



Talend Open Studio

Guide Utilisateur

Version 2.3_c

Convient pour l'utilisation de **Talend Open Studio** v2.3.x.

Copyright

Une copie de la licence GNU Free Documentation License est fournie avec les fichiers source de cette documentation.

Talend Open Studio

Guide Utilisateur..... i

A propos de ce manuel...	1
Historique des versions	1
Remarques et Support	1
A propos de l'intégration de données...	1
Intégration décisionnelle avec Talend	
Open Studio 2	
Intégration opérationnelle avec Talend	
Open Studio 3	
Monitoring de l'exécution	4
 Présentation de Talend Open Studio	 5
Accès à Talend Open Studio	5
Connexion au référentiel local	6
Créer un projet	8
Description de l'interface utilisateur	9
Barre d'outils et Menus	10
Barre d'outils	10
Menus	11
Repository	11
Business Models	12
Job Designs	12
Contexts	12
Code	12
Routines	13
Documentation	13
Metadata	13
Recycle bin	13
Graphical workspace	14
Palette	14
Déplacer la palette	15
Modifier l'affichage et les paramètres de la palette	15
Onglets de configuration	15
Component	15
Properties	15
Problems	16
Run Job	16
Error Log	16

Modules	16
Open Scheduler	17
Stats & Logs	18
Outline et Code Viewer	19
Outline	19
Code viewer	19
Configuration de Talend Open Studio	20
Interpréteurs Perl/Java	20
Status	21
Composants externes ou utilisateur	22
 Conception d'un Business Model	 25
Objectifs	25
Créer ou ouvrir un Business Model	25
Ouvrir un Business Model	26
Créer un Business Model	26
Modélisation d'un Business Model	27
Formes (Shapes)	28
Connexions	29
Commenter et réarranger un Business Model	31
Ajouter une note ou du texte libre	31
Réarranger une vue métier	32
Propriétés	33
Rulers and Grid	33
Appearance	34
Assignment	34
Affectation d'éléments du Repository à un Business Model	35
Modifier un Business Model	36
Renommer un Business Model	36
Copier et coller un Business Model	36
Déplacer un Business Model	37
Supprimer un Business Model	37
Enregistrer un Business Model	37
 Conception d'un Job	 39
Objectifs	39
Créer ou Ouvrir un Job	39
Ouvrir un job	40
Créer un job	40
Conception d'un Job	41
Afficher, cacher et déplacer la palette	41
Cliquer & Déposer des composants	42
Cliquer & Déposer des composants du	

répertoire Metadata	43		
Ajouter une Note au Job	44		
Réorganiser les onglets d'un Job	45		
Avertissements et erreurs sur un composant	46		
Connexion des composants	46		
Types de connexion	46		
Connexion de type Row	46		
Main	47		
Lookup	47		
Filter	48		
Rejects	48		
Output	48		
Uniques/Duplicates	49		
Connexion de type Iterate	49		
Connexion de type Trigger	49		
Connexion de type Link	51		
Job à entrées/sorties multiples	51		
Définition des propriétés d'un composant	51		
Basic Settings	52		
Paramétrage d'un schéma built-in	53		
Paramétrage d'un schéma repository	53		
Affichage dynamique d'un champ (Ctrl+Espace)	54		
Advanced settings	55		
View	56		
Documentation	56		
Définition du composant Start	57		
Définition des schémas de métadonnées	58		
Configurer une connexion DB	59		
Etape 1 : Propriétés générales	59		
Etape 2 : Connexion	60		
Etape 3 : Chargement de la table	62		
Etape 4 : Définition du schéma	62		
Configurer un schéma File Delimited	63		
Etape 1 : Propriétés générales	63		
Etape 2 : Chargement du fichier	63		
Etape 3 : Définition du schéma	64		
Etape 4 : Schéma final	67		
Configurer un schéma File Positional	68		
Etape 1 : Propriétés générales	69		
Etape 2 : Connexion et chargement du fichier	69		
Etape 3 : Affinage du schéma	70		
Etape 4 : Schéma final	70		
Configurer un schéma File Regex	71		
Etape 1 : Propriétés générales	71		
Etape 2 : Chargement du fichier	71		
Etape 3 : Définition du schéma	72		
Etape 4 : Schéma final	73		
Configurer un schéma File LDIF	73		
Etape 1 : Propriétés générales	74		
Etape 2 : Chargement du fichier	74		
Etape 3 : Définition du schéma	74		
Etape 4 : Schéma final	75		
Configurer un Schéma FileXML	75		
Etape 1 : Propriétés générales	76		
Etape 2 : Chargement du fichier	76		
Etape 3 : Définition du schéma	76		
Etape 4 : Schéma final	79		
Configurer un schéma LDAP	80		
Etape 1 : Propriétés générales	80		
Etape 2 : Connexion au serveur	80		
Etape 3 : Authentication et acquisition du DN	81		
Etape 4 : Définition du schéma	82		
Etape 5 : Schéma final	83		
Configurer un schéma générique	84		
Etape 1 : Propriétés générales	85		
Etape 2 : Définition du schéma	85		
Configurer un schéma WSDL	85		
Etape 1 : Propriétés générales	85		
Etape 2 : URI et définition des méthodes	85		
Etape 3 : Schéma final	86		
Créer une requête à l'aide de SQLBuilder	87		
Comparaison des structures de base de données	89		
Construction d'une requête	89		
Stocker une requête dans le Repository	91		
Mapping de flux de données dans un Job	92		
Présentation du fonctionnement tMap	92		
Interface tMap	93		
Configurer le flux d'entrée du tMap	95		
Renseigner les tables Input à l'aide d'un schéma	95		

95	Contenu des tables Main et Lookup	StoreSQLQuery 120
	Variables 96	Centraliser les contextes dans le Repository 120
	Jointure explicite 96	Exécuter un Job dans un contexte défini 122
	Unique Match (java) 98	Exécuter un job 123
	First ou Last Match (java) 98	Exécuter en mode normal 124
	All Matches (java) 98	Afficher les Statistiques 124
	Inner join 98	Afficher les Traces 125
	All rows (java) 99	Afficher la durée d'exécution et d'autres options 126
	Filtrer un flux d'entrée (java) 99	Exécuter en mode debug 126
	Retirer des entrées de la table Input 100	Enregistrer ou exporter un Job 127
	Mapping de variables 100	Enregistrer un Job 127
	Accéder aux variables globales et de contexte 102	Exporter des job scripts 127
	Supprimer des variables 102	Documentation du Job 128
	Paramètres Output 102	Générer la documentation HTML .. 128
	Construire des expressions complexes	Définir les options dans la vue Job Settings 128
103	Filtres 103	Automatiser l'utilisation de statistiques & logs 129
	Rejets 104	Fonctions Extra 130
	Rejets Inner Join 105	Raccourcis clavier 130
	Retirer des entrées de la table Output	
105	Expression editor 105	
	Schema editor 106	
	Ecrire du code à l'aide du Expression Builder 107	
	Activer/Désactiver un job ou un sous-job 111	
	Désactiver un composant Start 112	
	Désactiver un composant non-Start 112	
	Définir les Contextes et les Variables 112	
	Utiliser les variables dans un Job .. 112	
	Définir des variables à partir de la vue Contexts 113	
	Variables 113	
	Values as tree 114	
	Values as table 116	
	Configurer les contextes 117	
	Créer un contexte 117	
	Renommer et éditer un contexte 118	
	Définir les variables de contexte à partir de la vue Component 118	
	Création rapide des variables de contexte 119	
		De la théorie à la pratique : Exemple de Job 131
		Présentation du scénario 131
		Données en entrée 131
		Données en sortie 131
		Données de référence 132
		Du scénario au Job 133
		Etape 1 : Création du Job, définition des données d'entrée, lecture du fichier 133
		Etape 2 : Mapping et transformations 136
		Etape 3 : Définition du fichier de référence, mapping des données de référence, sélection du mode Inner Join 138
		Etape 4 : Sortie vers une table MySQL 140
		Gestion des Jobs & Projets 143
		Importer des projets 143
		Importer des échantillons (Demos) 144
		Importer des éléments 145

Exporter des projets	148	Supprimer un projet	155
Exporter des job scripts	149	Déployer un job sur un serveur SpagoBI	
Exporter un job en Java	151	156	
Exporter un job en tant que POJO	151	Créer un nouveau serveur SpagoBI	157
Exporter un Job en tant que Webservice		Editer ou supprimer un serveur SpagoBI	
152		158	
Exporter un job en Perl	153	Déployer vos jobs sur le serveur SpagoBI	
Changer les paramètres de contexte d'un		158	
job script	154	Tâches de migration	159
Exporter des éléments	154		

A propos de ce manuel...

Ce manuel s'adresse aux utilisateurs et administrateurs de **Talend Open Studio**.

Historique des versions

Le tableau ci-dessous liste les modifications apportées au Guide Utilisateur.

Version	Date	Historique
v 0.5	25/10/06	Distribution de la version française du Guide Utilisateur Talend Open Studio .
v 0.5.5	21/01/07	Suite de la distribution de Talend Open Studio version 1.1.0 Mise à jour du Guide Utilisateur version 0.5.5 Ajout des composants de Talend Open Studio v1.0 Nouveaux composants de Talend Open Studio v1.1
v0.7	05/03/07	Mise à niveau des composants de la version 0.7 du Guide Utilisateur anglais
v0.8	18/05/07	Distribution de la version française du Guide Utilisateur de Talend Open Studio version 2.0 Nouveaux composants v2.0 ajoutés Refonte des chapitres existants Ajout d'information pour Perl et Java Ajout du chapitre Gestion de Projets
v2.3_b	19/03/08	Mise à niveau du Guide Utilisateur de Talend Open Studio version 2.3.x : Inlus informations générales, utilisation du Studio.
v2.3_c	05/05/08	Mise à niveau du Guide Utilisateur de Talend Open Studio version 2.3.x. Séparation en deux guides : <ul style="list-style-type: none">- Guide Utilisateur Studio- Guide de référence des composants

Remarques et Support

N'hésitez pas à nous faire part de vos remarques, suggestions, requêtes concernant la documentation ou le produit, et obtenez le support de notre équipe **Talend**, sur le Forum **Talend** (en anglais) :

<http://talendforge.org/forum>

A propos de l'intégration de données...

Il est bien connu que les systèmes d'information des entreprises tendent à se complexifier. Les raisons sont les suivantes : une tendance à "l'empilage de couches" (une nouvelle application est déployée bien que l'ancien système est maintenu), le fait que les systèmes d'information s'ouvrent de plus en plus vers

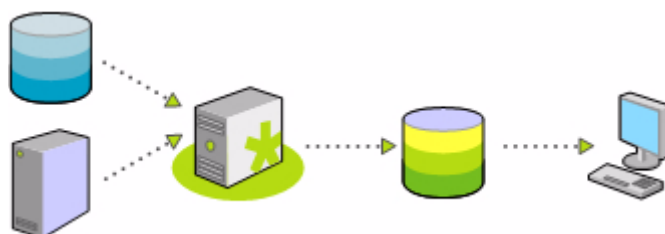
ceux de leurs fournisseurs, partenaires et clients, et la multiplication des formats de stockage de données (fichiers XML, plats positionnels, plats délimités, multi-valués, etc.), protocoles (FTP, HTTP, SOAP, SCP...) et les technologies des bases de données.

On se pose donc la question suivante : Comment procéder à une intégration de données réussie de ces données éparpillées dans tous ces systèmes d'information ? Plusieurs fonctions sont cachées derrière le principe d'intégration données : l'intégration décisionnelle (alimentation d'entrepôts de données) et l'intégration opérationnelle (capture et migration de données, synchronisation de bases de données, échanges inter-applicatifs, etc.).

Ces deux besoins ETL d'intégration décisionnelle et opérationnelle sont fournis par **Talend Open Studio**.

Intégration décisionnelle avec Talend Open Studio

Bien qu'ils soient généralement invisibles pour les utilisateurs de la plate-forme BI, les processus ETL reprennent les données de tous les systèmes opérationnels et les pré-traitent pour les outils d'analyse et de reporting.



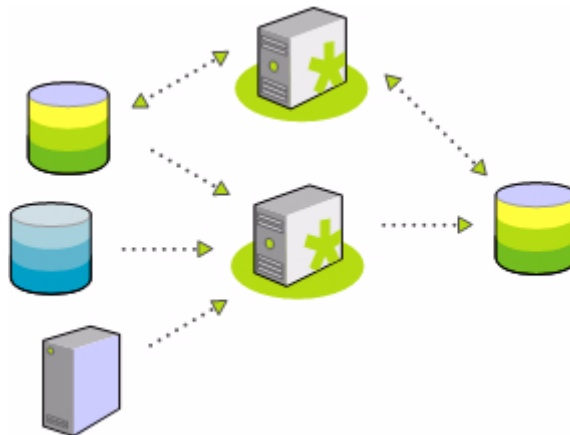
Talend Open Studio offre une très grande connectivité aux :

- Progiciels (ERP, CRM, etc.), bases de données, serveurs centraux, fichiers, Web Services, etc. pour couvrir la disparité grandissante des sources.
- Entrepôts de données, magasins de données, applications OLAP (Online Analytical Processing) - pour analyse, reporting, dashboarding, scorecarding, etc.
- Composants ETL avancés stockés localement, incluant des manipulations de chaînes, Dimensions à évolution lente, traitement automatique des références, support de charges en vrac, etc.

La plupart des connecteurs dont il est question ci-dessus sont détaillés dans les *Composants Talend Open Studio Guide de référence*. Pour plus d'informations sur leur fonctionnement dans **Talend Open Studio**, consultez *Conception d'un Job*, page 39. Pour la modélisation de processus métier, consultez *Conception d'un Business Model*, page 25.

Intégration opérationnelle avec Talend Open Studio

L'intégration opérationnelle de données est souvent utilisée pour implémenter les programmes et routines habituels, complétée en fonction de besoins spécifiques.



Les applications de chargement/migration de données et de synchronisation/réplication de données sont les plus répandues en matière d'intégration opérationnelle de données. Elles requièrent :

- des mappings et transformations complexes avec fonctions d'agrégation, calculs, etc. pour palier les différences dans la structure des données.
- le traitement et la résolution des conflits de données en tenant compte des mises à jour des enregistrements ou des "propriétaires des enregistrements".
- la synchronisation de données en quasi temps réel étant donné que les systèmes impliquent une latence lente.

La plupart des connecteurs dont il est question ci-dessus sont détaillés dans les *Composants Talend Open Studio Guide de référence*. Pour plus d'informations sur leur fonctionnement dans **Talend**

Open Studio, consultez *Conception d'un Job*, page 39. Pour la modélisation de processus métier, consultez *Conception d'un Business Model*, page 25.

Monitoring de l'exécution

Un des plus grands challenges des développeurs de processus d'intégration et des responsables informatiques chargés de contrôler leur exécution, est de contrôler et monitorer l'exécution de ces processus "mission critique". En effet, la gestion et la notification d'erreurs peuvent et devraient être inclus dans l'intégration de données.

De plus, au delà des notifications d'erreurs, il est aussi crucial de monitorer l'intégrité des processus d'intégration et de surveiller toute dégradation des performances.

La console **Activity Monitoring Console** surveille l'exécution du job (succès, échecs, avertissements, etc.), le temps d'exécution et les volumes de données, tout ceci en une seule console, disponible en environnement indépendant.

Pour plus d'informations concernant le fonctionnement de l'**Activity Monitoring Console**, consultez le **Guide Utilisateur de l'Activity Monitoring Console**.

—Présentation de Talend Open Studio—

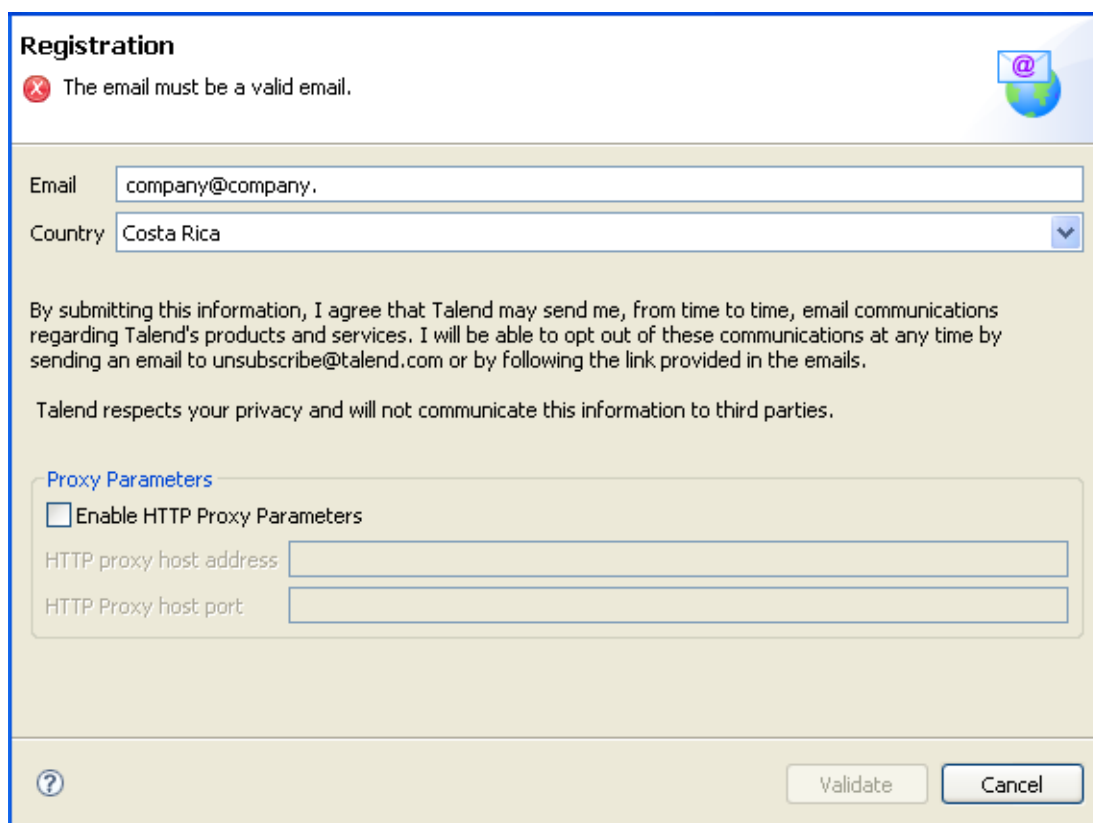
Présentation de Talend Open Studio

Accès à Talend Open Studio

L'assistant Setup vous aide à installer l'exécutable **Talend Open Studio**. Si vous le décompressez manuellement, des instructions d'installation sont disponibles.

Lisez et acceptez les termes de la licence pour continuer.

Le formulaire d'enregistrement **Talend Open Studio** s'affiche. Remplissez votre adresse électronique et votre lieu de résidence pour recevoir des informations sur **Talend Open Studio**. Cette étape est facultative, cliquez sur **Cancel** si vous ne souhaitez pas la renseigner.



The image shows a 'Registration' dialog box with a title bar. Inside, there's a red error icon and text: 'The email must be a valid email.' in the top left. In the top right is an '@' icon over a globe. Below this, there's an 'Email' text box containing 'company@company.' and a 'Country' dropdown menu showing 'Costa Rica'. A paragraph of text follows: 'By submitting this information, I agree that Talend may send me, from time to time, email communications regarding Talend's products and services. I will be able to opt out of these communications at any time by sending an email to unsubscribe@talend.com or by following the link provided in the emails.' Below that is another line: 'Talend respects your privacy and will not communicate this information to third parties.' Then, a section titled 'Proxy Parameters' contains a checkbox 'Enable HTTP Proxy Parameters' which is currently unchecked. Below the checkbox are two text boxes: 'HTTP proxy host address' and 'HTTP Proxy host port'. At the bottom left is a help icon '?'. At the bottom right are two buttons: 'Validate' and 'Cancel'.

Cochez **enable HTTP Proxy parameters** si votre situation le requiert et renseignez les informations de proxy appropriées. Veillez à fournir votre adresse e-mail si vous renseignez les informations de proxy.

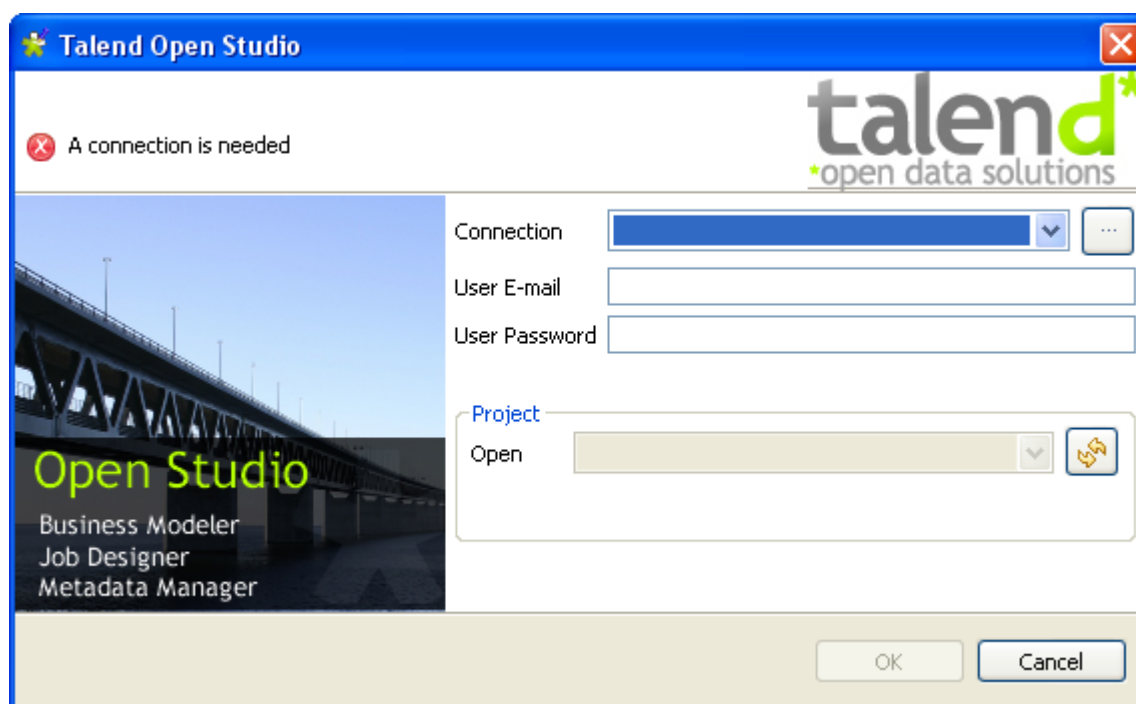
Vous pouvez accéder à ces informations d'enregistrement et les modifier à tout moment en cliquant sur *Window > Preference > Talend > Install/Update*.

WARNING—Soyez assuré que toutes les informations personnelles que vous fournissez à Talend ne sont pas transmises à des tiers et ne sont pas utilisées à d'autres fins que de vous informer sur Talend et les produits Talend.

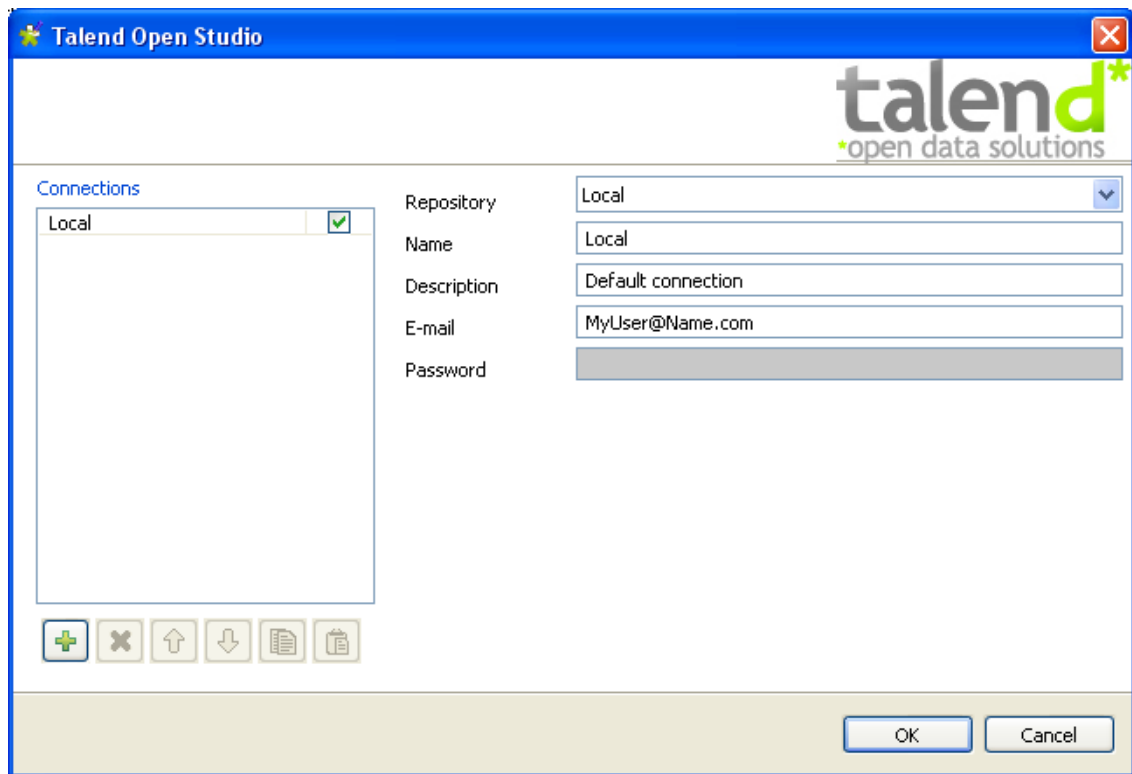
Vous accédez ensuite à la fenêtre de Login **Talend Open Studio**.

Connexion au référentiel local

Connectez-vous à **Talend Open Studio** à partir de la fenêtre de Login.



- Sélectionnez l'entrée correspondant à votre connexion dans la liste **Connection**, si vous l'avez déjà configurée.
- Si vous vous connectez pour la première fois, cliquez sur le bouton [...] pour saisir vos informations de connexion.



- Pour ajouter une nouvelle Connexion de référentiel, cliquez sur le bouton (+) de la vue de gauche.
- Saisissez l'adresse e-mail qui servira de login utilisateur (**user login**). Ce champ est obligatoire pour l'utilisation de **Talend Open Studio**. Sachez que l'e-mail ne sera jamais utilisé pour un autre usage que la connexion.
- Renseignez le champ **Password** si nécessaire. Ce champ est grisé lorsque la connexion est locale.
- Cliquez sur **OK** pour valider.

Cliquez sur le bouton **Refresh** pour mettre à jour la liste des projets, si nécessaire. Puis choisissez le nom du projet que vous souhaitez ouvrir et cliquez sur **OK**.

Si vous avez déjà créé des projets avec une version antérieure de **Talend Open Studio**, vous pouvez les importer dans la version courante de **Talend Open Studio** à l'aide du bouton **Import**.

Voir également : *Importer des projets, page 143*

Si vous créez un projet pour la première fois, aucun projet n'est proposé par défaut dans la liste. Cliquez sur le bouton **Create** pour lancer l'assistant de création.

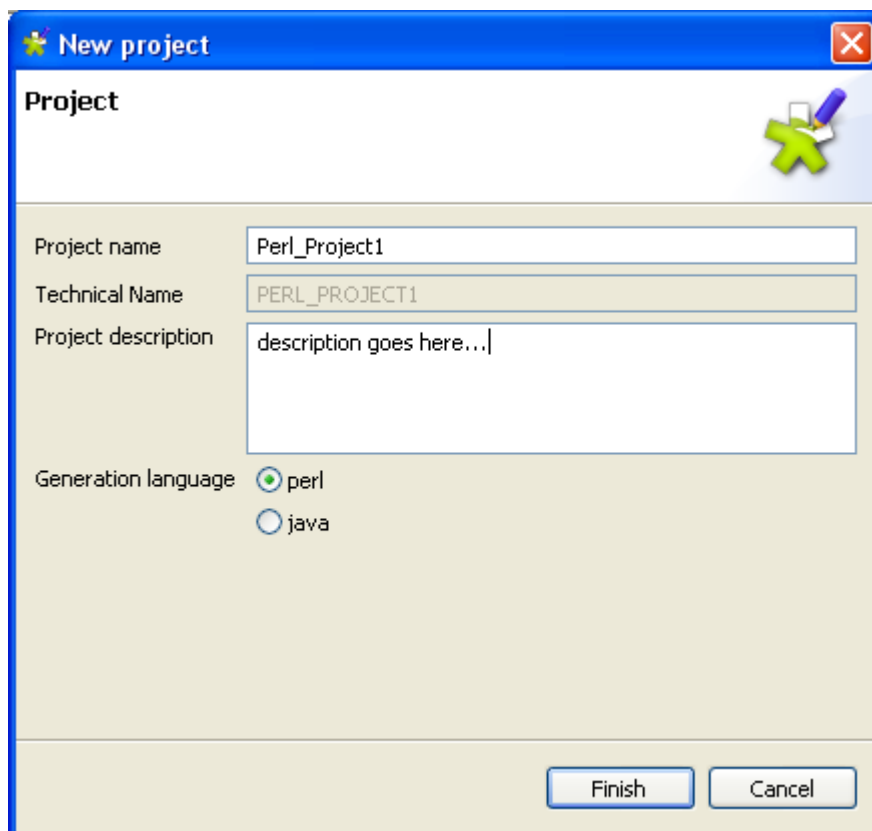
Voir également : *Créer un projet, page 8*

Pour faciliter votre prise en main de **Talend Open Studio** des exemples de jobs sont à votre disposition via le bouton **Import demos**. Le dossier de projet Demos est installé automatiquement sur votre répertoire Workspace. Et ce projet est directement accessible depuis la fenêtre de Login, dans le champ **Projects**.

La création d'un projet met automatiquement en place une arborescence de dossiers dans le répertoire **Workspace** de votre serveur. Celle-ci correspond en tout point à l'arborescence **Repository** qui s'affiche dans la fenêtre principale de **Talend Open Studio**.

Créer un projet

Commencez par donner un **Nom** à votre nouveau projet. Ce champ est obligatoire pour poursuivre le processus de création.



Un message contextuel s'affiche dans la partie supérieure de la fenêtre selon les caractères que vous saisissez ou l'endroit où vous placez votre curseur. Il vous informe de la nature de données attendues, ou le cas échéant, des caractères interdits.

Note: Les chiffres ne sont pas autorisés en début de nom de fichier. Le nom de fichier n'est pas sensible à la casse, par conséquent MonProjet et MONPROJET sont identiques.

Le nom technique, **Technical name**, est utilisé par l'application pour créer le nom du fichier projet, il correspond généralement au nom du projet saisi, mis en majuscule et concaténé à l'aide de tirets soulignés si nécessaire.

Sélectionnez le langage de génération de code, dans le champ **Generation language**, en l'occurrence : **Perl** ou **Java**. Lorsque vous avez choisi le langage, vous devez l'utiliser dans tous les jobs de votre projet : Perl pour les projets perl et Java pour les projets java.

Si vous souhaitez passer d'un projet à un autre, cliquez sur **File > Switch Projects...**

Note: Nous vous déconseillons d'avoir de nombreux projets dans des langages différents et de passer d'un projet Perl à un projet Java trop souvent. Si néanmoins vous souhaitez créer des projets dans les deux langages, veillez à bien les distinguer dans le répertoire Workspace, voire à lancer deux instances **Talend Open Studio** distinctes afin d'éviter tout conflit de langage.

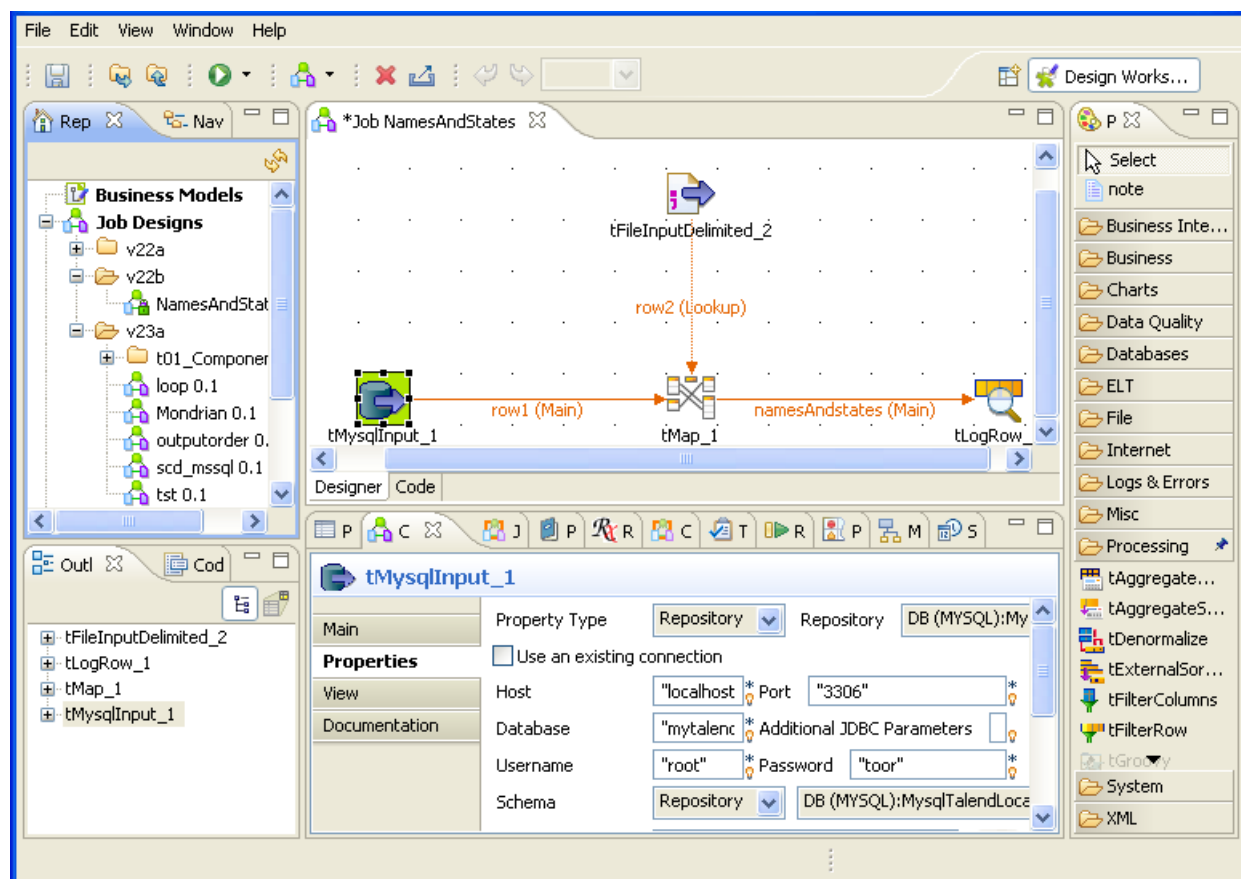
Si vous avez déjà créé un projet dans une version antérieure de **Talend Open Studio** et que vous souhaitez l'importer dans la version courante, consultez *Importer des projets*, page 143.

Dans la fenêtre de Login, sélectionnez le projet que vous souhaitez ouvrir, puis cliquez sur **OK** pour lancer **Talend Open Studio**.

Note: Une fenêtre d'initialisation du générateur apparaît au lancement de l'application. Patientez jusqu'à ce que l'initialisation se termine.

Description de l'interface utilisateur

Talend Open Studio s'ouvre sur une fenêtre à zones multiples.



La fenêtre de **Talend Open Studio** est composée des vues suivantes :

- Barres d'outils et menus
- Référentiel (**Repository**)
- Espace de modélisation (**Graphical designer**)
- Diverses vues de configuration organisées en onglets
- Aperçu schéma (**Outline**) et Aperçu du code (**Code Viewer**)

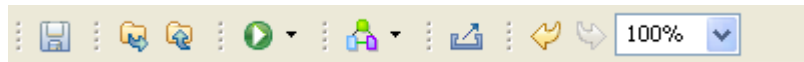
Les différentes vues et leurs fonctionnalités associées sont décrites dans la suite de ce manuel.

Barre d'outils et Menus

En haut de la fenêtre principale de **Talend Open Studio**, les barres d'outils et menus rassemblent les fonctionnalités communes ainsi que certaines fonctionnalités Eclipse.

Barre d'outils

La barre d'outils vous permet un accès rapide aux fonctions les plus courantes. Elle diffère légèrement en fonction de si vous travaillez sur un job design ou un business model.



La barre d'outils d'un job rassemble les fonctionnalités suivantes :

- **Save** : Enregistre le job design courant.
- **Export items** : Exporte les éléments du Repository dans un fichier archive, pour les déployer en dehors de **Talend Open Studio**. Si vous souhaitez importer les éléments exportés dans une version plus récente de **Talend Open Studio** ou d'un autre ordinateur, assurez-vous d'avoir inclu les fichiers source dans l'archive, en cochant la case adéquate.
- **Import items** : Importe les éléments de Repository d'un fichier archive dans la version courant de **Talend Open Studio**. Pour plus d'informations concernant l'Import/Export d'éléments, consultez *Importer des éléments*, page 145.
- **Run Job** : Exécute le job courant ouvert dans l'espace de modélisation. Pour plus d'informations concernant l'exécution d'un job, consultez *Exécuter un job*, page 123.
- **Create** : Lance l'assistant de création. A partir de ce menu, vous pouvez créer n'importe quel élément du Repository : Business models, Job Designs, contextes, routines, métadonnées, etc.
- **Export project** : Lance l'assistant d'Export de projet. Pour plus d'informations concernant l'Export de projet, consultez *Exporter des projets*, page 148.
- **Undo/Redo** : Permet d'annuler ou de répéter une action que vous avez effectuée.
- **Zoom in/out** : Sélectionnez le pourcentage de zoom avant ou arrière sur votre Job.

Menus

Les menus de **Talend Open Studio** permettent d'accéder à :

- la plupart des fonctionnalités standard, notamment Enregistrer (**Save**), Imprimer (**Print**), Quitter (**Exit**), accessibles au niveau de l'application.
- certaines fonctions natives d'Eclipse, accessibles au niveau du **Job Designer** seulement.
- ainsi que certaines fonctions spécifiques **Talend Open Studio**.

Sous **Window > Preferences > Talend**, vous pouvez paramétrer vos préférences. Pour plus d'informations concernant les préférences, consultez *Configuration de Talend Open Studio*, page 20.

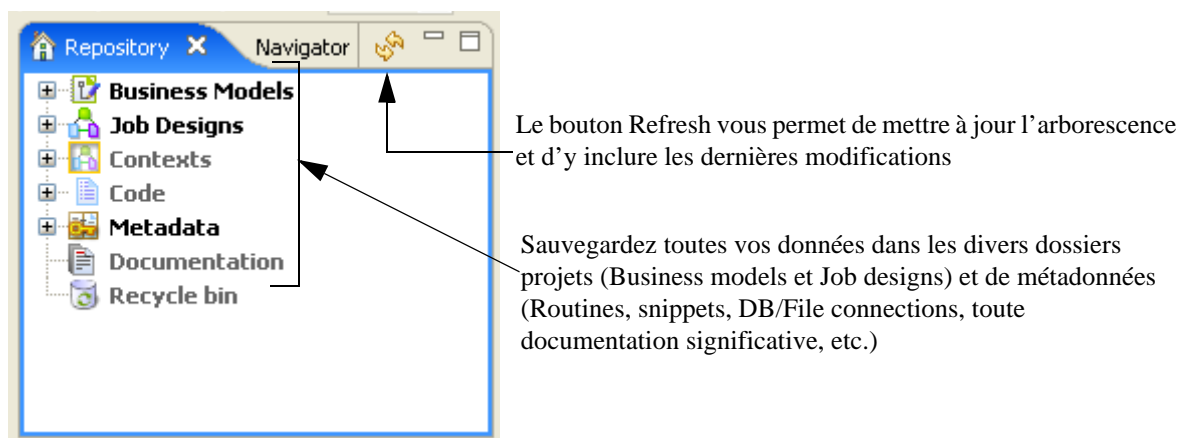
Sous **Window > Show views**, vous pouvez gérer les différentes vues qui s'affichent dans **Talend Open Studio**.

Repository

Le référentiel, **Repository**, regroupe les éléments techniques disponibles pour la description des Business models et la conception des Job designs. Le Repository vous donne accès aux Business

models et Job designs ainsi qu'à toutes les routines et documentations réutilisables pour leur conception.

Ce référentiel centralise et conserve localement tous les éléments contenus dans un projet.



Le repository regroupe les éléments suivants sous forme d'une arborescence :

Business Models

Toutes les représentations graphiques des processus métier d'un projet sont regroupées sous le noeud **Business Models**. Double-cliquez sur le nom pour l'ouvrir dans l'espace de modélisation graphique, aussi appelé workspace ou éditeur.

Voir également : *Conception d'un Business Model, page 25*

Job Designs

Le dossier **Job designs** rassemble tous les jobs du projet courant. Double-cliquez sur le nom d'un job pour l'ouvrir dans l'espace de modélisation.

Voir également : *Conception d'un Job, page 39*

Contexts

Le dossier **Contexts** rassemble les fichiers contenant les variable de contexte que vous souhaitez utiliser dans différents jobs, tels que les chemins d'accès ou les informations de connexion aux bases de données (**DB Connections**).

Voir également : *Définir les Contextes et les Variables, page 112*

Code

La bibliothèque de **Code** rassemble les routines ainsi que les snippets (fonctionnalités à venir) disponibles pour ce projet. Vous pouvez ainsi mutualiser ce code et le réutiliser dans divers composants ou jobs.

Cliquez sur une entrée de l'arborescence pour ouvrir le fichier de code correspondant.

Voir également : *Conception d'un Job, page 39*

Routines

Une **Routine** est un morceau de code plus ou moins complexe généralement utilisé de façon itérative dans un ou plusieurs jobs techniques.

Dans le référentiel, sous **Routines**, un dossier **System** regroupe toutes les routines Talend prédéfinies. Développez ce noeud pour accéder aux différents fichiers de routine tels que **Dates**, **Misc** et **String**.

Double-cliquez sur l'un des fichiers, l'**Editeur de routines** s'ouvre dans un nouvel onglet. Vous pouvez déplacer cet onglet dans l'espace de modélisation en le glissant simplement à l'aide de la souris vers l'emplacement souhaité.

Utilisez ces modèles de routines comme référence pour en construire de nouvelles ou copiez la routine qui vous convient vers la vue **Component** de votre job pour l'utiliser.

Pour créer une nouvelle routine, cliquez droit sur l'entrée **Routines** du Repository, puis sélectionnez **Create a routine** dans le menu contextuel. L'éditeur de routine s'ouvre sur un fichier template contenant par défaut des morceaux de code tels que :

```
sub printTalend {  
  
    print "Talend\n"
```

Remplacez ce modèle par votre propre morceau de code et refermez le fichier pour enregistrer. La nouvelle routine apparaît dans le **Repository** sous le noeud **Routines**.

Vous avez également la possibilité d'organiser vos routines dans des dossiers.

Note: Le dossier **System**, ainsi que son contenu est en lecture seule.

Documentation

Le répertoire **Documentation** regroupe tout type de documents, de tout format, tels que des spécifications ou la description d'un format technique particulier d'un fichier. Double-cliquez pour ouvrir le document dans l'application appropriée.

Voir également : *Générer la documentation HTML*, page 128

Metadata

Le répertoire **Documentation** regroupe tout type de documents, de tout format, tels que des spécifications ou la description d'un format technique particulier d'un fichier. Double-cliquez pour ouvrir le document dans l'application appropriée.

Voir également : *Définition des schémas de métadonnées*, page 58

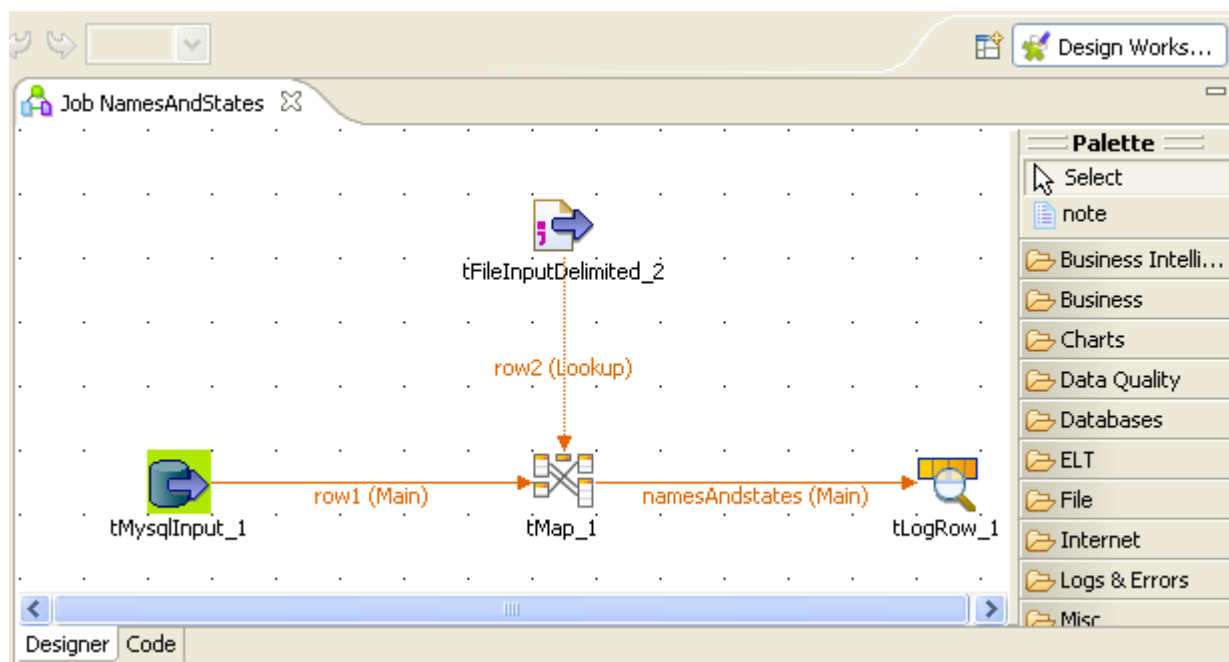
Recycle bin

Glissez et déposez les éléments inutiles ou obsolètes du **Repository** vers la corbeille ou appuyez sur la touche *Suppr* de votre clavier pour les supprimer.

Note: Les éléments supprimés sont conservés sur votre système, dans la corbeille, tant que vous ne la videz pas. Cliquez-droit sur la corbeille et sélectionnez **Empty Recycle bin** pour supprimer définitivement son contenu.

Graphical workspace

Le **Graphical workspace**, ou espace de modélisation, est l'éditeur de graphique des jobs de **Talend Open Studio** qui vous permet de concevoir graphiquement les business models (processus métier) mais également les job designs (processus technique).



Les job designs et business models ouverts dans l'éditeur graphique sont organisés par un système d'onglet.

Une **Palette** est rattachée à l'espace de modélisation pour vous permettre d'accéder rapidement à tous les éléments graphiques (formes et composants techniques) répondant à vos besoins.

Palette

A partir de la **Palette**, selon que vous travaillez sur un job design ou un business model, cliquez et déposez les formes, branches, notes ou composants techniques dans l'espace de modélisation. Puis définissez et formatez les éléments graphiques à l'aide des paramètres de propriétés disponibles dans la vue **Component**.

Voir également :

- *Conception d'un Business Model, page 25*
- *Conception d'un Job, page 39*

Déplacer la palette

Si la **Palette** n'apparaît pas ou si vous souhaitez la détacher de la fenêtre générale, cliquez sur **Window > Show view...> General > Palette**. La palette apparaît dans une vue séparée que vous pouvez déplacer n'importe où dans la fenêtre **Talend Open Studio**.

Modifier l'affichage et les paramètres de la palette

Vous pouvez modifier l'affichage des composants de la palette et les faire apparaître en colonne, liste, icônes uniquement ou avec une courte description.

Pour une meilleure visibilité, vous pouvez aussi agrandir la taille des icônes des composants.

Pour ce faire, cliquez-droit sur un élément de la palette et sélectionnez l'option que vous souhaitez utiliser dans le menu contextuel ou cliquez sur **Settings** pour ouvrir la fenêtre de configuration et modifier l'affichage.

Onglets de configuration

Les onglets **Properties**, **Run Jobs** et **Error Log** regroupent toutes les informations relatives aux éléments graphiques sélectionnés dans l'espace de modélisation ou à l'exécution elle-même du job complet.

Les onglets **Modules** et **Scheduler** se trouvent dans la même zone que les onglets **Properties**, **Logs** et **Run Job**. Ces deux vues sont indépendantes des jobs actifs ou inactifs ouverts dans l'espace de modélisation.

Component

Lorsque vous travaillez sur un Job Design, l'onglet **Component** vous permet de paramétrer les propriétés de vos composants.

Le contenu de l'onglet **Component** varie en fonction de l'élément sélectionné dans l'espace de modélisation.

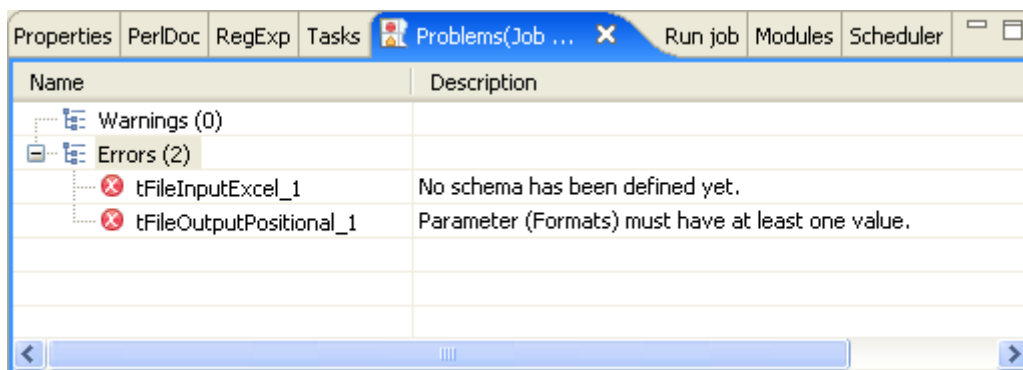
Properties

Lorsque vous travaillez sur une Job Design, l'onglet **Properties** regroupe les propriétés générales des éléments présents dans le Repository. Ces propriétés sont en lecture seule. Si vous souhaitez les modifier, sélectionnez et cliquez droit sur l'élément correspondant puis cliquez sur **Edit**.

Lorsque vous travaillez sur un Business Model, l'onglet **Properties** vous permet mettre en forme les différentes formes que vous avez ajoutées à l'espace de modélisation.

Problems

L'onglet **Problems** affiche tous les messages d'erreur et d'avertissement d'un composant lorsqu'il n'est pas correctement configuré ou qu'il manque une connexion vers un autre composant. Consultez la Description des icones **Error**, **Warning** et/ou **Info**.



Run Job

L'onglet **Run Job**, comme son nom le suggère, affiche l'exécution du job technique courant. Cet onglet joue le rôle d'une console de log pour afficher le résultat d'une exécution.

Pour plus d'informations concernant l'exécution d'un job, consultez *Exécuter un job*, page 123.

Error Log

L'onglet **Error Log** est généralement utile lors de l'exécution des jobs techniques, car il affiche les éventuelles erreurs d'exécution.

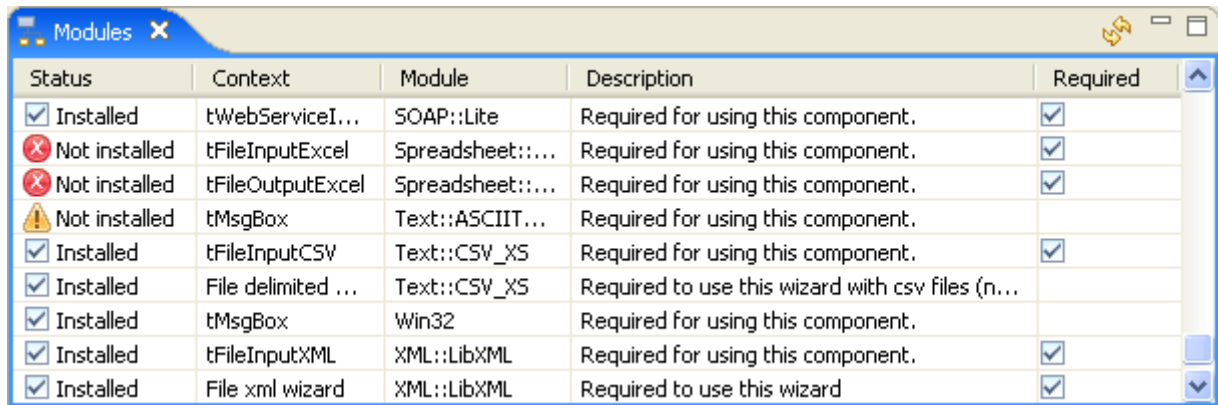
Note: L'onglet **Error Log** a également une fonction informative, notamment pour indiquer qu'un module Perl est en cours de fonctionnement.

Cet onglet ne s'affiche pas par défaut. Pour l'afficher, cliquez sur le menu **Window** > **Show view...** développez le noeud **PDE Runtime** et sélectionnez **Error Log**.

Modules

L'utilisation de certains composants requiert l'installation de modules spécifiques Perl. Consultez la vue **Modules** pour vérifier les modules présents ou manquants pour l'exécution de vos jobs.

Si l'onglet **Modules** ne s'affiche pas dans votre workspace, cliquez sur le menu **Window** > **Show View...** > **Talend**, puis sélectionnez **Modules**.



Status	Context	Module	Description	Required
✓ Installed	tWebServiceI...	SOAP::Lite	Required for using this component.	✓
✗ Not installed	tFileInputExcel	Spreadsheet::...	Required for using this component.	✓
✗ Not installed	tFileOutputExcel	Spreadsheet::...	Required for using this component.	✓
⚠ Not installed	tMsgBox	Text::ASCIIT...	Required for using this component.	
✓ Installed	tFileInputCSV	Text::CSV_XS	Required for using this component.	✓
✓ Installed	File delimited ...	Text::CSV_XS	Required to use this wizard with csv files (n...	
✓ Installed	tMsgBox	Win32	Required for using this component.	
✓ Installed	tFileInputXML	XML::LibXML	Required for using this component.	✓
✓ Installed	File xml wizard	XML::LibXML	Required to use this wizard	✓

Cette vue indique quel module est requis pour l'utilisation du composant correspondant.

La colonne **Status** indique si les modules répertoriés sont ou non installés sur votre système. Le triangle d'avertissement montre que le module est recommandé, mais pas requis pour l'utilisation de ce composant.

Par exemple, le module *DBD::Oracle* n'est requis que si vous utilisez *tDBSQLRow* avec une base Oracle. De la même façon, *DBD::Pg* n'est requis que pour l'utilisation de PostgreSQL. Mais tous peuvent devenir nécessaires dans leur contexte d'utilisation respectif.

Le cercle rouge à croix blanche signifie que le module est indispensable à l'utilisation du composant en regard.

Si la colonne **Context** n'est pas renseignée pour un module particulier, ce module est alors requis pour l'utilisation générale de **Talend Open Studio**.

Pour installer les modules Perl manquants, consultez le guide d'installation (en anglais) sur <http://talendforge.org/wiki/>

Open Scheduler

Open Scheduler est un outil de planification d'exécution basé sur la commande *crontab*, couramment utilisée sous Unix et systèmes d'exploitation similaires. Cette commande *cron* peut également être installée sous Windows.

Open Scheduler génère des entrées compatibles *cron* vous permettant d'exécuter une tâche, par exemple : lancer un job à une échéance définie via le *crontab*.

Si l'onglet **Scheduler** n'apparaît pas dans votre workspace, cliquez sur le menu **Window > Show View... > Talend**, et sélectionnez **Scheduler**.

Open Scheduler - Add a task

Talend Jobs | Command

Project: ETLProject1 Job: /Connections Context: Default

Day of month

1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31

Everyday

Month

January February March April May
June July August September October
November December

Day of week

Monday Tuesday Wednesday
Thursday Friday Saturday
Sunday

Hours/minutes

Hours AM: 0 1 2 3 4 5 6 7 8 9 10 11
PM: 12 13 14 15 16 17 18 19 20 21 22 23
Minutes: 0 5 10 15 20 25 30 35 40 45 50 55

☒ Generated crontab entry

Minute	Hour	Month	Day	Weekday	Command
0	0,13	6	*	1	C:\Perl\bin\perl.exe -IC:\T

☐ Custom crontab entry

0 0,13 * 6 1 C:\Perl\bin\perl.exe -IC:\Talend builds\JasperETL-Win32-r1732-V1.1.1\JasperETL-Win32-r1732-V1

Copy

Add this entry Cancel

Paramétrez l'heure et la date complète d'exécution de cette tâche.

Open Scheduler génère automatiquement la commande correspondante qui sera fournie au programme *crontab*.

Stats & Logs

L'onglet **Stats & Log**, situé dans la vue **Job Settings**, vous permet d'activer ou de désactiver les statistiques et logs de votre job.

Vous pouviez déjà utiliser ces fonctions pour chaque composant de votre job en utilisant et configurant les composants adéquats : **tFlowMeterCatcher**, **tStatCatcher**, **tLogCatcher**.

Pour plus d'informations concernant les composants, consultez les *Composants Talend Open Studio Guide de référence*.

Dorénavant, vous pouvez appliquer ces fonctionnalités à la totalité de votre job actif (i.e. tous les composants de votre job) en une seule fois, sans utiliser les composants Catcher mentionnés plus haut. Ainsi, tous les composants sont récupérés et transférés dans un fichier log ou une table de données en fonction de votre job.

Vous pouvez aussi définir les paramètres courant comme préférences en cliquant sur le bouton **Save to preferences**.

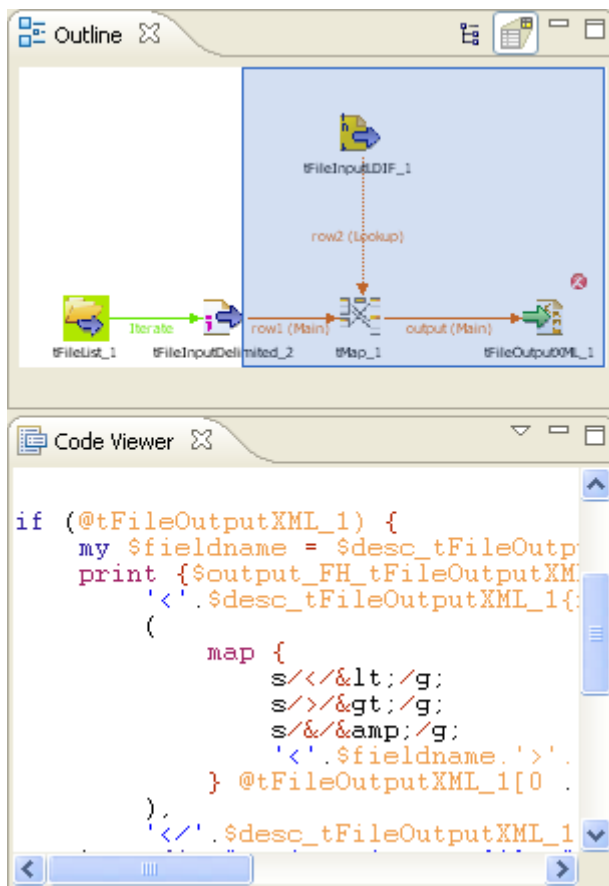
Pour plus d'informations concernant le fonctionnement automatique de Stats & Logs, consultez *Automatiser l'utilisation de statistiques & logs*, page 129.

Outline et Code Viewer

La zone Information est composée de deux onglets, **Outline** et **Code Viewer**, qui fournissent un aperçu du diagramme affiché dans le workspace (qu'il s'agisse d'un job design ou d'un business model).

Outline

L'onglet **Outline** offre un aperçu rapide du business model ou du job design ouvert dans l'espace de modélisation. En cliquant sur l'icône **Arborescence**, vous pouvez accéder aux variables disponibles pour les composants sélectionnés. Comme toutes les zones de la fenêtre générale de **Talend Open Studio**, la zone Outline peut être déplacée ou redimensionnée selon vos besoins. Cet aperçu vous permet d'identifier quelle partie du job design ou du business model est visible dans l'espace de modélisation.



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.

Remarquez que la vue du workspace se déplace de la même manière.

L'aperçu **Outline** peut également afficher les variables disponibles pour les composants utilisés dans le diagramme. En mode **Arborescence**, développez le noeud d'un composant pour afficher les variables appropriées disponibles.

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

Code viewer

L'onglet **Code viewer** donne un aperçu de lignes de code générées pour le composant affiché dans l'aperçu. Ce code est généré en arrière-plan du job et comprend les éléments Start, Body et End correspondant aux phases d'exécution de ce composant.

Note: Cette vue ne concerne que le code produit dans un job design. 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.

Configuration de Talend Open Studio

Vous pouvez définir les propriétés d'affichage de la fenêtre principale de **Talend Open Studio** pour les adapter à vos besoins.

Un grand nombre des configurations que vous avez définies peuvent être enregistrées dans **Preference** et seront donc utiliser par défaut pour tous les nouveaux que vous créez.

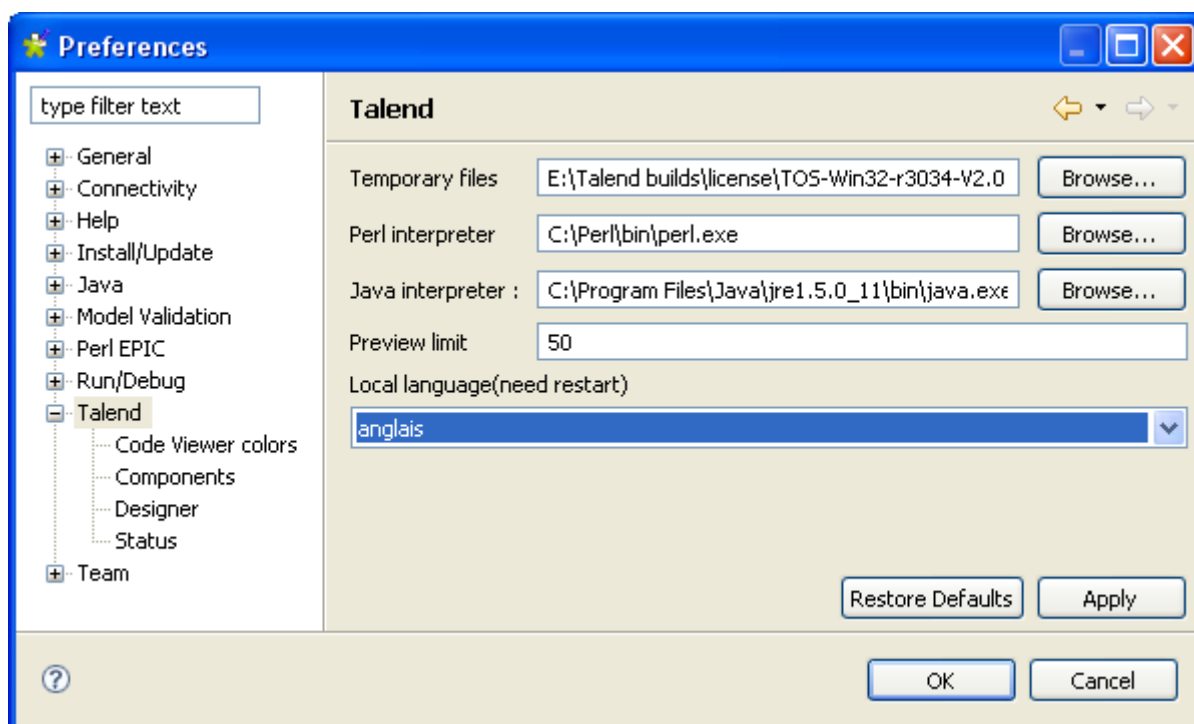
La section suivante décrit les configurations spécifiques que vous pouvez définir en tant que préférences.

Cliquez sur le menu **Window** de **Talend Open Studio**, puis sélectionnez **Preferences**.

Interpréteurs Perl/Java

Dans les préférences, il vous faudra peut-être faire pointer **Talend Open Studio** vers l'interpréteur.

- Si nécessaire, cliquez sur **Talend** dans l'arborescence **Preferences**.
- Si le répertoire d'installation Perl par défaut n'est pas correct, saisissez le chemin vers Perl.



Dans la même vue, vous pouvez également modifier la limite de l'aperçu et le chemin d'accès vers les fichiers temporaires ou la langue de votre système d'exploitation.

Status

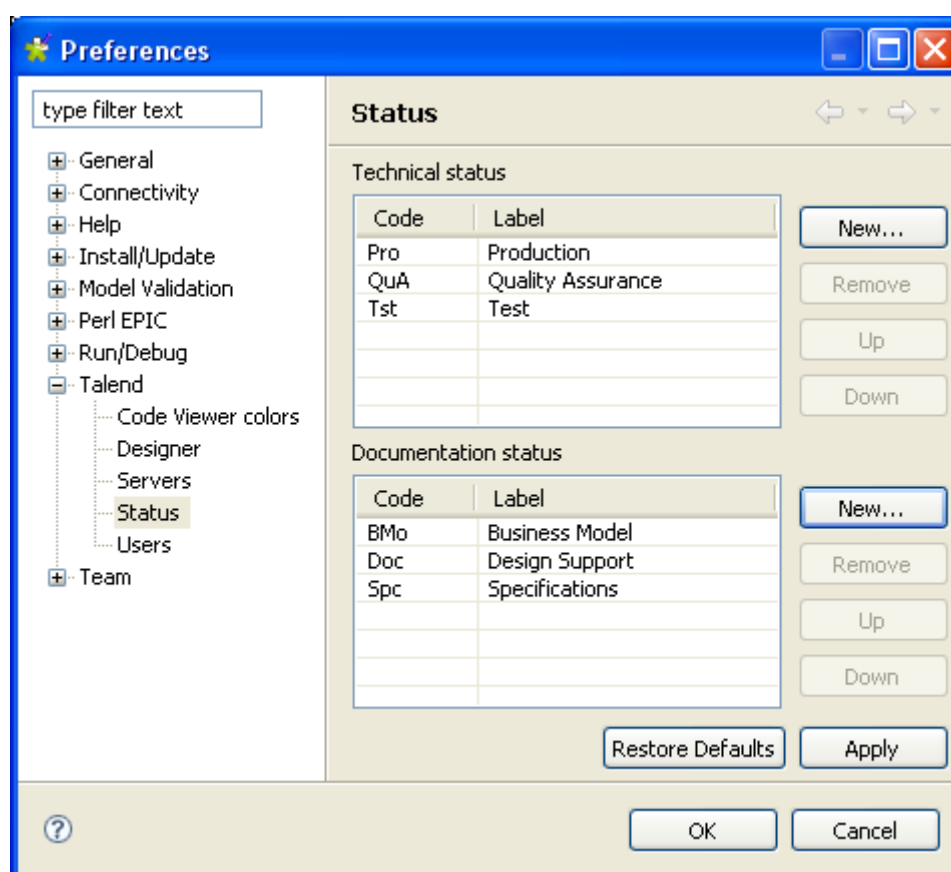
Dans les préférences, vous pouvez définir des niveaux de statuts pour vos jobs.

- Développez le noeud **Talend** et cliquez sur **Status** pour définir les propriétés principales des éléments du **Repository**.
- Le panneau des propriétés des éléments du Repository regroupe toutes les données générales, notamment : **Name**, **Purpose**, **Description**, **Author**, **Version** et **Status** de l'élément sélectionné.
- Alimentez la liste **Status** avec les valeurs appropriées à vos besoins d'entreprise. Notez que le **Code** ne peut pas dépasser trois caractères et le Libellé (**Label**) est obligatoire.



Talend fait la différence entre deux types de statuts : **Technical status** et **Documentation status**.

La liste de statuts 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 permet de classer les éléments du référentiel qui sont utilisés pour documenter les models/jobs. Cette liste de statuts n'est disponible que pour les Business Models et la Documentation.

Une fois la définition des **Status** terminée, cliquez sur **OK** pour enregistrer les paramètres.

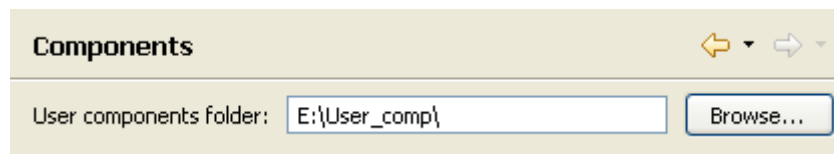
Désormais la liste **Status** vous permet d'appliquer vos paramètres de classification personnalisés à vos éléments du référentiel.

Composants externes ou utilisateur

Vous pouvez créer ou développer vos propres composants et les utiliser dans **Talend Open Studio**.

Pour plus d'informations concernant la création et le développement de composants utilisateur, consultez la page wiki (en anglais) *Component creation tutorial section*.

Dans l'arborescence **Preferences**, développez le noeud **Talend**, puis sélectionnez **Components**.



- Renseignez le champ **User components folder** pour indiquer le chemin d'accès aux composants à ajouter à la **Palette** de **Talend Open Studio**.
- Redémarrez **Talend Open Studio** pour que cet ajout soit effectif.

—Conception d'un Business Model—

Conception d'un Business Model

Talend Open Studio est l'outil optimal pour mettre en place une approche Top/Down vous permettant de garder le contrôle sur le processus métier, de son niveau d'abstraction le plus élevé à l'étape décisionnelle et même technique la plus pragmatique.

Ce chapitre s'adresse aux responsables métier, décisionnaires BI ou développeurs qui souhaitent modéliser leurs besoins en flux de données à un niveau macro.

Objectifs

Un business model est une vue non technique d'un besoin métier de gestion de flux de données.

Généralement, un business model intègre en premier lieu les systèmes stratégiques et étapes d'exécution déjà mis en place et opérationnels au sein de votre entreprise. Puis vous ajoutez brique par brique vos besoins supplémentaires et les reliez entre eux. Ces systèmes, connexions et autres besoins sont symbolisés dans **Talend Open Studio** par de multiples formes et liens disponibles dans la Palette.

Toutes les formes et tous les liens peuvent être décrits dans leur propriétés à l'aide des paramètres de documentation et de formats que vous définissez dans le référentiel de métadonnées, ou directement dans votre business model.

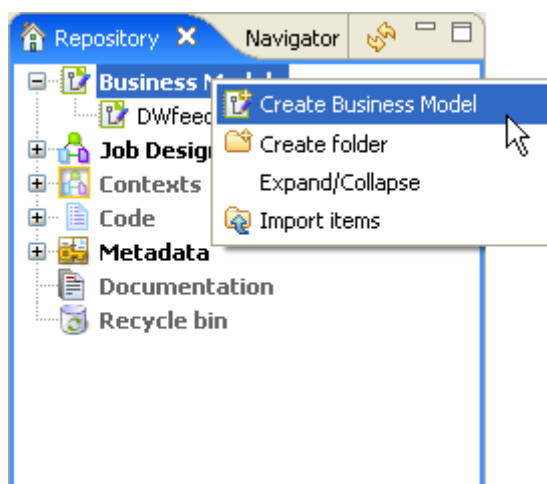
A partir de **Talend Open Studio**, vous disposez de nombreux outils vous permettant de :

- modéliser vos besoins métier
- créer des éléments dans le référentiel de métadonnées et les assigner à vos objets de Business model
- définir les propriétés d'apparence de vos objets de Business model.

Créer ou ouvrir un Business Model

Lancez **Talend Open Studio** suivant la procédure détaillée dans la section *Accès à Talend Open Studio*, page 5.

Dans la fenêtre principale de **Talend Open Studio**, cliquez sur **Business Models** dans le **Repository** pour développer l'arborescence.



Cliquez droit sur Business Models et sélectionnez l'option *Expand/Collapse*, pour afficher tous les modèles existants, s'il en existe déjà.

Ouvrir un Business Model

Double-cliquez sur le nom du business model que vous souhaitez ouvrir.

Le business model sélectionné s'ouvre dans l'espace de modélisation, généralement placé à droite du référentiel.

Créer un Business Model

Cliquez droit sur **Business Models** et sélectionnez

Create Business Model.

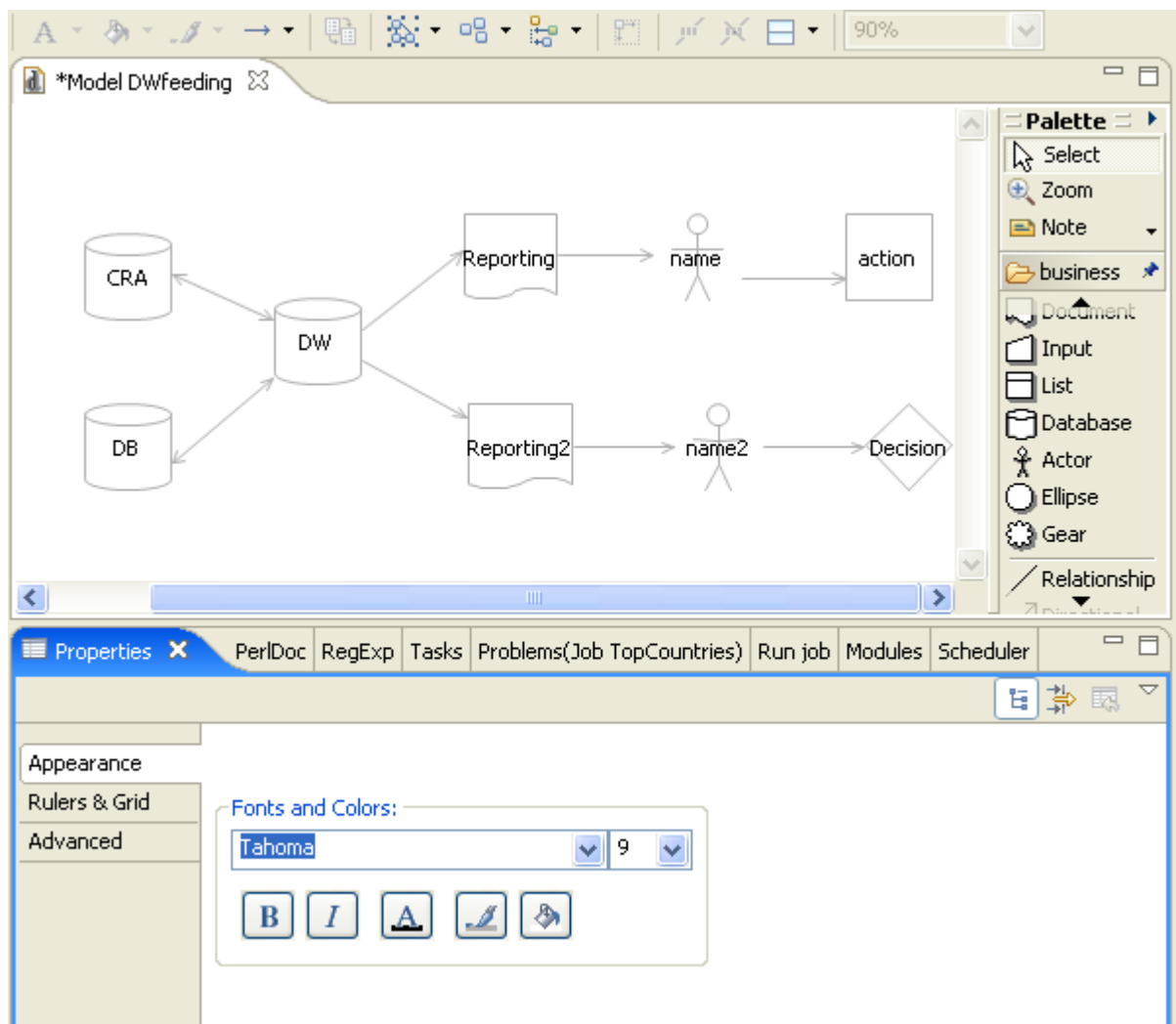
L'assistant de création vous guide dans les étapes de création d'un Business model.

Sélectionnez le répertoire de destination (**Location**) où vous souhaitez conserver votre nouveau model.

Puis donnez un nom à ce Business model dans le champ **Name**. Le nom que vous associez au fichier apparaît sous le noeud Business Models, dans le Repository.

Le **Modeler**, ou espace de modélisation, s'ouvre sur un espace de conception vide.

Vous pouvez créer autant de business models que vous le souhaitez. Un système d'onglets en haut de l'éditeur vous permet de naviguer facilement entre tous les processus ouverts.



Le **Modeler** est composé des vues suivantes :

- L'éditeur graphique (ou espace de modélisation) de **Talend Open Studio**
- Une **Palette** de formes et de connexions spécifiques à la modélisation métier
- Un onglet **Properties** présentant toutes les informations spécifiques sur tout ou partie du diagramme.

Modélisation d'un Business Model

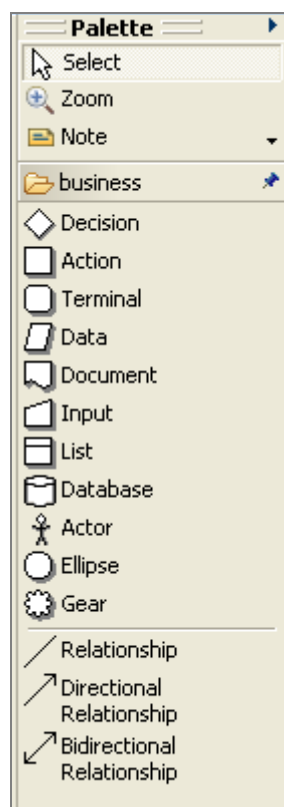
Si plusieurs onglets sont ouverts dans votre éditeur graphique, assurez-vous que la représentation graphique affichée est correcte, en cliquant sur l'onglet approprié.

Note: En effet, la fenêtre **Properties** ainsi que les éléments des **Menus** affichent des informations relatives au modèle actif seulement.

Cliquez et déposez les éléments de la **Palette**, puis connectez-les à l'aide de liens. Personnalisez l'aspect visuel du diagramme à l'aide du zoom avant et arrière.

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 affectations (**assignment**) que vous lui attribuez.



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

Note: Si les formes ne s'affichent pas dans la palette, cliquez sur l'icone du répertoire **business** pour dérouler la librairie 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.

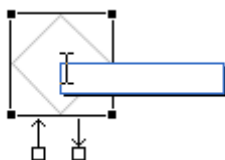
Alternativement, la barre d'accès rapide aux formes apparaît lorsque vous conservez votre curseur immobile quelques instants sur l'espace de modélisation :



Par exemple, si votre business model inclut une étape de décision, sélectionnez la forme losange dans la Palette.

Note: 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. Donnez un nom parlant qui vous permettra ensuite d'identifier rapidement le rôle de cette forme dans le processus.

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 processus.

Voir également : *Connexions*, page 29.

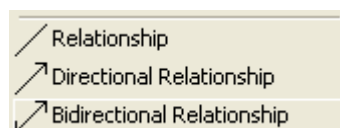
Les formes disponibles comprennent :

Libellé	Description
Decision	Représente une condition if dans le flux de données. Permet de prendre des mesures contextuelles.
Action	Tout type d'action, notamment, transformation, traduction ou format par exemple.
Terminal	Tout type de terminal de sortie.
Data	Tout type de données numériques ou alphanumériques.
Document	Ajoute un objet document qui peut servir en entrée ou en sortie pour fournir les données à traiter.
Input	Ajoute un objet entrée permettant à l'utilisateur de saisir ou de fournir manuellement les données à traiter.
List	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.
Database	Ajoute un objet base de données qui peut servir en entrée ou en sortie du traitement de données.
Actor	Représente schématiquement tous les acteurs du décisionnel à l'implémentation technique.
Ellipse	Ajoute une forme elliptique.
Gear	Cette forme d'engrenage est généralement utilisée pour illustrer les morceaux de code et moulinettes développés manuellement et qui pourraient être remplacés par des jobs Talend par exemple.

Connexions

Une des phases essentielles de la conception d'un business model consiste à mettre en place les relations entre les formes source et cible.

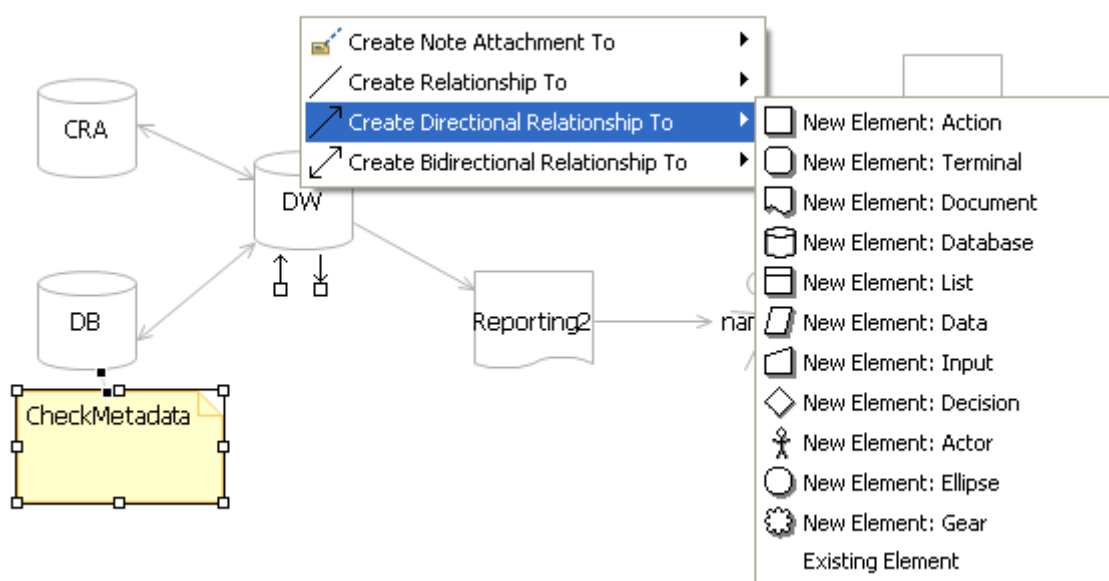
Plusieurs solutions sont offertes :



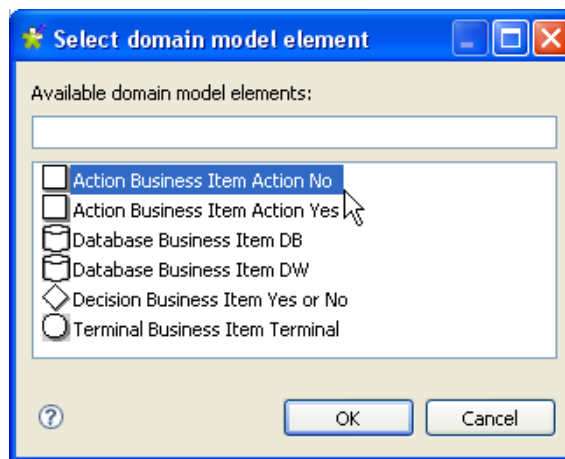
Sélectionnez l'outil **Relationship** de la **Palette**. Puis dans l'éditeur graphique, tirez un lien d'une forme vers une autre pour dessiner la connexion.

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.

- Simplement placez votre souris au dessus de la forme source pour faire apparaître les doubles flèches.
- Sélectionnez la flèche correspondant à la connexion dont vous avez besoin.
- Faites glisser l'une des flèches vers une zone vide de l'espace de modélisation et relâchez la souris pour afficher le menu contextuel.
- Sélectionnez la connexion appropriée dans la liste : **simple relationship**, **directional relationship** ou **bidirectional relationship**.
- 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 et sa mise en forme peut être paramétrée dans l'onglet **Properties**, consultez la section *Propriétés*, page 33.

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.

Note: Vous pouvez également ajouter de notes et des commentaires à votre processus pour permettre une reprise du modèle par la suite.

Voir également : *Commenter et réarranger un Business Model*, page 31

Commenter et réarranger un Business Model

Les outils suivants de la **Palette** vous permettent de personnaliser votre business model :

Libellé	Description
Select	Sélectionnez et déplacez les formes et relations dans l'éditeur graphique du Modeler .
Zoom	Zoomez sur une partie de la représentation graphique pour en voir le détail. Pour zoomer arrière, appuyez sur la touche Maj et cliquez sur l'éditeur graphique.
Note/Text/Note attachment	Permet d'ajouter des commentaires, des notes afin de conserver toute information utile concernant tout ou partie du processus.

Ajouter une note ou du texte libre

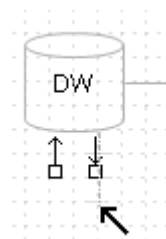
Pour ajouter une note, sélectionnez l'outil **Note** dans la **Palette**, à droite de l'éditeur graphique.

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.

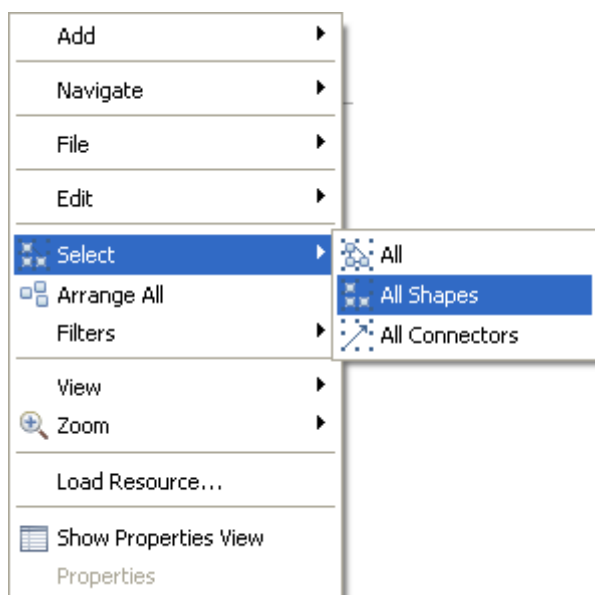
Pour ajouter une note attachée à une forme, sélectionnez **Note attachment** dans la Palette puis sélectionnez dans l'espace de modélisation la forme que vous souhaitez annotée. Lorsque la flèche noire apparaît, faites-la glisser vers une zone vide de l'éditeur. Relachez pour afficher le menu contextuel 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 processus.

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.

A partir de ce menu, vous pouvez également zoomer avant ou arrière sur une partie du modèle et ainsi changer la vue affichée.

Propriétés

Les informations de propriétés affichées dans l'onglet **Properties** 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, les propriétés fournissent des informations générales concernant l'espace de modélisation lui-même.

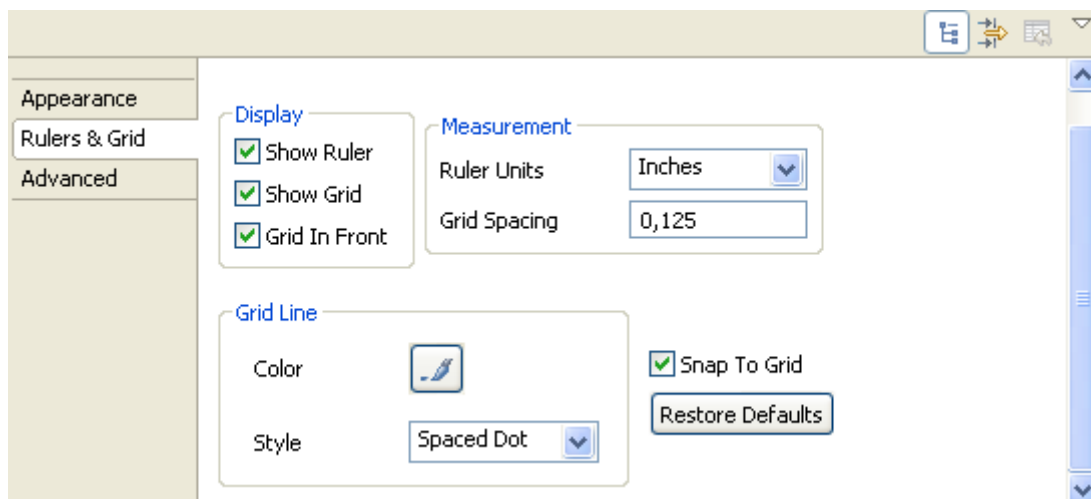
L'onglet **Properties** comporte différents types d'informations concernant :

- Règles et grille (**Rulers and Grid**)
- Format (**Appearance**)
- Affectation (**Assignment**)

Rulers and Grid

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

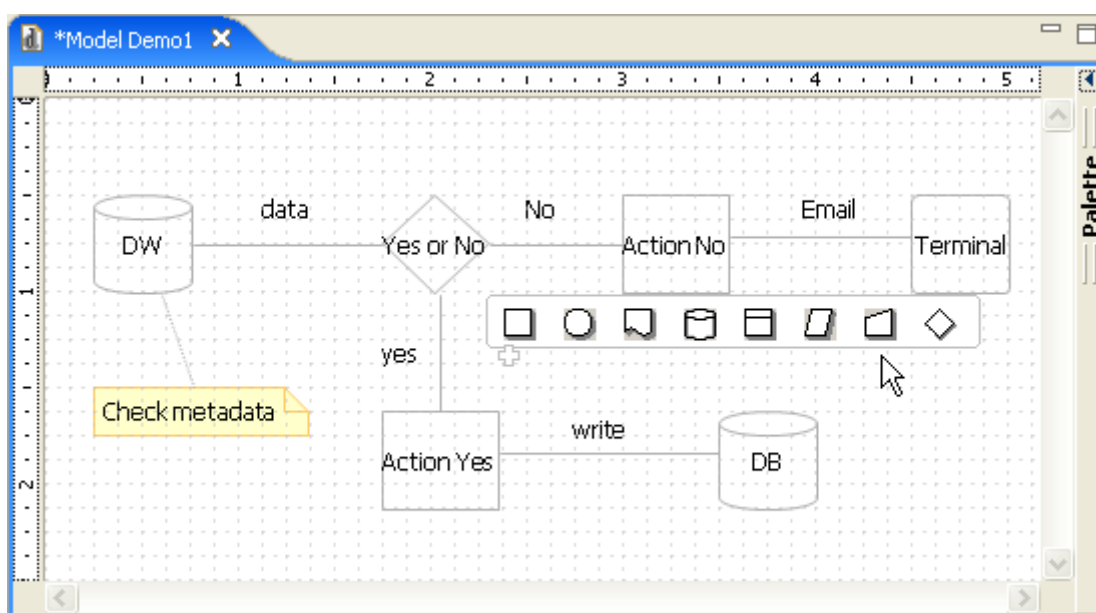
Cliquez sur l'onglet **Rulers & Grid** pour accéder aux paramètres de configuration des règles et de la grille.



Cochez les cases pour afficher la Règle (**Ruler**), ou la grille (**Grid**) ou les deux.

Grid in front place la grille devant toutes les autres formes du processus. Définissez l'unité de mesure de la règle **Centimeters** (centimètres), **Inches** (pouces) ou **Pixels**.

Vous pouvez également sélectionner la couleur ainsi que les styles des lignes de la grille ou restaurer les valeurs par défaut.



Appearance

A partir de l'onglet **Appearance** (Formats), vous pouvez appliquer des 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 **Properties** inclut les options de format suivants :

- remplir la forme avec la couleur sélectionnée
- colorer les bordures
- ajouter du texte
- ajouter un effet dégradé
- ajouter un effet ombré

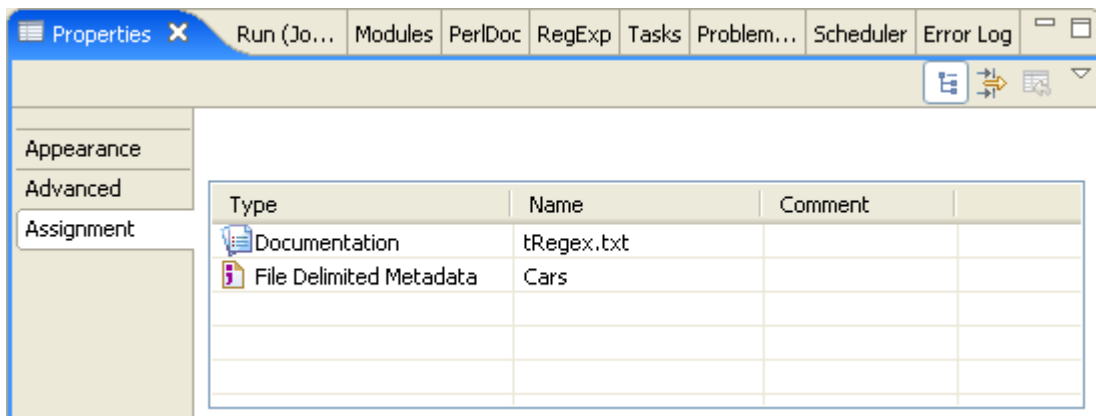
Vous pouvez également déplacer et gérer les différentes formes utilisées dans la vue de votre business model à l'aide des outils d'édition. Cliquez droit sur une des formes, pour faire apparaître les onglets appropriés.

Assignment

Le tableau des affectations affiche des informations détaillées sur les attributs du Repository (référentiel) affectés à la forme ou à la connexion sélectionnée.

Pour afficher les informations d'affectation, sélectionnez une forme ou une connexion dans la vue active.

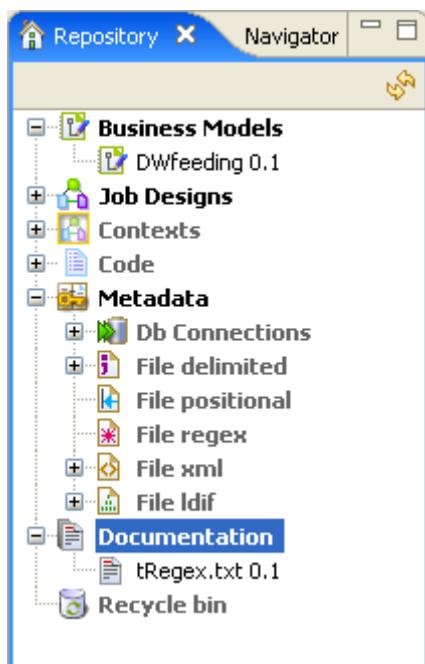
Vous pouvez modifier certaines informations ou lier un commentaire. En outre, si vous mettez à jour des données dans le référentiel, les informations d'affectation sont automatiquement mises à jour.



Pour plus d'informations concernant les affectations, voir *Affectation d'éléments du Repository à un Business Model*, page 35.

Affectation d'éléments du Repository à un Business Model

La vue **Assignment** répertorie les éléments sélectionnés dans le **Repository** et qui ont été affectés à un élément du Business model.



Une simple affectation (assignment) vous permet de définir et décrire, en d'autres termes, d'ajouter des métadonnées à votre business model, en lui associant différents types d'informations.

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

Pour affecter un élément, glissez et déposez l'élément du Repository vers une des formes composant votre business model courant.

Vous pouvez également utiliser le bouton *Un/Assign* pour effectuer cette opération d'affectation.

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

Vous pouvez retirer une affectation à l'aide du bouton *Un/Assign*.

Les types d'éléments du référentiel que vous pouvez affecter sont les suivants :

Élément	Description
Job designs	Si la représentation graphique d'un job est disponible, elle peut être réutilisée comme métadonnée du business model actif.
Metadata	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.
Business Models	Vous pouvez réutiliser comme métadonnées tout business model conservé dans le référentiel de ce projet.
Documentation	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.
Routines (Code)	Si vous avez développé des routines, d'automatisation de tâches, par exemple, vous pouvez les affecter à votre business model. Le répertoire Routine est conservé sous le noeud Code .

Pour plus d'informations concernant les éléments du **Repository**, consultez *Conception d'un Job*, page 39

Modifier un Business Model

Suivez la procédure qui convient au type de modification que vous souhaitez effectuer :

Renommer un Business Model

Cliquez sur le libellé du business model courant dans le **Repository** pour afficher les propriétés principales correspondantes, dans l'onglet **Main**.

Apportez vos modifications dans le champ **Name**. Le libellé est automatiquement changé dans le référentiel et apparaîtra dans l'onglet du processus au niveau de l'espace de modélisation la prochaine fois que vous l'ouvrez.

Copier et coller un Business Model

Dans **Repository > Business model**, cliquez droit sur le libellé du business model que vous souhaitez copier, et sélectionnez **Copy** dans le menu contextuel, ou utilisez le raccourci clavier **Ctrl+c**.

Puis cliquez droit à l'endroit où coller le business model en sélectionnant **Paste**.

Déplacer un Business Model

Pour déplacer un business model d'un emplacement à un autre dans votre répertoire Business model, sélectionnez un business model dans l'arborescence **Repository > Business Models**.

Puis glissez-déposez-le à son nouvel emplacement.

Vous pouvez également cliquer droit sur le libellé du business model et sélectionnez **Move** dans le menu contextuel.

Supprimer un Business Model

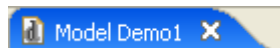
Cliquez droit sur le libellé du business model à supprimer et sélectionnez **Delete** dans le menu contextuel.

Vous pouvez également sélectionner le business model dans l'arborescence de référentiel et simplement le glisser-déposer dans la corbeille (**Recycle bin**) du **Repository**.

Enregistrer un Business Model

Pour sauvegarder un business model, cliquez sur **File > Save** ou utilisez le raccourci clavier **Ctrl+s**. Le modèle est sauvegardé sous le nom que vous lui avez donné au moment de sa création.

Un astérisque apparaît au niveau de l'onglet du nom de la vue métier dans l'éditeur graphique pour indiquer que des modifications ont été apportées à la vue courante mais qu'elles ne sont pas encore enregistrées.



—Conception d'un Job—

Conception d'un Job

Ce chapitre s'adresse aux programmeurs et responsables informatiques qui souhaitent mettre en place les aspects techniques d'un job basé (ou non) sur un business model. **Talend Open Studio** vous aide à développer et mettre en application votre job d'extraction, transformation et de chargement de flux de données.

Objectifs

Un job est la couche d'exécution ou l'implémentation technique d'un business model. Il traduit les besoins métier en code et se charge d'exécuter ce dernier. En d'autres termes, le job met en place votre flux de données.

Le Job design est la représentation graphique fonctionnelle de ce processus technique.

A partir de **Talend Open Studio**, vous pouvez :

- mettre en place les actions de votre job à l'aide d'une bibliothèque de composants techniques.
- changer les paramètres par défaut des composants ou créer de nouveaux composants ou famille de composants qui correspondent au mieux à vos besoins.
- paramétrer les connexions et relations entre composants afin de définir la séquence et la nature des actions.
- accéder au code généré du programme ou des composants pour le modifier ou le documenter.
- créer et ajouter des éléments au **Repository** (référentiel) à des fins de réutilisation et de partage (avec d'autres projets ou d'autres processus ou, à moyen terme, avec d'autres utilisateurs).

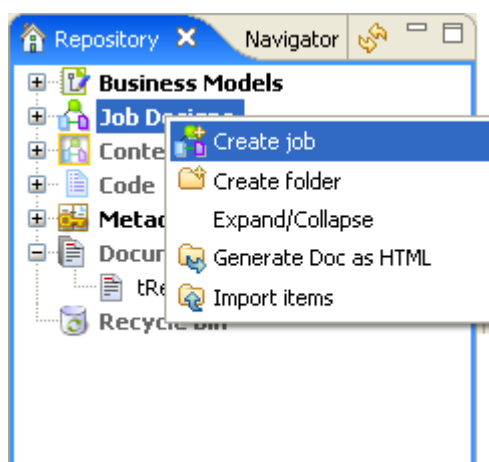
Créer ou Ouvrir un Job

Lancez **Talend Open Studio** suivant la procédure décrite dans la section *Accès à Talend Open Studio*, page 5.

Dans la fenêtre **Talend Open Studio**, cliquez sur **Job Designs** pour développer l'arborescence des jobs.

Vous pouvez créer des dossiers via le menu contextuel afin de faciliter la gestion de vos jobs. Cliquez droit sur le noeud **Job Designs** de l'arborescence et sélectionnez *Create folder*. Renseignez le champ **Name** et cliquez sur **OK**.

Si vous avez déjà créé des jobs que vous souhaitez déplacer dans ce nouveau dossier, glissez et déposez-les dans le dossier.



Ouvrir un job

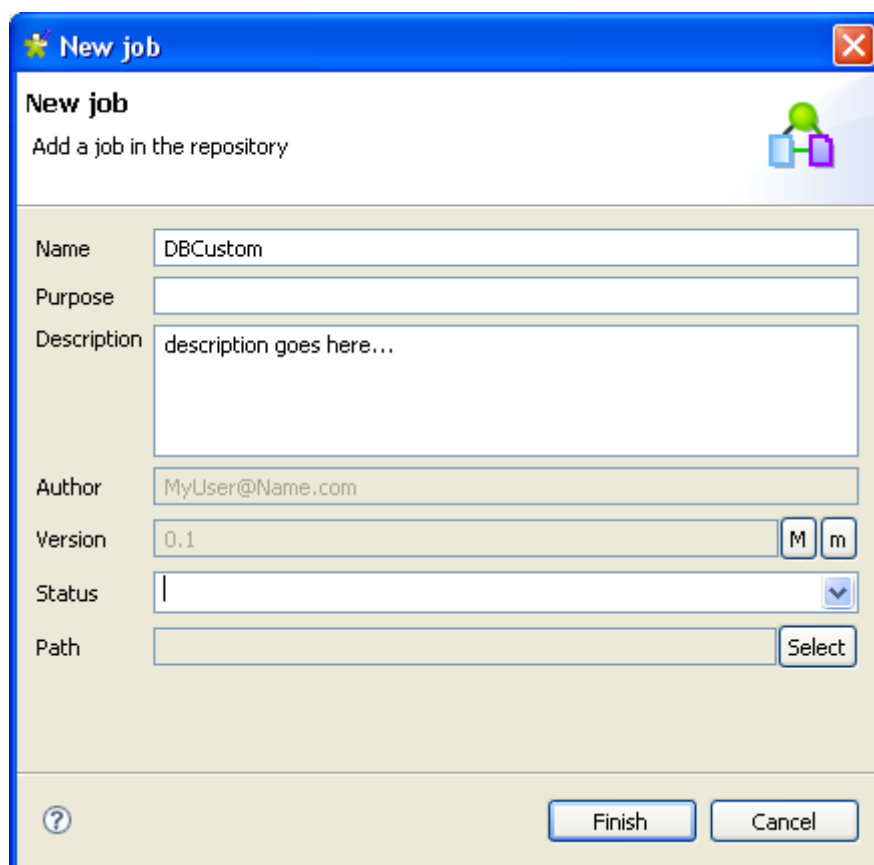
Double-cliquez sur le libellé du job que vous souhaitez ouvrir.

Le **Designer** s'ouvre sur la dernière vue enregistrée du job.

Note: Vous pouvez ouvrir autant de jobs que vous le souhaitez. Un système d'onglets en haut de l'éditeur vous permet de naviguer très facilement entre tous les processus ouverts.

Créer un job

Cliquez droit sur **Job Designs** dans le référentiel et sélectionnez *Create job* dans le menu contextuel. L'assistant de création vous guide dans les étapes de définition des propriétés de votre nouveau job.



Champ	Libellé
Name	Saisissez un nom pour le nouveau job. Un message s'affiche si vous saisissez des caractères interdits.
Purpose	Saisissez toute information que vous jugerez utile concernant l'utilisation du job.
Description	Saisissez une description si nécessaire.
Author	Ce champ est en lecture seule car il reprend par défaut le login utilisateur courant.
Version	La version indiquée est en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l'aide des boutons M et m.
Status	Vous pouvez définir le statut d'un job dans vos préférences Talend Open Studio . Par défaut aucun statut n'est défini. Pour les définir, cliquez sur le menu Window > Preferences > Talend > Status.

Le **Designer** s'ouvre sur un onglet de modélisation vide, indiquant simplement le nom du nouveau job.

Note: Vous pouvez créer autant de jobs que vous le souhaitez. Un système d'onglets en haut de l'éditeur vous permet de naviguer facilement entre tous les jobs ouverts.

Le **Designer** est composé des vues suivantes :

- L'éditeur graphique **Talend Open Studio**
- Une **Palette** de composants et de connexions spécifiques à la conception de Job designs
- Un onglet **Component** présente toutes les informations spécifiques sur tout ou partie de la représentation graphique du job.

Conception d'un Job

L'espace de modélisation et la palette sont grisés lorsqu'aucun job n'est affiché.

Si vous concevez un job pour la première fois dans **Talend Open Studio**, l'espace de modélisation s'ouvre sur un éditeur vide. Si vous ouvrez un job existant, l'éditeur s'ouvre sur la dernière vue sauvegardée du job.

Afficher, cacher et déplacer la palette

La **Palette** contient tous les éléments nécessaires à la création des jobs les plus complexes. Ces composants sont regroupés en familles et sous-familles.

Par défaut, la palette est 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.

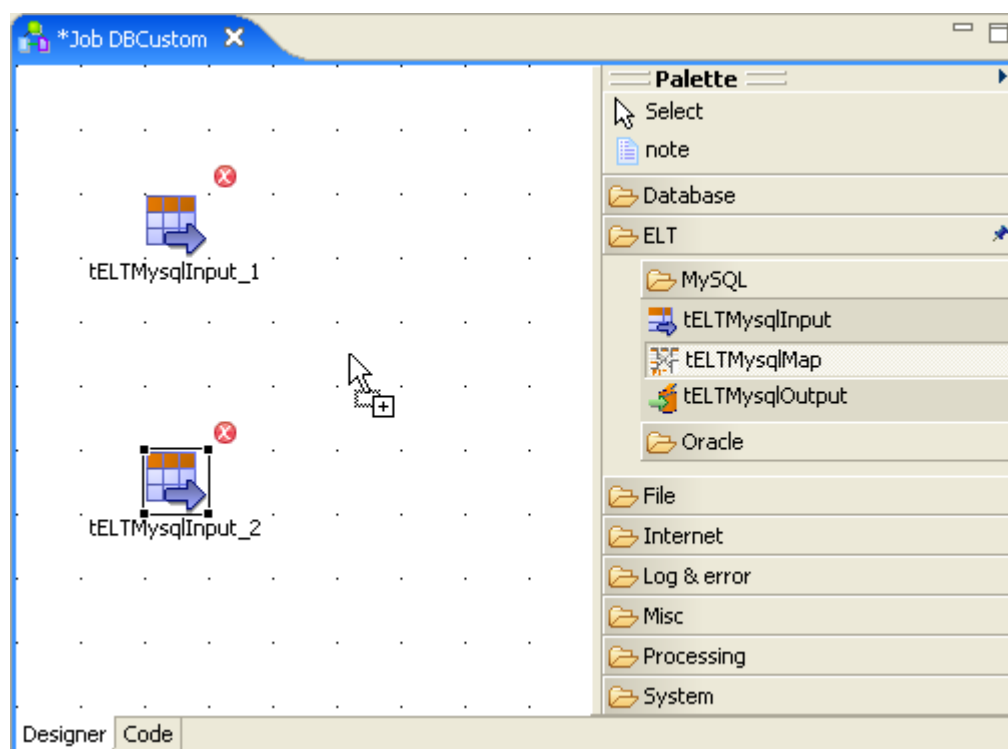
Pour connaître les propriétés spécifiques à chacun des composants, consultez les *Composants Talend Open Studio Guide de référence*.

Vous pouvez aussi détacher la **Palette** de l'espace de modélisation dans **Talend Open Studio**. Pour détacher l'onglet **Palette**, cliquez sur le menu **Window > Show View... > General > Palette**.

Cliquer & Déposer des composants

Pour commencer, cliquez sur un **Composant** ou une **Note** de la **Palette**. Puis cliquez dans l'espace de modélisation pour le déposer et ainsi l'ajouter à votre job design.

Si la Palette n'est pas visible, consultez *Afficher, cacher et déplacer la palette, page 41*.



Plusieurs types d'icônes d'information et d'avertissement peuvent s'afficher à côté du composant. Placez le pointeur sur l'icône pour faire apparaître la bulle d'information. Ces icônes s'affichent tant que vous n'avez pas terminé de paramétrer toutes les propriétés de base (et parfois avancées) des composants à partir de la vue **Component**.

WARNING—Vous devrez utiliser le code correspondant, c'est-à-dire Perl pour les projets en perl et Java pour les projets en java.

Voir également :

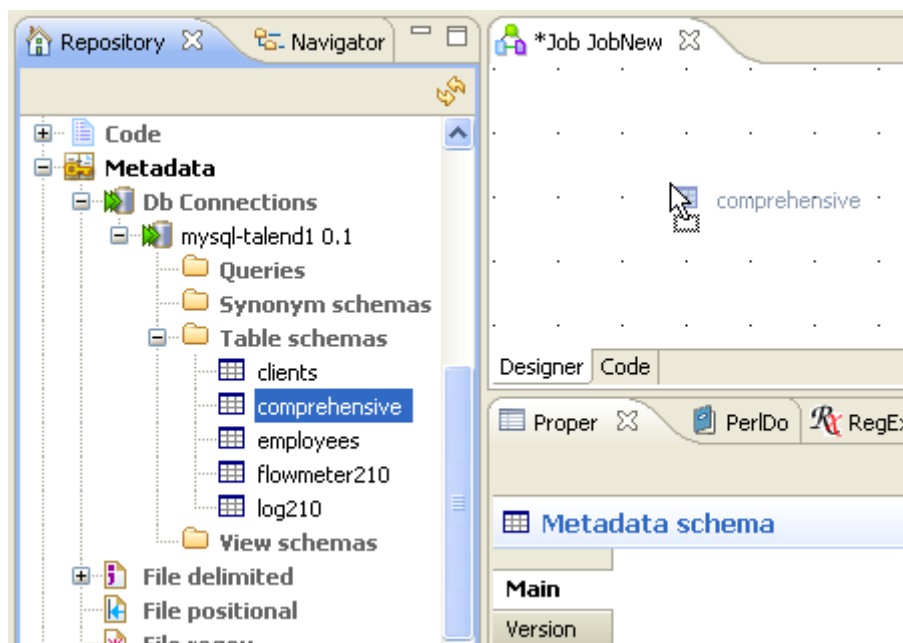
- *Connexion des composants, page 46*
- *Avertissements et erreurs sur un composant, page 46*
- *Définition des propriétés d'un composant, page 51*

Cliquer & Déposer des composants du répertoire Metadata

Si vous utilisez régulièrement les mêmes schémas de métadonnées pour plusieurs jobs, nous vous conseillons de les sauvegarder dans le Repository. Pour plus d'informations concernant l'assistant de gestion des métadonnées, consultez *Définition des schémas de métadonnées, page 58*.

Une fois les métadonnées sauvegardées dans le répertoire **Metadata** du **Repository**, vous pouvez directement cliquer sur les éléments et les déposer dans l'éditeur.

- Développez le noeud **Metadata** du Repository.
- Sélectionnez le type de connexion souhaité.
- Sélectionnez un schéma s'il y en a plusieurs pour la même connexion.



En fonction du type de composant (Input ou Output) que vous souhaitez utiliser, suivez les instructions suivantes :

- **Input** : Cliquez et déposez l'élément sélectionné dans l'éditeur graphique et ainsi l'ajouter à votre job design.
- **Output** : Maintenez la touche **Ctrl** enfoncée pour glisser et déposer le composant dans l'éditeur et ainsi l'ajouter à votre job design.

L'onglet **Component** affiche les détails des connexions sélectionnées ainsi que les informations du schéma sélectionné.

Note: Si vous avez sélectionné une connexion sans sélectionner de schéma, le premier schéma rencontré remplira les propriétés.

Ajouter une Note au Job

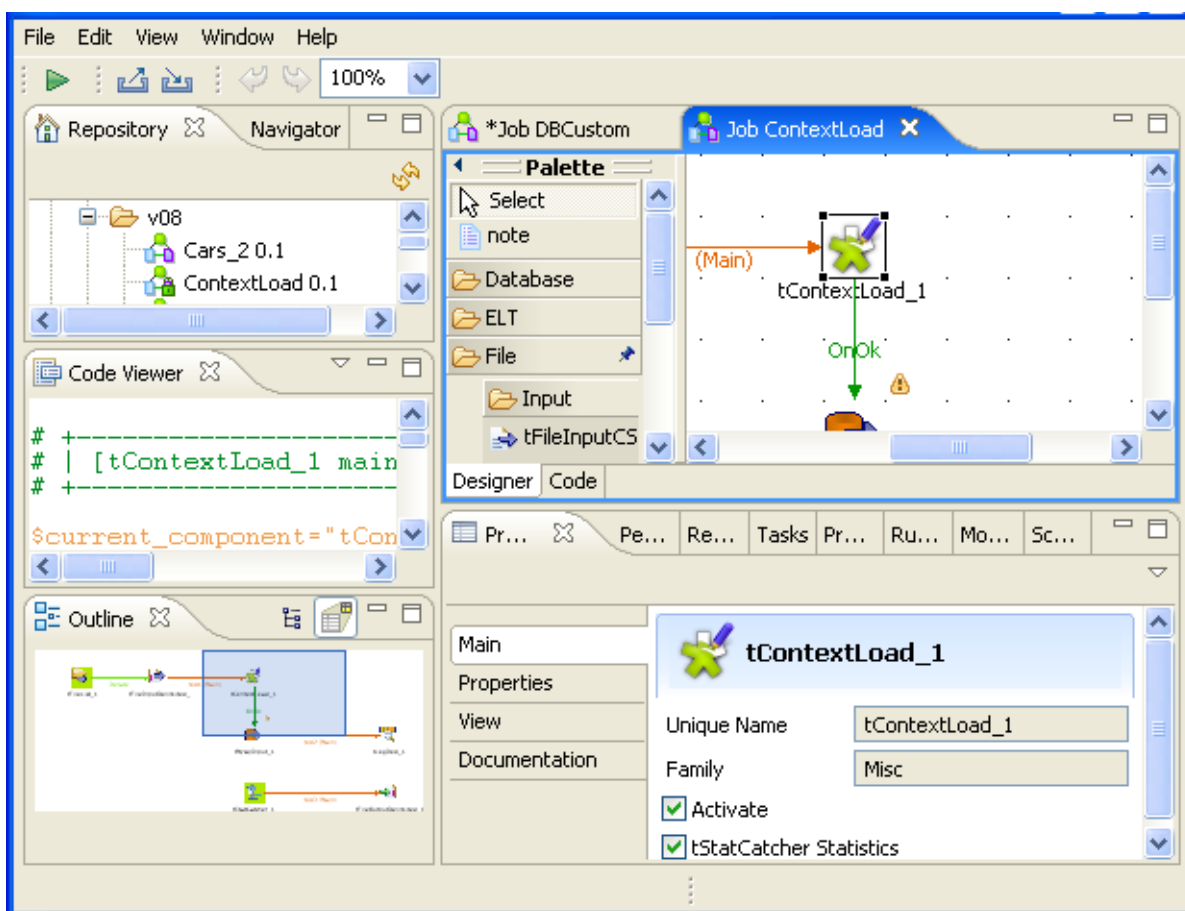
Sélectionnez l'élément Note dans la Palette.



Cliquez et déposez l'élément **Note** dans l'éditeur pour ajouter un commentaire sur un composant particulier ou sur l'ensemble de votre job.

Réorganiser les onglets d'un Job

Vous pouvez déplacer tous les onglets de **Talend Open Studio** selon vos besoins.



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.

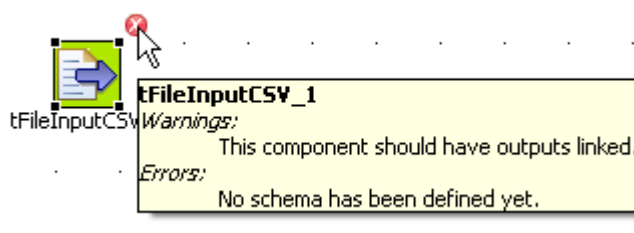
Cliquez sur la croix pour fermer une vue. Pour restaurer une vue, cliquez sur **Window > Show View... > Talend**, puis cliquez sur le nom de la vue que vous souhaitez afficher ou consultez *Raccourcis clavier*, page 130.

Si la **Palette** n'est pas visible ou si vous souhaitez la détacher, cliquez sur le menu **Window > Show view... > General > Palette**. La Palette s'ouvrira alors dans une vue séparée qui peut être déplacée dans la fenêtre principale **Talend Open Studio**.

Avertissements et erreurs sur un composant

Lorsqu'un composant n'est pas correctement configuré ou si une connexion vers un autre composant manque, une croix blanche sur fond rouge ou un simple signe attention apparaît.

Placez le pointeur de la souris sur le composant pour faire apparaître les messages d'erreurs et d'avertissement. Cette aide contextuelle vous informe sur les données manquantes ou le statut du composant.



Connexion des composants

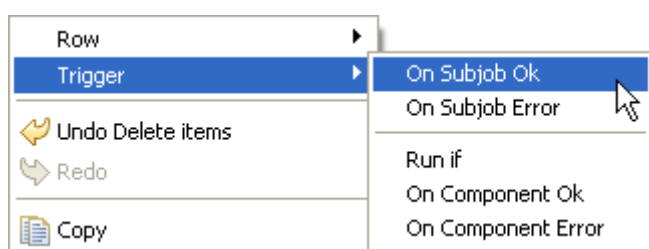
Il existe divers types de connexions qui définissent soit la donnée à traiter soit l'enchaînement logique du job.



Dans l'éditeur graphique, lorsque vous glissez le pointeur du composant source vers le composant cible, une prise électrique symbolise le branchement en cours et un signe *interdit* indique que le composant cible n'est pas valide. Le cercle barré ne disparaît qu'une fois le composant valide atteint.

Types de connexion

Seules les connexions autorisées pour le composant sélectionné dans l'éditeur graphique sont proposées dans la liste du menu contextuel.



Les types de connexions diffèrent selon la nature du composant et son rôle dans le job.

Les types de connexions disponibles varient également si les données proviennent d'un ou plusieurs fichiers d'entrée et si elles sont transmises vers un ou plusieurs fichiers de sortie.

Sélectionnez un composant dans l'éditeur graphique et cliquez droit pour afficher le menu contextuel. Tous les liens disponibles pour ce composant particulier apparaissent à l'écran.

Connexion de type Row

La connexion de type **Row** manipule les données elles-mêmes. Les liens de type Row sont **main**, **lookup**, **reject** ou **output** selon la nature du flux de données traité.

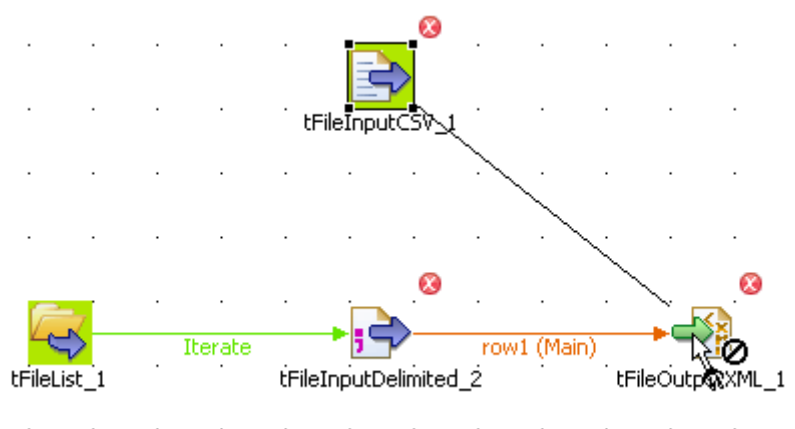
Main

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.

Note: Vous ne pouvez pas connecter deux composants d'entrée à l'aide d'une connexion de type **main Row**.

Note: Une seule connexion entrante de type Row est possible par composant. La seconde connexion Row se transforme en connexion Lookup (de référence) automatiquement. Mais de manière générale, vous ne pourrez pas relier deux fois le même composant cible avec une connexion main Row.

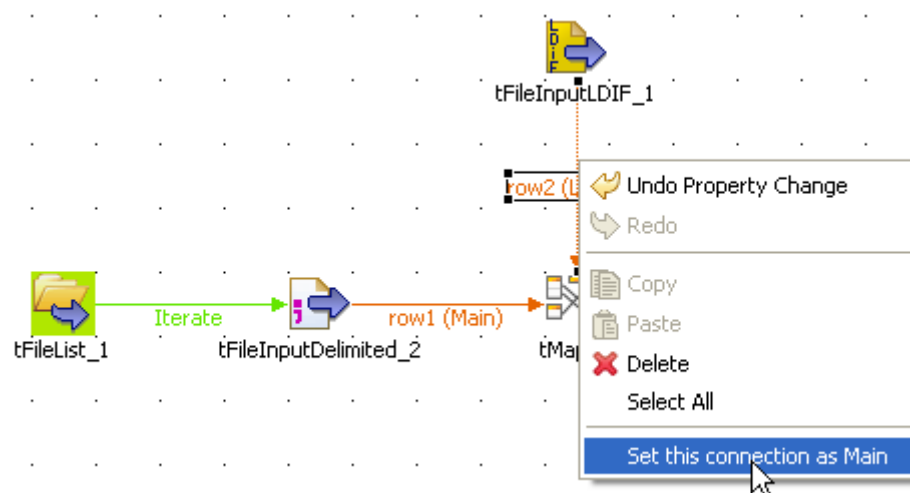


- Cliquez droit sur le composant d'entrée et sélectionnez **Row > Main** dans la liste des connexions.
- Sinon, cliquez sur le composant pour le sélectionner, puis, tout en cliquant droit, déplacez votre curseur vers le composant de destination. Une connexion de type sera automatiquement créée.

Pour en savoir plus sur les cas spécifiques de connexions Row multiples, consultez *Job à entrées/sorties multiples*, page 51.

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**.

Voir également : *Job à entrées/sorties multiples*, page 51.

Filter

La connexion de type **Filter** est une connexion reliant 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 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.).

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.

Note: Le système se souvient également des noms des connexions de sortie supprimées (ainsi que les propriétés définies). 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.

Voir également : *Job à entrées/sorties multiples*, page 51.

Uniques/Duplicates

Les connexions de type **Uniques** et **Duplicates** 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.

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'un seul lien d'itération, **Iterate**. La connexion **Iterate** est généralement utilisé sur le composant de début (Start) d'un flux principal ou secondaire.

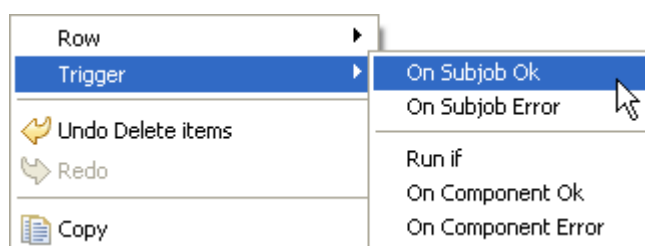
Certains composants ne peuvent fonctionner avec d'autres composants qu'à l'aide d'un lien d'itération, c'est le cas du composant tFilelist.

Note: A la différence des autres types de connexions, le nom du lien **Iterate** est en lecture seule.

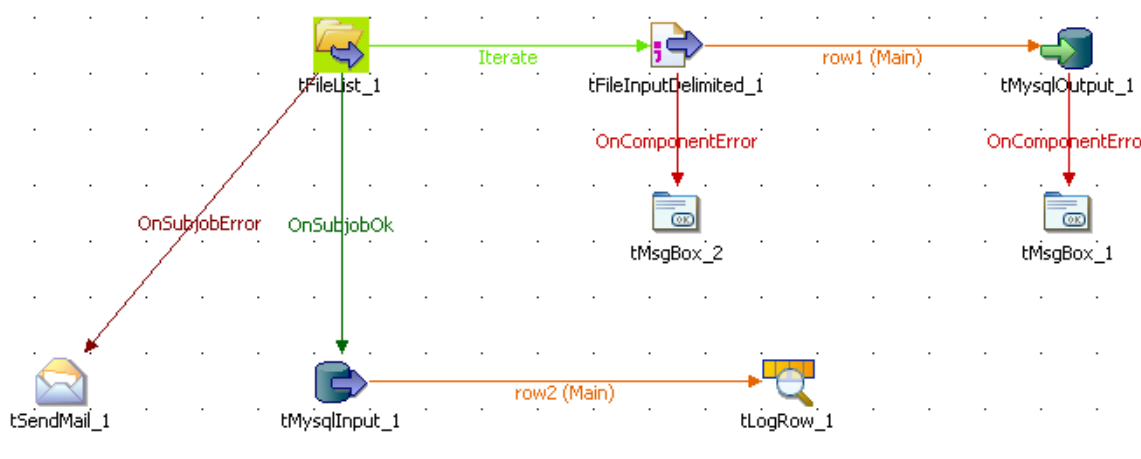
Connexion de type Trigger

Les connexions de déclenchement, **Trigger**, aident à définir l'ordre 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.



Il existe plusieurs types de déclenchements : les déclenchements de sous-job et les déclenchements de composant. Ces connexions sont utilisées pour orchestrer et vérifier plus facilement les sous-jobs qui composent le job ou traiter les erreurs qui n'étaient pas prévues.



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.

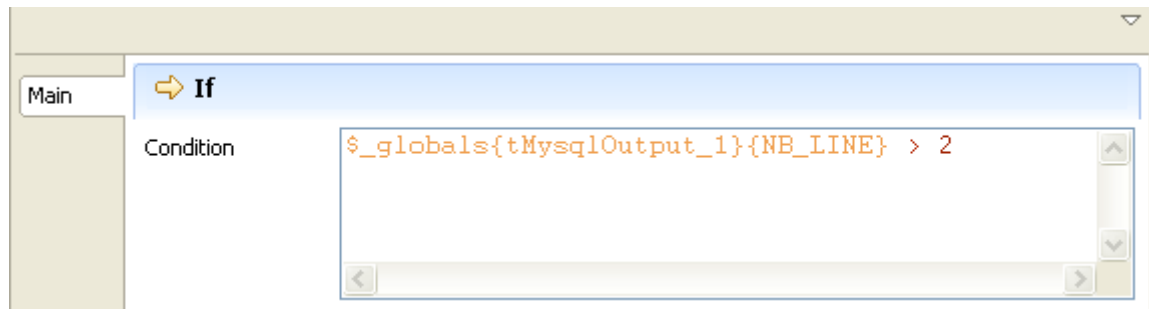
OnSubjobError : Cette connexion est utilisée pour déclencher le sous-job qui suit lorsque le premier sous-job (principal) ne s'est pas déroulé correctement. Ce sous-job "on error" vous permet d'identifier les éventuels goulots d'étranglement ou de corriger l'erreur si cela est possible.

Voir également : *Définition du composant Start*, page 57.

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. Cliquez sur la connexion pour afficher les propriétés de votre connexion **If**, puis définissez vos conditions en Perl ou en Java en fonction du langage de génération de code choisi pour votre projet. Le raccourci clavier **Ctrl+Barre d'espace** vous donne accès à toutes les variables globales et de contexte.



Connexion de type Link

La connexion **Link** n'est utilisée qu'avec les composants ELT. Ces liens 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.

Voir également : *Composants Talend Open Studio Guide de référence*

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**.

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

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

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 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 *Mapping de flux de données dans un Job*, page 92.

Pour plus d'informations concernant le composant tMap et son utilisation, consultez le composant *tMap*, page 119 du Guide de référence.

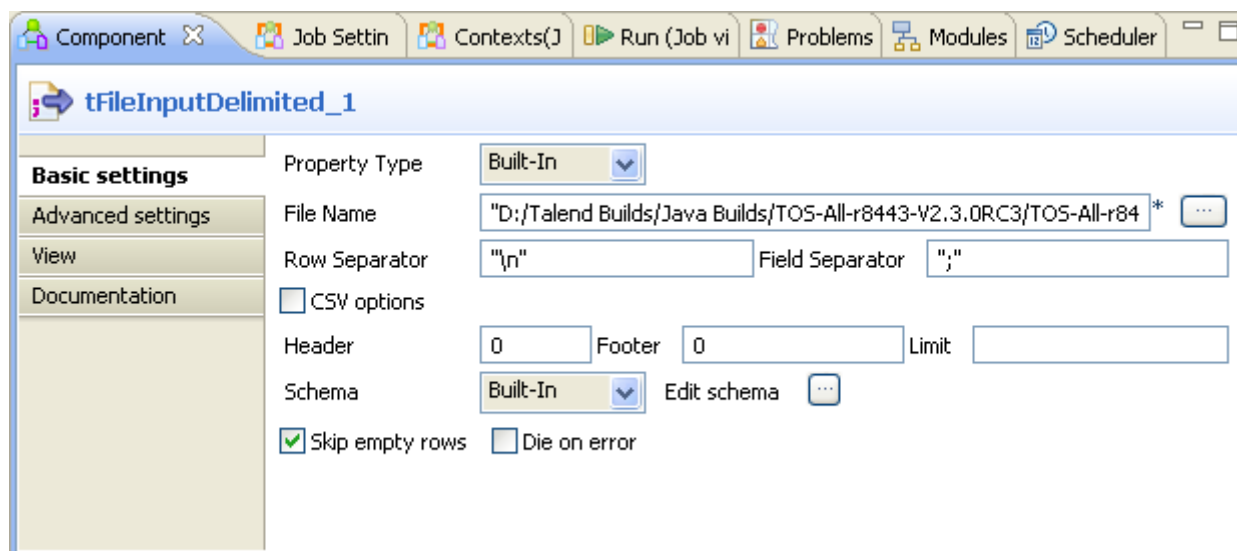
Definition des propriétés d'un composant

Les propriétés de chacun des composants formant un **Job** 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.

Basic Settings

L'onglet **Basic Settings** fait parti de la vue **Component**, qui est situé en bas de la fenêtre de **Talend Open Studio**.



Les paramètres de base des composants sont différents en fonction de ces exigences fonctionnelles dans le job. Pour une description détaillée des propriétés et de l'utilisation de chaque composant, consultez les *Composants Talend Open Studio Guide de référence*.

Note: Certains composants sont paramétrés à l'aide de code ou de fonctions. Veillez à bien utiliser le code approprié, i.e. le code Perl dans les propriétés en perl et le code Java dans les propriétés en java.

Pour les composants **File** (Fichier) et **Database** (Base de données), 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. Voir également : *Définition des schémas de métadonnées*, page 58.

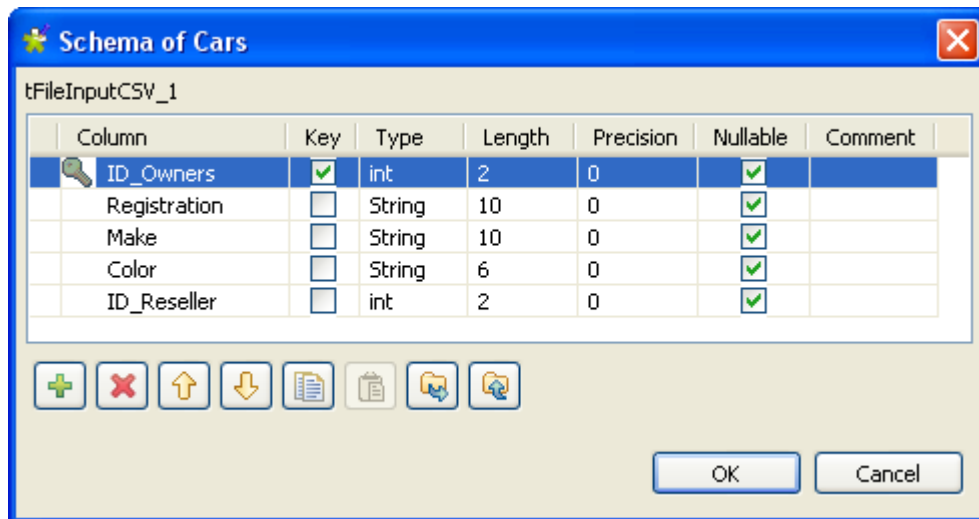
Pour tous les composants traitant des flux de données (la plupart des composants), vous pouvez définir un schéma **Talend** afin de décrire et sélectionner les données à traiter. Comme pour les propriétés, ce schéma peut être en local (**Built-in**) ou stocké dans le **Repository** dans les

métadonnées que vous avez créé. Vous trouverez une description détaillée d'un schéma dans la section suivante.

Paramétrage d'un schéma built-in

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.

Sélectionnez Built-in dans la liste, puis cliquez sur **Edit Schema**. Ensuite, créez votre schéma en y ajoutant des colonnes et en décrivant leur contenu selon la structure du fichier d'entrée.



Vous pouvez récupérer le schéma ponctuel d'un fichier d'entrée pour alimenter le schéma de sortie. Pour ce faire, cliquez sur le bouton **Sync columns** dans les propriétés du composant Output.

Note: En Java, des informations supplémentaires sont nécessaires. Pour plus d'informations sur les formats de date par exemple, visitez le site : <http://java.sun.com/j2se/1.5.0/docs/api/index.html>

Paramétrage d'un schéma repository

Vou pouvez éviter d'avoir à recréer les mêmes schémas et leur métadonnées en créant des fichiers de métadonnées que vous pouvez centraliser dans le répertoire Metadata du référentiel (Repository).

Pour rappeler un fichier de métadonnées dans votre processus courant, sélectionnez **Repository** dans la liste déroulante **Schema type** et sélectionnez le fichier approprié dans la liste **Repository**. Puis cliquez sur **Edit Schema** pour vérifier que les données sont appropriées.

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

Vous ne pouvez pas modifier le schéma distant à partir de cette fenêtre. Pour apporter une modification, accédez au fichier de métadonnées à partir du référentiel.

Voir également : *Définition des schémas de métadonnées*, page 58

Affichage dynamique d'un champ (Ctrl+Espace)

Dans tous les champs de l'onglet **Component**, 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**.

- Placez le curseur de votre souris dans n'importe quel champ de l'onglet **Component**.
- Appuyez sur **Ctrl+Barre d'espace** pour accéder à la liste des variables.
- Sélectionnez les paramètres adéquats dans la liste. En plus de la liste des variables, un panneau d'information fournit des détails sur les paramètres sélectionnés.

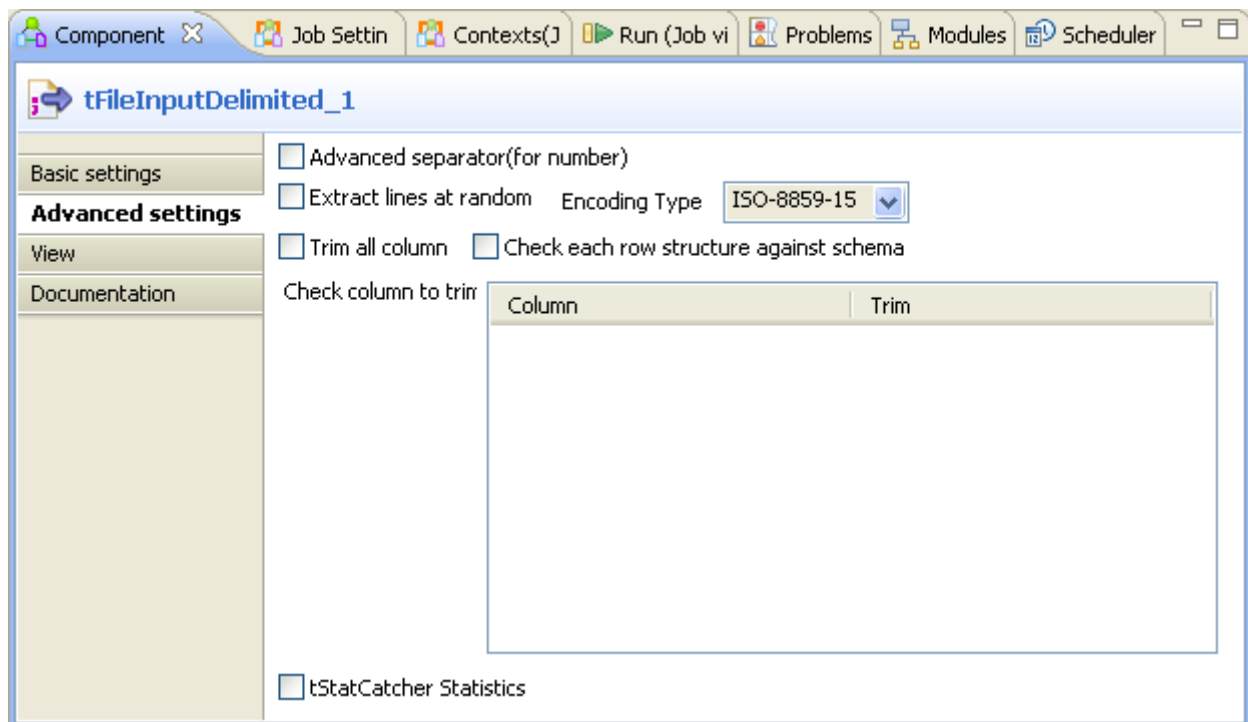
<p>Description: Error Message</p> <p>Global variable, property of component tMap [tMap_1].</p> <p>Type: String</p> <p>Availability: After</p> <p>Variable Name: ((String)globalMap.get("tMap_1_ERROR_MESSAGE"))</p>	<p>tFileInputDelimited_2.ERROR_MESSAGE</p> <p>tFileInputDelimited_2.NB_LINE</p> <p>tMap_1.ERROR_MESSAGE</p> <p>tFileOutputDelimited_1.ERROR_MESSAGE</p> <p>tFileOutputDelimited_1.NB_LINE</p> <p>tFileOutputDelimited_2.ERROR_MESSAGE</p> <p>tFileOutputDelimited_2.NB_LINE</p> <p>tFileInputDelimited_3.ERROR_MESSAGE</p> <p>tFileInputDelimited_3.NB_LINE</p> <p>tFlowMeter_1.ERROR_MESSAGE</p> <p>tFlowMeter_2.ERROR_MESSAGE</p>
---	---

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.

Voir également : *Définir les Contextes et les Variables*, page 112

Advanced settings

Certains composants, notamment les composants **File** et **Database**, 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.

Cet onglet propose aussi les options, notamment les statistiques avec StatCatcher. Pour plus d'informations concernant les fonctionnalités Stats & Log, consultez *Automatiser l'utilisation de statistiques & logs*, page 129.

View

L'onglet **View** de la fenêtre **Composant** vous permet de changer le format d'affichage par défaut d'un composant dans l'éditeur graphique.

Champ	Description
Label format	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.
Hint format	Bulle contextuelle apparaissant lorsque vous passez la souris au dessus du composant.
Show hint	Cochez cette case pour activer la fonctionnalité d'indication contextuelle.

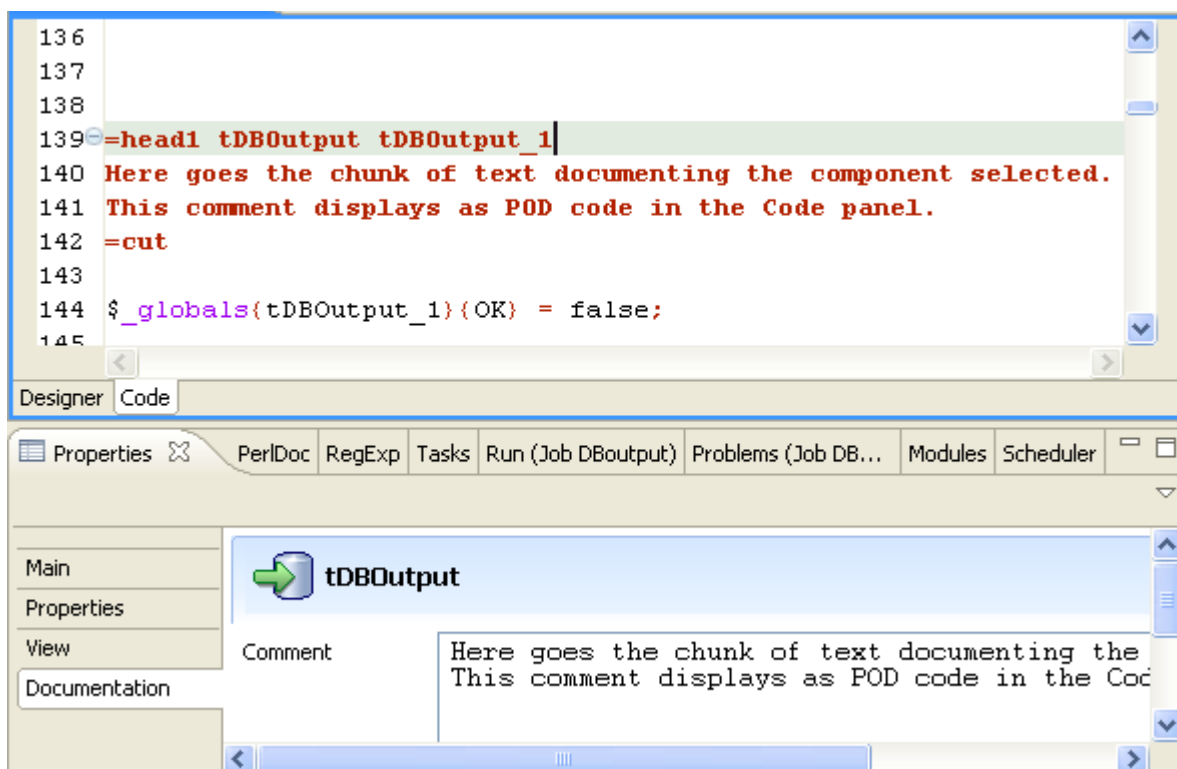
Vous pouvez personnaliser les textes des **Label** et **Hint** à l'aide des balises HTML suivantes :

- Gras : ` LibelléOuBulle `
- Italique : `<i> LibelléOuBulle </i>`
- Retour chariot : `LibelléOuBulle
 ContinueLigneSuiv`
- Couleur : ` LibelléOuBulle `

Pour changer vos préférences de l'onglet **View**, cliquez sur **Window > Preferences > Talend > Designer**.

Documentation

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



Le contenu de ce champ **Comment** sera formaté à l'aide du markup Pod et sera intégré au code généré sous forme de commentaires. Vous pouvez visualiser votre commentaire sous l'onglet **Code** de l'éditeur 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_`).

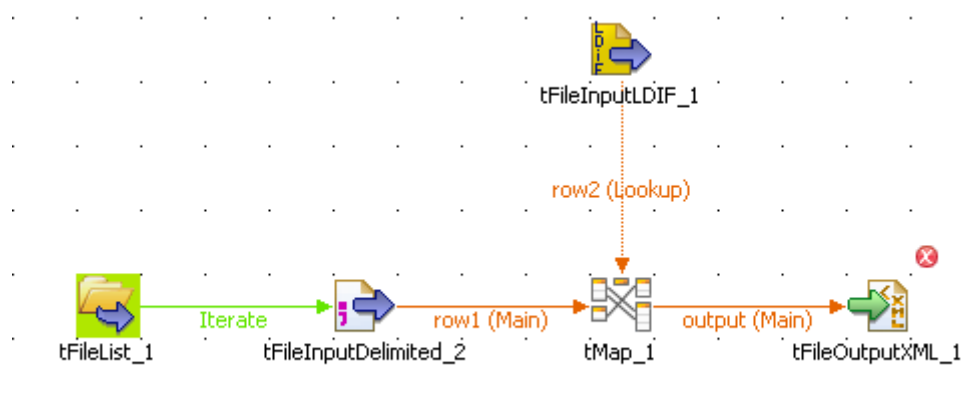
Pour une utilisation avancée de la fonction Documentation, il est préférable d'utiliser la partie Documentation du référentiel, qui vous permet de conserver et réutiliser tout type de documentation de façon centralisée.

Définition du composant Start

Le composant **Start** est le composant déclencheur des processus. Il peut y avoir plusieurs composants Start dans un job design si plusieurs traitements de flux s'exécutent en parallèle. Mais pour un même flux ainsi que les flux secondaires connectés, un seul composant peut être Start.

Cliquez et déposez un composant dans l'éditeur graphique. Tous les composants susceptibles d'être un composant Start, prennent une couleur d'arrière plan distinctive (vert). Vous pouvez remarquer que la plupart des composants peuvent être de composants **Start** et débiter un processus.

Seuls certains composants ne peuvent être de type start, notamment le composant tMap, par exemple.



Pour identifier quel composant débutera le processus, distinguez d'abord le flux principal des flux secondaires de votre job.

- Le flux principal doit connecter un composant à un autre à l'aide d'une connexion de type Row main. Le composant Start est alors automatiquement défini sur le premier composant du flux principal (icône dont l'arrière plan vert).
- Les flux secondaires sont connectés à l'aide de connexion de type Row lookup et non plus main. Cette connexion Lookup tient lieu de référence, qui enrichit le flux principal de nouvelles données.

Vous pouvez changer le statut du composant Start et par conséquent le flux principal et secondaire, en modifiant une connexion Row Main en connexion Lookup, simplement via un simple clic droit sur la connexion.

Voir également :

- *Connexion des composants, page 46*
- *Activer/Désactiver un job ou un sous-job, page 111*

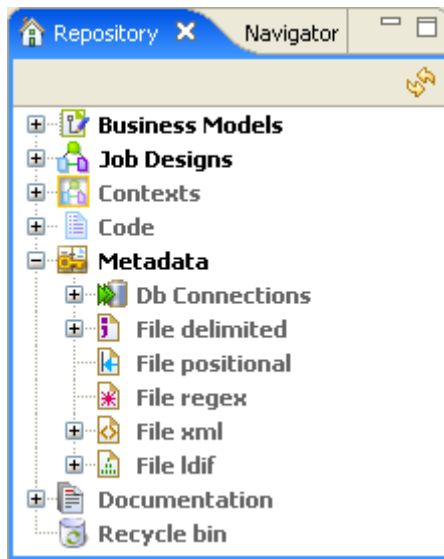
Définition des schémas de métadonnées

Talend Open Studio est une solution orientée métadonnées, et de ce fait, peut vous aider à garantir l'homogénéité et la qualité des données traitées par votre job design.

Sauvegardez vos métadonnées dans le référentiel pour pouvoir les réutiliser pour un autre composant. Ces métadonnées peuvent être : des connexions aux bases de données, des chemins d'accès et des schémas.

Deux étapes principales vous permettent de mettre en place des schémas de métadonnées à partir de fichiers ou de bases de données.

Dans un premier temps, configurez la connexion vers un fichier ou une base de données. Puis définissez le schéma basé sur une ou plusieurs tables de la base de données ou sur les métadonnées du fichier.



Cette procédure diffère légèrement selon le type de connexion choisie. Retrouvez ci-dessous les procédures respectives pour configurer et définir plusieurs schémas.

Cliquez sur Metadata dans le Repository pour développer l'arborescence.

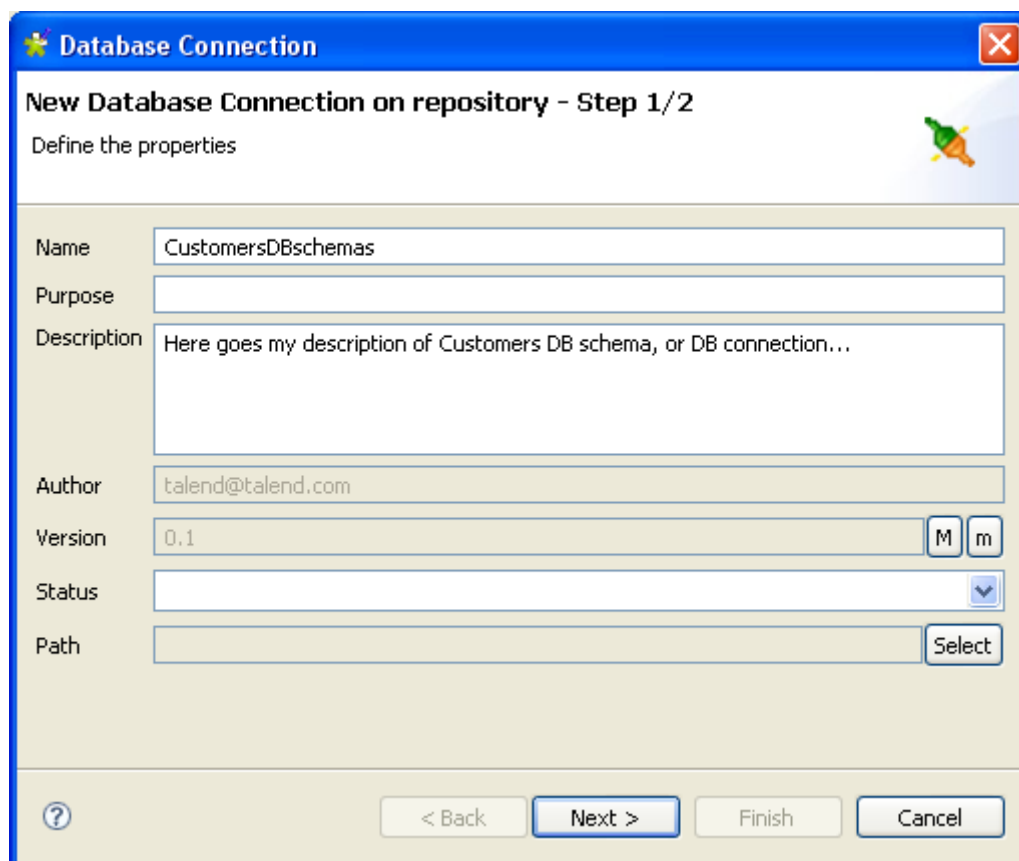
Chacun des noeuds regroupe différents types de connexions à configurer.

Configurer une connexion DB

Pour créer un fichier de schémas basés sur une base de données, suivez la procédure en deux phases, décrite ci-dessous. Cliquez droit sur **Db Connections** puis sélectionnez **Create connection** dans le menu contextuel.

Etape 1 : Propriétés générales

Un assistant de connexion s'ouvre. Renseignez les propriétés générales du schéma, notamment les champs **Name** et **Description**. Le champ **Status** est un champ personnalisé que vous pouvez définir dans les Préférences (*Window > Preferences*).



Database Connection

New Database Connection on repository - Step 1/2

Define the properties

Name: CustomersDBschemas

Purpose:

Description: Here goes my description of Customers DB schema, or DB connection...

Author: talend@talend.com

Version: 0.1 [M] [m]

Status: [v]

Path: [Select]

[?] < Back Next > Finish Cancel

Cliquez sur **Next** une fois terminé, l'étape suivante vous permet de renseigner les données de connexion à la BDD.

Etape 2 : Connexion

Sélectionnez le type de Base de données, champ **BD type**, à laquelle vous souhaitez vous connecter. Certains champs sont activés et d'autres grisés selon la nature de la connexion.

Database Connection

New Database Connection on repository - Step 2/2

i You must press the Check Button to check the Database Setting

Database Settings

DB Type: MySQL

String of Connection: jdbc:mysql://talend-dbms:3306/talend

Login: root

Password: •••••

Server: talend-dbms

Port: 3306

Database: talend

Schema:

DataSource:

MdbFile: Browse...

Check

Database Properties

SQL Syntax: SQL 92

String Quote: "

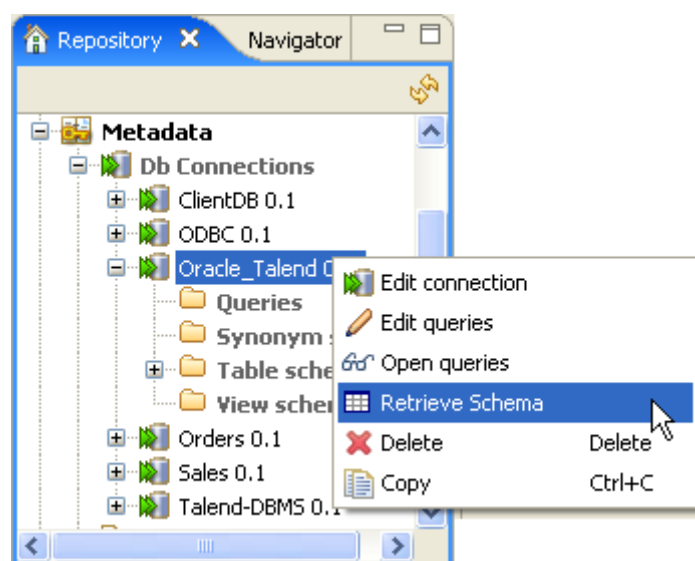
Null Char: 000

< Back Next > Finish Cancel

Renseignez les informations requises pour la connexion et vérifiez votre connexion à l'aide de la fonction **Check**.

Si nécessaire, remplissez les propriétés de base de données, dans la zone Database Properties. C'est tout pour la première phase de configuration d'une connexion base de données. Cliquez sur **Finish** pour valider.

La connexion base de données nouvellement créée est accessible depuis le Repository et contient quatre sous-dossiers dont **Queries** pour les requêtes SQL et **Table schemas** qui regroupe tous les schémas relatifs à cette connexion.



A présent, cliquez droit sur la connexion nouvellement créée et sélectionnez **Retrieve schema** dans le menu contextuel.

Etape 3 : Chargement de la table

Un nouvel assistant s'ouvre sur la première étape. La liste de tables propose toutes les tables présentes sur la connexion définie précédemment. Il peut s'agir de n'importe quel type de BDD.

Sélectionnez une ou plusieurs tables dans la liste pour les charger sur votre système de fichier Référentiel. Vos schémas de Repository seront basés sur ces tables.

Si aucun schéma n'est visible dans la liste, cliquez sur **Check connection** pour vérifier le statut de la connexion à la base de données.

Cliquez sur **Next** une fois la sélection terminée. Sur la fenêtre suivante, quatre panneaux de configuration vous permettent de définir les schémas à créer.

En Java, assurez-vous que le type de données est correctement défini. Pour plus d'informations concernant les types de données, formats de dates inclus, visitez le site <http://java.sun.com/j2se/1.5.0/docs/api/index.html>.

Etape 4 : Définition du schéma

Par défaut, le schéma affiché dans le panneau **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.

En effet, la barre d'outils vous permet d'ajouter, d'enlever ou déplacer les colonnes de votre schéma. Vous avez également la possibilité de charger un schéma XML ou exporter le schéma courant en XML.

Pour restaurer un schéma basé sur l'une des tables chargés, sélectionnez le schéma de la table dans la liste déroulante et cliquez sur **Retrieve schema**. Notez que la schéma ainsi chargé écrase le schéma courant et ne retient aucune des modifications ponctuelles.

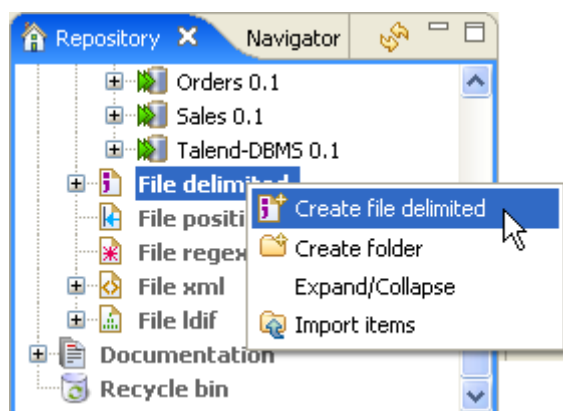
Cliquez sur **Finish** pour terminer la création des schémas de base de données. Tous les schémas ainsi créés apparaissent sous le noeud de connexion de base de données.

Configurer un schéma File Delimited

Les métadonnées de type File Delimited peuvent être utilisées pour définir les propriétés des composants InputFileDelimited et InputFileCSV car tous deux sont basés sur la même structure.

WARNING—La création de schémas de fichier est similaire pour tous les types de fichiers : *Delimited* (délimités), *Positional* (positionnels), *Regex* (Expr. régulières), *Xml* ou *Ldif*.

Dans le référentiel (Repository), cliquez droit sur File Delimited dans l'arborescence et sélectionnez **Create file delimited** dans le menu contextuel.



A la différence de l'assistant de connexion DB, l'assistant de connexion Delimited File regroupe la connexion au fichier et la définition du schéma, en une seule procédure à quatre étapes.

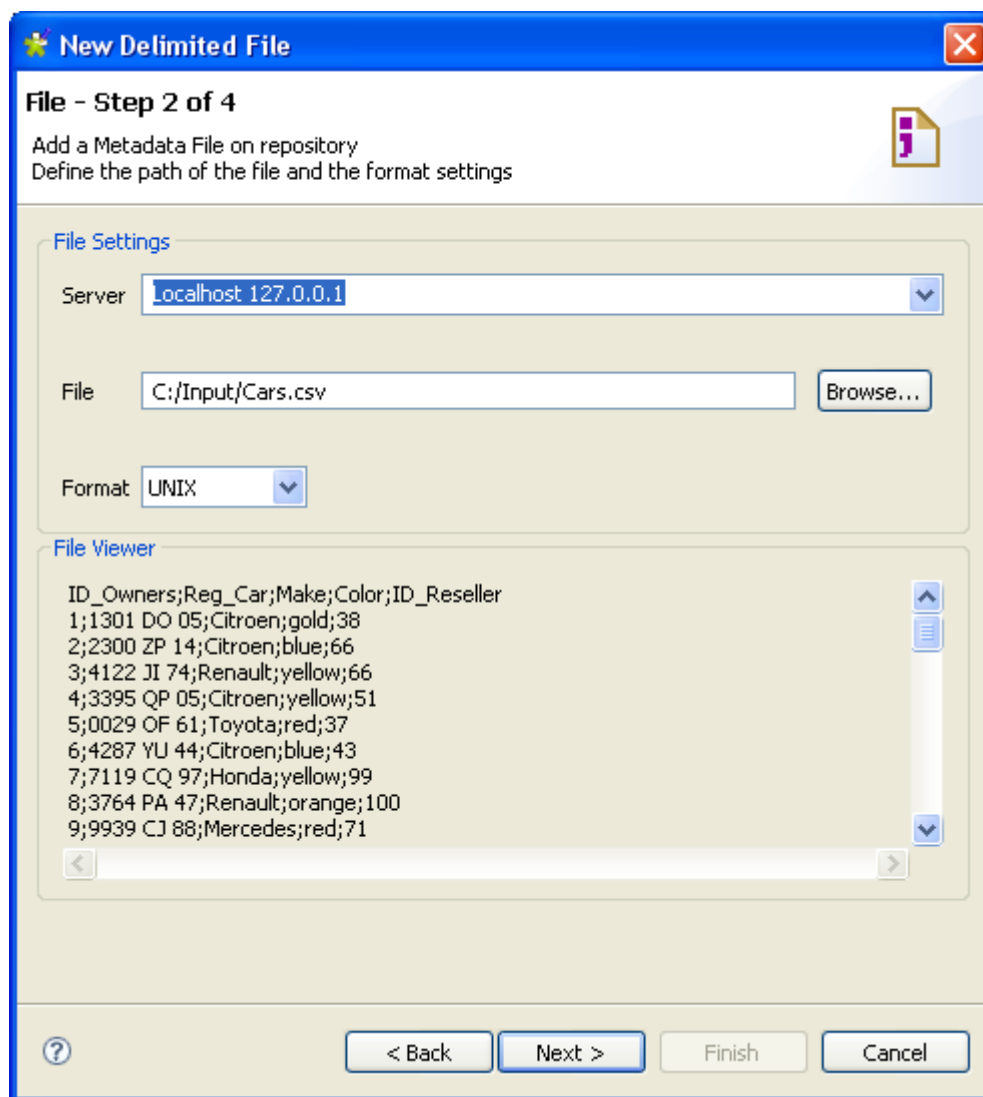
Etape 1 : Propriétés générales

Pour la première étape, renseignez les informations générales du schéma, notamment les champs **Name** et **Description**.

Pour plus d'informations, consultez *Etape 1 : Propriétés générales*, page 59.

Etape 2 : Chargement du fichier

Renseignez l'adresse IP du serveur où le fichier est conservé. Et cliquez sur **Browse...** pour définir le chemin d'accès au fichier, dans le champ **File path**.



Sélectionnez le **Format** du système d'exploitation dans lequel le fichier a été créé. Cette information est utilisée pour pré-remplir les champs des étapes suivantes. Si la liste ne propose pas le format approprié, ignorez ce champ.

Le **File viewer** donne un instantané du fichier chargé. Il vous permet de vérifier la cohérence du fichier, la présence d'un en-tête et plus généralement la structure du fichier.

Cliquez sur **Next** pour passer à l'étape 3.

Etape 3 : Définition du schéma

Dans cette vue, vous pouvez affiner la destination de vos données et les paramètres de fichier. Cliquez sur les rectangles pour des informations spécifiques.

New Delimited File

File - Step 3 of 4
Add a Metadata File on repository
Define the setting of the parse job

File Settings

Encoding: US-ASCII

Field length: Semicolon Corresponding Character: ;

Row Separator: Standard EOL Corresponding Character: \n

Escape Char Settings

☐ CSV ☒ Delimited

Escape Char: Empty

Text Enclosure: Empty

Rows To Skip

If any rows must be ignored, specify the following parameters

Header ☒ 6

Footer ☐

☐ Skip empty row

Limit Of Rows

If the number of lines must be limited, specify this number

Limit ☐

Preview

☒ Set heading row as column names Refresh Preview

id	CustomerName	CustomerAddress	idReseller	Make	RegTime	Column 6	Column 7
1	Griffith Paving and Sealcoat	355 Golf Rd.	7	1	03/11/1991 09:20	1973-01-17 06:26:40.000	67852.
2	Bill's Dive Shop	511 Maple Ave. Apt. 1B	35	1	19/11/1984 15:48	2010-06-07 09:40:00.000	88792.
3	Childress Child Day Care	662 Lyons Circle	39	1	16/02/1981 08:27	1990-04-01 21:00:00.000	35340.
4	Facelift Kitchen and Bath	220 Vine Ave.	41	1	22/08/2010 09:55	1972-04-23 18:00:00.000	6097.8
5	Terrinni & Son Auto and Truck	770 Exmoor Rd.	5	1	28/06/1995 09:15	1982-04-19 10:26:40.000	5146.5
6	Kermit the Pet Shop	1860 Parkside Ln.	28	1	17/08/2009 10:07	1991-05-27 17:00:00.000	16087.
7	Tub's Furniture Store	807 Old Trail Rd.	15	1	27/08/1976 03:13	1970-03-27 23:08:16.000	53216.
8	Toggle & Myerson Ltd	618 Sheriden rd.	9	1	24/03/2008 23:07	1981-08-02 01:26:40.000	74168.
9	Childress Child Day Care	700 Terrance Ave	12	1	16/02/1981 08:27	1990-04-01 21:00:00.000	35340.

< Back Next > Finish Cancel

Paramétrez l'**Encoding**, ainsi que les séparateurs de champs et de lignes (**Field separator** et **Row separator**) dans les paramètres Delimited File.

The image shows two panels from the Talend Open Studio interface. The top panel, titled 'File Settings', contains three rows of configuration: 'Encoding' set to 'US-ASCII', 'Field length' set to 'Semicolon' with a 'Corresponding Character' of ';', and 'Row Separator' set to 'Standard EOL' with a 'Corresponding Character' of '\n'. The bottom panel, titled 'Escape Char Settings', has two radio buttons: 'CSV' (selected) and 'Delimited'. Below these are two dropdown menus: 'Escape Char' set to 'Empty' and 'Text Enclosure' set to 'Empty'.

En fonction de votre type de fichier (csv ou delimited), vous pouvez également paramétrer des caractères d'inclusion et d'exclusion (respectivement **Text Enclosure** et **Escape character**).

Si l'aperçu du fichier indique un message d'en-tête, vous pouvez exclure cet en-tête de l'analyse. Indiquez le nombre de lignes d'en-tête à ignorer. En outre, si le fichier contient des informations de pied de page, indiquez le nombre de lignes à ignorer.

The image shows two panels from the Talend Open Studio interface. The top panel, titled 'Rows To Skip', includes the instruction 'If any rows must be ignored, specify the following parameters'. It has a 'Header' checkbox checked with a value of '3', a 'Footer' checkbox unchecked, and a 'Skip empty row' checkbox unchecked. The bottom panel, titled 'Limit Of Rows', includes the instruction 'If the number of lines must be limited, specify this number' and a 'Limit' checkbox which is unchecked.

Le champ **Limit of rows** vous permet de restreindre l'étendue du fichier qui fait l'analyse.

Dans le panneau File Preview, vous pouvez visualiser l'impact de votre paramétrage.

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.

Preview

☒ Set heading row as column names

ID	Registration	Make	Color	Reseller ID	Name	Insurance	
1	5776 ZQ 94	Volkswagen	gold	7	montmont	KVW2844	
3	2580 TT 77	Renault	orange	1	bouhnan	BNU9147	
4	1722 VE 11	Citroen	silver	10	kikla	TEU8220	

Cliquez sur **Refresh** dans le panneau d'aperçu, afin que les modifications de paramétrage prennent effet.

Etape 4 : Schéma final

La dernière étape affiche le schéma de fichier délimité généré. Vous pouvez personnaliser le tableau.

New Delimited File

File - Step 4 of 4

Add a Schema on repository
Define the Schema

Name: metadata

Comment:

Schema

Click to update schema preview **Guess**

Description of the Schema

Column	Key	Type	Nullable	Length	Preci...	D...	Comm...
ID_Owners	<input type="checkbox"/>	int	<input checked="" type="checkbox"/>	2			
Reg_Car	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>	10			
Make	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>	10			
Color	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>	6			
ID_Reseller	<input type="checkbox"/>	int	<input checked="" type="checkbox"/>	3			

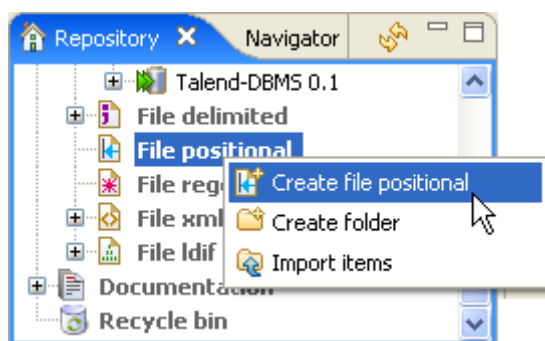
Navigation: ? < Back Next > Finish Cancel

Si le fichier délimité sur lequel est basé le schéma est modifié, utilisez le bouton **Guess** pour re-générer le schéma. Notez que si vous personnalisez le schéma, la fonctionnalité Guess ne retiendra pas ces modifications.

Cliquez sur **Finish** pour valider. Le nouveau schéma apparaît dans l'arborescence du Repository, sous le noeud de la connexion File Delimited appropriée.

Configurer un schéma File Positional

Dans le Repository, cliquez droit sur File Positional dans l'arborescence, puis sélectionnez **Create file positional** dans le menu contextuel.



Procédez de la même manière que pour une connexion au fichier délimité. Cliquez droit sous Metadata dans le Repository, puis sélectionnez **Create file positional**.

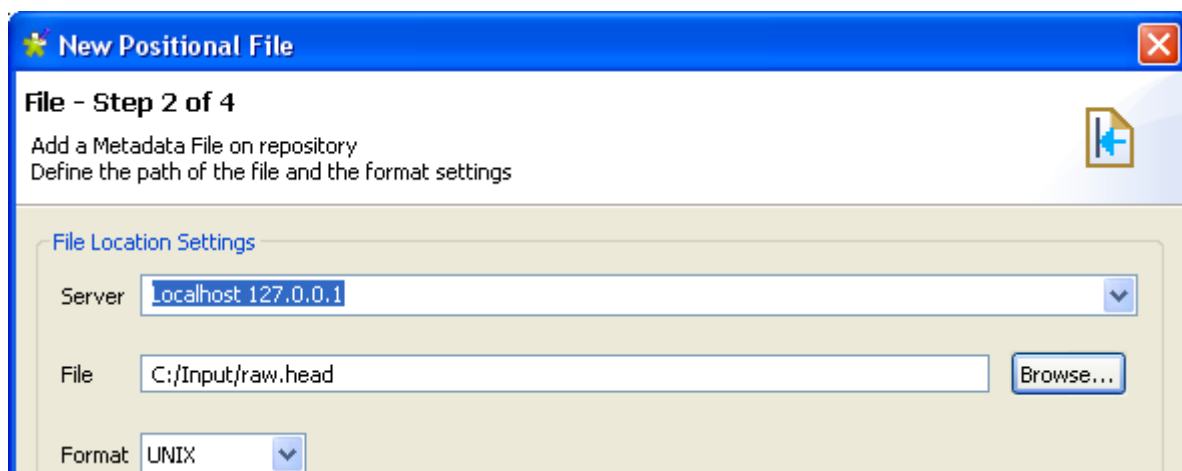
Etape 1 : Propriétés générales

Renseignez les informations générales du schéma, notamment **Name** et **Description**.

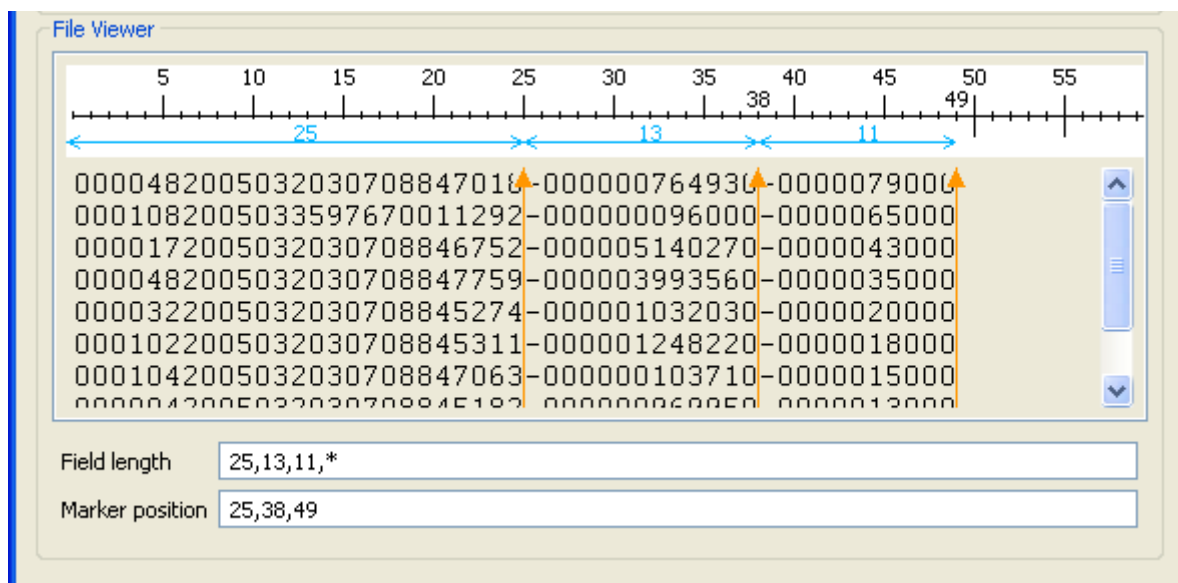
Etape 2 : Connexion et chargement du fichier

Puis définissez les paramètres de connexion au fichier positionnel, en renseignant les champs **Server IP address** et **File path**.

Comme pour la création d'un schéma Delimited File, le format est demandé pour pré-remplir des champs dans les étapes suivantes. Si le système d'exploitation de création du fichier n'est pas proposé dans la liste, ignorez ce champ.



L'aperçu du fichier donne un instantané du fichier et vous permet de placer les marqueurs de position.



Cliquez sur l'aperçu du fichier et définissez la position des marqueurs sur la règle. Une flèche de couleur orange vous aide à affiner la position du marqueur.

Le séparateur de champ (**Field length**) répertorie une série de chiffres séparés par des virgules. L'astérisque symbolise tous les caractères restants jusqu'à la fin de la ligne, à partir du marqueur précédent.

La position de champ (**Marker Position**) indique la position exacte du marqueur sur la règle. Vous pouvez affiner la position exacte du marqueur en saisissant la valeur exacte de position.

Vous pouvez placer autant de marqueurs que nécessaire. Pour retirer un marqueur, faites-le glisser vers la règle.

Cliquez sur **Next** pour continuer.

Etape 3 : Affinage du schéma

L'étape suivante ouvre la fenêtre de paramétrage du schéma. De la même manière que pour le schéma de fichier délimité, vous pouvez affiner la définition du schéma en précisant les séparateurs de champs et de lignes (**Field separator** et **Row separator**), le nombre de lignes d'en-tête, etc.

A ce stade, l'aperçu affiche le fichier délimité par la position de marqueurs. Si le fichier contient des libellés de colonne, cochez la case **Set heading row as column names**.

Etape 4 : Schéma final

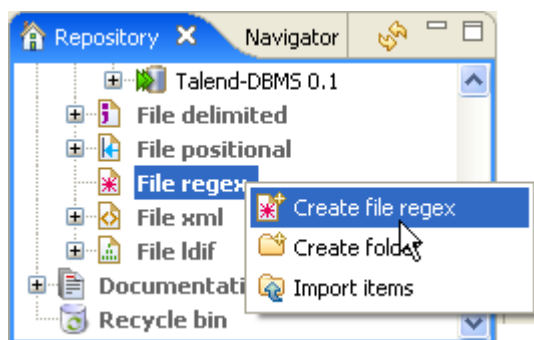
L'étape 4 affiche le schéma généré final. Remarquez que tous les caractères du fichier pouvant être mal interprété par le programme par la suite, sont remplacé par un caractère neutre. Par exemple, les tirets soulignés remplacent les astérisques.

Vous pouvez ajouter un nom personnalisé (par défaut, metadata) et apporter des modifications à l'aide de la barre d'outils.

Vous pouvez également restaurer ou mettre à jour le schéma du fichier positionnel en cliquant sur **Guess**. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur **Guess**.

Configurer un schéma File Regex

Les schémas de fichier Regex servent pour manipuler les fichiers dont les informations sont redondantes, notamment les fichiers log.



Procédez de la même façon que pour les connexions de fichiers délimités ou positionnels. Cliquez droit sur Metadata dans le Repository et sélectionnez **Create file regex**.

Etape 1 : Propriétés générales

Renseignez les informations générales du schéma, notamment **Name** et **Description**.

Etape 2 : Chargement du fichier

Puis définissez les paramètres de connexion au fichier Regex, en renseignant les champs **Server IP address** et **File path**.

The screenshot shows two panels from the Talend Open Studio interface. The top panel, titled 'File Settings', contains three fields: 'Server' with a dropdown menu showing 'localhost 127.0.0.1', 'File' with a text box containing 'C:\Input\Eclipse.log' and a 'Browse...' button, and 'Format' with a dropdown menu showing 'UNIX'. The bottom panel, titled 'File Viewer', displays the contents of the file 'Eclipse.log'. The text in the viewer is as follows:

```

***** Test *****
***** Test *****
!ENTRY org.talend.designer.runprocess 1 0 2006-09-07 10:38:55.050
!ENTRY org.talend.designer.runprocess 1 0 2006-09-07 10:39:25.537
!ENTRY org.talend.designer.runprocess 1 0 2006-09-07 10:39:28.430
!ENTRY org.talend.designer.runprocess 1 0 2006-09-07 10:39:51.377
    
```

Comme pour la création d'un schéma Delimited File, le format est demandé pour pré-remplir certains champs des étapes suivantes. Si le système d'exploitation de création du fichier n'est pas proposé dans la liste, ignorez ce champ.

L'aperçu de fichier fournit un instantané du fichier chargé. Cliquez sur **Next** pour définir la structure du schéma.

Etape 3 : Définition du schéma

Cette étape ouvre la fenêtre de paramétrage du schéma. Comme pour les autres schémas de fichier, vous pouvez affiner la définition du schéma en précisant les séparateurs de champs et de lignes (**Field separator** et **Row separator**), le nombre de lignes d'en-tête, etc.

Dans le panneau de paramétrage des expressions régulières, **Regular Expression settings**, saisissez l'expression régulière utilisée pour délimiter le fichier.

The screenshot shows the 'Regular Expression settings' dialog box. It has a title bar 'Regular Expression settings' and a large text area containing the following regular expression:

```

^!ENTRY\s+(\w+(?:\.\w+)*)\s+\d+\s+\d+\s+(\d{4}-\d{2}-\d{2})
    
```



Utilisez le bon langage Regex en fonction du langage de génération de code utilisé dans votre projet (Java ou Perl), ainsi que les bons guillemets (simples ou doubles).

Puis cliquez sur **Refresh preview** pour que les modifications soient prises en compte. Le bouton change d'apparence en **Wait** jusqu'à ce que l'aperçu soit rafraîchi.

Preview

☐ Set heading row as column names

Column 0	Column 1	
org.talend.designer.runprocess	2007-02-07	
org.talend.designer.runprocess	2007-02-07	
org.talend.designer.runprocess	2007-02-07	
org.talend.designer.runprocess	2007-02-07	

Cliquez sur **Next** une fois le paramétrage terminé. La dernière étape génère un schéma de fichier Regex.

Etape 4 : Schéma final

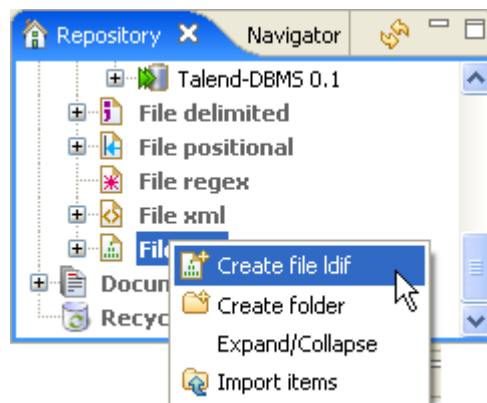
Vous pouvez ajouter un nom personnalisé (par défaut, metadata) et apporter des modifications à l'aide de la barre d'outils.

Vous pouvez également restaurer ou mettre à jour le schéma du fichier Regex en cliquant sur **Guess**. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur Guess.

Cliquez sur **Finish** pour valider. Le nouveau schéma apparaît dans le référentiel, sous le noeud correspondant à la connexion Regex appropriée.

Configurer un schéma File LDIF

Les fichiers LDIF sont des fichiers annuaires distincts par leurs attributs. Les métadonnées FileLDIF centralisent ces fichiers de type LDIF ainsi que leur description d'attributs.



Procédez de la même façon que pour les connexions de fichiers délimités ou positionnels. Cliquez droit sur **Metadata** dans le **Repository** et sélectionnez **Create file Ldif**.

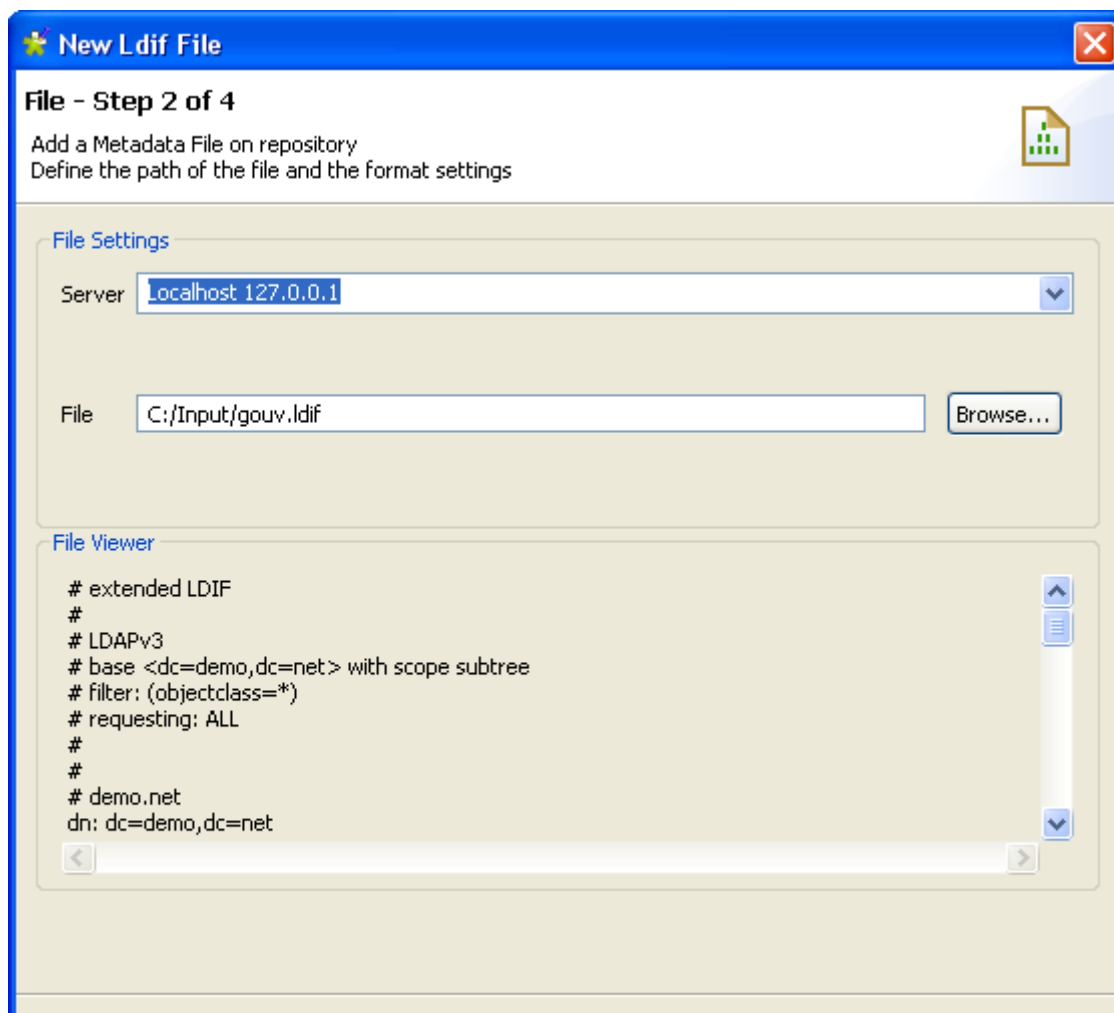
Note: Assurez-vous que les modules Perl requis pour l'utilisation de cette fonctionnalité sont bien installés. Pour plus d'informations, visitez <http://talendforge.org/wiki/doku.php>

Etape 1 : Propriétés générales

Pour commencer, renseignez les informations génériques, notamment **Name** et **Description**.

Etape 2 : Chargement du fichier

Puis définissez les paramètres de connexion au fichier Ldif en renseignant le champ **File path**.



Note: La fonctionnalité de connexion à un serveur distant n'est pour l'instant pas opérationnel pour la récupération de fichier LDIF.

L'aperçu de fichier affiche les 50 premières lignes du fichier.

Etape 3 : Définition du schéma

La liste des attributs du fichier de description s'affiche en haut de la fenêtre. Cochez les cases correspondant aux attributs que vous souhaitez extraire du fichier de description LDIF.

New Ldif File

File - Step 3 of 4

Add a Metadata File on repository
Define the setting of the parse job

List Attributes of Ldif file

Attributes
<input type="checkbox"/> password
<input checked="" type="checkbox"/> name
<input type="checkbox"/> changetype
<input checked="" type="checkbox"/> lastname

Preview

Refresh Preview

dn	name	lastname
jbzy	jbzy	taus
fkcd	fkcd	kznj
beyy	beyy	yhgw
kskb	kskb	vuxx
fpfc	fpfc	ryzy
doqp	doqp	eumq
jpxi	jpxi	ocnb

Cliquez sur **Refresh Preview** pour prendre en compte dans l'aperçu du fichier, les attributs sélectionnés.

Note: DN est volontairement omis de la liste des attributs proposés car cet attribut clé est automatiquement inclus dans l'aperçu de fichier, et par conséquent dans le schéma généré.

Etape 4 : Schéma final

Le schéma généré affiche les colonnes du fichier de description. Vous pouvez le personnaliser selon vos besoin ou recharger le schéma original à l'aide du bouton **Guess**.

Cliquez sur **Finish**. Le nouveau schéma s'affiche comme une nouvelle entrée de l'arborescence du Repository sous le noeud de connexion LDif approprié.

Configurer un Schéma FileXML

Centralisez vos requêtes XPath sur un fichier XML défini et regroupez les valeurs récupérées.

Procédez de la même façon que pour les connexions de fichier délimité ou positionnel. Cliquez droit sur Metadata dans le Repository et sélectionnez **Create file XML**.

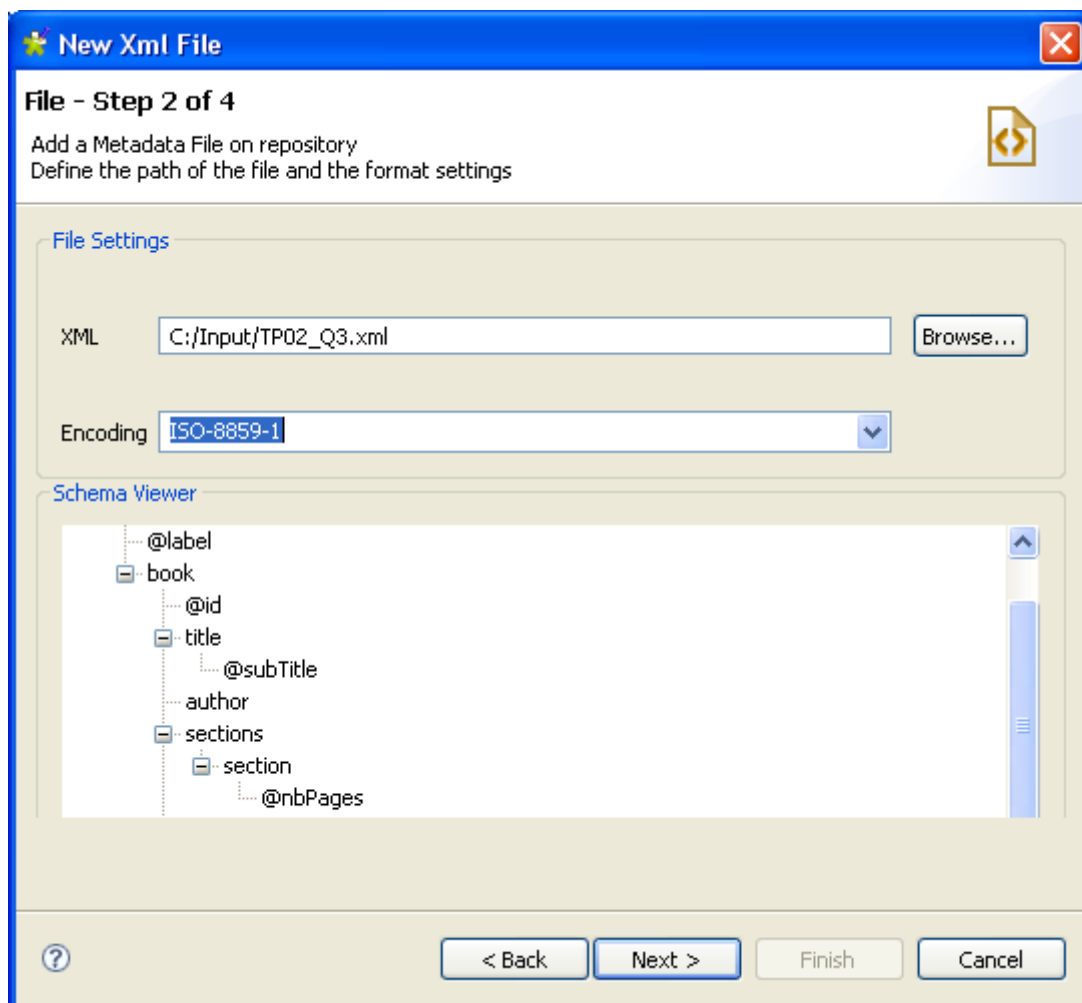
Etape 1 : Propriétés générales

Pour commencer, renseignez les informations génériques, notamment **Name** et **Description**.
Puis cliquez sur **Next**.

Etape 2 : Chargement du fichier

Renseignez le chemin d'accès au fichier XML à charger, puis renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement.

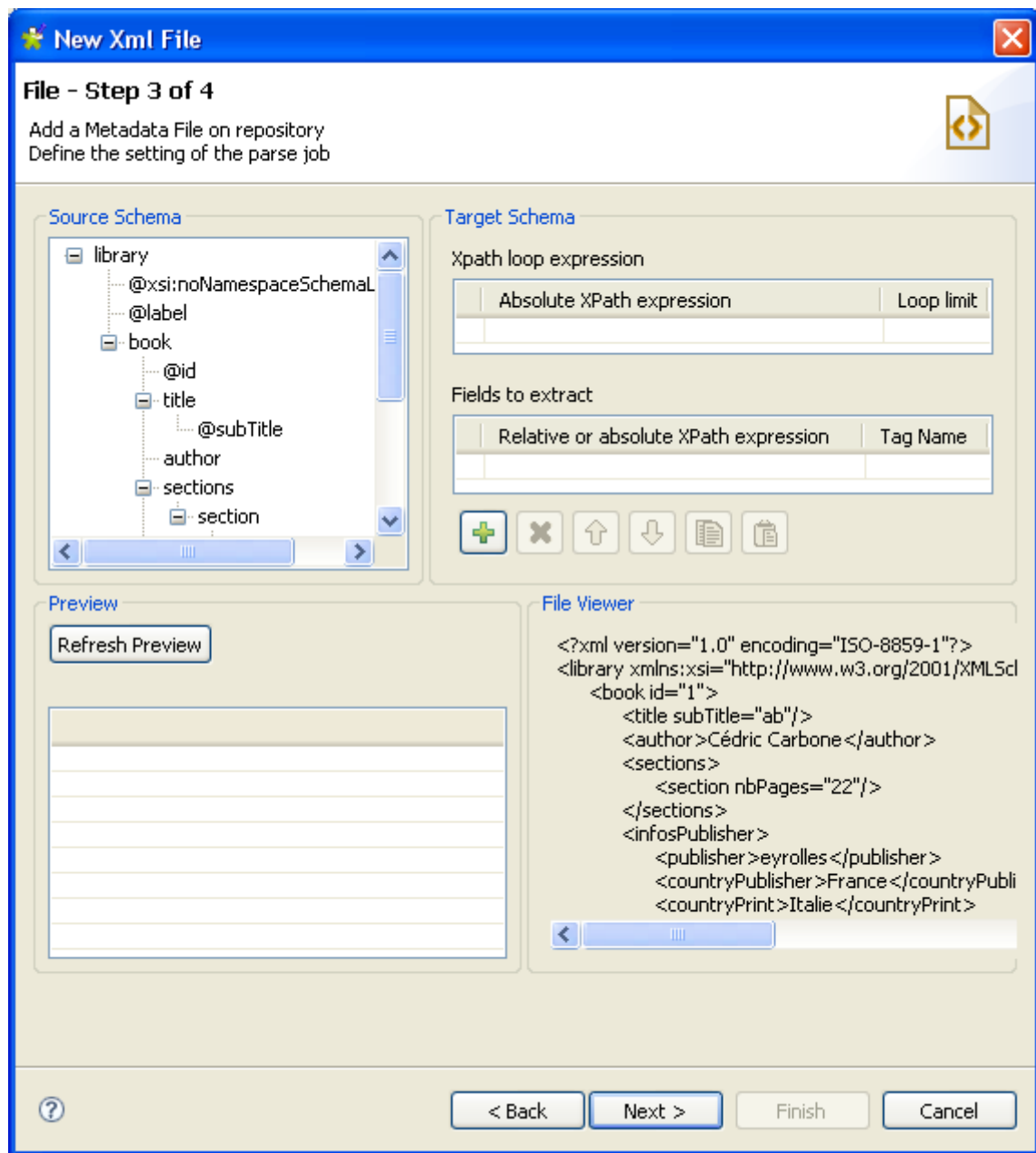
L'aperçu de fichier affiche la structure XML.



Cliquez sur **Next** pour continuer.

Etape 3 : Définition du schéma

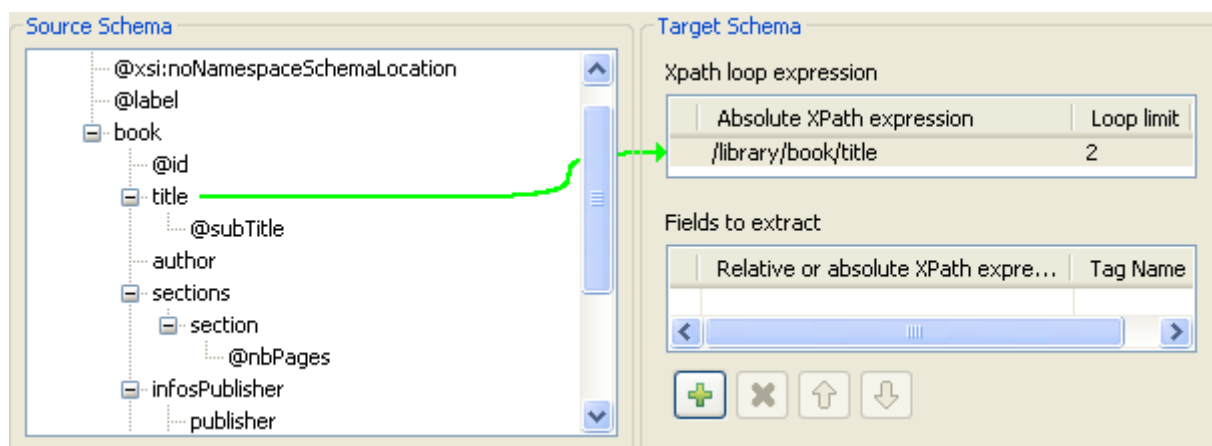
Renseignez les paramètres à prendre en compte pour la définition du schéma.



La fenêtre de définition du schéma se compose de quatre vues :

- **Source Schema** : Arborescence du fichier XML
- **Target Schema** : Information d'extraction et d'itération
- **Preview** : Aperçu du schéma cible
- **File viewer** : Fenêtre d'aperçu des données brutes

Dans le champ **Xpath loop expression**, saisissez l'expression de Xpath absolu du noeud qui fait l'objet de l'itération. Vous pouvez saisir l'expression entière ou appuyez sur Ctrl+Espace pour utiliser la liste d'autocomplétion.

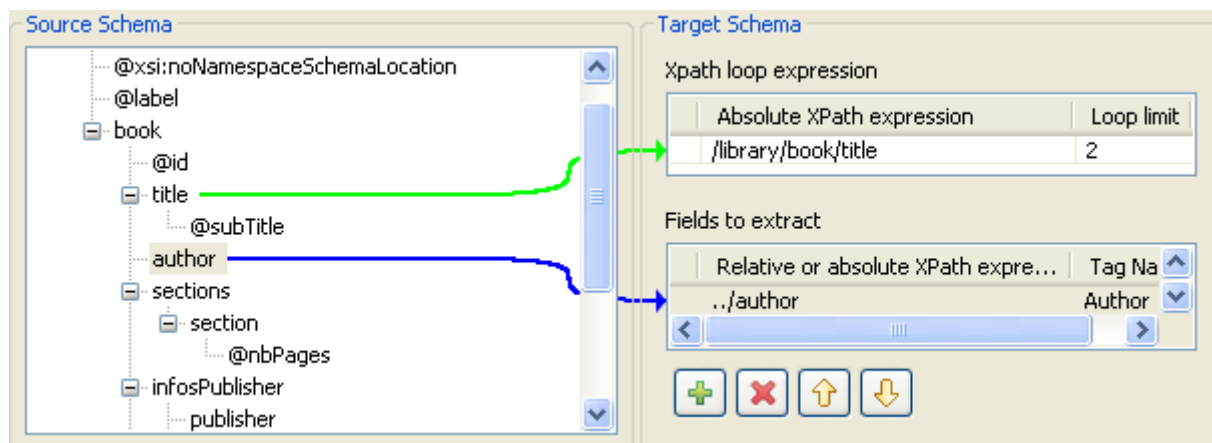


Alternativement, glissez et déposez le noeud sélectionné dans la structure source vers le champ Xpath du schéma cible.

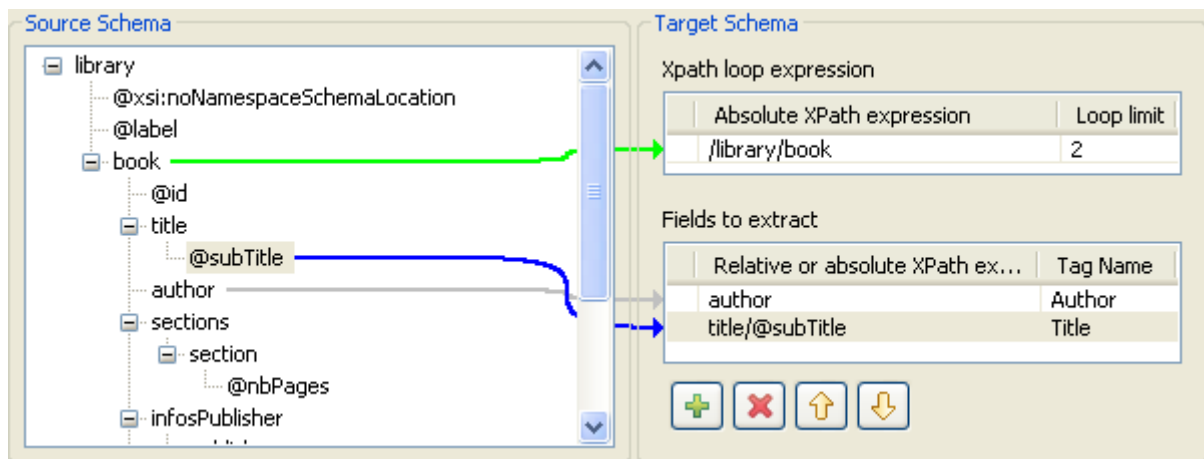
Note: Le champ **Xpath loop expression** est obligatoire.

Vous pouvez également définir une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de noeuds à itérer. Un lien de couleur orange est ainsi créé.

Puis définissez les champs à extraire en glissant et déposant les noeuds sélectionnés vers le champ **Relative or absolute XPath expression**.



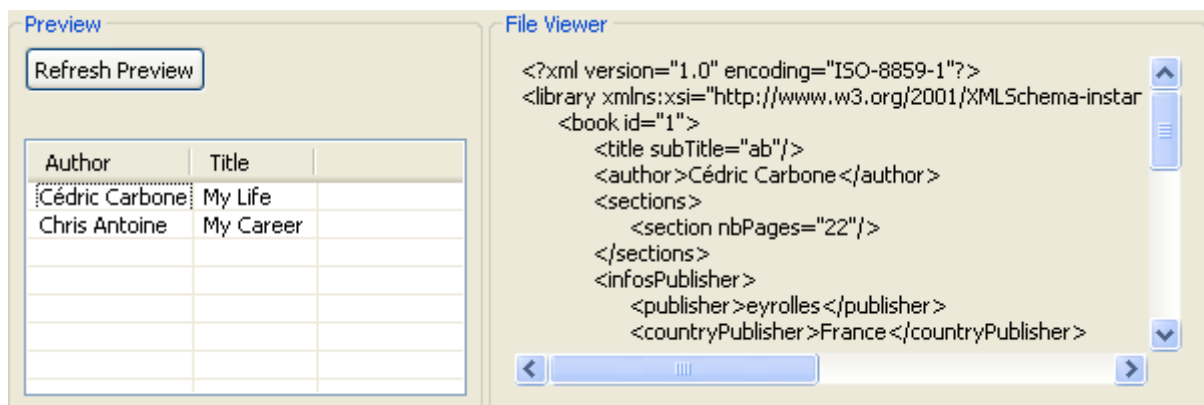
A l'aide du signe (+), ajoutez des lignes à la table et sélectionnez autant de champs à extraire que vous le souhaitez. Appuyez sur Ctrl ou Maj pour la sélection multiple des noeuds, qu'ils soient groupés ou non. Puis glissez et déposez-les sur la table.



Dans le champ **Tag name**, nommez les libellés de colonnes qui s'afficheront dans la vue d'aperçu du schéma (Schema preview).

Le lien de la sélection s'affiche de couleur bleue, tous les autres liens sont gris. Vous pouvez imposer un ordre dans les champs à extraire à l'aide des flèches haut et bas.

Cliquez sur **Refresh preview** pour afficher l'aperçu du schéma. Les champs apparaissent ainsi dans l'aperçu du schéma dans l'ordre imposé.



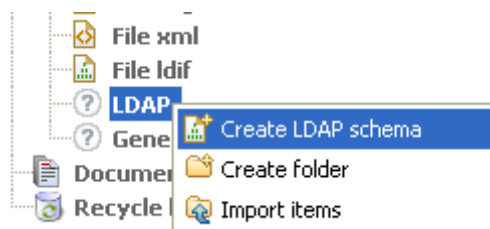
Etape 4 : Schéma final

Le schéma généré affiche les colonnes sélectionnées dans le fichier XML. Vous pouvez personnaliser ce schéma selon vos besoins ou recharger le schéma d'origine à l'aide du bouton **Guess**.

Cliquez sur **Finish**. Le nouveau schéma s'affiche dans le Repository sous le noeud File XML approprié.

Configurer un schéma LDAP

Dans le **Repository**, cliquez droit sur le noeud **LDAP**, puis sélectionnez **Create LDAP schema** dans le menu contextuel.



Contrairement à l'assistant de connexion DB, l'assistant LDAP regroupe à la fois la connexion au fichier et la définition du schéma.

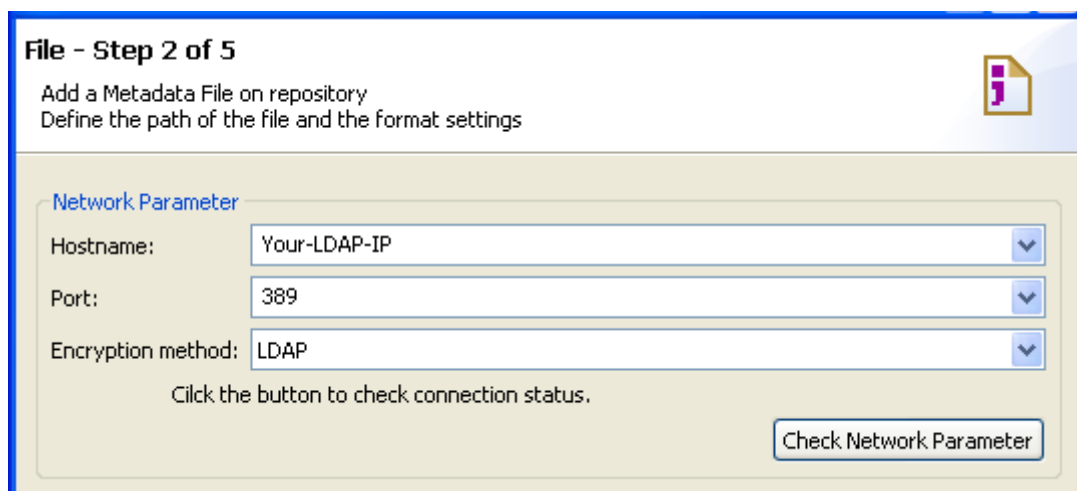
Etape 1 : Propriétés générales

Pour la première étape, renseignez les informations générales du schéma, notamment les champs **Name** et **Description**.

Pour plus d'informations, consultez *Etape 1 : Propriétés générales*, page 59.

Etape 2 : Connexion au serveur

Renseignez les informations de connexion.



Vérifiez votre connexion à l'aide de bouton **Check Network Parameter**, puis cliquez sur le bouton **Next**.

Champ	Description
Hostname	Serveur LDAP adresse IP
Port	Port d'écoute annuaire LDAP
Encryption method	LDAP : aucun codage n'est utilisé LDAPS : LDAP sécurisé TLS : le certificat est utilisé

Cliquez sur **Next** pour valider cette étape et continuer.

Etape 3 : Authentication et acquisition du DN

Dans cette fenêtre, renseignez les informations d'authentification et mode d'accès aux données.

Create new LDAP schema

File - Step 3 of 5

Add a Metadata File on repository
Define the setting of the parse job

Authentication Method
Simple Authentication

Authentication Parameter
Bind DN or user: cn=Directory Manager
Bind password:
☒ Save password
Check Authentication

Base DN
☒ Get base DN's from Root DSE
Fetch Base DN's
Base DN: o=directoryRoot

Aliases Dereferencing
☐ Finding
☐ Searching
☐ Never
☒ Always

Referrals Handling
☒ Ignore
☐ Follow

Limits
Count Limit: 100
Time Limit: 0

< Back Next > Finish Cancel

Cliquez sur **Check authentication** pour vérifier vos droits d'accès.

Champ	Description
Authentication method	Pour une Simple authentication , renseignez le champ Authentication Parameters 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 de login.
Get Base DN from Root DSE / Base DN	Chemin d'accès à l'arborescence de l'utilisateur Le bouton Fetch Base DNs redirige le DN automatiquement à la racine.
Alias Dereferencing	Never permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez Always . Always : les alias sont toujours déréférencés. Never : les alias ne jamais sont déréférencés. Searching : cette option déréférence les alias uniquement après la résolution du nom. Finding : cette option déréférence les alias uniquement lors de la résolution du nom.
Referral Handling	Redirection des requêtes utilisateur : Ignore ne tient pas compte des redirections des requêtes Follow tient compte des redirections des requêtes
Limit	Limite le nombre d'enregistrements lus

Cliquez sur **Fetch Base DNs** pour conserver le DN et cliquez sur **Next** pour continuer.

Etape 4 : Définition du schéma

Sélectionnez les attributs que vous souhaitez inclure dans le structure du schéma.

Ajoutez un filtre si vous souhaitez uniquement des données particulières.

Create new LDAP schema

File - Step 4 of 5

Add a Metadata File on repository
Define the setting of the parse job

List attributes of LDAP Schema

Attributes

- ☐ description
- ☐ userpassword
- ☒ uid
- ☐ sn
- ☐ cn
- ☐ ...

Filter

(&(objectClass=*))

Preview

Refresh Preview

uid	mail	givenname	telephonenumber
PIERRE DUPONT	Pierre.Dupont@talend.com	PIERRE	00149684750
PIERRE DUPON...	mhirt78@talend.com	PIERRE	
mhirt			
greg			
..			

< Back Next > Finish Cancel

Cliquez sur **Refresh Preview** pour afficher les colonnes sélectionnées ainsi qu'un échantillon des données.

Puis cliquez sur **Next** pour continuer.

Etape 5 : Schéma final

Cette dernière étape affiche le schéma LDAP généré. Vous pouvez modifier le schéma à l'aide de la barre d'outils sous le tableau.

Create new LDAP schema

File - Step 5 of 5

Add a Metadata File on repository
Define the setting of the parse job

Name: metadata

Comment:

Schema

Click to update schema preview

Description of the Schema

Column	Key	Type	Nullable	Date ...	Length	Precision
uid	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	
mail	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		24	
givenname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		6	
telephonenumber	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		11	

Navigation icons: +, -, up, down, list, clipboard, folder, refresh.

Buttons: ? < Back Next > Finish Cancel

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**.

Cliquez sur **Finish**. Le nouveau schéma s'affichera dans le **Repository**, sous le noeud LDAP correspondant.

Configurer un schéma générique

Talend Open Studio vous permet de créer entièrement un nouveau schéma si aucun des schémas proposés ne correspond à vos besoins ou si vous n'avez pas de schémas déjà constitués. La procédure de création se fait en deux étapes :

Cliquez droit sur **Generic Schema** dans le Repository et sélectionnez **Create generic schema**.

Etape 1 : Propriétés générales

Un assistant de connexion s'ouvre. Renseignez les informations générales de schéma, notamment **Name** et **Description**. Le champ **Status** est un champ optionnel, vous pouvez le modifier dans *Window > Preferences*.

Cliquez sur **Next** pour continuer.

Etape 2 : Définition du schéma

Aucun schéma n'est affiché par défaut si aucun schéma n'a été prédéfini.

- Vous pouvez nommer le schéma ou utiliser le nom par défaut (metadata) et ajouter un commentaire si vous le souhaitez.
- Puis dans la zone Schema, modifiez la structure du schéma selon vos besoins.
- La barre d'outils vous offre la possibilité d'ajouter, supprimer ou déplacer les colonnes de votre schéma. Vous avez aussi la possibilité d'importer et d'exporter des schémas au format XML.
- Cliquez sur **Finish** pour terminer la création du schéma générique. Tous les schémas créés sont sauvegardés sous le noeud **Generic Schemas** correspondant.

Configurer un schéma WSDL

Talend Open Studio vous permet de stocker vos connexions WSDL dans le repository. La procédure de création est composée de trois étapes.

Tout d'abord, cliquez droit sur **WSDL** dans le Repository puis sélectionnez **Create WSDL schema**.

Etape 1 : Propriétés générales

Un assistant de connexion s'ouvre. Renseignez les propriétés générales du schéma, notamment les champs **Name** et **Description**. Le champ **Status** est un champ personnalisé que vous pouvez définir dans les Préférences (*Window > Preferences*).

Cliquez sur **Next** pour continuer.

Etape 2 : URI et définition des méthodes

Dans l'étape 2, définissez l'URI et les paramètres nécessaires pour obtenir les valeurs.

The screenshot shows the 'WSDL Parameter' configuration window. It contains the following fields and options:

- WSDL:** A text field containing the URL `http://www.deeptraining.com/webservices/weather.asmx?WSDL`.
- Need authentication?** A checkbox that is currently unchecked.
- User:** A text field.
- Password:** A text field.
- Use http proxy:** A checkbox that is currently unchecked.
- host:** A text field.
- port:** A text field.
- user:** A text field.
- password:** A text field.
- Method:** A text field containing the value `GetWeather`.
- Parameters:** A table with one row containing the value `Melbourne`.
- Add** and **Remove** buttons are located below the parameters table.

- Renseignez l'URI qui transmettra les valeurs souhaitées dans le champ **WSDL**.
- Renseignez les informations d'authentification ou les informations de Proxy, si nécessaire.
- Définissez la méthode en la nommant dans le champ **Method**.
- Cliquez sur **Add** pour ajouter les paramètres exigés par la méthode.
- Puis cliquez sur **Refresh Preview** pour vérifier que les paramètres sont corrects.

The screenshot shows the 'Preview' window. It contains a 'Refresh Preview' button and a table displaying the result of the 'GetWeather' method.

GetWeather
Sunny

L'aperçu Preview affiche les valeurs qui doivent être transmises par la méthode Web Service basée sur les paramètres entrés.

Etape 3 : Schéma final

Vous pouvez modifier le nom (par défaut, metadata) et modifier le schéma à l'aide de la barre d'outils.

Create new WSDL schema

File - Step 3 of 3

Add a Metadata File on repository
Define the setting of the parse job

Name: metadata
Comment:

Schema

Click to update schema preview **Guess**

Description of the Schema

Column	Key	Type	Nullable	Date Pattern (Ctrl...)	Length	Precision
GetWeather	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		5	

+ - ↑ ↓ [Icon] [Icon]

? < Back Next > Finish Cancel

Vous pouvez aussi récupérer ou mettre à jour le schéma original en cliquant sur **Guess**. Notez, cependant, que lorsque vous cliquez sur le bouton **Guess**, l'ancienne version du schéma sera écrasé par la version à jour.

Cliquez sur **Finish**. Le nouveau schéma apparaît dans le Repository, sous le noeud **WSDL**.

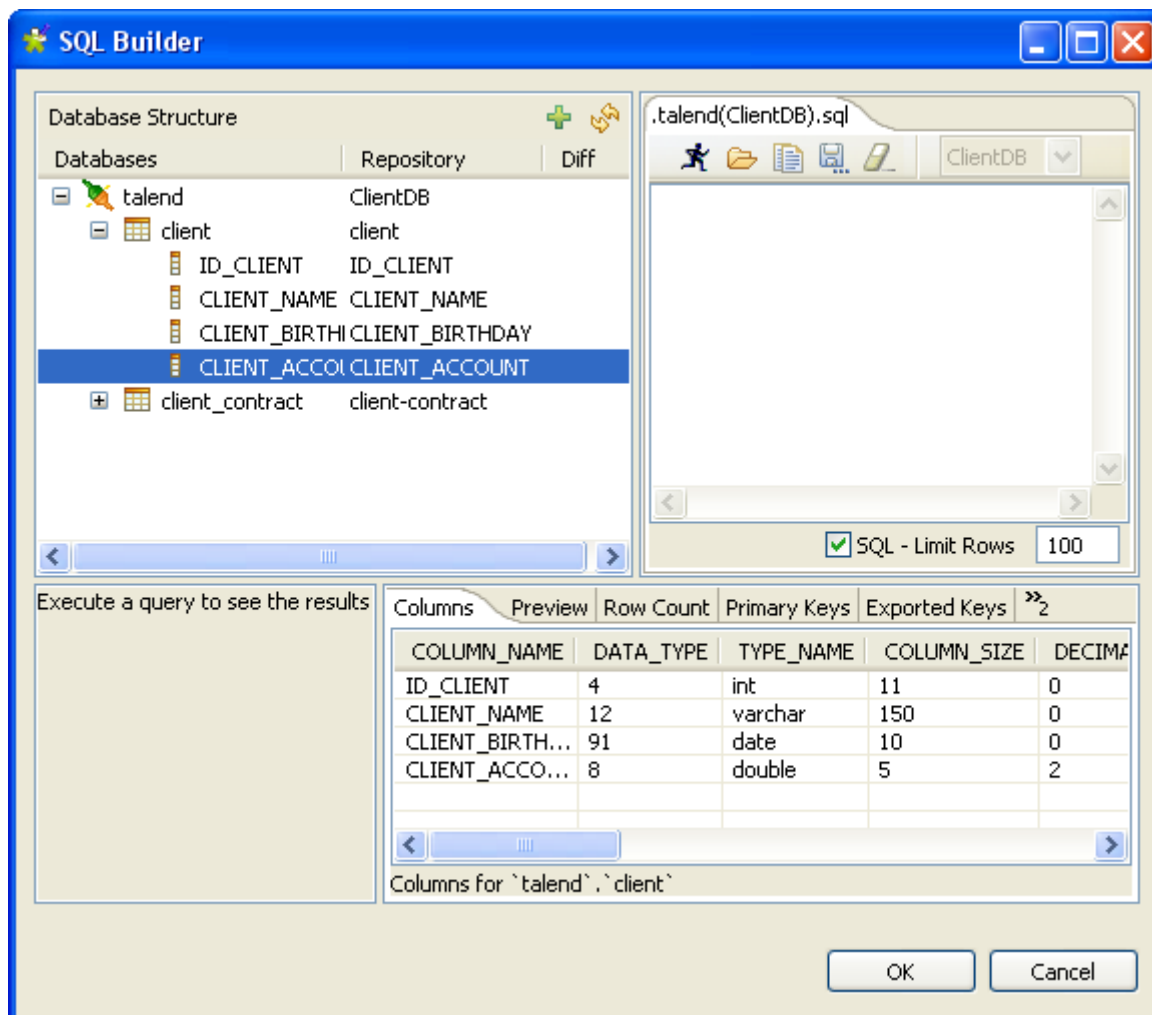
Créer une requête à l'aide de SQLBuilder

SQLBuilder vous permet de construire 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 construire un requête avec SQLbuilder que votre schéma de table soit stocké dans le référentiel 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** des propriétés du composant. Puis ouvrez l'éditeur de requête SQL Builder.



L'éditeur de requêtes SQL est composé de quatre vues :

- **Database structure** : Arborescence de la base de données
- **Query editor** : Editeur 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 des propriétés built-in de votre composant.

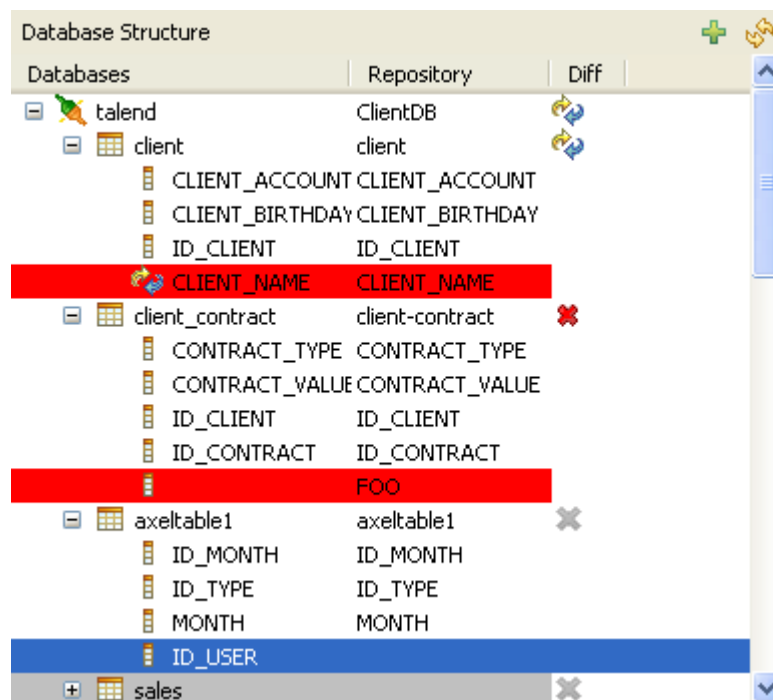
La vue Schema, en bas à droite de la fenêtre, fournit une description des colonnes.

Comparaison des 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.

Note: 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'icone 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.

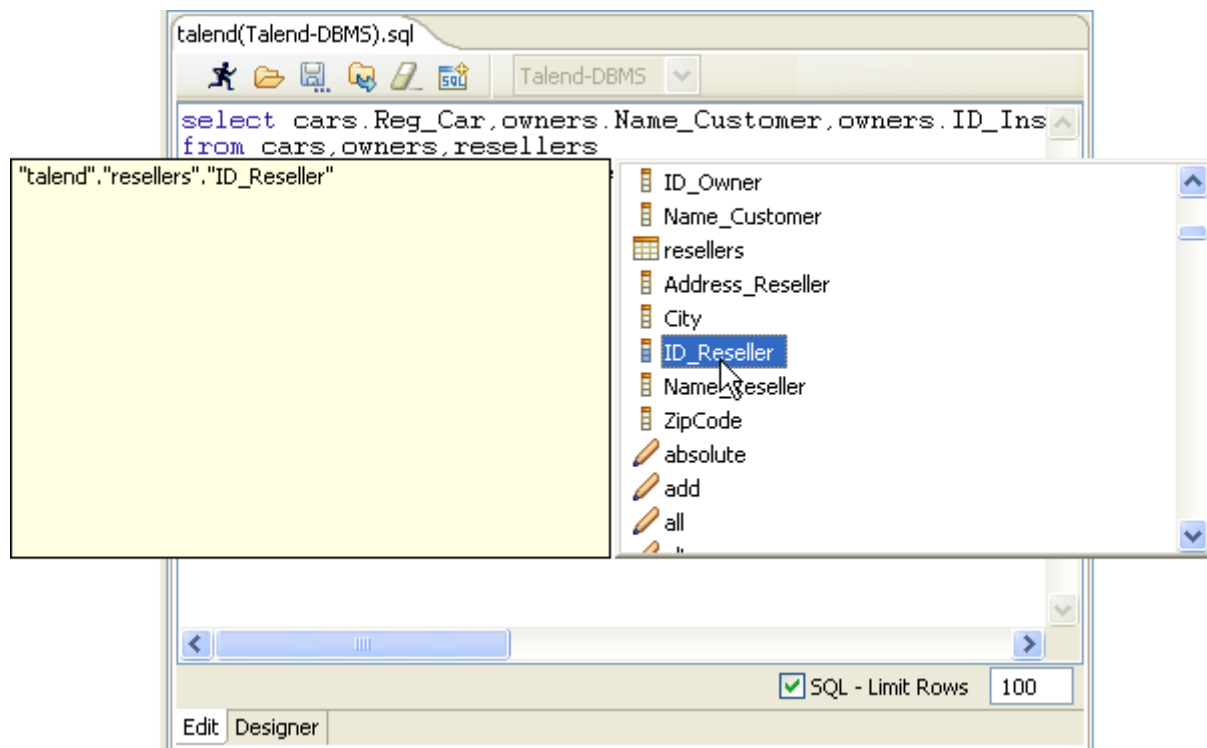
Le surlignage bleu indique que la colonne est présente dans la base de données elle-même mais qu'elle n'est pas présente dans la table de la métadonnée.

Construction d'une requête

L'éditeur de requête est un système multi-onglets vous permettant de construire autant de requêtes que vous le souhaitez.

Pour créer une nouvelle requête, cliquez droit sur la table ou sur une colonne de la table et sélectionnez **Generate Select Statement** dans le menu contextuel.

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**.

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

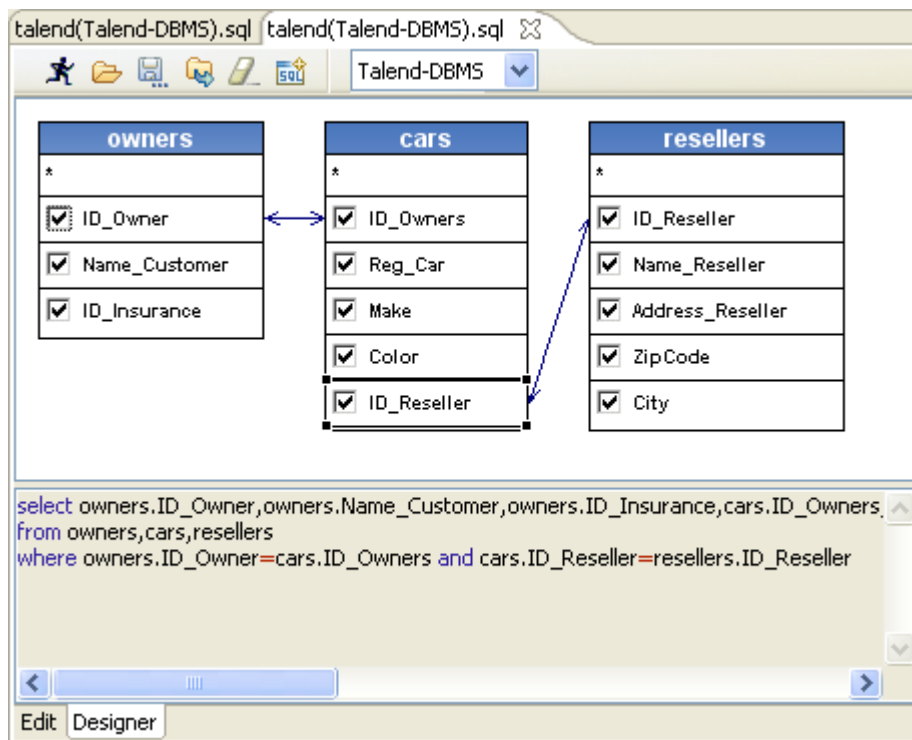
Note: Certaines déclarations SQL ne peuvent être interprétées par défaut. Décochez la case correspondant aux colonnes que vous souhaitez exclure de la sélection.

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

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. Le champ en sélection est automatiquement lié au champ de la seconde table.



La requête SQL correspondante s'affiche également dans la vue d'aperçu ou cliquez sur l'onglet **Edit** pour revenir au mode d'édition manuelle de requête.

Note: En mode **Designer**, vous ne pouvez pas ajouter graphiquement de critère de filtre. Pour ce faire, vous devez être en mode **Edit**.

Lorsque votre requête est terminée, exécutez-la en cliquant sur l'icone du coureur.

La barre d'outils au dessus de l'éditeur de requête vous permet d'accéder rapidement aux commandes principales, notamment : exécuter, ouvrir, enregistrer et effacer.

Les résultats de la requête active s'affichent dans la vue Query results.

La barre d'état en bas de la fenêtre fournit des informations relatives au temps d'exécution et au nombre de lignes traitées.

Stocker une requête dans le Repository

Pour pouvoir récupérer et réutiliser les requêtes que vous avez créées, nous vous recommandons de les conserver dans le **Repository**.

Dans l'éditeur SQL Builder, cliquez sur **Save** (représenté par une disquette 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.

Mapping de flux de données dans un Job

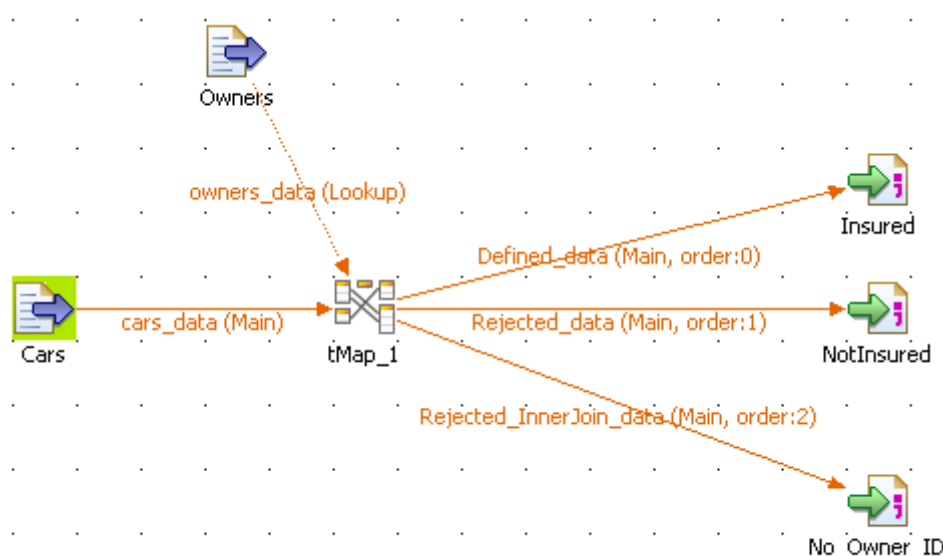
Pour gérer des flux multiples en entrée et/ou en sortie, ainsi que les transformations et redirection de données, il est recommandé d'utiliser le composant tMap. La section suivante présente les principes d'usage de ce composant, pour d'autres informations ou cas d'utilisation, consultez le composant du Guide de référence.

Présentation du fonctionnement tMap

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

Etant donnée que toutes ces opérations de transformation et/ou routage sont réalisées par le tMap, ce composant ne peut être ni un composant Start ni un composant End dans un Job design.



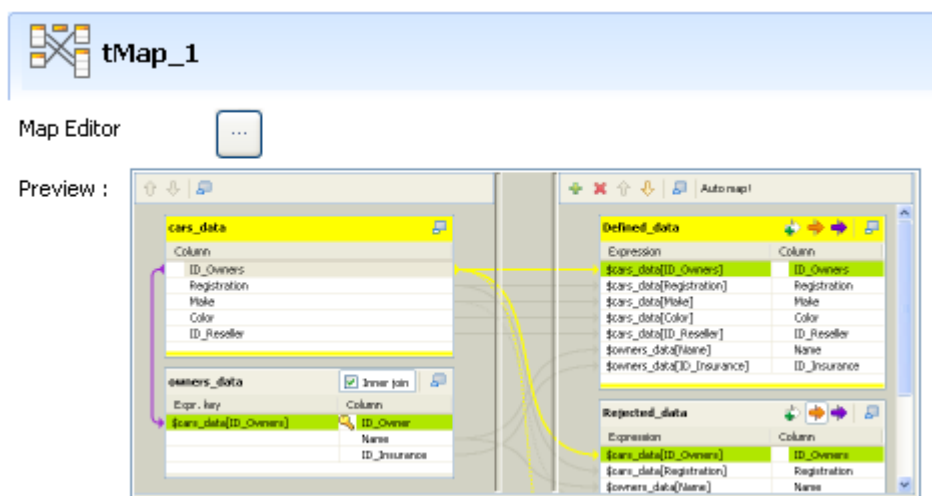
tMap utilise les connexions entrantes pour préremplir les schémas d'entrée dans le Mapper. Par conséquent, vous ne pouvez pas créer directement dans le Mapper, de nouveaux schémas d'entrée. Par contre, il vous faut mettre en place autant de connexion de type **Row** entrantes que nécessaire, pour que leur schéma s'affiche dans le Mapper.

De la même façon, créez autant de connexions de sortie que nécessaire dans l'éditeur graphique. Le Mapper vous permet de définir le type de données par sortie via un outil graphique de mapping.

Note: Il ne peut y avoir qu'une seule connexion de type **Main row**. Toute connexion entrante supplémentaire devient automatiquement une connexion **Lookup**. Voir : *Connexion de type Row*, page 46

Les connexions Lookup sont des connexions entrantes secondaires (ou de référence). Ces données de référence peuvent dépendre directement ou indirectement du flux principal. Cette relation de dépendance se traduit graphiquement par un Join et la création de clés d'expression.

Bien que le mapper 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 le panneau **Component** de l'espace de modélisation.

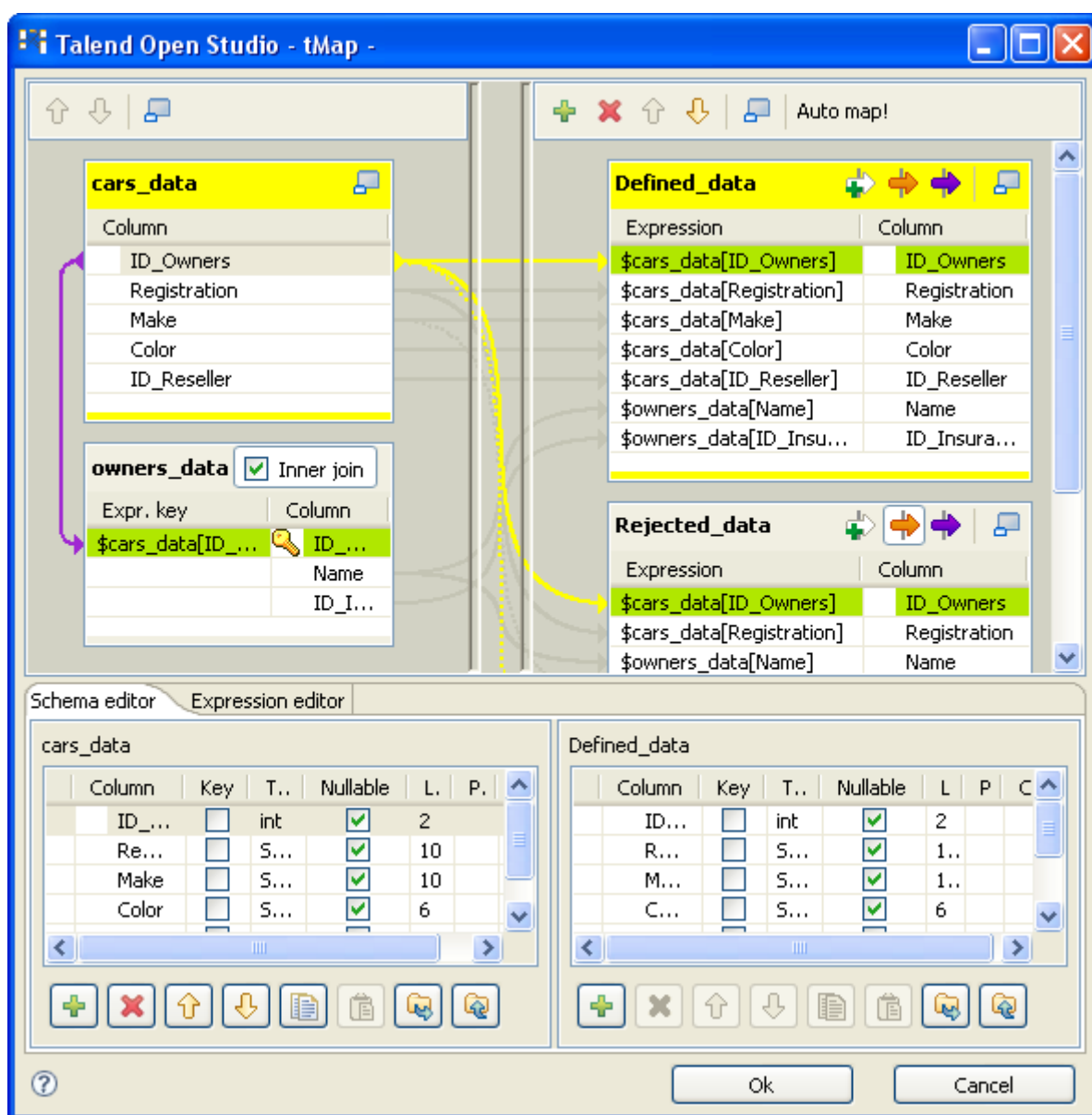


Double-cliquez sur l'icone tMap dans l'éditeur graphique ou cliquez sur le bouton [...] de l'onglet **Component**. L'éditeur de mapping s'ouvre dans une boîte de dialogue.

Interface tMap

tMap est un composant dit avancé qui requiert plus d'informations de propriétés que les autres composants. En effet, le **Mapper** 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 Mappers contenus dans l'éditeur de mapping, à l'aide des boutons dédiés situés dans le coin supérieur gauche de chacune des fenêtres.



Le Mapper est composé des panneaux suivants :

- La zone **Input** sur le coté gauche de l'éditeur. Les tableaux Input offrent une représentation structurée (schémas) de tous les flux entrants (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** est au centre de l'éditeur. Ce tableau de variable permet de centraliser des informations redondantes et de réaliser les transformations.
- La zone **Output** est située à droite de la fenêtre. Les tableaux Output permettent d'aiguiller les données et les champs provenant des schémas Input et des variables vers les schémas de sortie Output.

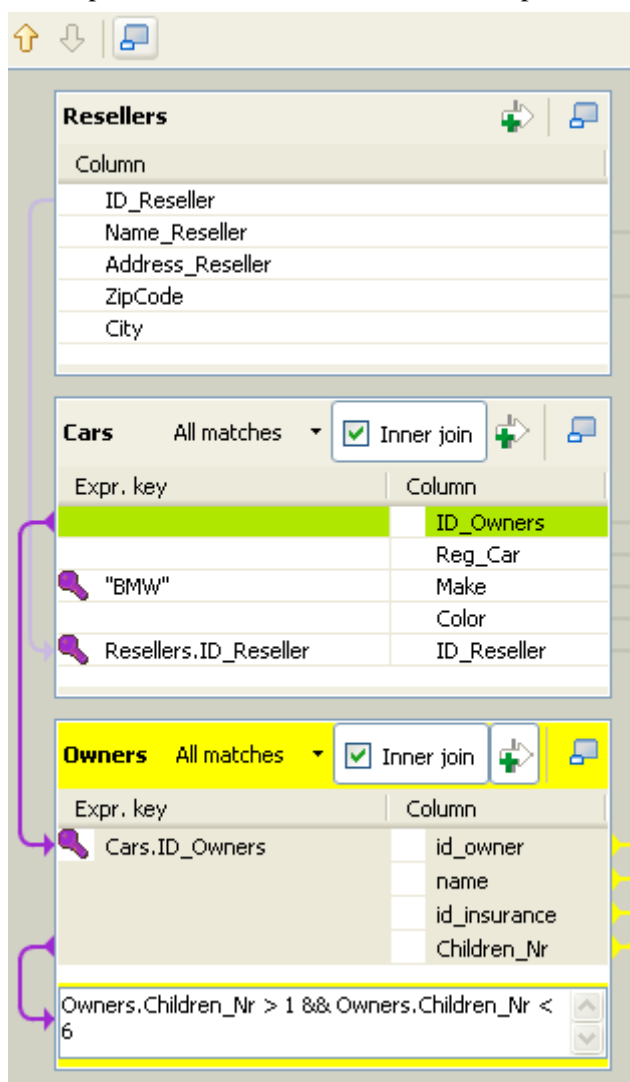
- Les deux panneaux situés en bas de la fenêtre sont les descriptions des schémas d'entrée et de sortie. L'onglet **Schema editor** détaille les schémas Input et Output sélectionnés.
- L'onglet **Expression editor** est l'outil de rédaction des clés d'expression des schémas Input/Output, des variables ou des conditions de filtre.

Le nom des schémas Input/Output dans l'éditeur de mapping reflète le nom des connexions entrantes et sortantes du composant tMap (connexion de type Row).

Configurer le flux d'entrée du tMap

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 ce schéma vers un niveau inférieur, sans risquer de perdre les éventuelles relations de dépendance (Join) que vous avez créées.



Vous pouvez utiliser les flèches haut et bas pour intervertir des tables secondaires (provenant d'une connexion **Lookup**), mais vérifiez qu'elles ne sont pas liées par un lien **Join**. Si elles le sont, veillez à ce que le lien Join ne soit pas perdu.

Voir également : *Jointure explicite*, page 96.

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.

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 reproduit dans la première table de la zone Input du Mapper.

Le contenu des connexions secondaires (**Lookup**) est reproduit dans les autres tables dépendant de la table **Main**. Si vous n'avez pas encore défini de schéma pour un composant d'entrée, la table correspondante du mapper apparaîtra comme vide.

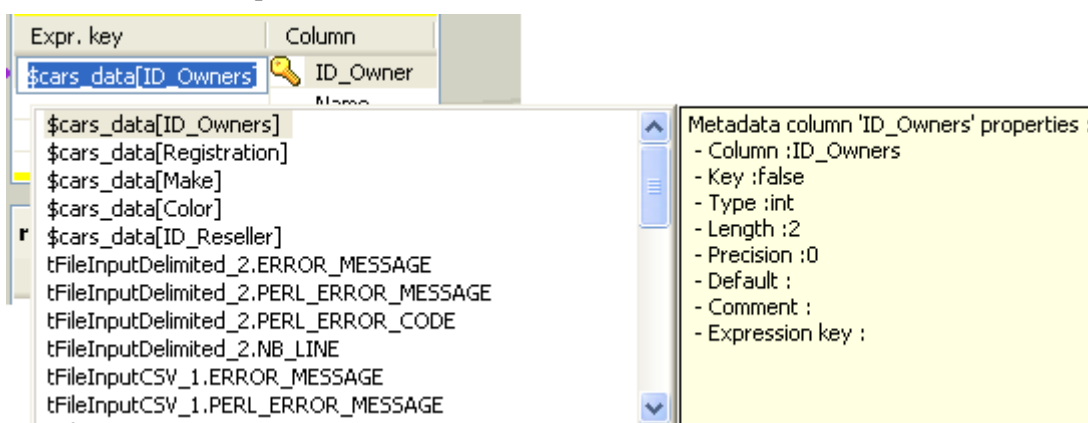
La clé (**Key**) est également repris du schéma défini dans les propriétés du composant Input. Cette **Clé** correspond à la clé définie dans le

schéma d'entrée, quand cela est possible. La clé de hachage (interne au Mapper) se différencie de la clé primaire en apparaissant 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. Elle fournit des informations concernant la colonne sélectionnée.

Voir également : *Mapping de variables, page 100*

Jointure explicite

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, les données **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.

Glissez et déposez les noms des colonnes d'une table vers sa table subordonnée, afin de créer une jointure **Join** entre les deux tables. Ainsi, vous pouvez appeler des données à partir d'entrées multiples.

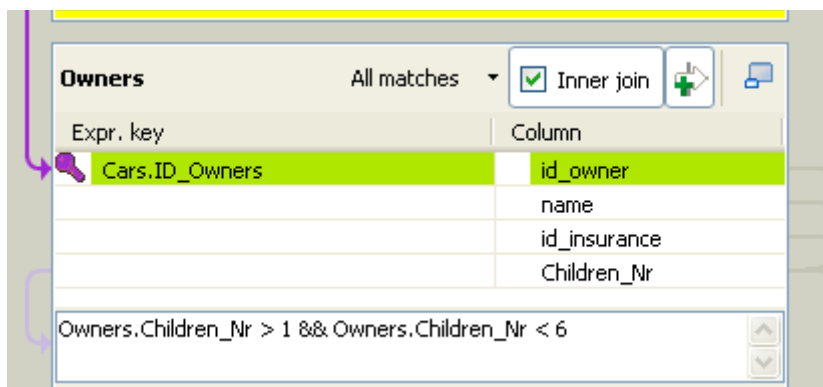
Les jointures apparaissent sous forme de liens violets et crée automatiquement une clé qui aura la même fonction qu'une clé de hachage, c'est-à-dire d'accélérer le processus de recherche.

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. Pour cela, il faut que vous ayez créé une jointure entre ses tables.

Note: 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**. 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.



Pour plus d'informations concernant les types de glisser-déposer possibles, consultez *Paramètres Output, page 102*.

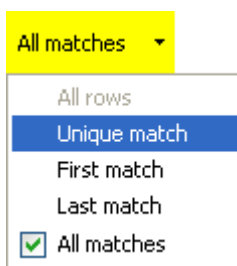
Note: Si vous avez beaucoup de tables Input, vous pouvez les réduire ou les restaurer à l'aide de l'icône correspondante dans la zone **Input**. La jointure liant les deux tables reste visible même lorsque la table est réduite.

La création d'un *Join* assigne automatiquement une clé de hachage au niveau du nom de champ de la table reliée. Le symbole d'une clé violette apparaît sur la table Input et disparaît quand la jointure est supprimée.

Voir également :

- *Schema editor, page 106*
- *Inner join, page 98*

Avec une jointure explicite, vous pouvez choisir de mettre un filtre pour limiter le nombre de matches. Vous pouvez soit prendre en compte un match, soit le premier ou le dernier, soit tous les matches.



Unique Match (java)

Elle correspond à l'option par défaut lorsque vous effectuez une jointure explicite. Cela signifie que zéro ou un match du flux secondaire sera pris en compte et passera dans la table de sortie.

Si plusieurs matches sont disponibles, un message d'avertissement apparaîtra.

First ou Last Match (java)

Cette option signifie que plusieurs matches peuvent être attendues dans le flux secondaire. L'option First ou Last Match signifie que seulement le premier ou le dernier match du flux secondaire sera pris en compte et passera dans le flux principal de sortie.

Les autres matches seront donc ignorés.

All Matches (java)

Cette option signifie que plusieurs matches sont attendus dans le flux secondaire. Ainsi, tous les matches sont pris en compte et passent dans le flux principal de sortie.

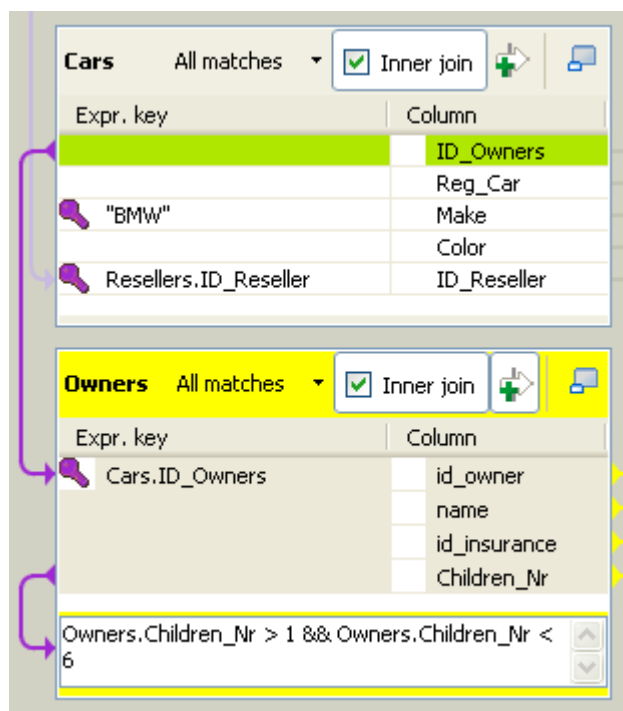
Inner join

La jointure **Inner join** est un type particulier de jointure qui se distingue par la façon dont le rejet est effectué.

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**.

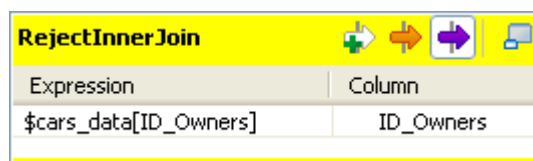
Si les données cherchées ne peuvent être récupérées avec la jointure explicite ou un filtre de jointure (Inner join), en d'autres termes la jointure Inner Join ne peut être établie quelle qu'en soit la raison, dans ce cas, les données demandées sont rejetées vers une table de sortie appelée **Inner Join Reject**.

Ainsi, vérifiez la case **Inner Join** située en haut de la table lookup et cochez-la.



Dans la zone **Output**, cliquez sur le bouton **Inner Join Reject** pour définir votre table comme table de sortie Inner Join Reject.

Note: Une table en Inner Join doit toujours être couplée à une table Inner Join Reject.



Vous pouvez utiliser le bouton de filtre pour diminuer le nombre de lignes à traiter et ainsi améliorer les performances (en java).

Voir également :

- *Rejets Inner Join, page 105*
- *Filtrer un flux d'entrée (java), page 99*

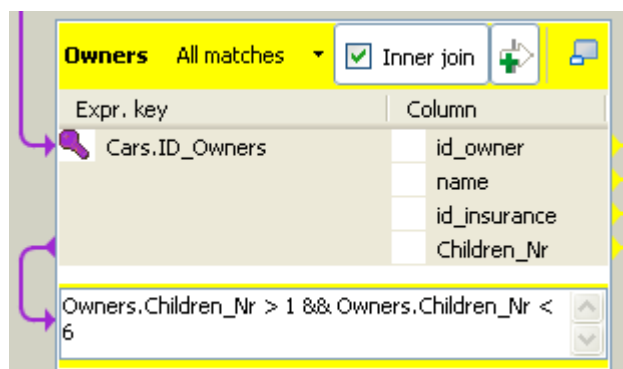
All rows (java)

Lorsque vous cochez la case **All rows**, la fonction **Inner Join** est automatiquement grisée. Cette option indique que toutes les lignes du flux secondaire sont traitées en fonction du flux principal.

La sortie correspond au produit cartésien des deux tables (ou plus, selon vos besoins).

Filtrer un flux d'entrée (java)

Cliquez sur le bouton **Filter** à côté du bouton Inner join pour ajouter une zone de **Filtre**.



Dans le zone de Filtre, saisissez les conditions que vous souhaitez appliquées, ce qui 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.



Cette fonctionnalité n'est disponible qu'en Java, c'est pourquoi les conditions doivent être rédigées en Java.

Retirer des entrées de la table Input

Pour enlever des entrées Input, cliquez sur la croix rouge, en bas, dans le Schema Editor de la table sélectionnée. Appuyez sur Ctrl ou Maj pour réaliser une sélection multiple de champs/colonnes à retirer de la table.

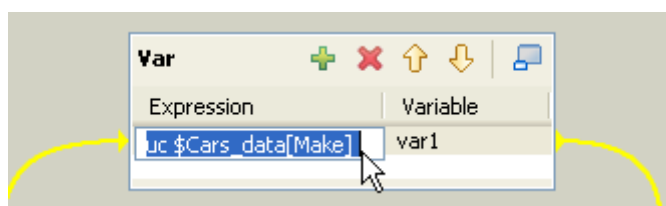
Note: Vous pouvez enlever des entrées Input d'un schéma dans le Mapper, cependant cette suppression se répercutera sur la définition du schéma au niveau des propriétés du composant en entrée.

Mapping de variables

La table **Variable** regroupe toutes les variables de mapping qui peuvent être utilisées à différents endroits du mapper.

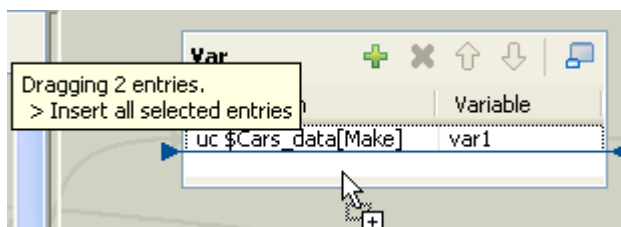
Vous pouvez également utiliser le champ Expression de la table Var pour réaliser des transformations, en langage Perl ou Java.

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 Perl. Saisissez les chaînes de caractères entre guillemets simples ou concaténez plusieurs fonctions à l'aide de l'opérateur Perl approprié.
- Ajoutez de nouvelles lignes à la table à l'aide du bouton (+) et retirez des lignes à l'aide du bouton (-). Puis appuyez sur Ctrl+Espace pour récupérer les variables globales et de contexte déjà existantes.
- Glissez et déposez une ou plusieurs entrées Input dans 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 grisée. 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.

Puis déposez la sélection vers la table Var. Plusieurs types de glisser-déposer sont possibles, notamment :

Comment faire pour...	Actions associées
Insérer toutes les entrées sélectionnées comme variables séparées.	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.
Concaténer toutes les entrées sélectionnées avec une entrée Var existante.	Glissez-déposez vers l'entrée Var à concaténer, elle se met alors en surbrillance. Toutes les entrées sont concaténées dans une seule cellule. Ajoutez les opérateurs requis à l'aide des signes opérateurs Perl/Java approprié. Le point permet la concaténation.
Ecraser une entrée Var par les entrées Input sélectionnées, et les concaténer.	Glissez-déposez vers l'entrée Var appropriée qui se met alors en surbrillance. Puis appuyez sur Ctrl et relâchez. Toutes les entrées sélectionnées sont concaténées et écrasent la variable sélectionnée.
Concaténer les entrées Input sélectionnées, avec plusieurs entrées Var mises en surbrillance. Créer de nouvelles lignes en cas de besoin.	Glissez-déposez vers des Var existantes puis appuyez sur Maj. Les premières entrées sont concaténées avec les entrées Var en surbrillance, et si nécessaire, de nouvelles lignes sont créées pour contenir les entrées restantes.

Accéder aux variables globales et de contexte

Appuyez sur **Ctrl+Esapce** 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. Elle fournit des informations concernant la colonne sélectionnée.

Supprimer des variables

Pour retirer une entrée Var sélectionnée, cliquez sur la croix rouge. La ligne entière est supprimée ainsi 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 rouge pour supprimer l'ensemble des lignes sélectionnées.

Paramètres Output

Dans l'espace de modélisation de **Talend Open Studio**, 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 Mapper.

Vous pouvez également ajouter un schéma Output dans votre Mapper, à l'aide du signe plus de la barre d'outils de la zone Output.

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.

Glissez-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.

Action	Résultat
Glisser-déposer vers des expressions existantes	Concatène l'expression sélectionnée, vers des expressions existantes.
Glisser-déposer vers une ligne d'insertion	Insère une ou plusieurs nouvelle entrée en début ou en fin de table ou entre deux lignes existantes.
Glisser-déposer + Ctrl	Remplace les expressions en surbrillance par les expressions sélectionnées.
Glisser-déposer + Maj	Ajoute les champs sélectionnés à toutes les expressions en surbrillance. Insère de nouvelles lignes si nécessaire.
Glisser-déposer + Ctrl + Maj	Remplace toutes les expressions en surbrillance par les champs sélectionnés. Insère de nouvelles lignes si nécessaire.

Vous pouvez ajouter des filtres et un rejet pour personnaliser vos sorties.

Construire des expressions complexes

Si vous devez construire des expressions complexes ou faire des changements majeurs sur le flux de sortie, utilisez la fenêtre **Expression Builder**.

Cliquez sur le champ **Expression** de votre table Input ou Output pour afficher le bouton [...]. Puis cliquez sur ce bouton pour ouvrir l'**Expression Builder**.

Pour plus d'informations concernant l'Expression Builder, consultez *Ecrire du code à l'aide du Expression Builder*, page 107.

Filtres

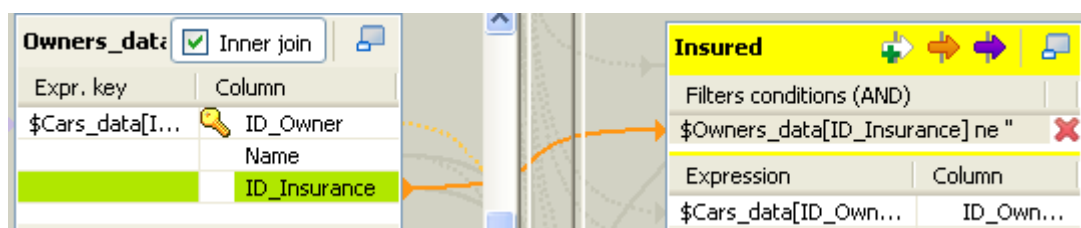
Les conditions de filtre vous permettent de sélectionner les champs et de les envoyer vers les sorties appropriées.

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 Perl/Java.

Glissez-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 Perl/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.

Rejets

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.

Créez une table dédiée et cliquez sur le bouton Output reject pour la définir comme contrepartie ELSE des tables régulières.



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 le 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 d'une sortie et qu'elle est donc routée vers elle, elle est également vérifiée contre les autres filtres et peut également être routée vers d'autres sorties.

Rejets Inner Join

L'Inner Join est un Join Lookup (de table secondaire). 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, cliquez et déposez un nouveau composant Output dans votre job design et connectez-le au tMap. Dans le Mapper, ce nouveau flux de sortie apparaît dans la zone Output. Cliquez alors sur le bouton Inner Join Reject pour définir cette table comme table de rejet Inner Join Reject.



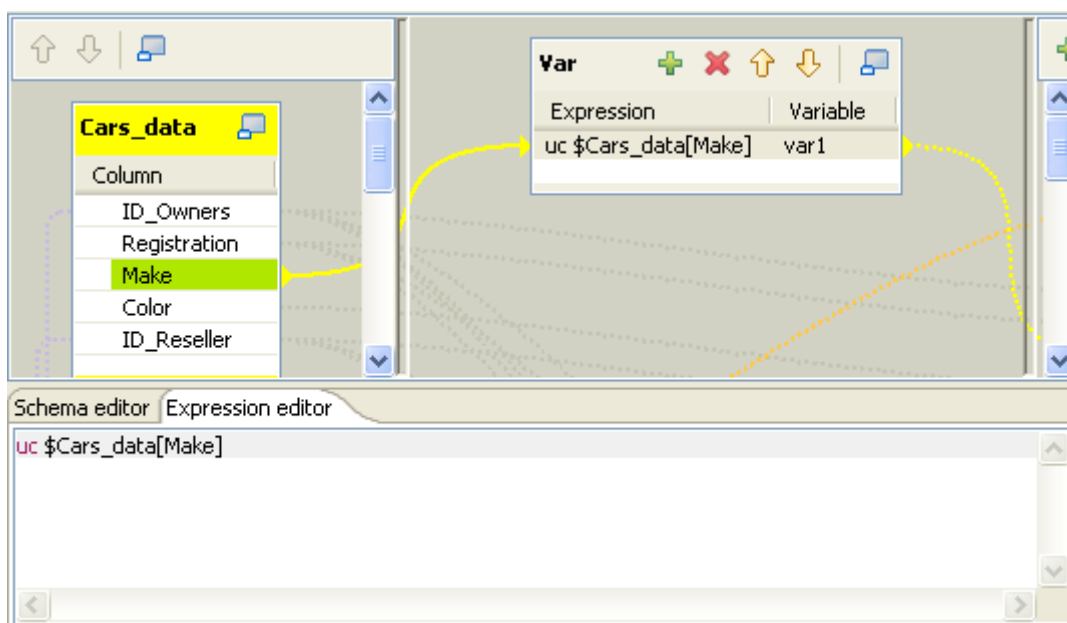
Retirer des entrées de la table Output

Pour retirer des entrées d'une table Output, cliquez sur la croix rouge, en bas, dans le Schema Editor de la table sélectionnée.

Expression editor

Toutes les déclarations d'expressions (Input, Var or Output) et de filtres peuvent être visualisées et modifiées à partir de l'éditeur d'expression. Cet éditeur fournit le confort d'une fenêtre dédiée pour écrire des fonctions ou des commandes de transformation.

Sélectionnez l'expression à modifier. Cliquez sur l'onglet **Expression editor**.



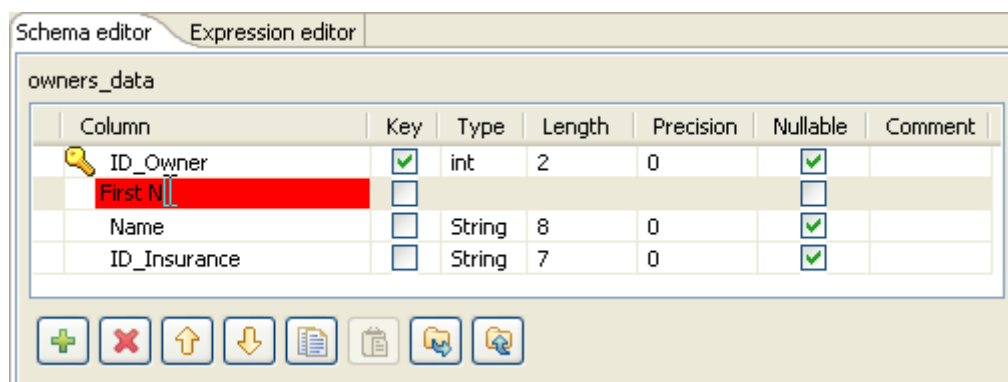
Saisissez le code Perl ou Java approprié pour l'opération à effectuer. La colonne Expression de la table correspondante est automatiquement mise à jour.

Note: Reportez-vous à la documentation Perl ou Java appropriée pour plus d'informations concernant les fonctions et les opérations.

L'Expression Builder peut vous aider à construire les expressions complexes dont vous avez besoin. Pour plus d'informations, consultez *Ecrire du code à l'aide du Expression Builder*, page 107.

Schema editor

Le Schema Editor fournit les informations concernant les champs d'une table sélectionnée.



Utilisez la barre d'outils placée sous la table de schéma pour ajouter, déplacez et supprimer des colonnes du schéma.

Vous pouvez également charger un schéma à partir du référentiel ou exporter le schéma courant vers un fichier.

Métadonnées	Description
Column	Nom de colonne tel que défini dans le schéma du Mapper et dans le schéma du composant Input ou Output correspondant.
Key	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.
Type	Type de données : Chaîne de caractères (String) ou entier (Integer). Note: Cette colonne est obligatoire dans la version Java.
Length	-1 indique qu'aucune valeur de longueur a été définie dans le schéma.
Precision	Précise la valeur de longueur de chaîne, si spécifiée.
Nullable	Décochez cette case si vous ne souhaitez pas autoriser les valeurs de champs nulles.
Default	Indique les valeurs par défaut, si elles sont définies.
Comment	Champ texte libre. Saisissez tout commentaire que vous jugerez utile.

Note: Les schémas Input et les schémas 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 toutes modifications apportées au schéma sont immédiatement reportées 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).

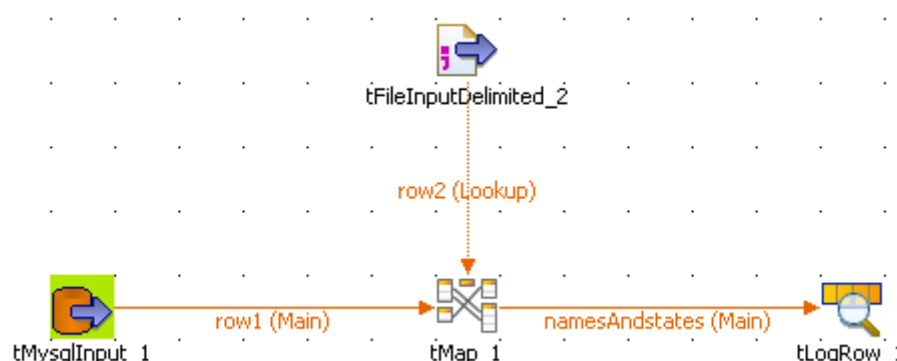
Un arrière-plan rouge s'affiche lorsqu'un caractère non valide a été saisi. La plupart des caractères spéciaux sont interdits pour réduire les risques de mauvaise interprétation dans le code. Les caractères autorisés sont : les minuscules et majuscules, les chiffres (à l'exception du caractère de début de chaîne).

Passez la souris sur le champ invalide pour faire apparaître le message d'erreur.

Ecrire du code à l'aide du 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 la construction de ce code (en Java ou Perl).

L'exemple suivant vous montre un exemple d'utilisation du **Expression Builder** pour le composant **tMap**.

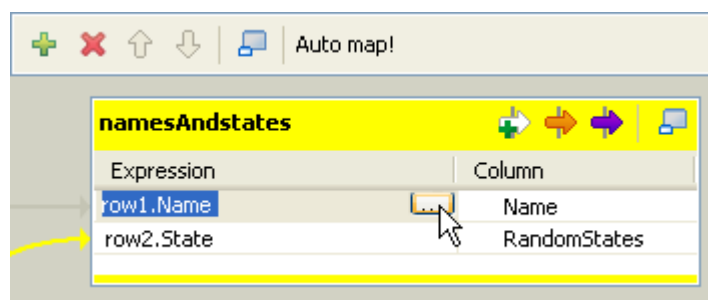


Deux flux d'entrée sont connectés au **tMap**.

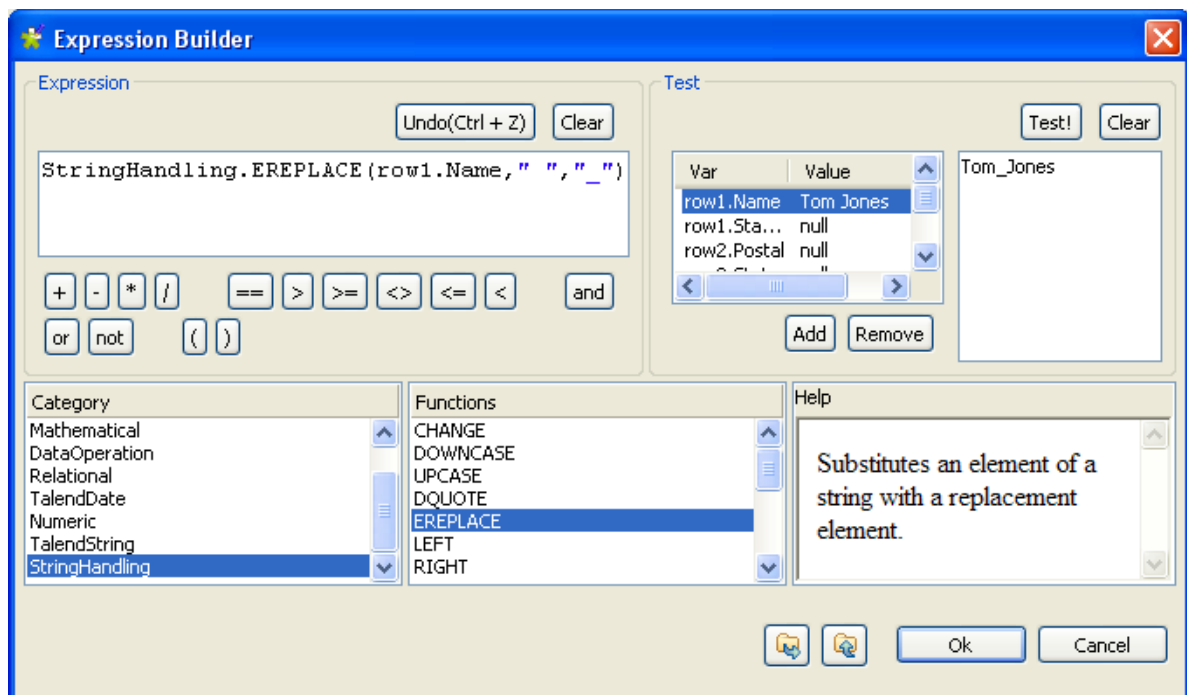
- Un premier flux DB input, contenant une liste de personnes dont le nom et le prénom sont séparés par un espace.
- Un autre flux File input, contenant la liste des états des Etats-Unis, en minuscule.

Dans le **tMap**, utilisez l'Expression builder pour : 1) Remplacer l'espace entre le nom et le prénom par un tiret souligné ; 2) Mettre en majuscule le nom des états.

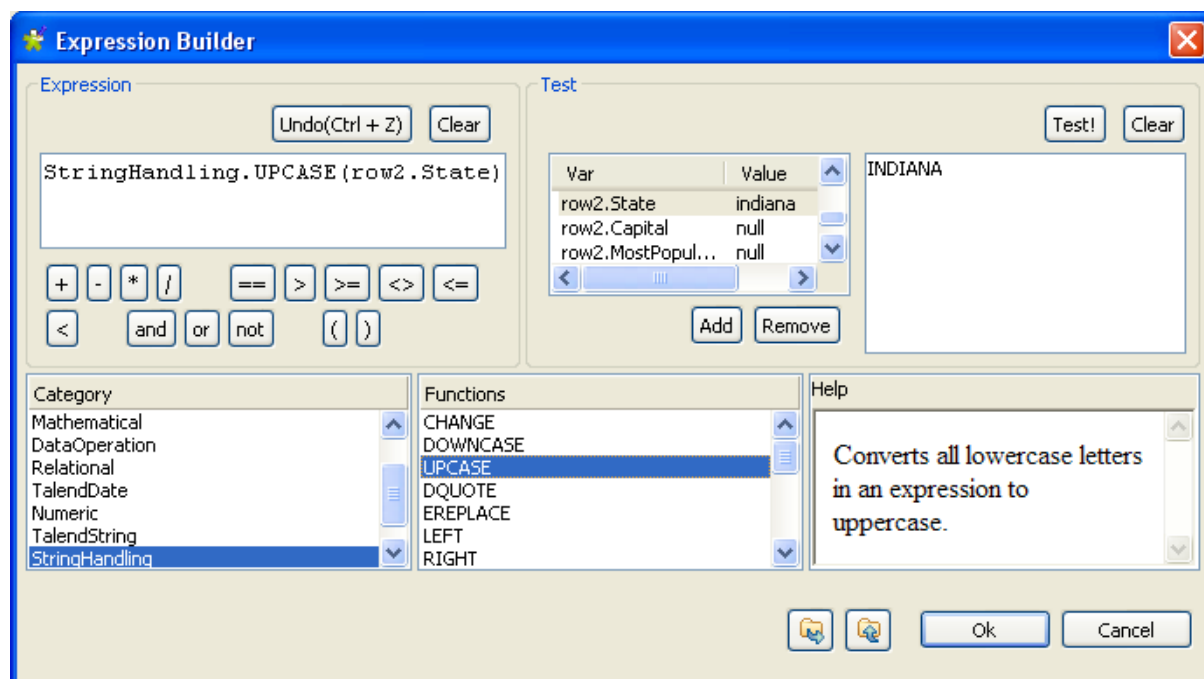
- Dans le **tMap**, créez la jointure Inner Join correspondante pour créer le mapping de référence. Pour plus d'informations concernant l'utilisation du tMap, consultez *Mapping de flux de données dans un Job*, page 92.
- Cliquez et déposez la colonne *Name* du flux d'entrée principal Main (*row1*) de la zone Input dans la zone Output, et la colonne *State* du flux d'entrée secondaire Lookup (*row2*) vers la même zone Output.
- Puis cliquez sur le premier champ **Expression** (*row1.Name*) pour afficher le bouton [...].



La fenêtre **Expression Builder** s'ouvre.

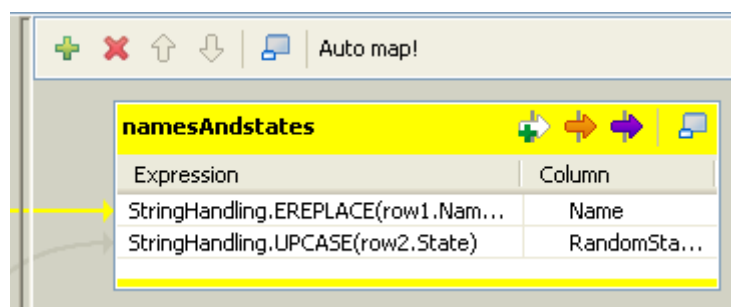


- Dans la liste **Category**, sélectionnez l'action que vous souhaitez effectuer. Pour cet exemple, choisissez **StringHandling**, puis sélectionnez la fonction **EREPLACE** dans la liste **Functions**.
- Dans la zone **Expression**, collez **row1.Name** à la place du texte, pour obtenir : `StringHandling.EREPLACE(row1.Name, ' ', '_')`. Cette expression remplacera l'espace par un tiret souligné dans la chaîne de caractères fournie.
- Vérifiez votre code en renseignant le champ **Value** correspondant dans la zone **Test**, par exemple saisissez *Tom Jones* en face de la variable *row1.Name*.
- Puis cliquez sur **Test!** et vérifiez que les changements ont bien été effectués, ici : *Tom_Jones*
- Cliquez sur **OK** pour valider.
- Répétez la même opération pour la deuxième colonne (*State*).
- 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 : `StringHandling.UPCASE(row2.State)`.
- Vérifiez que la syntaxe de votre code est correct, 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.

Les deux expressions sont maintenant affichées dans le champ **tMap Expression**.



Ces changements seront reportés tout au long du processus et le flux de sortie est affiché ci-dessous.

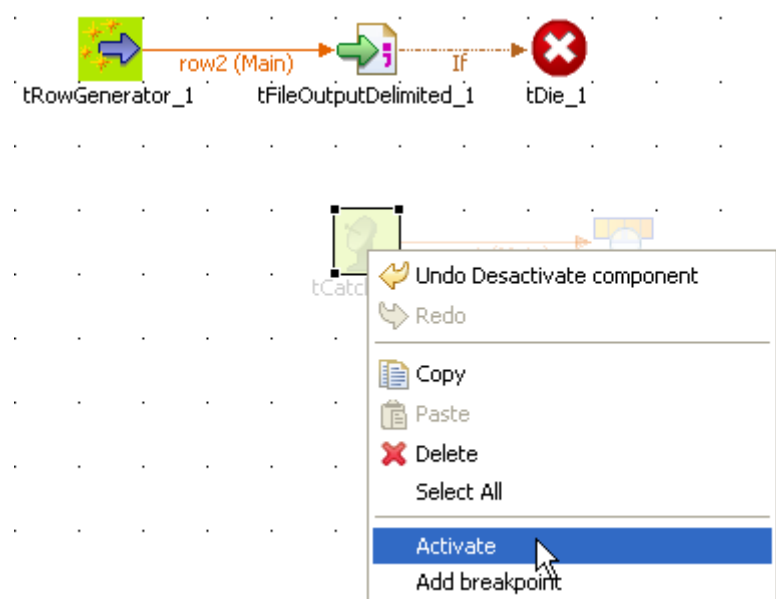
Starting job NamesAndStates at 10:02 10/10/2007.

tLogRow_1	
Name	RandomStates
William_Grant	IOWA
William_Hoover	NEW YORK
Grover_Lincoln	NORTH DAKOTA
Lyndon_Jefferson	OHIO
Gerald_Hayes	WASHINGTON
Benjamin_Grant	MAINE
George_Pierce	CONNECTICUT
Jimmy_Reagan	ALASKA
Martin_Hayes	WASHINGTON
Franklin_Jefferson	IOWA
Andrew_Nixon	NEW HAMPSHIRE

Activer/Désactiver un job ou un sous-job

Vous pouvez activer ou désactiver un composant, et par ce biais, un job entier ou un job secondaire (subjob) directement connecté au composant sélectionné. Par défaut, le composant est activé.

Dans la fenêtre de Propriétés **Main** du composant sélectionné, cochez ou décochez la case **Activer**.



Alternativement, cliquez-droit sur le composant et sélectionnez la commande **Activer/Désactiver** qui s'affiche en fonction du statut courant du composant.

Si vous désactivez un composant, aucun code ne sera généré et vous ne pourrez ni ajouter ni modifier de liens à partir de ce composant vers des composants actifs ou nouveaux.

Voir également : *Définition du composant Start*, page 57.

Désactiver un composant Start

Dans le cas où un composant **Start** est désactivé, les composants de tout type, ainsi que les connexions de toute nature, reliés directement et indirectement à lui, seront désactivés également.

Désactiver un composant non-Start

Lorsque vous décochez la case **Activer** d'un composant standard (non Start), seul le composant en question et ses connexions directes sont désactivés.

Si le lien reliant un composant désactivé à un sous-job est un lien direct de type Row main (principal), alors tous les composants de ce sous-job seront également désactivés.

Définir les Contextes et les Variables

Les contextes vous donnent la possibilité de gérer différemment vos jobs selon les conditions d'utilisation. Par exemple, vous pouvez intégrer dans votre cycle de validation plusieurs phases de test avant que votre job design soit mis en production.

Talend Open Studio vous offre la possibilité de créer des contextes multiples. De plus, vous pouvez créer des contextes ponctuels à partir de l'onglet Context ou vous avez la possibilité de regrouper tous vos contextes sous le noeud **Contexts** du Repository afin de les réutiliser dans différents jobs.

Un contexte est défini 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+Barre d'espace** dans les propriétés du composant dans la vue **Component**.

Utiliser les variables dans un Job

Dans tous les champs de l'onglet **Component** définissant les propriétés d'un composant, vous pouvez utiliser une variable globale définie ou des variables de contexte.

Appuyez sur les touches **Ctrl+Espace** pour afficher la liste complète des variables globales et de contexte utilisées dans les fonctions Perl prédéfinies. Les variables de contexte sont créées par l'utilisateur pour un contexte d'utilisation particulier, tandis que les variables globales sont des variables système.

Description: Error Message	tFileInputDelimited_2.ERROR_MESSAGE
Global variable, property of component tMap [tMap_1].	tFileInputDelimited_2.NB_LINE
Type: String	tMap_1.ERROR_MESSAGE
Availability: After	tFileOutputDelimited_1.ERROR_MESSAGE
	tFileOutputDelimited_1.NB_LINE
	tFileOutputDelimited_2.ERROR_MESSAGE
	tFileOutputDelimited_2.NB_LINE
	tFileInputDelimited_3.ERROR_MESSAGE
	tFileInputDelimited_3.NB_LINE
	tFlowMeter_1.ERROR_MESSAGE
	tFlowMeter_2.ERROR_MESSAGE

La liste s'allonge au fur et à mesure que vous définissez des variables (variables de contexte).

Voir également :

- Définir les variables de contexte à partir de la vue **Component**, page 118
- Définir des variables à partir de la vue **Contexts**, page 113

Définir des variables à partir de la vue Contexts

Les variables peuvent être créées et définies de différentes manières : soit à partir de la vue **Component**, soit directement à partir de la vue **Context**.

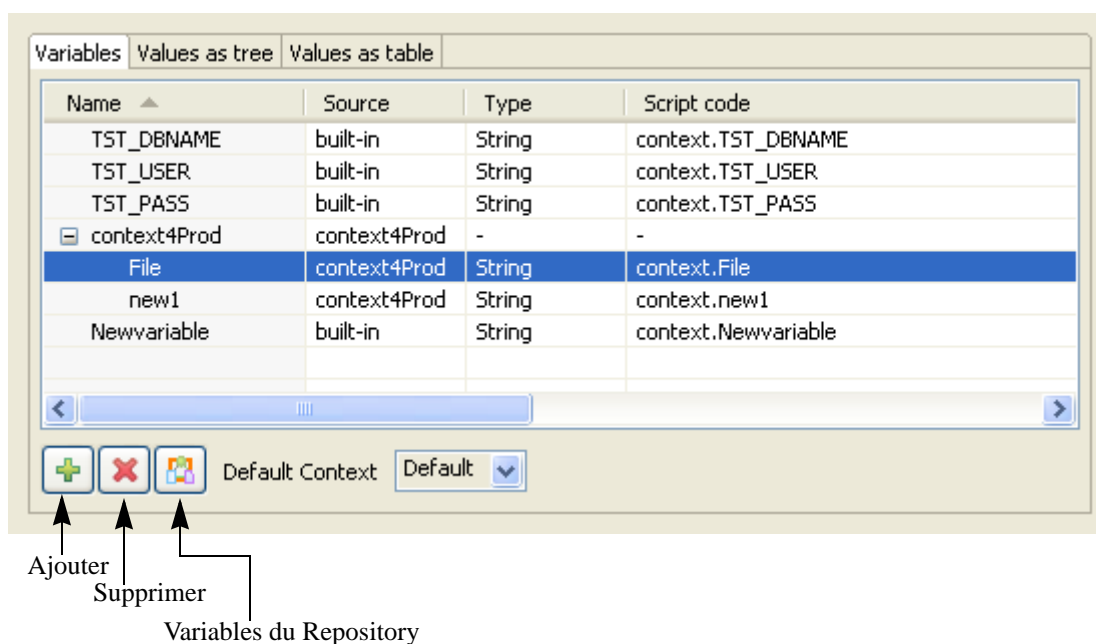
Pour plus d'informations concernant la définition des variables directement à partir de la vue **Component**, consultez *Définir les variables de contexte à partir de la vue Component*, page 118.

La vue **Contexts** se situe dans la partie inférieure du **Job Designer** et se compose de trois onglets : **Variables**, **Values as tree** et **Values as table**.

Note: Si la vue **Contexts** n'apparaît pas dans le fenêtre de **Talend Open Studio**, cliquez sur le menu **Window > Show view > Talend**, puis sélectionnez **Contexts**.

Variables

L'onglet **Variables** fait partie de l'onglet **Contexts** et affiche toutes les variables définies pour chaque composant du job en cours.



A partir de cette vue, vous pouvez gérer vos variables built-in (local) :

- Cliquez sur le bouton (+) pour ajouter une ligne de paramètre au tableau.
- Editez le nom **Name** de la nouvelle variable et saisissez le nouveau nom *<Newvariable>*.

- Supprimez les variables built-in. (Rappel : les variables stockées sur le Repository sont en lecture seule.)
- Importez les variables à partir d'une source contexte du Repository à l'aide du bouton **Repository variables**.

Pour définir la valeur actuelle de la variable nouvellement créée, cliquez sur l'onglet **Value as tree**.

Vous pouvez ajouter autant d'entrée que nécessaire dans l'onglet **Variables**. Par défaut, la variable créée est de type built-in.

Champs	Description
Name	Nom de la variable. Vous pouvez éditer ce champ, à condition que la variable soit de type Built-in. En effet, les variables de type Repository sont en lecture seule.
Source	Built-in : La variable est créée pour ce job et sera utilisée uniquement dans ce job. <Repository entry name> : La variable a été définie dans un contexte stockés sur le Repository. Ainsi, la source est le groupe de contexte actuel que vous avez créé dans le Repository.
Type	Sélectionnez le type de données traitées. Ce champ est obligatoire en Java.
Script code	Le code correspondant à la valeur de la variable. Le langage de génération de code utilisé pour ce champ dépend du langage sélectionné lors de la création de votre projet (Java ou Perl). En Perl, le script affiché sera : <code>\$_context {VotreNomDeParametre . Ce Script de code se génère automatiquement lorsque vous définissez la variable dans la vue Component du composant.</code>
Comment	Ajoutez un commentaire utile.

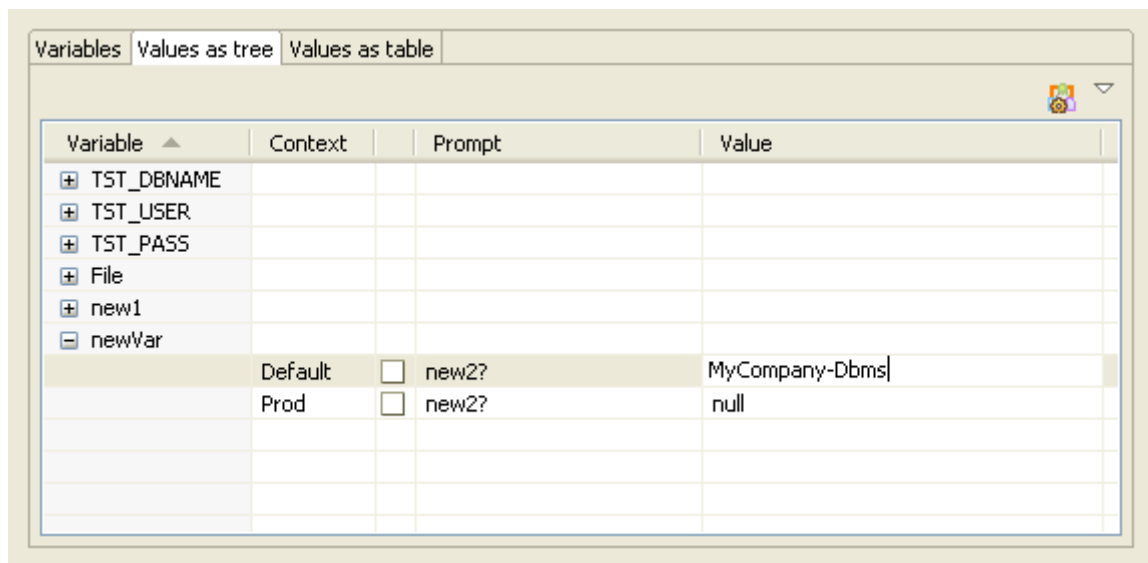
Note: Vous ne pouvez pas créer de contexte à partir de l'onglet **Variables**, mais seulement à partir des onglets **Values as table** et **as tree**.

Pour plus d'informations concernant la définition des variables à partir de la vue **Component**, consultez *Définir les variables de contexte à partir de la vue Component*, page 118.

Pour plus d'informations concernant les variables du Repository, consultez *Centraliser les contextes dans le Repository*, page 120.

Values as tree

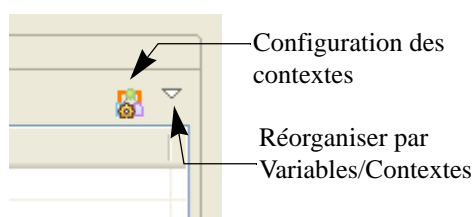
Cet onglet affiche les variables, ainsi que leurs valeurs, sous forme d'arbre.



A partir de cet onglet, vous pouvez :

- Définir la valeur d'une variable built-in directement dans le champ **Value**. Notez que les valeurs des variables de type Repository sont en lecture seule et ne peuvent être éditées uniquement dans le contexte Repository adéquat.
- Définir un message qui apparaîtra à l'utilisateur afin qu'il confirme la valeur de la variable lors de l'exécution.
- Créer ou éditer un contexte à l'aide du bouton dédié en haut à droite.
- Réorganiser les variables/contextes.

Champs	Description
Variable	Nom des variables.
Context	Nom des contextes.
Prompt	Cochez cette case, si vous souhaitez que la variable s'affiche dans la boîte de dialogue de confirmation lors de l'exécution.
	Si vous avez coché l'option Prompt, renseignez le champ associé qui correspond au message qui s'affichera dans la boîte de dialogue.
Value	La valeur correspondant à la variable. Définissez la valeur des variables built-in. Notez que les variables de type Repository sont en lecture seule.



Gérez vos contextes à partir de cet onglet, à l'aide du bouton correspondant situé en haut à droite de l'onglet **Contexts**. Pour plus d'informations concernant la gestion des contextes, consultez *Configurer les contextes, page 117*.

Dans l'onglet **Values as tree**, vous pouvez afficher les valeurs en fonction des *contextes* ou des *variables* pour plus de clarté.

Pour modifier l'affichage des valeurs dans l'arbre, cliquez sur la petite flèche vers le bas, puis sélectionnez l'option de réorganisation **group by** qui vous souhaitez.

Pour plus d'information concernant la définition des variables, consultez *Définir les variables de contexte à partir de la vue Component, page 118* et *Centraliser les contextes dans le Repository, page 120*.

Values as table

L'onglet **Values as table** affiche les paramètres des contextes et des variables sous forme de tableau.

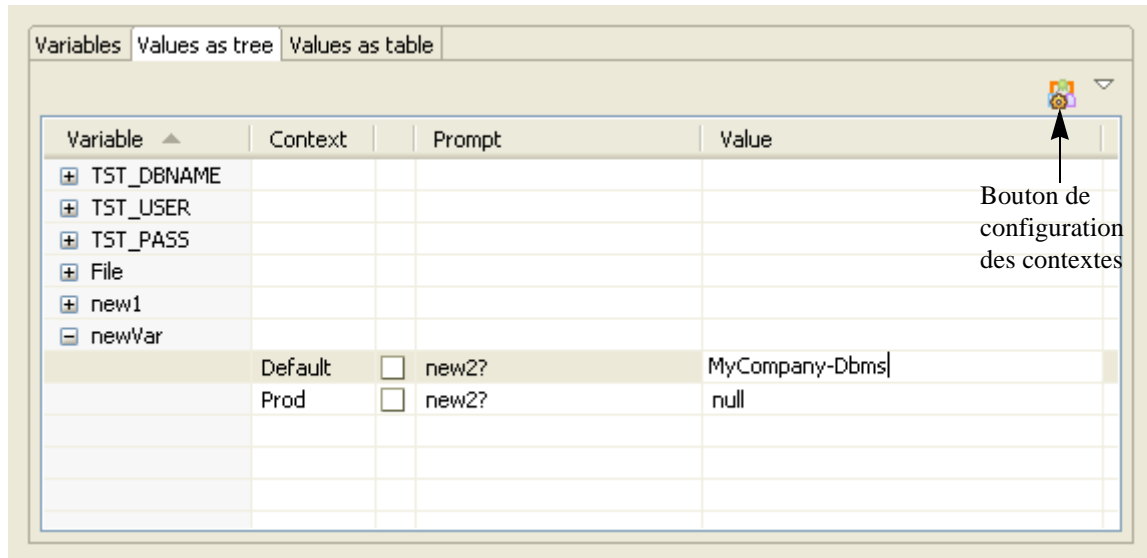
Champs	Description
Name	Le nom de la variable.
<YourContextName>	La valeur correspondant à la variable.

Gérez vos contextes à partir de cet onglet, à l'aide du bouton **Configure contexts** situé en haut à droite de l'onglet **Contexts**. Pour plus d'informations concernant la gestion des contextes, consultez *Configurer les contextes, page 117*.

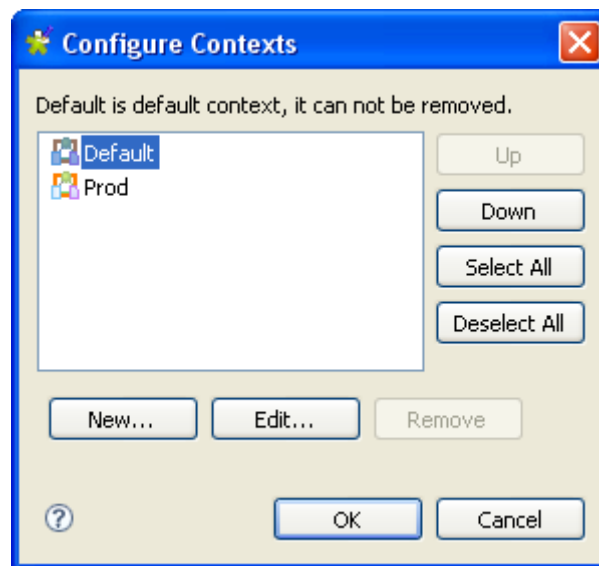
Pour plus d'informations concernant la définition des variables, consultez *Définir les variables de contexte à partir de la vue Component, page 118* et *Centraliser les contextes dans le Repository, page 120*.

Configurer les contextes

Vos contextes peuvent être gérés uniquement à partir des onglets **Values as table** ou **Values as tree** à l'aide du bouton correspondant en haut à droite de l'onglet **Contexts**.



Sélectionnez **Configure Contexts...** pour ouvrir la fenêtre de gestion.

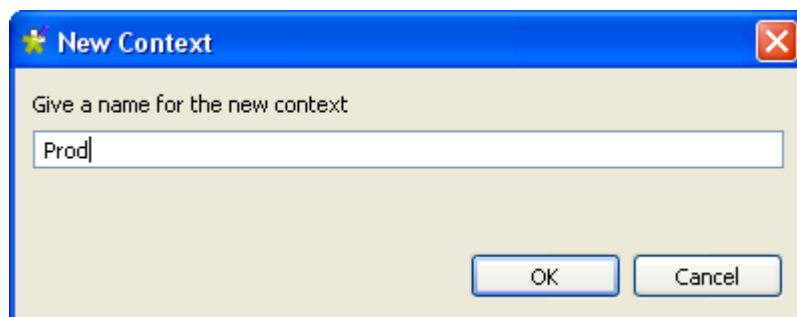


Note: Le contexte par défaut ne peut être supprimé, c'est pourquoi le bouton **Remove** est grisé. Pour le rendre accessible, sélectionnez un autre contexte dans la liste **Default Context** dans l'onglet **Contexts**.

Créer un contexte

Vous pouvez créer autant de contextes que vous le souhaitez. Ces contextes seront basés sur les paramètres des contextes par défaut.

- Pour créer un nouveau contexte, cliquez sur **Configure Contexts > New...**
- Saisissez le nom de votre nouveau contexte.



Cliquez sur **OK** pour valider votre création.

Lorsque vous créez un nouveau contexte, la totalité des informations par défaut est copiée vers le nouveau contexte. Il vous suffit donc d'éditer les champs nécessaires dans l'onglet **Value as tree** pour modifier votre contexte comme vous le souhaitez.

La liste déroulante **Default Context** propose tous les contextes créés pour ce job design.

Vous avez la possibilité de changer de contexte par défaut en sélectionnant un nouveau contexte dans la liste **Default Context** dans l'onglet **Variable** de la vue **Contexts**.

Notez qu'il est impossible de supprimer le contexte par défaut (ou le dernier) car l'exécution d'un job requiert toujours un contexte. Laissez-lui le nom de Default.

Renommer et éditer un contexte

Pour renommer un contexte existant, cliquez sur **Edit** dans la boîte de dialogue **Configure contexts** et saisissez le nouveau nom.

Cliquez sur **OK** pour valider les changements.

Les changements des valeurs des variables de contexte se font à partir des onglets **Values as tree** ou **Values as table**. Pour plus d'informations sur ces onglets, consultez *Définir des variables à partir de la vue Contexts*, page 113.

Définir les variables de contexte à partir de la vue Component

Les variables peuvent être créées et définies de différentes manières : soit à partir de la vue **Contexts**, soit directement à partir de la vue **Component**.

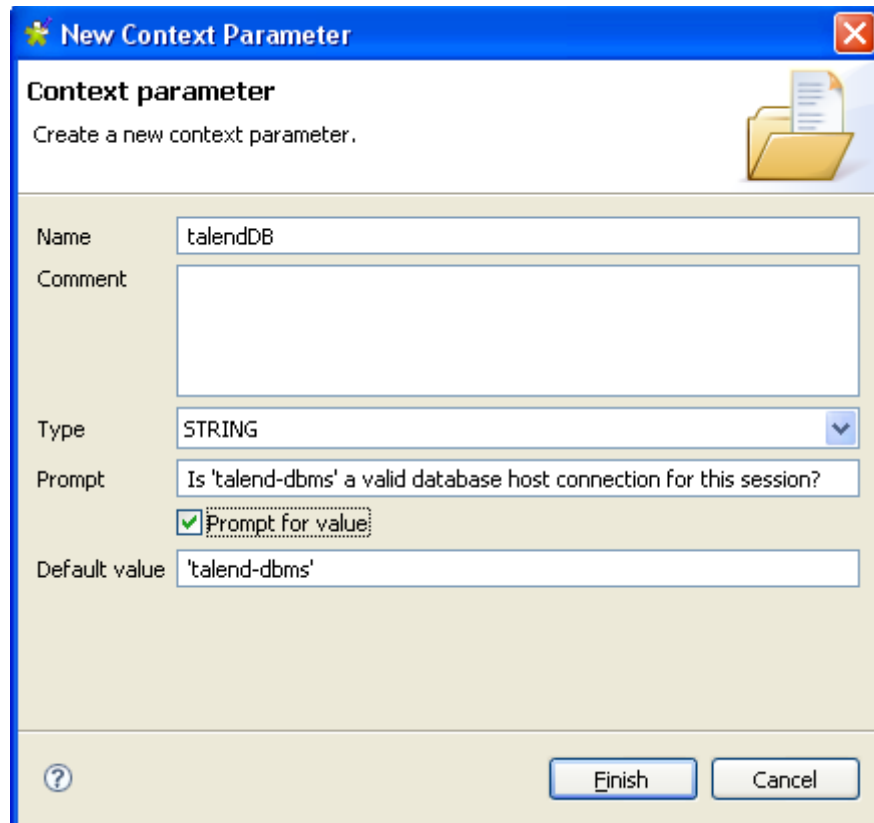
Pour plus d'informations concernant la définition des variables à partir de la vue **Contexts**, consultez *Définir des variables à partir de la vue Contexts*, page 113.

Pour plus d'informations concernant la définition des variables du Repository, consultez *Centraliser les contextes dans le Repository*, page 120.

Création rapide des variables de contexte


Appuyez sur la touche **F5** pour créer rapidement des variables de contexte :

- Dans la vue **Component** correspondante, placez votre curseur dans le champ que vous souhaitez paramétrer.
- Appuyez sur **F5** pour afficher la boîte de dialogue de paramétrage de contexte :



- Donnez un nom à cette nouvelle variable dans le champ **Name**, renseignez le champ **Comment** et choisissez le **Type**.
- Saisissez un message **Prompt** à afficher pour confirmer l'utilisation de cette variable lors de l'exécution de job courant (généralement utilisé comme test). Cochez la case **Prompt for value** pour rendre le champ éditable.
- Si vous avez déjà renseigné le champ correspondant dans la vue **Component** du composant, cette valeur est automatiquement reprise dans le champ **Default value**. Sinon, saisissez la valeur par défaut que vous souhaitez utiliser.
- Cliquez sur **Finish** pour valider.
- Les nouvelles variables que vous venez de créer apparaissent dans la liste des variables de contexte de l'onglet **Contexts**.

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

La variable créée est automatiquement stockée dans le contexte **Default**, mais vous pouvez renommer le contexte à l'aide du bouton **Configure Context** .

Pour plus d'informations concernant la création ou l'édition d'un contexte, consultez *Configurer les contextes*, page 117.

StoreSQLQuery

StoreSQLQuery est une variable définie par l'utilisateur. Elle est principalement dédiée au débogage.

StoreSQLQuery diffère des autres variables de contexte dans le fait qu'elle sert principalement de paramètre pour une variable globale spécifique, dite **Query**. Elle vous permet d'alimenter dynamiquement la variable globale de requête.

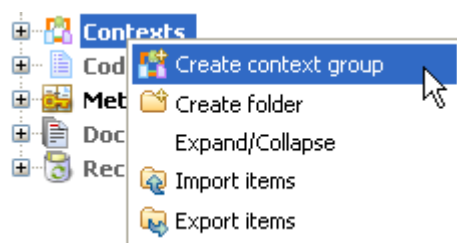
La variable globale **Query** est disponible dans la liste d'autocomplétion (**Ctrl+Esapce**) des propriétés des composants de type DB input.

Pour plus d'informations concernant le paramétrage de **StoreSQLQuery**, reportez-vous aux *Composants Talend Open Studio Guide de référence*.

Centraliser les contextes dans le Repository

Vous avez la possibilité de centraliser tous les contextes si vous avez besoin de les réutiliser dans plusieurs jobs.

Cliquez droit sur le noeud **Contexts** du Repository et sélectionnez **Create new context group** dans le menu contextuel.



L'assistant vous aide à définir les différents contextes et leurs paramètres, que vous pourrez sélectionner à partir de la vue **Contexts** de votre Job Designer.

- Etape 1 : renseignez le nom du context group que vous êtes en train de créer.
- Ajoutez une description si nécessaire.
- Cliquez sur **Next**.

L'étape 2 vous permet de définir les différents contextes et les différentes variables dont vous avez besoin.

Name	Type	Script code	Comment
Servername	String	context.Servername	
Dbname	String	context.Dbname	

Tout d'abord, définissez les variables de contexte que vous souhaitez utiliser comme base pour vos autres contextes.

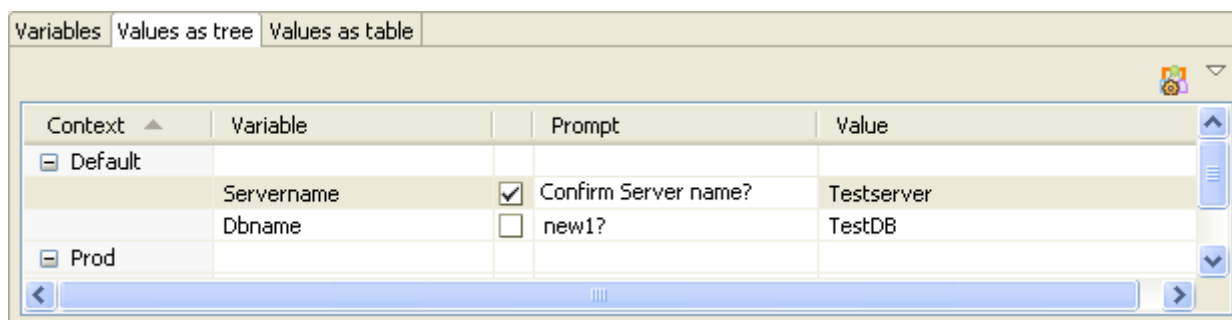
- Dans l'onglet **Variables**, cliquez sur le bouton (+) pour ajouter autant de lignes de variable que nécessaire et nommez ces variables. Dans cet exemple, nous avons défini les variables à partir de l'onglet **Variable** de la vue **Context**.
- Sélectionnez le **Type** de variable dans la liste.
- Le **Script code** varie en fonction du type de variable sélectionné (et du langage de génération de code). Il sera rédigé dans le langage de génération du projet. La capture d'écran ci-dessous vous montre le code Java généré.

Dans les onglets **Tree** ou **Table**, définissez les différents contextes et les valeurs des variables.

Name	Default	Prod
Servername	Testserver	ProdServer
Dbname	TestDB	ProdDB

- Définissez d'abord les valeurs des variables de contexte par défaut.
- Puis créez un nouveau contexte qui sera basé sur les valeurs des variables que vous venez juste de créer. Pour plus d'informations concernant la création de nouveau contexte, consultez *Configurer les contextes*, page 117.

Dans l'onglet **Values as tree**, vous pouvez cocher la case **Prompt** si vous souhaitez que la variable soit modifiable au moment d'exécuter le processus.



- Pour ajouter un message prompt, cochez la case qui précède votre message.
- Puis saisissez le message à afficher lors de l'exécution.

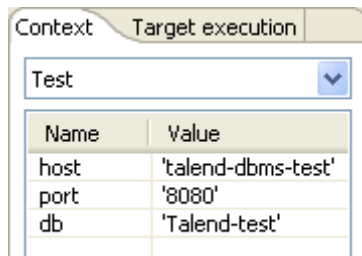
Une fois que vous avez créé et adapté autant de contextes que nécessaire, cliquez sur **Finish** pour valider. Le groupe de contextes configurés apparaît sous le noeud **Contexts** du **Repository**.

Pour appliquer un contexte à un job, cliquez sur l'onglet **Contexts**, sélectionnez **Repository** comme type de contexte dans le champ **Context type**.

Puis sélectionnez le **Context** dans le Repository. Les paramètres du sélectionné apparaissent en lecture seule.

Exécuter un Job dans un contexte défini

Vous pouvez sélectionner le contexte dans lequel vous souhaitez exécuter votre job.



Cliquez sur la vue **Run Job**, et dans la zone **Context**, sélectionnez le contexte approprié.

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 leur valeur associée apparaissent dans un tableau. Si vous avez coché la case **Prompt** lors de la définition du contexte, une

boîte de dialogue s'affiche pour vous demander de confirmer ou modifier la valeur de la variable utilisée pour cette exécution.

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.

Voir également :

- *Définir des variables à partir de la vue Contexts, page 113*
- *Centraliser les contextes dans le Repository, page 120*

Exécuter un job

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.

Si vous avez un niveau utilisateur avancé et souhaitez exécuter votre projet pas à pas, consultez la section *Exécuter en mode debug, page 126*.

Si vous n'avez pas de connaissances avancées (Perl ou Java) et souhaitez exécuter votre job en mode normal, consultez *Exécuter en mode normal, page 124*.

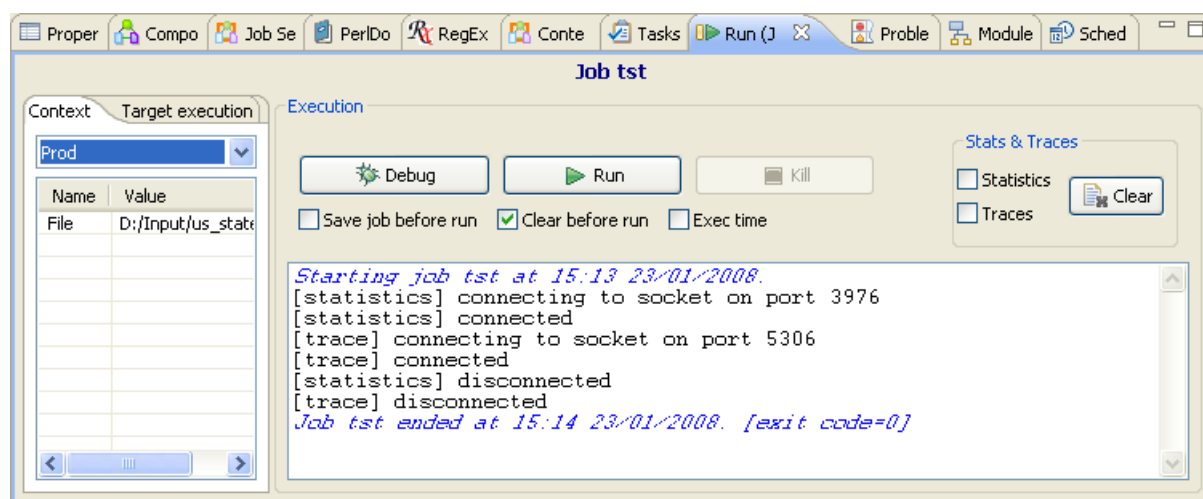
Exécuter en mode normal

Veillez à sauvegarder votre job avant de l'exécuter afin que toutes les propriétés puissent être prises en compte.

- Cliquez sur l'onglet **Run Job** pour accéder au panneau.
- Dans la zone **Context**, sélectionnez le contexte approprié pour ce processus. 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. Voir également : *Définir les Contextes et les Variables, page 112*

- Cliquez sur **Run** pour lancer l'exécution.
- Sur le même panneau, la console de log affiche la progression de l'exécution. La log inclut les messages d'erreurs ainsi que les messages de début et de fin de processus. Elle affiche également le résultat du processus si le composant terminal le prévoit, par exemple tLogRow.



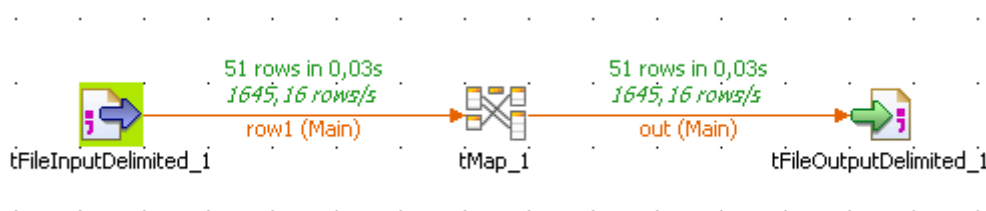
Avant d'exécuter à nouveau un job, vous pouvez vider le contenu de la log, en cliquant sur la case à cocher **Clear before run**. La log sera désormais vidée à chaque exécution.

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.

Talend Open Studio offre d'autres fonctionnalités informatives, notamment Statistics et Traces, qui facilitent la supervision du job ainsi que le travail de débogage.

Afficher les Statistiques

La fonction **Statistics** affiche sous 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. Vous pouvez ainsi plus facilement repérer d'éventuels goulots d'étranglement dans le traitement de votre flux de données.

Note: 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.

Cochez la case **Statistics** pour activer la fonction Statistiques et décochez la case pour la désactiver.

Le calcul des statistiques ne commence qu'au début de l'exécution du job et s'arrête lorsque l'exécution s'arrête.

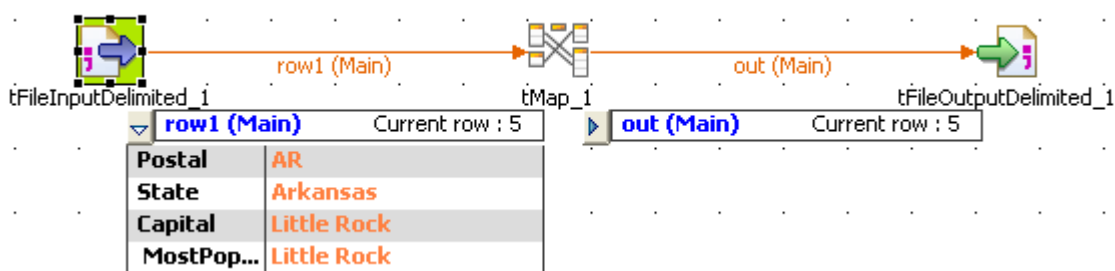
Cliquez sur le bouton **Clear** pour effacer les statistiques affichées. Cochez la case **Clear before Run** pour que la fonction de statistiques se réinitialise automatiquement avant chaque exécution.

Note: Les statistiques peuvent ralentir sensiblement les performances d'exécution du job car il doit envoyer ces données à l'application afin qu'elles soient affichées.

Afficher les Traces

Pour l'instant, la fonctionnalité de suivi est relativement simple dans **Talend Open Studio**. Mais elle devrait être améliorée à l'avenir.

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 superviser tous les composants d'un job, sans avoir besoin de passer en mode Debug, et par conséquent sans avoir besoin de connaissances Perl/Java particulières.

La fonction Traces affiche le contenu des lignes traitées dans un tableau.

Note: Les composants externes ne peuvent offrir cette fonctionnalité que si leur conception le prévoit.

Cochez la case **Traces** pour activer la fonction de suivi du traitement et décochez-la pour la désactiver.

La fonction Trace ne s'exécute qu'à l'exécution du job et s'arrête à la fin de celui-ci. Si la table Trace prend trop de place dans votre Job Design, vous pouvez la réduire à l'aide de la flèche en haut à gauche de la table Trace.

Cliquez sur **Clear** pour effacer les statistiques affichées.

Note: Le tableau est limité horizontalement, néanmoins si vous passez votre souris au-dessus du tableau, la totalité des informations s'affiche. A contrario, il n'existe aucune limitation verticale de ce tableau. Cela peut poser problème si le tableau est très long.

Afficher la durée d'exécution et d'autres options

Il vous suffit de cocher la case **Exec time** avant de lancer votre job pour qu'à la fin de l'exécution de votre job, sa durée totale s'affiche.

De cette manière vous pouvez vérifier les résultats ou tester votre job avant de l'envoyer en production.

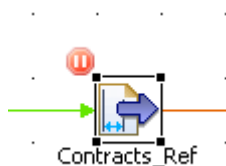
Vous pouvez aussi nettoyer votre espace de modélisation avant chaque exécution en cochant la case **Clear before run**.

Vous pouvez aussi enregistrer votre job avant que l'exécution ne commence, en cochant la case **Save job before run**.

Exécuter en mode debug

Avant d'exécuter un job en mode Debug, vérifiez que le module EPIC est installé.

Dans un premier temps nous vous recommandons d'ajouter des points de pause au niveau des étapes principales de votre processus.



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.

Pour ajouter des pauses (breakpoints) 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 pause s'affiche à gauche du composant dans l'éditeur graphique.

Pour passer en mode debug, cliquez sur le bouton **Debug** dans le panneau **Run Job**. **Talend Open Studio** ouvre l'éditeur en mode de débogage.

Vous pouvez exécuter le job pas à pas et vérifier le comportement de chacun des composants ainsi que les valeurs des variables utilisées.

Pour revenir en mode normal **Talend Open Studio**, cliquez sur **Window**, puis **Perspective** et sélectionnez **Talend Open Studio**.

Enregistrer ou exporter un Job

Enregistrer un Job

Lorsque vous fermez l'application **Talend Open Studio**, une boîte de dialogue vous demande de sauvegarder les jobs courants si vous ne l'avez pas déjà fait.

Si plusieurs jobs ne sont pas enregistrés, cochez la case correspondant aux jobs que vous souhaitez sauvegarder. Le job est enregistré dans le dossier du projet dont il fait partie, dans le répertoire Workspace.

Vous pouvez également cliquer sur le menu **File > Save** ou appuyer sur **Ctrl+S** pour enregistrer vos jobs.

Exporter des job scripts

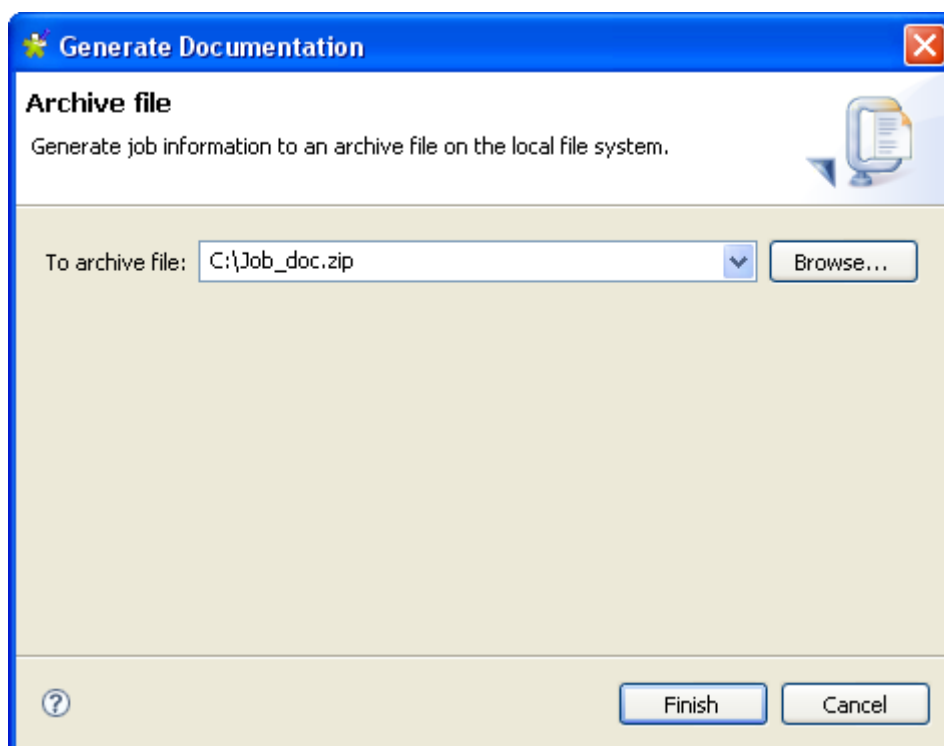
Pour plus d'informations sur l'export de job scripts, consultez *Exporter des job scripts*, page 127.

Documentation du Job

Générer la documentation HTML

Talend Open Studio vous permet de générer de la documentation détaillée au format HTML des jobs sélectionnés.

- Dans le **Repository**, cliquez droit sur l'un de vos **Jobs** ou sélectionnez plusieurs **Job Designs** pour générer une documentation multiple.
- Sélectionnez **Generate Doc as HTML** dans le menu contextuel.



- Renseignez le répertoire dans lequel doit être enregistré la documentation générée.
- Dans le même champ, renseignez le nom de l'archive regroupant les documents générés.
- Cliquez sur **Finish** pour valider.

Le fichier Archive contient les fichiers HTML ainsi que les dossiers correspondants. Ouvrez le fichier HTML dans votre navigateur préféré.

Définir les options dans la vue Job Settings

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

L'onglet **Stats & Logs** vous permet d'utiliser de manière automatique les fonctions **Stats & Logs** et la fonction d'automatisation des paramètres de Contexte.

L'onglet **Extra** liste les différentes options disponibles pour automatiser certaines fonctions.

Automatiser l'utilisation de statistiques & logs

Si vous utilisez souvent Logs, Statistics et toute autre mesure de votre flux de données, vous devez gérer beaucoup de composants liés aux logs dans vos job designs. A l'aide de l'onglet **Stats & Logs**, vous pouvez automatiser l'utilisation des fonctionnalités de **tFlowMeterCatcher**, **tStatCatcher** et **tLogCatcher** sans utiliser ses composants dans votre job.

Pour plus d'information sur le composant Log, consultez les .

L'onglet **Stats & Logs** est situé dans la vue **Job Settings** sous l'éditeur et vous permet d'éviter de surcharger vos job designs avec des composants.

Note: Cette configuration a les mêmes propriétés générale de log que les composants Log.

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

- Cliquez sur la vue **Job Settings**.
- Sélectionnez l'onglet **Stats & Logs** pour afficher la vue de configuration.

- Renseignez les informations en fonction de votre composant de sortie (console, fichier ou base de données).
- Vérifiez que l'option **Catch** correspond bien à vos besoin.

Note: Vous pouvez enregistrer automatiquement ces configurations dans vos Préférences. Ou sinon, accédez à vos préférences dans **Window > Preferences > Talend > Stats & Logs**.

Fonctions Extra

L'onglet **Extra** vous offre des options de paramétrage.

- Cochez la case **Multi-thread execution** pour lancer deux jobs au même moment.
- Configurez l'option **Implicit tContextLoad** pour éviter d'utiliser le composant **tContextLoad** dans votre job et ainsi automatiser l'utilisation des paramètres de contexte. Choisissez la source de vos paramètres de contexte : **File** ou **Database**. Configurez les messages (erreur/warning/info) lorsque des comportements inattendus liés aux paramètres de contexte surviennent.

Raccourcis clavier

Le tableau ci-dessous rassemble tous les raccourcis clavier utilisés dans **Talend Open Studio** :

Raccourci	Pour...	Contexte
F2	Afficher la vue Component .	Application globale
F4	Afficher la vue Run Job .	Application globale
F6	Exécuter le job courant ou afficher la vue Run Job si aucun job n'est ouvert.	Application globale
Ctrl + F2	Afficher la vue Module .	Application globale
Ctrl + F3	Afficher la vue Problems .	Application globale
Ctrl + H	Revenir à la vue Job Design courante.	Application globale
Ctrl + G	Afficher l'onglet Code du job courant.	Application globale
Ctrl + Maj + F3	Synchroniser les templates perljet des composants et les classes java associées.	Application globale
F7	Passer en mode Debug .	Dans la vue Run Job
F5	Créer une variable de contexte à partir d'un champ Settings.	Dans la vue Component d'un job
F8	Tuer le job courant.	Dans la vue Run Job
F5	Rafraichir le statut d'installation des Modules .	Dans la vue Modules
Ctrl+Barre d'espace	Accéder aux variables globales et de contexte. Cela peut être des messages d'erreur ou le numéro de ligne, en fonction du composant sélectionné.	Dans un champ de la vue Component

—De la théorie à la pratique : Exemple de Job—

De la théorie à la pratique : Exemple de Job

Ce chapitre est destiné aux utilisateurs de **Talend Open Studio** qui cherchent des cas réels d'utilisation de **Talend Open Studio** afin de maîtriser le mieux possible le produit. Ce chapitre est un complément du Guide de référence des composants.

Présentation du scénario

Pour illustrer le fonctionnement de **Talend Open Studio**, vous trouverez ci-dessous un scénario de la vie courante. Dans ce scénario, vous devez charger un fichier dans une table MySQL en appliquant des transformations à la volée. Et dans une étape suivante, vous sélectionnez les données à charger en appliquant un filtre dynamique.

Avant de commencer le job, vérifiez les données en entrée (Input) et les données attendues en sortie (Output).

Note: Ce scénario a été réalisé en Java, mais il peut aussi être réalisé dans un projet Perl. Dans ce cas, les étapes de transformation comporteront d'importantes différences.

Données en entrée

Le fichier en entrée, dont les données seront chargées dans une table de données, fournit une liste des clients de tous les états de la Californie.

La structure du fichier, communément appelée **Schéma** dans **Talend Open Studio** comprend les colonnes suivantes :

- *First name* (prénom)
- *Last name* (nom)
- *Address* (adresse)
- *City* (ville)

Données en sortie

Vous souhaitez uniquement charger les clients vivants dans certains comtés de la Californie dans la base de données : les comtés d'Orange et de Los Angeles.

La structure de la table est légèrement différente, ainsi les données devant être chargées dans le table de données doivent être structurées de la manière suivante :

- *Key* (Clé, Type entier)
- *Name* (Type chaîne, longueur max. 40)
- *Address* (Type chaîne, longueur max. 40)
- *County* (Type chaîne, longueur max. 40)

Pour charger cette table, vous devez utiliser les processus de mapping suivants :

La colonne *Key* est alimentée avec un entier auto-incrémenté.

La colonne *Name* est renseignée avec une concaténation des données *First Name* et *Last Name*.

Les données de la colonne *Address* sont les mêmes que celles de la colonne *Address* du fichier d'entrée, et elles seront mises en majuscule avant d'être chargées.

La colonne *County* est alimentée avec le nom du comté dans lequel se situe la ville. Un fichier de référence vous aidera à filtrer les villes des comtés d'Orange et de Los Angeles.

Données de référence

Etant donné que les données des comtés d'Orange et de Los Angeles doivent être chargées dans la base de données, vous devez mapper les villes de Californie avec leurs comtés respectifs, afin de pouvoir filtrer uniquement ceux d'Orange et de Los Angeles.

Pour cela, utilisez un fichier de référence contenant la liste des villes situées dans les comtés d'Orange et de Los Angeles, par exemple :

City	County
Agoura Hills	Los Angeles
Alhambra	Los Angeles
Aliso Viejo	Orange
Anaheim	Orange
Arcadia	Los Angeles

Le fichier de référence de ce job se nomme *LosAngelesandOrangeCounties.txt*.

Du scénario au Job

Pour mettre ce scénario en pratique, séparez ce job en quatre étapes.

- Etape 1 : Création du job, configuration des paramètres et lecture du fichier d'entrée
- Etape 2 : Mapping et transformations de données
- Etape 3 : Définition des paramètres du fichier de référence, mapping correspondant à l'aide du composant tMap et sélection du mode Inner Join.
- Etape 4 : Redirection des données en sortie dans une table MySQL

Etape 1 : Création du Job, définition des données d'entrée, lecture du fichier

Après avoir lancé **Talend Open Studio**, créez une connexion au référentiel local en appuyant sur le bouton “...” à droite de **Connection**.

Dans le champ **User Email**, saisissez votre email (!) puis cliquez sur **OK**.

Puis cliquez sur le bouton **Demo** et sélectionnez Java pour importer les jobs du projet DEMO JAVA. Cette opération demande un peu de temps mais elle vous donnera accès à plusieurs dizaines d'exemples de jobs, illustrant les fonctionnalités principales de **Talend Open Studio**.

Cliquez ensuite sur **OK** pour lancer le Studio. Puis cliquez sur le lien **Start using Talend Open Studio now!** pour accéder directement à la fenêtre principale.

Cette fenêtre est divisée en plusieurs espaces :

- A gauche : le **Référentiel** (Repository) dans lequel sont référencés tous les Jobs, Business Models, Métadonnées, Code partagé, Documentation, etc.
- Au centre : l'**Editeur** (espace de modélisation principal)
- En bas : les onglets **Component** et **Job Settings**, etc.
- A droite : la **Palette** des composants métier ou techniques en fonction de l'outil utilisé dans **Talend Open Studio**.

Sur la gauche du Studio, le Référentiel donne accès aux trois principaux outils de **Talend Open Studio**:

- Le **Business Modeler** : Pour plus d'informations, consultez *Modélisation d'un Business Model*, page 27.
- Le **Job Designer** : Pour plus d'informations, consultez *Conception d'un Job*, page 41.
- Le **Metadata Manager** : Pour plus d'informations, consultez *Définition des schémas de métadonnées*, page 58.

Pour créer le job, cliquez d'abord sur l'élément **Job Designs** du Référentiel avec le bouton droit de la souris et sélectionnez la première option du menu : **Create Job**.

Dans la boîte de dialogue qui apparaît alors à l'écran, seul le premier champ **Name** est obligatoire. Saisissez **California1** et cliquez sur **Finish**.

Un job vide s'ouvre ensuite dans la fenêtre principale et la Palette de composants techniques apparaît (par défaut, à droite du Studio) affichant une dizaine de familles de composants, notamment : Databases, Files, Internet, Data Quality, etc. Plus de 200 composants sont disponibles actuellement.

Pour lire le fichier *California_Clients*, utilisez le composant **tFileInputDelimited**. Ce composant se trouve dans la famille **File > Input** de la Palette. Cliquez sur ce composant puis cliquez sur la gauche de l'espace de modélisation pour le placer.

Définissez maintenant les propriétés de lecture de ce composant : chemin d'accès, séparateur de colonnes, encodage, etc. Pour ce faire, utilisez le **Metadata Manager**. Cet outil possède de nombreux assistants qui vous aideront à configurer les paramètres nécessaires et vous permettront de conserver ces propriétés qui pourront être réutilisées en un seul clic dans de futurs jobs.

Puisque votre fichier d'entrée est un fichier plat délimité, cliquez sur **Metadata > File Delimited** dans le Référentiel et dans le menu contextuel du noeud **File Delimited** sélectionnez l'option **Create file delimited**. L'assistant spécifique aux fichiers délimités apparaît :

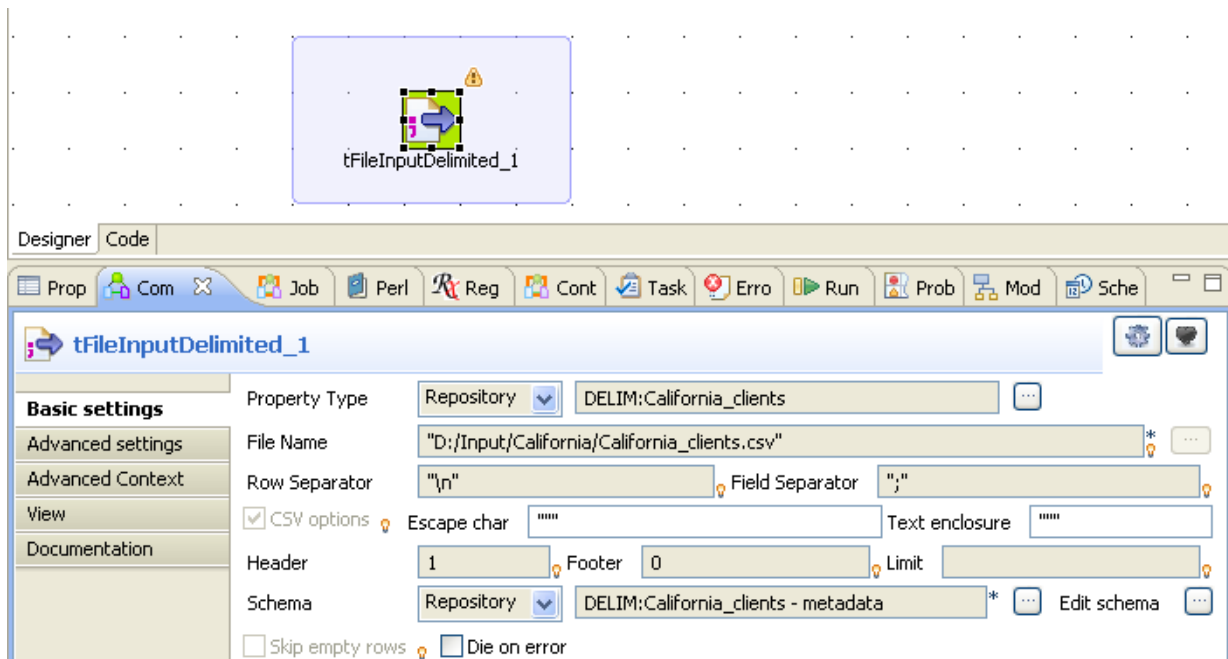
- A l'étape 1, seul le champ **Name** est obligatoire : saisissez le nom **California_clients** et passez à l'étape suivante.
- A l'étape 2, sélectionnez le fichier d'entrée (*California_Clients.csv*) via le bouton **Browse...**. Un extrait du fichier apparaît immédiatement dans la zone **File viewer** en bas de l'assistant, afin que vous puissiez en vérifier le contenu. Cliquez sur **Next**.
- A l'étape 3, définissez les paramètres du fichier : encodage, séparateurs de colonnes et de lignes, etc. Puisque votre fichier d'entrée est standard, vous pouvez garder la plupart des valeurs par défaut. La première ligne de votre fichier est un en-tête contenant les noms de colonnes. Pour récupérer automatiquement ces noms, cochez la case **Set heading row as column names** et cliquez sur **Refresh Preview**. Cliquez sur **Next** pour passer à l'étape suivante.
- A l'étape 4, définissez chaque colonne de votre fichier. L'assistant intègre des algorithmes qui essaient de deviner les types et la longueur des données contenues dans les colonnes du fichier en analysant les premières lignes. La description des données (appelé schéma dans **Talend Open Studio**) peut être modifiée à tout moment. Pour ce scénario particulier, ces informations peuvent être gardées tel quel.

La métadonnée **California_clients** est maintenant définie !

Vous pouvez donc l'utiliser dans votre composant d'entrée. Sélectionnez le composant **tFileInputDelimited** que vous avez déposé dans l'espace de modélisation, et sélectionnez la vue **Component Settings** dans le bas de la fenêtre.

Sélectionnez l'onglet vertical **Basic Settings**. Dans cet onglet, vous trouverez toutes les propriétés techniques nécessaires au composant. Au lieu de les saisir une à une, utilisez la métadonnée que vous venez de créer.

Sélectionnez **Repository** dans la liste déroulante **Property type**. Un nouveau champ apparaît : cliquez sur le bouton "... " et sélectionnez la métadonnées correspondante dans la liste : **California_clients**. Notez que tous les paramètres sont automatiquement renseignés.



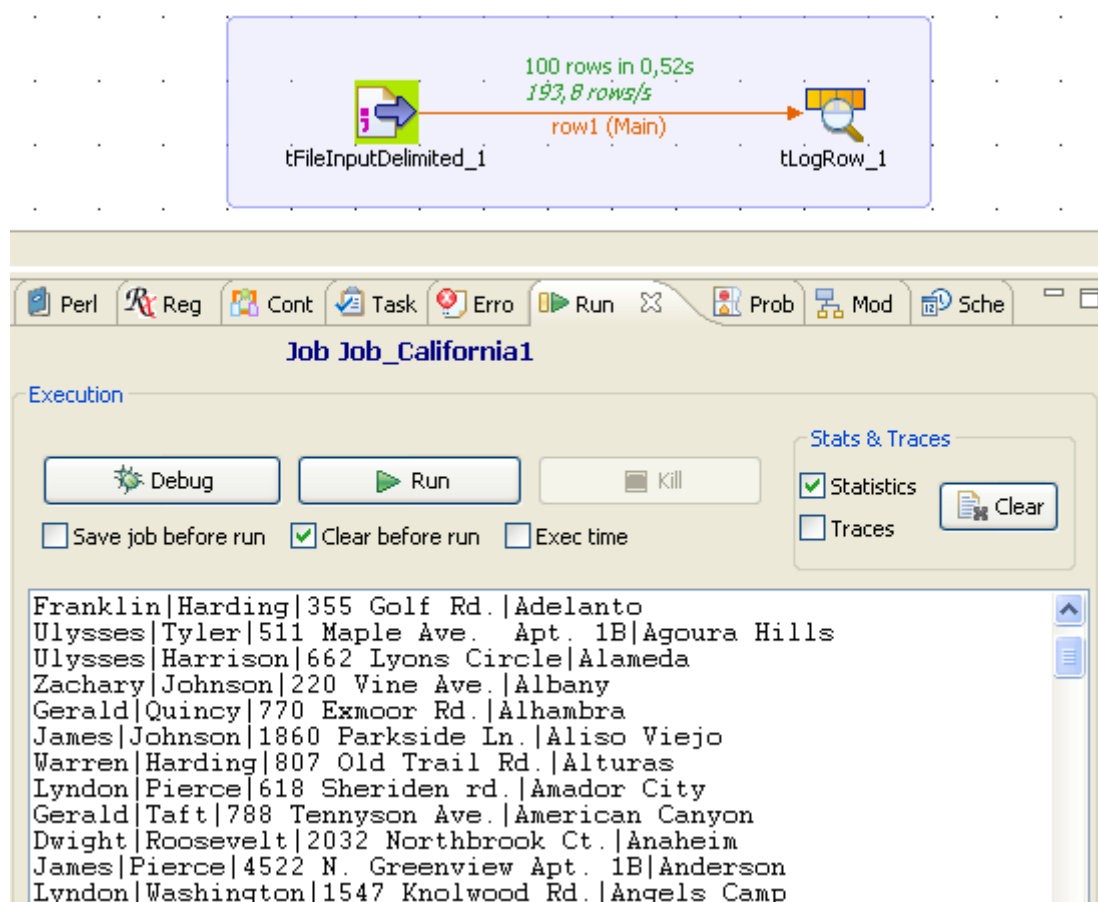
A cette étape, terminez votre flux en envoyant tout simplement les données lues dans le fichier d'entrée vers une sortie standard (StdOut).

Pour ce faire, ajoutez un composant **tLogRow** (de la famille **Logs & Errors**).

Pour lier ces deux composants, cliquez-droit sur le composant d'entrée et sélectionnez **Row > Main**. Puis cliquez sur le composant de sortie **tLogRow**.

Ce Job est maintenant prêt à être exécuter. Pour l'exécuter, sélectionnez la vue **Run** dans le bas de la fenêtre.

Activez les statistiques en cochant la case **Statistics**, puis exécutez le job en cliquant sur le bouton **Run**.



Le contenu du fichier d'entrée apparaît dans la console de la vue **Run Job**.

Etape 2 : Mapping et transformations

Vous allez maintenant enrichir votre job en ajoutant des transformations à la volée. Pour effectuer ses transformations, utilisez le composant **tMap** dans votre job. Ce composant est multiple et peut gérer :

- entrées et sorties multiples
- recherche de référence (simple, produit cartésien, première et dernière correspondance, etc.)
- jointure (inner join, outer join)
- transformations
- rejets
- etc.

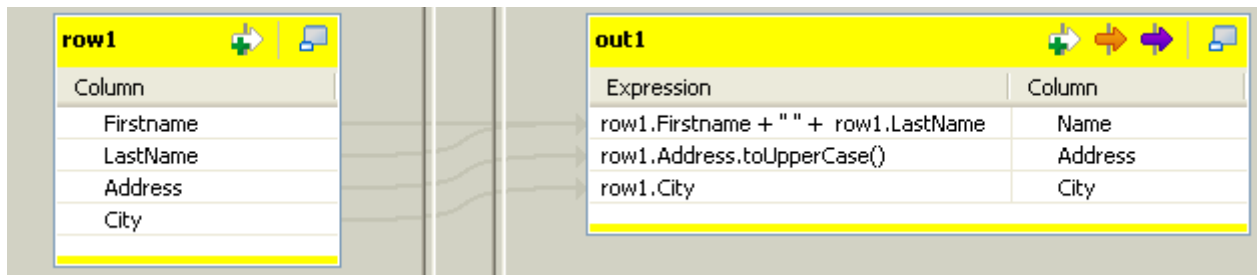
Supprimez la connexion reliant vos deux composants via un clic-droit sur cette connexion, et en sélectionnant l'option **Delete**. Puis placez **tMap** entre les deux autres composants et reliez le composant d'entrée au **tMap** comme vous l'avez fait précédemment.

Enfin, pour lier le composant **tMap** à la sortie standard, cliquez-droit sur le **tMap** et sélectionnez **Row > *New Output* (Main)**. Saisissez *out1* dans la boîte de dialogue et cliquez sur le composant **tLogRow** pour créer la connexion. Logiquement, une boîte de dialogue apparaît (pour la retro-propagation des schémas), ignorez-la en cliquant sur **No**.

Maintenant, double-cliquez sur le **tMap** pour accéder à son interface.

A gauche, vous trouverez le schéma (description) de votre fichier d'entrée (*row1*). A droite, votre sortie est encore vide pour le moment (*out1*).

Glissez-déposez les colonnes *FirstName* et *LastName* de la gauche vers la droite, comme le montre la capture d'écran suivante. Puis glissez-déposez les autres colonnes *Address* et *City* dans leur ligne respective.



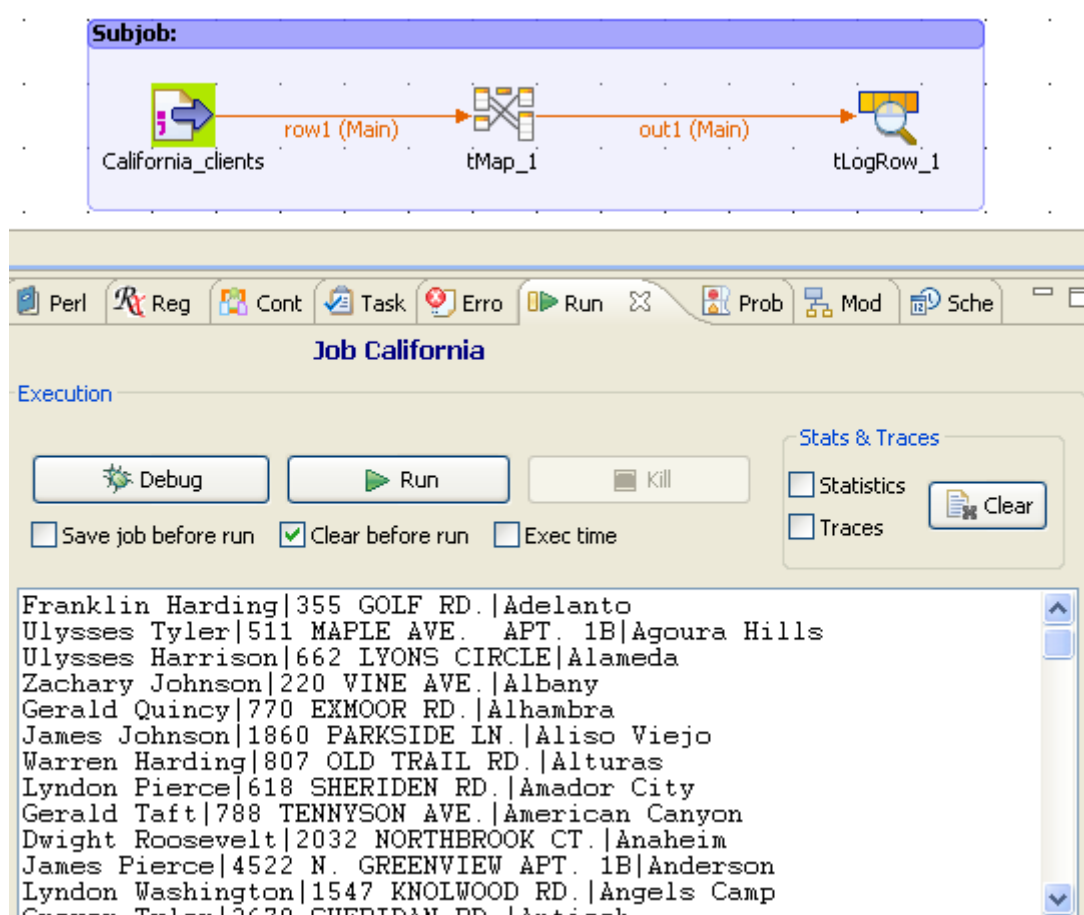
Puis effectuez les transformations suivantes sur chaque colonne :

- `row1.Firstname + " " + row1.LastName` : concatène les colonnes *Firstname* et *Lastname* dans une seule colonne.
- `row1.Address.toUpperCase()` : met l'adresse en majuscule.

Note: Ces transformations sont effectuées en Java pour ce Job. Vous utiliserez une syntaxe différente si vous êtes dans un projet Perl.

Aucune transformation n'est effectuée sur la colonne *City*. Cliquez sur **OK** pour valider les modifications et fermer l'interface du **tMap**.

Si vous exécutez votre job maintenant (via la vue **Run tab** comme dans l'étape précédente), vous pourrez observer que les modifications effectuées ont bien été appliquées.



Par exemple, l'adresse a été mise en majuscule et les prénoms et noms ont été regroupés dans une seule colonne.

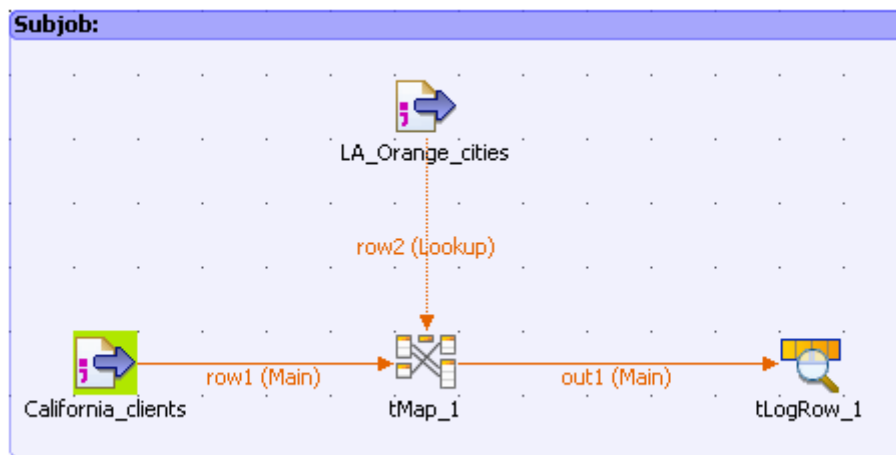
Etape 3 : Définition du fichier de référence, mapping des données de référence, sélection du mode Inner Join

Définissez la métadonnées correspondant au fichier *LosAngelesandOrangeCounties.txt* à l'aide de l'assistant, comme vous l'avez fait dans l'étape 1 avec le fichier *California_clients*.

A l'étape 1 de l'assistant, nommez cette métadonnée *LA_Orange_cities*.

Puis glissez-déposez cette nouvelle métadonnée en haut de l'espace de modélisation, cela créera automatiquement un composant de lecture pointant vers cette métadonnées.

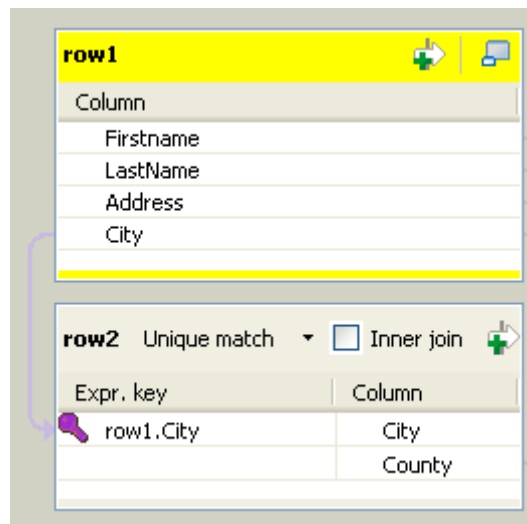
Reliez ce composant au **tMap**.



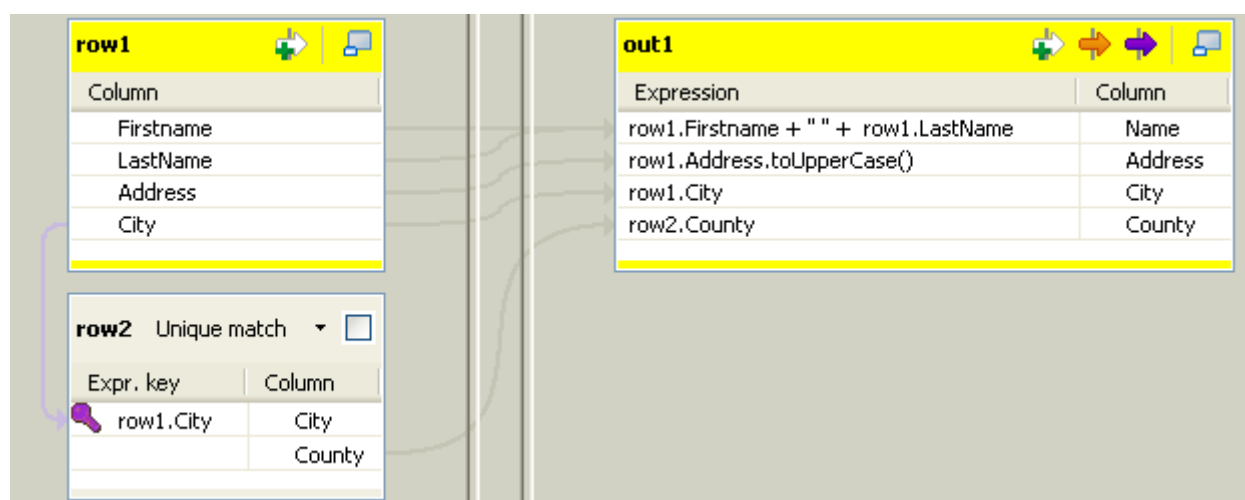
Double-cliquez de nouveau sur le composant **tMap** pour ouvrir son interface. Notez que la table de référence (*row2*) correspondant au fichier *LosAngelesandOrangeCounties.txt*, apparaît à gauche de la fenêtre dans la zone Input d'entrée, juste en dessus de votre flux d'entrée principal (*row1*).

Maintenant, définissez la jointure entre le flux principal et le flux de référence. Dans ce scénario, La jointure est simple à définir puisque la colonne *City* est présente dans les deux fichiers d'entrée et que les données correspondent parfaitement. Mais si ça n'avait pas été le cas, il aurait été possible de faire des opérations directement à ce niveau pour rapprocher les données (padding, chagement de casse, etc.)

Pour établir le jointure, glissez-déposez la colonne *City* de la première table d'entrée vers la colonne *City* de la table de référence. Un lien violet apparaît pour matérialiser cette jointure.

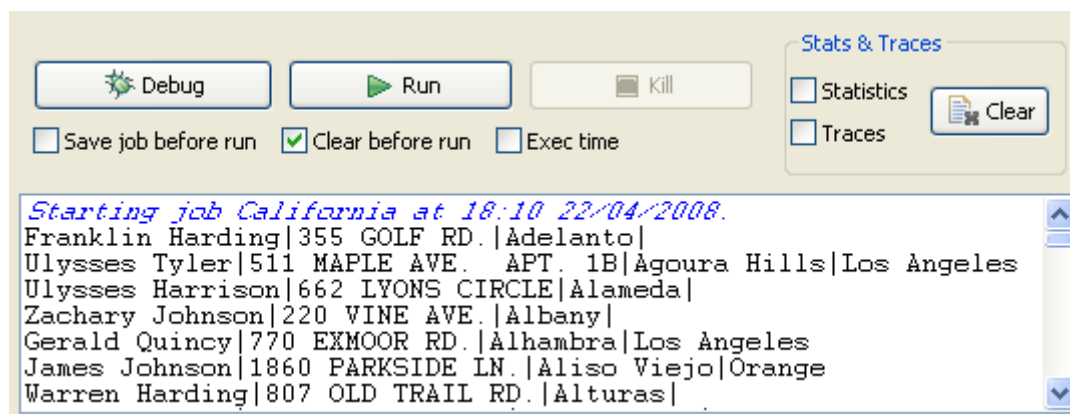


Maintenant, vous pouvez utiliser la colonne *County* de la table de référence dans la table de sortie (*out1*).



Enfin, cliquez sur le bouton **OK** pour valider les modifications et exécutez ce nouveau job.

La sortie suivante s'affichera dans la console.



Comme vous pouvez le voir, la dernière colonne ne contient que les villes des comtés d'*Orange* et de *Los Angeles*. Pour les autres villes, cette colonne reste vide. Ceci est dû au fait que par défaut le **tMap** établit une jointure Inner Join. Si vous souhaitez appliquer un filtre permettant de n'afficher que les données pour lesquelles une correspondance a été trouvée par le **tMap**, ouvrez de nouveau le **tMap** et cochez la case **Inner Join** sur la table de référence (*row2*).

Etape 4 : Sortie vers une table MySQL

Votre job fonctionne à merveille ! Pour le finaliser, redirigez le flux de sortie vers une table MySQL.

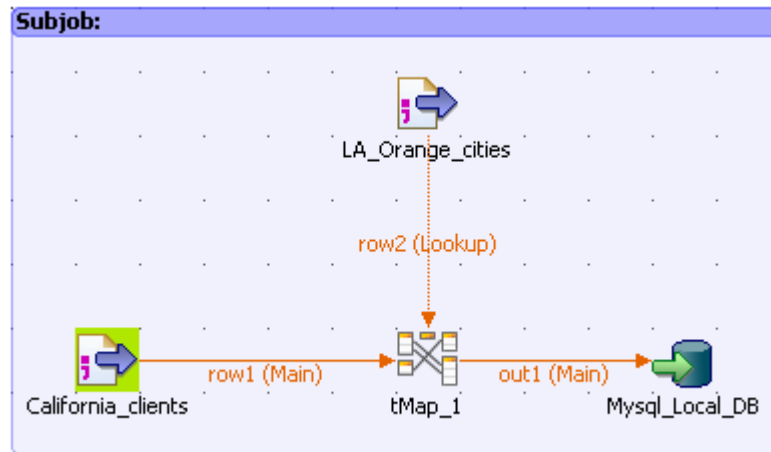
Pour cela, créez tout d'abord une métadonnée décrivant la connexion à la base de données MySQL. Double-cliquez sur **DemoMySQL** dans le répertoire **Metadata > MySQL** du Référentiel (à condition que vous ayez bien importé le projet **Demo**) pour lancer l'assistant Metadata.

A l'étape 2 de l'assistant, renseignez les paramètres de connexion à la base de données. Vérifiez la validité de cette connexion en cliquant sur le bouton **Check**. Enfin, validez vos modifications en cliquant sur **Finish**.

Glissez-déposez cette métadonnée à droite de l'espace de modélisation en maintenant la touche **Ctrl** enfoncée pour créer automatiquement un composant **tMysqlOutput**.

Supprimez le composant **tLogRow** de votre job.

Reconnectez le flux de sortie *out1* du tMap vers le composant tMysqlOutput
(Clique-droit > Row > out1) :



Dans l'onglet **Basic Settings** de ce composant :

- Saisissez **LA_Orange_clients** dans le champ **Table** pour nommer votre table cible qui va être créée à la volée.
- Sélectionnez l'option **Drop table if exists and create** dans le champ **Action on table**.

Exécutez à nouveau le job. La table cible devrait être automatiquement créée et remplie en moins d'une seconde !

Dans ce scénario, seuls quatre composants différents sont utilisés, mais la **Palette** en contient plus de 200 (bases de données, Webservices, Ftp, etc.) !

D'autres composants, réalisés cette fois par la communauté, sont disponibles sur le site communautaire de l'éditeur : talendforge.org.

Pour plus d'informations concernant les composants, consultez les *Composants Talend Open Studio Guide de référence*.

—Gestion des Jobs & Projets—

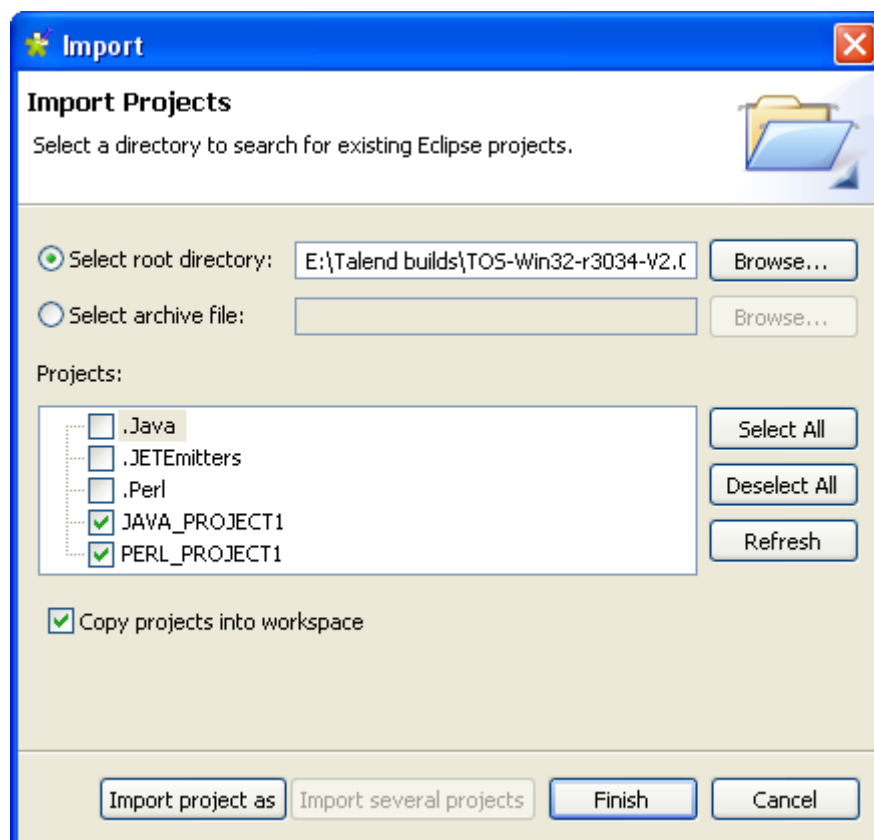
Gestion des Jobs & Projets

Vous pouvez importer les projets d'une version antérieure de **Talend Open Studio**. Vous avez aussi la possibilité d'importer ou d'exporter des jobs d'un projet à un autre ou d'une machine à une autre.

Importer des projets

Sur la fenêtre de connexion, cliquez sur **Import** pour lancer l'assistant d'Import de projet. Cliquez sur **Open several projects** pour importer plusieurs projets à la fois.

Ou à partir de la fenêtre principale de **Talend Open Studio**, cliquez sur le bouton **Import projects** dans la barre d'outils.



Cliquez sur **Browse...** pour sélectionner le répertoire **Workspace** ou le dossier contenant les projets à importer. Par défaut, le workspace sélectionné est le dossier workspace courant. Remontez

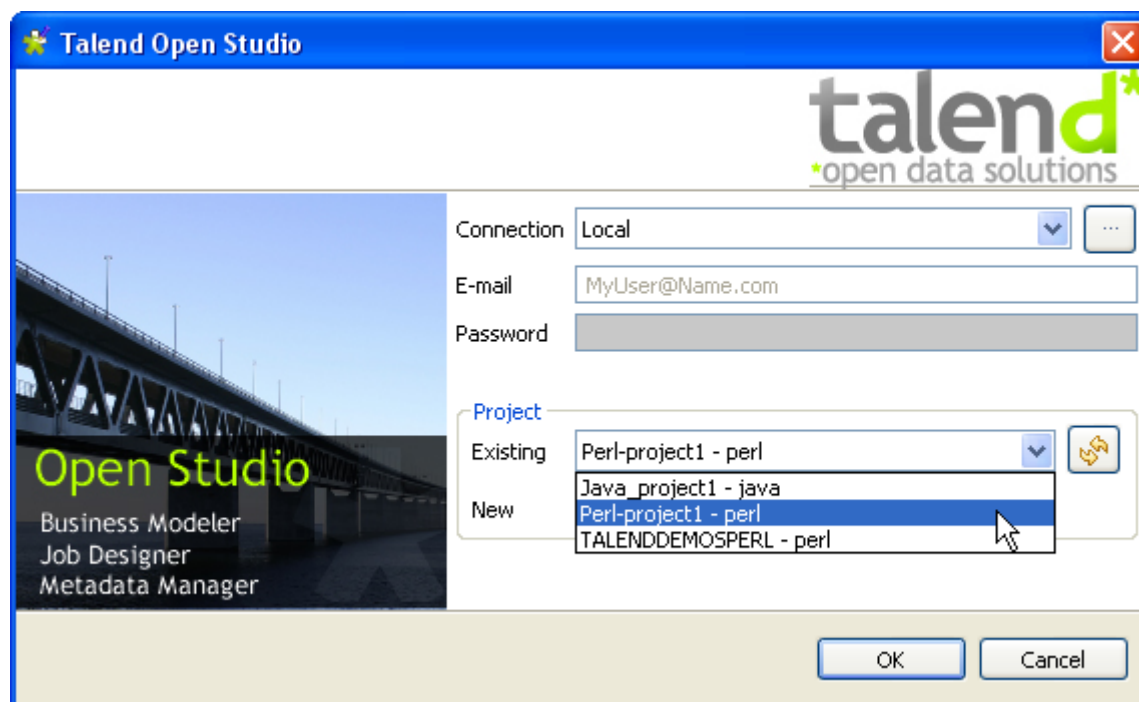
l'arborescence vers le répertoire **Workspace** de la version antérieure de **Talend Open Studio** et sélectionnez-le.

Cochez la case **Copy projects into workspace** pour faire une copie des projets importés. Si vous souhaitez déplacer les projets de leur emplacement initial vers le nouveau workspace, décochez cette case. Nous vous recommandons de conserver une version de sauvegarde.

Sélectionnez les projets à importer et cliquez sur **Finish** pour valider.

Sur la fenêtre de connexion, les projets importés apparaissent désormais dans la liste **Project** proposée.

Ou si vous vous trouvez dans la fenêtre principale de **Talend Open Studio**, cliquez sur **File > Switch projects...** pour revenir à la fenêtre de connexion et la liste des projets proposés.



Sélectionnez le projet que vous souhaitez ouvrir et cliquez sur **OK** pour lancer **Talend Open Studio**.

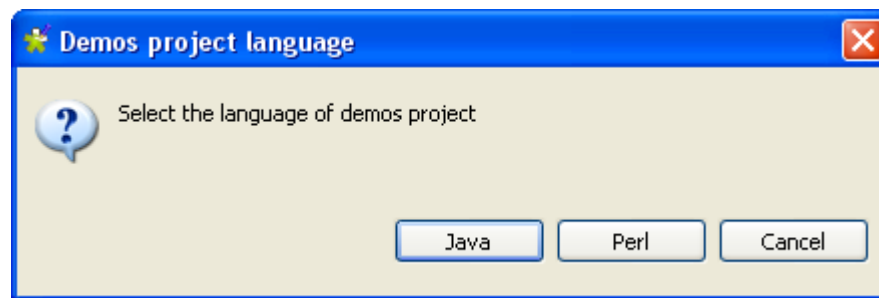
Note: Une fenêtre d'initialisation du générateur de code peut apparaître lors du lancement de l'application. Il est recommandé d'attendre que l'initialisation se termine.

Si vous ne souhaitez pas importer la totalité d'un projet, vous pouvez sélectionner les éléments séparément. Pour plus d'informations, consultez *Importer des échantillons (Demos)*, page 144.

Importer des échantillons (Demos)

Comme pour l'import de projets, vous pouvez importer dans votre répertoire Workspace courant le dossier Demos contenant de nombreux exemples de jobs.

Dans la fenêtre de connexion de **Talend Open Studio**, cliquez sur le bouton **Demos**.



Sélectionnez le langage de génération entre Perl et Java.

Des exemples de jobs couvrant tous les types de besoins sont automatiquement importés dans votre espace de travail (workspace) et accessibles via le référentiel (Repository).

Un message s'affiche confirmant le bon déroulement de l'opération d'import.

Le projet Demos s'ajoute à la liste des projets existants (**existing Projects**) dans la fenêtre de connexion.

Vous pouvez utiliser ces échantillons de jobs comme base pour votre propre job design.

Importer des éléments

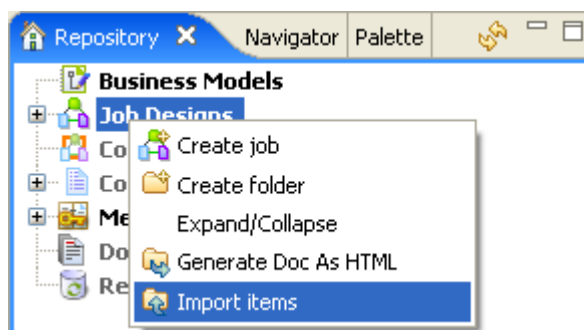
Vous pouvez à présent importer des éléments stockés dans le workspace des versions antérieures de **Talend Open Studio** ou d'un autre projet de votre workspace courant.

Les éléments que vous pouvez importer sont :

- Business Models
- Jobs Designs
- Routines
- Documentation
- Metadata

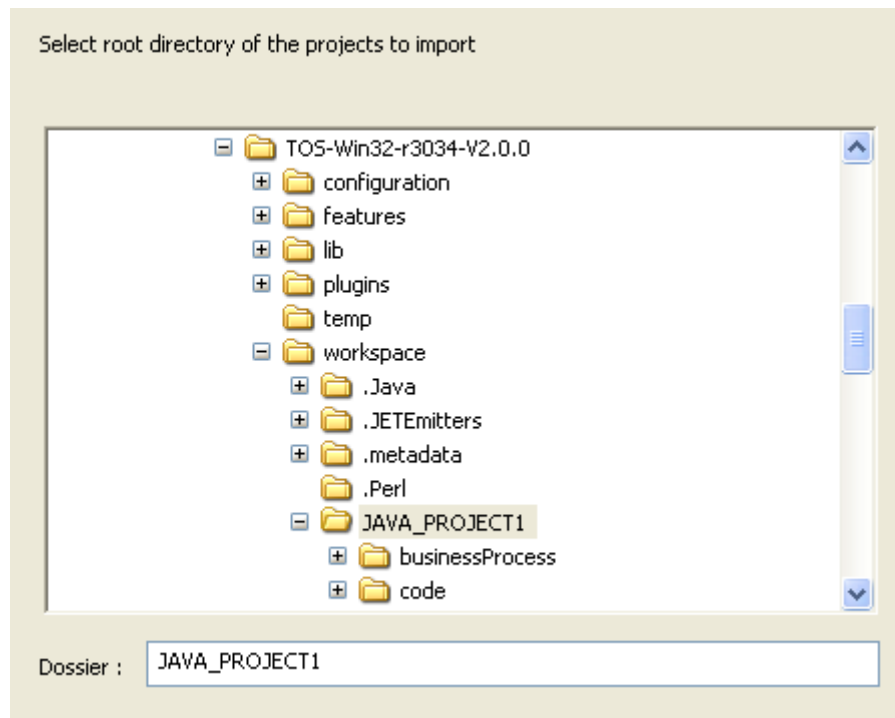
Pour importer des éléments vers un Référentiel , suivez les étapes suivantes :

- Dans le Repository, cliquez droit sur n'importe quelle entrée, notamment Job Designs ou Business Models.
- Dans le menu contextuel, sélectionnez l'option **Import Items**.

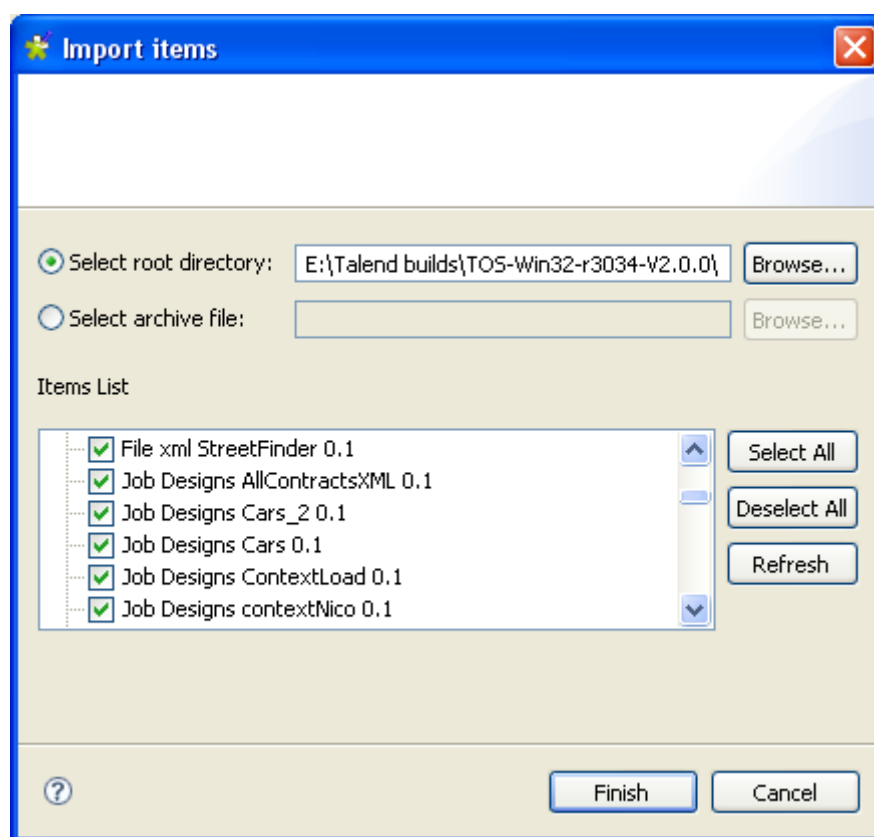


- Une boîte de dialogue vous demande de sélectionner le répertoire ou le fichier archive contenant les éléments que vous souhaitez extraire.
- Si vous avez exporté les éléments de votre référentiel local dans un fichier (fichiers source et scripts inclus), sélectionnez le dossier contenant l'archive puis cliquez sur **OK**.

- Si les éléments à importer sont encore stockés sur le référentiel local, sélectionnez le dossier du projet sur votre système.
- Sélectionnez le dossier correspondant au **Projet** dans le répertoire **Workspace**. Il correspond au nom du projet que vous créez auparavant.



- Si vous souhaitez importer des éléments spécifiques, notamment les **Job Designs**, sélectionnez le dossier correspondant, tel que Process dans lequel tous les job designs du projet sont stockés. Si vous souhaitez importer des **Business Models**, sélectionnez le dossier : **BusinessProcess**.
- Mais si votre projet regroupe différents types d'éléments (Business Models, Jobs Designs, Métadonnées, Routines...), nous vous conseillons de sélectionner le **Projet** en entier afin de tout importer en une seule fois.
- Cliquez sur **OK** pour continuer.

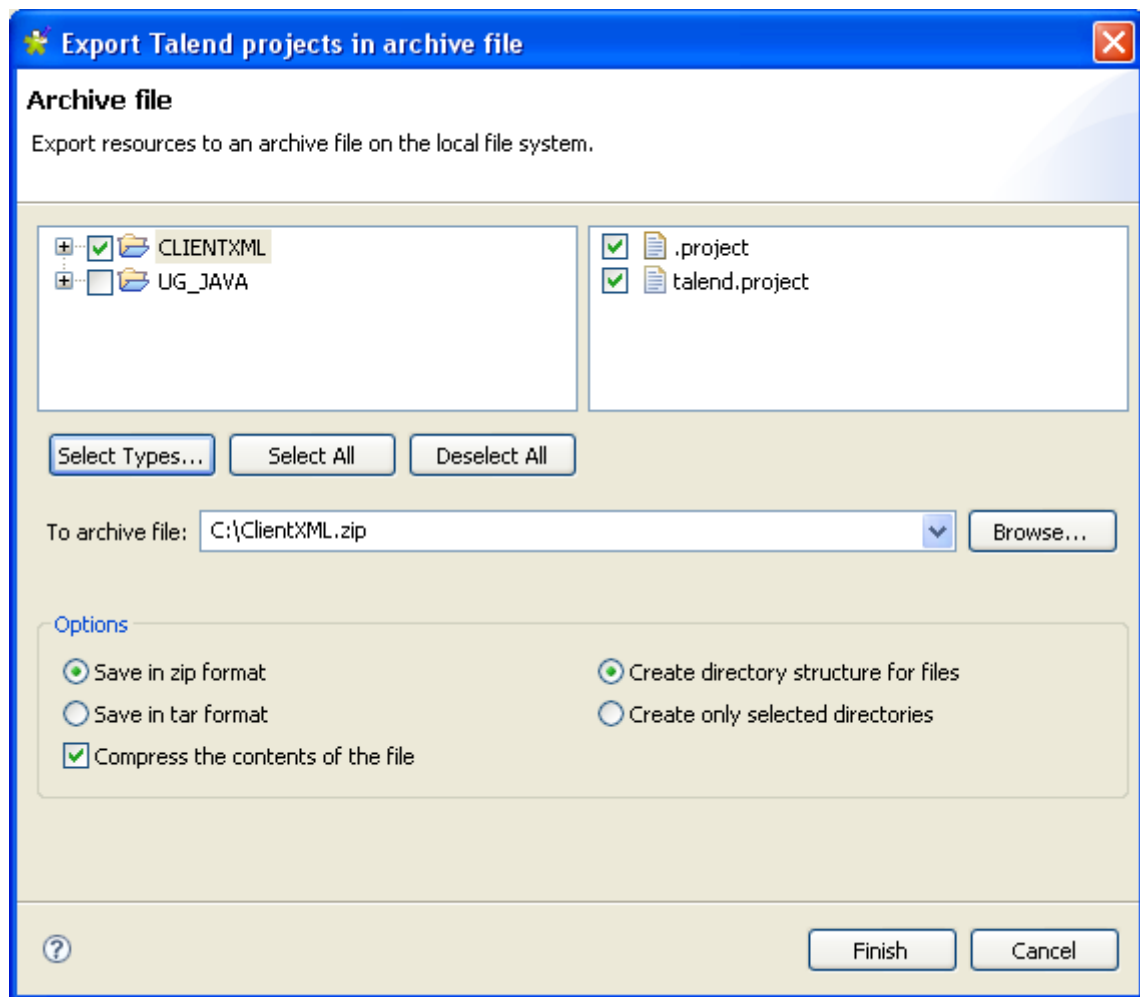


- Tous les éléments valides pouvant être exportés sont affichés dans la liste **Items List**. Par défaut, tous les éléments sont sélectionnés mais vous pouvez les désélectionner tous ou individuellement.
- Cliquez sur **Finish** pour valider l'import.
- Les éléments importés seront sauvegardés dans le dossier correspondant sur le Référentiel en fonction de sa nature.

Exporter des projets

Dans la fenêtre **Talend Open Studio**, la barre d'outils vous permet d'exporter le projet courant.

Cliquez sur le bouton **Export project** de la barre d'outils en haut de la fenêtre principale de **Talend Open Studio**.



La fenêtre d'**Export** s'ouvre sur le répertoire workspace, listant les projets que vous pouvez exporter dans un fichier archive.

- Vous pouvez, en fonction de vos besoins, sélectionner plusieurs projets ou une partie d'un projet à partir du bouton **Select Types** (pour les utilisateurs expérimentés).
- Saisissez un nom pour l'archive ou sélectionnez le fichier archive s'il existe déjà.
- Dans la zone **Option**, sélectionnez le format de compression et le type de structure que vous souhaitez utiliser.

Cliquez sur **Finish** pour valider.

Exporter des job scripts

La fonction **Export Job Scripts** vous permet de déployer et d'exécuter un job à partir de n'importe quel serveur, indépendamment de **Talend Open Studio**.

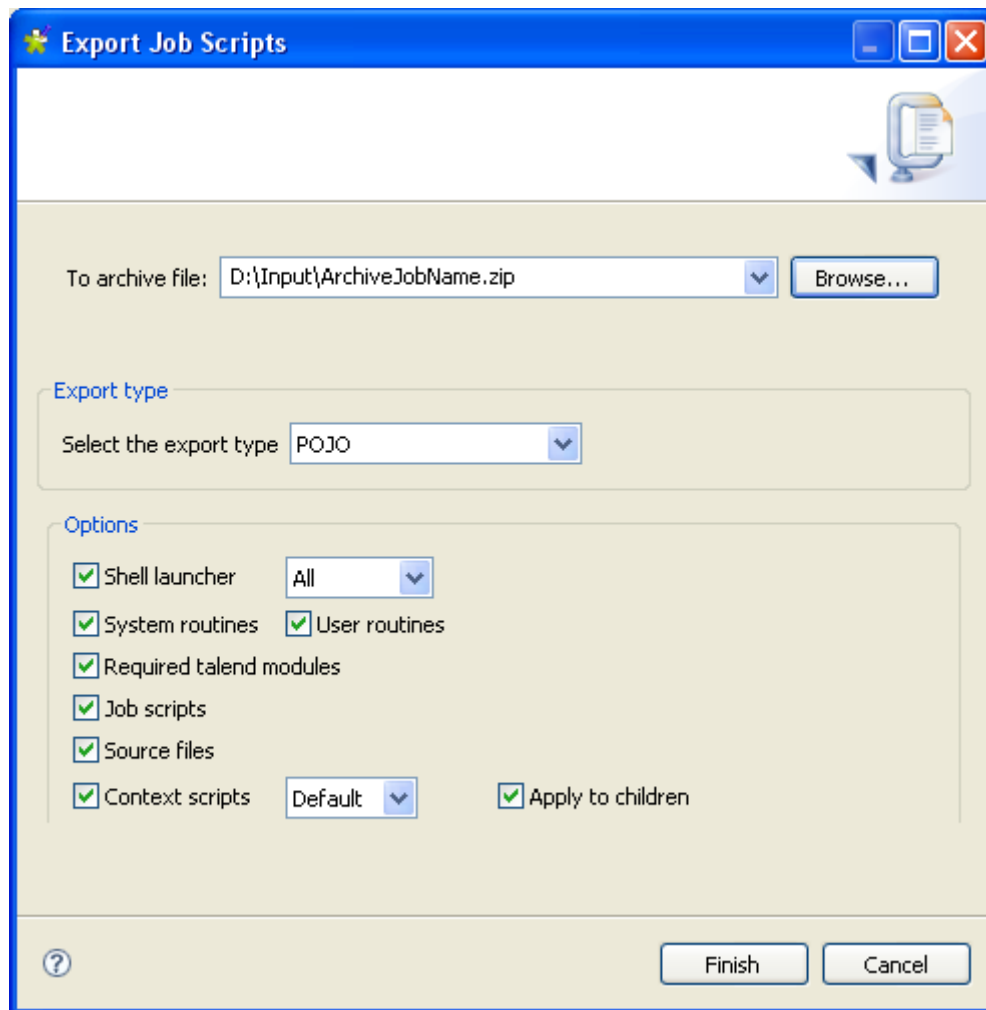
L'export de job scripts ajoute à l'archive tous les fichiers nécessaires à l'exécution d'un job, à savoir les fichiers .bat et .sh, ainsi que tous les fichiers de paramètres de contexte ou fichiers liés.

Pour exporter un job scripts :

- Cliquez droit sur le job correspondant dans le Repository.
- Sélectionnez **Export Job Scripts**.
- Donnez le chemin d'accès au fichier archive.

La boîte de dialogue **Export Job Scripts** n'est pas la même lorsque vous travaillez en Java ou en Perl.

Exporter un job en Java



- Sélectionnez le type d'export dans la liste **Export Type**. Les différents types sont : **POJO**, **Axis Webservice (WAR)** et **Axis Webservice (Zip)**
- Puis sélectionnez le type de fichiers que vous souhaitez ajouter à l'archive. Les différents types de fichiers diffèrent en fonction du type d'export que vous avez choisi.

Cliquez sur **Finish** pour valider.

Exporter un job en tant que POJO

Si vous exportez en POJO (Plain Old Java Object) et vous souhaitez réutiliser un job dans un **Talend Open Studio** installé sur une autre machine, cochez la case **Source files**. Ces fichiers source (.item et .properties) sont uniquement utilisés dans **Talend Open Studio**.

Lorsque que plusieurs contextes sont proposés dans la liste **Context script**, sélectionnez celui qui correspond à vos besoins. Ainsi, les fichiers .bat ou .sh contiendront les paramètres de contexte par défaut. Cependant, la totalité des fichiers de paramètres de contexte seront exportés

et pas uniquement ceux que vous aurez sélectionnés dans la liste. Ensuite, vous pourrez, si vous le souhaitez, éditer les fichiers .bat ou .sh pour modifier manuellement le type du contexte.

Si vous souhaitez modifier le type du contexte, il vous suffit d'éditer les fichiers .bat ou .sh et de modifier le paramètre suivant : `--context=Default` pour le contexte adéquat.

Si vous souhaitez modifier les paramètres de contexte, éditez le fichier **.properties** de contexte.

Exporter un Job en tant que Webservice

Dans la boîte de dialogue **Export Job**, vous pouvez modifier le type d'export afin d'exporter le job sélectionné dans une archive Webservice.



Sélectionnez le type d'archive que vous souhaitez utiliser dans votre application Web.

Type d'archive	Description
WAR	Les options sont en lecture seule. En effet, l'archive WAR générée comprend tous les fichiers de configuration nécessaire à l'exécution ou au déploiement pour une application Web.
ZIP	Toutes les options sont disponibles. Si tous les fichiers de configuration de votre application Web sont configurés, vous avez la possibilité de ne sélectionner que les paramètres de contexte et de n'exporter que les Classes dans l'archive.

Une fois que l'archive est générée, placez le fichier WAR ou le dossier Class du ZIP (ou des fichiers décompressés) dans le répertoire souhaité de votre serveur d'application Web.

L'URL utilisée pour déployer votre job ressemble à la suivante :

`http://localhost:8080/Webappname/services/JobName?method=runJob&args=null`

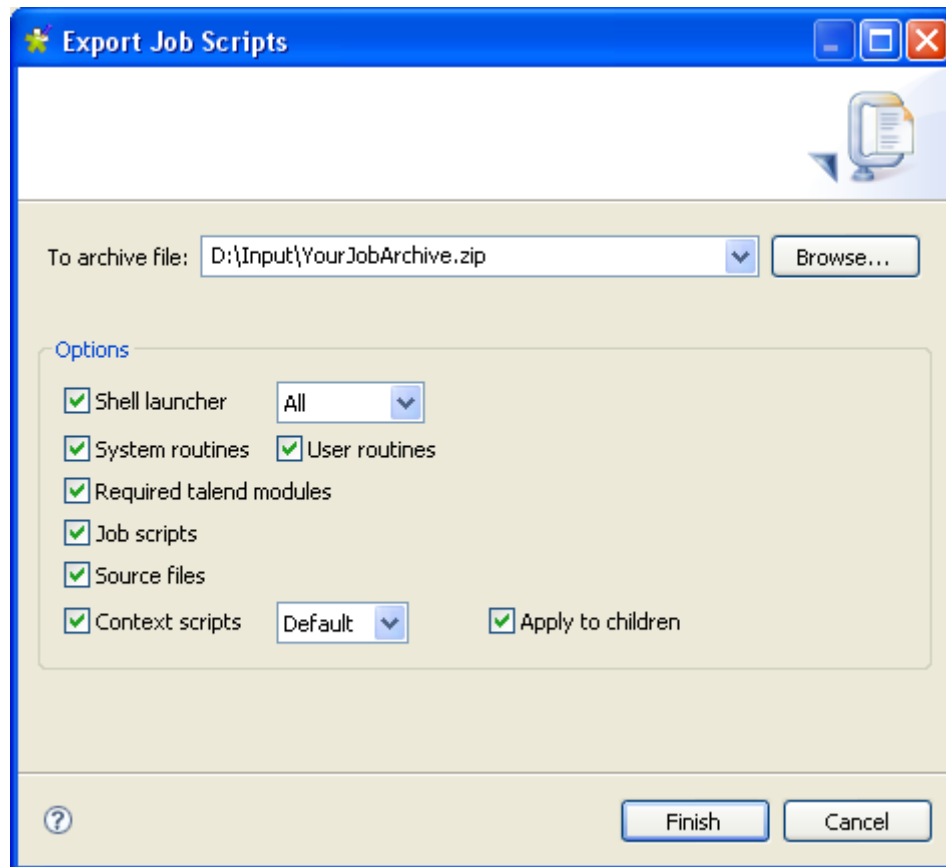
Les paramètres sont les suivants :

Paramètres de l'URL	Description
<code>http://localhost:8080/</code>	Saisissez les host et port de votre application Web.
<code>/NomAppWeb/</code>	Saisissez le nom de votre application Web.
<code>/services/</code>	Saisissez "services" comme terme d'appel standard pour les services Web.
<code>/NomJob</code>	Saisissez le nom exact du job que vous souhaitez exécuter.
<code>?method=runJob&args=null</code>	La méthode utilisée pour exécuter le job est RunJob.

Le retour d'appel de l'application Web s'est déroulé sans erreur (0), en cas d'erreur le chiffre affiché sera différent de 0.

Le composant tBufferOutput a été conçu spécialement pour ce type de déploiement. Pour plus d'informations concernant ce composant, consultez les *Composants Talend Open Studio Guide de référence*.

Exporter un job en Perl



- Sélectionnez le type de fichiers que vous souhaitez ajouter à l'archive.
- Si vous souhaitez réutiliser un job dans un **Talend Open Studio** installé sur une autre machine, assurez-vous d'avoir cocher la case **Source files**. Ces fichiers source (.item et .properties) sont uniquement utilisés dans **Talend Open Studio**.
- Lorsque que plusieurs contextes sont proposés dans la liste **Context script**, sélectionnez celui qui correspond à vos besoins. Ainsi, les fichiers .bat ou .sh contiendront les paramètres de contexte par défaut. Cependant, la totalité des fichiers de paramètres de contexte seront exportés et pas uniquement ceux que vous aurez sélectionnés dans la liste. Ensuite, vous pourrez, si vous le souhaitez, éditer les fichiers .bat ou .sh pour modifier manuellement le type de contexte.

Note: Si vous souhaitez modifier le type de contexte, il vous suffit d'éditer les fichiers .bat ou .sh et de modifier le paramètre suivant : `--context=Prod`.

Note: Si vous souhaitez modifier les paramètres de contexte, éditez le fichier **.properties** de contexte.

Cliquez sur **Finish** pour valider.

WARNING—Vous ne pouvez pas importer de job script d'une version différente de **Talend Open Studio** que celle utilisée lors de la création du job. Pour réutiliser des jobs d'une version antérieure de **Talend Open Studio**, il faut utiliser la fonction *Import Project*. Consultez *Importer des projets*, page 143.

Changer les paramètres de contexte d'un job script

Comme expliqué dans les sections *Exporter un job en Java*, page 151 et *Exporter un job en Perl*, page 153, vous pouvez éditer les paramètres de contexte :

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 ajouter le paramètre dont vous avez besoin parmi ceux du tableau suivant :

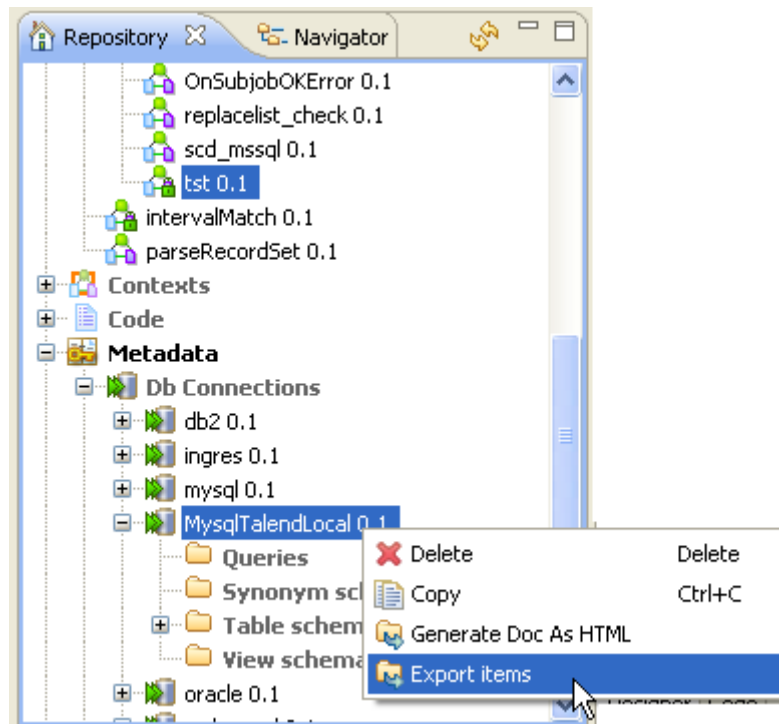
Pour...	Paramètre
Changer la <i>valeur1</i> pour le paramètre <i>key1</i>	<code>--context_param key1=valeur1</code>
Changer <i>valeur1</i> et <i>valeur2</i> des paramètres respectifs <i>key1</i> et <i>key2</i>	<code>--context_param key1=valeur1</code> <code>--context_param key2=valeur2</code>
Changer une valeur contenant des caractères d'espace comme dans les chemins d'accès	<code>--context_param key1="chemin d'accès"</code>

Exporter des éléments

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 Script ou votre Business Model, par exemple.

- Dans le Repository, sélectionnez les éléments que vous souhaitez exporter
- Pour une sélection multiple, maintenez la touche **Ctrl** enfoncé puis sélectionnez les éléments souhaités.

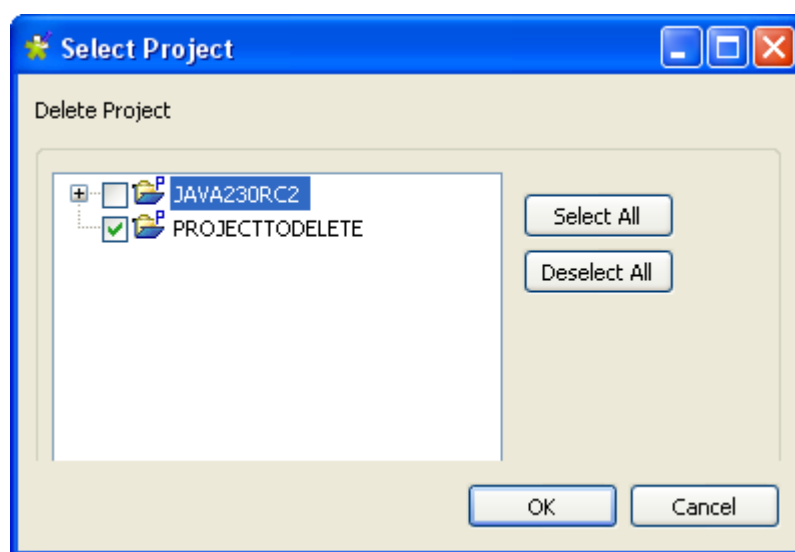


Note: Si vous souhaitez exporter les métadonnées d'une table de base de données, veillez à bien sélectionner la totalité de votre DB connection et pas uniquement la table souhaitée, afin que le processus d'export se déroule correctement.

- Maintenez la touche **Ctrl** enfoncée et cliquez-droit sur l'élément à exporter, puis sélectionnez **Export items** dans le menu contextuel.
- Puis sélectionnez le répertoire dans lequel vous souhaitez enregistrer les éléments exportés. Sinon, définissez le fichier archive dans lequel les fichiers des éléments sélectionnés seront compressés.
- Cliquez sur **Finish**.

Supprimer un projet

Dans la fenêtre de **Login**, cliquez sur **Delete** pour ouvrir la fenêtre de gestion.



- Cochez la case correspondante au projet que vous souhaitez supprimer. Vous pouvez sélectionner plusieurs projets en même temps.
- Cliquez sur **OK** pour valider la suppression.
- La liste des projets est rafraîchie automatiquement.

Note: Faites attention lorsque vous cliquez sur OK, car aucune fenêtre de confirmation n'apparaît. De plus, les projets supprimés ne peuvent pas être récupérés.

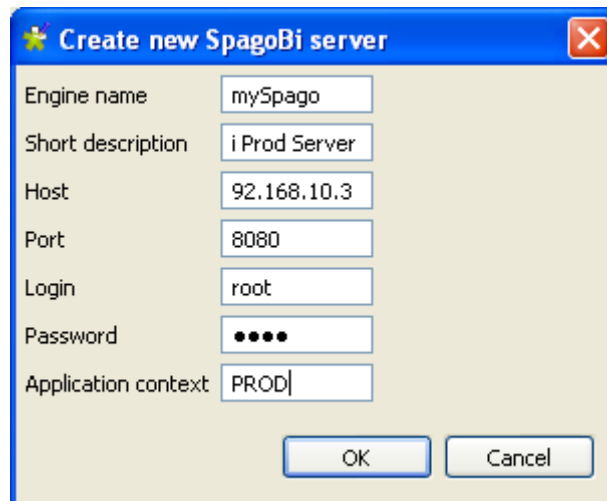
Déployer un job sur un serveur SpagoBI

A partir de l'interface **Talend Open Studio**, vous pouvez facilement déployer vos jobs sur un serveur SpagoBI afin de les exécuter de votre administrateur SpagoBI.

Créer un nouveau serveur SpagoBI

Avant toute chose, renseignez les informations concernant votre serveur SpagoBI simple ou multiple dans **Talend Open Studio**.

- Cliquez sur **Preferences > Talend > SpagoBI servers**
- Cliquez sur **New** pour ajouter un nouveau serveur à la liste.

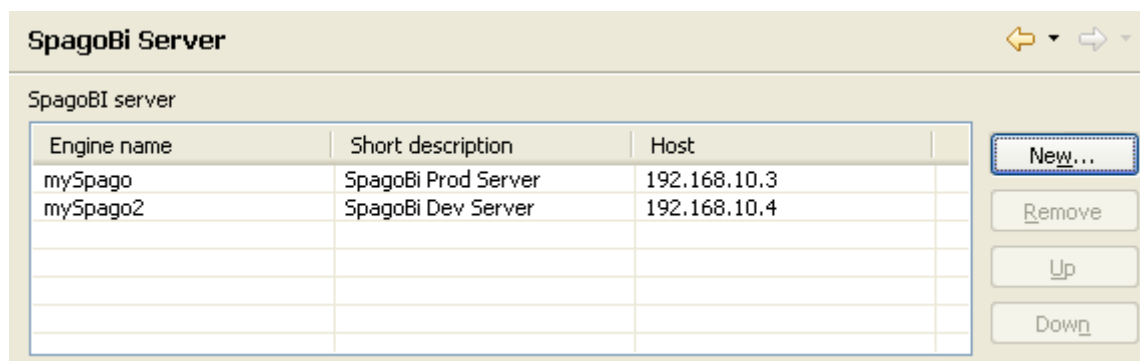


A dialog box titled "Create new SpagoBI server" with a close button (X) in the top right corner. It contains several input fields for server configuration:

Engine name	mySpago
Short description	i Prod Server
Host	92.168.10.3
Port	8080
Login	root
Password	••••
Application context	PROD

At the bottom right are "OK" and "Cancel" buttons.

- Dans le champ **Engine Name**, saisissez le nom interne utilisé dans **Talend Open Studio**. Ce nom n'est pas utilisé dans le code généré.
- Dans le champ **Short description**, saisissez une description du serveur que vous êtes en train de créer.
- Renseignez les informations **Host** et **Port** de votre serveur. Le Host peut correspondre à l'adresse IP ou au nom DNS de votre serveur.
- Saisissez le **Login** et le **Password** de connexion au serveur SpagoBI.
- Dans le champ **Application Context**, saisissez le nom du contexte créé dans **Talend Open Studio**
- Cliquez sur **OK** pour valider ces informations.



A window titled "SpagoBI Server" with a toolbar containing navigation arrows. Below the title bar is a section labeled "SpagoBI server" containing a table and a set of control buttons.

Engine name	Short description	Host
mySpago	SpagoBI Prod Server	192.168.10.3
mySpago2	SpagoBI Dev Server	192.168.10.4

Control buttons on the right: New..., Remove, Up, Down.

La nouvelle entrée est ajoutée au tableau des serveurs disponibles.

Vous pouvez ajouter autant d'entrée SpagoBi que vous le souhaitez.

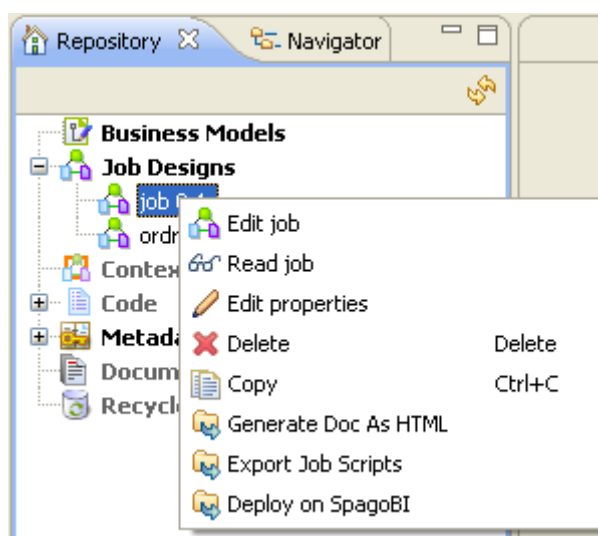
Editer ou supprimer un serveur SpagoBI

Pour supprimer une entrée obsolète, sélectionnez l'entrée dans le tableau, puis cliquez sur le bouton **Remove**.

Puis, si nécessaire, créez une nouvelle entrée contenant les informations mises à jour.

Déployer vos jobs sur le serveur SpagoBI

Suivez les étapes suivantes pour déployer vos jobs sur un serveur SpagoBI :



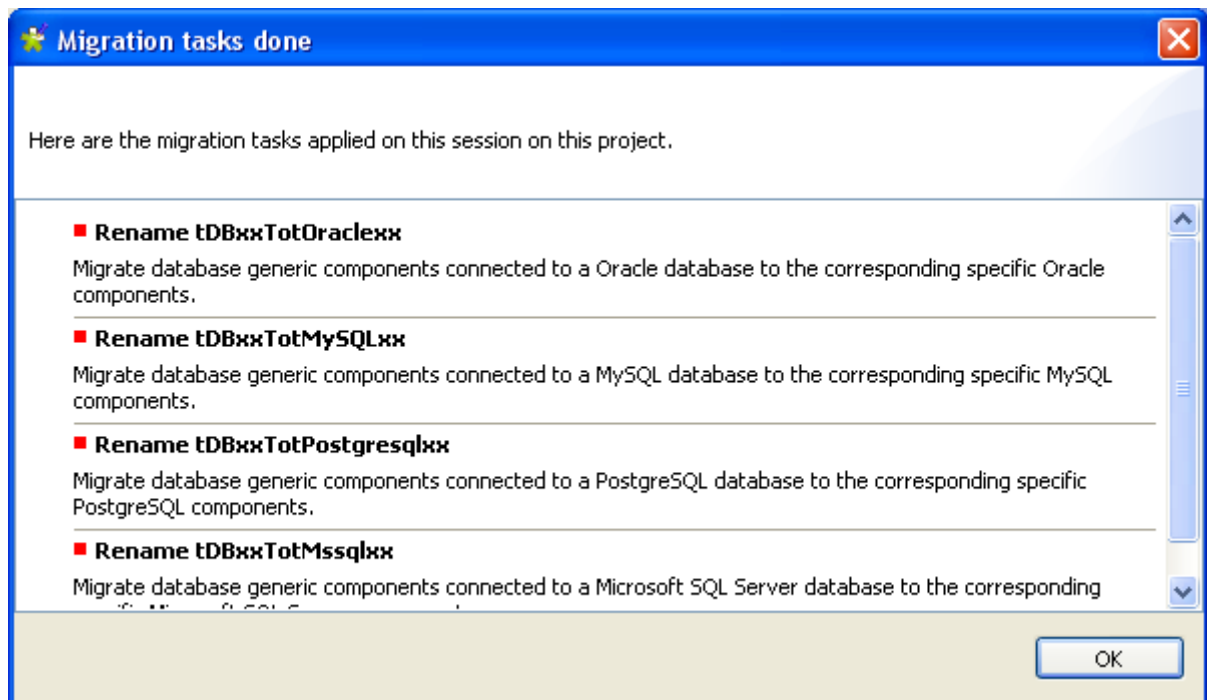
- Dans le Job designer, sélectionnez le job à déployer puis cliquez droit pour faire apparaître le menu contextuel.
- Sélectionnez **Deploy on SpagoBI**.
- Comme pour tout export de job script, sélectionnez le nom (**Name**) du job qui sera exporté et renseignez le champ **To archive file**.
- Sélectionnez le **SpagoBI server** correspondant dans la liste déroulante.
- Les champs **Label**, **Name** et **Description** seront pré-remplis avec les propriétés principales définies lors de la création du job.
- Sélectionnez le contexte adéquat dans la liste.
- Cliquez sur **OK** lorsque toutes les informations sont renseignées.

Les jobs sont maintenant déployés sur le serveur SpagoBI sélectionné. Ouvrez votre administrator SpagoBI pour exécuter vos jobs.

Tâches de migration

Les tâches de migration servent à assurer la compatibilité des projets créés dans une version antérieure de **Talend Open Studio** avec la version courante.

Afin que les modifications soient plus visibles, nous avons décidé de partager ces mises à jour avec vous grâce à une fenêtre d'information.



Parmi les modifications affectant l'utilisation de **Talend Open Studio**, voici quelques exemples :

- tDBInput utilisé avec une base de données MySQL, devient un composant spécifique tDBMysqlInput. Son aspect est ainsi modifié dans les jobs où il est utilisé.
- tUniqRow était basé sur les clés de schéma Input, alors que l'actuel tUniqRow permet à l'utilisateur de sélectionner la colonne où faire la correspondance.

Cette fenêtre d'information s'ouvre lorsque vous lancez le projet que vous avez importé (créé) d'une version antérieure de **Talend Open Studio**.

Elle liste et fournit une courte description des tâches qui ont été réalisées avec succès afin que vous puissiez exécuter vos projets normalement.

A		E	
Activer/Désactiver	111	Edit Schema	53
Advanced settings	55	Elément	
Appearance	34	Importer	145
B		Error Log	15
Basic settings	52	Exporter	
Breakpoint	126	Projet	148
Built-in		Expression Builder	107
Variable	112	Expressions régulières	72
Business Model	12, 25, 26	Extra	130
Créer	26	F	
Ouvrir	26	Fichier Delimited	63
Business Modeler	27	Fichier LDIF	73
Business Models	25	Fichier Positional	68
C		Fichier Regex	71
Clé	95	Fichier XML	75
Clé de hachage	95	Loop limit	78
Clé primaire	95	Filtre, Filter	48
Code	12	Forme	28
Code Viewer	19	G	
Component, Composant	42	Graphical workspace	14
Composant	42	Grid	33
Advanced settings	55	I	
Basic settings	52	Importer	
Externe	22	Eléments	145
Paramétrer	51	Inner join	98
Start	57	Inner Join Reject	98
Connexion		Input/Output Multiple	51
Iterate	49	Iterate	49
Link	51	J	
Lookup	46	Job	
Main	46	Créer	41
Output	46	Exécuter	122, 124
Rejet	46	Ouvrir/Créer	39
Row	46	Job Design	12, 39, 40
Contexte	124	Job Designer	41
Contexts	12	Onglets	45
Corbeille, Recycle bin	37	Job script	
D		Exporter	127
Debug mode	126	Job Settings	
Documentation	13	Extra	130
Duplicates	49		

L	View	56
Langage de génération	Propriétés	14, 15
Link		
Log	R	
Lookup	Recycle bin, Corbeille	14, 37
M	Référentiel, Repository	39
Main properties	Refresh	7
Main row	Reject row	46
Metadata, Métadonnées	Rejets	48
Métadonnées	Relation	29
Schéma DB Connection	Bidirectionnelle	30
Schéma FileDelimited	Directionnelle	30
Schéma FileLDIF	Simple	30
Schéma FilePositional	Repository	8, 11, 25
Schéma FileRegex	Repository, Référentiel	39
Schéma FileXML	Requête	
Mode Debug	SQLBuilder	87
Model	Routine	13
Affectation	Row	46
Commenter	Main	46, 47
Copier	Reject	46
Déplacer	Rulers	33
Enregistrer	Run After (devient OnSubjobOK)	50
Réarranger	Run Before (devient OnSubjobOK)	50
Supprimer	Run Job	15, 122, 124
Modeler	S	
O	Scheduler	17
Objet	Schéma	
Outline	Built-in	54
Output	SQLBuilder	87
P	Start	57
Palette	Statistiques	124
Note	StoreSQLQuery	120
Note attachment	Sync columns	53
Select	T	
Zoom	Table d'affectation	35
Projet	Technical name	8
Exporter	tFlowMeterCatcher	18, 129
Properties	ThenRun (devient OnSubjobOK)	50
Comment	tLogCatcher	18, 129
Main	tMap	51, 95
Rulers & Grid	Traces	126
	Trigger	

OnComponentError	50	V	
OnComponentOK	50	Variable	112
OnSubjobError	50	StoreSQLQuery	120
OnSubjobOK	50	Vues	
Run if	50	Réorganiser	45
tStatCatcher	18, 129	X	
U		Xpath	78
Uniques	49		