



Talend Open Studio **4.X**

Guide Utilisateur

**Intentionally
Blank**

Version 4.1_b

Convient pour l'utilisation de **Talend Open Studio** v4.1.x. Annule et remplace toute version antérieure du Guide Utilisateur.

Copyleft

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

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

Talend Open Studio

Guide Utilisateur i

Préface	xi
Objectif	xi
Public visé	xi
Conventions typographiques	xi
Historique des versions	xi
Remarques et Support	xii

CHAPITRE 1

A propos de l'intégration de données...	1
1.1 Intégration décisionnelle avec Talend Open Studio	2
1.2 Intégration opérationnelle avec Talend Open Studio	2
1.3 Monitoring de l'exécution	3

CHAPITRE 2

Présentation de Talend Open Studio	5
2.1 Concepts importants dans Talend Open Studio	6
2.2 Lancer Talend Open Studio	7
2.2.1 Se connecter à un ou plusieurs référentiels ..	10
Se connecter à un référentiel local	10
2.2.2 Configurer un projet dans le référentiel	11
2.3 Travailler avec différents répertoires workspace	12
2.3.1 Créer un nouveau répertoire workspace	12
2.3.2 Se connecter à un répertoire workspace différent	13
2.4 Travailler avec les projets	14
2.4.1 Créer un projet	14
2.4.2 Importer le projet Démo	16
2.4.3 Importer un projet	17
2.4.4 Ouvrir un projet	18
2.4.5 Supprimer un projet	18
2.4.6 Exporter un projet	19
2.4.7 Tâches de migration	20
2.5 Configurer les préférences de Talend Open Studio	21

2.5.1 Interpréteurs Perl/Java	21
2.5.2 Composants externes ou utilisateur	22
2.5.3 Préférences de langue	23
2.5.4 Préférences d'exécution et de débogage	23
2.5.5 Préférences du Designer	25
2.5.6 Ajouter du code par défaut	26
2.5.7 Préférences de Performance	26
2.5.8 Préférences de la documentation	28
2.5.9 Afficher des caractères spéciaux dans les colonnes de schémas	28
2.5.10 Préférences du SQL Builder	28

2.5.11 Préférences des schémas	29
2.5.12 Préférences des bibliothèques	30
2.5.13 Préférences des conversions de type	31

2.6 Personnaliser les paramètres du projet

2.6.1 Personnaliser la Palette	32
2.6.2 Gérer les versions	34
2.6.3 Gérer les statuts	35
2.6.4 Personnaliser les paramètres du Job	37
2.6.5 Configurer les informations de logs	37
2.6.6 Définir les paramètres de contexte	39
2.6.7 Appliquer les paramètres du projet	40
2.6.8 Définir les paramètres des Statuts	40
2.6.9 Afficher/masquer les paramètres de sécurité	42

CHAPITRE 3

Conception d'un Business Model

3.1 Objectifs	44
3.2 Création ou ouverture d'un Business Model	44
3.2.1 Ouvrir un Business Model	45
3.2.2 Créer un Business Model	45
3.3 Modélisation d'un Business Model	46
3.3.1 Formes (Shapes)	47
3.3.2 Connexions	48
3.3.3 Commenter et réarranger un Business Model ...	50

Ajouter une note ou du texte libre	50
Réarranger une vue métier	50
3.3.4 Propriétés	51
Onglet Appearance	51
Onglet Rulers and Grid	52
Onglet Assignment	53

3.4 Affectation d'éléments du Repository à un Business Model

3.5 Modification d'un Business Model	54
3.5.1 Renommer un Business Model	54
3.5.2 Copier et coller un Business Model	55
3.5.3 Déplacer un Business Model	55
3.5.4 Supprimer un Business Model	55
3.6 Enregistrement d'un Business Model	55

CHAPITRE 4

Conception d'un Job d'intégration de données

4.1 Objectifs	58
4.2 Utiliser un Job Design simple	58
4.2.1 Créer un Job	58
4.2.2 Cliquer & déposer des composants	61
Déposer des composants à partir de la Palette	61
Déposer des composants à partir du répertoire Metadata	62
4.2.3 Chercher des composants dans la Palette	64
4.2.4 Connecter des composants	64
4.2.5 Définition des propriétés d'un composant	65
Onglet Basic Settings	65

Onglet Advanced settings	69	les composants	119
Onglet Dynamic settings	69	Icônes d'erreur et d'avertissement sur les composants	119
Onglet View	71	Icône d'erreur sur un Job	120
Onglet Documentation	72	4.6.4 Ajouter une Note au Job	121
4.2.6 Exécution d'un job	72	4.6.5 Afficher les onglets Code ou Outline de votre Job	121
Exécuter en mode normal	72	Outline	121
Exécuter un Job en mode Java debug	73	Code viewer	122
Exécuter un Job en mode Traces Debug	74	4.6.6 Gérer l'affichage d'un sous-job	122
Configurer les paramètres d'exécution avancés	76	Personnaliser les sous-jobs	123
4.2.7 Personnaliser l'espace de travail	78	Réduire l'affichage des sous-jobs	123
Modifier la disposition et les paramètres de la Palette	78	Supprimer le surlignage d'un sous-job	123
Réorganiser les onglets d'un Job	82	4.6.7 Paramétrage des options dans la vue Job	124
Afficher les onglets/vues de configuration des Jobs	82	Automatiser l'utilisation de statistiques & logs	124
4.3 Utiliser les connexions	83	Fonctions Extra	126
4.3.1 Types de connexions	83	4.6.8 Chercher un Job utilisant un composant particulier	126
Connexion de type Row	84		
Connexion de type Iterate	86		
Connexion de type Trigger	86		
Connexion de type Link	88		
4.3.2 Définir les paramètres des connexions	88		
Paramètres d'une connexion Row	88		
Paramètres d'une connexion Iterate	89		
Paramètres d'une connexion Trigger	90		
4.4 Utiliser le Metadata Manager	90		
4.4.1 Centraliser les métadonnées	91		
4.4.2 Centraliser les Contextes et les Variables	91		
Utiliser les variables dans un Job	91		
Définir les variables à partir de la vue Contexts	92		
Configurer les contextes	96		
Définir les variables à partir de la vue Component	97		
Centraliser les contextes dans le Repository	99		
Appliquer des variables de contexte dans un Job du Repository	101		
Exécuter un Job dans un contexte défini	103		
4.4.3 Utilisation des SQL Templates	104		
4.5 Gestion des Jobs : Utilisation avancée	104		
4.5.1 Mapper des flux de données	105		
4.5.2 Créer une requête à l'aide de SQLBuilder	105		
Comparer les structures de base de données	106		
Construire une requête	107		
Stocker une requête dans le Repository	109		
4.5.3 Télécharger des composants de la communauté	110		
Installer des composants de la communauté à partir de Talend Exchange	110		
Gérer les composants installés	111		
4.5.4 Installer des modules externes	111		
4.5.5 Lancer un Job périodiquement	113		
4.5.6 Utiliser les composants tPrejob et tPostjob	115		
4.6 Gestion des Jobs : Autres fonctions	116		
4.6.1 Partager une connexion à une base de données	117		
4.6.2 Définition du composant Start	118		
4.6.3 Gérer les icônes d'avertissement/d'erreur sur	118		
les composants	119		
Icônes d'erreur et d'avertissement sur les composants	119		
Icône d'erreur sur un Job	120		
4.6.4 Ajouter une Note au Job	121		
4.6.5 Afficher les onglets Code ou Outline de votre Job	121		
Outline	121		
Code viewer	122		
4.6.6 Gérer l'affichage d'un sous-job	122		
Personnaliser les sous-jobs	123		
Réduire l'affichage des sous-jobs	123		
Supprimer le surlignage d'un sous-job	123		
4.6.7 Paramétrage des options dans la vue Job	124		
Automatiser l'utilisation de statistiques & logs	124		
Fonctions Extra	126		
4.6.8 Chercher un Job utilisant un composant particulier	126		
CHAPITRE 5			
Gestion des Jobs d'intégration de données ..	129		
5.1 Activation/Désactivation d'un Job ou d'un sous-job	130		
5.1.1 Désactiver un composant de début (Start) ..	130		
5.1.2 Désactiver un composant intermédiaire ..	130		
5.2 Import/export d'éléments ou de Jobs	131		
5.2.1 Import d'éléments	131		
5.2.2 Export de Job en Java	133		
Exporter un Job en Autonomous Job	135		
Exporter un Job en tant que WebService	136		
Exemple d'export de Job en tant que service Web ..	137		
Exporter un Job en tant que JBoss ESB	140		
Exporter un Job en tant que Petals ESB	141		
5.2.3 Export de Job scripts en Perl	145		
5.2.4 Export d'éléments	147		
5.2.5 Changer les paramètres de contexte d'un Job script	149		
5.3 Gestion des éléments du Repository	150		
5.3.1 Gérer les mises à jour des éléments du Repository	150		
Modifier un élément du Repository	150		
Mettre à jour automatiquement les Jobs impactés ..	151		
Mettre à jour manuellement les Jobs impactés ..	152		
5.4 Recherche d'un Job dans le référentiel	153		
5.5 Gestion des versions d'un Job	154		
5.6 Documentation du Job	155		
5.6.1 Générer la documentation HTML	156		
5.6.2 Mettre la documentation à jour	156		
5.7 Gestion de l'exécution des Jobs	157		
5.7.1 Déploiement d'un Job sur un serveur SpagoBI ..	157		
Créer une nouvelle connexion au serveur SpagoBI ..			

157	7.4.2 Etape 2 : Connexion	200
	7.4.3 Etape 3 : Chargement de la table	202
	7.4.4 Etape 4 : Définition du schéma	203
CHAPITRE 6	7.5 Configurer une connexion SAS	203
Mapping de flux de données	7.5.1 Conditions préalables	203
6.1 Présentation du fonctionnement du tMap	7.5.2 Etape 1 : Propriétés générales	203
6.2 Interface du tMap	7.5.3 Etape 2 : Connexion	203
6.3 Configuration du flux d'entrée du tMap	7.6 Configurer un schéma File Delimited	205
6.3.1 Renseigner les tables Input à l'aide d'un schéma 166	7.6.1 Etape 1 : Propriétés générales	206
Contenu des tables Main et Lookup	7.6.2 Etape 2 : Chargement du fichier	206
Variables	7.6.3 Etape 3 : Définition du schéma	207
6.3.2 Utiliser les jointures explicites	7.6.4 Etape 4 : Schéma final	209
Unique Match	7.7 Configurer un schéma File Positional	210
First Match	7.7.1 Etape 1 : Propriétés générales	210
All Matches	7.7.2 Etape 2 : Connexion et chargement du fichier ..	210
6.3.3 Utiliser la fonction Inner Join	7.7.3 Etape 3 : Affinage du schéma	211
6.3.4 Utiliser la fonction All Rows	7.7.4 Etape 4 : Schéma final	211
6.3.5 Filtrer un flux d'entrée	7.8 Configurer un schéma File Regex	212
6.3.6 Retirer des entrées de la table Input	7.8.1 Etape 1 : Propriétés générales	212
6.4 Mapping de variables	7.8.2 Etape 2 : Chargement du fichier	212
6.4.1 Accéder aux variables globales ou de contexte 173	7.8.3 Etape 3 : Définition du schéma	213
6.4.2 Supprimer des variables	7.8.4 Etape 4 : Schéma final	213
6.5 Utilisation de l'Expression editor	7.9 Configurer un Schéma File XML	214
6.5.1 Accéder à l'Expression editor	7.9.1 Configurer un schéma XML pour un fichier d'entrée (Input)	214
6.5.2 Créer du code à l'aide de l'Expression Builder 175	Etape 1 : Propriétés générales	214
6.6 Paramètres Output	Etape 2 : Sélection du type de schéma (Input)	215
6.6.1 Construire des expressions complexes	Etape 3 : Chargement du fichier Input	216
6.6.2 Utiliser les filtres	Etape 4 : Définition du schéma	217
6.6.3 Utiliser la fonction Rejet	Etape 5 : Schéma final	219
6.6.4 utiliser les Rejets Inner Join	7.9.2 Configurer un schéma XML pour un fichier de sortie (Output)	220
6.6.5 Retirer des entrées de la table Output	Etape 1 : Propriétés générales	220
6.6.6 Gestion des erreurs	Etape 2 : Sélection du type de schéma (Output)	220
6.7 Onglet Schema editor	221	
6.8 Résolution des problèmes de mémoire lors de l'utilisation du tMap	Etape 3 : Définition du fichier Output	222
6.9 Gestion des références	Etape 4 : Définition du schéma	224
	Etape 5 : Schéma final	225
CHAPITRE 7	7.10 Configurer un schéma File Excel	226
Gestion des métadonnées	7.10.1 Etape 1 : Propriétés générales	227
7.1 Objectifs	7.10.2 Etape 2 : Chargement du fichier	227
7.2 Configurer une connexion DB	7.10.3 Etape 3 : Affinage du schéma	228
7.2.1 Etape 1 : Propriétés générales	7.10.4 Etape 4 : Schéma final	229
7.2.2 Etape 2 : Connexion	7.11 Configurer un schéma File LDIF	230
7.2.3 Etape 3 : Chargement de la table	7.11.1 Etape 1 : Propriétés générales	230
7.2.4 Etape 4 : Définition du schéma	7.11.2 Etape 2 : Chargement du fichier	230
7.3 Configurer une connexion FTP	7.11.3 Etape 3 : Définition du schéma	231
7.3.1 Etape 1 : Propriétés générales	7.11.4 Etape 4 : Schéma final	232
7.3.2 Etape 2 : Connexion	7.12 Configurer un schéma LDAP	232
7.4 Configurer une connexion JDBC.....	7.12.1 Etape 1 : Propriétés générales	233
7.4.1 Etape 1 : Propriétés générales	7.12.2 Etape 2 : Connexion au serveur	233
	7.12.3 Etape 3 : Authentification et acquisition du DN	234
	7.12.4 Etape 4 : Définition du schéma	235
	7.12.5 Etape 5 : Schéma final	236
	7.13 Configurer un schéma Salesforce	237

7.13.1 Etape 1 : Propriétés générales	238	A.2.2 Barre d'outils de Talend Open Studio	293
7.13.2 Etape 2 : Connexion au compte Salesforce	238	A.3 Repository	294
7.13.3 Etape 3 : Affinage du schéma	238	A.4 Espace de modélisation graphique	296
7.13.4 Etape 4 : Schéma final	239	A.5 Palette	296
7.14 Configurer un schéma générique	241	A.6 Onglets de configuration	297
7.14.1 Etape 1 : Propriétés générales	241	A.7 Panneau des onglets outline et code	300
7.14.2 Etape 2 : Définition du schéma	241	A.8 Raccourcis clavier	300
7.15 Configurer un schéma Web Service	242		
7.15.1 Configurer un schéma simple	242		
Etape 1 : Propriétés générales	242		
Etape 2 : URI et définition des méthodes	243		
Etape 3 : Schéma final	244		
7.16 Configurer une connexion MDM	245		
7.16.1 Etape 1 : Définir la connexion	246		
7.16.2 Etape 2 : Définir un schéma MDM	248		
Définir un schéma d'entrée MDM	248		
Définir un schéma de sortie MDM	253		
Définir un schéma de réception MDM	258		
7.17 Exporter une métadonnée en tant que contexte	262		

CHAPITRE 8

Gestion des routines	265
8.1 Définition des routines	266
8.2 Accès aux routines système	266
8.3 Personnalisation des routines système	267
8.4 Gestion des routines personnalisées	268
8.4.1 Créer des routines personnalisées	269
8.4.2 Editer des routines	271
8.4.3 Editer les bibliothèques des routines utilisateur	
271	
8.5 Appel d'une routine à partir d'un Job	272
8.6 Cas d'utilisation : créer un fichier à la date du jour	273

CHAPITRE 9	
SQL Templates	275
9.1 Qu'est-ce que l'ELT ?	276
9.2 Définition des SQL Templates Talend	276
9.3 Gérer les SQL Templates Talend	277
9.3.1 Les types de SQL Templates système	277
9.3.2 Accéder aux SQL Templates système	278
9.3.3 Créer des SQL Templates personnalisés	280
9.3.4 Exemple d'utilisation des SQL Templates	
système	282

ANNEXE A	
Interface de Talend Open Studio	289
A.1 Fenêtre principale	290
A.2 Barre d'outils et Menus	291
A.2.1 Barre de menu de Talend Open Studio ...	291

A.2.2 Barre d'outils de Talend Open Studio	293
A.3 Repository	294
A.4 Espace de modélisation graphique	296
A.5 Palette	296
A.6 Onglets de configuration	297
A.7 Panneau des onglets outline et code	300
A.8 Raccourcis clavier	300

ANNEXE B

De la théorie à la pratique, exemple de Job . 303

B.1 Présentation du scénario	304
B.1.1 Données en entrée	304
B.1.2 Données en sortie	304
B.1.3 Données de référence	305
B.2 Du scénario au Job	305
B.2.1 Etape 1 : Création du Job, définition des données d'entrée, lecture du fichier	305
B.2.2 Etape 2 : Mapping et transformations	308
B.2.3 Etape 3 : Définition du fichier de référence, mapping des données de référence, sélection du mode Inner Join	310
B.2.4 Etape 4 : Sortie vers une table MySQL	312

ANNEXE C

Règles d'écriture des SQL Templates	315
C.1 Instructions SQL	316
C.2 Lignes de commentaire	316
C.3 La syntaxe <%...%>	316
C.4 La syntaxe <%=...%>	317
C.5 La syntaxe </.../>	317
C.6 Le code pour accéder aux éléments du schéma des composants	318
C.7 Le code pour accéder aux propriétés de la matrice du composant	319

ANNEXE D

Routines système	321
D.1 Routines de type Numeric	322
D.1.1 Exemple de création séquentielle	322
D.1.2 Exemple de conversion décimale implicite	322
D.2 Routines de type Relational	323
D.3 Routines de type StringHandling	323
D.3.1 Exemple de vérification de tri alphabétique	325
D.3.2 Exemple de vérification de type alphabétique .	
325	
D.3.3 Exemple de remplacement de chaîne	325
D.3.4 Exemple de vérification de chaîne	325
D.3.5 Exemple de calcul de longueur de chaîne ..	326
D.3.6 Exemple de nettoyage d'espaces inutiles ...	326
D.4 Routines de type TalendDataGenerator	326
D.4.1 Exemple de génération de données factices	

D.5 Routines de type TalendDate	328
D.5.1 Exemple de formatage d'une date	329
D.5.2 Exemple de vérification d'une date	329
D.5.3 Exemple de comparaison de dates	330
D.5.4 Exemple de configuration de date	330
D.5.5 Exemple de parsage de date	330
D.5.6 Exemple de formatage de la date courante	331
D.6 Routines de type TalendString	331
D.6.1 Exemple de formatage XML d'une chaîne	332
D.6.2 Exemple de trimming d'une chaîne	332
D.6.3 Exemple de désaccentuation d'une chaîne	332

Préface

Objectif

Le présent Guide Utilisateur explique comment utiliser les fonctions de **Talend Open Studio** dans un contexte opérationnel normal.

Les informations contenues dans ce document s'appliquent aux versions **4.1.x** de **Talend Open Studio**.

Public visé

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



Dans ce document, l'interface présentée par le biais des captures d'écran peut être légèrement différente de la vôtre.

Conventions typographiques

Ce guide utilise les conventions typographiques suivantes :

- texte en **gras** : boutons et champs d'une fenêtre ou d'une boîte de dialogue, touches du clavier, menus et options des menus ;
- texte en **[gras]** : titre des fenêtres, assistants et boîtes de dialogue ;
- texte en **courier** : paramètres système renseignés par l'utilisateur ;
- texte en *italique* : nom des fichiers, schémas, colonnes, lignes ou variables dont il est question dans les exemples utilisés.
- L'icône introduit un paragraphe apportant une information supplémentaire sur un point important. Elle introduit également les commentaires relatifs à un tableau ou une image.
- L'icône introduit un message apportant une information relative aux modalités d'exécution, ou des recommandations particulières. Elle est également utilisée pour attirer l'attention de l'utilisateur sur certaines informations ou situations particulièrement importantes.

Historique des versions

Le tableau ci-dessous liste les modifications apportées au Guide utilisateur de **Talend Open Studio**.

Version	Date	Historique
v4.0_a	23/04/2010	Les mises à jour du Guide Utilisateur Talend Open Studio comprennent : -Un nouveau chapitre sur les routines. -Une nouvelle section concernant la documentation autogénérée. -Des modifications dans les sections concernant le CDC. -Des modifications dans la jointure par défaut du tMap . -Des modifications dans la section des préférences de langage.
v4.0_b	28/05/2010	Les mises à jour du Guide Utilisateur Talend Open Studio comprennent : -Un nouveau chapitre : SQL Templates. -Une nouvelle annexe concernant les règles des SQL Templates. -Une nouvelle section concernant l'export de Jobs en tant que Petals ESB.
v4.1.a	05/10/2010	Les mises à jour du Guide Utilisateur Talend Open Studio comprennent : -une mise à jour de l'option Traces . -une mise à jour de la vue Run . -une nouvelle fenêtre de login. -un nouvel assistant MDM. -un nouvel assistant FTP. - une mise à jour de la documentation pour refléter les changements du tMap .
v4.1_b	10/12/2010	Les mises à jour du Guide Utilisateur Talend Open Studio comprennent : -une réorganisation du Chapitre 4 et ajout d'une nouvelle section concernant les paramètres des connexions. -l'ajout de l'Annexe D : Routines système.

Remarques et Support

Votre opinion nous intéresse, 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) disponible à l'adresse :

<http://talendforge.org/forum>



CHAPITRE 1

A propos de l'intégration de données...

Le fait que les systèmes d'information des entreprises deviennent de plus en plus complexes n'est pas vraiment nouveau. Une première raison à cela tient au fait que les entreprises déploient généralement de nouvelles applications tout en maintenant les anciennes solutions ("empilage de couches").

En outre les entreprises sont contraintes d'ouvrir de plus en plus leur systèmes d'information à leur fournisseurs, leurs partenaires et leurs clients.

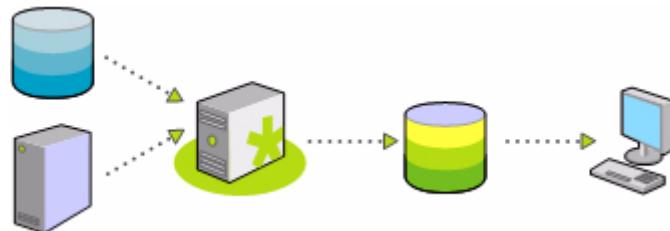
Enfin cette complexification s'explique aussi par la multiplication des formats de stockage de données (fichiers XML, plats positionnels, plats délimités, multi-valués, etc.), des protocoles (FTP, HTTP, SOAP, SCP, etc.) et des technologies des bases de données.

Une question ressort de ces trois constats : Comment réussir son intégration de données lorsque ses données sont éparpillées dans tous ces types de systèmes d'information ? Les fonctions qui se cachent derrière le principe d'intégration données sont variées : 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.).

Talend Open Studio couvre tous ces besoins qu'il s'agisse d'intégration décisionnelle ou opérationnelle.

1.1 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.



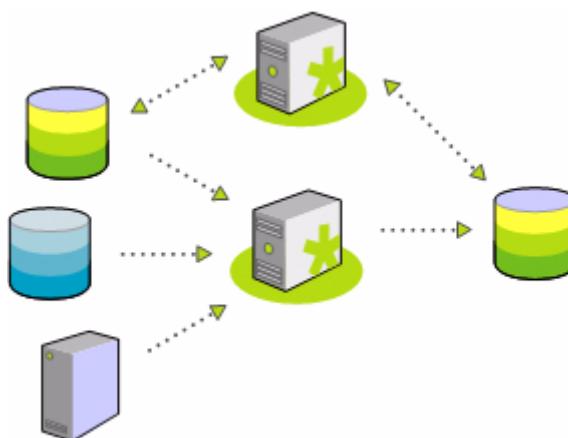
Talend Open Studio offre une très grande connectivité aux :

- 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 le **Guide de Référence des Composants Talend Open Studio**. Pour plus d'informations sur leur fonctionnement dans **Talend Open Studio**, consultez *Conception d'un Job d'intégration de données*, page 57. Pour la modélisation de processus métier, consultez *Conception d'un Business Model*, page 43.

1.2 Intégration opérationnelle avec Talend Open Studio

L'intégration opérationnelle de données est souvent utilisée pour implémenter les programmes et routines habituels, complétée en fonction des 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 pallier 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 le **Guide de Référence des Composants Talend Open Studio**. Pour plus d'informations sur leur fonctionnement dans **Talend Open Studio**, consultez *Conception d'un Job d'intégration de données*, page 57. Pour la modélisation de processus métier, consultez *Conception d'un Business Model*, page 43.

1.3 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 moniterer 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 moniterer l'intégrité des processus d'intégration et de surveiller toute dégradation des performances.

La console **Talend 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, ou disponible en environnement indépendant.

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

A propos de l'intégration de données...

Monitoring de l'exécution



CHAPITRE 2

Présentation de **Talend Open Studio**

Ce chapitre est une introduction à **Talend Open Studio**. Il fournit des informations de configuration de base nécessaire au démarrage de **Talend Open Studio**.

Ce chapitre vous guide dans vos premiers pas dans la création de projets en local à la personnalisation des préférences et de votre espace de travail dans **Talend Open Studio**.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

2.1 Concepts importants dans Talend Open Studio

Lorsque vous travaillez avec **Talend Open Studio**, vous rencontrez les mots : référentiel, projet, répertoire workspace, composant et élément.

Il est important de comprendre l'application de chacun de ses mots pour saisir le fonctionnement de **Talend Open Studio**.

Qu'est-ce qu'un référentiel ? Un référentiel, repository en anglais, est un espace de stockage utilisé par **Talend Open Studio** pour regrouper toutes les données liées aux éléments techniques utilisés soit pour décrire les business models, soit pour créer les Jobs.

Talend Open Studio peut se connecter à autant de référentiel local que nécessaire. Pour plus d'informations, consultez *Se connecter à un ou plusieurs référentiels, page 10*.

Qu'est-ce qu'un projet ? Un projet est un ensemble structuré d'éléments techniques et de leurs métadonnées associées. Ce qui signifie que tous les Jobs et business models que vous créez sont organisés en Projets.

Vous pouvez créer autant de projets que vous le souhaitez dans un référentiel. Pour plus d'informations concernant les projets, consultez *Configurer un projet dans le référentiel, page 11*.

Qu'est-ce qu'un répertoire workspace ? Un répertoire workspace est un répertoire dans lequel sont stockés tous les dossiers de vos projets. Vous devez avoir un répertoire workspace par connexion (connexion au référentiel). **Talend Open Studio** vous permet de vous connecter à différents répertoires workspace, si vous ne souhaitez pas utiliser celui créé par défaut.

Pour plus d'informations concernant le répertoire workspace, consultez *Travailler avec différents répertoires workspace, page 12*.

Qu'est-ce qu'un Job ? Un Job est un concept graphique, composé d'un ou plusieurs composants reliés entre eux. Il vous permet de mettre en place des processus opérationnels de gestion des flux. Il traduit des besoins métier en code, routines et programmes. Les Jobs utilisent les différentes sources et cibles nécessaires aux processus d'intégration de données ou tout autre processus lié.

Pour plus d'informations concernant la création de processus d'intégration de données dans **Talend Open Studio**, consultez *Conception d'un Job d'intégration de données, page 57*.

Qu'est-ce qu'un composant ? Un composant, component en anglais, est un connecteur pré-configuré exécutant une opération d'intégration de données spécifique, quel que soit le type de données que vous souhaitez intégrer (bases de données, applications, fichiers plats, services Web, etc.). Un composant permet d'éviter le plus possible le codage manuel nécessaire au traitement des données provenant de sources multiples et hétérogènes.

Les composants sont regroupés dans des familles en fonction de leur utilisation et apparaissent dans la **Palette de Talend Open Studio**.

Pour plus d'informations concernant les différents types de composants et leurs utilisations, consultez le **Guide de Référence** des **Composants Talend Open Studio**.

Qu'est-ce qu'un élément ? Les éléments, item en anglais, sont les unités techniques constituant un projet. Ces éléments sont regroupés en fonction de leur type : Job Design, Business model, Context, Code, Metadata, etc. Un élément peut comprendre plusieurs autres éléments. Ainsi, les business models et les Jobs que vous créez sont des éléments ; les métadonnées et les contextes que vous utilisez dans les Jobs sont aussi des éléments.

2.2 Lancer Talend Open Studio

Pour ouvrir **Talend Open Studio**, procédez comme suit :

- Dézippez le fichier zip **Talend Open Studio** et, dans le dossier, double-cliquez sur le fichier exécutable correspondant à votre système d'exploitation.
La fenêtre **[License]** s'affiche.
- Lisez et acceptez les termes de la licence afin de passer à l'étape suivante.
Une fenêtre d'enregistrement s'affiche
- Si besoin, suivez les instructions fournies pour rejoindre la communauté **Talend**, ou cliquez sur **Register later** pour passer à l'étape suivante.
- Si vous travaillez avec un proxy, cliquez sur **Network setting** et renseignez les champs **Proxy Host** et **Proxy Port** de la boîte de dialogue.

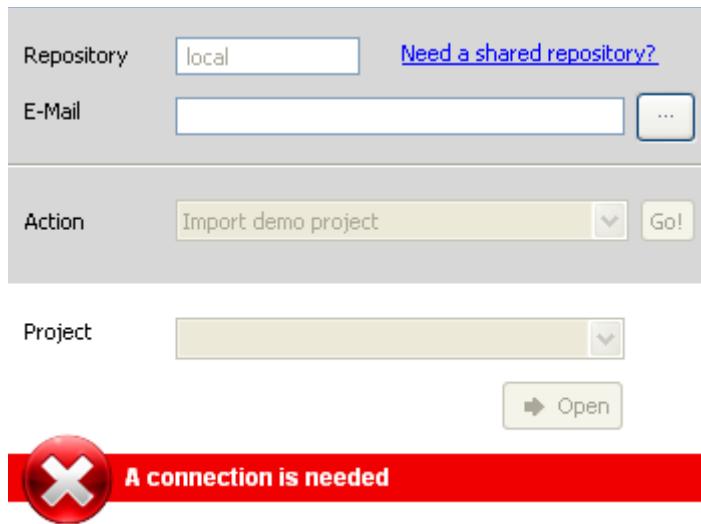


Vous pouvez cliquer sur le lien **You want to know more**, dans la fenêtre d'enregistrement, pour ouvrir une page expliquant les avantages que vous apporte la communauté Talend.



Soyez assurés qu'aucune information personnelle que vous communiquez à **Talend** ne sera transmise à des tiers ni utilisée à d'autres fins que celles de vous informer sur **Talend** ou les produits **Talend**.

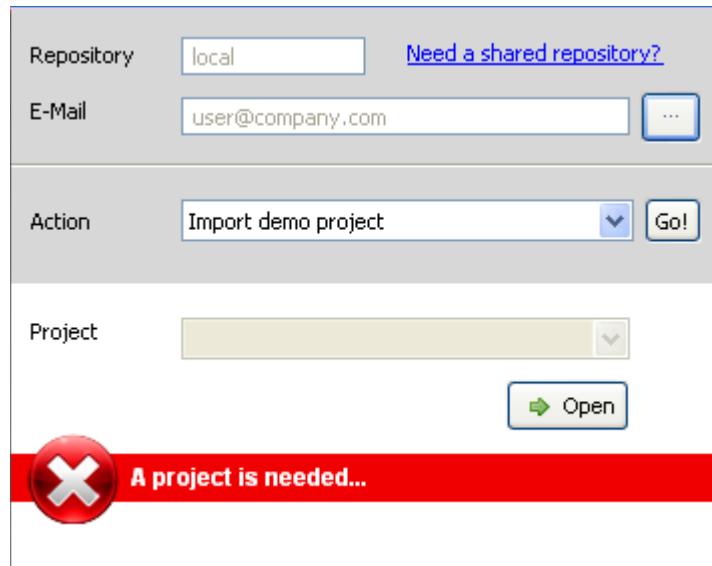
La fenêtre principale de **Talend Open Studio** s'ouvre et vous demande de définir une connexion.



- Cliquez sur le bouton [...] et paramétrez la connexion à un référentiel local.
Ce référentiel va stocker toutes les données et métadonnées liées aux projets et aux éléments du projet que vous créez dans la perspective **Talend Open Studio**. Il stockera également les différentes versions des éléments du projet. Pour plus d'informations concernant la connexion au référentiel, consultez *Se connecter à un ou plusieurs référentiels, page 10* du **Guide Utilisateur Talend Open Studio**

Présentation de Talend Open Studio

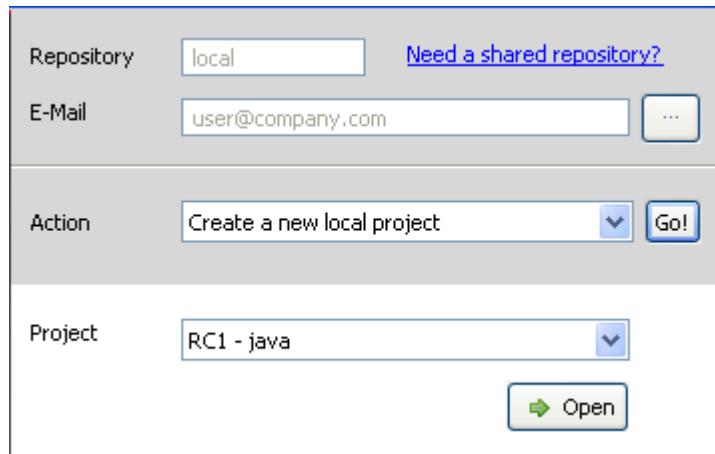
Lancer Talend Open Studio



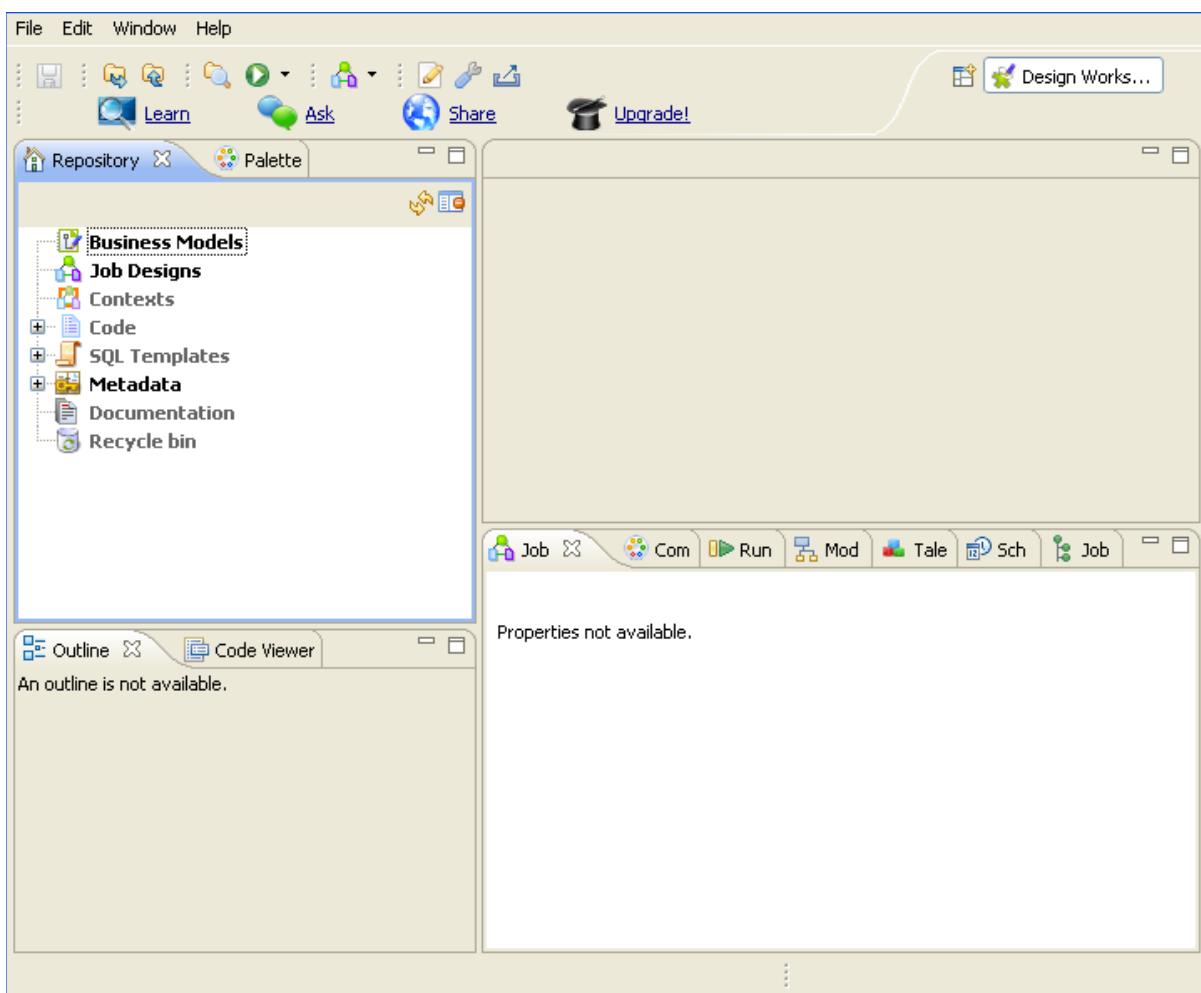
- Dans le champ **Action**, sélectionnez :

Option	Pour...
Import demo project	importer le projet Demo qui comprend de nombreux Jobs prêts à l'emploi. Ce projet Demo peut vous aider à comprendre les fonctionnalités des différents composants Talend. Pour plus d'informations, consultez <i>Importer le projet Démo, page 16.</i>
Create a new local project	créer un projet local dans lequel vous pouvez stocker tous les Jobs et les Business Models créés dans le Studio. Pour plus d'informations, consultez <i>Créer un projet, page 14.</i>
Import existing project(s) as local	importer tout projet que vous possédez localement. Pour plus d'informations, consultez <i>Importer un projet, page 17.</i>
Delete local projects	ouvrir une boîte de dialogue dans laquelle vous pouvez supprimer n'importe quel projet déjà créé ou importé dont vous n'avez plus besoin. Pour plus d'informations, consultez <i>Supprimer un projet, page 18.</i>

- Cliquez sur **Go!** pour passer à l'étape suivante. Celle-ci varie selon l'option sélectionnée. Le but de cette procédure est de créer un nouveau projet local.
- Dans la boîte de dialogue [New project], saisissez un nom pour le projet, sélectionnez un langage de génération (Perl ou Java), puis cliquez sur **Finish** afin de fermer la boîte de dialogue. Le nom du nouveau projet s'affiche dans le champ **Project**.



- Cliquez sur **Open**.
Une barre de progression et une fenêtre de bienvenue s'affichent l'une après l'autre. A partir de cette page, vous avez des liens directs vers la documentation utilisateur, les tutorials, le forum **Talend**, **Talend Exchange** et les dernières nouvelles de **Talend**.
- Cliquez sur **Start now** afin d'ouvrir la fenêtre principale de **Talend Open Studio**. Pour plus d'informations concernant l'ouverture d'un projet, consultez *Ouvrir un projet, page 18*.



2.2.1 Se connecter à un ou plusieurs référentiels

A partir de la fenêtre de login de **Talend Open Studio**, vous pouvez vous connecter à un ou plusieurs référentiels locaux contenant les données de tous vos projets, notamment les Jobs, les Business models, les métadonnées, les routines, etc.

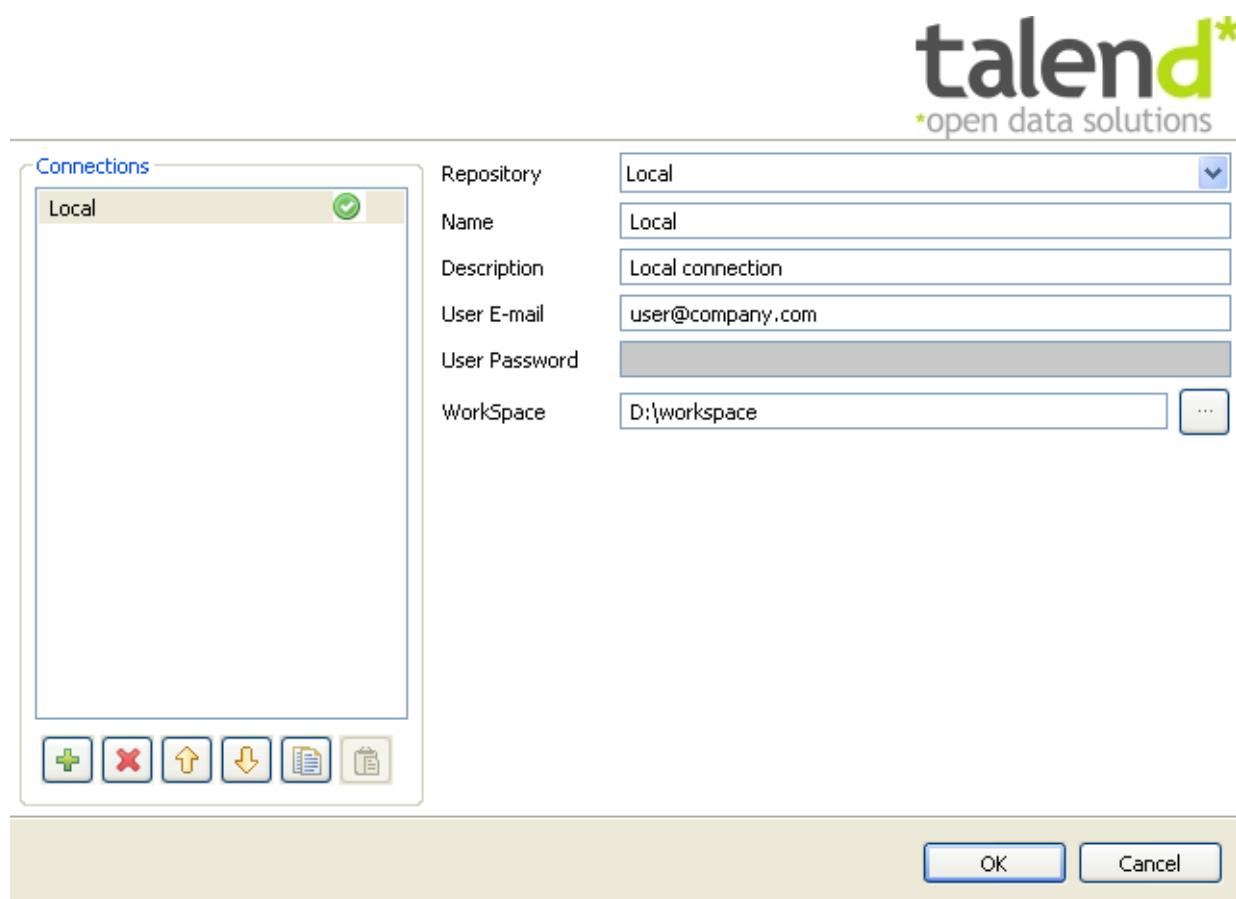
Se connecter à un référentiel local

Pour vous connecter à un référentiel local, procédez comme suit :

- A partir de la fenêtre de login de **Talend Open Studio**, cliquez sur le bouton [...] pour ouvrir une nouvelle vue dans la fenêtre de login et définir les informations de connexion.



Si vous avez déjà défini votre nom d'utilisateur et vos paramètres de connexion, vous pouvez sélectionner l'entrée correspondante à partir de la liste déroulante **Repository**.



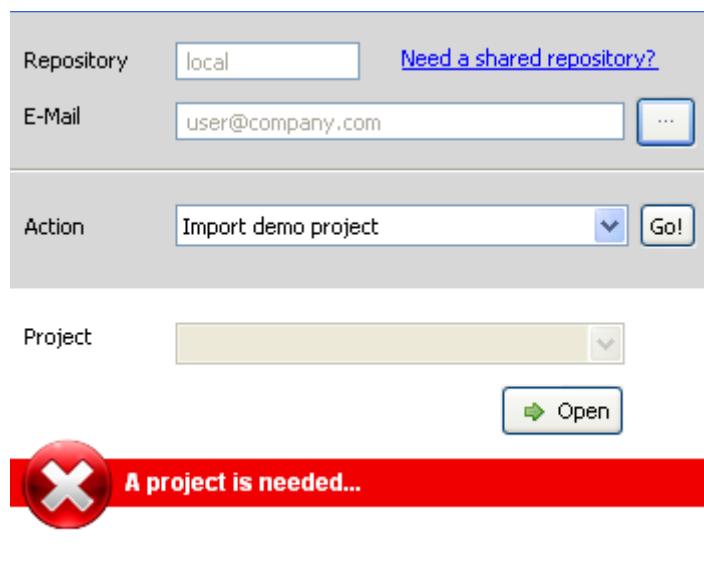
- Entrez si besoin est, dans les champs correspondants, un nom et une description pour votre connexion.
- Dans le champ **User E-mail**, saisissez l'adresse e-mail qui servira d'identifiant utilisateur. Ce champ est obligatoire pour l'utilisation de **Talend Open Studio**. Sachez que l'e-mail ne sera jamais utilisé pour un autre usage que pour la connexion.
- Cliquez sur le bouton [+] dans le coin inférieur gauche de la vue pour ajouter autant de connexions que vous voulez.

- Par défaut, le champ **Workspace** affiche le chemin d'accès à l'espace de travail courant contenant tous les dossiers du projet créé.
- Vous pouvez néanmoins modifier ce chemin d'accès ou indiquer le chemin d'accès d'un autre espace de travail en cliquant sur le bouton [...].



Lorsque vous modifiez le chemin d'accès à l'espace de travail, un message vous demandant de redémarrer le Studio et un bouton **Restart** apparaissent dans la fenêtre de login. Cliquez sur le bouton **Restart** pour redémarrer le Studio et accéder au(x) projet(s) du nouveau workspace indiqué. Pour plus d'informations sur comment vous connecter à un workspace différent, consultez *Travailler avec différents répertoires workspace, page 12*.

- Cliquez sur **OK** pour valider les modifications et fermer la vue.



La fenêtre de login réapparaît et vous invite à choisir une option pour le projet à partir de la liste déroulante **Action**.

Lorsque vous vous connectez pour la première fois au Studio, vous devez créer un nouveau projet. Pour plus d'informations concernant la création d'un projet, consultez *Créer un projet, page 14*.

Pour plus d'informations concernant les autres opérations possibles sur les projets, consultez *Travailler avec les projets, page 14*.

2.2.2 Configurer un projet dans le référentiel

Pour ouvrir **Talend Open Studio**, vous devez d'abord configurer un projet dans le référentiel auquel vous venez de vous connecter.

Pour configurer un projet dans le référentiel, vous pouvez :

- créer un nouveau projet. Pour plus d'informations, consultez *Créer un projet, page 14*.
- importer un ou plusieurs projets locaux que vous avez déjà créé dans une autre instance de **Talend Open Studio**. Pour plus d'informations, consultez *Importer un projet, page 17*.

- importer le projet Démo. Pour plus d'informations, consultez *Importer le projet Démo, page 16.*

2.3 Travailler avec différents répertoires workspace

Dans **Talend Open Studio**, vous pouvez créer plusieurs répertoires workspace et vous connectez à un workspace différent de celui auquel vous êtes actuellement connecté, si nécessaire.

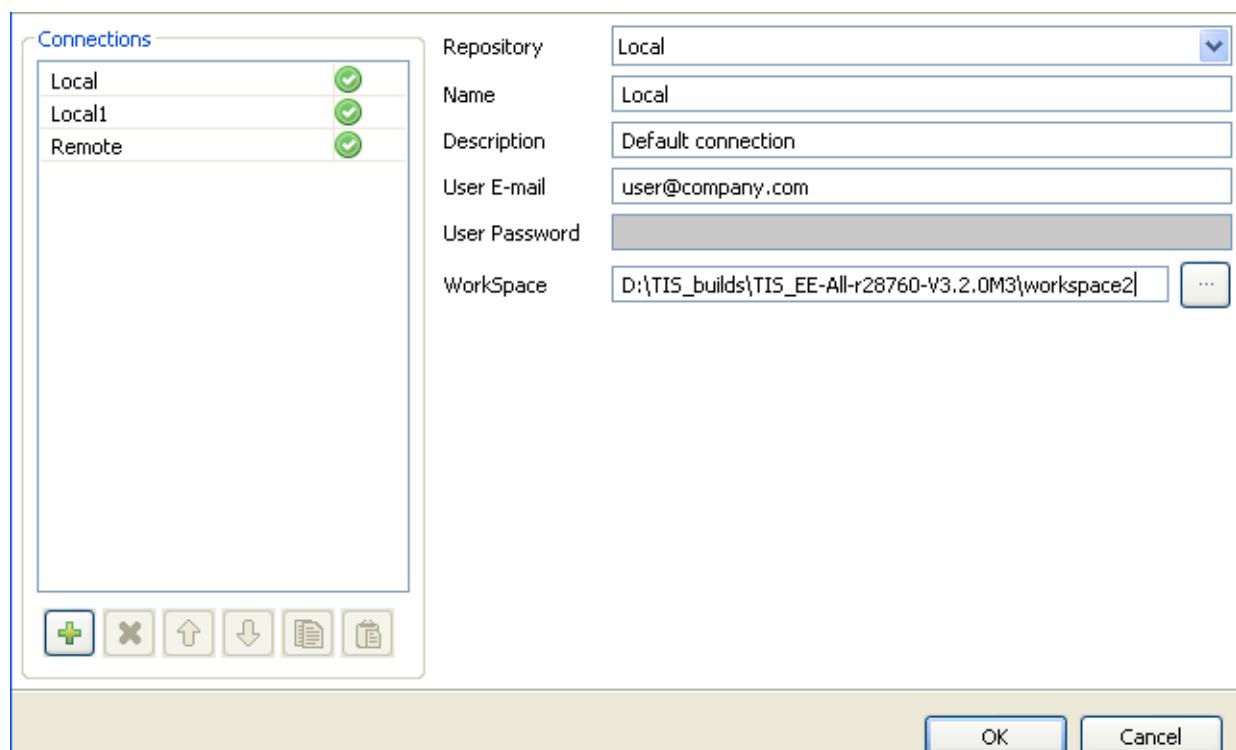
Cette flexibilité vous permet de stocker les répertoires où vous le souhaitez et vous permet de nommer deux ou plusieurs projets de la même manière, du moment que vous stockez les projets dans différents répertoires.

2.3.1 Créez un nouveau répertoire workspace

Talend Open Studio est livré avec un répertoire workspace par défaut. Cependant, vous pouvez créer autant de répertoires que vous le souhaitez et y stocker vos projets en fonction de vos préférences.

Pour créer un nouveau répertoire workspace :

- Dans la fenêtre de login de **Talend Open Studio**, cliquez sur le bouton [...] à côté du champ **Repository** pour ouvrir une autre vue de la fenêtre.



- Dans le champ **WorkSpace**, configurez le chemin d'accès vers ce nouveau répertoire workspace que vous souhaitez créer et cliquez sur **OK** pour fermer la vue. Dans la fenêtre de login, un message d'avertissement s'affiche vous demandant de redémarrer le Studio.
- Cliquez sur **restart** pour redémarrer le Studio.

- Dans la fenêtre de login réinitialisée, configurez un projet pour ce nouveau répertoire workspace.
Pour plus d'informations, consultez *Configurer un projet dans le référentiel, page 11.*
- Sélectionnez le projet dans la liste **Project** et cliquez sur **Open** pour ouvrir **Talend Open Studio**.

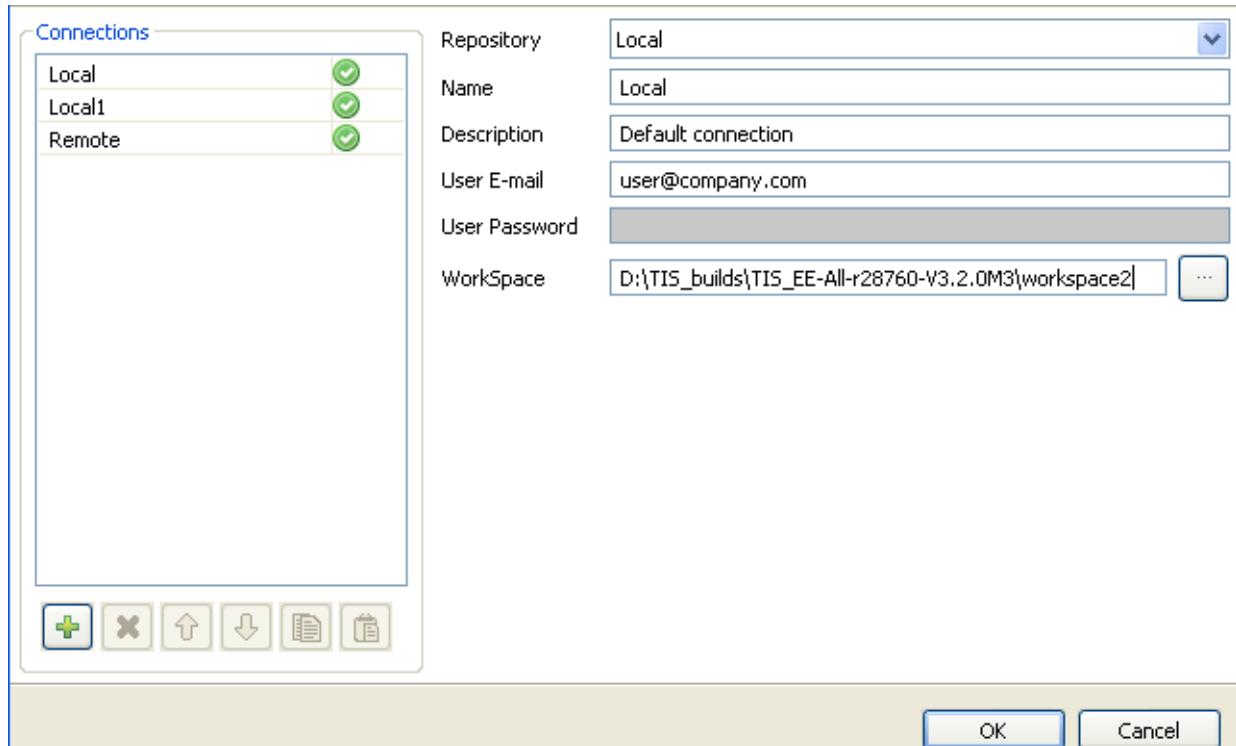
Tous les business models ou Jobs que vous créez dans l'instance courante du Studio seront stockés dans le nouveau répertoire workspace que vous venez de créer. Pour plus d'informations concernant la connexion à différents workspace, consultez la section suivante.

2.3.2 Se connecter à un répertoire workspace différent

Dans la fenêtre de login de **Talend Open Studio**, vous pouvez sélectionner le répertoire workspace dans lequel vous souhaitez stocker vos projets en fonction de vos préférences.

Pour changer de répertoire workspace :

- Dans la fenêtre de login de **Talend Open Studio**, cliquez sur le bouton [...] à côté du champ **Repository** pour ouvrir une nouvelle vue de la fenêtre.



- Cliquez sur le bouton [...] à côté du champ **WorkSpace** pour ouvrir une boîte de dialogue permettant de rechercher le répertoire workspace auquel vous souhaitez vous connectez.
- Sélectionnez le répertoire workspace dans la boîte de dialogue de votre système d'exécution et cliquez sur **OK**.
La boîte de dialogue se ferme et le chemin d'accès au répertoire workspace sélectionné est créé dans le champ **WorkSpace**.
- Cliquez sur **OK** pour fermer la vue.
Dans la fenêtre de login, un message s'affiche vous demandant de redémarrer le Studio.
- Cliquez sur **restart** pour redémarrer le Studio.

- Dans la fenêtre de login réinitialisée, sélectionnez un projet et cliquez sur **Open** pour ouvrir **Talend Open Studio**.

Tous les business models ou Jobs que vous créez dans l'instance courante du Studio seront stockés dans le répertoire workspace sélectionné.

2.4 Travailler avec les projets

Dans **Talend Open Studio**, le “projet” correspond au noeud racine stockant les différents types de Jobs et Business models d'intégration de données, métadonnées, routines, etc.

A partir de la fenêtre de login de **Talend Open Studio**, vous pouvez :

- importer le projet Demo pour découvrir les fonctionnalités de **Talend Open Studio** grâce à des exemples de Jobs prêts à utiliser. Lorsque vous importez le projet Demo, il est automatiquement installé dans le workspace de votre instance courante du Studio.

Pour plus d'informations, consultez *Importer le projet Démo, page 16*.

- créer un projet local. Lorsque vous vous connectez à **Talend Open Studio** pour la première fois, aucun projet par défaut n'apparaît dans la liste. Vous devez créer un projet local et l'ouvrir dans le Studio pour stocker tous les Jobs et Business models que vous créez. Lorsque vous créez un nouveau projet, une arborescence est automatiquement créée dans le répertoire workspace du serveur de votre repository. Il correspond à la vue **Repository** de la fenêtre principale de **Talend Open Studio**.

Pour plus d'informations, consultez *Créer un projet, page 14*.

- Si vous avez déjà créé un projet dans une version antérieure de **Talend Open Studio**, vous pouvez l'importer dans le workspace courant de **Talend Open Studio** via l'option **Import existing project(s) as local**.

Pour plus d'informations, consultez *Importer un projet, page 17*.

- ouvrir un projet que vous avez créé ou importé dans le Studio.

Pour plus d'informations, consultez *Ouvrir un projet, page 18*.

- supprimer des projets locaux que vous avez déjà créés ou importés et dont vous n'avez plus besoin.

Pour plus d'informations, consultez *Supprimer un projet, page 18*.

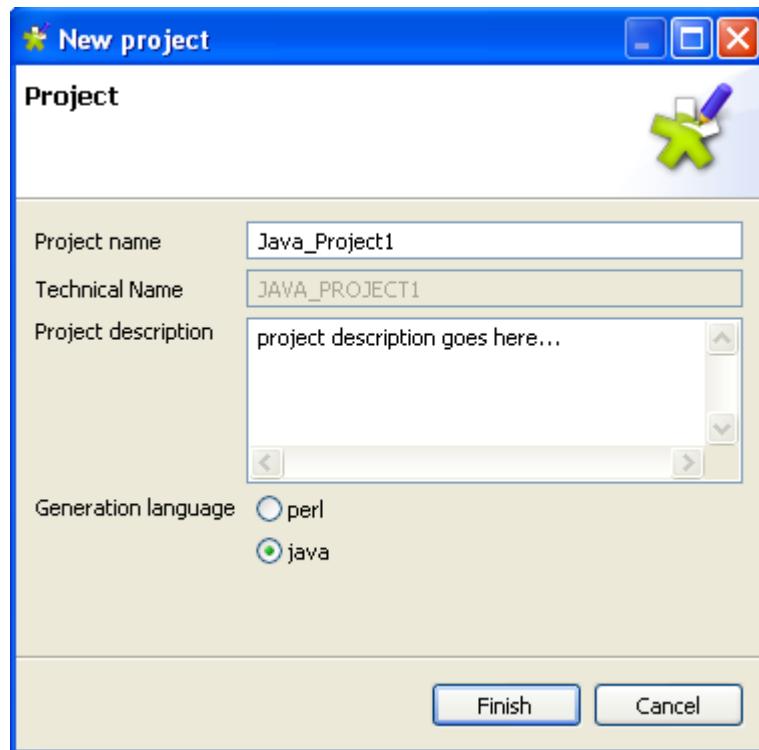
Une fois que vous avez lancé **Talend Open Studio**, vous pouvez exporter les ressources d'un ou de plusieurs projets créés à partir de l'instance courante du Studio. Pour plus d'informations, consultez *Exporter un projet, page 19*.

2.4.1 Créer un projet

Lorsque vous vous connectez pour la première fois au référentiel local du Studio courant, aucun projet par défaut n'apparaît dans la liste. Vous devez créer un projet qui contiendra les Jobs et Business Models d'intégration de données que vous avez créés à partir de l'instance courante du Studio.

Pour créer un projet :

- Lancez **Talend Open Studio** et connectez-vous au référentiel local.
- Dans la liste **Action** de la zone **Project** de la fenêtre de login du Studio, sélectionnez **Create a new local project** et cliquez sur **Go**.
La boîte de dialogue **[New project]** apparaît.

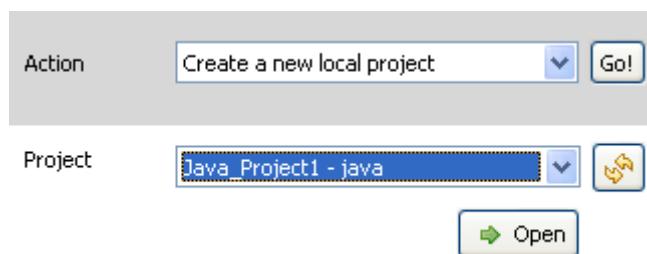


- Dans le champ **Project name**, saisissez le nom du nouveau projet. Ce champ est obligatoire. Un message apparaît en haut de l'assistant en fonction de l'endroit où se trouve votre curseur pour vous informer de la nature de l'information à renseigner, ainsi que des caractères interdits.



Le champ **Technical Name** est en lecture seule et contient le nom technique du projet. Il est utilisé par l'application pour créer le nom du fichier correspondant au projet. Il correspond généralement au nom du projet saisi, mis en majuscule et concaténé à l'aide de tirets soulignés si nécessaire.

- Dans le champ **Generation language**, sélectionnez le langage de génération de code à utiliser entre **Perl** et **Java**. Lorsque vous créerez des Jobs et que vous travaillerez avec les différents composants **Talend** d'intégration de données, vous devrez utiliser le code Perl dans les projets créés en Perl et le code Java dans les projets créés en Java.
- Cliquez sur **Finish**. Le nom de nouveau projet créé apparaît dans la liste **Project** de la fenêtre de login de **Talend Open Studio**.



Pour ouvrir le nouveau projet créé dans **Talend Open Studio**, sélectionnez-le dans la liste **Project** et cliquez sur **Open**. Une fenêtre montrant la progression de l'initialisation du générateur apparaît. Attendez que l'initialisation se termine.

Dans le Studio, si vous souhaitez passer d'un projet à un autre, cliquez sur **File > Switch Projects** dans la barre de menu.



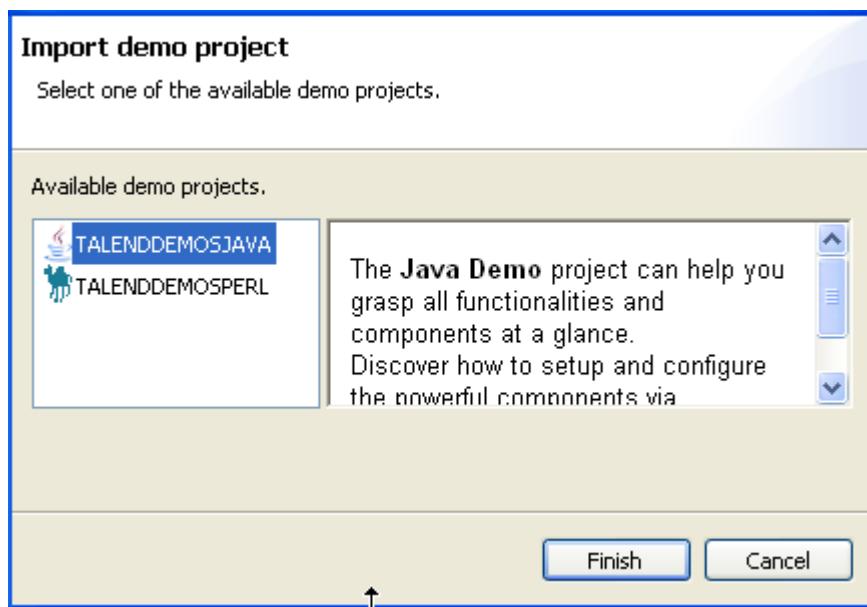
*Nous vous déconseillons d'avoir de nombreux projets dans des langages différents et de passer d'un projet Perl à un projet Java. Si néanmoins vous souhaitez créer des projets dans les deux langages, veillez à bien les distinguer dans le répertoire Workspace, voire à lancer deux instances distinctes de **Talend Open Studio** 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 un projet, page 17*.

2.4.2 Importer le projet Démo

Dans **Talend Open Studio**, vous pouvez importer, dans votre workspace, le projet Démo contenant de nombreux exemples de Jobs prêts à utiliser. Ce projet Démo peut vous aider à comprendre les fonctionnalités des différents composants **Talend**.

- Dans la liste **Project** de la fenêtre de login de **Talend Open Studio**, sélectionnez **Import demo project** et cliquez sur **Go**. La boîte de dialogue [**Import demo project**] apparaît.



- Sélectionnez votre langage de génération favori entre Perl et Java et cliquez sur **Finish** pour fermer la boîte de dialogue.

Un message de confirmation s'affiche vous indiquant que l'import a été réalisé avec succès dans l'instance courante du Studio.

- Cliquez sur **OK** pour fermer ce message de confirmation. Le projet Démo importé apparaît dans la liste **Project** de la fenêtre de login.

Pour ouvrir le projet Démo importé dans **Talend Open Studio**, sélectionnez-le dans la liste **Project** et cliquez sur **Open**. Une fenêtre montrant la progression de l'initialisation du générateur apparaît. Attendez que l'initialisation se termine.

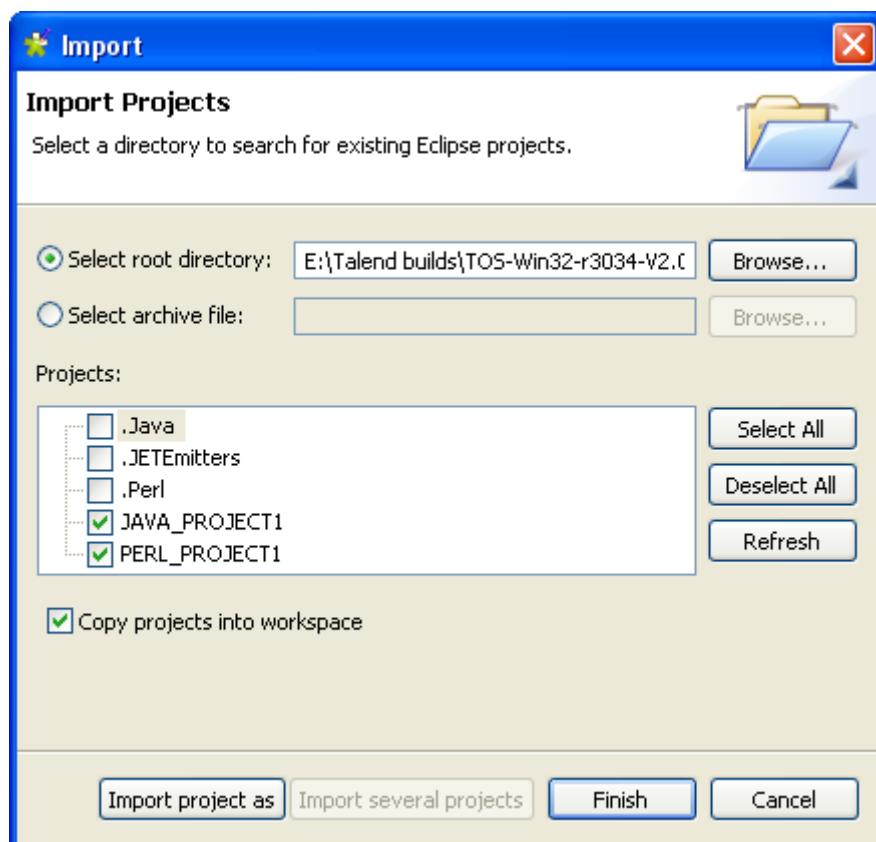
Les exemples de Jobs du projet Démo sont automatiquement importés dans votre workspace et sont disponibles à partir de la vue **Repository** sous le noeud **Job Designs**.

Vous pouvez utiliser ces exemples comme base pour vos propres Job designs.

2.4.3 Importer un projet

Dans **Talend Open Studio**, vous pouvez importer les projets que vous avez déjà créé dans une version antérieure du Studio.

- Dans la liste **Project** de la fenêtre de login de **Talend Open Studio**, sélectionnez **Import existing project(s) as local** et cliquez sur **Go**. L'assistant [**Import**] apparaît.



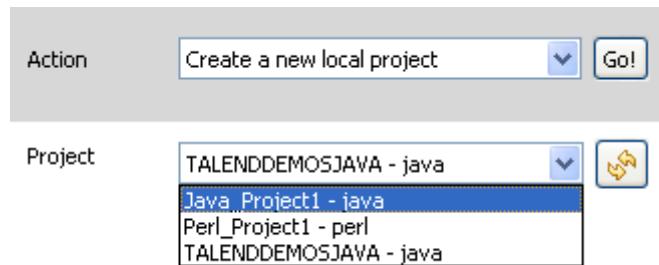
- Cliquez sur **Import several projects** si vous souhaitez importer plus d'un projet à la fois.
- Cliquez sur **Select root directory** ou **Select archive file** en fonction du type de source à partir de laquelle vous souhaitez faire l'import : respectivement une répertoire ou un fichier archive.
- Cliquez sur **Browse...** pour sélectionner le répertoire workspace ou le fichier archive contenant les projets à importer. Par défaut, le workspace sélectionné est le dossier workspace courant de votre Studio. Remontez l'arborescence vers le répertoire workspace de la version antérieure de **Talend Open Studio** ou le fichier archive contenant le ou les projets à importer et sélectionnez-le.
- Cochez la case **Copy projects into workspace** pour faire une copie des projets importés dans le nouveau workspace.



Si vous ne souhaitez pas déplacer les projets de leur emplacement initial vers le nouveau workspace, décochez cette case. Nous vous recommandons de conserver une version de sauvegarde.

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

Dans la fenêtre de login, le ou les nom(s) des projet(s) importé(s) apparaissent désormais dans la liste **Project**.



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



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.

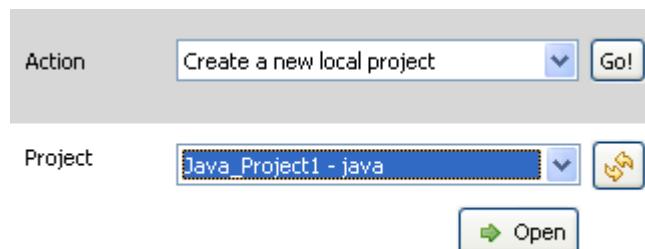
2.4.4 Ouvrir un projet



*Lorsque vous lancez **Talend Open Studio** pour la première fois, aucun projet n'apparaît dans la liste **Project**. Vous devez donc créer ou importer un projet local ou un projet Démo afin d'alimenter la liste **Project** avec des projets que vous pouvez ouvrir dans le Studio.*

Pour ouvrir un projet dans **Talend Open Studio** :

- Dans la fenêtre de login du Studio, cliquez sur le bouton **Refresh** pour mettre à jour la liste des projets, sélectionnez le projet que vous souhaitez ouvrir et cliquez sur **Open**.



Une barre de progression apparaît puis la fenêtre principale de **Talend Open Studio** s'ouvre. Une fenêtre d'initialisation du générateur de code apparaît. Il est recommandé d'attendre que l'initialisation se termine.

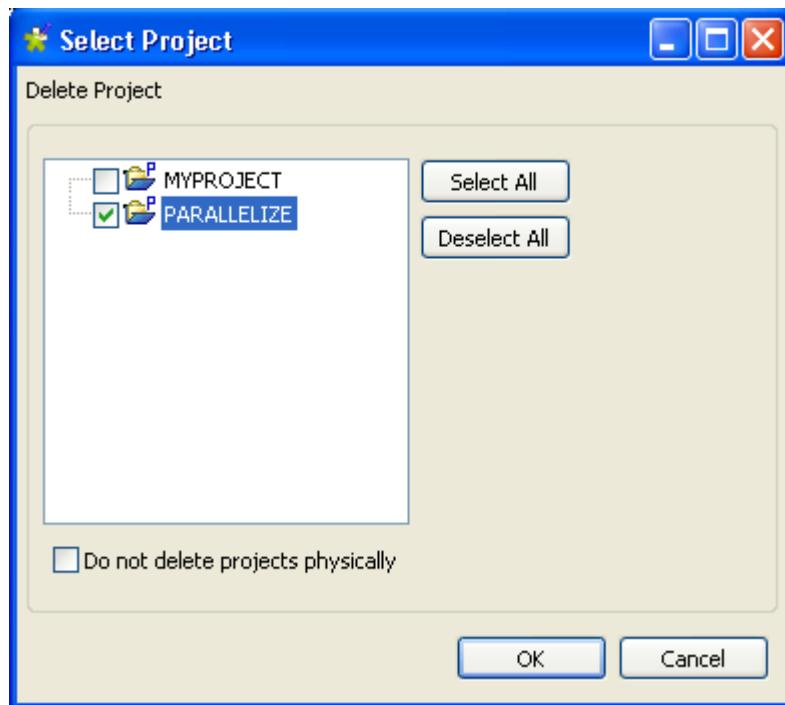


Lorsque vous ouvrez un projet importé d'une version précédente du Studio, une fenêtre d'information s'ouvre et affiche une courte description des tâches de migration réalisées avec succès. Pour plus d'informations, consultez *Tâches de migration, page 20*.

•

2.4.5 Supprimer un projet

- Dans la liste **Project** de la fenêtre de login du Studio, sélectionnez **Delete local project(s)** et cliquez sur **Go** pour ouvrir la boîte de dialogue **[Select Project]**.



- Cochez la ou les cases correspondantes aux projets que vous souhaitez supprimer.
- Cliquez sur **OK** pour valider la suppression.
La liste des projets est rafraîchie automatiquement.



*Faites attention lorsque vous cliquez sur **OK**, car aucune fenêtre de confirmation n'apparaît. De plus, les projets supprimés ne peuvent pas être récupérés.*

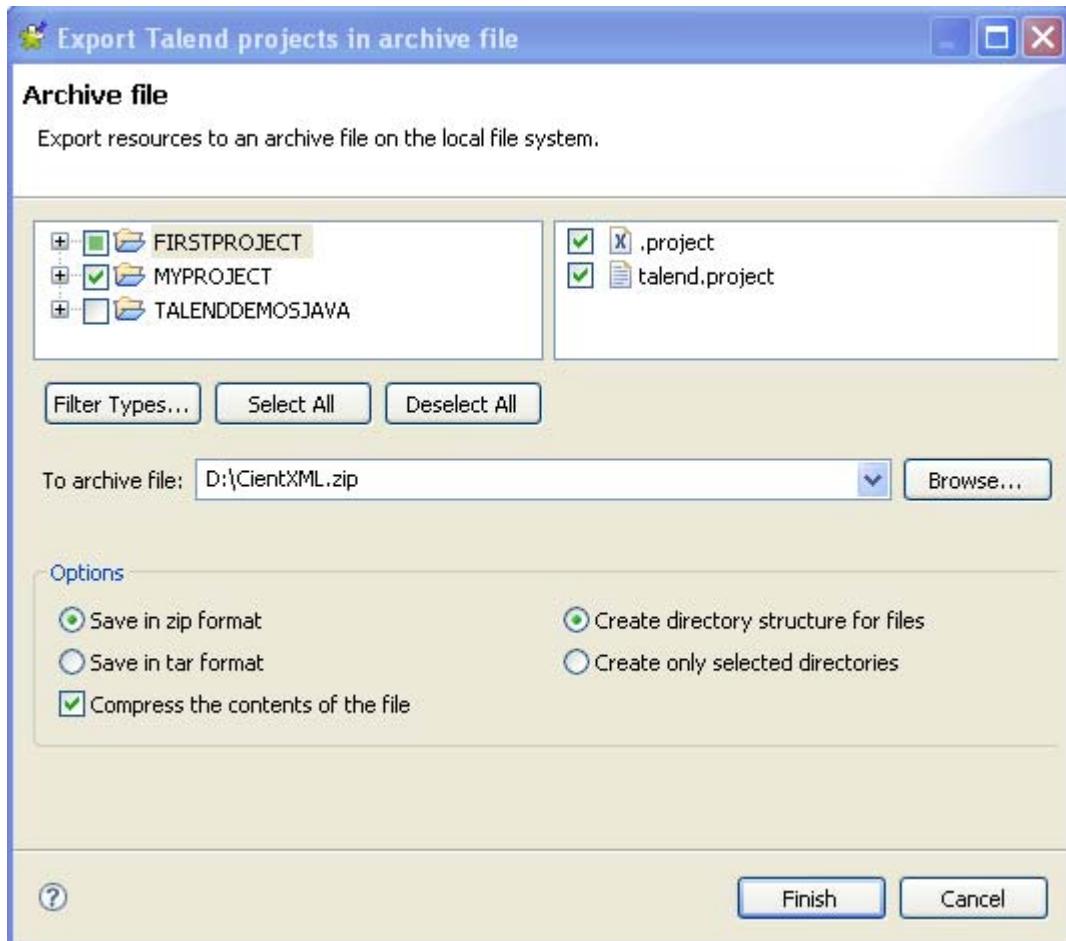


Si vous cochez la case **Do not delete projects physically**, le ou les projet(s) sélectionné(s) seront uniquement supprimés de la liste et seront donc toujours présents dans le répertoire *workspace* de **Talend Open Studio**. De plus, vous pouvez, à tout moment, récupérer un projet supprimé de cette manière via l'option **Import local project(s)** de la liste **Or** de la fenêtre de login.

2.4.6 Exporter un projet

Talend Open Studio vous permet d'exporter les projets créés ou importés dans l'instance courante de **Talend Open Studio**.

- Dans la barre d'outils du Studio, cliquez sur  pour ouvrir l'assistant [**Export Talend projects in archive file**].



- Cochez les cases correspondant aux projets que vous souhaitez exporter. Vous pouvez, en fonction de vos besoins, sélectionner une partie d'un projet à partir du bouton **Filter Types...** (pour les utilisateurs expérimentés).
- Dans le champ **To archive file**, cliquez sur le bouton **Browse...** pour spécifier le répertoire dans lequel exporter votre ou vos projet(s) et saisissez le nom que vous souhaitez donner au fichier archive.
- Dans la zone **Options**, sélectionnez le format de compression et le type de structure que vous souhaitez utiliser.
- Cliquez sur **Finish** pour valider.

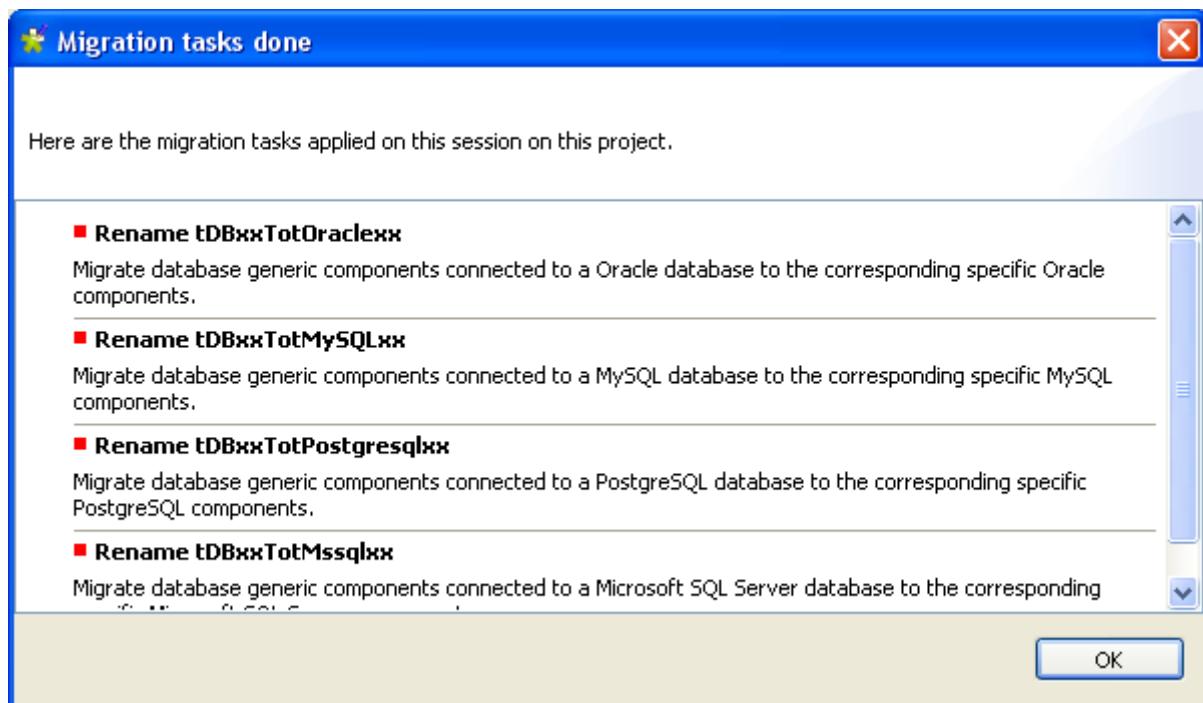
Le fichier archive contenant les projets exportés est créé dans le répertoire spécifié.

2.4.7 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.

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.



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.

2.5 Configurer les préférences 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 et à vos préférences.

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

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

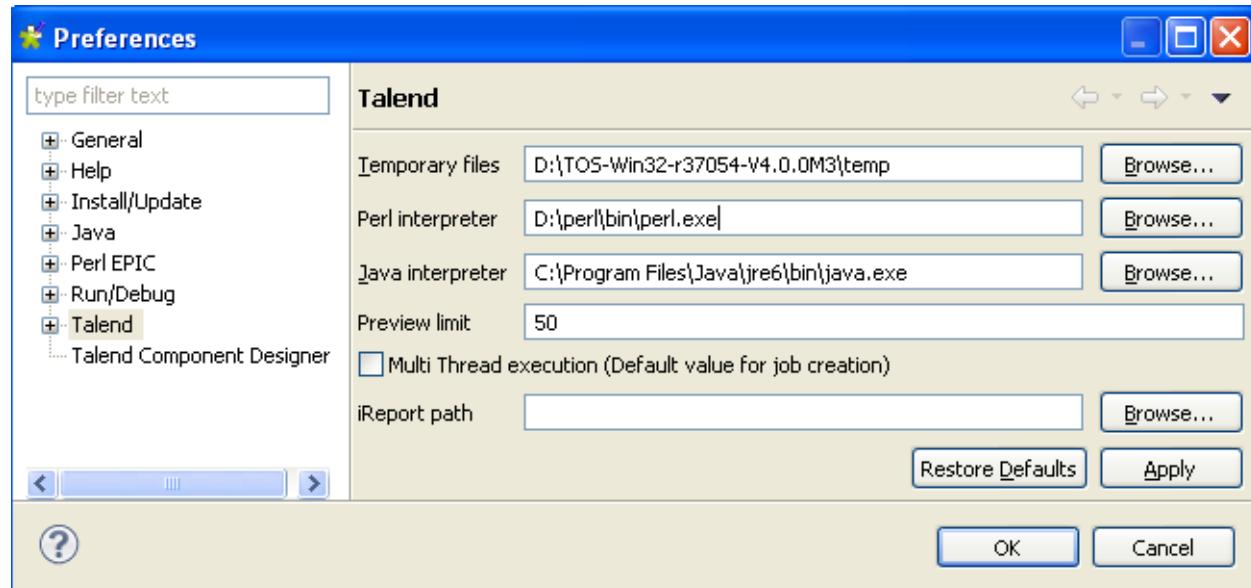
Cliquez sur le menu **Window** de **Talend Open Studio**, puis sélectionnez **Preferences**.

2.5.1 Interpréteurs Perl/Java

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

Présentation de Talend Open Studio

Configurer les préférences de Talend Open Studio



- Si nécessaire, cliquez sur **Talend** dans l’arborescence de la boîte de dialogue [**Preferences**].
- Si le répertoire d’installation Perl/Java par défaut n’est pas correct, rectifiez le chemin d’accès.

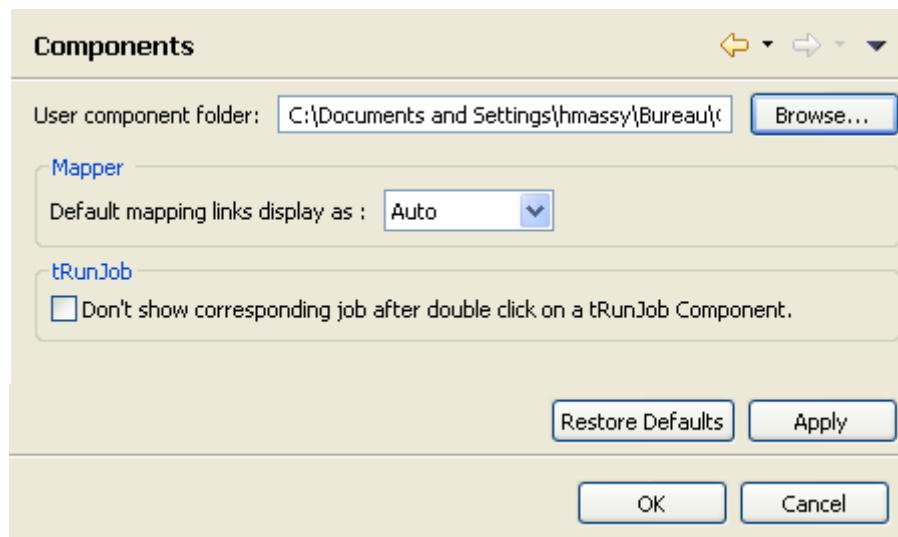
Dans la même vue, vous pouvez également modifier le nombre de lignes de données affichées dans l’aperçu et le chemin d’accès vers les fichiers temporaires.

2.5.2 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 de la boîte de dialogue [**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**.
- Dans le champ **Default mapping links dispaly as**, sélectionnez le type de liens que vous souhaitez utiliser pour la mise en correspondance dans le **tMap**.
- Cochez la case **Don't show corresponding job after double click on a tRunJob component** si vous ne souhaitez pas que le Job fils appelé par le **tRunJob** s'ouvre lorsque vous double-cliquez sur le composant.



Vous pourrez toujours ouvrir le Job correspondant en cliquant-droit sur le composant **tRunJob** et en sélectionnant **Open tRunJob Component**.

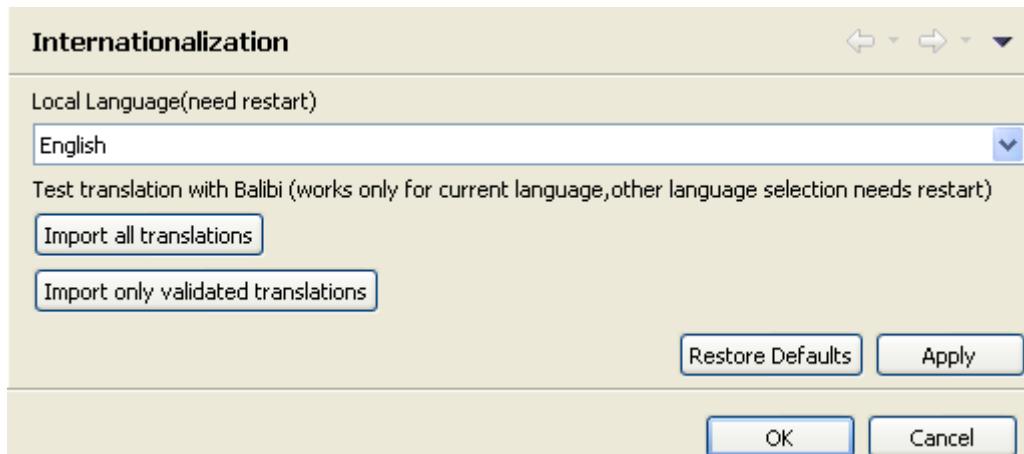
- Cliquez sur **Apply** puis sur **OK** afin de valider les préférences définies et fermer la boîte de dialogue.

Les composants externes sont ajoutés à la **Palette**.

2.5.3 Préférences de langue

Vous pouvez configurer les préférences de langue dans **Talend Open Studio**.

- Dans le menu **Window > Preferences**, développez le noeud **Talend** pour ouvrir la boîte de dialogue **[Preferences]**.
- Développez le noeud **Talend** et cliquez sur **Internationalization** pour afficher les préférences de langue.



- Dans le champ **Local Language**, sélectionnez la langue que vous souhaitez utiliser pour l'interface de **Talend Open Studio**.
- Cliquez sur **Apply**, puis sur **OK** pour valider vos changements et fermer la boîte de dialogue **[Preferences]**.
- Redémarrez **Talend Open Studio** pour que ce changement soit pris en compte.

2.5.4 Préférences d'exécution et de débogage

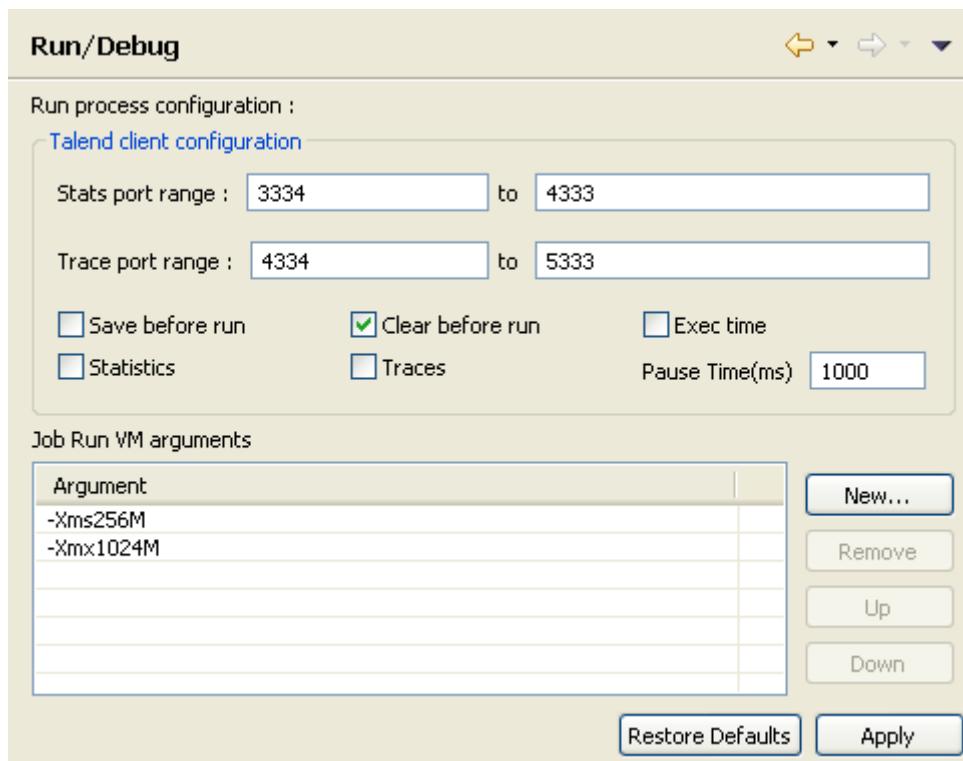
Vous pouvez configurer les préférences d'exécution et de débogage dans **Talend Open Studio**.

- Dans le menu, cliquez sur **Window > Preferences** pour afficher la boîte de dialogue **[Preferences]**.

Présentation de Talend Open Studio

Configurer les préférences de Talend Open Studio

- Développez le noeud **Talend** et cliquez sur **Run/Debug** pour afficher les préférences.



Dans la zone **Talend client configuration**, vous pouvez définir les options d'exécution à utiliser par défaut.

- Dans le champ **Stats port range**, vous pouvez modifier les ports utilisés pour la génération des Statistics. Notamment, si les ports définis par défaut sont déjà occupés par d'autres applications.
- Dans le champ **Trace port range**, vous pouvez modifier les ports utilisés pour la génération des Traces. Notamment, si les ports définis par défaut sont déjà occupés par d'autres applications.
- Cochez la case **Save before run** pour automatiquement enregistrer votre Job avant de l'exécuter.
- Cochez la case **Clear before run** pour nettoyer les résultats d'une exécution précédente avant d'exécuter de nouveau le Job.
- Cochez la case **Exec time** pour afficher la durée d'exécution du Job.
- Cochez la case **Statistics** pour afficher le suivi de flux au cours de l'exécution du Job.
- Cochez la case **Traces** pour afficher le suivi du traitement des données au cours de l'exécution du Job.
- Dans le champ **Pause time**, indiquez le temps de pause souhaitée entre chaque ligne de données du tableau Traces.

- Dans la liste **Job Run VM arguments**, vous pouvez définir les paramètres de votre JVM en fonction de votre utilisation. Les paramètres par défaut, **-Xms256M** et **-Xmx1024M** correspondent respectivement à la mémoire réservée minimale et maximale pour l'exécution de vos Jobs.

Si vous souhaitez utiliser des paramètres spécifiques pour l'exécution d'un Job, par exemple si vous voulez afficher les résultats d'exécution de ce Job en japonais, vous devez ouvrir la vue **Run**. Dans cette vue, configurez les paramètres d'exécution avancés correspondants.

Pour plus d'informations concernant les paramètres d'exécution avancés d'un Job spécifique, consultez *Configurer les paramètres d'exécution avancés, page 76*.

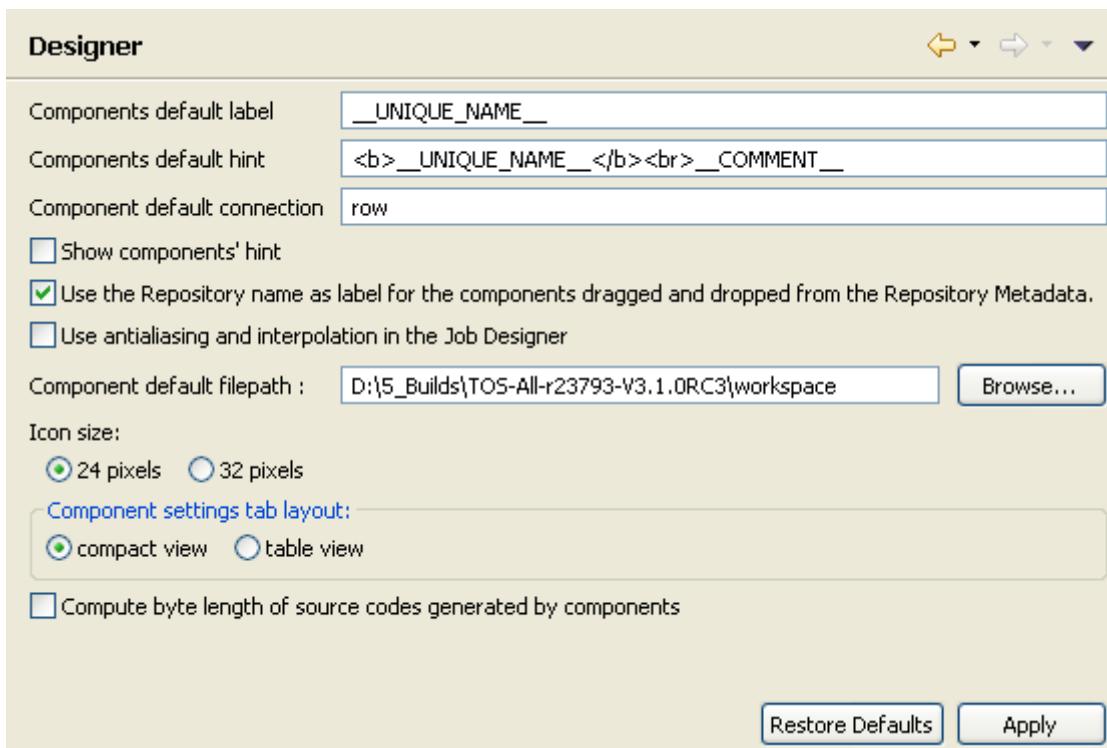
Pour plus d'informations sur les paramètres possibles, consultez le site <http://java.sun.com/javase/technologies/hotspot/vmoptions.jsp>.

2.5.5 Préférences du Designer

Vous pouvez configurer les préférences d'affichage des composants et des Jobs de manière permanente dans le Studio.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue **[Preferences]**.
- Développez le noeud **Talend > Appearance**.
- Cliquez sur **Designer** pour afficher les préférences d'affichage de l'espace de modélisation.

A partir de cette vue, vous pouvez définir l'affichage des noms et des aides contextuelles de chaque composant.

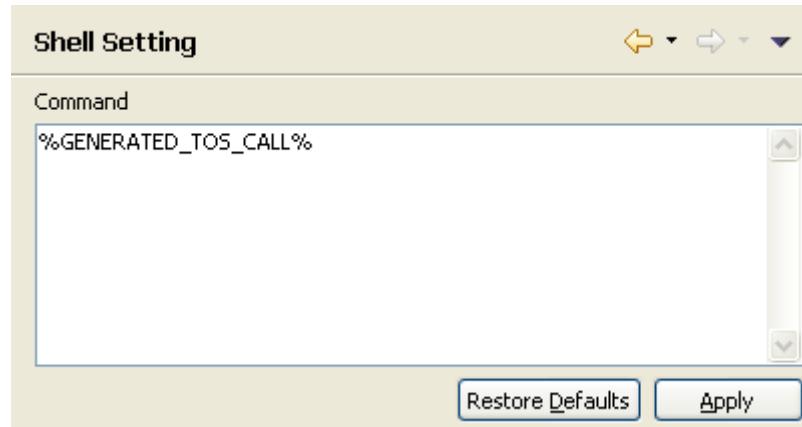


Cochez les cases adéquates pour personnaliser l'espace de modélisation de **Talend Open Studio** en fonction de votre utilisation.

2.5.6 Ajouter du code par défaut

Dans les préférences, vous pouvez ajouter du code par défaut au début ou à la fin du code de votre Job.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [Preferences].
- Développez les noeuds **Talend** et **Import/Export** et cliquez sur **Shell Settings** pour afficher le champ permettant d'ajouter du code par défaut.

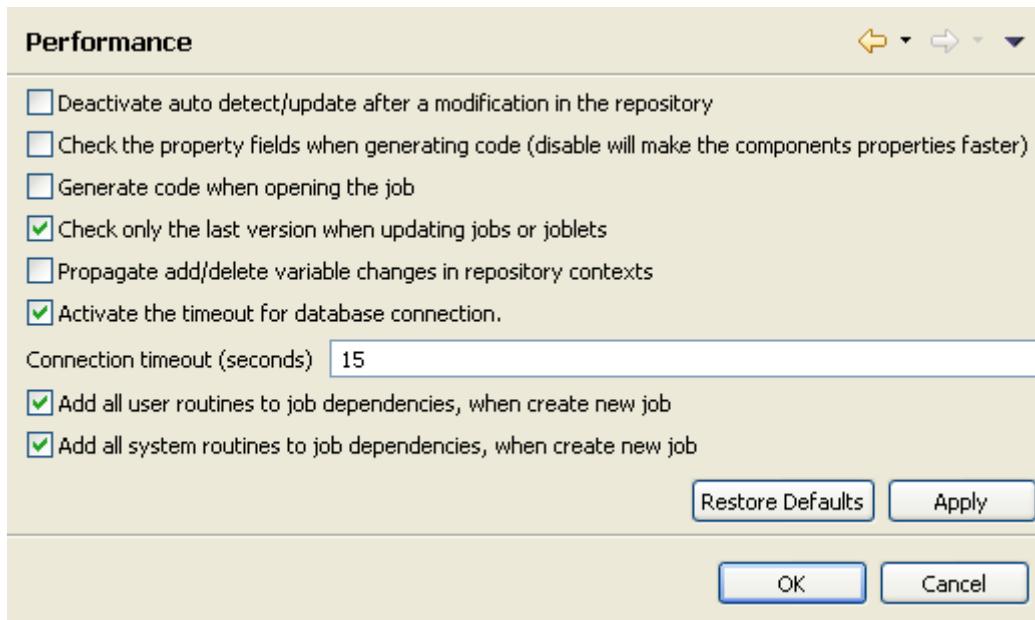


- Dans le champ **Command**, ajoutez votre/vos ligne(s) de code avant ou après le code %GENERATED_TOS_CALL% pour qu'elles apparaissent respectivement au début ou à la fin du code de votre Job.

2.5.7 Préférences de Performance

Dans les préférences, vous pouvez définir les options de rafraîchissement du **Repository** en fonction de votre utilisation de **Talend Open Studio**.

- Dans le menu, cliquez sur **Window > Preferences**, pour ouvrir la boîte de dialogue [Preferences].
- Développez le noeud **Talend** et cliquez sur **Performance** pour afficher les préférences de **Performance**.



La désactivation du rafraîchissement automatique permet un gain de performance.

Paramétrez les préférences de performance de **Talend Open Studio** en fonction de votre utilisation :

- Cochez la case **Deactivate auto detect/update after a modification in the repository** pour désactiver la détection et la mise à jour automatique du Repository après modification.
- Cochez la case **Check the property fields when generating code** pour activer la vérification des champs de propriétés des composants. Lorsque l'un des champs de propriétés d'un composant est mal renseigné, le composant est entouré en rouge.
 La désactivation de la vérification des champs de propriétés des composants permet un gain de performance. Désactivez la case **Check the property fields when generating code**.
- Cochez la case **Generate code when opening the job** pour générer le code à l'ouverture du Job.
- Cochez la case **Check only the last version when updating jobs or joblets** pour ne vérifier que la dernière version des Jobs et Joblets.
- Cochez la case **Propagate add/delete variable changes in repository contexts** pour propager les ajouts et suppressions de variables dans les métadonnées Contexts du Repository.
- Cochez la case **Activate the timeout for database connection** pour mettre en place un délai d'expiration des connexions aux bases de données. Puis dans le champ **Connection timeout (seconds)**, saisissez, en secondes, la durée souhaitée avant expiration.
- Cochez la case **Add all user routines to job dependencies, when create new job**, afin d'ajouter toutes les routines utilisateur dans les dépendances des Jobs lors de la création de nouveaux Jobs.
- Cochez la case **Add all system routines to job dependencies, when create job**, pour ajouter toutes les routines système lors de la création de nouveaux Jobs.

2.5.8 Préférences de la documentation

Dans les préférences, vous pouvez inclure le code source dans vos documentations générées.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [**Preferences**].
- Développez le noeud **Talend** et cliquez sur **Documentation** pour afficher les préférences de la documentation.



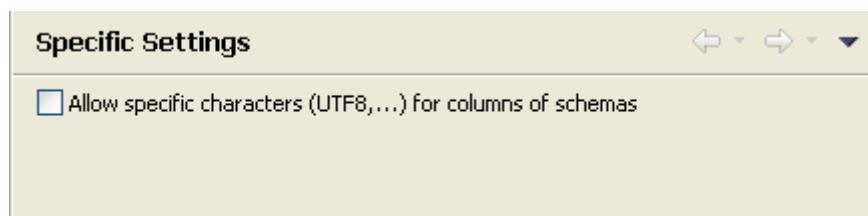
- Cochez la case **Source code to HTML generation** pour intégrer le code source dans la documentation HTML que vous générez.

Pour plus d'informations concernant la documentation, consultez *Générer la documentation HTML, page 156* et *Onglet Documentation, page 72*.

2.5.9 Afficher des caractères spéciaux dans les colonnes de schémas

Vous pouvez avoir besoin de récupérer le schéma d'une table contenant des colonnes avec caractères spéciaux comme des caractères chinois, japonais ou coréens. Dans ce cas, vous devez activer l'option permettant de lire les caractères spéciaux dans **Talend Open Studio**. Pour ce faire :

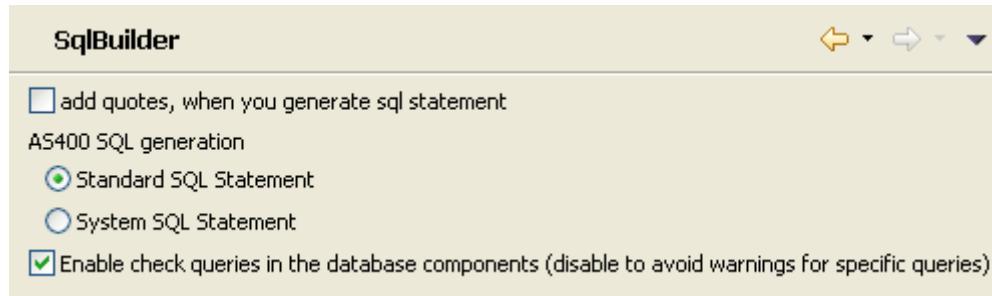
- Dans la barre de menu, cliquez sur **Window > Preferences** afin d'ouvrir la boîte de dialogue [**Preferences**].
- Dans l'arborescence de la boîte de dialogue, développez le noeud **Talend**.
- Cliquez sur le noeud **Specific settings** pour afficher la vue correspondante dans la partie droite de la boîte de dialogue.
- Cochez la case **Allow specific characters (UTF8,...) for columns of schemas**.



2.5.10 Préférences du SQL Builder

Dans les préférences, vous pouvez définir les préférences du SQL Builder.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [**Preferences**].
- Développez les noeuds **Talend** et **Specific Settings** et cliquez sur **SqlBuilder** pour afficher les préférences du SQL Builder.

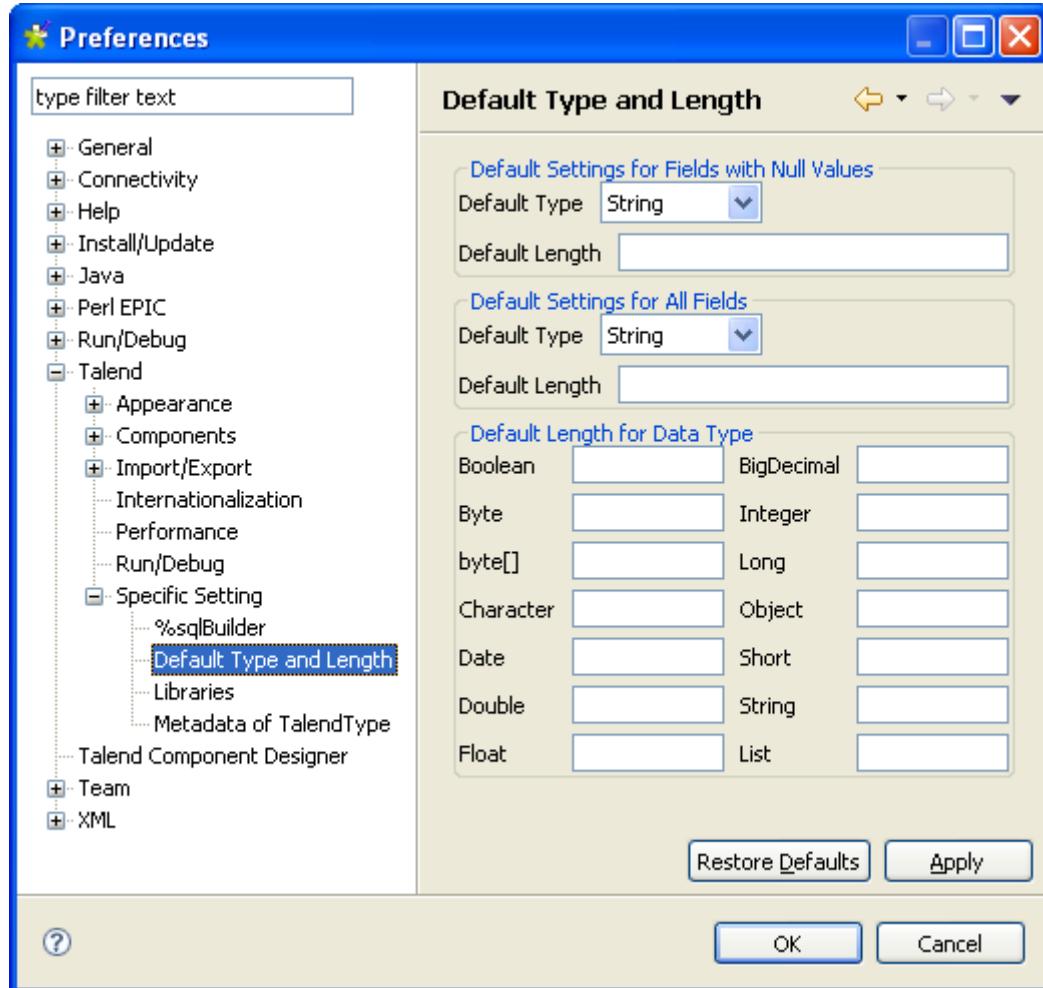


- Cochez la case **add quotes, when you generated sql statement** pour protéger le nom des colonnes et des tables par des guillemets dans vos requêtes SQL.
- Pour le champ **AS400 SQL generation**, sélectionnez **Standard SQL Statement** pour utiliser des commandes SQL standard ou **System SQL Statement** pour des commandes SQL system lorsque vous utilisez des bases de données de type AS400.
- DÉCOchez la case **Enable check queries in the database components (disable to avoid warnings for specific queries)** pour désactiver la vérification des requêtes dans les composants base de données.

2.5.11 Préférences des schémas

Dans les préférences, vous pouvez définir la longueur et le type de données par défaut des champs des schémas de vos composants.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [Preferences].
- Développez les noeuds **Talend** et **Specific Settings** et cliquez sur **Default Type and Length** pour afficher les préférences de longueur et de type des champs de vos schémas.



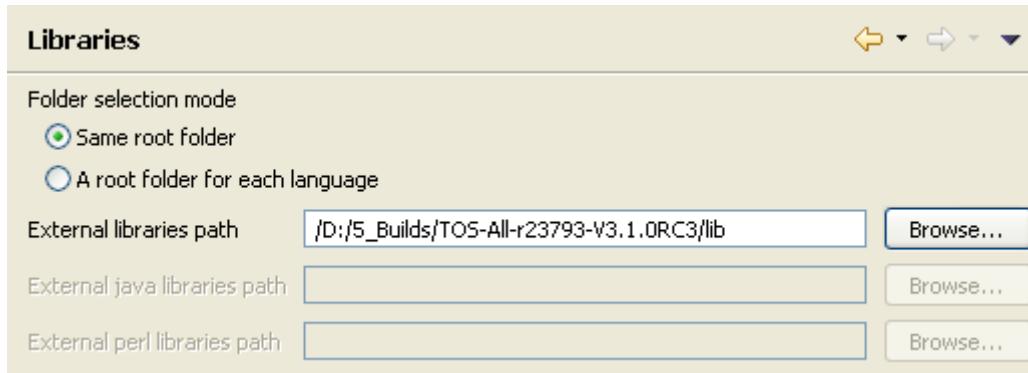
Paramétrez ces préférences en fonction de vos besoins :

- Dans la zone **Default Settings for Fields with Null Values**, renseignez le type de données et la longueur du champ à attribuer par défaut aux champs de valeurs nulles.
- Dans la zone **Default Settings for All Fields**, renseignez le type de données et la longueur du champ à attribuer à tous les champs du schéma.
- Dans la zone **Default Length for Data Type**, renseignez la longueur à attribuer en fonction du type de données du champ.

2.5.12 Préférences des bibliothèques

Dans les préférences, vous pouvez définir le répertoire dans lequel stocker les différentes bibliothèques utilisées dans **Talend Open Studio**.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [Preferences].
- Développez les noeuds **Talend** et **Specific Settings** et cliquez sur **Libraries** pour afficher les préférences des bibliothèques.

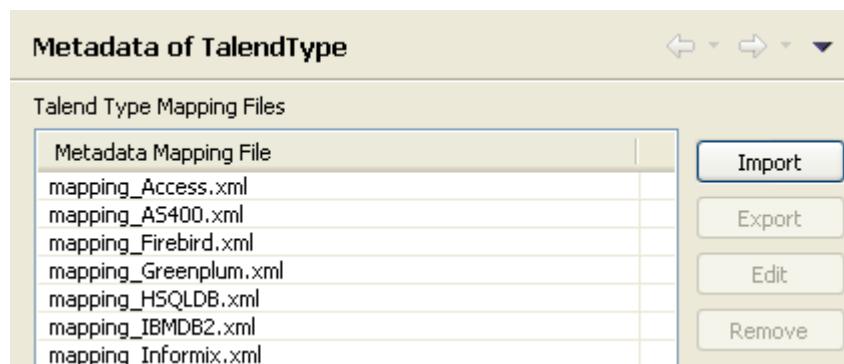


- Dans le champ **Folder selection mode**, sélectionnez **Same root folder** pour stocker les bibliothèques Java et Perl dans le même répertoire. Le chemin d'accès peut être indiqué dans le champ **External libraries path** en cliquant sur le bouton **Browse...**
- Si vous souhaitez stocker les bibliothèques Java et Perl dans deux répertoires bien distincts, sélectionnez **A root folder for each language** et spécifiez les chemins d'accès dans les champs **External java libraries path** pour les bibliothèques Java et **External perl libraries path** pour les bibliothèques Perl.

2.5.13 Préférences des conversions de type

Dans les préférences, vous pouvez paramétriser les préférences de conversion de type dans **Talend Open Studio**, de Java vers les bases de données et des bases de données vers le langage Java.

- Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue **[Preferences]**.
- Développez les noeuds **Talend** et **Specific Settings** et cliquez sur **Metadata of Talend Type** pour afficher les préférences des conversions.



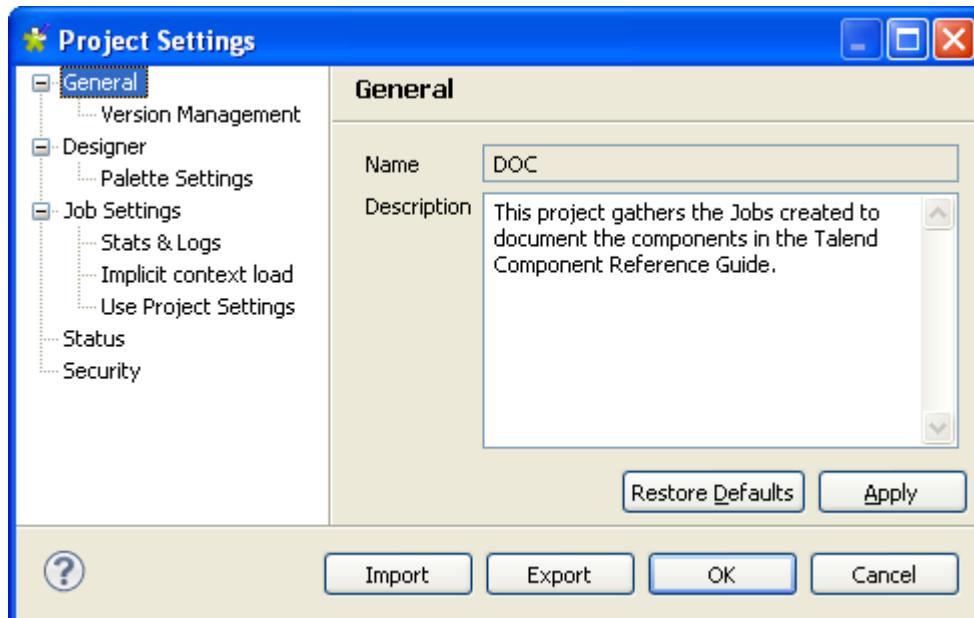
Dans la liste **Metadata Mapping File**, un fichier xml contenant les paramètres de conversion apparaît pour chaque type de base de données utilisé dans **Talend Open Studio**.

- Vous pouvez importer, exporter ou supprimer chaque fichier de conversion en cliquant respectivement sur les boutons **Import**, **Export** et **Remove**.
- Vous pouvez ouvrir et modifier chaque fichier afin d'affiner les conversions en fonction de votre utilisation en cliquant sur le bouton **Edit** et en modifiant le code XML directement dans la fenêtre **[Edit mapping file]**.

2.6 Personnaliser les paramètres du projet

Talend Open Studio vous permet de personnaliser les informations et les paramètres du projet en cours, notamment les paramètres de la Palette et du Job, la gestion du versionnement des Jobs, etc.

- Pour accéder aux paramètres du projet, cliquez sur  dans la barre d'outils du Studio ou cliquez sur **File > Edit Project Properties** dans la barre de menu. La boîte de dialogue **[Project Settings]** s'ouvre.



- Dans l'arborescence à gauche de la boîte de dialogue, sélectionnez le paramètre que vous souhaitez personnaliser et personnalisez-le dans la partie droite de la boîte de dialogue.

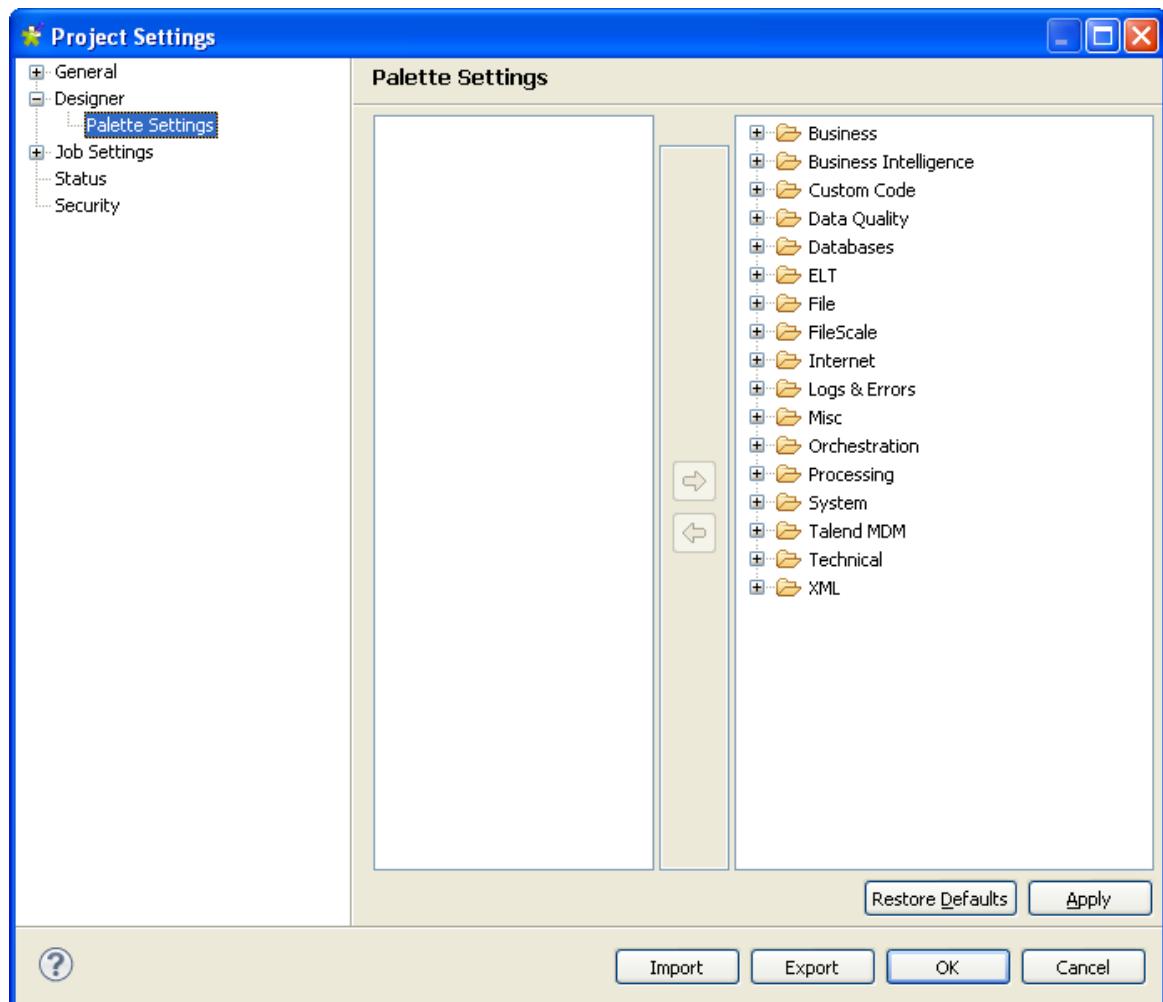
A partir de cette boîte de dialogue, vous pouvez aussi exporter ou importer l'intégralité des paramètres de votre projet.

- Pour exporter les paramètres, cliquez sur le bouton **Export**. L'export générera un fichier XML contenant tous les paramètres de votre projet.
- Pour importer les paramètres, cliquez sur le bouton **Import** et sélectionnez le fichier XML contenant les paramètres d'un autre projet à appliquer au projet en cours.

2.6.1 Personnaliser la Palette

Vous pouvez personnaliser l'affichage de la **Palette**, afin de ne charger que les composants que vous utilisez dans votre Projet, ce qui permet de lancer votre Studio plus rapidement :

- Dans la barre d'outils du Studio, cliquez sur  ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.



Dans la vue **General** de la boîte de dialogue **[Project Settings]**, ajoutez une description du projet en cours si vous ne l'avez pas fait lors de sa création.

Pour personnaliser les paramètres d'affichage de la **Palette** :

- Dans l'arborescence de la fenêtre **[Project Settings]**, développez le noeud **Designer** et cliquez sur **Palette Settings**. Les paramètres de la **Palette** en cours sont affichés dans la partie droite de la fenêtre.
- Sélectionnez un ou plusieurs composants, ou un ou plusieurs groupes de composants pour les supprimer de la **Palette** en cours.
- Puis utilisez la flèche vers la gauche pour déplacer la sélection vers la **Palette** de composants cachés, à gauche de la fenêtre. Notez que l'arborescence de la **Palette** est conservée dans la zone de gauche.
- Pour afficher de nouveau un composant caché, sélectionnez-le dans la zone de gauche et faites-la passer dans la zone de droite en cliquant sur la flèche vers la droite.
- Cliquez sur **Apply** pour valider les modifications effectuées à la **Palette** du projet en cours et cliquez sur **OK** pour fermer la boîte de dialogue.



Pour revenir aux paramètres par défaut, cliquez sur le bouton **Restore Defaults**.



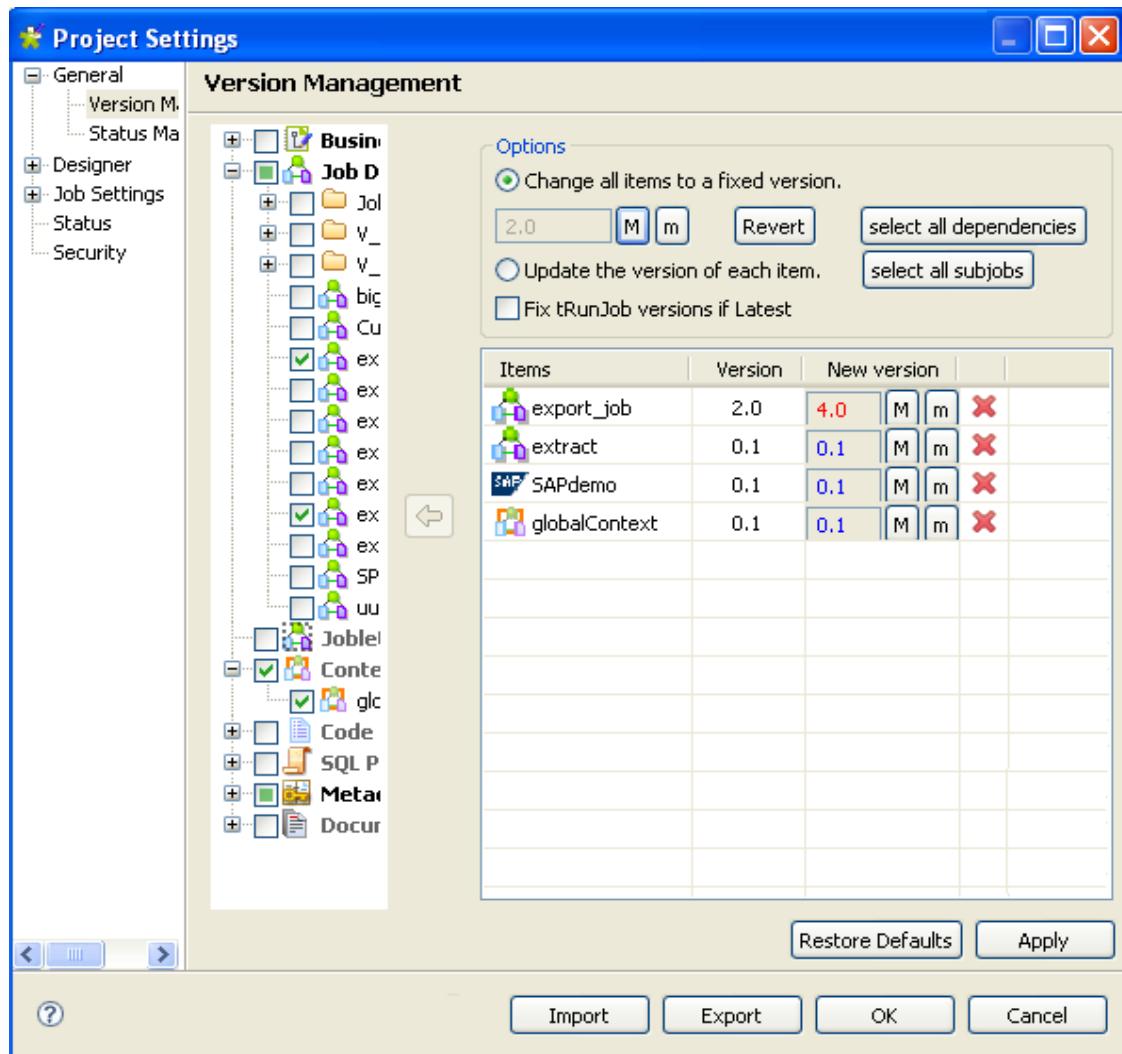
Si vous souhaitez cacher des composants qui sont utilisés dans votre Projet, une boîte de dialogue s'ouvre et vous précise que ces composants seront visibles automatiquement.

Pour plus d'informations concernant la **Palette**, consultez *Modifier la disposition et les paramètres de la Palette, page 78*.

2.6.2 Gérer les versions

Vous pouvez aussi gérer les versions des éléments du **Repository** à partir de la fenêtre **[Project Settings]**. Pour cela, développez le noeud **General** et cliquez sur **Version Management**.

- Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence de la boîte de dialogue, développez le noeud **General** et cliquez sur **Version Management** pour ouvrir la vue correspondante.



- Dans l'arborescence, développez le noeud correspondant aux éléments dont vous souhaitez gérer les versions et cochez les cases correspondant. Les éléments cochés apparaissent dans la liste **Items** avec leur version actuelle dans la colonne **Version** et la nouvelle version appliquée dans la colonne **New Version**.

- Dans la zone **Option**, sélectionnez l'option **Change all items to a fixed version** pour passer tous les éléments sélectionnés à la même version fixe.
- Cliquez sur **Revert** si vous souhaitez annuler ces modifications.
- Cliquez sur **Select all dependencies** si vous souhaitez mettre à jour tous les éléments dépendants des éléments sélectionnés à la fois.
- Cliquez sur **Select all subjobs** si vous souhaitez mettre à jour tous les sous-jobs dépendants des éléments sélectionnés à la fois.
- Pour incrémenter individuellement la version de chaque élément du Repository, sélectionnez l'option **Update the version of each item** dans la zone **Option**.
- Cochez la case **Fix tRunjob versions if Latest** si vous souhaitez que le Job parent conserve le Job enfant de la version actuelle dans le **tRunJob** à versionner, sans tenir compte de la mise à jour de leurs versions. Par exemple, un **tRunJob** fera une mise à jour à partir de la version actuelle *1.0* to *1.1* à la fois au niveau du Job parent et du Job enfant. Une fois cette case cochée, le Job parent *1.0* continuera à utiliser le Job enfant *1.0* plutôt que la dernière version (la version *1.1*), comme il le ferait habituellement.



Pour utiliser cette case, le Job parent doit utiliser le(s) Job(s) enfant(s) de la dernière version en tant que version actuelle dans le **tRunJob** à versionner, ce qui est possible en sélectionnant l'option **Latest** de la liste déroulante des versions dans la vue *Component du(d)e Job(s) enfant(s)*. Pour plus d'informations concernant le **tRunJob**, consultez la section **tRunJob** du Guide de Référence des Composants Talend Open Studio.

- Cliquez sur **Apply** pour appliquer les changements et sur **OK** pour fermer la boîte de dialogue.

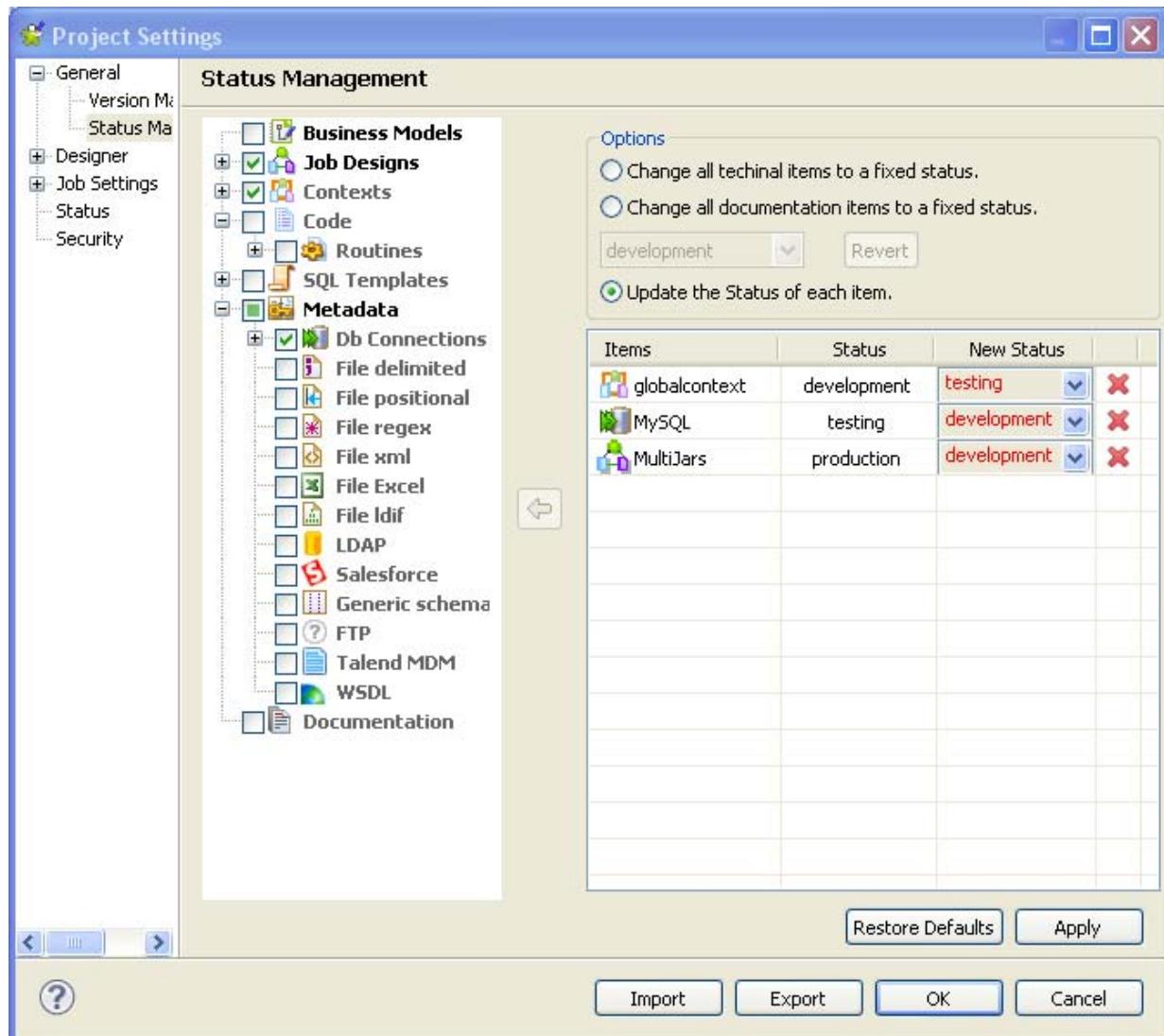


Pour plus d'informations concernant la gestion des versions, consultez *Gérer les versions*, page 34.

2.6.3 Gérer les statuts

Vous pouvez également gérer le statut de chaque élément dans la vue **Repository** via le chemin **General > Status Management** de la boîte de dialogue **[Project Settings]**.

- Dans la barre d'outils de la fenêtre principale du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre de menu afin d'ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence de la boîte de dialogue, développez le noeud **General** puis sélectionnez **Status Management** afin d'ouvrir la vue correspondante.



- Dans la vue **Repository**, développez le noeud contenant les éléments dont vous souhaitez gérer le statut et cochez la case de ces éléments.
- Les éléments sélectionnés s'affichent dans la liste **Items** de droite, avec leur statut, dans la colonne **Status**. Le nouveau statut défini s'affiche dans la colonne **New Status**.
- Dans la zone **Options**, cochez la case **Change all technical items to a fixed status** pour changer le statut de tous les éléments sélectionnés en un même statut fixe.
 - Cliquez sur **Revert** si vous souhaitez annuler les modifications.
 - Pour mettre à jour le statut des éléments, sélectionnez l'option **Update the version of each item** et changez-le manuellement.
 - Cliquez sur **Apply** afin d'appliquer vos modifications, puis sur **OK** pour fermer la boîte de dialogue.

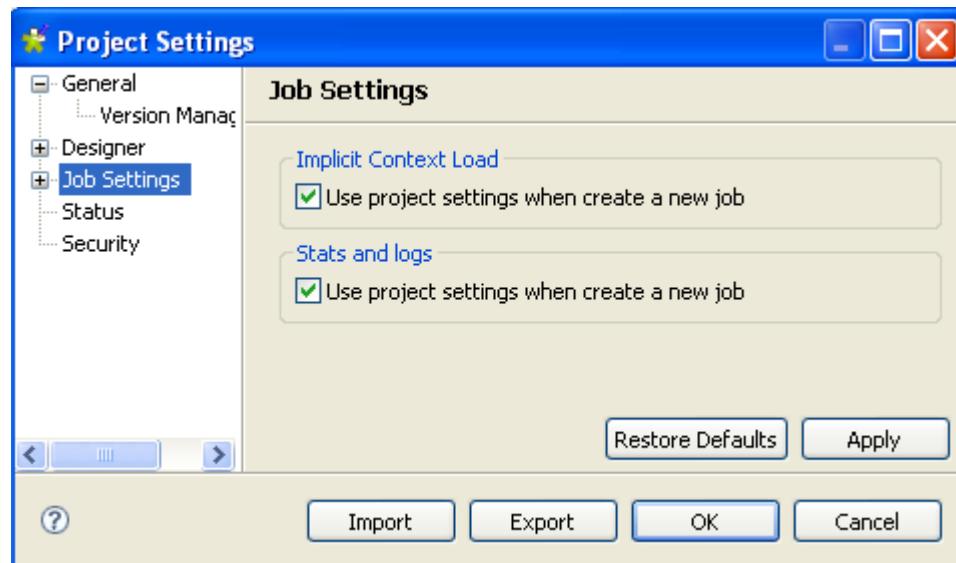


Pour plus d'informations concernant les statuts des Jobs, consultez *Personnaliser les paramètres du Job, page 37*.

2.6.4 Personnaliser les paramètres du Job

Vous pouvez automatiquement utiliser les paramètres **Implicit Context Load** et **Stats and Logs** définis dans les **Project Settings** du projet en cours lorsque vous créez un nouveau Job. Pour cela :

- Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence de la boîte de dialogue, cliquez sur le noeud **Job Settings** pour ouvrir la vue correspondante.
- Cochez la case **Use project settings when create a new job** des zones **Implicit Context Load** et **Stats and Logs**.



- Cliquez sur **Apply** pour valider les changements et cliquez sur **OK** pour fermer la boîte de dialogue.

2.6.5 Configurer les informations de logs

Lorsque vous exécutez un Job, vous avez la possibilité de monitorer son exécution à l'aide de l'option **tStatCatcher Statistics** ou des composants de log adéquats. Vous pouvez ensuite collecter les données récueillies dans des fichiers CSV ou dans une base de données.

Vous pouvez définir le chemin d'accès à ces fichiers et/ou cette base de données de log de votre projet de manière permanente, afin que les données de log soient toujours stockées dans le même répertoire.

- Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre d'outils pour ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence de la boîte de dialogue, développez le noeud **Job Settings** et cliquez sur **Stats & Logs** pour afficher la vue correspondante.

Présentation de Talend Open Studio

Personnaliser les paramètres du projet

Stats _Logs

<input checked="" type="checkbox"/> Use statistics (tStatCatcher)	<input checked="" type="checkbox"/> Use logs (tLogCatcher)	<input checked="" type="checkbox"/> Use volumetrics (tFlowMeterCatcher)
<input type="checkbox"/> On Console		
<input checked="" type="checkbox"/> On Files		
File Path	"D:/IS_Builds/TIS_EE-All-r23530-V3.1.0RC2/TIS_EE-All-r23530-V3.1.0RC2/workspace/.metadata"	
Stats File Name	"stats_file.txt"	
Log File Name	"logs_file.txt"	
Meter file name	"meter_file.txt"	
<input checked="" type="checkbox"/> On Databases		
Property Type	Repository	DB (MYSQL):localhost
Db Type	MySQL	*
Host	"localhost"	Port "3306"
Db Name	"talend"	Additional parameters "noDatetimeStringSync=
User	"root"	* Password *****
Stats Table	"statistics"	
Logs Table	"logs"	
Meter Table	"volumetrics"	
<input checked="" type="checkbox"/> Catch runtime errors	<input checked="" type="checkbox"/> Catch user errors	<input checked="" type="checkbox"/> Catch user warnings
<input type="checkbox"/> Catch components statistics (tStatCatcher Statistics)		



Si les paramètres des Stats & Logs ne varient pas en fonction du contexte d'exécution, vous pouvez les définir de manière permanente. Si vous souhaitez appliquer les mêmes paramètres de Stats & Logs en fonction de chaque Job, faites-le directement à partir de la vue Stats & Logs du Job. Pour plus d'informations sur cette vue, consultez *Automatiser l'utilisation de statistiques & logs, page 124*.

- Cochez les cases **Use statistics**, **Use logs** et **Use volumetrics** en fonction de vos besoins, puis renseignez le chemin d'accès des données de log.
- Sélectionnez le format de stockage des données de log : cochez **On Files** ou **On Database**. Ou cochez la case **On Console** pour afficher les données dans la console.
- Cliquez sur **Apply** pour appliquer les changements et sur **OK** pour fermer la boîte de dialogue.

Ainsi, les champs apparaissent en fonction des paramètres sélectionnés. Dans les champs **File Name** ou **DB Name**, saisissez respectivement le nom du fichier ou de la base de données entre guillemets en fonction du type de format choisi.

Notez que vous pouvez maintenant stocker les informations de connexion à la base de données dans le **Repository**. Pour cela, sélectionnez **Repository** dans la liste déroulante **Repository Type** et cliquez sur le bouton [...] pour stocker ces informations dans la métadonnée correspondante. Les champs suivants sont renseignés automatiquement.

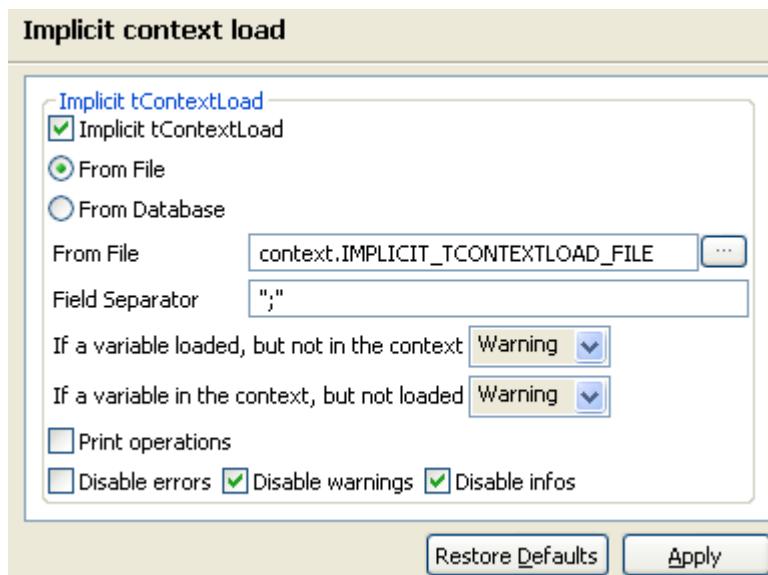


Si vous avez sauvé les informations de connexion dans une variable de contexte, vous pouvez y accéder via le raccourci **Ctrl+Espace**.

2.6.6 Définir les paramètres de contexte

Dans les paramètres du projet, vous pouvez définir des paramètres de contexte automatiques utilisés par défaut dans vos Jobs.

- Dans la barre d'outils du Studio, cliquez sur  ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence de la boîte de dialogue, développez le noeud **Job Settings** et cliquez sur **Implicit Context Load** pour afficher les paramètres de configuration de l'option **Implicit tContextLoad**.

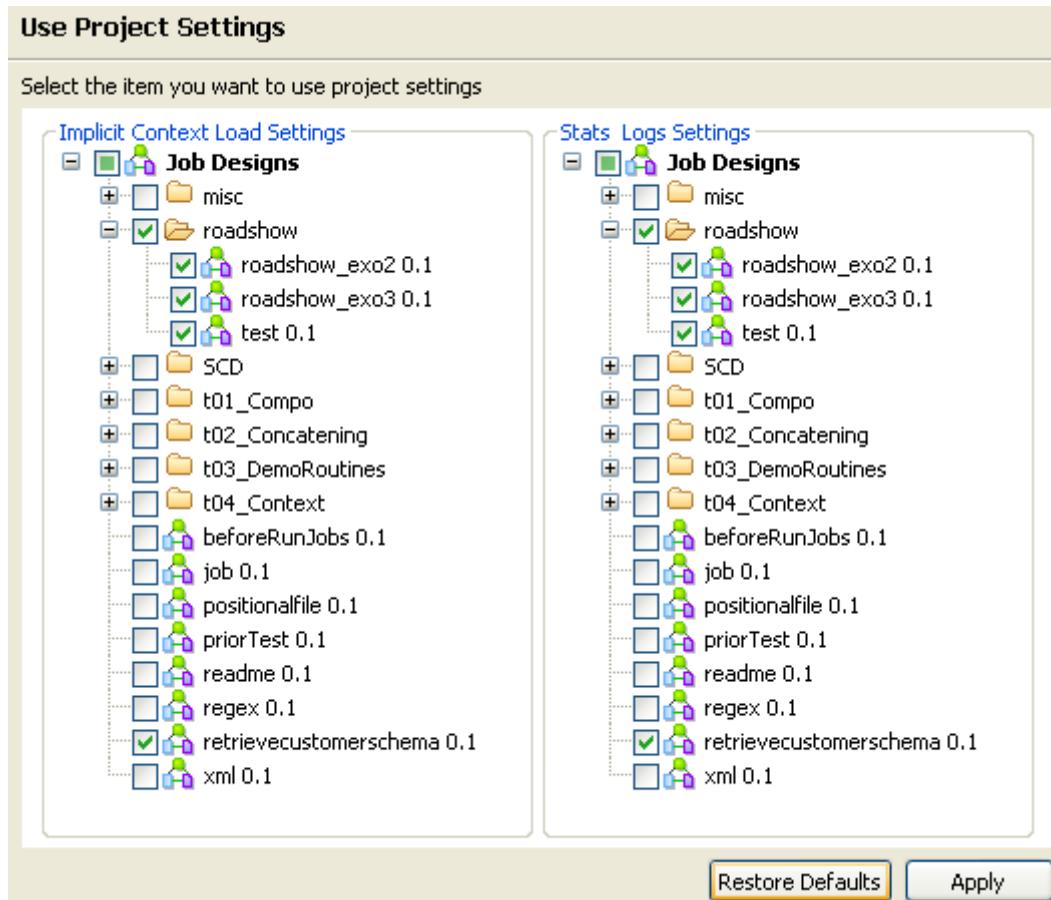


- Cochez la case **Implicit tContextLoad**.
- Sélectionnez le type de fichier dans lequel les données de contextes seront conservées, soit sous forme de fichier en sélectionnant le champ **From File**, soit sous forme de base de données en sélectionnant **From Database**.
- Pour un fichier, renseignez le chemin d'accès et le séparateur de champ du fichier contenant les paramètres de contexte dans les champs **From File** et **Field Separator**.
- Pour une base de données, sélectionnez le mode adéquat, **Built-in** ou **Repository**, dans le champ **Repository type** et renseignez les champs suivants.
- Dans les champs **Table Name** et **Query Condition**, renseignez le nom de la table contenant les paramètres de contexte et la requête à utiliser.
- Dans les champs **If a variable loaded, but not in the context** et **If a variable in the context, but not loaded**, sélectionnez le type d'avertissement que vous souhaitez recevoir en fonction de votre utilisation des variables.
- Cliquez sur **Apply** pour valider les changements et cliquez sur **OK** pour fermer la boîte de dialogue.

2.6.7 Appliquer les paramètres du projet

Dans les paramètres du projet, vous pouvez définir à quels Jobs du Repository vous souhaitez appliquer les paramètres **Implicit Context Load** et **Stats and Logs**. Pour cela :

- Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence de la boîte de dialogue, développez le noeud **Job Settings** et cliquez sur **Use Project Settings** pour afficher l'utilisation des options **Implicit Context Load** et **Stats and Logs** dans les Jobs.



- Dans la zone **Implicit Context Load Settings**, cochez les cases correspondant aux Jobs auxquels vous souhaitez appliquer l'option **Implicit Context Load**.
- Dans la zone **Stats Logs Settings**, cochez les cases correspondant aux Jobs auxquels vous souhaitez appliquer l'option **Stats and Logs**.
- Cliquez sur **Apply** pour valider les changements et sur **OK** pour fermer la boîte de dialogue.

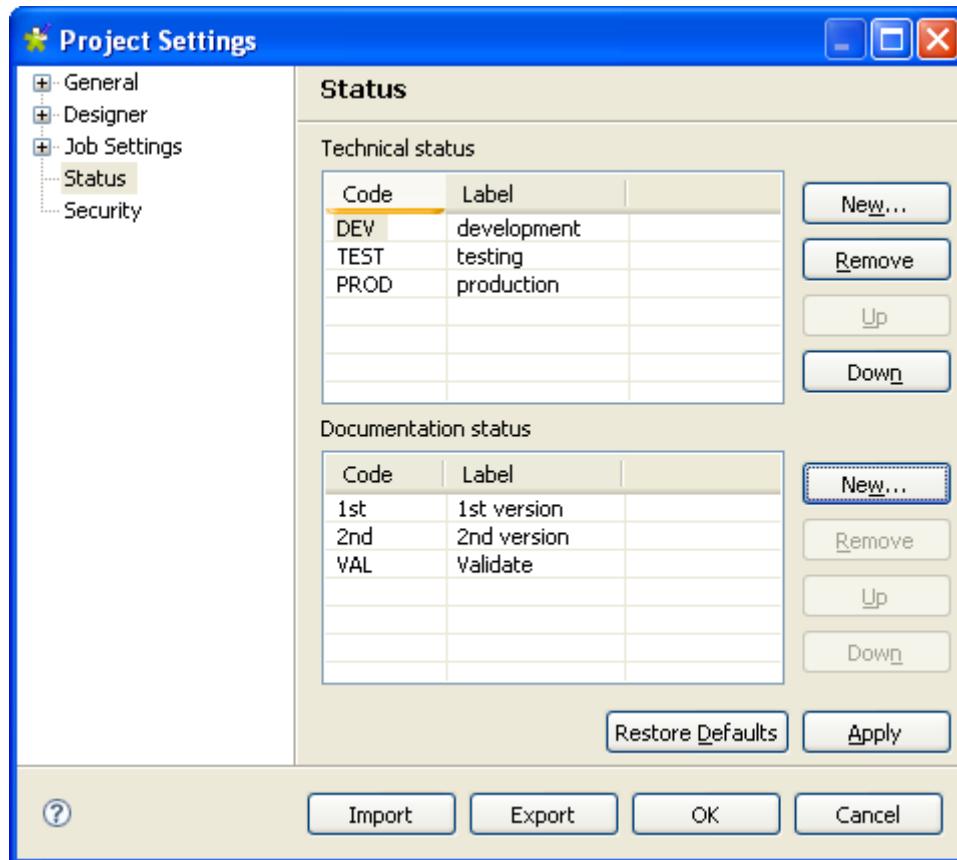
2.6.8 Définir les paramètres des Statuts

Dans les paramètres du projet, vous pouvez définir des niveaux de statuts.

- Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.

- Dans l'arborescence, cliquez sur le noeud **Status** pour paramétrer les propriétés principales des éléments du **Repository**.

Les propriétés principales d'un élément du repository comprend des informations telles que **Name**, **Purpose**, **Description**, **Author**, **Version** and **Status** de l'élément sélectionné. La plupart des propriétés sont des champs de saisie, cependant le champ **Status** est une liste déroulante.



- Cliquez sur le bouton **New...** pour afficher un boîte de dialogue et alimenter la liste **Status** avec les valeurs appropriées aux besoins de votre entreprise. Notez que le champ **Code** ne peut pas dépasser trois caractères et le champ **Label** contenant le libellé de votre statut est obligatoire.



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

La liste de statuts **Technical status** affiche les codes de classification des éléments qui sont utilisés lors de l'exécution de Jobs, de définition de métadonnées ou de routines.

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

- Une fois le statut renseigné, cliquez sur **OK** pour l'ajouter à la liste.

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

- Dans la boîte de dialogue **[Project Settings]**, cliquez sur **Apply** pour valider vos changements et sur **OK** pour fermer la boîte de dialogue.

2.6.9 Afficher/masquer les paramètres de sécurité

Dans les paramètres de projet, vous pouvez afficher ou masquer les mots de passe de vos documentations, métadonnées, contextes, etc. lorsqu'ils sont centralisés dans le **Repository**.

- Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans le menu pour ouvrir la boîte de dialogue **[Project Settings]**.
- Dans l'arborescence, cliquez sur le noeud **Security** pour ouvrir la vue correspondante.
- Cochez la case **Hide passwords** pour masquer vos mots de passe.



Si vous cochez la case **Hide passwords**, vos mots de passe seront masqués dans toutes vos documentations, contextes, etc., ainsi que dans les propriétés de vos composants si vous avez sélectionné l'option **Repository** dans le champ **Property Type** de l'onglet **Basic settings** de la vue **Component** comme illustré dans la capture d'écran ci-dessous. Si vous sélectionnez l'option **Built-in** dans le champ **Property Type** de l'onglet **Basic settings** de la vue **Component**, votre mot de passe ne sera pas masqué.

The screenshot shows the 'Edit Project Properties' dialog with the 'Basic settings' tab selected. The 'Property Type' dropdown is set to 'Repository' and the 'Connection' dropdown is set to 'DB (MYSQL):demoMysql'. The 'Host' field contains 'localhost', 'Port' is set to '3306', 'Database' is 'test', 'Username' is 'root', and 'Password' is masked as '****'. A checkbox 'Use an existing connection' is checked. The 'Basic settings' tab has several tabs: 'General', 'Advanced', 'Performance', 'Security', and 'Audit'.

- Dans la boîte de dialogue **[Project Settings]**, cliquez sur **Apply** pour valider vos changements et sur **OK** pour fermer la boîte de dialogue.



CHAPITRE 3

Conception d'un Business Model

Talend Open Studio a été spécialement conçu pour permettre une représentation simple et claire des besoins métiers de l'entreprise, ainsi que leur interaction. **Talend Open Studio** permet ainsi de schématiser les systèmes d'information, connexions, process et besoins à l'aide de diagrammes standard de flux de données disponibles dans une boîte à outils graphiques intuitive.

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

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

3.1 Objectifs

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

La conception des Business Models est une bonne pratique que les entreprises devraient adopter dès le début d'un projet d'intégration de données, afin de garantir la réussite de leur projet. Les Business Models permettent généralement d'identifier et de résoudre rapidement les goulots d'étranglement et autres points faibles du projet à mettre en place, ainsi que de limiter les dépassements de budget, voire de réduire l'investissement initial. Puis, pendant et après la mise en place du projet, les Business Models peuvent être revus et corrigés, si besoin est.

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 opérationnels au sein d'une entreprise. Il faut ajouter brique par brique les besoins supplémentaires et les relier 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 leurs 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.

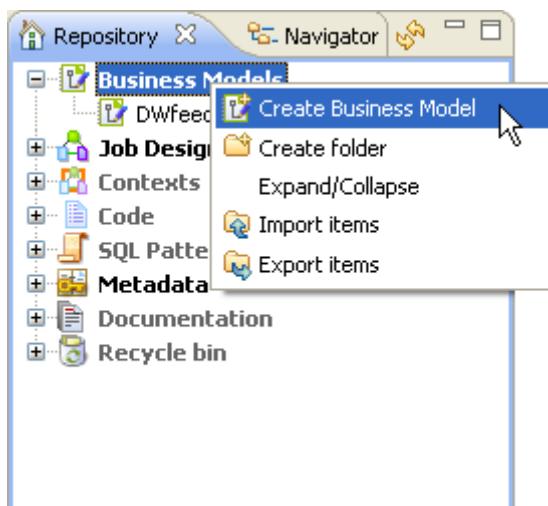
A partir de **Talend Open Studio**, vous disposez de nombreux outils vous permettant de :

- 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.

3.2 Crédit ou ouverture d'un Business Model

Lancez **Talend Open Studio** suivant la procédure détaillée dans la section *Lancer Talend Open Studio, page 7*.

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.

3.2.1 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.

3.2.2 Créer un Business Model

Cliquez-droit sur **Business Models** et sélectionnez **Create Business Model**.

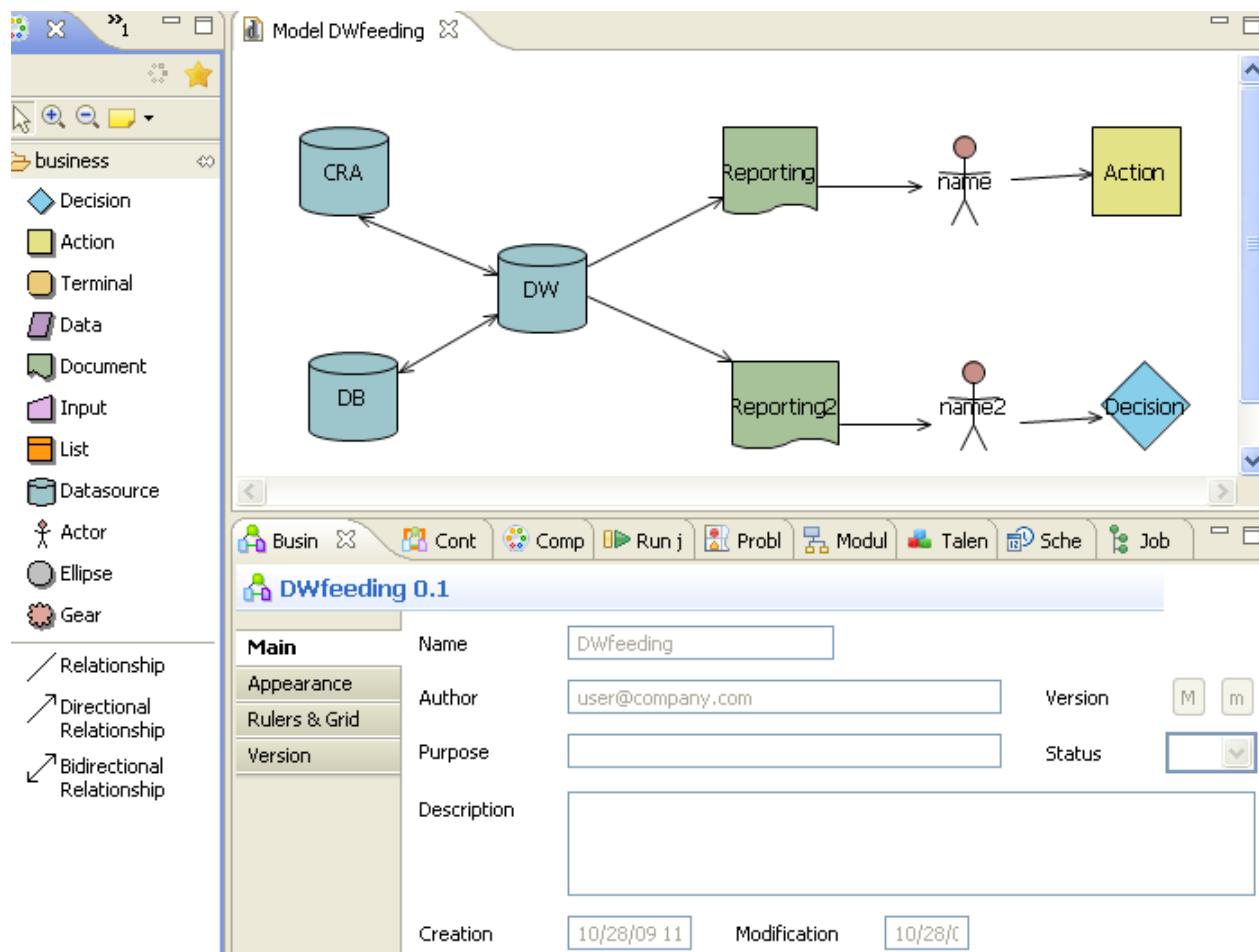
L'assistant de création vous guide dans les étapes de création d'un Business Model.

Sélectionnez le répertoire de destination (**Location**) dans lequel vous souhaitez conserver votre nouveau modèle.

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** s'ouvre sur un espace de modélisation 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 process ouverts.

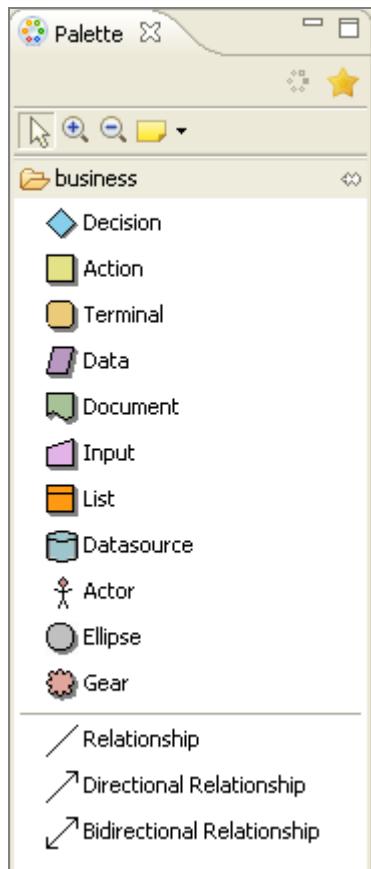


Le **Modeler** est composé des vues suivantes :

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

3.3 Modélisation d'un Business Model

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



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

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

3.3.1 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 :



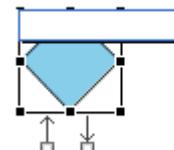
Par exemple, si votre Business Model inclut une étape de décision, sélectionnez la forme losange dans la **Palette**.



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 process.

Deux flèches opposées apparaissent sur cette forme, elles vous permettent de créer des connexions vers d'autres formes. Vous pouvez ainsi rapidement définir une séquence et des dépendances entre les éléments formant votre process.

Voir également : *Connexions, page 48.*

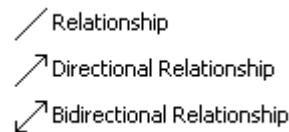
Les formes disponibles comprennent :

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 d'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.

3.3.2 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 :

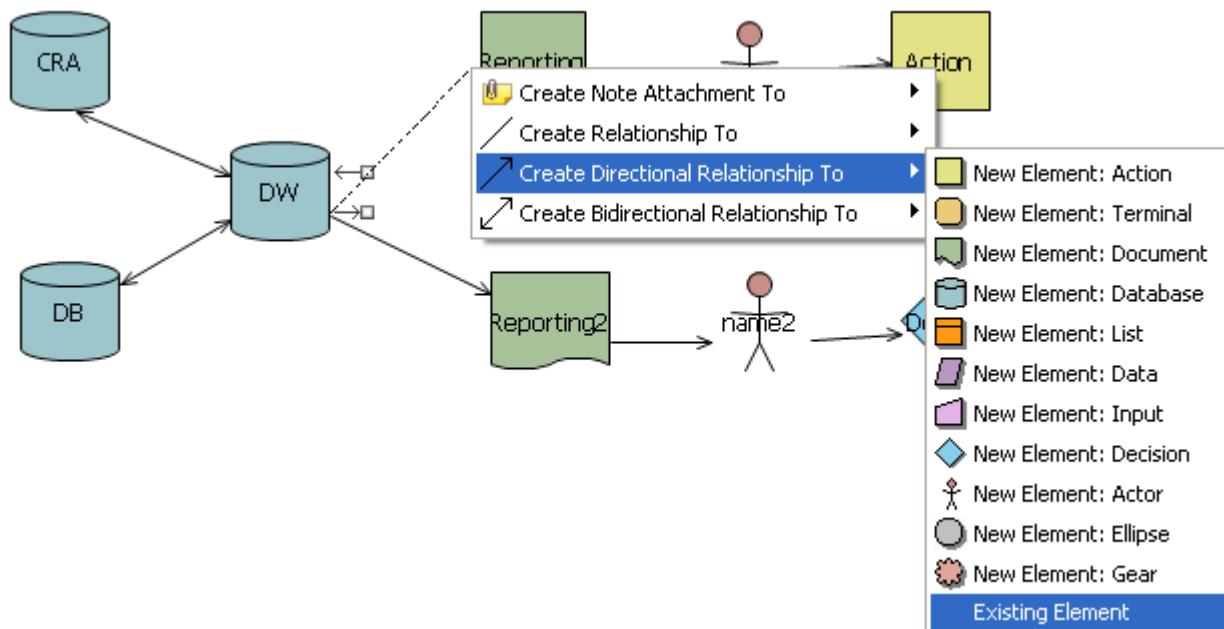


Sélectionnez l'outil **Relationship** de la **Palette**. Puis dans l'éditeur graphique, tirez un lien d'une forme vers une autre pour dessiner la connexion.

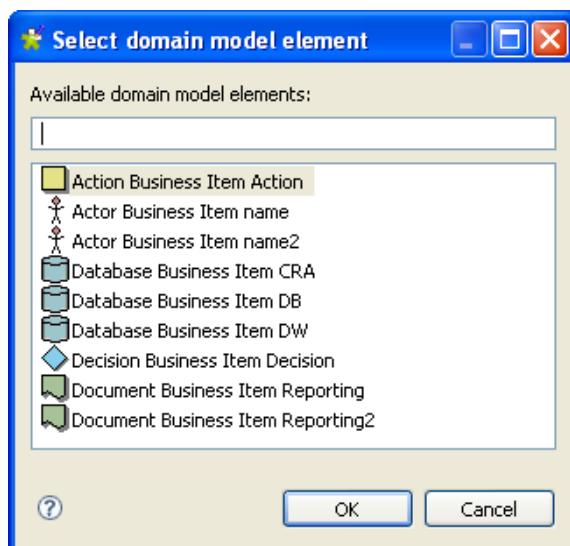
Ou si vous n'avez pas encore placé la forme cible de la connexion, vous pouvez créer la relation et la cible en une fois :

- Placez simplement 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 : **Create Relationship To**, **Create Directional Relationship To** ou **Create Bidirectional Relationship To**.
- Puis sélectionnez l'élément cible de la connexion parmi les éléments proposés.



Vous pouvez créer une connexion vers un élément déjà en place dans votre modèle. Sélectionnez **Existing Element** dans le menu contextuel puis choisissez l'élément à connecter.



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

La nature de cette connexion peut être décrite à l'aide d'éléments du référentiel de métadonnées (Repository) et sa mise en forme peut être paramétrée dans l'onglet **Business Model**, consultez la section *Propriétés, page 51*.

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.



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

Voir également : *Commenter et réarranger un Business Model, page 50*

3.3.3 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 effectuer un zoom 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 process.

Ajouter une note ou du texte libre

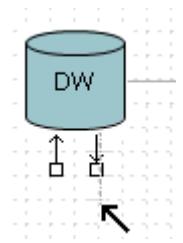
Pour ajouter une note, sélectionnez l'outil **Note** dans la **Palette**, à droite de l'éditeur graphique.

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.

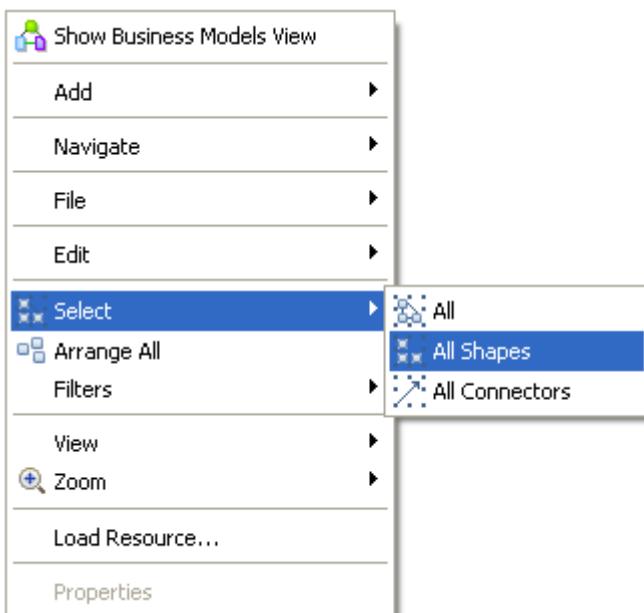
Pour ajouter une note attachée à une forme, sélectionnez **Note attachment** dans la **Palette** puis sélectionnez dans l'espace de modélisation la forme que vous souhaitez annoter. Lorsque la flèche noire apparaît, faites-la glisser vers une zone vide de l'éditeur. Relâchez pour afficher le menu contextuel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



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

Réarranger une vue métier

Vous pouvez adapter le "look and feel" de votre Business Model via le menu contextuel.



Placez votre curseur dans l'éditeur, cliquez-droit pour afficher le menu, et sélectionnez *Arrange all*. Les formes se déplacent automatiquement pour une lisibilité optimale du Business Model.

Vous pouvez également sélectionner manuellement tout ou partie du process.

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

Vous pouvez sélectionner :

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

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

3.3.4 Propriétés

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

La vue **Business Model** comporte différents types d'informations concernant :

- les informations de création du Business Model dans l'onglet **Main**
- le format dans l'onglet **Appearance**
- les règles et la grille dans l'onglet **Rulers and Grid**
- les affectations dans l'onglet **Assignment**

L'onglet **Main** affiche des informations de base sur l'élément sélectionné dans l'espace de modélisation, qu'il s'agisse d'un Business Model ou d'un Job. Pour plus d'informations sur l'onglet **Main**, consultez *Onglets de configuration, page 297*.

Onglet Appearance

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

L'onglet **Business Model** inclut les options de format suivants :

- changer la couleur de la forme

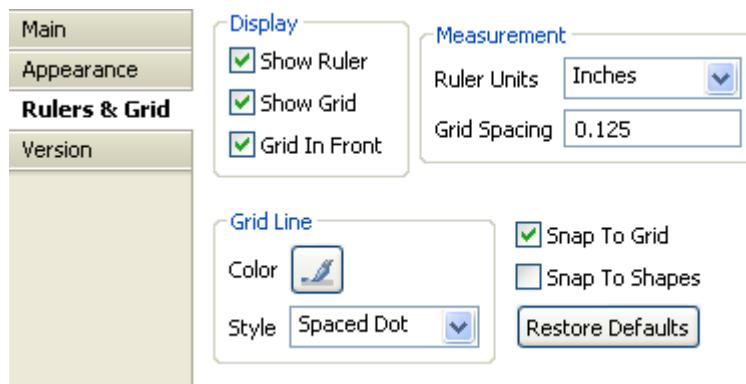
- colorer les bordures
- ajouter du texte
- ajouter un effet dégradé
- ajouter un effet ombré

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

Onglet Rulers and Grid

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.



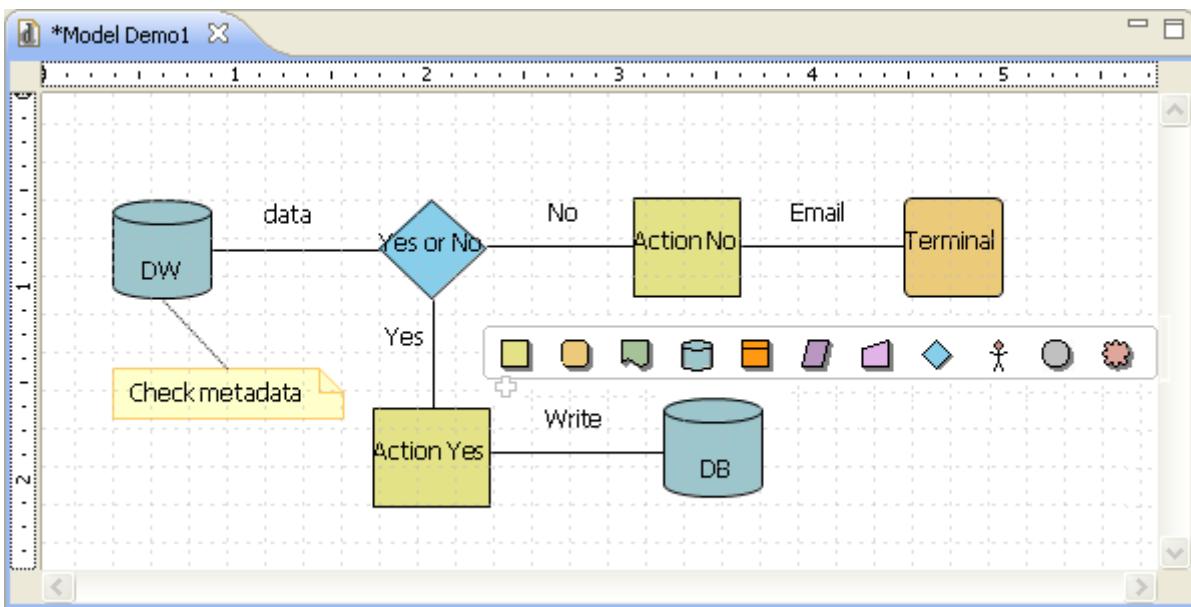
Dans la zone **Display**, cochez la case **Show Ruler** pour afficher la Règle (**Ruler**), ou **Show Grid** pour afficher la Grille (**Grid**) ou les deux. **Grid in front** place la grille devant toutes les autres formes du process.

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

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

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

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



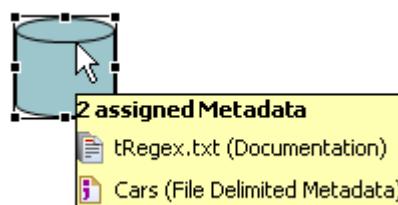
Onglet Assignment

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, puis cliquez l'onglet **Assignment** de la vue **Business Model**.

Type	Name	Comment
Documentation	tRegex.txt	
File Delimited Metadata	Cars	

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

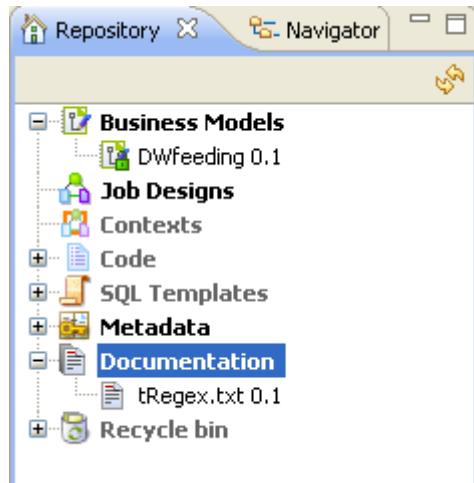


Vous pouvez modifier certaines informations ou lier un commentaire. En outre, si vous mettez à jour des données dans le référentiel, les informations d'affectation sont automatiquement mises à jour.

Pour plus d'informations concernant les affectations, voir *Affectation d'éléments du Repository à un Business Model, page 54.*

3.4 Affectation d'éléments du Repository à un Business Model

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



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

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

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

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

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

Elément	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 d'intégration de données, page 57*

3.5 Modification d'un Business Model

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

3.5.1 Renommer un Business Model

Assurez-vous que le Business Model est fermé.

Cliquez-droit sur le nom du Business Model dans le **Repository** et sélectionnez **Edit properties** pour afficher les propriétés principales dans la boîte de dialogue [**Edit properties**].

Apportez vos modifications dans le champ **Name** et cliquez sur **Finish** pour fermer la boîte de dialogue. Le nom du Business Model change automatiquement dans le **Repository**. Si votre Business Model est ouvert, refermez-le, et le nouveau nom apparaîtra dans l'onglet en haut de l'espace de modélisation lorsque vous l'ouvrez de nouveau.



Si le Business Model est ouvert, les informations de la boîte de dialogue [**Edit properties**] seront en lecture seule et ne pourront donc pas être modifiées.

3.5.2 Copier et coller un Business Model

Dans **Repository > Business Model**, cliquez-droit sur le nom 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 et sélectionnez **Paste**.

3.5.3 Déplacer un Business Model

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

Puis glissez-le à son nouvel emplacement.

3.5.4 Supprimer un Business Model

Cliquez-droit sur le nom 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**.

3.6 Enregistrement d'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.



Pour sauvegarder un Business Model et incrémenter sa version dans le même temps, cliquez sur **File>Save as....** La boîte de dialogue [**Save as**] s'ouvre.

- Dans le champ **Version**, cliquez sur le bouton **[M]** pour incrémenter la version majeure et sur le bouton **[m]** pour incrémenter la version mineure.
- Cliquez sur **Finish** pour valider.



Par défaut, lorsque vous ouvrez un Business Model, vous ouvrez sa dernière version.
Les versions précédentes sont en lecture seule et ne peuvent donc plus être modifiées.

Pour consulter les différentes versions de votre Business Model :

- Sélectionnez-le dans le **Repository**.
- Cliquez sur la vue **Business Model** et sur l'onglet **Version**.
- La liste des versions apparaît sous forme de tableau.
- Faites un clic-droit sur la version que vous souhaitez consulter.
- Sélectionnez l'option :

Select	To...
Edit properties	Pour consulter les propriétés du Job en lecture seule
Read Business Model	Pour consulter le Business Model en lecture seule.



Vous pouvez ouvrir et modifier la dernière version d'un Business Model, à partir de la vue **Version**, si vous sélectionnez **Edit Business Model** dans le menu contextuel.



CHAPITRE 4

Conception d'un Job d'intégration de données

Les fonctionnalités de **Talend Open Studio** permettent de traiter toutes les sources et les cibles utilisées dans un processus d'intégration de données, ainsi que tous les opérations associées.

Les Job Designs que vous modélez dans **Talend Open Studio** vous permettent de mettre en place des processus de gestion des flux de données directement exécutables.

Ce chapitre s'adresse aux programmeurs et responsables informatiques qui souhaitent mettre en place la couche technique d'un processus, basé (ou non) sur un Business Model **Talend Open Studio**.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

4.1 Objectifs

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

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

Avec **Talend Open Studio**, vous pouvez :

- modéliser des actions dans votre Job grâce à la bibliothèque de composants techniques,
- changer les paramètres par défaut des composants et même créer de nouveaux composants ou familles de composants qui correspondent au mieux à vos besoins,
- paramétriser les connexions et les relations entre composants afin de définir la nature des actions et leur déroulement,
- accéder au code généré pour le programme ou les composants afin de le modifier ou de le documenter,
- créer et ajouter des éléments au **Repository** (Référentiel) à des fins de réutilisation et de partage (avec d'autres projets, d'autres processus ou, à moyen terme, avec d'autres utilisateurs).



*Pour pouvoir exécuter les Jobs que vous avez créé dans **Talend Open Studio**, vous devez installer une JVM Sun 1.5 ou une version ultérieure (les JVM IBM ne sont pas supportées). Vous pouvez la télécharger sur <http://java.sun.com/javase/downloads/index.jsp>*

4.2 Utiliser un Job Design simple



Jusqu'à ce qu'un Job soit créé, l'espace de modélisation graphique est indisponible, et la Palette ne s'affiche pas.

Un Job Design est composé d'un ou plusieurs sous-job(s), eux-mêmes définis par un ou, plus vraisemblablement, par plusieurs composants reliés les uns aux autres. Les propriétés de chaque composant doivent être configurées individuellement, afin que tout fonctionne correctement.

Pour plus d'informations, consultez *Types de connexions, page 83* et *Définition des propriétés d'un composant, page 65*.

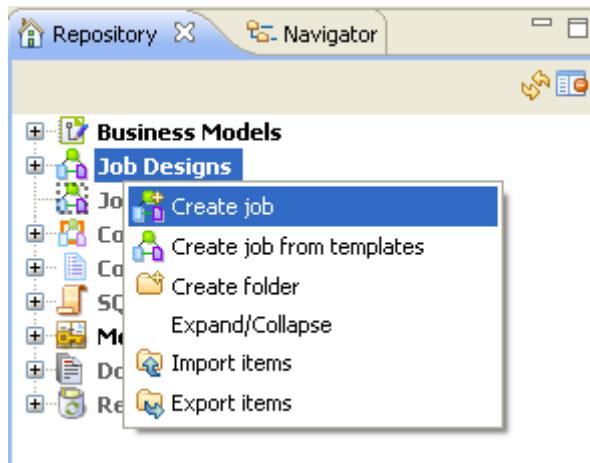
4.2.1 Crée un Job

Talend Open Studio vous permet de créer un Job d'intégration de données en déposant différents composants techniques de la **Palette** dans l'espace de modélisation graphique, puis en connectant ces composants les uns aux autres.

Vous pouvez également créer différents dossiers pour mieux classifier ces Jobs.

Pour créer un Job d'intégration de données, procédez comme suit :

- Ouvrez **Talend Open Studio** en suivant la procédure détaillée dans le chapitre *Lancer Talend Open Studio, page 7.*
- Dans la vue **Repository**, cliquez-droit sur le noeud **Job Designs**, et sélectionnez **Create job** dans le menu contextuel.



L'assistant de création [**New Job**] vous guide dans les étapes de définition des propriétés de votre nouveau Job.



- Renseignez les propriétés du Job comme suit :

Conception d'un Job d'intégration de données

Utiliser un Job Design simple

Champ	Description
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.
Locker	Ce champ est en lecture seule car il reprend par défaut le login utilisateur ayant verrouillé le Job courant. Ce champ est vide lorsque vous créez un Job, et n'a des données que lorsque vous éditez les propriétés d'un Job existant.
Version	La version indiquée est en lecture seule. Vous pouvez cependant incrémenter manuellement la version à l'aide des boutons M et m . Pour plus d'informations, consultez <i>Gérer les versions, page 34</i> .
Status	Sélectionnez dans la liste le statut du Job que vous créez.
Path	Sélectionnez le dossier dans lequel vous souhaitez créer le Job

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

- Glissez-déposez les composants que vous souhaitez utiliser de la **Palette** dans l'espace de modélisation graphique, et connectez-les. Pour plus d'informations, consultez *Cliquer & déposer des composants, page 61* et *Types de connexions, page 83*.
- Définissez les propriétés de chaque composant utilisé dans le Job. Pour plus d'informations, consultez *Définition des propriétés d'un composant, page 65*.
- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter. Pour plus d'informations, consultez *Exécution d'un job, page 72*.

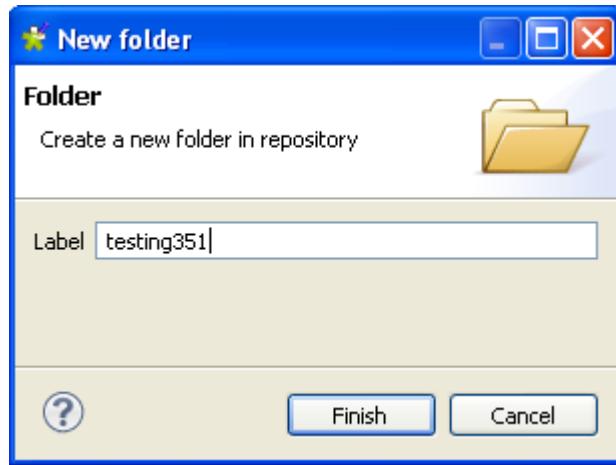
Le Job que vous avez créé s'affiche sous le noeud **Job Designs**, dans la vue **Repository**.



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

Pour créer différents dossiers afin de classer vos Jobs, procédez comme suit :

- Dans la vue **Repository**, cliquez-droit sur **Job Designs** et sélectionnez **Create Folder** dans le menu contextuel.
La boîte de dialogue **[New Folder]** apparaît.



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



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

Pour un scénario expliquant comment créer un vrai Job d'intégration de données, consultez *De la théorie à la pratique, exemple de Job, page 303*.

4.2.2 Cliquer & déposer des composants

Déposer des composants à partir de la Palette

Pour commencer la création d'un Job, cliquez sur un **Composant de la Palette**. Puis cliquez dans l'espace de modélisation pour le déposer et ainsi l'ajouter à votre Job Design.

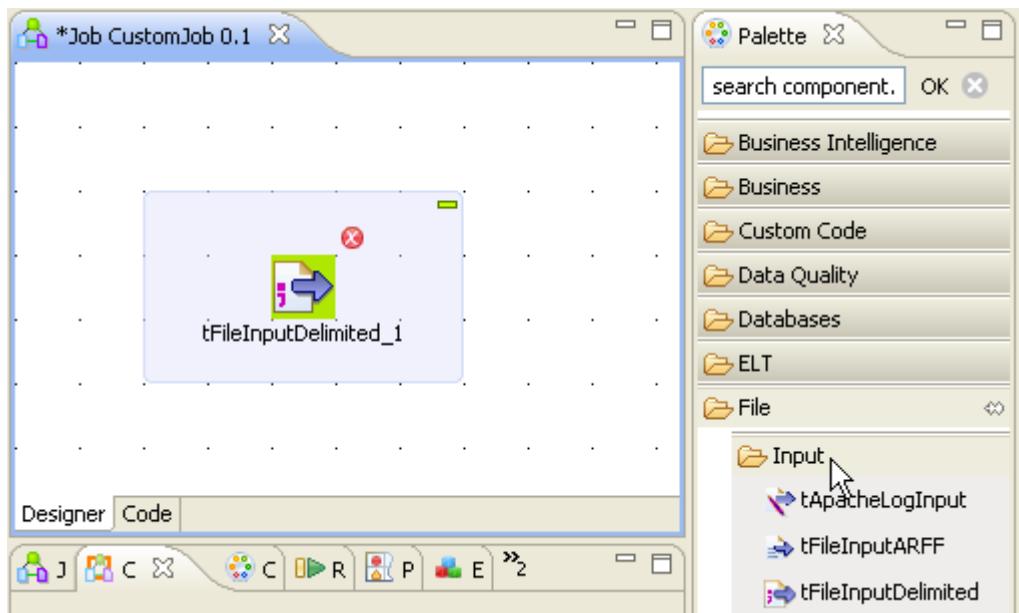


Si la **Palette** n'est pas visible, consultez *Afficher, cacher et déplacer la Palette, page 79*.



Vous pouvez également déposer une Note de la même façon que vous déposez des composants. Pour plus d'informations, consultez *Ajouter une Note au Job, page 121*.

Chaque composant ajouté au Job apparaît généralement dans un rectangle bleu symbolisant le sous-job.



Connectez les composants les uns aux autres dans un ordre logique à l'aide des connexions disponibles, afin de créer un Job ou un sous-job complet. Pour plus d'informations concernant les différents types de connexion, consultez *Types de connexions, page 83*.

Les Jobs ou sous-jobs sont mis en évidence par un seul rectangle bleu. Pour plus d'informations concernant le surlignage des Jobs et des sous-jobs, consultez *Personnaliser les sous-jobs, page 123*.

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 paramétrier toutes les propriétés de base (et parfois avancées) des composants à partir de la vue **Component**.



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 :

- *Types de connexions, page 83*
- *Gérer les icônes d'avertissement/d'erreur sur les composants, page 119*
- *Définition des propriétés d'un composant, page 65*

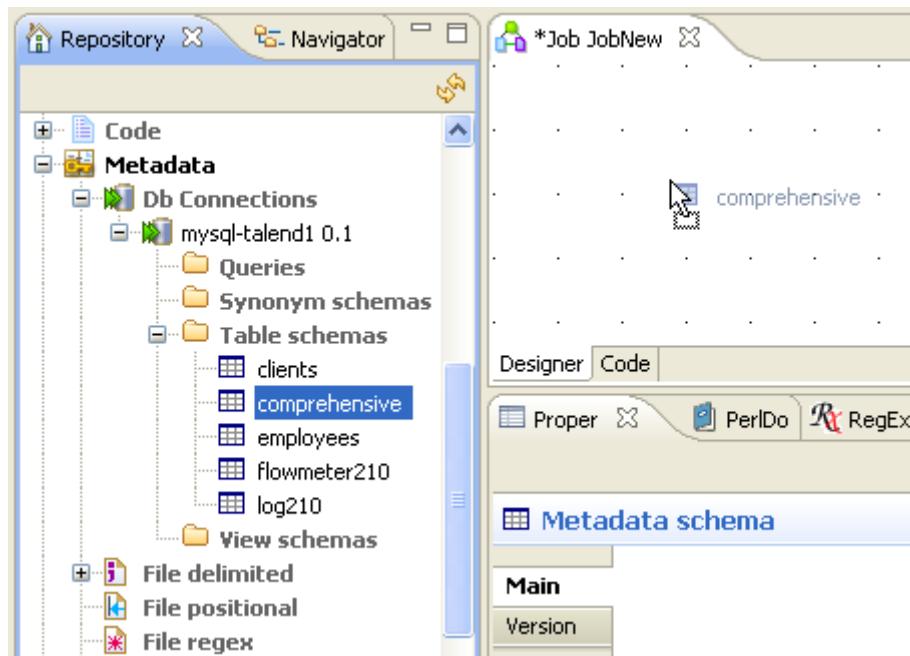
Déposer des composants à partir du répertoire Metadata

Si vous utilisez régulièrement les mêmes fichiers et les mêmes connexions aux bases de données pour plusieurs Jobs, nous vous conseillons de les sauvegarder dans le **Repository**, sous le noeud **Metadata**. Différents dossiers sous ce noeud regroupent les connexions créées, qu'elles soient des connexions aux bases de données, aux fichiers ou aux systèmes.

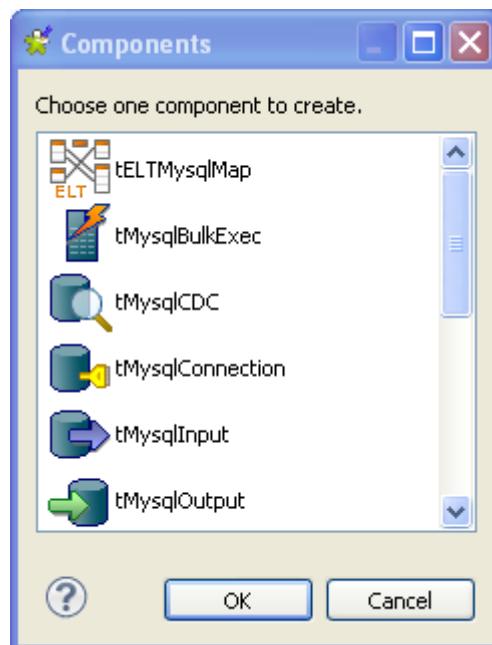
Différents assistants vous permettront de centraliser les connexions et les métadonnées des schémas, dans la vue Repository. Pour plus d'informations concernant les assistants de gestion des métadonnées, consultez *Centraliser les métadonnées, page 91*.

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** ainsi que le(s) dossier(s) où sont stockées les données que vous voulez utiliser.
- Déposez la connexion/le schéma choisi(e) dans l'éditeur graphique.



Une boîte de dialogue vous propose de sélectionner le composant à utiliser parmi la liste des composants disponibles.



- Sélectionnez le composant que vous souhaitez utiliser et cliquez sur **OK**.

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

- **Input** : Maintenez la touche **Alt** enfoncee pour déposer l'élément sélectionné dans l'éditeur graphique et ainsi l'ajouter à votre Job Design.
- **Output** : Maintenez la touche **Ctrl** enfoncee pour déposer le composant dans l'éditeur et ainsi l'ajouter à votre Job Design.

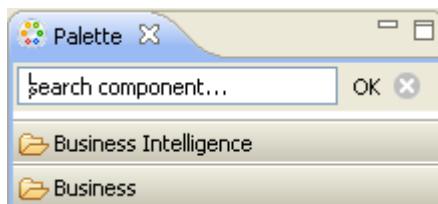
Si vous double-cliquez sur le composant, l'onglet **Component** affiche les détails des connexions sélectionnées ainsi que les informations du schéma sélectionné.



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

4.2.3 Chercher des composants dans la Palette

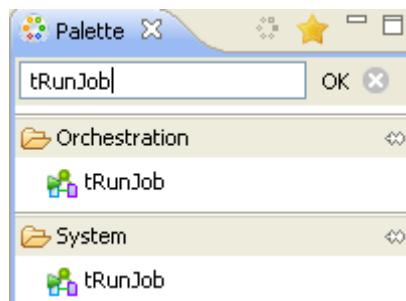
Si vous ne voulez pas parcourir les familles de composants dans la **Palette** pour trouver les composants que vous souhaitez utiliser dans votre Job, vous pouvez chercher le composant directement dans le champ de recherche en haut de la **Palette**.



Pour chercher un composant, procédez comme suit :

- Cliquez sur le bouton pour effacer tout texte du champ de recherche.
- Saisissez le nom du composant que vous cherchez puis cliquez sur **OK**.

La **Palette** n'affiche plus que la (les) famille(s) contenant le composant.



Pour retourner à la **Palette** par défaut, cliquez à nouveau sur le bouton.

4.2.4 Connecter des composants

Un Job ou un sous-job est composé par un groupe de composants interconnectés de manière logique. Un Job peut être composé de plusieurs sous-jobs chargés de différents traitements de données.

Le composant formant un sous-job, ainsi que les sous-jobs sont connectés entre eux via différents types de connexion.

De plus, un Job (composé d'un ou plusieurs sous-jobs) peut être précédé par un Pré-Job et suivit d'un Post-Job, afin de s'assurer que certaines tâches spécifiques (souvent sans lien avec le traitement

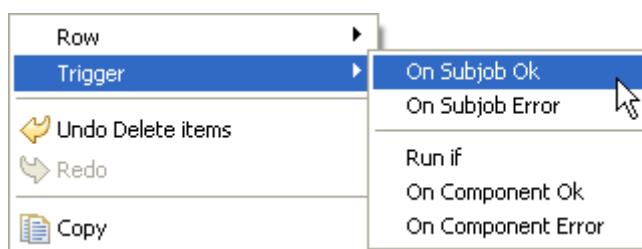
de données en cours) sont effectuées un tout premier ou en tout dernier. Pour plus d'informations, consultez *Utiliser les composants tPrejob et tPostjob, page 115*.

Pour connecter deux composants, cliquez-droit sur le composant source dans votre espace de modélisation graphique, sélectionnez le type de connexion dans le menu contextuel, puis cliquez sur le composant cible.



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 signe interdit ne disparaît qu'une fois le composant valide atteint.

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.

Les sections ci-dessous décrivent tous les types de connexions disponibles.

4.2.5 Définition des propriétés d'un composant

Les propriétés de chacun des composants formant un **Job** ou un **Sous-job** permettent de paramétrier 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.

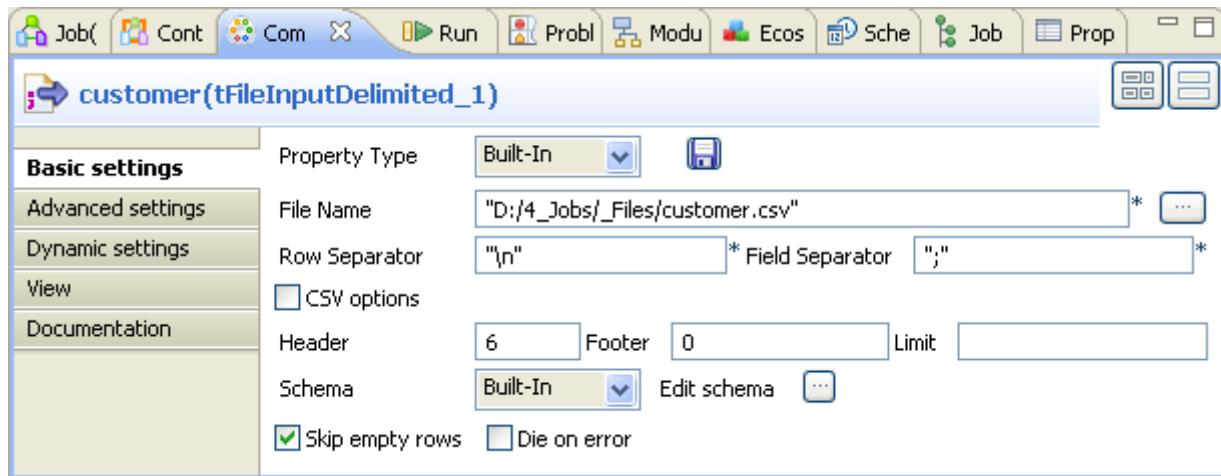
Pour plus d'informations concernant les composants présents dans la **Palette**, consultez le **Guide de référence** des composants **Talend Open Studio**.

Onglet Basic Settings

L'onglet **Basic settings** fait partie de la vue **Component**, qui est située en bas de la fenêtre de **Talend Open Studio**.

Conception d'un Job d'intégration de données

Utiliser un Job Design simple



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 le **Guide de référence** des composants **Talend Open Studio**.



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 paramétrier 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 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.

Selectionnez **Repository** dans le champ **Property type** et désignez les métadonnées comportant les paramètres appropriés. Voir également : *Centraliser les métadonnées, page 91*.

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

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

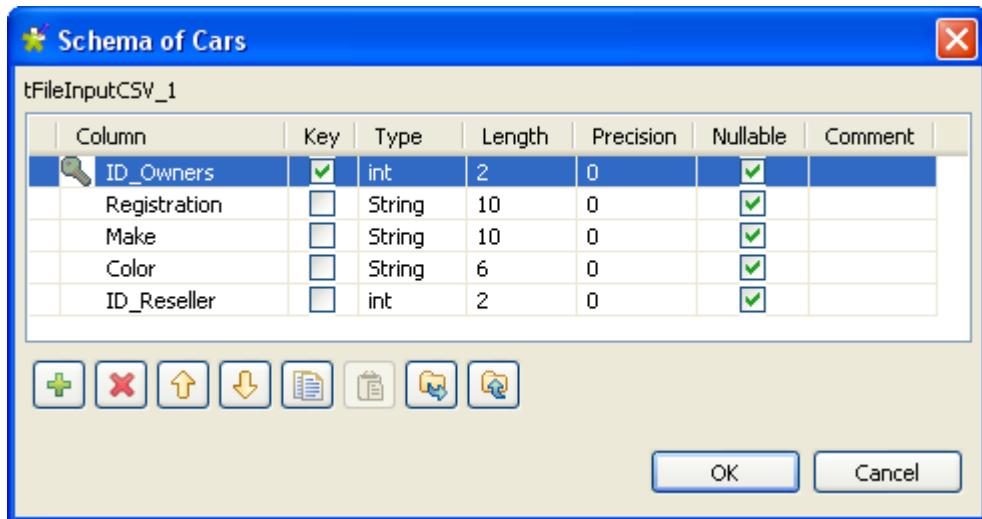
- Cliquez sur l'icône de disquette. L'assistant de création de métadonnée correspondant à votre composant s'ouvre.
- Suivez les étapes de l'assistant. Pour plus d'informations sur la création de métadonnées, consultez *Gestion des métadonnées, page 189*.
- La métadonnée apparaît alors sous le noeud **Metadata du Repository**.

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 créées. Vous trouverez une description détaillée d'un schéma dans la section suivante.

Paramétrer 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.



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>

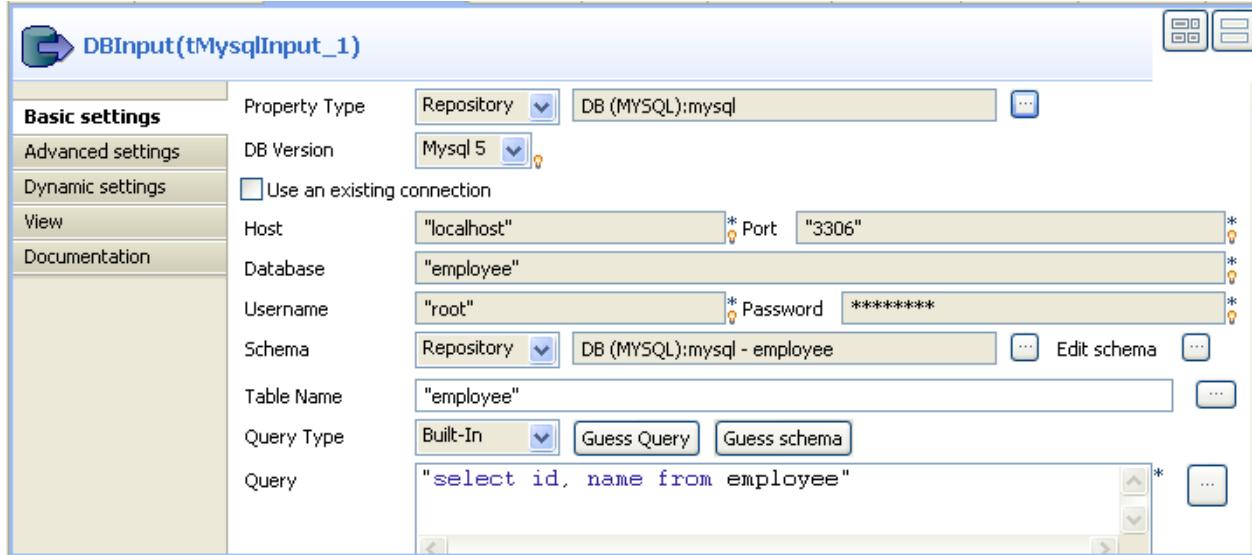
Paramétrer un schéma du Repository

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

Pour rappeler un fichier de métadonnées dans votre job, sélectionnez **Repository** dans la liste déroulante **Schema type** et sélectionnez le fichier approprié dans la liste **Repository**. Ou bien, cliquez sur la métadonnée sous le noeud **Metadata** du **Repository** et glissez-la jusqu'au composant que vous avez préalablement déposé dans l'espace de modélisation. Puis cliquez sur **Edit Schema** pour vérifier que les données sont appropriées.

Conception d'un Job d'intégration de données

Utiliser un Job Design simple



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



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. Pour cela, cliquez-droit sur le noeud correspondant et sélectionnez l'option correspondante (**Edit connection** ou **Edit file**) dans le menu contextuel.

Voir également : *Centraliser les métadonnées, page 91*

Afficher un champ de manière dynamique (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.

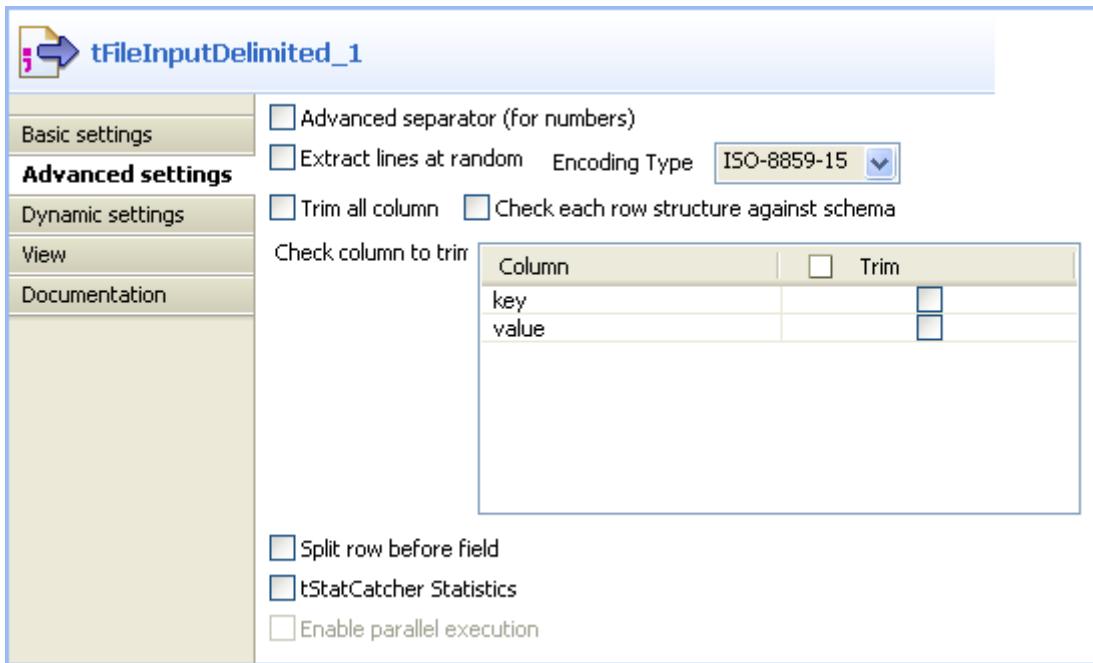
Description: Error Message Global variable, property of component tMap [tMap_1]. Type: String Availability: After Variable Name: ((String)globalMap.get("tMap_1_ERROR_MESSAGE"))	tFileInputDelimited_2.ERROR_MESSAGE tFileInputDelimited_2.NB_LINE tMap_1.ERROR_MESSAGE tFileOutputDelimited_1.ERROR_MESSAGE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

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 : *Centraliser les Contextes et les Variables, page 91*

Onglet 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.

Mesurer les flux de données

L'onglet **Advanced settings** propose aussi les fonctionnalités de statistiques avec l'option **tStatCatcher**. Si vous sélectionnez cette option, elle vous permet d'afficher les statistiques et les données de log sur le Job en cours sans avoir recours aux composants dédiés. Pour plus d'informations concernant les fonctionnalités Stats & Log, consultez *Automatiser l'utilisation de statistiques & logs, page 124*.

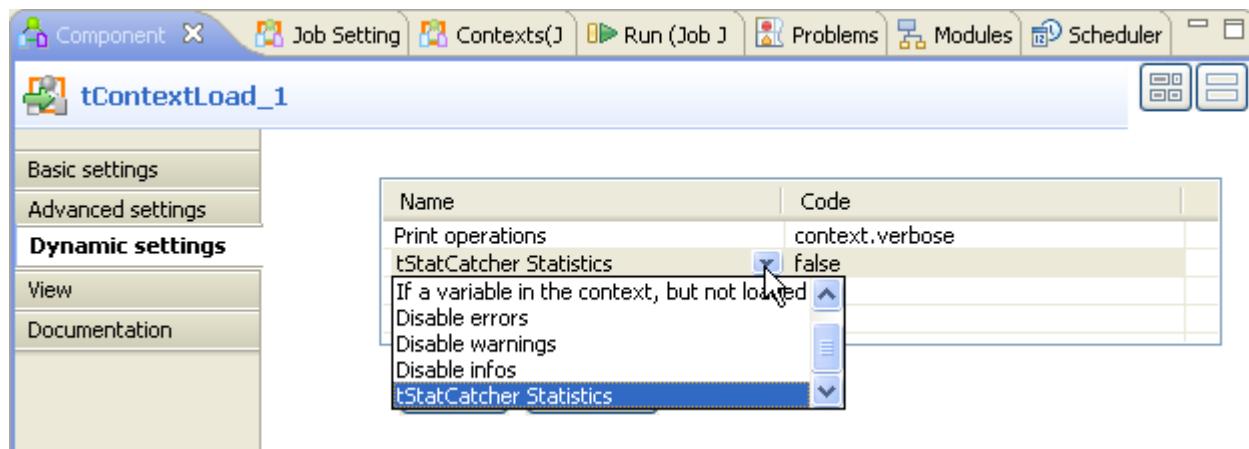
Onglet Dynamic settings

Les vues **Basic** et **Advanced settings** de tous les composants comportent divers cases à cocher et listes déroulantes permettant de paramétriser chaque composant. D'une manière générale, les valeurs de ces types de paramètres sont soit true (vrai), soit false (faux) et ne peuvent être éditées qu'au moment de la création.

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

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

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



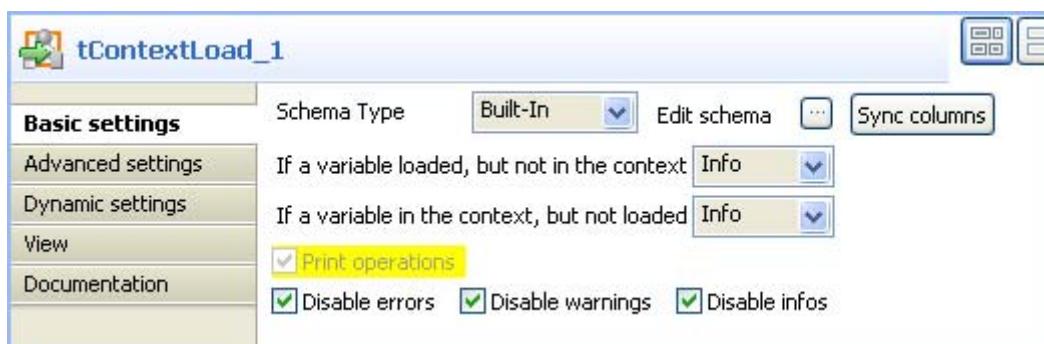
Pour personnaliser ces types de paramètres, notamment les variables de contexte, suivez les étapes qui suivent :

- Sélectionnez le composant dont les onglets **Basic** et **Advanced settings** contiennent les paramètres que vous souhaitez définir comme variable.
- Cliquez sur l'onglet **Dynamic settings**
- Cliquez sur le bouton **[+]** pour créer une nouvelle ligne de paramètres dans le tableau.
- Cliquez dans la cellule **Name** du paramètre pour afficher la liste des paramètres disponibles. Par exemple : *Print operations*.
- Puis cliquez dans la cellule **Code** en face du paramètre et paramétrez le code à utiliser. Par exemple : *context.verbose* si vous avez créé la variable de contexte que vous avez appelée *verbose*.



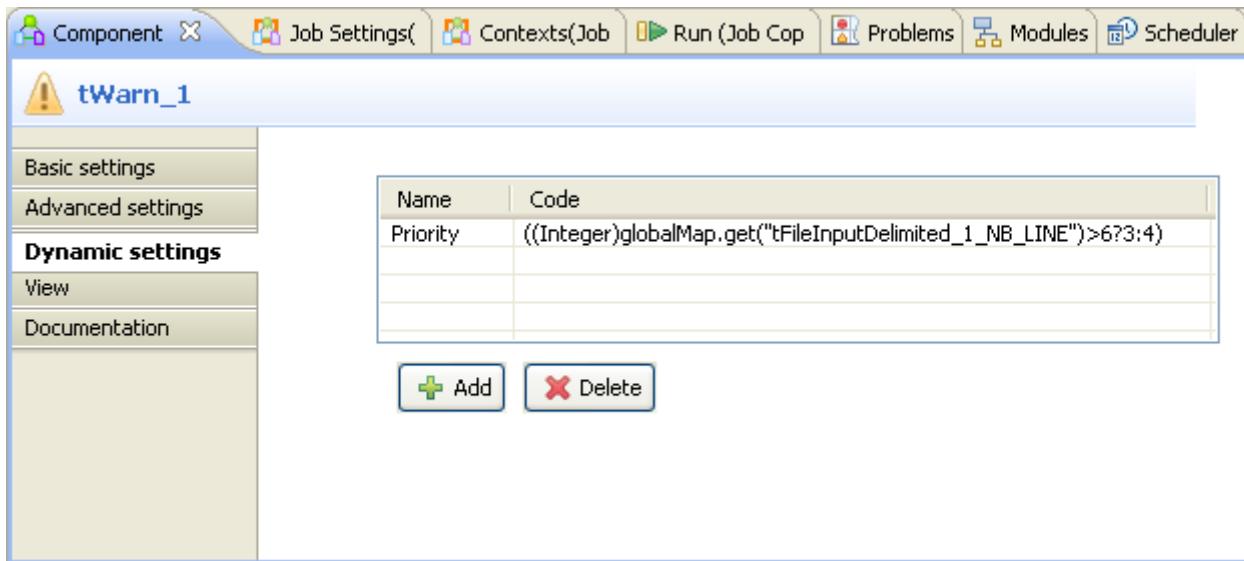
Dans la cellule **Code**, vous pouvez entrer une variable de contexte mais aussi un bout de code (en Java ou en Perl en fonction du langage utilisé dans votre projet).

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



 Si vous souhaitez configurer un paramètre comme variable de contexte, assurez-vous d'avoir bien créé la variable dans la vue **Context**. Pour plus d'informations concernant la définition des variables de contexte, consultez *Définir les variables à partir de la vue Contexts, page 92*

Vous pouvez aussi utiliser une variable globale ou des bouts de code Java ou Perl, pour stocker les valeurs à utiliser pour chaque paramètre.



Par exemple, utilisez certaines variables globales disponibles à l'aide du raccourci clavier **Ctrl+Espace**, et adaptez-les à votre contexte.

Onglet View

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

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.

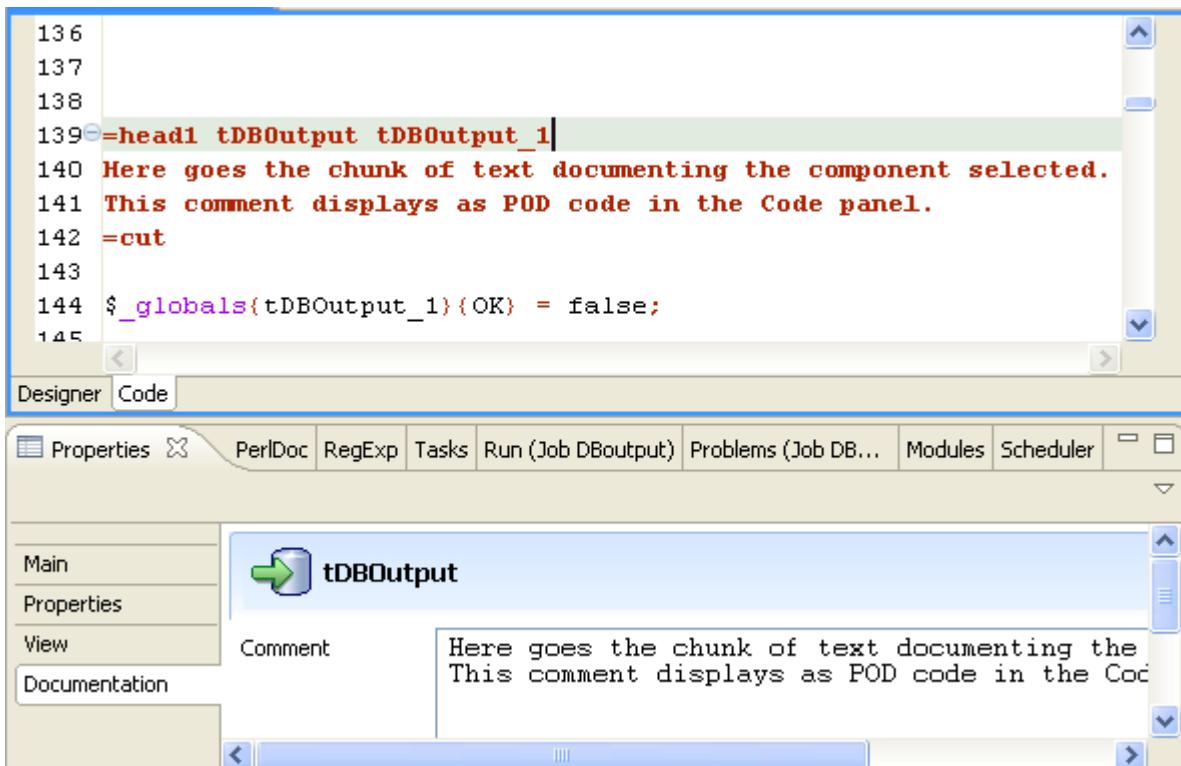
Vous pouvez personnaliser les textes des **Label** et **Hint** à l'aide des balises HTML suivantes :

- 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 > Appearance > Designer**.

Onglet Documentation

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

4.2.6 Exécution d'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 un Job en mode Java debug, page 73*.

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 72*.

Exécuter en mode normal



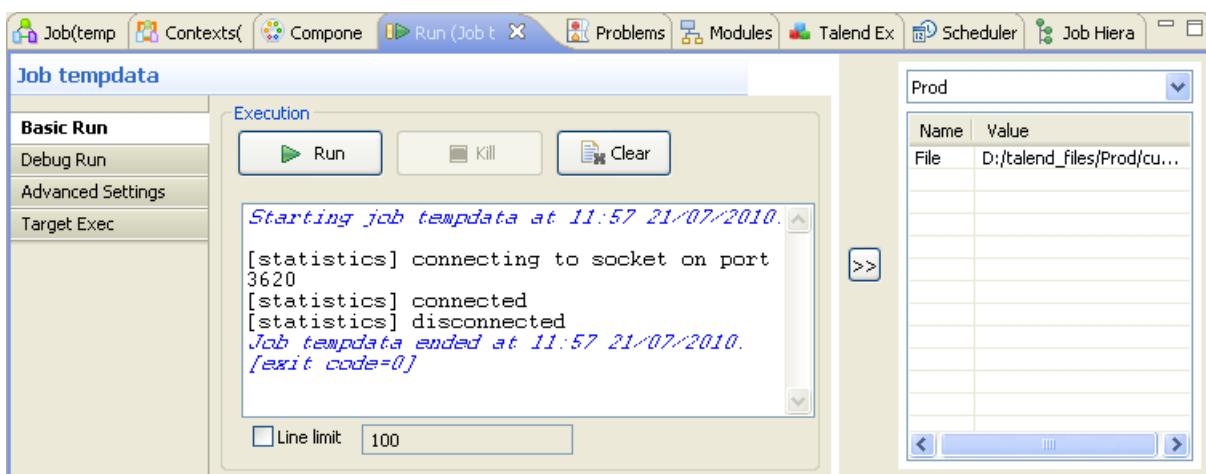
Veillez à sauvegarder votre Job avant de l'exécuter, afin que toutes les propriétés puissent être prises en compte.

Pour exécuter votre Job en mode normal, procédez comme suit :

- 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 variables à partir de la vue Contexts, page 92.*

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



Avant d'exécuter à nouveau un Job, vous pouvez vider le contenu de la vue de log, en cliquant sur la case à cocher **Clear before run**. La vue de 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. Pour plus d'informations, consultez les sections suivantes.

Exécuter un Job en mode Java debug

Pour suivre pas à pas les étapes de l'exécution d'un Job afin d'identifier les bugs possibles, vous pouvez exécuter ce Job en mode Debug.

Pour accéder au mode Debug :

- Cliquez sur l'onglet **Run** pour accéder à cette vue.
- Cliquez sur l'onglet **Debug Run** pour accéder aux modes d'exécution Debug.



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

Conception d'un Job d'intégration de données

Utiliser un Job Design simple

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.

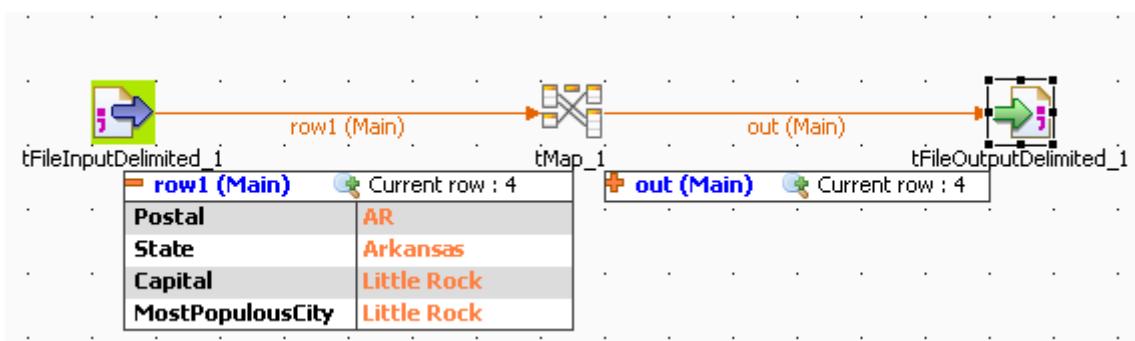
Vous pouvez exécuter le Job pas à pas et vérifier le comportement de chacun des composants ainsi que les valeurs des variables utilisées.

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

Exécuter un Job en mode Traces Debug

La fonctionnalité **Traces** permet un suivi du traitement des données au cours de l'exécution du Job dans **Talend Open Studio**.

Cette option fournit un aperçu ligne par ligne du comportement du composant et affiche le résultat dynamique de cette vue à côté de la connexion **Row**.



Cette fonctionnalité vous permet de surveiller tous les composants d'un Job, sans avoir besoin de passer en mode Debug, et par conséquent sans avoir besoin de connaissances Perl/Java particulières.

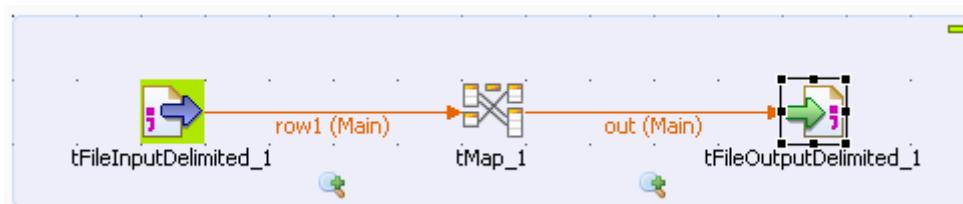
La fonction **Traces** affiche le contenu des lignes traitées dans un tableau.



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

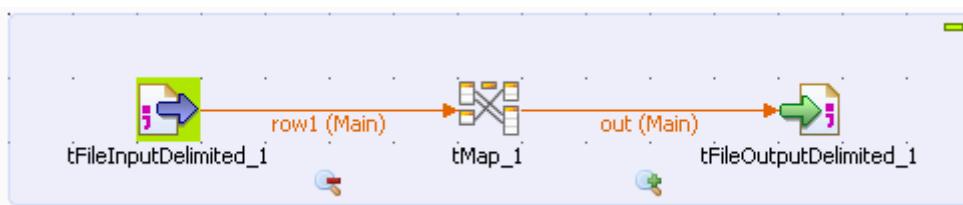
Vous pouvez activer ou désactiver le mode **Traces** ou décider quelles colonnes traitées afficher dans le tableau des traces qui s'affichera dans l'espace de modélisation graphique lors de l'exécution du Job.

Pour activer le mode **Traces** dans un Job :



- Cliquez sur la vue **Run**.
- Cliquez sur l'onglet **Debug Run** pour accéder aux modes d'exécution **Debug** et **Traces**.
- Cliquez sur la flèche descendante du bouton **Java Debug** puis sélectionnez l'option **Traces**. Une icône apparaît sous chaque flux de votre Job pour indiquer que la fonction de suivi du traitement est activée.

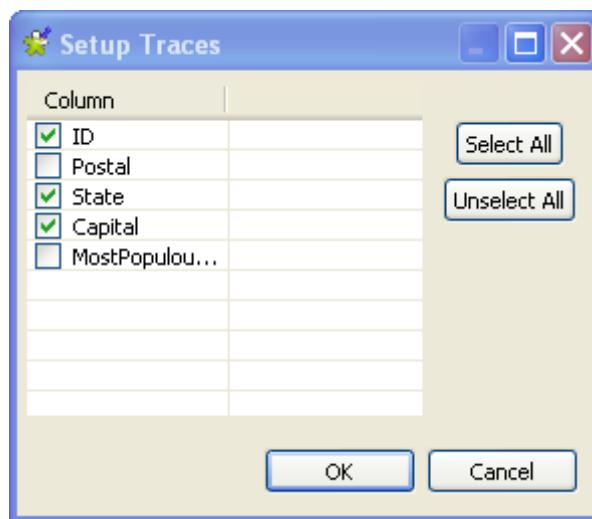
Pour désactiver le mode **Traces** d'un flux de votre Job :



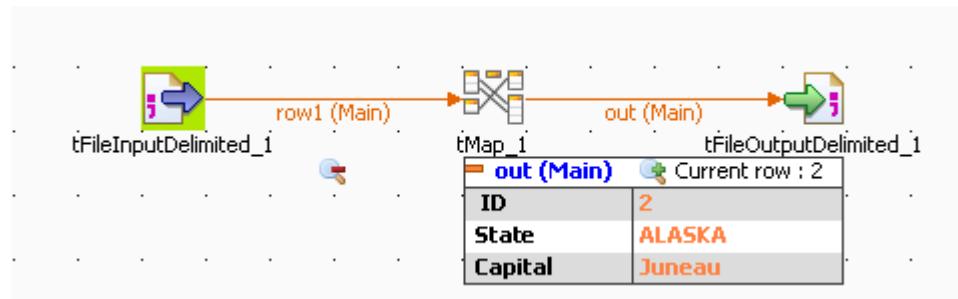
- Cliquez-droit sur l'icône de **Traces** du flux concerné et sélectionnez **Disable Traces** dans le menu contextuel. Un signe moins rouge vient remplacer le signe plus vert sur l'icône pour indiquer que le mode **Traces** a été désactivé pour ce flux.

Pour choisir quelles colonnes des données traitées afficher dans le tableau des traces, procédez comme suit :

- Cliquez-droit sur l'icône de **Traces** du flux concerné et sélectionnez **Setup Traces** dans le menu contextuel. La boîte de dialogue **[Setup Traces]** s'ouvre.



- Dans la boîte de dialogue, décochez les cases correspondant aux colonnes que vous ne souhaitez pas afficher dans le tableau Traces.
- Cliquez sur **OK** pour fermer la boîte de dialogue.



La fonction Traces ne s'exécute qu'à l'exécution du Job et s'arrête à la fin de celui-ci.

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

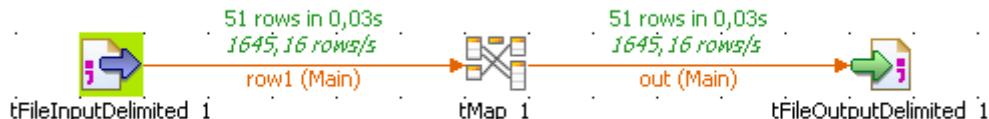
Configurer les paramètres d'exécution avancés

Plusieurs paramètres d'exécution avancés sont disponibles pour rendre l'exécution des Jobs plus pratique :

- **Statistics**, cette fonction affiche le taux de performance de traitement. Pour plus d'informations, consultez *Afficher les Statistiques, page 76*.
- **Exec time**, cette fonction affiche le temps d'exécution dans la console, à la fin de l'exécution. Pour plus d'informations, consultez *Afficher la durée d'exécution et d'autres options, page 77*.
- **Save Job before execution**, cette fonction permet de sauvegarder automatiquement le Job avant son exécution.
- **Clear before run**, cette fonction efface tous les résultats de l'exécution précédente avant de ré-exécuter le Job.
- **JVM Setting**, cette fonction vous permet de définir les paramètres de votre JVM selon vos besoins, par exemple les paramètres utilisés pour afficher des caractères spéciaux.

Afficher les Statistiques

La fonction **Statistics** affiche pour chacun des composants son taux de performance, en dessous des connexions dans l'espace de modélisation.



Sont indiqués le nombre de lignes traitées et la vitesse de traitement en ligne par seconde. Vous pouvez ainsi plus facilement repérer d'éventuels goulets d'étranglement dans le traitement de votre flux de données.

Pour les liens de déclenchement de type **If**, **OnComponentOk**, **OnComponentError**, **OnSubjobOk** et **OnSubjobError**, l'option **Statistics** affiche l'état de ce déclenchement durant l'exécution de votre Job : **Ok** ou **Error** et **True** ou **False**.



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.

Dans la vue **Run**, 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.



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.

Vous pouvez également sauvegarder votre Job avant le début de l'exécution. Cochez la case de l'option correspondante, **Save Job before execution**.

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**.

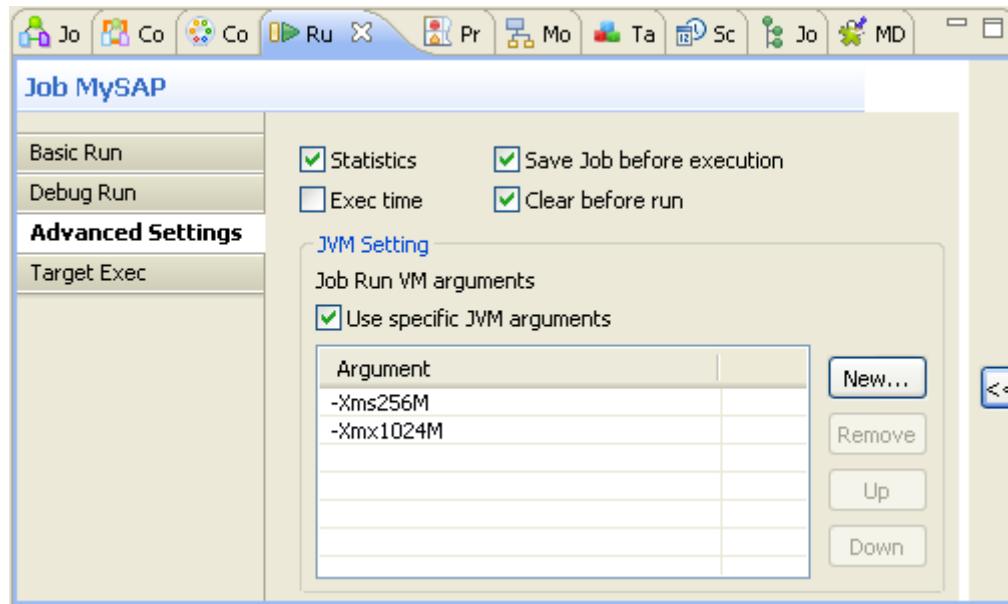
Vous pouvez aussi enregistrer votre Job avant que l'exécution ne commence, en cochant la case **Save Job before run**.

Afficher des caractères spéciaux dans la console

Talend Open Studio peut afficher des caractères spéciaux dans la console, comme les caractères chinois, japonais ou coréens, par exemple. Pour les afficher, procédez comme suit, avant d'exécuter le Job :

Conception d'un Job d'intégration de données

Utiliser un Job Design simple



- Cliquez sur l'onglet **Advanced settings**.
- Dans la zone **JVM settings**, cochez la case **Use specific JVM arguments** afin d'activer le tableau **Argument**.
- A côté du tableau **Argument**, cliquez sur le bouton **New...**, afin d'ouvrir la boîte de dialogue **[Set the VM argument]**.
- Dans la boîte de dialogue, saisissez `-Dfile.encoding=UTF-8`.
- Cliquez sur **OK** pour fermer la boîte de dialogue.

Cet argument peut être appliqué à toutes vos exécutions de Jobs dans **Talend Open Studio**. Pour plus d'informations concernant l'application d'un argument JVM à toutes les exécutions de Jobs, consultez *Préférences d'exécution et de débogage, page 23*.

4.2.7 Personnaliser l'espace de travail

Lorsque vous utilisez **Talend Open Studio** pour créer un Job d'intégration de données, vous pouvez personnaliser la disposition et les paramètres de la **Palette** selon vos besoins. Vous pouvez également modifier la position de tous les onglets existants dans le Studio afin de répondre à vos besoins.

Modifier la disposition et les paramètres de la Palette

La **Palette** contient tous les composants techniques de base nécessaire à la création de Jobs complexes dans l'espace de modélisation graphique. Ces composants sont regroupés par familles et sous-familles.

Pour la configuration d'un composant spécifique, consultez le Guide de Référence des **Composants Talend Open Studio**.

Talend Open Studio vous permet de changer la disposition et la position de votre **Palette** selon vos besoins. Les sections suivantes expliquent toutes les options de gestion disponibles pour la **Palette**.

Afficher, cacher et déplacer la Palette

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** peut être cachée sur le côté droit de l'espace de modélisation.



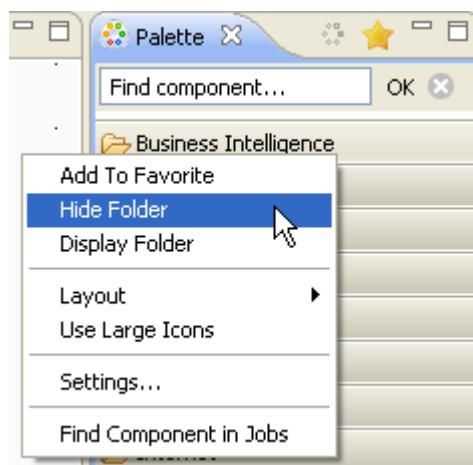
Si vous souhaitez que la **Palette** soit visible en permanence, cliquez sur la flèche de gauche dans le coin supérieur droit de l'éditeur graphique.

Pour connaître les propriétés spécifiques à chacun des composants, consultez le **Guide de référence** des composants [Talend Open Studio](#).

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**.

Afficher/masquer les familles de composants

Vous pouvez masquer ou afficher les familles de composants selon vos besoins, dans un souci de visibilité, par exemple. Pour ce faire, cliquez-droit sur la **Palette** et sélectionnez l'option **Display folder** pour afficher les dossiers et **Hide folder** pour masquer les dossiers.



L'option display/hide (afficher/masquer) peut être très utile lorsque vous êtes dans la vue **Favorite** de la **Palette**. Dans cette vue, vous avez généralement un nombre limité de composants. Si vous les affichez sans les familles, ils seront affichés par ordre alphabétique pour faciliter leur utilisation. Pour plus d'informations concernant la **Palette** favorite, consultez *Configurer la Palette favorite, page 80*.

Maintenir ouverte une famille de composants

Si vous utilisez souvent une ou plusieurs famille(s) de composants, vous pouvez ajouter une punaise sur leur nom pour les empêcher de se réduire lorsque vous sélectionnez des composants d'autres familles.



Pour ajouter une punaise, cliquez sur l'icône de punaise dans l'angle en haut à droite du nom de la famille.

Filtrer la Palette

Vous pouvez sélectionner les composants à afficher ou à masquer dans votre **Palette**. Vous pouvez également ajouter à la **Palette** les composants que vous avez développé vous-même.

Pour plus d'informations sur comment filtrer la **Palette**, consultez *Personnaliser la Palette, page 32*.

Pour plus d'informations concernant l'ajout de composants à la **Palette**, à partir de **Talend Exchange** ou développés par vous-même, consultez *Télécharger des composants de la communauté, page 110* et/ou *Composants externes ou utilisateur, page 22*.

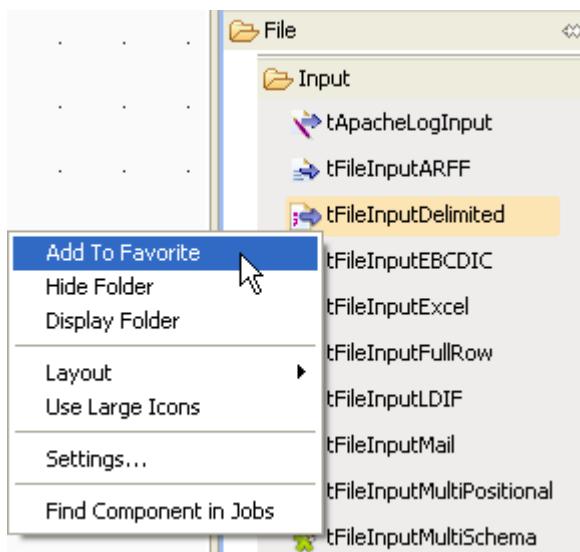
Configurer la Palette favorite

La **Palette** offre des fonctionnalités de recherche et de favoris facilitant son utilisation.

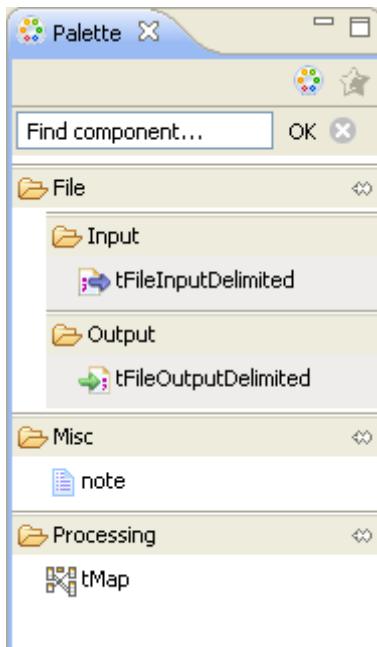
Vous pouvez ajouter et retirer des composants à votre **Palette** favorite de **Talend Open Studio**, afin d'accéder plus rapidement aux composants que vous utilisez le plus souvent.

Pour ce faire :

- Dans la **Palette**, cliquez-droit sur le composant que vous souhaitez ajouter à vos favoris et sélectionnez l'option **Add To Favorite**.



- Répétez cette action pour tous les composants que vous souhaitez ajouter à votre **Palette** favorite, puis cliquez sur le bouton **Favorite** en haut de la **Palette** pour afficher la **Palette** favorite.



Seuls les composants ajoutés aux Favoris apparaissent.

Pour retirer un composant de la **Palette** favorite :

- Cliquez-droit sur le composant et sélectionnez **Remove From Favorite**.

Pour retourner à la **Palette** standard, cliquez sur le bouton **Standard** en haut de la **Palette**.

Modifier la disposition des composants dans la Palette

Vous pouvez modifier la disposition de la liste des composants dans la **Palette** pour les afficher en colonnes ou en listes, avec seulement une icône ou une icône avec une courte description.

Vous pouvez également agrandir l'icône des composants pour une meilleure lisibilité de la liste des composants.

Pour ce faire, cliquez-droit sur une famille de composants dans la **Palette** et sélectionnez l'option désirée dans le menu contextuel ou cliquez sur **Settings** pour ouvrir la fenêtre [**Palette Settings**] et personnaliser la disposition.

Ajouter des composants externes à la Palette

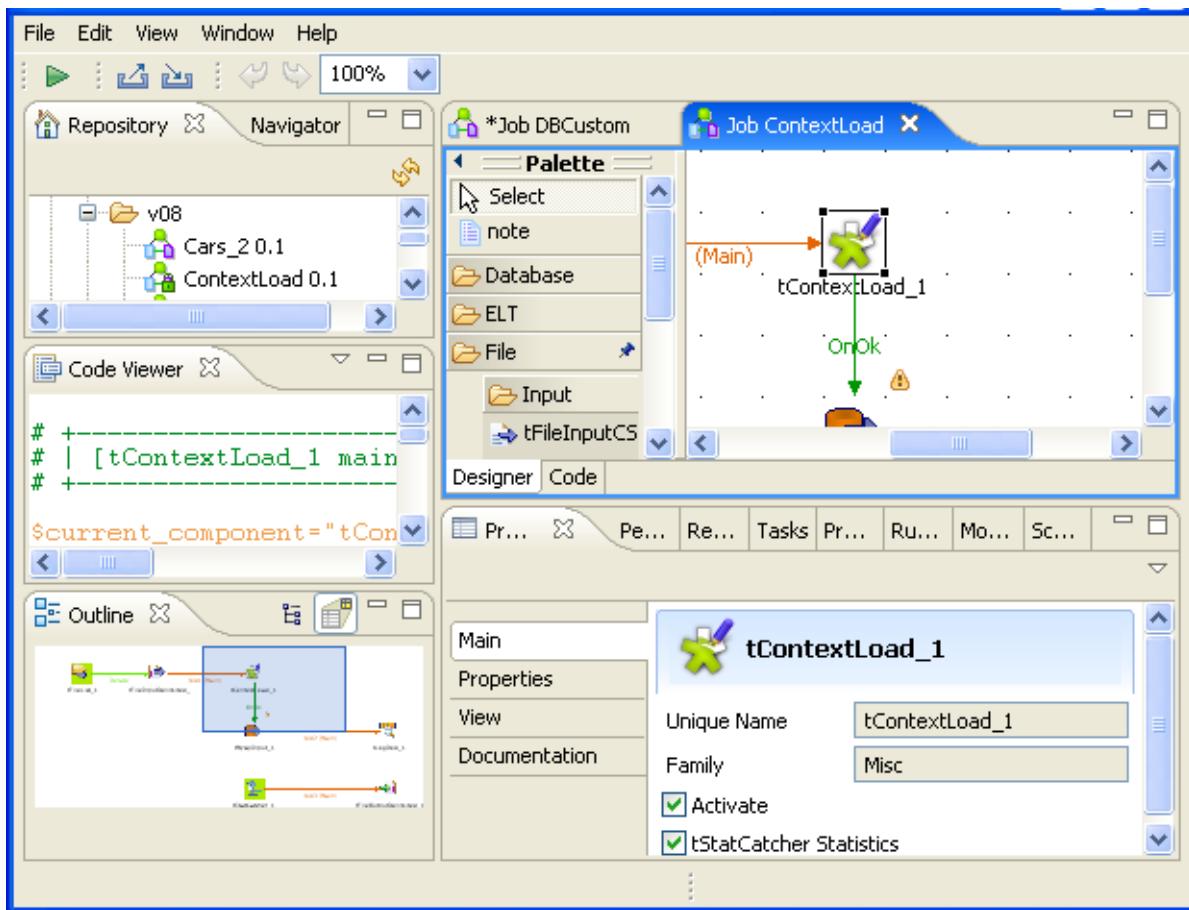
Talend Open Studio vous permet d'ajouter des composants externes à la **Palette** de votre Studio et des utiliser dans vos Jobs.

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

Pour plus d'informations concernant le téléchargement de composants utilisateur dans votre Studio, consultez *Composants externes ou utilisateur, page 22*.

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 l'icône minimiser/maximiser (pour réduire ou agrandir le panneau correspondant. Pour plus d'informations sur comment afficher ou masquer un panneau ou une vue, consultez *Afficher les onglets/vues de configuration des Jobs, page 82*.

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 300*.

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 de **Talend Open Studio**.

Afficher les onglets/vues de configuration des Jobs

Les onglets de configuration sont situés dans la partie inférieure de l'espace de modélisation graphique. Chaque onglet ouvre une vue affichant des informations précises concernant l'élément sélectionné dans l'espace de modélisation graphique.



Les onglets **Component**, **Run Job**, **Problems** et **Error Log** rassemblent toutes les informations relatives aux éléments graphiques sélectionnés dans l'espace de modélisation ou à l'exécution du Job ouvert.

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

Certains onglets de configuration sont masqués par défaut, comme les onglets **Error Log**, **Job Hierarchy**, **Modules** et **Scheduler**. Vous pouvez afficher les onglets masqués dans la même zone que les autres, et ouvrir directement la vue correspondante si vous sélectionnez **Window > Show view**, puis, dans la boîte de dialogue, développez le noeud correspondant et sélectionnez l'élément que vous souhaitez afficher.

Pour une description détaillée concernant ces onglets, consultez *Onglets de configuration, page 297*.

4.3 Utiliser les connexions

Dans **Talend Open Studio**, un Job ou un sous-job comprend un groupe de composants reliés logiquement les uns aux autres via des connexions. Cette section décrit les types de connexions et les paramètres correspondants.

4.3.1 Types de connexions

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

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

Les sections suivantes décrivent tous les types de connexions disponibles.

Connexion de type Row

La connexion de type **Row** manipule les données elles-mêmes. Les liens de type **Row** sont **Main**, **Lookup**, **Reject** ou **Output** selon la nature du flux de données traité.

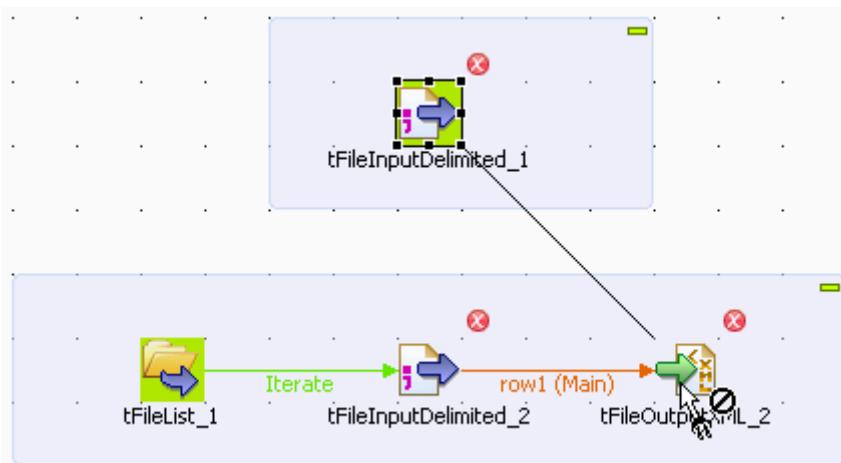
Main

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.



Vous ne pouvez pas connecter deux composants d'entrée à l'aide d'une connexion de type **main Row**. *Une seule* connexion entrante de type **Row** est possible par composant. La seconde connexion **Row** se transforme en connexion **Lookup** (de référence) automatiquement.

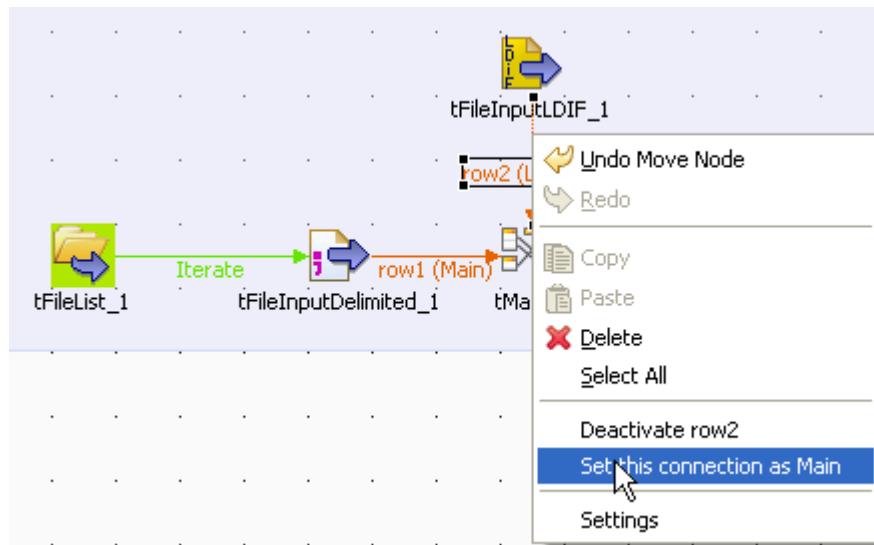


- Cliquez droit sur le composant d'entrée et sélectionnez **Row > Main** dans la liste des connexions.
- Sinon, cliquez sur le composant pour le sélectionner, puis, tout en cliquant droit, déplacez votre curseur vers le composant de destination. Une connexion de type **Row > Main** sera automatiquement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

Pour en savoir plus sur les cas spécifiques de connexions **Row** multiples, consultez *Job à entrées/sorties multiples, page 86*.

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 86.*

Filter

La connexion de type **Filter** relie le composant spécifique **tFilterRow** à un composant de sortie. Cette connexion Row regroupe les données répondant aux critères du filtre. Ce composant particulier permet aussi une connexion de Rejet 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 inappropriate, valeur Null non définie, etc.). Pour certains composants, ce lien est activé lorsque l'option **Die on error** est désactivée. Pour plus d'informations, référez-vous aux propriétés du composant correspondant, disponibles dans le **Guide de Référence Talend Open Studio**.

ErrorReject

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

Consultez également : *Gestion des erreurs, page 181.*

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.



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 resaisir toutes les informations de propriétés.

Voir également : *Job à entrées/sorties multiples, page 86.*

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.

Job à entrées/sorties multiples

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

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

Pour plus d'informations concernant le mapping et la transformation de données, consultez *Mapper des flux de données, page 105.*

Pour plus d'informations concernant le composant **tMap** et son utilisation, consultez le composant *tMap* du **Guide de référence des composants Talend Open Studio**.

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 d'un Job ou d'un sous-job.

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**. Pour plus d'informations concernant la configuration d'une connexion **Iterate**, consultez *Paramètres d'une connexion Iterate, page 89.*

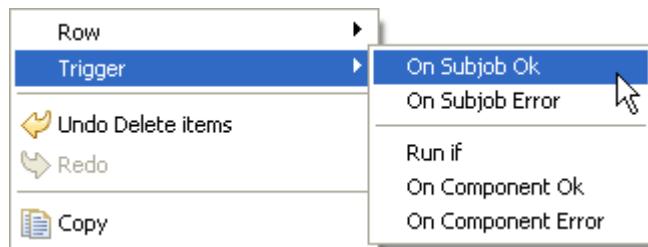


A la différence des autres types de connexions, le nom du lien **Iterate** est en lecture seule.

Connexion de type Trigger

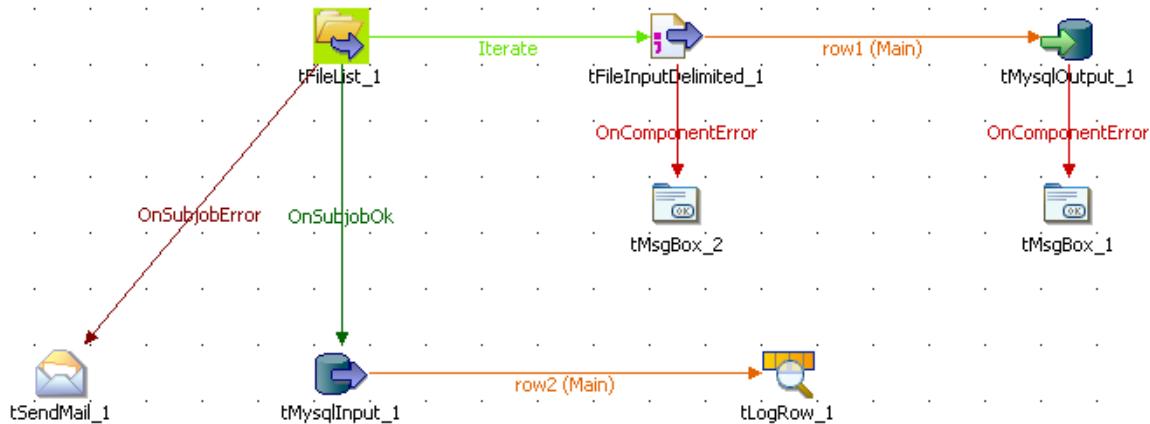
Les connexions de déclenchement, **Trigger**, aident à définir le processus de traitement. Les connexions de type **Trigger** ne servent pas au transfert de données.

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



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

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



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

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

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

Voir également : *Définition du composant Start, page 118.*

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

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

Run if déclenche un sous-job ou un composant si les conditions définies sont réunies.

Pour plus d'informations concernant la configuration d'une connexion Trigger,
consultez *Paramètres d'une connexion Trigger, page 90*

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 le **Guide de référence** des composants [Talend Open Studio](#).

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**.



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 générée par le composant ELT **tMap**, un même nom ne devrait jamais être utilisé deux fois.

4.3.2 Définir les paramètres des connexions

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

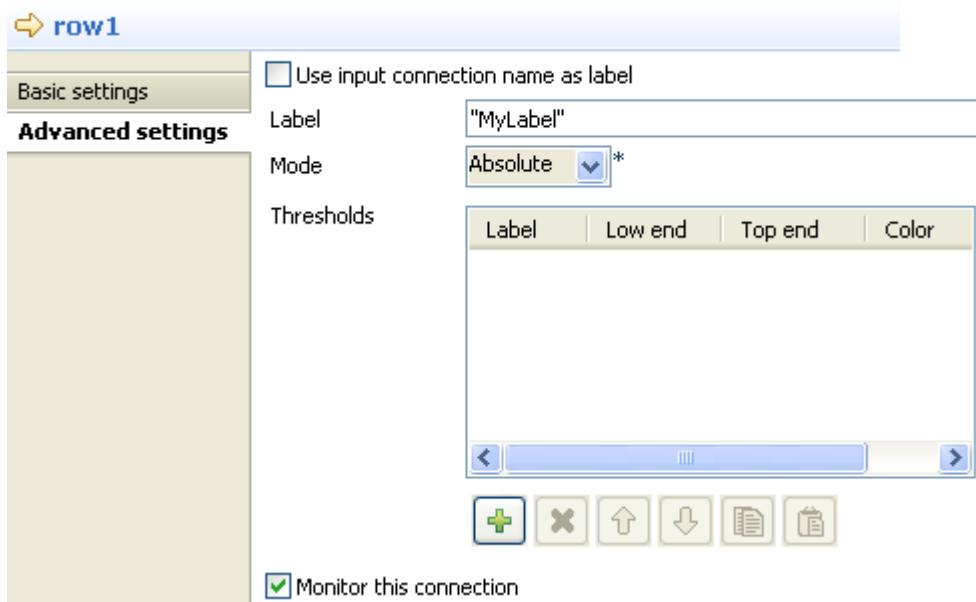
Paramètres d'une connexion Row

L'onglet **Basic settings** de la vue **Component** de la connexion affiche le schéma du flux de données géré par la connexion. Vous pouvez modifier le schéma en cliquant sur le bouton **Edit schema**. Une fois le schéma du flux de données modifié, le schéma type des deux composants aux deux extrémités de la connexion devient **Built-In**. Pour plus d'informations, consultez *Paramétriser un schéma built-in, page 67*.

Column	Key	T...	N...	Date...	L...	P...	D...	C...
id	<input checked="" type="checkbox"/>	In...	<input checked="" type="checkbox"/>		2	0		
Customer...	<input type="checkbox"/>	St...	<input checked="" type="checkbox"/>		255	0		
Customer...	<input type="checkbox"/>	St...	<input checked="" type="checkbox"/>		255	0		
idState	<input type="checkbox"/>	In...	<input checked="" type="checkbox"/>		2	0		
id2	<input type="checkbox"/>	In...	<input checked="" type="checkbox"/>		2	0		
RegTime	<input type="checkbox"/>	St...	<input checked="" type="checkbox"/>		50	0		
Register...	<input type="checkbox"/>	St...	<input checked="" type="checkbox"/>		50	0		

L'onglet **Advanced settings** vous permet de monitorer le flux de données d'une connexion dans un Job, sans avoir à utiliser un composant **tFlowMeter**. les informations mesurées seront

interprétées et affichées dans un outil de supervision, comme [Talend Activity Monitoring Console](#). Pour plus d'informations concernant [Talend Activity Monitoring Console](#), consultez le [Guide Utilisateur Talend Activity Monitoring Console](#).



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

- Cochez la case **Monitor this connection**.
- Cochez la case **Use input connection name as label** afin d'utiliser le nom du flux d'entrée comme libellé des données à enregistrer dans le log, ou saisissez le libellé dans le champ **Label**.
- Dans la liste **Mode**, sélectionnez **Absolute** pour enregistrer dans le log le nombre de lignes passant dans la connexion, ou **Relative** pour évaluer le ratio (%) du nombre de lignes passé dans cette connexion par rapport à une connexion de référence. Si vous sélectionnez **Relative**, vous devez également sélectionner une connexion de référence dans la liste **Connections List**.
- Cliquez sur le bouton **[+]** pour ajouter une ligne au tableau **Thresholds** et définir une plage pour le nombre lignes à enregistrer dans le log.

Consultez également : *Scénario : Mesurer le flux d'un Job, page 1271* dans le [Guide de Référence des Composants Talend Open Studio](#) et *Contrôle des connexions* dans le [Guide Utilisateur Talend Activity Monitoring Console](#).

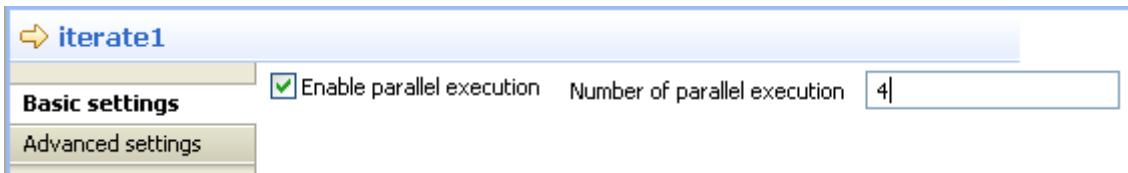
Paramètres d'une connexion **Iterate**

Vous pouvez utiliser un lien **Iterate** pour exécuter des itérations en parallèle :

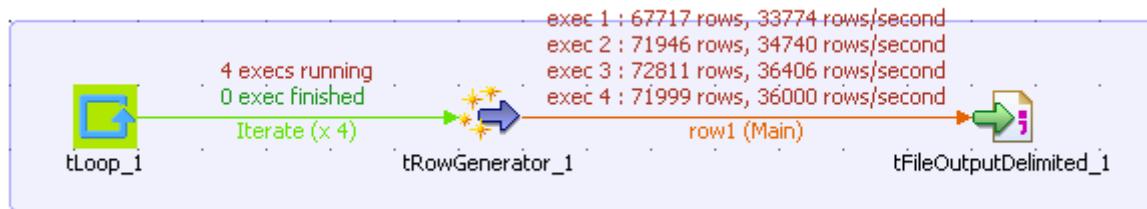
- Sélectionnez le lien **Iterate** de votre sous-job puis affichez l'onglet **Basic settings** de la vue **Components**.
- Cochez la case **Enable parallel execution** et configurez le nombre d'exécutions à effectuer en parallèle.

Conception d'un Job d'intégration de données

Utiliser le Metadata Manager



Lors de l'exécution de votre Job, les itérations effectuées en parallèle seront réparties sur les différents processeurs disponibles.

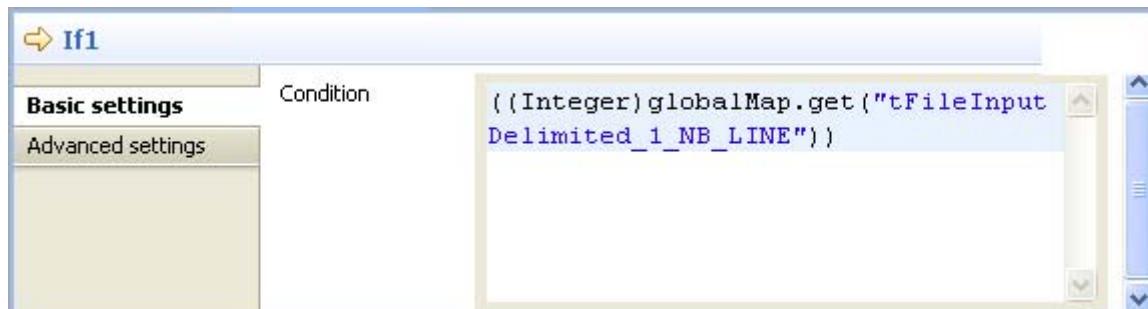


- Cochez la case **Statistics** de l'onglet **Run**, pour afficher, en temps réel, les exécutions en parallèles dans l'espace de modélisation.

Paramètres d'une connexion Trigger

Paramètres d'une connexion Run if

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.



4.4 Utiliser le Metadata Manager

Talend Open Studio est une solution orientée métadonnées, et peut donc vous permettre d'assurer la cohérence et la qualité du Job complet à travers le gestionnaire Metadata dans le Repository.

Talend Open Studio fournit un gestionnaire des métadonnées consolidant toutes les informations des projets dans un Repository centralisé, partagé par tous les utilisateurs durant les processus d'intégration de données. Ce Repository partagé facilite le travail collaboratif et permet aux membres d'une équipe de stocker et partager leurs Business Models, leurs Jobs d'intégration et leurs métadonnées.

4.4.1 Centraliser les 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 grâce à un gestionnaire de métadonnées centralisé dans le **Repository**.

Le répertoire **Metadata** du **Repository** regroupe plusieurs assistants permettant de créer des connexions à des fichiers, des bases de données et/ou des systèmes auxquels vous vous connectez souvent.

De cette manière, vous pouvez centraliser les informations de connexion dont vous avez habituellement besoin pour vos composants et les réutilisez facilement pour tous vos Jobs designs sans avoir à les saisir à chaque fois.

Grâce à l'assistant de métadonnées, vous pouvez collecter et centraliser les informations de connexion aux composants de type suivant :

- DB Connection : informations de connexion et description (schéma) des tables de toutes bases de données et connexions JDBC.
- File Delimited/Positional/Regex/XML/Excel/Ldif : informations d'accès et description des données des types de fichier listés.
- LDAP : informations d'accès et description des données d'un répertoire LDAP.
- Salesforce : informations d'accès et description d'une table Salesforce.
- WSDL : informations d'accès et description des données d'un service Web.
- Generic schema : informations d'accès et description de toutes sortes de sources.

Pour plus d'informations concernant les procédures de création de métadonnées, consultez *Gestion des métadonnées, page 189*.

4.4.2 Centraliser 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+Espace** dans les propriétés du composant dans la vue **Component**.

Utiliser les variables dans un Job

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

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 Java/Perl prédéfinies.

Description: Error Message Global variable, property of component tMap [tMap_1]. Type: String Availability: After Variable Name: ((String)globalMap.get("tMap_1_ERROR_MESSAGE"))	tFileInputDelimited_2.ERROR_MESSAGE tFileInputDelimited_2.NB_LINE tMap_1.ERROR_MESSAGE 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 à partir de la vue Component, page 97.*
- *Définir les variables à partir de la vue Contexts, page 92.*

Définir les 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 à partir de la vue Component, page 97*.

La vue **Contexts** se situe dans la partie inférieure de l'espace de modélisation et se compose de trois onglets : **Variables**, **Values as tree** et **Values as table**.



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**.

Onglet Variables

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

Name	Source	Type	Script code
TST_DBNAME	built-in	String	context.TST_DBNAME
TST_USER	built-in	String	context.TST_USER
TST_PASS	built-in	String	context.TST_PASS
context4Prod	context4Prod	-	-
File	context4Prod	String	context.File
new1	context4Prod	String	context.new1
Newvariable	built-in	String	context.Newvariable

Ajouter
 Supprimer

 Original order

Variables du Repository

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 dans le **Repository** sont en lecture seule.)
- Importez les variables à partir d'une source contexte du **Repository** à l'aide du bouton **Repository variables**.
- Vérifiez l'ordre initial des variables de contexte dans lequel elles sont automatiquement classées dans le studio lors de leur création ou de leur importation à partir du **Repository**. Pour ce faire, vous devez cocher la case **Original order**.
- Réorganisez les variables de contexte selon l'ordre initial en utilisant les boutons et . Pour ce faire, vous devez cocher la case **Original order** afin d'activer ces deux boutons.

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

Vous pouvez ajouter autant d'entrées 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.

Conception d'un Job d'intégration de données

Utiliser le Metadata Manager

Champs	Description
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 stocké 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 : \$_context{VotreNomDeParametre}. En Java, le script affiché sera : context.VotreNomDeParametre Ce Script de code se génère automatiquement lorsque vous définissez la variable dans la vue Component du composant.
Comment	Ajoutez un commentaire utile.



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 à partir de la vue Component, page 97*.

Pour plus d'informations concernant les variables du **Repository**, consultez *Centraliser les contextes dans le Repository, page 99*.

Onglet Values as tree

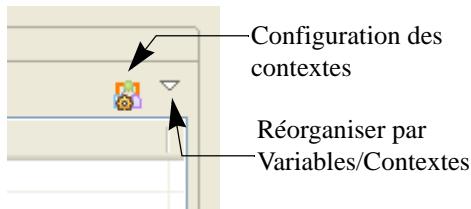
Cet onglet affiche les variables, ainsi que leurs valeurs, sous forme d'arbre.

Variable	Context	Prompt	Value
+ TST_DBNAME			
+ TST_USER			
+ TST_PASS			
+ File			
+ new1			
+ newVar			
	Default	<input type="checkbox"/> new2?	MyCompany-Dbms
	Prod	<input type="checkbox"/> new2?	null

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 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 96*.

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 à partir de la vue Component, page 97* et *Centraliser les contextes dans le Repository, page 99*.

Onglet Values as table

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

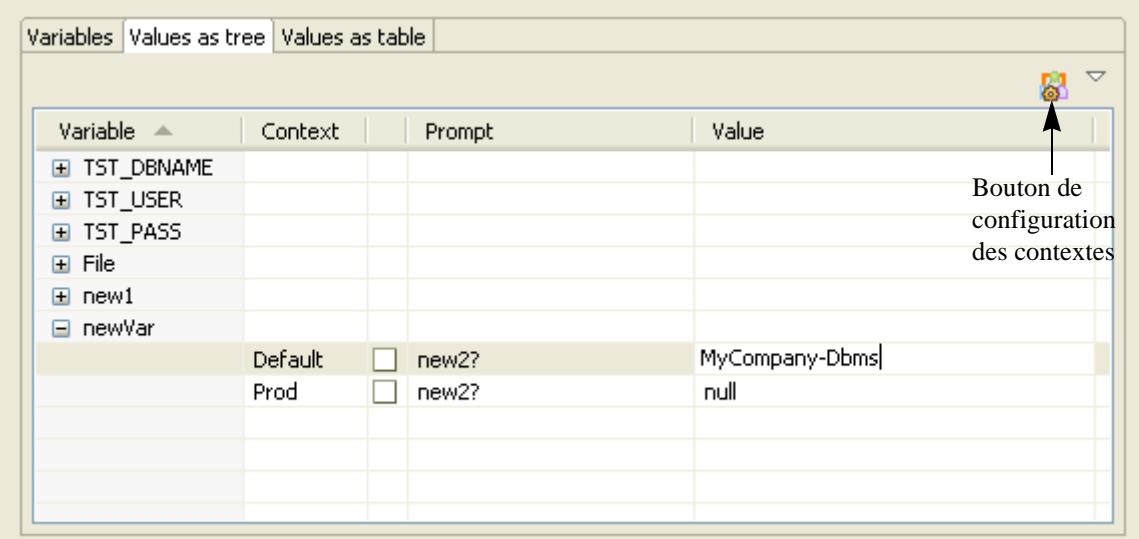
Champs	Description
Name	Le nom de la variable.
<YourContextName>	La valeur correspondant à la variable.

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 *Configurer les contextes, page 96*.

Pour plus d'informations concernant la définition des variables, consultez *Définir les variables à partir de la vue Component, page 97* et *Centraliser les contextes dans le Repository, page 99*.

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**.



Variable	Context	Prompt	Value
+ TST_DBNAME			
+ TST_USER			
+ TST_PASS			
+ File			
+ new1			
- newVar			
Default		<input type="checkbox"/> new2?	MyCompany-Dbms
Prod		<input type="checkbox"/> new2?	null

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

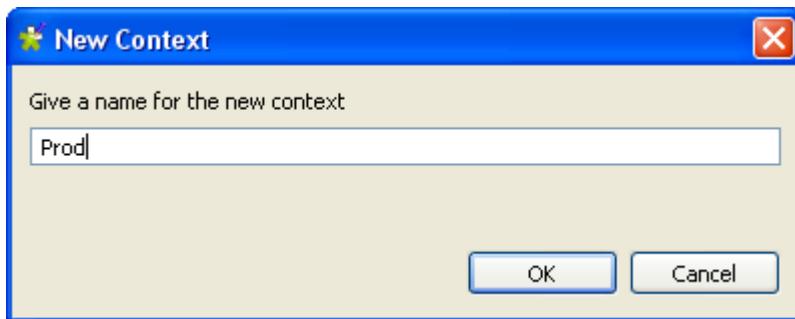


Le contexte par défaut ne peut être supprimé, c'est pourquoi le bouton **Remove** est grisé. Pour le rendre accessible, sélectionnez un autre contexte dans la liste **Default Context** dans l'onglet **Contexts**.

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 New dans la boîte de dialogue [Configure Contexts].
- 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 les variables à partir de la vue Contexts, page 92*.

Définir les variables à partir de la vue Component

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

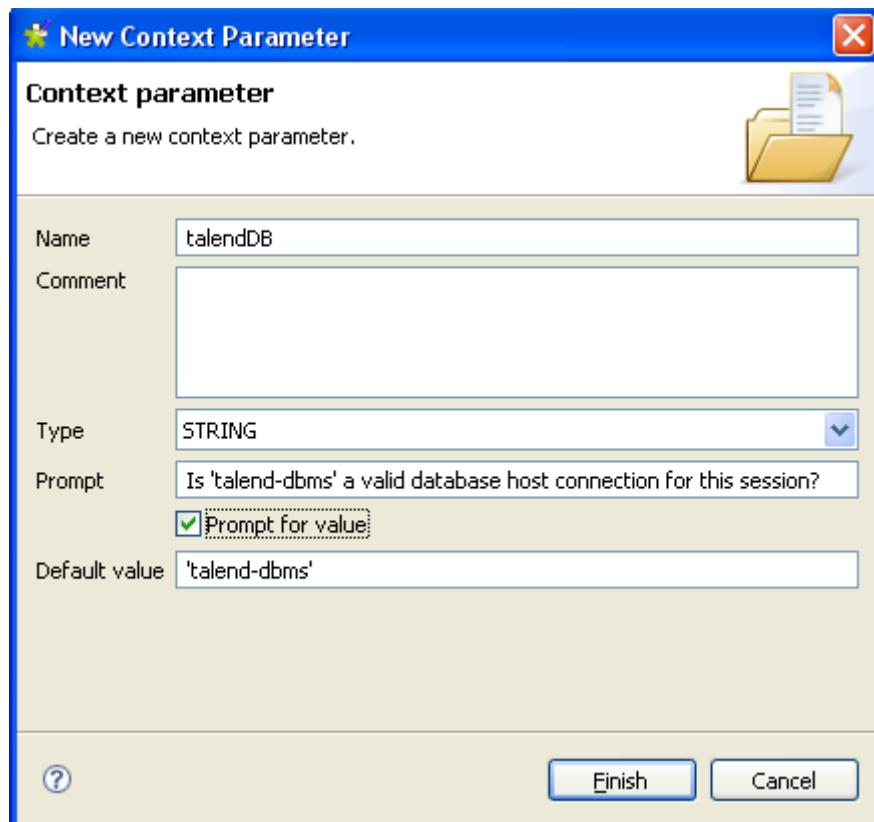
Pour plus d'informations concernant la définition des variables à partir de la vue **Contexts**, consultez *Définir les variables à partir de la vue Contexts, page 92*.

Pour plus d'informations concernant la définition des variables du **Repository**, consultez *Centraliser les contextes dans le Repository, page 99*.

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 paramétrier.
- 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 du 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**.



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 tous les contextes existants, mais vous pourrez par la suite changer la valeur de manière indépendante dans chaque contexte.

Pour plus d'informations concernant la création ou l'édition d'un contexte, consultez *Configurer les contextes, page 96*.

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+Espace**) des propriétés des composants de type DB input.

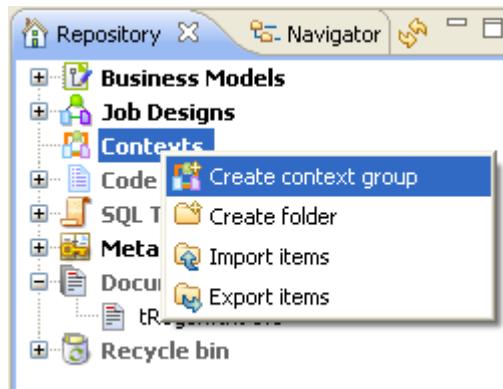
Pour plus d'informations concernant le paramétrage de **StoreSQLQuery**, reportez-vous au **Guide de référence des Composants Talend Open Studio**, et plus particulièrement au **Scénario 2 : Utiliser la variable StoreSQLQuery du composant tDBInput**.

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.

Créer un groupe de contexte

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 espace de modélisation.

- 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.

Conception d'un Job d'intégration de données

Utiliser le Metadata Manager

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 cette exemple, vous avez 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 96*.

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.

Context	Variable	Prompt	Value
Default	Servername	<input checked="" type="checkbox"/> Confirm Server name?	Testserver
	Dbname	<input type="checkbox"/> new1?	TestDB
Prod			

- 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**.

Créer un contexte à partir d'une métadonnée

Lorsque vous créez une métadonnée (à l'aide de l'assistant de connexion File ou DB), vous avez la possibilité d'exporter les informations de connexion à cette métadonnée en tant que Contexte.

Pour plus d'informations concernant cette option, consultez *Exporter une métadonnée en tant que contexte, page 262*.

Appliquer des variables de contexte dans un Job du Repository

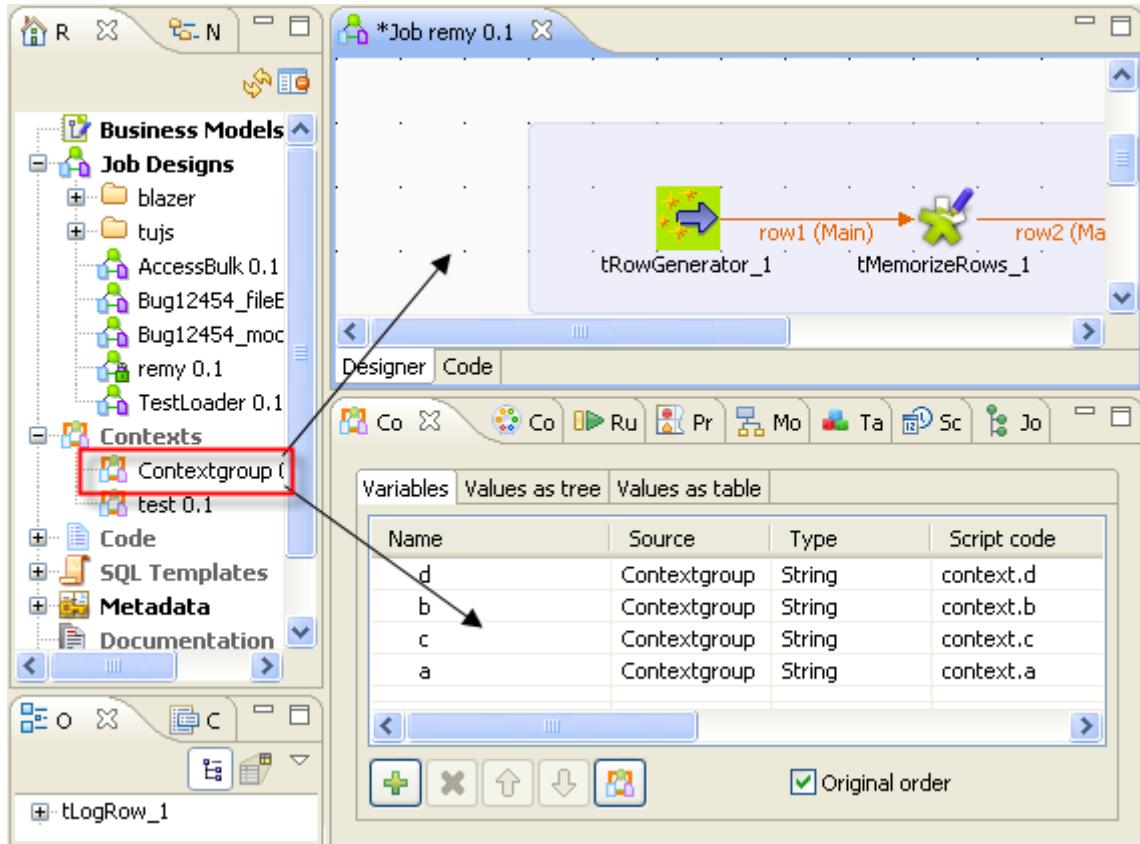
Une fois le groupe de contexte créé et stocké dans le **Repository**, vous pouvez l'appliquer à un Job par deux moyens :

- Déposez un groupe de contexte. Ainsi, le groupe sera appliqué comme un tout.
- Utilisez l'icône de contexte . De cette manière, les variables d'un groupe de contexte pourront être appliquées séparément.

Déposer un groupe de contexte dans un Job

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

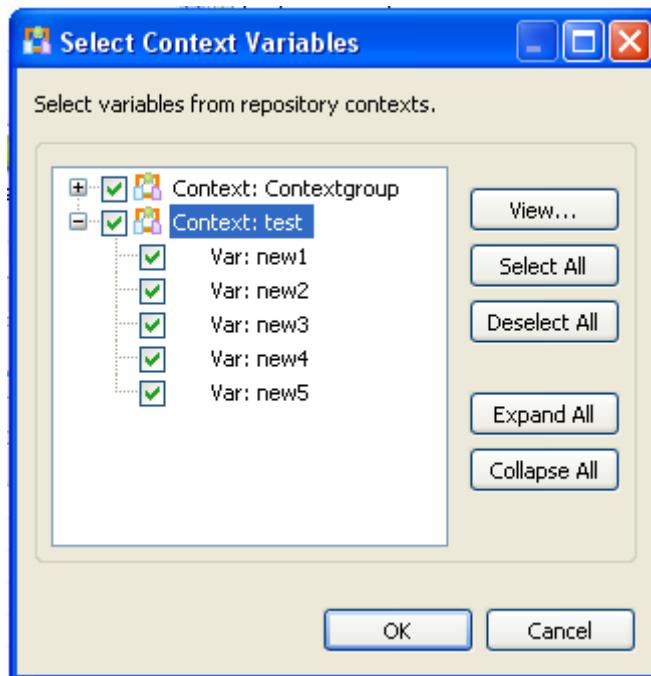
- Double-cliquez sur le Job auquel jouter un groupe de contexte.
- Une fois le Job ouvert, glissez le groupe de contexte de votre choix, soit dans l'espace de modélisation graphique du Job, soit dans la vue **Contexts** sous l'espace de travail.



Utiliser l'icône de contexte

Pour utiliser l'icône de contexte  afin d'appliquer les variables de contexte dans un Job, procédez comme suit :

- Double-cliquez sur le Job auquel ajouter une variable de contexte.
- Une fois le Job ouvert dans l'espace de modélisation graphique, cliquez sur la vue **Contexts** sous l'espace de travail afin de l'ouvrir.
- Au bas de la vue **Contexts**, cliquez sur l'icône  afin d'ouvrir l'assistant pour sélectionner les variables de contexte à appliquer.



- Dans l'assistant, sélectionnez les variables de contexte que vous souhaitez appliquer, ou effacez celles dont vous n'avez pas besoin.



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

- Cliquez sur **OK** pour appliquer les variables de contexte dans le Job.

Exécuter un Job dans un contexte défini

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

Context	Target execution
Test	

Name	Value
host	'talend-dbms-test'
port	'8080'
db	'Talend-test'

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**.

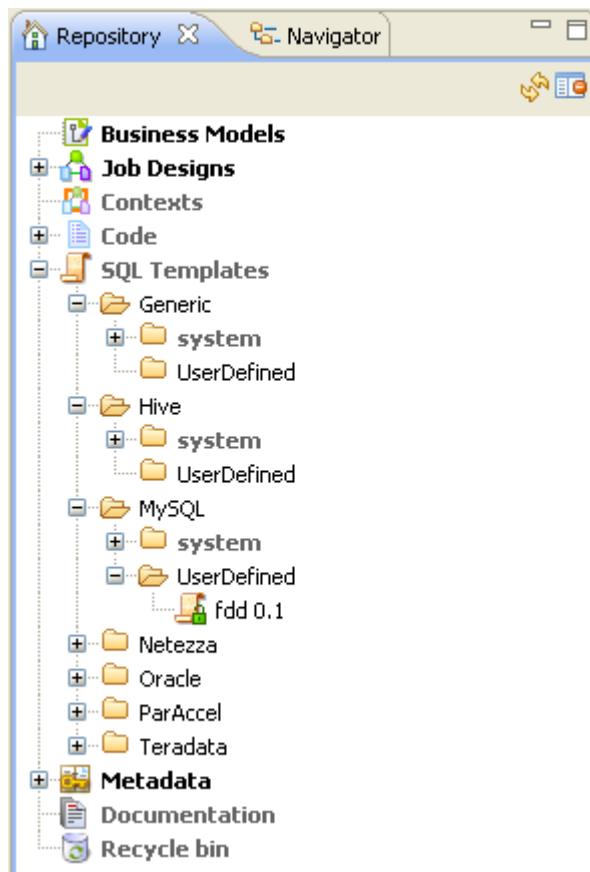
Voir également :

- Définir les variables à partir de la vue Contexts, page 92*
- Centraliser les contextes dans le Repository, page 99*

4.4.3 Utilisation des SQL Templates

Talend Open Studio vous permet de bénéficier des modèles SQL système puisque beaucoup de structures de requêtes sont standardisées à partir des mêmes approches.

Les modèles SQL du système sont regroupés sous le noeud **SQL Templates** dans le **Repository** de Talend Open Studio. Différents types de modèles SQL standardisés sont disponibles : des modèles génériques sous le noeud **Generic** et des modèles spécifiques aux bases de données, comme **MySQL**, **Oracle** ou **Teradata**.



Vous pouvez créer vos propres modèles SQL dans chacune des catégories ci-dessus grâce à l'assistant **[SQL Template]**.

Pour plus d'informations concernant l'utilisation des SQL Templates dans **Talend Open Studio**, consultez *SQL Templates, page 275*.

Pour plus d'informations concernant la création de modèles SQL personnalisés et leur utilisation dans un Job, consultez le *Scénario : Effectuer une opération d'itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l'utilisateur (SQL Template), page 651* dans le **Guide de Référence des Composants Talend Open Studio**.

4.5 Gestion des Jobs : Utilisation avancée

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

4.5.1 Mapper des flux de données

Le moyen le plus courant de gérer de multiples flux d'entrée et de sortie dans votre Job, y compris les transformations et la redirection des données est d'utiliser le composant **tMap**.

Pour plus d'informations concernant les principes d'utilisation de ce composant, consultez *Mapping de flux de données, page 161*.

Pour des exemples de Jobs utilisant ce composant, consultez *tMap* dans le **Guide de Référence des Composants Talend Open Studio**.

4.5.2 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.

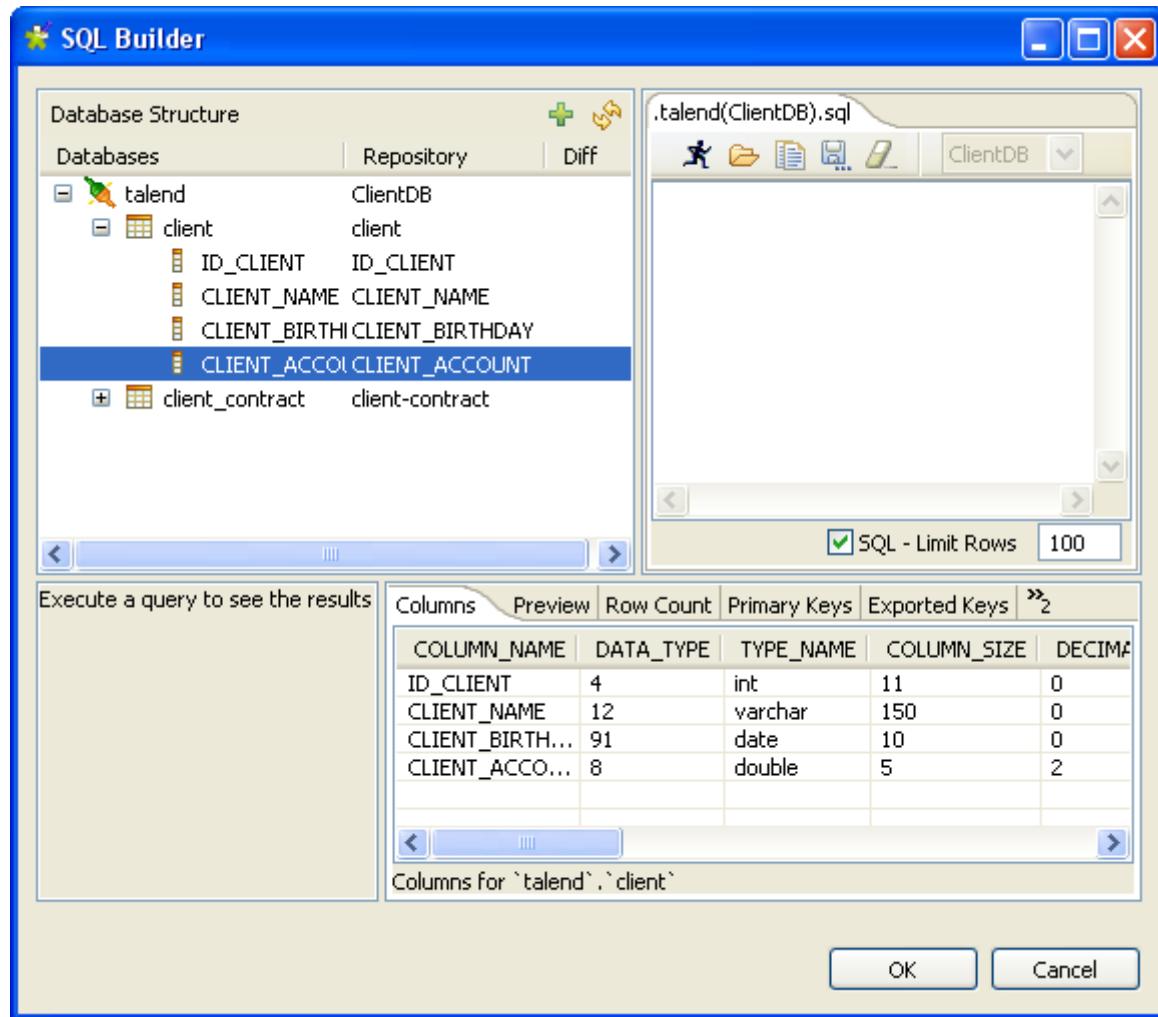
Vous pouvez construire un requête avec SQLbuilder que votre schéma de table soit stocké dans le référentiel ou intégrées en “built-in” directement dans les propriétés du composant.

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 en cliquant sur le bouton [...].

Conception d'un Job d'intégration de données

Gestion des Jobs : Utilisation avancée



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.

Comparer les structures de base de données

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



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.

The screenshot shows the 'Database Structure' interface. At the top, there are tabs for 'Databases', 'Repository', and 'Diff'. The 'Diff' tab is selected. Below the tabs, there's a tree view of tables. Red highlights are applied to several entries: 'CLIENT_NAME' under 'client', 'FOO' under 'client_contract', and 'ID_USER' under 'axeltable1'. A blue highlight is applied to 'sales'. To the right of the tree, there are icons for creating a new table ('+'), refreshing ('refresh'), and deleting ('X'). The bottom of the interface has a toolbar with various icons.

L'icône de la colonne **Diff** indique la présence de différences ou d'écart 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.

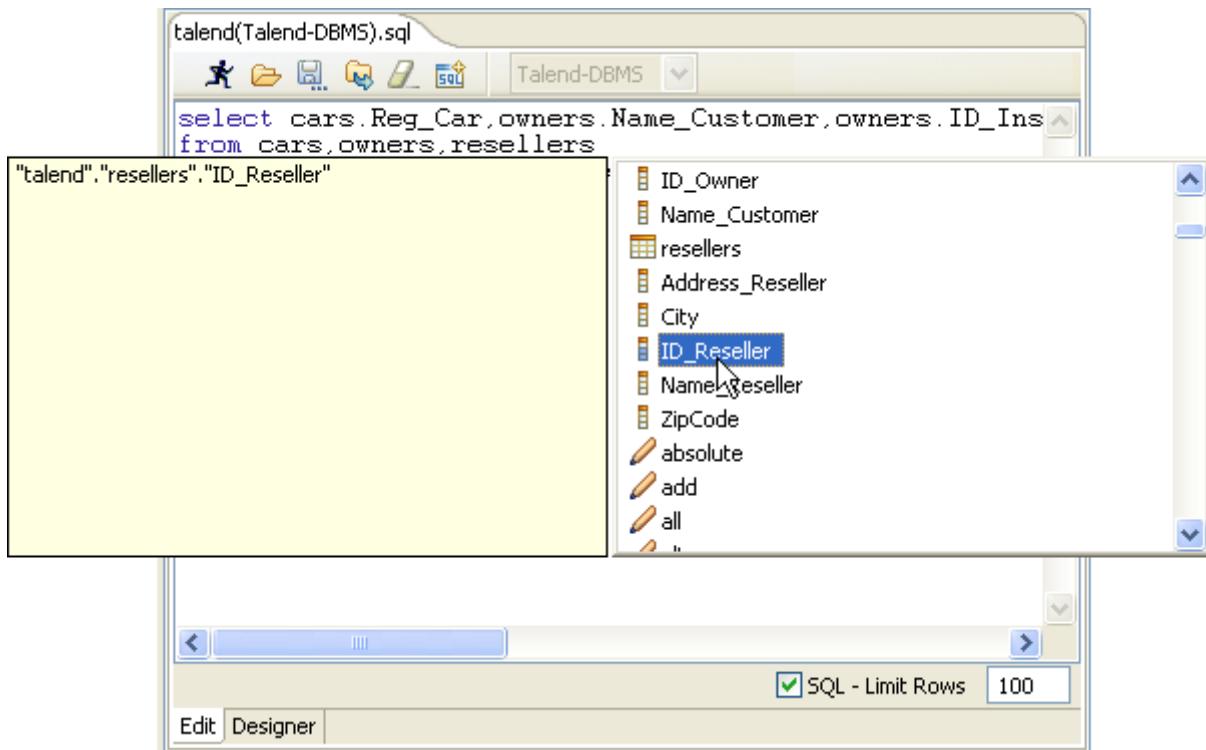
Le surlignage bleu indique que la colonne est présente dans la base de données elle-même mais qu'elle n'est pas présente dans la table de la métadonnée.

Construire 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.



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.

```

select owners.ID_Owner, owners.Name_Customer, owners.ID_Insurance, cars.ID_Owners,
from owners, cars, resellers
where owners.ID_Owner=cars.ID_Owners and cars.ID_Reseller=resellers.ID_Reseller

```

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.



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'icône du courrier.

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.

4.5.3 Télécharger des composants de la communauté

La vue **Talend Exchange** vous donne accès à une liste de composants compatibles avec votre version de **Talend Open Studio** mais qui ne sont pas installés dans votre **Palette**. Vous pouvez ensuite télécharger ces composants pour les réutiliser dans les Jobs Designs.

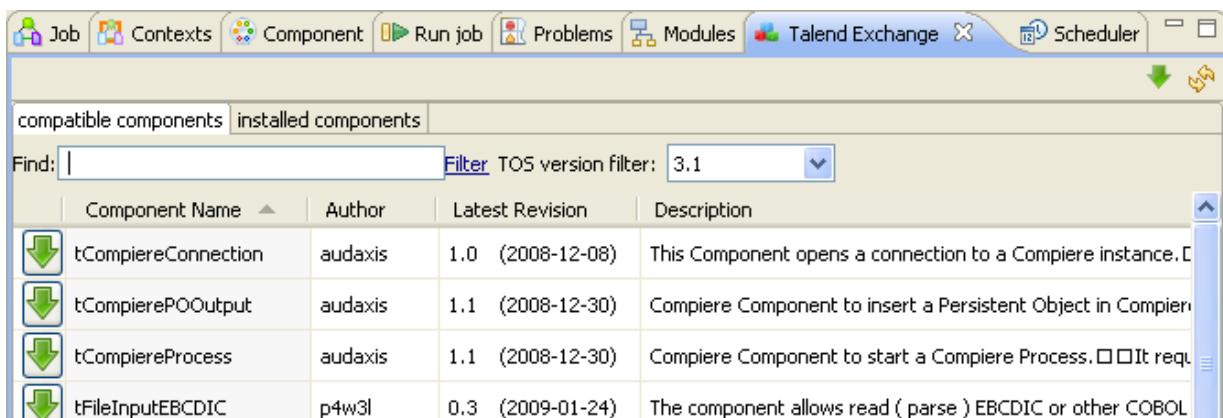


Les composants disponibles au téléchargement dans la vue **Talend Exchange** ne sont pas validés par **Talend**. Il est possible que vous soyez confrontés à des erreurs de chargements lorsque vous essayez de télécharger certains composants à partir de **Talend Exchange**.

Installer des composants de la communauté à partir de Talend Exchange

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

- Cliquez sur l'onglet **Talend Exchange** dans la partie inférieure de **Talend Open Studio** pour ouvrir la liste des composants compatibles (**Compatible components**). Cette liste apparaît comme vide jusqu'à ce que vous cliquez sur le bouton **Refresh** en haut à droite de la vue pour faire apparaître les composants disponibles.
- Cliquez sur le bouton **Refresh** en haut à droite de la vue. Les composants de **Talend Exchange** compatibles avec votre version de **Talend Open Studio** sont listés dans cette vue.



- Cliquez sur le bouton **Install Component** pour installer le composant sélectionné dans votre **Palette**. Une barre de progression apparaît, elle vous indique l'évolution du téléchargement en pourcentage. Lorsque la procédure d'installation du composant est terminée, une boîte de dialogue s'ouvre pour confirmer la réalisation de l'opération de téléchargement et indiquer à quel endroit de la **Palette** le composant a été chargé.



Pour télécharger simultanément un groupe de composants, sélectionnez les composants à télécharger, puis cliquez sur le bouton d'installation dans le coin supérieur droit de la vue.

- Cliquez sur **OK** pour fermer la boîte de dialogue de confirmation et redémarrez **Talend Open Studio**. Tous les composants téléchargés apparaissent maintenant dans leurs familles respectives dans la **Palette**.

Vous pouvez aussi télécharger des composants compatibles avec une version antérieure de **Talend Open Studio**. Pour cela :

- Dans la vue **Talend Exchange**, sélectionnez la version souhaitée, dans la liste **Talend Studio version filter**. La liste **Compatible components** est rafraîchie automatiquement.

Gérer les composants installés

A partir de la vue **Talend Exchange**, vous pouvez mettre à jour ou supprimer les composants que vous avez déjà installés dans la **Palette**.

Vous pouvez installer une version mise à jour d'un composant que vous avez déjà téléchargé à partir de la liste **Compatible components**. Pour cela :

- Dans le coin supérieur gauche de la vue, cliquez sur l'onglet **Installed components** pour ouvrir la liste de tous les composants que vous avez téléchargés à partir de **Talend Exchange**.



		Component Name	Author	Installed Revision	Description
		tFileInputXbase	plegall	0.3 (2009-05-12)	Read DBase and FoxPro files
		tOneToMany	plegall	2 (2009-04-29)	This component is a bridge between multiple sources

- Cliquez sur le bouton de rafraîchissement à côté du nom du composant pour lequel vous souhaitez installer la nouvelle version.



Si le composant de la communauté déjà téléchargé est modifié dans **Talend Exchange**, le bouton **Refresh** à côté du nom du composant s'affichera en bleu. Sinon, il s'affiche en gris.

Vous pouvez supprimer un composant que vous avez téléchargé à partir de la liste **Compatible components**. Pour cela :

- Dans la liste **Installed components**, cliquez sur le bouton de suppression situé à côté du nom du composant que vous voulez supprimer. Vous pourrez toujours le réinstaller plus tard si vous le souhaitez.

4.5.4 Installer des modules externes

L'utilisation de certains composants dans **Talend Open Studio** requiert l'installation de modules spécifiques.

La vue **Modules** liste tous les modules requis pour utiliser les composants embarqués dans le Studio. Certains de ces modules n'existent pas par défaut et doivent donc être installés, pour que vous puissiez exécuter correctement les Jobs utilisant ces composants.

Conception d'un Job d'intégration de données

Gestion des Jobs : Utilisation avancée



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

Pour accéder à la vue **Modules**, cliquez sur l'onglet **Modules** dans l'espace de modélisation graphique.

Status	Context	Module	Description	Required
Installed	tWebService	geronimo-ws-metadata_2....	Required for using this component.	<input checked="" type="checkbox"/>
Installed	tWebService	geronimo-jaxws_2.1_spec...	Required for using this component.	<input checked="" type="checkbox"/>
Installed	tWebService	wstx-asl-3.2.8.jar	Required for using this component.	<input checked="" type="checkbox"/>
Installed	tWebService	saaj-api-1.3.jar	Required for using this component.	<input checked="" type="checkbox"/>
Installed	tWebService	saaj-impl-1.3.2.jar	Required for using this component.	<input checked="" type="checkbox"/>
Installed	tWebService	jcifs-1.3.11.jar	Required for using this component.	<input checked="" type="checkbox"/>
Not installed	JBoss ESB Export	jbossesb-rosetta.jar	Library needed to export to JBoss...	<input checked="" type="checkbox"/>
Not installed	Db connection wizard	sas.core.jar	Required to define SAS connection	<input checked="" type="checkbox"/>
Not installed	Db connection wizard	sas.svc.connection.jar	Required to define SAS connection	<input checked="" type="checkbox"/>
Not installed	Db connection wizard	sas.intrnet.javatools.jar	Required to define SAS connection	<input checked="" type="checkbox"/>
Not installed	tHL7Input	hapi-structures-v51-0.6.jar	Required for using this component.	<input checked="" type="checkbox"/>
Not installed	tJasperOutput	poi-3.2-FINAL-20081019.jar	Required for using this component.	<input checked="" type="checkbox"/>
Not installed	tJasperOutputExec	jasperreports-3.1.4.jar	Required for using this component.	<input checked="" type="checkbox"/>
Not installed	tJasperOutputExec	poi-3.2-FINAL-20081019.jar	Required for using this component.	<input checked="" type="checkbox"/>
Not installed	tLDAPAttributesInput	courtinessl.jar	Required for using this component.	<input checked="" type="checkbox"/>
Not installed	tMtomInput	log4j-1.2.13.jar	Required for using this component.	
Not installed	tMtomOutput	log4j-1.2.13.jar	Required for using this component.	

Le tableau ci-dessous décrit les informations présentées dans la vue **Modules**.

Colonne	Description
Status	Indique si un module est installé ou non sur votre système. L'icône indique que le module est recommandé, mais non requis pour l'utilisation du composant listé dans la colonne Context . L'icône signifie que le module est indispensable à l'utilisation du composant correspondant.
Context	liste le nom des composants Talend utilisant le module. Si cette colonne est vide, le module est requis pour l'utilisation générale de Talend Open Studio . Cette colonne liste les bibliothèques externes ajoutées aux routines créées et sauvegardées dans le dossier des bibliothèques de votre Studio. Pour plus d'informations, consultez <i>Editer les bibliothèques des routines utilisateur, page 271</i> .
Module	affiche le nom exact du module.
Description	explique pourquoi le module/la bibliothèque est requis(e).
Required	la case cochée indique que le module est requis.

Pour installer un module manquant, procédez comme suit :

- Dans la vue **Modules**, cliquez sur l'icône dans le coin supérieur droit de la vue. La boîte de dialogue [Open] de votre système d'exploitation s'ouvre.

- Parcourez jusqu'au module que vous souhaitez installer, sélectionnez-le et cliquez sur **Open** dans la boîte de dialogue.
- La boîte de dialogue se ferme et le module sélectionné est installé dans le dossier des librairies de votre Studio.
- Vous pouvez maintenant utiliser le composant nécessitant le module dans vos Job Designs.

4.5.5 Lancer un Job périodiquement

La vue **Scheduler** dans **Talend Open Studio** vous permet de planifier une tâche qui lancera à intervalles réguliers un Job via le programme de planification de tâche (crontab).

Dans la vue **Scheduler**, vous pouvez générer un fichier crontab contenant des entrées compatibles *cron* (les données requises pour lancer le Job). Ces entrées vous permettront de lancer un Job à intervalles réguliers via le programme crontab.

Cette fonction de lancement de Job est basée sur la commande *crontab*, couramment utilisée sous Unix et systèmes d'exploitation similaires. Elle peut également être installée sous Windows.

Pour accéder à la vue **Scheduler**, cliquez sur l'onglet **Scheduler** dans l'espace de modélisation graphique.



Si l'onglet **Scheduler** n'apparaît pas dans votre espace de modélisation, suivez le chemin **Window > Show View... > Talend**, puis sélectionnez **Scheduler** dans la liste.

Minute	Hour	Day	Month	Weekday	Command
0	12	7	1	0	C:/Program Files/Java/jre6/bin/java.exe -Xms256M -Xmx1024
30	10	*	6	1	C:/Program Files/Java/jre6/bin/java.exe -Xms256M -Xmx1024
0	8	22	12	5	C:/Program Files/Java/jre6/bin/java.exe -Xms256M -Xmx1024

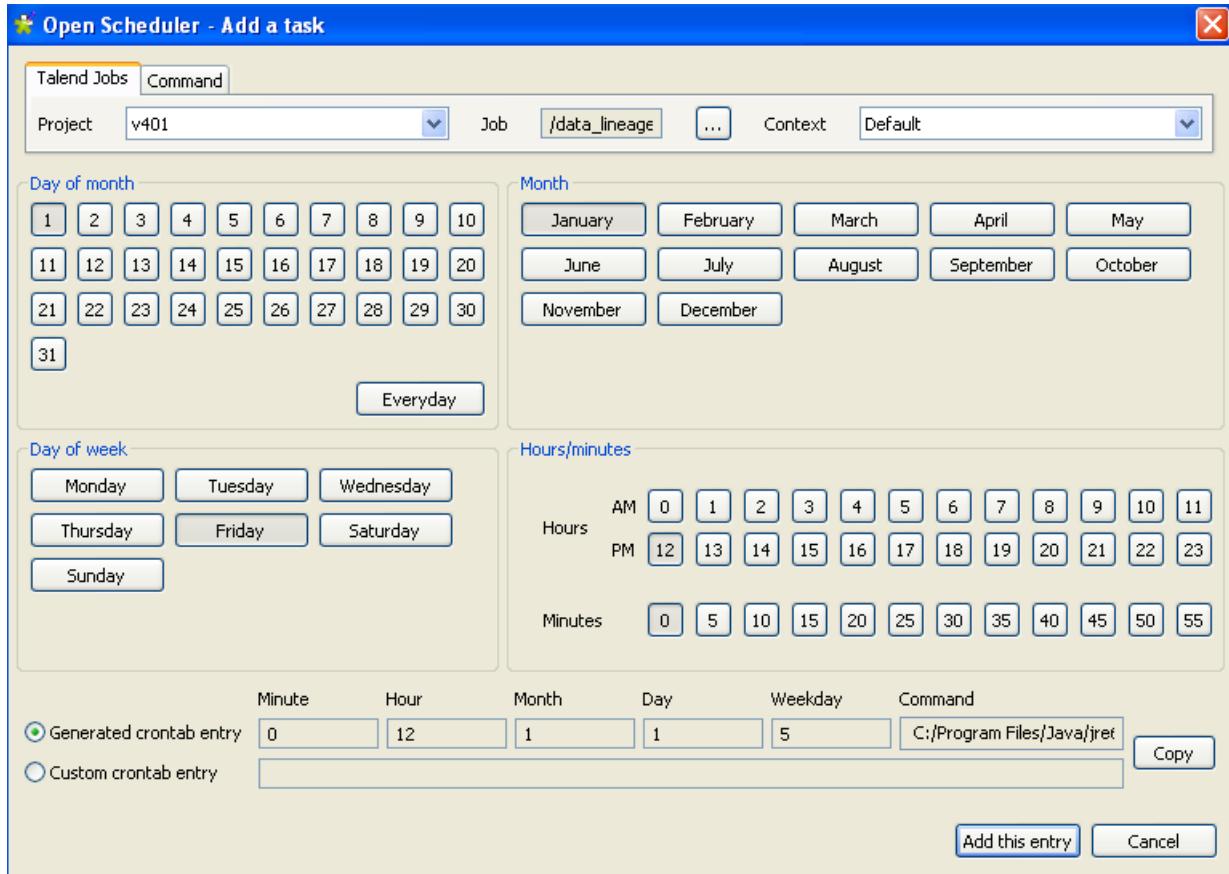
Cette vue est vide si vous n'avez programmé aucune tâche d'exécution de Job. Sinon, elle liste les paramètres de toutes les tâches planifiées.

La procédure ci-dessous explique comment planifier une tâche dans la vue **Scheduler** pour exécuter un Job spécifique à intervalles réguliers puis générer le fichier crontab qui contiendra toutes les données requises pour lancer le Job sélectionné. Elle indique également comment utiliser le fichier généré avec la commande *crontab* sous Unix ou le programme de planification des tâches sous Windows.

- Cliquez sur l'icône dans le coin supérieur droit de la vue **Scheduler**. La boîte de dialogue [Open Scheduler] s'ouvre.

Conception d'un Job d'intégration de données

Gestion des Jobs : Utilisation avancée



- Dans la liste **Project**, sélectionnez le projet contenant le Job que vous souhaitez lancer périodiquement.
- Cliquez sur le bouton [...] à côté du champ **Job** et sélectionnez le Job que vous souhaitez lancer à intervalles réguliers.
- Dans la liste **Context**, s'il y a plusieurs contextes, sélectionnez celui avec lequel vous souhaitez exécuter le Job.
- Paramétrez l'heure et la date complète d'exécution de cette tâche.
La commande qui sera utilisée pour lancer le Job sélectionné est générée automatiquement et jointe à la tâche définie.
- Cliquez sur **Add this entry** pour valider votre tâche et fermer la boîte de dialogue.
Les paramètres de la tâche planifiée sont listés dans la vue **Scheduler**.
 - Cliquez sur l'icône dans le coin supérieur droit de la vue **Scheduler** pour générer un fichier crontab qui contiendra toutes les données requises pour lancer le Job sélectionné. La boîte de dialogue [Save as] s'ouvre.
 - Parcourez votre répertoire pour définir le chemin d'accès au fichier crontab que vous générez.
 - Dans le champ **File name**, saisissez un nom pour le fichier crontab puis cliquez sur **Save** pour fermer la boîte de dialogue.
Le fichier crontab correspondant à la tâche sélectionnée est généré et stocké localement à l'emplacement défini.

- Sous Unix, collez le contenu du fichier crontab dans la configuration crontab de votre système Unix.
- Sous Windows, installez un programme de planification de tâche qui utilisera le fichier crontab généré pour lancer le Job.

Vous pouvez utiliser l'icône pour supprimer l'une des tâches affichées et l'icône pour éditer les paramètres de l'une des tâches affichées.

4.5.6 Utiliser les composants tPrejob et tPostjob

Les parties pré-job et post-job s'affichent en tant que composants dans l'espace de modélisation graphique, ils sont donc disponibles dans la **Palette** de composants. Pour utiliser ces composants **tPrejob** et **tPostjob**, déposez-les simplement dans l'espace de modélisation graphique comme vous le feriez avec d'autres composants. Un carré orange affiche les parties pré et post-job qui sont des types différents de sous-jobs.

Notez cependant que leur utilisation diffère légèrement de celle des composants habituels, puisque ces deux composants ne traitent ni des données ni des flux, mais ont été créés pour vous permettre de rendre votre Job plus clair.

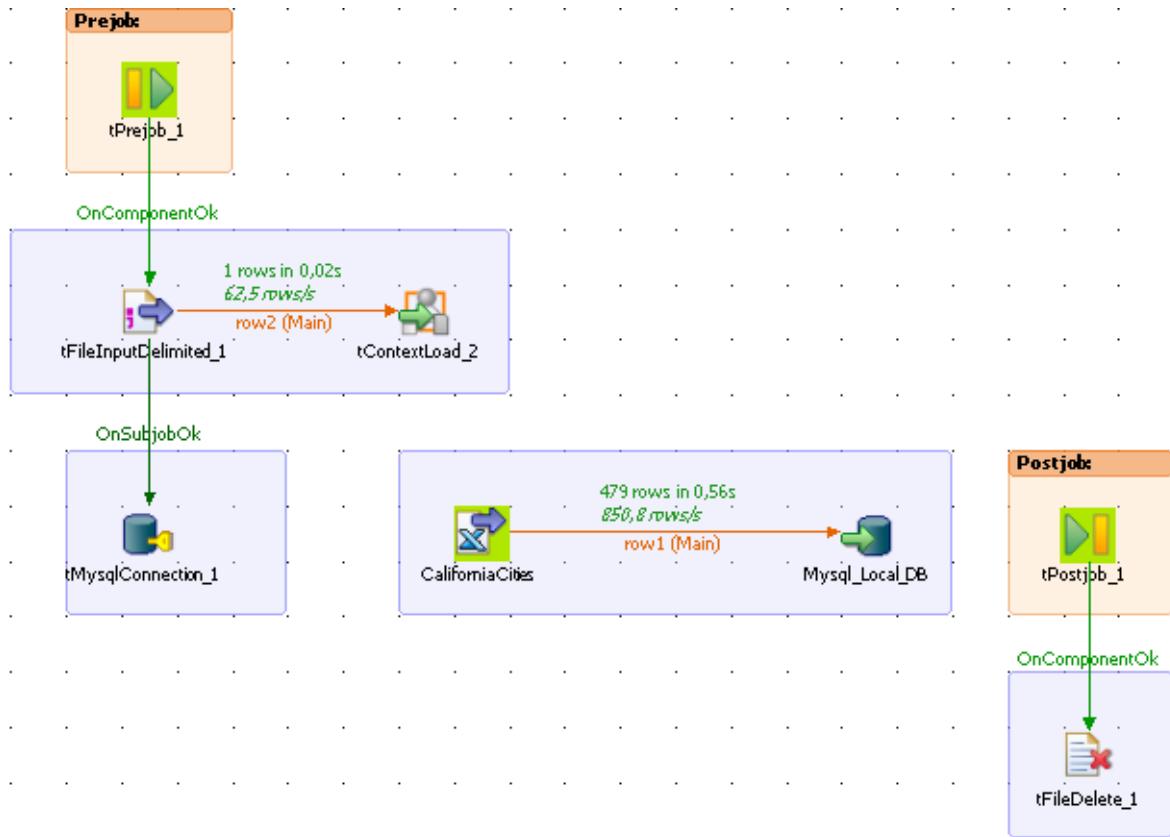


Puisque les composants **tPrejob** et **tPostjob** n'ont pas été créés pour le traitement de données, ils ne peuvent être exécutés en multi thread. Les tâches du **tPrejob** et du **tPostjob** sont exécutées une fois pour les sous-jobs suivants, qu'ils soient exécutés en séquence ou en parallèle.

Connectez aux **tPrejob** et **tPostjob** tous les composants organisant les tâches qui ne sont pas directement reliées au traitement des données, ou le sous-job principal pour permettre l'orchestration du traitement qui suivra.

Conception d'un Job d'intégration de données

Gestion des Jobs : Autres fonctions



Les tâches nécessitant l'utilisation d'un composant **tPrejob** sont par exemple :

- le chargement d'informations de contexte requises pour l'exécution du sous-job,
- l'ouverture d'une connexion à une base de données,
- la vérification qu'un fichier existe.

De nombreuses autres tâches collatérales à votre Job pouvant affecter la lisibilité de l'ensemble de votre Job peuvent aussi nécessiter un composant pré-job.

Les tâches nécessitant l'utilisation d'un composant **tPostjob** sont par exemple :

- la suppression des fichiers d'un dossier ou la suppression d'un fichier,
- toute tâche devant être exécutée même si le sous-job précédent a échoué.

4.6 Gestion des Jobs : Autres fonctions

Les sections suivantes fournissent des informations détaillées concernant divers sujets liés à la gestion de Jobs d'intégration de données, comme la définition des composants de début (start), la gestion des erreurs, l'utilisation des composants **tPrejob** et **tPostjob**, ainsi que la recherche de Jobs utilisant des composants spécifiques.

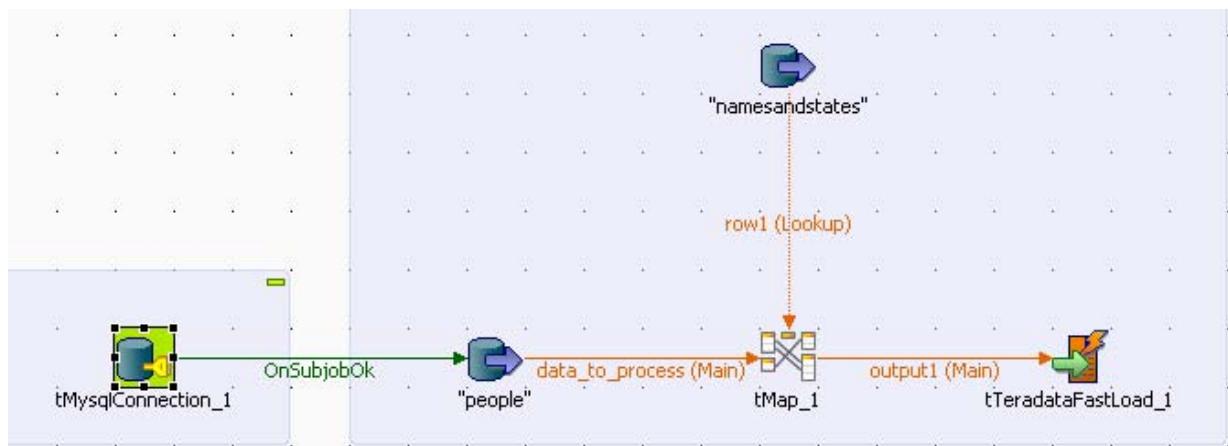
4.6.1 Partager une connexion à une base de données

Si vous avez plusieurs Jobs utilisant une connexion à la même base de données, vous pouvez dorénavant factoriser cette connexion en utilisant l'option **Use or Register a shared connection**.

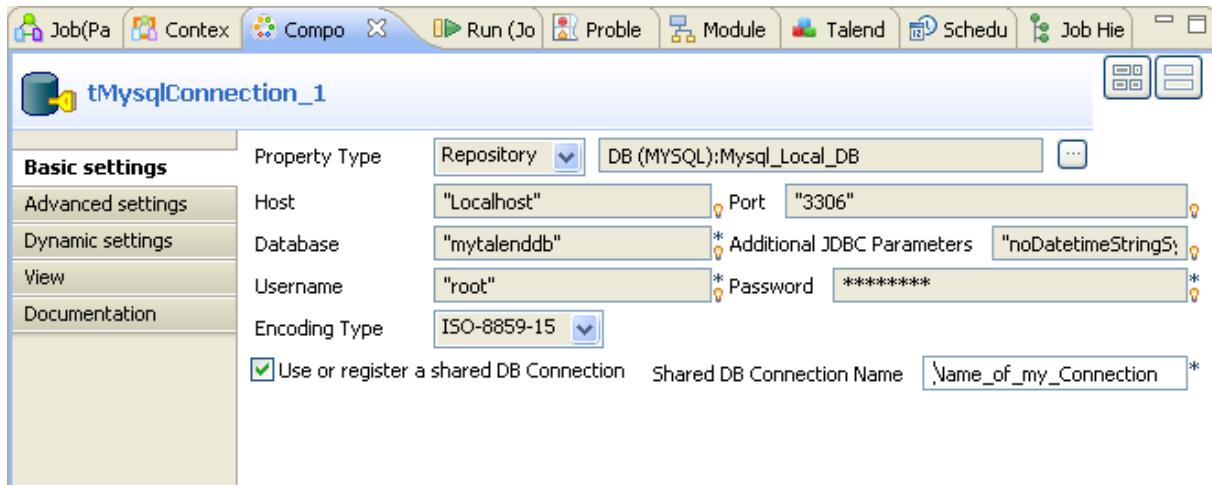
Cette case à cocher a été ajoutée à tous les composants de connexion afin de réduire le nombre d'ouvertures et de fermetures de connexion.

Si vous avez deux Jobs liés (un Job parent et un Job enfant) et que ces deux Jobs nécessitent une connexion à la même base de données MySQL distante, effectuez la procédure suivante :

- Cliquez-glissez un composant tMysqlConnection (dans le cas où vous travaillez avec une base de données MySQL).
- Connectez-le au premier composant de votre Job parent.



- Dans la vue **Component** du composant de connection, cochez la case **Use or Register a shared connection**.
- Dans le champ **Shared DB Connection Name**, donnez un nom à cette connexion que vous souhaitez partager.



- Maintenant vous pouvez réutiliser cette connexion dans votre Job enfant (et dans tout autre Job utilisant une connexion à cette même base de données).
- Suivez les mêmes étapes présentées ci-dessus et utilisez le même nom dans le champ **Shared DB Connection Name**.

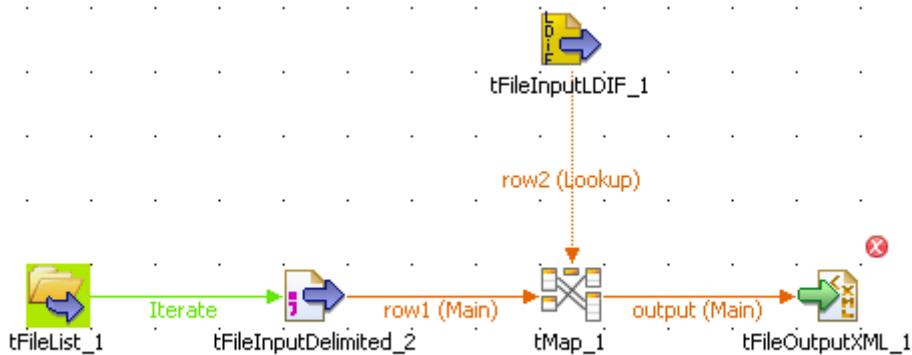
Pour plus d'informations concernant l'utilisation des composants de connexion, consultez le [Guide de Référence des Composants Talend Open Studio](#).

4.6.2 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.

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 débuter 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 est 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 :

- *Types de connexions, page 83*
- *Activation/Désactivation d'un Job ou d'un sous-job, page 130*

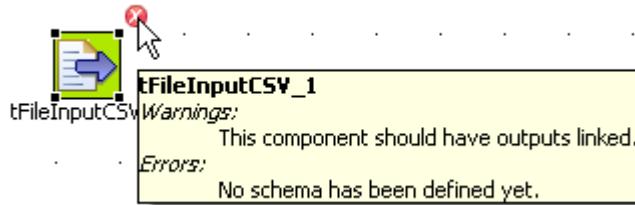
4.6.3 Gérer les icônes d'avertissement/d'erreur sur les composants

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

Icônes d'erreur et d'avertissement sur les composants

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

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



Icône d'erreur sur un Job

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



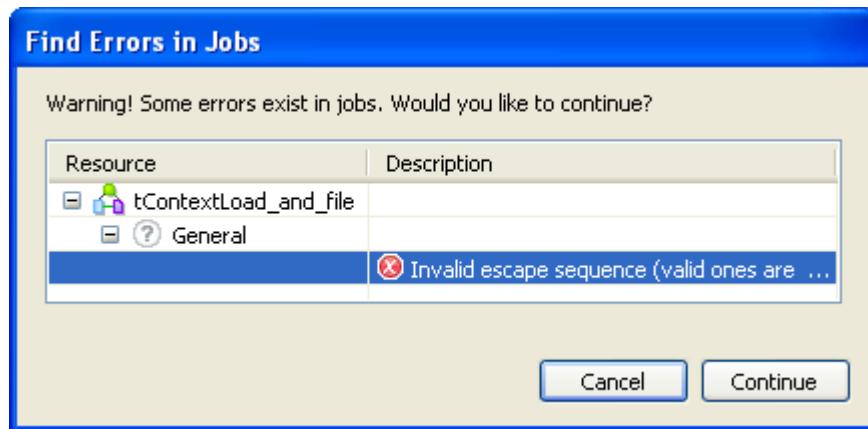
L'icône d'erreur apparaît aussi à côté du nom du Job dans l'onglet de l'espace de modélisation.

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

- l'ouverture d'un Job,
- le passage à l'onglet **Code viewer**,
- l'exécution du Job (en cliquant sur **Run**),
- l'enregistrement du Job.

L'icône d'erreur apparaîtra lorsque vous effectuerez l'une de ces actions.

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

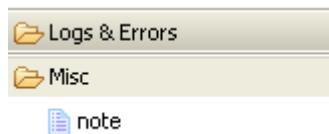


Cliquez sur **Cancel** pour arrêter l'exécution ou sur **Continue** pour continuer l'exécution de votre Job.

Pour plus d'informations concernant les erreurs dans les composants, consultez *Gérer les icônes d'avertissement/d'erreur sur les composants, page 119*

4.6.4 Ajouter une Note au Job

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



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



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

Les options **Fonts and Colors** permettent de modifier le style, la taille, la couleur, etc. de la police, ainsi que la couleur du fond et des bordures de votre note.

Les options **Adjust horizontal** et **Adjust vertical** permettent de définir l'alignement vertical et horizontal du texte dans la note.

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

4.6.5 Afficher les onglets Code ou Outline de votre Job

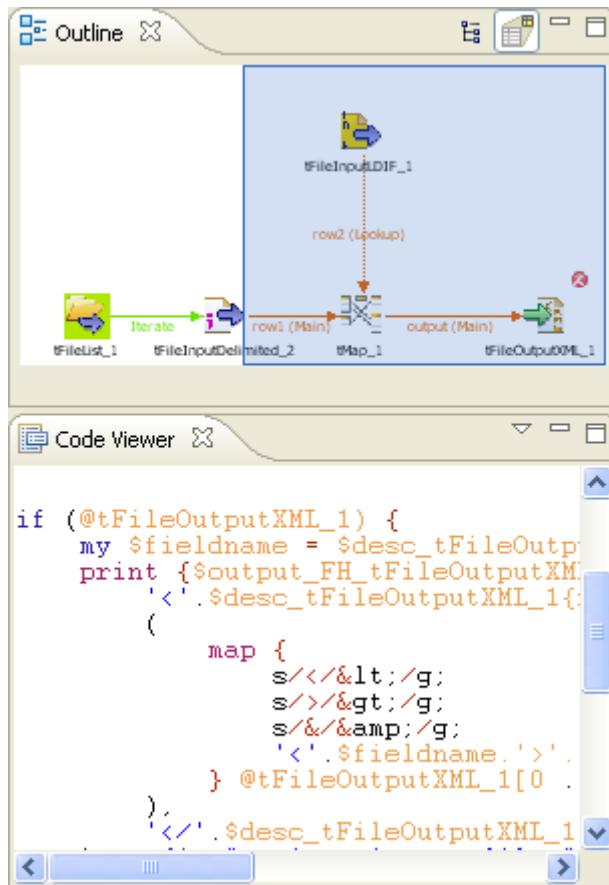
Ce panneau est situé sous la vue **Repository**. Il fournit des informations précises concernant le Job ou le Business Model ouvert dans l'espace de modélisation graphique.

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

Outline

L'onglet offre un aperçu rapide du Business Model ou du Job Design ouvert dans l'espace de modélisation graphique mais aussi une arborescence de tous les éléments utilisés dans le Job ou le Business Model. Comme l'espace de modélisation graphique, et comme toute zone de toute

fenêtre peut être réduit(e) ou agrandi(e) selon vos besoins, la vue **Outline** est utile pour savoir où vous êtes dans l'espace de modélisation graphique.



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 de l'espace de travail se déplace de la même manière.

La vue **Outline** peut également afficher un arborescence des composants utilisés dans le diagramme. Développez le noeud d'un composant pour afficher la liste des variables disponibles pour ce composant.

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

Code viewer

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

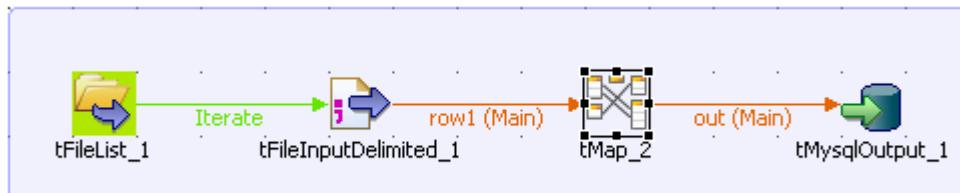


Cette vue ne concerne que le code produit dans un Job Design. En effet, aucun code n'est généré à partir des Business Models.

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.

4.6.6 Gérer l'affichage d'un sous-job

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



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



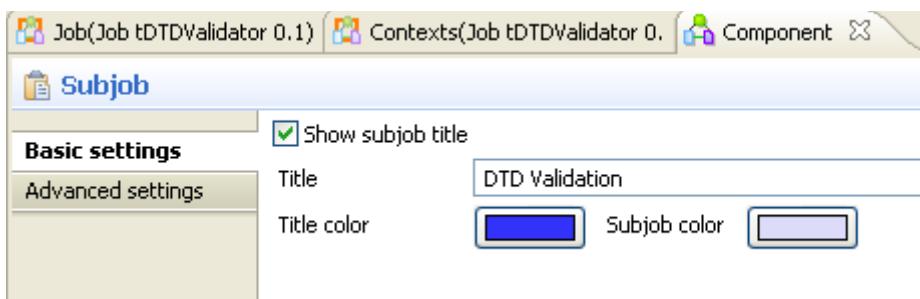
Un Job peut être composé d'un seul sous-job. Les rectangles oranges correspondent aux pré-jobs et aux post-jobs, qui sont différents des sous-jobs.

Pour plus d'informations sur les pré-jobs et les post-jobs, consultez la section *Utiliser les composants tPrejob et tPostjob, page 115*.

Personnaliser les sous-jobs

Ces rectangles représentant vos sous-jobs peuvent être mis en forme dans un soucis de clarté.

Pour cela, sélectionnez votre sous-job et cliquez sur la vue **Component**.



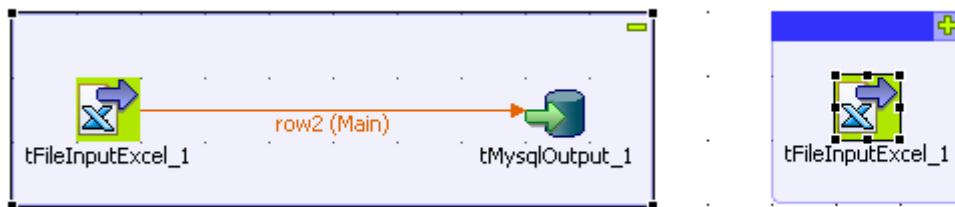
Dans la vue **Basic settings**, cochez la case **Show subjob title** si vous souhaitez ajouter un titre au sous-job et dans le champ **Title**, saisissez le titre.

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

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

Réduire l'affichage des sous-jobs

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

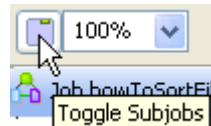


- Cliquez sur le signe moins [-] pour réduire le sous-job (capture d'écran de gauche). Quand il est réduit, seul le premier composant du sous-job apparaît.
- Cliquez sur le signe [+] pour le restaurer (capture d'écran de droite).

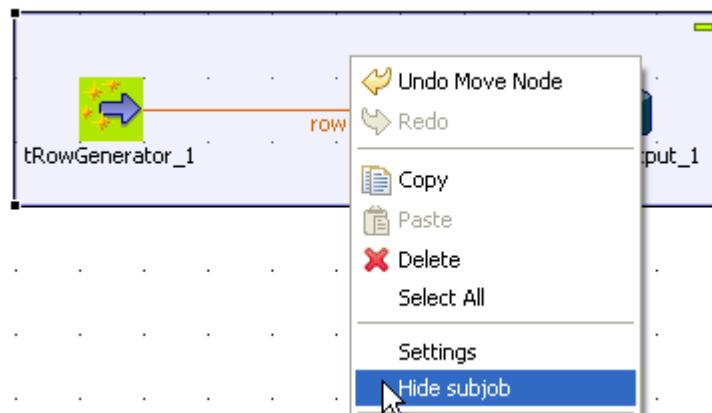
Supprimer le surlignage d'un sous-job

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

Pour enlever le surlignage de tous vos sous-jobs, cliquez sur l'icône **Toggle Subjobs** dans la barre d'outils de **Talend Open Studio**.



Pour enlever le surlignage d'un sous-job, cliquez-droit sur ce sous-job et sélectionnez l'option **Hide subjob** dans le menu contextuel.



4.6.7 Paramétrage des options dans la vue Job

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

L'onglet **Stats & Logs** vous permet d'utiliser de manière automatique les fonctions **Stats & Logs** et la fonction d'automatisation des paramètres de Contexte. Pour plus d'informations, consultez *Automatiser l'utilisation de statistiques & logs, page 124*.

L'onglet **Extra** liste les différentes options disponibles pour automatiser certaines fonctions, notamment l'utilisation des paramètres de contexte dans la zone **Implicit Context Loading**. Pour plus d'informations, consultez *Fonctions Extra, page 126*.

Automatiser l'utilisation de statistiques & logs

Si vous utilisez souvent les fonctions 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'informations sur les composants de Log, consultez le **Guide de Référence** des composants **Talend Open Studio**.

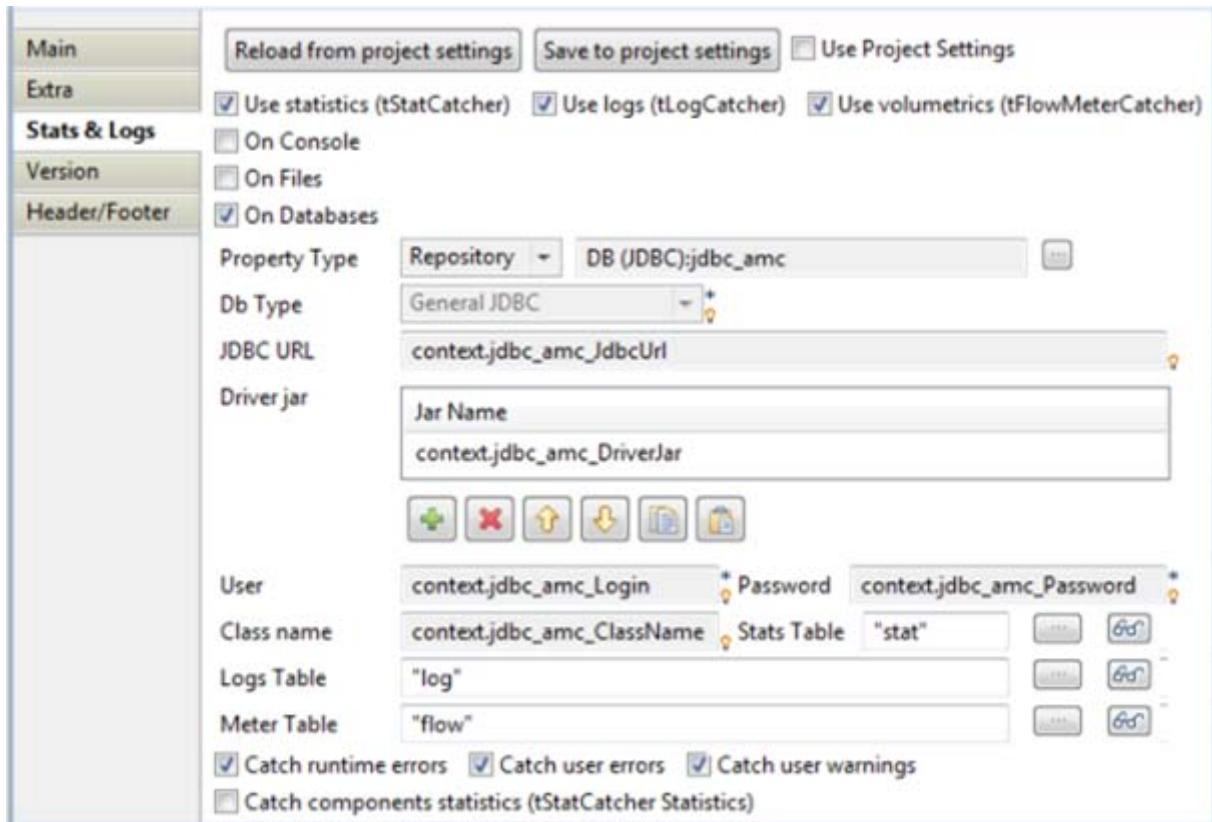
L'onglet **Stats & Logs** est situé dans la vue **Job** sous l'espace de modélisation et vous permet d'éviter de surcharger vos Jobs avec des composants.



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

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

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



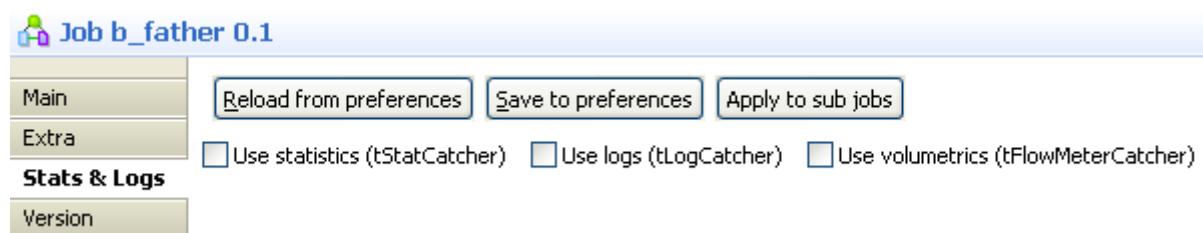
Lorsque vous utilisez une connexion à une base de données, par exemple JDBC, vous devez créer manuellement les tables log, stats et meter.

- 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 besoins.



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

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



Pour cela, cliquez sur le bouton **Apply to sub jobs** dans l'onglet **Stats & Logs** de la vue **Job**. Ainsi, les options sélectionnées pour votre Job principal seront sélectionnées pour tous ses sous-jobs.

Fonctions Extra

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

- Cochez la case **Multithread execution** pour lancer deux Jobs au même moment.
- Vous pouvez cocher l'option **Implicit tContextLoad** pour éviter d'utiliser le composant **tContextLoad** dans votre Job et ainsi automatiser l'utilisation des paramètres de contexte.
Choisissez la source de vos paramètres de contexte : **File** ou **Database** et configurez manuellement l'accès aux fichiers ou à la base de données.
Configurez les messages (erreur/warning/info) lorsque des comportements inattendus liés aux paramètres de contexte surviennent.
- Si vous avez renseigné manuellement l'option **Implicit tContextLoad**, vous avez la possibilité d'enregistrer ces paramètres dans les paramètres de votre projet en cliquant sur le bouton **Save to project settings**, et, ainsi, réutiliser ces paramètres pour d'autres composants et dans d'autres jobs.
- Vous avez la possibilité de récupérer les paramètres de contexte déjà définis dans les paramètres du projet (**Project Settings**) en cochant la case **Use Project Settings**. Pour plus d'informations sur les paramètres de contexte, consultez *Définir les paramètres de contexte, page 39*. En cochant cette case, l'option **Implicit tContextLoad** sera activée et tous les champs seront renseignés automatiquement.
- Vous avez aussi la possibilité de rafraîchir les paramètres de contexte récupérés des paramètres du projet en cliquant sur le bouton **Reload from project settings**.

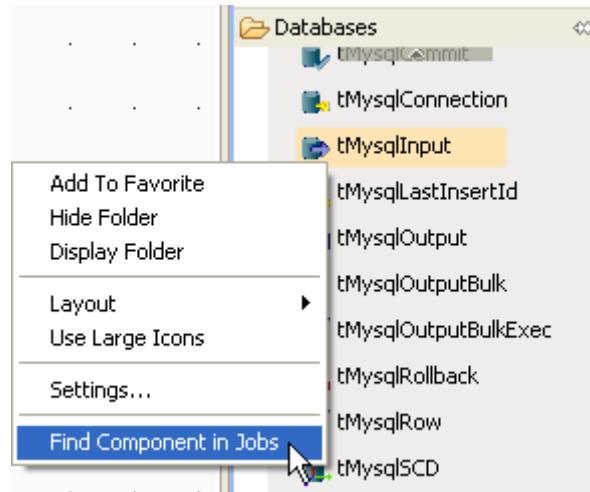
4.6.8 Chercher un Job utilisant un composant particulier



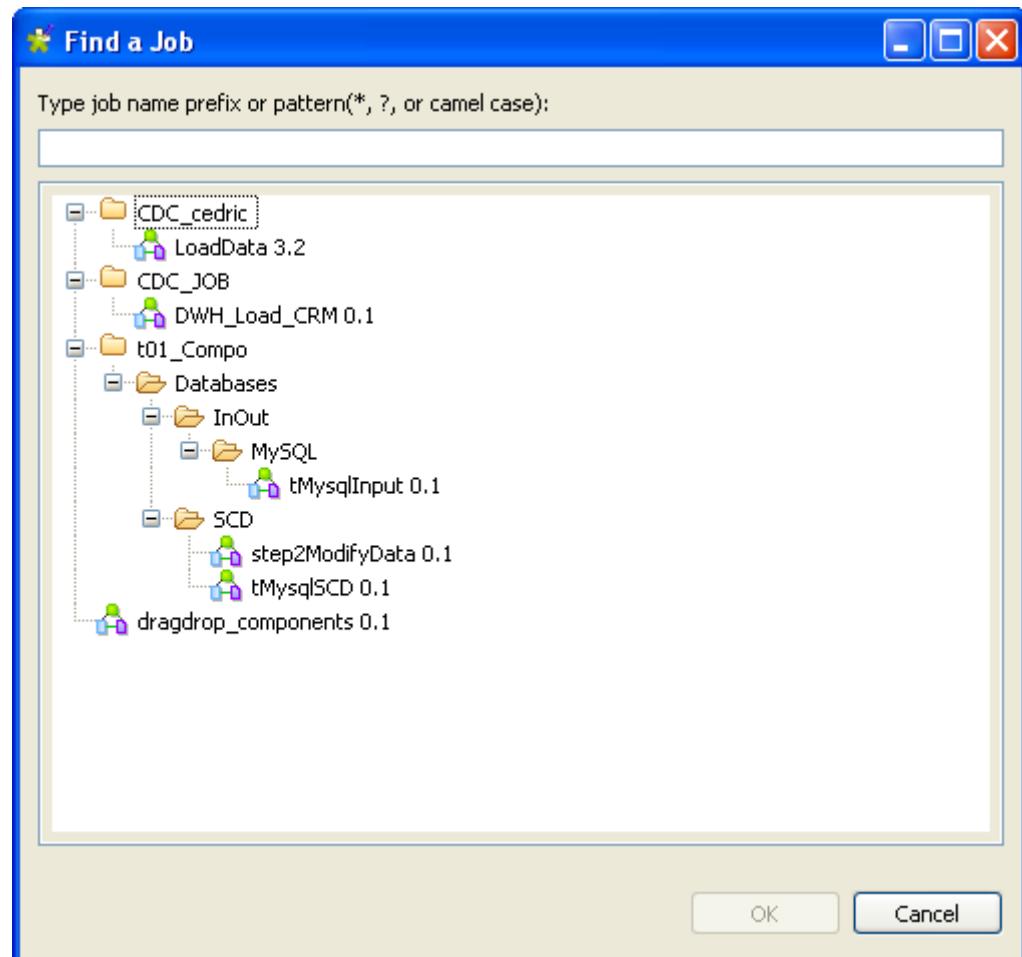
Vous devez ouvrir au moins un Job dans votre Studio afin d'afficher la **Palette** à droite de l'espace de modélisation graphique et lancer la recherche.

A partir de la **Palette**, vous avez la possibilité de rechercher tous les Jobs utilisant le composant sélectionné. Pour ce faire :

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



Une barre de progression s'affiche pour vous indiquer l'évolution de la recherche en pourcentage, puis la boîte de dialogue **[Find a Job]** s'affiche à l'écran et donne la liste de tous les Jobs utilisant ledit composant.



- Dans la liste des Jobs, sélectionnez celui qui vous intéresse puis cliquez sur **OK** pour l'ouvrir dans l'espace de modélisation graphique.



CHAPITRE 5

Gestion des Jobs d'intégration de données

Ce chapitre décrit les procédures de gestion des Jobs créés dans **Talend Open Studio**, ainsi que des autres éléments inclus dans un projet, par exemple les routines ou les métadonnées.

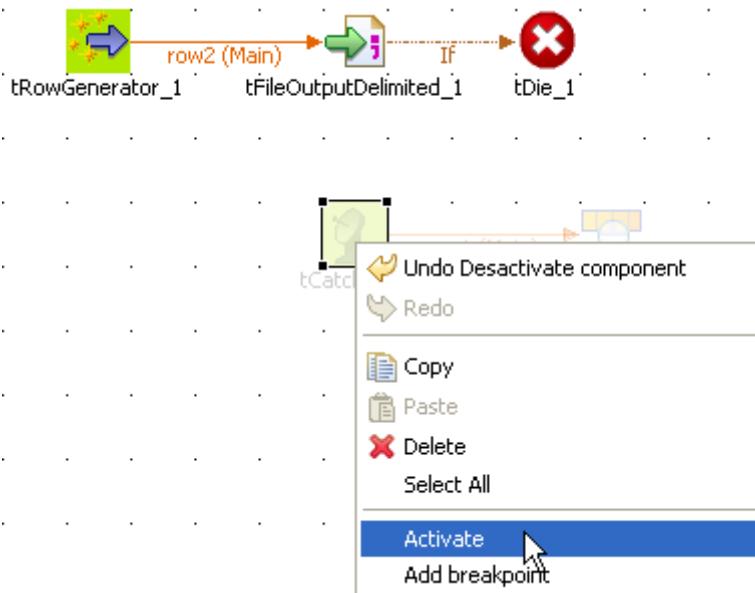
Ces procédures de gestion comprennent l'import et l'export de Jobs et d'éléments entre différents projets ou ordinateurs, ainsi que la programmation de l'exécution et l'exécution de Jobs et leur déploiement sur des serveurs distants, ainsi que la copie des Jobs dans différentes branches SVN.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

5.1 Activation/Désactivation d'un Job ou d'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 **Activate**.



Vous pouvez aussi faire un clic-droit sur le composant et sélectionnez la commande **Activate/Deactivate** 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 118.*

5.1.1 Désactiver un composant de début (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.

5.1.2 Désactiver un composant intermédiaire

Lorsque vous décochez la case **Activate** 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.

5.2 Import/export d'éléments ou de Jobs

Talend Open Studio vous permet d'import/exporter vos Jobs ou vos éléments à partir de/vers différents projets ou différentes versions du Studio. Il vous offre également la possibilité d'exporter des job scripts et ainsi de déployer et d'exécuter les Jobs créés dans le Studio sur n'importe quel serveur.

5.2.1 Import d'éléments

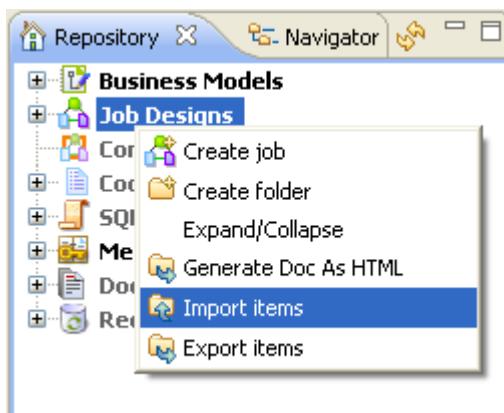
Vous pouvez à présent importer des éléments stockés dans le workspace des versions antérieures de Talend Open Studio ou d'un autre projet de votre workspace courant.

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 :

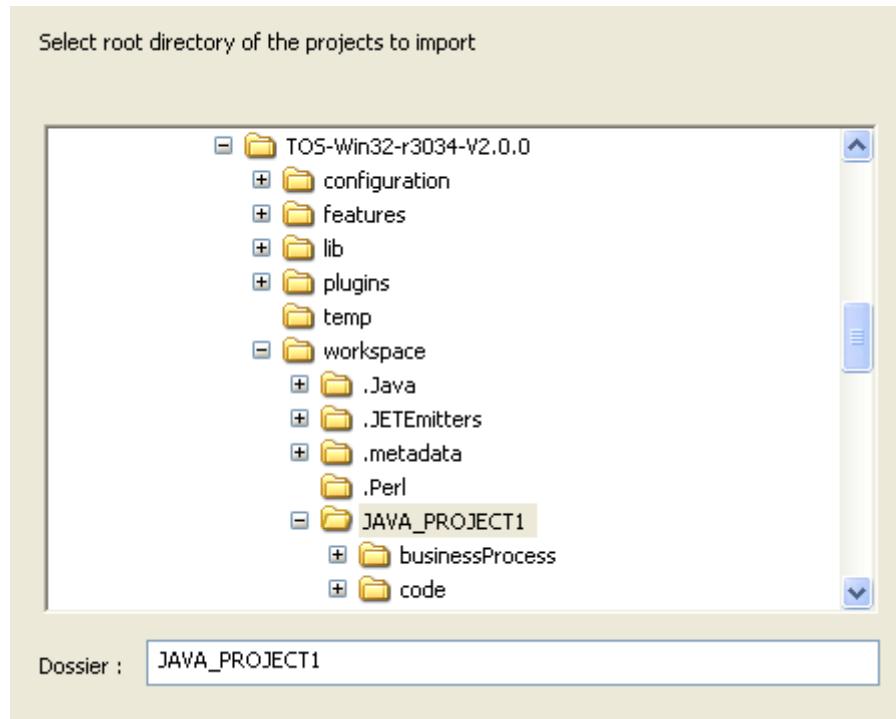
- Dans le **Repository**, cliquez-droit sur n'importe quelle entrée, notamment **Job Designs** ou **Business Models**.
- Dans le menu contextuel, sélectionnez l'option **Import Items**.



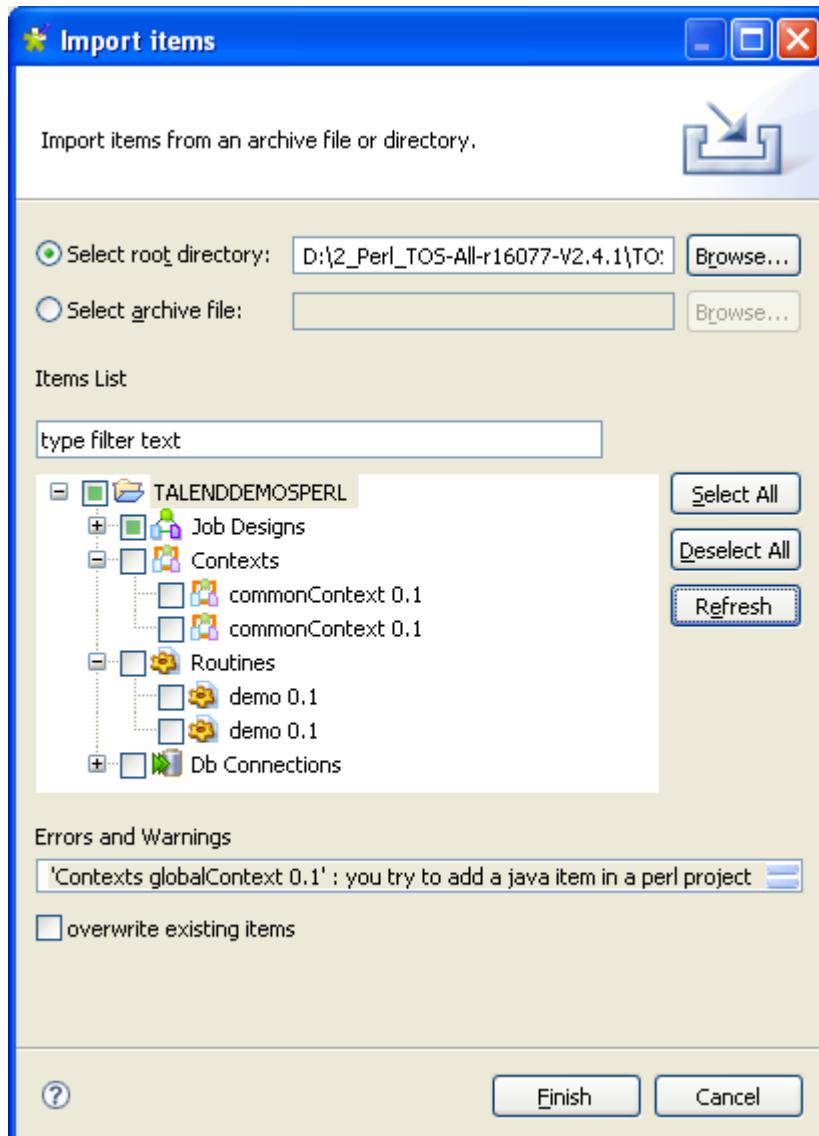
- Une boîte de dialogue vous demande de sélectionner le répertoire ou le fichier archive contenant les éléments que vous souhaitez extraire.
- Si vous avez exporté les éléments de votre référentiel local dans un fichier archive (fichiers source et scripts inclus), sélectionnez le dossier contenant l'archive 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.

Gestion des Jobs d'intégration de données

Import/export d'éléments ou de Jobs



- 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 leur nature.



S'il y a plusieurs versions d'un même élément, elles seront toutes importées dans le Projet en cours, à condition que vous n'ayez pas d'élément identique déjà existant.

5.2.2 Export de Job en Java

La fonction **Export Job** 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 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.

Gestion des Jobs d'intégration de données

Import/export d'éléments ou de Jobs



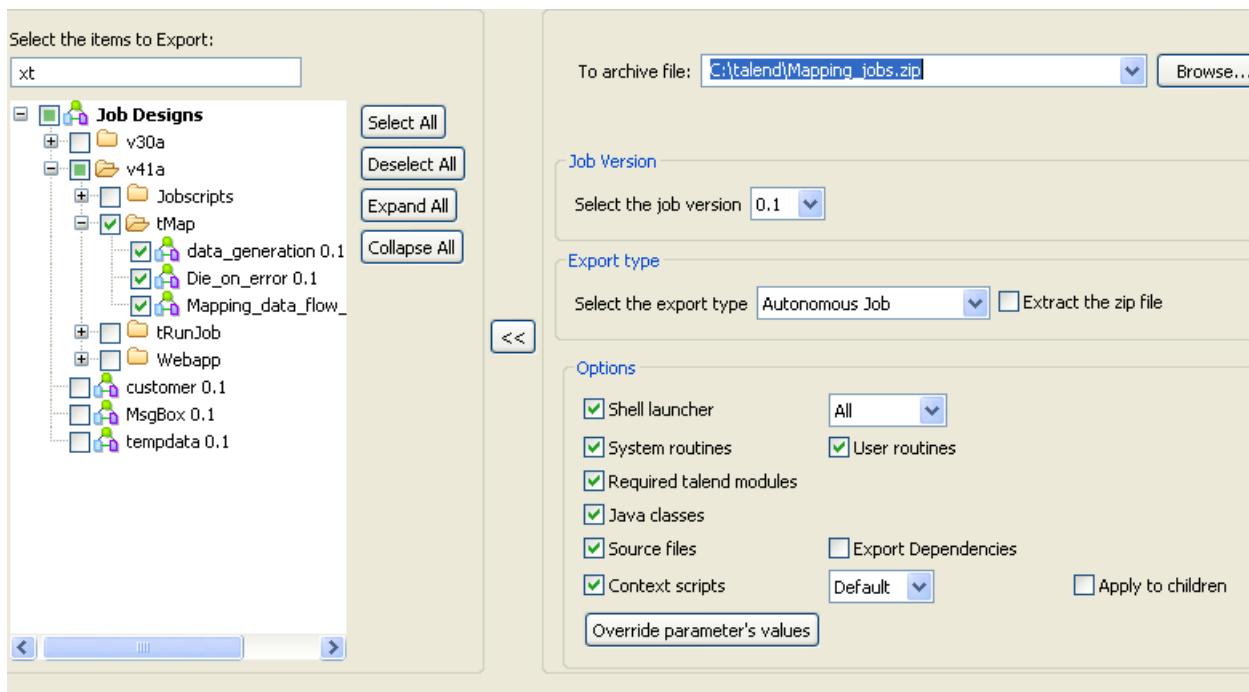
La boîte de dialogue [Export Job] n'est pas la même lorsque vous travaillez en Java ou en Perl.

Pour exporter des Job scripts :

- Dans l'arborescence **Repository**, cliquez-droit sur le Job que vous voulez exporter.
- Sélectionnez **Export Job** pour ouvrir la boîte de dialogue [**Export Job**].



Vous pouvez directement afficher ou cacher, dans la boîte de dialogue [**Export Job**], l'arborescence de tous les Jobs ayant été créés dans **Talend Open Studio**, en cliquant respectivement sur les boutons **>>** et **<<**. Les Jobs que vous avez précédemment sélectionnés dans l'arborescence du Studio s'affichent, précédés d'une case cochée. Cette fonctionnalité permet de modifier les éléments sélectionnés pour les exporter directement à partir de la boîte de dialogue, sans avoir à la fermer et retourner à l'arborescence **Repository** dans **Talend Open Studio**.



- Dans le champ **To archive file**, cliquez sur le bouton **Browse...** et parcourez jusqu'au répertoire dans lequel vous souhaitez sauvegarder votre Job exporté.
- Dans la zone **Job Version**, sélectionnez la version du Job que vous souhaitez exporter, si vous avez créé plusieurs versions de votre Job.
- Dans la liste **Export Type**, sélectionnez le type d'export. Les différents types sont : **Autonomous Job**, **Axis WebService (WAR)**, **Axis WebService (Zip)** et **JBoss ESB**.
- Dans la zone **Extract the zip file**, cochez la case pour exporter l'archive et sa version extraite de votre Job.
- Dans la zone **Options**, sélectionnez le type de fichiers que vous souhaitez ajouter à l'archive. Les options correspondant aux fichiers nécessaires à l'exécution du Job sont cochées par défaut, vous pouvez néanmoins les décocher en fonction de ce que vous souhaitez exporter.

- Pour exporter les fichiers .bat et/ou .sh nécessaires au lancement du Job exporté, cochez la case **Shell launcher**. Dans la liste correspondante, vous pouvez sélectionner :
 - **All** pour exporter les fichiers .bat et .sh,
 - **Unix** pour exporter le fichier .sh,
 - **Windows** pour exporter le fichier .bat.
- Pour exporter les routines système ou utilisateur utilisées dans le Job, cochez les cases **System routines** et **User routines**.
- Pour exporter les modules **Talend**, cochez la case **Required Talend modules**.
- Pour exporter les sources (les classes Java nécessaires à l'exécution du Job en Java), cochez la case **Java classes**.
- Pour exporter les sources utilisées par le Job lors de son exécution, notamment les fichiers .item, .properties et les sources Java et Talend, cochez la case **Source files**.



Si vous cochez la case **Source files**, vous pourrez réutiliser ce Job dans un **Talend Open Studio** installé sur une autre machine. Ces fichiers source sont uniquement utilisés dans **Talend Open Studio**

- Cochez la case **Export Dependencies** si vous souhaitez exporter les dépendances de votre job, notamment les contextes, les routines, les connexions, les bibliothèques externes requises, etc.
- Si vous cochez la case **Context script**, la totalité des fichiers de paramètres de contexte seront exportés et pas uniquement ceux que vous aurez sélectionnés dans la liste.
- Dans la liste située à coté de la case **Context script**, sélectionnez le contexte qui correspond à vos besoins, ainsi les fichiers .bat ou .sh contiendront les paramètres de contexte adéquats. Ensuite, vous pourrez, si vous le souhaitez, éditer les fichiers .bat ou .sh pour modifier manuellement le type de contexte.
- Cochez la case **Apply to children** si vous souhaitez appliquer le contexte sélectionné dans la liste à tous les Jobs fils.
- Cliquez sur le bouton **Override parameters' values**. Une fenêtre s'ouvre, dans laquelle vous pouvez mettre à jour les paramètres et les valeurs de contexte du Job sélectionné ou ajouter de nouveaux paramètres ou de nouvelles valeurs, selon vos besoins.
- Cliquez sur **Finish** pour valider les changements effectués, finaliser l'export et fermer la boîte de dialogue.

Un fichier zip des Jobs scripts est alors créé à l'endroit prédéfini.

Exporter un Job en Autonomous Job

Si vous exportez votre Job en **Autonomous Job** et que vous souhaitez le réutiliser 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. Ensuite, cliquez sur le bouton **Override parameters' values**, sous la case **Context script**. Une fenêtre s'ouvre alors et affiche tous les paramètres du contexte sélectionné. Dans cette fenêtre, vous pouvez configurer le contexte sélectionné selon vos besoins.

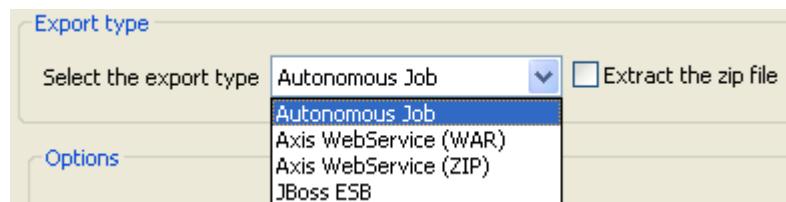
Tous les fichiers de paramètres de contexte sont exportés et pas uniquement celui que vous avez sélectionné dans la liste.



Après l'export, les informations de contexte sélectionnées sont stockées dans le fichier .bat/.sh. Les paramètres de contexte sont stockés dans le fichier de contextes **.properties**.

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 nécessaires à l'exécution ou au déploiement à partir d'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/NomAppWeb/services/NomJob?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. Pour un cas réel de création et d'export de Job en tant que service Web et de déploiement de ce Job à partir d'un navigateur Web, consultez *Exemple d'export de Job en tant que service Web, page 137*.

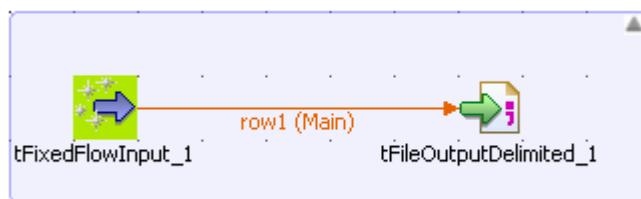
Le composant **tBufferOutput** a été conçu spécialement pour ce type de déploiement. Pour plus d'informations concernant ce composant, consultez le composant *tBufferOutput* du Guide de référence des composants [Talend Open Studio](#).

Exemple d'export de Job en tant que service Web

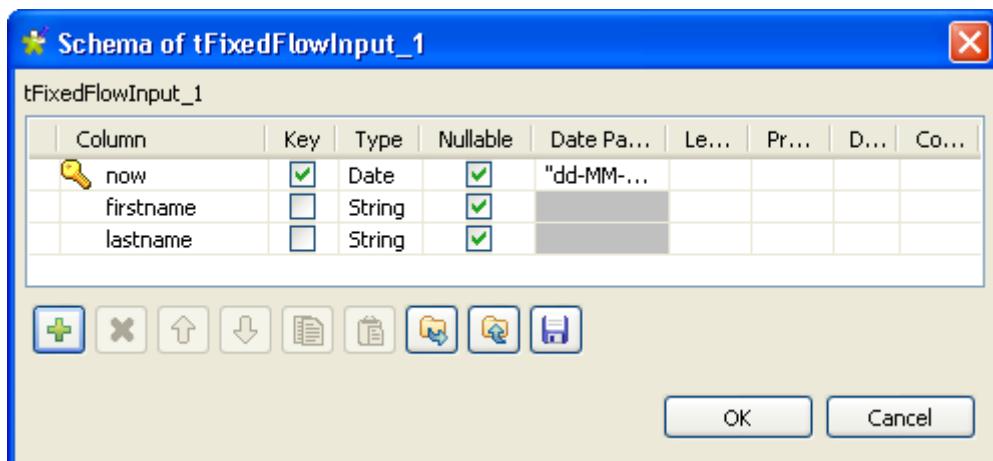
Ce scénario décrit, en premier temps, un simple Job créant un fichier .txt contenant la date en cours, ainsi que des noms et prénoms. En deuxième temps, ce Job est exporté en tant que service Web à partir d'un navigateur Web. Le Job exporté en tant que service Web renverra simplement le "retour d'appel" fourni par le système d'exploitation.

Créer le job :

- Déposez les composants suivants de la **Palette** dans l'espace de modélisation : **tFixedFlowInput** et **tFileOutputDelimited**.
- Connectez le **tFixedFlowInput** au **tFileOutputDelimited** à l'aide d'un lien **Row Main**.



- Dans l'espace de modélisation, sélectionnez le **tFixedFlowInput**.
- Cliquez sur la vue **Component** pour paramétriser les propriétés du **tFixedFlowInput**.
- Dans le champ **Schema**, sélectionnez l'option **Built-In** et cliquez sur le bouton [...] à côté du champ **Edit Schema** afin de décrire la structure des données que vous souhaitez créer à partir des variables internes. Dans ce scénario, le schéma est composé de trois colonnes : *now*, *firstname* et *lastname*.



- Cliquez sur le bouton [+] pour ajouter vos trois lignes de paramètres et définir vos variables.

Gestion des Jobs d'intégration de données

Import/export d'éléments ou de Jobs

- Cliquez sur **OK** pour fermer la boîte de dialogue et acceptez la propagation des modifications lorsque le message de confirmation vous le demande. Les trois colonnes définies apparaissent dans le tableau **Values** de l'onglet **Basic settings** du composant **tFixedFlowInput**.

Values

Column	Value
now	
firstname	
lastname	

- Cliquez sur la cellule **Value** de chaque colonne définie et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales.
- Dans la liste des variables globales, sélectionnez :
 - *TalendDate.getCurrentDate()* pour la colonne *now*,
 - *talendDatagenerator.getFirstName* pour la colonne *firstname*,
 - *talendDataGenerator.getLastName* pour la colonne *lastname*.
- Dans le champ **Number of rows**, saisissez le nombre de lignes à générer.

tFixedFlowInput_1

Basic settings

Schema: Built-In | Edit schema

Number of rows: 3

Values

Column	Value
now	TalendDate.getCurrentDate()
firstname	TalendDataGenerator.getFirstName
lastname	TalendDataGenerator.getLastName

- Dans l'espace de modélisation, sélectionnez le composant **tFileOutputDelimited**.
- Cliquez sur la vue **Component** du **tFileOutputDelimited** et dans le champ **File name**, sélectionnez le fichier de sortie. Paramétrez les autres propriétés du composant, si nécessaire.

tFileOutputDelimited_1

Basic settings

Property Type: Built-In

File Name: "C:/Output/out.csv"

Row Separator: "\n" | Field Separator: ";"

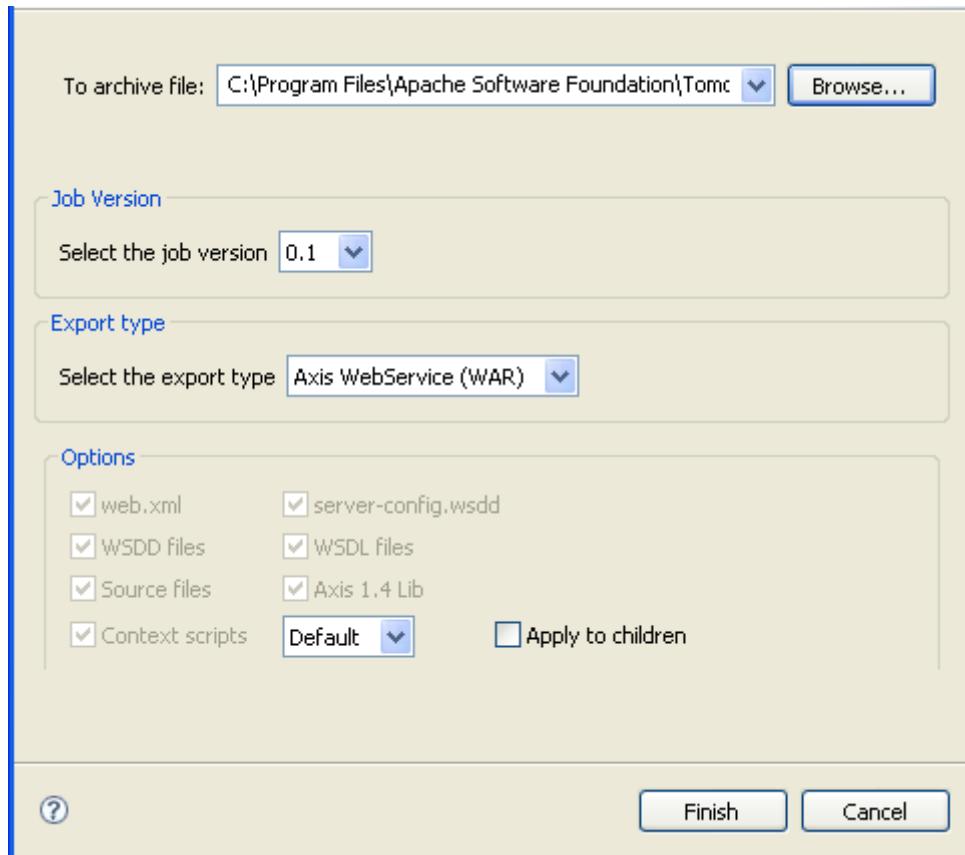
Append | Include Header

Schema: Built-In | Edit schema | Sync columns

Si vous appuyez sur **F6** pour exécuter le job, trois lignes contenant la date en cours et des noms et prénoms seront écrits dans le fichier de sortie défini.

Exporter le Job en tant que service Web :

- Dans la vue **Repository**, cliquez-droit sur le Job que vous venez de créer et sélectionnez **Export Job Scripts**.
- La boîte de dialogue **Export Job Scripts** suivante apparaît :

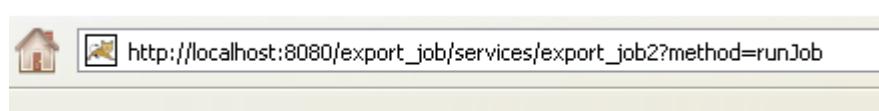


- Dans le champ **To archive file**, cliquez sur le bouton **Browse...** et sélectionnez un répertoire dans lequel enregistrer le fichier archive contenant votre Job.
- Dans la liste **Job Version**, vous pouvez renseigner la version que vous souhaitez exporter, si vous avez créé plusieurs versions de votre Job.
- Dans la liste **Export type**, sélectionnez le type d'export que vous souhaitez utiliser dans votre application Web (WAR pour cet exemple) et cliquez sur **Finish**. La boîte de dialogue [**Export Job Scripts**] disparaît.
- Copiez le dossier War et collez-le dans le répertoire du serveur d'application Tomcat.

Invoquer le Job à partir d'un navigateur Web :

Saisissez l'URL suivante dans votre navigateur :

`http://localhost:8080//export_job/services/export_job2?method=runJob` dans lequel “export_job” correspond au nom de l'application Web utilisée pour le déploiement sur le Tomcat et “export_job2” correspond au nom de votre job.



Cliquez sur **Enter** pour exécuter le Job à partir de votre navigateur Web.

Ce fichier XML ne semble pas avoir d'information de style lui

```
- <soapenv:Envelope>
  - <soapenv:Body>
    - <runJobReturn xsi:type="ns1:runJobReturn">
      - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
        <ns1:item xsi:type="xsd:string">0</ns1:item>
      </ns1:item>
    </runJobReturn>
  </soapenv:Body>
</soapenv:Envelope>
```

Le code retourné à partir de votre application Web est 0 lorsqu'il n'y a pas d'erreur et 1 si une erreur est apparue.

Pour un cas réel de création et d'export d'un Job en tant que service Web via le composant **tBufferOutput**, consultez le *Scénario 2 : Mettre les données de sortie en mémoire tampon du serveur d'application Web* du [Guide de Référence des Composants Talend Open Studio](#).

Exporter un Job en tant que JBoss ESB

Talend Open Studio vous permet d'exposer des Jobs **Talend** en tant que services dans JBoss ESB (Enterprise Service Bus) afin d'exécuter ces Jobs dans le moteur de messagerie (le bus).



*Si vous souhaitez déployer et exécuter votre Job sur un serveur JBoss ESB, assurez-vous que le jar spécifique à JBoss ESB est bien installé dans la bibliothèque Java et apparaît comme tel dans la vue Modules de **Talend Open Studio**. Pour plus d'informations concernant la vue Modules, consultez *Installer des modules externes*, page 111.*

Dans la boîte de dialogue **[Export Job Scripts]**, vous pouvez modifier le type d'export afin d'exporter le Job sélectionné dans une archive ESB. Le Job contenu dans cette archive peut ensuite être automatiquement déployé sur le serveur JBoss.

Pour exporter un Job en ESB :

- Dans la zone **Job Version**, sélectionnez la version du Job que vous souhaitez exécuter sur le serveur JBoss ESB.
- Dans la liste **Select export type** de la zone **Export type**, sélectionnez **JBoss ESB**.
- Dans la zone **Option**, sélectionnez le type de fichiers que vous souhaitez ajouter à l'archive. Lorsque plusieurs contextes sont proposés dans la liste **Context script**, sélectionnez celui qui correspond à vos besoins.
- Cochez la case **Apply to children** si vous souhaitez appliquer le contexte sélectionné dans la liste à tous les Jobs fils.
- Pour exporter les sources utilisées par le Job durant son exécution, y compris les fichiers .item, .properties et les sources Java des Jobs et des routines, cochez la case **Source files**.



Si vous cochez la case **Source files**, vous pouvez réutiliser le Job exporté dans un **Talend Open Studio** installé sur une autre machine. Ces fichiers source sont utilisés uniquement dans **Talend Open Studio**.

- Cochez la case **Export Dependencies** si vous souhaitez exporter les dépendances de votre Job, c'est-à-dire les contextes, les routines, les connexions, etc.
- Dans la liste **ESB Export type**, sélectionnez soit *JBoss MQ*, soit *JBoss Messaging*.
- Dans le champ **Service name**, saisissez le nom du service sur lequel déployer votre Job.
- Dans le champ **Category**, saisissez le nom de la catégorie du service sur lequel le Job sera déployé.
- Dans le champ **Message Queue Name**, saisissez le nom de la queue utilisée pour le déploiement du Job.
- Cliquez sur le bouton **Browse...** à côté du champ **To archive file** et définissez le chemin d'accès au fichier archive dans lequel vous souhaitez exporter le Job.
- Cliquez sur **Finish**.
La boîte de dialogue se ferme. Un indicateur de progression s'affiche et vous montre le pourcentage d'avancement de l'opération d'export. Le Job est exporté dans l'archive sélectionnée.

Au lancement du serveur, une fois l'archive ESB copiée dans le répertoire de déploiement, le Job est automatiquement déployé et prêt à être exécuté sur le serveur ESB.

Exporter un Job en tant que Petals ESB

Avec **Talend Open Studio**, il est possible d'exporter des Jobs **Talend** en tant que services dans Petals ESB, afin de les exécuter dans le serveur de messagerie applicative (le bus).

Intégrer Petals ESB dans Talend

Talend fournit une méthode aisée pour exposer des services dans Petals ESB, et permet :

- l'intégration de l'application dans le bus : ceci permettra l'intégration des systèmes et applications dans toute l'entreprise.
- l'interaction des services : l'ESB établit la connectivité entre les services, c'est-à-dire qu'il permet la communication entre les services, même si ceux-ci possèdent différentes interfaces.

L'Intégration Métier Java (Java Business Integration, JBI) est l'approche utilisée pour implémenter une architecture orientée service (service-oriented architecture, SOA) et exporter des Jobs **Talend** dans Petals ESB.

Petals ESB est complété par des Binding Components (BC) et le Service Engine (SE) **Talend** pour fournir : d'abord, l'accès aux méthodes inhérentes aux différents types de services comme le transfert de fichier(s), les services Web, le MOM, et ensuite, le moteur permettant de déployer le service. Pour plus d'informations concernant les interactions entre Petals et les Jobs **Talend**, consultez :

<http://doc.petalslink.com/display/petalsesb/A+Simple+Talend+Job>.

Avec l'intégration de **Talend** et Petals ESB, vous pouvez exécuter les Jobs créés dans **Talend Open Studio** dans Petals ESB. Pour plus d'informations, consultez *Exporter un Job*

en tant que Petals ESB, page 141. Différents mécanismes sont fournis pour passer les informations et les données à un Job et pour récupérer des informations et des données à partir d'un Job.

A l'aide de **Talend Open Studio** et de Petals ESB, vous pouvez exécuter un Job n'ayant aucune interaction spécifique avec Petals. Vous pouvez :

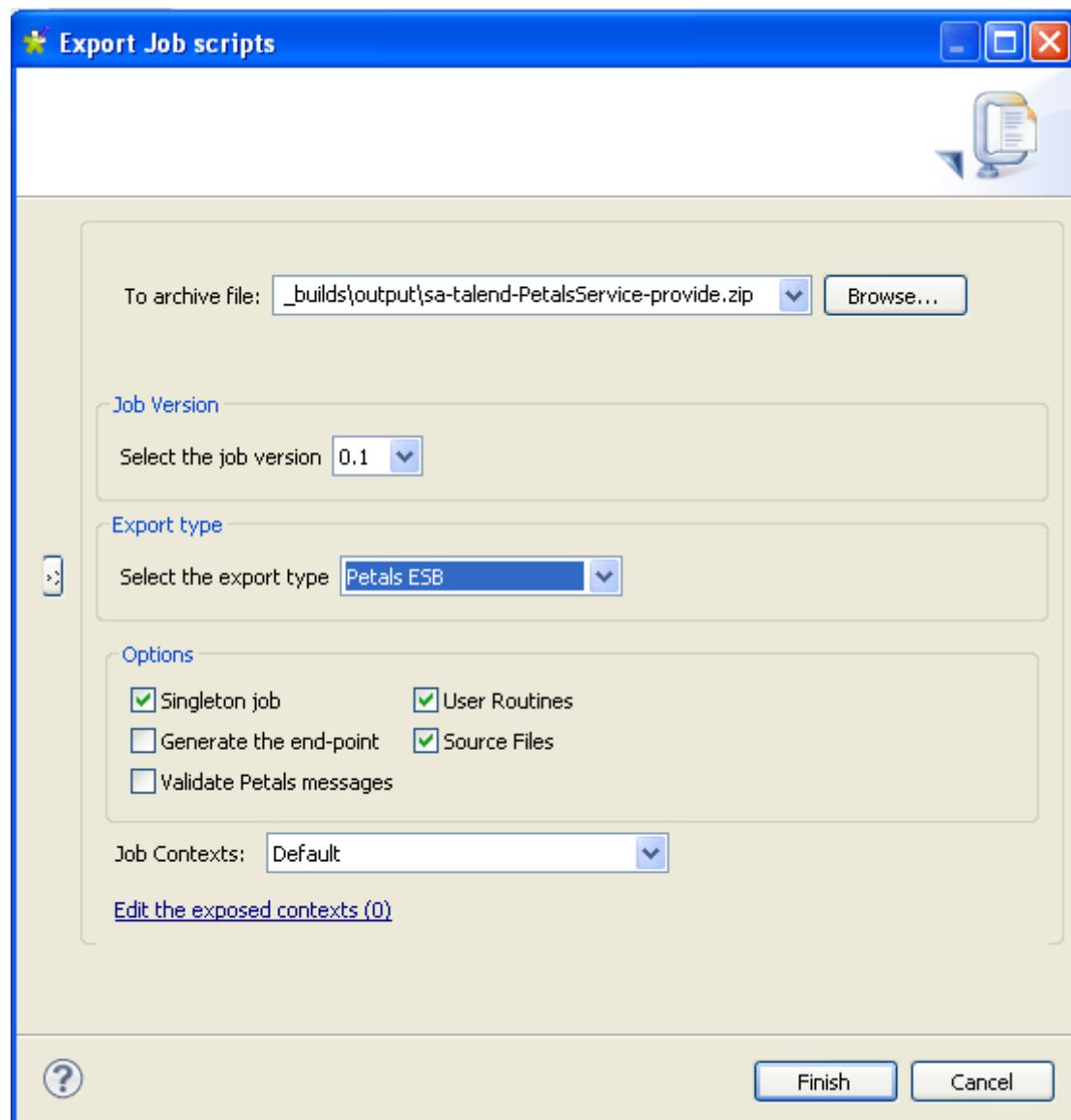
- exposer un contexte en tant que paramètre dans le service WSDL,
- passer des fichiers joints à un Job,
- passer des paramètres et options natifs à un Job,
- obtenir les résultats d'exécution du Job.

Comment exporter des Jobs vers Petals ESB

Dans la boîte de dialogue [**Export Job Scripts**], vous pouvez exporter un Job sélectionné en tant qu'archive Petals ESB. Vous pouvez exécuter le Job exporté dans le bus (le serveur de messagerie applicative).

Pour exporter un Job en tant qu'archive Petals ESB, procédez comme suit :

- Dans la vue **Repository**, cliquez-droit sur le Job que vous souhaitez exporter, puis sélectionnez **Export Job Scripts** dans le menu contextuel.
La boîte de dialogue [**Export Job Scripts**] s'ouvre.



- Dans le champ **To archive file**, parcourez votre répertoire afin de définir le chemin d'accès à l'archive dans laquelle vous souhaitez exporter le script.
- Dans la liste **Select the job version**, sélectionnez la version du Job que vous voulez exporter.
- Dans la liste **Select export type**, dans la zone **Export type**, sélectionnez **Petals ESB**.
Les trois options suivantes, dans la zone **Options** sont sélectionnées par défaut : **Singleton job**, **User Routines** et **Source file**. Vous pouvez sélectionner toute autre option qui vous paraît nécessaire.

Le tableau ci-dessous vous explique les options d'export :

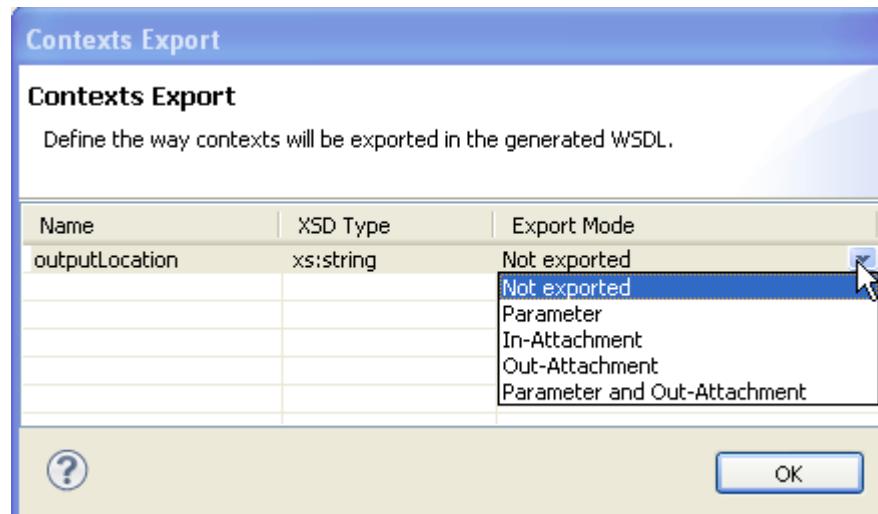
Option	Description
Singleton job	Exporter le Job en tant que singleton : un Job singleton ne peut avoir qu'une seule instance lancée à la fois sur un serveur de messagerie applicative Talend dans Petals ESB.

Gestion des Jobs d'intégration de données

Import/export d'éléments ou de Jobs

Option	Description
Generate the end-point	Génère l'end-point au moment du déploiement. Si cette option n'est pas sélectionnée, le nom de l'end-point sera le même nom que le Job, avec le suffixe Endpoint.
Validate Petals messages	Valide tous les messages / requêtes par rapport au WSDL. 💡 Sélectionner cette option réduit les performances de votre système (accès au disque).
User routines	Embarque les routines personnalisées dans le service-unit.
Source files	Embarque les fichiers source dans le service-unit généré.
Jobs contexts	Liste dans laquelle sélectionner le contexte qui sera utilisé par défaut dans le Job.

- Dans la boîte de dialogue [Export Job], cliquez sur le lien **Edit the exposed contexts** afin d'ouvrir la boîte de dialogue [Context Export].



La boîte de dialogue affichera une liste des toutes les variables de contexte utilisées dans le Job exporté. Vous pouvez ici spécifier comment chaque variable de contexte doit être exportée dans le fichier WSDL généré.

- Cliquez dans le champ **Export Mode** et sélectionnez dans la liste le mode d'export pour chaque variable de contexte.

Le tableau ci-dessous présente les différentes options d'export :

Export Mode	Description
Not exported	Le contexte n'est pas exporté (non visible en tant que paramètre). Le contexte pourra toujours être exporté en utilisant les paramètres natifs (options) du Job.
Parameter	Le contexte est exporté en tant que paramètre dans l'opération WSDL.
In-Attachment	Le contexte transmettra l'emplacement d'un fichier temporaire dont le contexte était joint au message d'entrée.

Export Mode	Description
Out-Attachment	<p>Le contexte sera lu après exécution du Job.</p> <ul style="list-style-type: none"> -Ce contexte doit indiquer un fichier, -Le contenu du fichier sera lu par le serveur de messagerie applicative et joint à la réponse, -Le nom du contexte sera utilisé comme nom de pièce jointe, -Le fichier sera supprimé par le serveur de messagerie applicative juste après le chargement de son contenu.
Parameter and Out-Attachment	<p>Cette option est un mélange des modes Parameter et Out-Attachment.</p> <ul style="list-style-type: none"> -Le contexte est exposé en tant que paramètre, Il sera également lu après exécution du Job, -Le fichier sera supprimé dans tous les cas, -L'avantage de ce mode d'export est la définition dynamique du fichier de sortie.

- Cliquez sur **OK** afin de valider votre choix et fermer la boîte de dialogue [**Context Export**].
- Dans la boîte de [**Export Job Scripts**], cliquez sur **Finish**.
La boîte de dialogue se ferme. Un indicateur de progression s'affiche, afin de montrer le pourcentage d'avancement de l'export. Le Job est exporté dans l'archive sélectionnée.

Le Job **Talend** est maintenant exposé en tant que service dans l'ESB Petals et peut être exécuté dans le bus.

5.2.3 Export de Job scripts en Perl

La fonction **Export Job** 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.



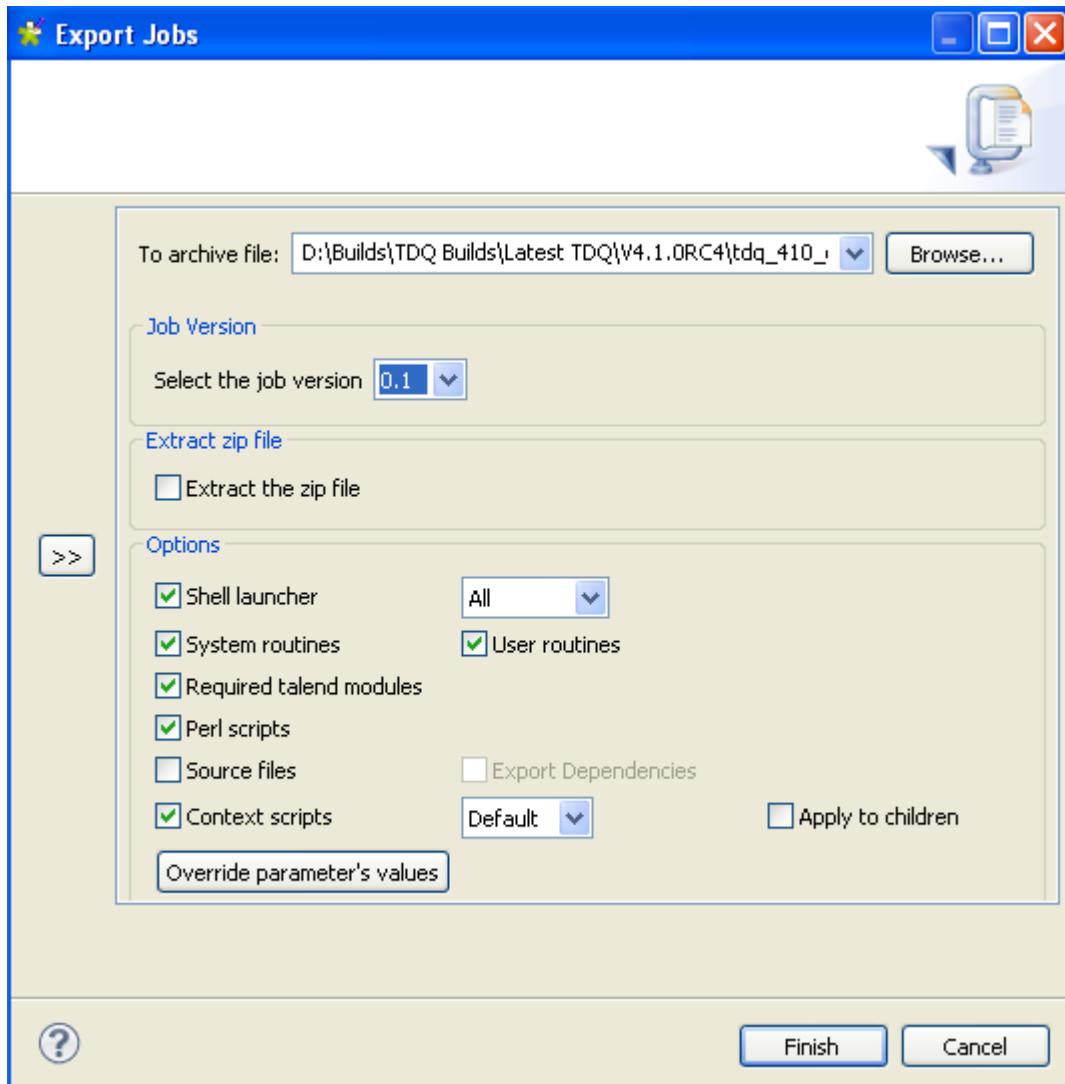
La boîte de dialogue [**Export Job**] n'est pas la même lorsque vous travaillez en Java ou en Perl.

Pour exporter des Job scripts :

- Cliquez-droit sur le Job concerné à partir de l'arborescence **Repository**.
- Sélectionnez **Export Job** pour ouvrir la boîte de dialogue [**Export Job**].



Vous pouvez directement afficher ou cacher, dans la boîte de dialogue [**Export Job**], l'arborescence de tous les Jobs ayant été créés dans **Talend Open Studio**, en cliquant respectivement sur les boutons et . Les Jobs que vous avez précédemment sélectionnés dans l'arborescence du Studio s'affichent, précédés d'une case cochée. Cette fonctionnalité permet de modifier les éléments sélectionnés pour les exporter directement à partir de la boîte de dialogue, sans avoir à la fermer et retourner à l'arborescence **Repository** dans **Talend Open Studio**.



- Dans le champ **To archive file**, cliquez sur **Browse...** et sélectionnez un répertoire dans lequel enregistrer le fichier archive contenant votre job.
- Dans la liste **Select the job version**, sélectionnez la version du Job que vous souhaitez exporter, si vous avez créé plusieurs versions de votre job.
- Dans la zone **Extract the zip file**, cochez la case pour exporter le fichier archive et sa version extraite de votre job.
- Dans la zone **Options**, sélectionnez le type de fichiers que vous souhaitez ajouter à l'archive. Les options correspondant aux fichiers nécessaires à l'exécution du Job sont cochées par défaut, vous pouvez néanmoins les décocher en fonction de ce que vous souhaitez exporter.
- Pour exporter les fichiers .bat et .sh permettant de lancer l'exécution du Job, cochez la case **Shell launcher**.
- Pour exporter les routines système ou utilisateur utilisées dans le Job, cochez les cases **System routines** et **User routines**.
- Pour exporter les modules Talend, cochez la case **Required Talend modules**.
- Pour exporter le fichier .jar contenant les classes Perl nécessaires à l'exécution du Job en Perl, cochez la case **Perl scripts**.

- Pour exporter les sources utilisées par le Job lors de son exécution, notamment les fichiers .item, .properties, cochez la case **Source files**.



ISi vous cochez la case **Source files**, vous pourrez réutiliser ce Job dans un **Talend Open Studio** installé sur une autre machine. Ces fichiers source sont uniquement utilisés dans **Talend Open Studio**.

- Cochez la case **Export Dependencies** si vous souhaitez exporter les dépendances de votre job, notamment les contextes, les routines, les connexions, etc.
- Si vous cochez la case **Context script**, la totalité des fichiers de paramètres de contexte seront exportés et pas uniquement ceux que vous aurez sélectionnés dans la liste.
- Dans la liste située à coté de la case **Context script**, sélectionnez le contexte qui correspond à vos besoins, ainsi les fichiers .bat ou .sh contiendront les paramètres de contexte adéquats. Ensuite, vous pourrez, si vous le souhaitez, éditer les fichiers .bat ou .sh pour modifier manuellement le type de contexte.
- Cochez la case **Apply to children** si vous souhaitez appliquer le contexte sélectionné dans la liste à tous les Jobs fils.



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.

Si vous souhaitez modifier les paramètres de contexte, éditez le fichier **.properties** de contexte.

- Cliquez sur **Finish** pour valider les modifications, finaliser l'export et fermer la boîte de dialogue.

Un fichier zip des Jobs scripts est alors créé à l'endroit prédéfini.

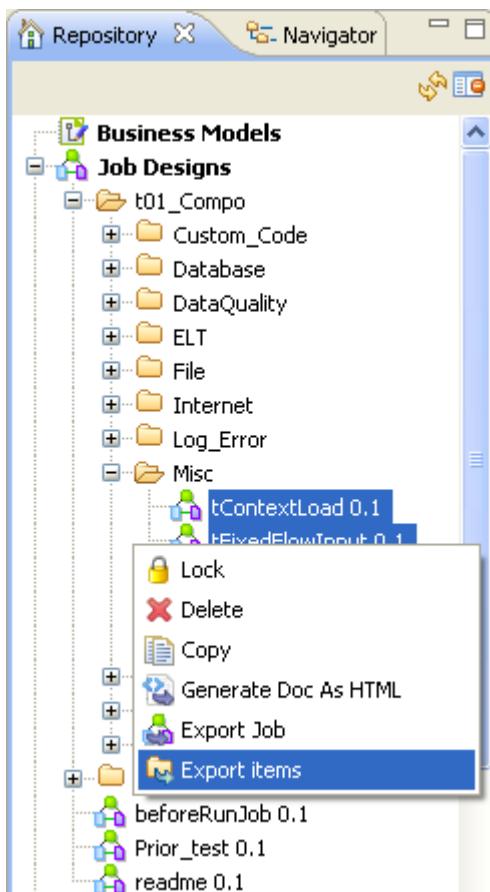


Vous ne pouvez pas importer de Job scripts d'une version différente de Talend Open Studio que celle utilisée lors de la création du job. Pour réutiliser des Jobs d'une version antérieure de Talend Open Studio, il faut utiliser la fonction Import Project. Consultez Importer un projet, page 17.

5.2.4 Export d'éléments

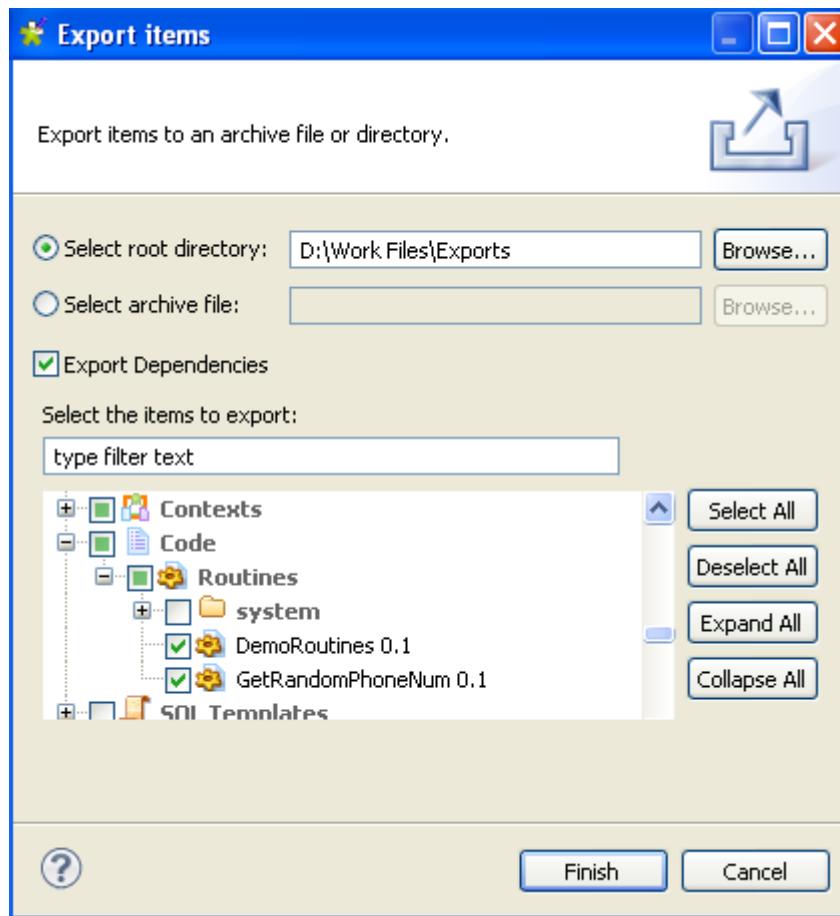
Vous pouvez exporter plusieurs éléments du **Repository** dans un répertoire ou dans un fichier archive. Vous pouvez donc exporter des métadonnées, notamment les informations de connexion aux bases de données (**DB connection**) ou de la **Documentation** en même temps que votre **Job** ou votre **Business Model**, par exemple.

- Dans le **Repository**, sélectionnez les éléments que vous souhaitez exporter
- Pour une sélection multiple, maintenez la touche **Ctrl** enfoncee puis sélectionnez les éléments souhaités.



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 :



Au besoin, vous pouvez sélectionner des éléments supplémentaires à exporter.

- Sélectionnez le répertoire dans lequel vous souhaitez enregistrer les éléments exportés en cliquant sur **Browse....** Sinon, définissez le fichier archive dans lequel les fichiers des éléments sélectionnés seront compressés.



Si vous avez plusieurs versions d'un même élément, elles seront toutes exportées.



Cochez la case **Export Dependencies** si vous souhaitez définir et exporter les dépendances des routines avec les Jobs que vous exportez. Par défaut, toutes les routines utilisatrices sont sélectionnées. Pour plus d'informations, consultez *Définition des routines*, page 266

- Cliquez sur **Finish** pour fermer la boîte de dialogue et exporter les éléments.

5.2.5 Changer les paramètres de contexte d'un Job script

Comme expliqué dans les sections *Export de Job en Java*, page 133 et *Export de Job scripts en Perl*, page 145, vous pouvez éditer les paramètres de contexte :

Si vous souhaitez changer le contexte sélectionné, il vous suffit d'éditionner les fichiers .bat ou .sh et de modifier le paramètre suivant : `--context=Prod`, pour le contexte correspondant.

Si vous souhaitez changer des paramètres particuliers d'un contexte, éditez le fichier .bat ou .sh et ajoutez le paramètre dont vous avez besoin parmi ceux du tableau suivant :

Pour...	Paramètre
Changer la <i>valeur1</i> pour le paramètre <i>key1</i>	--context_param key1= <i>valeur1</i>
Changer <i>valeur1</i> et <i>valeur2</i> des paramètres respectifs <i>key1</i> et <i>key2</i>	--context_param key1= <i>valeur1</i> --context_param key2= <i>valeur2</i>
Changer une valeur contenant des caractères d'espace comme dans les chemins d'accès	--context_param key1="chemin d'accès"

5.3 Gestion des éléments du Repository

Talend Open Studio vous permet d'éditer les éléments centralisés dans le Repository et de mettre à jour les Jobs qui utilisent ces éléments.

5.3.1 Gérer les mises à jour des éléments du Repository

Vous pouvez mettre à jour les paramètres des métadonnées, des contextes ou des joblets centralisés à chaque fois dans la vue **Repository**, afin d'actualiser la connexion à la base de données ou par exemple les détails du groupe de contexte.

Lorsque vous modifiez l'un des paramètres d'une entrée de la vue **Repository**, tous les Jobs utilisant cette entrée seront impactés par cette modification. C'est pourquoi le système vous proposera de propager ces modifications à tous les Jobs utilisant l'entrée du Repository.

Les sections ci-dessous expliquent comment modifier les paramètres d'entrée du Repository et comment propager les modifications pour tous les Jobs ou une partie d'entre eux utilisant l'entrée en question.

Modifier un élément du Repository

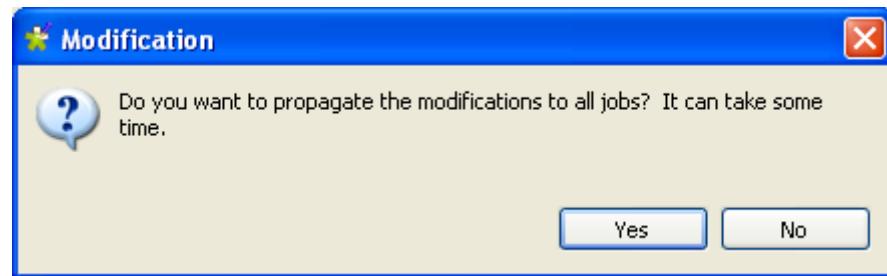
Pour actualiser les paramètres d'un élément du Repository, procédez comme suit :

- Développez le noeud **Metadata**, **Contexts** ou **Joblets Designs** dans la vue **Repository** et parcourez jusqu'à l'entrée correspondante que vous voulez actualiser.
- Cliquez-droit sur l'entrée et sélectionnez dans le menu contextuel l'option **edit** correspondante.

L'assistant respectif apparaît, vos permettant d'éditer chaque étape de la définition des paramètres d'entrée.

Pour mettre à jour les paramètres d'entrée, vous devez propager les modifications de plusieurs Jobs ou de tous les Jobs utilisant cette entrée.

Une boîte de dialogue apparaît automatiquement à la fin de votre mise à jour ou de vos modifications, quand vous cliquez sur le bouton **Finish** dans l'assistant.



- Cliquez sur **Yes** (Oui) pour fermer la boîte de dialogue et exécuter les modifications apportées à tous les Jobs concernés. Pour plus d'informations concernant la première méthode de propagation des modifications, consultez *Mettre à jour manuellement les Jobs impactés, page 152.*
- Cliquez sur **No** (Non) si vous voulez fermer la boîte de dialogue sans propager les changements. Cela vous permettra de propager manuellement, une par une, les modifications apportées aux Jobs concernés. Pour plus d'informations concernant la deuxième méthode de propagation des modifications, consultez *Mettre à jour manuellement les Jobs impactés, page 152.*

Mettre à jour automatiquement les Jobs impactés

Après avoir mis à jour les paramètres de chaque élément centralisé dans la vue **Repository** et utilisés dans différents Jobs, une boîte de dialogue s'ouvre et vous demande si vous souhaitez propager les modifications à tous les Jobs utilisant ces paramètres.

Cliquez sur **Yes** (Oui), si vous voulez que le système cherche dans votre Repository les Jobs impactés par les modifications que vous venez de faire. Ceci a pour but de propager automatiquement la mise à jour dans tous les Jobs (ouverts ou non), en un seul clic.

La boîte de dialogue [**Update Detection**] s'ouvre et affiche tous les Jobs impactés par les changements.

Items	Operations	Remarks
<input checked="" type="checkbox"/> Job tExtractXMLField_prelim 0.1 (*)	-	-
<input checked="" type="checkbox"/> tMysqlOutput_1	-	-
<input checked="" type="checkbox"/> Property	Update from repository	DB (MYSQL):localhost
<input type="checkbox"/> Job tExtractXMLField 0.1 (*)	-	-
<input type="checkbox"/> "customerdetails" (tMysqlInput_1)	-	-
<input type="checkbox"/> Property	Update from repository	DB (MYSQL):localhost



Vous pouvez ouvrir la boîte de dialogue **[Update Detection]** à tout moment, si vous cliquez-droit sur l'élément centralisé dans le Repository et que vous sélectionnez **Manage Dependencies** dans le menu contextuel. Pour plus d'informations, consultez *Mettre à jour manuellement les Jobs impactés*, page 152.

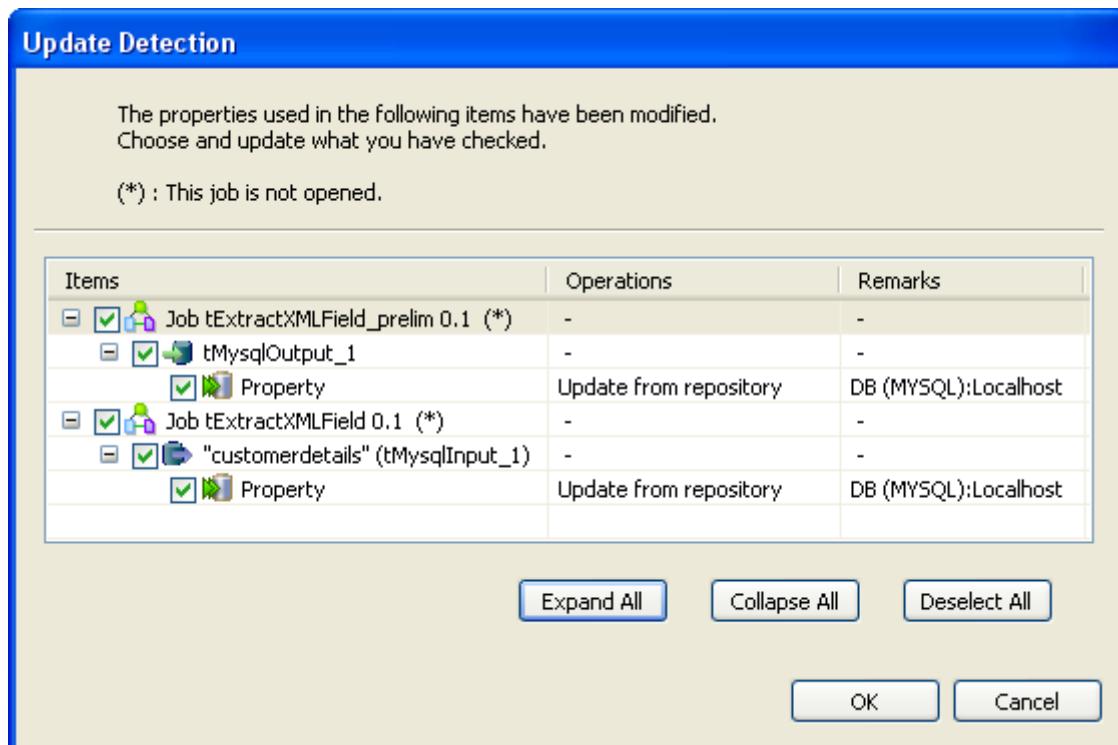
- Décochez la case correspondant aux Jobs que vous ne souhaitez pas mettre à jour. Vous pouvez les mettre à jour manuellement ultérieurement. Pour plus d'informations, consultez *Mettre à jour manuellement les Jobs impactés*, page 152.
- Cliquez sur **OK** pour fermer la boîte de dialogue et mettre à jour les Jobs sélectionnés.

Mettre à jour manuellement les Jobs impactés

Avant de propager les modifications des **Métadonnées** ou **Contextes** du **Repository** dans tous vos jobs, vous souhaitez peut-être voir les impacts que cela peut entraîner. Pour ce faire, effectuez les opérations suivantes :

- Dans la vue **Repository**, développez le noeud contenant l'entrée dans laquelle vous souhaitez vérifier les Jobs l'utilisant.
- Puis cliquez-droit sur cette métadonnée ou ce contexte.
- Sélectionnez l'option **Detect Dependencies** dans le menu contextuel.

Une barre de progression indique le processus de vérification de tous les Jobs utilisant les paramètres de métadonnée ou de contexte modifiés. Une boîte de dialogue s'ouvre et affiche tous les Jobs utilisant cet élément mis à jour.



- Cochez ou décochez les Jobs à mettre à jour.
- Cliquez sur **OK** lorsque vous avez sélectionné les Jobs que vous souhaitez mettre à jour.



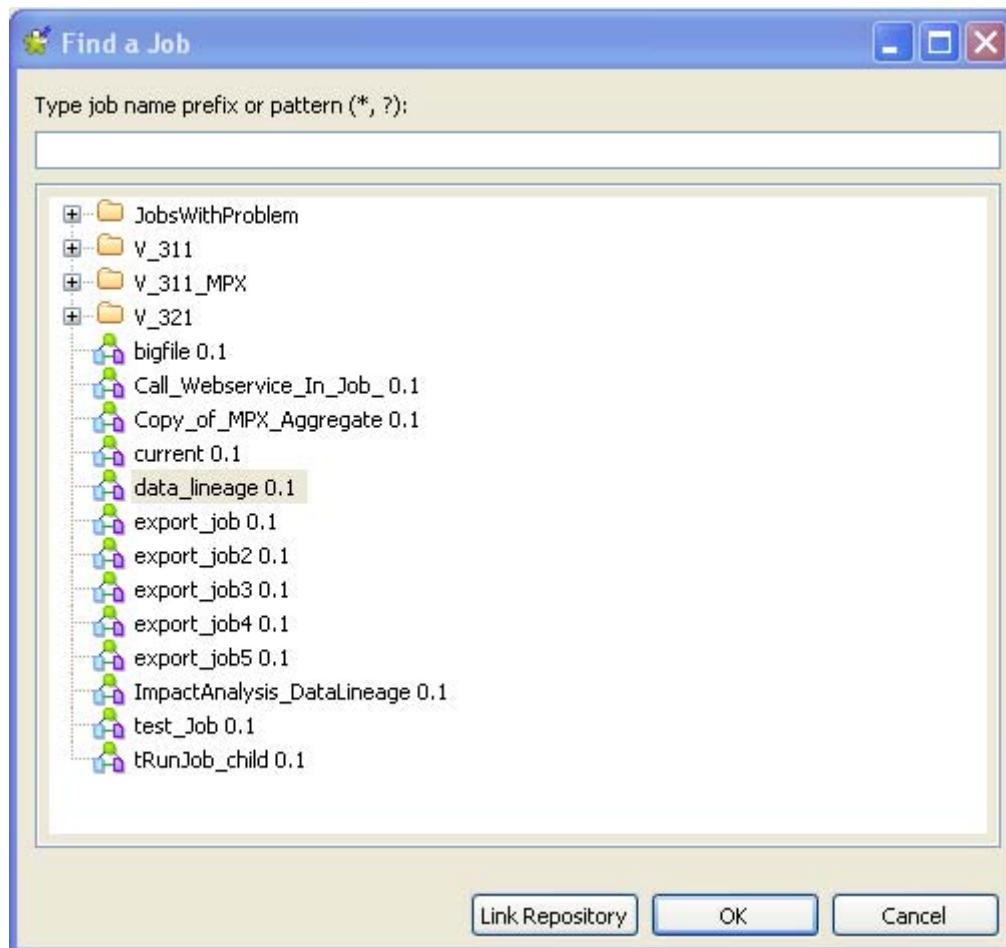
Les Jobs que vous n'aurez pas mis à jour passeront automatiquement en **Built-in**, puisque le lien vers le **Repository** ne pourra pas être maintenu. Cependant, les paramètres resteront les mêmes que ceux définis avant les modifications.

5.4 Recherche d'un Job dans le référentiel

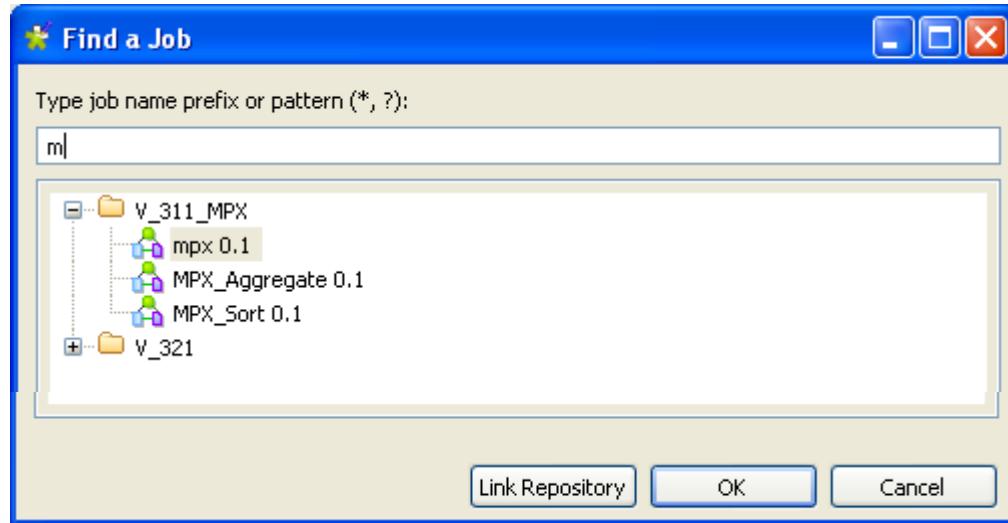
Si vous souhaitez ouvrir un Job précis dans **Talend Open Studio** et que vous n'arrivez pas à le retrouver dans la vue **Repository**, cliquez simplement sur le bouton de la barre d'outils.

Pour chercher un Job dans la vue **Repository** :

- Dans la barre d'outils de **Talend Open Studio**, cliquez sur le bouton pour ouvrir la boîte de dialogue **[Find a Job]** listant automatiquement tous les Jobs créés dans le Studio courant.



- Saisissez le nom ou une partie du nom du Job dans le champ en haut de la fenêtre. Lorsque vous commencez à saisir le nom dans le champ, la liste des Jobs est automatiquement mise à jour afin de n'afficher que les Jobs dont le nom correspond aux lettres saisies.



- Sélectionnez le Job souhaité dans la liste et cliquez sur **Link Repository** pour afficher automatiquement le Job sélectionné dans l'arborescence de la vue **Repository**.
- Si nécessaire, cliquez sur **Cancel** pour fermer la boîte de dialogue et cliquez-droit sur le Job sélectionné dans le **Repository** pour effectuer l'une des opérations disponibles dans le menu contextuel.
- Sinon, cliquez sur **Ok** pour fermer la boîte de dialogue et ouvrir le Job sélectionné dans l'espace de modélisation.

5.5 Gestion des versions d'un Job

Lorsque vous créez un Job dans **Talend Open Studio**, son numéro de version par défaut est 0.1 : 0 correspond à la version majeure et 1 à la version mineure.

Si vous souhaitez créer plusieurs versions d'un même Job :

- Veillez à fermer votre Job, sinon ses propriétés seront en lecture seule et vous ne pourrez donc pas les modifier.
- Cliquez-droit sur le libellé du Job dans le **Repository** et sélectionnez **Edit properties** dans le menu contextuel.
- La boîte de dialogue [**Edit properties**] s'ouvre.
- Dans le champ **Version**, cliquez sur le bouton **[M]** pour incrémenter la version majeure et sur le bouton **[m]** pour incrémenter la version mineure.
- Cliquez sur **Finish** pour valider.



Par défaut, lorsque vous ouvrez un Job, vous ouvrez sa dernière version.
Les versions précédentes sont en lecture seule et ne peuvent donc plus être modifiées.

Pour changer la version du Job, vous pouvez également :

- Fermer votre Job s'il est ouvert dans l'espace de modélisation graphique, sinon ses propriétés seront en lecture seule et vous ne pourrez pas les modifier.
- Dans la vue **Repository**, cliquez-droit sur votre Job et sélectionnez **Open another version** dans la liste déroulante.

- Dans la boîte de dialogue, cochez la case **Create new version and open it**, puis cliquez sur le bouton **M** pour incrémenter la version majeure, ou sur le bouton **m** pour incrémenter la version mineure.
- Cliquez sur **Finish** pour valider la modification et ouvrir la nouvelle version de votre Job.

Vous pouvez également sauvegarder un Job et incrémenter sa version simultanément, en cliquant sur **File>Save as...** dans le menu contextuel.



Cette option n'écrase pas votre Job actuel, mais l'enregistre "en tant que nouveau Job" avec une nouvelle version.

Vous pouvez accéder à la liste des différentes versions de votre Job et effectuer certaines opérations. Pour ce faire :

- Dans la vue **Repository**, sélectionnez le Job dont vous souhaitez consulter les versions.
- Cliquez sur la vue **Job** et sur l'onglet **Version**.
- La liste des versions apparaît sous forme de tableau.
- Cliquez-droit sur la version que vous souhaitez consulter.
- Sélectionnez l'option :

Sélectionnez	Pour...
Edit Job	ouvrir la dernière version du Job.  Cette option n'est disponible que lorsque vous sélectionnez la dernière version du Job.
Read job	consulter le Job en lecture seule.
Open Job Hierarchy	consulter la hiérarchie du Job.
Edit properties	éditer les propriétés du Job. Note : le Job ne doit pas être ouvert dans l'espace de modélisation graphique, sinon il sera en lecture seule.  Cette option n'est disponible que lorsque vous sélectionnez la dernière version du Job.
Run job	exécuter le Job.

Vous pouvez également gérer la version de plusieurs Jobs et/ou métadonnées simultanément, ainsi que les Jobs et leurs dépendances et/ou leur(s) Job(s) fils, dans les paramètres des projets. Pour plus d'informations, consultez *Gérer les versions, page 34*.

5.6 Documentation du Job

Talend Open Studio vous permet de générer une documentation fournissant des informations générales concernant vos projets, vos Jobs ou vos joblets. Vous pouvez automatiser la génération d'une telle documentation, et éditer n'importe quel document généré.

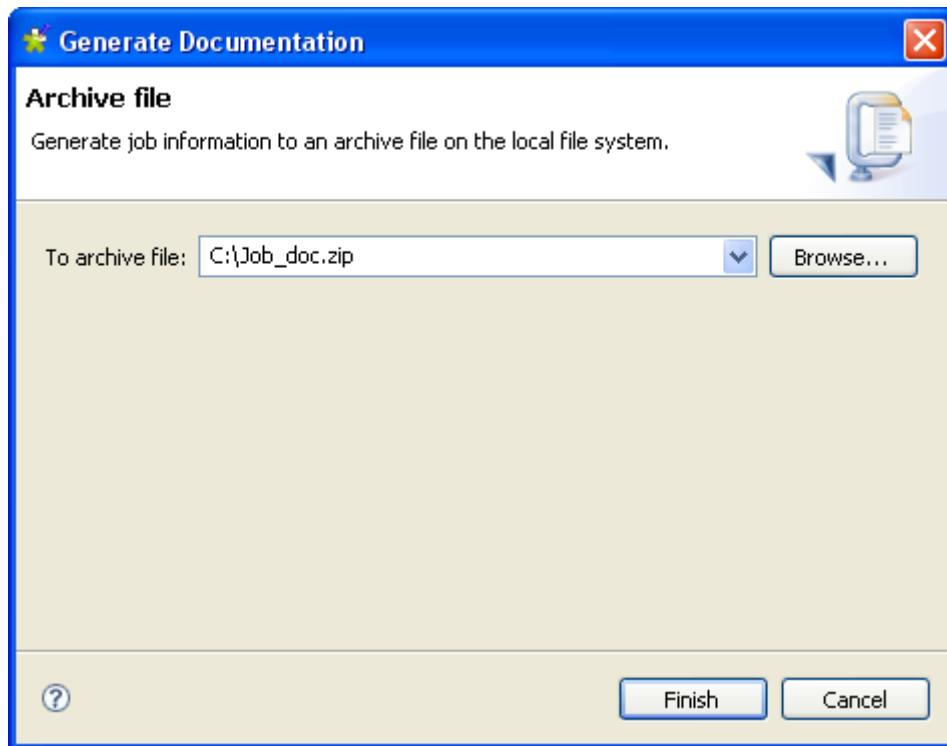
5.6.1 Générer la documentation HTML

Talend Open Studio vous permet de générer de la documentation détaillée au format HTML des Jobs sélectionnés dans le **Repository** de votre Studio. Cette auto-documentation comprend :

- les propriétés du projet dans lequel les Jobs ont été créés,
- les propriétés et les paramètres des Jobs sélectionnés ainsi qu'une image de chaque Job,
- la liste de tous les composants utilisés dans chaque Job sélectionné, ainsi que les paramètres de ces composants.

Pour générer un document HTML pour un Job, procédez comme suit :

- 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.

5.6.2 Mettre la documentation à jour

Vous pouvez mettre manuellement à jour votre documentation.

Pour mettre un seul document à jour : cliquez-droit sur la documentation que vous souhaitez mettre à jour et sélectionnez **Update documentation**.

5.7 Gestion de l'exécution des Jobs

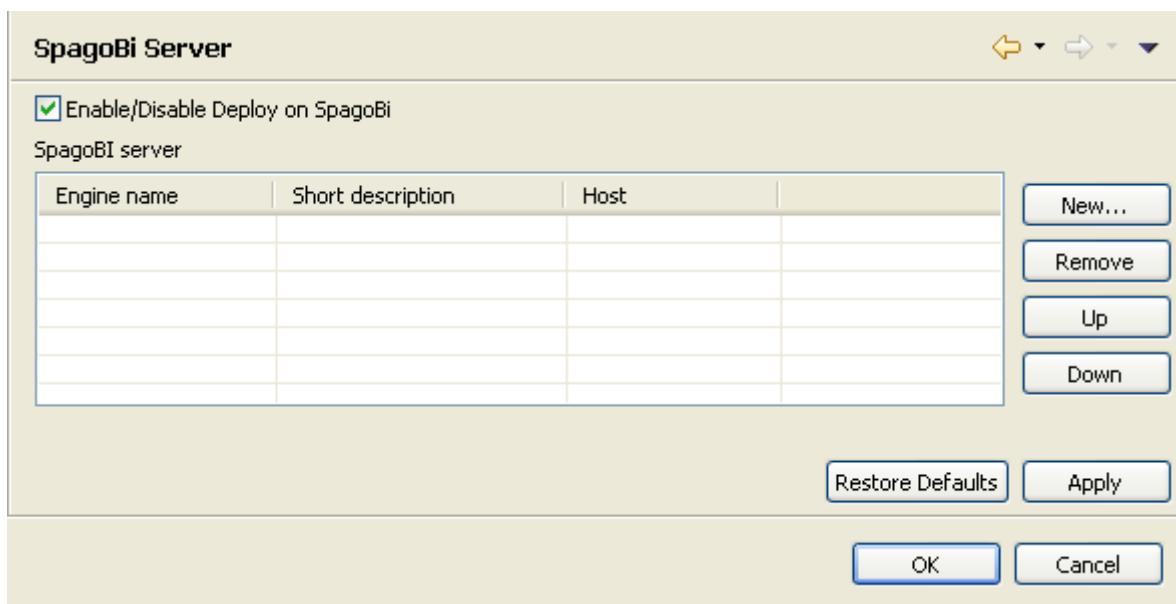
5.7.1 Déploiement d'un Job sur un serveur SpagoBI

A partir de l'interface **Talend Open Studio**, vous pouvez facilement déployer vos Jobs sur un serveur afin de les exécuter à partir de votre administrateur.

Créer une nouvelle connexion au serveur SpagoBI

Avant toute chose, renseignez les informations concernant votre serveur simple ou multiple dans **Talend Open Studio**.

- Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **[Preferences]**.
- Développez les noeuds **Talend > Import/Export** et sélectionnez **SpagoBI server** pour afficher la vue adéquate.



- Cochez la case **Enable/Disable Deploy on SpagoBI** pour activer le déploiement.
- Cliquez sur **New** pour ajouter un nouveau serveur à la liste server.



- 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.
- Cliquez sur **OK** pour valider ces informations.

Engine name	Short description	Host
mySpago	i Prod Server	92.168.10.3
mySpago2	i Dev server	92.168.10.4

La nouvelle entrée est ajoutée au tableau des serveurs disponibles. Vous pouvez ajouter autant d'entrée que vous le souhaitez.

- Cliquez sur **Apply** et **OK** pour fermer la boîte de dialogue **[Preferences]**.

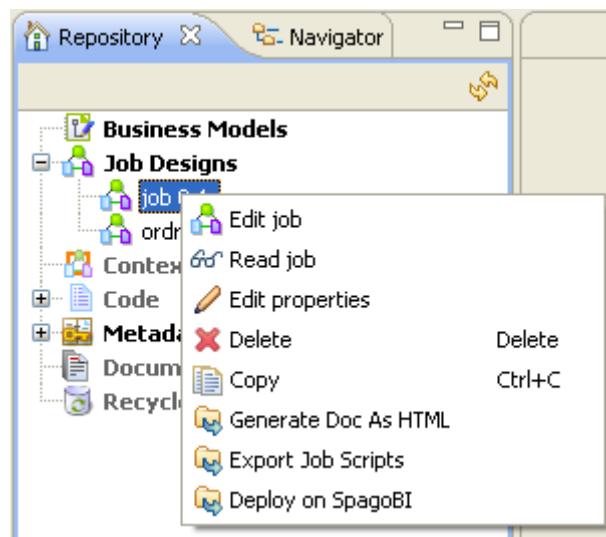
Editer ou supprimer un serveur

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 un serveur SpagoBI

Suivez les étapes suivantes pour déployer vos Jobs sur un serveur :



- A partir du **Repository**, développez le noeud **Job Designs** puis cliquez-droit sur le Job à déployer.
- Dans la liste déroulante, 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**.
- Dans le champ **SpagoBI server**, sélectionnez le serveur 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 pour exécuter vos jobs.



CHAPITRE 6

Mapping de flux de données

Le moyen le plus courant de gérer des flux multiples en entrée et en sortie, ainsi que les transformations et redirection des données est d'utiliser le composant **tMap**.

Ce chapitre fournit les informations concernant les principes d'usage de ce composant. Pour plus d'informations, ou pour des cas d'utilisation, consultez *tMap* dans le **Guide de Référence des Composants Talend Open Studio**.

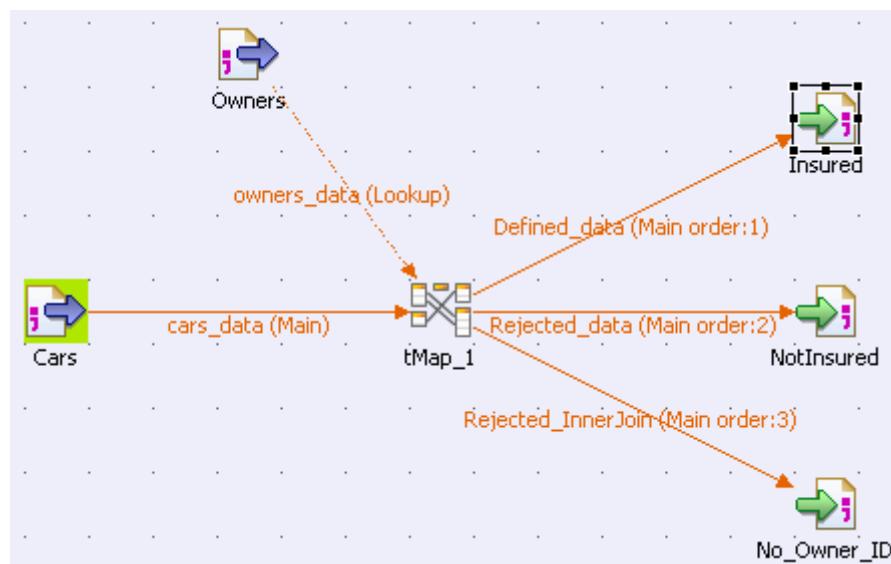
Avant de commencer un processus d'intégration de données, vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

6.1 Présentation du fonctionnement du tMap

Le **tMap** permet de réaliser les opérations suivantes :

- multiplexage et démultiplexage des données
 - transformation des données sur tout type de champs
 - concaténation et inversion de champs
 - filtrage de champs à l'aide de contraintes
 - gestion des rejets de données

Etant donné 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.



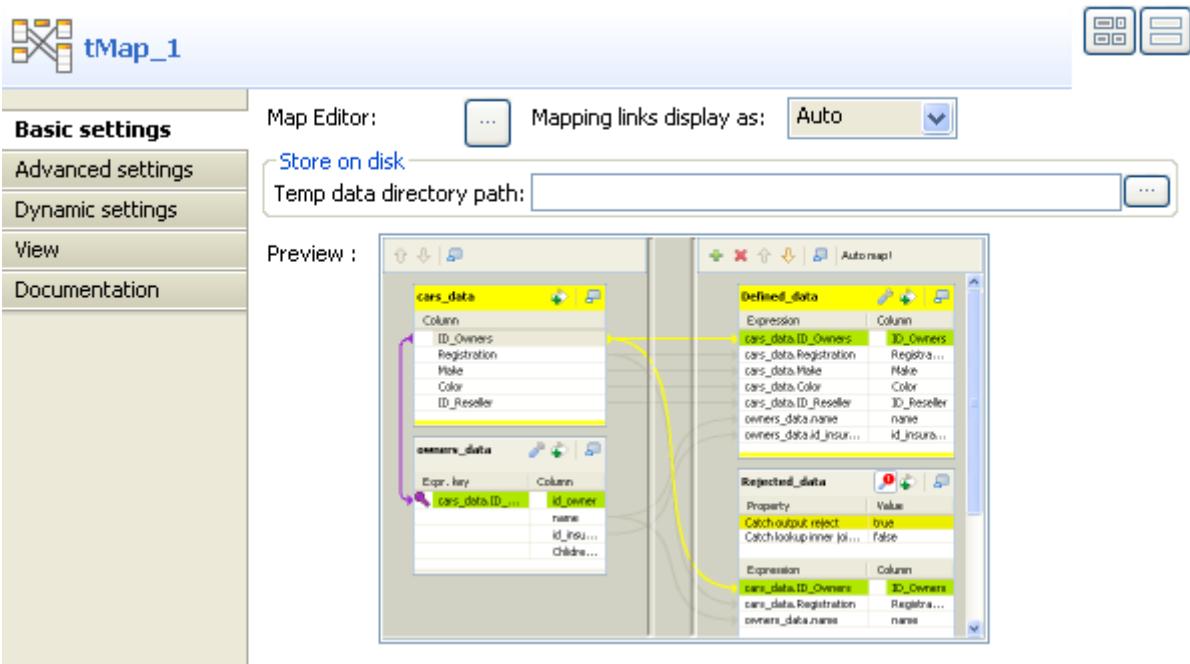
Le **tMap** utilise les connexions entrantes pour pré-remplir les schémas d'entrée dans le **[Map Editor]**. Par conséquent, vous ne pouvez pas créer directement dans le **[Map Editor]**, de nouveaux schémas d'entrée. Par contre, il vous faut mettre en place autant de connexions de type **Row** entrantes que nécessaire, pour que leur schéma s'affiche dans le **[Map Editor]**.

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

Il ne peut y avoir qu'une seule connexion de type **Main** row. Toute connexion entrante supplémentaire devient automatiquement une connexion **Lookup**. Voir : *Connexion de type Row, page 84*.

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 **tMap** requiert que les connexions soient mises en place pour définir les flux d'entrée et de sortie (respectivement Input et Output), vous devez également implémenter un mapping pour que la fonction d'aperçu de l'éditeur de mapping soit disponible dans le panneau **Component** de l'espace de modélisation.



Double-cliquez sur l’icône **tMap** dans l’éditeur graphique ou cliquez sur le bouton [...] de l’onglet **Component**, dans la vue **Basic settings** du composant **tMap**. L’éditeur de mapping s’ouvre dans une nouvelle fenêtre.

Les sections suivantes vous donnent les informations nécessaires à l’utilisation du composant **tMap** dans vos Job Designs.

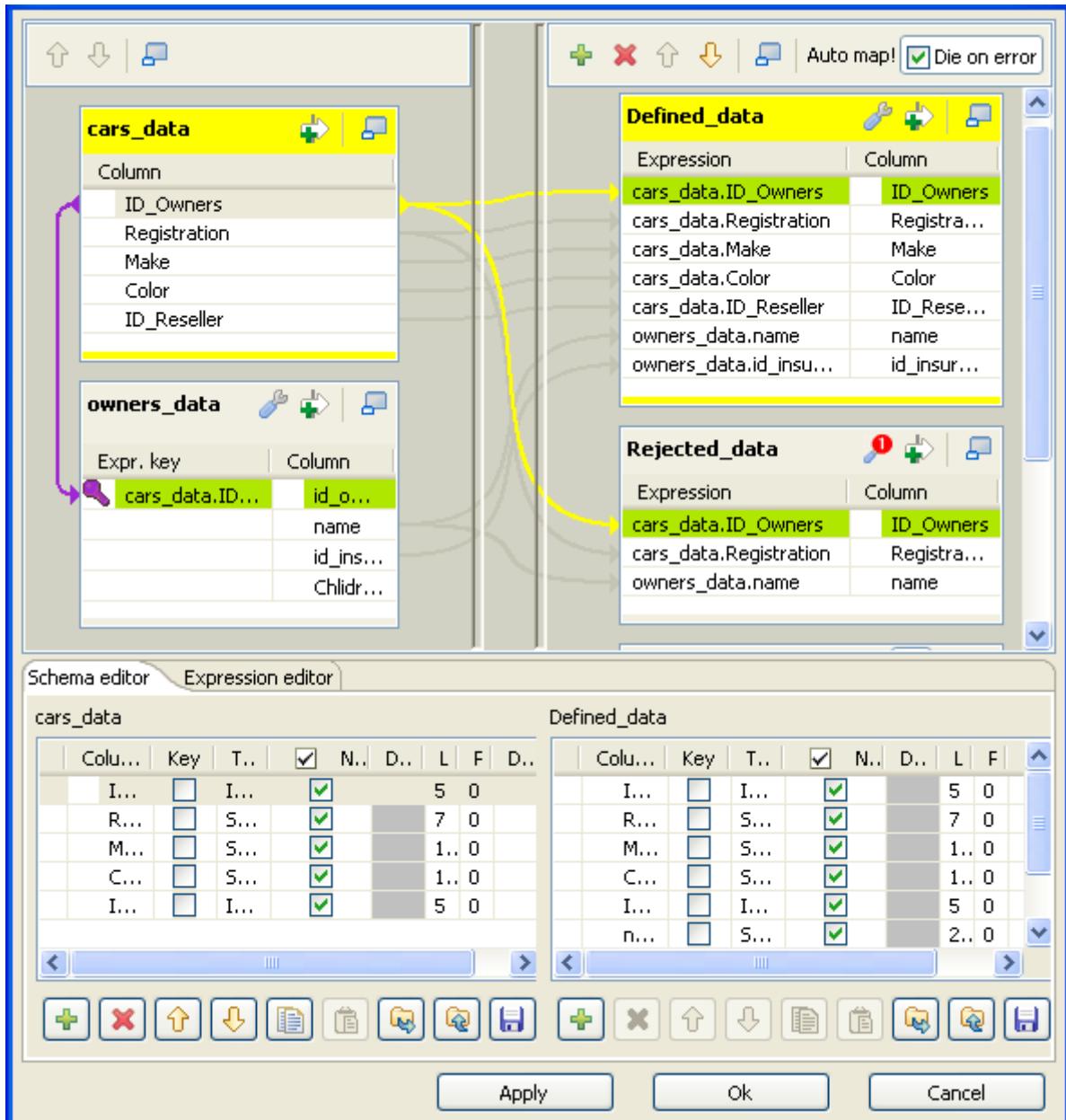
6.2 Interface du tMap

Le **tMap** est un composant dit avancé qui requiert plus d’informations de propriétés que les autres composants. En effet, le **[Map Editor]** est un outil complet vous permettant de définir tous les paramètres nécessaires au mapping, à la transformation et l’aiguillage des données dans votre processus, grâce à son interface graphique conviviale.

Vous pouvez réduire et restaurer les fenêtres de toutes les tables et la fenêtre **[Map Editor]** contenus dans l’éditeur de mapping, à l’aide des boutons dédiés situés dans le coin supérieur gauche de chacune des fenêtres.

Mapping de flux de données

Interface du tMap



Le **Map Editor** est composé de plusieurs zones :

- La zone **Input** sur le côté 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 **Variables** est au centre de l'éditeur. Ce tableau de variables 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.

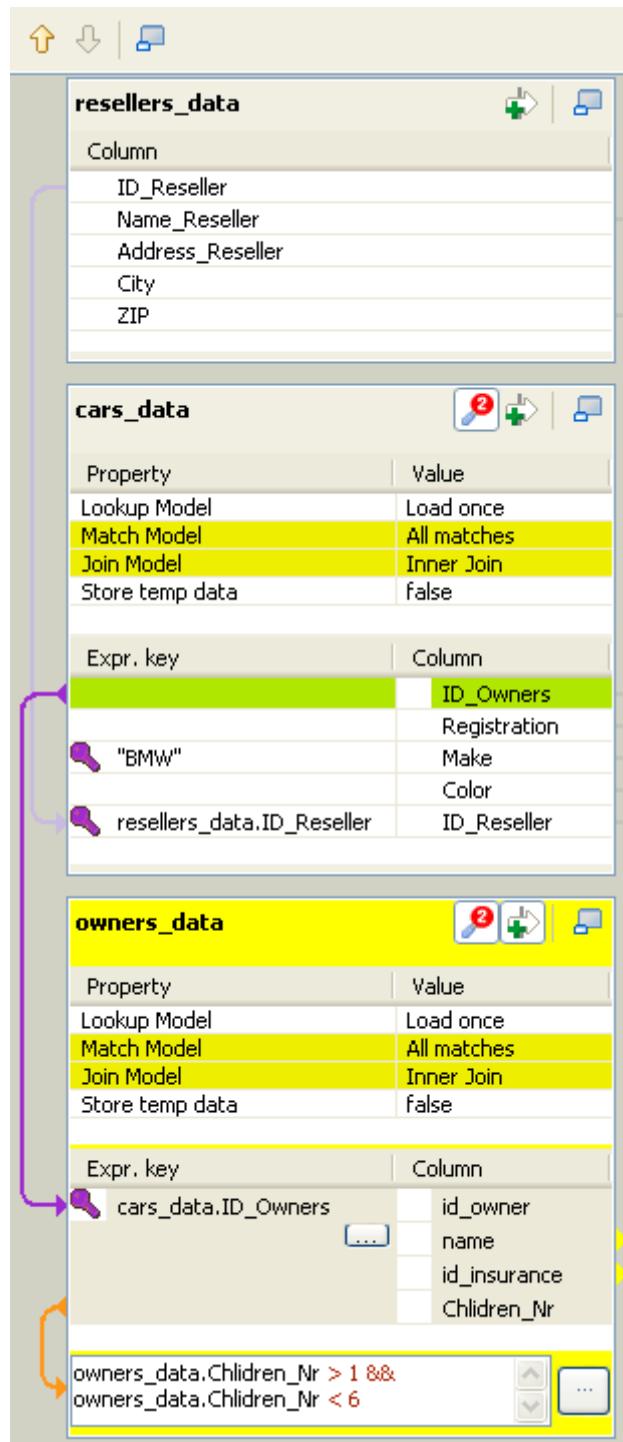
Le **[Map Editor]** est composé des panneaux suivants :

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).

6.3 Configuration du 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 : *Utiliser les jointures explicites, page 167.*

6.3.1 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 [Map Editor].

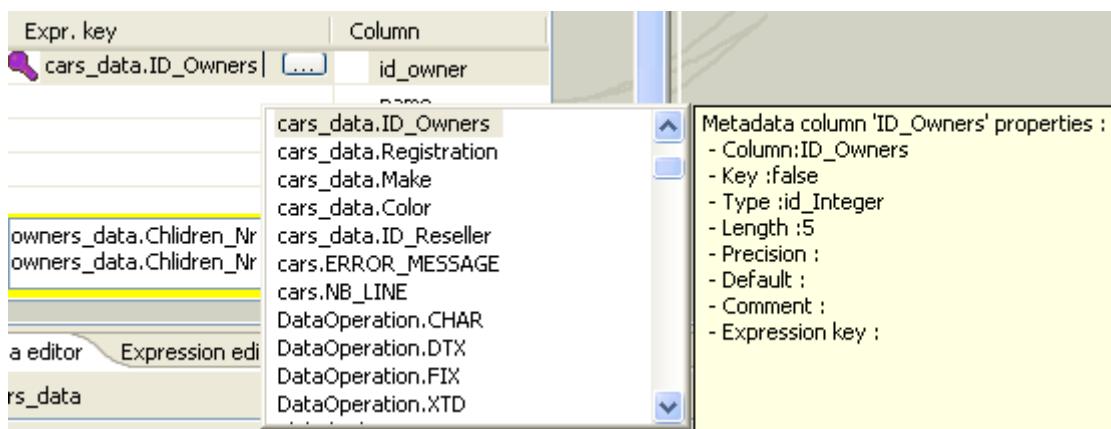
Le contenu des connexions secondaires (**Lookup**) est reproduit dans les autres tables dépendant de la table **Main**. Si vous n'avez pas encore défini de schéma pour un composant d'entrée, la table correspondante du [Map Editor] apparaîtra comme vide.

La clé (**Key**) est également reprise 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 [Map Editor]) 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 172*

6.3.2 Utiliser les jointures explicites

En effet, un lien **Joins** vous permet de sélectionner des données d'une table input via une autre table input. Dans le contexte d'un mapping, 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.

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.

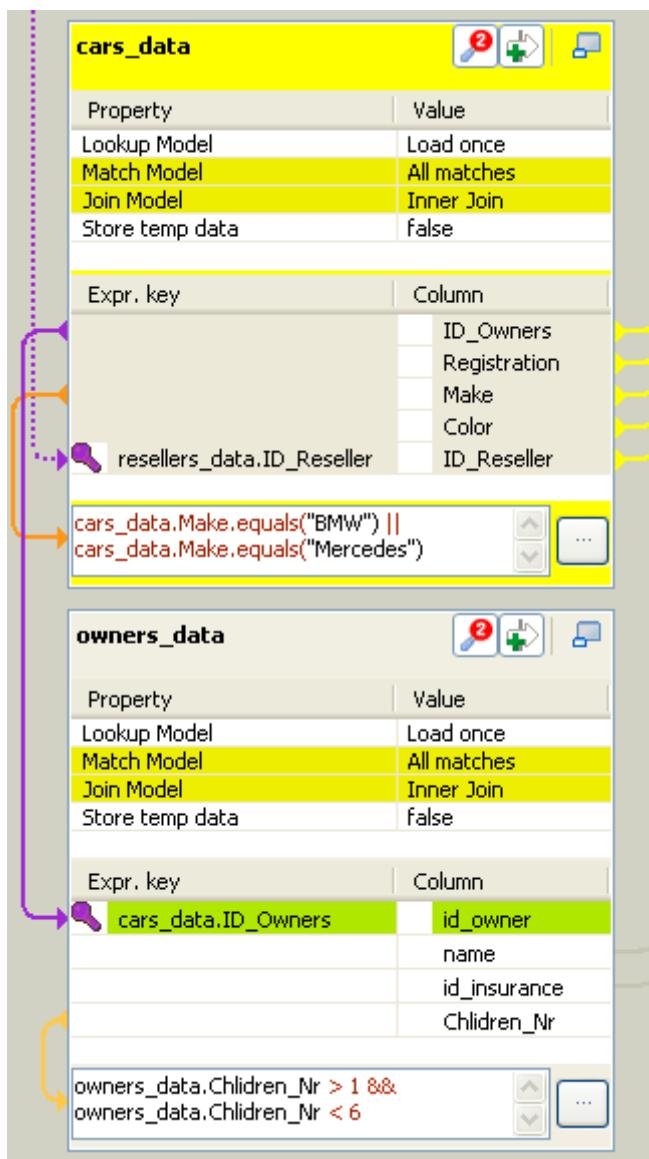
Vous pouvez créer des Joins directs entre la table principale et les tables secondaires. Mais vous pouvez également créer des Joins indirects à partir d'une table principale vers une table secondaire, elle-même reliée à une autre table secondaire. Pour cela, il faut que vous ayez créé une jointure entre ses tables.



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 la section *Paramètres Output, page 178.*



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 :

- *Onglet Schema editor, page 183*
- *Utiliser la fonction Inner Join, page 169*

Avec une jointure explicite, vous pouvez choisir de mettre un filtre pour limiter le nombre de correspondances. Vous pouvez soit prendre en compte une seule correspondance (la dernière), soit la première, soit toutes les correspondances.



Unique Match

Elle correspond à l'option par défaut lorsque vous effectuez une jointure explicite. Cela signifie que seulement la dernière correspondance du flux secondaire sera prise en compte et passera dans la table de sortie.

Les autres correspondances seront donc ignorées.

First Match

Cette option signifie que plusieurs correspondances peuvent être attendues dans le flux secondaire. L'option First signifie que seulement la première correspondance du flux secondaire sera prise en compte et passera dans le flux principal de sortie.

Les autres correspondances seront donc ignorées.

All Matches

Cette option signifie que plusieurs correspondances sont attendues dans le flux secondaire. Ainsi, toutes les correspondances sont prises en compte et passent dans le flux principal de sortie.

6.3.3 Utiliser la fonction 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**.

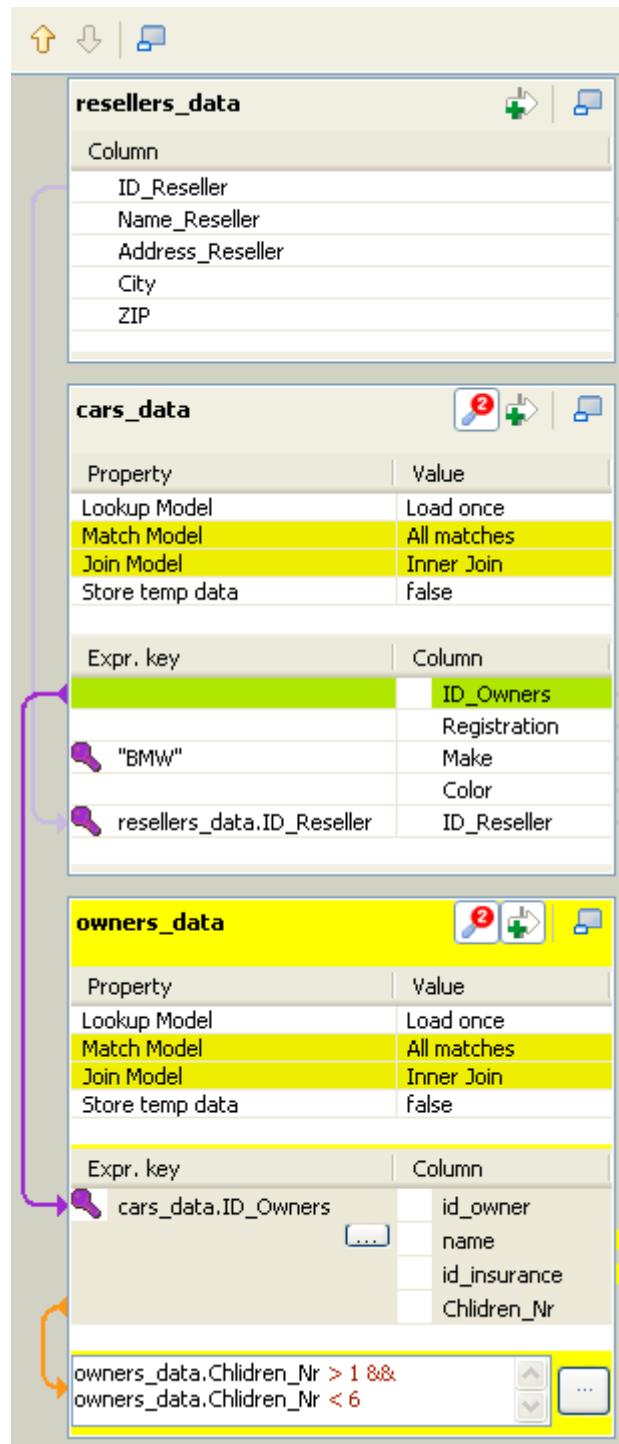
Si les données cherchées ne peuvent être récupérées avec la jointure explicite ou un filtre de jointure (Inner join), en d'autres termes la jointure Inner Join ne peut être établie quelle qu'en soit la raison, dans ce cas, les données demandées sont rejetées vers une table de sortie appelée **Inner Join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

Déposez simplement le nom des colonnes d'une table vers une table subordonnée, afin de créer une relation de jointure entre les deux tables. La jointure s'affiche graphiquement comme un lien violet, et crée automatiquement une clé qui sera utilisée comme une clé de hachage pour augmenter la vitesse de correspondance.

Cliquez sur le bouton **tMap settings** situé en haut de la table de référence et sélectionnez l'option **Inner Join** dans la liste **Join Model**, afin de définir cette table comme table **Inner Join**.

Mapping de flux de données

Configuration du flux d'entrée du tMap



Dans la zone **Output**, cliquez sur le bouton **tMap settings** et sélectionnez **true** dans la liste **Catch lookup inner join reject**, afin de définir la sortie **Inner Join Reject**.



Une table en **Inner Join** doit toujours être reliée à une table **Inner Join Reject**.

Rejected_InnerJoin	
Property	Value
Catch output reject	false
Catch lookup inner join reject	true
Expression	Column
cars_data.ID_Owners	ID_Owners

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

Voir également :

- *utiliser les Rejets Inner Join, page 181*
- *Filtrer un flux d'entrée, page 171*

6.3.4 Utiliser la fonction All Rows

Par défaut, dans chaque table d'entrée de la zone d'entrée de l'éditeur du **tMap**, le modèle de correspondance **All rows** est sélectionné. Cette option **All rows** signifie que toutes les lignes sont chargées à partir du flux **Lookup**, et comparées au flux **Main**.

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



Si vous créez une jointure explicite ou une jointure Inner Join entre deux tables, l'option **All rows** n'est plus disponible. Vous devez choisir entre les modèles Unique match, First match et All matches. Pour plus d'informations, consultez *Utiliser les jointures explicites, page 167* et *Utiliser la fonction Inner Join, page 169*.

6.3.5 Filtrer un flux d'entrée

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

owners_data	
Property	Value
Lookup Model	Load once
Match Model	All matches
Join Model	Inner Join
Store temp data	false
Expr. key	Column
cars_data.ID_Owners	<input type="button" value="..."/> id_owner <input type="button" value="..."/> name <input type="button" value="..."/> id_insurance <input type="button" value="..."/> Children_Nr
owners_data.Children_Nr > 1 && owners_data.Children_Nr < 6	

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 langage Java, c'est pourquoi les conditions doivent être rédigées en Java.

6.3.6 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 sélectionner les champs/colonnes à retirer de la table.



Vous pouvez enlever des entrées Input d'un schéma dans le **[Map Editor]**, cependant, cette suppression se répercutera sur la définition du schéma au niveau des propriétés du composant d'entrée.

6.4 Mapping de variables

La table **Var** regroupe toutes les variables de mapping qui peuvent être utilisées à différents endroits du **[Map Editor]**.

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

Les variables vous permettent de gagner du temps et vous évitent d'avoir à resaisir plusieurs fois les mêmes données.

Var	Type	Variable
StringHandling.UPCASE(cars_data.Make)	String	var1

Il existe plusieurs possibilités pour créer des variables :

- Saisissez librement vos variables en Java/Perl. Saisissez les chaînes de caractères entre guillemets simples ou concaténez plusieurs fonctions à l'aide de l'opérateur approprié.
- Ajoutez de nouvelles lignes à la table à l'aide du bouton **[+]** et retirez des lignes à l'aide du bouton **[x]**. Puis appuyez sur **Ctrl+Espace** pour récupérer les variables globales et de contexte déjà existantes.
- Déposez une ou plusieurs entrées Input dans la table **Var**.

Var	Type	Variable
StringHandling.UPCASE(cars_data.Make)	String	var1
Color		
ID_Reseller		

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** enfoncee 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.	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.	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.	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.

6.4.1 Accéder aux variables globales ou de contexte

Appuyez sur **Ctrl+Espace** pour accéder à la liste de semi-complétion des variables globales et de contexte.

Une fenêtre de métadonnées est annexée à la liste de semi-complétion des variables. Elle fournit des informations concernant la colonne sélectionnée.

6.4.2 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.

6.5 Utilisation de l'Expression editor

Toutes les déclarations d'expressions (**Input**, **Var** ou **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.

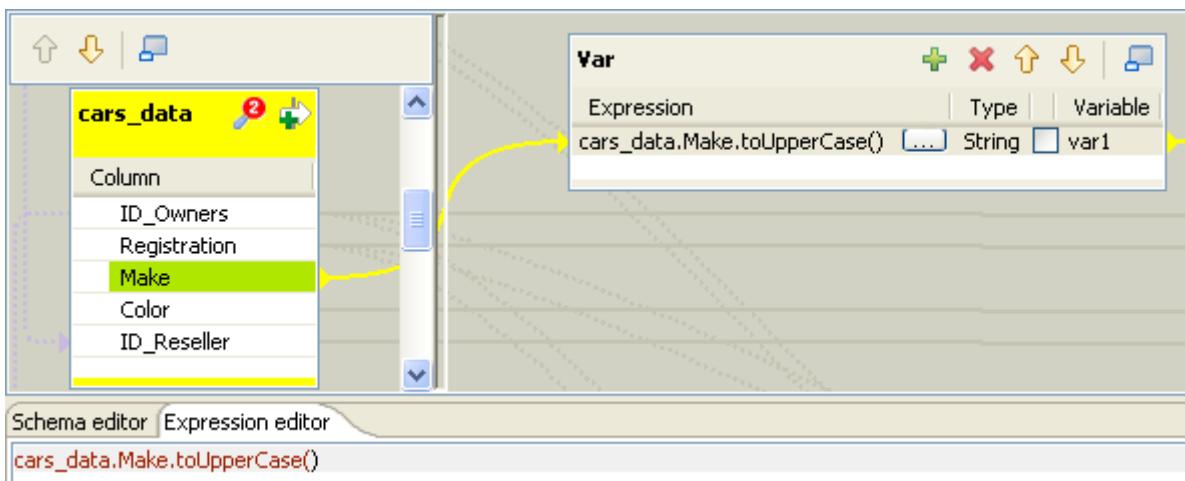
6.5.1 Accéder à l'Expression editor

Vous pouvez écrire les expressions dont vous avez besoin pour la transformation de données directement dans la vue **Expression editor** située dans la partie inférieure de l'**Expression editor**, ou vous pouvez ouvrir la boîte de dialogue [**Expression Builder**] dans laquelle vous pouvez écrire les expressions pour la transformation de données.

Pour ouvrir la vue **Expression editor**, procédez comme suit :

- Double-cliquez sur le composant **tMap** dans votre Job pour ouvrir le **Map Editor**.
- Dans la partie inférieure de l'éditeur, cliquez sur l'onglet **Expression editor** afin d'ouvrir la vue correspondante.

 Pour éditer une expression, sélectionnez-la dans le panneau **Input** puis cliquez sur l'onglet **Expression editor** et modifiez l'expression comme nécessaire.

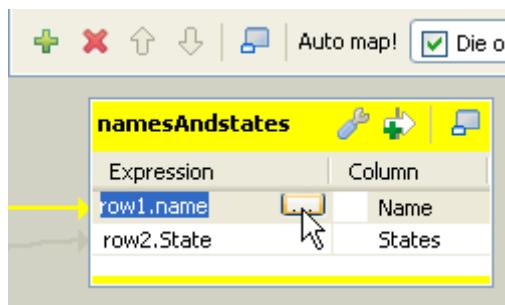


- Saisissez le code Perl ou Java selon vos besoins. L'expression correspondante dans le panneau output est synchronisée.

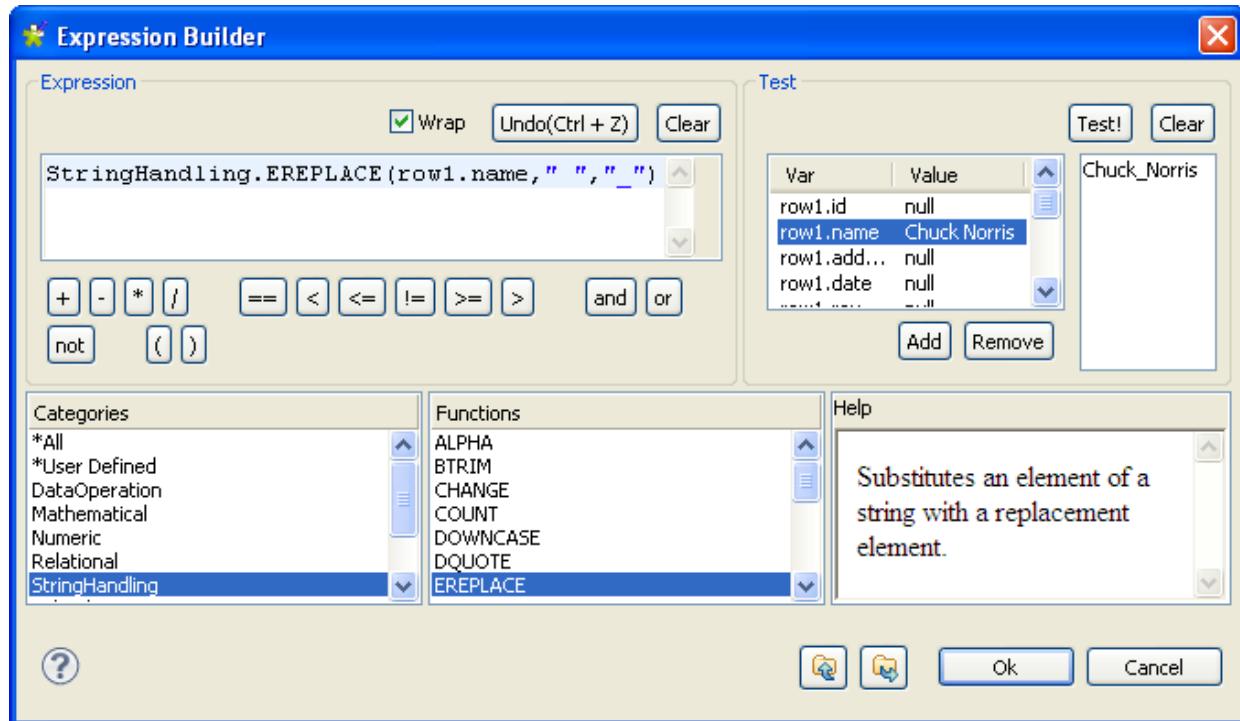
 Référez-vous à la documentation Perl ou Java adéquate pour plus d'informations concernant les fonctions et les opérations.

Pour ouvrir la boîte de dialogue [**Expression Builder**], procédez comme suit :

- Dans les panneaux **Var** ou **Output** dans le **Map Editor**, cliquez sur le bouton [...] à côté de l'expression que vous souhaitez ouvrir.



La boîte de dialogue [**Expression Builder**] s'ouvre sur l'expression sélectionnée.

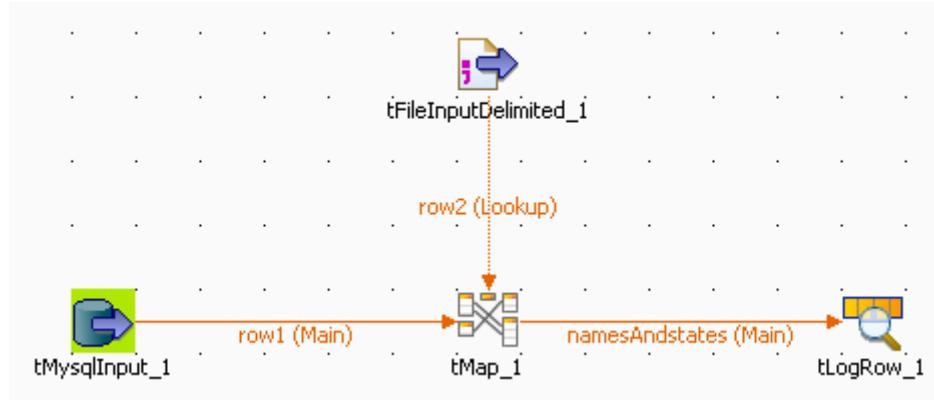


Pour un exemple d'utilisation de l'**Expression editor**, consultez la section suivante.

6.5.2 Créez le code à l'aide de l'Expression Builder

Pour certains Jobs, il est nécessaire de rédiger du code afin de paramétriser 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 en Perl).

L'exemple suivant vous montre l'utilisation de l'**Expression Builder** pour le composant **tMap**.

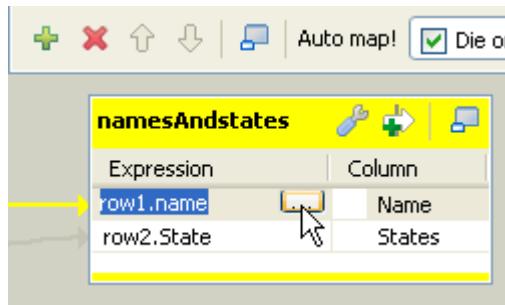


Deux flux d'entrée sont connectés au **tMap**.

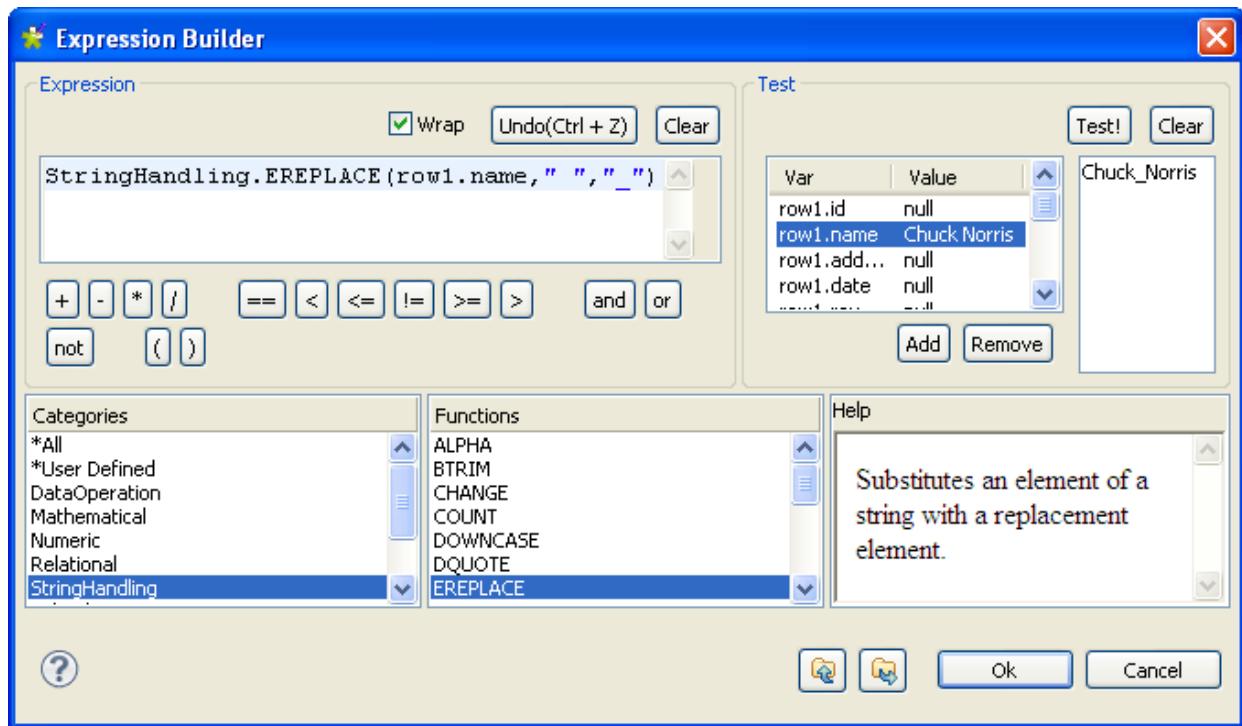
- Un premier flux DB input, contenant une liste de personnes dont le nom et le prénom sont séparés par un espace.
- Un autre flux File input, contenant la liste des états des Etats-Unis, en minuscule.

Dans le **[Map Editor]**, 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 [Map Editor], 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, page 161*.
- 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 [...].

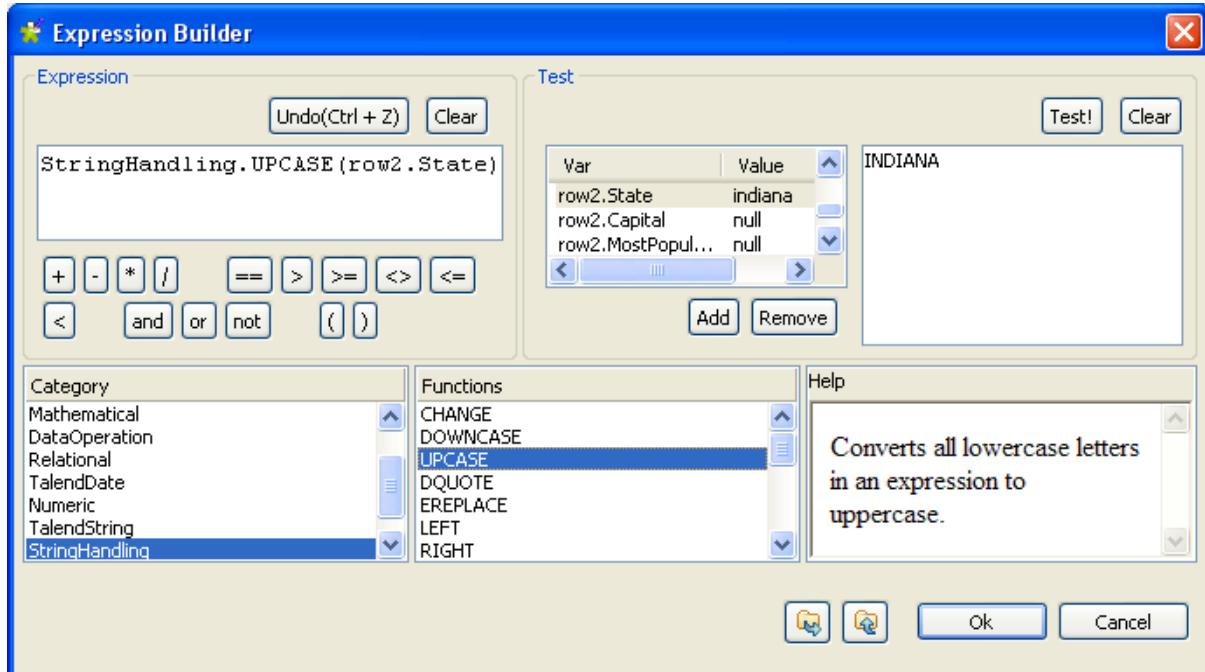


La fenêtre [Expression Builder] s'ouvre.



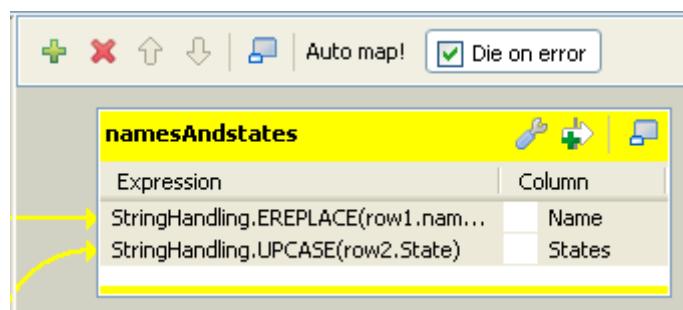
- Dans la liste **Category**, sélectionnez l'action que vous souhaitez effectuer. Pour cet exemple, choisissez **StringHandling**, puis sélectionnez la fonction **EREPLACE** dans la liste **Functions**.
- Dans la zone **Expression**, collez *row1.Name* à la place du texte, pour obtenir : `StringHandling.EREPLACE(row1.Name, " ", "_")`. 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. La sortie de cet exemple s'affiche comme suit :

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

6.6 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 **[Map Editor]**.

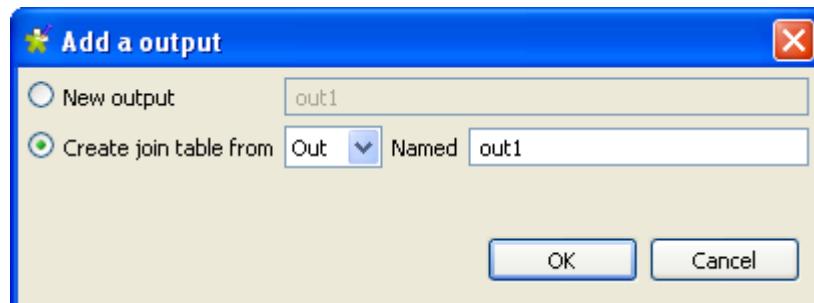
Vous pouvez également ajouter un schéma Output dans votre **[Map Editor]**, à l'aide du signe **[+]** de la barre d'outils de la zone **Output**.

Vous pouvez aussi effectuer une jointure entre vos tables de sortie. La jointure sur les tables de sortie permet de traiter les flux séparément, mais de les unifier en sortie. Pour plus d'informations concernant cette fonction, consultez le *Scénario 7 : Mapping avec des tables de sortie, page 1438* dans le **Guide de Référence Talend Open Studio**.



La table de jointure récupère le schéma de la table source.

Lorsque vous cliquez sur le bouton **[+]**, afin d'ajouter un schéma Output ou d'effectuer une jointure entre vos tables de sortie, une boîte de dialogue s'ouvre, vous proposant deux possibilités.



Sélectionner...	Pour...
New output	Ajouter une table de sortie.
Create join table from	Effectuer une jointure entre vos tables de sortie. Dans la liste déroulante, sélectionnez la table à partir de laquelle effectuer la jointure, puis dans le champ Named , saisissez le nom de la table à créer.

A la différence de la zone **Input**, l'ordre des tables de schémas Output n'a pas une grande importance, car il n'existe aucune relation de subordination (Join) entre les sorties.

Une fois que vous avez créé toutes les connexions de sortie, et de ce fait, tous les schémas Output, vous pouvez sélectionner et organiser les données de ces sorties.

Déposez une ou plusieurs entrées à partir de la zone **Input** directement vers la table Ouput appropriée.

Appuyez sur **Ctrl** ou **Maj** pour pour une sélection multiple.

Ou vous pouvez utiliser des expressions de la zone **Var** par glisser-déposer dans la table **Ouput** 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(s) entrée(s) 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.

6.6.1 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 *Créer du code à l'aide de l'Expression Builder, page 175*

6.6.2 Utiliser les 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.

Mapping de flux de données

Paramètres Output



Vous pouvez saisir librement vos conditions de filtre à l'aide d'opérateurs et de fonctions Perl/Java.

Déposez des expressions d'une zone **Input** ou d'une zone **Var** vers l'entrée de Filtre de la table Output appropriée.



Un lien graphique de couleur orange apparaît. Ajoutez l'opérateur 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.

6.6.3 Utiliser la fonction Rejet

L'option **Reject** définit la nature d'une table Ouput particulière.

Cette table de rejet rassemble toutes les données qui ne satisfont pas un ou plusieurs des filtres définis dans les tables Ouput 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, cliquez sur le bouton **tMap settings** et sélectionnez **true** dans la liste **Catch Output Reject** afin de la définir comme la partie ELSE des tables régulières.

Rejected_data	
Property	Value
Catch output reject	true
Catch lookup inner join reject	false

Vous pouvez définir plusieurs tables Reject afin d'affiner les sorties multiples. Pour différencier les variantes de rejets, ajoutez des tables Reject, des lignes de filtre en cliquant sur la flèche au bouton [+].

Une fois qu'une table est définie comme table de rejet, le processus de vérification des données commencera par les tables régulières avant de prendre en considération les filtres possibles des tables de rejet.

Les données ne sont pas exclusivement traitées vers une seule sortie. Même si une donnée satisfait le filtre de sortie et qu'elle est donc routée vers elle, elle est également vérifiée contre les autres filtres et peut être également routée vers d'autres sorties.

6.6.4 utiliser les Rejets Inner Join

L'Inner Join est un Join Lookup. La table de rejet Inner Join est un type particulier de table de rejet de sortie. Elle regroupe les données rejetées de la table du flux principal lorsqu'un Inner Join n'a pu être établi.

Pour configurer un flux de sortie (Output) afin qu'il transmette les données de rejet d'un Inner Join, déposez un nouveau composant Output dans votre Job Design et connectez-le au **tMap**. Dans le **[Map Editor]**, cliquez sur le bouton **tMap Settings** et sélectionnez **true** dans la liste **Catch lookup inner join reject** pour définir cette table comme table de rejet Inner Join Reject.

Rejected_InnerJoin	
Property	Value
Catch output reject	false
Catch lookup inner join reject	true

6.6.5 Retirer des entrées de la table Output

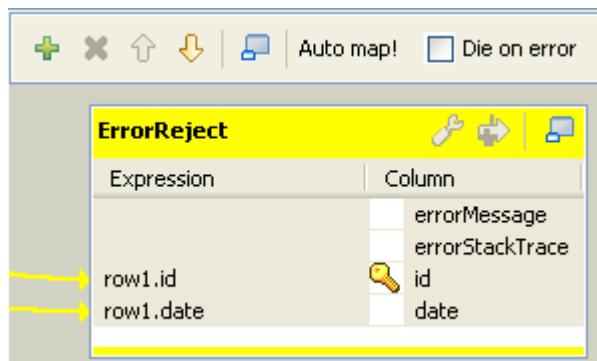
Pour retirer des entrées d'une table Output, cliquez sur la croix rouge, en bas, dans le Schema Editor de la table sélectionnée.

6.6.6 Gestion des erreurs

L'option **Die on error** empêche le traitement des erreurs. Pour ce faire, elle arrête l'exécution du Job aussitôt qu'une erreur est rencontrée. Le composant **tMap** fournit cette option afin d'empêcher le traitement des données erronées. L'option **Die on error** est activée par défaut dans le **tMap**.

Désactiver l'option **Die on error** vous permettra d'ignorer les lignes en erreur et de terminer le processus pour les lignes sans erreur. Cela vous permettra également de retrouver les lignes en erreur et de les gérer, si nécessaire.

Pour désactiver l'option **Die on error**, double-cliquez sur le composant **tMap** dans l'espace de modélisation graphique, afin d'ouvrir son **Editor**, et décochez la case **Die on error**, en haut à droite de l'éditeur.



ErrorReject	
Expression	Column
row1.id	errorMessage
row1.date	errorStackTrace
	id
	date

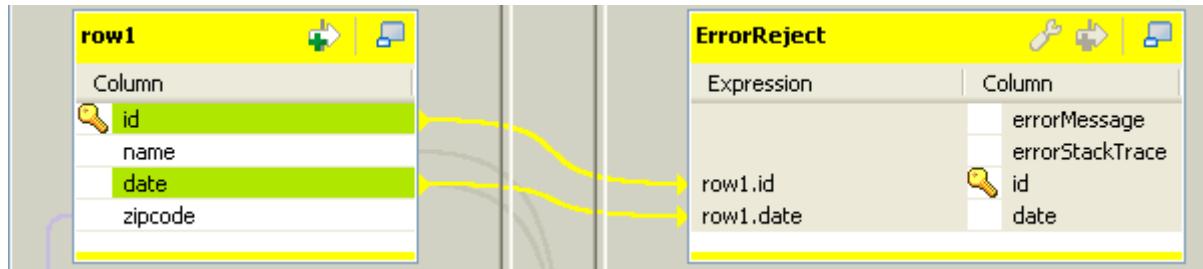
Une nouvelle table appelée **ErrorReject** s'affiche dans l'éditeur du **tMap**, dans la zone de sortie. Cette table de sortie comprend automatiquement deux colonnes : **errorMessage** et **errorStackTrace**, qui retrouvent le message et la trace de la pile de l'erreur rencontrée durant

Mapping de flux de données

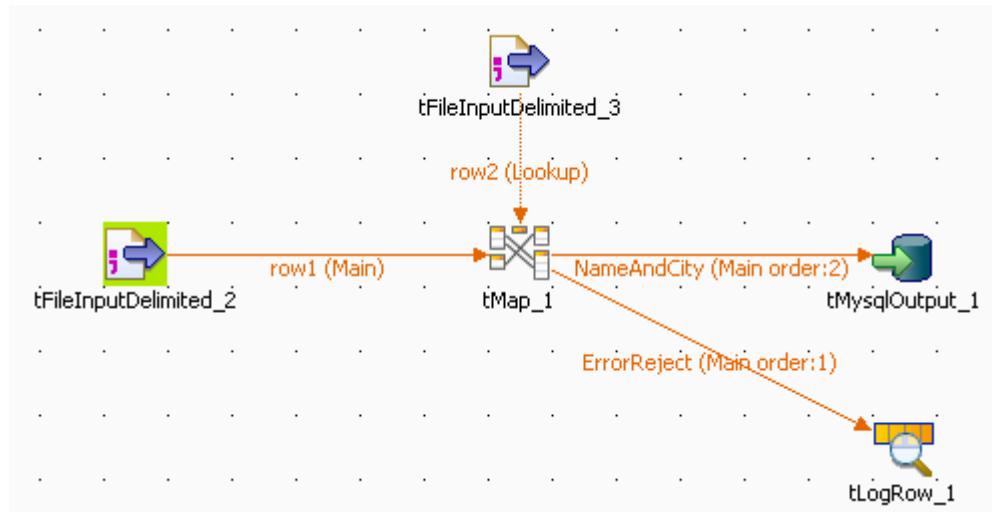
Paramètres Output

l'exécution du Job. Les erreurs peuvent être des erreurs de format de dates, des null pointer exceptions, des problèmes de conversion, etc.

Vous pouvez également glisser-déposer des colonnes (ici *id* et *date*) des tables d'entrée vers cette table de sortie de rejets. Ces données en erreur peuvent être retrouvées avec le message d'erreur correspondant et être corrigées ensuite.



Une fois la table *ErrorReject* définie, les flux correspondants peuvent être envoyés vers un composant de sortie.



Pour ce faire, dans l'espace de modélisation graphique, cliquez-droit sur le composant **tMap**, sélectionnez **Row > ErrorReject** dans le menu, et cliquez sur le composant de sortie correspondant, ici, le **tLogRow**.

Lorsque vous exécutez le Job, les erreurs sont retournées via la flux **ErrorReject**.

```

Starting job Die_on_error at 17:30 01/09/2010.

java.text.ParseException: Unparseable date: "08 01
1980" | java.lang.RuntimeException:
java.text.ParseException: Unparseable date: "08 01 1980"
    at routines.TalendDate.parseDate(TalendDate.java:503)
    at
doc.die_on_error_0_1.Die_on_error.tFileInputDelimited_2Pro
cess(Die_on_error.java:1409)
    at
doc.die_on_error_0_1.Die_on_error.runJobInTOS(Die_on_error.
java:2262)
    at
doc.die_on_error_0_1.Die_on_error.main(Die_on_error.java:2
160)
Caused by: java.text.ParseException: Unparseable date: "08
01 1980"
    at java.text.DateFormat.parse(Unknown Source)
    at routines.TalendDate.parseDate(TalendDate.java:501)
    ... 3 more
|1|08 01 1980
Job Die_on_error ended at 17:30 01/09/2010. [exit code=0]

```

Le résultats contient le message d'erreur, la trace de sa pile, et les deux colonnes *id* et *date*, glissées et déposées dans la table **ErrorReject**, séparées par une barre verticale “|”.

6.7 Onglet Schema editor

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

Column	Key	Type	Length	Precision	Nullable	Comment
ID_Owner	<input checked="" type="checkbox"/>	int	2	0	<input checked="" type="checkbox"/>	
First Name	<input type="checkbox"/>				<input type="checkbox"/>	
Name	<input type="checkbox"/>	String	8	0	<input checked="" type="checkbox"/>	
ID_Insurance	<input type="checkbox"/>	String	7	0	<input checked="" type="checkbox"/>	

Below the table is a toolbar with icons for adding (+), deleting (-), moving up (^), moving down (^), saving (disk), and other operations.

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

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

Métadonnées	Description
Column	Nom de colonne tel que défini dans le schéma du [Map Editor] 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.

Métadonnées	Description
Type	Type de données : Chaîne de caractères (String) ou entier (Integer).  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.



Les schémas Input et Output sont indépendants les uns des autres. Vous pouvez par exemple changer le libellé d'une colonne Output sans que le libellé du schéma Input ne change.

Cependant, toute modification apportée au schéma est immédiatement reportée sur le schéma correspondant de la zone **Input** ou **Output** appropriée, mais également au niveau des propriétés elles-mêmes du composant Input ou Output concerné.

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

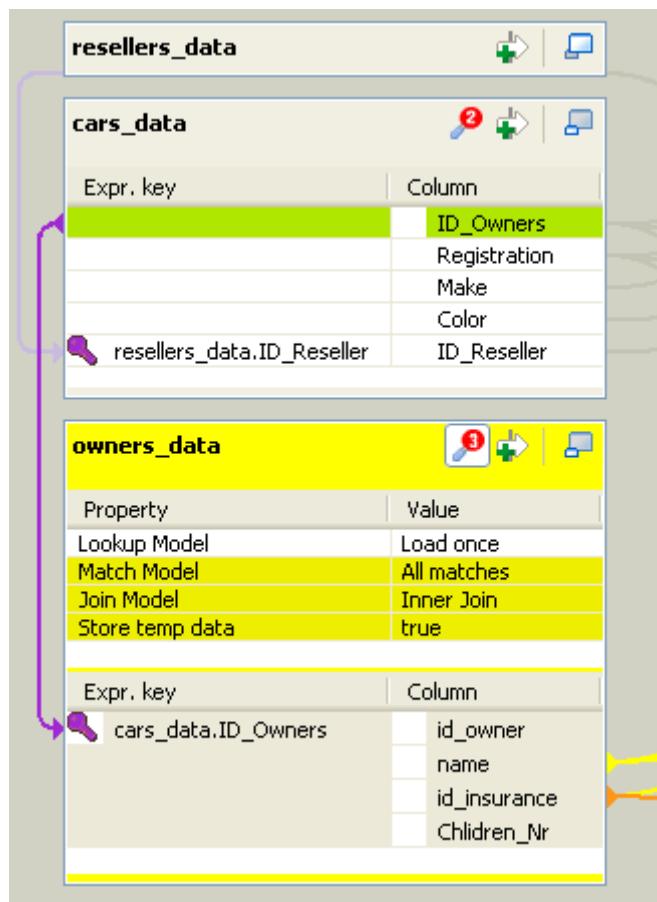
Passez la souris sur le champ invalide pour faire apparaître le message d'erreur.

6.8 Résolution des problèmes de mémoire lors de l'utilisation du tMap

Lorsque vous devez traiter un nombre important de données, par exemple, de nombreuses colonnes, différents types de colonnes ou lignes, votre système peut rencontrer des problèmes de mémoire insuffisante empêchant votre Job de s'exécuter correctement, et plus particulièrement lorsque vous utilisez un composant **tMap** pour effectuer des transformations.

Une option (uniquement disponible en Java pour le moment) a été ajoutée au composant **tMap**, pour utiliser moins de mémoire lors du traitement des données de référence (lookup). En effet, au lieu de stocker les données de référence dans la mémoire système et ainsi en atteindre les limites, l'option **Store temp data** vous permet de stocker les données de référence dans un dossier temporaire sur votre disque dur.

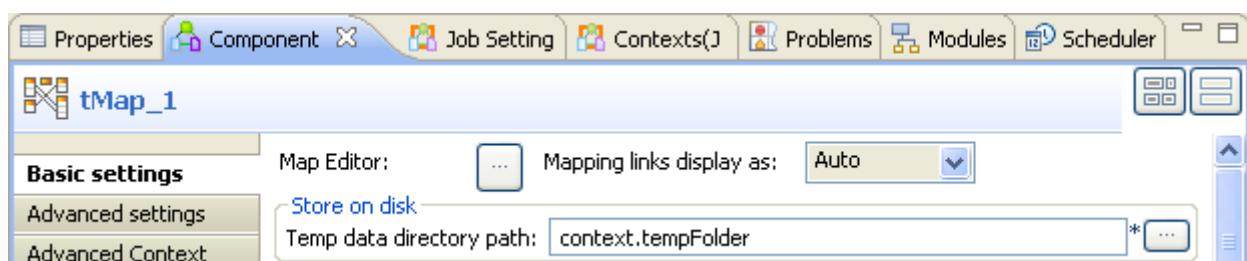
Cette option peut être sélectionnée sur la table Lookup de la zone **Input** (à gauche) de l'éditeur du **tMap**, le **[Map Editor]**.



Pour activer cette option, suivez les étapes suivantes :

- Double-cliquez sur le composant **tMap** de votre Job pour lancer le **[Map Editor]**.
- Dans le panneau de gauche, correspondant aux sources **Input** et **Lookup**, cliquez sur la table **Lookup** affichant les données de Lookup que vous souhaitez charger sur le disque pour ne pas encombrer la mémoire.
- Cliquez sur le bouton **Store temp data** pour activer cette option.
- Cliquez sur **OK** pour valider les modifications effectuées au **[Map Editor]** et retourner à l'espace de modélisation.

Pour que cette option soit totalement active, vous devez définir le répertoire de votre disque dans lequel les données seront stockées.



- Cliquez sur l'onglet **Component** pour afficher les propriétés du **tMap** dans l'onglet **Basic settings**.

- Dans la zone **Store on disk**, renseignez le chemin d'accès au dossier temporaire dans lequel les données de lookup devront être stockées.
- Vous pouvez utiliser une variable de contexte à l'aide du raccourci **Ctrl+Espace** si vous avez configuré une variable dans un groupe de contexte du **Repository**. Pour plus d'informations concernant les contextes, consultez *Centraliser les Contextes et les Variables, page 91*.

A la fin de votre sous-job, les fichiers temporaires seront nettoyés.

Ainsi, vous limitez l'utilisation de mémoire dédiée pour le traitement des données de référence (lookup) sur votre disque.

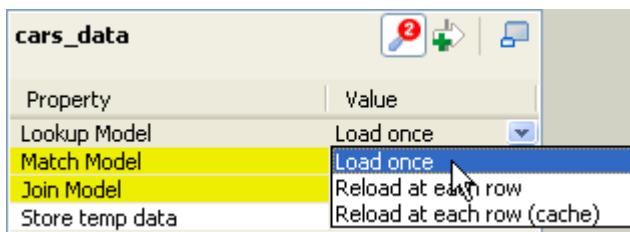


Etant donné que l'écriture du flux principal sur le disque nécessite le stockage de données, l'ordre des lignes en sortie peut ne pas être respecté.

Dans la vue **Advanced settings**, vous pouvez aussi ajouter un buffer, au besoin. Pour cela, renseignez le champ **Max. buffer size (in Nr of lines)** afin de séparer les données stockées sur le disque en autant de fichiers que nécessaire.

6.9 Gestion des références

Le composant **tMap** offre différents types de chargements possibles des données de référence pour vous adapter aux différents types de traitements, ainsi que pour gérer les problèmes de performance lorsque vous devez traiter de gros volumes de données dans vos bases de données de référence.



Dans la zone **lookup** de l'éditeur du **tMap**, cliquez sur la petite flèche vers le bas pour afficher la liste des différents types de chargements disponibles :

- **Load once** : Paramètre par défaut. Sélectionnez cette option pour charger la totalité du flux de référence avant de traiter le flux principal. Sélectionnez cette option si un nombre important de données de votre flux principal est sollicité par votre flux de référence (lookup), ou si vos données de référence proviennent d'un fichier qui peut être chargé facilement.
- **Reload at each row** : Le flux de référence est chargé à chaque ligne. Sélectionnez cette option lorsque vous utilisez une base de données de référence de gros volume, et que le flux principale est très petit. Notez que cette option vous permet d'utiliser des paramètres dynamiques de variable (de votre composant base de données), notamment la commande WHERE, pour modifier à la volée le flux de référence au moment où il est chargé, et avant que le flux principal ne soit traité. Cette option peut être considérée comme l'équivalent de l'option **Store temp data**, permettant de stocker les données de référence dans un dossier temporaire sur votre disque dur, disponible pour les fichiers de référence.
- **Reload at each row (cache)** : Les expressions (de la table de référence) sont d'abord évaluées et interrogées dans le cache. Le résultat des jointures ayant déjà été créées, est stocké dans le cache afin d'éviter de charger les mêmes résultats une deuxième fois. Cette option permet d'optimiser le temps de traitement et d'améliorer les performances de traitement du composant **tMap**.



Notez que, pour le moment, vous ne pouvez pas utiliser les options **Reload at each row (cache)** et **Store temp data** en même temps.

Pour un exemple d'utilisation de ces options, consultez la section **tMap** du **Guide de Référence des Composants Talend Open Studio**.



Lorsque vous utilisez une table en flux de référence, nous vous conseillons d'ouvrir une connexion à la base de données au début de votre Job, afin d'optimiser les performances.

Pour un exemple de scénario utilisant cette option, consultez le *Scénario 6 : Mapping avancé avec rechargement de la base de référence à chaque ligne* dans le **Guide de Référence des Composants Talend Open Studio**.



CHAPITRE 7

Gestion des métadonnées

Dans **Talend Open Studio**, les métadonnées permettent de définir, de décrire ou de documenter les données traitées par le logiciel.

Ce chapitre vous décrit les procédures de création et de gestion des différentes métadonnées pouvant être utilisées dans tous vos jobs.

Avant de commencer un processus de gestion de données, vous devez vous être familiarisé avec l’interface graphique de **Talend Open Studio**. Pour plus d’informations, consultez *Interface de Talend Open Studio, page 289*.

7.1 Objectifs

Dans le dossier **Metadata du Repository**, vous pouvez stocker toutes vos informations de connexion aux fichiers, bases de données et/ou systèmes, et les réutiliser pour la conception de vos jobs.

Les différents assistants de création vous aident à enregistrer ces informations qui vous permettent de paramétrier les composants d'entrée ou de sortie de vos jobs. Ils vous permettent également de stocker une description de vos données appelée “schéma” dans **Talend Open Studio**.

Les procédures de l'assistant diffèrent en fonction du type de connexion choisi.

Cliquez sur **Metadata** dans le **Repository** pour développer l'arborescence. Chaque dossier **Connection** contient les différentes connexions que vous avez créées.

Grâce à **Talend Open Studio**, il est possible de créer des connexions à des bases de données, de créer des connexions JDBC ou des schémas de type Fichier (file), des annuaires LDAP, des WebServices WSDL, des applications métier Salesforce, etc.



Si vous avez sélectionné le mode Built-in et que vous avez paramétré manuellement les propriétés de votre composant, vous avez encore la possibilité de sauvegarder ces propriétés sous forme de métadonnée dans le **Repository**. Pour plus d'informations sur les différentes manières de paramétrier les propriétés d'un composant, consultez *Onglet Basic Settings, page 65*.

Les sections suivantes décrivent en détail la procédure à suivre pour créer les différents schémas et connexions.

7.2 Configurer une connexion DB

Si vous vous connectez régulièrement à une base de données de tout type, vous pouvez centraliser les informations de connexion à cette base de données sous le noeud **Metadata** de la vue **Repository**.

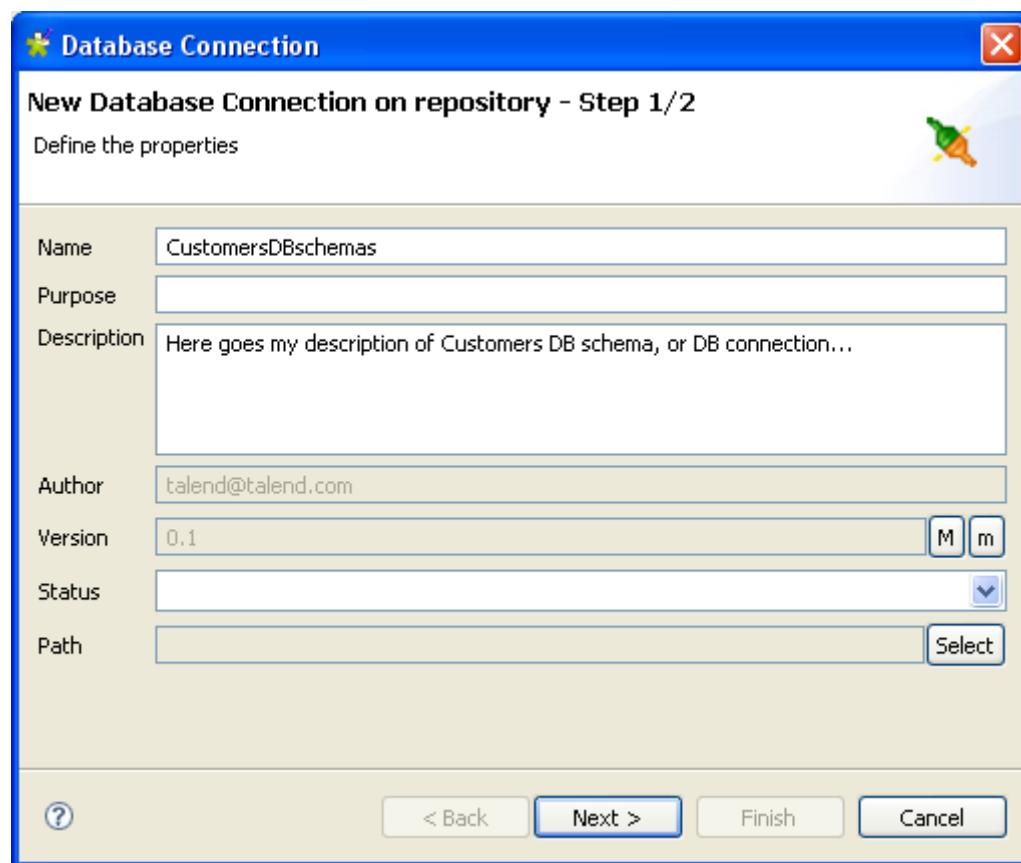


Vous pouvez également configurer cette connexion DB en cliquant sur l'icône dans la vue **Basic settings** de tous les composants DB input et output.

7.2.1 Etape 1 : Propriétés générales

Pour créer une connexion à une base de données, suivez la procédure décrite ci-dessous. Développez le noeud **Metadata** dans la vue **Repository**, cliquez-droit sur **Db Connections** puis sélectionnez **Create connection** dans le menu contextuel.

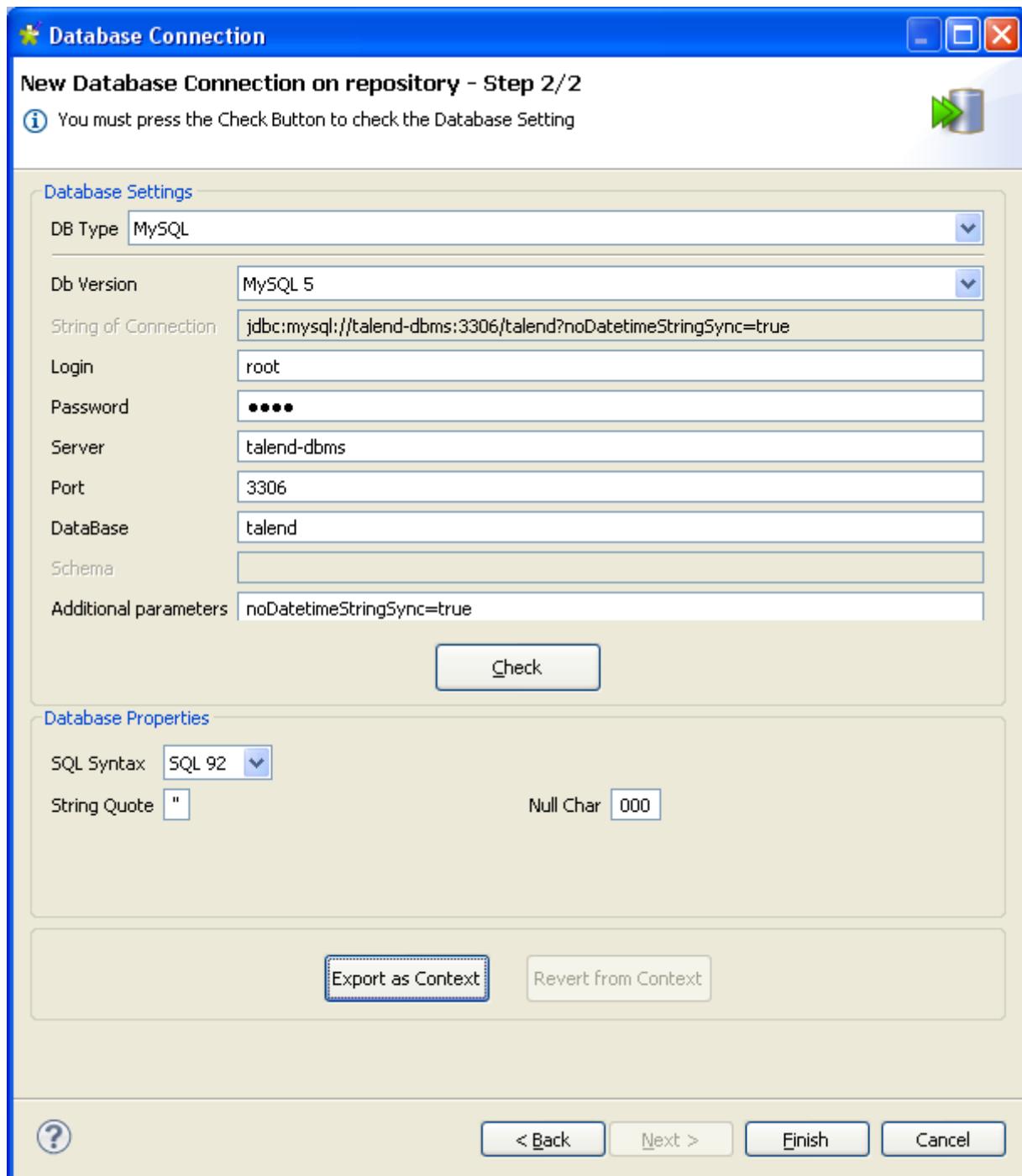
Un assistant de connexion s'ouvre. Renseignez les propriétés générales du schéma, notamment les champs **Name** et **Description**. Le champ **Status** est un champ personnalisé que vous pouvez définir dans les Préférences (**Window > Preferences**).



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

7.2.2 Etape 2 : Connexion

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



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

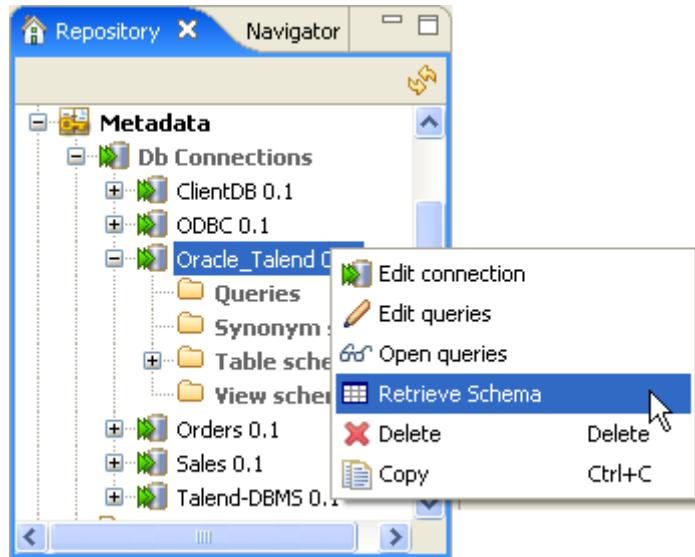


Pour pouvoir retrouver les schémas de toutes les tables dans la base de données :

- saisissez *dbo* dans le champ **Schema**, si vous vous connectez à MSSQL 2000,
- supprimez *dbo* dans le champ **Schema**, si vous vous connectez à MSSQL 2005/2008.

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

La connexion aux bases 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.



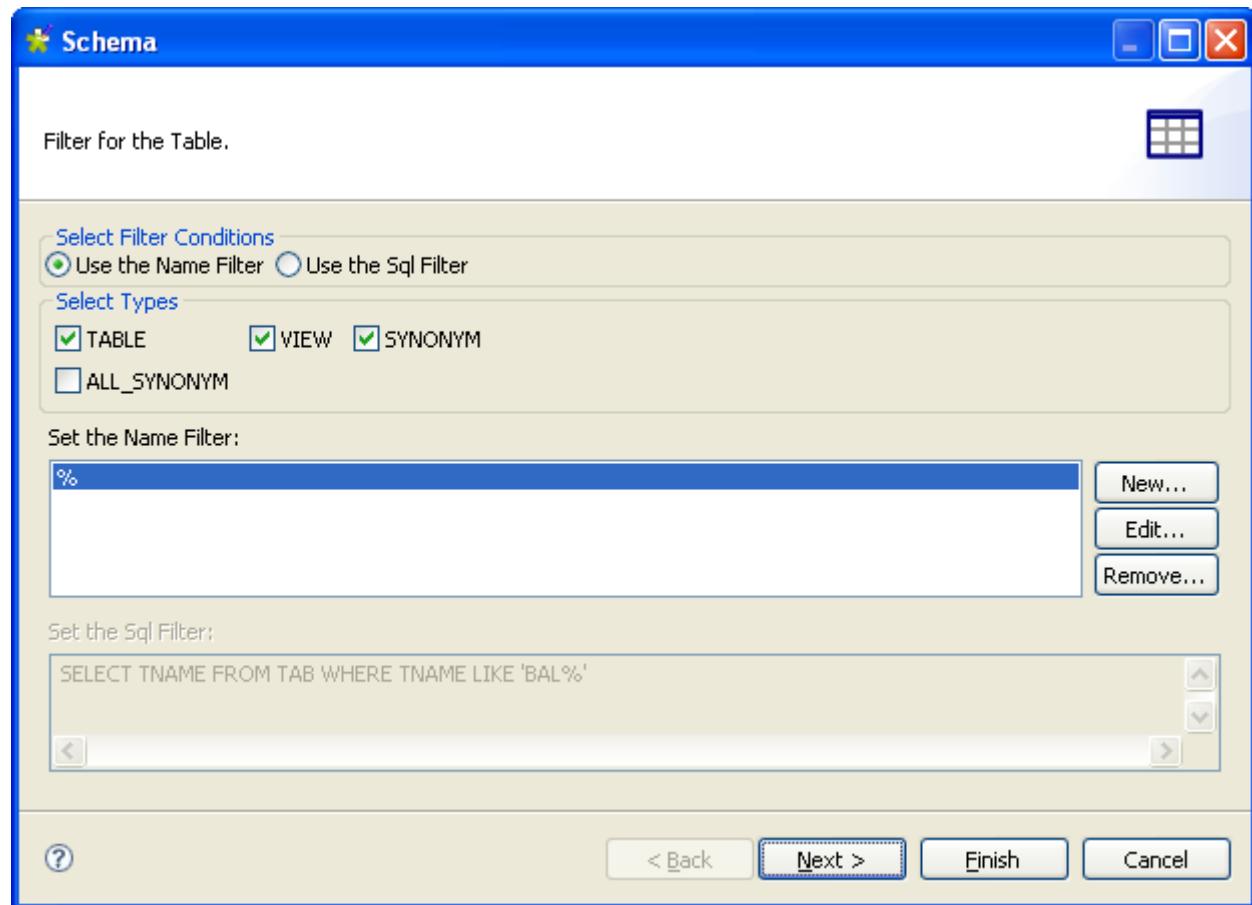
A présent, cliquez droit sur la connexion nouvellement créée et sélectionnez **Retrieve schema** dans le menu contextuel afin de charger le schéma de la table, dans la connexion établie.



Un message d'erreur s'affichera s'il n'y a aucune table à retrouver dans la base de données sélectionnée, ou si vous n'avez pas les droits d'accès à cette base de données.

7.2.3 Etape 3 : Chargement de la table

Lorsque vous cliquez sur **Retrieve schema**, un nouvel assistant apparaît qui vous donne la possibilité d'afficher et de filtrer les différents objets (tables, vues et synonymes) de votre connexion de base de données.



Dans la zone **Select Filter Conditions**, vous pouvez filtrer les objets de votre base de données soit à l'aide d'un assistant dans le champ **Set the Name Filter**, soit à l'aide d'une requête SQL dans le champ **Use the Sql Filter**.

Pour filtrer les objets d'une base de données par leur nom, observez les étapes suivantes :

- Dans la zone **Select Filter Conditions**, sélectionnez l'option **Use the Name Filter**.
- Dans la zone **Select Types**, cochez la/les case(s) du/des objet(s) de la base de données que vous voulez filtrer ou afficher.



Les options disponibles peuvent varier en fonction de la base de données sélectionnée.

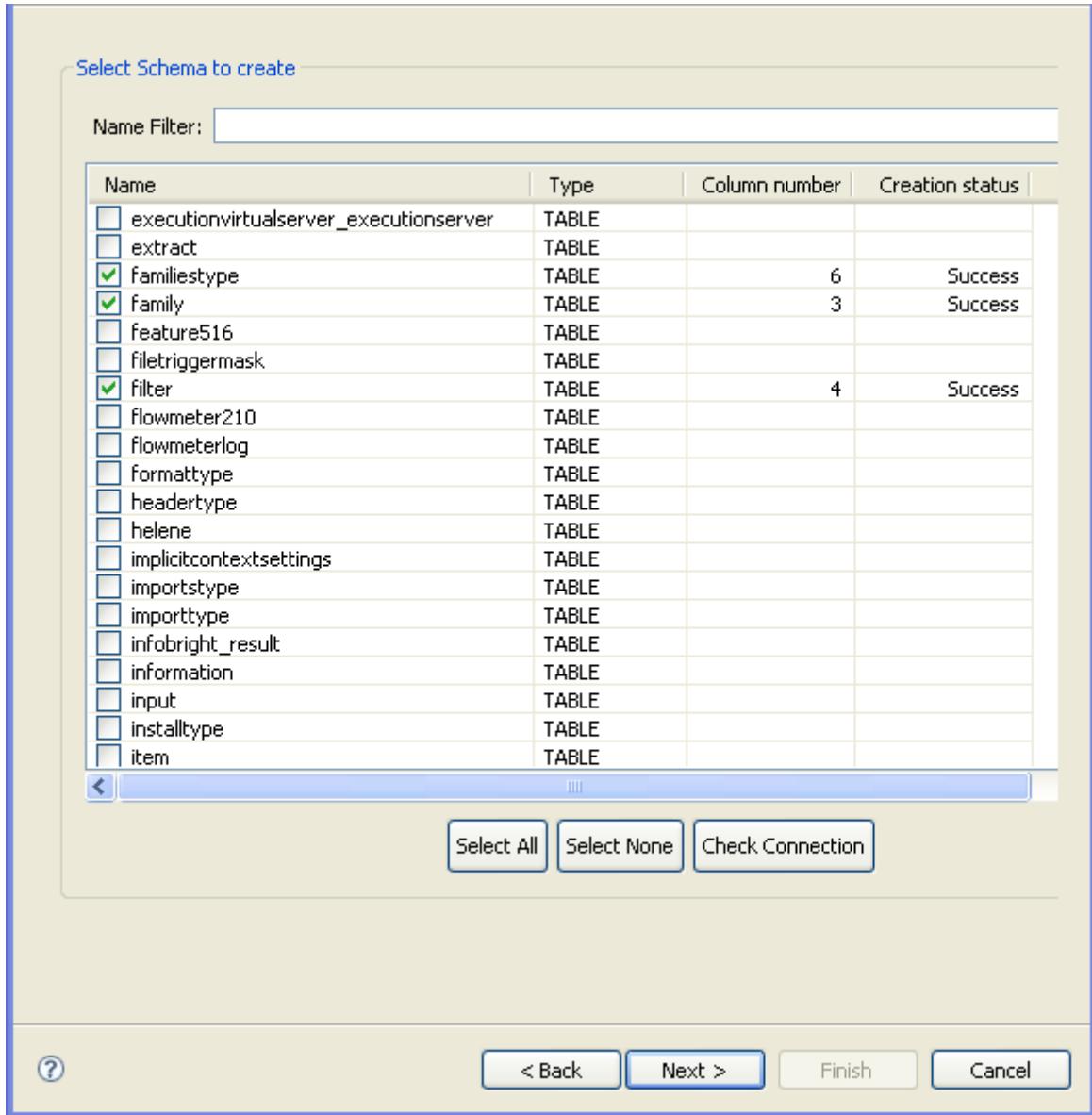
- Dans la zone **Set the Name Filter**, cliquez sur le bouton **Edit...** pour ouvrir la boîte de dialogue **[Edit Filter Name]**.
- Dans la boîte de dialogue, saisissez le filtre souhaité. Par exemple, si vous souhaitez récupérer les objets dont le nom commence par "A", saisissez le filtre "A%" ou si vous souhaitez récupérer les objets dont le nom termine par "type", saisissez "%type".
- Cliquez sur **OK** pour fermer la boîte de dialogue.
- Cliquez sur **Next** pour afficher la vue suivante de l'assistant, qui présente la liste des objets filtrés.

Pour filtrer les objets à l'aide d'une requête SQL :

- Dans la zone **Select Filter Conditions**, sélectionnez l'option **Use Sql Filter**.

- Dans le champ **Set the Sql Filter**, saisissez la requête SQL souhaitée.
- Cliquez sur **Next** pour afficher la vue suivante de l'assistant, qui présente la liste des objets filtrés.

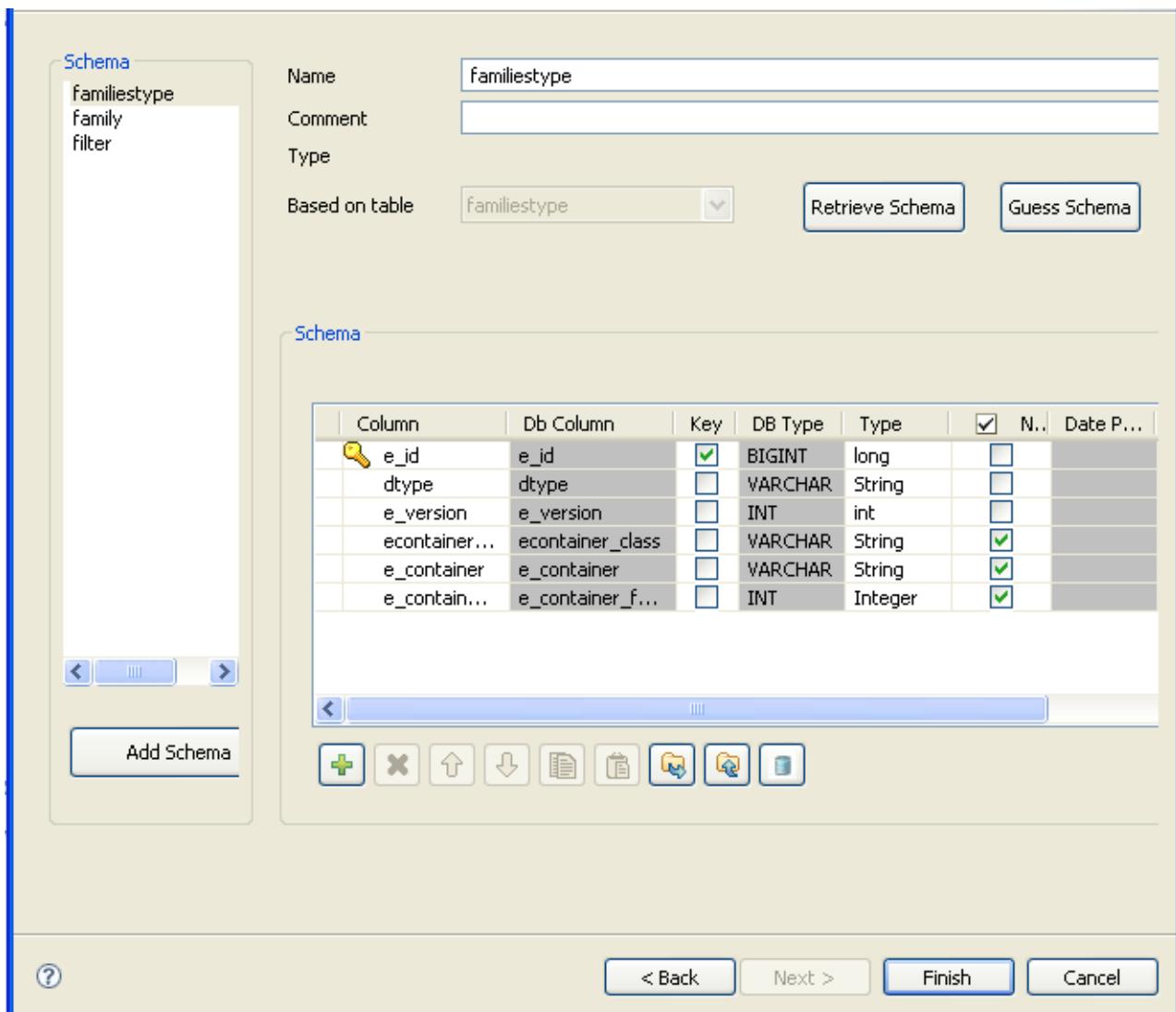
Une fois que vous avez obtenu la liste filtrée des différents objets de la base de données (table, vue et synonyme), observez les étapes suivantes pour charger les schémas des objets sélectionnés dans votre fichier système **Repository** :



- Sélectionnez un ou plusieurs objets à partir de la liste puis cliquez sur le bouton **Next** pour ouvrir la vue suivante de l'assistant, où vous pourrez consulter les schémas des objets sélectionnés.



Si aucun schéma n'est visible dans la liste, cliquez sur le bouton **Check connection** en dessous de la liste pour vérifier l'état de la connexion de base de données.



- Modifiez les schémas si besoin puis cliquez sur le bouton **Finish** pour fermer l'assistant.

Les schémas reposant sur les tables sélectionnées sont listés dans le répertoire **Table schemas**, qui correspond à la connexion de base de données que vous avez créée.



En Java, assurez-vous que le type de données, dans la colonne **Type**, est correctement défini.

Pour plus d'informations concernant les types de données, dont les modèles de date, consultez la page <http://java.sun.com/j2se/1.5.0/docs/api/index.html>.

7.2.4 Etape 4 : Définition du schéma

Par défaut, le schéma affiché dans la zone **Schema** est basé sur la première table sélectionnée dans la liste des schémas chargés (à gauche). Vous pouvez modifier le nom du schéma et selon vos besoins, vous pouvez également personnaliser la structure du schéma.

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

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

Cliquez sur **Finish** pour terminer la création du schéma de base de données. Tous les schémas ainsi créés apparaissent sous le noeud des connexions aux bases de données.

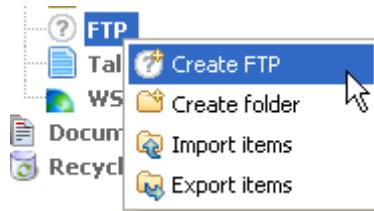
7.3 Configurer une connexion FTP

Si vous vous connectez régulièrement à un serveur FTP vous pouvez centraliser les informations de connexion à cette base de données sous le noeud **Metadata** de la vue **Repository**.

7.3.1 Etape 1 : Propriétés générales

Pour créer une connexion à un serveur FTP, suivez la procédure décrite ci-dessous.

- Développez le noeud **Metadata** dans la vue **Repository**.



- Cliquez-droit sur **FTP** puis sélectionnez **Create FTP** dans le menu contextuel.
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**). Pour plus d'informations concernant la définition des préférences, consultez *Configurer les préférences de Talend Open Studio, page 21*.
- Cliquez sur **Next** une fois terminé. L'étape suivante vous permet de renseigner les données de connexion au serveur FTP.

7.3.2 Etape 2 : Connexion

Dans cette étape, renseignez les informations et les paramètres de connexion à votre serveur FTP.



Dans la zone **Server** :

- Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant de connexion au serveur et votre mot de passe.
- Dans le champ **Host**, saisissez le nom de l'hôte de votre serveur FTP.
- Dans le champ **Port**, saisissez le numéro du port correspondant.
- Dans la liste **Encoding**, sélectionnez l'encodage utilisé.
- Dans la liste **Connection Model**, sélectionnez le mode de connexion que vous souhaitez utiliser.
Sélectionnez **Passive** si vous souhaitez que le serveur FTP détermine le port de connexion pour le transfert des données.
Sélectionnez **Active** si vous souhaitez déterminer vous-même ce port.

Dans la zone **Parameter**, sélectionnez le type d'utilisation du serveur FTP. Pour une utilisation standard de votre serveur FTP, vous n'avez pas besoin de sélectionner une option.

- Cochez la case **SFTP Support** pour utiliser votre serveur FTP via le protocole de sécurité SSH.
Une liste **Authentication method** apparaît alors. Sélectionnez **Public key** ou **Password** selon ce que vous utilisez.
- Cochez la case **FTPs Support** pour utiliser votre serveur via le protocole de sécurité SSL.
- Cochez la case **Use Socks Proxy** si vous souhaitez utiliser un proxy Socks, puis renseignez les informations du proxy (respectivement le nom de l'hôte, le numéro du port, l'identifiant et le mot de passe de l'utilisateur).

- Cliquez sur **Finish** pour fermer l'assistant.

Toutes les connexions ainsi créées apparaissent sous le noeud des connexions aux serveurs FTP, dans la vue **Repository**.

Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d'informations concernant l'action de déposer des métadonnées dans l'espace de modélisation graphique, consultez *Déposer des composants à partir du répertoire Metadata, page 62*.

7.4 Configurer une connexion JDBC

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

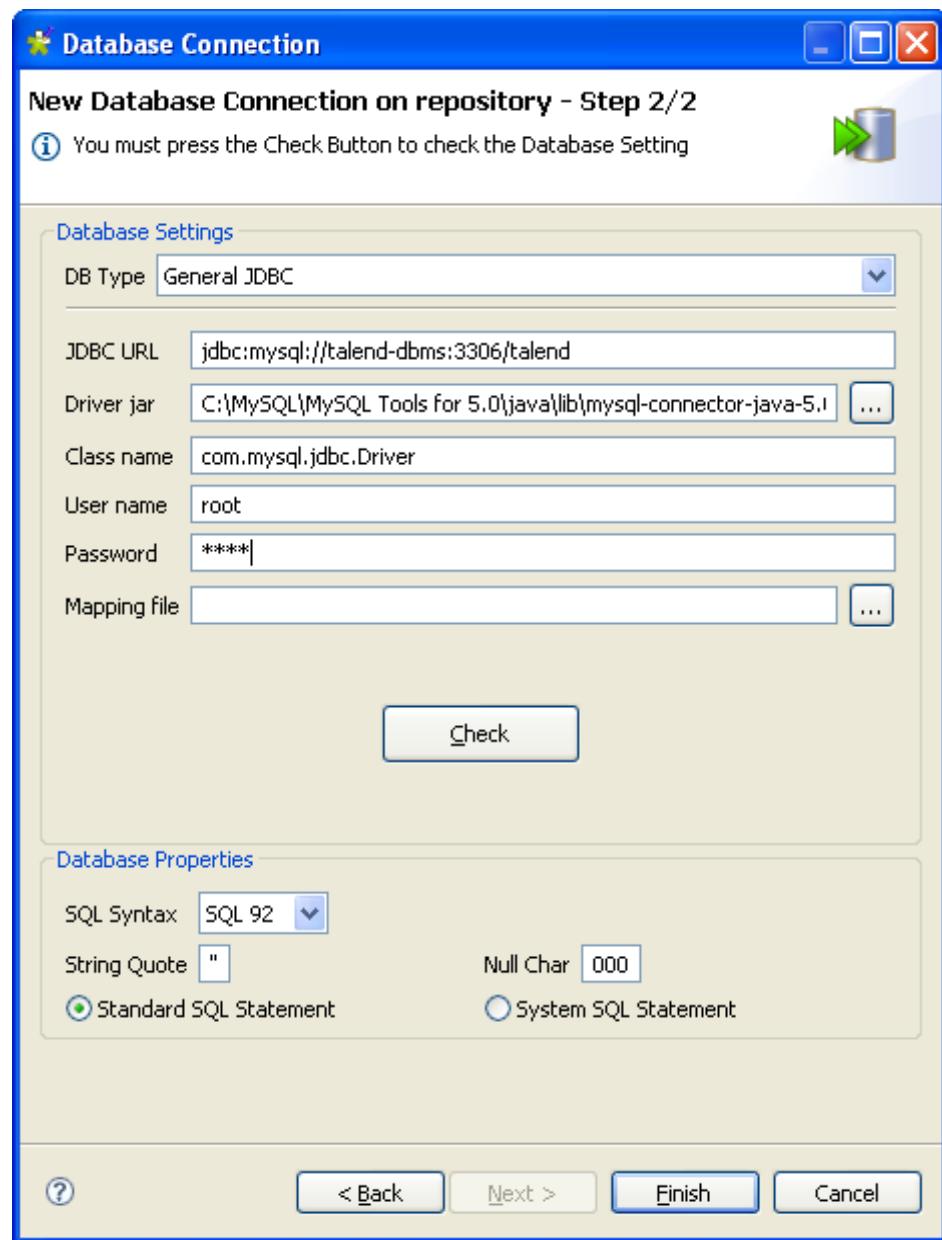
7.4.1 Etape 1 : Propriétés générales

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 190*.

7.4.2 Etape 2 : Connexion

Dans le champ **DB type**, sélectionnez l'option **General JDBC** et renseignez les champs qui suivent.



Renseignez les informations requises pour la connexion :

- Dans le champ **JDBC URL**, renseignez l'URL de connexion au serveur SGBD.
- Dans le champ **Driver jar**, sélectionnez le pilote jar permettant de valider votre connexion à la base de données.
- Dans le champ **Class name**, renseignez la classe principale du pilote permettant de communiquer avec la base de données.
- Dans les champs **User name** et **Password**, renseignez vos nom et mot de passe utilisateur.
- Dans le champ **Mapping File**, sélectionnez le mapping permettant de faire la corrélation entre le type de données de la base de données et le type de données en Java ou en Perl (respectivement les champ DB Type et Type dans le schéma). Par exemple : le type VARCHAR pour un base de données correspond au type String en Java et en Perl.

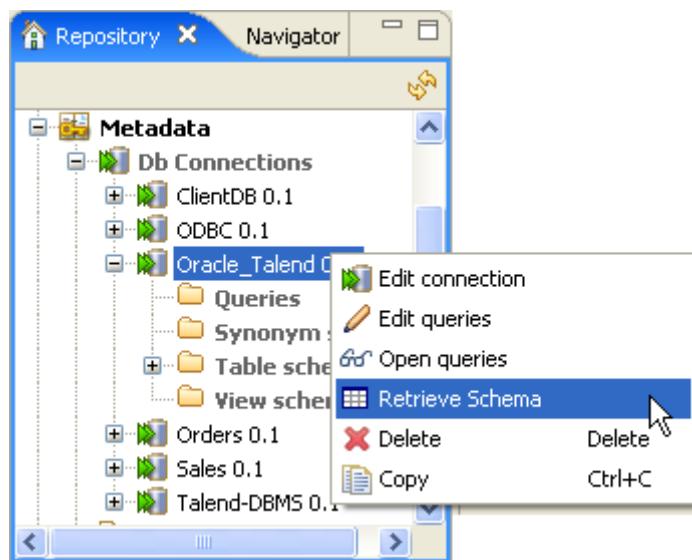


Les fichiers de mapping sont au format XML et sont définis dans le menu **Window > Preferences** sous le noeud **Talend > Metadata of TalendType**.

Et vérifiez votre connexion à l'aide de la fonction **Check**.

La première phase de configuration d'une connexion base de données est terminée. Cliquez sur **Finish** pour valider.

La connexion aux bases 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.

7.4.3 Etape 3 : Chargement de la table

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

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>.

7.4.4 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 d'importer un schéma XML à partir d'un fichier ou d'exporter le schéma courant en XML.

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

Cliquez sur **Finish** pour terminer la création du schéma de base de données. Tous les schémas ainsi créés apparaissent sous le noeud des connexions aux bases de données.

7.5 Configurer une connexion SAS

Talend Open Studio vous permet de configurer une connexion à un système distant SAS.

7.5.1 Conditions préalables

Avant de configurer votre connexion SAS, vous devez récupérer vos métadonnées à partir du serveur SAS et les exporter sous forme de fichier XML.

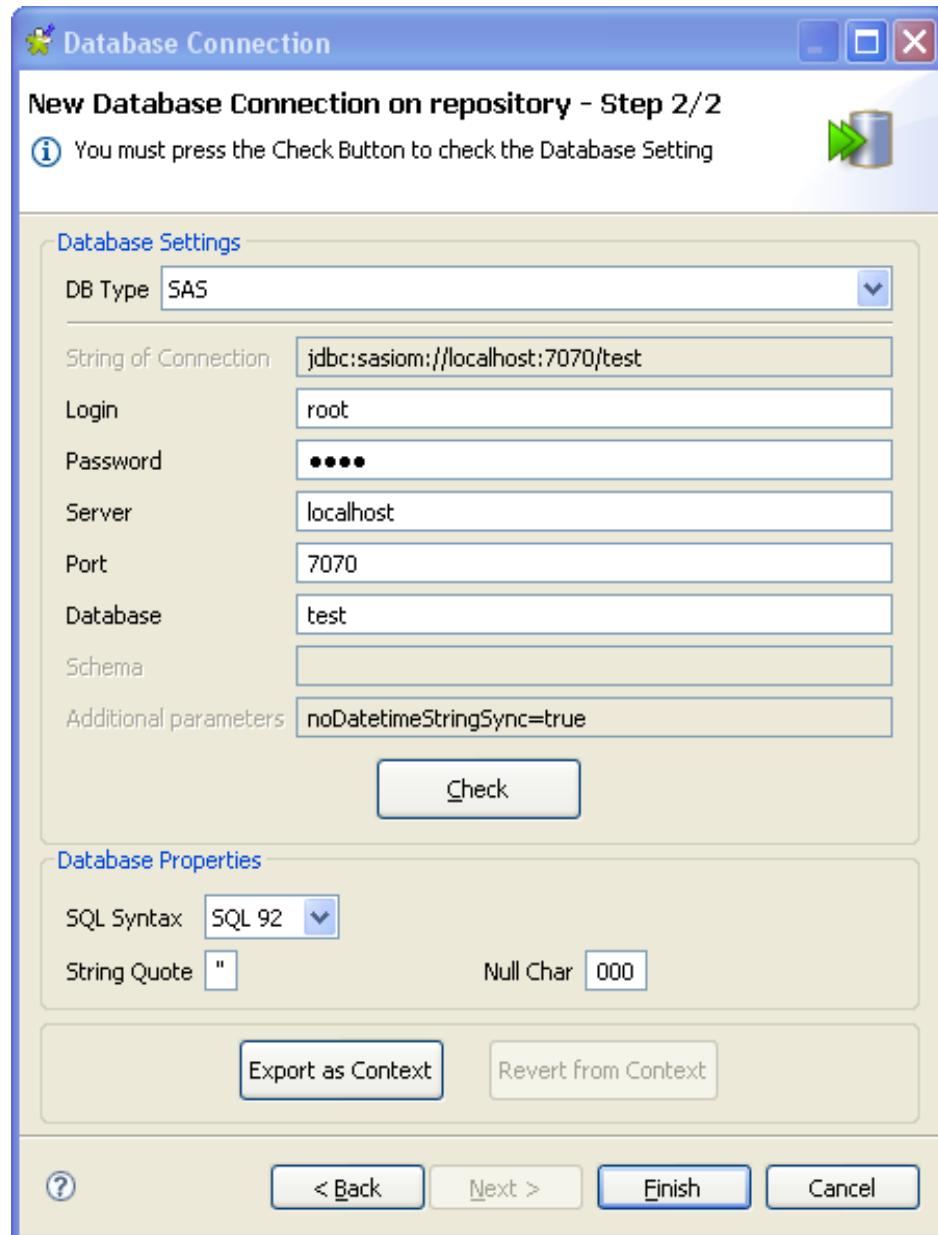
7.5.2 Etape 1 : Propriétés générales

- Dans l'arborescence **Repository** de Talend Open Studio, développez le noeud **Metadata** puis cliquez droit sur **DB Connection**.
- Sélectionnez **Create connection** dans le menu contextuel pour ouvrir l'assistant [**Database Connection**].
- Remplissez les informations génériques, comme le nom (**Name**) et la description (**Description**) puis cliquez sur le bouton **Next** pour passer à la vue suivante de l'assistant.

Pour plus d'informations, consultez *Etape 1 : Propriétés générales, page 190*.

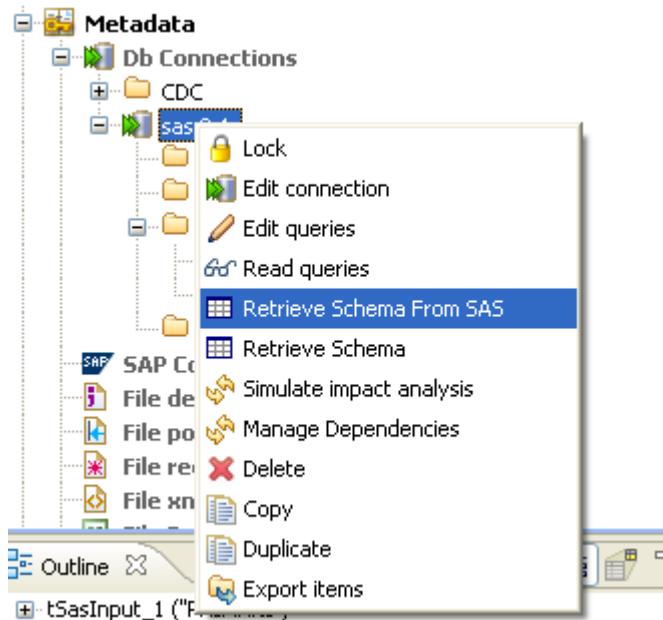
7.5.3 Etape 2 : Connexion

- Dans le champ **BD type** de l'assistant [**Database Connection**], sélectionnez l'option **SAS** et renseignez les champs qui suivent avec vos informations de connexion SAS.



- Si nécessaire, vérifiez votre connexion à l'aide de la fonction **Check**.
- Si nécessaire également, remplissez les propriétés de base de données dans la zone **Database Properties**.
- Cliquez sur **Finish** pour valider vos modifications et fermer l'assistant.

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



- Cliquez droit sur la connexion SAS nouvellement créée et sélectionnez **Retrieve Schema From SAS** dans le menu contextuel pour afficher tous les schémas de la base de données dans le sous-dossier **Table schemas**.

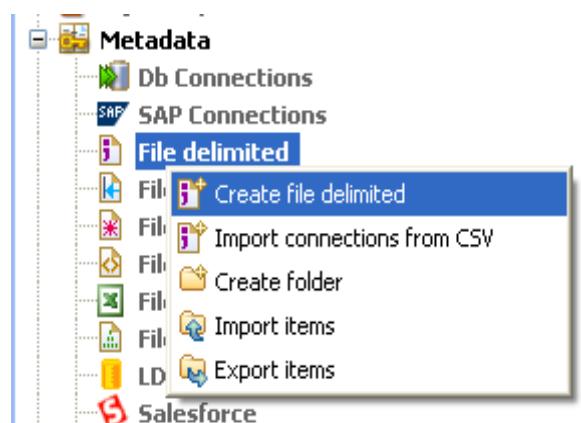
7.6 Configurer un schéma File Delimited

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.



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.

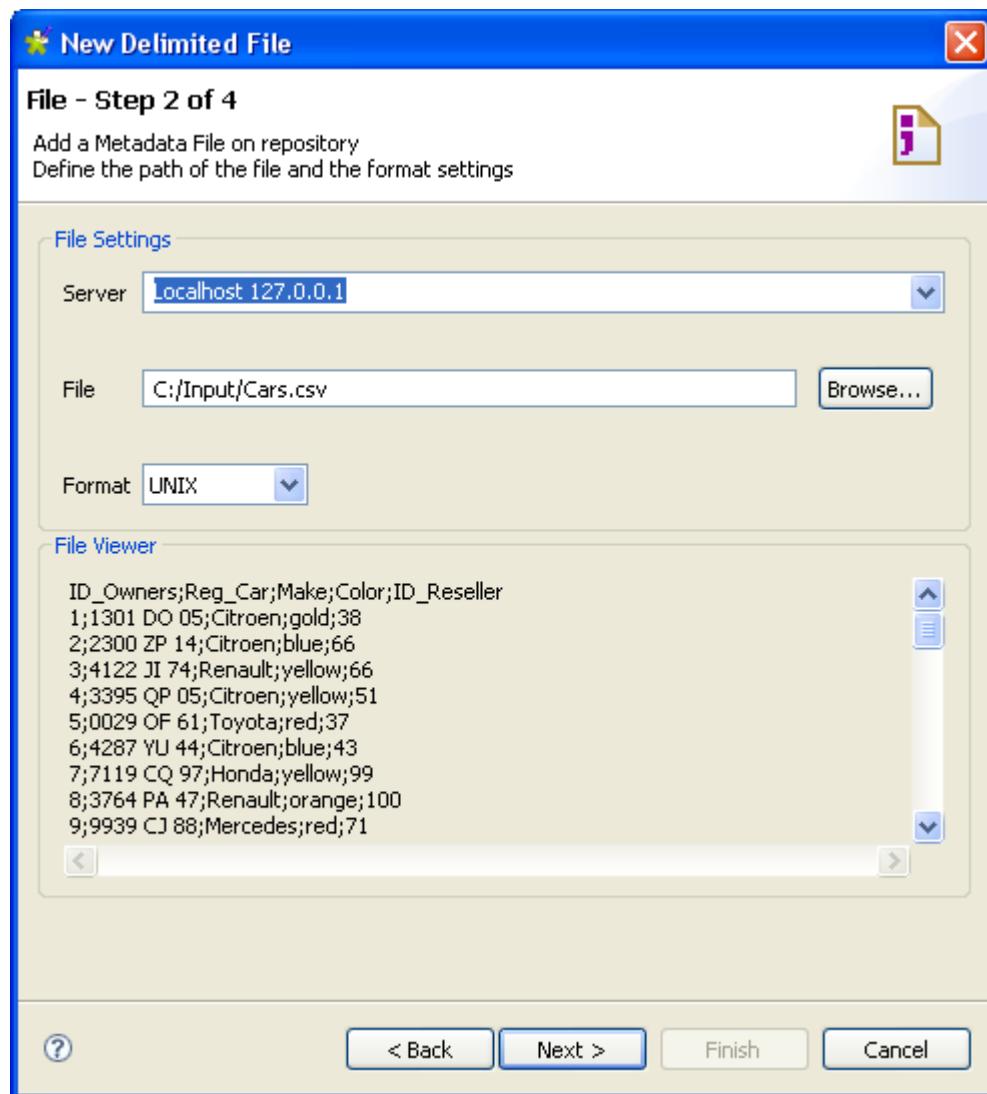
7.6.1 Etape 1 : Propriétés générales

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 190.*

7.6.2 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**.



Selectionnez 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.

7.6.3 Etape 3 : Définition du schéma

Dans cette vue, vous pouvez affiner la destination de vos données et les paramètres de fichier. Cliquez sur les différentes zones (bleu foncé) 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:

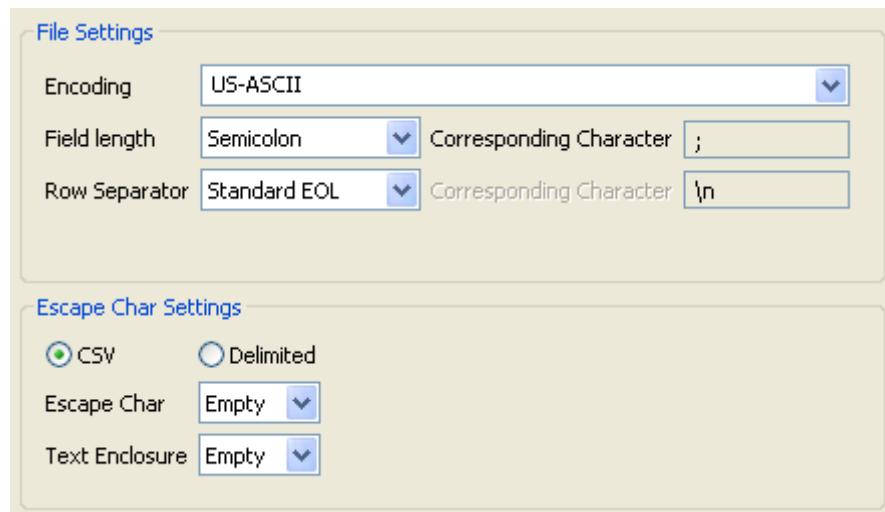
Preview

Set heading row as column names

id	CustomerName	CustomerAddress	idReseller	Make	RegTime	Column 6	Column 8
1	Griffith Paving and Sealcoatin	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	22/06/2010 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 Sheridan rd.	9	1	24/03/2008 23:07	1981-08-02 01:26:40.000	74168.
9	Childress Child Day Care	700 Tannery Ave.	12	1	10/06/2001 06:22	1991-07-02 11:12:20.000	92176.

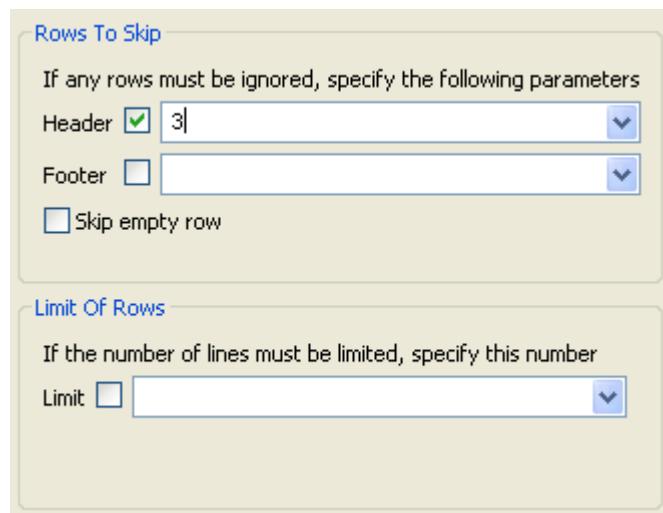
[?](#)

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



En fonction de votre type de fichier (csv ou delimited), vous pouvez également paramétriser 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.



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 vos paramétrages.

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						
<input checked="" type="checkbox"/> Set heading row as column names Refresh Preview						
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	hikha	TEUOOGCN

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

7.6.4 Etape 4 : Schéma final

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

The screenshot shows the 'New Delimited File' dialog box, specifically Step 4 of 4. The title bar says 'File - Step 4 of 4'. Below it, there's a note: 'Add a Schema on repository' and 'Define the Schema'. A large text input field contains the name 'metadata'. A smaller input field for comments is empty. Below these, a section titled 'Schema' contains a table preview with the heading 'Click to update schema preview' and a 'Guess' button. The table has columns: Column, Key, Type, Nullable, Length, Preci..., D..., Comm... . It lists five columns: ID_Owners (int, nullable, length 2), Reg_Car (String, nullable, length 10), Make (String, nullable, length 10), Color (String, nullable, length 6), and ID_Reseller (int, nullable, length 3). At the bottom of the schema section are several icons for managing the schema. The footer of the dialog box includes a question mark icon, '< Back' and 'Next >' buttons, and 'Finish' and 'Cancel' buttons.

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			

Si le fichier délimité sur lequel est basé le schéma a été 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.

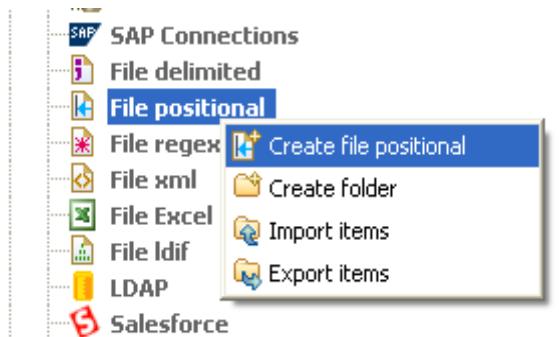
Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d'informations concernant l'action de déposer des métadonnées de composants dans l'espace de modélisation graphique, consultez *Déposer des composants à partir du répertoire Metadata, page 62*.

Pour plus d'informations concernant l'action de déposer des métadonnées de composants dans l'espace de modélisation graphique, consultez *Déposer des composants à partir du répertoire Metadata, page 62.*

7.7 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**.

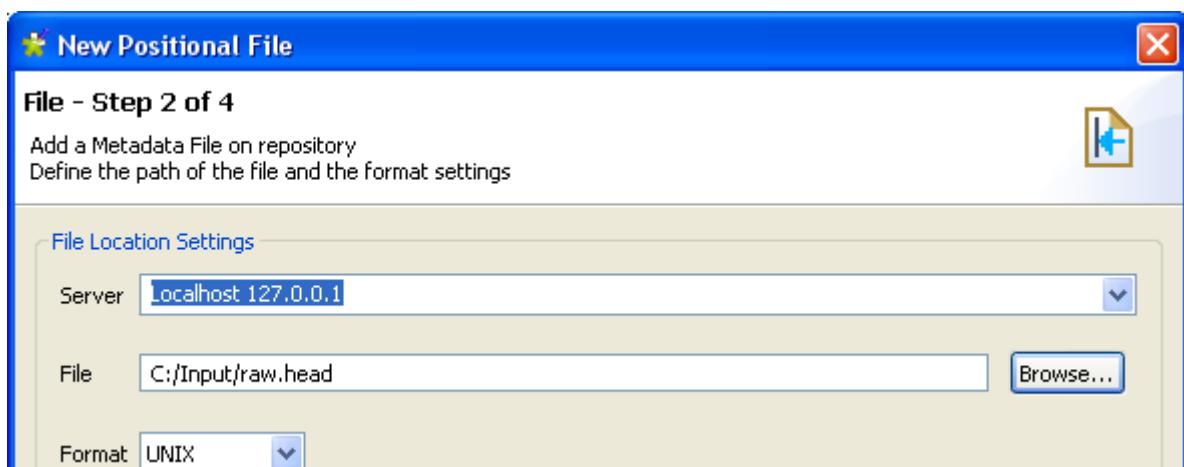
7.7.1 Etape 1 : Propriétés générales

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

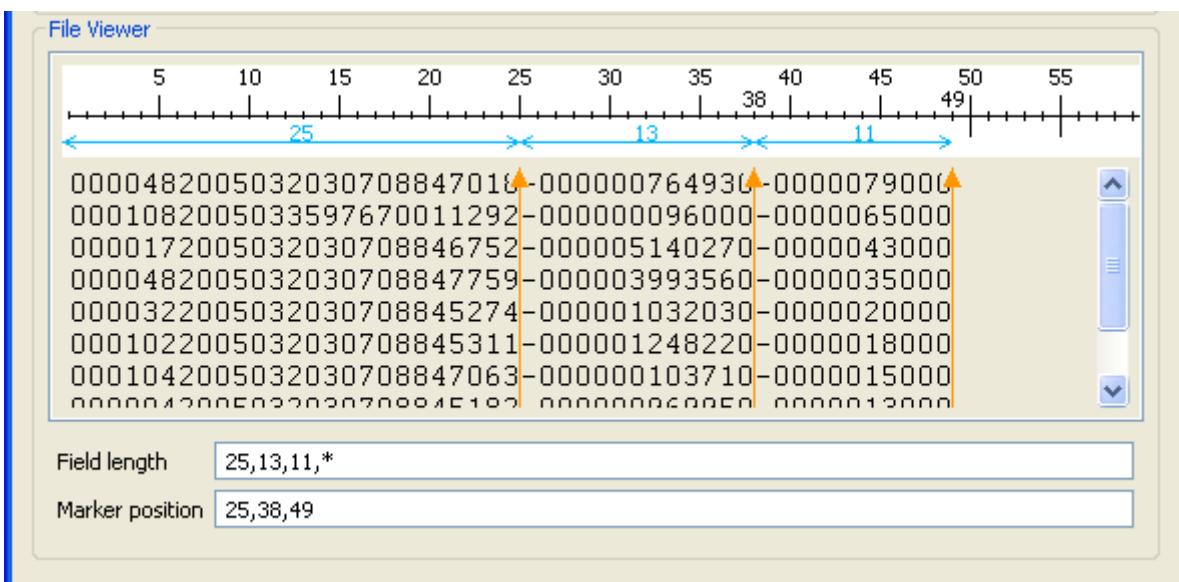
7.7.2 Etape 2 : Connexion et chargement du fichier

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 séparateur 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.

7.7.3 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**.

7.7.4 Etape 4 : Schéma final

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

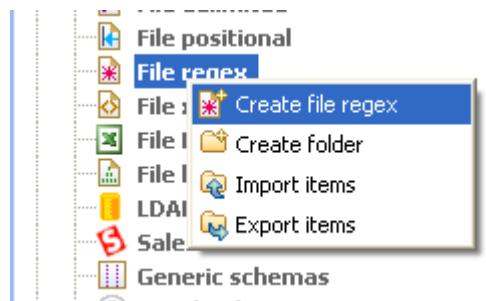
Vous pouvez personnaliser le nom de la métadonnée (par défaut, metadata) et apporter des modifications à l’aide de la barre d’outils.

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**.

Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

7.8 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**.

7.8.1 Etape 1 : Propriétés générales

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

7.8.2 Etape 2 : Chargement du fichier

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

A screenshot of the 'File Settings' configuration dialog. The 'File Settings' tab is active, showing fields for 'Server' (localhost 127.0.0.1), 'File' (C:\Input\Eclipse.log), and 'Format' (UNIX). Below this, the 'File Viewer' tab is open, displaying the contents of the log file: '***** 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', and '!ENTRY org.talend.designer.runprocess 1 0 2006-09-07 10:39:51.377'. The dialog has scroll bars on the right side.

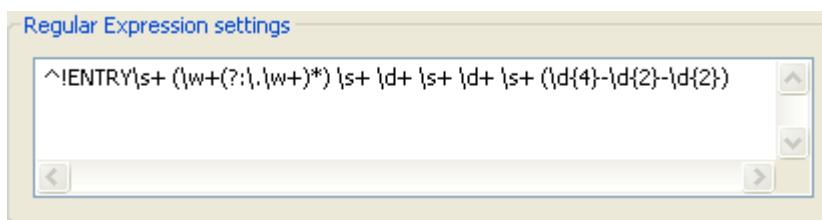
Comme pour la création d'un schéma Delimited File, le format est demandé pour pré-remplir certains champs des étapes suivantes. Si le système d'exploitation de création du fichier n'est pas proposé dans la liste, ignorez ce champ.

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

7.8.3 Etape 3 : Définition du schéma

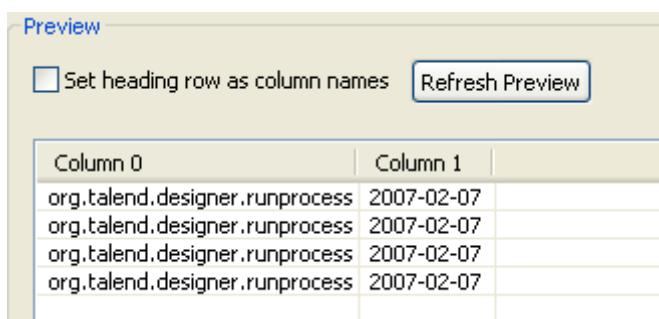
Cette étape ouvre la fenêtre de paramétrage du schéma. Comme pour les autres schémas de fichier, vous pouvez affiner la définition du schéma en précisant les séparateurs de champs et de lignes (**Field separator** et **Row separator**), le nombre de lignes d'en-tête, etc.

Dans le panneau de paramétrage des expressions régulières, **Regular Expression settings**, saisissez l'expression régulière utilisée pour délimiter le fichier (en Perl ci-dessous).



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.



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

7.8.4 Etape 4 : Schéma final

Vous pouvez personnaliser le nom de la métadonnée (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.

Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d'informations concernant l'action de déposer des métadonnées de composants dans l'espace de modélisation graphique, consultez *Déposer des composants à partir du répertoire Metadata, page 62.*

7.9 Configurer un Schéma File XML

Centralisez vos requêtes XPath à partir d'un fichier XML spécifié et regroupez les valeurs récupérées.

Procédez de la même façon que pour les connexions de fichier délimité ou positionnel.

Cet assistant intègre deux assistants en un seul. En effet, selon vos besoins, vous pouvez créer un schéma de lecture (Input), ou un schéma d'écriture (Output). Dans un Job, le composant **tFileInputXML** utilise le schéma de lecture créé pour lire des fichiers XML. Dans un Job, le composant **tAdvancedFileOutputXML** utilise le schéma d'écriture créé pour écrire un fichier XML, s'il n'existe pas, ou y ajouter des informations, s'il existe déjà.



L'étape 1, dans laquelle vous devez renseigner les propriétés générales du futur schéma, précède l'étape de sélection du type Input ou Output. Veillez donc à différencier vos Inputs et vos Outputs en saisissant des noms que vous pourrez reconnaître.

Si vous souhaitez lire un fichier XML, consultez la section *Configurer un schéma XML pour un fichier d'entrée (Input), page 214.*

Si vous souhaitez écrire un fichier XML, consultez la section *Configurer un schéma XML pour un fichier de sortie (Output), page 220.*

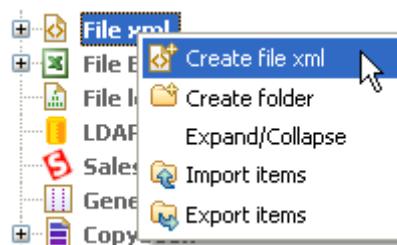
7.9.1 Configurer un schéma XML pour un fichier d'entrée (Input)

Cette section décrit la définition et le chargement d'un schéma XML pour un fichier d'entrée (Input). Pour définir et charger un fichier de sortie (Output), consultez *Configurer un schéma XML pour un fichier de sortie (Output), page 220.*

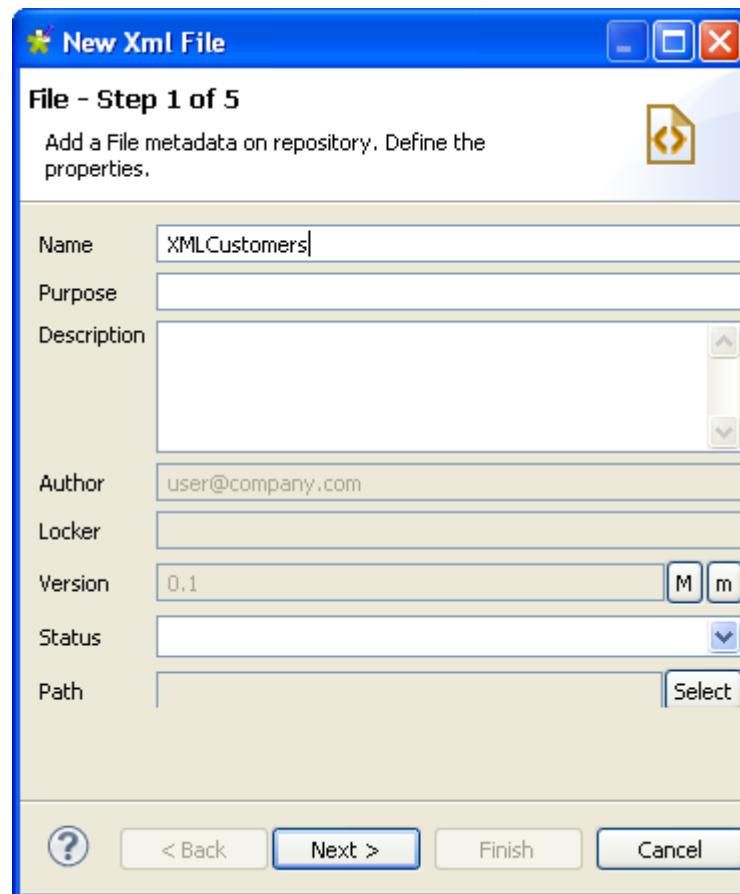
Etape 1 : Propriétés générales

Dans cette étape, vous allez définir les propriétés générales du fichier.

- Dans la vue **Repository**, développez le noeud **Metadata**.
- Cliquez-droit sur **File XML**, et sélectionnez **Create XML File** dans le menu contextuel.



- Remplissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.

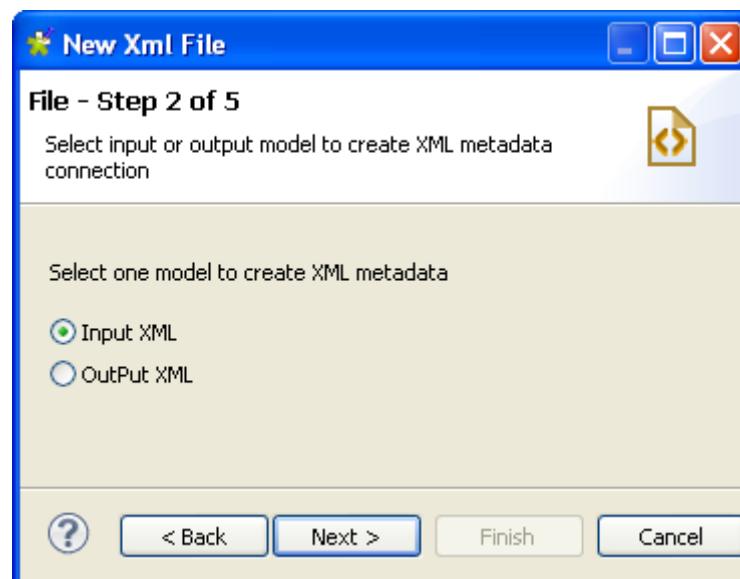


- Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

Etape 2 : Sélection du type de schéma (Input)

Dans cette étape, vous allez définir le type de schéma, Input ou Output. Dans cet exemple, sélectionnez le type Input.

- Dans la boîte de dialogue, sélectionnez **Input XML**.



- Cliquez sur **Next** pour continuer.

Etape 3 : Chargement du fichier Input

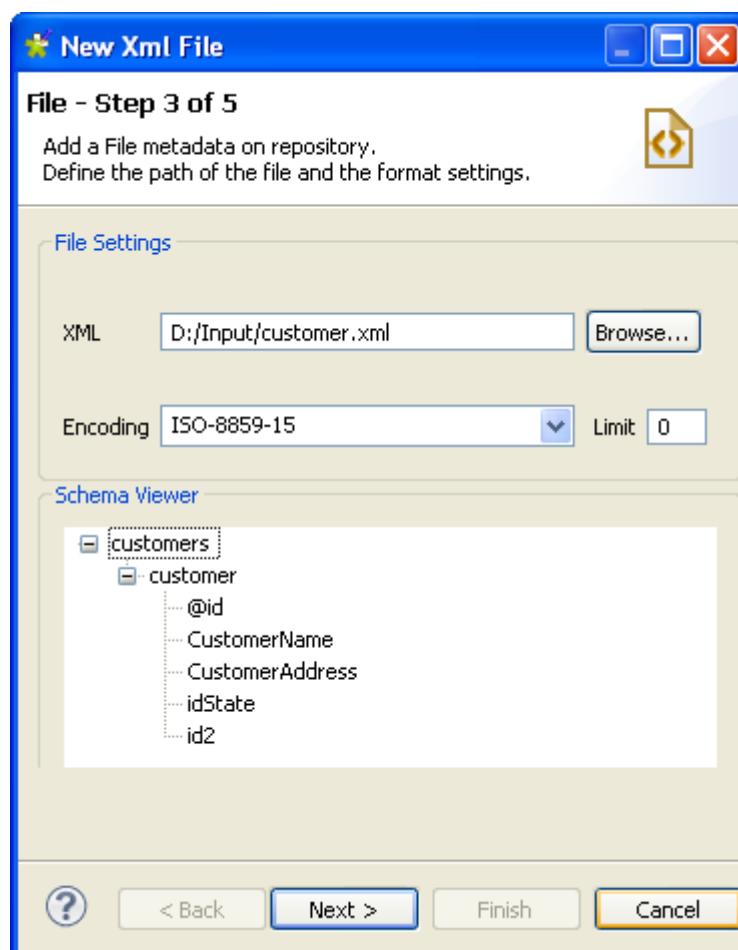
Dans cette étape, vous allez sélectionner votre fichier d'entrée, son encodage, définir le nombre de colonnes sur lequel la requête XPath devra être exécutée. Vous pouvez également prévisualiser la structure du fichier XML sélectionné.

- Cliquez sur le bouton [**Browse...**] et parcourez votre répertoire jusqu'au fichier à charger. Sinon, saisissez le chemin d'accès au fichier XML à charger.
Un aperçu de la structure du fichier lu s'affiche dans la zone **Schema Viewer**. Vous avez la possibilité de développer et de visualiser tous les niveaux de la structure XML du fichier.



Si votre fichier est un fichier XSD, il sera stocké dans le **Repository**. Les métadonnées ne seront donc pas affectées par la suppression ou le déplacement du fichier.

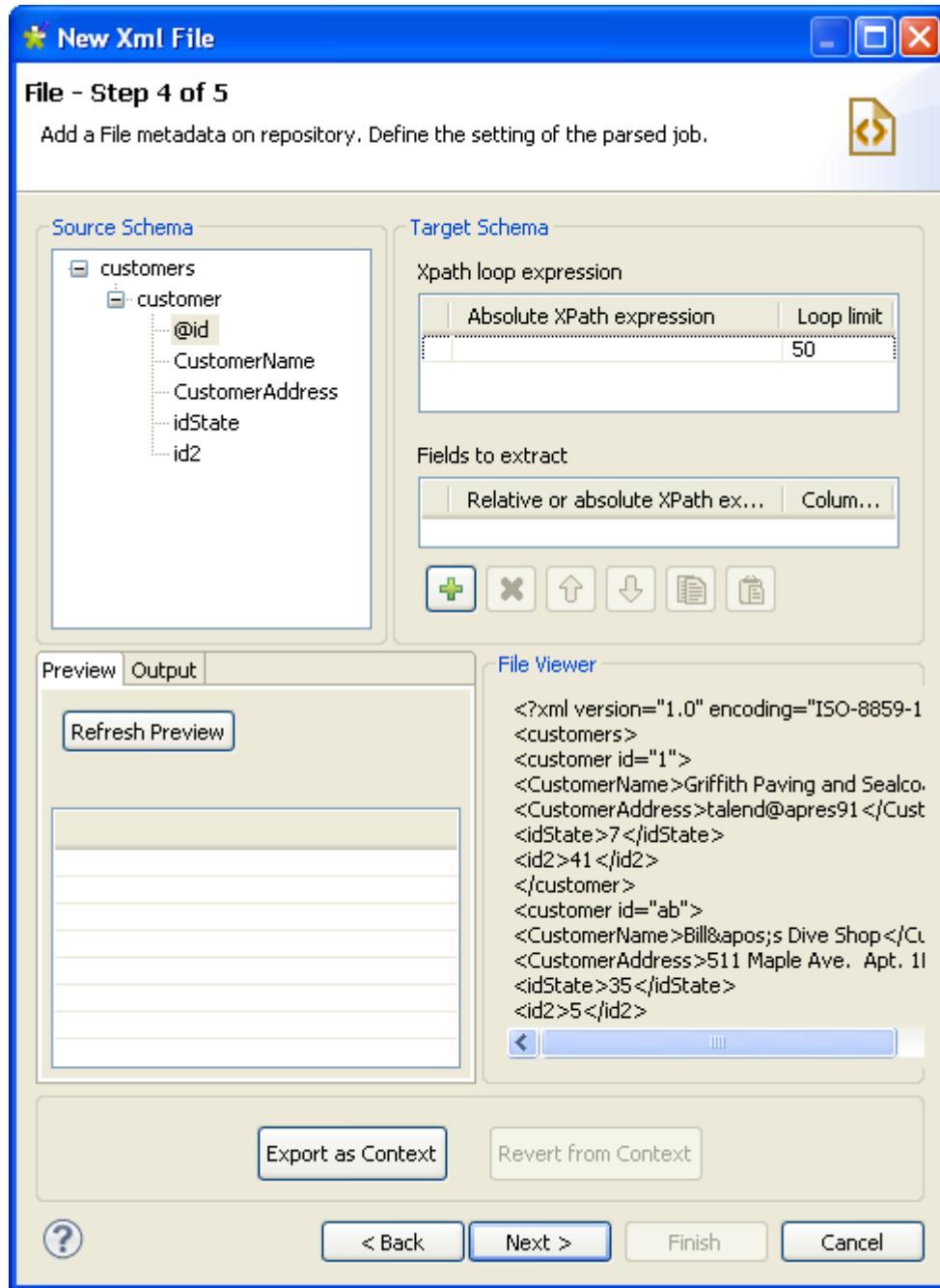
- Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement



- Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez 0 si vous souhaitez l'effectuer sur toutes les colonnes.
- Cliquez sur **Next** pour continuer et définir les paramètres du schéma.

Etape 4 : Définition du schéma

Dans cette étape, vous allez renseigner les paramètres à prendre en compte pour la définition du schéma.

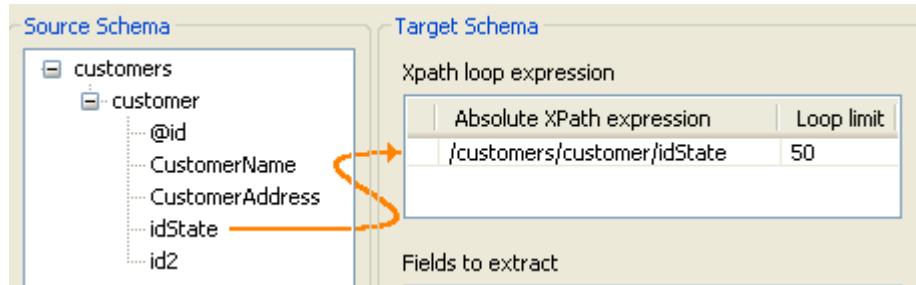


La fenêtre de définition du schéma est composée de quatre vues :

Vue	Description
Source Schema	Arborescence du fichier XML.
Target Schema	Information d'extraction et d'itération.
Preview	Aperçu du schéma cible.

Vue	Description
File viewer	Fenêtre d'aperçu des données brutes.

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



Ou,

- déposez le noeud sélectionné dans la structure source vers le champ Xpath du schéma cible. Un lien de couleur orange relie alors le noeud sélectionné à l'expression correspondante.



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

- Saisissez une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de noeuds à itérer.

Il est ensuite nécessaire de définir les champs à extraire. Pour ce faire :

- Glissez-déposez les noeuds appropriés de la vue **Source Schema** vers le champ **Relative or absolute XPath expression**.



Vous pouvez sélectionner plusieurs noeuds à déposer dans le tableau, en appuyant sur **Ctrl** ou **Maj**, et en cliquant sur les noeuds adéquats. La flèche reliant un noeud sélectionné dans la zone **Source Schema** à celui déposé dans la zone **Fields to extract** est bleue. Les autres liens sont gris.



Vous pouvez ajouter autant de colonnes à extraire que vous le souhaitez, supprimer des colonnes, ou en modifier l'ordre, à l'aide de la barre d'outils.

- Ajoutez ou supprimez une colonne à l'aide des boutons et .

- Modifiez l'ordre des colonnes à l'aide des boutons et .

- Dans le champ **Column name**, nommez les libellés des colonnes qui s'afficheront dans la vue d'aperçu du schéma (**Schema preview**).
- Cliquez sur **Refresh preview** pour afficher l'aperçu du schéma. Les champs apparaissent ainsi dans l'aperçu du schéma dans l'ordre imposé.

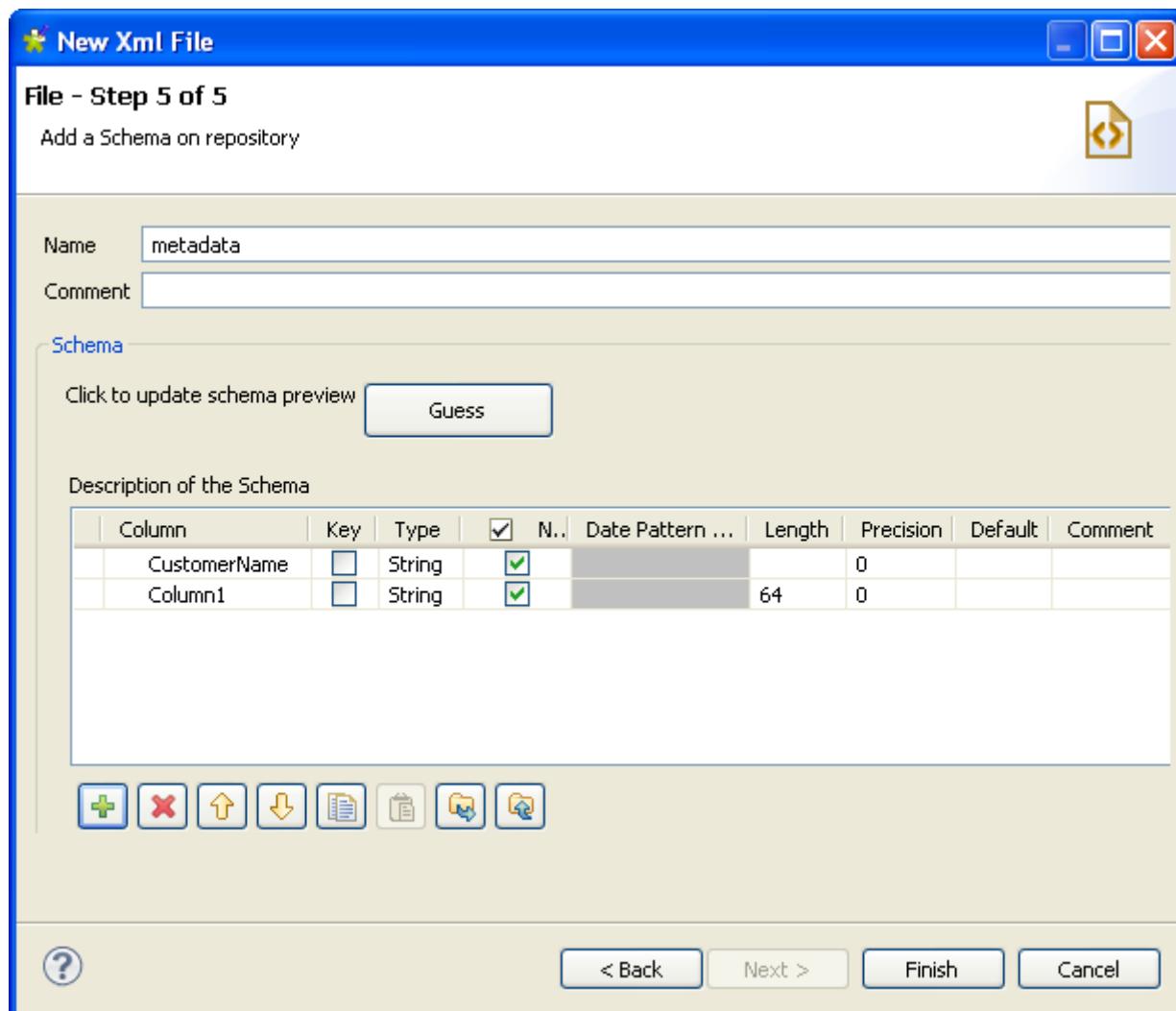
```

<?xml version="1.0" encoding="ISO-8859-15"?>
<customers>
<customer id="1">
<CustomerName>Griffith Paving and Sealcoatin</CustomerName>
<CustomerAddress>talend@apres91</CustomerAddress>
<idState>7</idState>
<id2>41</id2>
</customer>
<customer id="ab">
<CustomerName>Bill's Dive Shop</CustomerName>
<CustomerAddress>511 Maple Ave. Apt. 1B</CustomerAddress>
<idState>35</idState>
<id2>5</id2>
</customer>
<customer id="3">

```

Etape 5 : 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**.



- Ajoutez ou supprimez une colonne à l'aide des boutons et .
- Modifiez l'ordre des colonnes à l'aide des boutons et .
- Cliquez sur **Finish**.

Le nouveau schéma s'affiche dans le **Repository** sous le noeud **File xml** approprié.

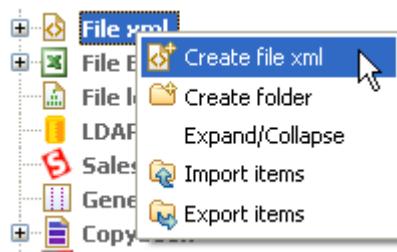
7.9.2 Configurer un schéma XML pour un fichier de sortie (Output)

Cette section décrit la définition et le chargement d'un schéma XML pour un fichier de sortie (Output). Pour définir et charger un schéma XML d'entrée (Input), consultez *Configurer un schéma XML pour un fichier d'entrée (Input)*, page 214.

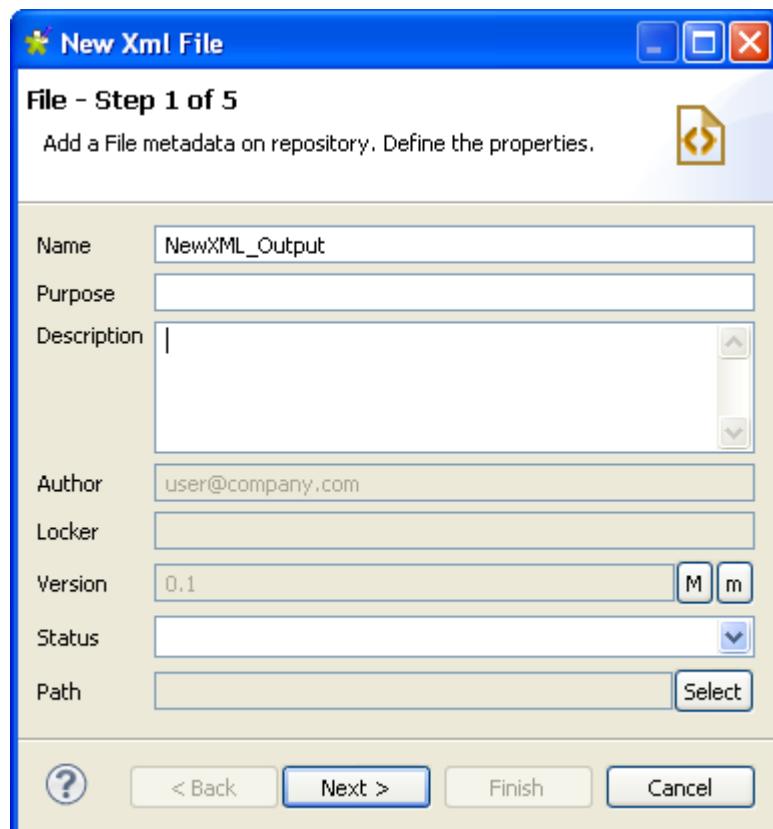
Etape 1 : Propriétés générales

Dans cette étape, vous allez définir les propriétés générales du fichier.

- Dans la vue **Repository**, développez le noeud **Metadata**.
- Cliquez-droit sur **File xml** et sélectionnez **Create file xml** dans le menu contextuel.



- Saisissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.

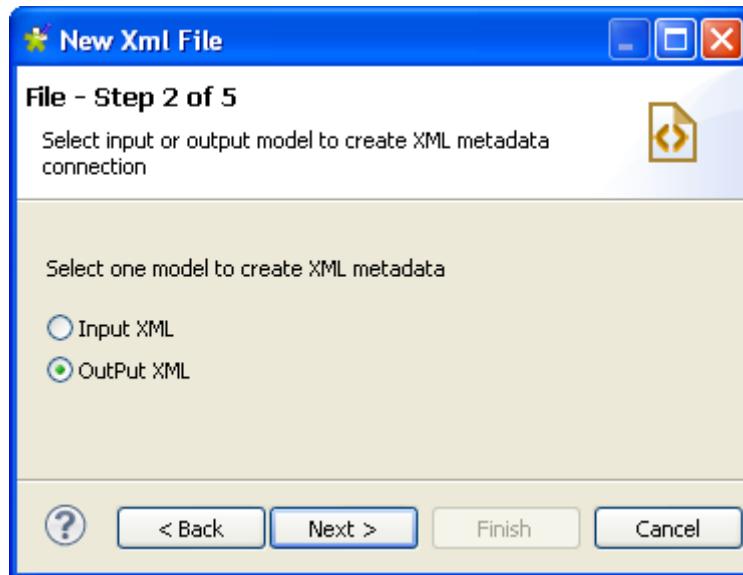


- Cliquez sur **Next** pour définir le type de schéma.

Etape 2 : Sélection du type de schéma (Output)

Dans cette étape, vous allez définir le type de schéma, Input ou Output. Dans cet exemple, sélectionnez le type Output.

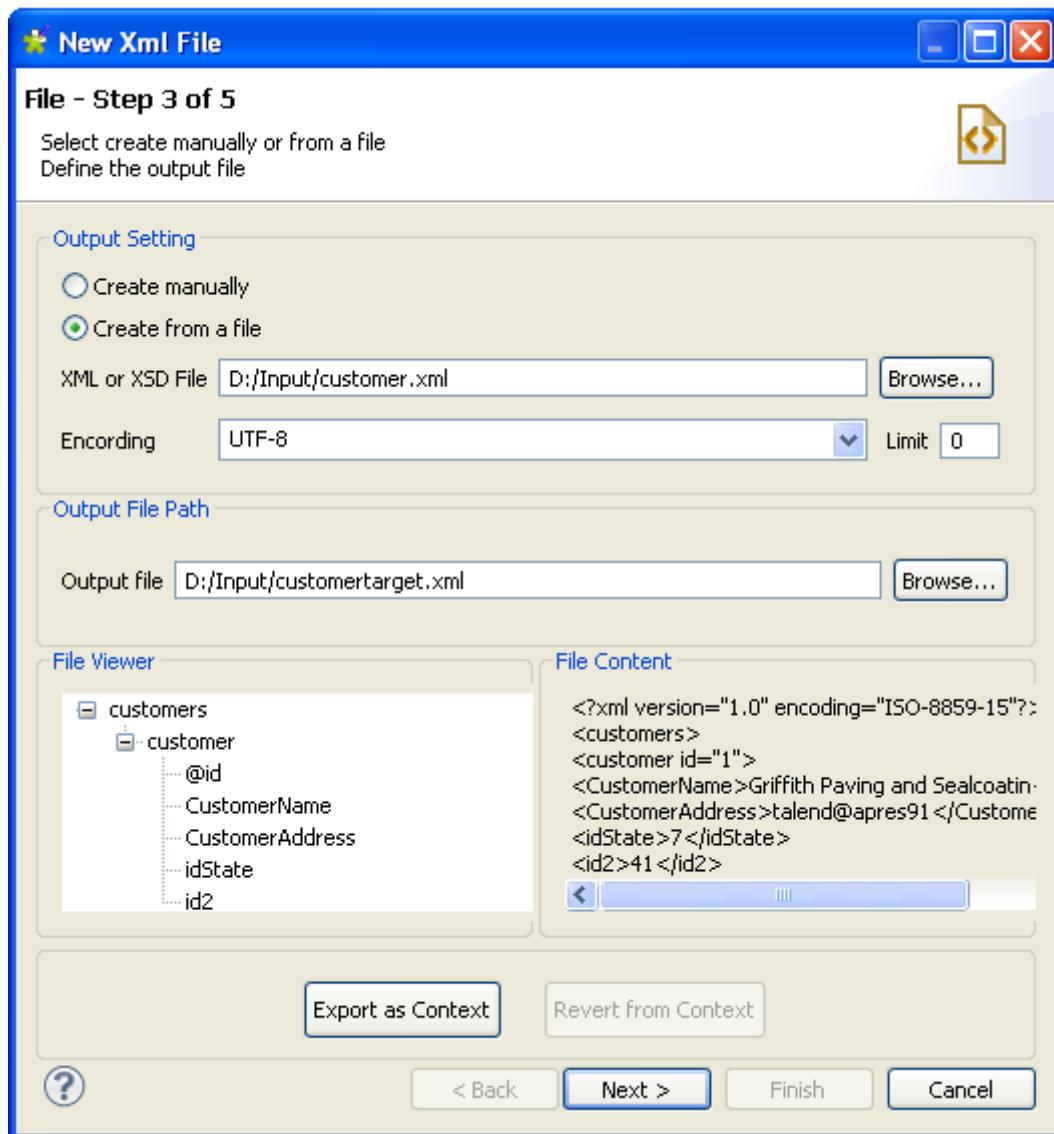
- Dans la boîte de dialogue, sélectionnez **Output XML**.



- Cliquez sur **Next** pour définir le fichier de sortie.

Etape 3 : Définition du fichier Output

Vous pouvez choisir de créer votre fichier manuellement ou de le créer à partir d'un fichier existant. Vous pouvez choisir de le créer manuellement (en sélectionnant **Create manually**), mais vous devrez alors configurer vous-même votre schéma, vos colonnes sources et vos colonnes cibles. Le fichier sera créé lors de l'utilisation dans un Job d'un composant de sortie XML (**tAdvancedFileOutputXML**).



- Sélectionnez **Create from a file**, afin de créer votre fichier à partir d'un fichier existant.
- Dans le champ **XML or XSD File**, renseignez le chemin d'accès au fichier XML ou XSD à charger, contenant la structure que vous souhaitez appliquer au fichier de sortie.



Si votre fichier est un fichier XSD, il sera stocké dans le **Repository**. Les métadonnées ne seront donc pas affectées par la suppression ou le déplacement du fichier.

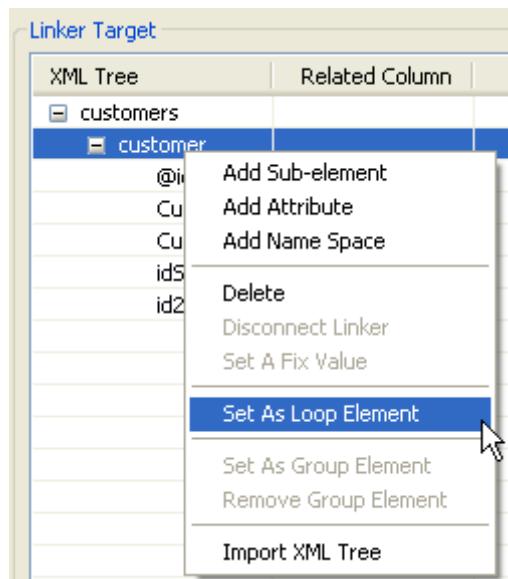
- Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement.
- Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez 0 si vous souhaitez l'effectuer sur toutes les colonnes.
- Dans la zone **Output File Path**, dans le champ **Output file**, renseignez le chemin d'accès à votre fichier de sortie. Si le fichier n'existe pas, il sera créé, lors de l'utilisation, dans un Job, du composant **tAdvancedFileOutputXML**. Si le fichier existe déjà, il sera écrasé.
- L'aperçu de fichier (dans la zone **File Viewer**) affiche la structure XML.

- Cliquez sur **Next** pour définir le schéma.

Etape 4 : Définition du schéma

Dans la zone **Linker Target**, vous devez définir l'élément sur lequel effectuer une boucle. Pour ce faire :

- Cliquez-droit sur l'élément souhaité, puis, dans le menu contextuel, sélectionnez **Set As Loop Element**.



Vous devez ensuite mapper les colonnes selon vos besoins.



Vous pouvez appuyer sur les touches **Ctrl** ou **Maj** si vous souhaitez effectuer les opérations de mapping via la sélection multiple. La sélection multiple fonctionne également pour les opérations disponibles par clic-droit.

Pour ce faire :

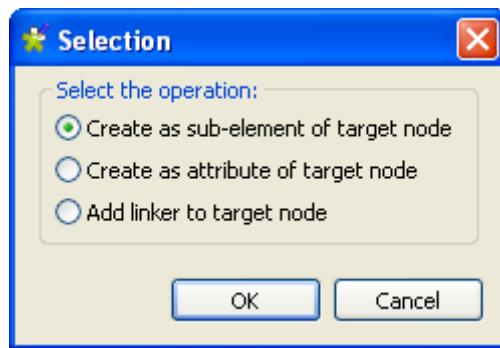
- Glissez la(es) colonne(s) source, dans la zone **Linker Source**, sur la(es) colonne(s) cible, dans la zone **Linker Target**.
Une boîte de dialogue s'ouvre et vous demande de choisir entre trois opérations.
- Sélectionnez **Create as sub-element of target node** si vous souhaitez ajouter un noeud sur un élément. Ce nouveau noeud sera considéré comme fils de l'élément.

Ou,

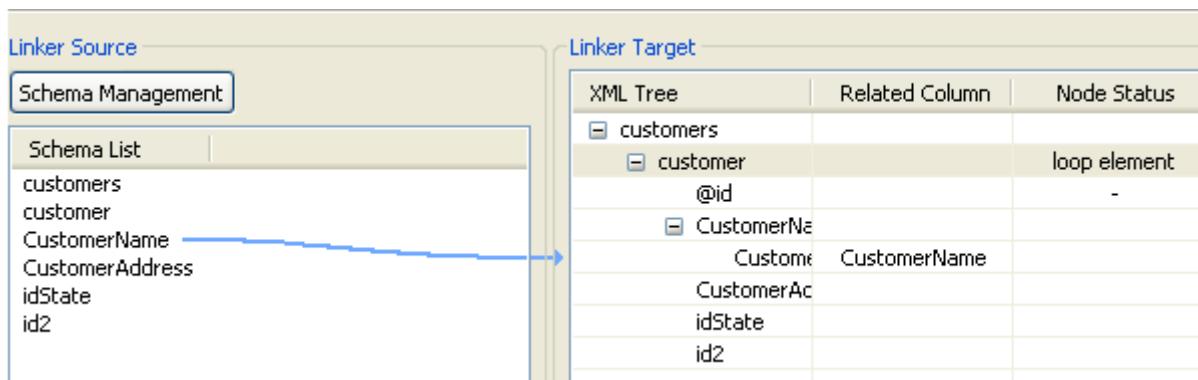
- Sélectionnez **Create as attribute of target node** si vous souhaitez ajouter un noeud sur un élément. Ce noeud sera considéré comme attribut de l'élément.

Ou,

- Sélectionnez **Add linker to target node** si vous souhaitez stocker la valeur de la colonne issue du schéma dans la balise XML correspondante.



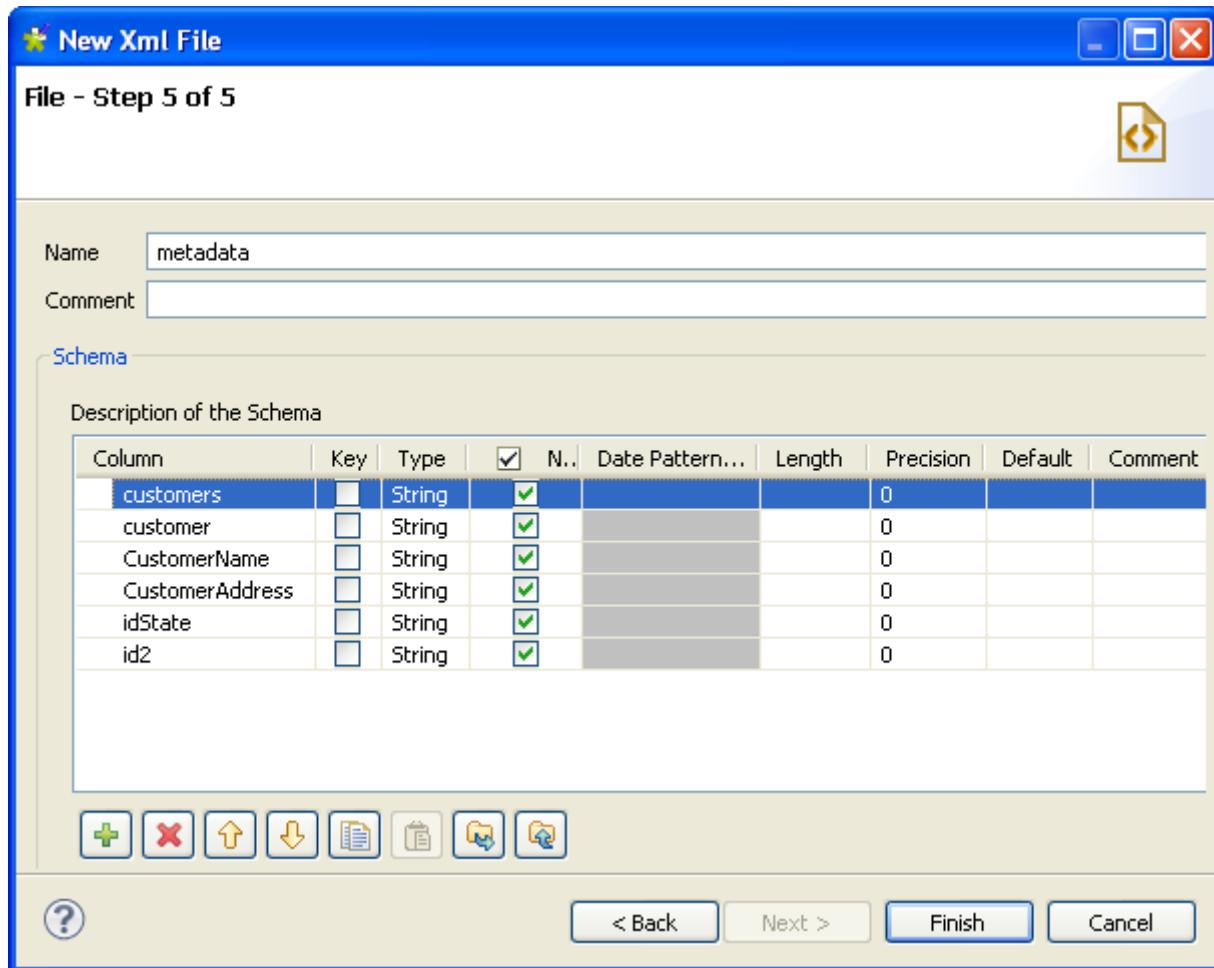
- Dans cet exemple, utilisez l'option **Create as sub-element of target node**. Cliquez sur **OK** pour valider l'opération.
- Une flèche bleue relie alors les deux colonnes.



- Cliquez sur **Next** pour passer à l'étape suivante.

Etape 5 : Schéma final

L'étape 5 affiche le schéma final généré.

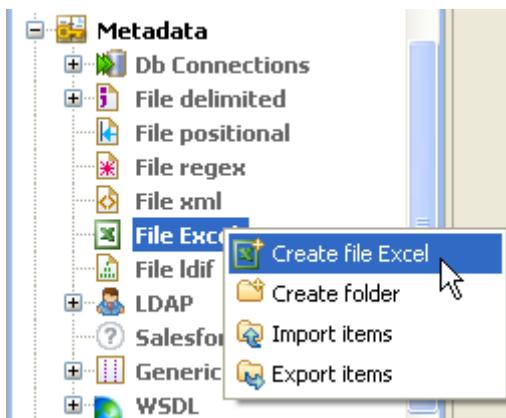


Vous pouvez personnaliser le nom de la métadonnée dans le champ **Name** (par défaut *metadata*), ajouter un commentaire dans le champ **Comment** et apporter des modifications à l'aide de la barre d'outils, par exemple :

- Ajoutez ou supprimez une colonne à l'aide des boutons et .
- Modifiez l'ordre des colonnes à l'aide des boutons et .
- Cliquez sur **Finish** pour terminer la création du fichier XML de sortie.
Le nouveau schéma apparaît dans le **Repository**, sous le noeud **File xml** correspondant.

7.10 Configurer un schéma File Excel

Dans le **Repository**, cliquez droit sur **File Excel** dans l'arborescence, puis sélectionnez **Create file Excel** 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 Excel**.

7.10.1 Etape 1 : Propriétés générales

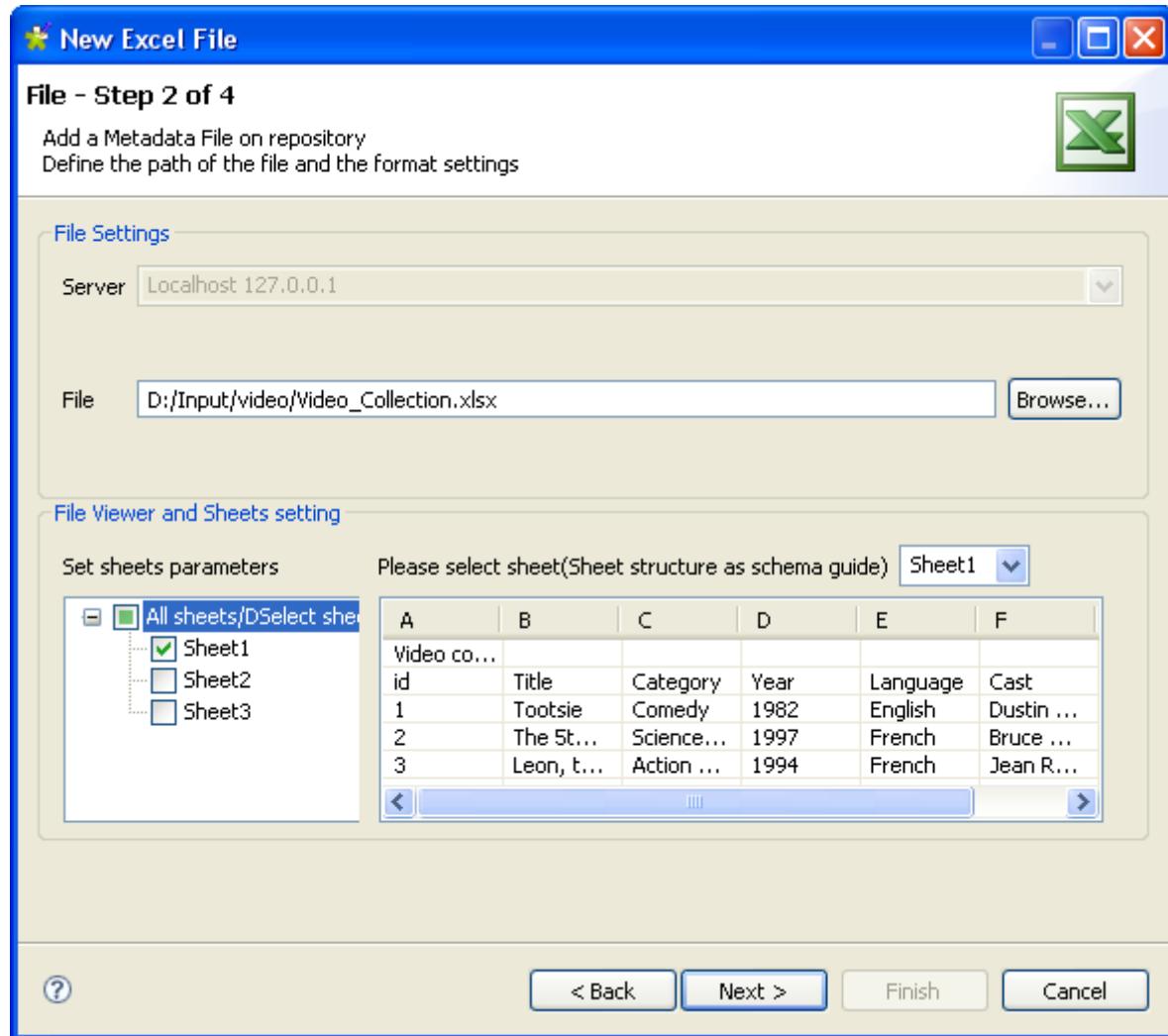
Renseignez les informations générales du schéma, notamment les champs **Name** et **Description**.

7.10.2 Etape 2 : Chargement du fichier

Cliquez sur **Browse** pour définir le chemin d'accès au fichier à charger et renseignez le champ **File**.



Tous les types de fichiers Excel sont pris en charge, notamment les fichiers .xlsx de la version Excel 2007.



La zone **File viewer and sheets settings** donne un instantané du fichier et vous permet de placer les marqueurs de position.

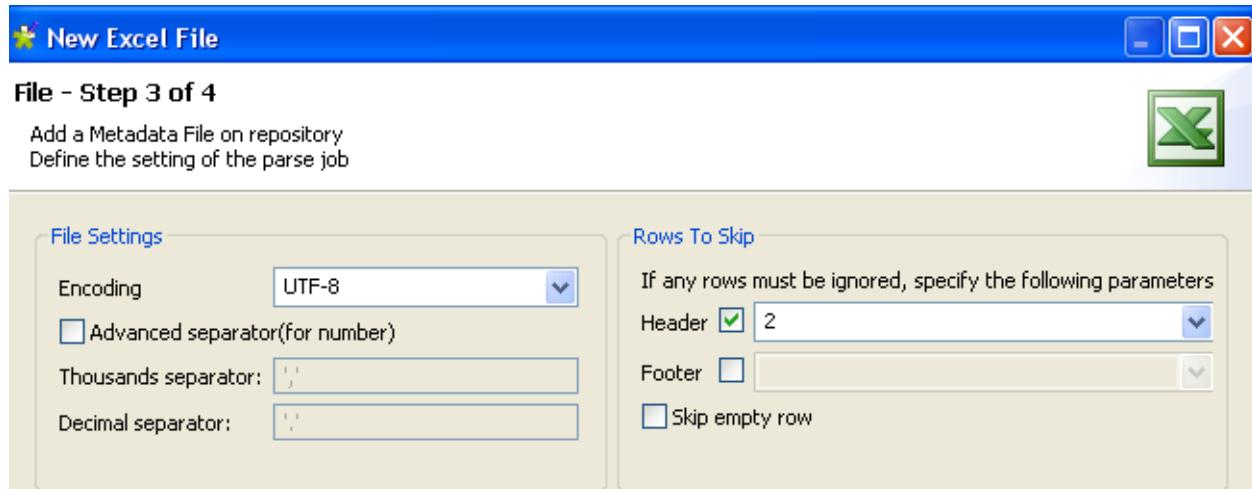
Dans la liste **Set sheets parameters**, cochez la case à gauche de la feuille Excel à charger. Par défaut, l'aperçu affiche le contenu de la première feuille Excel.

Vous pouvez aussi sélectionner une autre feuille dans la liste déroulante et vérifier que le fichier est lu correctement dans la table d'aperçu.

Cliquez sur **Next** pour continuer.

7.10.3 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** et **Row Separator**), le nombre de lignes d'en-tête, etc.

Renseignez les champs **First column** (première colonne) et **Last column** (dernière colonne), pour définir les colonnes devant être lues dans le fichier. Vous pouvez être amené à supprimer la colonne A par exemple, car elle peut contenir des données ne devant pas être traitées.

Cochez aussi la case **Set heading row as column names** pour conserver les libellés des colonnes. N'oubliez pas de cliquez sur le bouton **Refresh** pour avoir un aperçu des modifications apportées.

Puis cliquez sur **Next** pour continuer.

7.10.4 Etape 4 : Schéma final

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

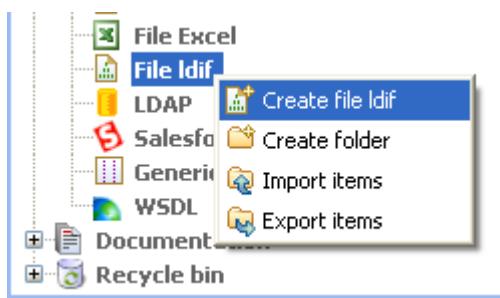
Vous pouvez personnaliser le nom de la métadonnée (par défaut, *metadata*) et apporter des modifications à l'aide de la barre d'outils.

Vous pouvez également restaurer ou mettre à jour le schéma du fichier Excel 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**. Le nouveau schéma apparaît dans le **Repository**, sous le noeud **File Excel** correspondant.

7.11 Configurer un schéma File LDIF

Les fichiers LDIF sont des fichiers annuaires distincts par leurs attributs. Les métadonnées **File LDIF** 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**.



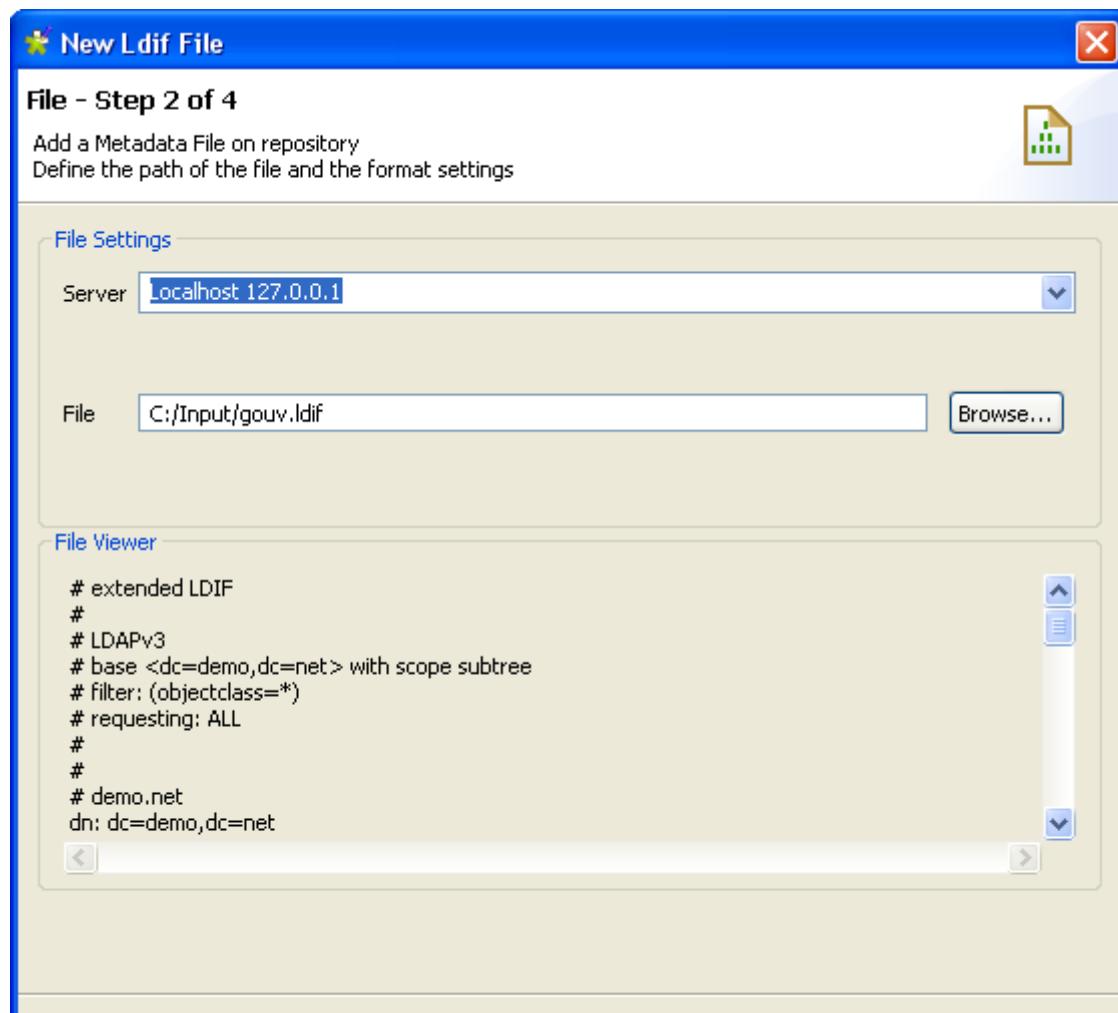
Assurez-vous que les modules requis pour l'utilisation de cette fonctionnalité sont bien installés. Pour plus d'informations, visitez <http://talendforge.org/wiki/doku.php>

7.11.1 Etape 1 : Propriétés générales

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

7.11.2 Etape 2 : Chargement du fichier

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

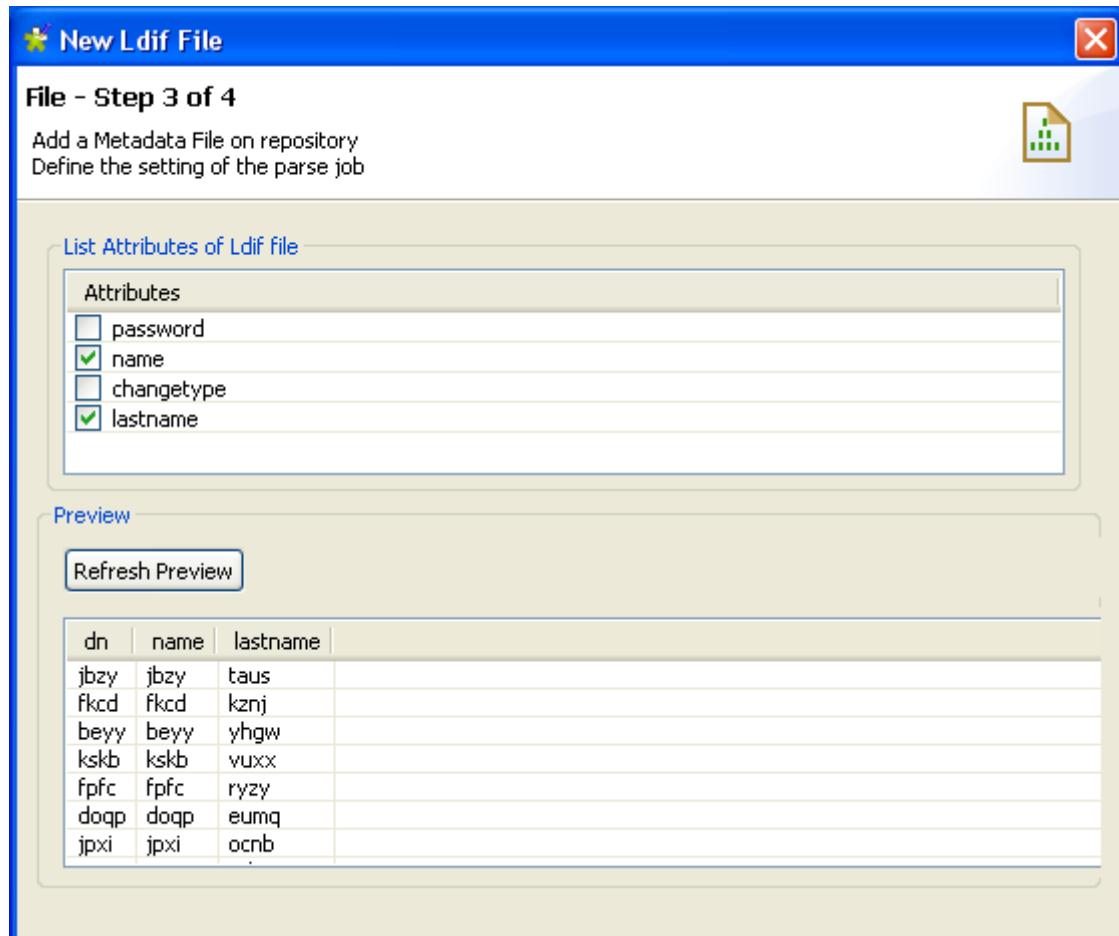


La fonctionnalité de connexion à un serveur distant n'est pour l'instant pas opérationnel pour la récupération de fichier LDIF.

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

7.11.3 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.



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



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.

7.11.4 Etape 4 : Schéma final

Le schéma généré affiche les colonnes du fichier de description. Vous pouvez le personnaliser selon vos besoins 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.

7.12 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.

7.12.1 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 190*.

7.12.2 Etape 2 : Connexion au serveur

Renseignez les informations de connexion.

File - Step 2 of 5

Add a Metadata File on repository
Define the path of the file and the format settings

Network Parameter

Hostname:	Your-LDAP-IP
Port:	389
Encryption method:	LDAP

Click the button to check connection status.

Check Network Parameter

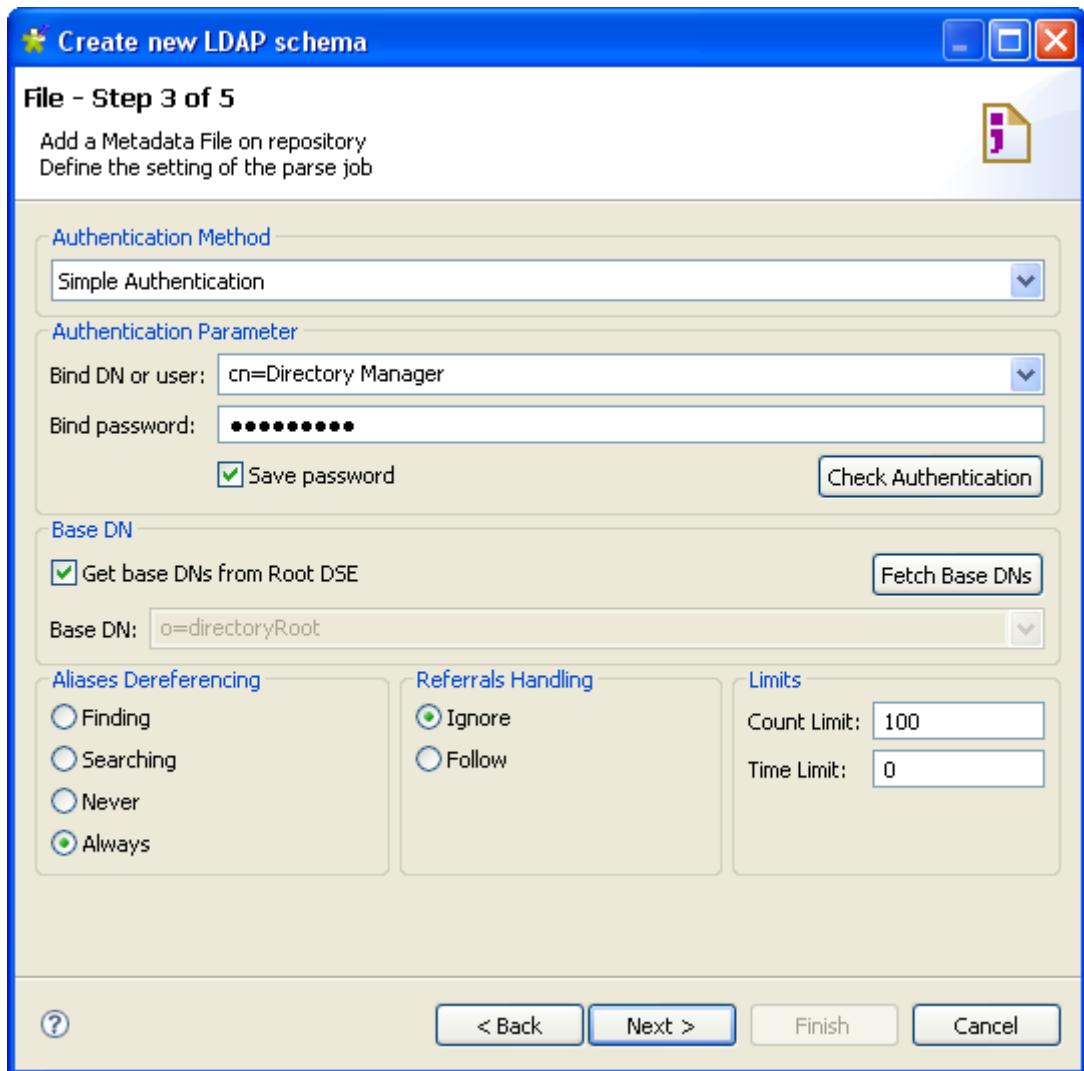
Vérifiez votre connexion à l'aide de bouton **Check Network Parameter**, puis cliquez sur le bouton **Next**.

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.

7.12.3 Etape 3 : Authentification et acquisition du DN

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



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 d'authentification.
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.

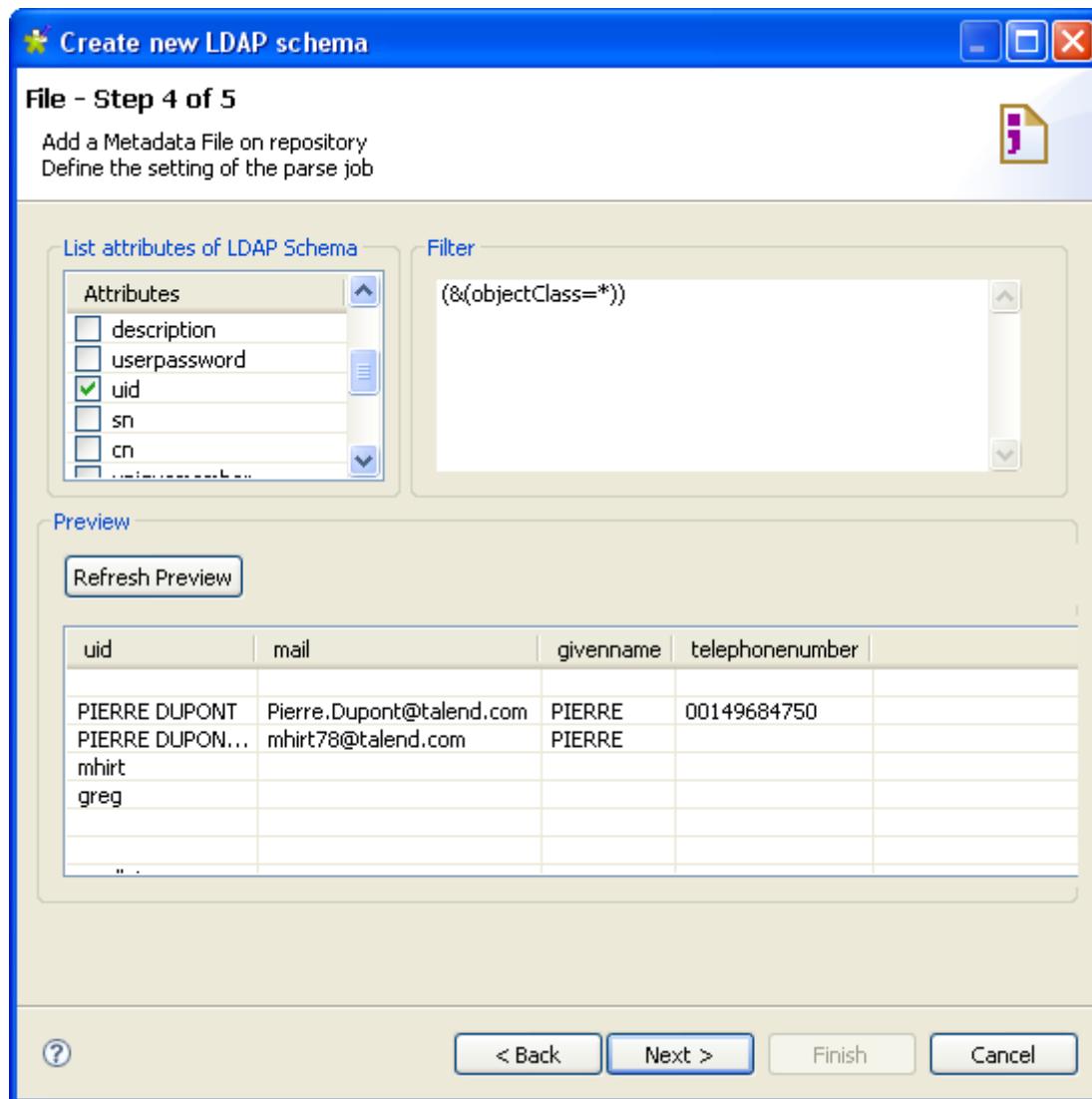
Champ	Description
Alias Dereferencing	<p>Never permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez Always.</p> <p>Always : les alias sont toujours déréférencés.</p> <p>Never : les alias ne jamais sont déréférencés.</p> <p>Searching : cette option déréfère les alias uniquement après la résolution du nom.</p> <p>Finding : cette option déréfère les alias uniquement lors de la résolution du nom.</p>
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.

7.12.4 Etape 4 : Définition du schéma

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

Ajoutez un filtre si vous souhaitez uniquement des données particulières.

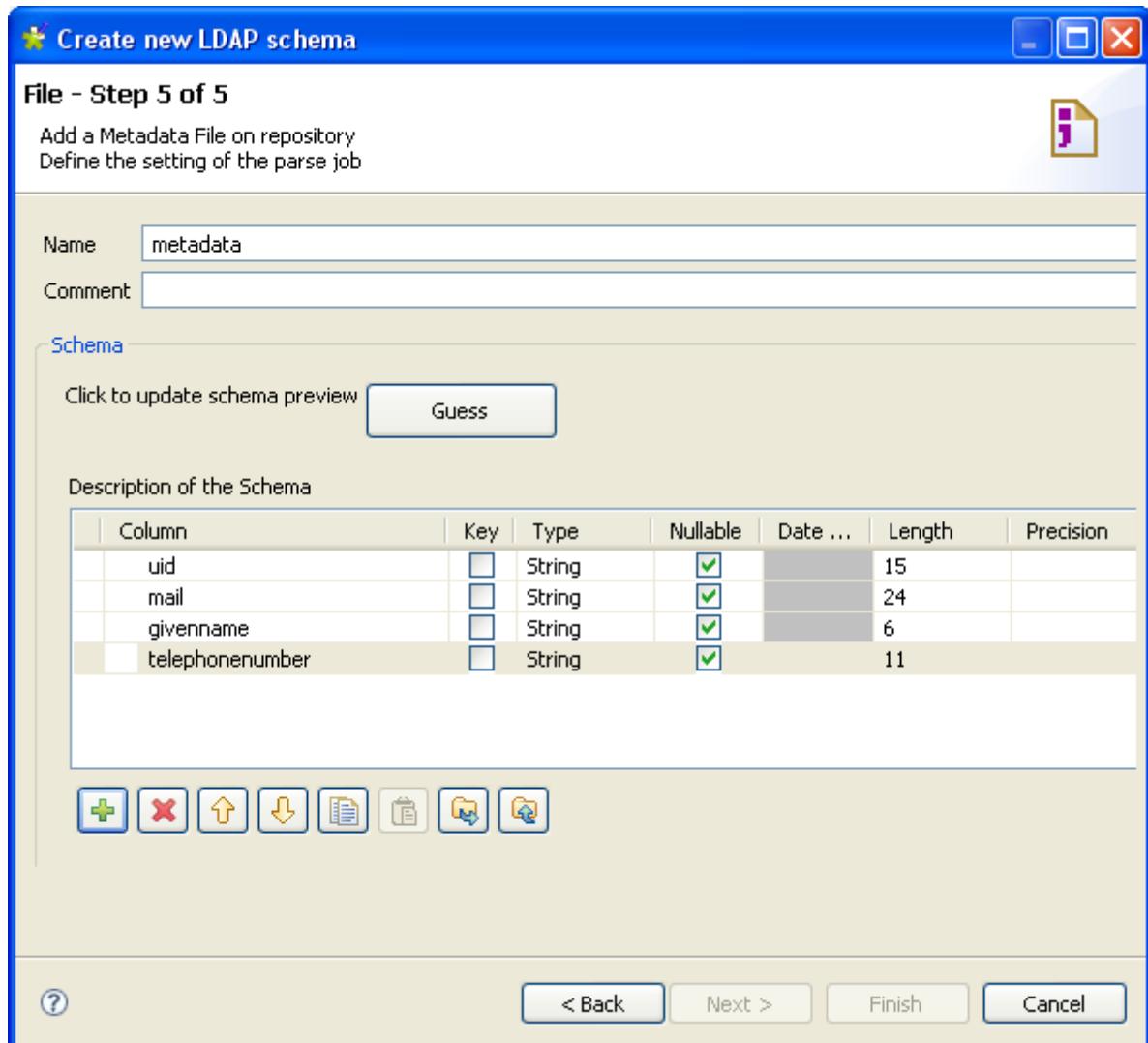


Cliquez sur **Refresh Preview** pour afficher les colonnes sélectionnées ainsi qu'un échantillon des données.

Puis cliquez sur **Next** pour continuer.

7.12.5 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.

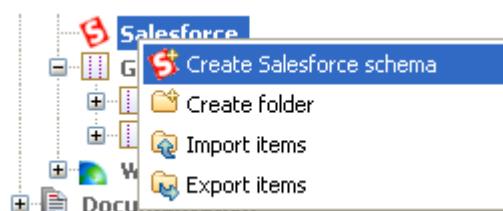


Si l’annuaire LDAP ayant donné lieu au schéma a changé, cliquez sur le bouton **Guess** pour générer le schéma une nouvelle fois. Notez que les modifications que vous avez apportées au schéma seront perdues si vous cliquez sur le bouton **Guess**.

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

7.13 Configurer un schéma Salesforce

Dans le **Repository**, cliquez-droit sur **Salesforce** dans l’arborescence, puis sélectionnez **Create Salesforce schema** dans le menu contextuel.



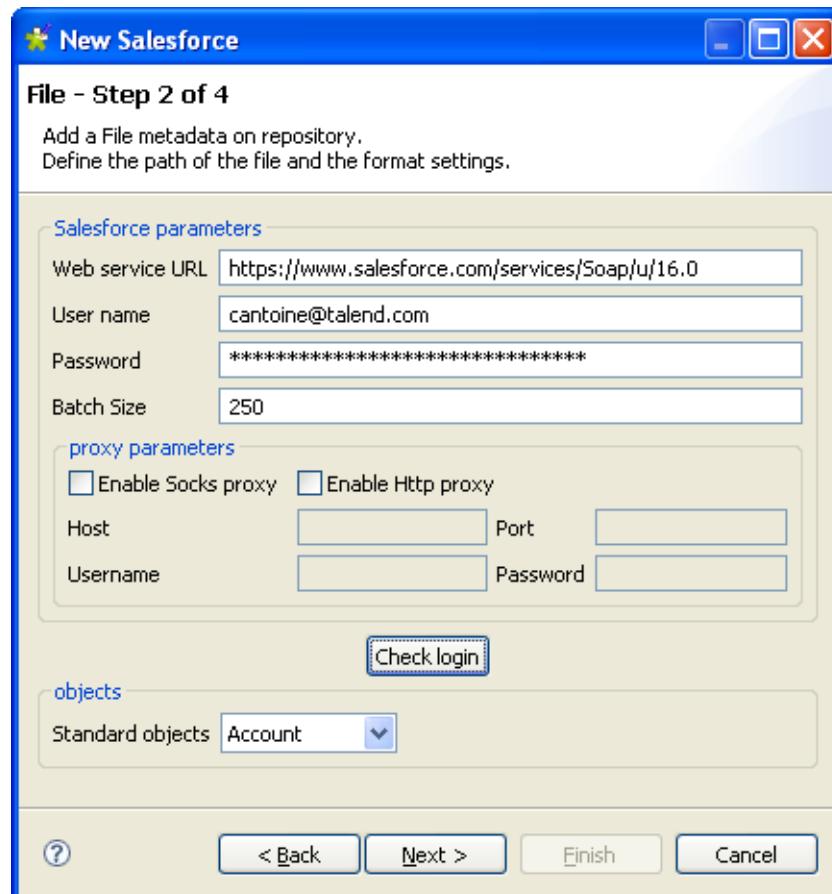
Procédez de la même manière que pour toute autre connexion de métadonnées.

7.13.1 Etape 1 : Propriétés générales

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

7.13.2 Etape 2 : Connexion au compte Salesforce

L'adresse du service Web Salesforce apparaît par défaut dans le champ **Web service URL**.



- Dans les champs **User name** et **Password**, saisissez votre identifiant et mot de passe de connexion au service Web Salesforce.
- Dans la liste **Standard objects**, sélectionnez l'objet standard ou personnalisé auquel vous souhaitez accéder et dont vous souhaitez récupérer le schéma.
- Cliquez sur **Check Login** pour vérifier que votre connexion fonctionne correctement.
- Cliquez sur **Next** pour continuer.

7.13.3 Etape 3 : Affinage du schéma

L'étape suivante ouvre la fenêtre de paramétrage du schéma en fonction de l'objet choisi dans l'étape 2.

New Salesforce

File - Step 3 of 4

Add a File metadata on repository. Define the setting of the parsed job.

Browse data column and set query condition

Query Condition name='United Oil & Gas'

Salesforce Object field detail order the fields

Column	Key	Type	Nullable	Data Pattern	Length	Precision	Default
<code>Id</code>	<code>false</code>		<code>false</code>		18	0	
<code>IsDeleted</code>	<code>false</code>		<code>false</code>		0	0	
<code>MasterRecordId</code>	<code>false</code>		<code>true</code>		18	0	
<code>Name</code>	<code>false</code>		<code>false</code>		255	0	
<code>... -</code>	<code>... -</code>		<code>... -</code>		

Dans la zone **Browse data column and set query condition**, vous pouvez filtrer les données affichées. Pour cela, saisissez la colonne que vous souhaitez filtrer puis la valeur que vous souhaitez mettre en avant dans le champ **Query Condition**.

Les colonnes de la liste **Column** sont triées par ordre alphabétique, si vous ne souhaitez pas trier la liste par ordre alphabétique, décochez la case **order the fields**.

Preview Output

Refresh Preview

AccountNumber	Active__c	AnnualRevenue	BillingCity	BillingCountry	BillingPostalCode	BillingState
CC978213	Yes	3.0E7	Fountain Valley	United States	92708	CA
CD355119-A	Yes	0.0	Mountain View			CA
CD355120-B	Yes	0.0	Singapore			UK
CD451796	Yes	1.39E8	Austin			Singapore
CD656092	No	3.5E8	Burlington	USA	27215	TX
CC213425	Yes	9.5E8	Paris	France	75251	NC
CC634267	Yes	5.0E7	Lawrence	USA	66045	KS
CD439877	Yes	5.0E8	Chicago			IL
CC947211	Yes	9.5E8	Portland			OR
CD736025	Yes	0.0	Tucson			AZ

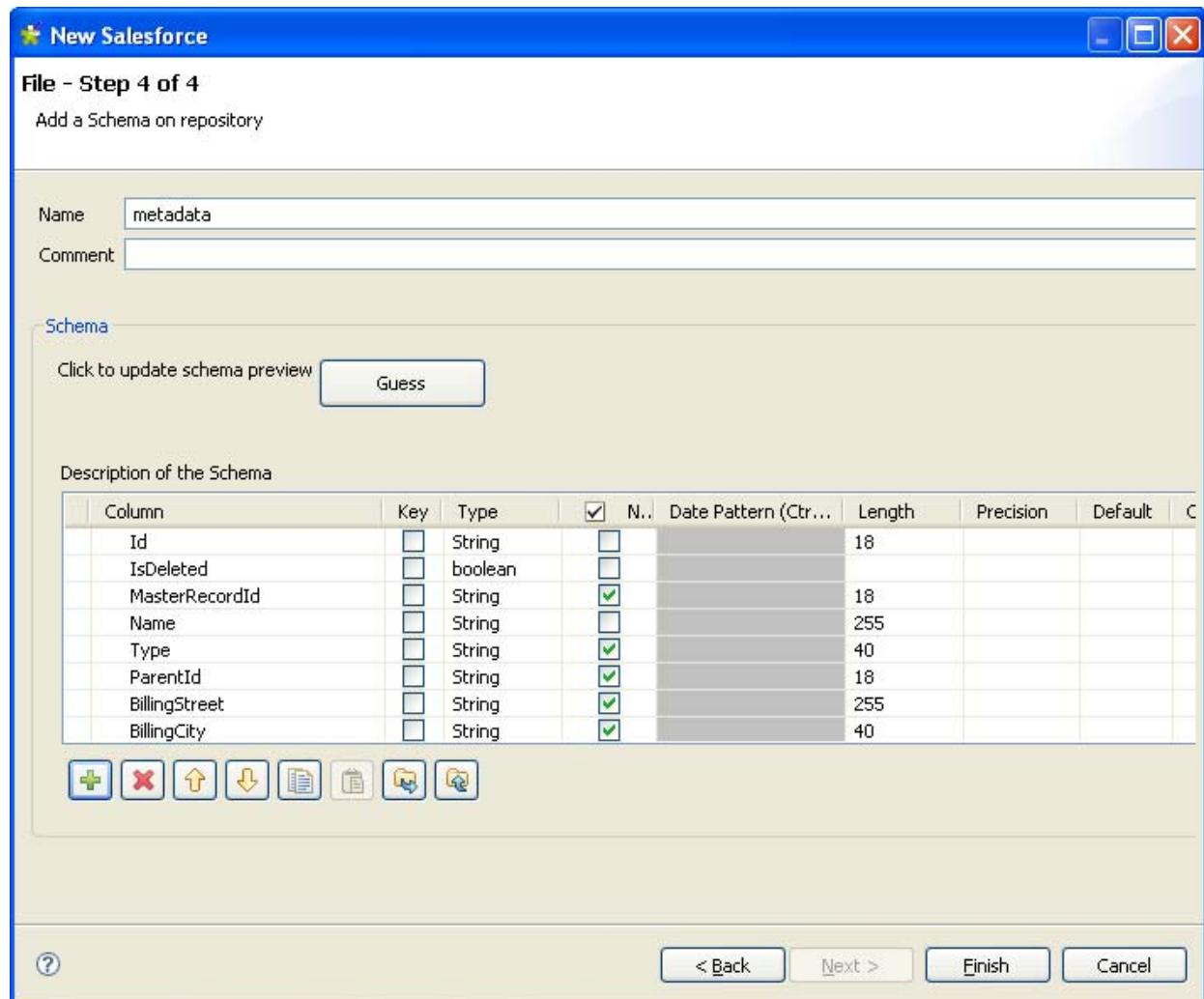
Cliquez sur **Refresh Preview** si vous avez saisi une requête, afin que l'aperçu soit mis à jour. Par défaut, l'aperçu montre toutes les colonnes de l'objet sélectionné.

Puis cliquez sur **Next** pour continuer.

7.13.4 Etape 4 : Schéma final

L'étape 4 affiche le schéma final généré.

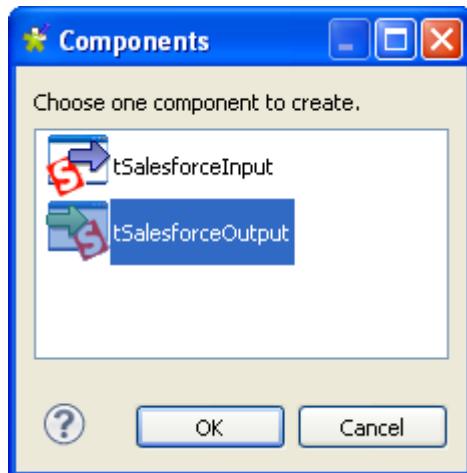
Vous pouvez personnaliser le nom de la métadonnée (par défaut, metadata) et apporter des modifications à l'aide de la barre d'outils.



Vous pouvez également restaurer ou mettre à jour le schéma Salesforce 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**. Le nouveau schéma apparaît dans le **Repository**, sous le noeud **Salesforce** correspondant.

Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir d'utiliser dans votre Job un **tSalesforceInput** ou un **tSalesforceOutput**.



Pour plus d'informations concernant l'action de déposer des métadonnées de composants dans l'espace de modélisation graphique, consultez *Déposer des composants à partir du répertoire Metadata, page 62.*

7.14 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 schéma déjà constitué dans un fichier source. La procédure de création se fait en deux étapes :

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

7.14.1 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.

7.14.2 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.

7.15 Configurer un schéma Web Service

Talend Open Studio vous permet de stocker vos connexions Web Service dans le **Repository**.

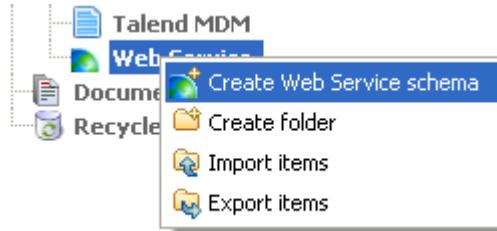
7.15.1 Configurer un schéma simple

Cette section décrit la définition d'un schéma Web Service.

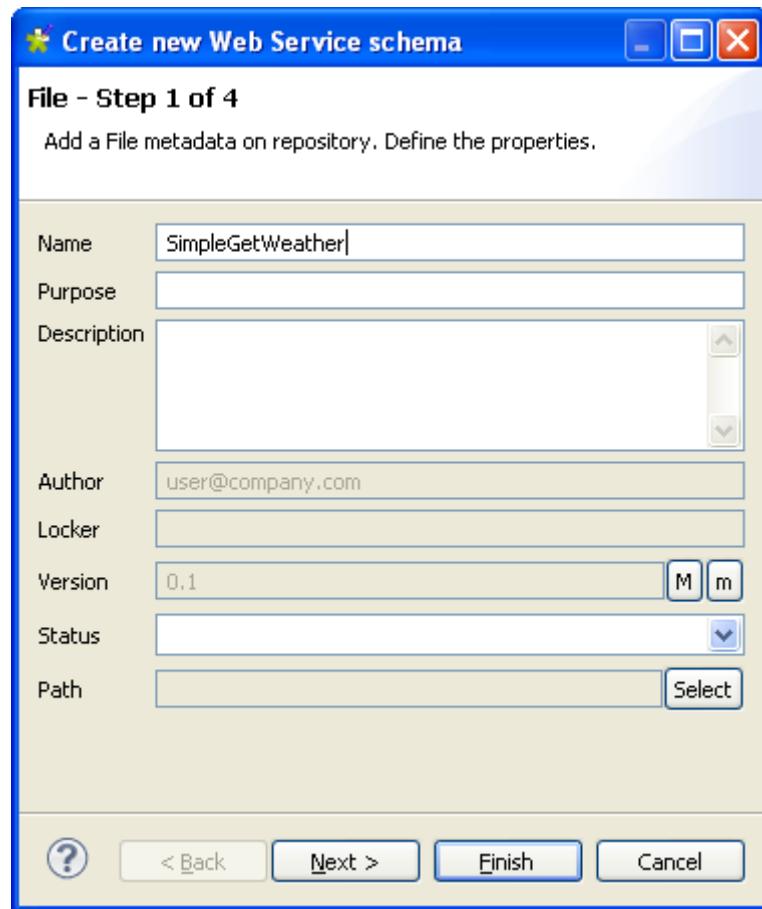
Etape 1 : Propriétés générales

Dans cette étape, vous allez définir les propriétés générales du fichier.

- Dans la vue **Repository**, développez le noeud **Metadata**.
- Cliquez-droit sur **Web Service**, et sélectionnez **Create Web Service schema** dans le menu contextuel.



- Remplissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.



- Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

Etape 2 : URI et définition des méthodes

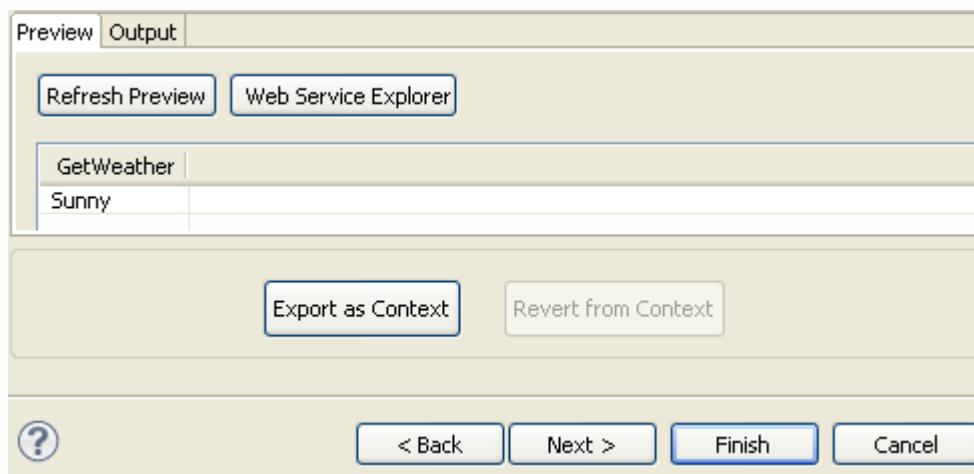
Dans cette étape, définissez l'URI et les paramètres nécessaires pour obtenir les valeurs.

Web Service Parameter

Web Service	<input type="text" value="http://www.deeptraining.com/webservices/weather.asmx?WSDL"/>					
<input type="checkbox"/> Need authentication?						
User	<input type="text"/>	Password <input type="text"/>				
<input type="checkbox"/> Use http proxy						
host	<input type="text"/>	Port <input type="text"/>				
user	<input type="text"/>	password <input type="text"/>				
Method	<input type="text" value="GetWeather"/>					
Time Out(Second)	<input type="text" value="20"/>					
Parameters						
<table border="1"> <tr> <td>Value</td> <td> </td> </tr> <tr> <td colspan="2">Melbourne</td> </tr> </table>			Value		Melbourne	
Value						
Melbourne						
<input type="button" value="Add"/> <input type="button" value="Remove"/>						

Dans la zone **Web Service Parameter** :

- Dans le champ **Web Service**, renseignez l'URI qui transmettra les valeurs souhaitées.
- Si nécessaire, cochez la case **Need authentication?** puis saisissez respectivement votre identifiant et votre mot de passe, dans les champs **User** et **Password**.
- Si vous utilisez un proxy http, cochez la case **Use http proxy**, et renseignez les propriétés (respectivement l'hôte, le port, l'identifiant et le mot de passe).
- Dans le champ **Method**, définissez la méthode en la nommant.
- Dans le tableau **Value**, ajoutez ou supprimez autant de valeurs que vous souhaitez, en cliquant sur le bouton **Add** (Ajouter) ou **Remove** (Supprimer).
- Puis cliquez sur **Refresh Preview** pour vérifier que les paramètres sont corrects.



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

Etape 3 : Schéma final

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

Add a File metadata on repository. Define the setting of the parsed job.

Name: metadata

Comment:

Schema

Click to update schema preview **Guess**

Description of the Schema

Column	Key	Type	Nullable	Date Pattern ...	Length	Precision	Def...	Comment
GetWeather	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		5	0		

Actions:

Buttons:

- Ajoutez ou supprimez une colonne à l'aide des boutons et .
 - Modifiez l'ordre des colonnes à l'aide des boutons et .
 - Cliquez sur **Finish**.
- Le nouveau schéma s'affiche dans le **Repository** sous le noeud **Web Service** approprié.

7.16 Configurer une connexion MDM

Talend Open Studio vous permet de centraliser les informations d'une ou plusieurs connexion(s) MDM sous le noeud **Metadata** de la vue **Repository**. Vous pouvez utiliser l'une de ces connexions établies pour vous connecter au serveur MDM.



Vous pouvez également paramétriser une connexion MDM en cliquant sur l'icône de la vue **Basic settings** des composants **tMDMInput** et **tMDMOutput**. Pour plus d'informations, consultez le chapitre *Composants Talend MDM* du **Guide de Référence Composants Talend Open Studio**.

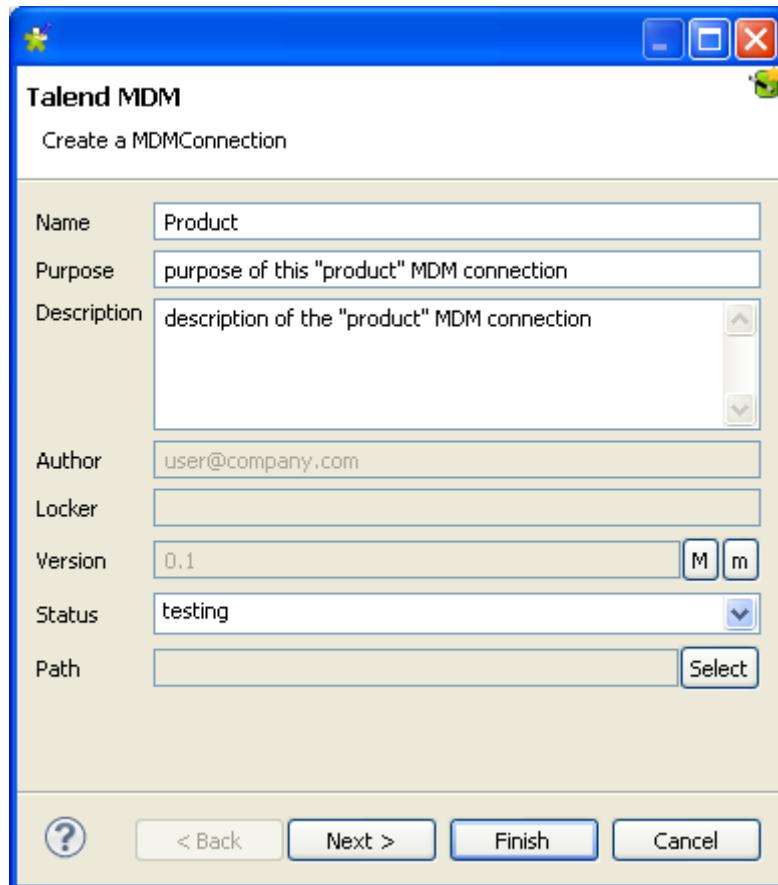
Selon l'option que vous sélectionnez, l'assistant vous permet de créer soit un schéma **XML Input** (d'entrée), soit un schéma **XML Output** (de sortie), soit un schéma **XML Receive** (de réception). Plus tard, dans un Job **Talend**, le composant **tMDMInput** utilise le schéma d'entrée défini pour lire des données maître stockées dans des documents XML. Le **tMDMOutput** utilise le schéma de sortie défini,

soit pour écrire des données maître dans un document XML sur le serveur MDM, soit pour mettre à jour des documents XML existants. Le **tMDMReceive** utilise le schéma XML défini pour recevoir, des déclencheurs et des processus MDM, un enregistrement MDM au format XML

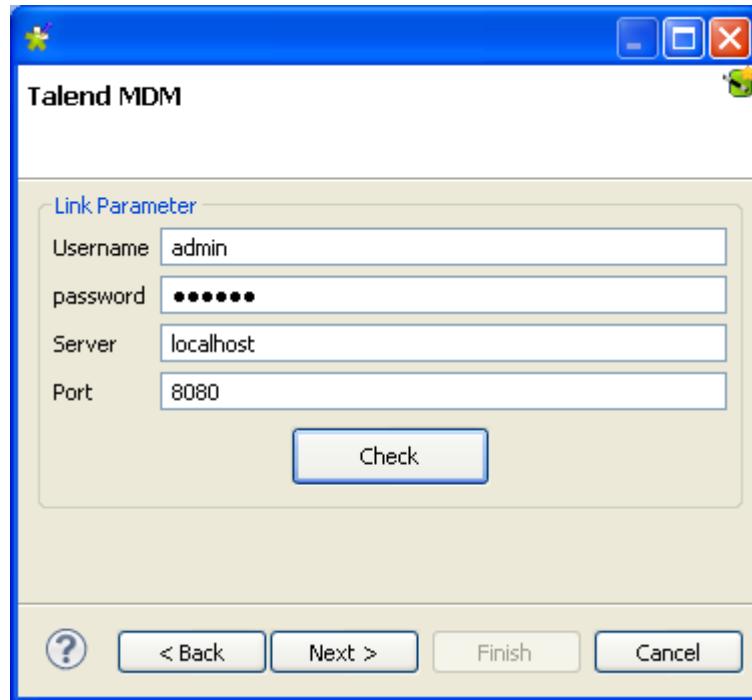
7.16.1 Etape 1 : Définir la connexion

Pour établir une connexion MDM, procédez comme suit :

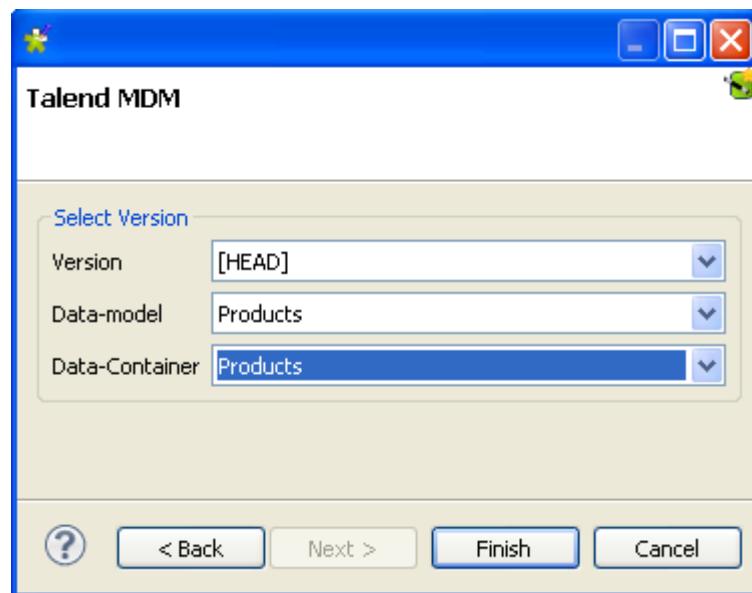
- Dans la vue **Repository**, développez le noeud **Metadata** et cliquez-droit sur **Talend MDM**.
- Sélectionnez **Create MDM** dans le menu contextuel.
L'assistant de connexion s'affiche.



- Remplissez les champs **Name**, **Purpose** et **Description**. Le champ **Status** est un champ personnalisé pouvant être défini. Pour plus d'informations, consultez *Définir les paramètres des Statuts, page 40*.
- Cliquez sur **Next** afin de passer à l'étape suivante.



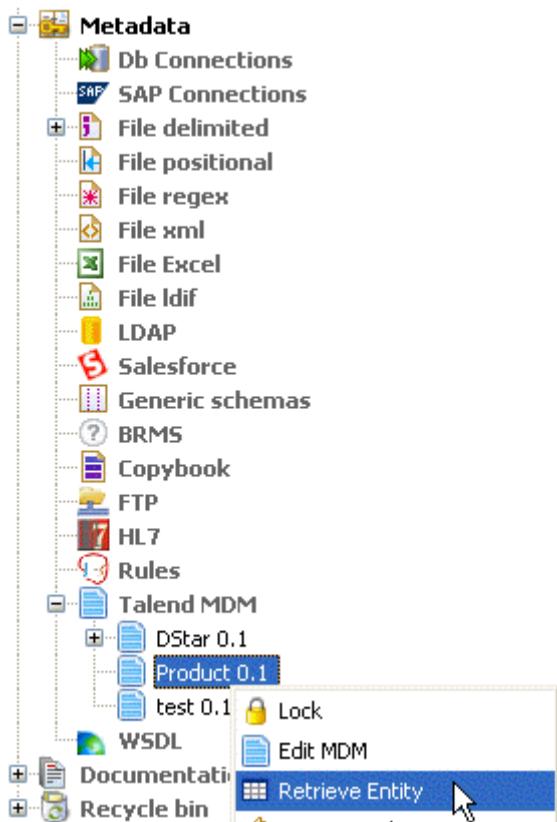
- Remplissez les informations de connexion au serveur MDM, y compris les informations d'authentification, puis cliquez sur **Check** afin de vérifier la connexion que vous avez créée. Une boîte de dialogue s'ouvre pour confirmer que votre connexion a été créée avec succès.
- Cliquez sur **OK** afin de fermer la boîte de dialogue de confirmation, puis cliquez sur **Next** afin de passer à l'étape suivante.



- Dans la liste **Version**, sélectionnez la version des données maître du serveur MDM auquel vous souhaitez vous connecter.
- Dans la liste **Data-Model**, sélectionnez le modèle de données par rapport auquel vous souhaitez valider les données maître.
- Dans la liste **Data-Container**, sélectionnez le conteneur de données où sont stockées les données maître auxquelles vous souhaitez accéder.

- Cliquez sur **Finish** afin de valider vos changements et fermer la boîte de dialogue.

La connexion nouvellement créée s'affiche sous le noeud **Talend MDM** dans le dossier **Metadata** dans la vue **Repository**.



Vous devez maintenant récupérer le schéma XML des entités métier liées à cette connexion MDM.

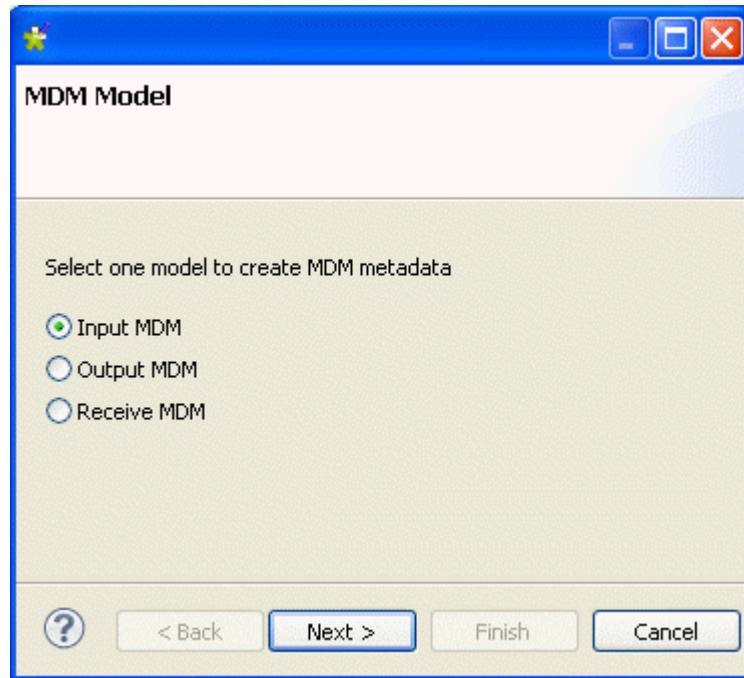
7.16.2 Etape 2 : Définir un schéma MDM

Définir un schéma d'entrée MDM

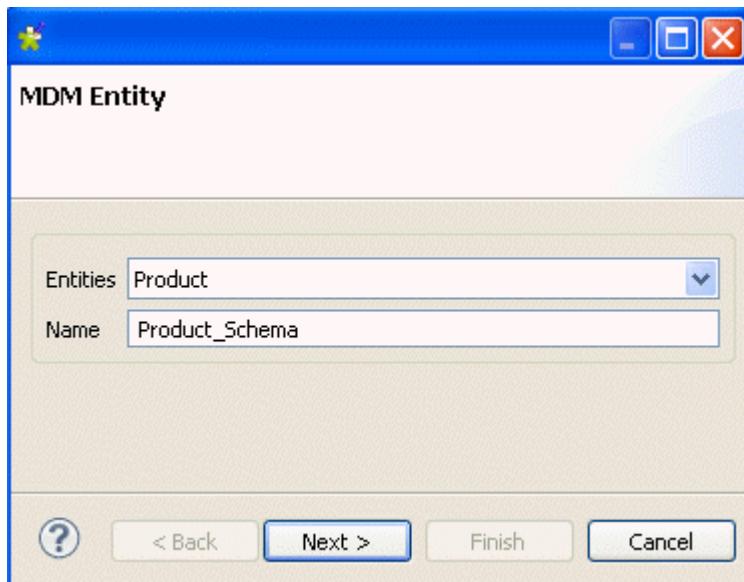
Cette section décrit la définition et le chargement d'un schéma MDM XML d'entrée. Pour définir et charger un schéma MDM XML de sortie, consultez *Définir un schéma de sortie MDM*, page 253.

Pour définir les valeurs à retrouver à partir d'une ou plusieurs entité(s) liée(s) à une connexion MDM procédez comme suit :

- Dans la vue **Repository**, développez le noeud **Metadata** puis cliquez-droit sur la connexion MDM pour laquelle vous souhaitez récupérer les valeurs de l'entité/des entités.
- Sélectionnez **Retrieve Entity** dans le menu contextuel.
Une boîte de dialogue s'affiche.



- Sélectionnez l'option Input MDM afin de charger un schéma XML d'entrée, puis cliquez sur **Next** afin de passer à l'étape suivante.

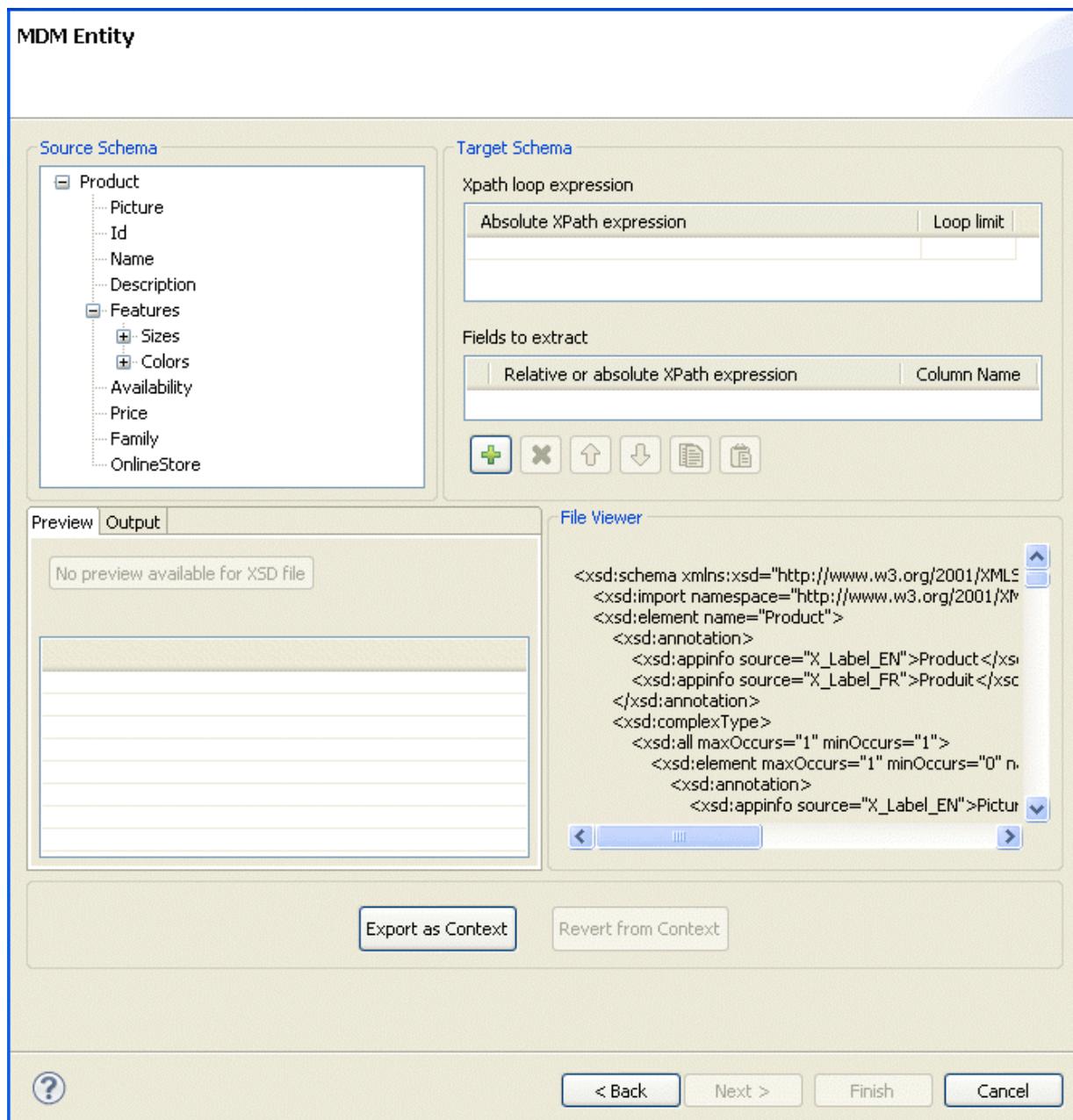


- Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) à partir de laquelle vous souhaitez récupérer les valeurs.
Le nom s'affiche automatiquement dans le champ **Name**.



Vous êtes libre de saisir le nom que vous voulez dans ce champ. Cependant, il est conseillé de saisir le nom de l'entité pour laquelle vous récupérez le schéma.

- Cliquez sur **Next** pour passer à l'étape suivante.



Le schéma de l'entité que vous avez sélectionnée, s'affiche dans le panneau **Source Schema**.

Dans la boîte de dialogue ouverte, vous pouvez configurer les paramètres à prendre en compte lors de la définition du schéma XML.

La boîte de dialogue relative au schéma est divisée en quatre panneaux :

Panneau	Description
Source Schema	Arborescence de l'entité chargée.
Target schema	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible.
File viewer	Visionneur des données brutes.

- Dans la zone **Xpath loop expression**, saisissez l'expression absolue du chemin xpath vers le noeud de la structure XML sur lequel appliquer l'itération. Vous pouvez également glisser le noeud du schéma source dans le champ Xpath du schéma cible. Le lien s'affiche en orange.



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

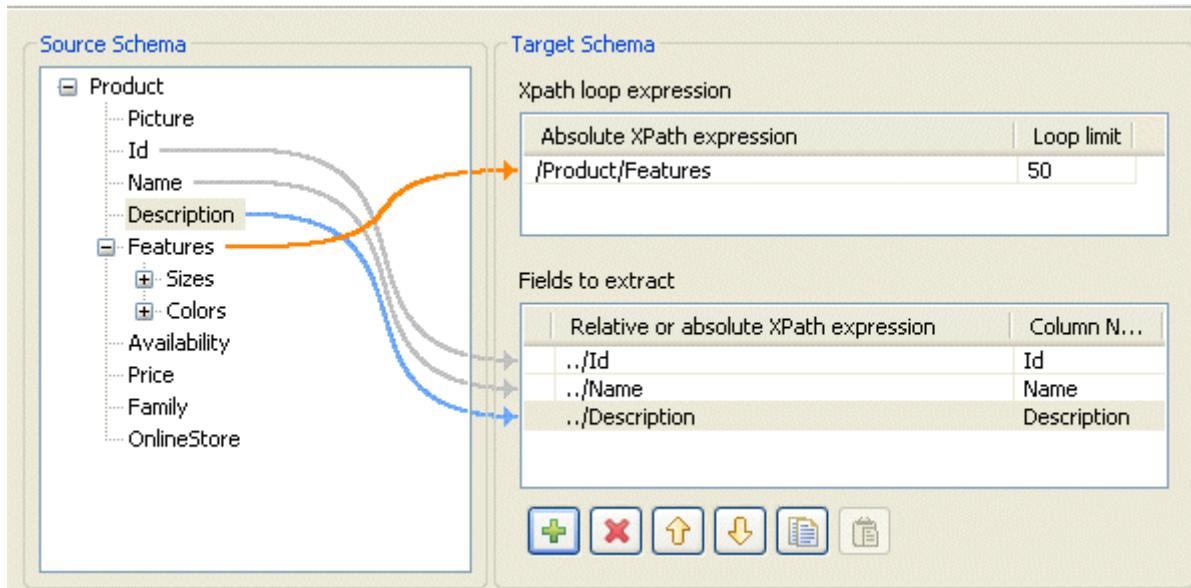
- Au besoin, définissez une limite (**Loop limit**) afin de restreindre l'itération à un certain nombre de noeuds.

Dans la capture d'écran ci-dessus, utilisez *Features* comme élément sur lequel effectuer une boucle, puisqu'il est répété dans l'entité *Product* :

```
<Product>
    <Id>1</Id>
    <Name>Cup</Name>
    <Description/>
    <Features>
        <Feature>Color red</Feature>
        <Feature>Size maxi</Feature>
    <Features>
        ...
    </Product>
<Product>
    <Id>2</Id>
    <Name>Cup</Name>
    <Description/>
    <Features>
        <Feature>Color blue</Feature>
        <Feature>Thermos</Feature>
    <Features>
        ...
    </Product>
```

Ce faisant, le composant **tMDMInput** utilisant la connexion MDM va créer une nouvelle ligne pour chaque élément Feature rencontré.

- Afin de définir les champs à extraire, glissez le noeud correspondant du schéma source dans le champ **Relative or absolute XPath expression**.



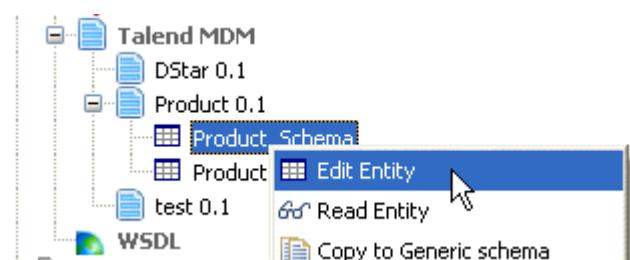
 Cliquez sur le bouton [+] afin d'ajouter des lignes à la table, puis sélectionnez les champs à extraire. Appuyez sur la touche **Ctrl** ou **Maj** pour sélectionner plusieurs noeuds, groupés ou séparés, et glissez-les dans la table.

- Au besoin, saisissez un nom pour chaque colonne récupérée, dans le champ **Column name**.

 Vous pouvez prioriser l'ordre des champs à extraire, en sélectionnant le champ et en utilisant les flèches montante et descendante. Le lien du champ sélectionné s'affiche en bleu, et tous les autres en gris.

- Cliquez sur **Finish** afin de valider vos modifications et fermer la boîte de dialogue.

Le schéma nouvellement créé s'affiche sous le noeud **Talend MDM** correspondant dans la vue **Repository**.



Pour modifier le schéma créé, procédez comme suit :

- Dans la vue **Repository**, développez les noeuds **Metadata** puis **Talend MDM** et parcourez votre répertoire jusqu'au schéma que vous souhaitez modifier.
- Cliquez-droit sur le nom du schéma puis sélectionnez **Edit Entity** dans le menu contextuel.
Une boîte de dialogue s'ouvre.

Name

Comment

Schema

Click Guess button to update the schema below according to your settings

Description of the Schema

Column	Key	Type	✓ N.	Date Pattern (Ctrl...)	Length	Precision	Default	Comment
Id	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Name	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			

- Modifiez le schéma comme vous le souhaitez. Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.
- Cliquez sur **Finish** afin de fermer la boîte de dialogue.

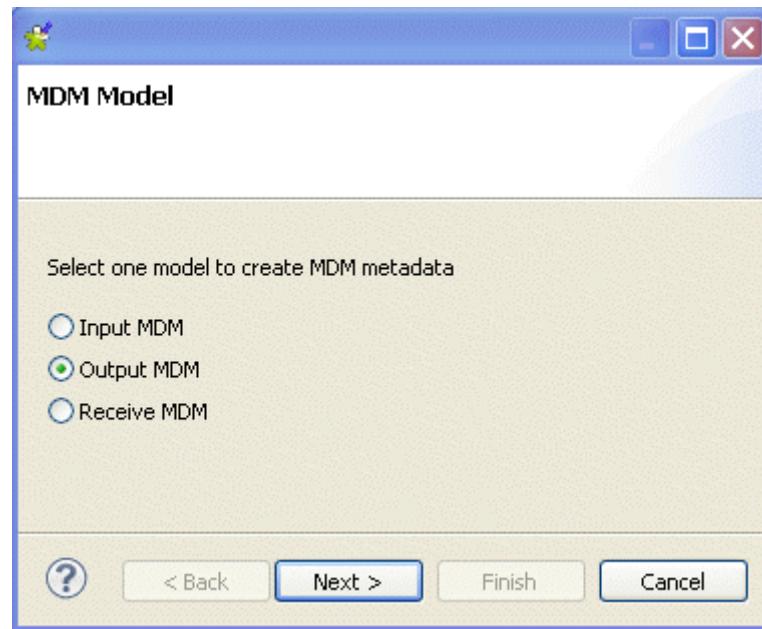
La connexion MDM d'entrée (**tMDMInput**) est maintenant prête à être déposée dans l'un de vos Jobs.

Définir un schéma de sortie MDM

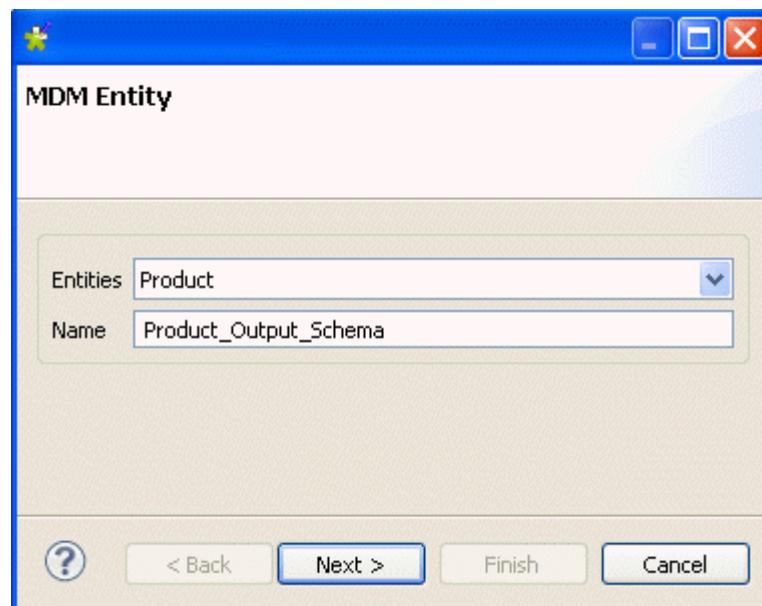
Cette section décrit la définition et le chargement d'un schéma XML MDM de sortie. Pour définir et charger un schéma XML MDM d'entrée, consultez *Définir un schéma d'entrée MDM, page 248*.

Pour définir les valeurs à écrire dans une ou plusieurs entité(s) liée(s) à une connexion MDM spécifique, procédez comme suit :

- Dans la vue **Repository**, développez le noeud **Metadata** et cliquez-droit sur la connexion MDM pour laquelle vous souhaitez écrire les valeurs de l'entité.
- Sélectionnez **Retrieve Entity** dans le menu contextuel.
Une boîte de dialogue s'affiche.



- Sélectionnez l'option **Output MDM** afin de définir un schéma XML de sortie, puis cliquez sur **Next** pour procéder à l'étape suivante.



- Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) dans laquelle vous souhaitez écrire les valeurs.
Le nom s'affiche automatiquement dans le champ **Name**.



Vous êtes libre de saisir le nom que vous voulez dans ce champ. Cependant, il est conseillé de saisir le nom de l'entité pour laquelle vous récupérez le schéma.

- Cliquez sur **Next** pour procéder à l'étape suivante.

MDM Entity

Linker Source

Schema Management

Schema List

Linker Target

XML Tree	Related Column	Node Status	Default Value
Product			
Picture			
Id			
Name			
Description			
Features			
Sizes			
Size			
Colors			
Color			
Availability			
Price			
Family			
OnlineStore			

Buttons: +, -, ↑, ↓

Buttons at the bottom: ? (Help), < Back, Next >, Finish, Cancel



Le schéma de l'entité sélectionnée s'affiche dans le panneau **Linker Target**.

Vous pouvez définir les paramètres à prendre en compte lors de la définition du schéma XML.

- Cliquez sur **Schema Management** pour ouvrir une boîte de dialogue.
- Cliquez sur le bouton **[+]** pour ajouter des lignes et définissez le schéma XML que vous souhaitez écrire dans l'entité sélectionnée.

Schema

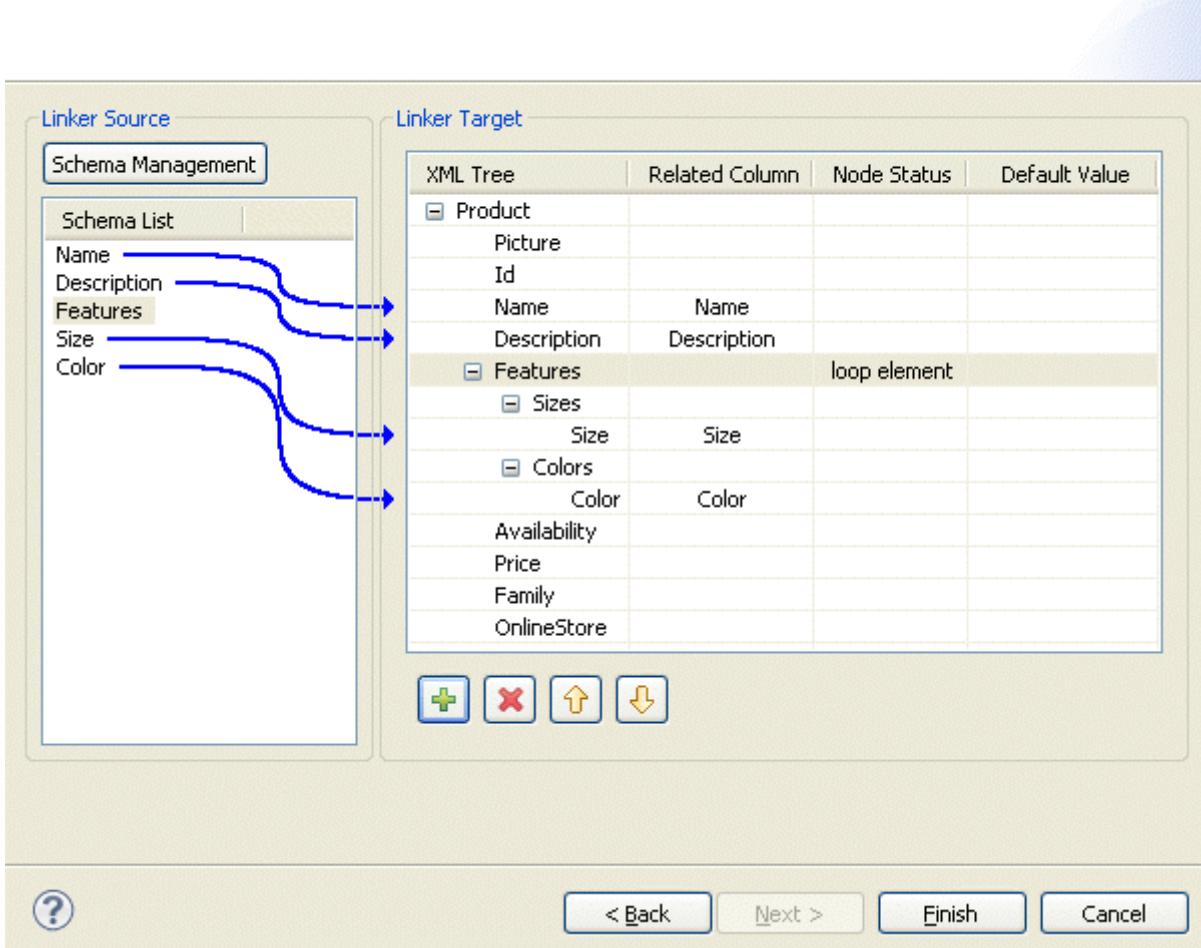
Column	Key	Type		N..	Date P...	Length	Precision	D.
Name	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				0	
Description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				0	
Features	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				0	
Size	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				0	
Color	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>				0	

Buttons at the bottom: +, -, ↑, ↓, OK, Cancel

Votre schéma dans la zone **Linker Source** doit correspondre au schéma dans la zone **Linker Target**, vous devez donc définir les éléments dans lesquels vous souhaitez écrire des valeurs.

- Cliquez sur **OK** pour fermer la boîte de dialogue.
Le schéma défini s'affiche dans la colonne **Schema list**.

MDM Entity



- Pour définir les champs à écrire, déposez les noeuds correspondants du schéma source vers ceux du schéma cible.



Vous pouvez appuyer sur les touches **Ctrl** ou **Maj** si vous souhaitez effectuer les opérations de mapping via la sélection multiple. La sélection multiple fonctionne également pour les opérations disponibles par clic-droit.

- Dans le panneau **Linker Target**, cliquez-droit sur l'élément que vous souhaitez définir comme élément de boucle et sélectionnez **Set as loop element**. Cela limitera l'itération à un ou plusieurs noeud(s).

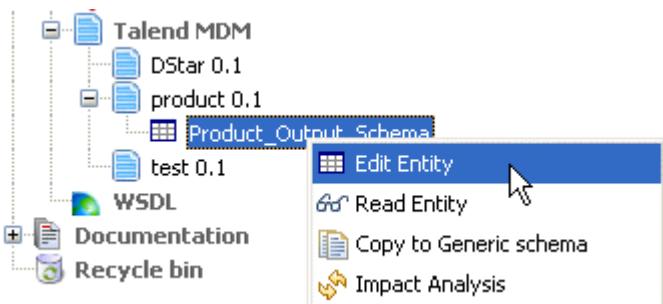
Ce faisant, le composant **tMDMOutput** utilisant cette connexion MDM créera une nouvelle ligne pour chaque élément ayant une “feature” différente.



Vous pouvez prioriser l'ordre des champs à écrire, en sélectionnant le champ et en utilisant les flèches montante et descendante.

- Cliquez sur **Finish** afin de valider vos modifications et fermer la boîte de dialogue.

Le schéma nouvellement créé s'affiche sous la connexion MDM correspondante dans la vue **Repository**.



Pour modifier le schéma créé, procédez comme suit :

- Dans la vue **Repository**, développez les noeuds **Metadata** et **Talend MDM** puis parcourez le référentiel jusqu'au schéma que vous souhaitez modifier.
- Cliquez-droit sur le nom du schéma et sélectionnez **Edit Entity** dans le menu contextuel. Une boîte de dialogue s'affiche.

Column	Key	Type	Null...	Date Pattern (Ctr...)	Length	Precision	Default	Comment
Name	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Features	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Size	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Color	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			

Buttons at the bottom: ? (Help), Finish (blue button), Cancel.

- Modifiez le schéma comme vous le souhaitez. Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.
- Cliquez sur **Finish** afin de fermer la boîte de dialogue.

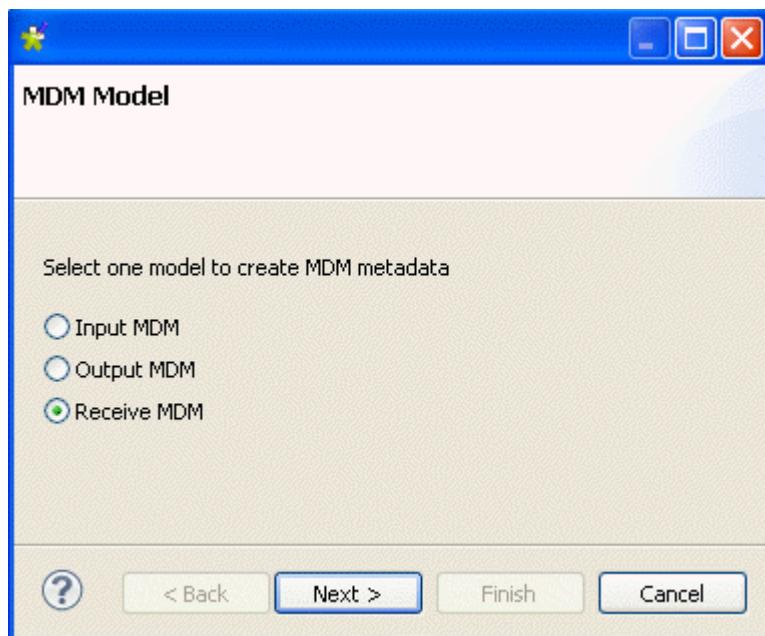
La connexion MDM de sortie (**tMDMOutput**) est maintenant prête à être déposée dans l'un de vos Jobs.

Définir un schéma de réception MDM

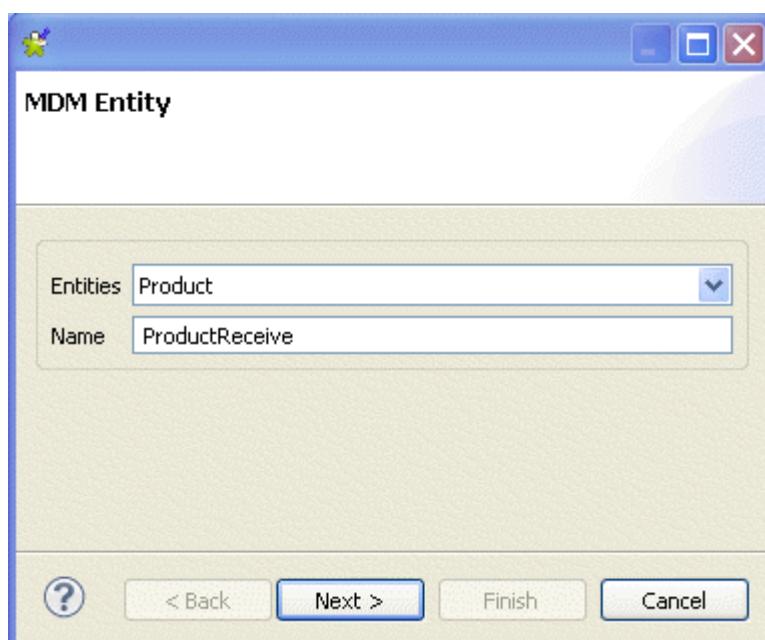
Cette section décrit la définition d'un schéma MDM XML de réception basé sur une connexion MDM.

Pour définir le schéma XML que vous souhaitez recevoir selon une connexion MDM spécifique, procédez comme suit :

- Dans la vue **Repository**, développez le noeud **Metadata** puis cliquez-droit sur la connexion MDM pour laquelle vous souhaitez récupérer les valeurs de l'entité.
- Sélectionnez **Retrieve Entity** dans le menu contextuel.
Une boîte de dialogue s'ouvre.



- Sélectionnez l'option **Receive MDM** afin de définir un schéma XML de réception puis cliquez sur **Next** pour passer à l'étape suivante.



- Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) selon laquelle vous souhaitez recevoir le schéma XML.
Le nom s'affiche automatiquement dans le champ **Name**.



Vous pouvez saisir tout texte dans ce champ, même s'il est recommandé de saisir le nom de l'entité selon laquelle vous souhaitez recevoir le schéma XML.

- Cliquez sur **Next** pour passer à l'étape suivante.

MDM Entity

Source Schema

```

Product
  - Picture
  - Id
  - Name
  - Description
Features
  + Sizes
  + Colors
  Availability
  Price
  Family
  OnlineStore

```

Target Schema

Xpath loop expression

Absolute XPath expression | Loop limit

Fields to extract

Relative or absolute XPath expression | Column Name

+ X Up Down Save

Preview
Output

No preview available for XSD file

File Viewer

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:import namespace="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="Product">
      <xsd:annotation>
        <xsd:appinfo source="X_Label_EN">Product</xsd:appinfo>
        <xsd:appinfo source="X_Label_FR">Produit</xsd:appinfo>
      </xsd:annotation>
      <xsd:complexType>
        <xsd:all maxOccurs="1" minOccurs="1">
          <xsd:element maxOccurs="1" minOccurs="0" name="Picture">
            <xsd:annotation>
              <xsd:appinfo source="X_Label_EN">Picture</xsd:appinfo>
            </xsd:annotation>
          </xsd:element>
        </xsd:all>
      </xsd:complexType>
    </xsd:element>
  </xsd:import>
</xsd:schema>

```

Export as Context
Revert from Context

?
< Back
Next >
Finish
Cancel



Le schéma de l'entité sélectionnée s'affiche dans le panneau **Source Schema**.

Vous pouvez définir les paramètres à prendre en compte lors de la définition du schéma XML.

La boîte de dialogue relative au schéma est divisée en quatre panneaux :

Panneau	Description
Source Schema	Arborescence de l'entité chargée.
Target schema	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible.
File viewer	Visionneur des données brutes.

- Dans la zone **Xpath loop expression**, saisissez l'expression absolue du chemin xpath vers le noeud de la structure XML sur lequel appliquer l'itération. Vous pouvez également glisser le noeud du schéma source dans le champ Xpath du schéma cible. Le lien s'affiche en orange.



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

- Au besoin, définissez une limite (**Loop limit**) afin de restreindre l'itération à un certain nombre de noeuds.

The screenshot shows the Talend Open Studio interface for configuring a connection. On the left, the **Source Schema** pane displays a hierarchical tree of an XML schema with nodes like Product, Picture, Id, Name, Description, and Features. The **Features** node is highlighted with an orange selection bar. On the right, the **Target Schema** pane includes an **Xpath loop expression** section where the absolute XPath expression `/Product/Features` is entered, and a **Loop limit** of 50 is specified. Below this, there's a **Fields to extract** section with a button for defining extraction rules.

Dans la capture d'écran ci-dessus, utilisez *Features* comme élément sur lequel effectuer une boucle, puisqu'il est répété dans l'entité *Product* :

```
<Product>
  <Id>1</Id>
  <Name>Cup</Name>
  <Description/>
  <Features>
    <Feature>Color red</Feature>
    <Feature>Size maxi</Feature>
  <Features>
  ...
</Product>
<Product>
  <Id>2</Id>
  <Name>Cup</Name>
  <Description/>
  <Features>
    <Feature>Color blue</Feature>
    <Feature>Thermos</Feature>
```

```
<Features>
...
</Product>
```

Ce faisant, le composant **tMDMReceive** utilisant la connexion MDM va créer une nouvelle ligne pour chaque élément Feature rencontré.

- Afin de définir les champs à extraire, glissez le noeud correspondant du schéma source dans le champ **Relative or absolute XPath expression**.

The screenshot shows the Talend Studio interface with the 'Source Schema' and 'Target Schema' tabs open. In the Source Schema, a 'Product' schema is defined with fields: Picture, Id, Name, Description, Features, Sizes, Colors, Availability, Price, Family, and OnlineStore. The 'Features' field is selected and highlighted in orange. In the Target Schema, under 'Xpath loop expression', the 'Absolute XPath expression' is set to '/Product/Features' with a 'Loop limit' of 50. Below it, the 'Fields to extract' table lists three columns: 'Relative or absolute XPath expression' (containing ../Id, ../Name, and ../Description), 'Column N...' (containing Id, Name, and Description), and a 'Delete' column. A lightbulb icon with a tip is positioned next to the table, providing instructions on how to add and select fields.

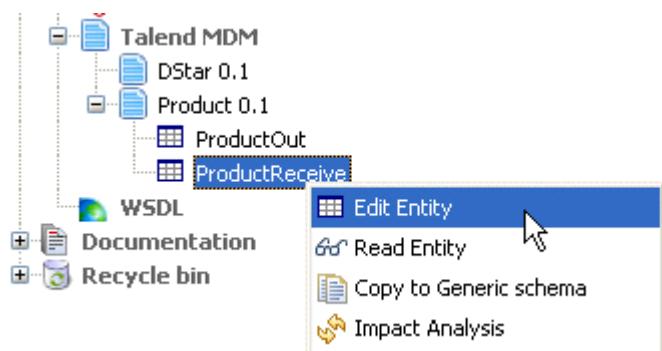
Cliquez sur le bouton [+] afin d'ajouter des lignes à la table, puis sélectionnez les champs à extraire. Appuyez sur la touche **Ctrl** ou **Maj** pour sélectionner plusieurs noeuds, groupés ou séparés, et glissez-les dans la table.

- Au besoin, saisissez un nom pour chaque colonne reçue, dans le champ **Column name**.

Vous pouvez prioriser l'ordre des champs à recevoir, en sélectionnant le champ et en utilisant les flèches montante et descendante. Le lien du champ sélectionné s'affiche en bleu, et tous les autres en gris.

- Cliquez sur **Finish** afin de valider vos modifications et fermer la boîte de dialogue.

Le nouveau schéma créé s'affiche sous la connexion MDM correspondante, dans la vue **Repository**.

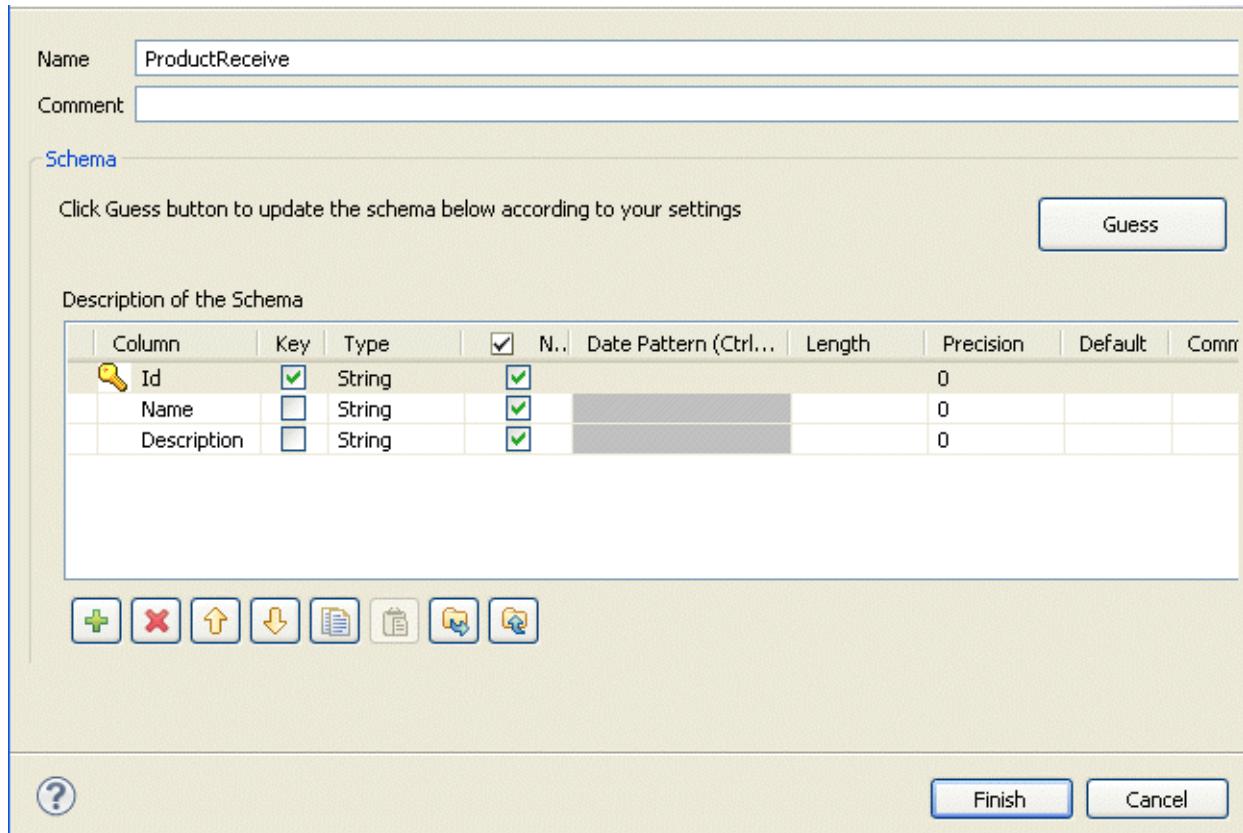


Pour modifier le schéma créé, procédez comme suit :

Gestion des métadonnées

Exporter une métadonnée en tant que contexte

- Dans la vue **Repository**, développez les noeuds **Metadata** et **Talend MDM** apuis parcourez le référentiel jusqu'au schéma que vous souhaitez modifier.
- Cliquez-droit sur le nom du schéma et sélectionnez **Edit Entity** dans le menu contextuel. Une boîte de dialogue s'affiche.



- Modifiez le schéma comme vous le souhaitez. Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.
- Cliquez sur **Finish** pour fermer la boîte de dialogue.
La connexion MDM de réception (**tMDMReceive**) est maintenant prête à être déposée dans l'un de vos Jobs.

7.17 Exporter une métadonnée en tant que contexte

Pour chaque métadonnée (fichier ou base de données), vous pouvez exporter les informations de connexion en tant que Contexte.

- Lors de la dernière ou avant dernière étape de configuration de schéma, cliquez sur **Export as Context**
- L'assistant de création de Contexte s'ouvre.
- Lors de l'étape 1, renseignez le nom du contexte. Par défaut, il prend le nom de la métadonnée.
- Cliquez sur Next.

- Lors de l'étape 2, les variables de contexte qui ont été créée automatiquement sont affichées dans une table à trois onglets. Vérifiez que les valeurs des variables des onglets **Values** sont correctes.
- Cliquez sur **Finish** pour valider la création.

Gestion des métadonnées

Exporter une métadonnée en tant que contexte



CHAPITRE 8

Gestion des routines

Ce chapitre définit les routines, fournit des cas d'utilisation et explique comment créer et gérer vos propres routines ou personnaliser les routines système. Pour avoir un aperçu des routines principales ainsi que d'autres exemples d'utilisation, consultez l'annexe *Routines système*.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez l'Annexe *Interface de Talend Open Studio, page 289*.

8.1 Définition des routines

Les routines sont des fonctions Java ou Perl plus ou moins complexes, généralement utilisées pour factoriser du code, et ainsi étendre les possibilités de traitement des données dans un ou plusieurs Job(s) technique(s).

Vous pouvez ainsi centraliser dans la vue **Repository** tous les morceaux de codes utilisés fréquemment ou tirer parti des méthodes déjà en place dans votre entreprise en les appelant via les routines. Cette factorisation facilite également la résolution des problèmes éventuels et permet la mise à jour à la volée du code utilisé dans des Jobs multiples.

En outre, certaines routines système reprennent les méthodes Java les plus courantes, dans une syntaxe **Talend**, qui vous permet de faire remonter les erreurs Java directement dans le Studio, facilitant ainsi l'identification et la résolution des problèmes au cours du développement de vos processus d'intégration avec **Talend**.

Les routines peuvent être de deux types :

- les routines système : un certain nombre de routines système sont fournies par défaut. Elles sont classées selon le type de données qu'elles traitent : numérique, chaîne de caractères, date...
- les routines utilisateur : ce sont les routines que vous créez ou adaptez à partir de routines existantes.



Il n'est pas nécessaire de connaître le langage Java pour créer et utiliser les routines **Talend**.

Toutes les routines sont conservées sous le noeud **Code > Routines** de la vue **Repository** sur la gauche du Studio.

Pour plus d'informations concernant les routines système, consultez *Accès aux routines système, page 266*.

Pour plus d'informations concernant la création des routines utilisateurs, consultez *Créer des routines personnalisées, page 269*.



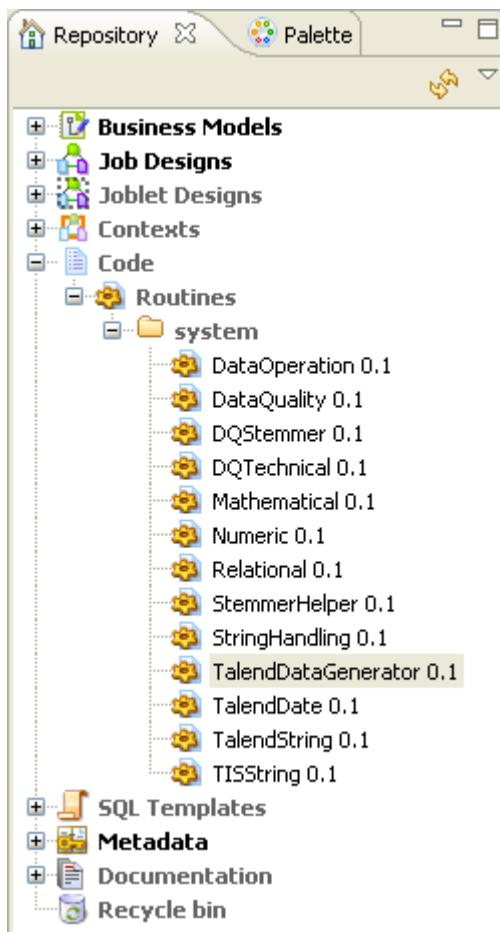
Vous pouvez également définir les dépendances des routines dans des Jobs. Pour ce faire, cliquez-droit sur un Job dans la vue **Repository** et sélectionnez **Set up routine dependancies**. Dans la boîte de dialogue qui s'ouvre, toutes les routines sont définies par défaut. Vous pouvez utiliser la barre d'outils afin de supprimer les routines, si nécessaire.

8.2 Accès aux routines système

Pour accéder aux routines système, cliquez sur **Code > Routines > system**. Les routines ou fonctions système sont regroupées par classe selon leur usage.



Le dossier **system**, ainsi que son contenu est en lecture seule.



Chaque classe répertoriée dans le dossier **system** contient plusieurs routines (fonctions). Double-cliquez sur la classe que vous souhaitez ouvrir.

Toutes les routines, autrement dit toutes les fonctions présentes dans une classe, se composent d'un commentaire explicatif suivi du code correspondant (Java ou Perl selon la langue de votre projet).

- Appuyez sur **Ctrl + O** dans la **vue Routines**.
Une boîte de dialogue affiche une liste des différentes routines de la catégorie.
- Cliquez sur la routine souhaitée.
La vue bascule vers la section comprenant le texte descriptif de la routine et le code correspondant.



La syntaxe d'appel des routines est sensible à la casse.

Pour plus d'informations concernant une sélection des routines les plus courantes (en Java), consultez l'annexe *Routines système*.

8.3 Personnalisation des routines système

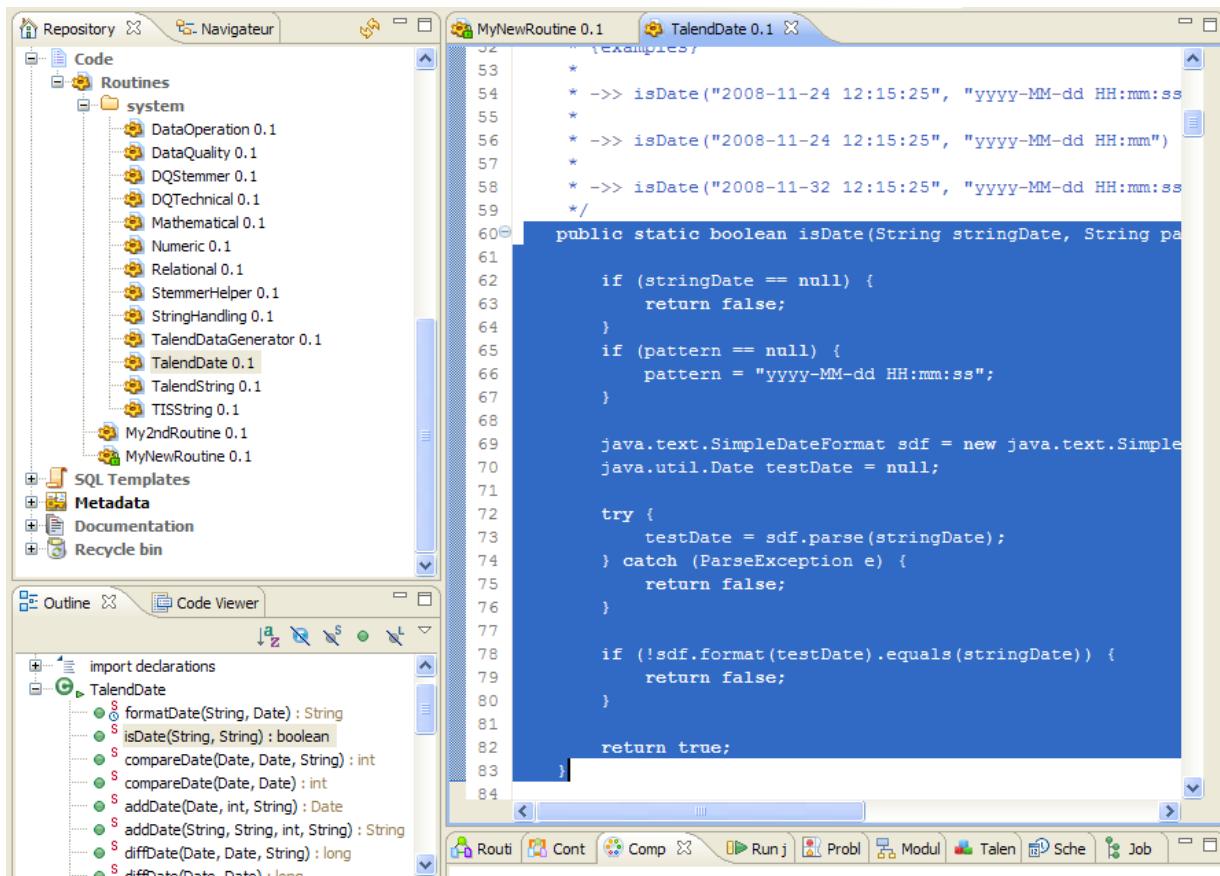
Si les routines système ne répondent pas exactement à vos besoins, vous pouvez les personnaliser en copiant leur contenu dans des routines utilisateur que vous créez.

Pour personnaliser une routine système

Gestion des routines

Gestion des routines personnalisées

- Créez d'abord une routine utilisateur en suivant les étapes décrites dans la section *Créer des routines personnalisées, page 269*. Cette routine s'ouvre dans l'espace de travail et contient l'exemple de base de routine.
- Puis dans **Code > Routines > system**, sélectionnez une classe de routines système qui contient la ou les routine(s) que vous souhaitez personnaliser.
- Double-cliquez sur la classe contenant la routine pour l'ouvrir dans l'espace de travail.
- Retrouvez la routine/fonction dont vous souhaitez copier tout ou partie, à l'aide du panneau **Outline** sur la gauche du studio.



- Dans l'espace de travail, sélectionnez tout ou partie du code, et copiez-le via **Ctrl+C**
- Cliquez sur l'onglet correspondant à votre routine utilisateur et collez le code du presse-papier (**Ctrl+V**).
- Adaptez ensuite le code à vos besoins, et appuyez sur **Ctrl+S** afin de l'enregistrer.

Nous vous recommandons d'utiliser la partie commentaire (en bleu) pour détailler les paramètres d'entrée et de sortie de votre routine et ainsi en faciliter la maintenance et la réutilisation.

8.4 Gestion des routines personnalisées

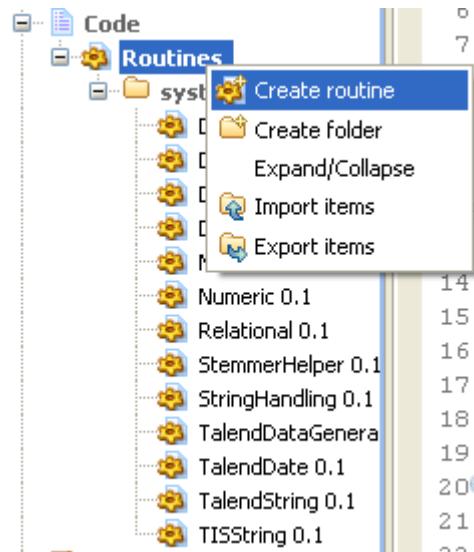
Talend Open Studio vous offre la possibilité de créer des routines personnalisées, de les modifier, ou de modifier des routines existantes, afin de répondre à vos besoins spécifiques.

8.4.1 Créeer des routines personnalisées

Vous pouvez créer vos propres routines pour répondre à vos besoins particuliers de factorisation. Comme les routines système, ces routines utilisateur seront centralisées dans la vue **Repository** (référentiel) sous **Code > Routines**. Vous pourrez ainsi les organiser par dossier selon vos besoins et les appeler facilement dans tous vos Jobs.

Pour créer un nouvelle routine :

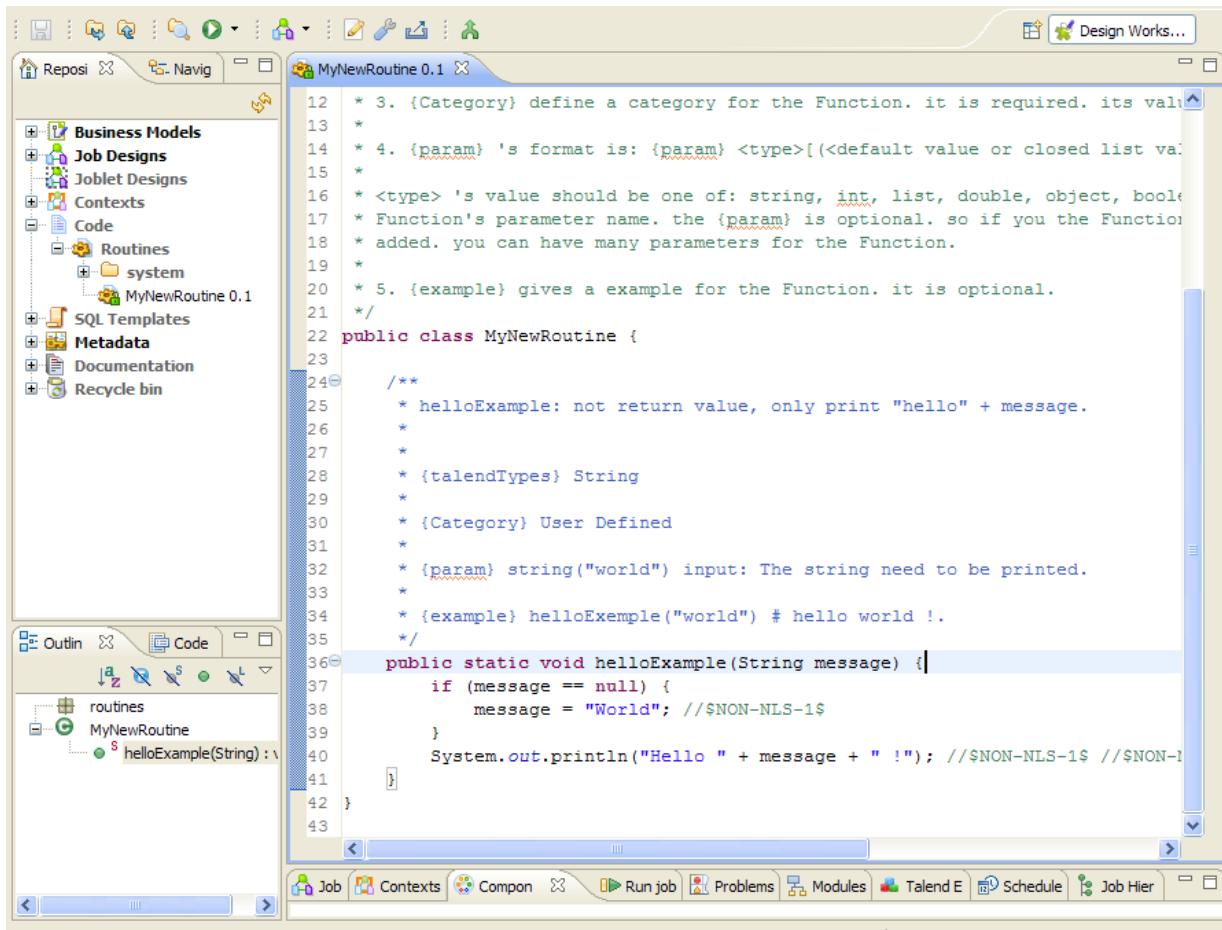
- Dans la vue **Repository**, cliquez sur **Code** pour accéder aux **Routines**,



- Puis cliquez-droit sur **Routines** et sélectionnez **Create Routine**.
- L'assistant [New routine] s'ouvre. Saisissez les informations nécessaires à la création de la routine, telles que son nom, sa description...
- Cliquez sur **Finish** pour valider la création et accéder à l'éditeur.

Gestion des routines

Gestion des routines personnalisées



La routine utilisateur nouvellement créée apparaît directement sous le noeud **Routines** dans la vue **Repository**. L'espace de travail s'ouvre sur un modèle de routine contenant par défaut un exemple simple, composé d'un commentaire en bleu, suivi du code correspondant.



Nous vous recommandons de documenter votre routine utilisateur à l'aide d'un commentaire détaillé. Ce commentaire inclut généralement les paramètres d'entrée et de sortie attendus pour l'utilisation de votre routine, ainsi que le résultat retourné par la routine et un exemple d'illustration. Ces informations sont généralement utiles pour la maintenance des routines et le travail collaboratif.

L'exemple de code suivant est fourni par défaut :

```
public static void helloExample(String message) {  
    if (message == null) {  
        message = "World"; //NON-NLS-1$  
    }  
    System.out.println("Hello " + message + " !");
```

- Adaptez ou remplacez ce modèle par votre propre code et sauvegardez-le à l'aide de **Ctrl+S**, ou refermez le fichier pour l'enregistrer automatiquement.



Vous pouvez également copier tout ou partie d'une classe ou routine système vers une routine utilisateur à l'aide des raccourcis du presse-papier : **Ctrl+C** et **Ctrl+V** afin de les adapter à vos besoins. Pour plus d'informations concernant la personnalisation des routines, consultez *Personnalisation des routines système, page 267*.

8.4.2 Editer des routines

Vous pouvez à tout moment modifier les routines que vous avez créées.



Rappel : Le dossier **system**, ainsi que toutes les routines système, est en lecture seule.

Pour éditer vos routines utilisateur :

- Cliquez-droit sur la routine que vous souhaitez éditer, puis sélectionnez **Edit Routine**.
- La routine s'ouvre dans l'espace de travail, et vous pouvez la modifier.
- Une fois adaptée à vos besoins, appuyez sur **Ctrl+S** afin de l'enregistrer.

Si vous souhaitez réutiliser une routine système pour vos besoins spécifiques, consultez la section *Personnalisation des routines système*.

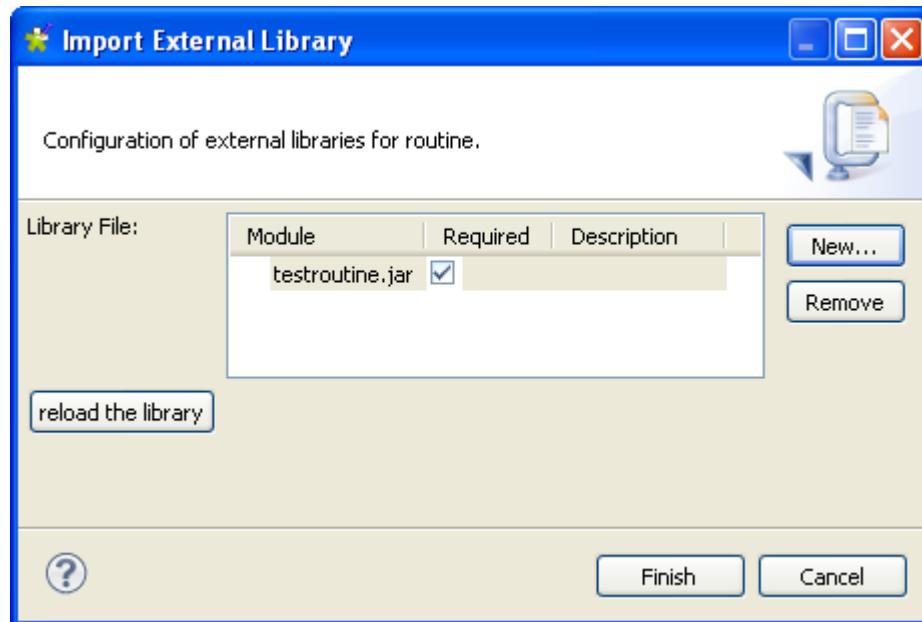
8.4.3 Editer les bibliothèques des routines utilisateur

Vous pouvez éditez la bibliothèque de chaque routine utilisateur en important des fichiers .jar externes pour la routine sélectionnée. Ces fichiers externes seront affichés, comme les modules, dans la vue **Modules** de votre Studio. Pour plus d'informations concernant la vue **Modules**, consultez *Installer des modules externes, page 111*.

Le fichier .jar de la bibliothèque importée sera également listé dans le fichier bibliothèque de votre Studio.

Pour éditer une bibliothèque de routine utilisateur, procédez comme suit :

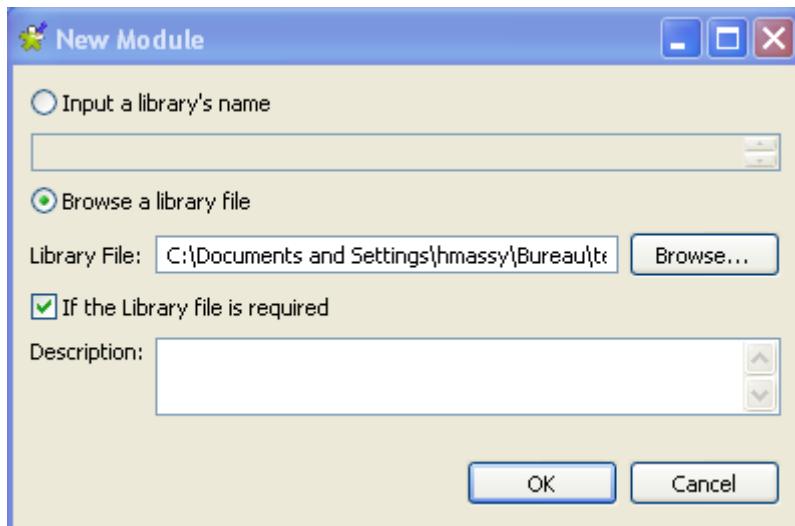
- Dans la vue **Repository**, développez les noeuds **Code > Routines**.
- Cliquez-droit sur la routine utilisateur dont vous souhaitez éditer la bibliothèque, puis sélectionnez **Edit Routine Library**.
La boîte de dialogue **[Import External Library]** s'ouvre.



- Cliquez sur **New** pour ouvrir une nouvelle boîte de dialogue dans laquelle vous pourrez importer la bibliothèque externe.



Vous pouvez supprimer tout fichier de routine déjà importé en sélectionnant le fichier dans la liste **Library File** et en cliquant sur le bouton **Remove**.



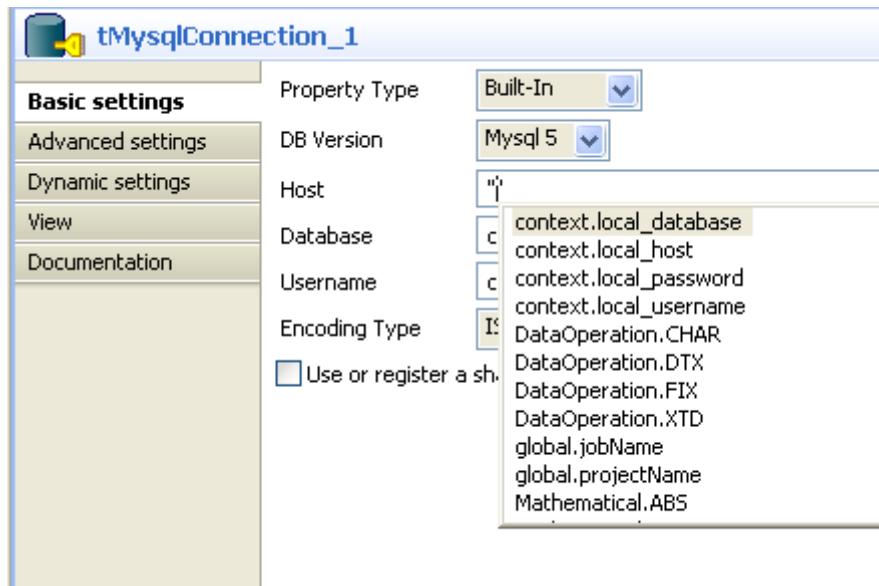
- Saisissez le nom du fichier bibliothèque dans le champ **Input a library's name** suivi du format de fichier (.jar), ou
- Sélectionnez l'option **Browse a library file** et cliquez sur **Browse** pour définir le chemin d'accès au fichier dans le champ correspondant.
- Au besoin, saisissez une description dans le champ **Description** puis cliquez sur **OK** afin de confirmer vos modifications.
Le fichier bibliothèque importé est affiché dans la liste **Library File**, dans la boîte de dialogue **[Import External Library]**.
- Cliquez sur **Finish** pour fermer la boîte de dialogue.
Le fichier bibliothèque est importé dans le dossier bibliothèque de votre Studio et s'affiche dans la vue **Module** de votre Studio.
Pour plus d'informations concernant la vue **Modules**, consultez *Installer des modules externes, page 111*.

8.5 Appel d'une routine à partir d'un Job

Pré-requis : Pour pouvoir exécuter votre routine, vous devez avoir créé au minimum un Job. Pour plus d'information sur la création d'un Job, consultez la section *Créer un Job, page 58* du **Guide Utilisateur Talend Open Studio**.

Vous avez la possibilité d'appeler toutes les routines (utilisateur et système) à partir des composants de vos Jobs afin d'exécuter automatiquement votre routine lors de l'exécution du Job qui la contient.

Les routines enregistrées dans le dossier **Routines** du référentiel sont accessibles dans tous les champs de saisie (texte ou code) des composants **Talend** utilisés dans vos Jobs, via la liste d'auto-complétion (**Ctrl+Espace**).

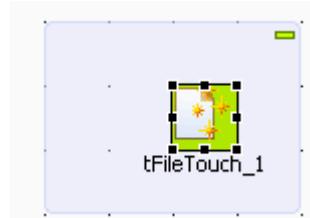


Pour faire appel à l'une de ces routines, il vous suffit d'appeler le nom de la classe suivie du nom de la routine puis des paramètres attendus, tel que :

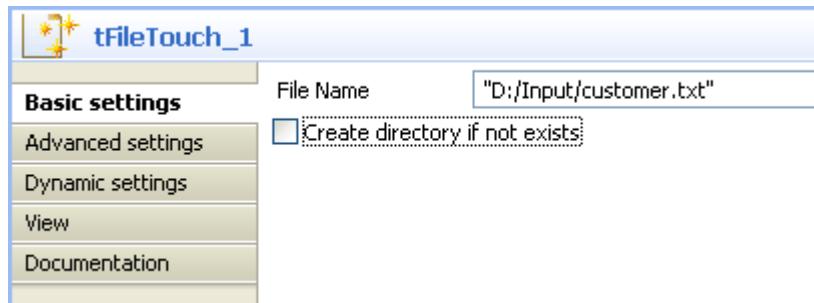
```
<ClassName>. <RoutineName>
```

8.6 Cas d'utilisation : créer un fichier à la date du jour

Le scénario suivant illustre l'utilisation d'une routine. Le Job se compose d'un composant unique qui appelle une routine système.



- Dans la **Palette**, cliquez sur le dossier **File > Management**, puis glissez un **tFileTouch** dans l'éditeur graphique. Ce composant permet de créer un fichier vide.
- Double-cliquez sur le composant afin d'afficher sa vue **Basic settings** dans l'onglet **Component**.
- Dans le champ **FileName**, saisissez le chemin d'accès à votre fichier, ou cliquez sur le bouton [...] afin de parcourir votre répertoire.

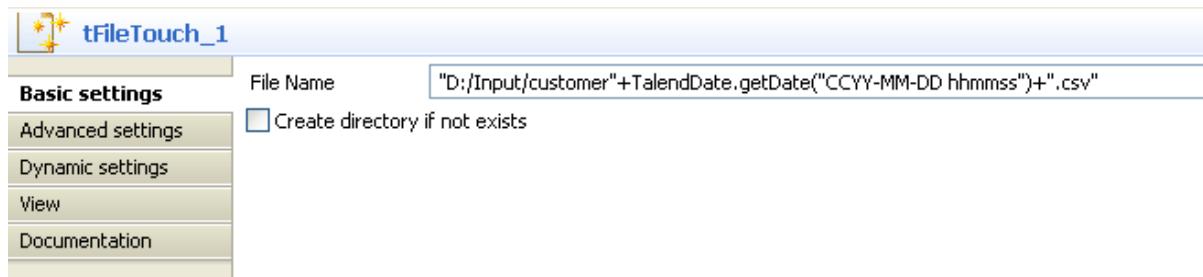


- Fermez les guillemets avant l'extension de votre fichier, tel que : "D:/Input/customer".txt.

Gestion des routines

Cas d'utilisation : créer un fichier à la date du jour

- Ajoutez le signe plus (+) entre les guillemets fermants et l'extension du fichier.
- Puis appuyez sur **Ctrl+espace** pour ouvrir la liste exhaustive des routines.
- Dans la liste d'auto-complétion qui s'affiche, sélectionnez *TalendDate.getDate* pour utiliser la routine de **Talend** qui permet d'obtenir la date courante.
- Modifiez le format de date fourni par défaut si besoin.
- Saisissez + après la variable *getDate* pour terminer l'appel de routine.
- Puis entourez de guillemets l'extension du fichier.



Si vous êtes sous Windows, les “:” entre les heures et les minutes, et entre les minutes et les secondes doivent être retirés.

- Appuyez sur **F6** afin d'exécuter votre Job.
- Le composant **tFileTouch** a créé un fichier vide qui porte la date du jour, telle que récupérée lors de l'exécution de la routine *GetDate* appelée.





CHAPITRE 9

SQL Templates

Les modèles SQL sont des groupes d'arguments de requêtes prédéfinis, exécutés en mode ELT. Ce chapitre donne une définition du mode ELT, indique ce que sont les modèles SQL et fournit à l'utilisateur des exemples de scénarios, afin d'expliquer comment utiliser les modèles SQL existants, ou comment créer vos modèles SQL personnalisés.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de [Talend Open Studio](#). Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

9.1 Qu'est-ce que l'ELT ?

Extract, Load and Transform (ELT), ou, en français, Extraire, Charger et Transformer, est un processus de manipulation de données lié aux bases de données, et plus particulièrement aux data warehouses. Ce mode est différent du mode ETL habituel (Extract, Transform, Load). En mode ELT, les données sont extraites, chargées dans la base de données, puis transformées dans la base de données, à l'emplacement défini, avant toute utilisation. Ces données sont migrées en masse et le processus de transformation s'effectue après le chargement des données dans le SGBD cible, en format brut. Cela permet de libérer de la bande passante.

Cependant, le mode ELT n'est pas optimal dans toutes les situations. Par exemple :

- SQL est moins puissant que Java, le nombre de transformations de données est limité.
- les utilisateurs du mode ELT doivent avoir des compétences en personnalisation de SQL et de SGBD.
- Si vous utilisez le mode ELT avec **Talend Open Studio**, vous ne pouvez ni passer ni rejeter une ligne de données, comme vous pouvez le faire avec l'ETL. Pour plus d'informations concernant le rejet de lignes, consultez la section *Types de connexions, page 83*.

Les modèles SQL sont conçus pour faciliter l'utilisation du mode ELT.

9.2 Définition des SQL Templates Talend

Le SQL est un langage standardisé de requêtes, utilisé pour accéder aux informations des bases de données et les gérer. Le SQL peut être utilisé pour les requêtes sur les données, les mises à jour, la création et la modification de schémas, et le contrôle d'accès aux données. **Talend Open Studio** fournit de nombreux modèles SQL pour simplifier les tâches les plus communes. Il comprend également un éditeur SQL vous permettant de personnaliser ou de créer vos propres modèles SQL, afin de répondre à vos besoins.

Ces modèles SQL sont utilisés avec les composants de la famille **Talend ELT**, tels que le **tELT**, le **tELTFilterColumns**, le **tELTCommit**, le **tELTFilterRows**, le **tELTRollback** et le **tELTAgregate**. Ces composants exécutent les instructions SQL sélectionnées. A l'aide des opérateurs UNION, EXCEPT et INTERSECT, vous pouvez modifier les données directement dans le SGBD sans utiliser le système de mémoire.

De plus, avec ces modèles SQL, vous pouvez optimiser les performances de votre SGBD, en stockant et en récupérant vos données selon vos besoins structurels.

Talend Open Studio fournit les types suivants de modèles SQL, sous le noeud **SQL templates** de la vue **Repository** :

- Modèles SQL **system** : ils sont classés selon le type de base de données.
- Modèles SQL personnalisés : ce sont les modèles que vous avez créés ou adaptés des modèles existants.

Des informations plus précises concernant les modèles SQL sont présentées dans les sections suivantes.

Pour plus d'informations concernant les composants de la famille ELT, consultez les *Composants ELT, page 921*, dans le **Guide de Référence** des composants de **Talend Open Studio**.



Comme la plupart des modèles SQL sont créés pour une base de données spécifique, si vous changez de base de données dans votre système, vous devrez également changer de modèles, ou en développer de nouveaux.

9.3 Gérer les SQL Templates Talend

Talend Open Studio vous permet, via le dossier **SQL Templates** de la vue **Repository**, d'utiliser des modèles SQL système ou personnalisés dans les Jobs que vous créez dans le Studio, à l'aide des composants de la famille ELT.

Les sections suivantes vous expliquent comment gérer ces deux types de modèles SQL.

9.3.1 Les types de SQL Templates système

Cette section fournit des informations détaillées concernant les différents types de modèles SQL prédéfinis.

Les instructions de chaque groupe de modèles varient d'une base de données à l'autre, selon les opérations à effectuer. Elles sont également groupées dans chaque dossier, selon leur type.

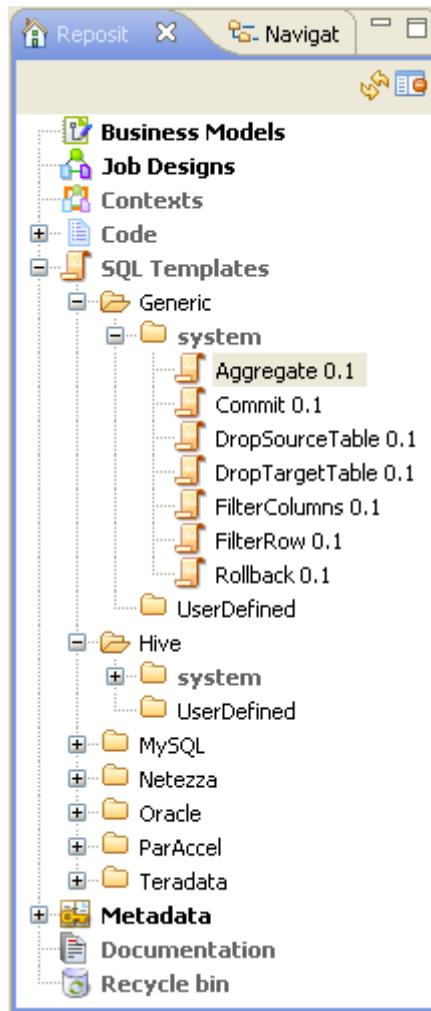
Le tableau ci-dessous présente ces types ainsi que les informations liées à chacun.

Nom	Fonction	Composants associés	Paramètres requis dans le composant
<i>Aggregate</i>	Réalise l'agrégation (sum, average, count, etc.) d'un ensemble de données.	tELTAggregate	Nom de la base de données Nom de la table source Nom de la table cible
<i>Commit</i>	Envoie une instruction de Commit au SGBDR.	tELT tELTAggregate tELTCommit tELTFilterColumns tELTFilterRows tELTRollback	Null
<i>Rollback</i>	Envoie une instruction Rollback au SGBDR.	tELT tELTAggregate tELTCommit tELTFilterColumns tELTFilterRow tELTRollback	Null
<i>DropSourceTable</i>	Supprime une source table.	tELT tELTAggregate tELTFilterColumns tELTRow	Nom de la table (lors de l'utilisation du tELT) Nom de la table source
<i>DropTargetTable</i>	Supprime une table cible.	tELTAggregate tELTFilterColumns tELTRow	Nom de la table cible.
<i>FilterColumns</i>	Sélectionne et extrait un ensemble de données dans des colonnes données du SGBDR.	tELTAggregate tELTFilterColumns tELTRows	Nom de la table cible (et schéma) Nom de la table source (et schéma)

<i>FilterRow</i>	Sélectionne et extrait un ensemble de données dans des lignes données du SGBDR.	tELTFilterRow	Nom de la table cible (et schéma) Nom de la table source (et schéma) Conditions
------------------	---	----------------------	---

9.3.2 Accéder aux SQL Templates système

Pour accéder à un modèle SQL système, développez le noeud **SQL Templates** de la vue **Repository**.



Chaque dossier contient un sous-dossier **system**, dans lequel se trouvent des instructions SQL prédefinies, et un sous-dossier **UserDefined** dans lequel vous pouvez stocker les instructions SQL que vous avez créées ou personnalisées.

Chaque dossier système contient différents types de modèles SQL, chaque modèle étant créé pour accomplir une tâche dédiée.

A l'exception du dossier **Generic**, les modèles SQL sont groupés dans différents dossiers, selon le type de base de données avec lequel les utiliser. Les modèles du dossier **Generic** ne sont dédiés à aucune base de données, ils sont standards et conviennent à toutes. A partir de ceux-ci, vous pouvez développer des modèles SQL plus spécifiques que ceux définis par [Talend Open Studio](#).



Les dossiers **system** et leur contenu sont en lecture seule.

Dans la vue **Repository**, procédez comme suit pour ouvrir un modèle SQL :

- Dans la vue **Repository**, développez le noeud **SQL Templates** et parcourez jusqu'au modèle que vous souhaitez ouvrir.
 - Double-cliquez sur la classe que vous voulez ouvrir, par exemple *aggregate*, dans le dossier **Generic**.
- La vue du modèle *aggregate* s'affiche dans l'espace de travail.

```

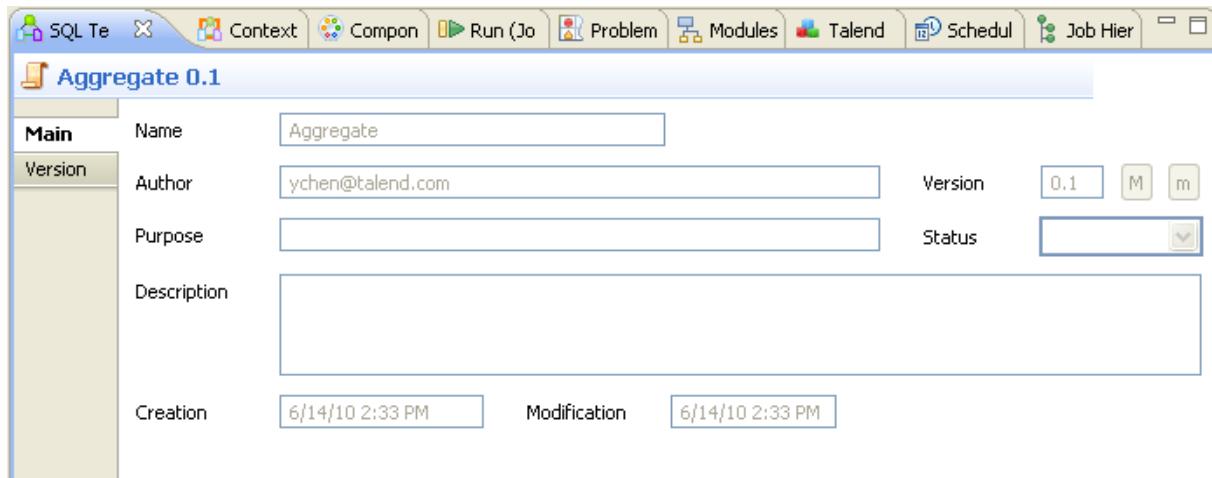
1 <%
2     EXTRACT(__GROUPBY__);
3     EXTRACT(__OPERATION__);
4     String operation = "";
5     boolean flag=false;
6     for(int i=0; i < __OPERATION_INPUT_COLUMN__.length; i++){
7         if(flag){
8             operation += ",";
9         }else{
10            flag=true;
11        }
12        operation += (__OPERATION_FUNCTION__[i] + "(" + __OPERATION_INPU
13    )
14
15 %>
16
17 INSERT INTO <%=__TABLE_NAME_TARGET__%> (<%=StringUtils.list(__OPERATI
18 SELECT <%= operation %>, <%= StringUtils.list(__GROUPBY_INPUT_COLUMN_
19 GROUP BY <%=StringUtils.list(__GROUPBY_INPUT_COLUMN__, "", "", "", "") %>
```

Vous pouvez lire les instructions *aggregate* prédéfinies dans la vue du modèle, ainsi que les paramètres, tels que **TABLE_NAME_TARGET**, **operation**, qui sont à définir lors de la création de Jobs liés. La configuration peut ensuite facilement être paramétrée, via les composants associés, comme expliqué dans la section *Définition des SQL Templates Talend*, page 276.

- A chaque fois que vous cliquez sur un modèle SQL ou que vous ouvrez un modèle SQL, la vue des propriétés correspondante à ce modèle s'affiche en bas du Studio. Par exemple, vous cliquez le modèle *aggregate* ou vous l'ouvrez, ses propriétés seront présentées comme suit :

SQL Templates

Gérer les SQL Templates Talend

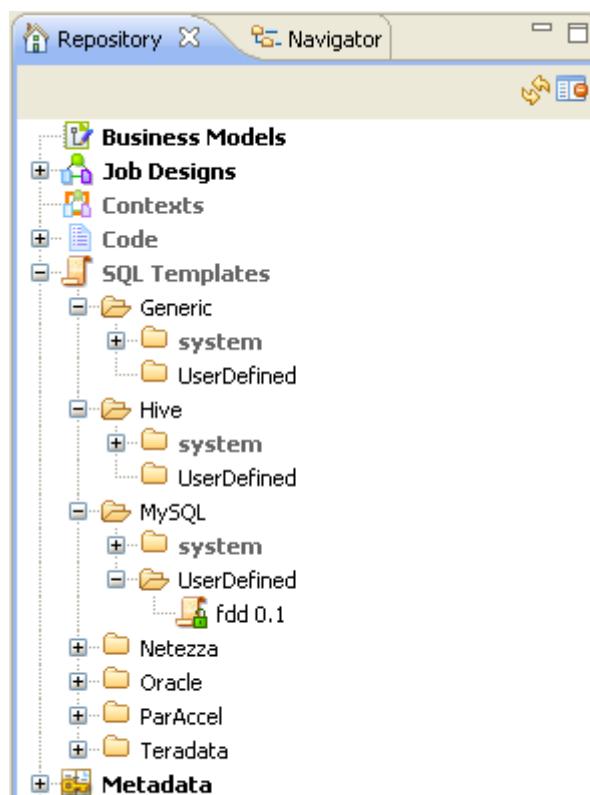


Pour plus d'informations concernant les types de modèles SQL, consultez la section *Les types de SQL Templates système, page 277*.

Pour plus d'informations concernant l'utilisation des modèles SQL avec les composants associés, consultez la section *Exemple d'utilisation des SQL Templates système, page 282*.

9.3.3 Créez des SQL Templates personnalisés

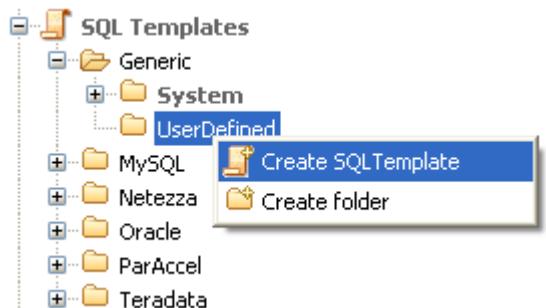
Comme la transformation que vous devez effectuer en mode ELT peut dépasser les possibilités des modèles SQL, **Talend Open Studio** vous permet de développer vos propres modèles SQL, à condition de respecter quelques règles d'écriture. Ces modèles SQL sont stockés dans les dossiers **User-defined**, regroupés selon la base de données dans laquelle ils seront utilisés.



Pour plus d'informations concernant les règles d'écriture des modèles SQL, consultez l'Annexe *Règles d'écriture des SQL Templates, page 315.*

Pour créer un modèle SQL personnalisé :

- Dans la vue **Repository**, développez le noeud **SQL Templates**, puis la catégorie dans laquelle vous souhaitez créer le modèle SQL.



- Cliquez-droit sur **UserDefined** et sélectionnez **Create SQL Template** pour ouvrir l'assistant **[SQL Templates]**.



- Saisissez les informations requises pour créer le modèle, puis cliquez sur **Finish** pour fermer l'assistant.

Le nouveau modèle apparaît sous le noeud **UserDefined**, dans la vue **Repository**. Un éditeur de modèles SQL s'ouvre dans l'espace de modélisation graphique. Vous pouvez saisir le code du nouveau modèle.

Pour plus d'informations concernant la création de modèles SQL personnalisés et leur utilisation dans des Jobs, consultez le *Scénario : Effectuer une opération d'itération sur des tables et en effacer le contenu grâce à un modèle SQL défini par l'utilisateur (SQL Template)* du **Guide de Référence Talend Open Studio**.

9.3.4 Exemple d'utilisation des SQL Templates système

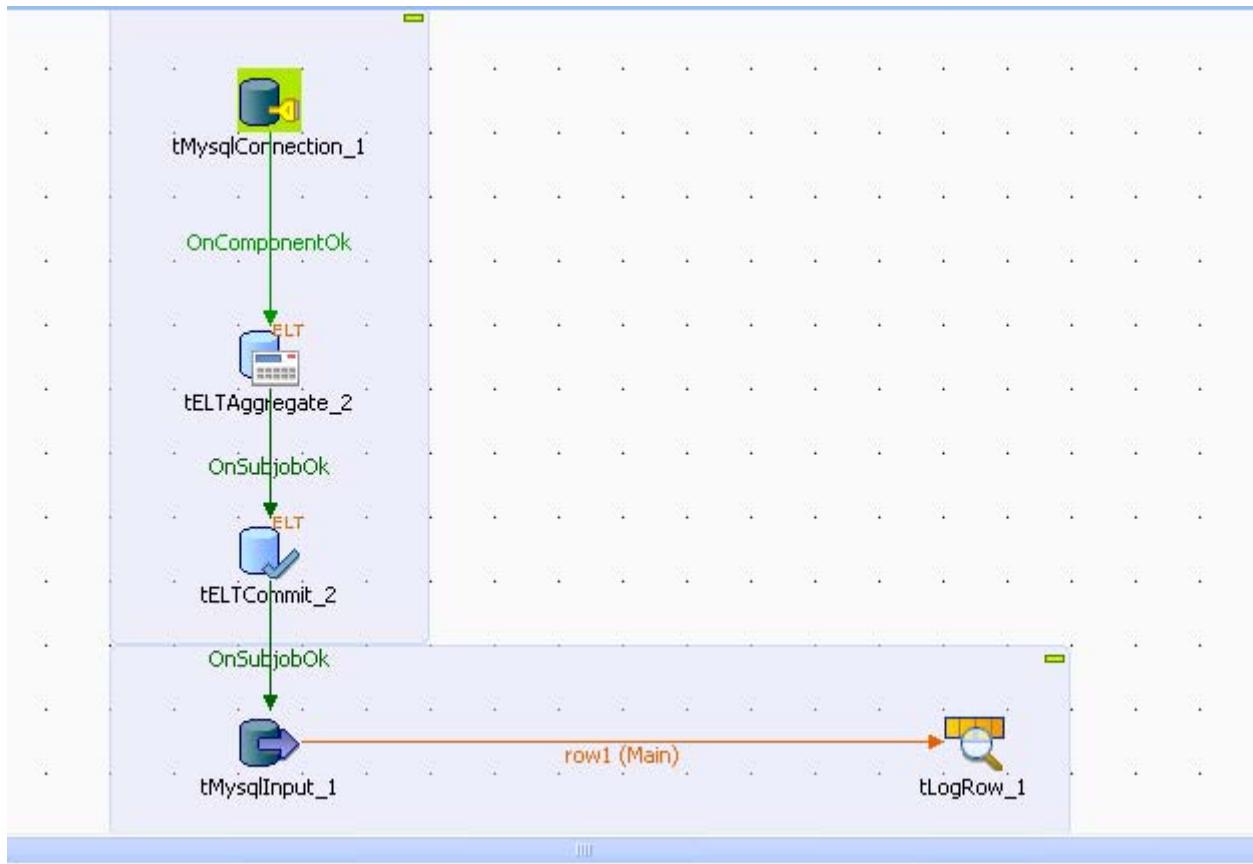
Puisque beaucoup d'instructions SQL sont communes, standardisées, **Talend Open Studio** vous permet de bénéficier de plusieurs modèles SQL système.

Cette section vous présente un exemple d'utilisation des modèles SQL système, à travers les étapes de l'utilisation de modèles SQL système Mysql dans un Job qui :

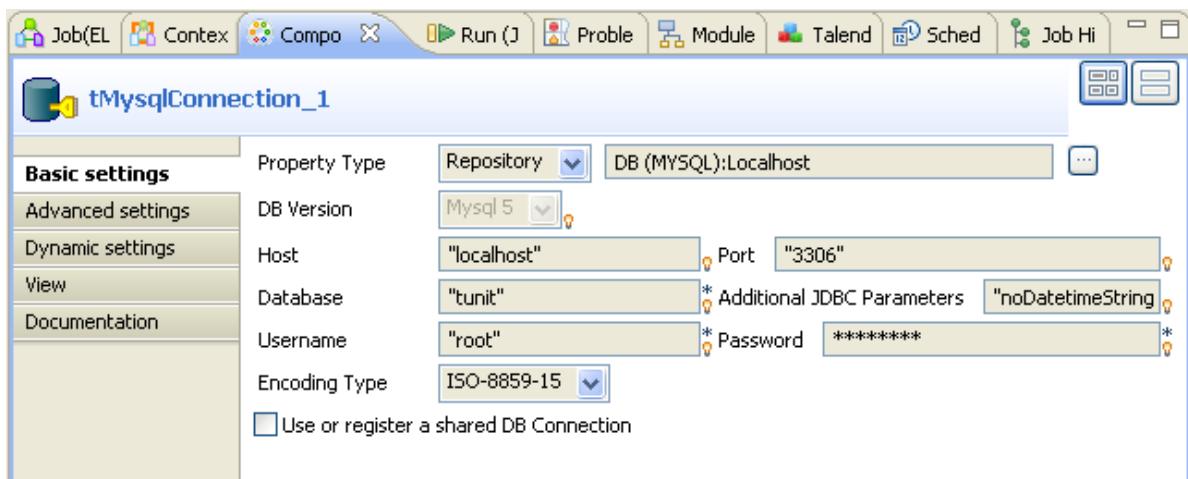
- ouvre une connexion à une base de données MySQL,
- collecte des données regroupées par valeurs spécifiques à partir d'une table de la base de données et écrit des données agrégées dans une table cible de la base de données,
- supprime la table source d'où proviennent les données agrégées,
- lit la table cible et liste les résultats d'exécution du Job.

Pour vous connecter à la base de données et agréger les colonnes de la table :

- Glissez les composants suivants de la **Palette** dans l'espace de modélisation graphique : **tMysqlConnection**, **tELTAggregate**, **tELTCommit**, **tMysqlInput**, et **tLogRow**.
- Cliquez-droit sur le **tMysqlConnection** et dans le menu contextuel, sélectionnez **Trigger > OnComponentOk** afin de lier le **tMysqlConnection** au **tELTAggregate**.
- Connectez le **tELTAggregate**, le **tELTCommit** et le **tMysqlInput** à l'aide de liens **OnComponentOk**.
- Connectez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Main > Row**.



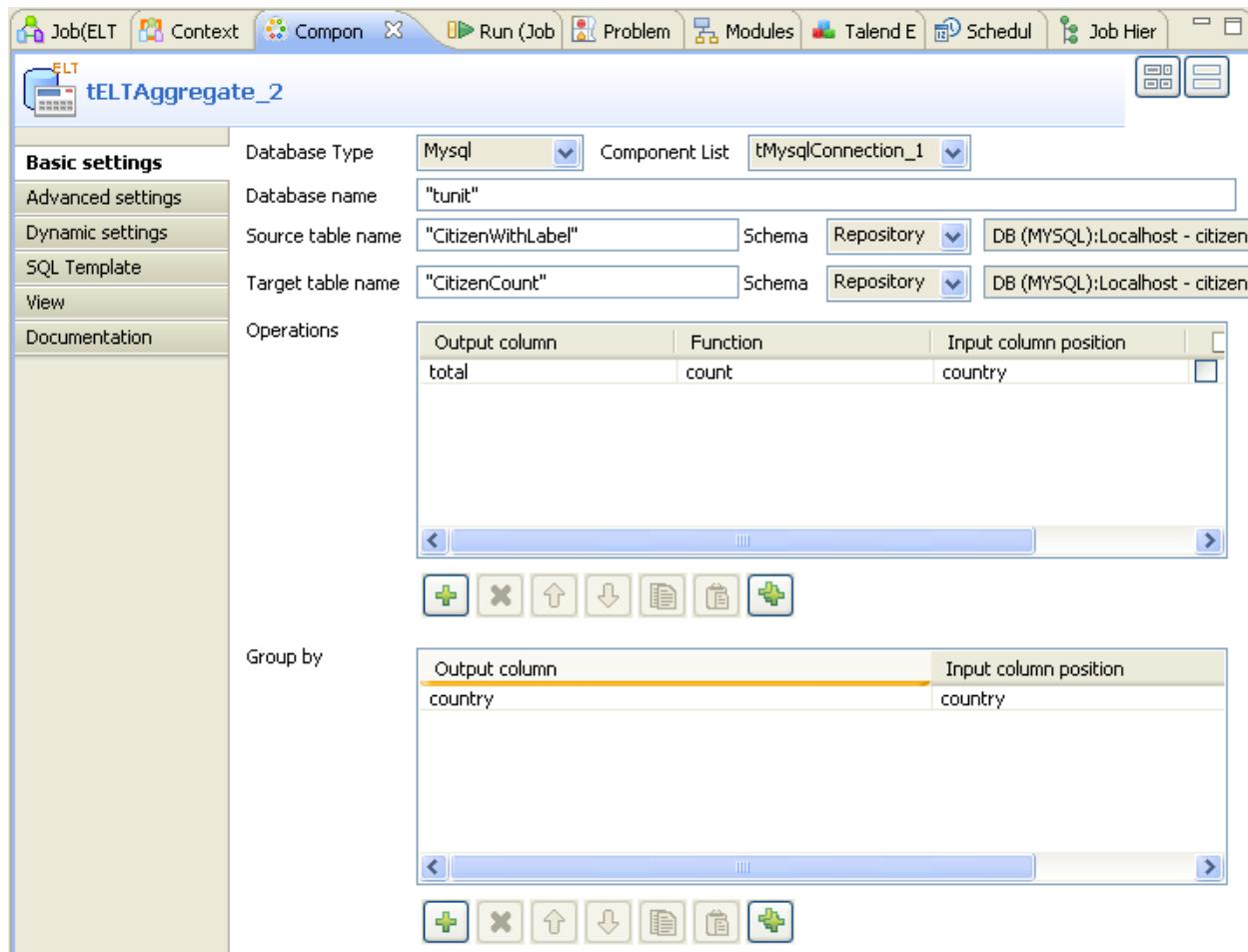
- Dans l'espace de modélisation graphique, sélectionnez le composant **tMysqlConnection** et cliquez sur l'onglet **Component** pour définir ses propriétés.
- Dans la vue **Basic settings**, configurez manuellement les paramètres de connexion.



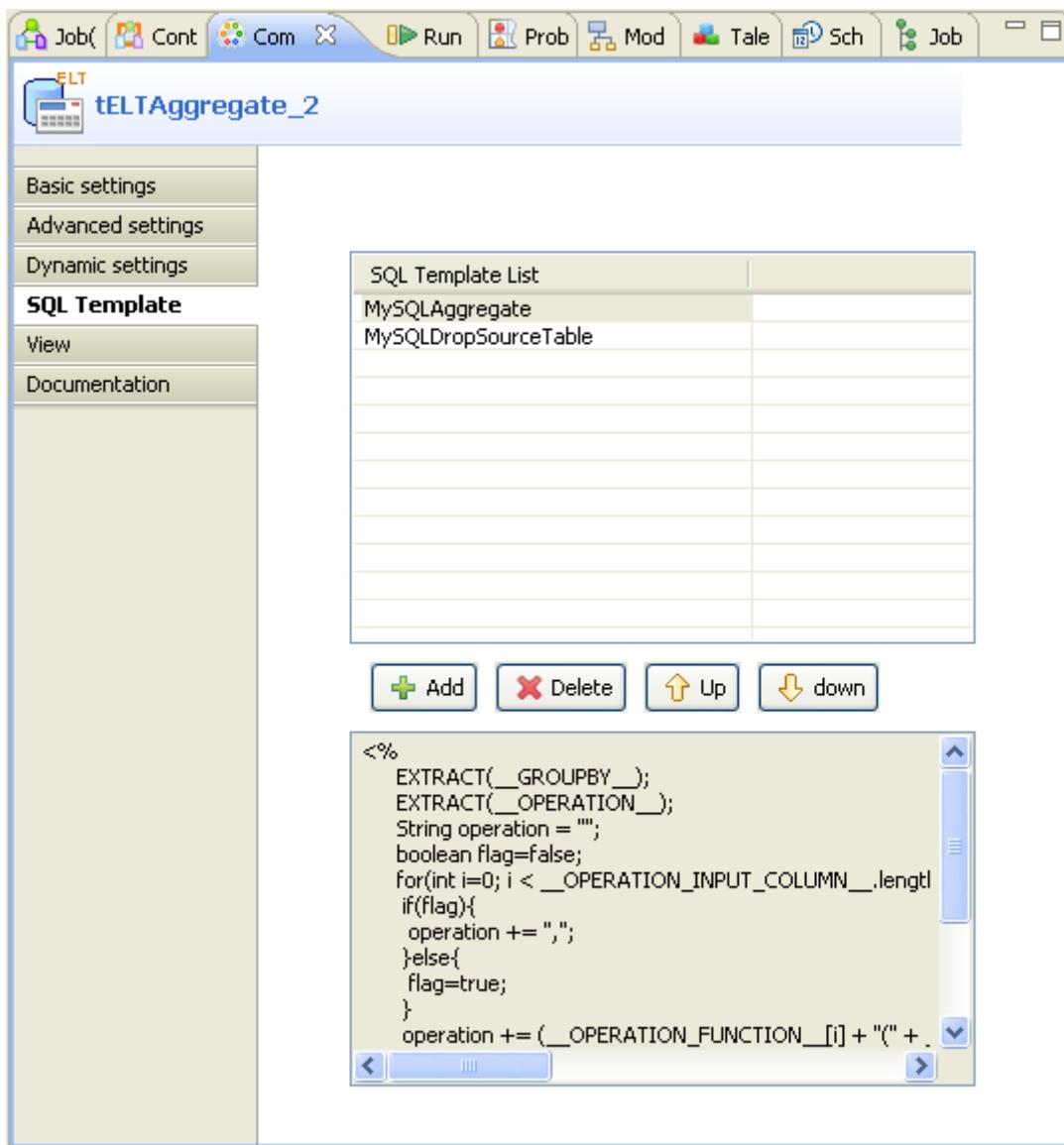
- Dans l'espace de modélisation graphique, sélectionnez le **tELTAggregate** et cliquez sur l'onglet **Component** pour définir ses propriétés (**Basic settings**).
- Dans la liste **Database type**, sélectionnez la base de données correspondante.
- Dans la liste **Component list**, sélectionnez le composant de connexion adéquat, si plus d'une connexion est ouverte dans le Job.

- Saisissez les noms de la base de données, de la table source et de la table source dans les champs correspondants, puis cliquez sur les boutons [...] à côté des champs **Edit schema**, afin de définir la structure des données, dans la table source et dans la table cible.

Le schéma de la table source comprend trois colonnes : *First_Name*, *Last_Name* et *Country*. Le schéma de la table cible comprend deux colonnes : *country* et *total*. Dans cet exemple, regroupez les citoyens par nationalité, et comptez le nombre de personnes dans chaque pays. Pour ce faire, définissez les paramètres **Operations** et **Groupby** selon vos besoins.



- Dans le tableau **Operations**, cliquez sur le bouton [+] afin d'ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** pour sélectionner la colonne de sortie qui contiendra les données comptées.
- Cliquez sur la ligne **Function**, puis sélectionnez l'opération à effectuer.
- Dans le tableau **Group by**, cliquez sur le bouton [+] afin d'ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** pour sélectionner la colonne de sortie qui contiendra les données agrégées.
- Cliquez sur l'onglet **SQL template** pour ouvrir la vue correspondante.



- Cliquez deux fois sur le bouton [+] sous le tableau **SQL template list** afin d'ajouter deux modèles SQL.
- Cliquez sur la ligne du premier modèle SQL et sélectionnez le modèle *MySQLAggregate* dans la liste déroulante. Ce modèle génère du code pour agréger des données, selon la configuration des **Basic settings**.
- Répétez l'opération et sélectionnez le modèle *MySQLDropSourceTable* pour la ligne du deuxième modèle. Ce modèle génère du code pour supprimer la table source d'où proviennent les données agrégées.

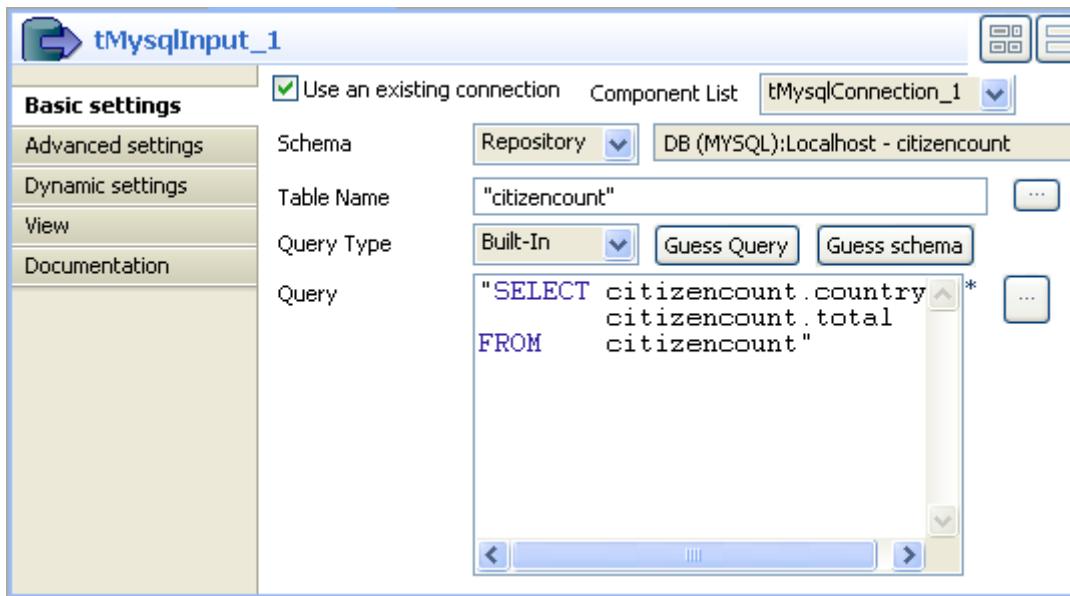


Afin d'ajouter de nouveaux modèles SQL à un composant ELT chargé de les exécuter, vous pouvez simplement glisser le(s) modèle(s) de votre choix à ce composant soit dans l'espace de travail soit dans le tableau **SQL template list** de ce composant.



Les modèles définis dans le tableau **SQL template list** sont prioritaires par rapport à la configuration paramétrée dans la vue **Basic settings**, et sont exécutés en ordre descendant. Dans cet exemple, si vous sélectionnez uniquement *MySQLDropSourceTable* dans la liste, la table source sera supprimée définitivement, et l'agrégation ne pourra être effectuée.

- Dans l'espace de modélisation graphique, sélectionnez le composant **tELTCommit** et cliquez sur l'onglet **Component** afin de définir ses propriétés.
- Dans la liste **Database type**, sélectionnez la base de données que vous utilisez.
- Dans la liste **Component list**, sélectionnez le composant de connexion adéquat, si plus d'une connexion est ouverte.
- Dans l'espace de modélisation graphique, sélectionnez le composant **tMysqlInput**, et cliquez sur l'onglet **Component** pour définir ses propriétés.



- Cochez la case **Use an existing connection** pour utiliser la connexion à la base de données définie dans le composant **tMysqlConnection**.
- Afin de définir le schéma, sélectionnez **Repository** et cliquez sur le bouton [...] pour sélectionner la table de la base de données dont le schéma est utilisé. Dans cet exemple, la table cible contenant les données agrégées est sélectionnée.
- Dans le champ **Table Name**, saisissez le nom de la table sur laquelle effectuer votre requête. Dans cet exemple, la table est celle qui contient les données agrégées.
- Dans la zone **Query**, saisissez l'instruction de la requête pour sélectionner les colonnes à afficher.
- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter. La table source est supprimée.

```
Starting job ELTYudong at 02:43 24/05/2010.  
[statistics] connecting to socket on port 3918  
[statistics] connected  
+-----+  
| tLogRow_1 |  
+-----+  
| country | total |  
+-----+  
| Canada  | 2030 |  
| China   | 2012 |  
| France  | 2009 |  
| Japan   | 1925 |  
| USA     | 2024 |  
+-----+  
[statistics] disconnected  
Job ELTYudong ended at 02:43 24/05/2010. [exit code=0]
```

Une table composée de deux colonnes, *citizenCount*, est créée dans la base de données. Elle regroupe les citoyens par nationalité, et affiche le total de personnes dans chaque pays.



ANNEXEA

Interface de **Talend Open Studio**

Cette annexe décrit l'interface de **Talend Open Studio**.

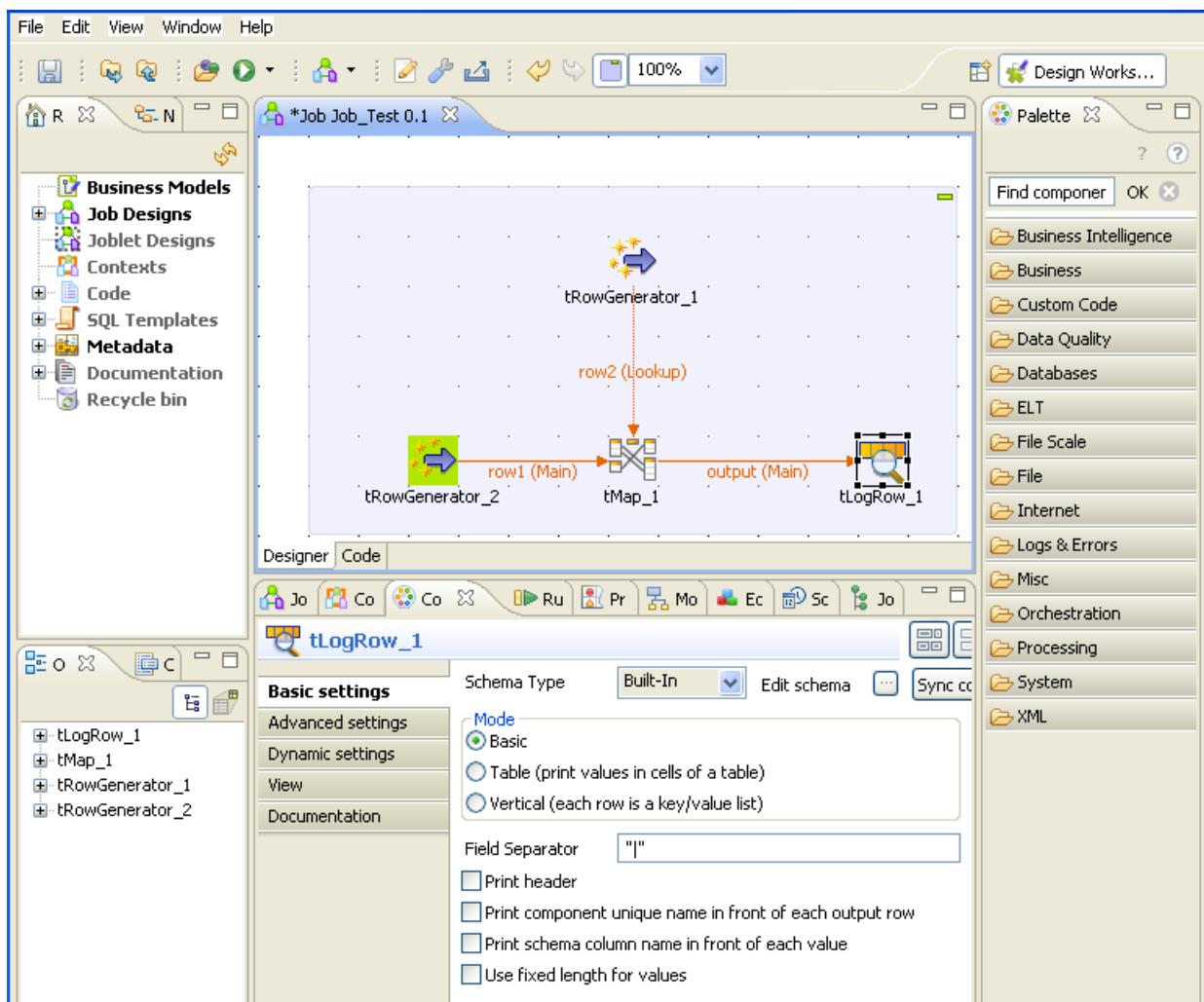
A.1 Fenêtre principale

La fenêtre principale de **Talend Open Studio** est l'interface à partir de laquelle vous pouvez gérer tous les types de processus d'intégration de données.

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

- Menus
- Barres d'outils
- Référentiel (**Repository**)
- Espace de modélisation (**Design workspace**)
- Diverses vues de configuration organisées en onglets pour chaque élément du Job d'intégration de données créé dans l'espace de modélisation
- Aperçu schéma (**Outline**) et Aperçu du code (**Code Viewer**)
- **Outline view and Code Viewer.**

La capture d'écran ci-dessous vous montre la fenêtre principale de **Talend Open Studio**, ses panneaux et ses vues.



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

A.2 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 à **Talend** ainsi que certaines fonctionnalités Eclipse.

A.2.1 Barre de menu de Talend Open Studio

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 à utiliser principalement au niveau du workspace, ainsi que certaines fonctions spécifiques **Talend Open Studio**.

Le tableau ci-dessous décrit les menus et sous-menus disponible dans la barre de menu de **Talend Open Studio**.



Les menus de la barre de menu sont légèrement différents selon l'élément avec lequel vous travaillez : un Business Model ou un Job.

Menu	Sous-menu	Description
File	Close	Ferme la vue courante ouverte dans l'espace de modélisation du Studio.
	Close All	Ferme toutes les vues ouvertes dans l'espace de modélisation du Studio.
	Save	Enregistre toutes les modifications apportées dans la vue courante. Pour plus d'informations, consultez <i>Enregistrement d'un Business Model, page 55</i>
	Save as	Enregistre en tant que nouveau Job.
	Save All	Enregistre toutes les modifications apportées dans toutes les vues ouvertes.
	Print	Option indisponible.
	Switch project	Ferme le projet courant et ouvre la fenêtre de login permettant d'ouvrir un nouveau projet dans le Studio.
	Edit project properties	Ouvre un boîte de dialogue permettant de personnaliser les propriétés du projet. Pour plus d'informations, consultez <i>Personnaliser les paramètres du projet, page 32</i> .
	Import	Ouvre un assistant permettant d'importer différents types de ressources (fichiers, éléments, préférences, catalogues XML, etc.) de différentes sources.
	Export	Ouvre un assistant permettant d'exporter différents types de ressources (fichiers, éléments, préférences, catalogues XML, etc.) vers différentes destinations.
	Exit	Ferme la fenêtre principale du Studio.
	Open File	Ouvre un fichier à partir du Studio.
Edit	Undo Move Node	Annule la dernière action effectuée dans l'espace de modélisation du Studio.
	Redo	Reproduit la dernière action effectuée dans l'espace de modélisation du Studio.
	Cut	Coupe l'objet sélectionné dans l'espace de modélisation du Studio.
	Copy	Copie l'objet sélectionné dans l'espace de modélisation du Studio.
	Paste	Colle l'objet préalablement copié dans l'espace de modélisation du Studio.
	Delete	Supprime l'objet sélectionné dans l'espace de modélisation du Studio.
	Select All	Sélectionne tous les composants présents dans l'espace de modélisation du Studio.
View	Zoom In	Affiche l'image du Job ouvert en plus grand.
	Zoom Out	Affiche l'image du Job ouvert en plus petit.
	Grid	Affiche la grille dans l'espace de modélisation graphique. Tous les éléments s'alignent alors sur la grille.
	Snap to Geometry	Active la fonction Snap to Geometry (Aligner sur la grille).

Menu	Sous-menu	Description
Window	Perspective	Ouvre une liste de différentes perspectives selon les éléments de la liste.
	Show View...	Ouvre la boîte de dialogue [Show View] permettant d'afficher les différentes vues dans le Studio.
	Maximize Active View or Editor...	Agrandit la vue courante.
	Preferences	Ouvre la boîte de dialogue [Preferences] permettant de paramétrier les préférences de Talend Open Studio . Pour plus d'informations concernant les préférences, consultez <i>Configurer les préférences de Talend Open Studio, page 21</i> .
Help	Welcome	Ouvre une page d'accueil contenant des liens vers la documentation de Talend Open Studio et vers les sites pratiques de Talend .
	Help Contents	Ouvre l'aide en ligne d'Eclipse.
	About Talend Open Studio	Affiche : -la version du logiciel utilisé -des informations détaillées sur la configuration du logiciel pouvant être utile en cas de problème -des informations détaillées sur les plug-in(s) -des informations détaillées concernant les fonctionnalités de Talend Open Studio .
	Export logs	Ouvre un assistant vous permettant d'exporter tous les fichiers .log générés par le Studio et des informations sur la configuration de votre ordinateur dans un fichier archive.
	Software Updates	Find and Install... : Ouvre un assistant [Install/Update] permettant de chercher les mises à jour des fonctionnalités couramment installées ou de chercher de nouvelles fonctionnalités à installer. Manage Configuration... : Ouvre une boîte de dialogue [Product Configuration] dans laquelle gérer la configuration de Talend Open Studio .

A.2.2 Barre d'outils de Talend Open Studio

La barre d'outils vous permet un accès rapide aux fonctions les plus courantes de **Talend Open Studio**.



Les icônes de la barre d'outils sont légèrement différentes selon l'élément avec lequel vous travaillez : un Business Model ou un Job.

Le tableau ci-dessous décrit les icônes de la barre d'outils et leurs fonctions.

Nom	Icône	Description
Save		Enregistre le Job Design courant.

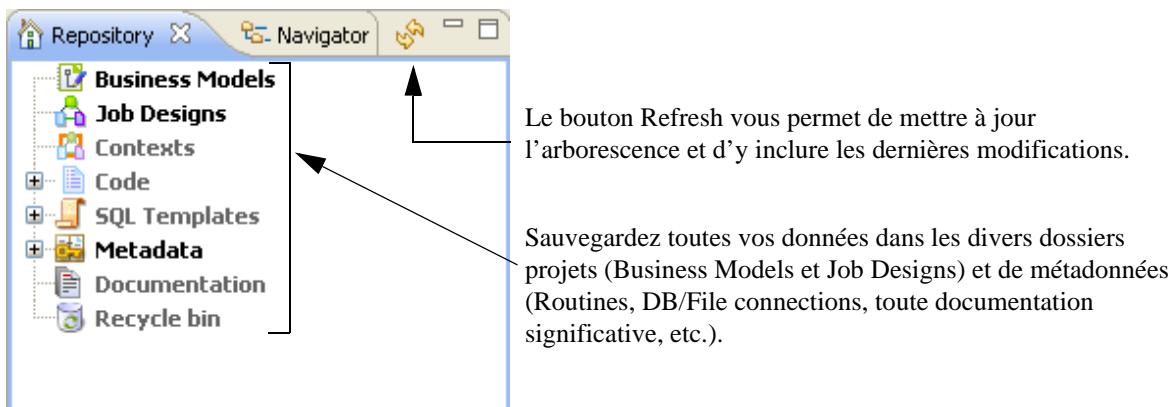
Nom	Icône	Description
Save as		Enregistre en tant que nouveau Job.
Export items		Exporte les éléments du Repository dans un fichier archive, pour les déployer en dehors de Talend Open Studio . Si vous souhaitez importer les éléments exportés dans une version plus récente de Talend Open Studio ou d'un autre ordinateur, assurez-vous d'avoir inclu les fichiers source dans l'archive, en cochant la case adéquate.
Import items		Importe les éléments du Repository d'un fichier archive dans la version courant de Talend Open Studio . Pour plus d'informations, consultez <i>Import d'éléments, page 131</i> .
Find a specific job		Lance l'assistant de recherche de Job vous permettant d'ouvrir tout Job listé dans la vue Repository .
Run job		Exécute le Job courant ouvert dans l'espace de modélisation. Pour plus d'informations, consultez <i>Exécution d'un job, page 72</i>
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.
Project settings		Lance la fenêtre [Project Settings] des paramètres du projet. A partir de cette fenêtre, vous pouvez ajouter une description au projet en cours et personnaliser l'affichage de la Palette . Pour plus d'informations, consultez <i>Personnaliser les paramètres du projet, page 32</i> .
Detect and update all jobs		Cherche les mises à jour disponibles pour vos Jobs.
Export Talend projects		Lance l'assistant [Export Talend projects] d'Export de projet. Pour plus d'informations concernant l'Export de projet, consultez <i>Exporter un projet, page 19</i> .

A.3 Repository

Le référentiel, **Repository**, est une arborescence regroupant 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 **Repository** centralise et conserve localement tous les éléments contenus dans un projet.

La capture d'écran ci-dessous montre les éléments stockés dans le **Repository**.



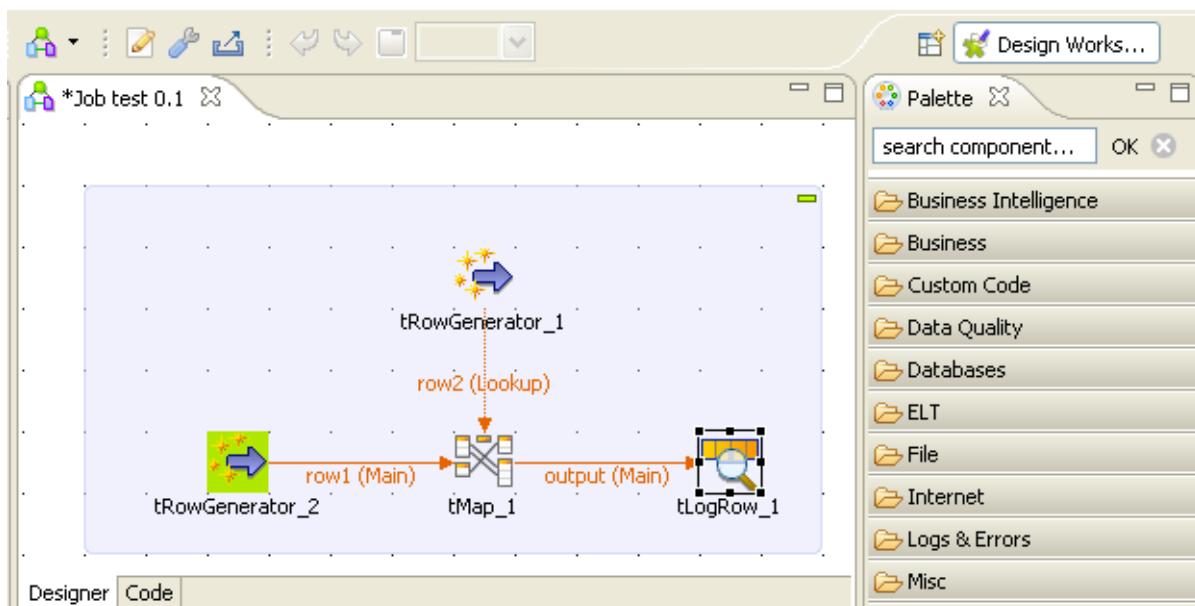
Le tableau ci-dessous décrit les noeuds de la vue **Repository**.

Noeud	Description
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. Pour plus d'informations, consultez <i>.Conception d'un Business Model, page 43</i>
Job Designs	Le dossier Job Designs montre l'arborescence des Jobs créés dans le projet courant. Double-cliquez sur le nom d'un Job pour l'ouvrir dans l'espace de modélisation graphique. Pour plus d'informations, consultez <i>Conception d'un Job d'intégration de données, page 57</i> .
Contexts	Le dossier Context groupe les fichiers contenant les variables de contextes que vous souhaitez réutiliser dans différents Jobs, comme les chemin d'accès ou les informations de connexion. Pour plus d'informations, consultez <i>Centraliser les Contextes et les Variables, page 91</i> .
Code	Le dossier Code est une librairie regroupant les routines disponibles dans ce projet et les autres pièces de code pouvant être réutilisées dans le projet. Cliquez sur l'entrée de l'arborescence afin de développer la pièce de code correspondante. Pour plus d'informations, consultez <i>Conception d'un Job d'intégration de données, page 57</i> .
SQL Templates	Le dossier SQL Templates regroupe tous les modèles SQL système et offre la possibilité de créer des modèles SQL personnalisés. Pour plus d'informations, consultez <i>Utilisation des SQL Templates, page 104</i> .
Metadata	Le dossier Metadata rassemble les fichiers contenant les informations redondantes que vous souhaitez réutiliser dans différents Jobs, comme les schémas ou les informations de propriétés. Pour plus d'informations, consultez <i>Gestion des métadonnées, page 189</i> .
Documentation	Le dossier Documentation regroupe tout type de documents, de tout format, par exemple, des spécifications ou la description d'un format technique particulier d'un fichier. Double-cliquez pour ouvrir le document dans l'application correspondante. Pour plus d'informations, consultez <i>Générer la documentation HTML, page 156</i> .

Noeud	Description
Recycle bin	<p>Recycle bin, la Corbeille rassemble tous les éléments supprimés à partir de tout dossier du Repository.</p> <p>Les éléments supprimés sont toujours présents dans vos fichiers système, dans la Corbeille, jusqu'à ce que vous cliquiez-droit sur l'icône de la Corbeille et sélectionnez Empty Recycle bin.</p> <p>Développez la Corbeille afin de voir les dossiers, les sous-dossiers ou éléments qu'elle contient. Vous pouvez agir directement sur un élément depuis la Corbeille, le restaurer ou le supprimer définitivement en cliquant-droit dessus et en sélectionnant l'action souhaitée dans la liste.</p>

A.4 Espace de modélisation graphique

Le **Design workspace**, ou espace de modélisation, est l'éditeur graphique des Jobs de **Talend Open Studio** qui vous permet de concevoir graphiquement les Business Models (processus métier) mais également les Job Designs (processus techniques).



Dans cet éditeur graphique, vous pouvez ouvrir à la fois les Job Designs et les Business Models. 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.

A.5 Palette

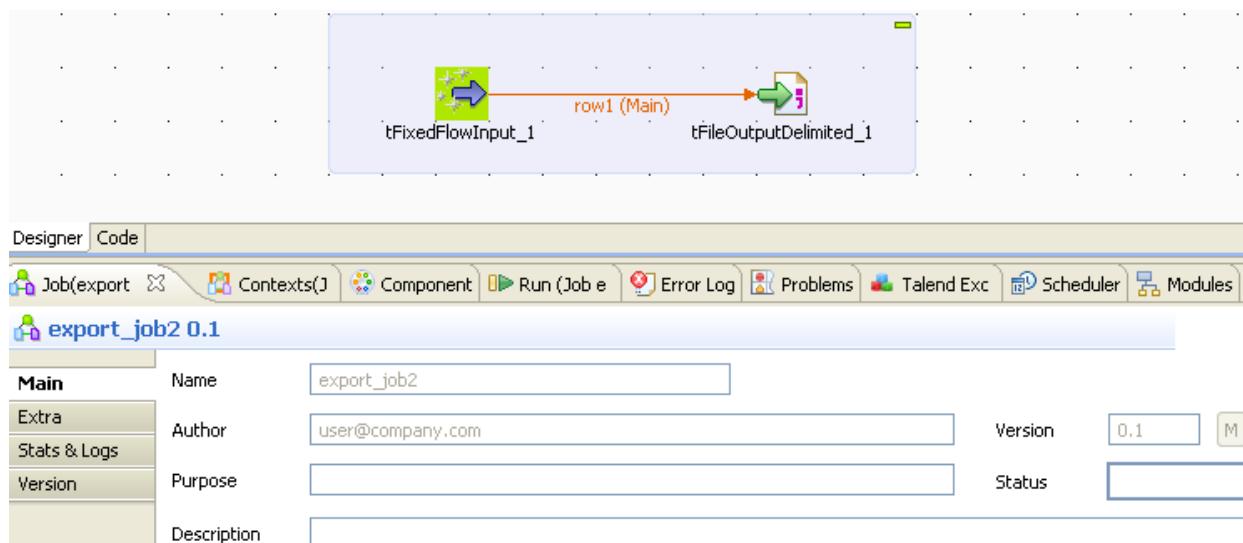
A partir de la **Palette**, selon que vous travaillez sur un Job Design ou un Business Model, déposez les formes, branches, notes ou composants techniques dans l'espace de modélisation. Puis définissez et formatez les éléments graphiques à l'aide des paramètres de propriétés disponibles dans la vue **Business Model** pour les Business Models et dans la vue **Component** pour les Job Designs.

Voir également :

- *Conception d'un Business Model, page 43.*
- *Conception d'un Job d'intégration de données, page 57.*
- *Personnaliser l'espace de travail, page 78.*

A.6 Onglets de configuration

Les onglets de configuration sont situés dans la partie inférieure de l'espace de modélisation graphique. Chaque onglet ouvre une vue affichant les propriétés de l'élément sélectionné dans l'espace de modélisation graphique. Ces propriétés peuvent être éditées pour modifier ou définir les paramètres relatifs à un composant précis ou à un Job complet.



Les onglets **Component**, **Run Job** 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 **Component**, **Run Job** et **Error Log**. Ces deux vues sont indépendantes des Jobs, actifs ou inactifs, ouverts dans l'espace de modélisation.



Vous pouvez afficher plus d'onglets et ouvrir la vue correspondante directement si vous sélectionnez **Window > Show view** puis, dans la boîte de dialogue, développez un noeud et sélectionnez l'élément que vous souhaitez afficher.

Les sections ci-dessous décrivent la vue de chaque onglet de configuration.

Onglet	Description
Component	Cette vue fournit des informations spécifiques sur les paramètres de chaque composant de la Palette. Pour créer un Job qui fonctionne, vous devez renseigner les champs de cette vue Component , et ce pour tous les composants du Job. Pour plus d'informations concernant la vue Component , consultez <i>Définition des propriétés d'un composant, page 65</i> .

Onglet	Description
Run Job	Cet onglet, comme son nom le suggère, affiche l'exécution du Job technique. 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 <i>Exécution d'un job, page 72</i> .
Error Log	Cette vue est généralement utile lors de l'exécution des Jobs, car elle affiche les éventuelles erreurs d'exécution.  L'onglet Error Log a également une fonction informative, notamment pour indiquer qu'un composant Perl/Java est en cours d'exécution. L'onglet Error Log est masqué par défaut. Comme pour tout autre onglet, cliquez sur Window > Show views , puis développez le noeud PDE Runtime et sélectionnez Error Log pour l'afficher.
Modules	Cette vue indique quel module est requis pour l'utilisation du composant correspondant.. Consultez la vue Modules pour vérifier les modules présents ou manquants pour l'exécution de vos Jobs. Pour plus d'informations, consultez <i>Installer des modules externes, page 111</i> .
Scheduler	Cette vue vous permet de planifier une tâche qui exécutera périodiquement le Job sélectionné via le programme crontab. Pour plus d'informations, consultez <i>Lancer un Job périodiquement, page 113</i> .

Onglet	Description	
Vue Job	La vue Job affiche de nombreuses informations relatives au Job ouvert dans l'espace de modélisation graphique. Cette vue contient les onglets suivants :	
	Onglet Main	Cet onglet affiche des informations simples concernant le Job ouvert dans l'espace de modélisation graphique, c'est-à-dire son nom, son auteur, son numéro de version, etc. Ces informations sont en lecture seule. Pour les éditer, vous devez fermer votre Job, cliquer-droit sur son libellé dans la vue Repository et cliquez sur Edit properties dans le menu contextuel.
	Onglet Extra	Cet onglet affiche les paramètres supplémentaires comme les fonctions de multi thread et de chargement de contexte implicite. Pour plus d'informations, consultez <i>Fonctions Extra, page 126</i>
	Onglet Stats & Logs	Cet onglet vous permet d'activer/désactiver les statistiques et les logs du Job complet. Vous pouvez 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 ces composants, consultez <i>tFlowMeterCatcher</i> , <i>tLogCatcher</i> et <i>tStatCatcher</i> dans le Guide de Référence Composants Talend Open Studio . De plus, vous pouvez dorénavant appliquer ces fonctionnalités à la totalité de votre Job actif (c'est-à-dire 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 également définir les paramètres courants comme préférences en cliquant sur le bouton Save to preferences . Pour plus d'informations concernant le fonctionnement automatique des Stats & Logs, consultez <i>Automatiser l'utilisation de statistiques & logs, page 124</i>
	Onglet Version	Cet onglet regroupe les différentes versions du Job ouvert, ainsi que leurs date et heure de création et de modification.
Talend Exchange	Cette vue vous donne accès dans Talend Exchange à une liste de composants compatibles avec votre version de Talend Open Studio mais qui ne sont pas installés dans votre Palette . Vous pouvez ensuite télécharger ces composants pour les réutiliser dans les Job Designs créés dans votre Studio. Pour plus d'informations, consultez <i>Télécharger des composants de la communauté, page 110</i> .	
Problems	Cette vue affiche tous les messages liés aux icônes liées aux composants en cas de problème, par exemple lorsqu'il manque une partie de la configuration. Trois types d'icônes/messages existent : Error , Warning et Infos . Pour plus d'informations, consultez <i>Gérer les icônes d'avertissement/d'erreur sur les composants, page 119</i> .	

Onglet	Description
Job Hierarchy	Cette vue affiche, sous forme d'arborescence, les Jobs enfant du Job parent sélectionné. Pour faire apparaître cette vue, cliquez-droit sur un Job parent dans le Repository et sélectionnez l'option Open Job Hierarchy dans le menu contextuel.. Vous pouvez également afficher cette vue en suivant le chemin Window > Show view... > Talend > Job Hierarchy . La hiérarchie d'un Job n'apparaîtra que si vous créez un Job parent et un ou plusieurs enfant(s) à l'aide du composant tRunJob . Pour plus d'informations concernant le tRunJob , consultez <i>tRunJob</i> dans le Guide de Référence Composants Talend Open Studio .
Properties	Lorsque vous insérez une forme dans l'éditeur graphique, l'onglet Properties vous permet d'utiliser un grand éventail d'outils de formatage pour vous aider à personnaliser votre Business Model et améliorer sa lisibilité.

A.7 Panneau des onglets outline et code

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

Ce panneau est composé de deux onglets, **Outline** et **Code Viewer**, qui fournissent des informations concernant le Job ou le Business Model affiché, ainsi que le code généré.

Pour plus d'informations, consultez *Afficher les onglets Code ou Outline de votre Job*, page 121.

A.8 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	Affiche l'onglet Designer du Job courant.	Application globale
Ctrl + G	Afficher l'onglet Code du Job courant.	Application globale
Ctrl + R	Restaurer la vue Repository initiale	Dans la vue Repository
Ctrl + Shift + F3	Synchroniser les templates perljet des composants et les classes java associées.	Application globale
Ctrl + Shift + J	Ouvrir un Job	Application globale (sous Windows)
F7	Passer en mode Debug .	Dans la vue Run Job
F5	Rafraîchir la vue Repository.	Dans la vue Repository
F8	Tuer le Job courant.	Dans la vue Run Job

Raccourci	Pour...	Contexte
F5	Rafraîchir le statut d'installation des Modules .	Dans la vue Modules
Ctrl+L	Exécuter les requêtes SQL	Commande Talend (sous Windows)
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



ANNEXEB

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 produit le mieux possible. Ce chapitre est un complément du Guide de référence des composants **Talend Open Studio**.

B.1 Présentation du scénario

Pour illustrer le fonctionnement de **Talend Open Studio**, vous trouverez ci-dessous un scénario reflétant un cas d'utilisation réel. 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).



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.

B.1.1 Données en entrée

Le contenu du fichier en entrée est une liste des clients de toutes les régions de l'état de Californie. Ces données seront donc chargées dans une table de données.

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)

B.1.2 Données en sortie

Vous souhaitez charger uniquement les données des clients habitant dans certaines régions (*Counties*) de la Californie dans la nouvelle base de données : les régions d'Orange et de Los Angeles.

La structure de la table est légèrement différente, ainsi les données devant être chargées dans la 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 par 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 par le nom de la région dans laquelle se situe la ville. Un fichier de référence vous aidera à filtrer les villes des régions d'Orange et de Los Angeles.

B.1.3 Données de référence

Etant donné que les données des régions d'Orange et de Los Angeles doivent être chargées dans la base de données, vous devez mapper les villes de Californie avec leur région respective, afin de pouvoir filtrer uniquement les villes d'Orange et de Los Angeles.

Pour cela, utilisez un fichier de référence contenant la liste des villes situées dans ces régions, par exemple :

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

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

B.2 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

B.2.1 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**.

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**, 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 46*.
- Le **Job Designer** : Pour plus d'informations, consultez *Créer un Job, page 58*.
- Le **Metadata Manager** : Pour plus d'informations, consultez *Centraliser les métadonnées, page 91*.

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 400 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 et placez-le à la gauche de l'espace de modélisation.

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 définir 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 s'ouvre :

- 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 des 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.

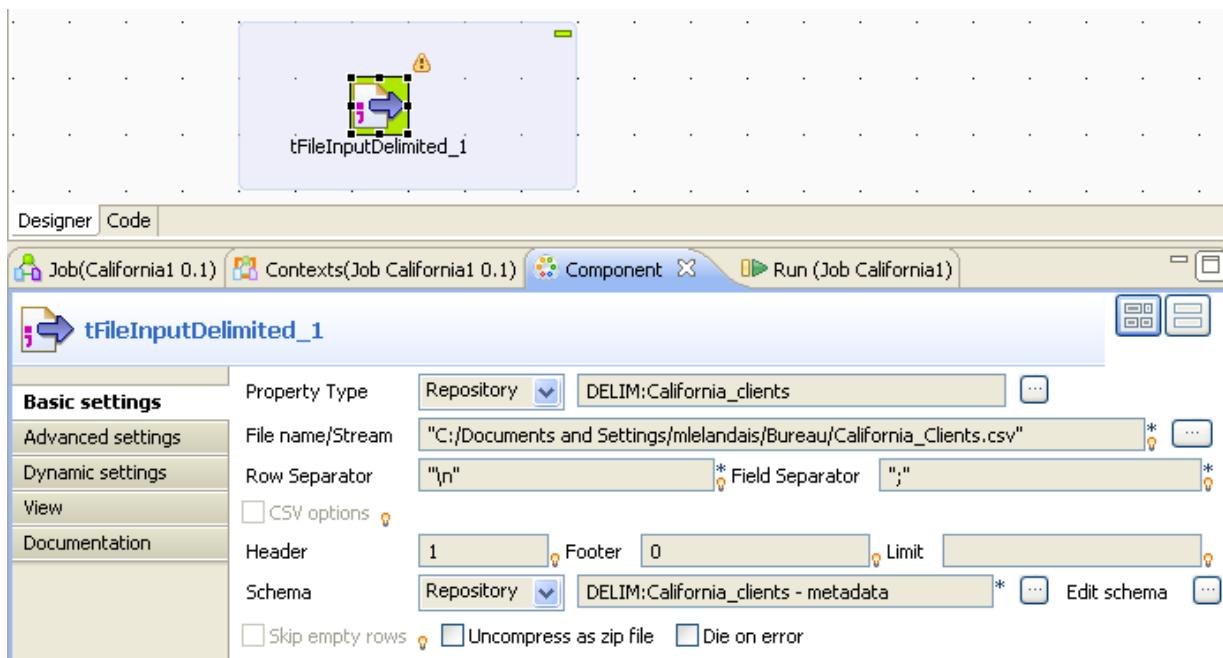
- A l'étape 4, définissez chaque colonne de votre fichier. L'assistant intègre des algorithmes qui essayent de deviner le type et la longueur des données contenues dans les colonnes du fichier en analysant les premières lignes. La description des données (appelé schéma dans **Talend Open Studio**) peut être modifiée à tout moment. Pour ce scénario particulier, ces informations peuvent être gardées telles quelles.

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 métadonnée correspondante dans la liste, **California_clients**. Notez que tous les paramètres sont automatiquement renseignés.



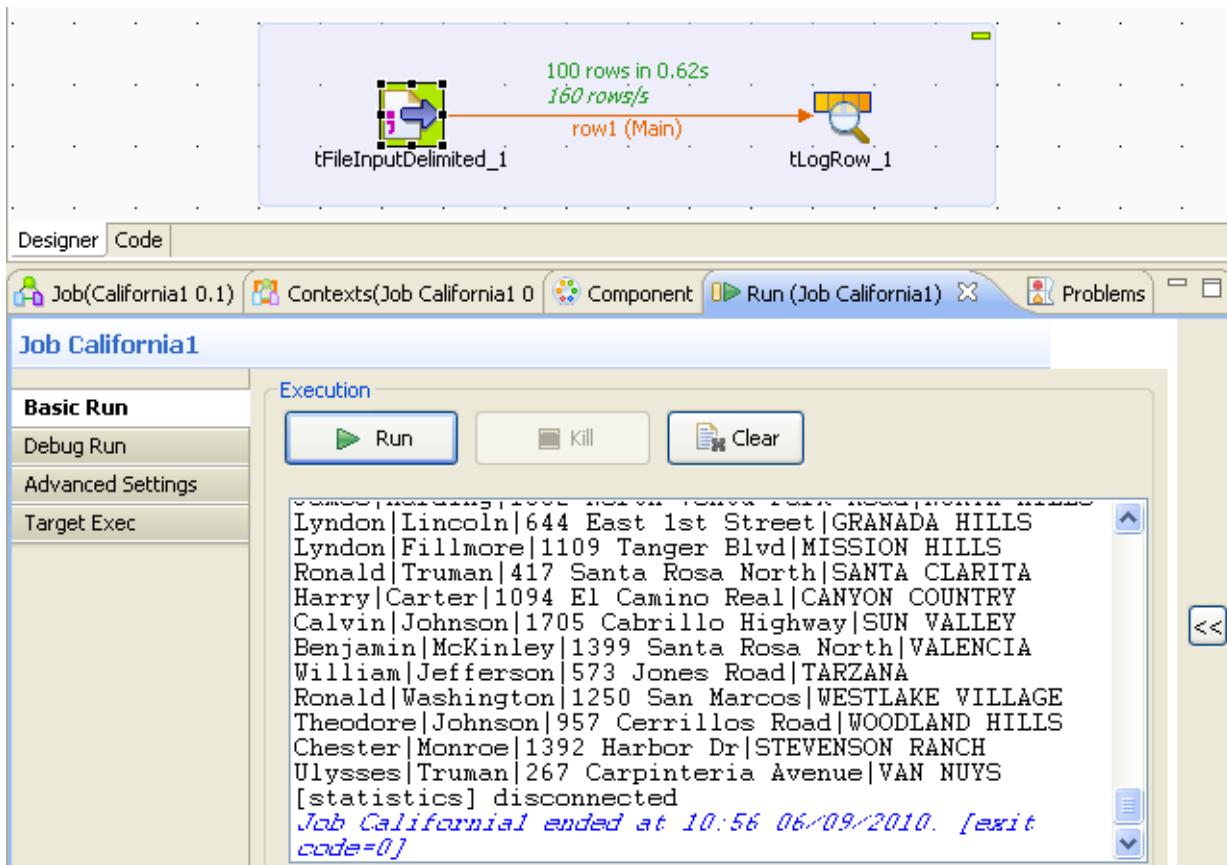
A cette étape, terminez votre flux en envoyant tout simplement les données lues dans le fichier d'entrée vers une sortie standard (StdOut).

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 exécuté. Pour l'exécuter, sélectionnez la vue **Run** dans le bas de la fenêtre.

Activez les statistiques en cochant la case **Statistics** dans l'onglet **Advanced settings** de la vue **Run**, puis exécutez le Job en cliquant sur le bouton **Run**, dans l'onglet **Basic Run**.



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

B.2.2 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 des :

- entrées et sorties multiples,
- recherches de référence (simple, produit cartésien, première et dernière correspondance, etc.),
- jointures (inner join, outer join),
- transformations,
- rejets ,
- etc.

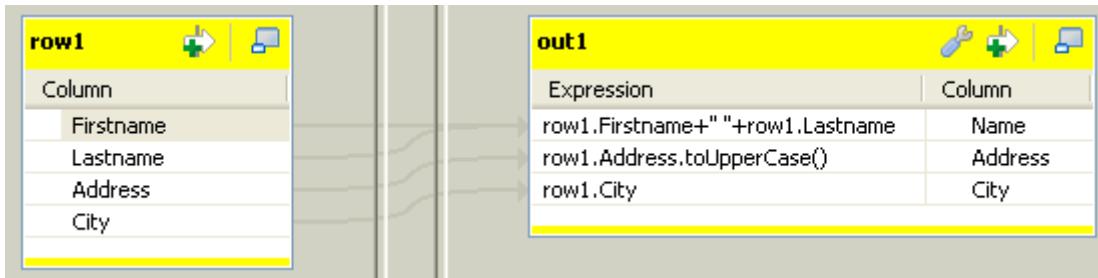
Supprimez la connexion reliant vos deux composants via un clic-droit sur cette connexion et en sélectionnant l'option **Delete**. Puis placez le **tMap** entre les deux autres composants et reliez-le au composant d'entrée comme vous l'avez fait précédemment.

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*).

Déposez les colonnes *FirstName* et *Lastname* de la gauche vers la droite dans la colonne *Name*, comme le montre la capture d'écran suivant. Puis déposez les autres colonnes *Address* et *City* dans leur ligne respective.



Puis effectuez les transformations suivantes sur chaque colonne :

- Changez les données de la colonne *Name* de la manière suivante : `row1.FirstName + " " + row1.LastName`. Cette action concatène les colonnes *Firstname* et *Lastname* dans une seule colonne en respectant la syntaxe Java employée.
- Changez les données de la colonne *Address* de la manière suivante : `row1.Address.toUpperCase()`. Cette action met l'adresse en majuscule.



Ces transformations sont effectuées en Java pour ce Job. Vous utiliserez une syntaxe différente si vous êtes dans un projet Perl.

Puis supprimez la colonne *Lastname* de la table *out1*, et augmentez la longueur des colonnes restantes. Pour cela, cliquez sur l'onglet **Schema Editor** situé en bas de l'éditeur du **[Map Editor]** et procédez comme suit :

Column	Key	Type	N.	Date Pattern...	Length	Precision	Def...	Comm...
Firstname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
Lastname	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
Address	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		26	0		
City	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		22	0		

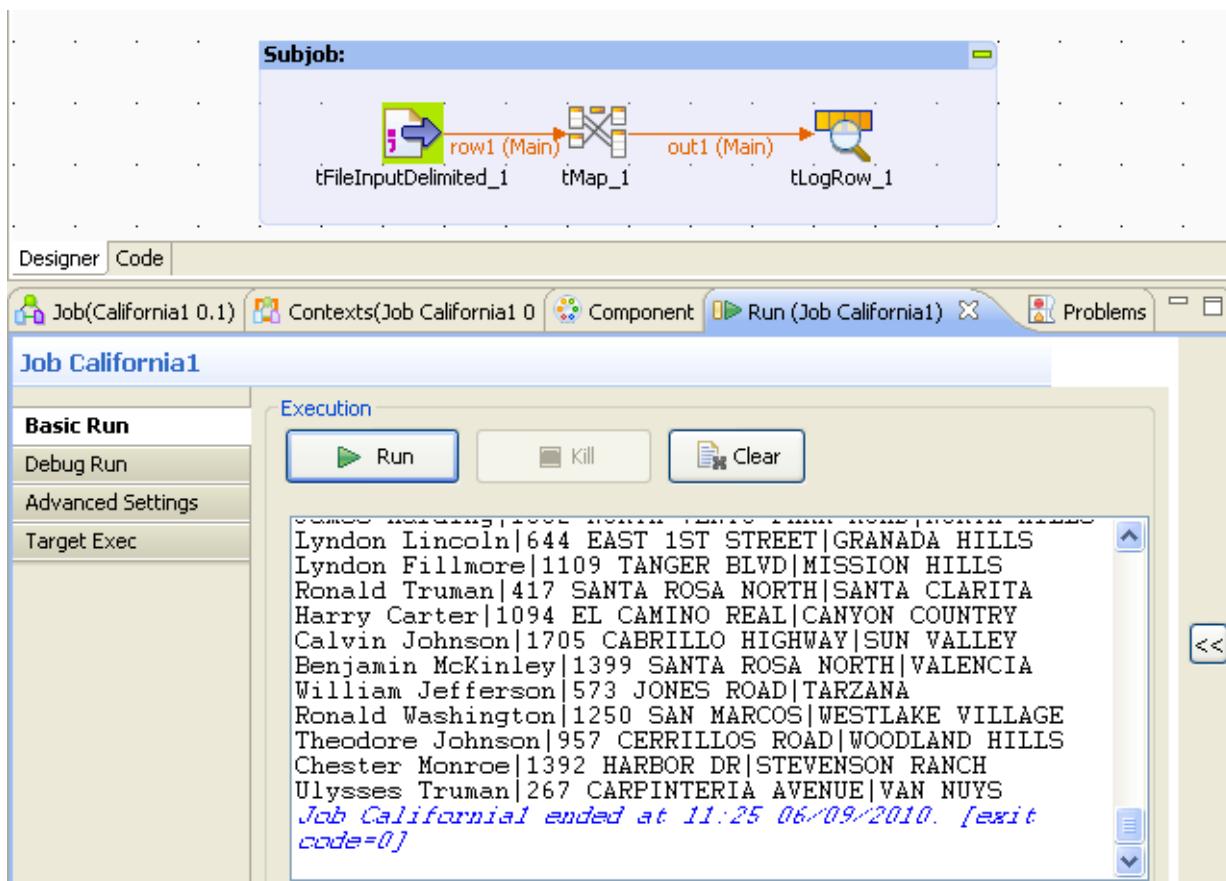
- Sélectionnez la colonne à supprimer du schéma, et cliquez sur l'icône représentant une croix rouge.
- Sélectionnez la colonne dont vous souhaitez augmenter la longueur.
- Saisissez la longueur que vous voulez dans la colonne **Length**. Dans cet exemple, modifiez la longueur de chaque colonne restante en 40.



Comme les noms et prénoms des clients sont concaténés, il est nécessaire d'augmenter la longueur de la colonne *name*, afin de prendre en compte la longueur complète du nom.

Aucune transformation n'est effectuée sur la colonne *City*. Cliquez sur **OK** pour valider les modifications et fermer l'éditeur.

Si vous exécutez votre Job à cette étape (via l'onglet **Run**, comme précédemment), vous remarquerez que les changements que vous avez apportés ont été implémentés.



L'adresse a été mise en majuscule et les prénoms et noms ont été regroupés dans une seule colonne.

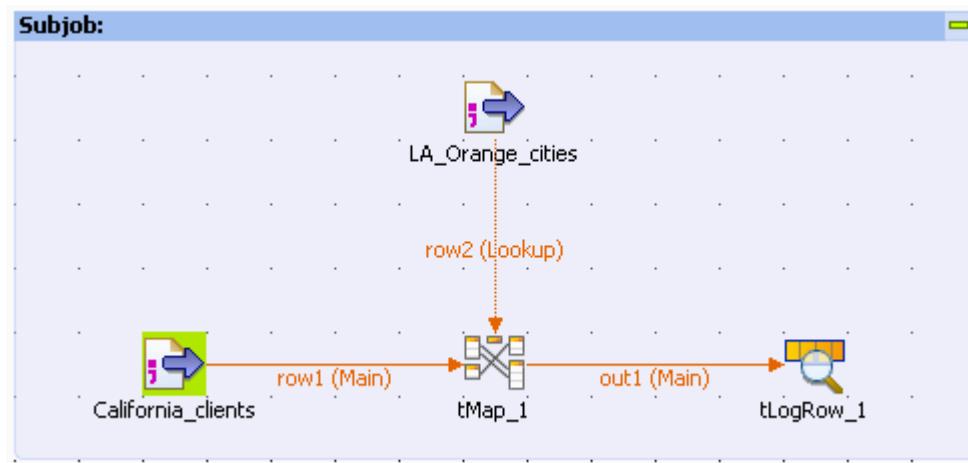
B.2.3 Etape 3 : Définition du fichier de référence, mapping des données de référence, sélection du mode Inner Join

Définissez la métadonnée 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 déposez cette nouvelle métadonnée en haut de l'espace de modélisation, cela créera automatiquement un composant de lecture pointant vers cette métadonnée.

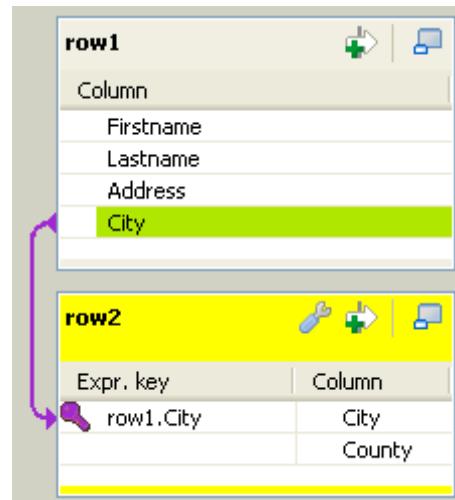
Reliez ce composant au **tMap**.



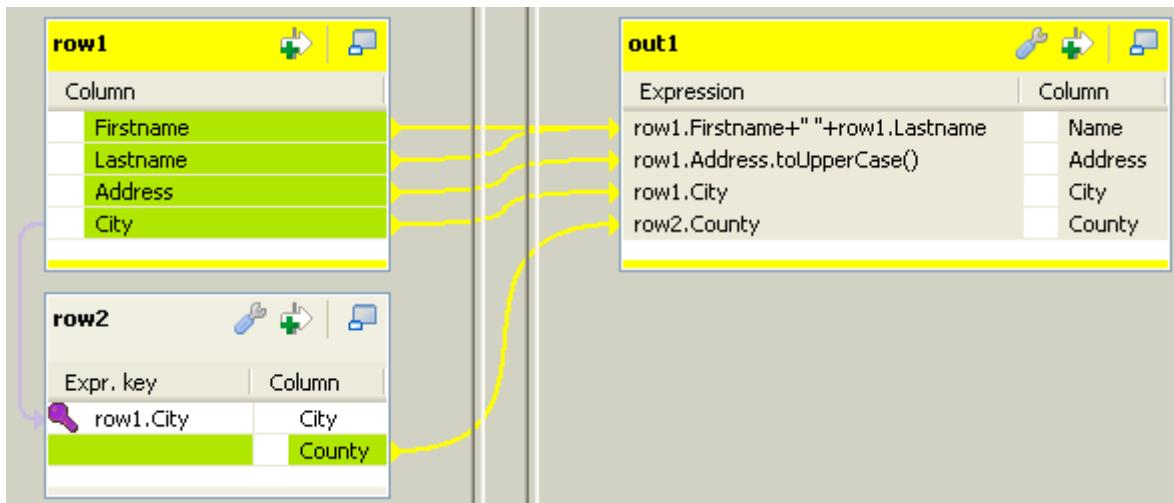
Double-cliquez de nouveau sur le composant **tMap** pour ouvrir son interface. Notez que la table de référence (*row2*) correspondant au fichier *LosAngelesandOrangeCounties.txt*, apparaît à gauche de la fenêtre dans la zone **Input** d'entrée, juste en dessus de votre flux d'entrée principal (*row1*).

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 rapprocher les données (padding, changement de casse, etc.) directement à ce niveau.

Pour établir la jointure, 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 :

```

Execution
Run Kill Clear

Ulysses Taft|1794 GRANDVIEW DRIVE|GARDEN GROVE|ORANGE
Theodore Grant|1895 PACIFIC HWY S|ORANGE|ORANGE
John Johnson|1554 SAN YSIDRO BLVD|NORCO|
Warren Jackson|897 MONROE STREET|VILLA PARK|ORANGE
Warren Van Buren|1633 WESTSIDE FREEWAY|PLACENTIA|ORANGE
Rutherford Eisenhower|448 BURNETT ROAD|CORONA|
Zachary Taft|385 W. RUSSELL ST.|YORBA LINDA|ORANGE
Zachary Pierce|1292 FONTAINE ROAD|VENTURA|
George Garfield|688 VIA REAL|Camarillo|
Warren Taylor|630 NORTH ATHERTON STREET|CARPINTERIA|

```

Comme vous pouvez le voir, la dernière colonne ne contient que les villes des régions 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 Left Outer Join. Si vous souhaitez appliquer un filtre permettant de n'afficher que les données pour lesquelles une correspondance a été trouvée par le **tMap**, cliquez sur le bouton **tMap settings** et sélectionnez **Inner Join** dans la liste **Join Model** sur la table de référence (*row2*).

B.2.4 Etape 4 : Sortie vers une table MySQL

Votre Job fonctionne à merveille ! Pour le finaliser, redirigez le flux de sortie vers une table MySQL.

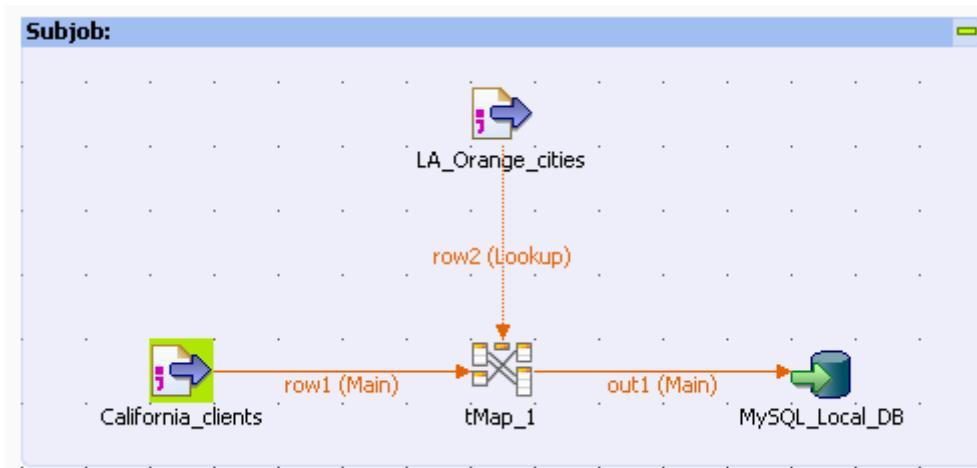
Pour cela, créez tout d'abord une métadonnée décrivant la connexion à la base de données MySQL. Double-cliquez sur **DemoMySQL** dans le répertoire **Metadata > MySQL** du Référentiel (à condition que vous ayez bien importé le projet **Demo**) pour lancer l'assistant Metadata.

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**.

Déposez cette métadonnée à droite de l'espace de modélisation en maintenant la touche **Ctrl** enfoncee 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** (Clic-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**.
- Cliquez sur **Edit Schema** et sur le bouton **Reset DB type** (le bouton en forme de base de données dans la barre d'outils) pour renseigner automatiquement le type de base de données, si nécessaire.

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 400 (bases de données, Webservices, Ftp, etc.) !

D'autres composants, réalisés cette fois par la communauté, sont disponibles sur le site communautaire : talendforge.org.

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



ANNEXEC

Règles d'écriture des SQL Templates

Ce chapitre illustre les règles appliquées à la création des modèles SQL, dont le but est de permettre aux utilisateurs de **Talend Open Studio** de lire, de comprendre et de développer les modèles SQL pour une utilisation plus personnelle de leur Studio.

Ces règles fournissent des informations que vous devez respecter lors de l'écriture d'une instruction modèle, d'une ligne de commentaire et d'une syntaxe.

Ces règles fournissent également des guides pour l'utilisation du code SQL dans des exemples d'utilisation précis, comme par exemple l'accès aux nombreux paramètres définis dans les composants.

C.1 Instructions SQL

Une instruction SQL peut être n'importe quelle instruction SQL valide exécutable par la JDBC liée. Le code des modèles SQL est un groupe d'instructions SQL. Les règles de base pour écrire une instruction SQL dans l'éditeur de modèles SQL sont :

- Une instruction SQL doit se terminer par ;.
- Une instruction SQL peut s'étendre sur plusieurs lignes. Dans ce cas, seule la dernière ligne doit se terminer par ;.

C.2 Lignes de commentaire

Une ligne de commentaire commence par # ou --. Chaque ligne débutant par # ou -- sera ignorée lors de la génération du code.



Aucune exception n'est faite pour les lignes dans la partie du milieu d'une instruction SQL, ou à l'intérieur de la syntaxe <%...%>.

C.3 La syntaxe <%...%>

Cette syntaxe peut s'étendre sur plusieurs lignes. Les points suivants listent ce que vous pouvez faire avec cette syntaxe, et ce à quoi vous devez faire attention.

- Vous pouvez définir de nouvelles variables, utiliser le code logique Java comme if, for et while, et également obtenir les valeurs des paramètres.
Par exemple, si vous souhaitez obtenir le paramètre FILE_NAME, utilisez le code comme suit :

```
<%
String filename = __FILE_NAME__;
%>
```

- Cette syntaxe ne peut être utilisée dans une instruction SQL. Elle doit être utilisée entre deux instructions SQL séparées.

Par exemple, la syntaxe dans le code suivant est valide :

```
#sql sentence
DROP TABLE temp_0;
<%
#loop
for(int i=1; i<10; i++){
%>
#sql sentence
DROP TABLE temp_<%=i %>;
<%
}
%>
#sql sentence
DROP TABLE temp_10;
```

Dans cet exemple, la syntaxe est utilisée entre deux modèles SQL séparés : DROP TABLE temp_0 ; et DROP TABLE temp_<%=i%> ;.

Les instructions SQL ont pour but de supprimer plusieurs tables, en commençant par *temp_0*. Le code entre <% et %> génère un nombre de séquences dans la boucle, afin d'identifier les tables à supprimer, et fermer la boucle après la génération du nombre de séquences.

- A l'intérieur de cette syntaxe, la syntaxe <%=...%> ou </.../> ne doit pas être utilisée.

<%=...%> et </.../> sont également des syntaxes liées aux modèles SQL. Les sections suivantes donnent des informations relatives à ces syntaxes.



Les paramètres auxquels les modèles SQL peuvent accéder grâce à cette syntaxe sont simples. Ils sont souvent utilisés lors de connexions et peuvent être facilement définis dans les composants, par exemple *TABLE_NAME*, *DB_VERSION*, *SCHEMA_TYPE*, etc.

C.4 La syntaxe <%=...%>

Cette syntaxe ne peut s'étendre sur plusieurs lignes, et elle est utilisée dans des instructions SQL. Les points suivants listent ce que vous pouvez faire avec cette syntaxe, et ce à quoi vous devez faire attention.

- Elle peut être utilisée pour générer toute valeur de variable et toute valeur des paramètres existants.
- Les caractères d'espacement sont autorisés après <%=.
- A l'intérieur de la syntaxe, la syntaxe <%...%> ou </.../> ne doit pas être utilisée.

L'instruction dans l'exemple ci-dessous est valide :

```
#sql sentence
DROP TABLE temp_<%=__TABLE_NAME__%>;
```

Le code est utilisé pour supprimer la table définie à l'aide d'un composant associé.

Pour plus d'informations concernant les composants associés aux modèles SQL, consultez le chapitre *SQL Templates, page 275*.

Pour plus d'informations concernant la syntaxe <%...%>, consultez la section *La syntaxe <%...%>, page 316*.

Pour plus d'informations concernant la syntaxe </.../>, consultez la section suivante.



Les paramètres auxquels les modèles SQL peuvent accéder grâce à cette syntaxe sont simples. Ils sont souvent utilisés lors de connexions et peuvent être facilement définis dans les composants, par exemple *TABLE_NAME*, *DB_VERSION*, *SCHEMA_TYPE*, etc.

C.5 La syntaxe </.../>

Cette syntaxe ne peut s'étendre sur plusieurs lignes. Les points suivants listent ce que vous pouvez faire avec cette syntaxe, et ce à quoi vous devez faire attention.

- Elle peut être utilisée pour générer la valeur des paramètres existants. La valeur générée ne doit pas être entourée de guillemets.
- Aucun caractère d'espacement n'est autorisé après </ ou avant />.
- A l'intérieur de cette syntaxe, la syntaxe <%...%> ou <%=...%> ne doit pas être utilisée.

L'instruction écrite dans l'exemple ci-dessous est valide :

```
#sql sentence
DROP TABLE temp_</TABLE_NAME/>;
```

L'instruction accède au paramètre *TABLE_NAME* et supprime la table correspondante.

Pour plus d'informations concernant la syntaxe <%...%>, consultez la section *La syntaxe <%...%>, page 316*.

Pour plus d'informations concernant la syntaxe <%=...%>, consultez la section *La syntaxe <%=...%>, page 317*.

Les sections suivantes présentent un code plus spécifique pour accéder à des paramètres plus complexes.



Les paramètres auxquels les modèles SQL peuvent accéder grâce à cette syntaxe sont simples. Ils sont souvent utilisés lors de connexions et peuvent être facilement définis dans les composants, par exemple *TABLE_NAME*, *DB_VERSION*, *SCHEMA_TYPE*, etc.

C.6 Le code pour accéder aux éléments du schéma des composants

Les éléments du schéma des composants sont présentés dans une liste comprenant le nom des colonnes des schémas (séparés par un point “.”). Ces éléments sont créés et définis par les utilisateurs dans les composants.

Le code ci-dessous propose un exemple d'accès à certains éléments du schéma des composants. Dans cet exemple, le nom de la variable *ELT_METADATA_SHEMA* est utilisé pour obtenir le schéma du composant.

```
<%
String query = "select ";
SCHEMA(__ELT_METADATA_SHEMA__);
for (int i=0; i < __ELT_METADATA_SHEMA__.length ; i++) {
query += (__ELT_METADATA_SHEMA__[i].name + ", ");
}
query += " from " + __TABLE_NAME__;
%>
<%=query %>;
```

Dans cet exemple, selon vos objectifs, le code `__ELT_METADATA_SHEMA__[i].name` peut être remplacé par `__ELT_METADATA_SHEMA__[i].dbType`, `__ELT_METADATA_SHEMA__[i].isKey`, `__ELT_METADATA_SHEMA__[i].length` ou `__ELT_METADATA_SHEMA__[i].nullable` pour accéder aux autres champs des colonnes du schéma.

L'instruction extraite est `SCHEMA(__ELT_METADATA_SHEMA__);`. Dans cette instruction, `ELT_METADATA_SHEMA` est le nom de la variable représentant les paramètres du schéma à extraire. Le nom utilisé dans ce code est simplement un exemple. Vous pouvez changer ce nom en nom d'une autre variable, afin de représenter les paramètres de schéma, selon votre système de nommage. Vérifiez que le nom que vous saisissez n'entrera en conflit avec aucun autre paramètre.

Pour plus d'informations concernant les schémas des composants, consultez *Définition des propriétés d'un composant, page 65*.

C.7 Le code pour accéder aux propriétés de la matrice du composant

Les propriétés de la matrice du composant sont créées et modifiées par les utilisateurs selon les différents objectifs de transformation des données. Ces propriétés sont définies par des paramètres dans des tableaux. Par exemple, les paramètres *operation* ou *groupby* peuvent être définis par les utilisateurs dans le composant **tELTAggregate**.

Vous pouvez accéder à ces paramètres de transformation, naturellement plus flexibles et complexes, de deux façons différentes.

- L'approche </ . . . />.

</ . . . /> est l'une des syntaxes utilisées par les modèles SQL. Cette approche nécessite souvent du code dur pour chaque paramètre à extraire.

Par exemple, un nouveau paramètre est créé par l'utilisateur et nommé *NEW_PROPERTY*. Si vous souhaitez y accéder en utilisant </*NEW_PROPERTY*/>, le code ci-dessous est requis.

```
else if (paramName.equals("NEW_PROPERTY")) {
    List<Map<String, String>> newPropertyValue = (List<Map<String, String>>)
        ElementParameterParser.getObjectValue(node, "__NEW_PROPERTY__");
    for (int ii = 0; ii < newPropertyValue.size(); ii++) {
        Map<String, String> newPropertyMap = newPropertyValue.get(ii);
        realValue += ...;//append generated codes
        ....
    }
}
```

- L'approche EXTRACT(__GROUPBY__);.

Le code ci-dessous montre la seconde méthode d'accès au paramètre de transformation (*GROUPBY*).

```
<%
String query = "insert into " + __TABLE_NAME__ + "(id, name, date_birth) select sum(id), name,
date_birth from cust_teradata group by";
EXTRACT(__GROUPBY__);
for (int i=0; i < __GROUPBY_LENGTH__ ; i++) {
query += (__GROUPBY_INPUT_COLUMN__[i] + " ");
}
%>
<%=query %>;
```

Lors de l'encodage des instructions, respectez les règles comme suit :

- L'instruction extraite doit utiliser EXTRACT(__GROUPBY__);. Les majuscules sont requises, et les caractères d'espacement ne sont pas autorisés. Cette instruction doit être utilisée entre <% et %>.
- Utilisez le code __GROUPBY_LENGTH__, dans lequel le nom du paramètre, suivi de _LENGTH, permet d'obtenir le numéro de ligne des paramètres du tableau *GROUPBY* définis dans la zone **Groupby** de la vue **Component**. Il peut être utilisé entre <% et %> ou <%= et %>.
- Utilisez le code __GROUPBY_INPUT_COLUMN__[i] afin d'extraire les valeurs des paramètres. Il peut être utilisé entre <% et %> ou entre <%= et %>.

- Pour accéder aux paramètres correctement, n'utilisez pas le même préfixe pour le nom de plusieurs paramètres. Par exemple, dans un composant, évitez de définir deux paramètres avec le nom PARAMETER_NAME et PARAMETER_NAME_2, car plusieurs préfixes semblables produisent des erreurs lors de la génération du code.



ANNEXED

Routines système

Cette annexe donne un aperçu d'une sélection des routines les plus couramment employées, ainsi que des exemples d'utilisation. Dans cette annexe, les routines sont présentées dans l'ordre dans lequel elles apparaissent dans le **Repository**. Elles sont rassemblées selon leur type et chaque type est détaillé dans une section différente.

Pour plus d'informations concernant la définition des routines, l'accès aux routines système ou la gestion des routines système ou personnalisées, consultez le chapitre *Gestion des routines*.

Avant de commencer un processus métier (Job), vous devez vous être familiarisé avec l'interface graphique de **Talend Open Studio**. Pour plus d'informations, consultez *Interface de Talend Open Studio, page 289*.

D.1 Routines de type Numeric

Les routines numériques permettent de retourner des nombres entiers ou décimaux afin de les utiliser comme paramètres dans un ou plusieurs composant(s) d'un Job, par exemple pour ajouter un identifiant numérique.

Pour accéder à ces routines, double-cliquez sur la classe **Numeric** du dossier **system**. La classe **Numeric** contient plusieurs routines, notamment : séquentielle (sequence), aléatoire (random) et décimale (convertImpliedDecimalFormat) :

Routine	Description	Syntaxe
<i>sequence</i>	Retourne un identifiant numérique incrémenté.	Numeric.sequence("Nom du Paramètre", valeur de début, valeur d'incrément)
<i>resetSequence</i>	Crée une séquence si elle n'existe pas et attribue une nouvelle valeur de début.	Numeric.resetSequence(Identifiant de la séquence, valeur de début)
<i>removeSequence</i>	Supprime une séquence.	Numeric.RemoveSequence(Identifiant de la séquence)
<i>random</i>	Retourne un entier au hasard entre les valeurs minimale et maximale.	Numeric.random(valeur limite de début, valeur limite de fin)
<i>convertImpliedDecimalFormat</i>	Retourne un décimal à l'aide d'un modèle décimal implicite.	Numeric.convertImpliedDecimalFormat("Format Cible", valeur à convertir)

D.1.1 Exemple de création séquentielle

Vous pouvez tester simplement la routine **sequence**, à l'aide d'un composant **tJava** par exemple, pour contrôler la création d'un identifiant incrémenté automatiquement :

```
System.out.println(Numeric.sequence("s1",1,1));
System.out.println(Numeric.sequence("s1",1,1));
```

L'identifiant numérique est généré et incrémenté automatiquement par la routine :

```
[statistics] connecting to socket on port 3360
[statistics] connected
1
2
```

D.1.2 Exemple de conversion décimale implicite

Vous pouvez tester simplement la routine **convertImpliedDecimalFormat** à l'aide d'un composant **tJava**, pour vérifier la conversion d'un nombre décimal implicite :

```
System.out.println(Numeric.convertImpliedDecimalFormat("9V99","123"));
```

La valeur saisie en paramètre est convertie automatiquement par la routine selon le format décimal implicite fourni :

```

1.23
[statistics] disconnected
Job test_routine ended at 14:12 04/02/2010. [exit code=0]

```

D.2 Routines de type Relational

Les routines relationnelles permettent de vérifier une affirmation basée sur des booléens.

Pour accéder à ces routines, double-cliquez sur la classe **Relational** du dossier **system**. La classe **Relational** contient plusieurs routines notamment :

Routine	Description	Syntaxe
ISNULL	vérifie si la variable donnée est de valeur nulle.	Relational.ISNULL(variable à vérifier)

Vous pouvez tester une routine Relational, comme la routine **ISNULL**, à l'aide d'un composant **tJava** par exemple :

```
System.out.println(Relational.ISNULL(null));
```

Dans cet exemple, le résultat de la vérification s'affiche dans la vue **Run** :

```

Starting job test_routine at 14:14 04/02/2010.

[statistics] connecting to socket on port 3375
[statistics] connected
true
[statistics] disconnected
Job test_routine ended at 14:14 04/02/2010. [exit code=0]

```

D.3 Routines de type StringHandling

Les routines de traitement des chaînes de caractères permettent d'effectuer différents types d'opérations et de vérifications, basées sur des méthodes Java, sur des expressions alphanumériques.

Pour accéder à ces routines, double-cliquez sur la classe **StringHandling** du dossier **system**. La classe **StringHandling** contient notamment les routines suivantes :

Routine	Description	Syntaxe
ALPHA	vérifie si l'expression est triée par ordre alphabétique. Retourne le booléen true si l'ordre alphabétique est vérifié, et false, dans le cas inverse.	StringHandling.ALPHA("chaîne à vérifier")
IS_ALPHA	vérifie si l'expression ne contient que des caractères alphabétiques. Retourne le booléen true si c'est le cas, et false dans le cas inverse.	StringHandling.IS_ALPHA("chaîne à vérifier")

Routine	Description	Syntaxe
<i>CHANGE</i>	remplace un élément d'une chaîne de caractères par l'élément de remplacement défini et retourne la nouvelle chaîne.	<code>StringHandling.CHANGE("chaîne à vérifier", "chaîne à remplacer", "chaîne de remplacement")</code>
<i>COUNT</i>	retourne le nombre d'occurrences d'une sous-chaîne dans une chaîne de caractères.	<code>StringHandling.COUNT("chaîne à vérifier", "chaîne à compter")</code>
<i>DOWNCASE</i>	convertit toutes les majuscules d'une expression en minuscules et retourne la nouvelle chaîne.	<code>StringHandling.DOWNCASE("chaîne à convertir")</code>
<i>UPCASE</i>	convertit toutes les minuscules d'une expression en majuscules et retourne la nouvelle chaîne.	<code>StringHandling.UPCASE("chaîne à convertir")</code>
<i>DQUOTE</i>	entoure une expression de guillemets doubles.	<code>StringHandling.DQUOTE("chaîne à traiter")</code>
<i>INDEX</i>	retourne la position, dans une chaîne de caractères, du premier caractère de la sous-chaîne recherchée. Si la sous-chaîne recherchée n'existe pas dans la chaîne, -1 est retourné.	<code>StringHandling.INDEX("chaîne à vérifier", "sous-chaîne recherchée")</code>
<i>LEFT</i>	retourne une sous-chaîne correspondant aux n premiers caractères d'une chaîne de caractères.	<code>StringHandling.LEFT("chaîne à vérifier", nombre de caractères)</code>
<i>RIGHT</i>	retourne une sous-chaîne correspondant aux n derniers caractères d'une chaîne de caractères.	<code>StringHandling.RIGHT("chaîne à vérifier", nombre de caractères)</code>
<i>LEN</i>	retourne la longueur d'une chaîne de caractères.	<code>StringHandling.LEN("chaîne à vérifier")</code>
<i>SPACE</i>	retourne une chaîne faite du nombre de caractères vides indiqué.	<code>StringHandling.SPACE(nombre d'espaces à créer)</code>
<i>SQUOTE</i>	entoure une expression de guillemets simples.	<code>StringHandling.SQUOTE("chaîne à traiter")</code>
<i>STR</i>	retourne un caractère répété le nombre de fois indiqué.	<code>StringHandling.STR('caractère à générer', nombre de répétition)</code>
<i>TRIM</i>	supprime les espaces et les tabulations en début et fin d'une chaîne de caractères et retourne la nouvelle chaîne.	<code>StringHandling.TRIM("chaîne à traiter")</code>
<i>BTRIM</i>	supprime tous les espaces et les tabulations après le dernier caractère non vide d'une chaîne de caractères et retourne la nouvelle chaîne.	<code>StringHandling.BTRIM("chaîne à traiter")</code>
<i>FTRIM</i>	supprime tous les espaces et les tabulations jusqu'au premier caractère non vide d'une chaîne de caractères et retourne la nouvelle chaîne.	<code>StringHandling.FTRIM("chaîne à traiter")</code>

D.3.1 Exemple de vérification de tri alphabétique

Vous pouvez tester simplement la routine **ALPHA**, à l'aide d'un composant **tJava** par exemple, pour vérifier si la chaîne est triée dans l'ordre alphabétique :

```
System.out.println(StringHandling.ALPHA("abcdefg"));
```

La vérification retourne un booléen.

```
Starting job test_routine at 14:29 04/02/2010.
```

```
[statistics] connecting to socket on port 3469
[statistics] connected
true
```

D.3.2 Exemple de vérification de type alphabétique

Vous pouvez tester simplement la routine **IS_ALPHA**, à l'aide d'un composant **tJava** par exemple, pour vérifier si la chaîne est de type alphabétique ou non :

```
System.out.println(StringHandling.IS_ALPHA("ab33cd"));
```

La vérification retourne un booléen.

```
Starting job routine1 at 11:45 23/02/2010.
```

```
[statistics] connecting to socket on port 3892
[statistics] connected
false
[statistics] disconnected
Job routine1 ended at 11:45 23/02/2010. [exit code=0]
```

D.3.3 Exemple de remplacement de chaîne

Vous pouvez tester simplement la routine **CHANGE**, à l'aide d'un composant **tJava** par exemple, pour contrôler le remplacement d'une chaîne par une autre :

```
System.out.println(StringHandling.CHANGE("hello
world!","world","guy"));
```

La vérification retourne un booléen.

```
hello guy!
```

D.3.4 Exemple de vérification de chaîne

Vous pouvez tester simplement la routine **INDEX**, à l'aide d'un composant **tJava** par exemple, pour vérifier si la chaîne contient ou pas un caractère ou une chaîne de caractères :

```
System.out.println(StringHandling.INDEX("hello world!", "hello"));
System.out.println(StringHandling.INDEX("hello world!", "world"));
System.out.println(StringHandling.INDEX("hello world!", "!="));
System.out.println(StringHandling.INDEX("hello world!", "?"));
```

La routine retourne un entier qui correspond à la position du premier caractère de la chaîne recherchée, ou retourne -1 si la chaîne recherchée n'a pu être trouvée :

```
Starting job routine1 at 15:47 24/02/2010.  
[statistics] connecting to socket on port 4027  
[statistics] connected  
0  
6  
11  
-1  
[statistics] disconnected  
Job routine1 ended at 15:47 24/02/2010. [exit code=0]
```

D.3.5 Exemple de calcul de longueur de chaîne

Vous pouvez tester simplement la routine **LEN**, à l'aide d'un composant **tJava** par exemple, pour contrôler la longueur d'une chaîne :

```
System.out.println(StringHandling.LEN("hello world!"));
```

La vérification retourne un entier correspondant à la longueur de la chaîne, y compris les espaces et caractères vides :

```
12
```

D.3.6 Exemple de nettoyage d'espaces inutiles

Vous pouvez tester simplement la routine **FTRIM**, à l'aide d'un composant **tJava** par exemple, afin de retirer, entre autres, une tabulation en tête de chaîne :

```
System.out.println(StringHandling.FTRIM("    Hello world  !));
```

La vérification retourne la chaîne nettoyée des espaces/tabulations présents en début de chaîne.

```
Starting job routine1 at 16:14 24/02/2010.  
[statistics] connecting to socket on port 3790  
[statistics] connected  
Hello world !  
[statistics] disconnected  
Job routine1 ended at 16:14 24/02/2010. [exit code=0]
```

D.4 Routines de type TalendDataGenerator

Les routines de génération de données factices sont des fonctions qui permettent de générer des ensembles de données de test. Elles se basent sur les listes (factices) de noms, prénoms, adresses, villes et Etats fournies par **Talend**. Ces routines sont généralement utilisées au moment du développement des Jobs, à l'aide d'un **tRowGenerator** par exemple, pour éviter d'utiliser des données de production ou de l'entreprise.

Pour accéder à ces routines, double-cliquez sur la classe **TalendDataGenerator** du dossier **system** :

Routine	Description	Syntaxe
<code>getFirstName</code>	retourne un prénom pris aléatoirement dans une liste factice de prénoms américains courants.	<code>TalendDataGenerator.getFirstName()</code>
<code>getLastName</code>	retourne un nom de famille pris aléatoirement dans une liste factice de noms de famille américains courants.	<code>TalendDataGenerator.getLastName()</code>
<code>getUsStreet</code>	retourne une adresse prise aléatoirement dans une liste factice de rues américaines courantes.	<code>TalendDataGenerator.getUsStreet()</code>
<code>getUsCity</code>	retourne une ville prise aléatoirement dans une liste de villes connues des Etats-Unis.	<code>TalendDataGenerator.getUsCity()</code>
<code>getUsState</code>	retourne le nom d'un Etat pris aléatoirement dans une liste des Etats américains.	<code>TalendDataGenerator.getUsState()</code>
<code>getUsStateId</code>	retourne le code pris aléatoirement dans une liste d'abréviations correspondant aux Etats américains.	<code>TalendDataGenerator.getUsStateId()</code>



Aucun paramètre d'entrée n'est attendu car la liste des données factices est fournie par **Talend**.

Vous pouvez personnaliser les données factices en modifiant les routines de génération de données. Pour plus d'informations concernant la personnalisation des routines, consultez *Personnalisation des routines système, page 267*.

D.4.1 Exemple de génération de données factices

Vous pouvez tester simplement les différentes fonctions de génération de données aléatoires, telles que `getFirstName`, `getLastName`, `getUSCity`, etc. à l'aide d'un composant **tJava**, pour tester, par exemple, la création d'une liste de données d'un client factice :

```
System.out.println(TalendDataGenerator.getFirstName());
System.out.println(TalendDataGenerator.getLastName());
System.out.println(TalendDataGenerator.getUsCity());
System.out.println(TalendDataGenerator.getUsState());
System.out.println(TalendDataGenerator.getUsStateId());
System.out.println(TalendDataGenerator.getUsStreet());
```

L'ensemble des données prises aléatoirement dans les listes de données factices est affiché dans la vue **Run** :

```

Starting job test_routine at 14:44 04/02/2010.
[statistics] connecting to socket on port 3907
[statistics] connected
Jimmy
Arthur
Des Moines
Wyoming
UT
Milpas Street
[statistics] disconnected
Job test_routine ended at 14:44 04/02/2010. [exit code=0]

```

D.5 Routines de type TalendDate

Les routines de traitement de dates permettent d'effectuer différents types d'opération et de vérification sur le format des expressions de type Date.

Pour accéder à ces routines, double-cliquez sur la classe **TalendDate** du dossier **system** :

Routine	Description	Syntaxe
<i>formatDate</i>	retourne une expression de type date formatée selon le modèle date spécifié.	TalendDate.formatDate("format de date - ex: yyyy-MM-dd HH:mm:ss", Date() à formater)
<i>isDate</i>	vérifie si l'expression est de type Date et correspond au modèle spécifié. Retourne le booléen true si c'est le cas, et false dans le cas inverse.	TalendDate.isDate(Date() à vérifier, "format de date cible - ex: yyyy-MM-dd HH:mm:ss")
<i>compareDate</i>	compare tout ou partie de deux dates, selon le modèle date si spécifié. Retourne 0 si les dates sont identiques, 1 si la première date est ultérieure à la deuxième et -1 si elle est antérieure à la deuxième.	TalendDate.compareDate(Date date1, Date date2, "format à comparer - ex: yyyy-MM-dd")
<i>addDate</i>	ajoute n jours, n mois, n heures, n minutes ou n secondes à une Date Java et retourne la nouvelle date. Le paramètre de format de donnée Date est : "yyyy", "MM", "dd", "HH", "mm", "ss" ou "SSS".	TalendDate.addDate("String date initiale", "format Date - ex: yyyy/MM/dd", entier n,"format de la donnée à ajouter - ex:yyyy").
<i>diffDate</i>	retourne la différence entre deux dates, en nombre de jours, mois, années, minutes ou secondes selon le paramètre de comparaison spécifié. Retourne 0 si aucune différence n'est détectée.	TalendDate.diffDate(Date1(), Date2(), "format de données à comparer - ex:yyyy")
<i>getFirstDayOfMonth</i>	change le jour d'une date en premier jour du mois courant et retourne la nouvelle date.	TalendDate.getFirstDayMonth(Date)
<i>getLastDayOfMonth</i>	change le jour d'une date en dernier jour du mois courant et retourne la nouvelle date.	TalendDate.getLastDayMonth(Date)

Routine	Description	Syntaxe
<i> setDate</i>	modifie une partie de la date en entrée par un entier spécifié et basé sur le format spécifié.	TalendDate.setDate(Date, entier n, "format de la donnée à changer - ex:yyyy")
<i> formatDateLocale</i>	transforme une date en une chaîne de caractères de type date/heure selon le modèle et la locale spécifiés. Retourne la nouvelle date formatée.	TalendDate.formatDateLocale("format cible", java.util.Date date, "code de la langue ou du pays")
<i> parseDate</i>	transforme une chaîne de caractères en Date. Retourne une date formatée en standard.	TalendDate.parseDate("format date de la chaîne à parser", "Chaîne de caractère au format date à parser")
<i> parseDateLocale</i>	parse une chaîne de caractères correspondant à un modèle spécifié, et en extrait une date. Retourne une date formatée selon la locale spécifiée.	TalendDate.parseDateLocale("format date de la chaîne à parser", "Chaîne de caractère au format date à parser", "code de la langue ou du pays")
<i> getDate</i>	retourne les date et heure courantes dans le format spécifié (optionnel). Cette chaîne peut contenir des chaînes de caractères fixes, ou des variables liées à la date. Par défaut, le format de la chaîne de caractères attendu est DD/MM/CCYY.	TalendDate.getDate("Format de chaîne de caractère - ex: CCYY-MM-DD")
<i> getCurrentDate</i>	retourne la date courante. Aucun paramètre d'entrée n'est attendu.	TalendDate.getCurrentDate()
<i> getRandomDate</i>	retourne une date aléatoire, au format ISO.	TalendDate.getRandomDate("Cahîne de caractère de type Date, de ", String maxDate)

D.5.1 Exemple de formatage d'une date

Vous pouvez tester simplement la routine **formatDate**, à l'aide d'un composant **tJava** par exemple, pour vérifier qu'une expression de type date est dans le format spécifié :

```
System.out.println(TalendDate.formatDate("dd-MM-yyyy", new Date()));
```

La date du jour est instanciée par la fonction Java `new date()` et s'affiche dans la vue **Run** :

```
Starting job routine1 at 17:28 25/02/2010.  
2010-02-25 17:28:07  
Job routine1 ended at 17:28 25/02/2010. [exit code=0]
```

D.5.2 Exemple de vérification d'une date

Vous pouvez tester simplement la routine **isDate**, à l'aide d'un composant **tJava** par exemple, pour vérifier que une expression de type date est dans le format spécifié :

```
System.out.println(TalendDate.isDate("2010-02-09 00:00:00", "yyyy-MM-dd  
HH:mm:ss"));
```

Un booléen s'affiche dans la vue **Run** :

```
Starting job routine1 at 17:36 25/02/2010.  
true  
Job routine1 ended at 17:36 25/02/2010. [exit code=0]
```

D.5.3 Exemple de comparaison de dates

Vous pouvez tester simplement la routine **formatDate**, à l'aide d'un composant **tJava** par exemple, pour vérifier si la date du jour est ultérieure à une date spécifiée, selon le format défini :

```
System.out.println(TalendDate.compareDate(new Date(),  
TalendDate.parseDate("yyyy-MM-dd", "2010/11/24"), "yyyy-MM-dd"));
```

La date du jour est instanciée par la fonction Java `new Date()` et la valeur `-1` s'affiche dans la vue **Run** pour indiquer que la date du jour est en fait antérieure à la date de référence :

```
Starting job routine1 at 18:09 25/02/2010.  
-1  
Job routine1 ended at 18:09 25/02/2010. [exit code=0]
```

D.5.4 Exemple de configuration de date

Vous pouvez tester simplement la routine **setDate**, à l'aide d'un composant **tJava**, pour changer l'année de la date courante par exemple :

```
System.out.println(TalendDate.formatDate("yyyy/MM/dd HH:mm:ss", new  
Date()));  
System.out.println(TalendDate.setDate(new Date(), 2011, "yyyy"));
```

La date courante suivie de la nouvelle date configurée s'affiche dans la vue **Run** :

```
Starting job routine1 at 18:03 26/02/2010.  
2010/02/26 18:03:14  
Sat Feb 26 18:03:14 CET 2011  
Job routine1 ended at 18:03 26/02/2010. [exit code=0]
```

D.5.5 Exemple de parsage de date

Vous pouvez tester simplement la routine **parseDate**, à l'aide d'un composant **tJava**, pour mettre une chaîne de type date au format Date par exemple :

```
System.out.println(TalendDate.parseDate("yyyy-MM-dd HH:mm:ss",  
"1979-10-20 19:00:59"));
```

La chaîne de caractères est transformée en Date et la date est renvoyée :

Starting job routine1 at 11:58 01/03/2010.

Sat Oct 20 19:00:59 CET 1979

Job routine1 ended at 11:58 01/03/2010. [exit code=0]

D.5.6 Exemple de formatage de la date courante

Vous pouvez tester simplement la routine **getDate**, à l'aide d'un composant **tJava**, pour récupérer la date courante et la formater selon un modèle spécifié, par exemple :

```
System.out.println(TalendDate.getDate("CCYY-MM-DD"));
```

La date courante est renournée selon le format spécifié (optionnel) :

Starting job routine1 at 10:58 02/03/2010.

2010-03-02

Job routine1 ended at 10:58 02/03/2010. [exit code=0]

D.6 Routines de type TalendString

Les routines de traitement de chaînes de caractères **Talend** permettent d'effectuer diverses opérations sur des expressions alphanumériques.

Pour accéder à ces routines, double-cliquez sur la classe **TalendString** du dossier **system**. La classe **TalendString** contient notamment les routines suivantes :

Routine	Description	Syntaxe
<i>replaceSpecialCharForXML</i>	retourne une chaîne de caractères où les caractères spéciaux (ex : <, >, &...) ont été remplacés par des caractères XML équivalents.	TalendString.replaceSpecialCharForXML("chaîne de caractères contenant les caractères spéciaux - ex: Thelma & Louise")
<i>checkCDATAForXML</i>	identifie les chaînes de caractères commençant par <! [CDATA[et se terminant par]]> en tant que XML et les retourne sans modification. Transforme les chaînes identifiées comme non-XML sous une forme compatible XML et les retourne ainsi formatées.	TalendString.checkCDATAForXML("chaîne de caractères à parser")
<i>talendTrim</i>	parse la chaîne de caractères en entrée et en retire le/les caractères de remplissage en début ou en fin de chaîne selon la valeur d'alignement spécifiée: -1 pour les caractères de remplissage de fin de chaîne, 1 pour ceux de début de chaîne et 0 pour les deux. Puis retourne la chaîne nettoyée.	TalendString.talendTrim("chaîne de caractères à parser", "caractère de remplissage à retirer", position du caractère)

Routine	Description	Syntaxe
<i>removeAccents</i>	enlève les accents d'une chaîne de caractères et retourne cette chaîne non accentuée.	TalendString.removeAccents("Chaîne de caractères")
<i>getAsciiRandomString</i>	génère une chaîne de caractères aléatoire, du nombre de caractères spécifiés.	TalendString.getAsciiRandomString(entier de la longueur de chaîne)

D.6.1 Exemple de formatage XML d'une chaîne

Vous pouvez tester simplement la routine **replaceSpecialCharForXML**, à l'aide d'un composant **tJava**, pour formater une chaîne de caractères pour le XML, par exemple :

```
System.out.println(TalendString.replaceSpecialCharForXML("Thelma & Louise"));
```

Dans cet exemple, le caractère "&" est remplacé pour être intelligible en XML :

```
Starting job routine1 at 15:48 02/03/2010.  
Thelma & Louise  
Job routine1 ended at 15:48 02/03/2010. [exit code=0]
```

D.6.2 Exemple de trimming d'une chaîne

Vous pouvez tester simplement la routine **talendTrim**, à l'aide d'un composant **tJava**, pour retirer des caractères de remplissage en début et en fin de chaîne, par exemple :

```
System.out.println(TalendString.talendTrim("**talend open studio****",  
'*', -1));  
System.out.println(TalendString.talendTrim("**talend open studio****",  
'*', 1));  
System.out.println(TalendString.talendTrim("**talend open studio****",  
'*', 0));
```

Les caractères étoiles sont retirés alternativement en début puis en fin de chaîne, et enfin des deux côtés :

```
Starting job routine1 at 14:19 02/03/2010.  
**talend open studio  
talend open studio****  
talend open studio  
Job routine1 ended at 14:19 02/03/2010. [exit code=0]
```

D.6.3 Exemple de désaccentuation d'une chaîne

Vous pouvez tester simplement la routine **removeAccents**, à l'aide d'un composant **tJava**, pour remplacer les caractères accentués, par exemple :

```
System.out.println(TalendString.removeAccents("sâcrebleü!"));
```

Les caractères avec accent sont remplacés par des caractères sans accent :

```
Starting job routine1 at 16:02 02/03/2010.
sacrebleu!
Job routine1 ended at 16:02 02/03/2010. [exit code=0]
```

A	
Activer/Désactiver	130
Aperçu du code	300
Assistants	
Connexion MDM	245
DB connection	190
FTP	197
JDBC	200
Salesforce	237
SAS	203
Schéma file delimited	205
Schéma file Excel	226
Schéma file LDIF	230
Schéma file positional	210
Schéma file Regex	212
Schéma file XML	214
Schéma générique	241
schéma LDAP	232
Schéma Web Service	242
B	
Barre d'outils	293
Breakpoint, Pause	74
Built-in	
Variable	91
Business Model	44, 45, 295
Affectation	54
Créer	45
Ouvrir	45
Business Modeler	46
C	
Centralisation des contextes	91
Centralisation des éléments	91
Centraliser les contextes	91
Centraliser les éléments	91
Changer de workspace	12
Chercher un Job	153
Clé	166
Clé de hachage	166
Clé primaire	166
Code & Structure	121
Code Viewer	121, 122, 300
Component	83
Component, Composant	61
Composant	
Advanced Settings	69
Advanced settings	69
Basic settings	65
Externe	22
onglet view	71
Paramétrtrer	65
Start	118
Composant de début	118
Composant de début, Start component	130
composant post-job	115
Composant pré-job	115
Composant Start	118
Composant, Component	61
Composants	
connexion main	84
connexions filter	85
connexions lookup	84
connexions rejects	85
connexions row	84
connexions uniques/duplicates	86
documentation	72
Dynamic settings	69
Configurer les préférences	21
Connexion	
Duplicates	86
Iterate	86
Link	88
Lookup	84
Main	84
Output	84, 85
Référentiel local	10
Rejet	84
Rejets	85
Row	84
Trigger	86
Uniques	86
Connexion des composants	64
Connexion MDM Input	248
Connexion MDM Output	253
Connexion MDM Receive	258
Contexte, Context	73
Corbeille, Recycle bin	55
D	
Debug mode	74
Déclenchement	86
Déployer un Job	157
Documentation	72
documenter un Job	156
Duplicates	86

E	
Edit Schema	67
Elément	
Importer	131
Entrée/sortie multiples	86
Erreurs & avertissements	119
Espace de modélisation, Job Designer	296
Espace de travail	296
Exécuter le Job, Run Job	297
Exécution	103
Exporter	
exemple	137
Exporter des métadonnées	262
Exporter un Job	
JBoss ESB	140
Petals ESB	141
Exporter un job	
autonomous	135
Service Web	136
Exporter vers l'ESB	142
Expression editor	
accéder	174
exemple d'utilisation	175
Expressions régulières	213
External modules	111
F	
Fenêtre principale	290
Fichier Delimited	205
Fichier LDIF	230
Fichier Positional	210
Fichier Regex	212
Fichier XML	214
Loop limit	218
File XML	
Loop limit	251, 260
Filtre, Filter	85
Footer	116
Forme, Shape	47
G	
Gérer les composants installés	111
Gestion des SQL Templates	277
Grid	52
H	
Header	116
Header/Footer	116
I	
Importer	
Eléments	131
Importer des extraits de job	16
Inner join	169
Inner Join Reject	169
Input/Output Multiple	86
Installer les composants	110
Interface	
Aperçu du code	300
Barre d'outils	293
Fenêtre principale	290
Menus	291
onglets de configuration des Jobs	297
Palette	296
Repository	294
structure de Job	300
Workspace	296
Iterate	86
J	
Job	58
activer	130
code & outline	121
connecter les composants	64
couleur	122
créer	58
déployer	157
déposer des composants (palette)	61
déposer des composants (repository)	62
désactiver	130
document HTML	156
éditer la documentation	156
erreurs & warnings	119
Exécuter	73, 103
exécuter	72
exporter des éléments	147
exporter des scripts (java)	133
exporter des scripts (perl)	145
Extra	126
importer des éléments	131
note	121
Ouvrir/Créer	58
paramètres de contexte	149
propriétés des composants	65
version	154
Job design	58
Job Designer	
Onglets	82

Job Designer, Espace de modélisation	296
Job version	154
L	
Lancer le studio	7
Link	88
Log	297
Logs	83
Lookup	84
M	
Main row	84
Mapper	86
Mapping de flux	105
Menus	291
Mesurer les flux de données	69
Metadata	
DB Connection schema	246
Metadata manager	90
Metadata, Métadonnées	91
Métadonnées	
Schéma DB Connection	190, 197, 200
Schéma FileDelimited	205
Schéma FileLDIF	230
Schéma FilePositional	210
Schéma FileRegex	212
Schéma FileXML	214
Métadonnées, Metadata	91
Mode Debug	74
Model	
Commenter	50
Copier	55
Déplacer	55
Enregistrer	55
Réarranger	50, 51
Supprimer	55
Modeler	46
Modules externes	111
N	
Notes	121
O	
Objet	47
Onglet Advanced settings	69
Mesurer les flux de données	69
Onglet Appearance	51
Onglet Basic settings	65
Onglet Error Log	297
Onglet Extra	126
Onglet Outline	121
Onglet View	71
Onglets	
Component	297
Error log, Onglets Modules	297
Job hierarchy	297
Job View	297
problems	297
Properties	297
Run Job	297
Scheduler	297
Talend Exchange	297
Onglets de configuration	
vue modules	111
Vue Scheduler	113
Outline	121, 122, 300
Output	85
P	
Palette	47, 50, 61, 78, 79, 296
chercher des composants	64
masquer/afficher	79
Note	50
Note attachment	50
position	79
Select	50
trouver des jobs	126
Zoom	50
Palette favorite	80
Palette layout	81
Paramètre	
Préférences des Stats & Logs	37
Paramétriser un schéma	67
Paramètres de la Palette	78
Paramètres du projet	32
Paramètres dynamiques	69
Partager une connexion	117
Pause, Breakpoint	74
Position des onglets	82
Préférences	
Ajouter du code par défaut	26
Bibliothèques	30
Caractères spéciaux	28
composants	81
Composants externes	22
Composants utilisateur	22
Conversions de type	31

Documentation	28	Row	84
Espace de modélisation	25	Main	84
Exécution	23	Reject	84
Interpréteurs Perl/Java	21	Rulers	52
Langues	23	Run After (devient OnSubjobOK)	87
Performance	26	Run Before (devient OnSubjobOK)	87
Schémas	29	Run Job	73, 83, 103
SQL Builder	28	Run Job, Exécuter le Job	297
Problems	83		
Projet		S	
créer	14	Schéma	
exporter	19	Built-in	68
importer	17	Schéma fichier Excel	226
ouvrir	18	Schéma fichier LDIF	230
supprimer	18	Schéma fichier positionnel	210
Projets	14	Schéma fichier Regex	212
Properties, Propriétés	46, 51, 296, 297	Schéma fichier XML	214
Propriétés		Se connecter au studio	10
Champ Comment	72	Sous-jobs	
Onglet View	71	Personnaliser	123
Rulers & Grid	52	Réduire	123
Propriétés avancées	69	Surlignage	123
Propriétés d'un composant	65	SQL Templates	104
propriétés de base	65	SQLBuilder	105
Propriétés, Properties	46, 51, 296, 297	Start	118
		Start component, Composant de début	130
R		Statistiques	76
Rafraîchir, Refresh	18	StoreSQLQuery	99
Rechercher un Job	153	Structure du Job	300
Récupérer un schéma ponctuel	67	Sync columns	67
Recycle bin, Corbeille	55		
Référentiel, Repository	14, 44, 58, 294	T	
Refresh, Rafraîchir	18	Tableau d'affectation	54
Reject row	84	Talend et l'ESB	141
Rejets	85	Talend Exchange	110
Relation	48	tFlowMeterCatcher	124, 299
Bidirectionnelle	49	ThenRun (devient OnSubjobOK)	87
Directionnelle	49	tLogCatcher	124, 299
Simple	49	tMap	86, 165
Repository	90, 294	Traces	75
déTECTer les déPENDances	152	Trigger	
déTECTer les déPENDances de Job(s)	150	OnComponentError	87
édITER un élÉMENT	150	OnComponentOK	87
Mettre à jour les Jobs	151	OnSubjobError	87
Repository, Référentiel	14, 44, 58, 294	OnSubjobOK	87
Requête		Run if	88
SQLBuilder	105	Trouver des jobs	126
requêtes SQL	105	tStatCatcher	124, 299

U	Vues
Uniques	Réorganiser82
Utiliser des requêtes	
V	
Variable	Webservice136
StoreSQLQuery	Workspace12
Vue Job	
Vue Scheduler	Xpath218
W	
X	

