tâche
Pour activer ou désactiver tous les sous-jobs liés, procédez comme suit :
Services de données

**Procédure**

1. Cliquez-droit sur le composant de début.
2. Sélectionnez l’option correspondant à l’action que vous souhaitez effectuer :
   - **Activate all linked Subjobs** si vous souhaitez les activer.
   - **Deactivate all linked Subjobs** si vous souhaitez les désactiver.

**Activer ou désactiver une Route**

**Pourquoi et quand exécuter cette tâche**

Pour activer ou désactiver une Route, procédez comme suit :

**Procédure**

1. Cliquez-droit sur n’importe quel composant formant une Route.
2. Sélectionnez l’option correspondant à l’action que vous souhaitez effectuer :
   - **Activate whole Route flow** si vous souhaitez l’activer.
   - **Deactivate whole Route flow** si vous souhaitez le désactiver.

**Import/export d’éléments et construction de Routes et de Jobs**

Le Studio Talend vous permet d’importer/exporter vos Jobs, Routes, ou éléments dans vos Jobs, Routes et Services à partir de/vers différents projets ou différentes versions du Studio. Il vous offre également la possibilité de construire des Jobs ou des Routes et ainsi de déployer et d’exécuter les Jobs ou les Routes créés dans le Studio sur n’importe quel serveur.

**Import d’éléments**

Vous pouvez importer des éléments des versions antérieures du Studio Talend ou d’un autre projet de votre version courante.

Les éléments que vous pouvez importer sont :

- Business Models
- Jobs
- Routes
- Services
- Routines
- Documentation
- Métadonnées

Pour importer les éléments stockés dans un répertoire local, procédez comme suit :

1. Sélectionnez l’option **Select root directory** dans la boîte de dialogue **Import items**.
2. Cliquez sur **Browse** afin de parcourir l’arborescence de votre répertoire Workspace jusqu’au dossier portant le même nom que votre projet. Il doit correspondre au nom du projet sélectionné.
3. Si vous souhaitez importer des éléments en particulier, tels que des Jobs Designs, vous pouvez sélectionner le dossier correspondant, par exemple le dossier Process où sont stockés tous les Jobs du projet. Si vous n’avez que des Business Models à importer, sélectionnez le dossier BusinessProcess puis cliquez sur OK.

Cependant, si votre projet contient différents types d’éléments (Business Models, Jobs Designs, Routes, Services, Métadonnées, Routines...), il est recommandé de sélectionner le dossier du projet afin d’importer tous les éléments en une fois puis de cliquer sur OK.

4. Si nécessaire, cochez la case overwrite existing items afin d’écraser les éléments existants par ceux ayant le même nom et devant être importés. Cela rafraîchit la liste Items List.

**Remarque:** Vous ne pouvez pas écraser les éléments existants, si :
- l’élément est identique, mais le chemin d’accès est différent, ou
- le nom est identique, mais l’élément est différent.

5. À partir de la liste Items List, qui contient tous les éléments valides pouvant être exportés, vous pouvez choisir les éléments à importer en cochant les cases correspondantes.


Pour importer des éléments à partir d’une archive (fichiers sources et scripts inclus), procédez comme suit :

1. Sélectionnez l’option Select archive file dans la boîte de dialogue Import items.
2. Parcourez votre système jusqu’au fichier archive à importer puis cliquez sur Open.
3. Si nécessaire, cochez la case overwrite existing items afin d’écraser les éléments existants par ceux ayant le même nom et devant être importés. Cela rafraîchit la liste Items List.
Remarque: Vous ne pouvez pas écraser les éléments existants, si :
• l’élément est identique, mais le chemin d’accès est différent, ou
• le nom est identique, mais l’élément est différent.

4. À partir de la liste Items List, qui contient tous les éléments valides pouvant être exportés, vous pouvez choisir les éléments à importer en cochant les cases correspondantes.


Pour importer des modèles SQL à partir de Talend Exchange, procédez comme suit :


2. Dans la liste Category, sélectionnez la catégorie à importer puis, dans la liste TOS_VERSION_FIL TTER, sélectionnez la version souhaitée.

Une barre de progression s’affiche indiquant que les extensions sont en cours de téléchargement. Une fois le téléchargement terminé, les extensions correspondant à la catégorie et à la version sélectionnées s’affichent dans la boîte de dialogue.

3. Dans la liste, sélectionnez l’extension que vous souhaitez importer.

Cliquez sur Finish afin de fermer la boîte de dialogue.
4. Si nécessaire, cochez la case **overwrite existing items** afin d’écarter les éléments existants par ceux ayant le même nom et devant être importés. Cela réafiche la liste **Items List**.

**Remarque:** Vous ne pouvez pas écarter les éléments existants, si :
- l’élément est identique, mais le chemin d’accès est différent, ou
- le nom est identique, mais l’élément est différent.

5. À partir de la liste **Items List**, qui contient tous les éléments valides pouvant être exportés, vous pouvez choisir les éléments à importer en cochant les cases correspondantes.


**Remarque:** S’il y a plusieurs versions d’un même élément, elles sont toutes importées dans le Projet en cours, à condition que vous n’ayez pas d’élément identique déjà existant.

**Construction de Jobs**

La fonction **Build Job** vous permet de déployer et d’exécuter un Job à partir de n’importe quel serveur, indépendamment du Studio Talend.

**Pourquoi et quand exécuter cette tâche**

En exécutant des scripts de build générés depuis des modèles définis dans les paramètres du projet, la fonctionnalité **Build Job** ajoute tous les fichiers nécessaires à l’exécution d’un Job dans une archive, y compris les fichiers .bat et .sh, ainsi que tous les fichiers de paramètres de contextes ou fichiers liés.

**Remarque:** Votre Studio Talend fournit un ensemble de scripts de construction par défaut. Vous pouvez personnaliser ces modèles afin qu’ils répondent à vos besoins. Pour plus d’informations, consultez Personnaliser des modèles de commandes Shell à la page 467 et Personnaliser les modèles de scripts de build Maven à la page 468.

Par défaut, lorsqu’un Job est construit, tous les fichiers .jar requis sont inclus dans la commande .bat ou .sh. Pour un Job complexe impliquant de nombreux Jars, le nombre de caractères dans la commande batch peut dépasser la longueur autorisée par certains systèmes d’exploitation. Pour éviter l’échec de l’exécution de la commande batch à cause de cette limitation, avant de construire votre Job, cliquez sur **Window > Preferences > Talend > Import/Export** puis cochez la case **Add classpath jar in exported jobs**, afin de regrouper les Jars dans un fichier classpath.jar ajouté au Job construit.

**Remarque:** Notez que le Job de service de données SOAP contenant le composant **tESBProviderRequest** implémentant une opération de service n’est pas autorisé à être publié indépendamment. Le Job doit être publié avec le service auquel il est assigné. Lorsque vous exportez un service SOAP, tous les Jobs de services de données implémentant les opérations de service sont également construits dans le fichier d’archive du service.

**Procédure**

1. Dans l’arborescence **Repository**, cliquez-droit sur le Job que vous voulez construire et sélectionnez **Build Job** pour ouvrir la boîte de dialogue **Build Job**.
Remarque: Vous pouvez afficher/masquer l’arborescence de tous les Jobs créés dans le Studio Talend directement à partir de la boîte de dialogue Build Job en cliquant sur les boutons ➔ et ➚, respectivement. Les Jobs que vous avez précédemment sélectionnés dans l’arborescence du Studio s’affichent, précédés d’une case cochée. Cette possibilité vous permet de modifier les éléments sélectionnés à exporter, directement depuis l’assistant, sans avoir à le fermer et à retourner dans le Repository du Studio Talend.

2. Dans le champ To archive file, cliquez sur le bouton Browse et parcourez jusqu’au répertoire dans lequel vous souhaitez sauvegarder votre Job construit.

3. Dans la zone Select the Job version, sélectionnez la version du Job que vous souhaitez construire, si vous en avez créé plusieurs versions.

4. Sélectionnez le type de build Build Type dans la liste :
   - Job standalone
   - Bundle OSGI pour ESB

   Notez que les Jobs de Services de données comprenant un tRESTRequest ne peuvent être construits qu’en tant que OSGI Bundle For ESB. Si le Job de Services de données comprend un tRESTClient ou un tESBConsumer et qu’aucun Service Registry, Service Locator ou Service Activity Monitoring n’est activé dans le composant, le Job de Services de données peut être construit en tant que OSGI Bundle For ESB ou Standalone Job. Si le Service Registry, le Service Locator ou le Service Activity Monitor est activé, le Job de services de données incluant le composant tRESTClient ou le composant tESBConsumer ne peut être construit qu’en tant que Bundle OSGI pour ESB.

5. Cochez la case Extract the zip file, si vous souhaitez extraire automatiquement le fichier dans le répertoire cible.
6. Dans la zone **Options**, sélectionnez le type de fichiers que vous souhaitez ajouter à l’archive. Les options correspondant aux fichiers nécessaires à l’exécution du Job sont cochées par défaut, Vous pouvez néanmoins les décocher en fonction de ce que vous souhaitez construire.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shell launcher</td>
<td>Cochez cette case pour exporter les fichiers .bat et/ou .sh nécessaires au lancement du Job construit.</td>
</tr>
<tr>
<td></td>
<td>• <em>All</em>: exporte les fichiers .bat et .sh,</td>
</tr>
<tr>
<td></td>
<td>• <em>Unix</em>: exporte le fichier .sh,</td>
</tr>
<tr>
<td></td>
<td>• <em>Windows</em>: exporte le fichier .bat.</td>
</tr>
<tr>
<td>Context scripts</td>
<td>Cochez cette case pour exporter TOUS les fichiers de paramètres de contexte et pas uniquement ceux sélectionnés dans la liste correspondante.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque:</strong> Pour exporter un contexte, sélectionnez le contexte répondant à vos besoins dans la liste Context scripts, notamment les fichiers .bat ou .sh contenant les paramètres de contexte appropriés. Vous pouvez ensuite, si vous le souhaitez, éditez les fichiers .bat et .sh pour modifier manuellement le type de contexte.</td>
</tr>
<tr>
<td>Apply to children</td>
<td>Cochez cette case si vous souhaitez appliquer le contexte sélectionné dans la liste à tous les Jobs fils.</td>
</tr>
<tr>
<td>Items</td>
<td>Cochez cette case pour exporter les sources utilisées par le Job durant son exécution, y compris les fichiers .item et .properties, ainsi que les sources Java et Talend.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque:</strong> Si vous cochez la case Items ou Source files, vous pouvez réutiliser le Job construit dans un Studio Talend installé sur une autre machine. Ces fichiers sources sont uniquement utilisés dans le Studio Talend.</td>
</tr>
<tr>
<td>Java sources</td>
<td>Cochez cette case pour exporter le fichier .java contenant les classes Java générées par le Job lors de sa création.</td>
</tr>
</tbody>
</table>

7. Cliquez sur le bouton **Override parameters’ values**, si nécessaire.
   Une fenêtre s’ouvre, dans laquelle vous pouvez mettre à jour les paramètres et les valeurs de contexte du Job sélectionné ou ajouter de nouveaux paramètres ou de nouvelles valeurs, selon vos besoins.

8. Cliquez sur **Finish** pour valider les changements effectués, finaliser la construction et fermer la boîte de dialogue.

**Résultats**

Un fichier zip des Jobs est alors créé à l’endroit prédéfini.

**Remarque:** Si le Job à construire appelle une Routine personnalisée contenant une ou plusieurs classe(s) Java en parallèle avec la classe publique nommée de la même façon que la routine utilisateur, la ou les classe(s) supplémentaire(s) ne sera (seront) pas incluse(s) dans le fichier exporté. Pour exporter ces classes, vous devez les inclure en tant que sous-classes dans la classe ayant le même nom que la routine. Pour plus d’informations concernant les routines utilisateur, consultez Gestion des routines personnalisées à la page 442. Pour plus d’informations concernant les classes et les sous-classes, consultez les manuels Java correspondants.
**Construire un Job en tant que Job standalone**

Dans le cas d'un export de Plain Old Java Object, si vous souhaitez réutiliser le Job dans le Studio Talend installé sur une autre machine, cochez la case **Items**. Ces fichiers sources (.item et .properties) sont uniquement utilisés dans le Studio Talend.

Lorsque que plusieurs contextes sont proposés dans la liste Context script, sélectionnez celui qui correspond à vos besoins. Ensuite, cliquez sur le bouton **Override parameters' values**, sous la case **Context scripts**. Une fenêtre s'ouvre alors et affiche tous les paramètres du contexte sélectionné. Une fenêtre s'ouvre alors et affiche tous les paramètres du contexte sélectionné.

Dans cette fenêtre, vous pouvez configurer le contexte sélectionné selon vos besoins.

**Remarque:**

Après l'export, les informations de contexte sélectionnées sont stockées dans le fichier .bat ou .sh. Les paramètres de contexte sont stockés dans le fichier de contextes .properties.

**Construire un Job en tant que bundle OSGI pour ESB**

**Pourquoi et quand exécuter cette tâche**

Dans la boîte de dialogue **Build Job**, vous pouvez changer le type de construction pour construire le Job sélectionné en tant que bundle OSGI, afin de déployer votre Job dans votre conteneur **Talend ESB Container**.
Procédure

1. Dans la zone **Job Version**, sélectionnez le numéro de version du Job que vous souhaitez construire si vous en avez créé plus d’une version.

2. Dans la zone **Build type**, sélectionnez **OSGI Bundle For ESB** afin de construire votre Job en tant que bundle OSGI.
   
   L’extension de votre construction change automatiquement en .jar puisque c’est l’extension attendue par **Talend ESB Container**.

3. Cliquez sur le bouton **Browse...** pour spécifier le dossier dans lequel construire votre Job.

4. Cliquez sur **Finish** afin de le construire.

Construction de Routes

L’option **Build Route** vous permet de construire une Route vers un fichier ESB Kar Runtime afin de le déployer dans Talend ESB Container.

**Construire une Route vers un fichier ESB Kar Runtime**

**Pourquoi et quand exécuter cette tâche**

Pour construire une Route vers un fichier ESB Kar Runtime, procédez comme suit :

Procédure

1. Dans la vue **Repository**, cliquez-droit sur la Route que vous souhaitez créer et sélectionnez **Build Route** pour ouvrir la boîte de dialogue **Build Route**.

**Remarque:**

Vous pouvez afficher/masquer l’arborescence de tous les Jobs créés dans le Studio Talend directement à partir de la boîte de dialogue **Build Route** en cliquant sur les boutons **»** et **«**, respectivement. Cette possibilité vous permet de modifier les éléments sélectionnés à exporter, directement depuis l’assistant, sans avoir à le fermer et à retourner dans le **Repository** du Studio Talend.
2. Dans le champ **To archive file**, parcourez votre système jusqu’au répertoire dans lequel vous souhaitez sauvegarder votre Route.

3. Dans la zone **Route Version**, sélectionnez le numéro de version de la Route, si vous en avez créé plus d’une version.

4. Dans la zone **Build Type**, le type **ESB Runtime Kar file** est sélectionné par défaut.

5. Cliquez sur **Finish** pour valider vos modifications, terminer l’export et fermer la boîte de dialogue.

**Résultats**

Un fichier *.kar* pour la Route est créé à l’emplacement défini.

**Modifier le Manifest d’une Route**

**Pourquoi et quand exécuter cette tâche**

Le Route designer vous permet de gérer les dépendances OSGI d’une Route. Pour cela :

**Procédure**

1. Double-cliquez sur la Route pour l’ouvrir Cliquez sur l’onglet **Dependencies** pour afficher l’éditeur de Manifest.
2. Dans la zone **Import Packages**, les packages système sont affichés en italique. Cliquez sur **Add** pour ajouter un nouveau package.
3. Dans l’assistant **Add Import-Package**, saisissez les propriétés comme expliqué dans le tableau ci-dessous puis cliquez sur **OK**. Les packages ajoutés manuellement s’affichent en noir. Vous pouvez également modifier les propriétés de ceux-ci en cliquant sur le bouton **Edit** après avoir ajouté le package. Vous pouvez aussi le supprimer en cliquant sur le bouton **Remove**.

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Saisissez le nom du Import-Package.</td>
</tr>
<tr>
<td><strong>Properties / Optional</strong></td>
<td>Cochez cette case pour résoudre le package uniquement s’il est présent.</td>
</tr>
<tr>
<td><strong>Version to match/MaxVersion</strong></td>
<td>Spécifiez la version maximale du package correspondant à l’Export-Package. Si cet attribut n’est pas spécifié, la valeur par défaut est ∞. Vous pouvez choisir entre <strong>Inclusive</strong> et <strong>Exclusive</strong> pour inclure ou exclure, respectivement, la version spécifiée. Cela est visible dans le nom du package. ( et ) indiquent l’exclusion de la version spécifiée. [ et ] indiquent l’inclusion de la version spécifiée. (xxx) indique une version unique.</td>
</tr>
</tbody>
</table>

4. Dans la zone **Export Packages**, les packages système sont affichés en italique. Cliquez sur **Add** pour ajouter un nouveau package.

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Saisissez le nom de l’Export-Package.</td>
</tr>
<tr>
<td>Version exported/Version</td>
<td>Spécifiez la version du package. Le format major.mid.min.qualifiername est supporté. Vous pouvez choisir d’utiliser qualifiername ou non. Si vous saisissez une version contenant moins de trois chiffres, le format major.mid.min est utilisé. Par exemple, si vous saisissez 1, la version affichée sera 1.0.0.</td>
</tr>
</tbody>
</table>

6. Le chemin de recherche dans le bundle est affiché dans la zone Bundle Classpath. Cliquez sur le bouton Select All ou Deselect All afin de tout inclure ou exclure du bundle OSGI.

7. Dans la zone Require Bundles, les bundles système sont affichés en italique. Cliquez sur Add pour ajouter un nouveau bundle.

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Saisissez le nom du Require-Bundle.</td>
</tr>
<tr>
<td>Properties / Optional</td>
<td>Cochez cette case pour résoudre le bundle uniquement s’il est présent.</td>
</tr>
<tr>
<td>Version to match/MaxVersion</td>
<td>Spécifiez la version maximum du bundle. Si cet attribut n’est pas spécifié, la valeur par défaut est ∞. Vous pouvez choisir entre <strong>Inclusive</strong> et <strong>Exclusive</strong> pour inclure ou exclure, respectivement, la version spécifiée. Cela est visible dans le nom du package. ( et ) indiquent l’exclusion de la version spécifiée. ( et ) indiquent l’inclusion de la version spécifiée. (xxx) indique une version unique.</td>
</tr>
</tbody>
</table>

**Résultats**

Dans l’éditeur de Manifest, vous pouvez également :

- filtrer les packages, bundles, ou les classpaths en saisissant des caractères à rechercher dans le champ **Search** et supprimer la recherche en cliquant sur le bouton 

- cocher la case **Hide Built-In Items** afin de masquer les éléments système,

- cliquer sur le bouton **Up** ou **Down** pour déplacer vers le haut ou vers le bas dans la liste, un Import-Package ou un Require-Bundle ajouté manuellement.

**Export d’éléments**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez exporter plusieurs éléments du Repository dans un répertoire ou dans un fichier archive. Vous pouvez donc exporter des métadonnées, notamment les informations de connexion aux bases de données (DB connection) ou de la Documentation en même temps que votre Job ou votre Business Model, ainsi que vos Routes et vos Services, par exemple. Pour cela :

**Procédure**

1. Dans le **Repository**, sélectionnez les éléments que vous souhaitez exporter.
2. Pour une sélection multiple, maintenez la touche **Ctrl** enfoncée puis sélectionnez les éléments souhaités.
Avertissement: Si vous souhaitez exporter les métadonnées d’une table de base de données, veillez à bien sélectionner la totalité de votre connexion à la base de données et pas uniquement la table souhaitée, afin que le processus d’export se déroule correctement.

3. Maintenez la touche Ctrl enfoncée et cliquez-droit sur l’élément à exporter, puis sélectionnez Export items dans le menu contextuel:
Au besoin, vous pouvez sélectionner des éléments supplémentaires à exporter.

4. Sélectionnez le répertoire dans lequel vous souhaitez enregistrer les éléments exportés en cliquant sur **Browse**. Sinon, définissez le fichier archive dans lequel les fichiers des éléments sélectionnés seront compressés.

**Remarque:** Si vous avez plusieurs versions d’un même élément, elles seront toutes exportées.

Cochez la case **Export Dependencies** si vous souhaitez définir et exporter les dépendances des routines avec les Jobs que vous exportez. Par défaut, toutes les routines utilisateur sont sélectionnées. Pour plus d’informations, consultez **Définition des routines** à la page 440.

5. Cliquez sur **Finish** pour fermer la boîte de dialogue et exporter les éléments.

**Changer les paramètres de contexte dans des Jobs et des Routes**

**Pourquoi et quand exécuter cette tâche**

Comme expliqué dans **Construction de Jobs** à la page 173, vous pouvez éditer les paramètres de contexte :
**Procédure**

- Si vous souhaitez changer le contexte sélectionné, il vous suffit d’éditer les fichiers `.bat` ou `.sh` et de modifier le paramètre suivant : `--context=Prod`, pour le contexte correspondant.
- Si vous souhaitez changer des paramètres particuliers d’un contexte, éditez le fichier `.bat` ou `.sh` et ajoutez le paramètre dont vous avez besoin parmi ceux du tableau suivant :

<table>
<thead>
<tr>
<th>Operation</th>
<th>Paramètre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changer la valeur pour le paramètre <code>key1</code></td>
<td><code>--context_param key1=valeur1</code></td>
</tr>
<tr>
<td>Changer <code>valeur1</code> et <code>valeur2</code> des paramètres respectifs <code>key1</code> et <code>key2</code></td>
<td><code>--context_param key1=valeur1 --context_param key2=valeur2</code></td>
</tr>
<tr>
<td>Changer une valeur contenant des caractères d’espace comme dans les chemins d’accès</td>
<td><code>--context_param key1=&quot;chemin d'accès&quot;</code></td>
</tr>
</tbody>
</table>

**Gestion des éléments du Référentiel**

Le Studio Talend vous permet d’éditer les éléments centralisés dans le Repository et de mettre à jour les Jobs qui utilisent ces éléments.

**Gérer les mises à jour des éléments du Repository**

Vous pouvez mettre à jour les paramètres des métadonnées, des contextes centralisés à chaque fois dans la vue **Repository**, afin d’actualiser la connexion à la base de données ou par exemple les détails du groupe de contexte.

Lorsque vous modifiez l’un des paramètres d’une entrée de la vue **Repository**, tous les Jobs utilisant cette entrée seront impactés par cette modification. C’est pourquoi le système vous proposera de propager ces modifications à tous les Jobs utilisant l’entrée du référentiel.

La boîte de dialogue **Update Detection** s’ouvre pour vous laisser mettre à jour les Jobs impactés lorsque :

- vous modifiez une entrée de référentiel centralisée utilisée dans n’importe quel Job et cliquez sur **Yes** dans la boîte de dialogue **Modification** qui s’ouvre automatiquement.
- vous sélectionnez **Detect Dependencies** en cliquant droit sur le menu d’une entrée de référentiel modifiée utilisée dans n’importe quel Job, ou cliquez sur l’icône ![Detect Dependencies](image) dans la barre d’outils après avoir modifié une entrée de référentiel centralisée utilisée dans n’importe quel Job.

Pour plus d’informations concernant la mise à jour des Jobs impactés, consultez Mettre à jour automatiquement les Jobs impactés à la page 186 et Mettre à jour manuellement les Jobs impactés à la page 187.

Les sections ci-dessous expliquent comment modifier les paramètres d’entrée du Repository et comment propager les modifications pour tous les Jobs ou une partie d’entre eux utilisant l’entrée en question.

**Modifier un élément du Repository**

**Pourquoi et quand exécuter cette tâche**

Pour actualiser les paramètres d’un élément du Repository, procédez comme suit :
**Procédure**

1. Développez le nœud **Metadata, Contexts** dans la vue **Repository** et parcourez jusqu'à l'entrée correspondante que vous voulez actualiser.
2. Cliquez-droit sur l'entrée et sélectionnez dans le menu contextuel l'option edit correspondante. L'assistant respectif apparaît, vos permettant d'éditer chaque étape de la définition des paramètres d'entrée.
   Pour mettre à jour les paramètres d'entrée, vous devez propager les modifications de plusieurs Jobs ou de tous les Jobs utilisant cette entrée.
   Une boîte de dialogue apparaît automatiquement à la fin de votre mise à jour ou de vos modifications, quand vous cliquez sur le bouton **Finish** dans l'assistant.

![Modification](image)

   *Do you want to propagate the modifications to all jobs? And it might be quite long.*

3. Cliquez sur **Yes** (Oui) pour fermer la boîte de dialogue et exécuter les modifications apportées à tous les Jobs concernés. Pour plus d'informations concernant la première méthode de propagation des modifications, consultez **Mettre à jour automatiquement les Jobs impactés** à la page 186.
   Cliquez sur **No** (Non) si vous voulez fermer la boîte de dialogue sans propager les changements. Cela vous permettra de propager manuellement, une par une, les modifications apportées aux Jobs concernés. Pour plus d'informations concernant la deuxième méthode de propagation des modifications, consultez **Mettre à jour manuellement les Jobs impactés** à la page 187.

**Mettre à jour automatiquement les Jobs impactés**

**Pourquoi et quand exécuter cette tâche**

Après avoir mis à jour les paramètres de chaque élément centralisé dans la vue **Repository** et utilisés dans différents Jobs, une boîte de dialogue s'ouvre et vous demande si vous souhaitez propager les modifications à tous les Jobs utilisant ces paramètres.

Pour mettre à jour les Jobs impactés, procédez comme suit :

**Procédure**

1. Dans la boîte de dialogue **Modification**, cliquez sur **Yes** si vous voulez que le système cherche dans votre **Repository** les Jobs impactés par les modifications que vous venez de faire. Ceci a pour but de propager automatiquement la mise à jour dans tous les Jobs (ouverts ou non), en un seul clic.
   La boîte de dialogue **Update Detection** s'ouvre et affiche tous les Jobs impactés par les changements.
2. Cochez les cases des Jobs à mettre à jour avec les métadonnées ou les paramètres de contexte modifiés et décochez les cases des Jobs que vous ne souhaitez pas mettre à jour. Vous pouvez les mettre à jour ultérieurement via le menu **Detect Dependencies** ou en cliquant sur l’icône ⌁. Pour plus d’informations, consultez **Mettre à jour manuellement les Jobs impactés** à la page 187.

3. Cliquez sur **OK** pour fermer la boîte de dialogue et mettre à jour les Jobs sélectionnés.

**Mettre à jour manuellement les Jobs impactés**

**Pourquoi et quand exécuter cette tâche**

Avant de propager les modifications des Métadonnées ou Contextes du Repository dans tous vos Jobs, vous souhaitez peut-être voir les impacts que cela peut entraîner. Pour ce faire, effectuez les opérations suivantes :

**Procédure**


2. Puis cliquez-droit sur cette métadonnée ou ce contexte et sélectionnez l’option **Detect Dependencies** dans le menu contextuel.

   Une barre de progression indique le processus de vérification de tous les Jobs utilisant les paramètres de métadonnée ou de contexte modifiés. Une boîte de dialogue s’ouvre et affiche tous les Jobs utilisant cet élément mis à jour.
3. Cochez les cases des Jobs à mettre à jour avec les métadonnées ou les paramètres de contexte modifiés et décochez les cases des Jobs que vous ne souhaitez pas mettre à jour.

4. Cliquez sur OK pour valider et fermer la boîte de dialogue.

Résultats

Remarque: Les Jobs que vous n’aurez pas mis à jour passeront automatiquement en Built-in, puisque le lien vers le Repository ne pourra pas être maintenu. Cependant, les paramètres resteront les mêmes que ceux définis avant les modifications.

Recherche d’un Job dans le référentiel

Pourquoi et quand exécuter cette tâche
Si vous souhaitez ouvrir un Job précis dans la vue Repository dans la perspective Integration du Studio Talend et que vous n’arrivez pas à le retrouver, cliquez simplement sur le bouton dans la barre d’outils.

Pour chercher un Job dans la vue Repository :

Procédure

1. Dans la barre d’outils Studio Talend, cliquez sur pour ouvrir la boîte de dialogue Find a Job listant automatiquement tous les Jobs créés dans le Studio courant.
2. Saisissez le nom ou une partie du nom du Job dans le champ en haut de la fenêtre. Lorsque vous commencez à saisir le nom dans le champ, la liste des Jobs est automatiquement mise à jour afin de n'afficher que le ou les Job(s) dont le nom correspond aux lettres saisies.
3. Sélectionnez le Job souhaité dans la liste et cliquez sur **Link Repository** pour afficher automatiquement le Job sélectionné dans l’arborescence de la vue **Repository**.

4. Si nécessaire, cliquez sur **Cancel** pour fermer la boîte de dialogue et cliquez-droit sur le Job sélectionné dans le **Repository** pour effectuer l’une des opérations disponibles dans le menu contextuel.
   Sinon, cliquez sur **OK** pour fermer la boîte de dialogue et ouvrir le Job sélectionné dans l’espace de modélisation.

**Gestion des versions d’un Job ou d’une Route**

Lorsque vous créez un Job ou une Route dans le Studio Talend, son numéro de version par défaut est 0.1 : 0 correspond à la version majeure et 1 à la version mineure.

Les sections suivantes décrivent comment gérer la version d’un Job ou d’une Route.

Vous pouvez également gérer la version de plusieurs Jobs, Routes et/ou métadonnées simultanément, ainsi que les Jobs et leurs dépendances et/ou leur(s) Job(s) fils ou Routes filles, dans les paramètres des projets. Pour plus d’informations, consultez **Gestion des versions** à la page 479.

**Mettre à jour la version d’un Job ou d’une Route inactif**

**Procédure**

1. Veillez à fermer votre Job ou votre Route s’il est ouvert dans l’espace de modélisation graphique. Autrement, ses propriétés seront en lecture seule et vous ne pourrez donc pas les modifier.
2. Cliquez-droit sur le libellé du Job ou de la Route dans le Repository et :
   - sélectionnez Edit properties à partir du menu contextuel. La boîte de dialogue [Edit properties] s’ouvre, puis cliquez sur le bouton [M] dans le champ Version pour incrémenter la version majeure ou sur le bouton [m] pour incrémenter la versionmineure.
   - sélectionnez Open another version à partir de la liste déroulante, et cochez la case Create new version and open it dans la boîte de dialogue, puis cliquez sur le bouton M pour incrémenter la version majeure, ou sur le bouton m pour incrémenter la version mineure.

3. Cliquez sur Finish pour valider.

Résultats
Vous avez créé une nouvelle version pour le Job ou la Route.

Remarque: Par défaut, lorsque vous ouvrez un Job ou une Route, vous ouvrez sa dernière version. Les versions précédentes du Job ou de la Route sont en lecture seule et ne peuvent donc plus être modifiées.

Mettre à jour la version d’un Job actif ou d’une Route active

Vous pouvez également sauvegarder le Job actif ou la Route active et incrémenter sa version simultanément.

Procédure
1. Cliquez sur File > Save As....
2. Indiquez une nouvelle version dans la boîte de dialogue Save As, puis cliquez sur Finish.

Remarque: Donner un nouveau nom à votre Job ou votre Route n’écrit pas la version actuellement ouverte, mais l’enregistre en tant que nouveau Job ou nouvelle Route avec la même version que le Job ou la Route actuel(le) ou une nouvelle version si vous en spécifiez une.

Utiliser différentes versions d’un Job ou d’une Route

Vous pouvez accéder à la liste des différentes versions de votre Job ou de votre Route et effectuer certaines opérations.

Procédure
1. Dans la vue Repository, sélectionnez le Job ou la Route dont vous souhaitez consulter les versions.
2. Dans le panneau des onglets de configuration, cliquez sur l’onglet Job et sélectionnez Version pour afficher la liste des versions du Job sélectionné ou de la Route sélectionnée.
3. Cliquez-droit sur la version du Job ou de la Route que vous souhaitez consulter.
4. Sélectionnez l’option :

<table>
<thead>
<tr>
<th>Select</th>
<th>Pour...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Job/Route</td>
<td>Ouvrir la dernière version du Job ou de la Route.</td>
</tr>
</tbody>
</table>

Remarque: Cette option n’est disponible que lorsque vous sélectionnez la dernière version du Job ou de la Route.
<table>
<thead>
<tr>
<th>Select</th>
<th>Pour...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read job/Route</td>
<td>Consulter le Job en lecture seule.</td>
</tr>
<tr>
<td>Open Job Hierarchy</td>
<td>Consulter la hiérarchie du Job.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque:</strong> Cette option n’est disponible que pour un Job d’intégration et non pour une Route.</td>
</tr>
<tr>
<td>Edit properties</td>
<td>Editer les propriétés du Job ou de la Route.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong> le Job ou la Route ne doit pas être ouvert(e) dans l’espace de modélisation graphique, sinon il/elle sera en lecture seule.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque :</strong> Cette option n’est disponible que lorsque vous sélectionnez la dernière version du Job ou de la Route.</td>
</tr>
<tr>
<td>Run job /Route</td>
<td>Exécuter le Job ou la Route.</td>
</tr>
<tr>
<td>Generate Doc As HTML</td>
<td>Générer le détail des informations du Job ou de la Route.</td>
</tr>
</tbody>
</table>

### Supprimer une version d’un Job ou d’une Route

Si une version d’un Job ou d’une Route ne vous est plus utile, vous pouvez la supprimer en effaçant ses fichiers de ressources.

**Avertissement:**

- Un Job ainsi supprimé ou une Route ainsi supprimée ne va pas dans la Corbeille. Sa restauration est donc impossible.
- La mauvaise suppression d’un fichier de ressources peut endommager l’intégrité du Job correspondant ou de la Route correspondante et l’empêcher de fonctionner.

**Procédure**

1. Si vous souhaitez supprimer la dernière version d’un Job ouvert ou d’une Route ouverte, fermez son onglet d’abord.
2. Sélectionnez *Window > Show view...* dans le menu puis, dans la boîte de dialogue *Show View*, sélectionnez *General > Navigator* et cliquez sur *OK* pour ouvrir la vue *Navigator* dans la zone de configuration.
   
   Ignorez cette étape si la vue *Navigator* est déjà affichée.
3. Dans la vue *Navigator*, développez le nœud nommé d’après votre projet.
   
   Le nom du nœud est en lettres capitales, *MY_PROJECT* par exemple.
4. Allez dans le dossier *process* pour afficher les fichiers de ressources de votre Job.
   
   Pour supprimer une version d’une Route, allez dans le dossier *routes*.
   
   Si votre Job ou votre Route se trouve dans un sous-dossier, allez dans ce sous-dossier pour afficher les fichiers de ressources correspondant.
5. Sélectionnez les trois fichiers de ressources correspondant au nom de votre Job ou Route ainsi que la version à supprimer, cliquez-droit sur la sélection et sélectionnez *Delete* dans le menu contextuel. Cliquez sur *OK* dans la boîte de dialogue *Delete Resources*.
**Exemple**

Pour supprimer la version 0.1 d'un Job nommé *my_job*, supprimez ces fichiers :

- `my_job_0.1.item`
- `my_job_0.1.properties`
- `my_job_0.1.screenshot`

**Documentation du Job ou d'une Route**

Le Studio Talend vous permet de générer une documentation fournissant des informations générales concernant vos projets, vos Routes, vos Jobs ou vos Joblets. Vous pouvez automatiser la génération d'une telle documentation et éditer n'importe quel document généré.

**Générer la documentation HTML**

Le Studio Talend vous permet de générer de la documentation détaillée au format HTML des Jobs ou des Routes sélectionné(e)s dans le *Repository* de votre Studio, dans la perspective *Integration*. Cette auto-documentation comprend :

- les propriétés du projet dans lequel les Jobs ou les Routes ont été créés,
- les propriétés et les paramètres des Jobs ou des Routes sélectionnés ainsi qu'une image de chaque Job ou Route,
- la liste de tous les composants utilisés dans chaque Job ou Route sélectionné(e), ainsi que les paramètres de ces composants.

Pour générer un document HTML pour un Job ou une Route, procédez comme suit :

**Procédure**

1. Dans le *Repository*, cliquez-droit sur l'un de vos *Jobs* ou *Route* ou sélectionnez plusieurs éléments pour générer une documentation multiple.
2. Sélectionnez *Generate Doc as HTML* dans le menu contextuel.
3. Renseignez le répertoire dans lequel doit être enregistré la documentation générée.
4. Dans le même champ, renseignez le nom de l’archive regroupant les documents générés.
5. Si vous souhaitez utiliser un fichier CSS, cochez la case **Use CSS file as a template to export** afin d’activer le champ **CSS File**.
7. Cliquez sur **Finish** pour valider.

**Résultats**

Le fichier archive contient les fichiers HTML ainsi que les dossiers correspondants. Ouvrez le fichier HTML dans votre navigateur Web. Ouvrez le fichier HTML dans votre navigateur Web.

**Mettre la documentation à jour**

Vous pouvez mettre manuellement à jour votre documentation.

**Procédure**

- Pour mettre un seul document à jour : cliquez-droit sur la documentation ou de la Route que vous souhaitez mettre à jour et sélectionnez **Update documentation**.
Comparaison des Jobs

Le Studio Talend fournit l'option **Compare Job** qui vous permet de comparer des Jobs sur la même branche ou sur différentes branches, afin de lister les différences entre les éléments utilisés dans les deux Jobs. En utilisant cette option, vous pouvez :

- Comparer les différentes versions du même Job.
- Comparer le même Job dans différentes versions du Studio, pour voir si des modifications ont été apportées au Job dans les versions précédentes ou actuelle.
- Comparer les Jobs créés à partir du même template, mais comprenant des paramètres différents, afin de contrôler les différences entre les Jobs.

Les différences entre les Jobs comparés sont affichées dans la vue **Compare Result**. Le détail des résultats s’affiche sous trois catégories : **Jobsettings**, **Components** et **Connectors**.

Le tableau ci-dessous donne la description des résultats de comparaison de chaque catégorie ci-dessus.

<table>
<thead>
<tr>
<th>Catégorie</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobsettings</td>
<td>liste toutes les différences relatives aux paramètres du Job comparé.</td>
</tr>
<tr>
<td>Components</td>
<td>liste les différences dans les composants et le paramètre de composant utilisé dans les deux Jobs. Un signe moins s'affiche en haut du composant listé dans la vue <strong>Compare Result</strong> ce qui indique que ce composant manque dans le design d'un des deux Jobs comparés. Un signe plus s'affiche en haut du composant listé dans la vue ce qui indique que ce composant a été ajouté à l'un des deux Jobs comparés. Toutes les différences dans les paramètres de composant seront listées dans des tables qui s'afficheront sous le composant correspondant.</td>
</tr>
<tr>
<td>Connectors</td>
<td>liste les différences dans les liens utilisés pour connecter les composants de deux Jobs.</td>
</tr>
</tbody>
</table>

Le processus de comparaison de deux Jobs ou de deux versions différentes d'un même Job est identique.

Pour comparer deux versions différentes d'un même Job, faites comme suit :

**Procédure**

La vue du *Compare Result* s’affiche dans l’espace de travail du Studio. Le nom du Job sélectionné et la version apparaissent, par défaut, dans les champs correspondants.

2. Si l’autre version du Job à laquelle vous souhaitez comparer la version actuelle se trouve dans une autre branche, sélectionnez la branche dans la liste *Another Branch*.

3. Cliquez sur le bouton [...] à côté du champ *Another job*, et sélectionnez dans la liste le Job auquel vous souhaitez comparer le Job actuel. Dans cet exemple, il s’agit du même Job.

4. Dans le champ *Name Filter*, saisissez le nom du Job ou du Joblet que vous souhaitez utiliser pour cette comparaison. La boîte de dialogue vous retourne le Job ou Joblet que vous cherchez.

5. Sélectionnez le Job ou Joblet retourné dans la liste et cliquez sur *OK*.


7. Cliquez sur le bouton pour lancer le processus d’exécution.
Les deux versions du Job sélectionnées s’affichent dans l’espace de modélisation graphique.

Les différences entre les deux versions sont listées dans la vue **Compare Result**.

**Résultats**

Dans cet exemple, les différences entre les deux versions du Job sont relatives aux composants et aux liens (connecteurs). La capture d’écran ci-dessous montre les différents composants utilisés dans les deux versions.
Par exemple, il y a une différence dans les schémas de sortie utilisés pour les composants tMap et tFileOutputXML : la longueur de la colonne Revenue est de 15 dans la deuxième version du Job alors qu'elle est de 11 dans la première version du même Job. Le signe moins s'affiche au-dessus du composant tMysqlOutput, ce qui indique que le composant manque à la création d'un des deux Jobs comparés. Le signe plus s'affiche au-dessus du composant tOracleOutput, ce qui indique que ce composant a été ajouté à l'un des deux Jobs comparés.

**Remarque :** Si vous cliquez sur un composant listé dans la vue Compare Result, le composant sera automatiquement sélectionné, donc identifié, dans le Job ouvert dans l'espace de modélisation graphique.

La capture d'écran ci-dessous montre les différences dans les liens utilisés pour relier les composants dans les versions du même Job.
Dans cet exemple, il y a une différence liée au lien `reject` utilisé dans les deux versions : la cible de ce lien dans la première version est un composant `tMysqlOutput`, alors qu’il s’agit d’un composant `tOracleOutput` dans la deuxième version.

**Remarque:** Vous pouvez exporter les résultats de la comparaison des Jobs dans un fichier HTML, en cliquant sur `Export to html`. Parcourez ensuite votre répertoire jusqu’à l’emplacement où vous souhaitez enregistrer le fichier et saisissez un nom pour ce fichier. Vous pouvez utiliser un modèle CSS par défaut ou personnalisé. Le dossier de destination contiendra le fichier HTML, un fichier CSS, un fichier XML et un dossier d’images. Pour un scénario associé, consultez Exporter les résultats de l’analyse d’impact/du lignage de données au format HTML.

### Exécution des Jobs et des Routes

Vous disposez de plusieurs manières pour exécuter votre Job. Le choix du type d’exécution dépend de l’objectif ainsi que de votre niveau utilisateur.

Cette section présente :

- Exécuter un Job ou une Route en mode normal à la page 199
- Exécuter un Job ou une Route en mode Java debug à la page 200
- Exécuter un Job en mode Traces Debug à la page 201
- Configurer les paramètres d’exécution avancés à la page 203
- Montrer l’utilisation des ressources de la JVM durant une exécution de Job ou de Route à la page 206
- Déploiement d’un Job sur un serveur SpagoBI (déprécié) à la page 207

#### Exécuter un Job ou une Route en mode normal

**Remarque:**

Veillez à sauvegarder votre Job ou Route avant de l’exécuter, afin que toutes les propriétés puissent être prises en compte.

Pour exécuter votre Job ou Route en mode normal, procédez comme suit :

2. Cliquez sur l’onglet Run Job pour accéder au mode normal d’exécution.
3. Dans la zone **Context**, sélectionnez le contexte approprié pour l’exécution de ce Job ou de cette Route. Vérifiez également les valeurs de variables si nécessaire.

Si vous n’avez pas défini de contexte d’exécution, le tableau des paramètres de contexte est vide et le contexte est celui par défaut. Pour plus d’informations, consultez Utiliser les contextes et les variables à la page 88.

1. Cliquez sur **Run** pour lancer l’exécution.
2. Sur le même panneau, la console de log affiche la progression de l’exécution. La console inclut les messages d’erreurs ainsi que les messages de début et de fin de processus.
3. Pour paramétrer le nombre de lignes à afficher dans la console lors de l’avancement de l’exécution, cochez la case **Line limit** et saisissez une valeur dans le champ.
4. Cochez la case **Wrap** pour activer les retours automatiques à la ligne. Cette case est sélectionnée par défaut. Lorsqu’elle est décochée, une barre de défilement horizontale s’affiche, vous permettant de voir la fin des lignes.

Avant d’exécuter à nouveau un Job, vous pouvez vider le contenu de la vue de log. Pour ce faire, cliquez sur le bouton **Clear**.

Si pour une raison quelconque, vous souhaitez stopper la progression du Job, cliquez simplement sur le bouton **Kill**. Vous devrez cliquer sur **Run** à nouveau pour reprendre l’exécution du Job.

Le Studio Talend offre d’autres fonctionnalités informatives, notamment Statistics et Traces, qui facilitent la supervision du Job ainsi que le travail de débogage. Pour plus d’informations, consultez les sections suivantes.

**Exécuter un Job ou une Route en mode Java debug**

**Pourquoi et quand exécuter cette tâche**

Pour suivre pas à pas les étapes de l’exécution d’un Job ou d’une Route afin d’identifier les bugs possibles, vous pouvez exécuter ce Job en mode Debug.

Avant d’exécuter votre Job en mode Debug, vous pouvez ajouter des points de pause au niveau des étapes principales de votre Job. Cela vous permettra d’arrêter automatiquement le Job à chaque point de pause. De cette manière, le processus s’arrêtera automatiquement régulièrement, vous permettant ainsi de vérifier pas à pas les composants et leurs variables respectives et de corriger les bugs éventuels.
**Remarque:** Vous pouvez ajouter des points de pause dans les Jobs, mais pas dans les Routes.

Pour ajouter des points de pauses au niveau d’un composant, cliquez-droit sur le composant dans l’espace de modélisation puis sélectionnez **Add breakpoint** dans le menu contextuel.

L’icône de pause s’affiche à gauche du composant dans l’éditeur graphique.

Pour accéder au mode Debug :

**Procédure**

2. Cliquez sur l’onglet **Debug Run** pour accéder aux modes d’exécution Debug.

   Pour passer en mode Debug, cliquez sur le bouton **Java Debug** de l’onglet **DebugRun** du panneau **Run**. La fenêtre principale du Studio Talend est alors réorganisée pour le débogage.

**Résultats**

Vous pouvez ensuite exécuter le Job étape par étape et, si vous avez ajouté des points de pause, vérifiez chaque point de pause afin de constater que le comportement et les valeurs des variables sont attendus.

**Exécuter un Job en mode Traces Debug**

La fonctionnalité Traces permet un suivi du traitement des données au cours de l’exécution du Job dans la perspective **Integration** du Studio Talend.

Cette option fournit un aperçu ligne par ligne du comportement du composant et affiche le résultat dynamique de cette vue à côté de la connexion **Row**.

Cette fonctionnalité vous permet de surveiller tous les composants d’un Job, sans avoir besoin de passer en mode Debug, et par conséquent sans avoir besoin de connaissances Java particulières.

La fonction **Traces** affiche le contenu des lignes traitées dans un tableau.

**Remarque:**

Cette option est disponible pour tous les composants, à l’exception des composants externes, qui ne peuvent offrir cette fonctionnalité que si leur conception le prévoit.
Vous pouvez activer ou désactiver le mode **Traces** ou décider quelles colonnes traitées afficher dans le tableau des traces qui s'affichera dans l'espace de modélisation graphique lors de l'exécution du Job.

**Pour activer le mode Traces dans un Job :**

1. Cliquez sur la vue **Run**.
2. Cliquez sur l'onglet **Debug Run** pour accéder aux modes d'exécution Debug et Traces.
3. Cliquez sur la flèche descendante du bouton **Java Debug** puis sélectionnez l'option **Traces**. Une icône apparaît sous chaque flux de votre Job pour indiquer que la fonction de suivi du traitement est activée.
4. Cliquez sur **Traces Debug** pour exécuter le Job en mode Traces.

**Pour désactiver le mode Traces d’un flux de votre Job :**

2. Sélectionnez **Disable Traces** dans le menu contextuel. Un signe moins rouge vient remplacer le signe plus vert sur l’icône pour indiquer que le mode **Traces** a été désactivé pour ce flux.

**Pour choisir quelles colonnes des données traitées afficher dans le tableau des traces, procédez comme suit :**

1. Cliquez-droit sur l’icône de **Traces** du flux concerné et sélectionnez **Setup Traces** dans le menu contextuel. La boîte de dialogue **Setup Traces** s'ouvre.

2. Dans la boîte de dialogue, décochez les cases correspondant aux colonnes que vous ne souhaitez pas afficher dans le tableau Traces.
3. Cliquez sur OK pour fermer la boîte de dialogue.

La fonction Traces ne s’exécute qu’à l’exécution du Job et s’arrête à la fin de celui-ci.
Cliquez sur le bouton Clear dans l’onglet Debug Run pour effacer les statistiques affichées.

**Configurer les paramètres d’exécution avancés**

Dans l’onglet Advanced settings de la vue Run, plusieurs paramètres d’exécution avancés sont disponibles afin de rendre l’exécution des Jobs ou Route plus pratique.

**Afficher les Statistiques**

La fonction Statistics affiche pour chacun des composants son taux de performance, en dessous des connexions dans l’espace de modélisation.

Sont indiqués le nombre de lignes traitées et la vitesse de traitement en ligne par seconde.

Pour les liens de déclenchement de type If, OnComponentOk, OnComponentError, OnSubjobOk et OnSubjobError, l’option Statistics affiche l’état de ce déclenchement durant l’exécution de votre Job : Ok ou Error et True ou False.

**Remarque:** Cette option est disponible pour tous les composants, à l’exception des composants externes, qui ne peuvent offrir cette fonctionnalité que si leur conception le prévoit.
Services de données

Procédure

- Dans la vue Run, cliquez sur l’onglet Advanced settings et cochez la case Statistics pour activer la fonctionnalité de statistiques ou décochez la case pour la désactiver.

Le calcul des statistiques ne commence qu’au début de l’exécution du Job ou de la Route et s’arrête lorsque l’exécution s’arrête.

Remarque: Les statistiques peuvent ralentir sensiblement les performances d’exécution du Job ou de la Route car le Job ou la Route doit envoyer ces données à l’application afin qu’elles soient affichées.

- Cliquez sur le bouton Clear dans la vue Basic ou Debug Run pour supprimer les statistiques calculées affichées.

- Cochez la case Clear before Run pour que la fonction de statistiques se réinitialise automatiquement avant chaque exécution.

Afficher la durée d’exécution et d’autres options

Procédure

- Pour afficher le temps total d’exécution, dans l’onglet Advanced settings de la vue Run, cochez la case Exec time avant d’exécuter votre Job ou Route.

De cette manière vous pouvez vérifier les résultats ou tester votre Job ou votre Route avant de l’envoyer en production.

- Pour nettoyer votre espace de modélisation avant chaque exécution de Job ou Route, cochez la case Clear before run.

- Pour enregistrer votre Job ou votre Route avant que l’exécution ne commence, cochez la case Save Job before run.

Afficher des caractères spéciaux dans la console

Pourquoi et quand exécuter cette tâche

Le Studio Talend peut afficher des caractères spéciaux dans la console. Pour activer l’affichage des caractères chinois, japonais ou coréens, par exemple, procédez comme suit avant d’exécuter le Job ou la Route :
Procédure

1. Cliquez sur l’onglet **Advanced settings**.
2. Dans la zone **JVM settings**, cochez la case **Use specific JVM arguments** afin d’activer le tableau **Argument**.
3. À côté du tableau **Argument**, cliquez sur le bouton **New** afin d’ouvrir la boîte de dialogue **Set the VM argument**.
4. Dans la boîte de dialogue, saisissez **-Dfile.encoding=UTF-8**.
5. Cliquez sur **OK** pour fermer la boîte de dialogue.

**Spécifier les limites de mémoire de la machine virtuelle pour un Job ou une Route**
Dans le Studio Talend, vous pouvez définir les paramètres de votre JVM avant d’exécuter votre Job ou Route selon vos besoins.

Les paramètres par défaut **-Xms256M** et **-Xmx1024M** correspondent respectivement aux tailles minimale et maximale de mémoire réservées pour vos exécutions de Jobs ou Routes. Modifiez ces paramètres selon vos besoins.

Pour spécifier ces paramètres de manière globale dans le Studio Talend, consultez ** Préférences d’exécution et de débogage (Talend > Run/Debug)** à la page 513.

Procédure

1. Dans l’onglet **Advanced settings** de la vue **Run**, cochez la case **Use specific JVM arguments**.
2. Cliquez sur le bouton **New** et, dans la boîte de dialogue **Set the VM Argument** qui s’ouvre, saisissez l’argument à utiliser.
   Par exemple, pour exécuter avec succès un Job gérant un fichier Excel contenant un million d’enregistrements, vous devez spécifier **-Xmx8192M** pour augmenter la taille de mémoire de la machine virtuelle à 8 Go.
3. Cliquez sur **OK** pour ajouter l’argument.
**Personnaliser le niveau de sortie du log4j à l’exécution**

**Pourquoi et quand exécuter cette tâche**

À l’exécution et lorsqu’il est activé pour les composants, log4j, l’utilitaire de log d’Apache, écrit en sortie les informations de log relatives aux composants. Par défaut, tous les messages de log supérieurs ou égaux au niveau de log défini dans la configuration du log4j sont écrits en sortie dans la cible définie.

Vous pouvez changer le niveau de sortie de log pour l’exécution d’un Job ou d’une Route. Pour ce faire, procédez comme suit :

**Procédure**

1. Dans la vue Run, cliquez sur l’onglet Advanced settings.
2. Cochez la case log4jLevel et sélectionnez le niveau de sortie de log souhaité dans la liste déroulante.
   - Cette case ne s’affiche que lorsque le log4j est activé pour les composants.
   - Pour plus d’informations concernant les niveaux de log, consultez la documentation Apache sur http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/Level.html (en anglais).
3. Exécutez votre Job ou Route.

**Résultats**

Tous les messages de log supérieurs ou égaux au niveau de log défini sont écrits en sortie dans la cible définie.

Pour plus d’informations concernant l’activation de log4j pour les composants et la configuration globale des comportements de log, consultez Configurer Log4j à la page 486.

Pour plus d’informations concernant les composants avec lesquels vous pouvez utiliser la fonctionnalité log4j, consultez Liste des composants supportant la fonctionnalité Log4j sur Talend Help Center (https://help.talend.com).

**Monitorer l’utilisation des ressources de la JVM durant une exécution de Job ou de Route**

**Pourquoi et quand exécuter cette tâche**

L’onglet Memory Run de la vue Run de votre Studio Talend vous permet de moniterer en temps réel l’utilisation des ressources de la JVM durant l’exécution d’un Job ou d’une Route, notamment la consommation de mémoire et l’utilisation de CPU hôte. Cela vous permet de prendre des décisions lorsque l’utilisation des ressources est trop importante et que votre Studio Talend subit une perte de performance. Vous pouvez augmenter la taille de la mémoire allouée à la JVM, arrêter des Jobs ou des Routes qui n’ont pas forcément à s’exécuter, etc.

Pour moniterer l’utilisation des ressources de la JVM lors de l’exécution d’un Job ou d’une Route, procédez comme suit :

**Procédure**

1. Ouvrez votre Job ou votre Route.
   - Dans la vue Run, cliquez sur l’onglet Memory Run.
2. Cliquez sur Run pour exécuter le Job ou la Route.
Vous pouvez cliquez sur le bouton **Run** de l’onglet **Memory Run** pour monitorer l’utilisation des ressources de la JVM par votre Job ou Route à tout moment, même après avoir lancé l’exécution du Job ou de la Route dans l’onglet **Basic Run**.

La console du Studio affiche des graphiques montrant l’utilisation des ressources de la JVM et du processeur, respectivement, durant l’exécution du Job ou de la Route. Des messages d’avertissement s’affichent en rouge dans la zone **Job execution information** lorsque les seuils sont atteints.

3. Pour voir des informations concernant les ressources utilisées à un moment donné de l’exécution du Job ou de la Route, placez votre curseur sur ce point, dans le graphique correspondant. Selon le graphique sur lequel vous placez votre curseur, vous pouvez voir les informations relatives à la taille du tas, le seuil des 90 % et des 70 % du tas, ou l’usage du CPU à un moment donné.

4. Pour exécuter le ramasse-miettes (Garbage Collector) durant un intervalle de temps spécifié, cochez la case **With Garbage Collector pace set to** et sélectionnez un intervalle en secondes. Le ramasse-miettes s’exécute automatiquement lors des périodes indiquées.

Pour exécuter le ramasse-miettes immédiatement, cliquez sur le bouton **Trigger GC**.

5. Pour exporter les informations de log dans un fichier texte, cliquez sur le bouton **Export** et sélectionnez un fichier pour sauvegarder le log.

6. Pour arrêter le Job, cliquez sur le bouton **Kill**.

**Déploiement d’un Job sur un serveur SpagoBI (déprécié)**

Cette fonctionnalité est dépréciée depuis la version 7.1 de Talend.

À partir de l’interface du Studio Talend, vous pouvez facilement déployer vos Jobs sur un serveur afin de les exécuter à partir de votre administrateur.

** Créer une nouvelle connexion au serveur SpagoBI**

**Pourquoi et quand exécuter cette tâche**

Avant toute chose, renseignez les informations concernant votre serveur simple ou multiple dans le Studio Talend.

**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.

2. Développez les noëuds **Talend > Import/Export** et sélectionnez **SpagoBI server (Deprecated)** pour afficher la vue adéquate.
3. Cochez la case **Enable/Disable Deploy on SpagoBI** pour activer le déploiement.

4. Cliquez sur **New** pour ouvrir la boîte de dialogue **Create new SpagoBI server** et ajouter un nouveau serveur à la liste des serveurs.

5. Renseignez les informations du serveur SpagoBI, comme décrit ci-dessous :

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine Name</td>
<td>Saisissez le nom interne utilisé dans le Studio Talend. Ce nom n'est pas utilisé dans le code généré.</td>
</tr>
<tr>
<td>Short description</td>
<td>Saisissez une description du serveur que vous êtes en train d'enregistrer.</td>
</tr>
<tr>
<td>Host</td>
<td>Adresse IP ou nom de l'hôte de la machine sur laquelle SpagoBI fonctionne.</td>
</tr>
<tr>
<td>Login</td>
<td>Identifiant requis pour vous connecter au serveur SpagoBI.</td>
</tr>
<tr>
<td>Password</td>
<td>Mot de passe pour vous connecter au serveur SpagoBI.</td>
</tr>
</tbody>
</table>

6. Cliquez sur **OK** pour valider ces informations et fermer la boîte de dialogue.
La nouvelle entrée est ajoutée au tableau des serveurs disponibles. Vous pouvez ajouter autant d’entrées que vous le souhaitez.

7. Cliquez sur **Apply** puis sur **OK** pour fermer la boîte de dialogue **Preferences**.

**Modifier ou supprimer un serveur SpagoBI**

**Procédure**

1. Pour supprimer une entrée obsolète, sélectionnez l’entrée dans le tableau, puis cliquez sur le bouton **Remove**.
2. Puis, si nécessaire, créez une nouvelle entrée contenant les informations mises à jour.

**Déployer vos Jobs sur un serveur SpagoBI**

**Pourquoi et quand exécuter cette tâche**

Suivez les étapes suivantes pour déployer vos Jobs sur un serveur :

1. **Edit job**
2. **Read job**
3. **Open another version**
4. **Open Job Hierarchy**
5. **Setup routine dependencies**
6. **Edit properties**
7. **Delete**
8. **Copy**
9. **Duplicate**
10. **Run job**
11. **Build Job**
12. **Deploy on SpagoBI**
13. **Generate Doc As HTML**
14. **Export items**
**Procédure**

1. À partir du Repository, développez le nœud Job Designs puis cliquez-droit sur le Job à déployer.
2. Dans la liste déroulante, sélectionnez Deploy on SpagoBI pour ouvrir la boîte de dialogue Deploy on SpagoBI.
3. Dans le champ SpagoBI server, sélectionnez le serveur correspondant dans la liste déroulante.
4. Les champs Label, Name et Description seront pré-remplis avec les propriétés principales définies lors de la création du Job.
5. Sélectionnez le contexte adéquat dans la liste.
6. Cliquez sur Finish lorsque vous avez terminé la configuration.

**Résultats**

Les Jobs sont maintenant déployés sur le serveur SpagoBI sélectionné. Ouvrez votre administrateur SpagoBI pour exécuter vos Jobs.

**Exécuter une Route ou un Job de Service de données dans un conteneur de test Runtime Talend local**

Le Studio Talend vous permet de configurer l’utilisation d’un Runtime Talend local dans le Studio pour tester les Routes et les Jobs de Service de données facilement, au sein de l’environnement du Runtime OSGI.

Pour configurer le Runtime de test :

1. Ouvrez la fenêtre Preferences.
2. Dans l’arborescence, cliquez sur Talend > Run/Debug > ESB Runtime pour afficher la vue correspondante.
3. La case **ESB Studio Runtime** est cochée par défaut, pour utiliser le Runtime Talend local. Si cette case est décochée, la Route ou le Job de Service de données sera exécuté(e) dans le Studio et non dans le Runtime local.

Cette option fonctionne uniquement pour les Routes et les Jobs de Services de données.

4. Dans la zone **Connection Information**, cliquez sur le bouton **Add Server...** pour ouvrir l’assistant **Add ESB Runtime Server**.

5. Choisissez l’une des méthodes suivantes pour sélectionner un Runtime à utiliser :

   - Si vous avez préparé un conteneur de Runtime que vous souhaitez utiliser, sélectionnez **Use an existing local Talend Runtime Container (ESB OSGi Container)**. Parcourez votre système ou saisissez le chemin d’accès au Runtime Talend, puis cliquez sur **Finish**. Ce chemin est renseigné dans le champ **Location** de la vue **ESB Runtime**. Vous pouvez également saisir le chemin d’accès au conteneur du Runtime dans le champ **Location**.

   - Si vous souhaitez utiliser une nouvelle archive du Runtime **Talend**, sélectionnez **Install the Runtime container from Talend Runtime (ZIP) or Talend ESB (ZIP)**. Parcourez votre système ou saisissez le chemin d’accès au Runtime **Talend** ou au fichier .zip ESB de **Talend**. Vous pouvez choisir d’installer le Runtime local dans le dossier cible par défaut (**default target folder**), le dossier `<Studio_Home>/esb/container`, ou dans un dossier cible personnalisé (**custom target folder**) de votre choix. Cliquez sur **Finish** pour fermer l’assistant. Le serveur du Runtime est extrait vers le dossier cible sélectionné. Ce chemin cible est renseigné dans le champ **Location** de la vue **ESB Runtime**.

   Un fichier de script d’initialisation **initlocal.sh** est créé dans le dossier `<LocalRuntimeContainerPath>/scripts` pour démarrer tous les bundles ESB, utiliser les gestionnaires
d’authentification JAAS pour le service de sécurité et démarrer un broker ActiveMQ dans le conteneur du Runtime. Vous pouvez modifier le script, si nécessaire.

```
tesb:start-all
tesb:switch-sts-jaas
feature:install activemq-broker
```

6. Dans la vue **ESB Runtime**, cliquez sur le bouton **Initialize...** afin de démarrer le serveur du Runtime et installer les bundles OSGI.

   Notez que vous pouvez également démarrer le serveur du Runtime à l’aide du fichier `trun` dans `<LocalRuntimeContainerPath>` :
   - `bin\trun.bat` (sous Windows)
   - `bin\trun` (sous UNIX)

7. Dans la vue **ESB Runtime**, si vous cochez la case **Filter system logs**, les informations de log Apache Karaf ne seront pas affichées lors de l’exécution des Routes ou des Jobs de Service de données sur le serveur du Runtime local.

8. Cliquez sur **OK** pour fermer la boîte de dialogue **Preferences**.

   Notez que si vous avez activé ou désactivé l’option **ESB Studio Runtime** dans cette vue, les paramètres du conteneur doivent fermer tous les éditeurs des artefacts ESB, avant de sauvegarder vos modifications.

Une fois le serveur du Runtime ajouté, vous pouvez exécuter votre Route ou votre Job de Service de données dans le Runtime local à des fins de test :

2. Cliquez sur le bouton **Run** de l’onglet **Basic Run**, pour déployer et exécuter la Route ou le Job de Service de données.

Vous pouvez arrêter le serveur du Runtime à partir de la console du Runtime :

1. Cliquez sur **Window > Show view** pour afficher la vue **Console** sous l’espace de modélisation graphique.
2. Cliquez sur le bouton **Console** dans la vue **Console** et sélectionnez **ESB Runtime** dans la liste, pour afficher la console du Runtime ESB. Vous pouvez exécuter les commandes du Runtime dans la console.
3. Cliquez sur 🗝 dans la vue Console pour arrêter le serveur du Runtime. Pour démarrer à nouveau le serveur du Runtime, cliquez sur 🛡️.

Notez que, si le Runtime Talend local dans le Studio est activé, lors de l'exécution d’une Route ou d’un Job de Service de données, le serveur du Runtime sera automatiquement démarré, s’il est arrêté.

**Utilisation de la parallélisation pour optimiser les performances des Jobs**

La parallélisation, en termes de Jobs Talend, signifie accomplir différents processus techniques via des exécutions parallèles. Lorsqu’il est bien conçu, un processus technique parallélisé peut être exécuté en moins de temps.

Le Studio Talend vous permet d’implémenter différents types de parallélisation selon les circonstances. Les circonstances peuvent être :

2. Itérations parallèles pour lecture de données. Pour plus d’informations, consultez Lancer des itérations parallèles pour lire des données à la page 214.

La parallélisation est une fonctionnalité avancée et nécessite des connaissances basiques concernant les Jobs Talend, notamment comment créer et exécuter un Job ou un sous-job, comment utiliser des composants et comment utiliser les différents types de liens reliant les composants ou les Jobs. Si vous ne possédez pas ces connaissances, consultez Qu’est-ce qu’un Job ? à la page 29.

**Exécuter plusieurs sous-jobs en parallèle**

La fonctionnalité Multi thread execution vous permet d’exécuter en parallèle différents sous-jobs actifs dans l’espace de modélisation.

Un Job ouvert dans l’espace de modélisation graphique peut contenir plusieurs sous-Jobs et vous pouvez personnaliser leur ordre d’exécution à l’aide des liens de déclenchement (Trigger), comme le lien OnSubjobOk. Cependant, il est possible de lancer les sous-jobs n’ayant aucune dépendance entre eux en même temps. Par exemple, l’image suivante présente quatre sous-jobs au sein d’un même Job sans aucune dépendance entre eux.
Plusieurs composants **tRunJob** sont utilisés dans cet exemple. Chacun appelle le sous-job qu’il représente.

Une fois le Job ouvert dans l’espace de modélisation graphique, procédez comme suit pour exécuter les sous-jobs en parallèle :

**Procédure**

1. Cliquez sur l’onglet **Job** puis sur l’onglet **Extra**.

   ![Exemple d’onglet Extra](image)

2. Cochez la case **Multi thread execution** pour activer l’exécution parallèle.
   
   Cette fonctionnalité est optimale lorsque le nombre de processus (en général un sous-job compte pour un processus) ne dépasse pas le nombre de processeurs de l’ordinateur utilisé pour ces exécutions parallèles. Sinon, certains des sous-jobs devront attendre qu’un processeur soit libre.

**Lancer des itérations parallèles pour lire des données**

Un lien **Iterate** permettant la parallélisation permet au composant recevant les threads de la connexion de les lire en parallèle.

**Avertissement**: Notez que **globalMap** n’est pas sécurisé. Soyez prudent(e) lorsque vous utilisez *globalMap.put(“key”, “value”) et globalMap.get(“key”)* pour créer vos propres variables globales et que vous récupérez leurs valeurs dans vos Jobs, notamment après un lien **Iterate**, avec l’option d’exécution parallèle activée.

**Pourquoi et quand exécuter cette tâche**

Procédez comme suit pour configurer les itérations parallèles :

**Procédure**

1. Sélectionnez le lien **Iterate** de votre sous-job afin d’afficher la vue **Basic settings** de l’onglet **Components**.

2. Cochez la case **Enable parallel execution** et configurez le nombre d’exécutions à effectuer en parallèle.

   ![Exemple d’onglet Extra](image)

Lorsque vous exécutez votre Job, le nombre d’itérations parallèles est distribué sur les processeurs disponibles.
3. Cochez la case **Statistics** de la vue **Run** afin d’afficher les exécutions parallèles en temps réel dans l’espace de modélisation graphique.

### Mapping de flux de données

#### Interfaces de mapping

Les composants de mapping sont des composants “avancés” qui requièrent plus d’informations de propriétés que les autres composants.

Les composants de mapping sont des composants avancés qui requièrent plus d’informations de propriétés que les autres composants Talend. En effet, le **Map Editor** est un outil complet vous permettant de définir tous les paramètres nécessaires au mapping, à la transformation et l’aiguillage des données dans votre processus, grâce à son interface graphique conviviale.

Vous pouvez réduire et restaurer les fenêtres de toutes les tables et la fenêtre **Map Editor** contenus dans le **Map Editor**, à l’aide des boutons dédiés situés dans le coin supérieur gauche de chacune des fenêtres.
La capture d'écran présente l'interface du tMap. Celle des autres composants de mapping a une apparence légèrement différente. Par exemple, en plus des onglets Schema editor et Expression editor dans la partie inférieure de l'interface, le tXMLMap contient un troisième onglet nommé Tree schema editor. Pour plus d'informations concernant le composant tXMLMap, consultez Présentation du fonctionnement du tXMLMap à la page 253.

Le Map Editor se compose de différentes zones :

- La zone d'entrée (Input), à gauche de l'éditeur. Elle offre une représentation graphique de tous les flux de données (Main et Lookup). Les données sont regroupées dans plusieurs colonnes des schémas Input. Notez que le nom de la table reflète le nom du lien Main ou Lookup dans l'espace de modélisation graphique du Job.

- La zone Variable panel est au centre de l'éditeur Map Editor. Ce tableau de variables permet de centraliser des informations redondantes et de réaliser les transformations.

- Le panneau de recherche (Search panel), au-dessus de la zone Variable. Il permet chercher dans l'éditeur des colonnes ou des expressions contenant le texte saisi dans le champ Find.

- La zone de sortie (Output), à droite de l'éditeur. Les tableaux Output permettent d'aiguiller les données et les champs provenant des schémas Input et des variables vers les schémas de sortie Output.

- Les deux panneaux situés en bas de la fenêtre sont les descriptions des schémas d'entrée et de sortie. L’onglet Schema editor propose une vue schématique de toutes les colonnes des tables Input et Output sélectionnées.

- L’onglet Expression editor est l'outil de rédaction des clés d’expression des schémas Input/Output, des variables ou des conditions de filtre.

Le nom des schémas Input/Output dans le Map Editor reflète le nom des connexions entrantes et sortantes (connexion de type Row).

Les sections suivantes présentent séparément les différents composants de mapping, pouvant chacun mapper des flux natures différentes.

**Présentation du fonctionnement du tMap**

Le tMap permet de réaliser les opérations suivantes :

- multiplexage et démultiplexage des données ;
- transformation des données sur tout type de champs ;
- concaténation et inversion de champs,
- filtrage de champs à l'aide de contraintes,
- gestion des rejets de données.

Comme toutes ces opérations de transformation et/ou routage sont réalisées par le tMap, ce composant ne peut être ni un composant de début ni un composant de fin de process dans un Job Design.
Le tMap utilise les connexions entrantes pour pré-remplir les schémas d’entrée dans le [Map Editor]. Par conséquent, vous ne pouvez pas créer directement dans le [Map Editor], de nouveaux schémas d’entrée. Par contre, il vous faut mettre en place autant de connexions de type Row entrantes que nécessaire, pour que leur schéma s’affiche dans le [Map Editor].

De la même façon, créez autant de connexions de sortie que nécessaire dans l’espace de modélisation. Cependant, vous pouvez définir le type de données en sortie directement dans le [Map Editor] via un outil graphique de mapping.

Il ne peut y avoir qu’une seule connexion de type Main row. Toute connexion entrante supplémentaire devient automatiquement une connexion Lookup. Pour plus d’informations, consultez Connexion de type Row à la page 78.


Bien que le tMap requiert que les connexions soient mises en place pour définir les flux d’entrée et de sortie (respectivement Input et Output), vous devez également implémenter un mapping pour que la fonction d’aperçu de l’éditeur de mapping soit disponible dans la vue Component de l’espace de modélisation. Vous devez créer le mapping dans votre Job afin d’afficher le Map Editor dans la zone Preview de l’onglet Basic settings du tMap.
Double-cliquez sur l’icône **tMap** dans l’espace de modélisation ou cliquez sur le bouton [...] à côté du champ **Map Editor** de l’onglet **Basic settings**, dans la vue **Component** du composant tMap.

Les sections suivantes vous donnent les informations nécessaires à l'utilisation du composant **tMap** dans vos Job Designs.

**Configuration du flux d’entrée dans le Map Editor**

L’ordre des tables **Input** (ou schémas) est essentiel. La première table reflète la connexion de flux principal (**Main** row), et pour cette raison, est traitée en priorité dans le composant **tMap**.

Par conséquent, vous ne pouvez pas déplacer cette table du flux **Main**. Cela assure qu’aucune jointure n’est perdue.
Vous pouvez utiliser les flèches haut et bas pour intervertir les tables secondaires (provenant de connexions **Lookup**), mais vérifiez qu’elles ne sont pas liées par un lien **Join**.

Pour plus d’informations, consultez **Utiliser les jointures explicites** à la page 221.
Renseigner les tables Input à l’aide d’un schéma

Pour renseigner les tables Input, définissez les schémas de tous les composants d’entrée connectés au composant tMap de votre Job, à partir du Map Editor.

Pour plus d’informations concernant la configuration d’un schéma d’un composant, consultez Définir les propriétés des composants à la page 55.

Pour plus d’informations concernant la configuration d’un schéma d’entrée dans le Map Editor, consultez Configuration des schémas dans le Map Editor à la page 241.

Contenu des tables Main et Lookup

L’ordre des tables Input (ou schémas) est essentiel.

La connexion Main Row détermine le contenu de la table Main. Ce flux entrant est représenté dans la première table de la zone Input du Map Editor.

Le contenu des connexions secondaires (Lookup) est représenté dans les autres tables apparaissant en dessous de la table Main. Si vous n’avez pas encore défini de schéma pour un composant d’entrée, la table correspondante du Map Editor apparaîtra comme vide.

La clé (Key) est également reprise du schéma défini dans les propriétés du composant Input. La clé (Key) est également reprise du schéma défini dans les propriétés du composant Input. La clé de hachage (interne au Map Editor) se différencie de la clé primaire en s’affichant dans une couleur différente.

Variables

Vous pouvez utiliser ou réutiliser les variables globales ou de contexte définies dans la zone Variables. Appuyez sur Ctrl+Espace pour accéder à la liste complète des variables, regroupant les variables globales, de contexte et de mapping. La liste des variables varie selon le contexte et s’accroît au fur et à mesure des nouvelles créations.

Seules les variables pour le mapping en cours sont affichées dans la liste d’auto-complétion. Une fenêtre de métadonnées est attachée à la liste de semi-complétion des variables.
Attachée à la liste **Variable**, une astuce relative aux métadonnées s'affiche pour fournir des informations concernant la colonne sélectionnée.

Consultez également : **Mapping de variables** à la page 226

**Utiliser les jointures explicites**

En effet, un lien **Joins** vous permet de sélectionner des données d'une table input via une autre table input. Dans le contexte d'un mapping dans le **Map Editor**, les données des tables **Main** et **Lookup** peuvent être liées à l'aide des clés d'expression (**expression keys**). C'est pourquoi l'ordre des tables a de l'importance.

Déposez simplement le nom des colonnes d'une table vers une table subordonnée, afin de créer une relation de jointure (**Join**) entre les deux tables. Ainsi, vous pouvez appeler des données à partir d'entrées multiples.

Les jointures apparaissent sous forme de liens violets et créent automatiquement une clé qui aura la même fonction qu'une clé de hachage, c'est-à-dire d'accélérer le processus de recherche.

Vous pouvez créer des Joins directs entre la table principale et les tables secondaires. Mais vous pouvez également créer des Joins indirects à partir d'une table principale vers une table secondaire, elle-même reliée à une autre table secondaire. Cela nécessite une jointure de l'une des tables **Lookup** à la table **Main**.

**Remarque:** Vous ne pouvez pas créer un **Join** à partir d'une table subordonnée vers une table de niveau supérieur de la zone **Input**.

Le champ **Expression key** peut être renseigné par glisser-déposer des données en Join. Ce champ est modifiable dans la table de la zone **Input** ou à partir de la vue Schema editor.

Lors de votre glisser-déposer, vous avez la possibilité soit d'insérer les données sélectionnées dans une nouvelle entrée, soit de remplacer une entrée existante ou encore de concaténer une sélection multiple dans une seule cellule.
Pour plus d’informations concernant les types de glisser-déposer possibles, consultez **Paramètres Output** à la page 235.

**Remarque:** Si vous avez beaucoup de tables input, vous pouvez les réduire ou les restaurer à l’aide de l’icône correspondante dans la zone **Input**. Par contre, le nom des colonnes est modifiable uniquement à partir du Schema editor, correspondant à la table Input sélectionnée.

Lors de votre glisser-déposer, vous avez la possibilité soit d’insérer les données sélectionnées dans une nouvelle entrée, soit de remplacer une entrée existante ou encore de concaténer une sélection multiple dans une seule cellule. Si vous avez beaucoup de tables input, vous pouvez les réduire ou les restaurer à l’aide de l’icône correspondante dans la zone Input.

Consultez également :

- Configuration des schémas dans le Map Editor à la page 241
- Utiliser la fonction Inner Join à la page 224
Avec une jointure explicite, vous pouvez choisir de mettre un filtre pour limiter le nombre de correspondances. Dans ce cas, vous pouvez choisir de prendre en considération uniquement la première correspondance, uniquement la dernière, ou toutes les correspondances.

Définir le modèle de rapprochement de la jointure explicite

Avant de commencer

Pour définir le modèle de rapprochement de la jointure explicite :

Procédure

1. Cliquez sur le bouton tMap settings situé en haut de la table de référence vers laquelle pointe la jointure pour afficher les propriétés de la table.
2. Cliquez dans le champ Value correspondant à Match Model puis cliquez sur le bouton [...] qui apparaît afin d’ouvrir la boîte de dialogue Options.
3. Dans la boîte de dialogue Options, double-cliquez sur le modèle souhaité ou sélectionnez-le et cliquez sur OK pour valider les paramètres et fermer la boîte de dialogue.

Unique Match (correspondance unique)

Elle correspond à l’option par défaut lorsque vous effectuez une jointure explicite. Cela signifie que seulement la dernière correspondance du flux secondaire sera prise en compte et passera dans la table de sortie.

Les autres correspondances seront donc ignorées.

First Match (première correspondance)

Cette option signifie que plusieurs correspondances peuvent être attendues dans le flux secondaire. L’option First signifie que seulement la première correspondance du flux secondaire sera prise en compte et passera dans le flux principal de sortie.

Les autres correspondances seront donc ignorées.
All Matches (toutes les correspondances)
Cette option signifie que plusieurs correspondances sont attendues dans le flux secondaire. Ainsi, toutes les correspondances sont prises en compte et passent dans le flux principal de sortie.

Utiliser la fonction Inner Join
La jointure interne Inner join est un type particulier de jointure qui se distingue par la façon dont les rejets sont traités.

Cette fonction empêche les valeurs nulles de passer dans le flux principal de sortie. Elle permet aussi de faire passer les données rejetées dans une table définie comme table de rejet Inner Join Reject.

Si les données cherchées ne peuvent être récupérées avec la jointure explicite ou un filtre de jointure (Inner join), en d’autres termes la jointure interne ne peut être établie quelle qu’en soit la raison, dans ce cas, les données demandées sont rejetées vers une table de sortie appelée Inner Join Reject.

Déposez simplement le nom des colonnes d’une table vers une table subordonnée, afin de créer une relation de jointure (Join) entre les deux tables. La jointure s’affiche graphiquement comme un lien violet, et crée automatiquement une clé qui sera utilisée comme une clé de hachage pour augmenter la vitesse de correspondance.

Pourquoi et quand exécuter cette tâche
Pour définir le type de jointure explicite :

Procédure
1. Cliquez sur le bouton tMap settings situé en haut de la table de référence vers laquelle pointe la jointure pour afficher les propriétés de la table.
2. Cliquez dans le champ Value correspondant à l’option Join Model pour faire apparaître le bouton [...] et cliquez dessus pour ouvrir la boîte de dialogue Options.
3. Dans la boîte de dialogue Options, double-cliquez sur le type de jointure souhaité, ou sélectionnez-le et cliquez sur OK pour activer l’option et fermer la boîte de dialogue.

Remarque: Une table Inner Join devrait toujours être associée à une table de rejet Inner Join Reject. Pour savoir comment définir une table de sortie comme table Inner Join Reject, consultez Utiliser les Rejets Inner Join à la page 238.
Vous pouvez utiliser le bouton de filtre pour diminuer le nombre de lignes à traiter et ainsi améliorer les performances.

Consultez également :
- Utiliser les Rejets Inner Join à la page 238.
- Filtrer un flux d’entrée à la page 225.

**Utiliser la fonction All Rows**

Par défaut, dans chaque table d’entrée de la zone d’entrée de l’éditeur du tMap, le modèle de correspondance All rows est sélectionné. Cette option All rows signifie que toutes les lignes sont chargées à partir du flux Lookup, et comparées au flux Main.

La sortie correspond au produit cartésien des deux tables (ou plus, selon vos besoins).

Rémarque: Si vous créez une jointure explicite ou une jointure Inner Join entre deux tables, l’option All rows n’est plus disponible. Vous devez choisir entre les modèles Unique match, First match et All matches. Pour plus d’informations, consultez Utiliser les jointures explicites à la page 221 et Utiliser la fonction Inner Join à la page 224.

**Filtrer un flux d’entrée**

Cliquez sur le bouton Filter à côté du bouton Inner join pour ajouter une zone de Filtre.

Dans la zone de Filtre, saisissez les conditions que vous souhaitez appliquer. Cela vous permet de réduire le nombre de lignes à traiter en fonction du flux principal et ainsi améliorer les performances sur des flux importants et hétérogènes.

Vous pouvez utiliser l’outil d’auto-complétion via le raccourci Ctrl+Espace pour reprendre les colonnes du schéma dans la déclaration des variables.

**Retirer des entrées de la table Input**

Pour enlever des entrées de la table Input, cliquez sur la croix rouge, en bas, dans le Schema Editor de la table sélectionnée. Appuyez sur Ctrl ou Maj pour sélectionner les champs/colonnes à retirer de la table.

Rémarque: Vous pouvez enlever des entrées Input d’un schéma dans le [Map Editor], cependant, cette suppression se répercutera sur la définition du schéma au niveau des propriétés du composant d’entrée.
Mapping de variables

La table Var regroupe toutes les variables de mapping qui peuvent être utilisées à différents endroits du [Map Editor].

Vous pouvez également utiliser le champ Expression de la table Var pour réaliser des transformations en langage Java.

Les variables vous permettent de gagner du temps et vous évitent d’avoir à ressaisir plusieurs fois les mêmes données.

Il existe plusieurs possibilités pour créer des variables :

- Saisissez librement vos variables en Java. Saisissez les chaînes de caractères entre guillemets simples ou concaténez plusieurs fonctions à l’aide de l’opérateur approprié.
- Ajoutez de nouvelles lignes à la table à l’aide du bouton [+] et retirez des lignes à l’aide du bouton [x]. Puis appuyez sur Ctrl+Espace pour récupérer les variables globales et de contexte déjà existantes.
- Déposez une ou plusieurs entrées de la table Input à la table Var.

Sélectionnez une entrée dans la zone Input ou appuyez sur la touche Maj pour effectuer une sélection multiple.

Appuyez sur la touche Ctrl pour sélectionner des entrées dispersées dans une table ou pour sélectionner des entrées de diverses tables. Lorsque vous sélectionnez plusieurs entrées, la première sélection peut prendre une couleur grisé. Maintenez la touche Ctrl enfoncée pour glisser-déposer toutes les entrées en une fois. Une bulle d’aide contextuelle vous indique le nombre d’entrées sélectionnées.

Plusieurs types de glisser-déposer sont possibles, notamment :

<table>
<thead>
<tr>
<th>Pour...</th>
<th>Actions associées</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insérer toutes les entrées sélectionnées comme variables séparées.</td>
<td>Faites simplement un glisser-déposer vers la table Var. Des flèches latérales vous indiquent où la nouvelle entrée Var peut être insérée. Chacune des Input est insérée dans une cellule séparée.</td>
</tr>
<tr>
<td>Concaténer toutes les entrées sélectionnées avec une entrée Var existante.</td>
<td>Déposez vers l’entrée Var à concaténer, elle se met alors en surbrillance. Toutes les entrées sont concaténnées dans une seule cellule. Ajoutez les opérateurs requis à l’aide des signes opérateurs Java appropriés. Le point permet la concaténation.</td>
</tr>
</tbody>
</table>
Pour... | Actions associées
---|---
Écrasez une entrée Var par les entrées Input sélectionnées, etconcaténez-les. | Puis appuyez sur Ctrl et relâchez. Toutes les entrées sélectionnées sontconcaténées et écrasent la variable sélectionnée.
Concaténer les entrées Input sélectionnées, avec plusieurs entrées Var mises en surbrillance. | Déposez vers des Var existantes puis appuyez sur Maj. Les premièresentrées sont concaténées avec les entrées Var en surbrillance, et sinécessaire, de nouvelles lignes sont créées pour contenir les entréesrestantes. Si nécessaire, de nouvelles lignes sont créées pour contenirles entrées restantes.

**Accéder aux variables globales ou de contexte**

Appuyez sur Ctrl+Espace pour accéder à la liste de semi-complétion des variables globales et de contexte.

Une fenêtre de métadonnées est annexée à la liste de semi-complétion des variables.

**Supprimer des variables**

Pour retirer une entrée Var sélectionnée, cliquez sur la croix rouge. La ligne entière est suppriméeainsi que tous les liens avec des tables Input ou Output.

Utilisez les touches Ctrl ou Maj pour effectuer une sélection multiple, puis cliquez sur la croix rougepour supprimer l’ensemble des lignes sélectionnées.

**Utilisation des expressions**

Toutes les déclarations d’expressions (Input, Var ou Output) et de filtres peuvent être visualisées etmodifiées directement dans les champs des expressions, dans l’éditeur d’expression.

**Accéder à l’Expression editor**

**Pourquoi et quand exécuter cette tâche**

Cet éditeur fournit le confort d’une fenêtre dédiée pour écrire des fonctions ou des commandes de transformation.

Vous pouvez écrire les expressions dont vous avez besoin pour la transformation de donnéesdirectement dans la vue Expression editor située dans la partie inférieure de l’Expression editor.

Pour ouvrir la vue Expression editor, procédez comme suit :

**Procédure**

1. Double-cliquez sur le composant tMap dans votre Job pour ouvrir le Map Editor.
2. Dans la partie inférieure de l’éditeur, cliquez sur l’onglet Expression editor pour ouvrir la vue correspondante.

**Remarque:** Pour éditer une expression, sélectionnez-la dans le panneau Input puis cliquez surl’onglet Expression editor et modifiez l’expression comme nécessaire.
3. Saisissez le code Java selon vos besoins. L’expression correspondante dans le panneau output est synchronisée.

Résultats

**Remarque:** Référez-vous à la documentation Java adéquate pour plus d’informations concernant les fonctions et les opérations.

**Créer du code à l’aide de l’Expression Builder**

Pour certains Jobs, il est nécessaire de rédiger du code afin de paramétrer les composants. Dans la vue **Component** de certains composants, une fenêtre **[Expression Builder]** peut vous aider à rédiger ce code (en Java), connu sous le nom d’expressions.

À l’aide de l’éditeur d’expressions du **tMap**, vous pouvez modifier l’expression pour une colonne d’entrée, une colonne de sortie ou une variable, ou modifier les expressions pour plusieurs colonnes de sortie en même temps.

**Modifier des expressions individuelles**

**Pourquoi et quand exécuter cette tâche**

L’exemple suivant vous montre l’utilisation de l’**Expression Builder** pour modifier deux expressions individuelles.

Dans cet exemple, deux flux d’entrée sont connectés au **tMap**.
• Un premier flux DB input, contenant une liste de personnes dont le nom et le prénom sont séparés par un espace.
• Un autre flux File input, contenant la liste des états des États-Unis, en minuscule.

Dans le **Map Editor**, utilisez l’Expression builder pour : 1) Remplacer l’espace entre le nom et le prénom par un tiret souligné ; 2) Mettre en majuscules le nom des états.

**Procédure**

1. Dans le **tMap**, créez la jointure Inner Join correspondante pour créer le mapping de référence. Pour plus d’informations concernant l’utilisation du **tMap**, consultez Présentation du fonctionnement du **tMap** à la page 216 et Interfaces de mapping à la page 215.

2. Déposez la colonne *Names* du flux d’entrée principal Main (*row1*) de la zone Input dans la zone Output, et la colonne *State* du flux d’entrée secondaire Lookup (*row2*) vers la même zone Output.

3. Cliquez sur le premier champ Expression (*row1.Name*) puis cliquez sur le bouton [...] qui s’affiche à côté de l’expression.

   ![Expression Builder](image)

   La fenêtre **[Expression Builder]** s’ouvre.

4. Dans la liste **Category**, sélectionnez l’action que vous souhaitez effectuer. Pour cet exemple, choisissez **StringHandling**, puis sélectionnez la fonction **ERREPLACE** dans la liste Functions.

5. Dans la zone Expression, collez *row1.Name* à la place du texte, pour obtenir :

   ```
   StringHandling.ERREPLACE(row1.Name," ", "-\_")
   ```

   Cette expression remplacera l’espace par un tiret souligné dans la chaîne de caractères fournie.

   Notez que les fonctions **CHANGE** et **ERREPLACE** de la catégorie **StringHandling** sont utilisées pour substituer toutes les sous-chaînes de caractères correspondant à l’expression régulière donnée.
dans l’ancienne chaîne de caractères, par le remplacement donné et retourner une nouvelle chaîne de caractères. Leurs trois paramètres sont :

- **oldStr** : l’ancienne chaîne de caractères.
- **newStr** : l’expression régulière avec laquelle effectuer la correspondance.
- **replacement** : la chaîne de caractères à substituer pour chaque correspondance.

6. Vérifiez votre code en renseignant le champ **Value** correspondant dans la zone **Test**, par exemple **Chuck Norris**, et cliquez sur **Test!**. Vérifiez que les changements ont bien été effectués, ici : **Chuck_Norris**.

7. Cliquez sur **OK** pour valider et répétez la même opération pour la deuxième colonne (**State**).

8. Dans le tMap de sortie, sélectionnez l’expression **row2.State** et cliquez sur le bouton [...] pour ouvrir de nouveau l’Expression builder.

Cette fois, la fonction **StringHandling** à utiliser est **UPCASE**. L’expression complète est :

```java
StringHandling.UPCASE(row2.State)
```

9. Vérifiez que la syntaxe de votre code est correcte, en saisissant par exemple **indiana** dans le champ **Value** de la zone **Test**. Cliquez sur **Test!** et le résultat affiché est **INDIANA**. Cliquez sur **OK** pour valider ces modifications.

Les deux expressions sont maintenant affichées dans le champ **tMap Expression**.

**Résultats**

Ces changements seront reportés tout au long du processus et le flux de sortie est affiché ci-dessous. La sortie de cet exemple s’affiche comme suit :

```java
230
```
Configurer des expressions pour plusieurs colonnes de sortie simultanément

Pourquoi et quand exécuter cette tâche

Le tMap vous permet de définir le comportement de transformation de plusieurs colonnes de sortie en même temps.

En utilisant un Job simple de transformation, l'exemple suivant vous montre comment définir les expressions sur plusieurs colonnes, à la manière d’un lot, dans le tMap.

Voici le contenu du fichier CSV d’entrée utilisé dans cet exemple :

```
id;firstname;lastname;city;state
1; Andrew; Adams; Madison; Rhode Island
2; Andrew; Garfield; Saint Paul; Colorado
3; Woodrow; Eisenhower; Juneau; New Hampshire
4; Woodrow; Jackson; Denver; Maine
5; Lyndon; Buchanan; Pierre; Kentucky
6; Bill; Tyler; Helena; New York
7; George; Adams; Oklahoma City; Alaska
8; Ulysses; Garfield; Santa Fe; Massachusetts
9; Thomas; Coolidge; Charleston; Mississippi
10; John; Polk; Carson City; Louisiana
```

Dans cet exemple, dans toutes les colonnes de sortie de type String, les espaces en début et fin de chaîne seront supprimés et les noms de famille et d’États seront passés en majuscules.

Procédure

1. Dans l’éditeur de mapping, effectuez les mappings de l’entrée vers la sortie.
2. Sélectionnez les colonnes de type String dans la table de sortie, respectivement firstname, lastname, city et state dans cet exemple, puis cliquez-droit sur la sélection, afin d’afficher le bouton Apply Routine.

3. Cliquez sur le bouton Apply Routine pour ouvrir la boîte de dialogue Expression Builder.
4. Sélectionnez StringHandling dans la zone Categories, puis double-cliquez sur la fonction TRIM dans la zone Functions, pour obtenir StringHandling.TRIM(${0}) dans le champ Expression.

5. Cliquez sur OK pour fermer la boîte de dialogue Expression Builder.

6. Sélectionnez les colonnes lastname et state dans la table de sortie de l’éditeur de mapping, cliquez-droit sur la sélection puis cliquez sur le bouton Apply Routine pour ouvrir la boîte de dialogue Expression Builder.

7. Sélectionnez StringHandling dans la zone Categories, puis double-cliquez sur la fonction UPPERCASE dans la zone Functions pour obtenir la fonction StringHandling(${0}) dans le champ Expression.
8. Cliquez sur OK pour fermer la boîte de dialogue Expression Builder.

Résultats

Les expressions de ces colonnes ressemblent à ceci :

<table>
<thead>
<tr>
<th>Expression</th>
<th>Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>StringHandling.TRIM(row1.firstname)</td>
<td>firstname</td>
</tr>
<tr>
<td>StringHandling.UPCASE(StringHandling.TRIM(row1.lastname))</td>
<td>lastname</td>
</tr>
<tr>
<td>StringHandling.TRIM(row1.city)</td>
<td>city</td>
</tr>
<tr>
<td>StringHandling.UPCASE(StringHandling.TRIM(row1.state))</td>
<td>state</td>
</tr>
</tbody>
</table>

Les fonctions seront exécutées au cours du traitement du flux. La sortie de cet exemple s’affiche comme suit :
Conseil:
Il n'y a pas d'ordre dans les flux de sortie du tMap. Pour exécuter les flux de sortie un par un, vous pouvez les écrire en sortie dans des fichiers temporaires ou dans la mémoire, puis les lire et les insérer dans des fichiers ou bases de données à l'aide de différents sous-jobs reliés par des liens Trigger > OnSubjobOK.

Dans l'espace de modélisation du Studio Talend, la création d'une connexion Row à partir du composant tMap vers des composants en sortie, a pour effet d'ajouter les schémas Output correspondants dans la zone Output du Map Editor.

Vous pouvez également ajouter un schéma Output dans votre [Map Editor], à l'aide du signe [+] de la barre d'outils de la zone Output.

Vous pouvez aussi effectuer une jointure entre vos tables de sortie. La jointure sur les tables de sortie permet de traiter les flux séparément, mais de les unifier en sortie.

Remarque: La table de jointure récupère le schéma de la table source.

Lorsque vous cliquez sur le bouton [+], afin d'ajouter un schéma Output ou d'effectuer une jointure entre vos tables de sortie, une boîte de dialogue s'ouvre, vous proposant deux possibilités. Sélectionnez...
A la différence de la zone Input, l’ordre des tables de schémas Output n’a pas une grande importance, car il n’existe aucune relation de subordination (Join) entre les sorties.

Une fois que vous avez créé toutes les connexions de sortie, et de ce fait, tous les schémas Output, vous pouvez sélectionner et organiser les données de ces sorties.

Déposez une ou plusieurs entrées à partir de la zone Input directement vers la table Output appropriée.

Appuyez sur Ctrl ou Maj pour une sélection multiple.

Ou vous pouvez utiliser des expressions de la zone Var par glisser-déposer dans la table Output avec les données réutilisables appropriées.

Notez que si vous apportez des modifications à la colonne Input du Schema Editor, une boîte de dialogue vous demande de confirmer la propagation des modifications sur toutes les entrées Input/Variable/Output concernées.

<table>
<thead>
<tr>
<th>Action</th>
<th>Résultat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glisser-déposer vers des expressions existantes</td>
<td>Concatène l’expression sélectionnée, vers des expressions existantes.</td>
</tr>
<tr>
<td>Glisser-déposer vers une ligne d’insertion</td>
<td>Insère une ou plusieurs nouvelle(s) entrée(s) en début ou en fin de table ou entre deux lignes existantes.</td>
</tr>
<tr>
<td>Glisser-déposer + Ctrl</td>
<td>Remplace les expressions en surbrillance par les expressions sélectionnées.</td>
</tr>
<tr>
<td>Glisser-déposer + Maj</td>
<td>Ajoute les champs sélectionnés à toutes les expressions en surbrillance. Insère de nouvelles lignes si nécessaire.</td>
</tr>
<tr>
<td>Glisser-déposer + Ctrl + Maj</td>
<td>Remplace toutes les expressions en surbrillance par les champs sélectionnés. Insère de nouvelles lignes si nécessaire.</td>
</tr>
</tbody>
</table>

Vous pouvez ajouter des filtres et un rejet pour personnaliser vos sorties.

**Créer des expressions complexes**

Si vous devez créer des expressions complexes ou faire des changements majeurs sur le flux de sortie, utilisez la fenêtre Expression Builder.

**Procédure**

1. Cliquez sur le champ Expression de votre table Input ou Output pour afficher le bouton [...].
2. Puis cliquez sur ce bouton pour ouvrir l’Expression Builder.

Pour plus d’informations concernant l’Expression Builder, consultez Créer du code à l’aide de l’Expression Builder à la page 228
Filtres

Les conditions de filtre vous permettent de sélectionner les champs et de les envoyer vers les sorties appropriées.

Cliquez sur le bouton pour ajouter une ligne de filtre.

Vous pouvez saisir librement vos conditions de filtre à l'aide d'opérateurs et de fonctions Java.

Déposez des expressions d'une zone Input ou d'une zone Var vers l'entrée de Filtre de la table Output appropriée.

Un lien graphique de couleur orange apparaît. Ajoutez l'opérateur Java nécessaire pour finaliser votre formule de filtre.

Vous pouvez créer plusieurs filtres sur différentes lignes. L'opérateur AND est la conjonction logique de tous les filtres formulés.

Utiliser la fonction Rejet

Pourquoi et quand exécuter cette tâche

L'option Reject définit la nature d’une table Output particulière.

Cette table de rejet rassemble toutes les données qui ne satisfont pas un ou plusieurs des filtres définis dans les tables Output standard. Notez que par table standard sont désignées toutes les tables qui ne sont pas des tables de rejet.

Ainsi les données rejetées des tables de sortie régulières sont regroupées dans une ou plusieurs tables dédiées, vous permettant par conséquent d'identifier les erreurs ou les cas imprévus.

Le principe de rejet (Reject) concatène tous les filtres des tables non-rejet et les définit comme formulation ELSE.

Pour définir une table de sortie comme la partie ELSE des tables régulières :

Procédure

1. Cliquez sur le bouton tMap settings en haut de la table de sortie pour afficher le tableau des propriétés.
2. Cliquez sur le champ Value correspondant à l’option Catch output reject et cliquez sur le bouton [...] qui apparaît pour ouvrir la boîte de dialogue Options.
3. Dans la boîte de dialogue **Options**, double-cliquez sur **true**, ou sélectionnez-le et cliquez sur **OK** pour activer l’option et fermer la boîte de dialogue.

![Options](image)

**Résultats**

Vous pouvez définir plusieurs tables Reject afin d’affiner les sorties multiples. Pour différencier les variantes de rejets, ajoutez des tables Reject, des lignes de filtre en cliquant sur la flèche au bouton [+].

Une fois qu’une table est définie comme table de rejet, le processus de vérification des données commencera par les tables régulières avant de prendre en considération les filtres possibles des tables de rejet.

Les données ne sont pas exclusivement traitées vers une seule sortie. Même si une donnée satisfait le filtre de sortie et qu’elle est donc routée vers elle, elle est également vérifiée contre les autres filtres et peut être également routée vers d’autres sorties.

**Utiliser les Rejets Inner Join**

**Pourquoi et quand exécuter cette tâche**

L’Inner Join est un Join Lookup. La table de rejet Inner Join est un type particulier de table de rejet de sortie. Elle regroupe les données rejetées de la table du flux principal lorsqu’un Inner Join n’a pu être établi.

Pour configurer un flux de sortie (Output) afin qu’il transmette les données de rejet d’un Inner Join, déposez un nouveau composant Output dans votre Job Design et connectez-le au **tMap**. Dans le **Map Editor**, suivez les étapes suivantes :

**Procédure**

1. Cliquez sur le bouton **tMap settings** en haut de la table de sortie pour afficher le tableau des propriétés.
2. Cliquez sur le champ **Value** correspondant à l’option **Catch lookup inner join reject** et cliquez sur le bouton [...] qui apparaît pour ouvrir la boîte de dialogue **Options**.
3. Dans la boîte de dialogue **Options**, double-cliquez sur **true**, ou sélectionnez-le et cliquez sur **OK** pour activer l’option et fermer la boîte de dialogue.
Retirer des entrées de la table Output

Pour retirer des entrées d’une table Output, cliquez sur la croix rouge, en bas, dans le Schema Editor de la table sélectionnée.

Gestion des erreurs

Pourquoi et quand exécuter cette tâche

L’option Die on error empêche le traitement des erreurs. Pour ce faire, elle arrête l’exécution du Job aussitôt qu’une erreur est rencontrée. Le composant tMap fournit cette option afin d’empêcher le traitement des données erronées. L’option Die on error est activée par défaut dans le tMap.

Désactiver l’option Die on error vous permettra d’ignorer les lignes en erreur et de terminer le processus pour les lignes sans erreur.

Pour désactiver l’option Die on error :

Procédure

1. Double-cliquez sur le composant tMap dans l’espace de modélisation pour ouvrir le Map Editor.
2. Cliquez sur le bouton Property Settings en haut de la zone input pour ouvrir la boîte de dialogue Property Settings.
3. Dans la boîte de dialogue Property Settings, décochez la case Die on error et cliquez sur OK.
Résultats
Une nouvelle table appelée ErrorReject s’affiche dans l’éditeur du tMap, dans la zone de sortie. Cette table de sortie comprend automatiquement deux colonnes : errorMessage et errorStackTrace, qui retrouvent le message et la trace de la pile de l’erreur rencontrée durant l’exécution du Job. Les erreurs peuvent être des erreurs de format de dates, des null pointer exceptions, des problèmes de conversion, etc.

Vous pouvez également glisser-déposer des colonnes (ici id et date) des tables d’entrée vers cette table de sortie de rejets. Ces données en erreur peuvent être retrouvées avec le message d’erreur correspondant et être corrigées ensuite.

Une fois la table ErrorReject définie, les flux correspondants peuvent être envoyés vers un composant de sortie.
Pour ce faire, dans l’espace de modélisation graphique, cliquez-droit sur le composant tMap, sélectionnez Row > ErrorReject dans le menu, et cliquez sur le composant de sortie correspondant, ici, le tLogRow.

Lorsque vous exécutez le Job, les erreurs sont retournées via le flux ErrorReject.

```
Starting job Die_on_error at 17:30 01/09/2010
java.text.ParseException: Unparseable date: "08 01
1980"|java.lang.RuntimeException:
java.text.ParseException: Unparseable date: "08 01 1980"
at routines.TalendDate.parseDate(TalendDate.java:503)
at doc.die_on_error_0_1.Die_on_error.tFileInputDelimited_2Pro
cess(Die_on_error.java:1409)
at doc.die_on_error_0_1.Die_on_error.runJobInTOS(Die_on_error.
java:2252)
at doc.die_on_error_0_1.Die_on_error.main(Die_on_error.java:2
160)
Caused by: java.text.ParseException: Unparseable date: "08
01 1980"
at java.text.SimpleDateFormat.parse(Unknown Source)
at routines.TalendDate.parseDate(TalendDate.java:501)
... 3 more
|1|08 01 1980
Job Die_on_error ended at 17:30 01/09/2010 [exit code=0]
```

Le résultat contient le message d’erreur, la trace de sa pile, et les deux colonnes id et date, glissées et déposées dans la table ErrorReject, séparées par une barre verticale "|".

**Configuration des schémas dans le Map Editor**

Dans le Map Editor, vous pouvez définir le type de schéma d’une table comme Built-In afin de pouvoir modifier la structure des données dans le panneau Schema editor, ou vous pouvez le définir comme Repository et récupérer la structure des données depuis le Repository. Par défaut, le type du schéma est défini comme Built-In pour toutes les tables.
Récupérer la structure d’un schéma depuis le Repository

Pourquoi et quand exécuter cette tâche

Pour récupérer la structure d’un schéma de la table sélectionnée dans le Repository :

Procédure

1. Cliquez sur le bouton tMap Settings en haut de la table pour afficher le tableau des propriétés.
2. Cliquez sur le champ Value correspondant à l’option Schema Type, et cliquez sur le bouton [...] qui apparaît pour ouvrir la boîte de dialogue Options.

3. Dans la boîte de dialogue Options, double-cliquez sur Repository, ou sélectionnez-le et cliquez sur OK pour fermer la boîte de dialogue et afficher une nouvelle option Schema Id au-dessus du champ Schema Type dans le tableau des propriétés.

   Remarque: Si vous fermez le Map Editor maintenant sans spécifier le schéma à utiliser, le type du schéma redevient Built-In.

4. Cliquez sur le champ Value de l’option Schema Id, et cliquez sur le bouton [...] qui apparaît dans la boîte de dialogue Repository Content.
5. Dans la boîte de dialogue Repository Content, sélectionnez votre schéma comme vous le faites pour n’importe quel autre composant et cliquez sur OK.

   Le champ Value de l’option Schema Id est renseigné avec le schéma que vous venez de sélectionner, et le schéma de cette table affiché dans le panneau Schema editor passe en lecture seule.
Avertissement: Si vous changez le type du schéma d'une table subordonnée contenant une jointure de Built-In à Repository, cette jointure sera perdue.

Remarque: Les modifications apportées au schéma d'une table à partir du Map Editor sont automatiquement répercutées sur le schéma du composant correspondant, relié au tMap.

Rechercher des colonnes de schéma

Pourquoi et quand exécuter cette tâche

Le filtre sur les colonnes du schéma du tMap vous permet de rechercher rapidement, en une fois, une colonne d’entrée ou de sortie parmi les centaines de colonnes.

L’exemple suivant vous montre comment trouver des colonnes contenant la chaîne de caractères “customer” dans la table de sortie, dans l’éditeur de mapping.

Procédure

1. Ouvrez l’éditeur de mapping et cliquez sur le bouton , en haut de la table, pour ouvrir la zone de filtre.
2. Dans la zone de filtre, saisissez votre recherche, *customer* dans cet exemple.
Lorsque vous commencez à saisir, la table affiche les colonnes correspondant aux caractères.

**Utiliser le Schema editor**

Le *Schema Editor* fournit les informations concernant les champs d’une table sélectionnée. Lorsque le type du schéma est défini comme *Built-in*, vous pouvez modifier ce schéma à partir du Schema Editor.
Utilisez la barre d'outils placée sous la table de schéma pour ajouter, déplacer et supprimer des colonnes du schéma.

Vous pouvez également charger un schéma à partir du référentiel ou exporter le schéma courant vers un fichier.

<table>
<thead>
<tr>
<th>Métadonnées</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column</td>
<td>Nom de colonne tel que défini dans le schéma du [Map Editor] et dans le schéma du composant Input ou Output correspondant.</td>
</tr>
<tr>
<td>Key</td>
<td>La clé indique si la valeur de la clé d’expression devrait être utilisée pour retrouver les données d’une autre table à travers un lien Join. Si cette case est décochée, la relation Join est désactivée.</td>
</tr>
<tr>
<td>Type</td>
<td>Type de données : Chaîne de caractères (String) ou entier (Integer). <strong>Remarque:</strong> Cette colonne est obligatoire dans la version Java.</td>
</tr>
<tr>
<td>Length</td>
<td>-1 indique qu’aucune valeur de longueur n’a été définie dans le schéma.</td>
</tr>
<tr>
<td>Precision</td>
<td>Définissez le nombre de chiffres après la virgule.</td>
</tr>
<tr>
<td>Nullable</td>
<td>Décochez cette case si vous ne souhaitez pas autoriser les valeurs de champs nulles.</td>
</tr>
<tr>
<td>Default</td>
<td>Indique les valeurs par défaut, si elles sont définies.</td>
</tr>
<tr>
<td>Comment</td>
<td>Champ texte libre. Saisissez tout commentaire que vous jugerez utile.</td>
</tr>
</tbody>
</table>

**Remarque:** Les schémas Input et Output sont indépendants les uns des autres. Vous pouvez par exemple changer le libellé d’une colonne Output sans que le libellé du schéma Input ne change.

Cependant, toute modification apportée au schéma est immédiatement reportée sur le schéma correspondant de la zone **Input** ou Output appropriée, mais également au niveau des propriétés elles-mêmes du composant Input ou Output concerné.

Un arrière-plan rouge s’affiche lorsqu’un caractère non valide a été saisi. La plupart des caractères spéciaux sont interdits pour réduire les risques de mauvaise interprétation dans le code. Les caractères autorisés sont : les minuscules et les majuscules, les chiffres (à l’exception du caractère de début de chaîne).

**Activation de la conversion automatique des types de données**

**Avant de commencer**

Lors du traitement des flux de données avec le **tMap**, si les colonnes d’entrée et de sortie dans le mapping sont de différents types de données, des erreurs de compilation peuvent survenir pendant l’exécution du Job. L’option **Enable Auto-Conversion of types** dans le **tMap** permet d’éviter ce type d’erreur.

Pour activer cette fonctionnalité dans un **tMap** dans un Job :

**Procédure**

1. Cliquez sur le bouton [ ] en haut de l’éditeur de mapping pour ouvrir la boîte de dialogue **Property Settings**.
2. Cochez la case **Enable Auto-Conversion of types** puis cliquez sur **OK**.

![](image)

**Que faire ensuite**

Vous pouvez activer l’option de conversion automatique au niveau du projet afin que cette fonctionnalité soit activée dans tous les composants **tMap** ajoutés par la suite au projet.

**Définir des règles pour écraser le comportement de conversion par défaut**

Si nécessaire, vous pouvez également définir des règles de conversion pour écraser le comportement de conversion par défaut du **tMap**.

**Procédure**

1. Cliquez sur ![fichier](image) dans la barre d’outils de la fenêtre du Studio ou cliquez sur **File > Edit Project Properties** dans la barre du menu pour ouvrir la boîte de dialogue **Project Settings**.
2. Dans l’arborescence de la boîte de dialogue, développez **General** et sélectionnez **Auto-Conversion of types** pour ouvrir cette vue.
3. Cochez la case **Enable Auto-Conversion of types** afin d’activer la fonctionnalité de conversion automatique du type pour tous les composants **tMap** ajoutés par la suite au projet.

4. Si nécessaire, cliquez sur le bouton **[+]** pour ajouter une ligne et sélectionnez les types source et cible. Définissez ensuite une fonction Java pour la conversion du type de données afin de créer une règle de conversion écrasant le comportement par défaut de conversion du **tMap** pour les données correspondant à la règle.

   Vous pouvez appuyer sur les touches **Ctrl+Espace** dans le champ **Conversion Function** pour accéder à une liste de fonctions Java disponibles.

   Dans cet exemple, la règle rapproche les mappings avec des données d’entrée de type String (chaîne de caractères) et des données de sortie de type Integer (entier).

   Vous pouvez créer autant de règles de conversion que vous le souhaitez.

5. Cliquez sur **Apply** pour appliquer vos changements, puis sur **OK** afin de fermer la boîte de dialogue.

**Résolution des problèmes de mémoire lors de l’utilisation du **tMap****

Lorsque vous devez traiter un nombre important de données, par exemple, de nombreuses colonnes, différents types de colonnes ou lignes, votre système peut rencontrer des problèmes de mémoire insuffisante empêchant votre Job de s’exécuter correctement, et plus particulièrement lorsque vous utilisez un composant **tMap** pour effectuer des transformations.

Une option (uniquement disponible en Java pour le moment) a été ajoutée au composant **tMap**, pour utiliser moins de mémoire lors du traitement des données de référence (lookup). En effet, au lieu de stocker les données de référence dans la mémoire système et ainsi en atteindre les limites, l’option **Store temp data** vous permet de stocker les données de référence dans un dossier temporaire sur votre disque dur.
Cette option peut être sélectionnée sur la table Lookup de la zone Input (à gauche) de l’éditeur du tMap, le [Map Editor].

Pour activer l’option **Store temp data**, suivez les étapes suivantes :

1. Double-cliquez sur le composant tMap de votre Job pour lancer le [Map Editor].
2. Dans le panneau de gauche, correspondant aux sources Input et Lookup, cliquez sur la table Lookup affichant les données de Lookup que vous souhaitez charger sur le disque pour ne pas encombrer la mémoire.
3. Cliquez sur le bouton tMap settings pour afficher le tableau des propriétés.
4. Cliquez sur le champ Value correspondant à l’option Store temp data, et cliquez sur le bouton [...] pour ouvrir la boîte de dialogue Options.
5. Dans la boîte de dialogue Options, double-cliquez sur true, ou sélectionnez-le et cliquez sur OK pour activer l’option et fermer la boîte de dialogue.

Pour que cette option soit totalement active, vous devez définir le répertoire de votre disque dans lequel les données seront stockées, et la taille du buffer, à savoir le nombre de lignes de données stockées dans chaque fichier temporaire. Vous pouvez spécifier le répertoire de stockage temporaire des données et la taille de la mémoire tampon (buffer) à la fois à partir du Map Editor et de la vue Component du tMap.

Pour paramétrer le répertoire de stockage temporaire et la taille du buffer à partir du Map Editor :

1. Cliquez sur le bouton Property Settings en haut de la zone input pour ouvrir la boîte de dialogue Property Settings.
2. Dans la boîte de dialogue Property Settings, renseignez le chemin d'accès complet vers le dossier dans lequel les données temporaire seront stockées dans le champ Temp data directory path.
3. Dans le champ Max buffer size (nb of rows), spécifiez le nombre maximum de lignes contenues dans chaque fichier temporaire. La valeur par défaut est de 2 000 000 lignes.
4. Cliquez sur OK pour valider les paramètres et fermer la fenêtre Property Settings.
Pour paramétrer le répertoire de stockage temporaire du composant tMap sans avoir à ouvrir le Map Editor :

1. Cliquez sur le composant tMap dans l’espace de modélisation pour le sélectionner, puis sélectionnez la vue Component pour afficher l’onglet Basic settings.

2. Dans la zone Store on disk, renseignez le chemin d’accès complet vers le dossier dans lequel les données temporaires seront stockées dans le champ Temp data directory path.

   Vous pouvez aussi utiliser une variable de contexte à l’aide du raccourci Ctrl+Espace si vous avez configuré une variable dans un groupe de contexte dans le Repository. Pour plus d’informations concernant les contextes, consultez Utiliser les contextes et les variables à la page 88.

   À la fin de votre sous-job, les fichiers temporaires seront nettoyés.

   Ainsi, vous limitez l’utilisation de mémoire allouée aux données de référence à écrire dans les fichiers temporaires stockés sur le disque.

**Remarque:** Étant donné que l’écriture du flux principal sur le disque nécessite le stockage de données, l’ordre des lignes en sortie peut ne pas être respecté.
Dans la vue **Advanced settings**, vous pouvez aussi ajouter un buffer, au besoin. Pour cela, renseignez le champ **Max. buffer size (Nb of rows)** afin de séparer les données stockées sur le disque en autant de fichiers que nécessaire.

### Gestion des références

Lorsque vous implémentez une jointure (notamment **Inner Join** et **Left Outer Join**) dans un **tMap**, sur différentes sources de données, il y a toujours un flux principal (Main) et un ou plusieurs flux de référence (Lookup) connectés au **tMap**. Tous les enregistrements du flux de référence doivent être chargés avant de traiter chaque enregistrement du flux principal. Trois types de modèles de chargement de Lookup sont fournis, correspondants à différents besoins métier et besoins en performances : **Load once, Reload at each row** et **Reload at each row (cache)**.

- **Load once** : charge une fois (une fois seulement) tous les enregistrements du flux de référence, soit dans la mémoire, soit dans un fichier local, avant de traiter chaque enregistrement du flux principal, si l’option **Store temp data** est configurée à true. Cette configuration est celle par défaut et est l’option recommandé si vous avez un jeu d’enregistrements volumineux à traiter dans le flux principal, à l’aide d’une jointure entre les deux flux.

- **Reload at each row** : charge tous les enregistrements du flux de référence pour chaque enregistrement du flux principal. Généralement, cette option augmente le temps d’exécution du Job, à cause du chargement répété des enregistrements du flux de référence à chaque enregistrement du flux principal. Cependant, cette option est recommandée dans les situations suivantes :
  - le flux de données de référence est constamment mis à jour et vous souhaitez charger les dernières données de référence pour chaque enregistrement du flux principal, pour obtenir les données les plus récentes après exécution de la jointure ;
  - les données sont peu nombreuses dans le flux principal et le flux de référence contient de nombreuses données d’une table de base de données. Dans ce cas, l’utilisation de l’option **Load once** peut causer une erreur de mémoire **OutOfMemory**. Vous pouvez utiliser des paramètres variables dynamiques comme une clause WHERE pour mettre à jour le flux de référence à la volée durant le chargement, avant traitement de la jointure du flux principal. Par exemple, consultez **Recharger des données à chaque ligne** à la page 251.

Notez que les options **Reload at each row** dans un Job Streaming sont supportées uniquement par les composants Lookup Input, comme le **tMongoDBLookupInput**.

- **Reload at each row (cache)** : fonctionne comme le modèle **Reload at each row**, tous les enregistrements du flux de référence sont chargés pour chaque enregistrement du flux principal. Cependant, ce modèle ne peut être utilisé avec l’option **Store temp data on disk**. Les données de référence sont en cache dans la mémoire et, lorsqu’un nouveau chargement survient, seuls les enregistrements qui n’existent pas déjà dans le cache seront chargés, pour éviter deux fois les mêmes enregistrements. Cette option permet d’optimiser le temps de traitement et d’améliorer les performances de traitement du composant **tMap**. Notez que vous ne pouvez pas utiliser les options **Reload at each row (cache)** et **Store temp data** en même temps.

Notez que, lorsque votre référence est une table de base de données, la bonne pratique est d’ouvrir la connexion au début du Job, afin d’optimiser les performances.

### Configurer le mode de chargement d’un flux de référence

#### Pourquoi et quand exécuter cette tâche

Pour configurer le mode de chargement d’un flux de référence :
Procédure

1. Cliquez sur le bouton **tMap settings** en haut à droite de la table de référence, pour afficher les propriétés de la table.

2. Cliquez sur le champ **Value** correspondant à l’option **Lookup Model**, puis cliquez sur le bouton [...] pour ouvrir la boîte de dialogue **Options**.

3. Dans la boîte de dialogue **Options**, double-cliquez sur le mode de chargement souhaité, ou sélectionnez-le et cliquez sur OK pour activer l’option et fermer la boîte de dialogue.

Résultats

Pour un exemple d’utilisation de ces options, consultez la documentation associée au composant **tMap**.

**Recharger des données à chaque ligne**

**Pourquoi et quand exécuter cette tâche**

L’option **Reload at each row** est utilisée pour charger tous les enregistrements d’un flux Lookup pour chaque enregistrement du flux principal.

Lorsque le flux principal contient moins de lignes que le flux de référence (par exemple, avec un ratio de 1000 ou plus) et que l’entrée de référence est un composant de base de données, l’avantage de cette approche est qu’elle permet de gérer des données de références de plus en plus nombreuses, puisque vous pouvez exécuter des requêtes par rapport aux données du flux principal dans le composant de base de données, pour sélectionner uniquement les données de référence correspondant à chaque enregistrement du flux principal.
Par exemple, ici, avec des données de référence d’une base de données MySQL.

Dans la base de données MySQL, vous pouvez sélectionner uniquement les données correspondant aux valeurs de la colonne `id` du flux principal. Pour ce faire, procédez comme suit :

**Procédure**

1. Double-cliquez sur le composant `tSetGlobalVar` pour ouvrir sa vue Component.
2. Cliquez sur le bouton [+] pour ajouter une ligne et, dans la colonne Key, saisissez id. Dans la colonne Value, saisissez row1.id.

3. Double-cliquez sur le tMysqlInput pour ouvrir sa vue Component.

4. Dans le champ Query, saisissez la requête pour sélectionner les données correspondant à la colonne id du flux principal. Dans cet exemple, la requête se présente comme suit:

   ```
   Select * from person where id="+(Integer)globalMap.get("id")
   ```

Résultats

Pour plus d’informations concernant les composants utilisés dans cet exemple, consultez la documentation associée.

Présentation du fonctionnement du tXMLMap

**Remarque:** Avant de lire les sections suivantes, il est recommandé de lire les sections précédentes concernant le tMap afin d’avoir les connaissances de base du composant Talend de mapping.

Le tXMLMap est conçu pour utiliser le type de données Document afin de traiter des données XML, en cas de transformation mélangant souvent des données hiérarchiques (XML) et des données plates. Le type Document contient un flux XML complet spécifique à l’utilisateur. En utilisant tXMLMap,
vous pouvez ajouter autant de flux d’entrée et de sortie que nécessaire dans un éditeur graphique de mapping dans lequel vous pouvez effectuer des opérations, comme :

• multiplexage et démultiplexage des données ;
• transformation des données sur tout type de champs, particulièrement sur le type Document,
• rapprochement de données via différents modes, par exemple le mode Unique match (consultez Utiliser les jointures explicites à la page 221),
• construction automatisée d’arbre XML des côtés d’entrée et de sortie,
• jointures Inner Join et Left Outer Join (consultez Utiliser la fonction Inner Join à la page 224),
• flux lookup entre tous types de sources de données, XML ou plates à l’aide de modèles comme Load once (consultez Gestion des références à la page 250),
• concaténation et inversion de champs,
• filtrage de champs à l’aide de contraintes,
• gestion des rejets de données.

Comme pour le tMap, un éditeur de mapping est requis pour configurer ces opérations. Pour ouvrir cet éditeur, vous pouvez double-cliquer sur l’icône du tXMLMap dans l’espace de modélisation graphique ou bien cliquer sur le bouton [...] à côté de Map Editor dans la vue Basic settings du composant tXMLMap.

Les composants tXMLMap et tMap ont un fonctionnement quasiment identique. Ainsi, les sections qui suivent ne couvriront que le fonctionnement spécifique au tXMLMap, à savoir le traitement des données hiérarchiques.

Les opérations relatives aux données hiérarchiques sont :

• l’utilisation du type Document.
• Pour plus d’informations, consultez.
• la gestion des données XML de sortie.

Pour plus d’informations, consultez.

Remarque: A la différence du tMap, le tXMLMap ne possède pas l’option Store temp data pour stocker les données temporaires dans un répertoire de votre disque. Pour plus d’informations concernant cette option du tMap, consultez Résolution des problèmes de mémoire lors de l’utilisation du tMap à la page 247.

Utiliser le type Document pour créer l’arborescence XML

Le type de données Document correspond parfaitement au concept de données structurées de type XML. Lorsque vous devez utiliser la structure XML pour mapper le flux d’entrée ou de sortie, ou les deux, utilisez ce type. Vous pouvez importer des arborescences XML de plusieurs sources XML et de les éditer directement dans son éditeur de mapping, afin de vous éviter de les éditer manuellement un à un.

Configurer le type Document

Pourquoi et quand exécuter cette tâche

Le type de données Document est un des types de données fournis par Talend. Le type Document se sélectionne lorsque vous définissez le schéma correspondant à vos données dans le panneau Schema editor. Pour plus d’informations concernant le Schema editor, consultez Utiliser le Schema editor à la page 244.
La capture d'écran ci-dessous montre un exemple de flux d'entrée, Customer, de type Document. Afin de le reproduire dans le Map editor, cliquez sur le bouton [+] pour ajouter une ligne du côté de l'entrée du Schema editor, renommez-la et sélectionnez Document dans la liste des types.

Dans la plupart des cas, le tXMLMap récupère le schéma du composant précédent ou suivant, par exemple, d'un tFileInputXML ou d'un scénario ESB, d'un tESBProviderRequest. Cela permet d'éviter les efforts manuels lors de la définition du type Document dans le flux XML à traiter. Cependant, pour continuer à modifier la structure XML ainsi que le contenu d'une ligne de type Document, vous devez utiliser le Map editor.

Remarque: Un flux Document comporte une arborescence XML personnalisée et représente un seul champ du schéma, pouvant contenir plusieurs champs de types différents. Pour plus d'informations concernant la configuration d'un schéma, consultez Onglet Basic settings à la page 55.

Une fois une ligne de données définie comme type Document dans la table du flux de données correspondante, une arborescence XML de base est créée automatiquement pour refléter les détails de la structure. L'arborescence représente le nombre d'éléments minimum requis pour une arborescence XML valide, dans le tXMLMap :

- Elément racine : il est l'élément requis par une arborescence XML à traiter et, si nécessaire, il est la base d'une arborescence XML plus sophistiquée.
- Élément boucle : il détermine l'élément sur lequel s'effectue l'itération, afin de lire les données hiérarchiques dans une arborescence XML. Par défaut, l'élément racine est défini comme l'élément répétable.
Cette capture d'écran vous montre un exemple du flux d'entrée XML, Customer. À partir de la racine (root) XML créée, vous pouvez développer l'arborescence XML qui vous intéresse.

Pour ce faire, vous devez :

**Procédure**

1. **Importer** l'arborescence XML personnalisée à partir de sources de type :
   - fichiers XML ou XSD (consultez Importer une structure XML à partir de fichiers XML et XSD à la page 258).
     
     **Remarque:** Lorsque vous importez un fichier XSD, vous créez la structure XML que décrit le fichier XSD.
     
     - connexions vers des fichiers XML créées et centralisées dans les métadonnées du **Repository** dans le Studio (consultez Importer une structure XML à partir du Repository à la page 259).

     **Remarque:** Si nécessaire, vous pouvez développer manuellement l’arborescence XML qui vous intéresse, à l’aide des options fournies dans le menu contextuel.

2. **Définir** l’élément répétable pour l’arborescence XML que vous créez. Vous pouvez définir autant de boucles que nécessaire. Pour cette étape, prenez en compte les situations suivantes :
   - Si vous avez créé plusieurs arborescences XML, vous devez définir un élément répétable pour chacune d’entre elles.

     Pour plus d’informations, consultez : Définir ou réinitialiser un élément répétable pour une structure XML créée à la page 260.

3. **Facultatif** : Si nécessaire, vous pouvez continuer à modifier l’arborescence XML à l’aide des options fournies dans le menu contextuel. Le tableau suivant présente les opérations que vous pouvez effectuer via ces options.

<table>
<thead>
<tr>
<th>Options</th>
<th>Opérations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Sub-element et Create Attribute</td>
<td>Ajoute des éléments ou des attributs à l’arborescence XML importée. Consultez également : Ajouter un sous-élément ou un attribut à une structure XML à la page 261</td>
</tr>
<tr>
<td>Set a namespace</td>
<td>Ajoute et gère des espaces de noms donnés dans la structure XML importée. Consultez également : Gérer un espace de noms à la page 262</td>
</tr>
<tr>
<td>Delete</td>
<td>Supprime un élément ou un attribut. Consultez également : Supprimer un élément ou un attribut depuis la structure XML à la page 262</td>
</tr>
<tr>
<td>Rename</td>
<td>Renomme un élément ou un attribut.</td>
</tr>
<tr>
<td>As loop element</td>
<td>Définit ou réinitialise un élément répétable. Les éléments répétibles multiples et optionnels sont supportés.</td>
</tr>
<tr>
<td>Options</td>
<td>Opérations</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>As optional loop</td>
<td>Cette option est disponible uniquement sur l’élément répétable que vous avez défini. Lorsque l’élément correspondant existe dans le fichier source, l’élément répétable facultatif fonctionne de la même manière qu’un élément répétable normal. Cependant, dans des cas d’utilisation réelle, avec des différences entre l’arborescence XML et la structure du fichier source, il est recommandé d’adapter l’arborescence XML au fichier source, pour améliorer les performances.</td>
</tr>
<tr>
<td>As group element</td>
<td>Dans la structure XML de sortie, définit un élément comme group element. Consultez également : Grouper les données de sortie à la page 264</td>
</tr>
<tr>
<td>As aggregate element</td>
<td>Dans la structure XML de sortie, Définit un élément aggregate. Consultez également : Agréger les données de sortie à la page 266</td>
</tr>
<tr>
<td>Add Choice</td>
<td>Configure l’élément Choice. Tous ses éléments fils développés seront contenus dans cette déclaration. Cet élément provient des concepts XSD. Il permet au tXMLMap d’exécuter la fonction de l’élément Choice XSD afin de lire ou d’écrire un flux Document. Lorsque le tXMLMap traite un élément Choice, les éléments contenus dans sa déclaration ne sont pas écrits en sortie, à moins que leurs expressions de mapping soient définies de manière appropriée.</td>
</tr>
<tr>
<td>Set as Substitution</td>
<td>Configure l’élément Substitution afin de spécifier l’élément substituable pour un élément Head donné, défini dans le XSD correspondant. L’élément de substitution permet au tXMLMap d’exécuter la fonction de l’élément Substitution XSD afin de lire ou d’écrire un flux Document. Lorsque le tXMLMap traite un élément de substitution, les éléments contenus dans sa déclaration ne sont pas écrits en sortie, à moins que leurs expressions de mapping soient définies de manière appropriée.</td>
</tr>
</tbody>
</table>

**Remarque:**

Le composant tXMLMap déclare automatiquement tout élément Choice défini dans le fichier XSD importé.

Les sections suivantes présentent de manière détaillée les processus de création d’arborescence XML.
**Importer une structure XML à partir de fichiers XML et XSD**

**Importer une structure XML à partir d’un fichier XML**

**Procédure**

1. Dans la table d’entrée correspondante, cliquez-droit sur le nom de la colonne pour ouvrir le menu contextuel. Dans cet exemple, cliquez-droit sur la colonne *Customer*.

2. Dans ce menu, sélectionnez **Import From File**.

3. Dans la boîte de dialogue, parcourez votre système jusqu’au fichier XML que vous souhaitez utiliser pour fournir la structure XML, puis double-cliquez sur le fichier.

**Importer la structure XML d’un fichier XSD**

**Procédure**

1. Dans la table d’entrée correspondante, cliquez-droit sur le nom de la colonne pour ouvrir le menu contextuel. Dans cet exemple, cliquez-droit sur la colonne *Customer*.

2. Dans ce menu, sélectionnez **Import From File**.

3. Dans la boîte de dialogue, parcourez votre système jusqu’au fichier XSD que vous souhaitez utiliser pour fournir la structure XML, puis double-cliquez sur le fichier.

4. Dans la boîte de dialogue qui apparaît, sélectionnez un élément racine dans la liste **Root** pour être la racine de votre arborescence XML puis cliquez sur **OK**. L’arborescence XML décrite par le fichier XSD importée est établie.
Remarque: La racine de l’arborescence XML est adaptable :

- Lors de l’import d’une structure XML d’entrée ou de sortie depuis un fichier XSD, vous pouvez choisir un élément pour en faire la racine de votre structure XML.
- Une fois qu’une structure XML a été importée, la balise root est automatiquement renommée avec le nom de la source XML. Pour modifier le nom de la racine manuellement, vous devez utiliser l’éditeur de schéma. Pour plus d’informations concernant cet éditeur, consultez Utiliser le Tree schema editor à la page 272.

Que faire ensuite

Puis spécifiez l’élément répétable de cette structure XML. Pour plus d’informations concernant le paramétrage de l’élément répétable, consultez Définir ou réinitialiser un élément répétable pour une structure XML créée à la page 260.

Importer une structure XML à partir du Repository

Pourquoi et quand exécuter cette tâche

Pour définir un espace de noms :

Procédure


2. Dans le menu contextuel, sélectionnez l’option Import From Repository.
3. Dans la boîte de dialogue qui s’ouvre alors, sélectionnez la connexion XML ou MDM souhaitée pour importer la structure XML correspondante.
Cette figure vous montre un exemple de connexion XML importée du Repository.

**Remarque:**
Pour importer une arborescence XML du Repository, la connexion XML correspondante doit déjà avoir été créée. Pour plus d'informations concernant la création d'une connexion à un fichier XML dans le Repository, consultez Centraliser des métadonnées d'un fichier XML à la page 309.

4. Cliquez sur **OK** pour valider la sélection.

**Résultats**

La structure XML est créée et une boucle est automatiquement définie car la boucle avait déjà été spécifiée lors de la création de la connexion XML stockée dans le Repository.

**Définir ou réinitialiser un élément répétable pour une structure XML créée**

**Pourquoi et quand exécuter cette tâche**

Vous devez définir un minimum un élément répétable pour tous les flux de données XML n'ayant pas d'élément boucle déjà défini. S'ils en ont déjà un, vous devez réinitialiser l'élément répétable lorsque c'est nécessaire.

Pour définir ou réinitialiser un élément répétable, procédez comme suit :

**Procédure**

1. Dans la structure XML créée, cliquez-droit sur l'élément que vous souhaitez définir comme élément boucle. Par exemple, vous souhaitez définir le nœud **Customer** comme élément boucle.

2. Dans le menu contextuel, sélectionnez **As loop element** afin de définir l'élément sélectionné comme élément boucle.

Une fois ceci effectué, l'élément sélectionné se voit ajouter le texte suivant : **loop**.
Résultats

**Remarque:**
Si vous fermez le **Map Editor** sans avoir défini les éléments répétables nécessaires, comme expliqué plus tôt dans ce scénario, l’élément racine est automatiquement défini comme élément répétable.

**Ajouter un sous-élément ou un attribut à une structure XML**

**Pourquoi et quand exécuter cette tâche**
Dans la structure XML, vous pouvez ajouter manuellement un sous-élément ou un attribut à la racine ou à l’un des éléments.

Pour effectuer une de ces opérations, procédez comme suit :

**Procédure**

1. Dans la structure que vous souhaitez modifier, cliquez-droit sur l’élément sur lequel vous souhaitez ajouter un sous-élément ou un attribut et sélectionnez **Create Sub-Element** ou **Create Attribute** en fonction de ce que vous souhaitez ajouter.

3. Cliquez **OK** pour valider la création du nouvel élément. Ce nouveau sous-élément ou attribut apparaît alors dans la structure XML.

**Supprimer un élément ou un attribut depuis la structure XML**

**Pourquoi et quand exécuter cette tâche**

Pour supprimer un élément ou un attribut d’une structure XML existante, procédez comme suit : Pour définir un espace de noms :

**Procédure**

1. Dans l’arborescence XML que vous souhaitez modifier, cliquez-droit sur l’élément ou l’attribut que vous souhaitez supprimer.

2. Dans le menu contextuel, sélectionnez **Delete**.
   
   L’élément ou l’attribut sélectionné est supprimé, ainsi que tous les sous-éléments ou attributs qu’il contenait.

**Gérer un espace de noms**

Vous pouvez définir et modifier un espace de noms pour chacun des éléments des arborescences XML de ces flux d’entrée ou de sortie.

**Définir un espace de noms**

**Pourquoi et quand exécuter cette tâche**

Pour définir un espace de noms :

**Procédure**

1. Dans la structure XML du flux d’entrée ou de sortie que vous souhaitez modifier, cliquez-droit sur l’élément auquel vous souhaitez associer un espace de noms. Par exemple, dans l’arborescence XML de **Customer**, vous devez créer un espace de noms pour le nœud racine.
2. Dans le menu contextuel, sélectionnez **Set A Namespace**. L’assistant **Namespace dialog** s’ouvre.

3. Dans cet assistant, saisissez l’URI à utiliser.

4. Si vous devez ajouter un préfixe à cet espace de noms, cochez la case **Prefix** dans l’assistant et saisissez le préfixe à utiliser. Dans cet exemple, cochez la case et saisissez `xhtml`.

5. Cliquez sur **OK** pour valider l’espace de noms.
Modifier la valeur par défaut d’un espace de noms

Pourquoi et quand exécuter cette tâche

Pour définir un espace de noms :

Procédure


2. Dans le menu, sélectionnez Change Namespace pour ouvrir l’assistant correspondant.

3. Saisissez la nouvelle valeur dans cet assistant.

4. Cliquez sur OK pour valider cette modification.

Supprimer un espace de noms

Pourquoi et quand exécuter cette tâche

Pour définir un espace de noms :

Procédure


2. Dans le menu, cliquez sur Delete pour valider cette suppression.

Grouper les données de sortie

Le composant tXMLMap utilise un élément “group” pour regrouper les données de sortie selon certaines conditions données. Cela vous permet d’entourer de balises “group element” les éléments répondant à la condition.

Pour définir un élément “group”, deux restrictions doivent être respectées :

1. le nœud racine ne peut être défini en tant qu’élément ”group” ;
2. l’élément “group” doit être défini sur l’élément dont le sous-élément répétable dépend directement.

Remarque:

L’option de configuration d’un élément “group” est visible à partir du moment où vous avez défini un élément répétable. Cette option est également invisible si un élément ne peut pas être configuré comme élément “group”.
Une fois l’élément "group" défini, tous ses sous-éléments sauf la boucle sont utilisés comme conditions pour regrouper les données de sortie.

Vous devez créer avec soin l’arborescence XML pour une utilisation optimale d’un élément "group" donné. Pour plus d’informations concernant l’utilisation d’un élément "group", consultez tXMLMap.

**Remarque:** Le tXMLMap propose des éléments "group" et "aggregate" afin de classer les données dans la structure XML. Lorsque vous gérez une ligne de données XML, la différence de comportement est la suivante :
- L’élément "group" traite toujours les données en un seul flux.
- L’élément "aggregate" sépare ce flux en différents flux XML complets.

Définir un group element

**Pourquoi et quand exécuter cette tâche**

Pour définir un group element, procédez comme suit :

**Procédure**

1. Dans l’arborescence XML de la sortie dans le Map editor, cliquez-droit sur l’élément que vous voulez définir comme group element.
2. Dans le menu contextuel qui s’ouvre, sélectionnez As group element.

Cet élément de la sélection devient le group element. La capture d’écran vous montre un exemple de l’arborescence XML avec le group element.

Révoquer un group element défini

**Pourquoi et quand exécuter cette tâche**

Pour révoquer un group element défini, procédez comme suit :

**Procédure**

1. Dans l’arborescence XML de la sortie dans le Map editor, cliquez-droit sur l’élément que vous avez défini comme group element.
2. Dans le menu contextuel, sélectionnez Remove group element.

Le group element défini est révoqué.
Agréger les données de sortie

Pourquoi et quand exécuter cette tâche

Avec le tXMLMap, vous pouvez définir autant d’éléments "aggregate" que nécessaire dans l’arborescence XML de sortie afin de classer les données XML. Ce composant écrit les données classées, chaque classification devenant un flux XML complet.

Procédure


2. Pour révoquer la définition de l’élément "aggregate", cliquez-droit sur cet élément puis sélectionnez Remove aggregate element, dans le menu contextuel.

Résultats

Remarque:
Pour définir un élément “aggregate”, assurez-vous que cet élément n’a pas d’enfant et que la fonctionnalité All in one est désactivée. L’option As aggregate element est disponible dans le menu contextuel si les deux conditions sont respectées.

Pour un exemple d’utilisation de l’élément "aggregate" avec le tXMLMap, consultez tXMLMap.

Remarque: Le composant tXMLMap propose les éléments “group” et “aggregate” pour classer les données dans une structure XML. Lors de la gestion d’une ligne de données (flux XML complet), la différence de comportement est la suivante :

- L’élément “group” traite toujours les données dans un flux unique.
- L’élément “aggregate” sépare ce flux en différents flux XML complets.
Définir le mode de sortie

Pour définir le mode de sortie des données de type Document, vous devez regrouper tous les éléments XML en un seul flux XML, et, lorsqu’un élément vide existe, choisir où les écrire. Ainsi, vous ne modifiez pas la structure de l’arborescence XML créée.

Ecrire dans un document les éléments en sortie

Pourquoi et quand exécuter cette tâche

Sauf si vous utilisez l’élément “aggregate” qui sépare toujours un flux XML, vous devez être capable de déterminer si un flux XML est écrit en un seul flux ou dans des flux séparés, à l’aide de la fonctionnalité All in one de l’éditeur du tXMLMap.

Pour ce faire, dans la partie droite du Map editor, procédez comme suit :

Procédure


2. Cliquez sur le champ All in one et, dans la liste déroulante, sélectionnez true ou false afin de choisir si le flux XML doit être écrit en un flux unique.
   - Si vous sélectionnez true, les données XML sont écrites en un seul flux. Dans cet exemple, le flux unique s’affiche comme suit :

   ![Diagramme de mapping](image)
La structure de ce flux est la suivante :
• Si vous sélectionnez **false**, les données XML sont écrites dans des flux séparés, chaque boucle représentant un flux. Dans cet exemple, les flux sont les suivants :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="1">
    <CustomerName>Griffith Painting and Sealcoating</CustomerName>
    <CustomerAddress>Talend@apres91</CustomerAddress>
    <idState>7</idState>
    <LabelState>Connecticut</LabelState>
  </customer>
  <customer id="56">
    <CustomerName>Glenn Oaks Office Supplies</CustomerName>
    <CustomerAddress>1859 Green Bay Rd.</CustomerAddress>
    <idState>7</idState>
    <LabelState>Connecticut</LabelState>
  </customer>
  <customer id="2">
    <CustomerName>Bill's Dive Shop</CustomerName>
    <CustomerAddress>511 Maple Ave. Apt. 1B</CustomerAddress>
    <idState>35</idState>
    <LabelState>Ohio</LabelState>
  </customer>
  <customer id="51">
    <CustomerName>DDN Bank</CustomerName>
    <CustomerAddress>456 Grossman Ln.</CustomerAddress>
    <idState>35</idState>
    <LabelState>Ohio</LabelState>
  </customer>
  <customer id="53">
    <CustomerName>Pivot Point College</CustomerName>
    <CustomerAddress>1547 Knollwood Rd.</CustomerAddress>
    <idState>9</idState>
    <LabelState>Florida</LabelState>
  </customer>
</customers>
```
Chaque flux contient une structure XML complète. La structure du premier flux, par exemple est la suivante :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="1">
    <CustomerName>Griffith Paving and Sealcoating</CustomerName>
    <CustomerAddress>9891</CustomerAddress>
    <idState>7</idState>
    <LabelState>Connecticut</LabelState>
  </customer>
</customers>
```

**Remarque :** La fonctionnalité All in one est désactivée si vous utilisez l’élément “aggregate”.

### Gérer les éléments vides dans le Map editor

#### Pourquoi et quand exécuter cette tâche

Il peut être nécessaire de créer des éléments de sortie vides durant le processus de transformation des données en un flux XML, par exemple quand le `tXMLMap` est associé à un `tWriteXMLField` créant des éléments vides ou lorsqu’il n’y a pas de colonne associée à un certain nœud XML dans le flux de données XML de sortie.

Vous pouvez garder ces éléments vides dans l’arborescence XML selon vos besoins, même si vous ne souhaitez pas les écrire en sortie.

Le composant `tXMLMap` vous permet de définir le booléen pour la création de l’élément vide. Pour ce faire, dans la partie droite du **Map editor**, effectuez les opérations suivantes :

1. Ouvrez la fenêtre du **Map editor**.
2. Sélectionnez l’élément `tXMLMap`.
3. Dans la partie droite de la fenêtre, cliquez sur la case correspondant à la création d’élément vide.
4. Cochez la case correspondant au booléen de création de l’élément vide.

En faisant cela, vous permettrez à l’élément vides d’être inclus dans le flux XML de sortie.
**Procédure**

1. Cliquez sur la clé anglaise afin d’ouvrir le panneau de paramétrage du mapping.

![Panneau de paramétrage du mapping](image)

2. Dans le panneau, cliquez sur le champ **Create empty element** et, dans la liste déroulante, sélectionnez **true** ou **false** afin de choisir d’écrire ou non l’élément vide.
   - Si vous sélectionnez **true**, l’élément vide est créé dans le flux XML de sortie et écrit, par exemple, comme suit: `<customer><LabelState/></customer>`.
   - Si vous sélectionnez **false**, l’élément vide n’est pas écrit.

**Définir la séquence des différents boucles en entrée**

**Pourquoi et quand exécuter cette tâche**

Si un élément répétable, ou si le flux de données plat reçoit des mappings d’un ou plusieurs élément(s) de boucle du flux d’entrée, vous devez définir la séquence des boucles en entrée. Le processus de transformation relatif à cette séquence effectuera d’abord une boucle sur cet élément, afin que les données écrites en sortie soient triées selon la valeur de cet élément.

![Séquence des boucles](image)

Par exemple, dans cette capture d’écran, l’élément **types** est la boucle primaire et les données écrites en sortie seront triées selon la valeur de cet élément.
Dans le cas d'une réception de plusieurs éléments répétables en entrée, un bouton [...] apparaît à côté de l'élément recevant la boucle, ou dans le cas de données plates, apparaît en haut de la table représentant le flux de données plat. Pour définir la séquence des boucles, procédez comme suit :

**Procédure**

1. Cliquez sur le bouton [...] pour ouvrir la fenêtre de configuration de séquence comme dans la capture d'écran présentée précédemment dans cette section.
2. Utilisez les boutons de montée ou de descente afin de définir l'ordre de votre séquence.

**Utiliser le Tree schema editor**

En plus des vues *Schema editor* et *Expression editor* proposées par l'éditeur du *tMap*, l'éditeur du composant *tXMLMap* fournit la vue *Tree schema editor* permettant de modifier le schéma de l'arborescence XML des flux d’entrée et de sortie.

Pour accéder au *Tree schema editor*, cliquez sur l'onglet correspondant en bas de l'éditeur.
Le schéma de l’arborescence XML du flux d’entrée apparaît à gauche de l’onglet Tree schema editor et celui du flux de sortie apparaît à droite.

Le schéma de l’arborescence XML du flux d’entrée s’affiche à gauche de l’onglet Tree schema editor et celui du flux de sortie s’affiche à droite.

<table>
<thead>
<tr>
<th>Métadonnées</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xpath</td>
<td>Le chemin absolu pointant sur chaque élément ou attribut de l’arborescence XML et éditez le nom de l’élément ou attribut correspondant.</td>
</tr>
<tr>
<td>Key</td>
<td>La clé indique si la valeur de la clé d’expression devrait être utilisée pour retrouver les données d’une autre table à travers un lien Join. Si cette case est décochée, la relation Join est désactivée.</td>
</tr>
<tr>
<td>Type</td>
<td>Type de données : Chaîne de caractères, entier, document, etc.</td>
</tr>
<tr>
<td>Nullable</td>
<td>Indique si les valeurs de champs nulles sont autorisées.</td>
</tr>
<tr>
<td>Pattern</td>
<td>Indique si un modèle est utilisé pour les données de type Date.</td>
</tr>
</tbody>
</table>

**Remarque:**

Les schémas Input et Output sont indépendants les uns des autres. Vous pouvez par exemple changer le libellé d’une colonne Output sans que le libellé du schéma Input ne change.

Cependant, toute modification apportée au schéma est immédiatement reportée sur le schéma correspondant dans la zone Input ou Output correspondante dans l’éditeur du tXMLMap, mais aussi au niveau des propriétés elles-mêmes des composants Input et Output concerné.
Gestion des métadonnées dans l'intégration de données

Objectifs

Dans le dossier **Metadata** du **Repository**, vous pouvez stocker toutes vos informations de connexion aux fichiers, bases de données et/ou systèmes, et les réutiliser pour la création de vos Jobs.

Les différents assistants de création vous aident à enregistrer ces informations qui vous permettent de paramétrer les composants d’entrée ou de sortie de vos Jobs. Ils vous permettent également de stocker une description de vos données appelée "schéma" de façon centralisée dans le Studio Talend.

Les procédures des différents assistants diffèrent en fonction du type de connexion choisi.

Cliquez sur **Metadata** dans le **Repository** pour développer l’arborescence. Chaque nœud regroupe les connexions et schémas que vous avez paramétrés.

Centraliser des métadonnées de base de données

Si vous vous connectez régulièrement à une table de base de données, vous pouvez centraliser les informations de connexion à cette base de données sous le nœud **Metadata** de la vue **Repository**.

Cette procédure est composée de deux tâches séparées mais proches :

1. La configuration d’une connexion à une base de données,
2. La récupération des schémas de table.

Le Studio Talend requiert l’installation de bibliothèques Java tierces ou de pilotes de bases de données spécifiques (fichiers `.jar`) afin de se connecter à des sources ou à des cibles.
En raison de restrictions liées aux licences, il se peut que Talend ne puisse pas intégrer certaines bibliothèques ou pilotes. Dans ce cas, l’assistant de connexion présenté dans les sections suivantes affiche les informations nécessaires pour vous aider à identifier et à installer les bibliothèques ou les pilotes en question.

**Configurer une connexion à une base de données**

Pour créer une connexion à une base de données à partir de zéro, développez le nœud Metadata du Repository. Ensuite, cliquez-droit sur Db Connections puis sélectionnez Create connection dans le menu contextuel afin d’ouvrir l’assistant de connexion à une base de données.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône dans la vue Basic settings du composant de base de données correspondant. Sa propriété Property Type doit être Built-in afin d’ouvrir l’assistant de configuration de connexion à une base de données.

Pour modifier une connexion à une base de données existante, cliquez-droit sur la connexion dans le Repository puis sélectionnez Edit connection afin d’ouvrir l’assistant de configuration de connexion.

Définissez ensuite les propriétés et les paramètres généraux de la connexion dans l’assistant.

**Définir les propriétés générales**

**Procédure**

1. Dans l’assistant de connexion qui s’ouvre, donnez un nom à votre connexion dans le champ Name. Ce nom apparaît comme nom de la connexion à la base de données sous le nœud Metadata du Repository.
2. Renseignez les champs facultatifs Purpose et Description. Les informations que vous renseignez dans le champ Description apparaissent dans une info-bulle lorsque vous bougez le pointeur de votre souris au-dessus de la connexion.


4. Si nécessaire également, cliquez sur le bouton Select à côté du champ Path afin de sélectionner un dossier sous le nœud Db connections pour sauvegarder la connexion à la base de données créée. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.

5. Une fois les champs remplis, cliquez sur Next pour passer à l’étape suivante. L’étape suivante nécessite que vous saisissiez ou que vous éditez des données de connexion à une base de données.

**Définir des paramètres de connexion**

**Procédure**

1. Sélectionnez le type de base de données auquel vous souhaitez vous connecter et renseignez les détails de connexion. Les champs que vous devez compléter varient selon le type de base de données sélectionné.
Gestion des métadonnées dans l'intégration de données
Remarque:
À partir de la version 6.0 du Studio Talend, à cause de limitations liées à Java 8, ODBC n’est plus supporté pour les connexions à la base de données Access. Le seul type de pilote supporté est JDBC.
À cause de ces limitations liées à Java 8, vous ne pouvez créer de connexion ODBC générique ou Microsoft SQL Server (ODBC) dans le Studio Talend version 6.0 et supérieures, à moins d’importer des connexions précédemment créées dans une ancienne version du Studio Talend - dans ce cas, vous pouvez créer des connexions ODBC génériques et Microsoft SQL Server (ODBC) mais elles ne fonctionneront qu’avec Java 7.
Pour une connexion à Microsoft SQL Server (JDBC), quand Microsoft est sélectionné dans la liste Db Version, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur le Centre de téléchargement Microsoft. Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décomprimé, selon votre version de JRE, renommer le fichier .jar en mssql-jdbc.jar et l’installer manuellement. Pour plus d’informations relatives au choix du fichier .jar, consultez la configuration système requise sur le Centre de téléchargement Microsoft.
Vous pouvez configurer une connexion à Oracle à l’aide du Wallet en sélectionnant Oracle Custom dans la liste déroulante DB Type, puis en cochant la case Use SSL Encryption et en spécifiant les propriétés relatives, notamment le chemin d’accès à vos fichiers TrustStore et KeyStore. Spécifiez également le mot de passe pour chacun d’eux et s’il faut désactiver l’utilisation de CBC (CipherBlock Chaining).
Si vous souhaitez vous connecter à Hive, il est recommandé d’utiliser l’une des solutions Big Data de Talend.

Avertissement:
Si vous créez une connexion MSSQL, afin de pouvoir récupérer les schémas des tables de la base de données, assurez-vous de :
- saisir dbo dans le champ Schema si vous vous connectez à MSSQL 2000,

2. (Facultatif) Spécifiez des propriétés de connexion supplémentaires dans le champ Additional parameters de la zone Database Settings.
3. Vérifiez votre connexion à l’aide de la fonction Check.
   En cas d’échec de la connexion, une fenêtre s’affiche, indiquant que la connexion a échoué. Depuis cette fenêtre, vous pouvez cliquer sur Details pour obtenir des informations supplémentaires.
   Si l’erreur est due à une bibliothèque ou un pilote (fichier .jar) manquant, ouvrez la vue Details.
4. Si nécessaire, renseignez les propriétés de base de données dans la zone Database Properties. La première phase de configuration d’une connexion à une base de données est terminée. Cliquez sur Finish pour fermer l’assistant de connexion.
La connexion à la base de données nouvellement créée est sauvegardée sous le nœud Db Connections du Repository. Plusieurs dossiers sont créés sous le nœud de connexion à la base de données, pour les requêtes SQL et les différents types de schémas, comme Calculation View.
Gestion des métadonnées dans l'intégration de données

schemas (uniquement pour SAP HANA) Synonym schemas (pour Oracle, IBM DB2 et MSSQL), Table schemas et View schemas.

Vous pouvez à présent déposer la connexion à la base de données dans l'espace de modélisation graphique en tant que composant, afin de réutiliser les détails de connexion à la base de données dans votre Job.

Récupérer les schémas de table

Pour récupérer les schéma de table à partir d'une connexion à une base de données définie, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve schema dans le menu contextuel.

Remarque:

Un message d'erreur s'affiche s'il n'y a aucune table à retrouver dans la base de données sélectionnée ou si vous n'avez pas les droits d'accès à cette base de données.

Un nouvel assistant s'ouvre et vous permet de spécifier le filtre pour chercher différents objets de base de données, comme des tables, vues et synonymes (pour Oracle, IBM DB2 et MSSQL) et la vue Calculation (uniquement pour SAP HANA).
Filtrer les objets de votre base de données

Dans la zone **Select Filter Conditions**, vous pouvez filtrer les tables de votre base de données soit par leur nom à l’aide d’un assistant dans le champ **Set the Name Filter**, soit à l’aide d’une requête SQL dans le champ **Use the Sql Filter**.

**Filtrer les tables d’une base de données par leur nom**

**Procédure**

1. Dans la zone **Select Filter Conditions**, sélectionnez l’option **Use the Name Filter**.
2. Dans la zone **Select Types**, cochez la/les case(s) du/des objet(s) de la base de données que vous voulez filtrer ou afficher.

   **Remarque:** Les options disponibles peuvent varier en fonction de la base de données sélectionnée.

3. Dans la zone **Set the Name Filter**, cliquez sur le bouton **Edit...** pour ouvrir la boîte de dialogue **Edit Filter Name**.
4. Dans la boîte de dialogue, saisissez le filtre souhaité.

   **Exemple**

   Par exemple, si vous souhaitez récupérer les objets dont le nom commence par "A", saisissez le filtre `A%` ou si vous souhaitez récupérer les objets dont le nom termine par "type", saisissez `%type`.

5. Cliquez sur **OK** pour fermer la boîte de dialogue.
6. Cliquez sur **Next** pour afficher la vue suivante de l’assistant, qui présente la liste des objets filtrés.
Filtrer les objets à l’aide d’une requête SQL

Procédure

1. Dans la zone Select Filter Conditions, sélectionnez l’option Use Sql Filter.
2. Dans le champ Set the Sql Filter, saisissez la requête SQL souhaitée pour filtrer les objets de bases de données.
3. Cliquez sur Next pour afficher la vue suivante de l’assistant, qui présente la liste des objets filtrés.
Gestion des métadonnées dans l'intégration de données

Sélectionner les tables et définir les schémas de table

Pourquoi et quand exécuter cette tâche

Une fois que vous avez filtré la liste des objets de base de données, procédez comme suit afin de charger les schémas des objets souhaités dans votre Repository :

Procédure

1. Sélectionnez un ou plusieurs objets à partir de la liste puis cliquez sur le bouton Next pour ouvrir la vue suivante de l'assistant, où vous pourrez consulter les schémas des objets sélectionnés.

   Remarque: Si aucun schéma n'est visible dans la liste, cliquez sur le bouton Check connection en dessous de la liste pour vérifier l'état de la connexion de base de données.

2. Modifiez les schémas si besoin.

   Assurez-vous que le type de données dans la colonne Type est correctement défini.

   Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

   Les types de données Talend les plus utilisés sont les suivants :

   - Object : est un type de données Talend générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être
traité à l'aide d'un composant `tFileInputRaw` en spécifiant qu'il comporte un type de données `Object`.

- **List** : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément `xsd:list`.
- **Document** : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

**Avertissement** : Si la table de la base de données source contient une valeur par défaut étant une fonction ou une expression et non une chaîne de caractères, assurez-vous de supprimer les guillemets simples entourant la valeur par défaut du schéma final, s'il y en a, afin d'éviter des résultats inattendus lors de la création de tables de bases de données à l'aide de ce schéma.

Pour plus d'informations, consultez Talend Help Center (https://help.talend.com).

**Conseil** : Si vous vous rendez compte qu'un certain type de données de la base de données n'est pas encore supporté par Talend, vous pouvez modifier le fichier de mapping pour cette base de données afin d'activer la conversion entre le type de données de la base de données et le type de données Talend correspondant. Pour plus d’informations, consultez Mapping de types à la page 473.

Par défaut, le schéma affiché dans la zone `Schema` est basé sur la première table sélectionnée dans la liste des schémas chargés (à gauche). Vous pouvez modifier le nom du schéma et selon vos besoins, vous pouvez également personnaliser la structure du schéma.

La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma. Vous avez également la possibilité d’importer un schéma XML à partir d'un fichier ou d'exporter le schéma courant en XML.

Pour restaurer un schéma basé sur l'une des tables chargées, sélectionnez le schéma de la table dans la liste déroulante et cliquez sur `Retrieve schema`. Notez que le schéma ainsi chargé écrase le schéma courant et ne retient aucune des modifications ponctuelles.

Quand vous avez apporté les changements souhaités, cliquez sur `Finish` pour terminer la création du schéma de base de données. Tous les schémas ainsi récupérés seront sauvegardés dans le dossier du schéma correspondant, sous le nœud relatif à votre base de données.

Vous pouvez à présent déposer n’importe quel schéma de table d’une connexion à une base de données depuis le `Repository` dans l’espace de modélisation graphique en tant que nouveau composant de base de données. Pour plus d’informations, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

### Centraliser des métadonnées JDBC

Pour centraliser des métadonnées basées sur des tables de base de données dans une connexion JDBC, sous le nœud `Metadata` du `Repository`, la procédure est composée de deux tâches séparées mais proches :

1. La configuration d'une connexion JDBC,
2. La récupération des schémas de table.

Les sections suivantes décrivent dans le détail comment effectuer ces tâches.

Pour un exemple d'utilisation d'une connexion JDBC, consultez Exemples de Jobs d’intégration de données dans le Talend Help Center (https://help.talend.com).
Créer une connexion JDBC et importer un pilote de base de données

**Procédure**

1. Pour créer une connexion JDBC à partir de zéro, développez le nœud *Metadata* du *Repository*. Ensuite, cliquez-droit sur *Db Connections* puis sélectionnez *Create connection* dans le menu contextuel afin d’ouvrir l’assistant de connexion à une base de données.

   Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône dans la vue *Basic settings* du composant de base de données correspondant. Sa propriété *Property Type* doit être *Built-in* afin d’ouvrir l’assistant de configuration de connexion à une base de données.

   Pour modifier une connexion JDBC existante, cliquez-droit sur la connexion dans le *Repository* puis sélectionnez *Edit connection* afin d’ouvrir l’assistant de configuration de connexion.

2. Renseignez les informations génériques du schéma, notamment les champs *Name* et *Description* puis cliquez sur *Next* afin de passer à l’étape de définition des détails de connexion.

   Pour plus d’informations, consultez la section sur la définition des propriétés générales dans Configurer une connexion à une base de données à la page 275.

3. Sélectionnez *JDBC* dans la liste *DB type*. 
4. Si la bibliothèque qui doit être importée n’est pas disponible sur votre machine, téléchargez-la ou installez-la en utilisant la vue Modules. Vous pouvez également la télécharger ou la stocker dans un dossier local.

**Remarque:** Selon la connexion, il est possible que vous deviez importer plusieurs fichiers de pilotes. Par exemple, lorsque vous vous connectez à Google BigQuery, importez chaque fichier .jar extrait de ce fichier .zip. Pour plus d’informations, consultez la procédure pour installer et utiliser le pilote JDBC (en anglais).

5. Dans la table **Drivers**, ajoutez une ligne en cliquant sur le bouton [+].
6. Cliquez sur la nouvelle ligne puis sur le bouton [...] pour ouvrir la boîte de dialogue Module dans laquelle vous pouvez importer la bibliothèque externe.

7. Si vous avez installé la bibliothèque à l’aide de la vue Modules :
   • Sélectionnez l’option Platform puis la bibliothèque dans la liste.
   • Sélectionnez l’option Artifact repository (local m2/nexus) > Find by name ou Artifact repository (local m2/nexus) > Find by Maven URI, puis spécifiez le nom complet ou l’URI Maven du module de la bibliothèque, puis cliquez sur le bouton Detect the module install status afin de valider son statut d’installation.

8. Si vous avez stocké le fichier de la bibliothèque dans un répertoire local :
   a) Sélectionnez l’option Artifact repository (local m2/nexus).
   b) Sélectionnez l’option Install a new module et cliquez sur le bouton [...] pour parcourir votre système jusqu’au fichier de bibliothèque.
   c) Si vous devez personnaliser l’URI Maven de la bibliothèque, cochez la case Custom MVN URI, spécifiez la nouvelle URI et cliquez sur le bouton Detect the module install status afin de valider son statut d’installation.
Gestion des métadonnées dans l'intégration de données

9. Cliquez sur OK afin de confirmer vos modifications.
Le fichier de bibliothèque importé est listé dans la table Drivers.

Remarque: Vous pouvez remplacer ou supprimer la bibliothèque importée, ou importer de nouvelles bibliothèques si nécessaire.

Renseigner les détails de connexion JDBC

Procédure

1. Renseignez les informations requises pour la connexion comme suit :
   • Dans le champ JDBC URL, renseignez l’URL de connexion au serveur SGBD.
   • Dans le champ Driver Class, spécifiez la classe principale du pilote permettant de communiquer avec la base de données.
   • Dans les champs User Id et Password, renseignez votre identifiant et mot de passe d’authentification à la base de données.
   • Dans la liste Mapping File, sélectionnez le mapping permettant au type de la base de données de correspondre au type Java des données dans le schéma, selon le type de base de données auquel vous vous connectez.

Remarque: Les fichiers de mapping sont au format XML et sont gérés dans le menu Window > Preferences > Talend > Specific Settings > Metadata of TalendType. Pour plus d’informations, consultez Accéder aux fichiers de mapping et définir les mappings de types à la page 473.

2. Cliquez sur le bouton Test connection afin de vérifier votre connexion.
3. Cliquez sur Finish pour fermer l’assistant de connexion.

La connexion JDBC nouvellement créée est accessible depuis le Repository et contient plusieurs sous-dossiers dont Queries pour les requêtes SQL et Table schemas qui regroupe tous les schémas relatifs à cette connexion une fois ces derniers récupérés.

Récupérer les schémas de table

Procédure

1. Pour récupérer les schéma de table à partir d’une connexion à une base de données définie, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve schema dans le menu contextuel.
Gestion des métadonnées dans l’intégration de données

Dans l’assistant qui s’ouvre, vous pouvez filtrer et afficher différents objets (tables, vues et synonymes) de votre connexion à la base de données, sélectionner des tables et définir des schémas de table.

2. Configurez le filtre sur les objets de bases de données selon vos besoins. Pour plus de détails, consultez Filtrer les objets de votre base de données à la page 280.

Cliquez sur Next pour ouvrir une vue listant les objets de bases de données que vous avez filtrés. La liste montre toutes les bases de données avec toutes leurs tables présentes dans la connexion de base de données qui correspondent aux conditions de filtre que vous avez définies.

Si aucune base de données n’est visible dans la liste, cliquez sur Check connection afin de vérifier la connexion à la base de données.


Assurez-vous que le type de données dans la colonne Type est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données Talend les plus utilisés sont les suivants :

- **Object** : est un type de données Talend générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant tFileInputRaw en spécifiant qu’il comporte un type de données Object.
Gestion des métadonnées dans l'intégration de données

- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

**Avertissement** : Si la table de la base de données source contient une valeur par défaut étant une fonction ou une expression et non une chaîne de caractères, assurez-vous de supprimer les guillemets simples entourant la valeur par défaut du schéma final, s’il y en a, afin d’éviter des résultats inattendus lors de la création de tables de bases de données à l’aide de ce schéma. Pour plus d’informations, consultez Talend Help Center (https://help.talend.com).

**Conseil** : Si vous vous rendez compte qu’un certain type de données de la base de données n’est pas encore supporté par Talend, vous pouvez modifier le fichier de mapping pour cette base de données afin d’activer la conversion entre le type de données de la base de données et le type de données Talend correspondant. Pour plus d’informations, consultez Mapping de types à la page 473.

Par défaut, le schéma affiché dans la zone Schema est basé sur la première table sélectionnée dans la liste des schémas chargés (à gauche). Vous pouvez modifier le nom du schéma et, selon vos besoins, vous pouvez également personnaliser la structure du schéma.

La barre d’tools vous permet d’ajouter, supprimer ou déplacer des colonnes dans votre schéma. Vous avez également la possibilité d’importer un schéma XML à partir d’un fichier ou d’exporter le schéma courant en XML.

Pour restaurer un schéma basé sur l’une des tables chargées, sélectionnez le schéma de la table dans la liste déroulante et cliquez sur **Retrieve schema**. Notez que le schéma ainsi chargé écrase le schéma courant et ne retient aucune des modifications ponctuelles.

Quand vous avez apporté les changements souhaités, cliquez sur **Finish** pour terminer la création du schéma de base de données. Tous les schémas ainsi créés apparaissent dans le sous-dossier **Table schemas** du noeud correspondant à votre base de données.

Vous pouvez à présent déposer n’importe quel schéma de table d’une connexion à une base de données depuis le Repository dans l’espace de modélisation graphique en tant que nouveau composant de base de données. Pour plus d’informations, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

### Centraliser des métadonnées SAS

Si vous devez souvent vous connecter à un système distant SAS, vous pouvez centraliser les informations de connexion dans le Repository.

Pour centraliser les métadonnées d’information d’une connexion SAS dans le Repository, vous devez au préalable :

1. Configurer une connexion SAS,
2. Récupérer les schémas de base de données.

**Prérequis** :

- Le Studio Talend requiert l’installation de bibliothèques Java tierces ou de pilotes de bases de données spécifiques (fichiers .jar) afin de se connecter à des sources ou à des cibles. En raison de restrictions liées aux licences, il se peut que Talend ne puisse pas intégrer certaines
bibliothèques ou pilotes. Dans ce cas, l’assistant de connexion présenté dans les sections suivantes affiche les informations nécessaires pour vous aider à identifier et à installer les bibliothèques ou les pilotes en question.

- Avant d’effectuer la procédure suivante pour configurer votre connexion SAS, assurez-vous de récupérer vos métadonnées depuis le serveur SAS et de les exporter au format XML.

**Configurer une connexion SAS**

**Procédure**

1. Dans l’arborescence du Repository du Studio Talend, cliquez-droit sur DB Connections sous le nœud Metadata et sélectionnez Create connection dans le menu contextuel afin d’ouvrir l’assistant Database Connection.

2. Renseignez les propriétés générales de la connexion, comme le nom (Name) et la description (Description) puis cliquez sur le bouton Next pour passer à la vue suivante de l’assistant afin de définir les détails de la connexion.

   Pour plus d’informations, consultez la section sur la définition des propriétés générales dans Configurer une connexion à une base de données à la page 275.

3. Dans le champ DB Type de l’assistant Database Connection, sélectionnez l’option SAS et renseignez les champs qui suivent avec vos informations de connexion SAS.
4. Si nécessaire, vérifiez votre connexion à l’aide de la fonction **Check**.

5. Si nécessaire également, renseignez les propriétés de base de données dans la zone **Database Properties**.

6. Cliquez sur **Finish** pour valider vos modifications et fermer l’assistant.
   
   La connexion à la base de données nouvellement créée est accessible depuis le **Repository** sous le nœud **DB Connections**. Elle contient plusieurs sous-dossiers dont **Table schemas** qui regroupe tous les schémas relatifs à cette connexion une fois ces derniers récupérés.

**Récupérer des schémas de table SAS**
**Procédure**


Dans l’assistant qui s’ouvre, vous pouvez filtrer et afficher différents objets (tables, vues) de votre connexion à la base de données, sélectionner des tables et définir des schémas de table.

2. Filtrez les objets de base de données selon vos besoins, sélectionnez une ou plusieurs tables et modifiez les tables de schéma si nécessaire. Pour plus de détails, consultez Récupérer les schémas de table à la page 279.

Assurez-vous que le type de données dans la colonne *Type* est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données *Talend* les plus utilisés sont les suivants :

- **Object** : est un type de données *Talend* générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant *tFileInputRaw* en spécifiant qu’il comporte un type de données *Object*.
- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément `xsd:list`.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

Quand vous avez apporté les changements souhaités, vous pouvez déposer n’importe quel schéma de table de la connexion SAS depuis le *Repository* dans l’espace de modélisation graphique en tant que nouveau composant. Pour plus d’informations, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du *Repository* dans un Job à la page 57.

**Centraliser des métadonnées File Delimited**

Si vous devez souvent lire et/ou écrire des données dans des fichiers délimités, vous pouvez centraliser les métadonnées de ces fichiers dans le *Repository* afin de les réutiliser facilement. Les
Gestion des métadonnées dans l’intégration de données

métadonnées de type File Delimited peuvent être utilisées pour définir les propriétés des composants tInputFileDelimited, tFileOutputDelimited et t*OutputBulk.

Remarque:
La création de schémas de fichier est similaire pour tous les types de fichiers : Delimited (délimités), Positional (positionnels), Regex (Expressions régulières), XML ou LDIF.

A la différence de l’assistant de connexion à une base de données, l’assistant New Delimited File regroupe la connexion au fichier et la définition du schéma, en une seule procédure de quatre étapes.

Pour créer une connexion à un fichier délimité à partir de zéro, développez le nœud Metadata du Repository. Ensuite, cliquez-droit sur File Delimited puis sélectionnez Create file delimited dans le menu contextuel afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône dans la vue Basic settings du composant de base de données correspondant. Sa propriété Property Type doit être Built-in afin d’ouvrir l’assistant de configuration de connexion à une base de données.

Définissez ensuite les propriétés générales et le schéma du fichier dans l’assistant.

Vous pouvez à présent déposer n’importe quelle connexion à un fichier ou n’importe quel schéma depuis le Repository dans l’espace de modélisation graphique en tant que nouveau composant. Pour plus d’informations concernant l’utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existant, cliquez-droit sur la connexion dans le Repository et sélectionnez Edit file delimited afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve Schema dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

Définir les propriétés générales

Procédure

1. Dans l’assistant de configuration des métadonnées du fichier, renseignez le champ Name, qui est obligatoire, et les champs Purpose et Description si vous le souhaitez. Les informations fournies dans le champ Description s’affichent en tant qu’info-bulle lorsque vous placez votre curseur sur la métadonnée.
2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d’un élément du **Repository** dans la boîte de dialogue Project Settings. Pour plus d’informations, consultez **Gestion des versions** à la page 479 et **Gestion du statut** à la page 481 respectivement.

3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File delimited** pour sauvegarder la connexion au fichier créé. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.

4. Cliquez sur **Next** lorsque vous avez défini les propriétés générales.

**Définir le chemin et le format du fichier**

**Procédure**

1. Spécifiez le chemin entier de votre fichier source dans le champ **File**, ou cliquez sur le bouton **Browse...** pour rechercher le fichier.

**Remarque:** La notation de chemin Convention Universelle de Nommage (UNC) n’est pas supportée. Si votre fichier source est sur un hôte LAN, vous pouvez d’abord mapper le dossier du réseau dans un lecteur local.
2. Sélectionnez le **Format** du système d’exploitation dans lequel le fichier a été créé. Ces informations sont utilisées afin de pré-remplir les champs suivants. Si la liste ne propose pas le format approprié, ignorez ce champ.

3. Le **File viewer** donne un instantané du fichier chargé. Vérifiez la cohérence du fichier, la présence d’un en-tête et plus généralement la structure du fichier.

4. Cliquez sur **Next** pour passer à l’étape suivante.

**Définir le parsing du fichier**

**Pourquoi et quand exécuter cette tâche**

Dans cette vue, vous pouvez affiner différents paramètres de votre fichier afin que le schéma soit correctement récupéré.
Procédure

1. Paramétrez le type d’encodage (Encoding) et les séparateurs de champs et de lignes (Field separator et Row separator) dans la zone File Settings.
2. En fonction de votre type de fichier (CSV ou Delimited), vous pouvez paramétrer des caractères d’inclusion et d’exclusion (respectivement Text Enclosure et Escape character).


4. Le champ Limit of Rows vous permet de restreindre l’étendue du fichier qui fait l’analyse. Si nécessaire, cochez la case Limit et définissez ou sélectionnez le nombre de lignes souhaitées.

5. Dans le panneau File Preview, vous pouvez visualiser l’impact de vos paramétrages.

6. Cochez la case Set heading row as column names pour transformer la première ligne analysée en libellés des colonnes du schéma. Notez que le nombre de lignes d’en-tête à ignorer (champ Rows To Skip) est incrémenté de 1.

7. Cliquez sur Refresh dans le panneau d’aperçu, afin que les modifications de paramétrage prennent effet.

8. Cliquez sur Next pour accéder à l’étape suivante permettant de vérifier et de personnaliser le schéma de fichier généré.

Vérifier et personnaliser le schéma du fichier

Pourquoi et quand exécuter cette tâche

La dernière étape affiche le schéma du fichier délimité généré. Vous pouvez personnaliser le tableau.
Procédure

1. Si le fichier délimité sur lequel est basé le schéma a été modifié, utilisez le bouton **Guess** pour générer le schéma à nouveau. Notez que si vous personnalisez le schéma, la fonctionnalité **Guess** ne retiendra pas ces modifications.

2. Modifiez les schémas si besoin.
   Assurez-vous que le type de données dans la colonne **Type** est correctement défini.
   Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).
   Les types de données **Talend** les plus utilisés sont les suivants :
   
   - **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant **tFileInputRaw** en spécifiant qu’il comporte un type de données **Object**.
   - **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément **xsd:list**.
   - **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

3. Cliquez sur **Finish**. Le nouveau schéma apparaît dans l’arborescence du **Repository**, sous le nœud de la connexion **File Delimited** appropriée.
Gestion des métadonnées dans l'intégration de données

Configurer un schéma File Positional

Si vous devez souvent lire et/ou écrire des données dans des fichiers positionnels, vous pouvez centraliser les métadonnées de ces fichiers dans le Repository afin de les réutiliser facilement. Les métadonnées de type File Positional peuvent être utilisées pour définir les propriétés des composants tInputFilePositional, tFileOutputPositional et tFileInputMSPositional.

L’assistant New Positional File regroupe la connexion au fichier et la définition du schéma en une seule procédure de quatre étapes.

Pour créer une connexion à un fichier délimité à partir de zéro, développez le nœud Metadata du Repository. Ensuite, cliquez-droit sur File Positional puis sélectionnez Create file positional dans le menu contextuel afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône dans la vue Basic settings du composant de base de données correspondant. Sa propriété Property Type doit être Built-in afin d’ouvrir l’assistant de configuration de connexion à une base de données.

Définissez ensuite les propriétés générales et le schéma du fichier dans l’assistant.

Le nouveau schéma apparaît dans l’arborescence du Repository, sous le nœud de la connexion File positional appropriée. Vous pouvez glisser-déposer la métadonnée du Repository dans l’espace de modélisation graphique en tant que nouveau composant. Pour plus d’informations concernant l’utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Edit file positional afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve Schema dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

Définir les propriétés générales de la connexion File Positional

Procédure

1. Dans l’assistant de configuration des métadonnées du fichier, renseignez le champ Name, qui est obligatoire, et les champs Purpose et Description si vous le souhaitez. Les informations fournies dans le champ Description s’affichent en tant qu’info-bulle lorsque vous placez votre curseur sur la métadonnée.
2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d’un élément du **Repository** dans la boîte de dialogue **Project Settings**. Pour plus d’informations, consultez **Gestion des versions** à la page 479 et **Gestion du statut** à la page 481 respectivement.

3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File positional**. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.

4. Cliquez sur **Next** lorsque vous avez défini les propriétés générales.

**Définir le chemin, le format et les marqueurs de position du fichier**

**Procédure**

1. Spécifiez le chemin entier de votre fichier source dans le champ **File**, ou cliquez sur le bouton **Browse...** pour rechercher le fichier.

   *Remarque:* La notation de chemin Convention Universelle de Nommage (UNC) n’est pas supportée. Si votre fichier source est sur un hôte LAN, vous pouvez d’abord mapper le dossier du réseau dans un lecteur local.

2. Sélectionnez le type d’encodage (**Encoding**) et le format du système d’exploitation (**Format**) dans lequel le fichier a été créé. Ces informations sont utilisées afin de pré-remplir les champs suivants. Si la liste ne propose pas le format approprié, ignorez ce champ.
Le fichier est chargé et la zone **File Viewer** donne un instantané du fichier et vous permet de placer les marqueurs de position.

Gestion des métadonnées dans l’intégration de données

Les champs **Field Separator** et **Marker position** sont automatiquement remplis par une série de chiffres séparés par des virgules.

Les chiffres dans le champ **Field Separator** représentent le nombre de caractères entre les séparateurs, c’est-à-dire la longueur des colonnes dans le fichier chargé. L’astérisque symbolise tous les caractères restants jusqu’à la fin de la ligne, à partir du séparateur précédent. Vous pouvez changer ces chiffres afin de spécifier la longueur des colonnes de façon précise.

Le champ **Marker position** indique la position exacte de chaque marqueur sur la règle, en nombre de caractères. Vous pouvez affiner la position exacte du marqueur en saisissant la valeur exacte de position.

Afin de déplacer un marqueur, cliquez sur une flèche et glissez-la vers sa nouvelle position. Pour retirer un marqueur, cliquez sur sa flèche et faites-la glisser vers la règle, jusqu’à ce qu’une icône apparaîsse.

4. Cliquez sur **Next** pour passer à l’étape suivante.

**Configurer les paramètres de parsing de votre fichier positionnel**

**Pourquoi et quand exécuter cette tâche**

Dans cette vue, vous pouvez définir les paramètres de parsing du fichier afin que le schéma du fichier soit correctement récupéré.

A ce stade, l’aperçu affiche les colonnes du fichier par la position de marqueurs.
Gestion des métadonnées dans l'intégration de données

Procédure

1. Définissez les séparateurs de champs et de lignes dans la zone **File Settings**.
   - Si nécessaire, vous pouvez modifier les chiffres dans le champ **Field Separator** afin de spécifier la longueur des colonnes de façon précise.
   - Si le séparateur de lignes de votre fichier n’est pas le caractère de fin de ligne standard, sélectionnez **Custom String** dans la liste **Row Separator** et spécifiez le caractère dans le champ **Corresponding Character**.

2. Si votre fichier possède des en-têtes à exclure du contenu des données, cochez la case **Header** dans la zone **Rows To Skip** et configurez le nombre de lignes à ignorer dans le champ correspondant. De plus, si vous savez que le fichier contient des informations de pied de page, cochez la case **Footer** et définissez le nombre de lignes à ignorer.

3. La zone **Limit of Rows** vous permet de restreindre l’étendue du fichier à analyser. Si nécessaire, cochez la case **Limit** et définissez ou sélectionnez le nombre de lignes souhaitées.

4. Si le fichier contient des libellés de colonne, cochez la case **Set heading row as column names** afin de transformer la première ligne en libellés de colonne. Notez que le nombre de lignes d’en-tête à ignorer (champ Rows To Skip) est incrémenté de 1.

5. Cliquez sur le bouton **Refresh Preview** dans le panneau de prévisualisation (**Preview**) afin que les paramètres prennent effet.
6. Cliquez sur **Next** afin passer à l’étape suivante pour vérifier et personnaliser le schéma de fichier généré.

**Vérifier et personnaliser le schéma de votre fichier positionnel**

**Pourquoi et quand exécuter cette tâche**

L’étape 4 affiche le schéma final généré. Remarquez que tous les caractères du fichier pouvant être mal interprétés par le programme par la suite sont remplacés par un caractère neutre. Par exemple, les tirets soulignés remplacent les astérisques.

**Procédure**

1. Renommez le schéma (par défaut, `metadata`) et modifiez les colonnes du schéma comme souhaité.

   Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

   Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](en anglais).

   Les types de données **Talend** les plus utilisés sont les suivants :
   
   - **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être
traité à l'aide d'un composant `tFileInputRaw` en spécifiant qu'il comporte un type de données `Object`.

- **List** : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément `xsd:list`.
- **Document** : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

2. Pour générer à nouveau le schéma du fichier positionnel, cliquez sur le bouton **Guess**. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur Guess.

3. Une fois vos changements effectués, cliquez sur **Finish** pour fermer l'assistant.

**Centraliser des métadonnées File Regex**

Les schémas de fichier Regex servent à manipuler les fichiers composés d'expressions régulières, notamment les fichiers log. Si vous devez souvent vous connecter à un fichier Regex, vous pouvez centraliser les informations de connexion et le schéma de ce fichier dans le Repository afin de les réutiliser facilement.

L'assistant **New RegEx File** regroupe la connexion au fichier et la définition du schéma en une seule procédure de quatre étapes.

**Remarque:**
Cette procédure nécessite une connaissance avancée de la syntaxe des expressions régulières.

Pour créer une connexion à un fichier Regex à partir de zéro, développez le noeud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **File Regex** puis sélectionnez **Create file regex** dans le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l'icône `Ubuntu` dans la vue **Basic settings** du composant de base de données correspondant. Sa propriété **Property Type** doit être **Built-in** afin d'ouvrir l'assistant de configuration de connexion à une base de données.

Définissez ensuite les propriétés générales et le schéma du fichier dans l'assistant.

Le nouveau schéma apparaît dans l'arborescence du **Repository**, sous le nœud de la connexion **File regex** appropriée. Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations concernant l'utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file regex** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.
Définir les propriétés générales de la connexion Regex File

Procédure

1. Dans l’assistant de configuration des métadonnées du fichier, renseignez le champ Name, qui est obligatoire, et les champs Purpose et Description si vous le souhaitez. Les informations fournies dans le champ Description s’affichent en tant qu’info-bulle lorsque vous placez votre curseur sur la métadonnée.

![Image](image.png)


3. Si nécessaire également, cliquez sur le bouton Select à côté du champ Path afin de sélectionner un dossier sous le nœud File regex. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.

4. Cliquez sur Next lorsque vous avez défini les propriétés générales.

Configurer le chemin et le format de votre fichier Regex

Procédure

1. Spécifiez le chemin entier de votre fichier source dans le champ File, ou cliquez sur le bouton Browse... pour rechercher le fichier.
Remarque: La notation de chemin Convention Universelle de Nommage (UNC) n’est pas supportée. Si votre fichier source est sur un hôte LAN, vous pouvez d’abord mapper le dossier du réseau dans un lecteur local.

2. Sélectionnez le type d’encodage (Encoding) et le format du système d’exploitation (Format) dans lequel le fichier a été créé. Ces informations sont utilisées afin de pré-remplir les champs suivants. Si le système d’exploitation souhaité n’est pas proposé dans la liste, ignorez ce champ.

Le File viewer donne un instantané du fichier chargé.

3. Cliquez sur Next pour définir la structure du schéma.

Définir les paramètres de parsing de votre fichier Regex

Pourquoi et quand exécuter cette tâche

Dans cette vue, vous pouvez définir les paramètres de parsing du fichier afin que le schéma du fichier soit correctement récupéré.

Procédure

1. Définissez les séparateurs de champs et de lignes dans la zone File Settings.
   - Si nécessaire, vous pouvez modifier les chiffres dans le champ Field Separator afin de spécifier la longueur des colonnes de façon précise.
• Si le séparateur de lignes de votre fichier n’est pas le caractère de fin de ligne standard, sélectionnez **Custom String** dans la liste **Row Separator** et spécifiez le caractère dans le champ **Corresponding Character**.

2. Dans le panneau de paramétrage des expressions régulières, **Regular Expression settings**, saisissez l’expression régulière utilisée pour délimiter le fichier.

![Regular Expression settings](image)

**Avertissement:**
Veillez à utiliser les bons guillemets (simples ou doubles) au moment de rédiger le code.

3. Si votre fichier possède des en-têtes à exclure du contenu des données, cochez la case **Header** dans la zone **Rows To Skip** et configurez le nombre de lignes à ignorer dans le champ correspondant. De plus, si vous savez que le fichier contient des informations de pied de page, cochez la case **Footer** et définissez le nombre de lignes à ignorer.

4. Le champ **Limit of Rows** vous permet de restreindre l’étendue du fichier qui fait l’analyse. Si nécessaire, cochez la case **Limit** et définissez ou sélectionnez le nombre de lignes souhaitées.

5. Si le fichier contient des libellés de colonne, cochez la case **Set heading row as column names** afin de transformer la première ligne en libellés de colonne. Notez que le nombre de lignes d’en-tête à ignorer (champ <Rows To Skip) est incrémenté de 1.

6. Cliquez ensuite sur **Refresh preview** pour que les modifications soient prises en compte. Le bouton change d’apparence en **Stop** jusqu’à ce que l’aperçu soit rafraîchi.

![Refresh preview](image)

7. Cliquez sur **Next** afin de passer à l’étape suivante pour vérifier et personnaliser le schéma de fichier Regex généré.

**Vérifier et personnaliser le schéma de votre fichier Regex**

**Procédure**

1. Renommez le schéma (par défaut, metadata) et modifiez les colonnes du schéma comme souhaité.
   Assurez-vous que le type de données dans la colonne **Type** est correctement défini.
Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données Talend les plus utilisés sont les suivants :

- **Object** : est un type de données Talend générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant tFileInputRaw en spécifiant qu’il comporte un type de données Object.
- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

2. Pour restaurer ou mettre à jour le schéma du fichier Regex, cliquez sur **Guess**. Notez cependant que toute modification ou personnalisation du schéma n’est pas conservée lorsque vous cliquez sur Guess.

3. Une fois vos changements effectués, cliquez sur **Finish** pour fermer l’assistant.

**Centraliser des métadonnées d’un fichier XML**

Si vous devez souvent vous connecter à un fichier XML, vous pouvez utiliser l’assistant New Xml File pour centraliser les informations de connexion et le schéma de ce fichier dans le Repository afin de les réutiliser facilement.

Selon l’option sélectionnée, cet assistant vous permet de créer une connexion en lecture (Input) ou une connexion en écriture (Output). Dans un Job, les composants **tFileInputXML** et **tExtractXMLField** utilisent la connexion en lecture créée pour lire des fichiers XML. Dans un Job, le composant **tAdvancedFileOutputXML** utilise le schéma d’écriture créé pour écrire un fichier XML, s’il n’existe pas, ou y ajouter des informations, s’il existe déjà.

Si vous souhaitez lire un fichier XML, consultez Configurer les métadonnées pour un fichier d’entrée (Input) XML à la page 309.

Si vous souhaitez écrire un fichier XML, consultez Configurer les métadonnées pour un fichier de sortie (Output) XML à la page 320.

Pour créer une connexion à un fichier XML à partir de zéro, développez le nœud Metadata du Repository. Ensuite, cliquez-droit sur File XML puis sélectionnez Create file XML dans le menu contextuel afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône dans la vue Basic settings du composant de base de données correspondant. Sa propriété Property Type doit être Built-in afin d’ouvrir l’assistant de configuration de connexion à une base de données.

Définissez ensuite les propriétés générales et le schéma du fichier dans l’assistant.

**Configurer les métadonnées pour un fichier d’entrée (Input) XML**

Cette section décrit la définition et le chargement d’une connexion à un fichier d’entrée (Input) XML.
Définir les propriétés générales de la connexion File XML pour un fichier d’entrée

Pourquoi et quand exécuter cette tâche
Dans cette étape, vous allez définir les propriétés générales des métadonnées, telles que le nom (champ Name), l’objectif (champ Purpose) et une Description.

Procédure
1. Dans l’assistant de configuration des métadonnées du fichier, renseignez le champ Name, qui est obligatoire, et les champs Purpose et Description si vous le souhaitez. Les informations fournies dans le champ Description s’affichent en tant qu’info-bulle lorsque vous placez votre curseur sur la métadonnée.

   **Remarque:**
   Lorsque vous saisissez les propriétés générales des métadonnées à créer, vous devez définir le type de connexion, entrée (input) ou sortie (output). Il est donc conseillé de saisir des informations susceptibles de vous aider à distinguer les schémas d’entrée et de sortie.

3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File XML**. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.

4. Cliquez sur **Next** pour sélectionner le type de métadonnées.

**Sélectionner le type de métadonnées (Input)**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous allez définir le type de métadonnées, Input ou Output. Dans cet exemple, sélectionnez le type Input.

**Procédure**

1. Dans la boîte de dialogue, sélectionnez **Input XML**.

![Input XML selection dialog](image)

2. Cliquez sur **Next** pour charger le fichier d’entrée.

**Charger un fichier Input**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous allez sélectionner votre fichier d’entrée, son encodage, définir le nombre de colonnes sur lequel la requête XPath devra être exécutée.
Gestion des métadonnées dans l’intégration de données

Le fichier XML d’entrée utilisé pour expliquer cette étape contient des informations de contact.

```
<contactInfo>
  <contact>
    <id>1</id>
    <firstName>Michael</firstName>
    <lastName>Jackson</lastName>
    <company>Talend</company>
    <city>Paris</city>
    <phone>2323</phone>
  </contact>
  <contact>
    <id>2</id>
    <firstName>Elisa</firstName>
    <lastName>Black</lastName>
    <company>Talend</company>
    <city>Paris</city>
    <phone>4499</phone>
  </contact>
  ...
</contactInfo>
```

Pour charger un fichier XML, procédez comme suit :

**Procédure**


   Un aperçu de la structure du fichier lu s’affiche dans la zone *Schema Viewer*. Vous avez la possibilité de développer et de visualiser tous les niveaux de la structure XML du fichier.
2. Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement.

3. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez 0 si vous souhaitez l'effectuer sur toutes les colonnes.

4. Cliquez sur **Next** afin de configurer les paramètres du schéma.

**Charger un fichier XSD**

**Pourquoi et quand exécuter cette tâche**

Cette procédure décrit comment charger un fichier XSD pour obtenir une structure XML.
Un fichier XSD est utilisé pour décrire le schéma des fichiers XML. La structure et les types d'éléments peuvent être décrits à l'aide du XSD suivant, utilisé comme XSD d'entrée d'exemple dans cette section.

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="contactInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="contact"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="contact">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="id"/>
        <xs:element ref="firstName"/>
        <xs:element ref="lastName"/>
        <xs:element ref="company"/>
        <xs:element ref="city"/>
        <xs:element ref="phone"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="id" type="xs:integer"/>
  <xs:element name="firstName" type="xs:NCName"/>
  <xs:element name="lastName" type="xs:NCName"/>
  <xs:element name="company" type="xs:NCName"/>
  <xs:element name="city" type="xs:NCName"/>
  <xs:element name="phone" type="xs:integer"/>
</xs:schema>
```


**Remarque:**
Lorsque vous chargez un fichier XSD,
- les données sont sauvegardées dans le Repository et la métadonnées n’est pas affectée par la suppression ou le déplacement d’un fichier.
- vous pouvez choisir un élément comme racine de votre arborescence XML.

Pour charger un fichier XSD, procédez comme suit :

**Procédure**

1. Cliquez sur Browse... et parcourez votre système jusqu’au répertoire du fichier XSD à charger. Sinon, saisissez le chemin d’accès au fichier XML à charger.
2. Dans la boîte de dialogue qui apparaît, sélectionnez un élément de la liste Root, afin d’en faire la racine de votre arborescence XML et cliquez sur OK.

Un aperçu de la structure du fichier lu s’affiche dans la zone Schema Viewer. Vous avez la possibilité de développer et de visualiser tous les niveaux de la structure XML du fichier.
3. Renseignez le champ **Encoding** si le système ne l’a pas détecté automatiquement.

4. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez 0 si vous souhaitez l’effectuer sur toutes les colonnes.

5. Cliquez sur **Next** afin de configurer les paramètres du schéma.

**Définir un schéma**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous allez renseigner les paramètres à prendre en compte pour la définition du schéma.
La fenêtre de définition du schéma est composée de quatre vues :

<table>
<thead>
<tr>
<th>Vue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Schema</td>
<td>Arborescence du fichier XML.</td>
</tr>
<tr>
<td>Target Schema</td>
<td>Informations d'extraction et d'itération.</td>
</tr>
<tr>
<td>Preview</td>
<td>Aperçu du schéma cible, ainsi que des données d'entrée des colonnes sélectionnées, affichées dans l'ordre défini.</td>
</tr>
<tr>
<td></td>
<td><strong>Remarque:</strong> La fonctionnalité d'aperçu n'est pas disponible si vous chargez un fichier XSD.</td>
</tr>
<tr>
<td>File viewer</td>
<td>Fenêtre d'aperçu des données brutes.</td>
</tr>
</tbody>
</table>
Gestion des métadonnées dans l'intégration de données

Tout d’abord, définissez la boucle XPath ainsi que le nombre maximum d’itérations à effectuer. Pour cela :

**Procédure**

1. Peuplez le champ **Xpath loop expression**, avec l’expression Xpath absolue du nœud qui fait l’objet de l’itération. Vous pouvez le faire de deux manières :
   - Vous pouvez saisir l’expression entière ou appuyer sur Ctrl+Espace pour utiliser la liste d’autocomplétion.
   - Déposez le nœud sélectionné de l’arborescence **Source Schema** dans le champ **Absolute XPath expression**.

   Une flèche orange relie le nœud à l’expression correspondante.

   **Remarque:** Le champ **Xpath loop expression** est obligatoire.

2. Saisissez une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de nœuds à itérer ou saisissez –1 si vous souhaitez l’effectuer sur toutes les colonnes.

3. Définissez les champs à extraire en déposant les nœuds appropriés de la vue **Source Schema** vers le champ **Relative or absolute XPath expression**.

   **Remarque:** Vous pouvez sélectionner plusieurs nœuds à déposer dans le tableau, en appuyant sur Ctrl ou Maj, et en cliquant sur les nœuds adéquats. La flèche reliant un nœud sélectionné dans la zone **Source Schema** à celui déposé dans la zone **Fields to extract** est bleue. Les autres liens sont gris.
4. Vous pouvez ajouter autant de colonnes à extraire que vous le souhaitez, supprimer des colonnes, ou en modifier l'ordre, à l'aide de la barre d'outils.
   
   - Ajoutez ou supprimez une colonne à l'aide des boutons + et -
   - Modifiez l'ordre des colonnes à l'aide des boutons ↑ et ↓

5. Dans le champ Column name, nommez les libellés des colonnes qui s'afficheront dans la vue d'aperçu du schéma (Schema Preview).


   **Remarque:** La fonctionnalité d'aperçu n'est pas disponible si vous chargez un fichier XSD.

7. Cliquez sur Next pour vérifier et éditer le schéma final.

   Finaliser le schéma du fichier

   **Pourquoi et quand exécuter cette tâche**

   Le schéma généré affiche les colonnes sélectionnées dans le fichier XML et vous permet de personnaliser ce schéma.
**Procédure**

1. Si nécessaire, renommez la métadonnée dans le champ **Name**, *(metadatata, par défaut)*, ajoutez un commentaire dans le champ **Comment** et procédez à d'autres modifications, par exemple :
   - Définissez les colonnes en éditant les champs correspondants.
   - Ajoutez ou supprimez une colonne à l'aide des boutons + et -.
   - Modifiez l'ordre des colonnes à l'aide des boutons ↑ et ↓.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez **Java API Specification** (en anglais).

Les types de données **Talend** les plus utilisés sont les suivants :

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
Gestion des métadonnées dans l’intégration de données

- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

2. Si le fichier XML sur lequel se base le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.

3. Cliquez sur **Finish**. La nouvelle connexion au fichier, ainsi que son schéma, s’affichent sous le nœud **File XML** du **Repository**.

**Résultats**

Vous pouvez à présent déposer n’importe quelle connexion à un fichier ou n’importe quel schéma depuis le **Repository** dans l’espace de modélisation graphique en tant que nouveau composant **tFileInputXML** ou **tExtractXMLField**. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d’informations concernant l’utilisation des métadonnées centralisées, consultez **Comment utiliser les métadonnées centralisées dans un Job** à la page 438 et **Paramétrer un schéma du Repository dans un Job** à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file xml** afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

**Configurer les métadonnées pour un fichier de sortie (Output) XML**

Cette section décrit la définition et le chargement d’une connexion à un fichier de sortie (Output) XML.

**Définir les propriétés générales de la connexion File XML pour un fichier de sortie**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous allez définir les propriétés générales des métadonnées, telles que le nom (champ **Name**), l’objectif (champ **Purpose**) et une **Description**.

**Procédure**

1. Dans l’assistant de configuration des métadonnées du fichier, renseignez le champ **Name**, qui est obligatoire, et les champs **Purpose** et **Description** si vous le souhaitez. Les informations fournies dans le champ **Description** s’affichent en tant qu’info-bulle lorsque vous placez votre curseur sur la métadonnée.

**Remarque**:

Lorsque vous saisissez les propriétés générales des métadonnées à créer, vous devez définir le type de connexion, entrée (input) ou sortie (output). Il est donc conseillé de saisir des informations susceptibles de vous aider à distinguer les schémas d’entrée et de sortie.
Gestion des métadonnées dans l'intégration de données


3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File XML**. Notez que vous ne pouvez pas sélectionner de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.

4. Cliquez sur **Next** pour sélectionner le type de métadonnées.

**Sélectionner le type de métadonnées (Output)**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous allez définir le type de métadonnées, Input ou Output. Dans cet exemple, sélectionnez le type Output.

**Procédure**

1. Dans la boîte de dialogue, sélectionnez **Output XML**.
2. Cliquez sur **Next** pour définir le fichier de sortie, à partir d’un fichier XML, XSD ou à partir de rien.

**Définir la structure du fichier de sortie à l’aide d’un fichier XML existant**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez choisir de créer votre fichier manuellement ou de le créer à partir d’un fichier existant. Vous pouvez choisir de le créer manuellement (en sélectionnant **Create manually**), mais vous devrez alors configurer vous-même votre schéma, vos colonnes sources et vos colonnes cibles à l’étape 4 de l’assistant. Le fichier est créé lors de l’utilisation dans un Job d’un composant de sortie, comme le `tAdvancedFileOutputXML`.

Pour créer la structure XML de sortie à partir d’un fichier XML, procédez comme suit :

**Procédure**

1. Sélectionnez l’option **Create from a file**.
2. Cliquez sur le bouton **Browse**... à côté du champ **XML or XSD File**, parcourrez votre système jusqu’au fichier XML duquel la structure doit être appliquée au fichier de sortie et double-cliquez sur le fichier.

La zone **File Viewer** affiche l’aperçu de la structure XML et la zone **File Content** affiche au maximum les cinquante premières lignes du fichier.
3. Renseignez le champ **Encoding** si le système ne l’a pas détecté automatiquement.
4. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez 0 si vous souhaitez l’effectuer sur toutes les colonnes.
5. Dans la zone **Output File Path**, dans le champ **Output file**, renseignez le chemin d’accès à votre fichier de sortie. Si le fichier n’existe pas, il sera créé, lors de l’utilisation, dans un Job, du composant **tAdvancedFileOutputXML**. Si le fichier existe déjà, il sera écrasé.
6. Cliquez sur **Next** pour définir le schéma.
Définir la structure du fichier de sortie à l'aide d'un fichier XSD

Pourquoi et quand exécuter cette tâche

Cette procédure décrit comment définir la structure d'un fichier XML de sortie à partir d'un fichier XSD.

Remarque:
Lorsque vous chargez un fichier XSD,
• les données sont sauvegardées dans le Repository et la métadonnées n'est pas affectée par la suppression ou le déplacement d'un fichier.
• vous pouvez choisir un élément comme racine de votre arborescence XML.

Pour créer la structure XML de sortie à partir d'un fichier XSD, procédez comme suit :

Procédure

1. Sélectionnez l’option Create from a file.
2. Cliquez sur le bouton Browse... à côté du champ XML or XSD File, parcourez votre système jusqu’au fichier XSD duquel la structure doit être appliquée au fichier de sortie et double-cliquez sur le fichier.
3. Dans la boîte de dialogue qui apparaît, sélectionnez un élément de la liste Root, afin d’en faire la racine de votre arborescence XML et cliquez sur OK.
   La zone File Viewer affiche l’aperçu de la structure XML et la zone File Content affiche au maximum les cinquante premières lignes du fichier.
4. Renseignez le champ **Encoding** si le système ne l’a pas détecté automatiquement.

5. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez 0 si vous souhaitez l’effectuer sur toutes les colonnes.

6. Dans la zone **Output File Path**, dans le champ **Output file**, renseignez le chemin d’accès à votre fichier de sortie. Si le fichier n’existe pas, il sera créé, lors de l’utilisation, dans un Job, du composant **tAdvancedFileOutputXML**. Si le fichier existe déjà, il sera écrasé.

7. Cliquez sur **Next** pour définir le schéma.
Définition du schéma de votre fichier de sortie

Pourquoi et quand exécuter cette tâche

Lorsque les opérations précédentes sont terminées, les colonnes de la zone **Linker Source** sont automatiquement mappées vers celles correspondantes dans la zone **Linker Target**, comme le montrent les flèches bleues.

Dans cette étape, configurez le schéma de sortie. Le tableau suivant décrit comment faire :

<table>
<thead>
<tr>
<th>Pour...</th>
<th>Effectuer...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Créer un schéma à partir de zéro ou éditer les colonnes du schéma source à passer au schéma cible</td>
<td>Dans la zone <strong>Linker Source</strong>, cliquez sur le bouton <strong>Schema Management</strong> afin d’ouvrir l’éditeur de schéma.</td>
</tr>
<tr>
<td>Spécifiez le nombre maximum de colonnes à afficher dans la liste de schéma.</td>
<td>Dans le champ <strong>Field Limit</strong> de la zone <strong>Linker Source</strong>, spécifiez le nombre maximum de colonnes que vous souhaitez afficher si le nombre de colonnes de fichiers sources est supérieur à la limite définie dans les préférences du Studio. Toutes les colonnes à passer au schéma de sortie ne sont pas affichées dans la liste de schéma. Puis, cliquez sur le bouton Réactualiser.</td>
</tr>
<tr>
<td>Pour...</td>
<td>Effectuer...</td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
| Définir un élément répétable | Dans la zone Linker Target, cliquez-droit sur l’élément qui vous intéresse et sélectionnez Set As Loop Element dans le menu contextuel.  
**Remarque:** Cette opération est obligatoire pour définir un élément sur lequel effectuer une boucle. |
| Définir un élément de groupe | Dans la zone Linker Target, cliquez-droit sur l’élément qui vous intéresse et sélectionnez Set As Group Element dans le menu contextuel.  
**Remarque:** Vous pouvez configurer l’élément parent de l’élément répétable en tant que group element à condition que l’élément parent ne soit pas la racine de l’arborescence XML. |
| Créer un élément fils pour un élément | Dans la zone Linker Target,  
- Cliquez-droit sur l’élément qui vous intéresse et sélectionnez Add Sub-element dans le menu contextuel, saisissez un nom pour le sous-élément dans la boîte de dialogue qui apparaît, puis cliquez sur OK.  
- Sélectionnez l’élément qui vous intéresse, cliquez sur le bouton [+] en bas, sélectionnez Create as sub-element dans la boîte de dialogue qui apparaît, puis cliquez sur OK. Ensuite, saisissez un nom pour le sous-élément dans la boîte de dialogue suivante et cliquez sur OK. |
| Créer un attribut pour un élément | Dans la zone Linker Target,  
- Cliquez-droit sur l’élément qui vous intéresse et sélectionnez Add Attribute dans le menu contextuel, saisissez un nom pour l’attribut dans la boîte de dialogue qui apparaît et cliquez sur OK.  
- Sélectionnez l’élément qui vous intéresse, cliquez sur le bouton [+ ]en bas, sélectionnez Create as attribute dans la boîte de dialogue qui apparaît, puis cliquez sur OK. Ensuite, saisissez un nom pour l’attribut dans la boîte de dialogue suivante et cliquez sur OK. |
| Créer un espace de noms pour un élément | Dans la zone Linker Target,  
- Cliquez-droit sur l’élément qui vous intéresse et sélectionnez Add Name Space dans le menu contextuel, saisissez un nom pour l’espace de noms dans la boîte de dialogue qui apparaît et cliquez sur OK.  
- Sélectionnez l’élément qui vous intéresse, cliquez sur le bouton [+ ]en bas, sélectionnez Create as name space dans la boîte de dialogue qui apparaît, puis cliquez sur OK. Ensuite, saisissez un nom pour l’espace de noms dans la boîte de dialogue suivante et cliquez sur OK. |
| Supprimer un ou plusieurs élément(s)/attribut(s)/espace(s) de noms | Dans la zone Linker Target,  
- Cliquez-droit sur les éléments/attributs/espace(s) de noms qui vous intéressent et sélectionnez Delete dans le menu contextuel.  
- Sélectionnez les éléments/attributs/espace(s) de noms qui vous intéressent et cliquez sur le bouton x en bas.  
- Sélectionnez les éléments/attributs/espace(s) de noms qui vous intéressent et appuyez sur la touche Suppr.  
**Remarque:** Supprimer un élément supprime également ses enfants, s’il en a. |
| Ajuster l’ordre d’un ou plusieurs élément(s) | Dans la zone Linker Target, sélectionnez l’élément qui vous intéressent et cliquez sur les boutons ↑ et ↓. |
Pour... | Effectuer...
---|---
Configurer une valeur statique pour un élément/attribut/espace de noms | Dans la zone **Linker Target**, cliquez-droit sur l’élément/l’attribut/l’espace de noms qui vous intéresse et sélectionnez **Set A Fix Value** dans le menu contextuel.

**Remarque:**
- La valeur que vous avez configurée va remplacer toute valeur récupérée du flux d’entrée dans votre Job.
- Vous pouvez configurer une valeur statique pour un élément fils de l’élément répétable uniquement, à condition que l’élément n’ait pas lui-même d’enfants et pas de mapping source-cible sur lui.

Créer un mapping source-cible | Sélectionnez la colonne qui vous intéresse, dans la zone **Linker Source**, déposez-la sur le nœud qui vous intéresse de la zone **Linker Target** et sélectionnez **Create as sub-element of target node**, **Create as attribute of target node**, ou **Add linker to target node** selon vos besoins, dans la boîte de dialogue qui apparaît, puis cliquez sur OK.
Si vous choisissez une option qui n’est pas permise pour le nœud cible, un message d’avertissement apparaît et l’opération échoue.

Supprimer un mapping source-cible | Dans la zone **Linker Target**, cliquez-droit sur le nœud qui vous intéresse et sélectionnez **Disconnect Linker** dans le menu contextuel.

Créer une arborescence XML à partir d’un autre fichier XML ou XSD | Cliquez-droit sur tout élément du schéma dans la zone **Linker Target** et sélectionnez **Import XML Tree** dans le menu contextuel afin de charger un autre fichier XML ou XSD.
Ensuite, vous devez créer manuellement les mappings source-cible et définir à nouveau le schéma de sortie.

**Remarque:**
Vous pouvez appuyer sur les touches Ctrl ou Maj si vous souhaitez effectuer les opérations de mapping via la sélection multiple. La sélection multiple fonctionne également pour les opérations disponibles par clic-droit.

**Procédure**
1. Dans la zone **Linker Target**, cliquez-droit sur l’élément sur lequel vous souhaitez effectuer une boucle, puis, dans le menu contextuel, sélectionnez **Set As Loop Element**.
2. Définissez les autres propriétés de fichier Output, puis cliquez sur **Next** pour passer à l’étape suivante.

**Finaliser le schéma de votre fichier de sortie**

**Pourquoi et quand exécuter cette tâche**

L’étape 5 de l’assistant affiche le schéma final généré et vous permet de le modifier.
Gestion des métadonnées dans l'intégration de données

Procédure

1. Si nécessaire, renommez la métadonnée dans le champ **Name** (*metadata*, par défaut), ajoutez un commentaire dans le champ **Comment** et procédez à d'autres modifications, par exemple :
   - Définissez les colonnes en modifiant les champs correspondants.
   - Ajoutez ou supprimez une colonne à l'aide des boutons [+ et -].
   - Modifiez l'ordre des colonnes à l'aide des boutons [↑ et ↓].

2. Si le fichier XML sur lequel se base le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.

3. Cliquez sur **Finish**. Le nouveau schéma apparaît dans le **Repository**, sous le nœud **File XML** correspondant.

Résultats

Vous pouvez à présent déposer n’importe quelle connexion à un fichier ou n’importe quel schéma depuis le **Repository** dans l’espace de modélisation graphique en tant que nouveau composant **tAdvancedFileOutputXML**.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file xml** afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.
Gestion des métadonnées dans l'intégration de données

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

Centraliser des métadonnées d'un fichier Excel

Si vous devez souvent écrire des données dans et/ou lire des données depuis une feuille de calcul Excel, vous pouvez centraliser les informations de connexion et le schéma de ce fichier dans le Repository afin de les réutiliser facilement. Ainsi, vous n'aurez pas à définir ces métadonnées manuellement pour chaque composant à chaque fois que vous souhaitez utiliser ce fichier.

Vous pouvez centraliser une connexion à un fichier Excel depuis un fichier Excel existant ou depuis les propriétés de ce fichier définies dans un Job.

Pour centraliser une connexion ainsi que le schéma d'un fichier Excel, développez le nœud Metadata du Repository, cliquez-droit sur File Excel et sélectionnez Create file Excel depuis le menu contextuel afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l'icône dans la vue Basic settings du composant de base de données correspondant. Sa propriété Property Type doit être Built-in afin d’ouvrir l’assistant de configuration de connexion à une base de données.

Suivez ensuite les étapes suivantes dans l'assistant afin de :

- Définir les informations qui identifient la connexion au fichier. Pour plus d’informations, consultez Définir les propriétés générales de la connexion File Excel à la page 331.
- Charger le fichier. Pour plus d’informations, consultez Charger le fichier à la page 332.
- Analyser le fichier afin de récupérer son schéma. Pour plus d’informations, consultez Analyser le fichier à la page 334.
- Finaliser le schéma du fichier. Pour plus d’informations, consultez Finaliser le schéma de votre fichier Excel à la page 335.

Vous pouvez à présent déposer la connexion au fichier, ou son schéma, depuis le Repository dans l’espace de modélisation graphique en tant que nouveau composant. Pour plus d’informations concernant l'utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Edit file Excel afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve Schema dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

Définir les propriétés générales de la connexion File Excel

Procédure

1. Dans l’assistant de configuration des métadonnées, renseignez le champ Name, qui est obligatoire, et les champs Purpose et Description si vous le souhaitez. Les informations fournies
dans le champ **Description** s’affichent en tant qu’info-bulle lorsque vous placez votre curseur sur la métadonnée.


3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File Excel**. Ce dossier contient la connexion au fichier juste créée.

4. Cliquez sur **Next** pour passer aux paramètres du fichier.

### Charger le fichier

**Procédure**

1. Spécifiez le chemin entier de votre fichier source dans le champ **File**, ou cliquez sur le bouton **Browse…** pour rechercher le fichier.

**Remarque:** La notation de chemin Convention Universelle de Nommage (UNC) n’est pas supportée. Si votre fichier source est sur un hôte LAN, vous pouvez d’abord mapper le dossier du réseau dans un lecteur local.

Passez cette étape si vous sauvegardez la connexion à un fichier Excel définie dans un composant car son chemin est déjà défini dans le champ **File**.
2. Si le fichier chargé provient de Excel 2007, assurez-vous que la case Read excel2007 file format(xlsx) soit cochée.

3. Par défaut, le mode Memory-consuming (User mode) est sélectionné. Cependant, si le fichier .xlsx chargé est volumineux, sélectionnez Less memory consumed for large excel(Event mode) dans la liste Generation mode pour éviter les erreurs de mémoire.

4. Dans la zone File viewer and sheets settings, sélectionnez la (les) feuille(s) que vous souhaitez utiliser.
   - Dans la liste Please select sheet, sélectionnez la feuille dont vous souhaitez avoir un aperçu. La table d’aperçu affiche le contenu de la feuille sélectionnée.
     Par défaut, la table affiche la première feuille du fichier.
   - Dans la liste Set sheets parameters, cochez la case à côté de la (les) feuille(s) que vous souhaitez charger.
     Si vous sélectionnez plus d’une feuille, le schéma final sera une combinaison des structures de toutes les feuilles sélectionnées.
5. Cliquez sur **Next** pour passer à l’étape suivante.

**Analyser le fichier**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous pouvez définir les différents paramètres de votre fichier afin que le schéma soit correctement récupéré.

**Procédure**

1. Selon votre fichier Excel, spécifiez l’encodage (dans le champ Encoding), le séparateur avancé pour les nombres (dans le champ Advanced separator (for number)) ainsi que le nombre de lignes à ignorer en en-tête (Header) ou en pied de page (Footer).

![Image de l'étape 1 de l'analyse de fichier](image1.png)

2. Si nécessaire, vous pouvez définir précisément le nombre de colonnes à lire à l’aide des champs **First column** et **Last column**. Par exemple, si vous souhaitez ignorer la première colonne car elle ne contient pas de données à utiliser, saisissez 2 dans le champ **First column** afin de définir la deuxième colonne du schéma comme étant la première.

   Afin de récupérer le schéma d’un fichier Excel, vous n’avez pas besoin d’analyser toutes les lignes du fichier, en particulier lorsque le fichier à charger est volumineux. Afin de limiter le nombre de lignes à analyser, cochez la case **Limit** dans la zone **Limit Of Rows** et définissez le nombre de lignes désirées.

![Image de l'étape 2 de l'analyse de fichier](image2.png)

3. Si le fichier Excel a une ligne d’en-tête, cochez la case **Set heading row as column names** afin de prendre en compte le nom des en-têtes. Cliquez sur **Refresh** afin de voir le résultat des changements précédents dans la table d’aperçu.

![Image de l'étape 3 de l'analyse de fichier](image3.png)
4. Cliquez sur **Next** pour continuer.

**Finaliser le schéma de votre fichier Excel**

**Pourquoi et quand exécuter cette tâche**

La dernière étape de l’assistant affiche le schéma final généré et vous permet de le personnaliser selon vos besoins.

Remarquez que tous les caractères du fichier pouvant être mal interprétés par le programme par la suite sont remplacés par un caractère neutre. Par exemple, les tirets bas remplacent les astérisques.
**Procédure**

1. Si nécessaire, renommez le schéma (par défaut, `metadata`) et saisissez un commentaire.
   Personnalisez le mapping, si nécessaire. Ajouter, supprimer ou déplacer les colonnes du schéma, exporter le schéma dans un fichier XML ou remplacer le schéma en important une définition de schéma depuis un fichier XML à l'aide de la barre d'outils.
   Assurez-vous que le type de données dans la colonne **Type** est correctement défini.
   Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](en anglais).

Les types de données **Talend** les plus utilisés sont les suivants :

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant `tFileInputRaw` en spécifiant qu'il comporte un type de données Object.
- **List** : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément `xsd:list`.
- **Document** : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
2. Si le fichier Excel sur lequel est basé le schéma a changé, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.

3. Cliquez sur **Finish**. Le nouveau schéma apparaît dans le **Repository**, sous le nœud **File Excel** correspondant.

## Centraliser les métadonnées d'un fichier LDIF

**Pourquoi et quand exécuter cette tâche**

Les fichiers LDIF sont des fichiers annuaires distincts par leurs attributs. Si vous devez souvent lire un fichier LDIF, vous pouvez centraliser les informations de connexion et les attributs de ce fichier dans le **Repository** afin de les réutiliser facilement. Ainsi, vous n’aurez pas à définir ces métadonnées manuellement pour chaque composant à chaque fois que vous souhaitez utiliser ce fichier.

Vous pouvez centraliser une connexion à un fichier LDIF depuis un fichier LDIF existant ou depuis les propriétés de ce fichier définies dans un Job.

Pour centraliser une connexion ainsi que le schéma d’un fichier LDIF, développez le nœud **Metadata** du **Repository**, cliquez-droit sur **File ldif** et sélectionnez **Create file ldif** depuis le menu contextuel afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône ![icône](...) dans la vue **Basic settings** du composant de base de données correspondant. Sa propriété **Property Type** doit être **Built-in** afin d’ouvrir l’assistant de configuration de connexion à une base de données.

Suivez ensuite les étapes suivantes dans l’assistant :

**Procédure**

1. Afin d’identifier la métadonnées LDIF, renseignez les informations génériques, notamment **Name**, **Purpose** et **Description**

   Le champ **Name** est obligatoire et les informations que vous renseignez dans le champ **Description** apparaissent en tant qu’info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.

3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File ldif**. Ce dossier contient la connexion au fichier juste créée.

   Cliquez sur **Next** pour passer aux paramètres du fichier.

4. Spécifiez le chemin entier de votre fichier source dans le champ **File**, ou cliquez sur le bouton **Browse**... pour rechercher le fichier.

   **Remarque:** La notation de chemin Convention Universelle de Nommage (UNC) n’est pas supportée. Si votre fichier source est sur un hôte LAN, vous pouvez d’abord mapper le dossier du réseau dans un lecteur local.

   Passez cette étape si vous sauvegardez la connexion à un fichier LDIF définie dans un composant car son chemin est déjà défini dans le champ **File**.
5. La zone **File Viewer** affiche les 50 premières lignes du fichier. Cliquez sur **Next** pour continuer.

7. Si besoin, personnalisez le schéma généré.

- Renommez le schéma (par défaut `metadata`) et laissez un commentaire.
- Ajouter, supprimer ou déplacer les colonnes du schéma, exporter le schéma dans un fichier XML ou remplacer le schéma en important une définition de schéma depuis un fichier XML à l'aide de la barre d'outils.
Assurez-vous que le type de données dans la colonne Type est correctement défini.
Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données Talend les plus utilisés sont les suivants :

- **Object** : est un type de données Talend générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant tFileInputRaw en spécifiant qu’il comporte un type de données Object.
- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

8. Si le fichier LDIF sur lequel est basé le schéma a changé, cliquez sur le bouton Guess afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité Guess ne retient pas ces modifications.
Résultats

Vous pouvez à présent déposer la connexion au fichier, ou son schéma, depuis le Repository dans l'espace de modélisation graphique en tant que nouveau composant. Pour plus d'informations concernant l'utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Edit file ldif afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve Schema dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

Centraliser les métadonnées d'un fichier JSON

Si vous utilisez fréquemment un fichier JSON, vous pouvez utiliser l'assistant New Json File pour centraliser la connexion au fichier, vos instructions de requêtes XPath et la structure des données dans le Repository afin de les réutiliser facilement.

Selon l'option sélectionnée, l'assistant vous permet de créer une connexion à un fichier d'entrée ou de sortie. Dans un Job, les composants tFileInputJSON et tExtractJSONFields utilisent le schéma d'entrée créé pour lire des fichiers ou des champs JSON et le tWriteJSONField utilise le schéma de sortie créé pour écrire un champ JSON pouvant être sauvegardé dans un fichier par le tFileOutputJSON ou extrait par le tExtractJSONFields.

Pour plus d'informations concernant la configuration des métadonnées JSON d'entrée, consultez Configurer les métadonnées JSON pour un fichier d'entrée à la page 342.

Pour plus d'informations concernant la configuration des métadonnées JSON de sortie, consultez Configurer les métadonnées JSON pour un fichier de sortie à la page 351.

Dans la vue Repository, développez le nœud Metadata, cliquez-droit sur File JSON et sélectionnez Create JSON Schema dans le menu contextuel pour ouvrir l'assistant New Json File.

Configurer les métadonnées JSON pour un fichier d'entrée

Cette section décrit comment définir et charger une connexion à un fichier et charger un schéma JSON pour un fichier d'entrée.

Vous pouvez glisser-déposer la connexion au fichier ou son schéma de la vue Repository dans l’espace de modélisation graphique, en tant que composant tFileInputJSON ou tExtractJSONFields, ou sur un composant existant afin de réutiliser les métadonnées. Pour plus d’informations concernant l’utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Edit JSON pour ouvrir l’assistant de configuration du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le Repository et sélectionnez Retrieve Schema dans le menu contextuel.
Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

**Définir les propriétés générales de la connexion File JSON pour un fichier d’entrée**

**Procédure**

1. Dans l’assistant, renseignez les informations générales dans les champs correspondants afin d’identifier les métadonnées du fichier JSON, notamment les champs Name, Purpose et Description.

Le champ Name est obligatoire et les informations que vous renseignez dans le champ Description apparaissent en tant qu’info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.

**Remarque:**

Dans cette étape, il est recommandé de saisir les informations vous permettant de distinguer vos connexions d’entrée et de sortie, puisque l’étape d’après vous demande de choisir entre les deux.

2. Si nécessaire, définissez la version et le statut de la connexion dans les champs Version et Status, respectivement.

Vous pouvez également gérer la version et le statut d’un élément du Repository dans la boîte de dialogue Project Settings. Pour plus d’informations, consultez Gestion des versions à la page 479 et Gestion du statut à la page 481 respectivement.
3. Au besoin, cliquez sur le bouton Select à côté du champ Path afin de sélectionner un dossier sous le nœud File Json pour contenir votre nouvelle connexion à un fichier.
4. Cliquez sur Next pour sélectionner le type de métadonnées.

**Configurer le type de métadonnées et charger le fichier d'entrée**

**Procédure**

1. Dans la boîte de dialogue, sélectionnez Input Json et cliquez sur Next pour passer à l’étape suivante et charger le fichier d'entrée.

![New Json File](image)

2. Dans la liste Read By, sélectionnez le type de requête pour lire le fichier source JSON.
   - **JwtPath** : lit les données JSON selon une requête JsonPath.
     Ce type est le type de requêtes par défaut recommandé pour lire des données JSON afin de gagner en performance et d’éviter les problèmes que vous pouvez rencontrer lors de la lecture de données JSON à partir d’une requête XPath.
   - **Xpath** : lit les données JSON selon une requête XPath.

3. Cliquez sur Browse... et parcourez votre répertoire jusqu’au fichier JSON à charger. Vous pouvez également saisir le chemin d’accès complet ou l’URL du fichier JSON.
Dans cet exemple, le fichier JSON d’entrée contient ceci :

```json
{ "movieCollection": [ 
    { "type": "Action Movie",  
      "name": "Brave Heart",  
      "details": {  
        "release": "1995",  
        "rating": "5",  
        "starring": "Mel Gibson"  
      }  
    },  
    { "type": "Action Movie",  
      "name": "Edge of Darkness",  
      "details": {  
        "release": "2010",  
        "rating": "5",  
        "starring": "Mel Gibson"  
      }  
    }  
] }
```

La zone **Schema Viewer** affiche une prévisualisation de la structure JSON. Vous pouvez développer et visualiser chaque niveau de la structure du fichier JSON.
4. Renseignez le champ **Encoding** si le système ne l’a pas détecté automatiquement.

5. Dans le champ **Limit**, définissez le nombre de niveaux dans la profondeur hiérarchique JSON, profondeur à laquelle vous souhaitez limiter la requête JsonPath ou XPath. Saisissez 0 pour ne pas configurer de limite.
   Configurez une valeur inférieure à 5 pour ce paramètre peut empêcher l’assistant de s’arrêter, dans le cas d’un fichier JSON volumineux.

6. Cliquez sur **Next** afin de configurer les paramètres du schéma.

**Définir le schéma JSON de votre fichier d’entrée**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, renseignez les paramètres du schéma.
La fenêtre de définition du schéma est composée de quatre vues :

<table>
<thead>
<tr>
<th>Vue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Schema</td>
<td>Arborescence du fichier JSON.</td>
</tr>
<tr>
<td>Target Schema</td>
<td>Informations d'extraction et d'itération.</td>
</tr>
<tr>
<td>Vue</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Preview</td>
<td>Aperçu du schéma cible, ainsi que des données d’entrée des colonnes sélectionnées, affichées dans l’ordre défini.</td>
</tr>
<tr>
<td>File viewer</td>
<td>Aperçu des données du fichier JSON.</td>
</tr>
</tbody>
</table>

**Procédure**

1. Alimentez le champ **Path loop expression**, avec l’expression absolue JsonPath ou Xpath, selon le type de requête sélectionné, du nœud qui fait l’objet de l’itération. Vous pouvez le faire de deux manières :
   - Déposez le nœud sélectionné de l’arborescence **Source Schema** dans le champ **Absolute path expression** de la table **Path loop expression**.

   Une flèche orange relie le nœud à l’expression correspondante.

   ![Diagramme de processus](image)

   **Remarque:** Le champ **Path loop expression** est obligatoire.

2. Saisissez une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de nœuds à itérer.

3. Glissez-déposez les nœuds appropriés de la vue **Source Schema** vers les champ **Relative or absolute path expression** de la table **Fields to extract**.

   ![Diagramme de processus](image)
Remarque: Vous pouvez sélectionner plusieurs nœuds à déposer dans le tableau, en appuyant sur Ctrl ou Maj, et en cliquant sur les nœuds adéquats.

4. Vous pouvez ajouter autant de colonnes à extraire que vous le souhaitez, supprimer des colonnes, ou en modifier l’ordre, à l’aide de la barre d’outils.
   • Ajoutez ou supprimez une colonne à l’aide des boutons [+] et [-].
   • Modifiez l’ordre des colonnes à l’aide des boutons ↑ et ↓.

5. Si vous souhaitez que votre schéma contienne des noms de colonnes différents de ceux récupérés dans le fichier d’entrée, saisissez dans la colonne Column name le libellé des colonnes à afficher dans la zone Preview du schéma.


7. Cliquez sur Next pour finaliser le schéma.

Finaliser le schéma JSON de votre fichier d’entrée

Pourquoi et quand exécuter cette tâche

La dernière étape de l’assistant affiche le schéma final généré et vous permet de le personnaliser selon vos besoins.
Procédure

1. Si nécessaire, renommez le schéma (par défaut, metadata) et saisissez un commentaire. Personnalisez le mapping, si nécessaire. Ajouter, supprimer ou déplacer les colonnes du schéma, exporter le schéma dans un fichier XML ou remplacer le schéma en important une définition de schéma depuis un fichier XML à l’aide de la barre d’outils.

Assurez-vous que le type de données dans la colonne Type est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données Talend les plus utilisés sont les suivants :

- **Object** : est un type de données Talend générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant tFileInputRaw en spécifiant qu’il comporte un type de données Object.
- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.
2. Si le fichier JSON sur lequel est basé le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.

3. Cliquez sur **Finish**. La nouvelle connexion au fichier, avec son schéma, s’affiche sous le nœud **File Json** de la vue **Repository**.

**Configurer les métadonnées JSON pour un fichier de sortie**

Cette section décrit comment définir des métadonnées JSON pour un fichier de sortie.

Vous pouvez glisser-déposer la connexion au fichier ou son schéma de la vue **Repository** dans l’espace de modélisation graphique, en tant que composant **tWriteJSONField**, ou sur un composant existant afin de réutiliser les métadonnées. Pour plus d’informations concernant l’utilisation des métadonnées centralisées, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438 et Paramétrer un schéma du Repository dans un Job à la page 57.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit JSON** pour ouvrir l’assistant de configuration du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

**Définir les propriétés générales de la connexion File JSON pour un fichier de sortie**

**Procédure**

1. Dans l’assistant, renseignez les informations générales dans les champs correspondants afin d’identifier les métadonnées du fichier JSON, notamment les champs **Name**, **Purpose** et **Description**.

   Le champ **Name** est obligatoire et les informations que vous renseignez dans le champ **Description** apparaissent en tant qu’info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.

   **Remarque:**

   Dans cette étape, il est recommandé de saisir les informations vous permettant de distinguer vos connexions d’entrée et de sortie, puisque l’étape d’après vous demande de choisir entre les deux.
2. Si nécessaire, définissez la version et le statut de la connexion dans les champs Version et Status, respectivement.
   Vous pouvez également gérer la version et le statut d’un élément du Repository dans la boîte de dialogue Project Settings. Pour plus d’informations, consultez Gestion des versions à la page 479 et Gestion du statut à la page 481 respectivement.
3. Au besoin, cliquez sur le bouton Select à côté du champ Path afin de sélectionner un dossier sous le nœud File Json pour contenir votre nouvelle connexion à un fichier.
4. Cliquez sur Next pour configurer le type de métadonnées.

Configurer le type de métadonnées et charger le fichier JSON modèle

Pourquoi et quand exécuter cette tâche
Dans cette étape, vous allez définir le type de schéma, Input ou Output. Dans cet exemple, sélectionnez le type Output.

Procédure
2. Choisissez de créer vos métadonnées manuellement ou à partir d’un fichier modèle JSON existant. Si vous choisissez de le créer manuellement (en sélectionnant Create manually), vous devez configurer vous-même votre schéma, vos colonnes sources et vos colonnes cibles. Le fichier sera créé lors de l’utilisation dans un Job d’un composant de sortie, comme le tWriteJSONField. Dans cet exemple, créez le schéma de sortie en chargeant un fichier JSON existant. Sélectionnez donc l’option Create from a file.

3. Cliquez sur le bouton Browse... à côté du champ JSON File, parcourez votre système jusqu’au fichier JSON dont la structure va être appliquée au fichier/champ de sortie JSON et double-cliquez sur le fichier. Sinon, vous pouvez saisir le chemin complet ou l’URL pointant vers le fichier JSON modèle.

La zone File Viewer affiche un aperçu de la structure JSON et la zone File Content affiche au maximum les cinquante premières lignes du fichier.
4. Renseignez le champ **Encoding** si le système ne l’a pas détecté automatiquement.

5. Dans le champ **Limit**, définissez le nombre de niveaux dans la profondeur hiérarchique JSON, profondeur à laquelle vous souhaitez limiter la requête JsonPath ou XPath. Saisissez 0 pour ne pas configurer de limite.
   
   Configurer une valeur inférieure à 5 pour ce paramètre peut empêcher l’assistant de s’arrêter, dans le cas d’un fichier JSON volumineux.

6. Vous pouvez également, de manière facultative, spécifier un chemin vers un fichier de sortie.

7. Cliquez sur **Next** pour définir le schéma.
Définir le schéma JSON de votre fichier de sortie

Pourquoi et quand exécuter cette tâche

Lorsque les opérations précédentes sont terminées, les colonnes de la zone **Linker Source** sont automatiquement mappées vers celles correspondantes dans la zone **Linker Target**, comme le montrent les flèches bleues.

Dans cette étape, configurez le schéma de sortie. Le tableau suivant décrit comment faire :

<table>
<thead>
<tr>
<th>Pour...</th>
<th>Effectuer...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Définir un élément répétable</td>
<td>Dans la zone <strong>Linker Target</strong>, cliquez-droit sur l’élément qui vous intéresse et sélectionnez <strong>Set As Loop Element</strong> dans le menu contextuel.</td>
</tr>
</tbody>
</table>

**Remarque:**
Cette opération est obligatoire pour définir un élément sur lequel effectuer une boucle.
<table>
<thead>
<tr>
<th>Pour...</th>
<th>Effectuer...</th>
</tr>
</thead>
</table>
| Définir un élément de groupe | Dans la zone **Linker Target**, cliquez-droit sur l’élément qui vous intéresse et sélectionnez **Set As Group Element** dans le menu contextuel.  
**Remarque:** Vous pouvez configurer l’élément parent de l’élément répétatif en tant que group element à condition que l’élément parent ne soit pas la racine de l’arborescence JSON. |
| Créer un élément fils pour un élément | Dans la zone **Linker Target**,  
- Cliquez-droit sur l’élément qui vous intéresse et sélectionnez **Add Sub-element** dans le menu contextuel, saisissez un nom pour le sous-élément dans la boîte de dialogue qui apparaît, puis cliquez sur OK.  
- Sélectionnez l’élément qui vous intéresse, cliquez sur le bouton [*] en bas, sélectionnez **Create as sub-element** dans la boîte de dialogue qui apparaît, puis cliquez sur OK.  
Ensuite, saisissez un nom pour le sous-élément dans la boîte de dialogue suivante et cliquez sur OK. |
| Créer un attribut pour un élément | Dans la zone **Linker Target**,  
- Cliquez-droit sur l’élément qui vous intéresse et sélectionnez **Add Attribute** dans le menu contextuel, saisissez un nom pour l’attribut dans la boîte de dialogue qui apparaît et cliquez sur OK.  
- Sélectionnez l’élément qui vous intéresse, cliquez sur le bouton [*] en bas, sélectionnez **Create as attribute** dans la boîte de dialogue qui apparaît, puis cliquez sur OK.  
Ensuite, saisissez un nom pour l’attribut dans la boîte de dialogue suivante et cliquez sur OK. |
| Créer un espace de noms pour un élément | Dans la zone **Linker Target**,  
- Cliquez-droit sur l’élément qui vous intéresse et sélectionnez **Add Name Space** dans le menu contextuel, saisissez un nom pour l’espace de noms dans la boîte de dialogue qui apparaît et cliquez sur OK.  
- Sélectionnez l’élément qui vous intéresse, cliquez sur le bouton [*] en bas, sélectionnez **Create as name space** dans la boîte de dialogue qui apparaît, puis cliquez sur OK.  
Ensuite, saisissez un nom pour l’espace de noms dans la boîte de dialogue suivante et cliquez sur OK. |
| Supprimer un ou plusieurs élément(s)/attribut(s)/espace(s) de noms | Dans la zone **Linker Target**,  
- Cliquez-droit sur les éléments/attributs/espaces de noms qui vous intéressent et sélectionnez **Delete** dans le menu contextuel.  
- Sélectionnez les éléments/attributs/espaces de noms qui vous intéressent et cliquez sur le bouton x en bas.  
- Sélectionnez les éléments/attributs/espaces de noms qui vous intéressent et appuyez sur la touche **Suppr**.  
**Remarque:** Supprimer un élément supprime également ses enfants, s’il en a. |
<p>| Ajuster l’ordre d’un ou plusieurs élément(s) | Dans la zone <strong>Linker Target</strong>, sélectionnez l’élément qui vous intéresse et cliquez sur les boutons et . |</p>
<table>
<thead>
<tr>
<th>Pour...</th>
<th>Effectuer...</th>
</tr>
</thead>
</table>
| Configurer une valeur statique pour un élément/attribut/espace de noms | Dans la zone **Linker Target**, cliquez-droit sur l’élément/l’attribut/l’espace de noms qui vous intéresse et sélectionnez **Set A Fix Value** dans le menu contextuel.  
**Remarque:**  
- La valeur que vous avez configurée va remplacer toute valeur récupérée du flux d’entrée dans votre Job.  
- Vous pouvez configurer une valeur statique pour un élément fils de l’élément répétable uniquement, à condition que l’élément n’ait pas lui-même d’enfants et pas de mapping source-cible sur lui. |
| Créer un mapping source-cible | Sélectionnez la colonne qui vous intéresse, dans la zone **Linker Source**, déposez-la sur le nœud qui vous intéresse de la zone **Linker Target** et sélectionnez **Create as sub-element of target node**, **Create as attribute of target node**, ou **Add linker to target node** selon vos besoins, dans la boîte de dialogue qui apparaît, puis cliquez sur **OK**.  
Si vous choisissez une option qui n’est pas permise pour le nœud cible, un message d’avertissement apparaît et l’opération échoue. |
| Supprimer un mapping source-cible | Dans la zone **Linker Target**, cliquez-droit sur le nœud qui vous intéresse et sélectionnez **Disconnect Linker** dans le menu contextuel. |
| Créer une structure JSON à partir d’un autre fichier JSON | Cliquez-droit sur tout élément du schéma dans la zone **Linker Target** et sélectionnez **Import XML Tree** dans le menu contextuel afin de charger un autre fichier JSON. Ensuite, vous devez créer manuellement les mappings source-cible et définir à nouveau le schéma de sortie. |

**Remarque:**  
Vous pouvez appuyer sur les touches **Ctrl** ou **Maj** si vous souhaitez effectuer les opérations de mapping via la sélection multiple. La sélection multiple fonctionne également pour les opérations disponibles par clic-droit.

**Procédure**  
1. Dans la zone **Linker Target**, cliquez-droit sur l’élément que vous souhaitez définir comme élément répétable et sélectionnez **Set As Loop Element** dans le menu contextuel.  
   Dans cet exemple, configurez la boucle sur l’élément `details`.  

357
2. Personnalisez le mapping, si nécessaire.
3. Cliquez sur Next pour finaliser le schéma.

**Finaliser le schéma JSON de votre fichier de sortie**

**Pourquoi et quand exécuter cette tâche**

La dernière étape de l’assistant affiche le schéma final généré et vous permet de le personnaliser selon vos besoins.
Gestion des métadonnées dans l'intégration de données

**Procédure**

1. Si nécessaire, renommez le schéma (par défaut, `metadata`) et saisissez un commentaire. Personnalisez le mapping, si nécessaire. Ajouter, supprimer ou déplacer les colonnes du schéma, exporter le schéma dans un fichier XML ou remplacer le schéma en important une définition de schéma depuis un fichier XML à l'aide de la barre d'outils.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données **Talend** les plus utilisés sont les suivants :

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant `tFileInputRaw` en spécifiant qu'il comporte un type de données Object.
- **List** : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément `xsd:list`.
- **Document** : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
2. Si le fichier JSON sur lequel est basé le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.

3. Cliquez sur **Finish**. La nouvelle connexion au fichier, avec son schéma, s’affiche sous le nœud **File Json** de la vue **Repository**.

**Centraliser les métadonnées d’une connexion LDAP**

Si vous devez souvent accéder à un répertoire LDAP, vous pouvez centraliser les informations de connexion à un serveur LDAP dans le **Repository** afin de les réutiliser facilement.

Vous pouvez créer une connexion LDAP depuis un répertoire LDAP disponible ou à partir des paramètres définis dans un Job.

Afin de créer une connexion depuis un répertoire LDAP disponible, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **LDAP** puis sélectionnez **Create LDAP schema** dans le menu contextuel afin d’ouvrir l’assistant **Create new LDAP schema**.

Pour centraliser les paramètres de connexion à la base de données définis dans un Job, cliquez sur l’icône dans la vue **Basic settings** du composant de base de données correspondant. Sa propriété **Property Type** doit être **Built-in** afin d’ouvrir l’assistant de configuration de connexion à une base de données (**Create new LDAP schema**).

Contrairement à l’assistant de connexion DB, l’assistant LDAP regroupe à la fois la connexion au serveur et la définition du schéma dans une procédure en cinq étapes.

Vous pouvez à présent déposer n’importe quelle connexion à un fichier ou n’importe quel schéma depuis le **Repository** dans l’espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit LDAP schema** afin d’ouvrir l’assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

**Définir les propriétés générales de la connexion LDAP**

**Procédure**

1. Dans cet assistant, définissez les propriétés générales du fichier, afin d’identifier les métadonnées de la connexion LDAP, telles que le nom (champ **Name**), l’objectif (champ **Purpose**) et une **Description**.

   Le champ **Name** est obligatoire et les informations que vous renseignez dans le champ **Description** apparaissent en tant qu’info-bulle lorsque vous placez le pointeur de votre souris sur la connexion LDAP.

2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d’un élément du **Repository**.
dans la boîte de dialogue **Project Settings**. Pour plus d’informations, consultez **Gestion des versions** à la page 479 et **Gestion du statut** à la page 481 respectivement.

3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **LDAP** pour contenir la nouvelle connexion LDAP créée.

4. Cliquez sur **Next** pour définir les détails de connexion.

### Définir la connexion au serveur

**Procédure**

1. Renseignez les informations de connexion.

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td>Serveur LDAP adresse IP</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>Port d’écoute pour l’annuaire LDAP</td>
</tr>
</tbody>
</table>
| **Encryption method** | LDAP : aucun chiffrement n’est utilisé  
|               | LDAPS : LDAP sécurisé.               
|               | TLS : le certificat est utilisé      |

2. Vérifiez votre connexion à l’aide de bouton **Check Network Parameter**, puis cliquez sur le bouton **Next**.

3. Cliquez sur **Next** pour passer à l’étape suivante.

###Configurer les paramètres d’accès à LDAP

**Procédure**

1. Dans cette fenêtre, renseignez les informations d’authentification et le mode d’accès aux données.
Gestion des métadonnées dans l’intégration de données

### Authentication Method
- **Simple Authentication**

### Authentication Parameters
- **Bind DN or User**: `cn=Directory Manager`
- **Bind password**: `*******`
- **Save password**

### Base DN
- **Get base DNs from Root DSE**
- **Base DN**: `o=directoryRoot`

### Aliases Dereferencing
- **Finding**
- **Searching**
- **Never**
- **Always**

### Referrals Handling
- **Ignore**
- **Follow**

### Limits
- **Count Limit**: 100
- **Time Limit**: 0

### Description

#### Authentication method
Pour une **Simple authentication**, renseignez le champ **Authentication Parameters**. **Anonymous authentication** ne nécessite aucun paramètre d’authentification. **Anonymous authentication** ne nécessite aucun paramètre d’authentification.

#### Authentication Parameters
- **Bind DN or User**: connectez-vous en fonction de la méthode d’authentification LDAP choisie.
- **Bind password**: renseignez le mot de passe.
- **Save password**: cochez cette case pour enregistrer les informations d’authentification.

#### Get Base DN from Root DSE / Base DN
Chemin d’accés à l’arborescence de l’utilisateur. Le bouton **Fetch Base DNs** redirige le DN automatiquement à la racine.
Gestion des métadonnées dans l'intégration de données

<table>
<thead>
<tr>
<th>Champ</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limit</td>
<td>Limite le nombre d’enregistrements lus</td>
</tr>
</tbody>
</table>

2. Cliquez sur **Check authentication** pour vérifier vos droits d’accès.
3. Cliquez sur **Fetch Base DNs** pour conserver le DN et cliquez sur **Next** pour continuer.
4. Si des bibliothèques tierces requises pour la configuration de la connexion LDAP sont manquantes, un assistant d’installation de modules externes s’affiche. Installez les bibliothèques requises en suivant les étapes de l’assistant.

**Définir le schéma de votre annuaire LDAP**

**Procédure**

1. Sélectionnez les attributs que vous souhaitez inclure dans la structure du schéma. Ajoutez un filtre si vous souhaitez uniquement des données particulières.
2. Cliquez sur **Refresh Preview** pour afficher les colonnes sélectionnées ainsi qu’un échantillon des données.

3. Cliquez sur **Next** pour passer à l’étape suivante.

**Finaliser le schéma de votre annuaire LDAP**

**Pourquoi et quand exécuter cette tâche**

Cette dernière étape affiche le schéma LDAP généré et vous permet de personnaliser le schéma final.
Procédure

1. Si nécessaire, renommez la métadonnée dans le champ **Name**, (metadata, par défaut), ajoutez un commentaire dans le champ **Comment** et procédez à d'autres modifications, par exemple :
   - Définissez les colonnes en éditant les champs correspondants.
   - Ajoutez ou supprimez une colonne à l'aide des boutons et .
   - Modifiez l'ordre des colonnes à l'aide des boutons et .

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez **Java API Specification** (en anglais).

Les types de données **Talend** les plus utilisés sont les suivants :

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant **tFileInputRaw** en spécifiant qu’il comporte un type de données **Object**.
• List : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
• Document : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

2. Si l’annuaire LDAP ayant donné lieu au schéma a changé, cliquez sur le bouton **Guess** pour générer le schéma une nouvelle fois. Notez que les modifications que vous avez apportées au schéma seront perdues si vous cliquez sur le bouton **Guess**.

3. Cliquez sur **Finish**. Le nouveau schéma s’affichera dans le **Repository**, sous le nœud LDAP correspondant.

**Centraliser les métadonnées Azure Storage**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez utiliser l’assistant de création de métadonnées Azure Storage fourni par le Studio Talend pour configurer rapidement une connexion à Azure Storage et récupérer le schéma des conteneurs, des files et des tables.

**Procédure**

1. Dans le **Repository**, développez le nœud **Metadata**, puis cliquez-droit sur le nœud **Azure Storage** et, dans le menu contextuel, sélectionnez **Create Azure Storage** pour ouvrir l’assistant **Azure Storage**.

2. Dans la boîte de dialogue **Azure Storage Connection Settings**, renseignez (ou mettez à jour, si nécessaire) les informations des champs listés dans le tableau suivant.
Gestion des métadonnées dans l'intégration de données

### Propriété Description

**Name**
Saisissez le nom de la connexion à créer.

**Account Name**

**Account key**
Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n’importe laquelle peut être utilisée pour accéder au compte.

**Protocol**
 Sélectionnez le protocole de la connexion à créer.

**Use Azure Shared Access Signature**
Cochez cette case pour utiliser une signature d’accès partagé, afin d’accéder aux ressources de stockage sans avoir besoin d’une clé associée au compte. Dans le champ Azure Shared Access Signature affiché, saisissez votre signature d’accès partagé, entre guillemets doubles. Pour plus d’informations, consultez Utilisation des signatures d’accès partagé (SAP).

3. **Cliquez sur Test connection** pour vérifier la configuration.
Lorsque les informations de connexion fournies sont correctes, un message indiquant le succès de la vérification de la connexion s’affiche. Cliquez sur OK pour fermer la boîte de dialogue. Le bouton Next n’est plus grisé.

4. Cliquez sur **Next**, puis dans la boîte de dialogue Add a new container schema in current connection qui s’affiche, cochez les cases correspondant au(x) conteneur(s) dont vous souhaitez récupérer le schéma.
5. Cliquez sur Next, puis dans la boîte de dialogue Add a new queue schema in current connection qui s’affiche, cochez les cases correspondant à la ou aux file(s) dont vous souhaitez récupérer le schéma.

6. Cliquez sur Next, puis dans la boîte de dialogue Add a new table schema in current connection qui s’affiche, cochez les cases correspondant à la ou aux table(s) dont vous souhaitez récupérer le schéma.
7. Cliquez sur **Finish** pour finaliser la procédure.
La nouvelle connexion à Azure Storage créée s’affiche sous le nœud **Azure Storage** du **Repository**, de même que les schémas des conteneurs, files et tables sélectionné(e)s.
Vous pouvez maintenant ajouter un composant Azure Storage dans l’espace de modélisation graphique, en glissant-déposant la connexion à Azure Storage créée ou n’importe quel(le) conteneur/file/table récupéré(e) depuis le Repository, afin de réutiliser la connexion et/ou les informations du schéma. Pour plus d’informations concernant le glisser-déposer de métadonnées dans l’espace de modélisation graphique, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438. Pour plus d’informations concernant les composants Azure Storage, consultez la documentation relative aux composants Azure Storage.

Pour modifier la métadonnée de connexion à Azure Storage créée, cliquez-droit sur le nœud de la connexion dans le Repository, puis dans le menu contextuel, sélectionnez Edit Azure Storage pour ouvrir l’assistant de configuration de la métadonnée.

Pour modifier le schéma d’un(e) conteneur/file/table, cliquez-droit sur le nœud du conteneur ou de la file/table dans le Repository, puis dans le menu contextuel, sélectionnez Edit Schema pour ouvrir l’assistant de mise à jour du schéma.

### Centraliser les métadonnées Google Drive

Le Studio Talend vous permet de centraliser les détails de votre connexion à Google Drive, dans le dossier Metadata de la vue Repository. Vous pouvez ensuite utiliser cette connexion établie pour vous connecter à Google Drive, lorsque vous utilisez des composants Google Drive.

#### Procédure


2. Spécifiez les valeurs pour les propriétés listées dans le tableau suivant, selon la méthode OAuth que vous utilisez.

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Nom de la connexion à Google Drive à créer.</td>
</tr>
<tr>
<td>Application Name</td>
<td>Nom de l’application requis par Google Drive pour obtenir l’accès à ses API.</td>
</tr>
<tr>
<td>OAuth Method</td>
<td>Sélectionnez une méthode OAuth utilisée pour accéder à Google Drive dans la liste déroulante.</td>
</tr>
<tr>
<td></td>
<td>• Access Token (deprecated) : utilise un jeton d’accès pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• Installed Application (Id &amp; Secret) : utilise l’ID et le secret du client créés via la console Google API pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Installed applications (en anglais).</td>
</tr>
<tr>
<td></td>
<td>• Installed Application (JSON) : utilise un fichier JSON de secret, créé via la console Google API contenant l’ID du client, son secret et d’autres paramètres OAuth 2.0 pour accéder à Google Drive.</td>
</tr>
<tr>
<td></td>
<td>• Service Account : utilise un fichier JSON de compte de service, créé via la console Google API, pour accéder à Google Drive. Pour plus d’informations concernant cette méthode, consultez Google Identity Platform &gt; Service accounts (en anglais).</td>
</tr>
<tr>
<td>Propriété</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Access Token</td>
<td>Jeton d'accès généré via Google Developers OAuth 2.0 Playground. Cette propriété est disponible uniquement lorsque l'option Access Token est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Client ID et Client secret</td>
<td>ID et secret du client. Ces deux propriétés sont disponibles uniquement lorsque l'option Installed Application (Id &amp; Secret) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Client Secret JSON</td>
<td>Chemin d'accès au fichier JSON contenant le secret du client. Cette propriété est disponible uniquement lorsque l'option Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>Service Account JSON</td>
<td>Chemin d'accès au fichier JSON du compte de service. Cette propriété est disponible uniquement lorsque l'option Service Account est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
<tr>
<td>DataStore Path</td>
<td>Chemin d'accès au fichier Credential contenant le jeton de rafraîchissement. Cette propriété est disponible uniquement lorsque l'option Installed Application (Id &amp; Secret) ou Installed Application (JSON) est sélectionnée dans la liste déroulante OAuth Method.</td>
</tr>
</tbody>
</table>
| Use Proxy                     | Cochez cette case, lorsque vous travaillez derrière un proxy. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :  
|                               | • Host : adresse IP du serveur du proxy HTTP.  
|                               | • Port : numéro du port du serveur du proxy HTTP. |
| Use SSL                       | Cochez cette case si une connexion SSL est utilisée pour accéder à Google Drive. Lorsque cette case est cochée, vous devez spécifier la valeur des paramètres suivants :  
|                               | • Algorithm : nom de l'algorithme de chiffrement SSL.  
|                               | • Keystore File : chemin d'accès au fichier TrustStore du certificat contenant la liste des certificat auxquels le client fait confiance.  
|                               | • Password : mot de passe utilisé pour vérifier l'intégrité des données TrustStore. |

3. Cliquez sur **Test connection** pour vérifier la configuration.

Si vous utilisez la méthode OAuth Access Token (deprecated), Installed Application (Id & Secret), ou Installed Application (JSON), une fenêtre s'ouvre dans votre navigateur Web, vous demandant de choisir votre compte et d'autoriser l'accès.
à Google Drive. Une fois l’authentification réussie dans le navigateur Web, une boîte de dialogue indiquant que la connexion est établie s’ouvre dans le Studio Talend.

4. **Cliquez sur OK** pour fermer la boîte de dialogue, puis cliquez sur **Finish**.

La nouvelle connexion à Google Drive s’affiche sous le nœud **Google Drive**, dans la vue **Repository**.

Vous pouvez ajouter un composant Google Drive dans l’espace de modélisation graphique en glissant-déposant la métadonnée de connexion à Google Drive, pour réutiliser les informations de connexion. Pour plus d’informations concernant le glisser-déposer de métadonnées dans l’espace de modélisation graphique, consultez **Comment utiliser les métadonnées centralisées dans un Job** à la page 438. Pour plus d’informations concernant l’utilisation des composants Google Drive, consultez la documentation des composants Google Drive.

Pour modifier la métadonnée de connexion à Google Drive créée, cliquez-droit sur le nœud de connexion dans la vue **Repository** et sélectionnez **Edit GoogleDrive Connection** dans le menu contextuel, pour ouvrir l’assistant de configuration de la métadonnée.

### Centraliser les métadonnées Marketo

**Pourquoi et quand exécuter cette tâche**

Vous pouvez utiliser l’assistant de création de métadonnées Marketo fourni par le Studio Talend pour configurer rapidement une connexion à Marketo et récupérer le schéma des objets personnalisés via l’API REST.

**Procédure**

1. Dans le **Repository**, développez le nœud **Metadata**, puis cliquez-droit sur le nœud **Marketo** dans l’arborescence. Dans le menu contextuel, sélectionnez **Create Marketo** pour ouvrir l’assistant **Marketo**.
2. Dans la boîte de dialogue *Marketo REST Connection Settings*, renseignez (ou mettez à jour, si nécessaire) les informations des champs listés dans le tableau suivant.
### Gestion des métadonnées dans l’intégration de données

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection name</strong></td>
<td>Saisissez le nom de la connexion à créer.</td>
</tr>
<tr>
<td><strong>Client Access ID</strong></td>
<td>Saisissez l’identifiant utilisateur permettant d’accéder au service Web Marketo.</td>
</tr>
<tr>
<td><strong>Secret key</strong></td>
<td>Saisissez la phrase secrète du client pour accéder au Service Web Marketo.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Saisissez le délai avant suspension (en millisecondes) de la connexion au Service Web Marketo, avant de terminer la tentative.</td>
</tr>
<tr>
<td><strong>Max reconnection attempts</strong></td>
<td>Spécifiez le nombre maximal de tentatives de connexion au service Web Marketo, avant d’abandonner.</td>
</tr>
<tr>
<td><strong>Attempt interval time</strong></td>
<td>Saisissez la période de temps (en millisecondes) entre deux tentatives de reconnexion subséquentes.</td>
</tr>
</tbody>
</table>

3. Cliquez sur **Test connection** pour vérifier la configuration.
   Lorsque les informations de connexion fournies sont correctes, un message indiquant le succès de la vérification de la connexion s’affiche. Cliquez sur **OK** pour fermer la boîte de dialogue. Le bouton **Next** n’est plus grisé.

4. Cliquez sur **Next** pour passer à l’étape suivante afin de sélectionner les objets personnalisés souhaités.
5. Cochez les cases correspondant aux objets personnalisés dont vous souhaitez récupérer le schéma, puis cliquez sur Finish.

La nouvelle connexion à Marketo créée s’affiche sous le nœud Marketo du Repository, de même que les schémas des objets personnalisés sélectionnés.
Vous pouvez maintenant ajouter un composant Marketo dans l'espace de modélisation graphique, en glissant-déposant la connexion à Marketo créée ou n'importe quelle table récupérée depuis le Repository, afin de réutiliser la connexion et/ou les informations du schéma. Pour plus d'informations concernant le glisser-déposer de métadonnées dans l'espace de modélisation graphique, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438. Pour plus d'informations concernant les composants Marketo, consultez la documentation relative aux composants Marketo.

Pour modifier la métadonnée de connexion à Marketo créée, cliquez-droit sur le nœud de la connexion dans le Repository, puis dans le menu contextuel, sélectionnez Edit Marketo pour ouvrir l'assistant de configuration de la métadonnée.

Pour modifier le schéma d'une table, cliquez-droit sur le nœud de la table dans le Repository, puis dans le menu contextuel, sélectionnez Edit Schema pour ouvrir l’assistant de mise à jour du schéma.

Centraliser les métadonnées d'une connexion Salesforce

Vous pouvez utiliser l’assistant de création de métadonnées Salesforce fourni par le Studio Talend pour configurer rapidement une connexion à un système Salesforce et ainsi réutiliser vos métadonnées Salesforce dans plusieurs Jobs.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l’assistant de création de métadonnées Salesforce fourni par le Studio Talend pour configurer rapidement une connexion à un système Salesforce et ainsi réutiliser vos métadonnées Salesforce dans plusieurs Jobs.

Procédure

1. Dans le Repository, développez le nœud Metadata, cliquez-droit sur le nœud Salesforce dans l’arborescence, puis sélectionnez Create Salesforce dans le menu contextuel pour ouvrir l’assistant Salesforce.
2. Saisissez le nom de votre connexion dans le champ **Name**, sélectionnez **Basic** ou **OAuth** dans la liste **Connection type**, puis renseignez les informations de la connexion selon le type de connexion que vous avez sélectionné.

![Image de Salesforce Connection Setting](image)

- Si l’option **Basic** est sélectionnée, vous devez spécifier les informations suivantes :
  - **User Id** : identifiant de l’utilisateur de Salesforce.
  - **Password** : mot de passe associé à l’identifiant de l’utilisateur.
  - **Security Key** : jeton de sécurité.
- Si l’option **OAuth** est sélectionnée, vous devez spécifier les informations suivantes :
  - **Client Id** et **Client Secret** : Consumer key/Consumer Secret OAuth, disponibles dans la zone **OAuth Settings** de l’application connectée que vous avez créée sur Salesforce.com.
  - **Callback Host** et **Callback Port** : Cette URL (l’hôte et le port) est définie durant la création d’une application connectée et sera affichée dans la zone **OAuth Settings** de l’application connectée.
  - **Token File** : chemin d’accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d’accès sans autorisation.

3. Si nécessaire, cliquez sur **Advanced...** pour ouvrir la boîte de dialogue [Salesforce Advanced Connection Settings], effectuez les tâches suivantes puis cliquez sur **OK** :

- saisissez l’URL du Service Web de Salesforce requise pour se connecter au système Salesforce.
- cochez la case **Bulk Connection** si vous devez utiliser la fonction de traitement des données de masse.
• cochez la case **Use or save the connection session** et, dans le champ **Session directory** qui s’affiche, spécifiez le chemin d’accès au fichier de sessions de la connexion utilisé ou à sauvegarder.

Ce fichier de sessions peut être partagé par différents Jobs afin de récupérer une session de connexion tant que l’ID correct de l’utilisateur est fourni par le composant. Ainsi, vous n’avez pas besoin de vous connecter au serveur pour récupérer la session.

Lorsqu’une session expirée est détectée, si les informations des connexion correctes (ID utilisateur, mot de passe, clé de sécurité) sont fournies, le composant se connecte au serveur afin de récupérer les informations de la nouvelle session et mettre à jour le fichier de sessions de connexion.

Cette case est disponible uniquement lorsque l’option **Basic** est sélectionnée dans la liste déroulante **Connection type**.

• cochez la case **Need compression** pour activer la compression des messages SOAP, ce qui peut augmenter les niveaux de performance.

• cochez la case **Trace HTTP message** pour écrire en sortie les interactions HTTP dans la console.

Cette option est disponible si la case **Bulk Connection** est cochée.

• cochez la case **Use HTTP Chunked** pour utiliser le mécanisme de transfert des données HTTP morcelées.

Cette option est indisponible lorsque la case **Bulk Connection** est cochée.

• saisissez l’identifiant d’un utilisateur réel dans le champ **Client Id** afin de différencier les utilisateurs d’un même compte et mot de passe pour accéder au site Web de Salesforce.

• Renseignez le champ **Timeout** avec la valeur de délai d’attente de connexion à Salesforce, en millisecondes.

• Si nécessaire, cochez la case **Use Proxy** pour configurer le type de proxy SOCKS et saisir les informations de configuration correspondantes. Notez que vous pouvez également configurer le type de proxy HTTP via **Window > Preferences > General > Network Connections**.
4. Cliquez sur **Test connection** pour vérifier les paramètres de connexion et lorsque le message indiquant le succès de la vérification de la connexion s’affiche, cliquez sur **OK** pour confirmer. Cliquez ensuite sur **Next** pour passer à l’étape suivante afin de sélectionner les modules dont vous souhaitez récupérer le schéma.

5. Cochez les cases correspondant aux modules souhaités puis cliquez sur **Finish** pour récupérer les schémas des modules sélectionnés.

Vous pouvez saisir du texte en filtre pour réduire votre sélection.

La nouvelle connexion à Salesforce créée s’affiche sous le nœud **Salesforce** du **Repository**, de même que les schémas des modules sélectionnés.
Résultats

Vous pouvez à présent glisser-déposer la connexion à Salesforce ou n’importe lequel de ses schémas depuis le Repository dans l’espace de modélisation graphique. Dans la boîte de dialogue qui s’ouvre, vous pouvez choisir un composant Salesforce à utiliser dans votre Job. Dans la boîte de dialogue qui s’ouvre, vous pouvez choisir un composant Salesforce à utiliser dans votre Job. Pour plus d’informations concernant le glisser-déposer de métadonnées dans l’espace de modélisation graphique, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438.

Pour modifier la métadonnée Salesforce, cliquez-droit sur le schéma dans le Repository, puis sélectionnez Edit Salesforce pour ouvrir l’assistant de configuration de la métadonnée du fichier.

Pour modifier un schéma Salesforce existant, cliquez-droit sur le schéma dans le Repository et sélectionnez Edit Schema dans le menu contextuel.

Centraliser les métadonnées Snowflake

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l’assistant de création de métadonnées Snowflake fourni par le Studio Talend pour configurer rapidement une connexion à Snowflake et récupérer le schéma des tables souhaitées.

Remarque: L’assistant de métadonnées Snowflake ne supporte pas la gestion des vues Snowflake, pour le moment.
Procédure


2. Dans la boîte de dialogue Snowflake Connection Settings, renseignez les informations des champs listés dans le tableau suivant.
3. Cliquez sur Advanced..., puis dans la boîte de dialogue **Snowflake Advanced Connection Settings** qui s’affiche, renseignez ou mettez à jour les valeurs des propriétés avancées listées dans le tableau suivant. Cliquez sur OK pour fermer la boîte de dialogue.
Gestion des métadonnées dans l’intégration de données

<table>
<thead>
<tr>
<th>Propriété</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Timeout</td>
<td>Spécifiez le temps d’attente d’une réponse lors de la connexion à Snowflake, avant de retourner une erreur.</td>
</tr>
<tr>
<td>Tracing</td>
<td>Sélectionnez le niveau de log pour le pilote JDBC de Snowflake. Si cette option est activée, un log standard Java est généré.</td>
</tr>
<tr>
<td>Role</td>
<td>Saisissez le rôle de contrôle des accès par défaut à utiliser pour initialiser la session Snowflake. Ce rôle doit déjà exister et doit avoir été assigné à l’ID de l’utilisateur que vous utilisez pour vous connecter Snowflake. Si vous laissez ce champ vide, le rôle PUBLIC est automatiquement assigné. Pour plus d’informations concernant le modèle de contrôle des accès de Snowflake, consultez la documentation Snowflake à l’adresse suivante Understanding the Access Control Model (en anglais).</td>
</tr>
</tbody>
</table>

4. Cliquez sur **Test connection** pour vérifier la configuration.
   Lorsque les informations de connexion fournies sont correctes, un message indiquant le succès de la vérification de la connexion s’affiche. Cliquez sur **OK** pour fermer la boîte de dialogue. Le bouton **Next** n’est plus grisé.

5. Cliquez sur **Next** pour passer à l’étape suivante afin de sélectionner les tables souhaitées.
6. Cochez les cases correspondant aux tables dont vous souhaitez récupérer le schéma, puis cliquez sur **Finish**.

La nouvelle connexion à Snowflake créée s'affiche sous le nœud **Snowflake** du **Repository**, de même que les schémas des tables sélectionnées.
Vous pouvez maintenant ajouter un composant Snowflake dans l'espace de modélisation graphique, en glissant-déposant la connexion à Snowflake créée ou n'importe quelle table récupérée depuis le Repository, afin de réutiliser la connexion et/ou les informations du schéma. Pour plus d'informations concernant le glisser-déposer de métadonnées dans l'espace de modélisation graphique, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438. Pour plus d'informations concernant les composants Snowflake, consultez la documentation relative aux composants Snowflake.

Pour modifier la métadonnée de connexion à Snowflake créée, cliquez-droit sur le nœud de la connexion dans le Repository, puis dans le menu contextuel, sélectionnez Edit Snowflake pour ouvrir l'assistant de configuration de la métadonnée.

Pour modifier le schéma d'une table, cliquez-droit sur le nœud de la table dans le Repository, puis dans le menu contextuel, sélectionnez Edit Schema pour ouvrir l'assistant de mise à jour du schéma.

**Configurer un schéma générique**

Le Studio Talend vous permet de créer un schéma afin de l'utiliser dans vos Jobs, si aucun des assistants de métadonnées ne correspond à vos besoins ou si vous n'avez pas de fichier source duquel prendre le schéma.

Vous pouvez créer un schéma générique :

- à partir de rien. Pour plus de détails, consultez Configurer un schéma générique à partir de rien à la page 385,
- à partir d'un fichier XML de définition de schéma. Pour plus de détails, consultez Configurer un schéma générique à partir d'un fichier XML à la page 388,
- et à partir du schéma défini dans un composant. Pour plus de détails, consultez Enregistrer le schéma d'un composant en tant que schéma générique à la page 390.

Pour utiliser le schéma générique sur un composant, utilisez l'une des méthodes suivantes :

- Sélectionnez Repository dans la liste Schema de la vue Basic settings du composant.
  Cliquez sur le bouton [...] pour ouvrir la boîte de dialogue [Repository Content] et sélectionnez le schéma correspondant sous le nœud Generic schemas. Cliquez sur OK.
- Sélectionnez le nœud metadata du schéma générique depuis le Repository et déposez-le sur un composant.

**Configurer un schéma générique à partir de rien**

**Pourquoi et quand exécuter cette tâche**

Pour créer un schéma générique à partir de rien, procédez comme suit :

**Procédure**

1. Sous le nœud Metadata, cliquez-droit sur Generic schemas dans la vue Repository et sélectionnez Create generic schema.
2. Dans l’assistant de création de schéma qui apparaît, renseignez les propriétés génériques du schéma, comme son nom (dans le champ **Name**) et sa **Description**. Le champ **Status** (pour le statut) est un champ personnalisable. Pour plus d’informations concernant ce champ, consultez **Définir les paramètres des Statuts** à la page 487.

   Cliquez sur **Next** pour passer à l’étape suivante.

3. Nommez votre schéma ou utilisez le nom par défaut (metadata) et ajoutez un commentaire, si nécessaire. Personnalisez la structure du schéma dans le panneau **Schema** selon vos besoins.

   La barre d’outils vous permet d’ajouter, supprimer ou déplacer des colonnes dans votre schéma.
Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez **Java API Specification** (en anglais).

Les types de données **Talend** les plus utilisés sont les suivants :

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant **tFileInputRaw** en spécifiant qu’il comporte un type de données **Object**.

- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.

- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

4. Cliquez sur **Finish** pour terminer la création du schéma générique. Le schéma créé s’affiche sous le nœud **Generic schemas**.
Configurer un schéma générique à partir d'un fichier XML

Pourquoi et quand exécuter cette tâche

**Avertissement:**
Le fichier XML source à partir duquel vous pouvez créer un schéma générique doit être un schéma exporté depuis le Studio ou depuis un fichier XML ayant la même structure.

Pour créer un schéma générique à partir d'un fichier source XML, procédez comme suit :

**Procédure**

1. Cliquez-droit sur *Generic schemas* dans la vue *Repository* et sélectionnez *Create generic schema from xml*.

2. Dans la boîte de dialogue qui apparaît, sélectionnez le fichier source XML duquel prendre le schéma et cliquez sur *Open*.

3. Dans l'assistant de création de schéma qui apparaît, renseignez le champ *Name* en saisissant un nom pour le schéma ou utilisez celui par défaut (*metadata*) et saisissez un commentaire, dans le champ *Comment*, si nécessaire.

   La structure du schéma du fichier source s'affiche dans le panneau *Schema*. Vous pouvez personnaliser les colonnes du schéma selon vos besoins.

   La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.
Assurez-vous que le type de données dans la colonne Type est correctement défini.

Pour plus d’informations concernant les types de données Java, à savoir le modèle de date, consultez Java API Specification (en anglais).

Les types de données Talend les plus utilisés sont les suivants :

- **Object** : est un type de données Talend générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l’aide d’un composant tFileInputRaw en spécifiant qu’il comporte un type de données Object.
- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.

Enregistrer le schéma d'un composant en tant que schéma générique

Pourquoi et quand exécuter cette tâche

Vous pouvez créer un schéma générique en enregistrant le schéma défini pour un composant. Pour ce faire, procédez comme suit :

Procédure


2. Cliquez sur l’icône représentant une disquette afin d’ouvrir la boîte de dialogue Select folder.

3. Si nécessaire, sélectionnez un dossier puis cliquez sur OK afin de fermer la boîte dialogue et ouvrir l’assistant de création Save as generic schema.
4. Renseignez le champ **Name** (obligatoire) ainsi que les autres champs (si nécessaire). Cliquez sur **Finish** pour sauvegarder le schéma. Fermez ensuite la boîte de dialogue **Schema** ouverte depuis la vue **Basic settings** du composant.
Le schéma est sauvegardé dans le dossier sélectionné sous le nœud **Generic schemas** dans le **Repository**.

**Centraliser les métadonnées MDM**

Le Studio Talend vous permet de centraliser les informations d’une ou plusieurs connexion(s) MDM sous le nœud **Metadata** de la vue **Repository**. Vous pouvez utiliser l’une de ces connexions établies pour vous connecter au serveur MDM.

**Remarque:**
Vous pouvez également paramétrer une connexion MDM en cliquant sur l’icône de la vue **Basic settings** des composants **tMDMInput** et **tMDMOutput**.

Selon l’option que vous sélectionnez, l’assistant vous permet de créer soit un schéma XML Input (d’entrée), soit un schéma XML Output (de sortie), soit un schéma XML Receive (de réception). Plus tard, dans un Job **Talend**, le composant **tMDMInput** utilise le schéma d’entrée défini pour lire des données maître stockées dans des documents XML. Le **tMDMOutput** utilise le schéma de sortie défini, soit pour écrire des données maître dans un document XML sur le serveur MDM, soit pour mettre à
jour des documents XML existants. Le **tMDMReceive** utilise le schéma XML défini pour recevoir, des déclencheurs et des processus MDM, un enregistrement MDM au format XML.

**Définir la connexion**

**Pourquoi et quand exécuter cette tâche**

Pour établir une connexion MDM, procédez comme suit :

**Procédure**

1. Dans la vue **Repository**, développez le nœud **Metadata** et cliquez-droit sur **Talend MDM**.
2. Sélectionnez **Create MDM Connection** dans le menu contextuel.
   L'assistant de connexion s'affiche.

![Assistant de connexion MDM](image)

3. Renseignez les champs **Name**, **Purpose** et **Description**. Le champ **Status** est un champ personnalisé pouvant être défini. Pour plus d'informations, consultez Définir les paramètres des Statuts à la page 487.
4. Cliquez sur **Next** pour passer à l'étape suivante.

**Remarque:**
La valeur par défaut dans le champ **Server URL** varie selon l'option sélectionnée dans la liste **Version**.

6. Remplissez les informations de connexion au serveur MDM, y compris les informations d'authentification, puis cliquez sur **Check** afin de vérifier la connexion que vous avez créée.
Une boîte de dialogue s'ouvre pour confirmer que votre connexion a été créée avec succès. Cliquez sur **OK** pour fermer la boîte de dialogue.

Si nécessaire, vous pouvez cliquer sur **Export as context** pour exporter les paramètres de connexion **Talend MDM** dans un nouveau groupe de contextes dans le Repository ou réutiliser des variables d'un groupe de contextes pour paramétrer la connexion de la métadonnée. Pour plus d'informations, consultez **Exporter une métadonnée en tant que contexte et réutiliser ses paramètres de contexte pour configurer une connexion** à la page 428.

7. Cliquez sur **Next** pour passer à l'étape suivante.
Gestion des métadonnées dans l’intégration de données

8. Dans la liste **Data-Model**, sélectionnez le modèle de données par rapport auquel vous souhaitez valider les données maître.

9. Dans la liste **Data-Container**, sélectionnez le conteneur de données où sont stockées les données maître auxquelles vous souhaitez accéder.

10. Cliquez sur **Finish** afin de valider vos changements et fermer la boîte de dialogue.

   La connexion nouvellement créée s’affiche sous le nœud **Talend MDM** dans le dossier **Metadata** dans la vue **Repository**.

**Résultats**

Vous devez maintenant récupérer le schéma XML des entités métier liées à cette connexion MDM.

**Définir un schéma MDM**

**Définir un schéma d’entrée MDM**

Cette section décrit la définition et le chargement d’un schéma MDM XML d’entrée. Pour définir et charger un schéma MDM XML de sortie, consultez **Définir un schéma de sortie MDM** à la page 400.

**Récupérer les valeurs des entités pour une connexion MDM**

**Pourquoi et quand exécuter cette tâche**

Pour définir les valeurs à retrouver à partir d’une ou plusieurs entité(s) liée(s) à une connexion MDM procédez comme suit:
Procédure

1. Dans la vue **Repository**, développez le nœud **Metadata** puis cliquez-droit sur la connexion MDM pour laquelle vous souhaitez récupérer les valeurs de l'entité, et sélectionnez **Retrieve Entity** dans le menu contextuel.

Exemple

2. Dans la boîte de dialogue **MDM Model**, sélectionnez l'option **Input MDM** afin de charger un schéma XML d'entrée, puis cliquez sur **Next** afin de passer à l'étape suivante.
3. Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) à partir de laquelle vous souhaitez récupérer les valeurs.
Le nom s'affiche automatiquement dans le champ **Name**.
Exemple

Remarque: Vous êtes libre de saisir le nom que vous voulez dans ce champ.

4. Cliquez sur **Next** pour passer à l’étape suivante.

Le schéma de l’entité que vous avez sélectionnée, s’affiche dans le panneau *Source Schema*. Dans la boîte de dialogue ouverte, vous pouvez configurer les paramètres à prendre en compte lors de la définition du schéma XML.

Exemple

![Image](image.png)

La boîte de dialogue relative au schéma est divisée en quatre panneaux :

<table>
<thead>
<tr>
<th>Panneau</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Schema</td>
<td>Arborescence de l’entité chargée.</td>
</tr>
<tr>
<td>Target Schema</td>
<td>Informations d’extraction et d’itération.</td>
</tr>
<tr>
<td>Preview</td>
<td>Aperçu du schéma cible.</td>
</tr>
<tr>
<td>File viewer</td>
<td>Visionneur des données brutes.</td>
</tr>
</tbody>
</table>

5. Dans la zone **Xpath loop expression**, saisissez l’expression absolue du chemin XPath vers le nœud de la structure XML sur lequel appliquer l’itération.
Vous pouvez également glisser le nœud du schéma source dans le champ Xpath du schéma cible. Le lien s’affiche en orange.

**Remarque:** Le champ **Xpath loop expression** est obligatoire.

6. Au besoin, définissez une limite (**Loop limit**) afin de restreindre l’itération à un certain nombre de nœuds.

**Exemple**

Dans la capture d’écran ci-dessus, utilisez **Features** comme élément sur lequel effectuer une boucle, puisqu’il est répété dans l’entité **Product** :

```xml
<Product>
  <Id>1</Id>
  <Name>Cup</Name>
  <Description/>
  <Features/>
  <Feature>Color red</Feature>
  <Feature>Size maxi</Feature>
  ...
</Product>
<Product>
  <Id>2</Id>
  <Name>Cup</Name>
  <Description/>
  <Features/>
  <Feature>Color blue</Feature>
  <Feature>Thermos</Feature>
  ...
</Product>
```

Ce faisant, le composant **tMDMInput** utilisant la connexion MDM va créer une nouvelle ligne pour chaque élément **Feature** rencontré.

7. Afin de définir les champs à extraire, glissez le nœud correspondant du schéma source dans le champ **Relative or absolute XPath expression**.
Exemple

Conseil: Cliquez sur le bouton [+] afin d’ajouter des lignes à la table, puis sélectionnez les champs à extraire. Appuyez sur la touche Ctrl ou Maj pour sélectionner plusieurs nœuds, groupés ou séparés, et glissez-les dans la table.

8. Au besoin, saisissez un nom pour chaque colonne récupérée, dans le champ Column name.

Conseil: Vous pouvez prioriser l’ordre des champs à extraire, en sélectionnant le champ et en utilisant les flèches montante et descendante. Le lien du champ sélectionné s’affiche en bleu, et tous les autres en gris.

9. Cliquez sur Finish afin de valider vos modifications et fermer la boîte de dialogue.

Résultats

Le schéma nouvellement créé s’affiche sous le nœud Talend MDM correspondant dans la vue Repository.

Modifier le schéma créé

Procédure

1. Dans la vue Repository, développez les nœuds Metadata puis Talend MDM et parcourez votre répertoire jusqu’au schéma que vous souhaitez modifier.

2. Cliquez-droit sur le nom du schéma puis sélectionnez Edit Entity dans le menu contextuel.
Une boîte de dialogue s’ouvre.

3. Modifiez le schéma comme vous le souhaitez.
Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.
Assurez-vous que le type de données dans la colonne **Type** est correctement défini.
Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez *Java API Specification* (en anglais).
Les types de données *Talend* les plus utilisés sont les suivants :
- **Object** : est un type de données *Talend* générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant *tFileInputRaw* en spécifiant qu’il comporte un type de données Object.
- **List** : est une liste d’éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l’aide de l’élément xsd:list.
- **Document** : est un type de données permettant le traitement d’un document XML en entier sans tenir compte de son contenu.
4. Cliquez sur **Finish** afin de fermer la boîte de dialogue.

**Résultats**
La connexion MDM d’entrée (*tMDMInput*) est maintenant prête à être déposée dans l’un de vos Jobs.

**Définir un schéma de sortie MDM**

**Pourquoi et quand exécuter cette tâche**
Cette section décrit la définition et le chargement d’un schéma XML MDM de sortie. Pour définir et charger un schéma XML MDM d’entrée, consultez *Définir la connexion* à la page 392.
Pour définir les valeurs à écrire dans une ou plusieurs entité(s) liée(s) à une connexion MDM spécifique, procédez comme suit :

**Procédure**

1. Dans la vue **Repository**, développez le nœud **Metadata** et cliquez-droit sur la connexion MDM pour laquelle vous souhaitez écrire les valeurs de l’entité, et sélectionnez **Retrieve Entity** dans le menu contextuel.

2. Dans la boîte de dialogue **MDM Model**, sélectionnez l’option **Output MDM** afin de définir un schéma XML de sortie, puis cliquez sur **Next** pour procéder à l’étape suivante.

**Exemple**

3. Dans le champ **Entities**, sélectionnez l’entité métier (schéma XML) dans laquelle vous souhaitez écrire les valeurs.
Exemple

Le nom s’affiche automatiquement dans le champ **Name**.

**Remarque:** Vous êtes libre de saisir le nom que vous voulez dans ce champ.

4. Cliquez sur **Next** pour passer à l’étape suivante.

Un schéma identique à celui de l’entité sélectionnée est automatiquement créé dans le panneau **Linker Target**, et les colonnes sont mappées automatiquement du panneau source au panneau cible. L’assistant définit automatiquement l’élément Id comme étant l’élément répétable. Vous pouvez toujours choisir d’effectuer une boucle sur un autre élément. Dans la boîte de dialogue ouverte, vous pouvez configurer les paramètres à prendre en compte lors de la définition du schéma XML.
5. Cliquez sur **Schema Management** pour ouvrir une boîte de dialogue.
6. Effectuez les modifications nécessaires à la définition du schéma XML que vous souhaitez écrire dans l’entité sélectionnée.

**Avertissement:** Votre schéma dans la zone **Linker Source** doit correspondre au schéma dans la zone **Linker Target**. Vous devez donc définir les éléments dans lesquels vous souhaitez écrire des valeurs.

7. Cliquez sur **OK** pour fermer la boîte de dialogue.
   Le schéma défini s’affiche dans la colonne **Schema list**.
Exemple

Dans le panneau **Linker Target**, cliquez-droit sur l’élément que vous souhaitez définir comme élément répétée et sélectionnez **Set as loop element**. Cela limitera l’itération à un ou plusieurs nœud(s).

Ce faisant, le composant **tMDMOutput** utilisant cette connexion MDM créera une nouvelle ligne pour chaque élément ayant une “feature” différente.
Exemple

Conseil: Vous pouvez prioriser l’ordre des champs à écrire, en sélectionnant le champ et en utilisant les flèches montante et descendante.

9. Cliquez sur Finish afin de valider vos modifications et fermer la boîte de dialogue.

Résultats

Le schéma nouvellement créé s’affiche sous le nœud Talend MDM correspondant dans la vue Repository. Vous pouvez modifier le schéma créé selon vos besoins et déposer la connexion en tant que composant tMDMOutput dans l’un de vos Jobs.

Pour plus d’informations concernant la modification du schéma, consultez Modifier le schéma créé à la page 399.
Gestion des métadonnées dans l’intégration de données

Définir un schéma de réception MDM

Avant de commencer
Cette section décrit la définition d’un schéma MDM XML de réception basé sur une connexion MDM.

Pour définir le schéma XML que vous souhaitez recevoir selon une connexion MDM spécifique, procédez comme suit :

Procédure
1. Dans la vue Repository, développez le nœud Metadata puis cliquez-droit sur la connexion MDM pour laquelle vous souhaitez récupérer les valeurs de l’entité, et sélectionnez Retrieve Entity dans le menu contextuel.
2. Dans la boîte de dialogue MDM Model, sélectionnez l’option Receive MDM afin de définir un schéma XML de réception puis cliquez sur Next pour passer à l’étape suivante.

Exemple

3. Dans le champ Entities, sélectionnez l’entité métier (schéma XML) selon laquelle vous souhaitez recevoir le schéma XML.
Le nom s’affiche automatiquement dans le champ **Name**.

**Exemple**

![Image du logiciel](image)

**Remarque:** Vous pouvez saisir tout texte dans ce champ, même s’il est recommandé de saisir le nom de l’entité selon laquelle vous souhaitez recevoir le schéma XML.

4. Cliquez sur **Next** pour passer à l’étape suivante.

Le schéma de l’entité sélectionnée s’affiche dans le panneau **Source Schema**. Dans la boîte de dialogue ouverte, vous pouvez configurer les paramètres à prendre en compte lors de la définition du schéma XML.
La boîte de dialogue relative au schéma est divisée en quatre panneaux :

<table>
<thead>
<tr>
<th>Panneau</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Schema</td>
<td>Arborescence de l’entité chargée.</td>
</tr>
<tr>
<td>Target Schema</td>
<td>Informations d’extraction et d’itération.</td>
</tr>
<tr>
<td>Preview</td>
<td>Aperçu du schéma cible.</td>
</tr>
<tr>
<td>File viewer</td>
<td>Visionneur des données brutes.</td>
</tr>
</tbody>
</table>

5. Dans la zone **XPath loop expression**, saisissez l’expression absolue du chemin XPath vers le nœud de la structure XML sur lequel appliquer l’itération. Vous pouvez également glisser le nœud du schéma source dans le champ Xpath du schéma cible. Le lien s’affiche en orange.

**Remarque:** Le champ **XPath loop expression** est obligatoire.

6. Au besoin, définissez une limite (**Loop limit**) afin de restreindre l’itération à un certain nombre de nœuds.
Exemple

Dans la capture d'écran ci-dessus, utilisez Features comme élément sur lequel effectuer une boucle, puisqu'il est répété dans l'entité Product :

```
<Product>
  <Id>1</Id>
  <Name>Cup</Name>
  <Description/>
  <Features>
    <Feature>Color red</Feature>
    <Feature>Size maxi</Feature>
    ...
  </Features>
</Product>
<Product>
  <Id>2</Id>
  <Name>Cup</Name>
  <Description/>
  <Features>
    <Feature>Color blue</Feature>
    <Feature>Thermos</Feature>
    ...
  </Features>
</Product>
```

Ce faisant, le composant tMDMReceive utilisant la connexion MDM va créer une nouvelle ligne pour chaque élément Feature rencontré.

7. Afin de définir les champs à extraire, glissez le nœud correspondant du schéma source dans le champ Relative or absolute XPath expression.
8. Au besoin, saisissez un nom pour chaque colonne reçue, dans le champ **Column name**.

Conseil: Vous pouvez prioriser l'ordre des champs à recevoir, en sélectionnant le champ et en utilisant les flèches montante et descendante. Le lien du champ sélectionné s'affiche en bleu, et tous les autres en gris.

9. Cliquez sur **Finish** afin de valider vos modifications et fermer la boîte de dialogue.

**Résultats**

Le schéma nouvellement créé s’affiche sous le nœud **Talend MDM** correspondant dans la vue Repository. Vous pouvez modifier le schéma créé selon vos besoins et déposer la connexion en tant que composant **tMDMReceive** dans l’un de vos Jobs.

Pour plus d’informations concernant la modification du schéma, consultez **Modifier le schéma créé** à la page 399.

---

**Centraliser les métadonnées Web Service**

Si vous accédez souvent à un service Web depuis le Studio Talend, vous pouvez stocker vos connexions Web Service dans le **Repository**.
L’assistant **Web Service** vous permet de créer un schéma simple (**Simple WSDL**), ou un schéma avancé (**Advanced WebService**), selon vos besoins.

**Remarque:**
Dans l’étape 1, vous devez saisir les métadonnées du schéma avant de choisir de créer un schéma simple ou un schéma avancé au cours de l’étape 2. Veillez donc à différencier vos schémas simples de vos schémas avancés, en saisissant des noms que vous pourrez reconnaître.

Si vous souhaitez créer un schéma simple, consultez *Configurer un schéma simple* à la page 411.
Si vous souhaitez créer un schéma avancé, consultez *Configurer un schéma avancé* à la page 416.

**Configurer un schéma simple**

Cette section décrit la définition d’un schéma Web Service simple (**Simple WSDL**).

**Définir les propriétés générales du schéma Web Service simple**

**Procédure**

1. Dans la vue **Repository**, développez le nœud **Metadata**.
2. Cliquez-droit sur **Web Service**, et sélectionnez **Create WDSL schema** dans le menu contextuel.

3. Remplissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.
4. Cliquez sur **Next** pour passer à l’étape suivante et sélectionner le type de schéma.

** Sélectionner le type de schéma (Simple)**

**Pourquoi et quand exécuter cette tâche**
Dans cette étape, vous allez définir le type de schéma, Simple ou Advanced. Dans cet exemple, sélectionnez le type Simple.

**Procédure**
1. Dans la boîte de dialogue, sélectionnez **Simple WSDL**.
2. Cliquez sur **Next** pour passer à l’étape suivante.

**Définir l’URI et la méthode**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, définissez l’URI et les paramètres nécessaires pour obtenir les valeurs.
Dans la zone **Web Service Parameter** :

**Procédure**

2. Si nécessaire, cochez la case **Need authentication?** puis saisissez respectivement votre identifiant et votre mot de passe, dans les champs **User** et **Password**.
3. Si vous utilisez un proxy http, cochez la case **Use http proxy**, et renseignez les propriétés dans les champs **host**, **Port**, **user** et **password**.
5. Dans le tableau **Value**, ajoutez ou supprimaitez autant de valeurs que vous souhaitez, en cliquant sur le bouton **Add** (Ajouter) ou **Remove** (Supprimer).
6. Cliquez sur **Refresh Preview** pour vérifier que les paramètres sont corrects.
L’aperçu **Preview** affiche les valeurs qui doivent être transmises par la méthode du Service Web basée sur les paramètres saisis.

**Finaliser le schéma Web Service simple**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez modifier le nom du schéma (par défaut, `metadata`) et modifier le schéma à l’aide de la barre d’outils.
Gestion des métadonnées dans l'intégration de données

Procédure

1. Ajoutez ou supprimez une colonne à l'aide des boutons et .
2. Modifiez l'ordre des colonnes à l'aide des boutons et .

Le nouveau schéma s’affiche dans le Repository sous le nœud Web Service approprié. Vous pouvez à présent le déposer dans l’espace de modélisation graphique en tant que composant tWebServiceInput dans votre Job.

Configurer un schéma avancé

Cette section décrit la définition d’un schéma Web Service avancé (Advanced WebService).

Définissez les schémas d’entrée et de sortie ainsi que les mappings entre les schémas et les paramètres dans les onglets Input mapping et Output mapping.

Remarque:
Selon le type de sortie, vous pouvez choisir de normaliser ou dénormaliser les résultats en cliquant sur les boutons Normalize et Denormalize.

Définir les propriétés générales du schéma Web Service avancé

Procédure

1. Dans la vue Repository, développez le nœud Metadata.
3. Remplissez les informations génériques du schéma, comme son nom (Name) et sa Description.
4. Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

**Sélectionner le type de schéma (Advanced)**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, vous allez définir le type de schéma, **Simple** ou **Advanced**. Dans cet exemple, sélectionnez le type **Advanced**.

**Procédure**

1. Dans la boîte de dialogue, sélectionnez **Advanced WebService**.
2. Cliquez sur **Next** pour définir plus précisément les paramètres du service web.

**Définir l’opération et le nom du port**

**Procédure**

1. Dans le champ **WSDL**, saisissez l’URI du fichier Web Service WSDL. Vous pouvez également cliquer sur le bouton **Browse...** afin de parcourir votre répertoire si votre WSDL est stocké localement.

2. Cliquez sur le bouton **Refresh** afin de récupérer la liste des noms de port et des opérations disponibles.
3. Dans la zone **Port Name**, sélectionnez le nom du port à utiliser, `countrySoap12` dans cet exemple.

4. Dans la zone **Operation**, sélectionnez l'opération à effectuer.
   

**Définir les schémas d’entrée et les mappings**

**Pourquoi et quand exécuter cette tâche**

Pour définir le schéma d’entrée et ses mappings, procédez comme suit :

**Procédure**

1. Cliquez sur l’onglet **Input mapping** afin de définir le schéma d’entrée et les paramètres nécessaires à l’exécution de l’opération.
2. Dans le tableau de droite, sélectionnez la ligne parameters et cliquez sur le bouton [+] afin d’ouvrir la boîte de dialogue ParameterTree.

3. Sélectionnez le paramètre que vous souhaitez utiliser et cliquez sur OK afin de fermer la boîte de dialogue.
Une nouvelle ligne s’affiche avec le paramètre que vous avez ajouté, CountryCode dans cet exemple.


5. Définissez le schéma d’entrée.
Dans cet exemple, le schéma n’a qu’une colonne : CountryCode.

6. Cliquez sur OK afin de valider votre ajout et fermer la boîte de dialogue.

7. Créez les mappings entre les colonnes du schéma et les paramètres.
Un lien rouge montre que la colonne est mappée.

Remarque:
S'il est disponible, utilisez le bouton Auto map situé en haut de l'onglet, il permet d'effectuer cette opération automatiquement.

Définir les schémas de sortie et les mappings

Pourquoi et quand exécuter cette tâche
Pour définir le schéma de sortie et ses mappings, procédez comme suit :

Procédure
1. Cliquez sur l'onglet Output mapping afin de définir le schéma de sortie et ses paramètres.
2. Dans le tableau de gauche, sélectionnez la ligne parameters et cliquez sur le bouton [+], afin d'ajouter un paramètre.
   La boîte de dialogue ParameterTree s'ouvre.
3. Sélectionnez le paramètre et cliquez sur OK pour fermer la boîte de dialogue.
   Une nouvelle ligne s'affiche avec le paramètre que vous avez ajouté, GetCountryByCountryCodeResult dans cet exemple.

4. Dans le tableau de droite, cliquez sur le bouton [...] afin d'ouvrir la boîte de dialogue Schema.

5. Définissez le schéma de sortie.
   Dans cet exemple, le schéma n'a qu'une colonne : Result.

6. Cliquez sur OK afin de valider votre ajout et fermer la boîte de dialogue.

7. Créez les mappings entre le schéma de sortie et les paramètres.
   Dans cet exemple, cliquez sur la ligne parameters.GetCountryByCountyCodeResult, dans le tableau de gauche, puis déposez-la dans la colonne Result, à droite.
8. Cliquez sur **Next** pour finaliser le schéma.

**Finaliser le schéma Web Service avancé**

**Pourquoi et quand exécuter cette tâche**

Dans cette étape, l’assistant affiche le schéma de sortie généré.
Vous pouvez personnaliser le nom de la métadonnee dans le champ **Name** (par défaut **Output**), ajouter un commentaire dans le champ **Comment** et apporter des modifications à l'aide de la barre d'outils, par exemple :

**Procédure**

1. Ajoutez ou supprimez une colonne à l'aide des boutons et .
2. Modifiez l'ordre des colonnes à l'aide des boutons et .
3. Cliquez sur **Finish** pour terminer la création du schéma avancé.

Le nouveau schéma apparaît dans le **Repository**, sous le nœud Web Service correspondant. Vous pouvez à présent le déposer dans l'espace de modélisation graphique de votre Job en tant que composant **tWebService**.

**Centraliser une connexion FTP**

Si vous vous connectez régulièrement à un serveur FTP vous pouvez centraliser les informations de connexion à cette base de données sous le nœud **Metadata** de la vue **Repository**.
Toutes les connexions ainsi créées apparaissent sous le nœud des connexions aux serveurs FTP, dans la vue Repository.

Vous pouvez glisser-déposer la métadonnée du Repository dans l’espace de modélisation graphique. Une boîte de dialogue s’ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d’informations concernant l’action de déposer des métadonnées dans l’espace de modélisation graphique, consultez Comment utiliser les métadonnées centralisées dans un Job à la page 438.

**Configurer les propriétés générales de la connexion FTP**

**Pourquoi et quand exécuter cette tâche**

Pour créer une connexion à un serveur FTP, suivez les étapes ci-dessous :

**Procédure**

1. Développez le nœud Metadata dans la vue Repository.

2. Cliquez-droit sur FTP puis sélectionnez Create FTP dans le menu contextuel. Un assistant de connexion s’ouvre.
3. Remplissez les informations génériques du schéma, comme son nom (Name) et sa Description.

   **Remarque:**
   Le champ Status est un champ personnalisé que vous pouvez définir dans les Préférences (Window > Preferences). Pour plus d'informations concernant la définition des préférences, consultez Configuration des préférences du Studio Talend à la page 502.

4. Cliquez sur Next une fois terminé, pour renseigner les informations de connexion au serveur FTP.

   **Se connecter à un serveur FTP**

   **Pourquoi et quand exécuter cette tâche**
   Dans cette étape, renseignez les informations et les paramètres de connexion à votre serveur FTP.

   **Procédure**
   1. Dans les champs Username et Password, saisissez respectivement votre identifiant de connexion au serveur et votre mot de passe.
2. Dans le champ **Host**, saisissez le nom de l’hôte de votre serveur FTP.
3. Dans le champ **Port**, saisissez le numéro du port correspondant.
5. Dans la liste **Connection Model**, sélectionnez le mode de connexion que vous souhaitez utiliser.
   - Sélectionnez **Passive** si vous souhaitez que le serveur FTP détermine le port de connexion pour le transfert des données.
   - Sélectionnez **Active** si vous souhaitez déterminer vous-même ce port.
6. Dans la zone **Parameter**, sélectionnez le type d’utilisation du serveur FTP. Pour une utilisation standard de votre serveur FTP, vous n’avez pas besoin de sélectionner une option.
   - Cochez la case **SFTP Support** pour utiliser votre serveur FTP via le protocole de sécurité SSH.
     Une liste **Authentication method** apparaît alors. Sélectionnez **Public key** ou **Password** selon ce que vous utilisez.
   - Cochez la case **FTPs Support** pour utiliser votre serveur via le protocole de sécurité SSL.
   - Cochez la case **Use Socks Proxy** si vous souhaitez utiliser un proxy Socks, puis renseignez les informations du proxy (respectivement le nom de l’hôte, le numéro du port, l’identifiant et le mot de passe de l’utilisateur).
7. Cliquez sur **Finish** pour fermer l’assistant.
Utilisation du mapper hiérarchique

Le Studio Talend vous permet d’accéder aux structures, mappings et conteneurs d’espaces de noms créés dans la perspective Mapping. Pour ce faire, développez le nœud Hierarchical Mapper situé dans le dossier Metadata du Repository dans la perspective Integration.

Pour plus d’informations concernant l’utilisation de ces éléments, consultez le guide Talend Data Mapper User Guide (en anglais).

Exporter une métadonnée en tant que contexte et réutiliser ses paramètres de contexte pour configurer une connexion

Si l’option Export as context est disponible pour une métadonnée, vous pouvez exporter les informations de connexion dans un nouveau groupe de contextes, dans le référentiel, afin de les réutiliser dans d’autres connexions ou dans différents Jobs, ou réutiliser des variables d’un groupe de contextes existant afin de configurer vos métadonnées.

Exporter des informations de connexion en tant que variables de contexte

Pourquoi et quand exécuter cette tâche

Pour exporter des informations de connexion en tant que variables de contexte dans un nouveau groupe de contextes dans le Repository, effectuez les étapes suivantes :

Procédure

1. Après avoir créé ou modifié une métadonnée dans l’assistant, cliquez sur Export as context.
2. Dans l’assistant **Create / Reuse a context group** qui s’ouvre, sélectionnez **Create a new repository context** et cliquez sur **Next**.
3. Saisissez un nom pour le groupe de contextes à créer et ajoutez les informations générales, comme une description, si nécessaire.

Le nom de la métadonnée est proposé par l’assistant comme nom du groupe de contextes et les informations que vous fournissez dans le champ **Description** apparaissent sous la forme d’une infobulle lorsque vous placez votre souris sur le groupe de contexte dans le Repository.
4. Cliquez sur **Next** pour créer et voir le groupe de contextes ou cliquez sur **Finish** afin de terminer la création du contexte et retourner directement à l’assistant de connexion.

Dans cet exemple, cliquez sur **Next**.

5. Vérifiez les résultats de génération du groupe de contextes.

Pour modifier les variables de contexte, allez dans le nœud **Contexts** de la vue **Repository**, cliquez-droit sur le nouveau groupe de contextes et sélectionnez **Edit context group** pour ouvrir l’assistant **Create / Edit a context group** après fermeture de l’assistant de connexion.

Pour modifier le contexte par défaut ou ajouter de nouveaux contextes, cliquez sur le bouton [+](#) dans le coin supérieur droit de l’assistant.

Pour ajouter une nouvelle variable de contexte, cliquez sur le bouton [+](#) au bas de l’assistant.

Pour plus d’informations concernant la gestion des contextes et des variables, consultez **Utiliser les contextes et les variables** à la page 88.
6. Cliquez sur **Finish** afin de terminer la création du contexte et retourner à l’assistant de connexion.
Les champs des informations de connexion de l’assistant sont paramétrés avec les variables de contexte.

Pour supprimer les paramètres définis, cliquez sur le bouton **Revert Context**.

**Utiliser les variables d’un groupe de contextes existant afin de configurer une connexion**

**Pourquoi et quand exécuter cette tâche**

Pour utiliser les variables d’un groupe de contextes existant stocké dans le référentiel afin de configurer une connexion, suivez les étapes ci-dessous :

**Procédure**

1. Après avoir créé ou modifié une métadonnée dans l’assistant, cliquez sur **Export as context**.
2. Dans l'assistant **Create / Reuse a context group** qui s'ouvre, sélectionnez **Reuse an existing repository context** et cliquez sur **Next**.
3. Sélectionnez un groupe de contextes dans la liste et cliquez sur **Next**.
4. Pour chaque variable, sélectionnez le champ correspondant des informations de connexion puis cliquez sur **Next** pour visualiser et modifier les variables de contexte, ou cliquez sur **Finish** pour afficher directement les résultats de configuration de la connexion.
Dans cet exemple, cliquez sur **Next**.

5. Modifiez les contextes et/ou les variables de contexte si nécessaire. Si vous avez effectué des modifications, votre groupe de contextes centralisé est automatiquement mis à jour.
Pour plus d’informations concernant la gestion des contextes et des variables, consultez **Utiliser les contextes et les variables** à la page 88.

6. Cliquez sur **Finish** pour valider la réutilisation du contexte et retourner à l’assistant de connexion.
Les champs des informations de connexion de l'assistant sont paramétrés avec les variables de contexte.

Pour supprimer les paramètres définis, cliquez sur le bouton Revert Context.
Comment utiliser les métadonnées centralisées dans un Job

Pourquoi et quand exécuter cette tâche

Si vous utilisez régulièrement les mêmes fichiers et les mêmes connexions aux bases de données pour plusieurs Jobs, il est conseillé de les sauvegarder dans le Repository, sous le nœud Metadata. Différents dossiers sous ce nœud Metadata regroupent les connexions créées, qu’elles soient des connexions aux bases de données, aux fichiers ou aux systèmes.

Différents assistants vous permettront de centraliser les connexions et les métadonnées des schémas, dans la vue Repository.

Une fois les métadonnées sauvegardées dans le répertoire Metadata du Repository, vous pouvez directement cliquer sur les composants et les déposer dans l’espace de travail.

Procédure

1. Dans la perspective Integration, développez le nœud Metadata du Repository ainsi que le(s) dossier(s) où sont stockées les données que vous voulez utiliser.

2. Déposez la connexion/le schéma choisi(e) dans l’espace de modélisation graphique.

Une boîte de dialogue vous propose de sélectionner le composant à utiliser parmi la liste des composants disponibles.
3. Sélectionnez le composant que vous souhaitez utiliser et cliquez sur OK. Le composant sélectionné s'affiche dans l'espace de modélisation graphique.

Résultats

Sinon, en fonction du type de composant (Input ou Output) que vous souhaitez utiliser, suivez les instructions suivantes :

- **Output** : Maintenez la touche Ctrl enfoncée pour déposer le composant sélectionné dans l'éditeur graphique et ainsi l'ajouter à votre Job Design.
- **Input** : Maintenez la touche Ctrl enfoncée pour déposer le composant sélectionné dans l'éditeur graphique et ainsi l'ajouter à votre Job Design.

Si vous double-cliquez sur le composant, l'onglet Component affiche les détails des connexions sélectionnées ainsi que les informations du schéma sélectionné.

**Remarque :**
Si vous avez sélectionné une connexion sans sélectionner de schéma, le premier schéma rencontré remplira les propriétés.
Utilisation des routines

Gestion des routines

Définition des routines

Les routines sont des fonctions Java plus ou moins complexes, généralement utilisées pour factoriser du code. Elles permettent d’étendre les possibilités de traitement des données dans un ou plusieurs Job(s) technique(s).

Vous pouvez ainsi centraliser dans la vue Repository tous les morceaux de codes utilisés fréquemment ou tirer parti des méthodes déjà en place dans votre entreprise en les appelant via les routines. Cette factorisation facilite également la résolution des problèmes éventuels et permet la mise à jour à la volée du code utilisé dans des Jobs multiples.

En outre, certaines routines système reprennent les méthodes Java les plus courantes, dans une syntaxe Talend, et vous permettent de faire remonter les erreurs Java directement dans le Studio, facilitant ainsi l’identification et la résolution des problèmes au cours du développement de vos processus d’intégration avec Talend.

Les routines peuvent être de deux types :

- Routines Système : un certain nombre de routines système sont fournies par défaut. Elles sont classées selon le type de données qu’elles traitent : numérique, chaîne de caractères, date...
- Routines utilisateur : ce sont les routines que vous créez ou adaptez à partir de routines existantes.

Remarque: Il n’est pas nécessaire de connaître le langage Java pour créer et utiliser les routines Talend.

Toutes les routines sont conservées sous le nœud Code > Routines de la vue Repository sur la gauche du Studio.

Pour plus d’informations concernant les routines système, consultez Accès aux routines système à la page 440.

Pour plus d’informations concernant la création des routines utilisateur, consultez Créer des routines personnalisées à la page 443.

Remarque: Vous pouvez également définir les dépendances des routines dans des Jobs. Pour ce faire, cliquez-droit sur un Job dans la vue Repository et sélectionnez Set up routine dependencies. Dans la boîte de dialogue qui s’ouvre, toutes les routines sont définies par défaut. Vous pouvez utiliser la barre d’outils afin de supprimer les routines, si nécessaire.

Accès aux routines système

Pourquoi et quand exécuter cette tâche

Pour accéder aux routines système, cliquez sur Code > Routines > system. Les routines ou fonctions système sont regroupées par classe selon leur usage.

Remarque: Le dossier system, ainsi que son contenu est en lecture seule.
Chaque classe répertoriée dans le dossier system contient plusieurs routines (fonctions). Double-cliquez sur la classe que vous souhaitez ouvrir.

Toutes les routines, autrement dit toutes les fonctions présentes dans une classe, se composent d’un commentaire explicatif suivi du code correspondant Java. Dans la vue Routines, vous pouvez utiliser la barre de défilement pour consulter les différentes routines. Ou :

**Procédure**

1. Appuyez sur *Ctrl+O* dans la routine ouverte.
   Une boîte de dialogue affiche une liste des différentes routines de la catégorie.
2. Cliquez sur la routine souhaitée.
   La vue bascule vers la section comprenant le texte descriptif de la routine et le code correspondant.

**Remarque:** La syntaxe d’appel des routines est sensible à la casse.

**Personnalisation des routines système**

**Pourquoi et quand exécuter cette tâche**

Si les routines système ne répondent pas exactement à vos besoins, vous pouvez les personnaliser en copiant leur contenu dans des routines utilisateur que vous créez.

Pour personnaliser une routine système :
**Procédure**

1. Créez d’abord une routine utilisateur en suivant les étapes décrites dans la Créer des routines personnalisées à la page 443. Cette routine s’ouvre dans l’espace de travail et contient l’exemple de base de routine.

2. Ensuite, sous Code > Routines > system, sélectionnez la classe des routines contenant la (les) routine(s) à personnaliser.


5. Dans l’espace de travail, sélectionnez tout ou partie du code, et copiez-le via Ctrl+C.


Il est recommandé d’utiliser la partie commentaire (en bleu) pour détailler les paramètres d’entrée et de sortie de votre routine et ainsi en faciliter la maintenance et la réutilisation.

**Gestion des routines personnalisées**

Le Studio Talend vous offre la possibilité de créer des routines personnalisées, de les modifier, ou de modifier des routines existantes, afin de répondre à vos besoins spécifiques.
**Utilisation des routines**

**Créer des routines personnalisées**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez créer vos propres routines pour répondre à vos besoins particuliers de factorisation. Comme les routines système, ces routines utilisateur seront centralisées dans la vue **Repository** (référentiel) sous **Code > Routines**. Vous pourrez ainsi les organiser par dossier selon vos besoins et les appeler facilement dans tous vos Jobs.

Pour créer un nouvelle routine :

**Procédure**

1. Dans la vue **Repository**, cliquez sur **Code** pour accéder aux **Routines**.

   ![Code et Routines](image)

2. Puis cliquez-droit sur **Routines** et sélectionnez **Create Routine**.

3. La boîte de dialogue **New routine** s'ouvre. Saisissez les informations nécessaires à la création de la routine, notamment son nom et sa description.

4. Cliquez sur **Finish** pour valider la création et accéder à l’éditeur.
La routine utilisateur nouvellement créée apparaît directement sous le nœud **Routines** dans la vue **Repository**. L’espace de travail s’ouvre sur un modèle de routine contenant par défaut un exemple simple, composé d’un commentaire en bleu, suivi du code correspondant.

**Remarque:** Il est recommandé de documenter votre routine utilisateur à l’aide d’un commentaire détaillé. Ce commentaire inclut généralement les paramètres d’entrée et de sortie attendus pour l’utilisation de votre routine, ainsi que le résultat retourné par la routine et un exemple d’illustration. Ces informations sont généralement utiles pour la maintenance des routines et le travail collaboratif.

L’exemple de code suivant est fourni par défaut :

```java
public static void helloExample(String message) {
    if (message == null) {
        message = "World";  //NON-NLS-1$
    }
    System.out.println("Hello " + message + " !");
}
```

5. Adaptez ou remplacez ce modèle par votre propre code et sauvegardez-le à l’aide de **Ctrl+S**.

Vous pouvez également copier tout ou partie d’une classe ou routine système vers une routine utilisateur à l’aide des raccourcis du presse-papier : **Ctrl+C** et **Ctrl+V** afin de les adapter à vos besoins.

**Résultats**

**Remarque:** Vous pouvez également copier tout ou partie d’une classe ou routine système vers une routine utilisateur à l’aide des raccourcis du presse-papier : **Ctrl+C** et **Ctrl+V** afin de les adapter à vos besoins.
**Modifier des routines**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez à tout moment modifier les routines que vous avez créées.

**Remarque:** Le dossier system, ainsi que toutes les routines système, est en lecture seule.

Pour éditer vos routines utilisateur :

**Procédure**

1. Cliquez-droit sur la routine que vous souhaitez éditer, puis sélectionnez Edit Routine.
2. La routine s’ouvre dans l’espace de travail, et vous pouvez la modifier.
3. Une fois adaptée à vos besoins, appuyez sur Ctrl+S afin de l’enregistrer.

**Résultats**

Si vous souhaitez réutiliser une routine système pour vos besoins spécifiques, consultez Personnalisation des routines système à la page 441.

**Modifier les bibliothèques des routines utilisateur**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez modifier la bibliothèque de chaque routine utilisateur en important des bibliothèques externes (généralement des fichiers .jar) pour la routine sélectionnée. Ces fichiers externes seront affichés, comme les modules, dans la vue Modules de votre Studio. Pour plus d’informations concernant la vue Modules, consultez le Guide d’installation et de migration Talend.

La bibliothèque importée sera également listée dans le fichier bibliothèque de votre Studio.

Pour éditer une bibliothèque de routine utilisateur, procédez comme suit :

**Procédure**

1. Si la bibliothèque qui doit être importée n’est pas disponible sur votre machine, téléchargez-la ou installez-la en utilisant la vue Modules. Vous pouvez également la télécharger ou la stocker dans un dossier local.
2. Dans la vue Repository, développez les nœuds Code > Routines.
3. Cliquez-droit sur la routine utilisateur dont vous souhaitez éditer la bibliothèque, puis sélectionnez Edit Routine Library.
   
   La boîte de dialogue [Import External Library] s’ouvre.
4. Cliquez sur **New** pour ouvrir la boîte de dialogue **[New Module]** dans laquelle vous pourrez importer la bibliothèque externe.

5. Si vous avez installé la bibliothèque à l’aide de la vue **Modules** :
   - Sélectionnez l’option **Platform** puis la bibliothèque dans la liste.
• Sélectionnez l’option Artifact repository (local m2/nexus) > Find by name ou Artifact repository (local m2/nexus) > Find by Maven URI, puis spécifiez le nom complet ou l’URI Maven du module de la bibliothèque, puis cliquez sur le bouton Detect the module install status afin de valider son statut d’installation.

6. Si vous avez stocké le fichier de la bibliothèque dans un répertoire local :
   a) Sélectionnez l’option Artifact repository (local m2/nexus).
   b) Sélectionnez l’option Install a new module et cliquez sur le bouton [...] pour parcourir votre système jusqu’au fichier de bibliothèque.
   c) Si vous devez personnaliser l’URI Maven de la bibliothèque, cochez la case Custom MVN URI, spécifiez la nouvelle URI et cliquez sur le bouton Detect the module install status afin de valider son statut d’installation.

**Remarque:**

Modifier l’URI Maven pour un module externe affecte toutes les connexions des composants et métadonnées utilisant ce module au sein du projet.

Lorsque vous travaillez sur un projet distant, vos paramètres d’URI Maven personnalisée seront automatiquement synchronisés avec Talend Artifact Repository et seront utilisés lorsque d’autres utilisateurs travailleront sur le même projet et installeront le module externe.

7. Cliquez sur OK afin de confirmer vos modifications.
   Le fichier bibliothèque importé est affiché dans la liste Library File, dans la boîte de dialogue Import External Library.

**Remarque:** Vous pouvez supprimer tout fichier de routine déjà importé en sélectionnant le fichier dans la liste Library File et en cliquant sur le bouton Remove.

8. Cliquez sur Finish pour fermer la boîte de dialogue.

### Appel d’une routine à partir d’un Job

Prérequis : Pour pouvoir exécuter votre routine, vous devez avoir créé au minimum un Job. Pour plus d’informations concernant la création d’un Job, consultez Créer un Job à la page 30.

Vous avez la possibilité d’appeler toutes les routines (utilisateur et système) à partir des composants de vos Jobs afin d’exécuter automatiquement votre routine lors de l’exécution du Job qui la contient.

Pour accéder à toutes les routines sauvegardées dans le dossier Routines de la vue Repository, appuyez sur Ctrl+Espace dans n’importe quel champ de la vue Basic settings de n’importe quel composant Talend utilisé dans votre Job. Sélectionnez ensuite la routine que vous souhaitez exécuter.
Pour faire appel à l'une de ces routines, il vous suffit d'appeler le nom de la classe suivi du nom de la routine puis des paramètres attendus, dans n'importe quel champ de Basic Settings, de la manière qui suit :

```
<ClassName>.<RoutineName>
```

**Cas d’utilisation : Cas d’utilisation : créer un fichier à la date du jour**

**Pourquoi et quand exécuter cette tâche**

Le scénario suivant illustre l’utilisation d’une routine. Le Job se compose d'un composant unique qui appelle une routine système.

**Procédure**

1. Dans la Palette, cliquez sur le dossier File > Management, puis glissez un tFileTouch dans l’éditeur graphique. Ce composant permet de créer un fichier vide.
2. Double-cliquez sur le composant afin d’afficher sa vue Basic settings dans l’onglet Component.
3. Dans le champ FileName, saisissez le chemin d’accès à votre fichier, ou cliquez sur le bouton [...] afin de parcourir votre répertoire.
4. Fermez les guillemets avant l’extension de votre fichier, comme suit : "D:/Input/customer.txt".

5. Ajoutez le signe plus (+) entre les guillemets fermants et l’extension du fichier.

6. Puis appuyez sur Ctrl+Espace pour ouvrir la liste exhaustive des routines. Dans la liste d’auto-complétion qui s’affiche, sélectionnez Talend Date.getDate pour utiliser la routine de Talend qui permet d’obtenir la date actuelle.

7. Modifiez le format de date fourni par défaut si besoin.

8. Saisissez + après la variable getDate pour terminer l’appel de routine, puis entourez de guillemets l’extension du fichier.


Le composant tFileTouch a créé un fichier vide qui porte la date du jour, telle que récupérée lors de l’exécution de la routine appelée GetDate.

Avertissement: Si vous êtes sous Windows, les ":" entre les heures et les minutes, et entre les minutes et les secondes doivent être retirés.

Routines Système

Routines de type Numeric

Les routines numériques permettent de retourner des nombres entiers ou décimaux afin de les utiliser comme paramètres dans un ou plusieurs composant(s) d’un Job, par exemple pour ajouter un identifiant numérique. Par exemple, pour ajouter un identifiant numérique
Pour accéder à ces routines, double-cliquez sur la classe **Numeric** du dossier **system**. La classe **Numeric** contient plusieurs routines, notamment : séquentielle (sequence), aléatoire (random) et décimale (convertImpliedDecimalFormat) :

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>sequence</td>
<td>Retourne un identifiant numérique incrémenté.</td>
<td>Numeric.sequence(&quot;Nom du Paramètre&quot;, valeur de début, valeur d'incrément)</td>
</tr>
<tr>
<td>resetSequence</td>
<td>Crée une séquence si elle n'existe pas et attribue une nouvelle valeur de début.</td>
<td>Numeric.resetSequence (Identifiant de la séquence, valeur de début)</td>
</tr>
<tr>
<td>removeSequence</td>
<td>Supprime une séquence.</td>
<td>Numeric.RemoveSequence (Identifiant de la séquence)</td>
</tr>
<tr>
<td>random</td>
<td>Retourne un entier au hasard entre les valeurs minimale et maximale.</td>
<td>Numeric.random(valeur limite de début, valeur limite de fin)</td>
</tr>
<tr>
<td>convertImpliedDecimalFormat</td>
<td>Retourne un décimal à l'aide d'un modèle décimal implicite.</td>
<td>Numeric.convertImpliedDecimalFormat(&quot;Format Cible&quot;, valeur à convertir)</td>
</tr>
</tbody>
</table>

Les trois routines **sequence**, **resetSequence** et **removeSequence** sont très proches.

- La routine **sequence** est utilisée pour créer un identifiant de séquence, nommé **s1** par défaut, dans le Job. Cet identifiant de séquence est global dans le Job.
- La routine **resetSequence** peut être utilisée pour initialiser la valeur de l'identifiant de séquence créé par la routine **sequence**.
- La routine **removeSequence** est utilisée pour supprimer l'identifiant de séquence d'une liste de variables globales dans le Job.

**Exemple de création séquentielle**

Vous pouvez tester simplement la routine **sequence**, à l'aide d'un composant **tJava** par exemple, pour contrôler la création d'un identifiant incrémenté automatiquement :

```java
System.out.println(Numeric.sequence("s1",1,1));
System.out.println(Numeric.sequence("s1",1,1));
```

L'identifiant numérique est généré et incrémenté automatiquement par la routine :

```
[statistics] connecting to socket on port 3360
[statistics] connected
1
2
```

**Exemple de conversion décimale implicite**

Vous pouvez tester simplement la routine **convertImpliedDecimalFormat** à l'aide d'un composant **tJava**, pour vérifier la conversion d'un nombre décimal implicite :

```java
System.out.println(Numeric.convertImpliedDecimalFormat("9V99","123"));
```

La valeur saisie en paramètre est convertie automatiquement par la routine selon le format décimal implicite fourni :
Utilisation des routines

Routines de type Relational

Les routines relationnelles permettent de vérifier une affirmation basée sur des booléens.

Pour accéder à ces routines, double-cliquez sur Relational, dans le dossier system. La classe Relational contient plusieurs routines notamment :

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISNULL</td>
<td>Vérifie si la variable donnée est de valeur nulle. Retourne true si la valeur est NULL et false si la valeur ne l’est pas.</td>
<td>Relational.ISNULL(variable)</td>
</tr>
<tr>
<td>Not</td>
<td>Retourne le complément de la valeur logique d’une expression.</td>
<td>Relational.NOT(expression)</td>
</tr>
<tr>
<td>isNull</td>
<td>Vérifie si la variable donnée est de valeur nulle. Retourne 1 si la valeur est NULL et 0 si la valeur n’est pas NULL.</td>
<td>Relational.isNull(variable)</td>
</tr>
</tbody>
</table>

Vous pouvez tester une routine Relational, comme la routine ISNULL, à l’aide d’un composant tJava par exemple :

```java
String str = null;
System.out.println(Relational.ISNULL(str));
```

Dans cet exemple, le résultat de la vérification s’affiche dans la vue Run :

```
Starting job test_routine at 14:14 04/02/2010.
[statistics] connecting to socket on port 3375
[statistics] connected
true
[statistics] disconnected
Job test_routine ended at 14:14 04/02/2010. [exit code=0]
```

Routines de type StringHandling

Les routines de traitement des chaînes de caractères permettent d’effectuer différents types d’opérations et de vérifications, basées sur des méthodes Java, sur des expressions alphanumériques.

Pour accéder à ces routines, double-cliquez sur la classe StringHandling du dossier system. La classe StringHandling contient notamment les routines suivantes :
<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>Vérifie si l’expression est triée par ordre alphabétique. Retourne le booléen true si l’ordre alphabétique est vérifié, et false, dans le cas inverse.</td>
<td><code>StringHandling.ALPHA(&quot;chaîne à vérifier&quot;)</code></td>
</tr>
<tr>
<td>IS_ALPHA</td>
<td>Vérifie si l’expression ne contient que des caractères alphabétiques. Retourne le booléen true si l’ordre alphabétique est vérifié, et false, dans le cas inverse.</td>
<td><code>StringHandling.IS_ALPHA(&quot;chaîne à vérifier&quot;)</code></td>
</tr>
<tr>
<td>CHANGE</td>
<td>Remplace un élément d’une chaîne de caractères par l’élément de remplacement défini et retourne la nouvelle chaîne.</td>
<td><code>StringHandling.CHANGE(&quot;chaîne à vérifier&quot;, &quot;chaîne à remplacer&quot;,&quot;chaîne de remplacement&quot;)</code></td>
</tr>
<tr>
<td>COUNT</td>
<td>Retourne le nombre d’occurrences d’une sous-chaîne dans une chaîne de caractères.</td>
<td><code>StringHandling.COUNT(&quot;chaîne à vérifier&quot;,&quot;chaîne à compter&quot;)</code></td>
</tr>
<tr>
<td>DOWNCASE</td>
<td>Convertit toutes les majuscules d’une expression en minuscules et retourne la nouvelle chaîne.</td>
<td><code>StringHandling.DOWNCASE(&quot;chaîne à convertir&quot;)</code></td>
</tr>
<tr>
<td>UPCASE</td>
<td>Convertit toutes les minuscules d’une expression en majuscules et retourne la nouvelle chaîne.</td>
<td><code>StringHandling.UPCASE(&quot;chaîne à convertir&quot;)</code></td>
</tr>
<tr>
<td>DQUOTE</td>
<td>Entoure une expression de guillemets doubles.</td>
<td><code>StringHandling.DQUOTE(&quot;chaîne à traiter&quot;)</code></td>
</tr>
<tr>
<td>EREPLACE</td>
<td>Remplace toutes les sous-chaînes de caractères correspondant à l’expression régulière donnée, dans l’ancienne chaîne de caractères par le remplacement donné et retourne une nouvelle chaîne de caractères.</td>
<td><code>StringHandling.EREPLACE(oldStr, regex, replacement)</code></td>
</tr>
<tr>
<td>Index</td>
<td>Retourne la position, dans une chaîne de caractères, du premier caractère de la sous-chaîne recherchée. Si la sous-chaîne recherchée n’existe pas dans la chaîne, −1 est retourné.</td>
<td><code>StringHandling.INDEX(&quot;chaîne à vérifier&quot;,&quot;sous-chaîne recherchée&quot;)</code></td>
</tr>
<tr>
<td>Routine</td>
<td>Description</td>
<td>Syntaxe</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>LEFT</td>
<td>Retourne une sous-chaîne correspondant aux n premiers caractères d'une chaîne de caractères.</td>
<td><code>StringHandling.LEFT(&quot;chaîne à vérifier&quot;, nombre de caractères)</code></td>
</tr>
<tr>
<td>RIGHT</td>
<td>Retourne une sous-chaîne correspondant aux n derniers caractères d'une chaîne de caractères.</td>
<td><code>StringHandling.RIGHT(&quot;chaîne à vérifier&quot;, nombre de caractères)</code></td>
</tr>
<tr>
<td>LEN</td>
<td>Retourne la longueur d'une chaîne de caractères.</td>
<td><code>StringHandling.LEN(&quot;chaîne à vérifier&quot;)</code></td>
</tr>
<tr>
<td>SPACE</td>
<td>Retourne une chaîne faite du nombre de caractères vides indiqué.</td>
<td><code>StringHandling.SPACE(nombre d'espaces à créer)</code></td>
</tr>
<tr>
<td>SQUOTE</td>
<td>Entoure une expression de guillemets simples.</td>
<td><code>StringHandling.SQUOTE(&quot;chaîne à traiter&quot;)</code></td>
</tr>
<tr>
<td>STR</td>
<td>Retourne un caractère répété le nombre de fois indiqué.</td>
<td><code>StringHandling.STR('caractère à générer', nombre de répétition)</code></td>
</tr>
<tr>
<td>TRIM</td>
<td>Supprime les espaces et les tabulations en début et fin d'une chaîne de caractères et retourne la nouvelle chaîne.</td>
<td><code>StringHandling.TRIM(&quot;chaîne à traiter&quot;)</code></td>
</tr>
<tr>
<td>BTRIM</td>
<td>Supprime tous les espaces et les tabulations après le dernier caractère non vide d'une chaîne de caractères et retourne la nouvelle chaîne.</td>
<td><code>StringHandling.BTRIM(&quot;chaîne à traiter&quot;)</code></td>
</tr>
<tr>
<td>FTRIM</td>
<td>Supprime tous les espaces et les tabulations jusqu'au premier caractère non vide d'une chaîne de caractères et retourne la nouvelle chaîne.</td>
<td><code>StringHandling.FTRIM(&quot;chaîne à traiter&quot;)</code></td>
</tr>
<tr>
<td>SUBSTR</td>
<td>Retourne une partie d'une chaîne de caractères. Compte tous les caractères, y compris les blancs, en commençant au début de la chaîne de caractères.</td>
<td><code>StringHandling.SUBSTR(string, start, length)</code></td>
</tr>
<tr>
<td>• <code>string</code> : chaîne de caractères à rechercher.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <code>start</code> : position dans la chaîne de caractères à partir de laquelle commencer à compter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <code>length</code> : nombre de caractères à retourner.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routine</td>
<td>Description</td>
<td>Syntaxe</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LTRIM</td>
<td>Supprime les blancs ou les caractères au début d'une chaîne de caractères.</td>
<td>StringHandling.LTRIM(string[, trim_set])</td>
</tr>
<tr>
<td></td>
<td>• string : chaîne de caractères à modifier.</td>
<td>• string : chaîne de caractères à modifier.</td>
</tr>
<tr>
<td></td>
<td>• trim_set : caractères à supprimer du début de la chaîne.</td>
<td>• trim_set : caractères à supprimer du début de la chaîne.</td>
</tr>
<tr>
<td></td>
<td>LTRIM va comparer trim_set à la chaîne, caractère par caractère, en</td>
<td>LTRIM va comparer trim_set à la chaîne, caractère par caractère, en</td>
</tr>
<tr>
<td></td>
<td>commençant par le côté gauche de la chaîne et supprimer les caractères</td>
<td>commençant par le côté gauche de la chaîne et supprimer les caractères</td>
</tr>
<tr>
<td></td>
<td>jusqu'à ne plus trouver de caractère correspondant dans trim_set. Si ce</td>
<td>jusqu'à ne plus trouver de caractère correspondant dans trim_set. Si ce</td>
</tr>
<tr>
<td></td>
<td>paramètre n'est pas spécifié, LTRIM va supprimer tous les caractères</td>
<td>paramètre n'est pas spécifié, LTRIM va supprimer tous les caractères</td>
</tr>
<tr>
<td></td>
<td>blancs du début de la chaîne.</td>
<td>blancs du début de la chaîne.</td>
</tr>
<tr>
<td>RTRIM</td>
<td>Supprime les blancs ou les caractères à la fin d'une chaîne de caractères.</td>
<td>StringHandling.RTRIM(string[, trim_set])</td>
</tr>
<tr>
<td></td>
<td>• string : chaîne de caractères à modifier.</td>
<td>• string : chaîne de caractères à modifier.</td>
</tr>
<tr>
<td></td>
<td>• trim_set : caractères à supprimer de la fin de la chaîne.</td>
<td>• trim_set : caractères à supprimer de la fin de la chaîne.</td>
</tr>
<tr>
<td></td>
<td>RTRIM va comparer trim_set à la chaîne, caractère par caractère, en</td>
<td>RTRIM va comparer trim_set à la chaîne, caractère par caractère, en</td>
</tr>
<tr>
<td></td>
<td>commençant par le côté droit de la chaîne et supprimer les caractères</td>
<td>commençant par le côté droit de la chaîne et supprimer les caractères</td>
</tr>
<tr>
<td></td>
<td>jusqu'à ne plus trouver de caractère correspondant dans trim_set. Si ce</td>
<td>jusqu'à ne plus trouver de caractère correspondant dans trim_set. Si ce</td>
</tr>
<tr>
<td></td>
<td>paramètre n'est pas spécifié, RTRIM va supprimer tous les caractères</td>
<td>paramètre n'est pas spécifié, RTRIM va supprimer tous les caractères</td>
</tr>
<tr>
<td></td>
<td>blancs de la fin de la chaîne.</td>
<td>blancs de la fin de la chaîne.</td>
</tr>
<tr>
<td>LPAD</td>
<td>Convertit une chaîne de caractères en une chaîne d'une longueur spécifiée en</td>
<td>StringHandling.LPAD(first_string, length[, second_string])</td>
</tr>
<tr>
<td></td>
<td>ajoutant des blancs ou des caractères au début de la chaîne de caractères.</td>
<td>• first_string : chaîne de caractères à modifier.</td>
</tr>
<tr>
<td></td>
<td>• length : longueur souhaitée pour la chaîne après remplissage.</td>
<td>• length : longueur souhaitée pour la chaîne après remplissage.</td>
</tr>
<tr>
<td></td>
<td>• second_string : caractères à ajouter à gauche de first_string.</td>
<td>• second_string : caractères à ajouter à gauche de first_string.</td>
</tr>
<tr>
<td>RPAD</td>
<td>Convertit une chaîne de caractères en une chaîne d'une longueur spécifiée,</td>
<td>StringHandling.RPAD(first_string, length[, second_string])</td>
</tr>
<tr>
<td></td>
<td>en ajoutant des blancs ou des caractères à la fin de la chaîne.</td>
<td>• first_string : chaîne de caractères à modifier.</td>
</tr>
<tr>
<td></td>
<td>• length : longueur souhaitée pour la chaîne après remplissage.</td>
<td>• length : longueur souhaitée pour la chaîne après remplissage.</td>
</tr>
<tr>
<td></td>
<td>• second_string : caractères à ajouter à droite de first_string.</td>
<td>• second_string : caractères à ajouter à droite de first_string.</td>
</tr>
<tr>
<td>INSTR</td>
<td>Retourne la position d’un jeu de caractères dans une chaîne de caractères,</td>
<td>StringHandling.INSTR(string, search_value, start, occurrence)</td>
</tr>
<tr>
<td></td>
<td>en comptant de gauche à droite et à partir de 1.</td>
<td>• string : chaîne de caractères à rechercher.</td>
</tr>
<tr>
<td></td>
<td>Notez que cette routine retourne 0 si la recherche n'obtient pas de résultat</td>
<td>• search_value : jeu de caractères à rechercher.</td>
</tr>
<tr>
<td></td>
<td>et NULL si la valeur de la recherche est NULL.</td>
<td>• start : position dans la chaîne de caractères à partir de laquelle</td>
</tr>
<tr>
<td></td>
<td>Par exemple, StringHandling.INSTR(&quot;Talend Technology&quot;, &quot;e&quot;, 3, 2) démarre la recherche au troisième caractère e et retourne 7, la position du second caractère e.</td>
<td>commencer la recherche. La valeur par défaut est 1, ce qui signifie</td>
</tr>
<tr>
<td>TO_CHAR</td>
<td>Convertit des valeurs numériques en chaînes de caractères textuelles.</td>
<td>StringHandling.TO_CHAR(numeric_value)</td>
</tr>
</tbody>
</table>
Exemple de vérification de tri alphabétique

Vous pouvez tester simplement la routine ALPHA, à l'aide d'un composant tJava par exemple, pour vérifier si la chaîne est triée dans l'ordre alphabétique :

```java
System.out.println(StringHandling.ALPHA("abcdefg"));
```

La vérification retourne un booléen.

Exemple de vérification de type alphabétique

Vous pouvez tester simplement la routine IS_ALPHA, à l'aide d'un composant tJava par exemple, pour vérifier si la chaîne est de type alphabétique ou non :

```java
System.out.println(StringHandling.IS_ALPHA("ab33cd"));
```

La vérification retourne un booléen.

Exemple de remplacement de chaîne

Vous pouvez tester simplement la routine CHANGE, à l'aide d'un composant tJava par exemple, pour contrôler le remplacement d'une chaîne par une autre :

```java
System.out.println(StringHandling.CHANGE("hello world!", "world", "guy"));
```

La vérification retourne un booléen.

Exemple de vérification de chaîne

Vous pouvez tester simplement la routine INDEX, à l'aide d'un composant tJava par exemple, pour vérifier si la chaîne contient ou pas un caractère ou une chaîne de caractères :

```java
System.out.println(StringHandling.INDEX("hello world!", "hello");
System.out.println(StringHandling.INDEX("hello world!", "world");
System.out.println(StringHandling.INDEX("hello world!", "!");
System.out.println(StringHandling.INDEX("hello world!", "?"));
```

La routine retourne un entier qui correspond à la position du premier caractère de la chaîne recherchée, ou retourne -1 si la chaîne recherchée n'a pu être trouvée : Sinon, la routine retourne -1 si la chaîne recherchée n'a pu être trouvée.
**Utilisation des routines**

*Exemple de calcul de longueur de chaîne*

Vous pouvez tester simplement la routine `LEN`, à l'aide d'un composant `tJava` par exemple, pour contrôler la longueur d'une chaîne :

```java
System.out.println(StringHandling.LEN("hello world!"));
```

La vérification retourne un entier correspondant à la longueur de la chaîne, y compris les espaces et caractères vides :

12

*Exemple de nettoyage d'espaces inutiles*

Vous pouvez tester simplement la routine `FTRIM`, à l'aide d'un composant `tJava` par exemple, afin de retirer, entre autres, une tabulation en tête de chaîne :

```java
System.out.println(StringHandling.FTRIM("  Hello world  "));
```

La vérification retourne la chaîne nettoyée des espaces/tabulations présents en début de chaîne.

*Routines de type TalendDataGenerator*

Les routines de génération de données factices (`TalendDataGenerator`) sont des fonctions qui permettent de générer des ensembles de données de test. Elles se basent sur les listes (factices) de noms, prénoms, adresses, villes et États fournies par `Talend`. Ces routines sont généralement utilisées au moment du développement des Jobs, à l'aide d'un `tRowGenerator` par exemple, pour éviter d’utiliser des données de production ou de l’entreprise.

Pour accéder à ces routines, double-cliquez sur la classe `TalendDataGenerator` du dossier `system` :

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getFirstName</code></td>
<td>retourne un prénom pris aléatoirement dans une liste factice de prénoms américains courants.</td>
<td><code>TalendDataGenerator.getFirstName()</code></td>
</tr>
<tr>
<td>Routine</td>
<td>Description</td>
<td>Syntaxe</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>getLastName</td>
<td>retourne un nom de famille pris aléatoirement dans une liste factice de noms de famille américains courants.</td>
<td>TalendDataGenerator.getLastName()</td>
</tr>
<tr>
<td>getUsStreet</td>
<td>retourne une adresse prise aléatoirement dans une liste factice de rues américaines courantes.</td>
<td>TalendDataGenerator.getUsStreet()</td>
</tr>
<tr>
<td>getUsCity</td>
<td>retourne une ville prise aléatoirement dans une liste de villes connues des États-Unis.</td>
<td>TalendDataGenerator.getUsCity()</td>
</tr>
<tr>
<td>getUsState</td>
<td>retourne le nom d’un État pris aléatoirement dans une liste des Etats américains.</td>
<td>TalendDataGenerator.getUsState()</td>
</tr>
<tr>
<td>getUsStateId</td>
<td>retourne le code pris aléatoirement dans une liste d'abréviations correspondant aux États américains.</td>
<td>TalendDataGenerator.getUsStateId()</td>
</tr>
</tbody>
</table>

**Remarque:** Aucun paramètre d’entrée n’est attendu car la liste des données factices est fournie par Talend.

Vous pouvez personnaliser les données factices en modifiant les routines de génération de données TalendDataGenerator. Pour plus d’informations concernant la personnalisation des routines, consultez Personnalisation des routines système à la page 441.

**Exemple de génération de données factices**

Vous pouvez tester simplement les différentes fonctions de génération de données aléatoires. Vous pouvez tester simplement les différentes fonctions de génération de données aléatoires, telles que `getFirstName()`, `getLastName()`, `getUSCity()`, etc. à l’aide d’un composant `tJava`, pour tester, par exemple, la création d’une liste de données d’un client factice :

```java
System.out.println(TalendDataGenerator.getFirstName());
System.out.println(TalendDataGenerator.getLastName());
System.out.println(TalendDataGenerator.getUsCity());
System.out.println(TalendDataGenerator.getUsState());
System.out.println(TalendDataGenerator.getUsStateId());
System.out.println(TalendDataGenerator.getUsStreet());
```

L’ensemble des données prises aléatoirement dans les listes de données factices est affiché dans la vue Run :
Routines de type TalendDate

Les routines de traitement de dates (TalendDate) permettent d’effectuer différents types d’opération et de vérification sur le format des expressions de type Date.

Pour accéder à ces routines, double-cliquez sur la classe TalendDate du dossier system :

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>addDate</td>
<td>Ajoute n jours, n mois, n heures, n minutes ou n secondes à une Date Java et retourne la nouvelle date. Le paramètre de format de données Date est : yyyy, MM, dd, HH, mm, ss ou SSS.</td>
<td>TalendDate.addDate(&quot;Chaîne de date initiale&quot;, &quot;format de date - ex: yyyy/MM/dd&quot;, entier n,&quot;format de la donnée à ajouter - ex : yyyy&quot;).</td>
</tr>
<tr>
<td>compareDate</td>
<td>Compare tout ou partie de deux dates, selon le modèle de date si spécifié. Retourne 0 si les dates sont identiques, -1 si la première date est antérieure et 1 si la deuxième est antérieure.</td>
<td>TalendDate.compareDate(Date date1, Date date2, &quot;format à comparer - ex : yyyy-MM-dd&quot;)</td>
</tr>
<tr>
<td>diffDate</td>
<td>Retourne la différence entre deux dates, en nombre de jours, mois ou années selon le paramètre de comparaison spécifié.</td>
<td>TalendDate.diffDate(Date1(), Date2(), &quot;format de données à comparer - ex yyyy&quot;)</td>
</tr>
<tr>
<td>diffDateFloor</td>
<td>Retourne la différence entre deux dates, en nombre d’années, mois, jours, heures, minutes, secondes ou millisecondes selon le paramètre de comparaison spécifié.</td>
<td>TalendDate.diffDateFloor(Date1(), Date2(), &quot;format de données à comparer - ex MM&quot;)</td>
</tr>
<tr>
<td>formatDate</td>
<td>Retourne une expression de type date formatée selon le modèle de date spécifié.</td>
<td>TalendDate.formatDate(&quot;format de date - ex : yyyy-MM-dd HH:mm:ss&quot;, Date() à formater)</td>
</tr>
<tr>
<td>formatDateLocale</td>
<td>Transforme une date en une chaîne de caractères de type date/heure selon le modèle et la locale spécifiés.</td>
<td>TalendDate.formatDateLocale(&quot;format cible&quot;, java.util.Date date, &quot;code de la langue ou du pays&quot;)</td>
</tr>
<tr>
<td>getCurrentDate</td>
<td>Retourne la date courante. Aucun paramètre d’entrée n’est attendu.</td>
<td>TalendDate.getCurrentDate()</td>
</tr>
<tr>
<td>Routine</td>
<td>Description</td>
<td>Syntaxe</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getDate</code></td>
<td>Retourne les date et heure courantes dans le format spécifié (optionnel). Cette chaîne peut contenir des chaînes de caractères fixes, ou des variables liées à la date. Par défaut, le format de la chaîne de caractères attendu est DD/MM/CCYY.</td>
<td><code>TalendDate.getDate(&quot;Format of the string - ex: CCYY-MM-DD&quot;)</code></td>
</tr>
<tr>
<td><code>getFirstDayOfMonth</code></td>
<td>Change le jour d’une date en premier jour du mois courant et retourne la nouvelle date.</td>
<td><code>TalendDate.getFirstDayMonth(Date)</code></td>
</tr>
<tr>
<td><code>getLastDayOfMonth</code></td>
<td>Change le jour d’une date en dernier jour du mois courant et retourne la nouvelle date.</td>
<td><code>TalendDate.getLastDayMonth(Date)</code></td>
</tr>
<tr>
<td><code>getPartOfDay</code></td>
<td>Retourne une partie d’une date dans le format spécifié. Cette chaîne peut contenir des chaînes de caractères fixes, ou des variables liées à la date.</td>
<td><code>TalendDate.getPartOfDate(&quot;Chaîne de caractères indiquant la partie de la date à récupérer&quot;, &quot;Chaîne de caractères au format date à parser&quot;)</code></td>
</tr>
<tr>
<td><code>getRandomDate</code></td>
<td>Retourne une date aléatoire, au format ISO.</td>
<td><code>TalendDate.getRandomDate(&quot;Chaîne de caractère de type Date, de &quot;, String maxDate)</code></td>
</tr>
<tr>
<td><code>isDate</code></td>
<td>Vérifie si l’expression est de type Date et correspond au modèle spécifié. Retourne le booléen true si c’est le cas, et false dans le cas inverse.</td>
<td><code>TalendDate.isDate(Date() à vérifier, &quot;format de date cible - ex : yyyy-MM-dd HH:mm:ss&quot;)</code></td>
</tr>
<tr>
<td><code>parseDate</code></td>
<td>Transforme une chaîne de caractères en Date. Retourne une date formatée en standard.</td>
<td><code>TalendDate.parseDate(&quot;format date de la chaîne à parser&quot;, &quot;Chaîne de caractères au format date à parser&quot;)</code></td>
</tr>
<tr>
<td><code>parseDateInUTC</code></td>
<td>modifie une chaîne de caractères en une Date UTC. Retourne une date au format UTC.</td>
<td><code>TalendDate.parseDateInUTC(&quot;format date de la chaîne à parser&quot;, &quot;chaîne de caractères au format date à parser&quot;, &quot;booléen indiquant si le parsing est Lenient, c'est-à-dire accepte la correspondance heuristique avec le format&quot;)</code></td>
</tr>
<tr>
<td><code>parseDateLocale</code></td>
<td>Parse une chaîne de caractères correspondant à un modèle spécifié, et en extrait une date. Retourne une date formatée selon la locale spécifiée.</td>
<td><code>TalendDate.parseDateLocale(&quot;format date de la chaîne à parser&quot;, &quot;Chaîne de caractères au format date à parser&quot;, &quot;code de la langue ou du pays&quot;)</code></td>
</tr>
<tr>
<td><code>setDate</code></td>
<td>Modifie une partie de la date en entrée par un entier spécifié et basé sur le format spécifié.</td>
<td><code>TalendDate.setDate(Date, entier n, &quot;format de la donnée à changer - ex :yyyy&quot;)</code></td>
</tr>
</tbody>
</table>

459
<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>TO_CHAR</td>
<td>Convertit une date en une chaîne de caractères.</td>
<td>TalendDate.TO_CHAR(date[, format])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• date : valeur de la date à convertir en une chaînes de caractères.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• format : chaîne de caractères définissant le format de la valeur à retourner.</td>
</tr>
<tr>
<td>TO_DATE</td>
<td>Convertit une chaîne de caractères en un type Date/Time.</td>
<td>TalendDate.TO_DATE(string[, format])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• string : chaîne de caractères à convertir en type Date/Time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• format : chaîne de caractères de format correspondant à la partie de l’argument string. Si elle n’est pas spécifiée, la valeur string doit être au format de date MM/dd/yyyy HH:mm:ss.SSS.</td>
</tr>
<tr>
<td>ADD_TO_DATE</td>
<td>Ajoute une somme spécifiée à une partie d’une valeur datetime et retourne une date au même format que la date passée à la fonction.</td>
<td>TalendDate.ADD_TO_DATE(date, format, amount)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• date : valeur de la date à modifier.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• format : chaîne de caractères de format spécifiant la partie de la valeur de date à modifier.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaînes de caractères de format valides pour les années : Y, YY, YYY, and YYYY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaînes de caractères de format valides pour les mois : MONTH, MM, and MON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaînes de caractères de format valides pour les jours : D, DD, DDD, DAY, and DY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaînes de caractères de format valides pour les heures : HH, HH12, and HH24.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaîne de caractères de format valide pour les minutes : MI.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaîne de caractères de format valide pour les secondes : SS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Chaîne de caractères de format valide pour les millisecondes : MS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• amount : la valeur de l’entier spécifiant la somme des années, mois, heures, etc via laquelle vous souhaitez modifier la valeur de la date.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Par exemple,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>si TalendDate.getCurrentDate() retourne Mon Apr 24 14:26:03 CST 2017,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TalendDate.ADD_TO_DATE(TalendDate currentDate(), &quot;YY&quot;, 1) va retourner Tue Apr 24 14:26:03 CST 2018.</td>
</tr>
</tbody>
</table>

**Remarque**

Les chaînes de caractères de format valides pour les années : Y, YY, YYY, et YYYY.
Les chaînes de caractères de format valides pour les mois : MONTH, MM, et MON.
Les chaînes de caractères de format valides pour les jours : D, DD, DDD, DAY, et DY.
Les chaînes de caractères de format valides pour les heures : HH, HH12, et HH24.
La chaîne de caractère de format valide pour les minutes : MI.
La chaîne de caractère de format valide pour les secondes : SS.
La chaîne de caractère de format valide pour les millisecondes : MS.
Avertissement:

Dans un format de date, "yyyy" et "YYYY" retournent la même année dans la plupart des cas.

- Cependant, il est possible que le format "YYYY" ne fonctionne pas comme attendu, lorsqu’il est utilisé :
- pour la première semaine de l’année, si l’année ne commence pas par le premier jour de la semaine ;

Par exemple, lorsque vous calculez le jour venant trois jours avant le 2 janvier 2016, le code ci-dessous retourne une date incorrecte :

```java
System.out.println(TalendDate.formatDate("YYYY-MM-dd", TalendDate.addDate(TalendDate.TO_DATE("01/02/2016 08:10:30.123"), -3, "dd")));
```

alors que ce code-ci retourne la date attendue :

```java
System.out.println(TalendDate.formatDate("yyyy-MM-dd", TalendDate.addDate(TalendDate.TO_DATE("01/02/2016 08:10:30.123"), -3, "dd")));
```

Il est donc recommandé d’utiliser "yyyy", qui représente les années calendaires.

Exemple de formatage d’une Date

Vous pouvez tester simplement la routine `formatDate`, à l’aide d’un composant `tJava` par exemple, pour vérifier qu’une expression de type date est dans le format spécifié :

```java
System.out.println(TalendDate.formatDate("dd-MM-yyyy", new Date()));
```

La date du jour est instanciée par la fonction Java `new Date()` et s’affiche dans la vue **Run** :

```
Starting job routine1 at 17:39 25/02/2010.
2010-02-25 17:28:07
Job routine1 ended at 17:28 25/02/2010. [exit code=0]
```

Exemple de vérification d’une Date

Vous pouvez tester simplement la routine `isDate`, à l’aide d’un composant `tJava` par exemple, pour vérifier qu’une expression de type date est dans le format spécifié :

```java
System.out.println(TalendDate.isDate("2010-02-09 00:00:00", "yyyy-MM-dd HH:mm:ss"));
```

Un booléen s’affiche dans la vue **Run** :

```
Starting job routine1 at 17:36 25/02/2010.
true
Job routine1 ended at 17:36 25/02/2010. [exit code=0]
```
Vous pouvez tester simplement la routine `compareTo`, à l’aide d’un composant `tJava` afin de comparer deux dates pour, par exemple, vérifier si la date du jour est identique, antérieure ou ultérieure à une date spécifiée, selon le format défini :

```
System.out.println(TalendDate.compareTo(new Date(), TalendDate.parseDate("yyyy-MM-dd", "2025/11/24")));
```

Dans cet exemple, la date du jour est instanciée par la fonction `new Date()` et la valeur -1 s’affiche dans la vue `Run` pour indiquer que la date du jour est antérieure à la deuxième date :

```
Starting job routine1 at 18:09 25/02/2010.
-1
Job routine1 ended at 18:09 25/02/2010. [exit code=0]
```

**Exemple de configuration de Date**

Vous pouvez tester simplement la routine `setDate`, à l’aide d’un composant `tJava`, pour changer l’année de la date courante par exemple :

```
System.out.println(TalendDate.formatDate("yyyy/MM/dd HH:mm:ss",new Date()));
System.out.println(TalendDate.setDate(new Date(),2011,"yyyy"));
```

La date courante suivie de la nouvelle date configurée s’affiche dans la vue `Run` :

```
Starting job routine1 at 18:03 26/02/2010.
2010/02/26 18:03:14
Sat Feb 26 18:03:14 CET 2011
Job routine1 ended at 18:03 26/02/2010. [exit code=0]
```

**Exemple de parsage de Date**

Vous pouvez tester simplement la routine `parseDate`, à l’aide d’un composant `tJava`, pour mettre une chaîne de type date au format `Date` par exemple :

```
System.out.println(TalendDate.parseDate("yyyy-MM-dd HH:mm:ss", "1979/10/20 19:00:59"));
```

La chaîne de caractères est transformée en `Date` et la date est retournée :

```
Starting job routine1 at 11:58 01/03/2010.
Sat Oct 20 19:00:59 CET 1979
Job routine1 ended at 11:58 01/03/2010. [exit code=0]
```

**Exemple de récupération d’une partie d’une Date**

Vous pouvez tester simplement la routine `getPartOfDate`, à l’aide d’un composant `tJava`, pour récupérer une partie d’une date, par exemple :

```
Date D=TalendDate.parseDate("dd-MM-yyyy HH:mm:ss", "13-10-2010 12:23:45");
System.out.println(D.toString());
System.out.println(TalendDate.getPartOfDate("DAY_OF_MONTH", D));
System.out.println(TalendDate.getPartOfDate("MONTH", D));
System.out.println(TalendDate.getPartOfDate("YEAR", D));
System.out.println(TalendDate.getPartOfDate("DAY_OF_YEAR", D));
System.out.println(TalendDate.getPartOfDate("DAY_OF_WEEK", D));
```
Dans cet exemple, sont retournés dans la vue **Run** le jour du mois (**DAY_OF_MONTH**), le mois (**MONTH**), l’année (**YEAR**), le numéro de jour de l’année (**DAY_OF_YEAR**) et le numéro de jour de la semaine (**DAY_OF_WEEK**). Toutes les données retournées sont de type numérique.

**Remarque:** Dans la console de la vue **Run**, la chaîne de caractères qui renvoie aux mois (**MONTH**) va de 0 à 11 : 0 correspondant à janvier, et 11 correspondant à décembre.

### Exemple de formatage de la Date courante

Vous pouvez tester simplement la routine **getDate**, à l’aide d’un composant **tJava**, pour récupérer la date courante et la formater selon un modèle spécifié, par exemple :

```
System.out.println(TalendDate.getDate("CCYY-MM-DD");
```

La date courante est retournée selon le format spécifié (optionnel) :

```
```

### Routines de type TalendString

Les routines de traitement de chaînes de caractères **Talend** permettent d’effectuer diverses opérations sur des expressions alphanumériques.

Pour accéder à ces routines, double-cliquez sur la classe **TalendString** du dossier **system**. La classe **TalendString** contient notamment les routines suivantes :

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>replaceSpecialCharForXML</td>
<td>retourne une chaîne de caractères où les caractères spéciaux (ex.:&lt;,&gt;,...) ont été remplacés par des caractères XML équivalents.</td>
<td>TalendString.replaceSpecialCharForXML (&quot;chaîne de caractères contenant les caractères spéciaux - ex : Thelma &amp; Louise&quot;)</td>
</tr>
<tr>
<td>Routine</td>
<td>Description</td>
<td>Syntaxe</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>checkCDATAForXML</td>
<td>identifie les chaînes de caractères commençant par <code>&lt;![CDATA[</code> et se terminant par <code>]]&gt;</code> en tant que XML et les retourne sans modification. Transforme les chaînes identifiées comme non-XML sous une forme compatible XML et les retourne ainsi formatées.</td>
<td>TalendString.checkCDATAForXML(&quot;chaîne de caractères à parser&quot;)</td>
</tr>
<tr>
<td>talendTrim</td>
<td>parse la chaîne de caractères en entrée et en retire le/les caractères de remplissage en début ou en fin de chaîne selon la valeur d’alignement spéciﬁée: -1 pour les caractères de remplissage de fin de chaîne, 1 pour ceux de début de chaîne et 0 pour les deux. Puis retourne la chaîne nettoyée.</td>
<td>TalendString.talendTrim(&quot;chaîne de caractères à parser&quot;, &quot;caractère de remplissage à retirer&quot;, position du caractère)</td>
</tr>
<tr>
<td>removeAccents</td>
<td>enlève les accents d’une chaîne de caractères et retourne cette chaîne non accentuée.</td>
<td>TalendString.removeAccents(&quot;Chaîne de caractères&quot;)</td>
</tr>
<tr>
<td>getAsciiRandomString</td>
<td>génère une chaîne de caractères aléatoire, du nombre de caractères spéciﬁés.</td>
<td>TalendString.getAsciiRandomString (entier de la longueur de chaîne)</td>
</tr>
</tbody>
</table>

**Exemple de formatage XML d’une chaîne**

Vous pouvez tester simplement la routine replaceSpecialCharForXML, à l’aide d’un composant tJava, pour formater une chaîne de caractères pour le XML, par exemple :

```java
System.out.println(TalendString.replaceSpecialCharForXML("Thelma & Louise"));
```

Dans cet exemple, le caractère & est remplacé pour être intelligible en XML :

```
Starting job routine1 at 15:48 02/03/2010.
Thelma &amp; Louise
Job routine1 ended at 15:48 02/03/2010. [exit code=0]
```

**Exemple de trimming d’une chaîne**

Vous pouvez tester simplement la routine talendTrim, à l’aide d’un composant tJava, pour retirer des caractères de remplissage en début et en fin de chaîne, par exemple :

```java
System.out.println(TalendString.talendTrim("***talend open studio****", '**', -1));
System.out.println(TalendString.talendTrim("***talend open studio****", '**', 1));
System.out.println(TalendString.talendTrim("***talend open studio****", '**', 0));
```

Les caractères étoiles sont retirés alternativement en début puis en fin de chaîne, et enfin des deux côtés :
Exemple de désaccentuation d’une chaîne

Vous pouvez tester simplement la routine removeAccents, à l’aide d’un composant tJava, pour remplacer les caractères accentués, par exemple :

```java
System.out.println(TalendString.removeAccents("sâcrebleü!");
```

Les caractères avec accent sont remplacés par des caractères sans accent :

Starting job routine at 16:02 02/03/2010.
sacrebleu!
Job routine ended at 16:02 02/03/2010. [exit code=0]

Routines TalendStringUtil

La classe TalendStringUtil contient une seule routine DECODE vous permettant de rechercher une valeur dans un port. Pour accéder à la routine, double-cliquez sur TalendStringUtil dans le dossier system. Pour accéder à la routine, double-cliquez sur TalendStringUtil, dans le dossier system.

<table>
<thead>
<tr>
<th>Routine</th>
<th>Description</th>
<th>Syntaxe</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECODE</td>
<td>Recherche un port pour une valeur que vous spécifiez. Si la fonction trouve la valeur, elle retourne une valeur de résultat, que vous définissez. Vous pouvez construire un nombre illimité de recherches dans une fonction DECODE</td>
<td>TalendStringUtil.DECODE(value, defaultValue, search1, result1[, search2, result2]...)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• value : valeur à rechercher.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• defaultValue : valeur à retourner si la recherche ne trouve pas de valeur correspondante. La valeur par défaut peut être configurée à null.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• search : valeur à rechercher. La valeur de la recherche doit être du même type de données que l’argument value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• result : valeur à retourner si la recherche trouve une valeur correspondante.</td>
</tr>
</tbody>
</table>

Voici un exemple d’utilisation de la routine DECODE avec un composant tJava. Vous devez ajouter un composant tJava à un nouveau Job, puis saisir le code suivant, qui recherchera la valeur pour 10, dans le champ Code de l’onglet Basic settings du composant tJava.

```java
TalendStringUtil<Integer,String> example = new TalendStringUtil<Integer,String>();
System.out.println(example.DECODE(10, "error", 5, "five", 10, "ten", 15, "fifteen", 20, "twenty");
```

Vous devez créer un nouvel objet de type TalendStringUtil. Il est recommandé d’utiliser le type générique pour contraindre les données d’entrée, puis d’utiliser l’objet pour appeler la routine DECODE.

Appuyez sur F6 pour exécuter le Job. Cela retourne ten, qui est le résultat de la valeur 10.
Annexes

Personnaliser le Studio Talend et ses préférences

Personnalisation des paramètres du projet

Pourquoi et quand exécuter cette tâche

Le Studio Talend vous permet de personnaliser les informations et les paramètres du projet en cours, notamment les paramètres de la Palette et du Job, la gestion du versionnement des Jobs, etc.

Pour accéder aux paramètres du projet :

Procédure

   La boîte de dialogue Project Settings s’ouvre.
2. Dans l’arborescence à gauche de la boîte de dialogue, sélectionnez le paramètre que vous souhaitez personnaliser et personnalisez-le dans la partie droite de la boîte de dialogue.

Résultats

À partir de cette boîte de dialogue, vous pouvez aussi exporter ou importer l’intégralité des paramètres de votre projet.

- Pour exporter les paramètres, cliquez sur le bouton Export. L’export générera un fichier XML contenant tous les paramètres de votre projet.
- Pour importer les paramètres, cliquez sur le bouton Import et sélectionnez le fichier XML contenant les paramètres d’un autre projet à appliquer au projet en cours.

Configuration du niveau de conformité du compilateur

Pourquoi et quand exécuter cette tâche

Le niveau de conformité du compilateur (Compiler compliance level) correspond à la version de Java utilisée pour la génération de code des Jobs.

Pour plus d’informations concernant la compatibilité des niveaux de conformité du compilateur, consultez le Guide d’installation et de migration Talend.

Procédure

2. Dans l’arborescence, développez le nœud Build et cliquez sur Java Version.
3. Dans la liste JDK Compiler compliance level, sélectionnez le niveau de conformité du compilateur que vous souhaitez utiliser puis cliquez sur Apply and Close.
Personnaliser des modèles de commandes Shell

Pourquoi et quand exécuter cette tâche
Votre Studio Talend fournit un ensemble de modèles pour les commandes Shell utilisées pour exécuter des Jobs. Vous pouvez personnaliser ces modèles selon vos besoins.

Procédure

2. Développez les nœuds Build > Shell Setting et cliquez sur Bat, Ps1 ou Sh, selon le système d’exploitation sur lequel vos Jobs s’exécuteront.
3. Modifiez le code dans la vue correspondante selon vos besoins, puis cliquez sur Apply and Close pour valider la personnalisation.
Personnaliser les modèles de scripts de build Maven

Votre Studio Talend fournit les modèles suivants par défaut pour générer des scripts de build :

- des paramètres de build d’images Docker
- des modèles de script Maven pour un export en tant que Job stand-alone
- un modèle de script Maven pour l’export d’un bundle OSGI de Jobs
- des modèles de script Maven pour l’export Karaf de Routes
- des modèles de script Maven pour l’export Karaf de Services

En vous basant sur les modèles de build globaux, par défaut, vous pouvez créer des scripts au niveau des dossiers. Les scripts de build générés, basés sur ces modèles, sont exécutés lors de la construction de Jobs, Routes ou Services.

Cette section fournit des informations concernant la personnalisation des modèles de scripts de build.

Personnaliser les modèles de scripts de build globaux

Pourquoi et quand exécuter cette tâche

Dans la boîte de dialogue Project Settings, vous pouvez voir et personnaliser les modèles de scripts de build globaux, sous le nœud Build > Maven > Default. Ces modèles de scripts s’appliquent à tous les Jobs ou Routes dans le dossier racine et tous les sous-dossiers, exceptés ceux ayant leurs propres modèles de scripts de build configurés.

L’exemple suivant vous montre comment personnaliser le modèle de script POM global pour les Jobs stand-alone :

Procédure

1. Dans le menu, cliquez sur File > Edit Project properties pour ouvrir la boîte de dialogue Project Settings.
2. Développez les nœuds Build > Maven > Default puis cliquez sur le nœud Standalone Job pour ouvrir la vue correspondante affichant le contenu du modèle de script POM.
Remarque:
Selon le Studio que vous utilisez, les éléments des paramètres de projet de votre Studio peuvent être différents de ce qui vous est présenté ici.

3. Modifiez le code du script dans le panneau de texte et cliquez sur Apply and Close afin de terminer votre personnalisation.

Personnaliser les modèles de scripts de build de niveau dossier

Pourquoi et quand exécuter cette tâche

En vous basant sur les modèles de scripts de build globaux, vous pouvez ajouter et personnaliser les modèles de scripts pour les Jobs dossier par dossier, sous le nœud Build > Maven > Setup custom scripts by folder. Les modèles de scripts de build ajoutés dans un dossier s'appliquent à tous les Jobs de ce dossier et ses sous-dossiers, sauf ceux possédant leurs propres modèles de scripts de build configurés.

L'exemple suivant explique comment ajouter et personnaliser le modèle de script POM pour la construction de Jobs standalone depuis les Jobs du dossier CA_customers :

Procédure

1. Dans le menu, cliquez sur File > Edit Project properties pour ouvrir la boîte de dialogue Project Settings.
2. Développez les nœuds Build > Maven > Setup custom scripts by folder > Job Designs > CA_customers puis cliquez sur le nœud Standalone Job pour ouvrir la vue correspondante, de laquelle vous pouvez ajouter des modèles de scripts ou supprimer tous les modèles existants.
Remarque:
Selon le Studio que vous utilisez, les éléments des paramètres de projet de votre Studio peuvent être différents de ce qui vous est présenté ici.

3. Cliquez sur le bouton **Create Maven files** afin de créer des modèles de scripts basés sur les modèles globaux pour les Jobs standalone.

4. Sélectionnez le modèle de script à personnaliser, `pom.xml` dans cet exemple, pour afficher le code du script dans la vue du code. Modifiez le code du script dans le panneau de texte et cliquez sur **Apply and Close** afin de terminer votre personnalisation.
Une fois les modèles de scripts de build créés pour un dossier, vous pouvez également aller dans le répertoire où sont stockés les fichiers XML, `<studio_installation_directory>\workspace\<project_name>\process\CA_customers` dans cet exemple et modifier directement le fichier XML du modèle à personnaliser. Vos modifications affectent tous les Jobs du dossier et des sous-dossiers, sauf ceux possédant leurs propres scripts configurés.

**Gestion des versions du déploiement des Jobs**

**Pourquoi et quand exécuter cette tâche**

Via la boîte de dialogue **Project Settings**, vous pouvez gérer individuellement ou en batch la version de déploiement de chaque élément de chaque Job à publier dans le référentiel d’artefacts.

**Procédure**

1. Cliquez sur l'icône de dossier dans la barre d’outils de la fenêtre du Studio ou cliquez sur **File > Edit Project Properties** dans la barre du menu pour ouvrir la boîte de dialogue **Project Settings**.
2. Dans l’arborescence de la boîte de dialogue, développez **Build > Maven** et sélectionnez **Deployment Versioning** pour ouvrir la vue correspondante.
3. Dans l’arborescence Repository, développez le nœud contenant les éléments dont vous souhaitez gérer les versions de déploiement et cochez les cases de ces éléments. Les éléments cochés sont affichés dans la liste Items à droite, à côté de leur version actuelle dans la colonne Version.

4. Effectuez les modifications nécessaires :
   - Pour définir une version de déploiement pour tous les éléments, saisissez la version dans le champ Project version, sélectionnez l’option Apply project version to items et cliquez sur Apply version.
   - Cliquez sur Select all subjobs si vous souhaitez mettre à jour tous les sous-Jobs dépendants des éléments sélectionnés au même moment.
   - Pour définir individuellement la version du déploiement pour un ou plusieurs éléments, sélectionnez l’option Change the version of each item individually, sélectionnez l’élément ou les éléments dans la table sous la zone Options, saisissez la version du déploiement dans le champ New version et cliquez sur Apply.
   - Sélectionnez l’option Use job versions si vous souhaitez utiliser les dernières versions du Job ou du Service en tant que versions de déploiement des éléments sélectionnés.
   - Si vous souhaitez publier une version Snapshot de l’élément ou des éléments, sélectionnez la case Use snapshot avant d’appliquer les paramètres de la version.

5. Cliquez sur Apply and Close pour appliquer vos modifications et fermer la boîte de dialogue.


Vous pouvez également configurer la version de déploiement d’un Job dans la vue Job, une fois ouvert dans l’espace de modélisation graphique. Pour plus d’informations, consultez Personnalisation du déploiement d’un Job.

Les paramètres de la version de déploiement que vous configurés dans la boîte de dialogue Project Settings sont reflétés dans l’onglet Deployment de la vue Job de l’élément ou des éléments concernés, et vice-versa.
**Paramètres de la Palette**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez personnaliser l’affichage de la Palette, afin de ne charger que les composants que vous utilisez dans votre Projet, ce qui permet de lancer votre Studio plus rapidement. Ce qui permet de lancer votre Studio plus rapidement.

Pour personnaliser l’affichage de la Palette, procédez comme suit :

**Procédure**

1. Cliquez sur 🔄 dans la barre d’outils de la fenêtre du Studio ou cliquez sur **File > Edit Project Properties** dans la barre du menu pour ouvrir la boîte de dialogue **Project Settings**.

   **Remarque:**
   Dans la vue **General** de la boîte de dialogue **Project Settings**, ajoutez une description du projet en cours si vous ne l’avez pas fait lors de sa création.

2. Dans l’arborescence de la fenêtre **Project Settings**, développez le nœud **Designer** et cliquez sur **Palette Settings**. Les paramètres de la Palette en cours sont affichés dans la partie droite de la fenêtre.

3. Sélectionnez un ou plusieurs composants, ou un ou plusieurs groupes de composants pour les supprimer de la Palette en cours.

4. Puis utilisez la flèche vers la gauche pour déplacer la sélection vers la Palette de composants cachés, à gauche de la fenêtre. Cela supprime les composants sélectionnés de la Palette.

5. Pour afficher de nouveau un composant caché, sélectionnez-le dans la zone de gauche et faites-la passer dans la Palette en cliquant sur la flèche vers la droite.

6. Cliquez sur **Apply** pour valider vos modifications puis sur **Apply and Close** pour fermer la boîte de dialogue.

**Résultats**

**Remarque:**
Pour revenir aux paramètres par défaut de la Palette, cliquez sur le bouton **Restore Defaults**.

Pour plus d’informations concernant la Palette, consultez Modifier la disposition et les paramètres de la Palette à la page 489.

**Mapping de types**

Vous pouvez configurer les paramètres pour la conversion de types dans le Studio Talend, de Java vers des bases de données et vice versa.

**Accéder aux fichiers de mapping et définir les mappings de types**

**Procédure**

1. Cliquez sur 🔄 dans la barre d’outils de la fenêtre du Studio ou cliquez sur **File > Edit Project Properties** dans la barre du menu pour ouvrir la boîte de dialogue **Project Settings**.
2. Dans l’arborescence de la fenêtre, développez **General** et sélectionnez **Metadata of Talend Type** pour ouvrir la vue associée.

La zone **Metadata Mapping File** liste les fichiers XML contenant les paramètres de conversion pour chaque type de base de données utilisé dans le Studio Talend.

Vous pouvez importer, exporter ou supprimer un fichier de conversion en cliquant sur **Import**, **Export** ou **Remove**, respectivement.

Vous pouvez modifier un fichier de conversion selon vos besoins en double-cliquant sur le fichier ou en sélectionnant le fichier et en cliquant sur le bouton **Edit** pour ouvrir la boîte de dialogue **Edit Mapping File**, puis en modifiant le code XML directement dans la boîte de dialogue.

Lorsque vous définissez un mapping de types, vous devez mapper depuis le type Talend vers le type de base de données et depuis le type de base de données vers le type Talend.

- L’élément `<dbTypes>` et ses éléments enfants `<dbType>` définissent les types de bases de données supportés. Pour ajouter un type de base de données dans le fichier de mapping, vous devez ajouter un élément `<dbType>` sous l’élément `<dbTypes>`. L’exemple ci-dessous ajoute deux types de bases de données BOOLEAN et YESNO.

```xml
<dbType type="BOOLEAN" ignoreLen="true" ignorePre="true"/>
<dbType type="YESNO" ignoreLen="true" ignorePre="true"/>
```

- L’élément `<talendToDbType>` et ses éléments enfants `<talendType>` définissent une liste de types de bases de données suggérés et le type de base de données par défaut lors de la configuration d’un type Talend pour une colonne de métadonnées. Pour mapper un type Talend vers un ou plusieurs type(s) de base(s) de données, vous devez ajouter un élément `talendType` sous l’élément `<talendToDbType>`. L’exemple ci-dessous mappe le type Talend `id_Boolean` à deux types de bases de données BOOLEAN et YESNO.

```xml
<talendType type="id_Boolean">
  <dbType type="YESNO" default="true"/>
  <dbType type="BOOLEAN"/>
</talendType>
```
L'élément `<dbToTalendTypes>` et ses éléments enfants `<dbType>` définissent une liste de types Talend suggérés et le type Talend par défaut lors de la récupération du schéma depuis la base de données. Pour mapper un ou plusieurs type(s) Talend de bases de données, vous devez ajouter un élément `dbType` sous l'élément `<dbToTalendTypes>`. L'exemple ci-dessous mappe le type de base de données `YESNO` au type Talend `id_Boolean`.

```xml
<dbType type="YESNO">
  <talendType type="id_Boolean" default="true" />
</dbType>
```
Ci-dessous se trouve le fichier de mapping de métadonnées XML pour la base de données Access :

```xml
<?xml version="1.0"?>
<mapping>
<dbms product="ACCESS" id="access_id" label="Mapping Access" default="true">
<dbTypes>
  <dbType type="BIT" ignoreLen="true" ignorePre="true"/>
  <dbType type="BOOLEAN" ignoreLen="true" ignorePre="true"/>
  <dbType type="COUNTER"/>
  <dbType type="DATE" ignoreLen="true" ignorePre="true"/>
  <dbType type="DOUBLE" ignoreLen="true" ignorePre="true"/>
  <dbType type="DECIMAL" ignoreLen="true" ignorePre="true"/>
  <dbType type="FLOAT" ignoreLen="true" ignorePre="true"/>
  <dbType type="INTEGER" ignoreLen="true" ignorePre="true"/>
  <dbType type="NUMERIC" ignoreLen="true" ignorePre="true"/>
  <dbType type="REAL" ignoreLen="true" ignorePre="true"/>
  <dbType type="SMALLINT" ignoreLen="true" ignorePre="true"/>
  <dbType type="TINYINT" ignoreLen="true" ignorePre="true"/>
  <dbType type="TIME" ignoreLen="true" ignorePre="true"/>
  <dbType type="TIMESTAMP" ignoreLen="true" ignorePre="true"/>
  <dbType type="VARCHAR" default="true" defaultLength="200" ignorePre="true"/>
  <dbType type="DATETIME" ignoreLen="true" ignorePre="true"/>
  <dbType type="MEMO" ignoreLen="true" ignorePre="true"/>
  <dbType type="YESNO" ignoreLen="true" ignorePre="true"/>
</dbTypes>

<language name="java">

talenToDbTypes
  </talendToDbTypes>
</language>
</mapping>
```
<dbType type="TINYINT"/>
<dbType type="COUNTER"/>
</talendType>
	<talendType type="id_Long">
		<dbType type="INTEGER" default="true"/>
		<dbType type="SMALLINT"/>
		<dbType type="TINYINT"/>
		<dbType type="COUNTER"/>
	</talendType>
	<talendType type="id_Object" />
	<talendType type="id_Short">
		<dbType type="SMALLINT" default="true"/>
		<dbType type="INTEGER"/>
		<dbType type="TINYINT"/>
		<dbType type="COUNTER"/>
	</talendType>
	<talendType type="id_String">
		<dbType type="VARCHAR" default="true"/>
		<dbType type="MEMO"/>
	</talendType>
</talendToDbTypes>
<dbToTalendTypes>
<dbType type="BIT">
	<talendType type="id_Boolean" default="true"/>
</dbType>
<dbType type="BOOLEAN">
	<talendType type="id_Boolean" default="true"/>
</dbType>
<dbType type="COUNTER">
	<talendType type="id_Integer" default="true"/>
</dbType>
<dbType type="DATE">
	<talendType type="id_Date" default="true"/>
</dbType>
<dbType type="DECIMAL">
	<talendType type="id_Double"/>
	<talendType type="id_BigDecimal" default="true"/>
	<talendType type="id_Float"/>
</dbType>
<dbType type="DOUBLE">
	<talendType type="id_Double" default="true"/>
	<talendType type="id_BigDecimal"/>
	<talendType type="id_Float"/>
</dbType>
<dbType type="FLOAT">
	<talendType type="id_Float" default="true"/>
	<talendType type="id_BigDecimal"/>
	<talendType type="id_Double"/>
</dbType>
<dbType type="INTEGER">
	<talendType type="id_Integer" default="true"/>
	<talendType type="id_Short"/>
	<talendType type="id_Long"/>
	<talendType type="id_Byte"/>
</dbType>
<dbType type="NUMERIC">
	<talendType type="id_Float"/>
	<talendType type="id_BigDecimal" default="true"/>
	<talendType type="id_Double"/>
</dbType>
<dbType type="REAL">
	<talendType type="id_Float" default="true"/>
	<talendType type="id_BigDecimal"/>
	<talendType type="id_Double"/>
</dbType>
<dbType type="SMALLINT">
	<talendType type="id_Short" default="true"/>
	<talendType type="id_Integer"/>
	<talendType type="id_Long"/>
	<talendType type="id_Byte"/>
</dbType>
<dbType type="TINYINT">
	<talendType type="id_Byte" default="true"/>
	<talendType type="id_Integer"/>
	<talendType type="id_Short"/>
	<talendType type="id_Long"/>
Types Talend supportés
Talend supporte les types suivants.

Remarque: Vous ne pouvez définir des types personnalisés dans Studio Talend.

<table>
<thead>
<tr>
<th>Type Talend</th>
<th>Type Java</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_Boolean</td>
<td>java.lang.Boolean</td>
</tr>
<tr>
<td>id_Byte</td>
<td>java.lang.Byte</td>
</tr>
<tr>
<td>id_byte[]</td>
<td>byte[]</td>
</tr>
<tr>
<td>id_Character</td>
<td>java.lang.Character</td>
</tr>
<tr>
<td>id_Date</td>
<td>java.util.Date</td>
</tr>
<tr>
<td>id_Double</td>
<td>java.lang.Double</td>
</tr>
<tr>
<td>id_Float</td>
<td>java.lang.Float</td>
</tr>
<tr>
<td>id_BigDecimal</td>
<td>java.math.BigDecimal</td>
</tr>
<tr>
<td>id_Integer</td>
<td>java.lang.Integer</td>
</tr>
<tr>
<td>id_Long</td>
<td>java.lang.Long</td>
</tr>
<tr>
<td>id_Object</td>
<td>java.lang.Object</td>
</tr>
<tr>
<td>id_Short</td>
<td>java.lang.Short</td>
</tr>
<tr>
<td>id_String</td>
<td>java.lang.String</td>
</tr>
<tr>
<td>id_List</td>
<td>java.util.List</td>
</tr>
</tbody>
</table>
**Attributs fréquemment utilisés dans l’éléments `dbType`**

Grâce à l’utilisation d’attributs dans les mappings de conversion, vous pouvez définir les valeurs ou le comportement par défaut des colonnes du schéma. Le tableau ci-dessous décrit les attributs les plus fréquemment utilisés dans l’élément `dbType`.

<table>
<thead>
<tr>
<th>Attribut</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ignoreLen</code></td>
<td>Lorsqu’il est configuré à <code>true</code>, la longueur paramétrée des nouvelles colonnes du type dans le schéma (via <code>Edit schema</code>) dans les composants sera ignorée.</td>
</tr>
<tr>
<td><code>ignorePre</code></td>
<td>Lorsqu’il est configuré à <code>true</code>, la précision configurée sur les nouvelles colonnes du type dans le schéma (via <code>Edit schema</code>) dans les composants sera ignorée.</td>
</tr>
<tr>
<td><code>Default</code></td>
<td>Lorsqu’il est configuré à <code>true</code>, le type défini dans cet élément sera le type par défaut des nouvelles colonnes ajoutées (via <code>Edit schema</code>) dans les composants.</td>
</tr>
<tr>
<td><code>defaultLength</code></td>
<td>Configure la longueur par défaut des nouvelles colonnes ajoutées du type (via <code>Edit schema</code>) dans les composants.</td>
</tr>
<tr>
<td><code>defaultPrecision</code></td>
<td>Configure la précision par défaut des nouvelles colonnes ajoutées du type (via <code>Edit schema</code>) dans les composants.</td>
</tr>
</tbody>
</table>

**Gestion des versions**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez aussi gérer les versions des éléments du `Repository` à partir de la fenêtre `General > Version Management` des paramètres du projet (`Project Settings`).

Pour cela :

**Procédure**

1. Cliquez sur ![icon](image) dans la barre d’outils de la fenêtre du Studio ou cliquez sur `File > Edit Project Properties` dans la barre du menu pour ouvrir la boîte de dialogue `Project Settings`.
3. Dans l’arborescence Repository, développez le nœud correspondant aux éléments dont vous souhaitez gérer les versions et cochez les cases correspondantes.


4. Effectuez les modifications nécessaires :
   - Dans la zone Option, sélectionnez l’option Change all items to a fixed version pour passer tous les éléments sélectionnés à la même version fixe.
   - Cliquez sur Revert si vous souhaitez annuler ces modifications.
   - Cliquez sur Select all dependencies si vous souhaitez mettre à jour tous les éléments dépendants des éléments sélectionnés à la fois.
   - Cliquez sur Select all subJobs si vous souhaitez mettre à jour tous les sous-Jobs dépendants des éléments sélectionnés au même moment.
   - Pour incrémenter individuellement la version de chaque élément du référentiel, sélectionnez l’option Update the version of each item et modifiez la version manuellement.
   - Cochez la case Fix tRunjob versions if Latest si vous souhaitez que le Job parent conserve le Job enfant de la version actuelle dans le tRunJob à versionner, sans tenir compte de la mise à jour de leurs versions. Par exemple, un tRunJob fera une mise à jour à partir de la version actuelle 1.0 vers 1.1 à la fois au niveau du Job parent et du Job enfant. Une fois cette case cochée, le Job parent 1.0 continuera à utiliser le Job enfant 1.0 plutôt que la dernière version (la version 1.1), comme il le ferait habituellement, lorsque la mise à jour est faite.

**Avertissement:** Pour utiliser cette case, le Job parent doit utiliser le(s) Job(s) enfant(s) de la dernière version en tant que version actuelle dans le tRunJob à versionner, ce qui est possible en sélectionnant l’option Latest de la liste déroulante des versions dans la vue Component du (des) Job(s) enfant(s).

5. Cliquez sur Apply and Close pour appliquer vos modifications et fermer la boîte de dialogue.
Gestion du statut

Pourquoi et quand exécuter cette tâche

Vous pouvez également gérer le statut de chaque élément dans la vue Repository via le chemin General > Status Management de la boîte de dialogue Project Settings.

Pour cela :

Procédure

2. Dans l’arborescence de la boîte de dialogue, développez le nœud General puis sélectionnez Status Management afin d’ouvrir la vue correspondante.
3. Dans la vue Repository, développez le nœud contenant les éléments dont vous souhaitez gérer le statut et cochez la case de ces éléments. Les éléments sélectionnés s’affichent dans la liste Items de droite, avec leur statut, dans la colonne Status. Le nouveau statut défini s’affiche dans la colonne New Status.
4. Dans la zone Options, cochez la case Change all technical items to a fixed status pour changer le statut de tous les éléments sélectionnés en un même statut fixe.
5. Cliquez sur Revert si vous souhaitez annuler les modifications.
6. Pour mettre à jour le statut des éléments, sélectionnez l’option Update the version of each item et changez-le manuellement.
7. Cliquez sur Apply and Close pour appliquer vos modifications et fermer la boîte de dialogue.

Résultats

Remarque:

Pour plus d’informations concernant les statuts des Jobs, consultez Définir les paramètres des Statuts à la page 487.
Annexes

Paramètres du Job

Pourquoi et quand exécuter cette tâche

Vous pouvez automatiquement utiliser les paramètres Implicit Context Load et Stats and Logs définis dans les Project Settings du projet en cours lorsque vous créez un nouveau Job.

Pour cela :

Procédure


2. Dans l’arborescence de la boîte de dialogue, cliquez sur le nœud Job Settings pour ouvrir la vue correspondante.

3. Cochez la case Use project settings when create a new job des zones Implicit Context Load et Stats and Logs.

4. Cliquez sur Apply pour valider vos modifications puis sur Apply and Close pour fermer la boîte de dialogue.

Stats et Logs

Pourquoi et quand exécuter cette tâche

Lorsque vous exécutez un Job, vous avez la possibilité de moniter son exécution à l’aide de l’option tStatCatcher Statistics ou des composants de log adéquats. Vous pouvez ensuite collecter les données recueillies dans des fichiers CSV ou dans une base de données.

Vous pouvez définir le chemin d’accès à ces fichiers et/ou cette base de données de log de votre projet de manière permanente dans la boîte de dialogue Project Settings, afin que les données de log soient toujours stockées dans le même répertoire.

Pour cela :

482
Procédure


2. Dans l’arborescence de la boîte de dialogue, développez le nœud Job Settings et cliquez sur Stats & Logs pour afficher la vue correspondante.

3. Cochez les cases Use statistics, Use logs et Use volumetrics en fonction de vos besoins, puis renseignez le chemin d’accès des données de log.

4. Sélectionnez un format pour le stockage des données de log : cochez la case On Files ou On Database, ou cochez la case On Console pour afficher les données dans la console.

Résultats

Les champs apparaissent en fonction des paramètres sélectionnés. Dans les champs File Name ou DB Name, saisissez respectivement le nom du fichier ou de la base de données entre guillemets en fonction du type de format choisi.
Notez que vous pouvez maintenant stocker les informations de connexion à la base de données dans le Repository. Pour cela, sélectionnez Repository dans la liste déroulante Repository Type et cliquez sur le bouton [...] pour stocker ces informations dans la métadonnée correspondante. Les champs suivants sont renseignés automatiquement.

**Remarque:**
Si vous avez sauvé les informations de connexion dans une variable de contexte, vous pouvez y accéder via le raccourci Ctrl+Espace.

**Définir les paramètres de contexte**

**Pourquoi et quand exécuter cette tâche**
Dans les paramètres du projet, vous pouvez définir des paramètres de contexte automatisques utilisés par défaut dans vos Jobs.

Pour cela :

**Procédure**

2. Dans l’arborescence de la boîte de dialogue, développez le nœud Job Settings et cochez la case Implicit tContextLoad pour afficher les paramètres de configuration de l’option Implicit tContextLoad.

![Implicit context load](image)

3. Sélectionnez le type de fichier dans lequel les données de contextes seront conservées, soit sous forme de fichier en sélectionnant le champ From File, soit sous forme de base de données en sélectionnant From Database.
4. Pour un fichier, renseignez le chemin d’accès et le séparateur de champ du fichier contenant les paramètres de contexte dans les champs From File et Field Separator.
5. Pour une base de données, sélectionnez le mode adéquat, Built-in ou Repository, dans le champ Property Type et renseignez les champs suivants.
6. Dans les champs Table Name et Query Condition, renseignez le nom de la table contenant les paramètres de contexte et la requête à utiliser.
7. Sélectionnez le type de message système à générer (avertissement, erreur ou info), si une variable est chargée mais n’est pas dans le contexte, ou vice versa.

8. Cliquez sur **Apply and Close** pour appliquer vos modifications et fermer la boîte de dialogue.

**Appliquer les paramètres du projet**

**Pourquoi et quand exécuter cette tâche**

Dans les paramètres du projet (**Project Settings**), vous pouvez définir à quels Jobs du **Repository** vous souhaitez appliquer les paramètres **Implicit Context Load** et **Stats and Logs**.

Pour cela :

**Procédure**

1. Cliquez sur **Éditer les paramètres du projet** dans la barre d’outils de la fenêtre du Studio ou cliquez sur **Fichier > Éditer les paramètres du projet** dans la barre du menu pour ouvrir la boîte de dialogue **Project Settings**.

2. Dans l’arborescence de la boîte de dialogue, développez le nœud **Job Settings** et cliquez sur **Use Project Settings** pour afficher l’utilisation des options **Implicit Context Load** et **Stats and Logs** dans les Jobs.


4. Dans la zone **Stats Logs Settings**, cochez les cases correspondant aux Jobs auxquels vous souhaitez appliquer l’option Stats and Logs.

5. Cliquez sur **Apply and Close** pour appliquer vos modifications et fermer la boîte de dialogue.
Configurer Log4j

Le Studio Talend inclut Log4j, l'utilitaire de log d'Apache. Cet utilitaire fournit des informations de log à l'exécution. Vous pouvez activer ou désactiver Log4j pour les composants et personnaliser sa configuration sur tout le projet.

Procédure

2. Dans l’arborescence de la boîte de dialogue, cliquez sur le nœud Log4j afin d’ouvrir la vue Log4j.
3. Cochez la case Activate log4j in components pour activer la fonctionnalité Log4j.

Par défaut, la fonction log4j est désactivée lorsqu’un projet est créé.

Lorsqu’un projet est importé à partir d’une version Talend 7.2 ou précédente, qui supporte uniquement Log4j 1, une liste déroulante Log4j version est affichée et Log4j déprécié est sélectionné par défaut. Tous les fichiers POM pour le projet sont synchronisés lorsque vous modifiez la version de Log4j.

4. Modifiez la configuration globale de Log4j en modifiant les instructions XML dans la zone Log4j template.

Par exemple, pour configurer le log racine de Log4j 2, pour écrire tous les messages de niveau debug ou supérieur, allez dans la zone Loggers et configurez la valeur de l’attribut level du nœud root à debug.

Pour plus d’informations concernant la configuration de Log4j 2, consultez http://logging.apache.org/log4j/2.x/manual/configuration.html#XML.

Pour plus d’informations concernant la configuration de Log4j 1, consultez https://cwiki.apache.org/confluence/display/LOGGINGLOG4J/Log4jXmlFormat.
Définir les paramètres des Statuts

Pourquoi et quand exécuter cette tâche
Dans les paramètres du projet (Project Settings), vous pouvez définir des niveaux de statuts.
Pour cela :

Procédure

2. Dans l’arborescence, cliquez sur le nœud Status pour paramétrer les propriétés principales des éléments du Repository.
   Les propriétés principales d’un élément du repository comprend des informations telles que Name, Purpose, Description, Author, Version and Status de l’élément sélectionné. La plupart des propriétés sont des champs de saisie, cependant le champ Status est une liste déroulante.

3. Cliquez sur le bouton New... pour afficher une boîte de dialogue et alimenter la liste Status avec les valeurs appropriées aux besoins de votre entreprise. Notez que le champ Code ne peut pas dépasser trois caractères et le champ Label contenant le libellé de votre statut est obligatoire.
**Talend** fait la différence entre deux types de statut : **Technical status** et **Documentation status**.

La liste de statuts **Technical status** affiche les codes de classification des éléments qui sont utilisés lors de l’exécution de Jobs, de définition de métadonnées ou de routines.

La liste de statuts **Documentation status** permet de classer les éléments du référentiel qui sont utilisés pour documenter les process. Cette liste de statuts n’est disponible que pour les Business Models et la Documentation.

4. Une fois le statut renseigné, cliquez sur **OK** pour sauvegarder.

Désormais la liste **Status** vous permet d’appliquer vos paramètres de classification personnalisés aux Jobs et aux Business Models du référentiel.

5. Dans la boîte de dialogue **Project Settings**, cliquez sur **Apply** afin de valider vos modifications, puis sur **Apply and Close** pour fermer la boîte de dialogue.

### Paramètres de sécurité

**Pourquoi et quand exécuter cette tâche**

Dans les paramètres de projet, vous pouvez afficher ou masquer les mots de passe de vos documentations, métadonnées, contextes, etc. lorsqu’ils sont centralisés dans le **Repository**.

Pour masquer votre mot de passe :

**Procédure**

1. Cliquez sur dans la barre d’outils de la fenêtre du Studio ou cliquez sur File > Edit Project Properties dans la barre du menu pour ouvrir la boîte de dialogue **Project Settings**.

2. Dans l’arborescence, cliquez sur le nœud **Security** pour ouvrir la vue correspondante.

3. Cochez la case **Hide passwords** pour masquer vos mots de passe.

### Remarque:

Si vous cochez la case **Hide passwords**, vos mots de passe seront masqués dans tous vos contextes, documentations, etc., ainsi que dans les propriétés de vos composants si vous avez sélectionné l’option **Repository** dans le champ **Property Type** de l’onglet **Basic settings** de la vue Component comme illustré dans la capture d’écran ci-dessous. Si vous sélectionnez l’option **Built-in**, le mot de passe ne sera pas masqué.

4. Dans la boîte de dialogue **Project Settings**, cliquez sur **Apply** afin de valider vos modifications, puis sur **Apply and Close** pour fermer la boîte de dialogue.

### Personnaliser l’espace de travail

Lorsque vous utilisez le Studio Talend pour créer un Job d’intégration de données, vous pouvez personnaliser la disposition et les paramètres de la **Palette** selon vos besoins. Vous pouvez également modifier la position de tous les onglets existants dans le Studio afin de répondre à vos besoins.
Remarque:
Tous les panneaux, onglets et vues décrites dans cette documentation sont spécifiques au Studio Talend. Certaines vues listées dans la boîte de dialogue Show view sont spécifiques à Eclipse et ne concernent pas cette documentation. Pour plus d'informations concernant ces vues, veuillez consulter la documentation Eclipse sur http://www.eclipse.org/documentation/

Modifier la disposition et les paramètres de la Palette

La Palette contient tous les composants techniques, les formes ainsi que les branches de base nécessaires à la création de Jobs et de Business Models dans l'espace de modélisation graphique. Ces composants, ces formes ainsi que ces branches sont regroupés par familles et sous-familles.

Le Studio Talend vous permet de changer la disposition et la position de votre Palette selon vos besoins. Les sections suivantes expliquent toutes les options de gestion disponibles pour la Palette.

Afficher, cacher et déplacer la Palette

Par défaut, la Palette peut être cachée sur le côté droit de l'espace de modélisation.

Si vous souhaitez que la Palette soit visible en permanence, cliquez sur la flèche de gauche dans le coin supérieur droit de l'éditeur graphique.

Vous pouvez aussi détacher la Palette de l'espace de modélisation dans la perspective Integration. Pour activer la vue autonome de la Palette, cliquez successivement dans le menu sur Window > Show View… > General > Palette.

Si vous souhaitez détacher la Palette, cliquez-droit sur la barre Palette et sélectionnez Detached dans le menu contextuel. La Palette s'ouvrira alors dans une vue séparée qui peut être déplacée dans la perspective.

Afficher/masquer les familles de composants

Vous pouvez masquer ou afficher les familles de composants selon vos besoins, dans un souci de visibilité, par exemple. Pour ce faire, cliquez-droit sur la Palette et sélectionnez l’option Display folder pour afficher les dossiers et Hide folder pour masquer les dossiers.
Remarque:

L’option display/hide (afficher/masquer) peut être très utile lorsque vous êtes dans la vue Favorite de la Palette. Pour plus d’informations concernant la Palette favorite, consultez Configurer la Palette favorite à la page 490.

Maintenir ouverte une famille de composants

Si vous utilisez souvent une ou plusieurs famille(s) de composants, vous pouvez ajouter une punaise sur leur nom pour les empêcher de se réduire lorsque vous sélectionnez des composants d’autres familles.

Pour ajouter une punaise, cliquez sur l’icône de punaise dans l’angle en haut à droite du nom de la famille.

Filtrer la Palette

Vous pouvez sélectionner les composants à afficher ou à masquer dans votre Palette. Vous pouvez également ajouter à la Palette les composants que vous avez développé vous-même.

Pour plus d’informations concernant comment filtrer la Palette, consultez Paramètres de la Palette à la page 473.

Pour plus d’informations concernant l’ajout de composants à la Palette, à partir de Talend ou développés par vous-même, consultez Télécharger/charger des composants de la communauté à la page 71 et/ou Définir le dossier de composants utilisateur (Talend > Components) à la page 506.

Configurer la Palette favorite

Pourquoi et quand exécuter cette tâche

La Palette offre des fonctionnalités de recherche et de favoris facilitant son utilisation.

Vous pouvez ajouter et retirer des composants à votre Palette favorite, afin d’accéder plus rapidement aux composants que vous utilisez le plus souvent.

Pour cela :
Annexes

Procédure

1. Dans la Palette, cliquez-droit sur le composant que vous souhaitez ajouter à vos favoris de la Palette et sélectionnez l’option Add To Favorite.

2. Répétez cette action pour tous les composants que vous souhaitez ajouter à votre Palette favorite, puis cliquez sur le bouton Favorite ⭐ dans le coin en haut à droite de la Palette pour afficher la Palette favorite.

Seuls les composants ajoutés aux Favoris apparaissent.

Pour retirer un composant de la Palette favorite, cliquez-droit sur le composant et sélectionnez Remove From Favorite.
Pour retourner à la Palette standard, cliquez sur le bouton Standard en haut de la Palette.

**Modifier la disposition des composants dans la Palette**

Vous pouvez modifier la disposition de la liste des composants dans la Palette pour les afficher en colonnes ou en listes, avec seulement une icône ou une icône avec une courte description.

Vous pouvez également agrandir l’icône des composants pour une meilleure lisibilité de la liste des composants.

Pour ce faire, cliquez-droit sur une famille de composants dans la Palette et sélectionnez l’option désirée dans le menu contextuel ou cliquez sur Settings pour ouvrir la fenêtre Palette Settings et personnaliser la disposition.

**Réorganiser les onglets d’un Job**

Vous pouvez déplacer tous les onglets selon vos besoins.

**Procédure**

- Cliquez sur la bordure ou sur l’onglet, maintenez le bouton de la souris enfoncé pendant que vous déplacez la fenêtre vers l’emplacement cible, puis relâchez.
- Cliquez sur l’icône minimiser/maximiser pour réduire ou agrandir le panneau correspondant.
Pour plus d'informations concernant comment afficher ou masquer un panneau ou une vue, consultez **Afficher les onglets/vues de configuration des Jobs** à la page 493.

- Cliquez sur la croix (❌) pour fermer une vue/un onglet. Pour restaurer une vue, cliquez sur **Window > Show View... > Talend**, puis cliquez sur le nom de la vue que vous souhaitez afficher.
- Si la **Palette** n'est pas visible ou si vous souhaitez la détacher, accédez à **Window > Show view... > General > Palette**.
  La **Palette** s'ouvrira alors dans une vue séparée qui peut être déplacée dans la perspective.

### Afficher les onglets/vues de configuration des Jobs

Les onglets de configuration sont situés dans la partie inférieure de l'espace de modélisation graphique de la perspective **Integration**. Chaque onglet ouvre une vue affichant des informations précises concernant l’élément sélectionné dans l’espace de modélisation graphique.

Les vues **Component**, **Run Job** et **Contexts** rassemblent toutes les informations relatives aux éléments graphiques sélectionnés dans l’espace de modélisation graphique et à l’exécution du Job ouvert.

<table>
<thead>
<tr>
<th>Remarque:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Par défaut, lorsque vous lancez le Studio Talend pour la première fois, l’onglet <strong>Problems</strong> ne s’affichera qu’au moment où vous créerez votre premier Job. L’onglet <strong>Problems</strong> s’affichera automatiquement.</td>
</tr>
</tbody>
</table>

Les onglets **Modules** et **Scheduler[deprecated]** se trouvent dans la même zone que les onglets **Component**, **Logs** et **Run Job**. Ces deux vues sont indépendantes des Jobs, actifs ou inactifs, ouverts dans l’espace de modélisation.

Certains onglets de configuration sont masqués par défaut, comme les onglets **Error Log**, **Navigator**, **Job Hierarchy**, **Problems**, **Modules** et **Scheduler[deprecated]**. Vous pouvez afficher les onglets masqués dans la même zone que les autres, et ouvrir directement la vue correspondante si vous...
sélectionnez **Window > Show view**, puis, dans la boîte de dialogue, développez le nœud correspondant et sélectionnez l’élément que vous souhaitez afficher.

**Filtrage des entrées listées dans la vue Repository**

Le Studio Talend offre la possibilité de choisir les nœuds, les Jobs ou éléments que vous souhaitez lister dans la vue **Repository**.

Vous pouvez filtrer la vue **Repository** par le nom du Job, le statut du Job ou l’utilisateur ayant créé le Job ou les éléments, simplement en cochant ou décochant la case située à côté du nœud ou de l’élément que vous souhaitez afficher ou cacher dans la vue. Vous pouvez aussi définir simultanément plusieurs filtres.

**Filtrer avec le nom du Job**

**Pourquoi et quand exécuter cette tâche**

Pour filtrer les Jobs listés dans la vue **Repository** par le nom du Job, procédez comme suit :

**Procédure**

1. Dans le Studio, cliquez sur dans le coin supérieur droit de la vue **Repository** et sélectionnez **Filter Setting...** dans le menu contextuel.
   La boîte de dialogue **Repository Filter Setting** s’ouvre.
2. Cochez la case **Filter By Name**.
Le champ correspondant devient accessible.

![Filter By Name](image)

3. Suivez les règles affichées sous le champ lorsque vous saisissez les modèles à utiliser pour filtrer les Jobs.
L'objectif de cet exemple est de lister tous les Jobs de l'arborescence commençant par **tMap** ou **test**.

Seuls les Jobs correspondants au filtre que vous avez défini apparaîtront dans l'arborescence, ceux commençant par **tMap** et **test** dans cet exemple.
Résultats

Remarque:
Vous pouvez retourner à l'arborescence par défaut, listant tous les nœuds, Jobs et éléments, simplement en cliquant sur l'icône. Alors le signe plus vert se change en signe moins rouge.

Filtrer avec l'utilisateur

Pourquoi et quand exécuter cette tâche
Pour filtrer des entrées dans la vue Repository avec l'utilisateur qui a créé les Jobs ou les éléments, procédez comme suit :

Procédure
1. Dans le Studio, cliquez sur dans le coin supérieur droit de la vue Repository et sélectionnez Filter Setting... dans le menu contextuel.
   La boîte de dialogue Repository Filter s'ouvre.
2. Décochez la case **All Users**.
Les champs correspondants dans la table ci-dessous deviennent accessibles.

La table liste les informations authentification de tous les utilisateurs qui se sont connectés au Studio Talend et qui ont créé un Job ou un élément.

3. Décochez la case à côté du nom de l’utilisateur si vous souhaitez cacher dans la vue **Repository** tous les Jobs ou éléments créés par l’utilisateur.

4. Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.
Tous les Jobs ou éléments créés par l’utilisateur sélectionné disparaîtront de l’arborescence.
Remarque:
Vous pouvez retourner à l’arborescence par défaut, listant tous les nœuds, Jobs et éléments, simplement en cliquant sur l’icône . Alors le signe plus vert se change en signe moins rouge ( ).

Filtrer avec le statut du Job

Pourquoi et quand exécuter cette tâche
Pour filtrer les Jobs dans la vue Repository par le statut du Job, procédez comme suit :

Procédure

1. Dans le Studio, cliquez sur dans le coin supérieur droit de la vue Repository et sélectionnez Filter Setting... dans le menu contextuel. La boîte de dialogue Repository Filter s’ouvre.

2. Dans la zone Filter By Status, décochez la case située à côté de statut si vous souhaitez cacher tous les Jobs ayant le statut sélectionné.
3. Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.
   Tous les Jobs ayant le statut sélectionné disparaîtront de l’arborescence.

**Résultats**

<table>
<thead>
<tr>
<th>Remarque:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vous pouvez retourner à l’arborescence par défaut, listant tous les nœuds, Jobs et éléments, simplement en cliquant sur l’icône <img src="image" alt="plus" />. Alors le signe plus vert se change en signe moins rouge <img src="image" alt="moins" />.</td>
</tr>
</tbody>
</table>

**Sélectionner les nœuds du Repository à afficher**

**Pourquoi et quand exécuter cette tâche**

Pour filtrer les nœuds du Repository, procédez comme suit :

**Procédure**

1. Dans la perspective **Integration** du Studio, cliquez sur ![folder](image) dans la vue **Repository** en haut à droite et sélectionnez **Filter settings** à partir du menu contextuel.
   La boîte de dialogue **Repository Filter** s’ouvre.
2. Cochez la case située à côté des nœuds que vous souhaitez afficher dans la vue Repository.
Par exemple, vous souhaitez montrer dans l’arborescence tous les Jobs listés sous le nœud **Job Designs**, et trois des dossiers listés sous le nœud **SQL Templates** et l’un des éléments des métadonnées listé sous le nœud **Metadata**.

3. Cliquez sur **OK** pour valider vos changements et fermer la boîte de dialogue.

   Seuls les nœuds/dossiers dont vous avez coché la case apparaissent dans l’arborescence.

Résultats

**Remarque:**

Si vous ne souhaitez pas afficher tous les Jobs listés sous le nœud **Job Designs**, vous pouvez filtrer les Jobs en cochant la case **Filter By Name**. Pour plus d’informations concernant le filtrage des Jobs, consultez **Filtrer avec le nom du Job** à la page 494.
Configuration des préférences du Studio Talend

Vous pouvez définir plusieurs propriétés d’affichage pour toutes les perspectives du Studio Talend pour les adapter à vos besoins et à vos préférences.

Un grand nombre des configurations que vous avez définies peuvent être enregistrées dans Preferences et seront donc utilisées par défaut pour tous les nouveaux Jobs que vous créerez.

La section suivante décrit les configurations spécifiques que vous pouvez définir en tant que préférences.

Cliquez sur le menu Window du Studio Talend, puis sélectionnez Preferences.

Interpréteur Java (Talend)

Pourquoi et quand exécuter cette tâche

Le chemin de l’interpréteur Java est défini par défaut selon l’emplacement du fichier Java de votre ordinateur (par exemple C:\Program Files\Java\jre1.8.0_51\bin\java.exe).

Pour personnaliser votre chemin d’accès à l’interpréteur Java :

Procédure

1. Si nécessaire, cliquez sur Talend dans l’arborescence de la boîte de dialogue Preferences.
2. Si le répertoire d’installation Java par défaut n’est pas correct, rectifiez le chemin d’accès dans le champ Java interpreter.

Résultats

Dans la même vue, vous pouvez également modifier le nombre de lignes de données affichées dans l’aperçu et le chemin d’accès vers les fichiers temporaires.

Préférences du Designer (Talend > Appearance)

Pourquoi et quand exécuter cette tâche

Vous pouvez configurer les préférences d’affichage des composants et des Jobs de manière permanente dans le Studio.
**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend > Appearance**.
3. Cliquez sur **Designer** pour afficher les préférences d’affichage de l’espace de modélisation. À partir de cette vue, vous pouvez définir l’affichage des noms et des aides contextuelles de chaque composant.

![Designer](image)

4. Cochez les cases adéquates pour personnaliser l’espace de modélisation du Studio Talend en fonction de votre utilisation.

**Préférences du référentiel d’artefacts (Talend > Artifact Repository)**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez configurer la durée pendant laquelle la connexion de votre Studio Talend est maintenue au référentiel d’artefacts, à partir duquel votre Studio Talend récupère les mises à jour et les bibliothèques personnalisées et sur lequel vous pouvez publier vos artefacts.

**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend** et cliquez sur **Artifact Repository** pour afficher les préférences des projets de référence.
3. Dans le champ **Timeout for artifact repository connection (ms)**, spécifiez, en millisecondes, le temps durant lequel vous souhaitez que votre Studio Talend attende une interaction avec le serveur Nexus, avant de suspendre la connexion, 0 pour une période de temps infinie.

4. Cliquez sur **Apply** pour appliquer vos modifications. Cliquez sur **Apply and Close** pour valider les paramètres et fermer la fenêtre **Preferences**.

**Mettre à jour les préférences du serveur (Talend > Artifact Repository > Libraries)**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez configurer les préférences du Studio Talend afin de vérifier les mises à jour des bibliothèques personnalisées sur le serveur Nexus.

**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Développez successivement les nœuds **Talend** et **Artifact Repository**, puis cliquez sur **Librairies** pour afficher la vue.
3. Configurez les préférences selon vos besoins :
   - Dans le champ **Jars check frequency**, spécifiez la manière dont vous souhaitez que votre Studio Talend vérifie les mises à jour :
     - –1 si vous ne souhaitez pas que votre Studio Talend vérifie les mises à jour disponibles.
     - 0 si vous souhaitez que votre Studio Talend vérifie les mises à jour lors de chaque action nécessitant au moins un fichier .jar, par exemple lorsqu’un Job est construit ou exécuté à partir du Studio.
     - un nombre correspondant à l’intervalle entre deux vérifications.
4. Cliquez sur **Apply** pour appliquer vos modifications. Cliquez sur **Apply and Close** pour valider les paramètres et fermer la fenêtre **Preferences**.

**Préférences de Bonita BPM Portal (Talend > Bonita BPM Portal)**

**Pourquoi et quand exécuter cette tâche**
Lorsque vous créez un service BPM, vous pouvez configurer son URI ainsi que les informations de connexion au Portail Bonita BPM.

**Procédure**
1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Dans l’arborescence, développez les nœuds **Talend > Bonita BPM Portal**.
3. Saisissez les informations comme suit.

<table>
<thead>
<tr>
<th>Nom du champ</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username et Password</td>
<td>Saisissez le nom d’utilisateur et le mot de passe de connexion à la console Web de BPM. Saisissez le nom d’utilisateur et le mot de passe de connexion à la console Web de BPM, install et install par défaut.</td>
</tr>
</tbody>
</table>
4. Cliquez sur Apply puis sur Apply and Close afin de valider les préférences et fermer la fenêtre.

**Définir le dossier de composants utilisateur (Talend > Components)**


Vous pouvez télécharger et installer les composants personnalisés à utiliser dans la perspective Integration de Studio Talend.


La procédure suivante s’applique uniquement aux composants externes.

Le dossier des composants contient les composants que vous créez et/ou ceux que vous téléchargez depuis Talend Exchange. Pour le définir, procédez comme suit :

**Procédure**

1. Dans l’arborescence de la boîte de dialogue Preferences, développez le nœud Talend puis sélectionnez Components.

2. Renseignez le champ User component folder pour indiquer le chemin d’accès au dossier contenant les composants à ajouter à la Palette du Studio.

   Afin qu’ils soient importés dans la Palette du Studio, les composants utilisateur doivent être dans des dossiers séparés à la racine du dossier juste défini.


   Le Studio redémarre et les composants externes sont ajoutés à la Palette.

**Résultats**

La configuration est stockée dans les métadonnées du workspace. Si le répertoire Workspace du Studio Talend change, vous devez redéfinir la configuration.

**Définir les préférences de composants spécifiques (Talend > Components)**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez modifier certaines préférences de composants spécifiques telles que l’affichage par défaut des links de mapping.

La procédure suivante s’applique aux composants externes et aux composants inclus dans le Studio.

Pour modifier ces paramètres de composants spécifiques, procédez comme suit :

**Procédure**

1. Dans l’arborescence de la boîte de dialogue Preferences, développez le nœud Talend, puis sélectionnez Components.
2. Configurez les préférences selon vos besoins.

- Dans le champ **Default mapping links display as**, sélectionnez le type de liens que vous souhaitez utiliser pour la mise en correspondance dans le tMap.

- Cochez la case **Don’t show corresponding job after double click on a tRunJob component** si vous ne souhaitez pas que le Job fils appelé par le tRunJob s’ouvre lorsque vous double-cliquez sur le composant.

  **Remarque:** Vous pourrez toujours ouvrir le Job correspondant en cliquant-droit sur le composant tRunJob et en sélectionnant **Open tRunJob Component**.

- Sous **Component Assist**, cochez la case **Enable Component Creation Assistant** afin de pouvoir ajouter un composant en saisissant son nom directement dans l’espace de modélisation graphique. Pour plus d’informations, consultez Ajouter des composants au Job à la page 32.

- Cochez la case **Enable online help for components** si vous souhaitez activer l’aide en ligne, qui vous permet de voir le contenu le plus à jour sur Talend Help Center (https://help.talend.com). Décrochez la case si vous souhaitez activer l’aide embarquée. Par défaut, cette case est cochée et l’aide en ligne est activée.

3. Cliquez sur **Apply** afin de valider les préférences. Cliquez sur **Apply and Close** afin de valider vos préférences et fermer la boîte de dialogue.

**Résultats**
La configuration est stockée dans les métadonnées du workspace. Si le répertoire Workspace du Studio Talend change, vous devez redéfinir la configuration.

**Préférences de la documentation (Talend > Documentation)**

**Pourquoi et quand exécuter cette tâche**
Dans les préférences, vous pouvez inclure le code source dans vos documentations générées.
Procédure

1. Cliquez sur le menu Window > Preferences pour ouvrir la boîte de dialogue Preferences.
2. Développez le nœud Talend et cliquez sur Documentation pour afficher les préférences de la documentation.

3. Personnalisez les préférences de la documentation selon vos besoins :
   - Cochez la case Source code to HTML generation pour intégrer le code source dans la documentation HTML que vous générerez.
   - Cochez la case Use CSS File as a template when export to HTML afin d’activer le champ CSS File, si vous devez utiliser un fichier CSS pour personnaliser les fichiers HTML exportés.

Résultats

Pour plus d’informations concernant la documentation, consultez Générer la documentation HTML à la page 193 et Onglet Documentation à la page 64.

Préférences des services REST (Talend > ESB)

Pourquoi et quand exécuter cette tâche

Vous pouvez configurer l’URI par défaut de l’endpoint pour les services REST et l’espace de noms par défaut du Service Locator. Pour cela :

Procédure

1. Cliquez sur le menu Window > Preferences pour ouvrir la boîte de dialogue Preferences.
2. Développez le nœud Talend > ESB.
3. Cliquez sur Service-REST pour afficher la vue correspondante.
   Dans cette vue, vous pouvez définir l’URI par défaut de l’endpoint URI pour les services REST et l’espace de noms par défaut du Service Locator.
Préférence Exchange (Talend > Exchange)

Avant de commencer

**Avertissement:** Assurez-vous que le paramètre `-Dtalend.disable.internet` n'est pas dans le fichier `.ini` du Studio ou défini comme `false`.

Pourquoi et quand exécuter cette tâche

Vous pouvez configurer les préférences relatives à votre connexion à Talend Exchange, site faisant partie de la Communauté Talend, dans le Studio Talend. Pour cela :

**Procédure**

1. Cliquez sur le menu `Window > Preferences` pour ouvrir la boîte de dialogue `Preferences`.
2. Développez le nœud `Talend` et cliquez sur `Exchange` afin d'afficher la vue `Exchange`.

![Image du panneau Exchange](image.png)

3. Configurez les préférences d'Exchange selon vos besoins :
   - Si vous n’êtes pas encore connecté(e) à la Communauté Talend, cliquez sur `Sign In` pour vous rendre sur la page `Connect to Talend Community` afin de vous connecter à la Communauté Talend à l’aide de votre identifiant/mot de passe ou pour vous créer un compte Talend et de vous y connecter.
   - Si vous êtes déjà connecté(e) à la Communauté Talend, votre compte s’affiche et le bouton `Sign In` devient `Sign Out`. Pour vous déconnecter de la Communauté Talend, cliquez sur `Sign Out`.
   - Par défaut, lorsque vous êtes connecté(e) à la Communauté Talend, lorsqu’une mise à jour d’un produit installé est disponible, une boîte de dialogue apparaît pour vous le signaler. Si vous vérifiez fréquemment les mises à jour disponibles et que vous ne souhaitez pas que cette boîte de dialogue réapparaisse, décochez la case `Notify me when updated extensions are available`.

Préférences de Talend Metadata Bridge (Talend > Import/Export)

Pourquoi et quand exécuter cette tâche

Vous pouvez configurer les préférences de Talend Metadata Bridge afin de le faire fonctionner comme vous le souhaitez.

**Procédure**

1. Dans le menu, cliquez sur `Window > Preferences` pour afficher la boîte de dialogue `Preferences`.
2. Développez les nœuds `Talend` et `Import/Export` successivement puis cliquez sur `Metadata Bridge` pour afficher la vue correspondante.
3. Configurez les préférences selon votre utilisation de Talend Metadata Bridge :
   - dans la zone Location, sélectionnez l’option Embedded pour utiliser l’outil MIMB embarqué dans Talend Metadata Bridge. Cette option est l’option par défaut. Pour utiliser l’outil MIMB installé localement, sélectionnez Local Directory et spécifiez le répertoire d’installation de l’outil MIMB.
   - Dans le champ Temp folder, spécifiez le répertoire qui contiendra les fichiers temporaires générés durant l’import/export de métadonnées, si vous ne souhaitez pas utiliser le répertoire par défaut.
   - Dans le champ Log folder, spécifiez le répertoire qui contiendra les fichiers de log générés durant l’import/export de métadonnées, si vous ne souhaitez pas utiliser le répertoire par défaut.
   - Cochez la case Show detailed logs pour générer des fichiers de log détaillés durant l’import/export de métadonnées.

Résultats
Pour plus d’informations concernant l’utilisation du Metadata Bridge Talend pour importer/exporter des métadonnées, consultez Import et export de métadonnées à l’aide de Talend Metadata Bridge sur Talend Help Center (https://help.talend.com).

Préférences de langue (Talend > Internationalization)

Pourquoi et quand exécuter cette tâche
Vous pouvez configurer les préférences de langue dans le Studio Talend. Pour cela :
**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend** et cliquez sur **Internationalization** pour afficher les préférences de langue.
3. Dans le champ **Local Language**, sélectionnez la langue que vous souhaitez utiliser pour l'interface du Studio Talend.
4. Cliquez sur **Apply** puis sur **Apply and Close** pour fermer la boîte de dialogue **Preferences**.
5. Redémarrez le Studio pour que ce changement soit pris en compte.

**Préférences de la Palette (Talend > Palette Settings)**

**Pourquoi et quand exécuter cette tâche**

Dans la vue **Palette Settings**, vous pouvez configurer les préférences liées à la recherche de composants dans la **Palette** et même dans la liste qui s’affiche dans l’espace de modélisation graphique lors de l’ajout d’un composant sans utilisation de la **Palette**.

**Procédure**

1. Dans le menu, cliquez sur **Window > Preferences** pour afficher la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend** et cliquez sur **Palette Settings** pour afficher la vue **Palette Settings**.
3. Afin de limiter le nombre de composants pouvant être affichés dans la liste **Recently Used**, saisissez le nombre maximal que vous souhaitez afficher dans le champ **Recently used list size**.
4. Pour activer la possibilité de rechercher un composant en saisissant une phrase décrivant sa fonctionnalité ou sa finalité dans le champ de recherche de la **Palette** ou dans le champ de texte.
s’affichant dans l’espace de modélisation graphique lorsque vous commencez à saisir, cochez la case **Also search from Help when performing a component searching**. Lorsque cette case est cochée, vous pouvez voir votre composant dans la **Palette** ou dans la liste s’affichant dans l’espace de modélisation graphique, tant qu’il est disponible dans l’aide (touche F1) lorsque vous saisissez une phrase descriptive incluse dans cette aide.

5. Pour limiter le nombre de résultats de recherche ou en afficher plus, lorsque vous saisissez une phrase dans le champ de recherche, saisissez le nombre de résultats que vous souhaitez voir retournés, dans le champ **Result limitation from Help**.

**Préférences des performances (Talend > Performance)**

Dans les préférences, vous pouvez définir des options de performance en fonction de votre utilisation du Studio Talend. Pour ce faire, procédez comme suit :

**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend** et cliquez sur **Performance** pour afficher les préférences de performances du référentiel.

![Performance settings](image)

**Remarque:** La désactivation du rafraîchissement automatique permet un gain de performance.

3. Dans les préférences, vous pouvez définir des options de performance en fonction de votre utilisation du Studio Talend :
   - Cochez la case **Deactivate auto detect/update after a modification in the repository** pour désactiver la détection et la mise à jour automatique du Repository après modification.
• Cochez la case **Check the property fields when generating code** pour activer la vérification des champs de propriétés des composants. Lorsque l’un des champs de propriétés d’un composant est mal renseigné, le composant est entouré en rouge.

**Remarque:** Décochez la case **Check the property fields when generating code**.

• Cochez la case **Generate code when opening the job** pour générer le code à l’ouverture du Job.
• Cochez la case **Check only the last version when updating jobs or joblets** pour ne vérifier que la dernière version des Jobs.
• Cochez la case **Propagate contexts added in repository context groups** pour permettre la propagation aux Jobs des nouveaux contextes dans les groupes de contextes du référentiel.

Lorsque cette option est activée, chaque fois que vous ouvrez un Job utilisant un groupe de contextes du référentiel, une boîte de dialogue s’ouvre, vous demandant si vous souhaitez effectuer une propagation des contextes, si un contexte a été ajouté au groupe mais n’a pas été synchronisé avec le groupe.

Cette option est désactivée par défaut.
• Cochez la case **Propagate add/delete variable changes in repository contexts** pour autoriser la propagation des modifications de variables dans les contextes du référentiel.
• Cochez la case **Activate the timeout for database connection** pour mettre en place un délai d’expiration des connexions aux bases de données. Puis dans le champ **Connection timeout (seconds)**, saisissez, en secondes, la durée souhaitée avant expiration.
• Cochez la case **Add all user routines to job dependencies, when create new job**, afin d’ajouter toutes les routines utilisateur dans les dépendances des Jobs lors de la création de nouveaux Jobs.
• Dans le champ **Code Format timeout (seconds)**, spécifiez le nombre de secondes après lequel votre Studio Talend doit arrêter de formater le code source après génération. Par exemple, lorsque vous passez de la vue Designer à la vue Code ou lorsque vous construisez un Job ou une Route. La valeur doit être un entier supérieur à 0. Configurer un courte période de temps permet d’éviter des problèmes de performances, au prix d’une lisibilité réduite du code source, particulièrement pour un Job ou une Route de grande taille et complexe.

**Préférences d’exécution et de débogage (Talend > Run/Debug)**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez configurer les préférences d’exécution et de débogage dans le Studio Talend. Pour cela :

**Procédure**

1. Dans le menu, cliquez sur **Window > Preferences** pour afficher la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend** et cliquez sur **Run/Debug** pour afficher les préférences.
3. Paramétrez ces préférences en fonction de vos besoins.

- Dans la zone **Talend client configuration**, vous pouvez définir les options d’exécution à utiliser par défaut.

<table>
<thead>
<tr>
<th>Paramètre</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stats port range</td>
<td>Spécifiez une plage pour les ports utilisés pour la génération des Statistics, notamment si les ports définis par défaut sont déjà occupés par d'autres applications.</td>
</tr>
<tr>
<td>Trace port range</td>
<td>Spécifiez une plage pour les ports utilisés pour la génération des Traces, notamment si les ports définis par défaut sont déjà occupés par d'autres applications.</td>
</tr>
<tr>
<td>Save before run</td>
<td>Cochez cette case pour automatiquement enregistrer votre Job avant de l’exécuter.</td>
</tr>
<tr>
<td>Clear before run</td>
<td>Cochez cette case pour nettoyer les résultats d’une exécution précédente avant d’exécuter de nouveau le Job.</td>
</tr>
<tr>
<td>Exec time</td>
<td>Cochez cette case pour afficher la durée d’exécution du Job.</td>
</tr>
<tr>
<td>Statistics</td>
<td>Cochez cette case le suivi de flux au cours de l’exécution du Job.</td>
</tr>
<tr>
<td>Traces</td>
<td>Cochez cette case pour afficher le suivi du traitement des données au cours de l’exécution du Job.</td>
</tr>
<tr>
<td>Pause time</td>
<td>Indiquez le temps de pause souhaitée entre chaque ligne de données du tableau Traces.</td>
</tr>
</tbody>
</table>

- Dans la liste **Job Run VM arguments**, vous pouvez définir les paramètres de votre JVM en fonction de votre utilisation. Les paramètres par défaut -Xms256M et -Xmx1024M correspondent respectivement aux capacités minimale et maximale de mémoire réservées pour vos exécutions de Jobs.
Annexes

Si vous souhaitez utiliser des paramètres spécifiques pour l'exécution d'un Job, par exemple si vous voulez afficher les résultats d'exécution de ce Job en japonais, vous devez ouvrir la vue Run. Dans cette vue, configurez les paramètres d'exécution avancés correspondants.

Pour plus d'informations concernant les paramètres d'exécution avancés d'un Job spécifique, consultez Configurer les paramètres d'exécution avancés à la page 203.
Pour plus d'informations concernant les paramètres possibles, consultez le site (en anglais) http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html.

Afficher des caractères spéciaux dans les colonnes de schémas (Talend > Specific settings)

Pourquoi et quand exécuter cette tâche

Vous pouvez avoir besoin de récupérer le schéma d'une table contenant des colonnes avec caractères spéciaux comme des caractères chinois, japonais ou coréens. Dans ce cas, vous devez activer l’option permettant de lire les caractères spéciaux dans le Studio Talend. Pour cela :

Procédure

1. Cliquez sur le menu Window > Preferences pour ouvrir la boîte de dialogue Preferences.
2. Dans l’arborescence de la boîte de dialogue, développez le nœud Talend.
3. Cliquez sur le nœud Specific settings pour afficher la vue correspondante dans la partie droite de la boîte de dialogue.
4. Cochez la case Allow specific characters (UTF8,...) for columns of schemas.

Préférences des schémas (Talend > Specific Settings)

Pourquoi et quand exécuter cette tâche

Dans les préférences, vous pouvez définir la longueur et le type de données par défaut des champs des schémas de vos composants.

Procédure

1. Cliquez sur le menu Window > Preferences pour ouvrir la boîte de dialogue Preferences.
2. Développez le nœud Talend et cliquez sur Specific Settings > Default Type and Length pour afficher les préférences de longueur et de type des champs de vos schémas.
3. Paramétrez ces préférences en fonction de vos besoins :

- Dans la zone **Default Settings for Fields with Null Values**, renseignez le type de données et la longueur du champ à attribuer par défaut aux champs de valeurs nulles.
- Dans la zone **Default Settings for All Fields**, renseignez le type de données et la longueur du champ à attribuer à tous les champs du schéma.
- Dans la zone **Default Length for Data Type**, renseignez la longueur à attribuer en fonction du type de données du champ.

**Préférences du SQL Builder (Talend > Specific Settings)**

**Pourquoi et quand exécuter cette tâche**

Dans les préférences, vous pouvez définir les préférences du SQL Builder. Pour cela :

**Procédure**

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **Preferences**.
2. Développez les nœuds **Talend et Specific Settings** et cliquez sur **Sql Builder** pour afficher les préférences du SQL Builder.

3. Personnalisez les performances du SQL Builder selon vos besoins :
• Cochez la case **add quotes, when you generated sql statement** pour protéger le nom des colonnes et des tables par des guillemets dans vos requêtes SQL.
• Pour le champ **AS400 SQL generation**, sélectionnez **Standard SQL Statement** pour utiliser des commandes SQL standard ou **System SQL Statement** pour des commandes SQL system lorsque vous utilisez des bases de données de type AS/400.
• Décochez la case **Enable check queries in the database components (disable to avoid warnings for specific queries)** pour désactiver la vérification des requêtes dans les composants base de données.

**Préférences des paramètres SSL (Talend > SSL)**

**Pourquoi et quand exécuter cette tâche**

Vous pouvez paramétrer les préférences du SSL afin de configurer votre Studio Talend pour des communications sécurisées avec des serveurs distants.

**Procédure**

1. Dans le menu, cliquez sur **Window > Preferences** pour afficher la boîte de dialogue **Preferences**.
2. Développez le nœud **Talend** et cliquez sur **SSL** pour afficher la vue correspondante.
3. Définissez la configuration du Keystore, dans la zone **Keystore Configuration** pour que le certificat local soit envoyé à l’hôte distant :
   a) Cliquez sur **Browse** à côté du champ **Path** et parcourez votre système jusqu’au fichier Keystore stockant vos identifiants locaux.
   b) Dans le champ **Password**, saisissez le mot de passe du Keystore.
   c) Dans la liste **Keystore Type**, sélectionnez le type de Keystore à utiliser.
4. Paramétrez la configuration du Truststore, dans la zone **Truststore Configuration** pour vérifier le certificat de l’hôte distant :
   a) Cliquez sur **Browse**, à côté du champ **Path** et parcourez votre système jusqu’au fichier Truststore.
   b) Dans le champ **Password**, saisissez le mot de passe du Truststore.
   c) Dans la liste **Keystore Type**, sélectionnez le type de Keystore à utiliser.
5. Cliquez sur **Apply** pour appliquer vos modifications. Cliquez sur **Apply and Close** pour valider les paramètres et fermer la fenêtre **Preferences**.
6. Redémarrez votre Studio Talend pour que la configuration soit prise en compte.

**Préférences du collecteur de données d’utilisation (Talend > Usage Data Collector)**

**Pourquoi et quand exécuter cette tâche**

En autorisant le Studio Talend à collecter vos statistiques d’utilisation du Studio, vous permettez aux utilisateurs de mieux comprendre les produits **Talend** et vous permettez à **Talend** de savoir comment les utilisateurs utilisent les produits. Cela permet à **Talend** d’améliorer la qualité des produits et des performances afin de mieux répondre aux besoins des utilisateurs.

Par défaut, le Studio Talend collecte automatiquement vos données d’utilisation et les envoie régulièrement aux serveurs hébergés par **Talend**. Vous pouvez voir la collection de données d’utilisation et le chargement d’informations, ainsi que personnaliser les performances du collecteur de données selon vos besoins.
Remarque: Soyez assuré que seules les statistiques d'utilisation du Studio sont collectées. Aucune de vos informations privées ne sera collectée et transmise à Talend.

Procédure

1. Dans le menu, cliquez sur Window > Preferences pour afficher la boîte de dialogue Preferences.
2. Développez le nœud Talend et cliquez sur Usage Data Collector pour afficher la vue Usage Data Collector.

3. Lisez le message concernant le collecteur de données d'utilisation et, si vous ne souhaitez pas que le collecteur collecte et envoie vos informations d'utilisation du Studio, décochez la case Enable capture.
4. Pour obtenir un aperçu des données d'utilisation capturées par le collecteur de données, développez le nœud Usage Data Collector et cliquez sur Preview.
5. Pour personnaliser l'intervalle d'envoi des données d'utilisation et voir la date du dernier envoi, cliquez sur **Uploading** sous le nœud **Usage Data Collector**.

- Par défaut, s’il est activé, le collecteur de données collecte les données d'utilisation du produit et les envoie aux serveurs **Talend** tous les dix jours. Pour modifier l'intervalle, saisissez une nouvelle valeur entière (en jours) dans le champ **Upload Period**.
- Le champ **Last Upload** en lecture seule affiche la date et l’heure du dernier envoi de données aux serveurs **Talend**.
Configurer un proxy HTTPS d’authentification pour une connexion Azure Storage

Cette tâche configure un proxy HTTPS avec authentification utilisateur activée pour une connexion Azure Storage existante.

Pourquoi et quand exécuter cette tâche

Cette tâche permet au Studio Talend de se connecter à Azure Storage via un serveur proxy HTTPS ayant l’authentification utilisateur activée. Cela présume que l’adresse du serveur HTTPS, l’identifiant et le mot de passe configurés sur le serveur du proxy sont disponibles.

Procédure

1. Dans le Repository, développez le nœud Metadata>Azure Storage.
   La connexion à Azure Storage s’affiche sous Azure Storage.
2. Cliquez-droit sur la connexion à Azure Storage et sélectionnez Azure Storage Connection dans le menu contextuel.
   La boîte de dialogue Azure Storage s’ouvre.
   La boîte de dialogue Preferences (Filtered) s’ouvre.
   Le panneau Network Connections s’affiche dans la boîte de dialogue, à droite.
5. Sélectionnez la ligne HTTPS dans la table Proxy entries, puis cliquez sur Edit....
   La boîte de dialogue Edit Proxy Entry s’ouvre.
6. Fournissez les paramètres suivants dans les champs correspondants :
   • Host : spécifiez l’adresse IP du serveur HTTPS.
   • Port : spécifiez le numéro de port utilisé.
   • Require Authentication : sélectionnez cette option pour saisir un identifiant et un mot de passe.
   • User : identifiant pour l’authentification.
   • Password : mot de passe pour l’authentification.
7. Cliquez sur OK puis sur Apply and Close afin de valider les paramètres.

SQL Templates

Qu’est-ce que l’ELT ?

Extract, Load and Transform (ELT), ou, en français, Extraire, Charger et Transformer, est un processus de manipulation de données lié aux bases de données, et plus particulièrement aux data warehouses. Ce mode est différent du mode ETL habituel (Extract, Transform, Load). Ces données sont migrées en masse et le processus de transformation s’effectue après le chargement des données dans le SGBD cible, en format brut. Cela permet de libérer de la bande passante.

Cependant, le mode ELT n’est pas optimal dans toutes les situations.

• SQL est moins puissant que Java, le nombre de transformations de données est limité,
• les utilisateurs du mode ELT doivent avoir des compétences en personnalisation de SQL et de SGBD.
si vous utilisez le mode ELT avec le Studio Talend, vous ne pouvez ni passer ni rejeter une ligne de données, comme vous pouvez le faire avec l'ETL. Pour plus d'informations concernant le rejet de lignes, consultez Connexion de type Row à la page 78.

Les modèles SQL sont conçus pour faciliter l'utilisation du mode ELT.

Définition des SQL Templates Talend

Le SQL est un langage standardisé de requêtes, utilisé pour accéder aux informations des bases de données et les gérer. Le SQL peut être utilisé pour les requêtes sur les données, les mises à jour, la création et la modification de schémas et le contrôle d'accès aux données. Le Studio Talend fournit de nombreux modèles SQL pour simplifier les tâches les plus communes. Il comprend également un éditeur SQL vous permettant de personnaliser ou de créer vos propres modèles SQL, afin de répondre à vos besoins.

Ces modèles SQL sont utilisés avec les composants de la famille Talend ELT, tels que le tSQLTemplate, le tSQLTemplateFilterColumns, le tSQLTemplateCommit, le tSQLTemplateFilterRows, le tSQLTemplateRollback, le tSQLTemplateAggregate et le tSQLTemplateMerge. Ces composants exécutent les instructions SQL sélectionnées. À l'aide des opérateurs UNION, EXCEPT et INTERSECT, vous pouvez modifier les données directement dans le SGBD sans utiliser le système de mémoire.

De plus, avec ces modèles SQL, vous pouvez optimiser les performances de votre SGBD, en stockant et en récupérant vos données selon vos besoins structurels.

Le Studio Talend fournit les types suivants de modèles SQL, sous le nœud SQL templates de la vue Repository :

- Modèles SQL system : ils sont classés selon le type de base de données.
- Modèles SQL personnalisés : ce sont les modèles que vous avez créés ou adaptés des modèles existants.

Des informations plus précises concernant les modèles SQL sont présentées dans les sections suivantes.

**Remarque:**

Comme la plupart des modèles SQL sont créés pour une base de données spécifique, si vous changez de base de données dans votre système, vous devrez également changer de modèles, ou en développer de nouveaux.

Gérer les SQL Templates Talend

Le Studio Talend vous permet, via le dossier SQL Templates de la vue Repository, d’utiliser des modèles SQL système ou personnalisés dans les Jobs que vous créez dans le Studio, à l’aide des composants de la famille ELT.

Les sections suivantes vous expliquent comment gérer ces deux types de modèles SQL.

Les types de SQL Templates système

Cette section fournit des informations détaillées concernant les différents types de modèles SQL prédéfinis.

Les instructions de chaque groupe de modèles varient d’une base de données à l’autre, selon les opérations à effectuer.
Le tableau ci-dessous présente ces types ainsi que les informations liées à chacun.

<table>
<thead>
<tr>
<th>Nom</th>
<th>Fonction</th>
<th>Composants associés</th>
<th>Paramètres requis dans le composant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate</td>
<td>Réalise l’agrégation (sum, average, count, etc.) d’un ensemble de données.</td>
<td>tSQLTemplateAggregate</td>
<td>Nom de la base de données Nom de la table source Nom de la table cible</td>
</tr>
<tr>
<td>Commit</td>
<td>Envoie une instruction de Commit au SGBDR.</td>
<td>tSQLTemplate, tSQLTemplateFilterColumns, tSQLTemplateCommit, tSQLTemplateFilterRows, tSQLTemplateRollback, tSQLTemplateAggregate et tSQLTemplateMerge.</td>
<td>Null</td>
</tr>
<tr>
<td>Rollback</td>
<td>Envoie une instruction Rollback au SGBDR.</td>
<td>tSQLTemplate, tSQLTemplateFilterColumns, tSQLTemplateCommit, tSQLTemplateFilterRows, tSQLTemplateRollback, tSQLTemplateAggregate et tSQLTemplateMerge.</td>
<td>Null</td>
</tr>
<tr>
<td>DropSourceTable</td>
<td>Supprime une source table.</td>
<td>tSQLTemplate, tSQLTemplateAggregate, tSQLTemplateFilterColumns, tSQLTemplateFilterRows</td>
<td>Nom de la table (lors de l’utilisation du tSQLTemplate) Nom de la table source</td>
</tr>
<tr>
<td>DropTargetTable</td>
<td>Supprime une table cible.</td>
<td>tSQLTemplateAggregate, tSQLTemplateFilterColumns, tSQLTemplateFilterRows</td>
<td>Nom de la table cible</td>
</tr>
<tr>
<td>FilterColumns</td>
<td>Sélectionne et extrait un ensemble de données dans des colonnes données du SGBDR.</td>
<td>tSQLTemplateAggregate, tSQLTemplateFilterColumns, tSQLTemplateFilterRows</td>
<td>Nom de la table cible (et schéma) Nom de la table source (et schéma)</td>
</tr>
<tr>
<td>FilterRow</td>
<td>Sélectionne et extrait un ensemble de données dans des lignes données du SGBDR.</td>
<td>tSQLTemplateAggregate, tSQLTemplateFilterColumns, tSQLTemplateFilterRows</td>
<td>Nom de la table cible (et schéma) Nom de la table source (et schéma) Conditions</td>
</tr>
<tr>
<td>MergeInsert</td>
<td>Insère des enregistrements d’une table source vers une table cible.</td>
<td>tSQLTemplateMerge, tSQLTemplateCommit</td>
<td>Nom de la table cible (et schéma) Nom de la table source (et schéma) Conditions</td>
</tr>
<tr>
<td>MergeUpdate</td>
<td>Met à jour la table cible avec les enregistrements de la table source.</td>
<td>tSQLTemplateMerge, tSQLTemplateCommit</td>
<td>Nom de la table cible (et schéma) Nom de la table source (et schéma) Conditions</td>
</tr>
</tbody>
</table>
Accéder aux SQL Templates système

Pour accéder à un modèle SQL système, développez le nœud SQL Templates de la vue Repository.

Chaque dossier contient un sous-dossier system, dans lequel se trouvent des instructions SQL prédéfinies, et un sous-dossier UserDefined dans lequel vous pouvez stocker les instructions SQL que vous avez créées ou personnalisées.

Chaque dossier système contient différents types de modèles SQL, chaque modèle étant créé pour accomplir une tâche dédiée.

À l’exception du dossier Generic, les modèles SQL sont groupés dans différents dossiers, selon le type de base de données avec lequel les utiliser. Les modèles du dossier Generic ne sont dédiés à aucune base de données, ils sont standards et conviennent à toutes. À partir de ceux-ci, vous pouvez développer des modèles SQL plus spécifiques que ceux définis par le Studio Talend.

Remarque:
Les dossiers system et leur contenu sont en lecture seule.
Dans la vue Repository, procédez comme suit pour ouvrir un modèle SQL :

**Procédure**

1. Dans la vue Repository, développez le nœud SQL Templates et parcourez jusqu’au modèle que vous souhaitez ouvrir.
2. Double-cliquez sur la classe que vous voulez ouvrir, par exemple Aggregate, dans le dossier Generic.

La vue du modèle Aggregate s’affiche dans l’espace de travail.

**Résultats**

Vous pouvez lire les instructions Aggregate prédéfinies dans la vue du modèle. Les paramètres, tels que TABLE_NAME_TARGET, operation, sont à définir lors de la création de Jobs liés. La configuration peut ensuite facilement être paramétrée, via les composants associés, comme expliqué dans la section précédente.

A chaque fois que vous cliquez sur un modèle SQL ou que vous ouvrez un modèle SQL, la vue des propriétés correspondante à ce modèle s’affiche en bas du Studio. Par exemple, vous cliquez le modèle Aggregate ou vous l’ouvrez, ses propriétés seront présentées comme suit :

Pour plus d’informations concernant les types de modèles SQL, consultez Les types de SQL Templates système à la page 521.
Créer des SQL Templates personnalisés

Comme la transformation que vous devez effectuer en mode ELT peut dépasser les possibilités des modèles SQL, le Studio Talend vous permet de développer vos propres modèles SQL, à condition de respecter quelques règles d’écriture. Ces modèles SQL sont stockés dans les dossiers User-Defined, regroupés selon la base de données dans laquelle ils seront utilisés.

Pour plus d’informations concernant les règles d’écriture des modèles SQL, consultez Instructions SQL à la page 531.

Pour créer un modèle SQL personnalisé :

Procédure

1. Dans la vue Repository, développez le nœud SQL Templates, puis la catégorie dans laquelle vous souhaitez créer le modèle SQL.

3. Saisissez les informations requises pour créer le modèle, puis cliquez sur **Finish** pour fermer l’assistant.

Le nouveau modèle apparaît sous le nœud **UserDefined**, dans la vue **Repository**. Un éditeur de modèles SQL s’ouvre dans l’espace de modélisation graphique.

Pour plus d’informations concernant la création d’un modèle SQL personnalisé et comment l’utiliser dans un Job, consultez la section **Effectuer une opération d’itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l’utilisateur (SQL Template) sur MySQL**.

**Exemple d’utilisation des SQL Templates système**

Puisque beaucoup d’instructions SQL sont communes, standardisées, le Studio Talend vous permet de bénéficier de plusieurs modèles SQL système.

Cette section vous présente un exemple d’utilisation des modèles SQL système, à travers les étapes de l’utilisation de modèles SQL système Mysql dans un Job qui :

- ouvre une connexion à une base de données MySQL,
- collecte des données regroupées par valeurs spécifiques à partir d’une table de la base de données et écrit des données agrégées dans une table cible de la base de données,
- supprime la table source d’où proviennent les données agrégées,
- lit la table cible et liste les résultats d’exécution du Job.

Pour vous connecter à la base de données et agréger les colonnes de la table :

![New SQL Template](image)
Configurer une connexion à une base de données MySQL

**Procédure**

1. À partir de la Palette, glissez les composants suivants dans l'espace de modélisation graphique : `tMysqlConnection`, `tSQLTemplateAggregate`, `tSQLTemplateCommit`, `tMysqlInput` et `tLogRow`.
2. Reliez le `tMysqlConnection` au `tSQLTemplateAggregate` à l'aide d'un lien `Trigger > On Subjob Ok`.
3. Répétez l'opération afin de connecter le `tSQLTemplateAggregate` au `tSQLTemplateCommit` et le `tSQLTemplateCommit` au `tMysqlInput`.
4. Connectez le `tMysqlInput` au `tLogRow` à l'aide d'un lien `Main > Row`.

5. Dans l'espace de modélisation graphique, double-cliquez sur le composant `tMysqlConnection` pour ouvrir sa vue `Basic settings`.

6. Dans la vue `Basic settings`, configurez manuellement les paramètres de connexion.
7. Dans l’espace de modélisation graphique, double-cliquez sur le composant **tSQLTemplateCommit** pour ouvrir sa vue **Basic settings**.

8. Dans la liste **Database Type**, sélectionnez la base de données correspondante et dans la liste **Component List**, sélectionnez le composant de connexion adéquat, si plus d’une connexion est ouverte dans le Job.

**Regrouper les données, écrire des données agrégées et supprimer la table**

**Procédure**

1. Dans l’espace de modélisation graphique, double-cliquez sur le **tSQLTemplateAggregate** pour ouvrir sa vue **Basic settings**.

2. Dans la liste **Database Type**, sélectionnez la base de données correspondante et dans la liste **Component List**, sélectionnez le composant de connexion adéquat, si plus d’une connexion est ouverte dans le Job.

3. Saisissez le nom de la base de données, de la table source et de la table cible dans les champs correspondants et définissez la structure des données, dans la table source et dans la table cible.

Le schéma de la table source comprend trois colonnes : **First_Name, Last_Name et Country**.

Le schéma de la table cible comprend deux colonnes : **country et total**. Dans cet exemple, regroupez les citoyens par nationalité, et comptez le nombre de personnes dans chaque pays.

Pour ce faire, définissez les paramètres **Operations et Groupby** selon vos besoins.
4. Dans le tableau **Operations**, cliquez sur le bouton [+] afin d’ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** et sélectionnez dans la liste déroulante la colonne de sortie qui contiendra les données comptées.

5. Cliquez sur la ligne **Function**, puis sélectionnez dans la liste l’opération à effectuer.

6. Dans le tableau **Group by**, cliquez sur le bouton [+] afin d’ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** et sélectionnez dans la liste la colonne de sortie qui contiendra les données agrégées.

7. Cliquez sur l’onglet **SQL Template** pour ouvrir la vue correspondante.

8. Cliquez deux fois sur le bouton [+] sous le tableau **SQL Template List** afin d’ajouter deux modèles SQL.

9. Cliquez sur la ligne du premier modèle SQL et sélectionnez le modèle **MySQLAggregate** dans la liste déroulante. Ce modèle génère le code pour agréger des données, selon la configuration des **Basic settings**.

10. Répétez l’opération et sélectionnez le modèle **MySQLDropSourceTable** pour la ligne du deuxième modèle. Ce modèle génère le code pour supprimer la table source d’où proviennent les données agrégées.

**Remarque:**
Afin d’ajouter de nouveaux modèles SQL à un composant ELT chargé de les exécuter, vous pouvez simplement glisser le(s) modèle(s) de votre choix à ce composant soit dans l’espace de travail soit dans le tableau **SQL Template List** de ce composant.

**Remarque:**
Les modèles définis dans le tableau **SQL Template List** sont prioritaires par rapport à la configuration paramétrée dans la vue **Basic settings**, et sont exécutés en ordre décroissant. Dans cet exemple, si vous sélectionnez uniquement **MySQLDropSourceTable** pour la première ligne de modèle et **MySQLAggregate** pour la seconde, la table source sera supprimée définitivement et l’agréation ne pourra être effectuée.
Lire la base de données cible et lister le résultat de l’exécution du Job

**Procédure**

1. Double-cliquez sur le **tMysqlInput** pour ouvrir sa vue **Basic settings**.

![Toujours](image.png)

2. Cochez la case **Use an existing connection** pour utiliser la connexion à la base de données définie dans le composant **tMysqlConnection**.

3. Afin de définir le schéma, sélectionnez **Repository** et cliquez sur le bouton [...] pour sélectionner la table de la base de données dont le schéma est utilisé. Dans cet exemple, la table cible contenant les données agrégées est sélectionnée.

4. Dans le champ **Table Name**, saisissez le nom de la table sur laquelle effectuer votre requête. Dans cet exemple, la table est celle qui contient les données agrégées.

5. Dans la zone **Query**, saisissez l'instruction de la requête pour sélectionner les colonnes à afficher.

6. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

La table source est supprimée.

![Start结束](image.png)

Une table **citizencount** à deux colonnes est créée dans la base de données. Elle regroupe les citoyens selon leur nationalité et donne le compte total pour chaque pays.

![Statistics](image.png)
Règles d’écriture des SQLTemplates

Instructions SQL

Une instruction SQL peut être n’importe quelle instruction SQL valide exécutable par la JDBC liée. Le code des modèles SQL est un groupe d’instructions SQL. Les règles de base pour écrire une instruction SQL dans l’éditeur de modèles SQL sont :

• une instruction SQL doit se terminer par ;,
• une instruction SQL peut s’étendre sur plusieurs lignes. Dans ce cas, seule la dernière ligne doit se terminer par ;.

Lignes de commentaire

Une ligne de commentaire commence par # ou --. Chaque ligne débutant par # ou -- sera ignorée lors de la génération du code.

Remarque:
Aucune exception n’est faite pour les lignes dans la partie du milieu d’une instruction SQL, ou à l’intérieur de la syntaxe <%...%>.

La syntaxe <%...%>

Cette syntaxe peut s’étendre sur plusieurs lignes. Les points suivants listent ce que vous pouvez faire avec cette syntaxe, et ce à quoi vous devez faire attention.

• Vous pouvez définir de nouvelles variables, utiliser le code logique Java, comme if, for et while et également obtenir les valeurs des paramètres.

Par exemple, si vous souhaitez obtenir le paramètre FILE_NAME, utilisez le code comme suit :

```<%
String filename = __FILE_NAME__;
%>```

• Cette syntaxe ne peut être utilisée dans une instruction SQL. Elle doit être utilisée entre deux instructions SQL séparées.

Par exemple, la syntaxe dans le code suivant est valide :

```#sql sentence
DROP TABLE temp_0;
<% #loop
for(int i=1; i<10; i++){ %>
#sql sentence
DROP TABLE temp_<%=i %>; <%
} %>
#sql sentence
DROP TABLE temp_10;``` 

Dans cet exemple, la syntaxe est utilisée entre deux modèles SQL séparés : DROP TABLE temp_0; et DROP TABLE temp_&lt;%=i%&gt;.
Les instructions SQL ont pour but de supprimer plusieurs tables, en commençant par temp_0. Le code entre <%% et %> génère un nombre de séquences dans la boucle, afin d’identifier les tables à supprimer et fermer la boucle après la génération du nombre de séquences.

- A l’intérieur de cette syntaxe, la syntaxe <%%=...%> ou </.../> ne doit pas être utilisée. <%%=...%> et </.../> sont également des syntaxes liées aux modèles SQL. Les sections suivantes donnent des informations relatives à ces syntaxes.

**Remarque:**
Les paramètres auxquels les modèles SQL peuvent accéder grâce à cette syntaxe sont simples. Ils sont souvent utilisés lors de connexions et peuvent être facilement définis dans les composants, par exemple TABLE_NAME, DB_VERSION, SCHEMA_TYPE, etc.

**La syntaxe <%%=...%>**
Cette syntaxe ne peut s’étendre sur plusieurs lignes et elle est utilisée dans des instructions SQL. Les points suivants listent ce que vous pouvez faire avec cette syntaxe, et ce à quoi vous devez faire attention.

- Elle peut être utilisée pour générer toute valeur de variable et toute valeur des paramètres existants.
- Les caractères d’espacement sont autorisés après <%%=.
- A l’intérieur de la syntaxe, la syntaxe <%% ...%> ou </.../> ne doit pas être utilisée.

L’instruction écrite dans l’exemple ci-dessous est valide :

```sql
#sql sentence
DROP TABLE temp_<%=__TABLE_NAME__ %>;
```

Le code est utilisé pour supprimer la table définie à l’aide d’un composant associé.

Pour plus d’informations concernant les composants associés aux modèles SQL, consultez Qu’est-ce qu’un Job ? à la page 29.

Pour plus d’informations concernant la syntaxe <%% ...%>, consultez La syntaxe <%% ...%> à la page 531.

Pour plus d’informations concernant la syntaxe </.../>, consultez la section suivante.

**Remarque:**
Les paramètres auxquels les modèles SQL peuvent accéder grâce à cette syntaxe sont simples. Ils sont souvent utilisés lors de connexions et peuvent être facilement définis dans les composants, par exemple TABLE_NAME, DB_VERSION, SCHEMA_TYPE, etc.

**La syntaxe " /> "**
Cette syntaxe ne peut s’étendre sur plusieurs lignes. Les points suivants listent ce que vous pouvez faire avec cette syntaxe, et ce à quoi vous devez faire attention.

- Elle peut être utilisée pour générer la valeur des paramètres existants. La valeur générée ne doit pas être entourée de guillemets.
- Aucun caractère d’espacement n’est autorisé après < ou avant />.
A l'intérieur de cette syntaxe, la syntaxe `<% ... %>` ou `<%= ... %>` ne doit pas être utilisée.

L'instruction écrite dans l'exemple ci-dessous est valide :

```sql
#sql sentence
DROP TABLE temp_</TABLE_NAME/>;
```

L'instruction accède au paramètre `TABLE_NAME` et supprime la table correspondante.

Pour plus d'informations concernant la syntaxe `<% ... %>`, consultez La syntaxe `<%...%>` à la page 531.

Pour plus d'informations concernant la syntaxe `<%=...%>`, consultez La syntaxe `<%=...%>` à la page 532.

Les sections suivantes présentent un code plus spécifique pour accéder à des paramètres plus complexes.

**Remarque:**

Les paramètres auxquels les modèles SQL peuvent accéder grâce à cette syntaxe sont simples. Ils sont souvent utilisés lors de connexions et peuvent être facilement définis dans les composants, par exemple `TABLE_NAME`, `DB_VERSION`, `SCHEMA_TYPE`, etc.

### Le code pour accéder aux éléments du schéma des composants

Les éléments du schéma des composants sont présentés dans une liste comprenant le nom des colonnes des schémas (séparés par un point "."). Ces éléments sont créés et définis par les utilisateurs dans les composants.

Le code ci-dessous propose un exemple d'accès à certains éléments du schéma des composants. Dans cet exemple, le nom de la variable `ELT_METADATA_SHEMA` est utilisé pour obtenir le schéma du composant.

```jsp
<% String query = "select ";
    SCHEMA(__ELT_METADATA_SHEMA__); 
    for (int i=0; i < __ELT_METADATA_SHEMA__.length ; i++) {
        query += (__ELT_METADATA_SHEMA__[i].name + ","); 
    }
    query += " from " + __TABLE_NAME__; 
%>
<%query %>
```

Dans cet exemple, selon vos objectifs, le code `__ELT_METADATA_SHEMA__[i].name` peut être remplacé par `__ELT_METADATA_SHEMA__[i].dbType`, `__ELT_METADATA_SHEMA__[i].isKey`, `__ELT_METADATA_SHEMA__[i].length` ou `__ELT_METADATA_SHEMA__[i].nullable` pour accéder aux autres champs des colonnes du schéma.

L'instruction extraite est `SCHEMA(__ELT_METADATA_SHEMA__);`. Dans cette instruction, `ELT_METADATA_SHEMA` est le nom de la variable représentant les paramètres du schéma à extraire. Le nom utilisé dans ce code est simplement un exemple. Vous pouvez changer ce nom en nom d'une autre variable, afin de représenter les paramètres de schéma, selon votre système de nommage.
Avertissement:
Vérifiez que le nom que vous saisissez n'entrera en conflit avec aucun autre paramètre.

Pour plus d'informations concernant les schémas des composants, consultez Onglet Basic settings à la page 55.

**Le code pour accéder aux propriétés de la matrice du composant**

Les propriétés de la matrice du composant sont créées et modifiées par les utilisateurs selon les différents objectifs de transformation des données. Par exemple, les paramètres `operation` ou `groupby` peuvent être définis par les utilisateurs dans le composant `tSQLTemplateAggregate`.

Vous pouvez accéder à ces paramètres de transformation, naturellement plus flexibles et complexes, de deux façons différentes.

- L'approche `<.../>`.
  
  `<.../>` est l'une des syntaxes utilisées par les modèles SQL. Cette approche nécessite souvent du code dur pour chaque paramètre à extraire.

  Par exemple, un nouveau paramètre est créé par l'utilisateur et nommé `NEW_PROPERTY`. Si vous souhaitez y accéder en utilisant `<NEW_PROPERTY/>`, le code ci-dessous est requis.

```java
else if (paramName.equals("NEW_PROPERTY")) {
    List<Map<String, String>> newPropertyTableValue = (List<Map<String, String>>)
        ElementParameterParser.getObjectValue(node, "__NEW_PROPERTY__");
    for (int ii = 0; ii < newPropertyTableValue.size(); ii++) {
        Map<String, String> newPropertyMap = newPropertyTableValue.get(ii);
        realValue += ...;//append generated codes
        ...
    }
}
```

- L'approche `EXTRACT(__GROUPBY__)`.

  Le code ci-dessous montre la seconde méthode d'accès au paramètre de transformation `GROUPBY`.

```java
<%=
String query = "insert into __TABLE_NAME__ (id, name, date_birth) select sum(id), name, date_birth from cust_teradata group by";
    EXTRACT(__GROUPBY__);
    for (int i=0; i < __GROUPBY_LENGTH__; i++) {
        query += (__GROUPBY_INPUT_COLUMN__[i] + " ");
    }

<%=
query %>
```

Lors de l'encodage des instructions, respectez les règles comme suit :

- L'instruction extraite doit utiliser `EXTRACT(__GROUPBY__)`; Les majuscules sont requises, et les caractères d'espacement ne sont pas autorisés. Cette instruction doit être utilisée entre `<%` et `%>`.

- Utilisez le code `__GROUPBY_LENGTH__`, dans lequel le nom du paramètre, suivi de _LENGTH_, permet d'obtenir le numéro de ligne des paramètres du tableau `GROUPBY` définis dans la zone `Groupby` de la vue `Component`. Il peut être utilisé entre `<%` et `%>` ou entre `<%=` et `%>`.

- Utilisez le code `__GROUPBY_INPUT_COLUMN__[i]` afin d'extraire les valeurs des paramètres. Il peut être utilisé entre `<%` et `%>` ou entre `<%=` et `>%`.
• Pour accéder aux paramètres correctement, n’utilisez pas le même préfixe pour le nom de plusieurs paramètres. Par exemple, dans un composant, évitez de définir deux paramètres avec le nom PARAMETER_NAME et PARAMETER_NAME_2, car plusieurs préfixes semblables produisent des erreurs lors de la génération du code.