



Talend Open Studio

for Data Integration

Guide utilisateur

5.0_b

Talend Open Studio: Guide utilisateur

Convient pour l'utilisation de Talend Open Studio for Data Integration v5.0.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/>

Notices

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

Table des matières

Préface	ix
1. Informations générales	ix
1.1. Objectif	ix
1.2. Public visé	ix
1.3. Conventions typographiques	ix
2. Historique des versions	ix
3. Remarques et Support	x
chapitre 1. Intégration de données et Studio Talend	1
1.1. Intégration décisionnelle avec Talend Open Studio for Data Integration	2
1.2. Intégration opérationnelle avec Talend Open Studio for Data Integration	2
1.3. Monitoring de l'exécution.....	3
chapitre 2. Présentation du Studio Talend	5
2.1. Concepts importants dans Talend Open Studio for Data Integration	6
2.2. Lancement de Talend Open Studio for Data Integration	7
2.2.1. Lancer le Studio pour la première fois	7
2.2.2. Configurer un projet.....	11
2.3. Travailler avec différents répertoires workspace	11
2.3.1. Créer un nouveau répertoire workspace	12
2.4. Travailler avec les projets.....	12
2.4.1. Créer un projet	13
2.4.2. Importer le projet Démo	15
2.4.3. Importer un projet	16
2.4.4. Ouvrir un projet	18
2.4.5. Supprimer un projet.....	19
2.4.6. Exporter un projet	19
2.4.7. Tâches de migration.....	20
2.5. Configuration des préférences de Talend Open Studio for Data Integration	21
2.5.1. Interpréteur Java	21
2.5.2. Composants externes ou utilisateur	22
2.5.3. Préférence Exchange	23
2.5.4. Préférences de langue.....	23
2.5.5. Préférences d'exécution et de débogage.....	24
2.5.6. Préférences du Designer	25
2.5.7. Ajouter du code par défaut	26
2.5.8. Préférences de Performance	27
2.5.9. Préférences de la documentation	28
2.5.10. Afficher des caractères spéciaux dans les colonnes de schémas	28
2.5.11. Préférences du SQL Builder	29
2.5.12. Préférences des schémas	29
2.5.13. Préférences des bibliothèques	30
2.5.14. Préférences des conversions de type	31
2.5.15. Préférences du collectionneur de données d'utilisation.....	32
2.6. Personnalisation des paramètres du projet	33
2.6.1. Personnaliser la Palette	34
2.6.2. Gérer les versions	36
2.6.3. Gérer les statuts	37
2.6.4. Personnaliser les paramètres du Job	38
2.6.5. Configurer les informations de logs	39
2.6.6. Définir les paramètres de contexte	40
2.6.7. Appliquer les paramètres du projet	41
2.6.8. Définir les paramètres des Statuts	42
2.6.9. Paramètres de sécurité	44
chapitre 3. Conception d'un Business Model	45
3.1. Objectifs	46
3.2. Crédit ou ouverture d'un Business Model	46
3.2.1. Ouvrir un Business Model	47
3.2.2. Crédit un Business Model	47
3.3. Modélisation d'un Business Model	48
3.3.1. Formes (Shapes)	49
3.3.2. Connexions	50
3.3.3. Commenter et réarranger un Business Model	51
3.3.4. Propriétés	53
3.4. Affectation d'éléments du Repository à un Business Model	55
3.5. Modification d'un Business Model	56

3.5.1. Renommer un Business Model	56	4.5.6. Utiliser les composants tPrejob et tPostjob	118
3.5.2. Copier et coller un Business Model	56	4.5.7. Utiliser la fonctionnalité Use Output Stream	119
3.5.3. Déplacer un Business Model	56	4.6. Gestion des Jobs : Autres fonctions	119
3.5.4. Supprimer un Business Model	56	4.6.1. Partager une connexion à une base de données	120
3.6. Enregistrement d'un Business Model	57	4.6.2. Configurer un composant Start	121
chapitre 4. Conception d'un Job d'intégration de données	59	4.6.3. Gérer les icônes d'avertissement/d'erreur sur les composants	121
4.1. Objectifs	60	4.6.4. Ajouter une Note au Job	123
4.2. Utilisation d'un Job Design simple	60	4.6.5. Afficher les onglets Code ou Outline de votre Job	124
4.2.1. Créer un Job	60	4.6.6. Gérer l'affichage d'un sous-job	125
4.2.2. Déposer des composants dans l'espace de modélisation graphique	63	4.6.7. Paramétrage des options dans la vue Job	127
4.2.3. Chercher des composants dans la Palette	65	4.6.8. Chercher un Job utilisant un composant particulier	129
4.2.4. Connecter des composants	66	4.6.9. Configurer les valeurs par défaut dans le schéma d'un composant	130
4.2.5. Cliquer et déposer des composants au milieu d'une connexion de type Row	66		
4.2.6. Définir les propriétés d'un composant	68		
4.2.7. Exécuter un Job	74		
4.2.8. Personnaliser l'espace de travail	80		
4.3. Utilisation des connexions	85		
4.3.1. Types de connexions.....	85		
4.3.2. Définir les paramètres des connexions	89		
4.4. Utilisation du Metadata Manager	91		
4.4.1. Centraliser les métadonnées	92		
4.4.2. Centraliser les Contextes et les Variables	92		
4.4.3. Utiliser les SQL Templates	103		
4.5. Gestion des Jobs : Utilisation avancée	104		
4.5.1. Mapper des flux de données	104		
4.5.2. Créer une requête à l'aide de SQLBuilder	104		
4.5.3. Télécharger/charger des composants de la communauté	108		
4.5.4. Installer des modules externes	115		
4.5.5. Lancer un Job périodiquement	116		
		chapitre 5. Gestion des Jobs d'intégration de données	133
		5.1. Activation/Désactivation d'un Job ou d'un sous-job	134
		5.1.1. Désactiver un composant de début (Start)	134
		5.1.2. Désactiver un composant intermédiaire	134
		5.2. Import/export d'éléments ou de Jobs	135
		5.2.1. Import d'éléments	135
		5.2.2. Export de Jobs vers une archive	137
		5.2.3. Export d'éléments	149
		5.2.4. Changer les paramètres de contexte d'un Job script.....	151
		5.3. Gestion des éléments du Repository	152
		5.3.1. Gérer les mises à jour des éléments du Repository....	152
		5.4. Recherche d'un Job dans le référentiel	155
		5.5. Gestion des versions d'un Job	156
		5.6. Documentation du Job	157
		5.6.1. Générer la documentation HTML	157
		5.6.2. Mettre la documentation à jour	158

5.7. Gestion de l'exécution des Jobs	158	7.4.2. Etape 1 : Propriétés générales	220
5.7.1. Déploiement d'un Job sur un serveur SpagoBI	158	7.4.3. Etape 2 : Connexion.....	220
chapitre 6. Mapping de flux de données	163	7.5. Configurer un schéma File Delimited	222
6.1. Interface du tMap et du tXMLMap	164	7.5.1. Etape 1 : Propriétés générales	223
6.2. Présentation du fonctionnement du tMap	165	7.5.2. Etape 2 : Chargement du fichier	223
6.2.1. Configuration du flux d'entrée dans le Map Editor....	166	7.5.3. Etape 3 : Définition du schéma	224
6.2.2. Mapping de variables....	174	7.5.4. Etape 4 : Schéma final	225
6.2.3. Utilisation de l'Expression editor	175	7.6. Configurer un schéma File Positional	226
6.2.4. Paramètres Output	179	7.6.1. Etape 1 : Propriétés générales	227
6.2.5. Configuration des schémas dans le Map Editor	185	7.6.2. Etape 2 : Connexion et chargement du fichier	227
6.2.6. Résolution des problèmes de mémoire lors de l'utilisation du tMap	188	7.6.3. Etape 3 : Affinage du schéma	228
6.2.7. Gestion des références	191	7.6.4. Etape 4 : Schéma final	228
6.3. Présentation du fonctionnement du tXMLMap	192	7.7. Configurer un schéma File Regex	229
6.3.1. Utiliser le type Document pour créer l'arborescence XML	193	7.7.1. Étape 1 : Propriétés générales	229
6.3.2. Définir le mode de sortie	203	7.7.2. Étape 2 : Chargement du fichier	229
6.3.3. Utiliser le Tree schema editor	207	7.7.3. Étape 3 : Définition du schéma	230
chapitre 7. Gestion des métadonnées	209	7.7.4. Étape 4 : Schéma final	230
7.1. Objectifs	210	7.8. Configurer un Schéma Fi-le XML	230
7.2. Configurer une connexion DB	211	7.8.1. Configurer un schéma XML pour un fichier d'entrée (Input)	231
7.2.1. Etape 1 : Propriétés générales	211	7.8.2. Configurer un schéma XML pour un fichier de sortie (Output)	238
7.2.2. Etape 2 : Connexion.....	211	7.9. Configurer un schéma File Excel	247
7.2.3. Etape 3 : Chargement de la table	213	7.9.1. Étape 1 : Propriétés générales	248
7.2.4. Etape 4 : Définition du schéma	216	7.9.2. Étape 2 : Chargement du fichier	248
7.3. Configurer une connexion JDBC	217	7.9.3. Étape 3 : Affinage du schéma	249
7.3.1. Etape 1 : Propriétés générales	217	7.9.4. Étape 4 : Schéma final	250
7.3.2. Etape 2 : Connexion....	217	7.10. Configurer un schéma File LDIF	251
7.3.3. Etape 3 : Chargement de la table	219	7.10.1. Étape 1 : Propriétés générales	251
7.3.4. Etape 4 : Définition du schéma	219	7.10.2. Étape 2 : Chargement du fichier	251
7.4. Configurer une connexion SAS	220		
7.4.1. Conditions préalables....	220		

7.10.3. Étape 3 : Définition du schéma	252
7.10.4. Étape 4 : Schéma final	253
7.11. Configurer un schéma LDAP	253
7.11.1. Étape 1 : Propriétés générales	254
7.11.2. Étape 2 : Connexion au serveur	254
7.11.3. Étape 3 : Authentification et acquisition du DN....	254
7.11.4. Étape 4 : Définition du schéma	256
7.11.5. Étape 5 : Schéma final	257
7.12. Configurer un schéma Salesforce	257
7.12.1. Étape 1 : Propriétés générales	258
7.12.2. Étape 2 : Connexion au compte Salesforce	258
7.12.3. Étape 3 : Récupérer les modules Salesforce	258
7.12.4. Étape 4 : Récupérer les schémas Salesforce	260
7.12.5. Étape 5 : Schéma final	261
7.13. Configurer un schéma générique	262
7.13.1. Etape 1 : Propriétés générales	262
7.13.2. Etape 2 : Définition du schéma	262
7.14. Configurer une connexion MDM	263
7.14.1. Étape 1 : Définir la connexion	263
7.14.2. Étape 2 : Définir un schéma MDM	265
7.15. Configurer un schéma Web Service	279
7.15.1. Configurer un schéma simple	279
7.16. Configurer une connexion FTP	282
7.16.1. Étape 1 : Propriétés générales	282
7.16.2. Étape 2 : Connexion....	283
7.17. Exporter une métadonnée en tant que contexte	285
chapitre 8. Gestion des routines	287
8.1. Définition des routines	288
8.2. Accès aux routines système....	288
8.3. Personnalisation des routines système	289
8.4. Gestion des routines personnalisées	290
8.4.1. Créer des routines personnalisées	290
8.4.2. Éditer des routines	292
8.4.3. Éditer les bibliothèques des routines utilisateur	292
8.5. Appel d'une routine à partir d'un Job	294
8.6. Cas d'utilisation : créer un fichier à la date du jour	294
chapitre 9. SQL Templates	297
9.1. Qu'est-ce que l'ELT ?	298
9.2. Définition des SQL Templates Talend	298
9.3. Gérer les SQL Templates Talend	299
9.3.1. Les types de SQL Templates système	299
9.3.2. Accéder aux SQL Templates système	300
9.3.3. Créer des SQL Templates personnalisés	302
9.3.4. Exemple d'utilisation des SQL Templates système	303
annexe A. Interface de <i>Talend Open Studio for Data Integration</i>	309
A.1. Fenêtre principale	310
A.2. Barre d'outils et Menus	311
A.2.1. Barre de menu de <i>Talend Open Studio for Data Integration</i>	311
A.2.2. Barre d'outils de <i>Talend Open Studio for Data Integration</i>	313
A.3. Repository	313
A.4. Espace de modélisation graphique	315
A.5. Onglets de configuration	316
A.6. Panneau des onglets outline et code	318
A.7. Raccourcis clavier	318
annexe B. De la théorie à la pratique, exemple de Job	321
B.1. Exemple de Job comprenant un tMap	322
B.1.1. Présentation du scénario	322
B.1.2. Du scénario au Job	323
B.2. Utilisation de la fonctionnalité Output Stream	330
B.2.1. Présentation du scénario	330
B.2.2. Création du Job	331

annexe C. Règles d'écriture des SQL Templates	339	D.6.2. Exemple de trimming d'une chaîne	356
C.1. Instructions SQL	340	D.6.3. Exemple de désaccentuation d'une chaîne	356
C.2. Lignes de commentaire	340		
C.3. La syntaxe <% . . . %>	340		
C.4. La syntaxe <%= . . . %>	341		
C.5. La syntaxe </ . . . />	341		
C.6. Le code pour accéder aux éléments du schéma des composants....	342		
C.7. Le code pour accéder aux propriétés de la matrice du composant	343		
annexe D. Routines système	345		
D.1. Routines de type Numeric	346		
D.1.1. Exemple de création séquentielle	346		
D.1.2. Exemple de conversion décimale implicite	346		
D.2. Routines de type Relational.....	347		
D.3. Routines de type StringHandling	347		
D.3.1. Exemple de vérification de tri alphabétique	348		
D.3.2. Exemple de vérification de type alphabétique	349		
D.3.3. Exemple de remplacement de chaîne	349		
D.3.4. Exemple de vérification de chaîne	349		
D.3.5. Exemple de calcul de longueur de chaîne	349		
D.3.6. Exemple de nettoyage d'espaces inutiles	350		
D.4. Routines de type TalendData-Generator	350		
D.4.1. Exemple de génération de données factices	351		
D.5. Routines de type TalendDate....	351		
D.5.1. Exemple de formatage d'une Date	353		
D.5.2. Exemple de vérification d'une Date	353		
D.5.3. Exemple de comparaison de Dates	353		
D.5.4. Exemple de configuration de Date	353		
D.5.5. Exemple de parsage de Date	354		
D.5.6. Exemple de récupération d'une partie d'une Date...	354		
D.5.7. Exemple de formatage de la Date courante	355		
D.6. Routines de type TalendString	355		
D.6.1. Exemple de formatage XML d'une chaîne	356		

Préface

1. Informations générales

1.1. Objectif

Le présent Guide utilisateur explique comment utiliser les fonctions de *Talend Open Studio for Data Integration* dans un contexte opérationnel normal.

Les informations contenues dans ce document s'appliquent aux versions **5.0.x** de *Talend Open Studio for Data Integration*.

1.2. Public visé

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



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

1.3. 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.

2. Historique des versions

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

Version	Date	Historique
v4.2_a	19/05/2011	<p>Les mises à jour du Guide utilisateur <i>Talend Open Studio for Data Integration</i> comprennent :</p> <ul style="list-style-type: none"> - mise à jour de la variable Copyright dans la couverture ; - mise à jour du chapitre : Mapping de flux de données ; - mise à jour de l'annexe Routines système ; - mise à jour du chapitre : Gestion des métadonnées ; - mise à jour du chapitre : Conception d'un Job d'intégration de données ; - mise à jour du chapitre : Gestion des Jobs d'intégration de données.
v4.2_b	12/07/2011	Nouvelle génération du Guide utilisateur <i>Talend Open Studio for Data Integration</i> .
v5.0_a	12/12/2011	<p>Les mises à jour du Guide utilisateur <i>Talend Open Studio for Data Integration</i> comprennent :</p> <ul style="list-style-type: none"> - la mise à jour de la documentation pour refléter les nouveaux noms de produits : - <i>Talend Open Studio</i> devient <i>Talend Open Studio for Data Integration</i>, <p>Pour plus d'informations sur ces changements, consultez le site Web de Talend.</p> <ul style="list-style-type: none"> - la restructuration post-migration ; - la mise à jour du chapitre Présentation du Studio <i>Talend</i> ; - la mise à jour du chapitre Conception d'un Job d'intégration de données ; - la mise à jour du chapitre Mapping de flux de données ; - la mise à jour du chapitre Gestion des métadonnées ; - la mise à jour de l'annexe De la théorie à la pratique, exemple de Job.
v5.0b	13/02/2012	Les mises à jour du Guide utilisateur <i>Talend Open Studio for Data Integration</i> comprennent :
		<ul style="list-style-type: none"> - L'ajout de mentions légales.

3. 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. Intégration de données et Studio Talend

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.

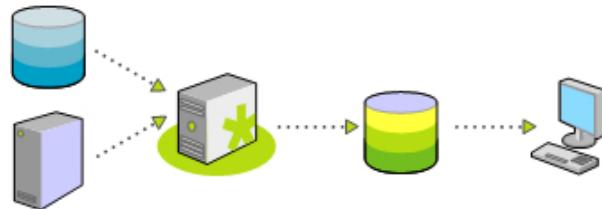
Enfin cette complexification s'explique aussi par la multiplication des formats de stockage de données (fichiers XML, plats positionnels, plats délimités, multi-valués, etc.), des protocoles (FTP, HTTP, SOAP, SCP, etc.) et des technologies des bases de données.

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 de 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 for Data Integration couvre tous ces besoins qu'il s'agisse d'intégration décisionnelle ou opérationnelle.

1.1. Intégration décisionnelle avec *Talend Open Studio for Data Integration*

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.



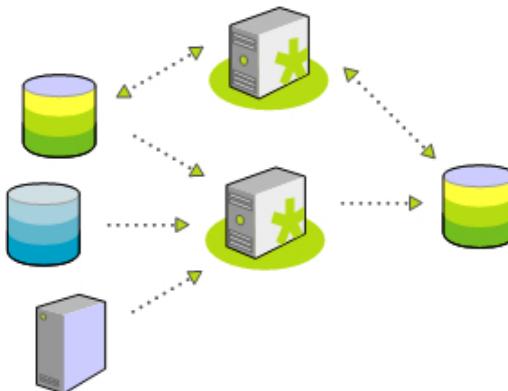
Talend Open Studio for Data Integration offre une très grande connectivité aux :

- 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 for Data Integration*, consultez le [Chapitre 7, Gestion des métadonnées](#). Pour la modélisation de processus métier, consultez le [Chapitre 3, Conception d'un Business Model](#).

1.2. Intégration opérationnelle avec *Talend Open Studio for Data Integration*

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 for Data Integration*, consultez le [Chapitre 7, Gestion des métadonnées](#). Pour la modélisation de processus métier, consultez le [Chapitre 3, Conception d'un Business Model](#). Pour plus d'informations concernant un Job d'intégration de données utilisant la fonctionnalité Output Stream, consultez [Section B.2, « Utilisation de la fonctionnalité Output Stream »](#).

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

La console *Activity Monitoring Console* surveille l'exécution du Job (succès, échecs, avertissements, etc.), le temps d'exécution et les volumes de données, tout ceci en une seule console, *Talend Open Studio for Data Integration* ou disponible en environnement indépendant.

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

Chapitre 2. Présentation du Studio Talend

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

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 for Data Integration*.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez [Annexe A, Interface de Talend Open Studio for Data Integration](#) .

2.1. Concepts importants dans *Talend Open Studio for Data Integration*

Lorsque vous travaillez avec *Talend Open Studio for Data Integration*, 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 for Data Integration*.

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

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 [Section 2.2.2, « Configurer un projet »](#).

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 for Data Integration* 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 [Section 2.3, « Travailler avec différents répertoires workspace »](#).

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 for Data Integration*, consultez le [Chapitre 4, Conception d'un Job d'intégration de données](#).

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 for Data Integration*.

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.

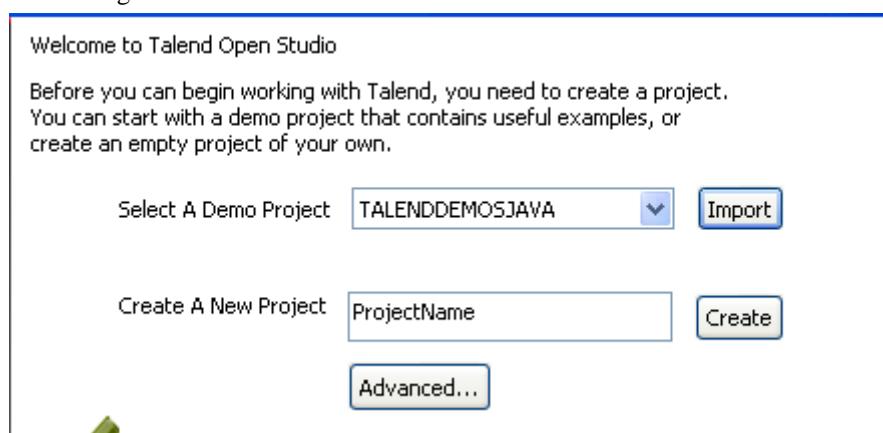
2.2. Lancement de *Talend Open Studio for Data Integration*

2.2.1. Lancer le Studio pour la première fois

Pour ouvrir *Talend Open Studio for Data Integration* pour la première fois, procédez comme suit :

1. Dézippez le fichier zip *Talend Open Studio for Data Integration* et, dans le dossier, double-cliquez sur le fichier exécutable correspondant à votre système d'exploitation.
2. Dans la boîte de dialogue [Licence] qui apparaît, lisez et acceptez les termes de la licence afin de passer à l'étape suivante.

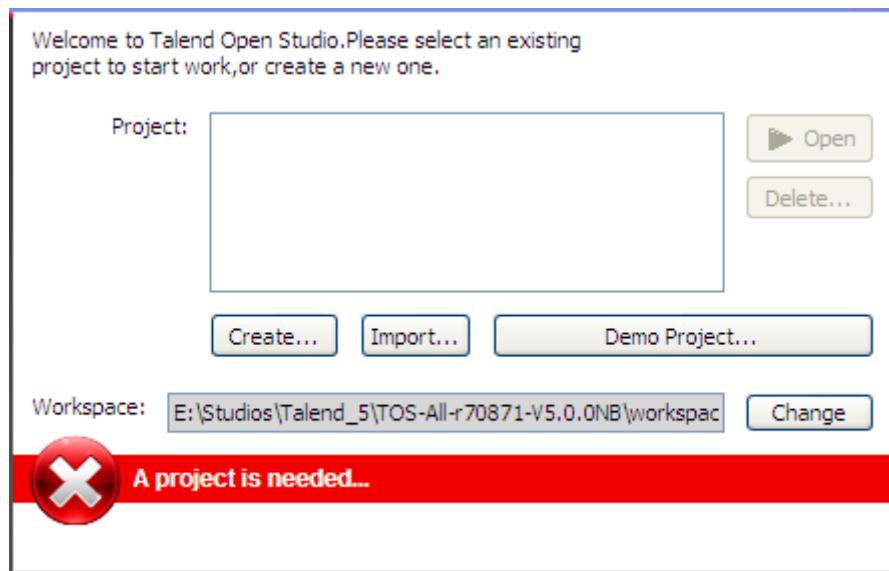
Une fenêtre de démarrage s'affiche.



Cet écran apparaît uniquement lorsque vous lancez *Talend Open Studio for Data Integration* pour la première fois ou si tous les projets existants ont été supprimés.

3. Cliquez sur le bouton **Import** afin d'importer le projet Démo sélectionné, ou saisissez un nom de projet dans le champ **Create A New Project**, puis cliquez sur le bouton **Create** pour créer le nouveau projet, ou sur le bouton **Advanced...** pour aller à la fenêtre de login.

Pour cette procédure, cliquez sur **Advanced...** pour accéder à la fenêtre de connexion. Pour plus d'informations sur les deux autres options, consultez [Section 2.4.1, « Crée un projet »](#) et [Section 2.4.2, « Importer le projet Démo »](#).

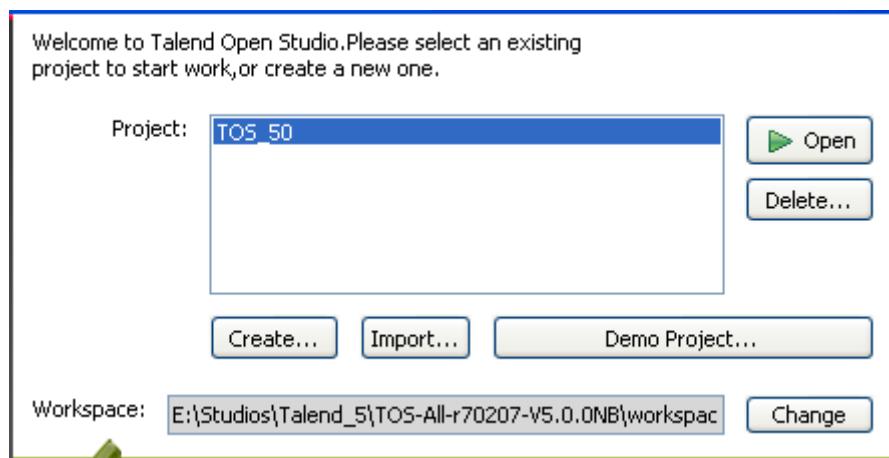


4. A partir de la fenêtre de connexion du Studio

Cliquez sur...	Pour...
Créer...	créer un nouveau projet qui contiendra tous les Jobs et les Business models conçus dans le Studio. Pour plus d'informations, consultez Section 2.4.1, « Créer un projet »
Importer...	importer tout projet stocké Pour plus d'informations, consultez Section 2.4.3, « Importer un projet »
Projet démo	importer un Projet démo incluant de nombreux Jobs prêts à être utilisés. Ce Projet démo peut vous aider à comprendre les fonctionnalités des différents composants Talend . Pour plus d'informations, consultez Section 2.4.2, « Importer le projet Démo »
Open	ouvrir le projet existant sélectionné. Pour plus d'informations, consultez Section 2.4.4, « Ouvrir un projet » .

Cette procédure ayant pour but de créer un nouveau projet , cliquez sur **Créer** pour ouvrir la boîte de dialogue [**Nouveau projet**].

5. Dans la boîte de dialogue, saisissez un nom pour votre projet et cliquez sur **Finish** pour fermer la boîte de dialogue. Le nom du nouveau projet s'affiche dans la liste **Project**.



6. Sélectionnez le projet et cliquez sur **Open**.

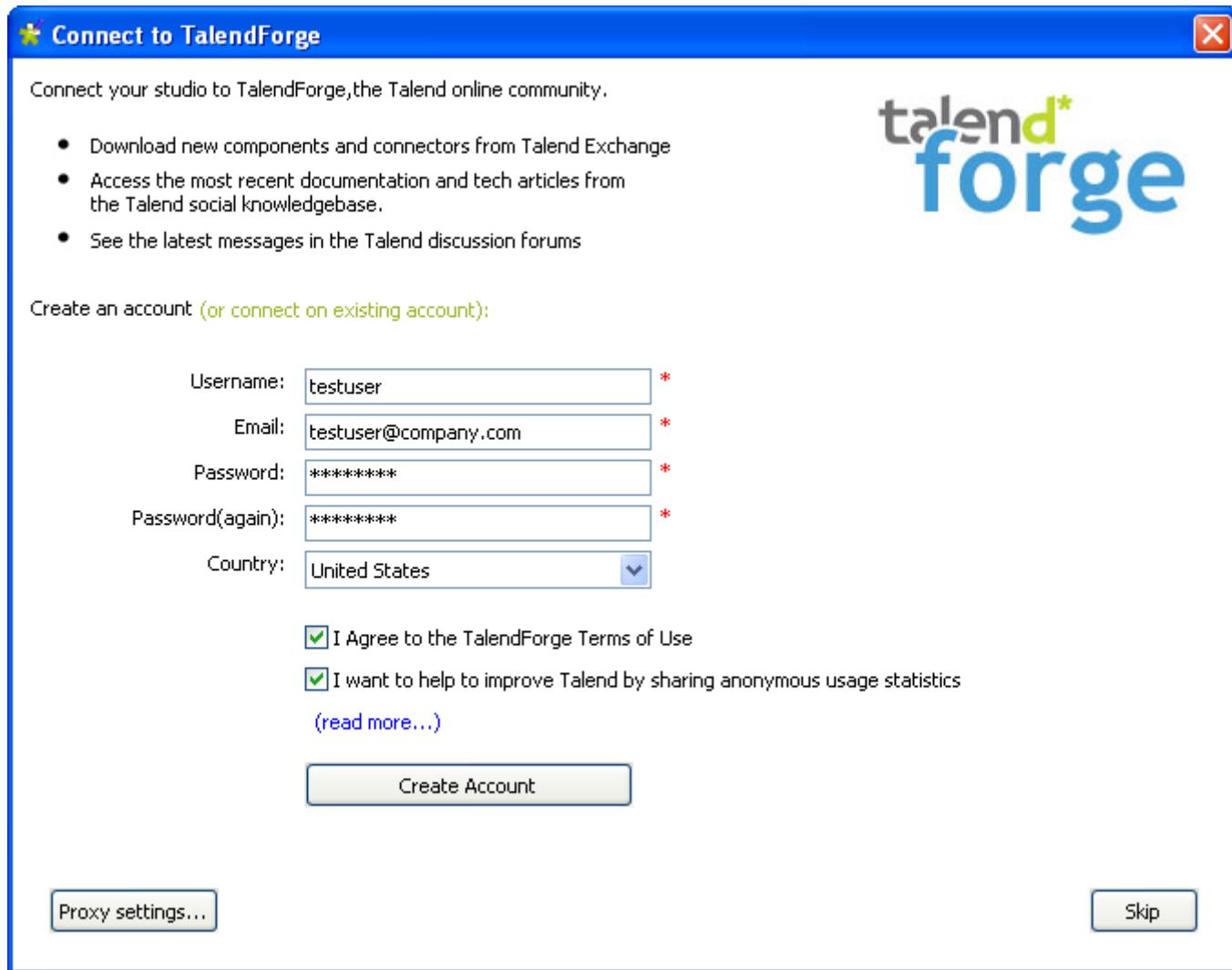
La page **Connect to TalendForge** apparaît, vous invitant à vous connecter à la Communauté **Talend** afin que vous puissiez regarder, télécharger et installer des composants externes, charger vos propres composants afin de les partager avec les autres utilisateurs de **Talend**, le tout directement à partir de la vue **Exchange** de la création de Job dans le Studio.

Pour en apprendre plus sur la Communauté **Talend**, cliquez sur le lien **en savoir plus**. Pour plus d'informations sur l'utilisation et le partage des composants avec la Communauté, consultez [Section 4.5.3, « Télécharger/charger des composants de la communauté »](#).

7. Si vous souhaitez vous connecter à la Communauté **Talend** ultérieurement, cliquez sur le bouton **Skip** pour continuer.
8. Si vous travaillez derrière un proxy, cliquez sur **Proxy settings...** et renseignez les champs **Proxy Host** et **Proxy Port** dans la boîte de dialogue **Network setting**.
9. Par défaut, le Studio collecte automatiquement les données d'utilisation et les envoie périodiquement aux serveurs hébergés par **Talend**, dans un but d'amélioration. Si vous ne souhaitez pas que le Studio effectue ces envois, décochez la case **I want to help to improve Talend by sharing anonymous usage statistics**. Pour plus d'informations concernant le collecteur de données d'utilisation, consultez [Section 2.5.15, « Préférences du collecteur de données d'utilisation »](#).
10. Renseignez les informations requises, cochez la case **I Agree to the TalendForge Terms of Use** et cliquez sur **Create Account** afin de créer votre compte et de vous connecter automatiquement à la Communauté **Talend**. Si vous avez déjà un compte sur <http://www.talendforge.org/>, cliquez sur le lien **or connect on existing account** afin de vous connecter.



Soyez assuré qu'aucune information personnelle que vous communiquez à **Talend** ne sera transmise à des tiers ou utilisée à d'autres fins que celles de vous connecter à la Communauté ou de vous informer sur **Talend** des dernières mises à jour de **Talend**.

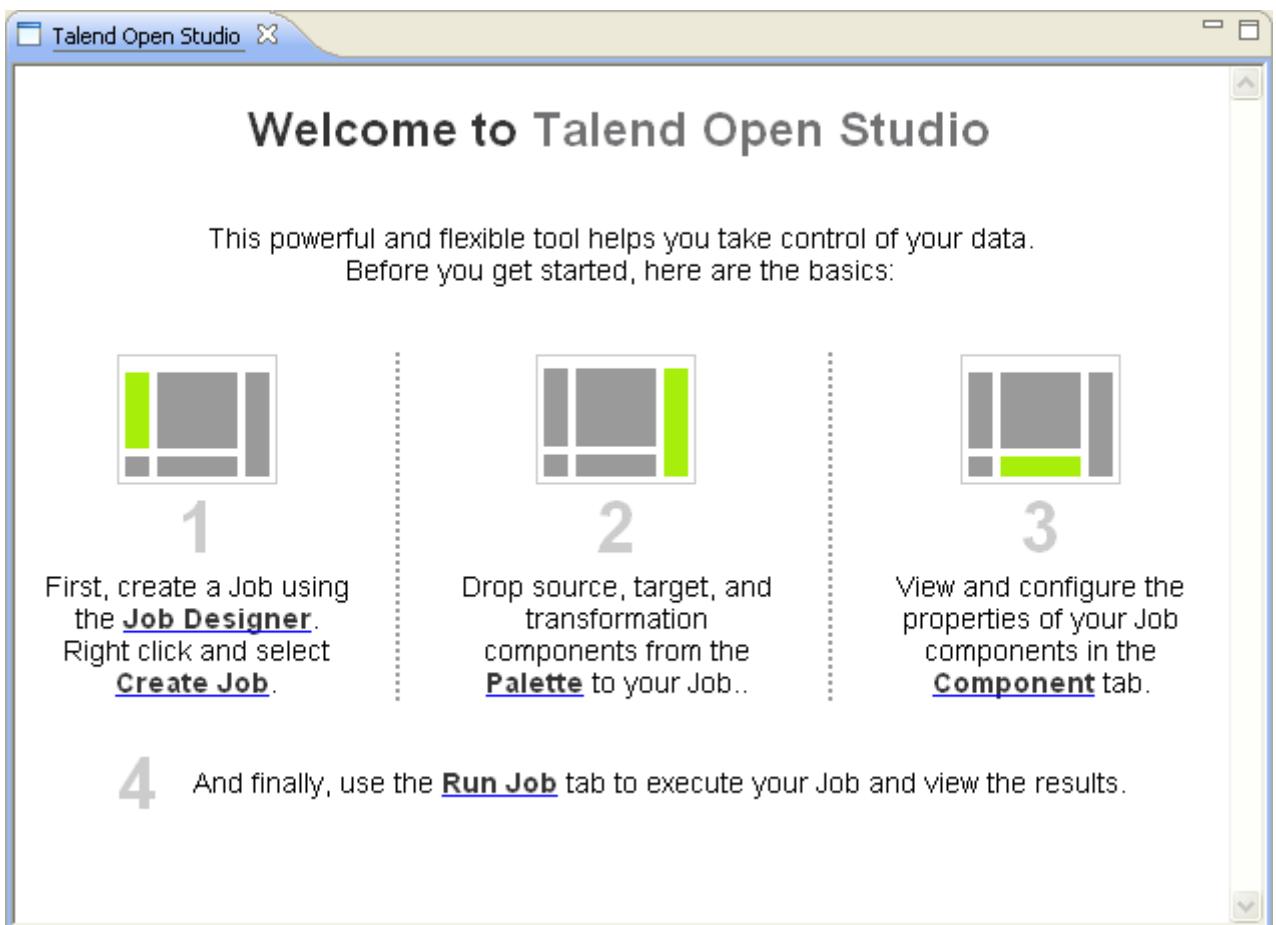


Cette page n'apparaît plus à l'ouverture du Studio une fois que vous vous êtes connecté avec succès à la Communauté **Talend**, ou si vous cliquez six fois d'affilée sur le bouton **Skip** lorsque vous ouvrez le Studio. Dans ce cas, pour afficher de nouveau la page, ouvrez la boîte de dialogue **[Preferences]**, sélectionnez **Talend > Exchange** puis cliquez sur **Logon Exchange**. Pour plus d'informations, consultez [Section 2.5.3, « Préférence Exchange »](#).

Une barre de progression et une fenêtre de bienvenue s'ouvrent consécutivement. Dans cette page, des liens directs vers la documentation, les tutoriels, le forum **Talend**, **Talend Exchange** et les dernières nouvelles de **Talend** sont affichés.

11.Cliquez sur **Start now** afin d'ouvrir la fenêtre principale de *Talend Open Studio for Data Integration* qui contient des astuces utiles pour les débutants concernant le fonctionnement du Studio. Cliquer sur un lien souligné vous amène à la vue ou à l'onglet correspondant, ou ouvre la boîte de dialogue relative.

Pour plus d'informations concernant l'ouverture d'un projet , consultez [Section 2.4.4, « Ouvrir un projet »](#).



2.2.2. Configurer un projet

Pour ouvrir *Talend Open Studio for Data Integration*, vous devez d'abord configurer un projet.

Pour configurer un projet dans le référentiel, vous pouvez :

- créer un nouveau projet. Pour plus d'informations, consultez [Section 2.4.1, « Créeer un projet »](#).
- importer un ou plusieurs projets que vous avez déjà créé dans une autre instance de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez [Section 2.4.3, « Importer un projet »](#).
- importer le projet Démo. Pour plus d'informations, consultez [Section 2.4.2, « Importer le projet Démo »](#).

2.3. Travailler avec différents répertoires workspace

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

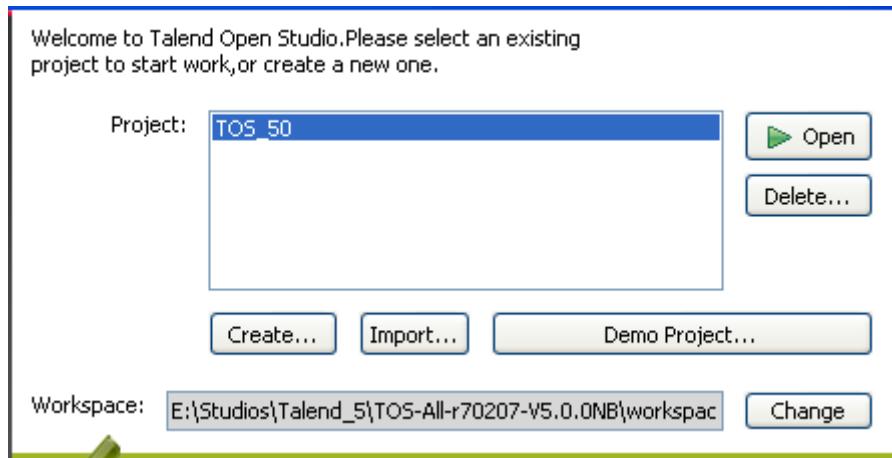
Cette flexibilité vous permet de stocker les répertoires où vous le souhaitez et vous permet de nommer deux ou plusieurs projets de la même manière, du moment que vous stockez les projets dans différents répertoires.

2.3.1. Créer un nouveau répertoire workspace

Talend Open Studio for Data Integration 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 :

1. Dans la fenêtre de connexion au projet, cliquez sur **Change** pour ouvrir la boîte de dialogue vous permettant de sélectionner le répertoire du nouveau workspace.



2. Dans la boîte de dialogue, 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.

3. Cliquez sur **Restart** pour redémarrer le Studio.

4. Dans la fenêtre de login réinitialisée, configurez un projet pour ce nouveau répertoire workspace.

Pour plus d'informations, consultez [Section 2.2.2, « Configurer un projet »](#).

5. Sélectionnez le projet dans la liste **Project** et cliquez sur **Open** pour ouvrir *Talend Open Studio for Data Integration*.

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.

Si vous souhaitez vous connecter à l'un des workspaces que vous avez créés, répétez la procédure décrite dans cette section.

2.4. Travailler avec les projets

Dans *Talend Open Studio for Data Integration*, le “projet” correspond au nœud 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 for Data Integration*, vous pouvez :

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

Pour plus d'informations, consultez [Section 2.4.2, « Importer le projet Démo ».](#)

- créer un projet local. Lorsque vous vous connectez à *Talend Open Studio for Data Integration* pour la première fois, aucun projet par défaut n'apparaît dans la liste. Vous devez créer un projet 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 for Data Integration*.

Pour plus d'informations, consultez [Section 2.4.1, « Créer un projet ».](#)

- importer des projets que vous avez déjà créés dans une version antérieure de *Talend Open Studio for Data Integration* vous pouvez l'importer dans le workspace courant de *Talend Open Studio for Data Integration Import...* .

Pour plus d'informations, consultez [Section 2.4.3, « Importer un projet ».](#)

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

Pour plus d'informations concernant l'ouverture d'un projet , consultez [Section 2.4.4, « Ouvrir un projet ».](#)

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

Pour plus d'informations, consultez [Section 2.4.5, « Supprimer un projet ».](#)

Une fois que vous avez lancé *Talend Open Studio for Data Integration*, 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 [Section 2.4.6, « Exporter un projet ».](#)

2.4.1. Créer un projet

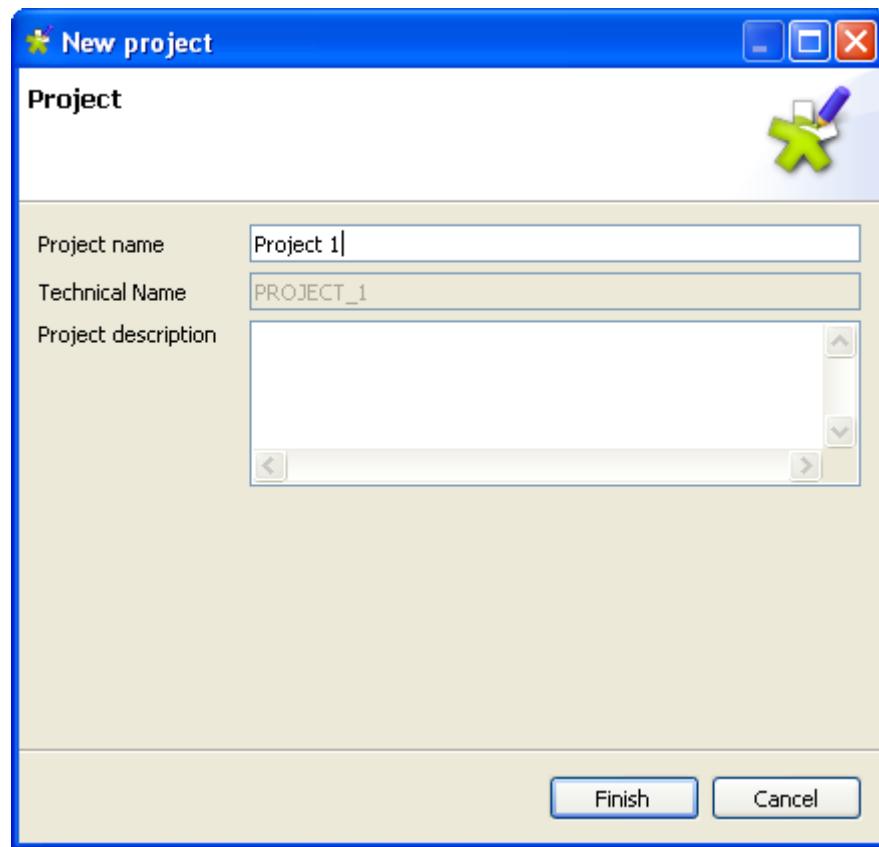
Lorsque vous lancez le Studio pour la première fois, aucun projet par défaut n'apparaît dans la liste. Vous devez créer un projet qui contiendra les Jobs et Business Models d'intégration de données que vous avez créés à partir de l'instance courante du Studio.

Pour créer un projet :

1. Lancez *Talend Open Studio for Data Integration* .

2. Utilisez l'une des deux options suivantes:

- Saisissez le nom du nouveau projet dans le champ **Create A New Project**, puis cliquez sur **Create** pour ouvrir la fenêtre de dialogue contenant le champ **[Project name]** rempli avec le nom spécifié.
- Cliquez sur **[Advanced]** et cliquez sur **Create...à partir de la fenêtre de connexion** pour ouvrir la boîte de dialogue **[New project]** contenant le champ **Project Name** vide.



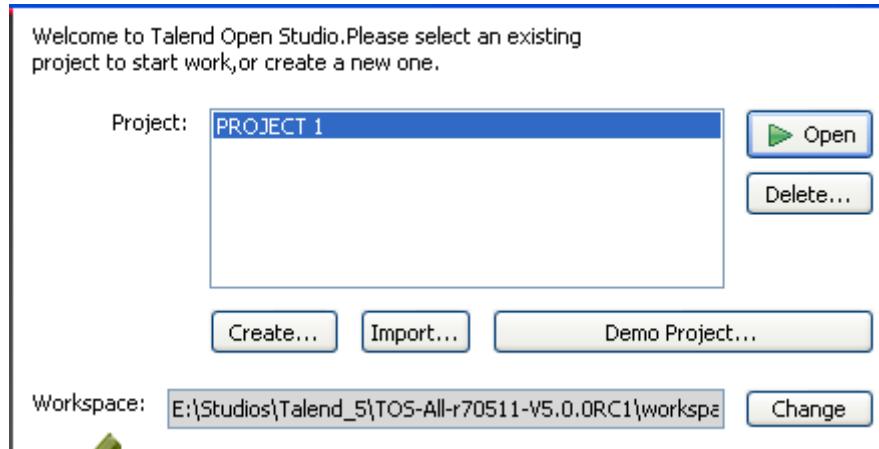
3. Dans le champ **Project name**, saisissez un nom pour le nouveau projet ou si besoin modifiez le nom de projet précédemment saisi. Ce champ est obligatoire

Selon l'endroit où se trouve votre curseur, un message en haut de l'assistant vous informe sur la nature des données à saisir, comme les 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.

4. Cliquez sur **Finish**. Le nom de nouveau projet créé apparaît dans la liste **Project** de la fenêtre de login de *Talend Open Studio for Data Integration*.



Depuis la version 5.0, seul le langage Java est généré.

Pour ouvrir le nouveau projet créé dans *Talend Open Studio for Data Integration*, 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.

Si vous avez déjà créé un projet dans une version antérieure de *Talend Open Studio for Data Integration* et que vous souhaitez l'importer dans la version courante [Section 2.4.3, « Importer un projet »](#).

2.4.2. Importer le projet Démo

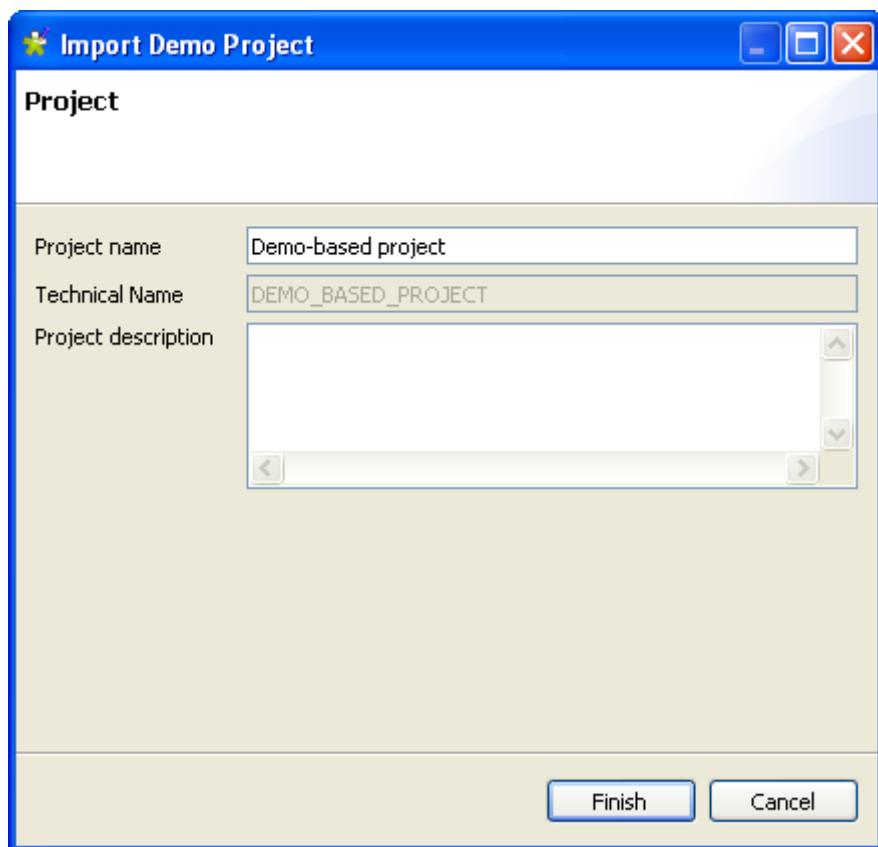
Dans *Talend Open Studio for Data Integration*, vous pouvez importer, dans votre workspace, le projet Démo contenant de nombreux exemples de Jobs prêts à utiliser. Ce projet Démo peut vous aider à comprendre les fonctionnalités des différents composants **Talend**.

Au premier lancement de *Talend Open Studio for Data Integration*, vous pouvez:

- créer un nouveau projet dans votre répertoire en utilisant le projet démo comme modèle,
- importer le projet démo *TALENDDEMONSJAVA* dans votre répertoire.

Pour créer un nouveau projet à partir du projet démo

1. Cliquez sur **Import**, situé à côté de la liste **Select A Demo Project**. La boîte de dialogue **[Import demo Project]** s'affiche.



2. Saisissez un nom pour votre nouveau projet, et cliquez sur **Finish** pour créer le projet.

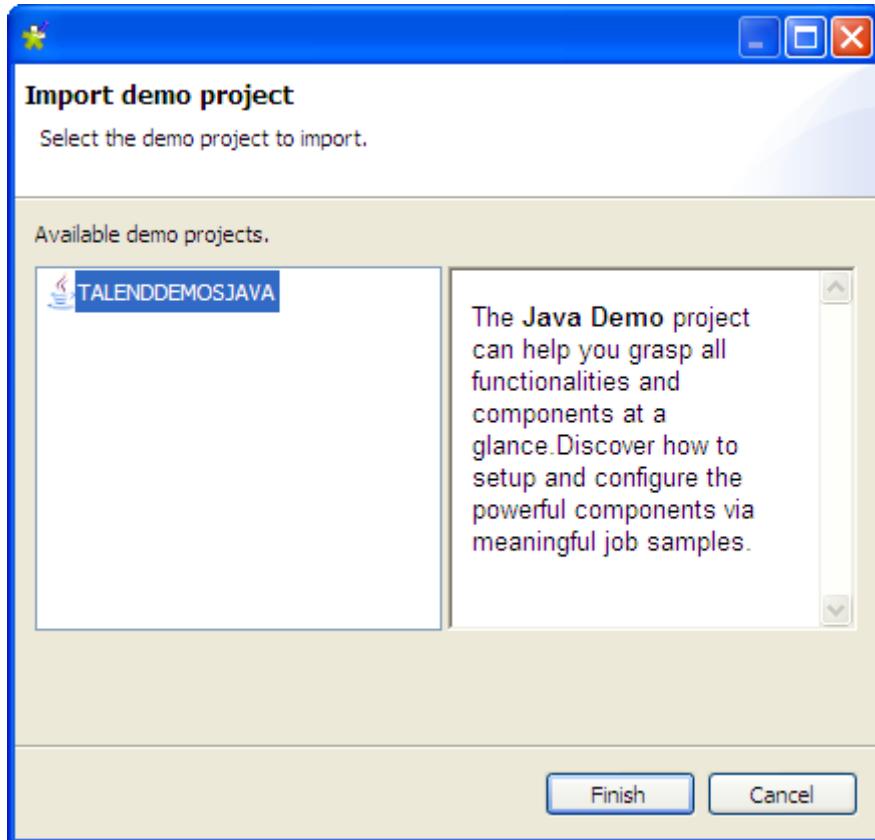
Un message de confirmation s'affiche vous indiquant que l'import a été réalisé avec succès dans l'instance courante du Studio.

3. Cliquez sur **OK** pour fermer le message de confirmation.

Tous les exemples du projet démo sont importés dans le projet nouvellement créé, et le nom du nouveau projet est affiché dans la liste **Projet** de l'écran de connexion.

Pour importer le projet démo *TALENDDDEMOSJAVA* dans votre répertoire:

1. Cliquez sur **Advanced** et, de la fenêtre de connexion cliquez sur **Demo Project...**. La boîte de dialogue [**Import demo project**] s'ouvre.



2. Sélectionnez le projet démo 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.

3. Cliquez sur **OK** pour fermer le message de confirmation.

Pour ouvrir le projet Démo importé dans *Talend Open Studio for Data Integration*, 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.

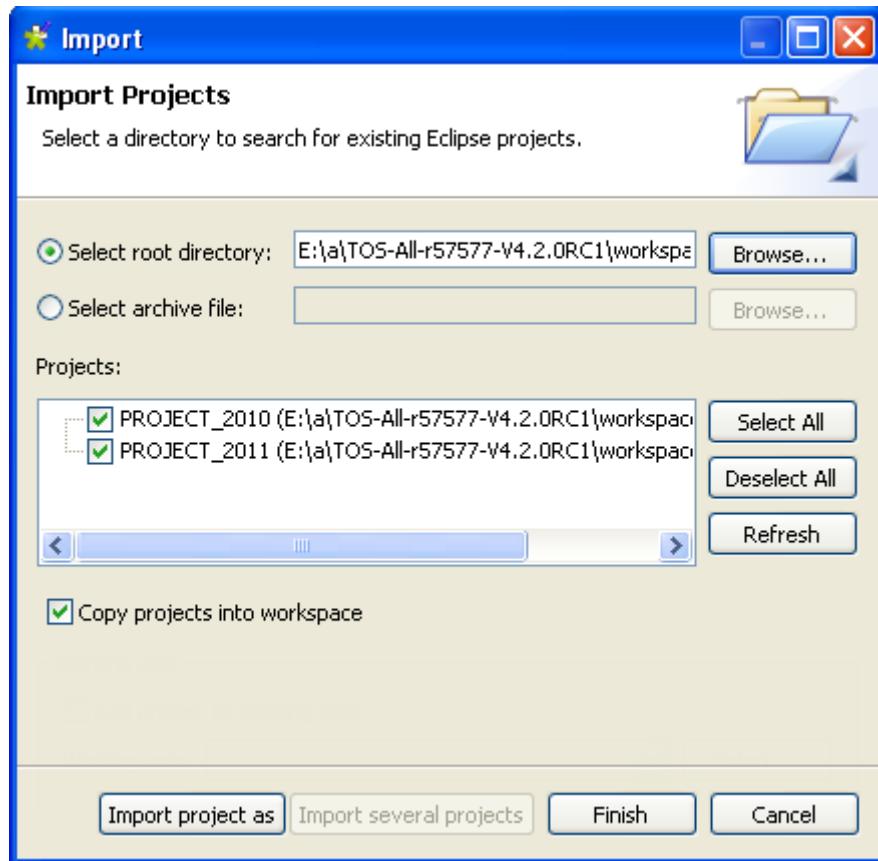
Les exemples de Jobs du projet Démo sont automatiquement importés dans votre workspace et sont disponibles à partir de la vue **Repository** sous le nœud **Job Designs**.

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

2.4.3. Importer un projet

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

1. Si vous lancez *Talend Open Studio for Data Integration* pour la première fois, cliquez sur **Advanced...** pour ouvrir la fenêtre de connexion.
2. A partir de la fenêtre connexion, cliquez sur **Import...** pour ouvrir l'assistant **[Import]**.



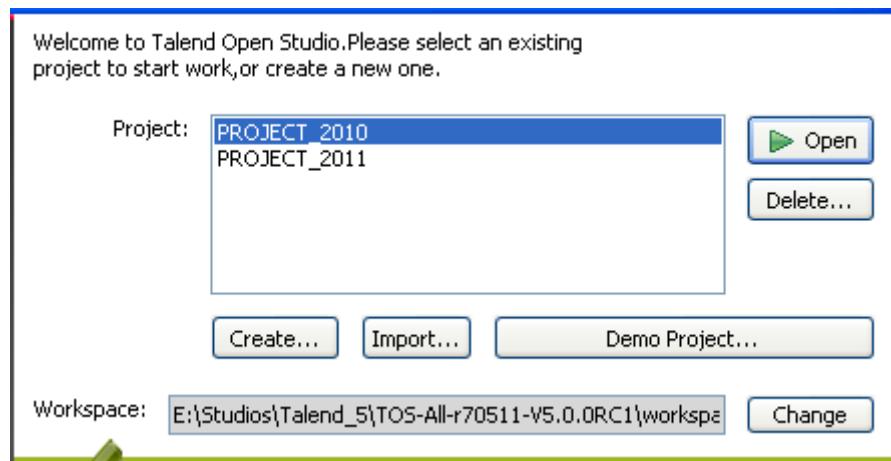
3. Cliquez sur **Import several projects** si vous souhaitez importer plus d'un projet à la fois.
4. Cliquez sur **Select root directory** ou **Select archive file** en fonction du type de source à partir de laquelle vous souhaitez faire l'import : respectivement une répertoire ou un fichier archive.
5. 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 for Data Integration* ou le fichier archive contenant le ou les projets à importer et sélectionnez-le.
6. 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.

7. 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 for Data Integration* et cliquer sur **Open** pour lancer le Studio.

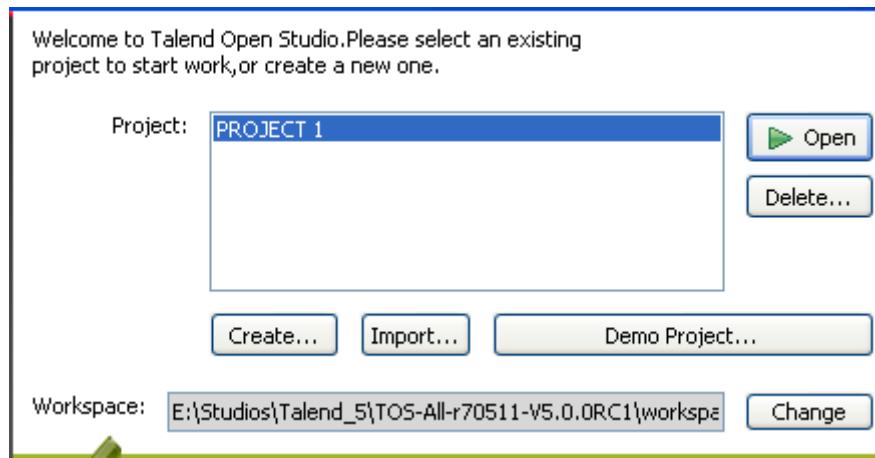
-  Une fenêtre d'initialisation du générateur de code peut apparaître lors du lancement de l'application. Il est recommandé d'attendre que l'initialisation se termine.

2.4.4. Ouvrir un projet

 Lorsque vous lancez *Talend Open Studio for Data Integration* pour la première fois, aucun projet n'apparaît dans la liste **Project**. Vous devez donc créer ou importer un projet 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 for Data Integration* :

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**.

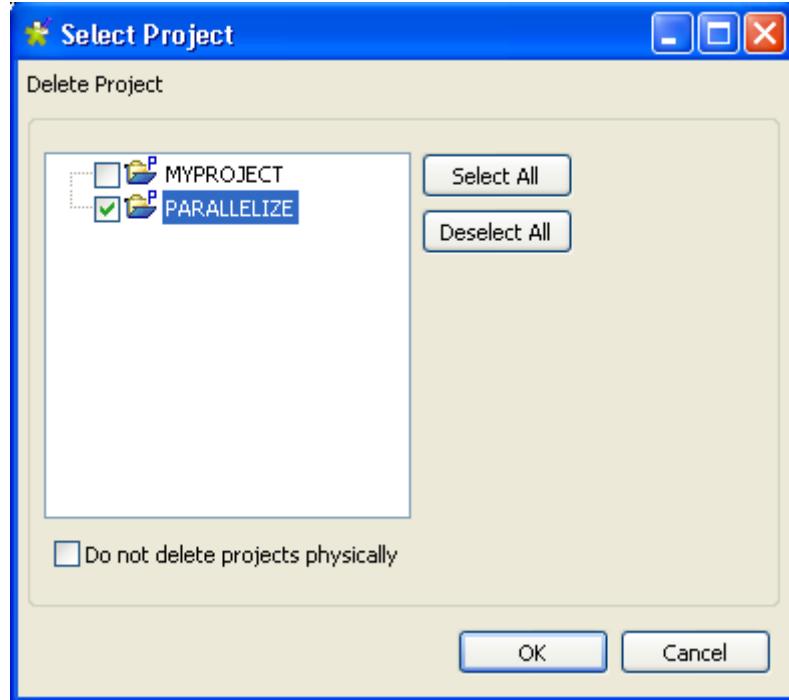


Une barre de progression apparaît puis la fenêtre principale de *Talend Open Studio for Data Integration* s'ouvre. Une fenêtre d'initialisation du générateur de code apparaît. Il est recommandé d'attendre que l'initialisation se termine.

-  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 [Section 2.4.7, « Tâches de migration »](#).

2.4.5. Supprimer un projet

- Dans la fenêtre de connexion, cliquez sur **Delete** pour ouvrir la boîte de dialogue **[Select Project]**.



- Cochez la ou les cases correspondant 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.*

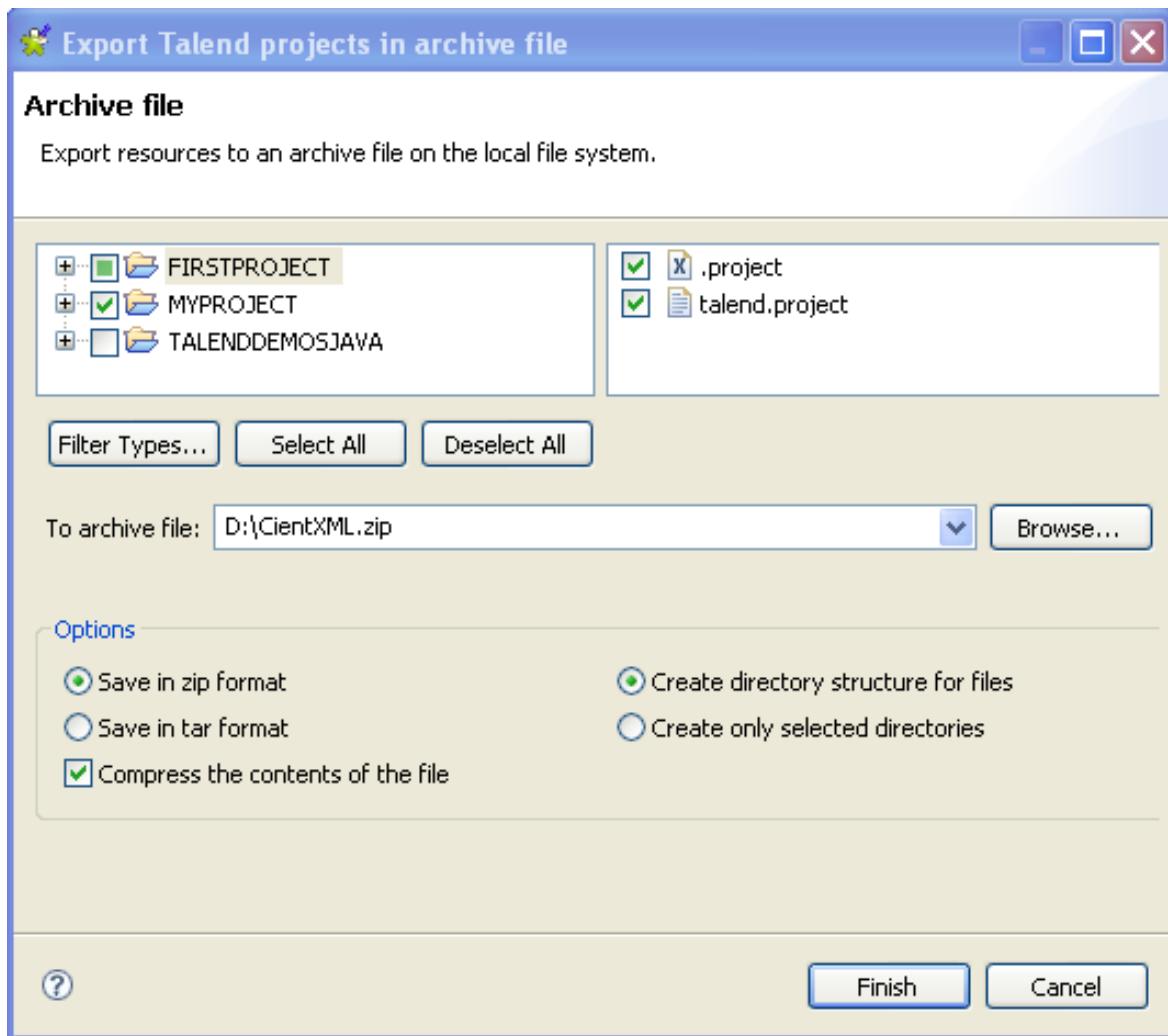


Si vous cochez la case **Do not delete projects physically**, le ou les projet(s) sélectionné(s) seront uniquement supprimés de la liste et seront donc toujours présents dans le répertoire *workspace* de *Talend Open Studio for Data Integration*. De plus, vous pouvez, à tout moment, récupérer un projet 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 for Data Integration vous permet d'exporter les projets créés ou importés dans l'instance courante de *Talend Open Studio for Data Integration*.

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



2. Cochez les cases correspondant aux projets que vous souhaitez exporter. Vous pouvez, en fonction de vos besoins, sélectionner une partie d'un projet à partir du bouton **Filter Types...** (pour les utilisateurs expérimentés).
3. Dans le champ **To archive file**, 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.
4. Dans la zone **Options**, sélectionnez le format de compression et le type de structure que vous souhaitez utiliser.
5. Cliquez sur **Finish** pour valider.

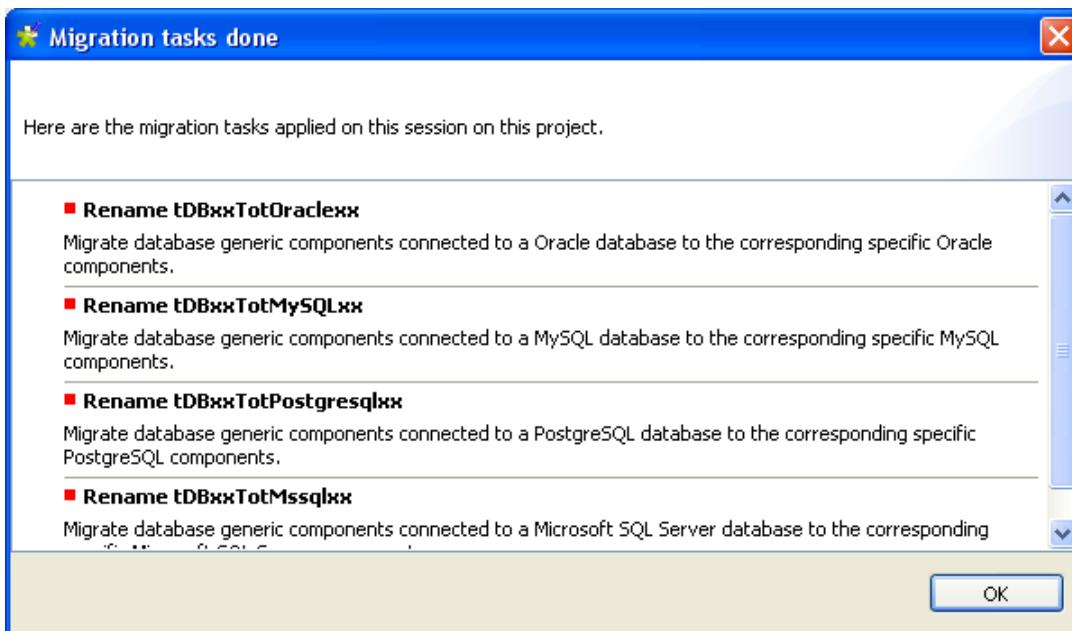
Le fichier archive contenant les projets exportés est créé dans le répertoire spécifié.

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 for Data Integration* 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 for Data Integration*. 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 for Data Integration*, 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. Configuration des préférences de *Talend Open Studio for Data Integration*

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

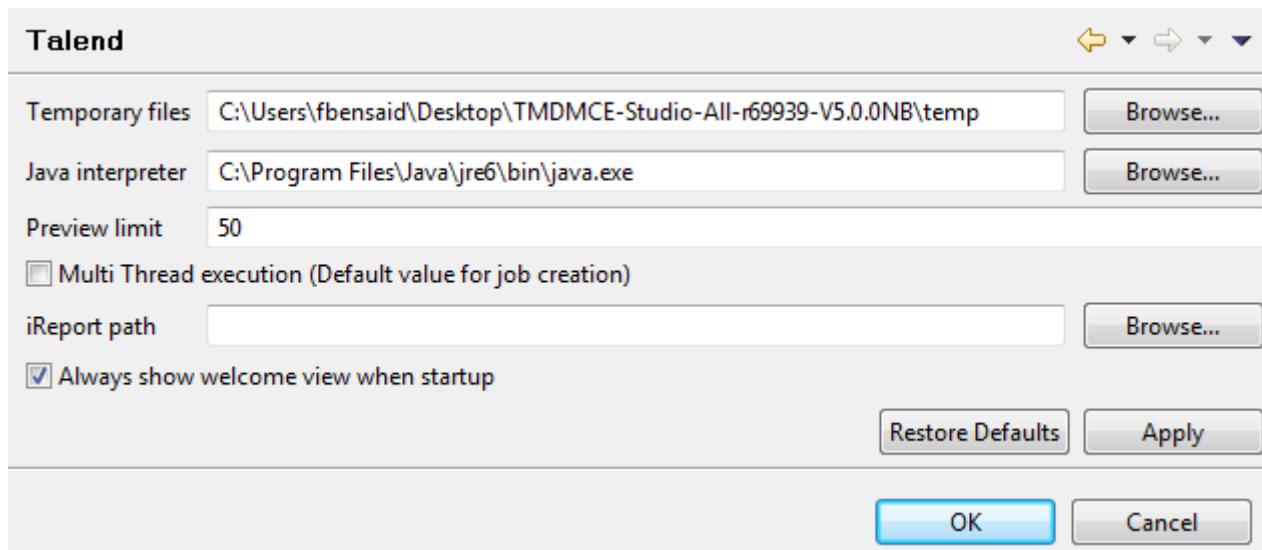
Un grand nombre des configurations que vous avez définies peuvent être enregistrées dans **Preference** et seront donc utilisées par défaut pour tous les nouveaux Jobs que vous créerez.

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

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

2.5.1. Interpréteur Java

Le chemin de l'interpréteur Java est défini par défaut dans le fichier Java de votre ordinateur (Program Files \Java\jre6\bin\java.exe par défaut).



Pour personnaliser votre chemin d'accès à l'interpréteur Java :

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

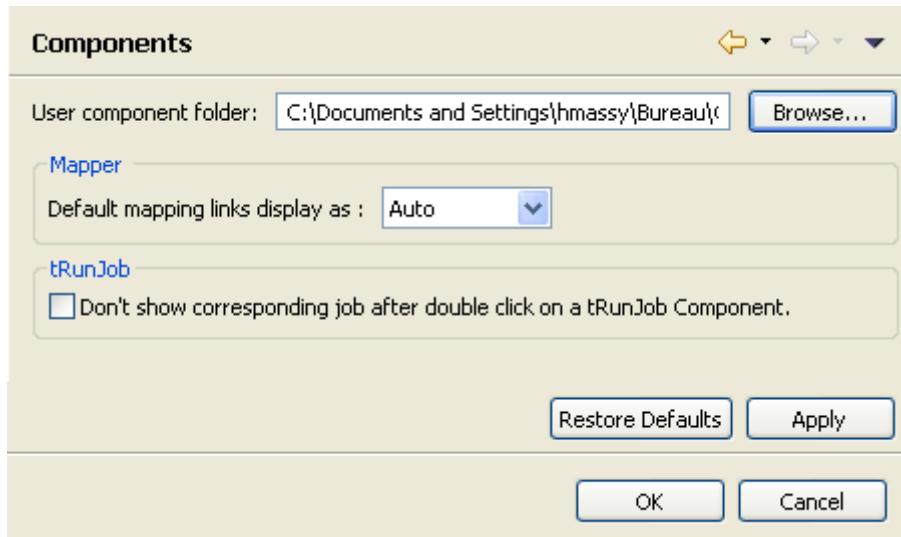
Dans la même vue, vous pouvez également modifier le nombre de lignes de données affichées dans l'aperçu et le chemin d'accès vers les fichiers temporaires.

2.5.2. Composants externes ou utilisateur

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

Pour plus d'informations concernant la création et le développement de composants utilisateur, consultez la page wiki (en anglais) [Component creation tutorial section](http://www.talendforge.org/wiki/). [<http://www.talendforge.org/wiki/>]

1. Dans l'arborescence de la boîte de dialogue **[Preferences]**, développez le nœud **Talend**, puis sélectionnez **Components**.



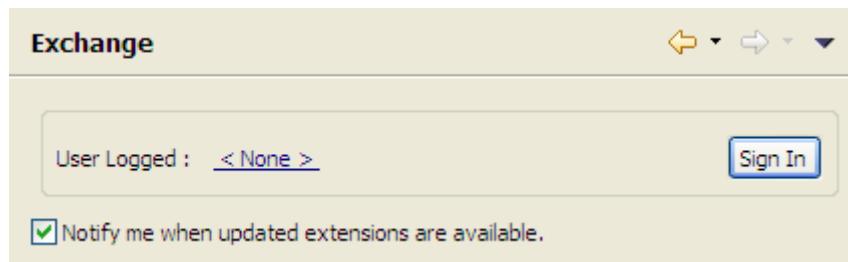
2. Renseignez le champ **User components folder** pour indiquer le chemin d'accès aux composants à ajouter à la **Palette** de *Talend Open Studio for Data Integration*.
 3. Dans le champ **Default mapping links dispaly as**, sélectionnez le type de liens que vous souhaitez utiliser pour la mise en correspondance dans le **tMap**.
 4. 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**.
5. 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. Préférence Exchange

Vous pouvez configurer les préférences relatives à votre connexion à **Talend Exchange**, site faisant partie de la Communauté **Talend**, dans *Talend Open Studio for Data Integration*. Pour ce faire :

1. Dans la barre de menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [**Preferences**].
2. Développez le nœud **Talend** et cliquez sur **Exchange** afin d'afficher la vue **Exchange**.



3. Configurez les préférences d'**Exchange** selon vos besoins :

- a. Si vous n'êtes pas encore connecté à la Communauté **Talend**, cliquez sur **Sign In** pour vous rendre sur la page **Connect to TalendForge** afin de vous connecter à la Communauté **Talend** à l'aide de votre identifiant/mot de passe ou pour vous créer un compte et de vous y connecter.

Si vous êtes déjà connecté à la Communauté **Talend**, votre compte s'affiche et le bouton **Sign In** devient **Sign Out**. Pour vous déconnecter de la Communauté **Talend**, cliquez sur **Sign Out**.

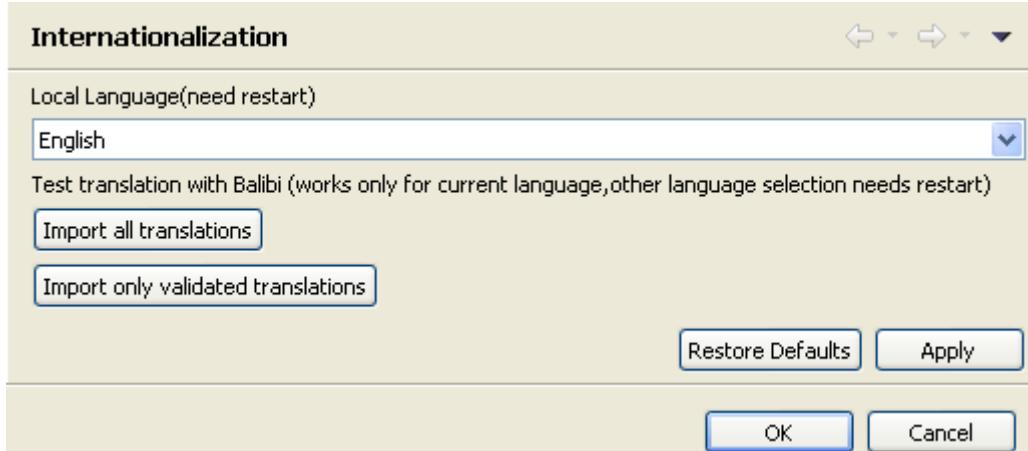
- b. Par défaut, lorsque vous êtes connecté à la Communauté **Talend**, lorsqu'une mise à jour d'un produit installé est disponible, une boîte de dialogue apparaît pour vous le signaler. Si vous vérifiez fréquemment les mises à jour disponibles et que vous ne souhaitez pas que cette boîte de dialogue réapparaisse, décochez la case **Notify me when updated extensions are available**.

Pour plus d'informations concernant la connexion à la Communauté **Talend**, consultez [Section 2.2, « Lancement de Talend Open Studio for Data Integration »](#). Pour plus d'informations concernant l'utilisation des composants de la communauté dans le Studio, consultez la [Section 4.5.3, « Télécharger/charger des composants de la communauté »](#).

2.5.4. Préférences de langue

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

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

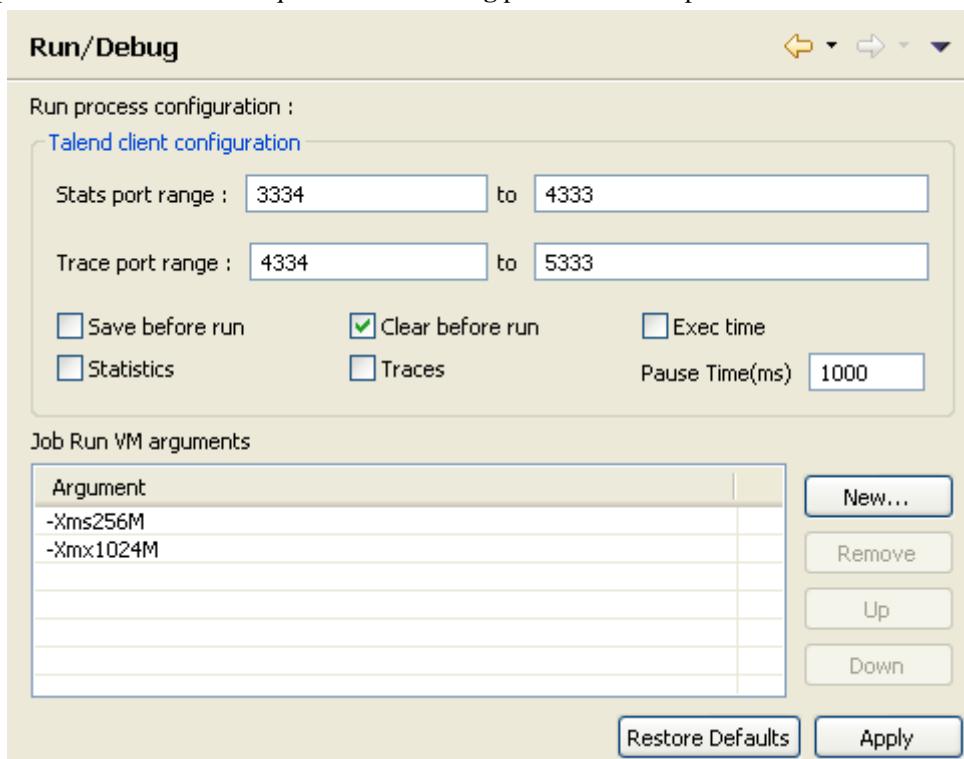


3. Dans le champ **Local Language**, sélectionnez la langue que vous souhaitez utiliser pour l'interface de *Talend Open Studio for Data Integration*.
4. Cliquez sur **Apply**, puis sur **OK** pour valider vos changements et fermer la boîte de dialogue [**Preferences**].
5. Redémarrez *Talend Open Studio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

2.5.5. Préférences d'exécution et de débogage

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

1. Dans le menu, cliquez sur **Window > Preferences** pour afficher la boîte de dialogue [**Preferences**].
2. Développez le nœud **Talend** et cliquez sur **Run/Debug** pour afficher les préférences.



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

Stats port range	Spécifiez une plage pour les ports utilisés pour la génération des Statistics, notamment si les ports définis par défaut sont déjà occupés par d'autres applications.
Trace port range	Spécifiez une plage pour les ports utilisés pour la génération des Traces, notamment si les ports définis par défaut sont déjà occupés par d'autres applications.
Save before run	Cochez cette case pour automatiquement enregistrer votre Job avant de l'exécuter.
Clear before run	Cochez cette case pour nettoyer les résultats d'une exécution précédente avant d'exécuter de nouveau le Job.
Exec time	Cochez cette case pour afficher la durée d'exécution du Job.
Statistics	Cochez cette case le suivi de flux au cours de l'exécution du Job.
Traces	Cochez cette case pour afficher le suivi du traitement des données au cours de l'exécution du Job.
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 [Section 4.2.7.4, « Configurer les paramètres d'exécution avancés »](#).

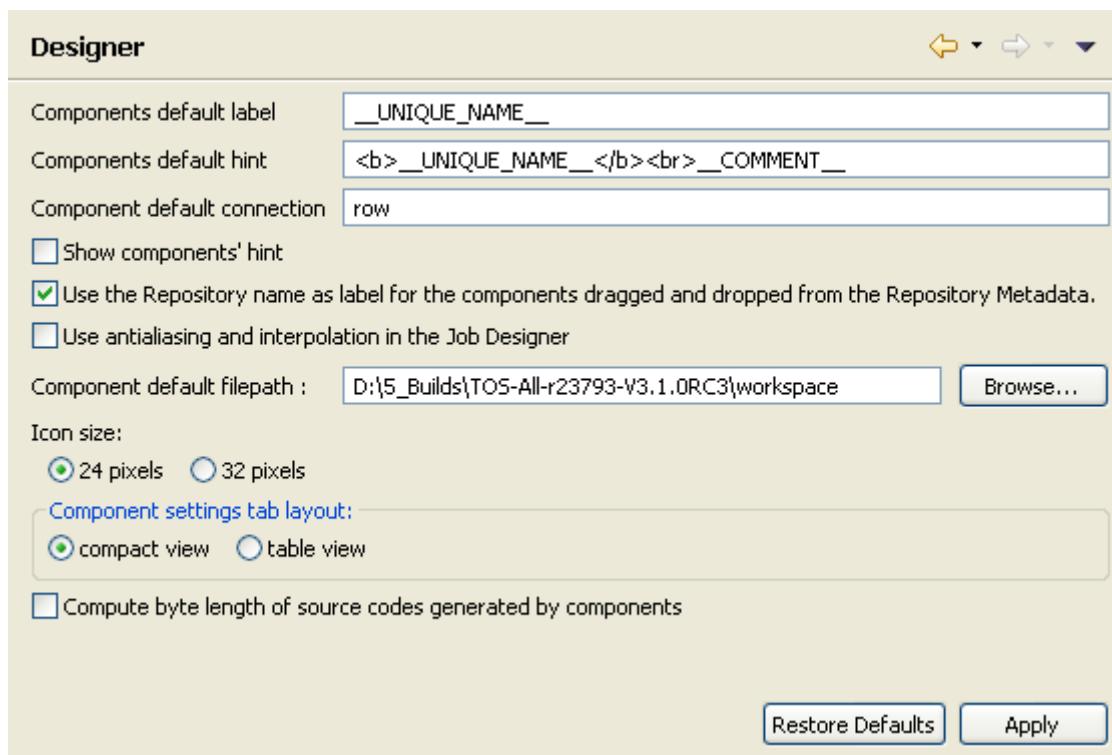
Pour plus d'informations sur les paramètres possibles, consultez le site <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>.

2.5.6. Préférences du Designer

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

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

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

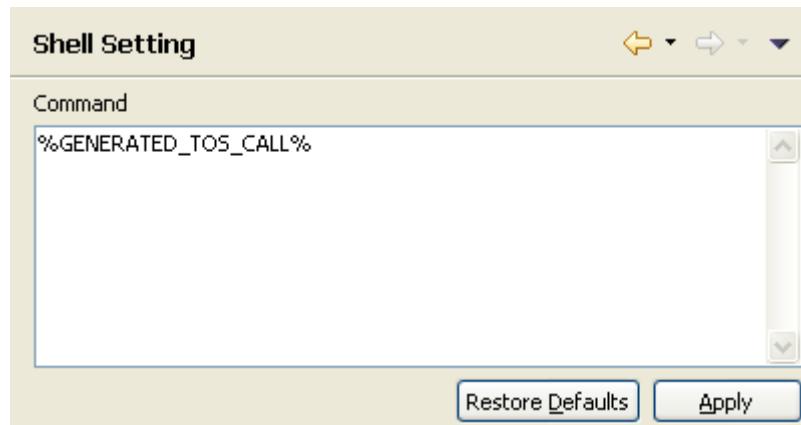


4. Cochez les cases adéquates pour personnaliser l'espace de modélisation de *Talend Open Studio for Data Integration* en fonction de votre utilisation.

2.5.7. 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.

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

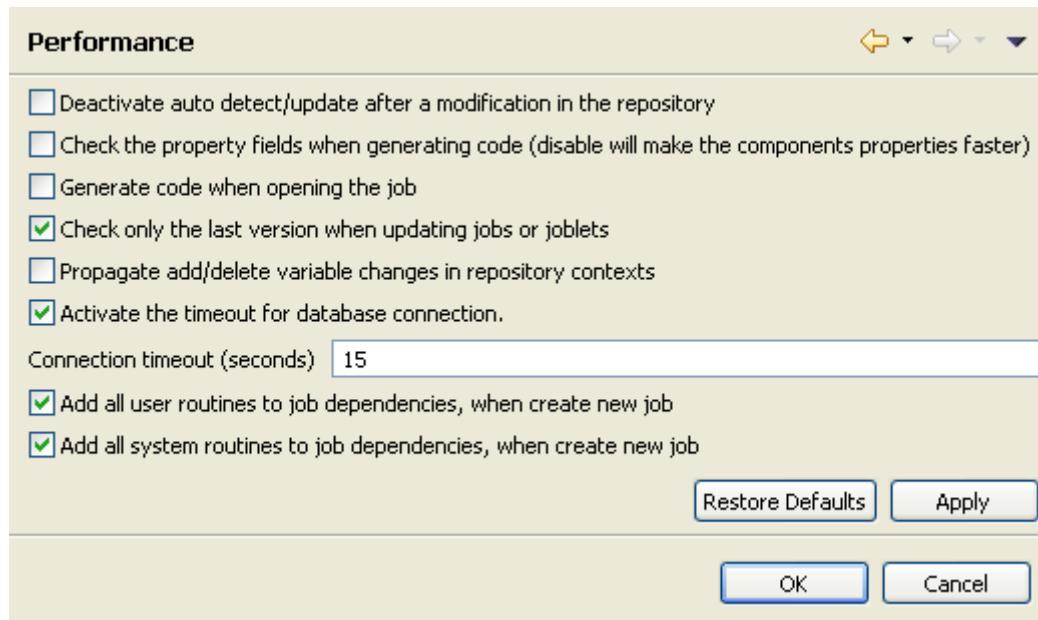


3. Dans le champ **Command**, ajoutez votre/vos ligne(s) de code avant ou après le code %GENERATED_TOS_CALL% pour qu'elles apparaissent respectivement au début ou à la fin du code de votre Job.

2.5.8. Préférences de Performance

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

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



La désactivation du rafraîchissement automatique permet un gain de performance.

3. Paramétrez les préférences de performance de *Talend Open Studio for Data Integration* en fonction de votre utilisation :

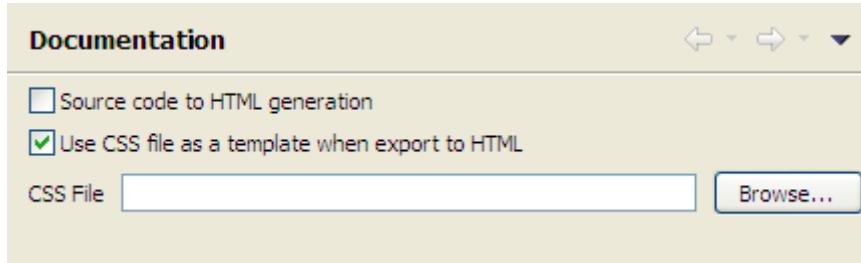
- Cochez la case **Deactivate auto detect/update after a modification in the repository** pour désactiver la détection et la mise à jour automatique du Repository après modification.
- Cochez la case **Check the property fields when generating code** pour activer la vérification des champs de propriétés des composants. Lorsque l'un des champs de propriétés d'un composant est mal renseigné, le composant est entouré en rouge.
- 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.

2.5.9. Préférences de la documentation

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

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



3. Personnalisez les préférences de la documentation selon vos besoins :

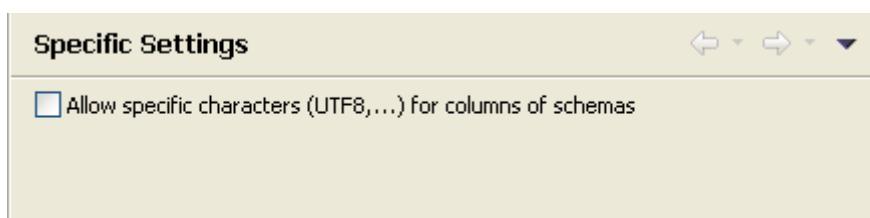
- Cochez la case **Source code to HTML generation** pour intégrer le code source dans la documentation HTML que vous générerez.
- Cochez la case **Use CSS File as a template when export to HTML** afin d'activer le champ **CSS File**, si vous devez utiliser un fichier CSS pour personnaliser les fichiers HTML exportés.

Pour plus d'informations concernant la documentation, consultez [Section 5.6.1, « Générer la documentation HTML »](#) et [Section 4.2.6.5, « Onglet Documentation »](#).

2.5.10. 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 for Data Integration*. Pour ce faire :

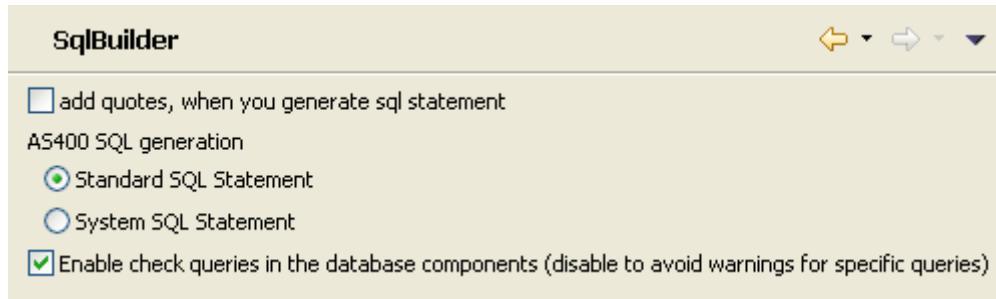
1. Dans la barre de menu, cliquez sur **Window > Preferences** afin d'ouvrir la boîte de dialogue [**Preferences**].
2. Dans l'arborescence de la boîte de dialogue, développez le noeud **Talend**.
3. Cliquez sur le noeud **Specific settings** pour afficher la vue correspondante dans la partie droite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



2.5.11. Préférences du SQL Builder

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

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



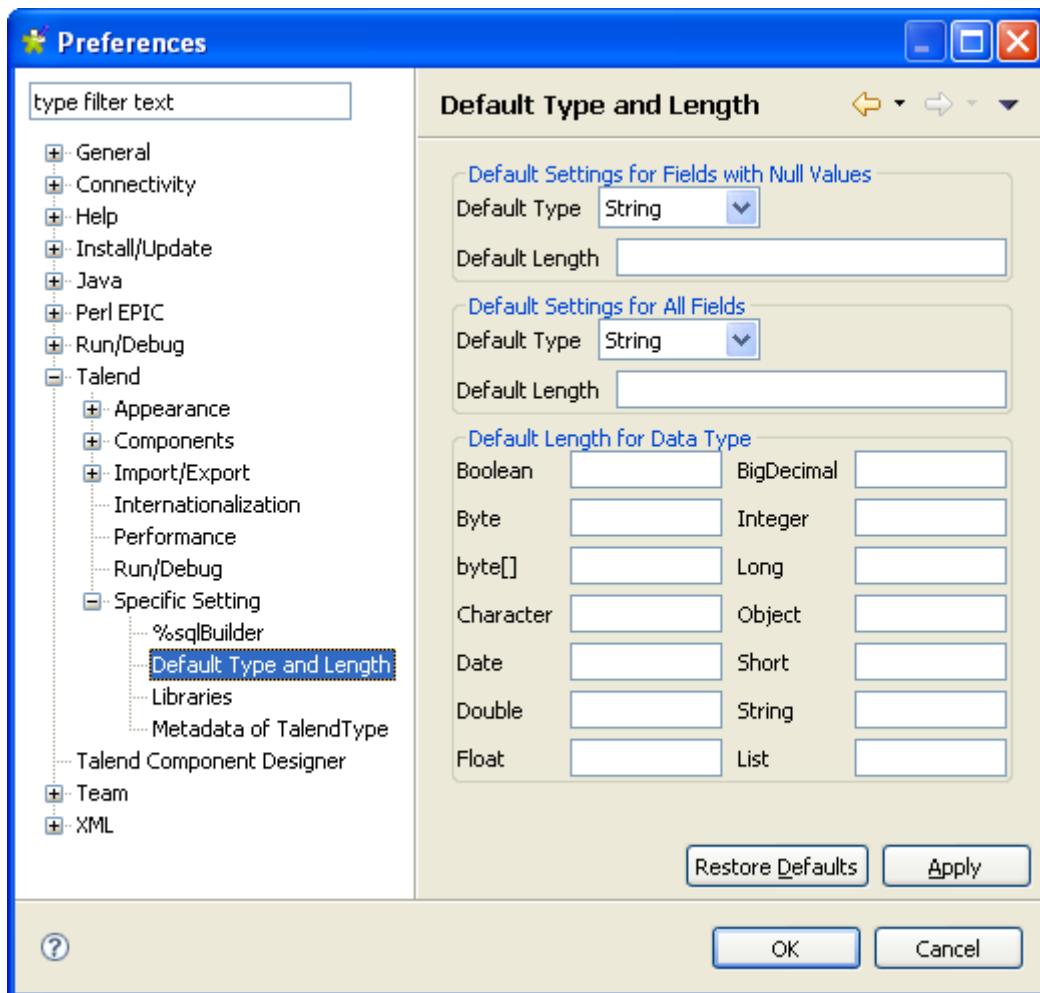
3. Personnalisez les performances du SQL Builder selon vos besoins :

- Cochez la case **add quotes, when you generated sql statement** pour protéger le nom des colonnes et des tables par des guillemets dans vos requêtes SQL.
- Pour le champ **AS400 SQL generation**, sélectionnez **Standard SQL Statement** pour utiliser des commandes SQL standard ou **System SQL Statement** pour des commandes SQL system lorsque vous utilisez des bases de données de type 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.

2.5.12. 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.

1. Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [**Preferences**].
2. Développez les nœuds **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.



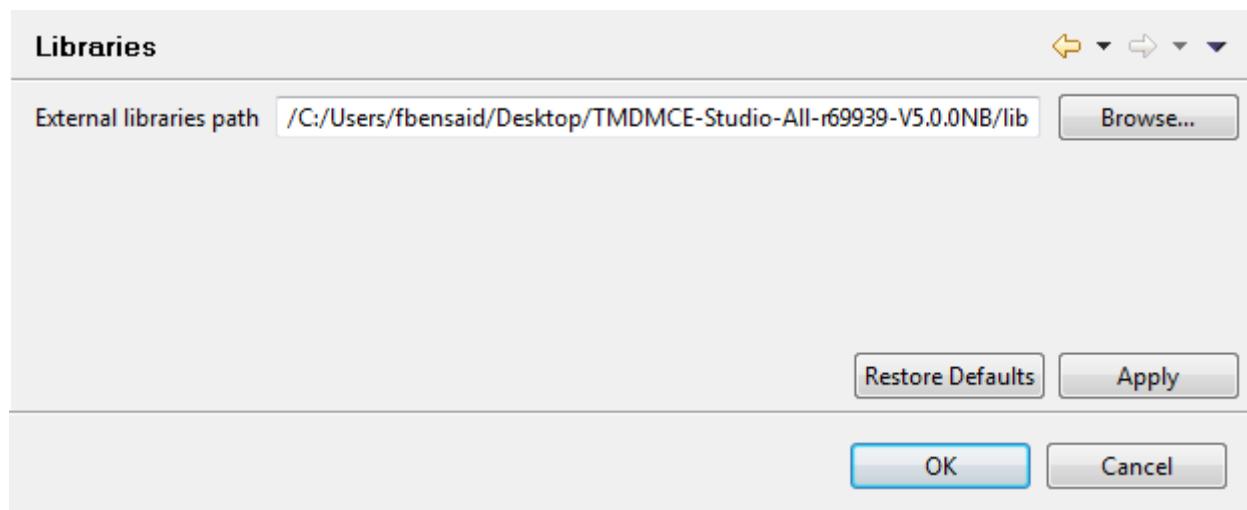
3. Paramétrez ces préférences en fonction de vos besoins :

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

2.5.13. Préférences des bibliothèques

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

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

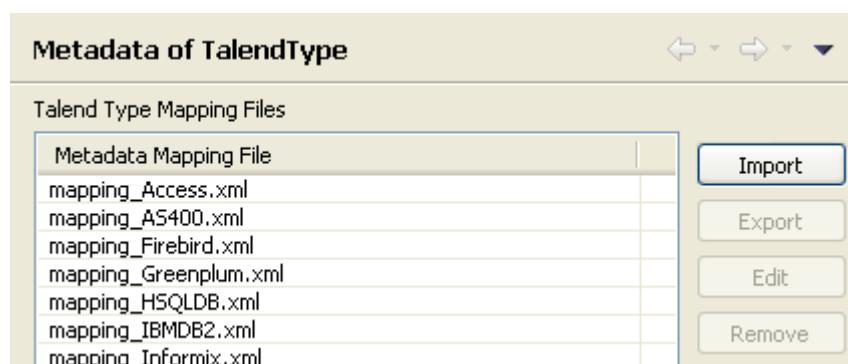


3. Le chemin d'accès peut être indiqué dans le champ **External libraries path** en cliquant sur le bouton **Browse...**.
Le chemin par défaut est celui de la version actuelle que vous utilisez.

2.5.14. Préférences des conversions de type

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

1. Dans le menu, cliquez sur **Window > Preferences** pour ouvrir la boîte de dialogue [**Preferences**].
2. Développez les nœuds **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 for Data Integration*.

- 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**].

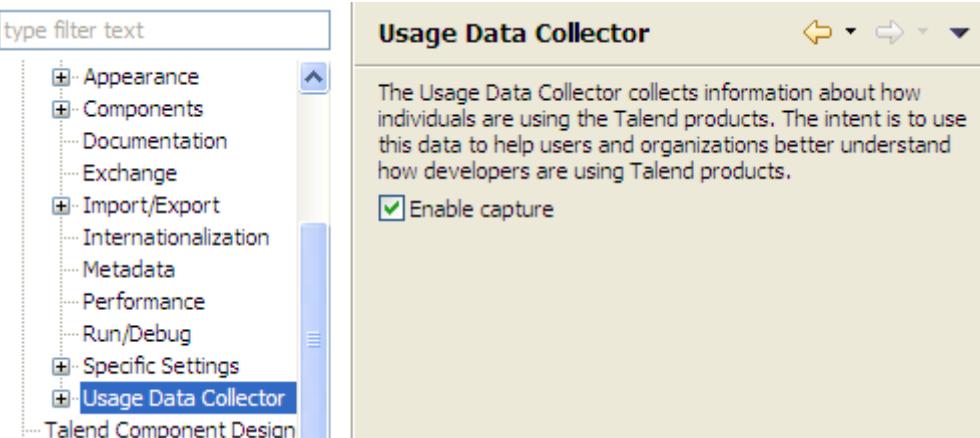
2.5.15. Préférences du collecteur de données d'utilisation

En autorisant *Talend Open Studio for Data Integration* à collecter vos statistiques d'utilisation du Studio, vous permettez aux utilisateurs de mieux comprendre les produits **Talend** et vous permettez à **Talend** de savoir comment les utilisateurs utilisent les produits. Cela rend possible l'amélioration de la qualité des produits et des performances afin de mieux répondre aux besoins des utilisateurs.

Par défaut, *Talend Open Studio for Data Integration* collecte automatiquement vos données d'utilisation et les envoie régulièrement aux serveurs hébergés par **Talend**. Vous pouvez voir la collection de données d'utilisation et le chargement d'informations, ainsi que personnaliser les performances du collecteur de données selon vos besoins.

-  Soyez assuré que seules les statistiques d'utilisation du Studio sont collectées. Aucune de vos informations privées ne sera collectée et transmise à **Talend**.

1. Dans la barre de menu, cliquez sur **Window > Preferences** afin d'afficher la boîte de dialogue [**Preferences**].
2. Développez le nœud **Talend** et cliquez sur **Usage Data Collector** pour afficher la vue **Usage Data Collector**.



3. Lisez le message concernant le collecteur de données d'utilisation et, si vous ne souhaitez pas que le collecteur collecte et envoie vos informations d'utilisation du Studio, décochez la case **Enable capture**.
4. Pour obtenir un aperçu des données d'utilisation capturées par le collecteur de données, développez le nœud **Usage Data Collector** et cliquez sur **Preview**.

Preview	
The following Keys/Values have been captured by the Usage Data Collector.	
Key	Value
tokenStudio	
typeStudio	tos
uniqueId	RuRU/jefScKhad/4RS3ElOpJz+KTLzG9crYBBuA7...
version	5.0.0.RC1_r70511
properties	
tos.count.businessModels	0
tos.count.componentsPerJob	0.0
tos.count.contextVariablesPerJob	0.0
tos.count.debugRuns	0
tos.count.generatedJobDocs	0
tos.count.jobExports	0
tos.count.jobs	0
tos.count.jobsPerProject	0.0
tos.count.localProjects	1
tos.count.metadata	0
tos.count.runs	0
tos.count.userComponents	0
tos.job.frequentComponents	<Empty>
tos.user.components	<Empty>

5. Pour personnaliser l'intervalle d'envoi des données d'utilisation et voir la date du dernier envoi, cliquez sur **Uploading** sous le nœud **Usage Data Collector**.

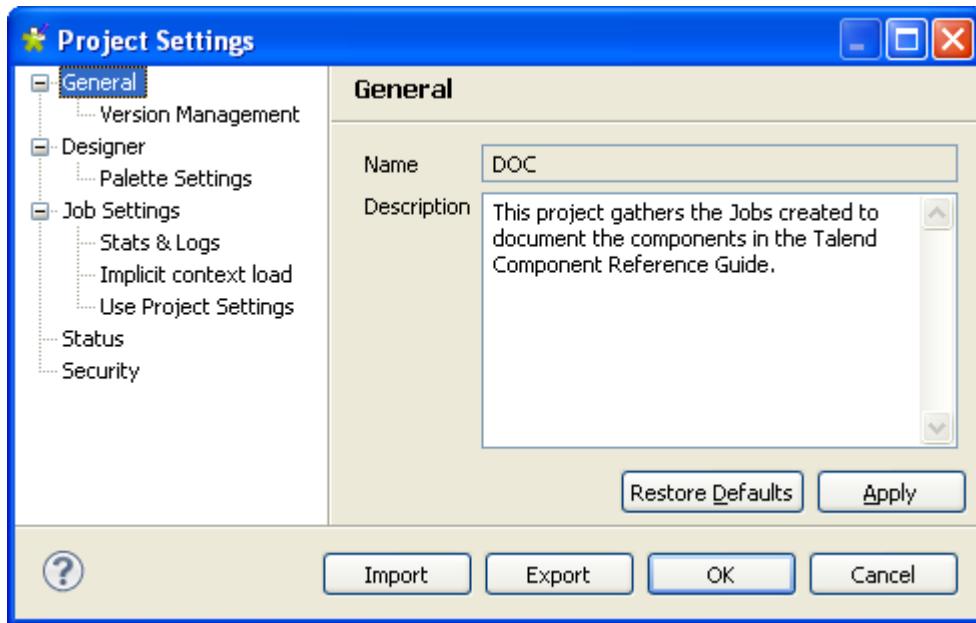
Uploading	
Information gathered by the Usage Data Collector is periodically uploaded to servers hosted by Talend.	
Upload Period	10 Days
Last Upload	

- Par défaut, s'il est activé, le collecteur de données collecte les données d'utilisation du produit et les envoie aux serveurs *Talend* tous les dix jours. Pour modifier l'intervalle, saisissez une nouvelle valeur entière (en jours) dans le champ **Upload Period**.
- Le champ **Last Upload** en lecture seule affiche la date et l'heure du dernier envoi de données aux serveurs *Talend*.

2.6. Personnalisation des paramètres du projet

Talend Open Studio for Data Integration 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.

1. 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.



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

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

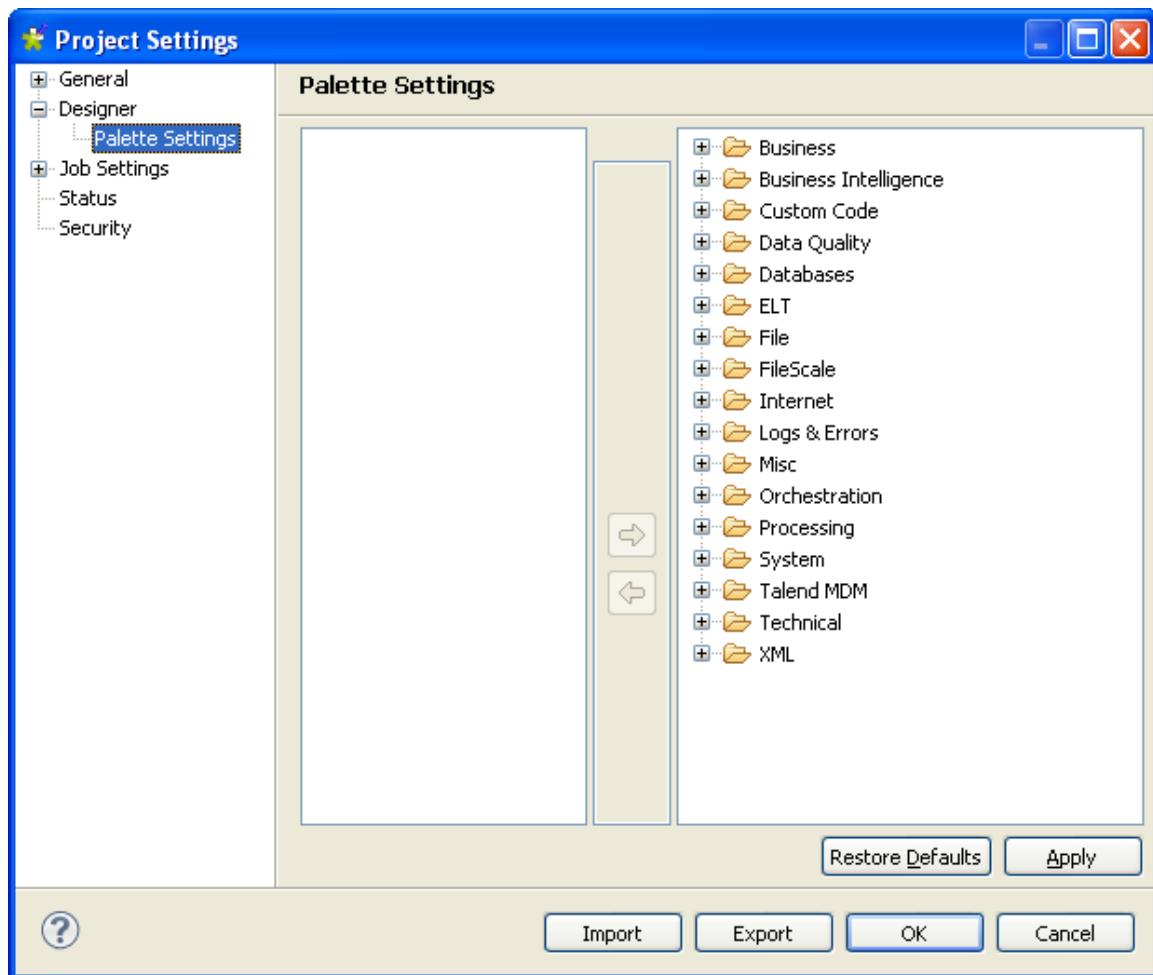
- Pour exporter les paramètres, cliquez sur le bouton **Export**. L’export générera un fichier XML contenant tous les paramètres de votre projet.
- Pour importer les paramètres, cliquez sur le bouton **Import** et sélectionnez le fichier XML contenant les paramètres d’un autre projet à appliquer au projet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

Pour personnaliser l’affichage de la Palette, procédez comme suit :

1. 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.

2. 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.
3. Sélectionnez un ou plusieurs composants, ou un ou plusieurs groupes de composants pour les supprimer de la **Palette** en cours.
4. Puis utilisez la flèche vers la gauche pour déplacer la sélection vers la **Palette** de composants cachés, à gauche de la fenêtre. Notez que l’arborescence de la **Palette** est conservée dans la zone de gauche.
5. 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.
6. 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**.

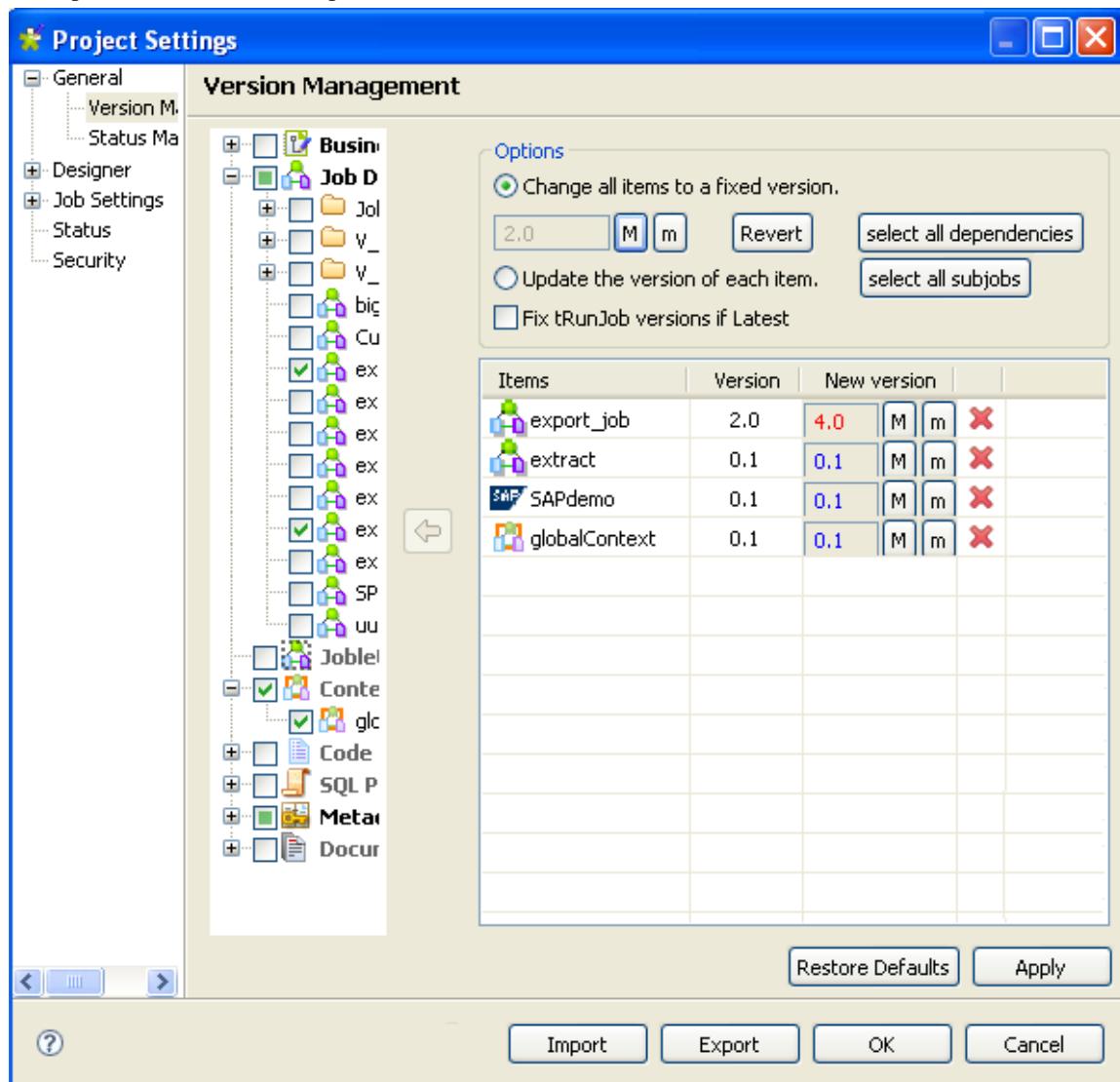
Pour plus d’informations concernant la **Palette**, consultez [Section 4.2.8.1, « Modifier la disposition et les paramètres de la Palette »](#).

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 nœud **General** et cliquez sur **Version Management**.

Pour ce faire :

1. 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]**.
2. Dans l'arborescence de la boîte de dialogue, développez le nœud **General** et cliquez sur **Version Management** pour ouvrir la vue correspondante.



3. Dans l'arborescence, développez le nœud correspondant aux éléments dont vous souhaitez gérer les versions et cochez les cases correspondantes.

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**.

4. Effectuez les modifications suivantes :

- 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 père conserve le Job fils de la version actuelle dans le **tRunJob** à versionner, sans tenir compte de la mise à jour de leurs versions. Par exemple, un **tRunJob** fera une mise à jour à partir de la version actuelle *1.0* vers *1.1* à la fois au niveau du Job père et du Job fils. Une fois cette case cochée, le Job père *1.0* continuera à utiliser le Job fils *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)es Job(s) fils. Pour plus d'informations concernant le **tRunJob**, consultez le **Guide de référence des Composants Talend Open Studio**.*

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

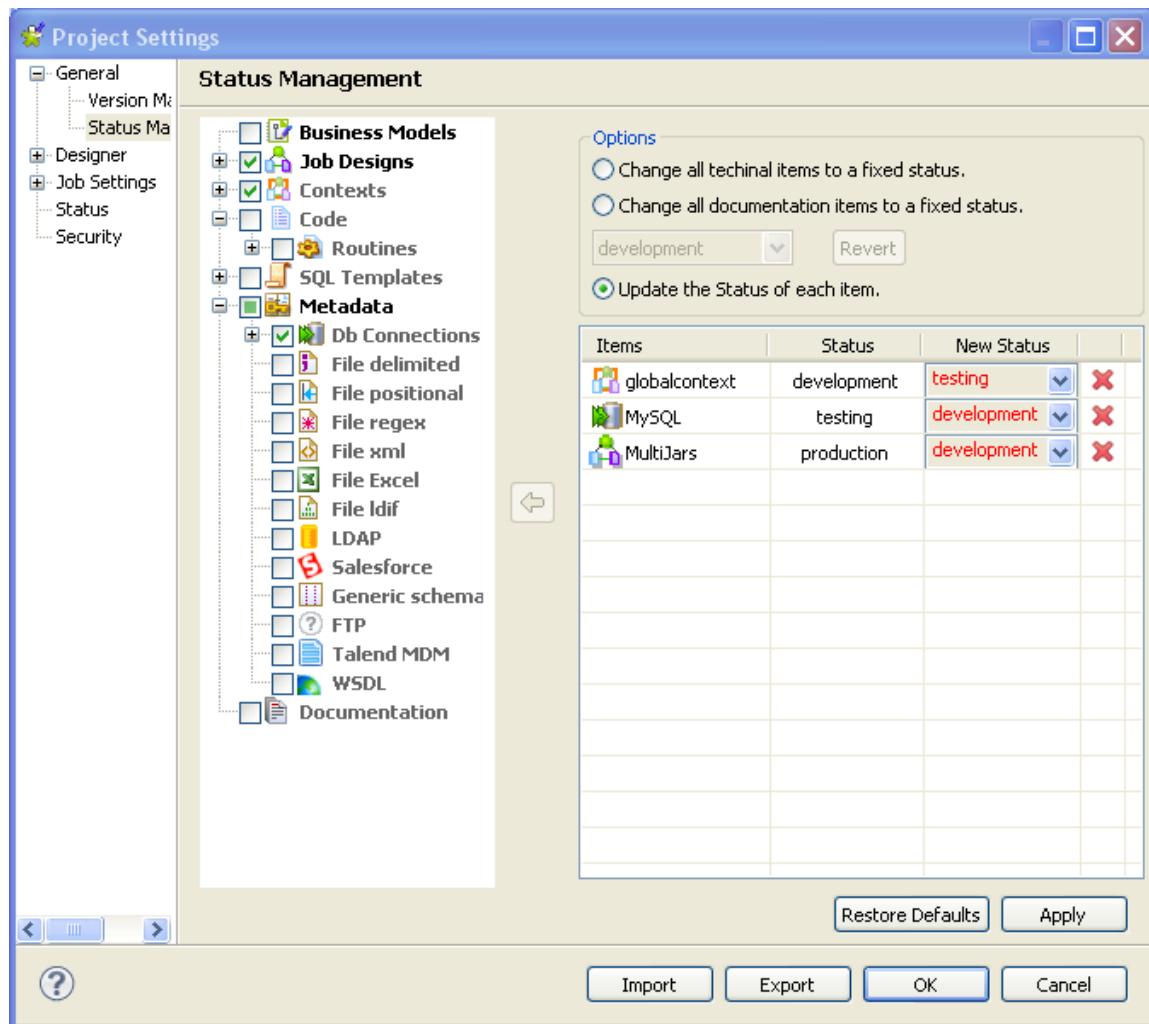


Pour plus d'informations concernant la gestion des versions, consultez [Section 2.6.2, « Gérer les versions »](#).

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]**.

1. 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]**.
2. Dans l'arborescence de la boîte de dialogue, développez le nœud **General** puis sélectionnez **Status Management** afin d'ouvrir la vue correspondante.



3. Dans la vue **Repository**, développez le nœud contenant les éléments dont vous souhaitez gérer le statut et cochez la case de ces éléments.

Les éléments sélectionnés s'affichent dans la liste **Items** de droite, avec leur statut, dans la colonne **Status**. Le nouveau statut défini s'affiche dans la colonne **New Status**.

4. Dans la zone **Options**, cochez la case **Change all technical items to a fixed status** pour changer le statut de tous les éléments sélectionnés en un même statut fixe.
5. Cliquez sur **Revert** si vous souhaitez annuler les modifications.
6. Pour mettre à jour le statut des éléments, sélectionnez l'option **Update the version of each item** et changez-le manuellement.
7. Cliquez sur **Apply** afin d'appliquer vos modifications, puis sur **OK** pour fermer la boîte de dialogue.

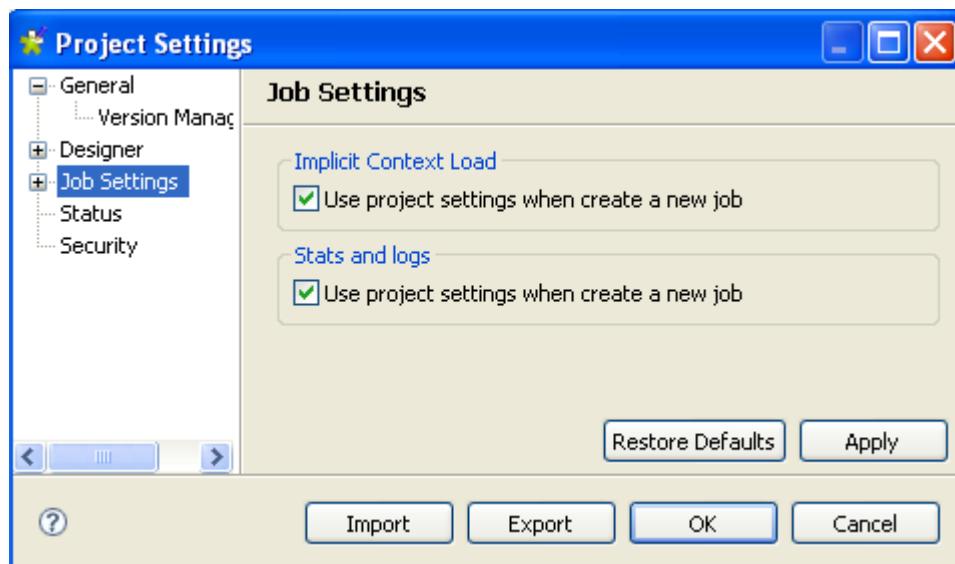
 Pour plus d'informations concernant les statuts des Jobs, consultez [Section 2.6.4, « Personnaliser les paramètres du Job »](#).

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 :

1. 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]**.
2. Dans l'arborescence de la boîte de dialogue, cliquez sur le noeud **Job Settings** pour ouvrir la vue correspondante.
3. Cochez la case **Use project settings when create a new job** des zones **Implicit Context Load** et **Stats and Logs**.



4. Cliquez sur **Apply** pour valider 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 recueillies dans des fichiers CSV ou dans une base de données.

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

Pour ce faire :

1. 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]**.
2. Dans l'arborescence de la boîte de dialogue, développez le noeud **Job Settings** et cliquez sur **Stats & Logs** pour afficher la vue correspondante.

The screenshot shows the 'Stats & Logs' configuration dialog. At the top, there are three checked checkboxes: 'Use statistics (tStatCatcher)', 'Use logs (tLogCatcher)', and 'Use volumetrics (tFlowMeterCatcher)'. Below these are two unchecked checkboxes: 'On Console' and 'On Files'. Under 'File Path', the path is set to 'D:/5_Builds/TIS_EE-All-r23530-V3.1.0RC2/TIS_EE-All-r23530-V3.1.0RC2/workspace/.metadata'. The 'Stats File Name' is 'stats_file.txt', 'Log File Name' is 'logs_file.txt', and 'Meter file name' is 'meter_file.txt'. In the 'On Databases' section, 'Property Type' is set to 'Repository' (selected from a dropdown) and 'DB (MYSQL):localhost' (selected from another dropdown). 'Db Type' is set to 'MySQL'. 'Host' is 'localhost', 'Port' is '3306'. 'Db Name' is 'talend', with 'Additional parameters' set to 'noDatetimeStringSync=true'. 'User' is 'root', with 'Password' masked as '*****'. Under 'Tables', 'Stats Table' is 'statistics', 'Logs Table' is 'logs', and 'Meter Table' is 'volumetrics'. At the bottom, there are three checked checkboxes: 'Catch runtime errors', 'Catch user errors', and 'Catch user warnings'. An unchecked checkbox 'Catch components statistics (tStatCatcher Statistics)' is also present. At the very bottom right are 'Restore Defaults' and 'Apply' buttons.



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 [Section 4.6.7.1, « Automatiser l'utilisation de statistiques & logs »](#).

3. Cochez les cases **Use statistics**, **Use logs** et **Use volumetrics** en fonction de vos besoins, puis renseignez le chemin d'accès des données de log.
4. Sélectionnez 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.
5. Cliquez sur **Apply** pour appliquer les changements et sur **OK** pour fermer la boîte de dialogue.

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.



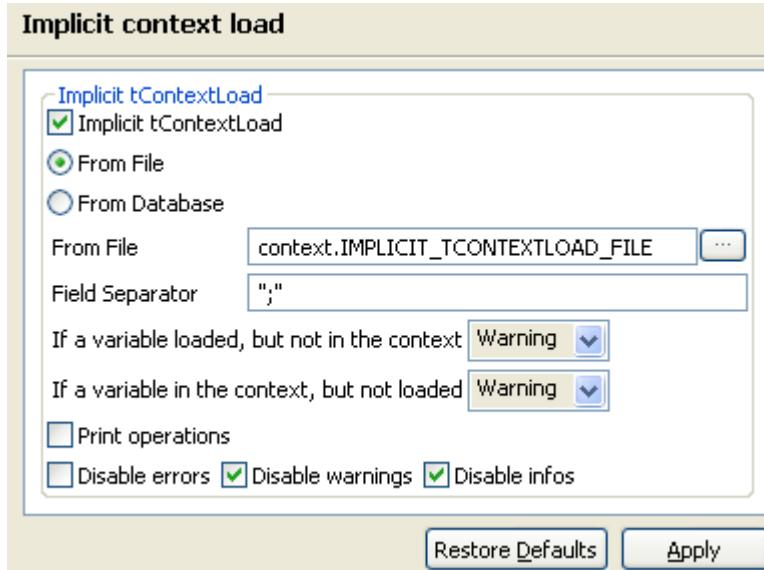
Si vous avez sauvé les informations de connexion dans une variable de contexte, vous pouvez y accéder via le raccourci **Ctrl+Espace**.

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.

Pour ce faire :

1. 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]**.
2. Dans l'arborescence de la boîte de dialogue, développez le nœud **Job Settings** et cochez la case **Implicit tContextLoad** pour afficher les paramètres de configuration de l'option **Implicit tContextLoad**.



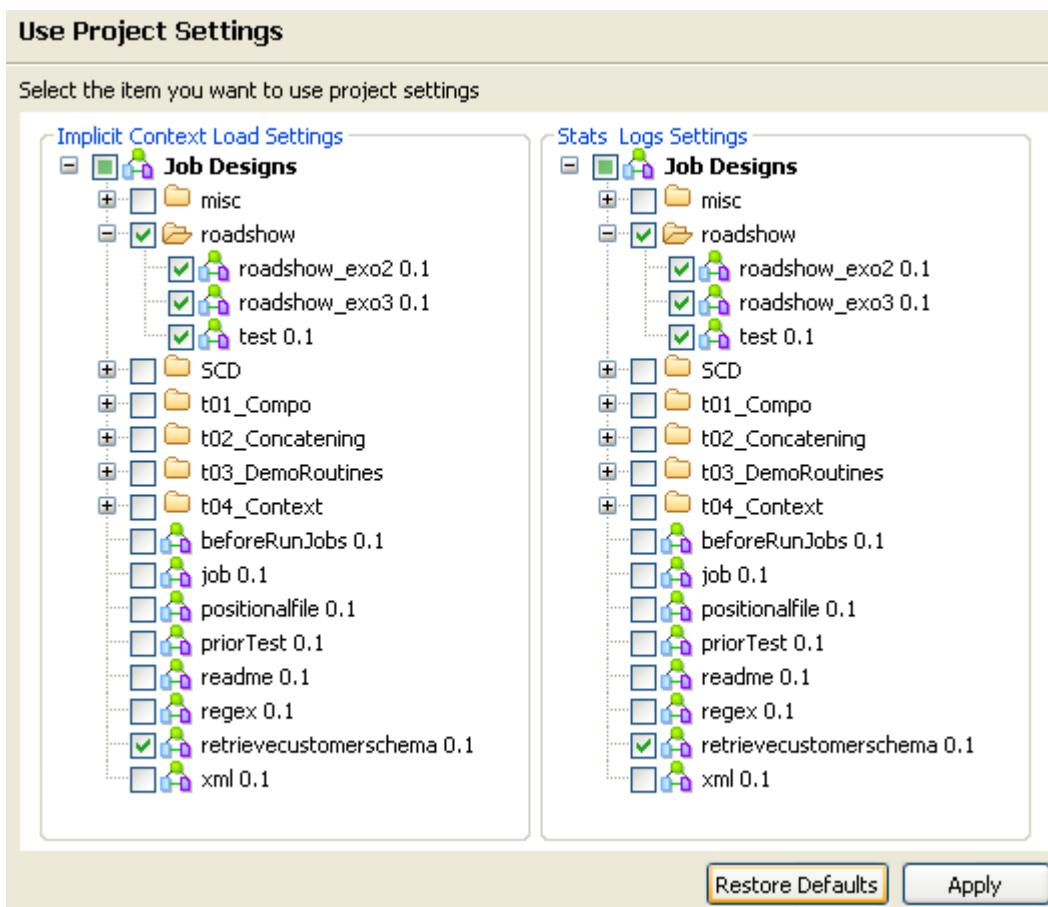
3. Sélectionnez le type de fichier dans lequel les données de contextes seront conservées, soit sous forme de fichier en sélectionnant le champ **From File**, soit sous forme de base de données en sélectionnant **From Database**.
4. Pour un fichier, renseignez le chemin d'accès et le séparateur de champ du fichier contenant les paramètres de contexte dans les champs **From File** et **Field Separator**.
5. Pour une base de données, sélectionnez le mode adéquat, **Built-in** ou **Repository**, dans le champ **Property type** et renseignez les champs suivants.
6. Dans les champs **Table Name** et **Query Condition**, renseignez le nom de la table contenant les paramètres de contexte et la requête à utiliser.
7. 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.
8. 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 :

1. 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]**.
2. Dans l'arborescence de la boîte de dialogue, développez le nœud **Job Settings** et cliquez sur **Use Project Settings** pour afficher l'utilisation des options **Implicit Context Load** et **Stats and Logs** dans les Jobs.



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

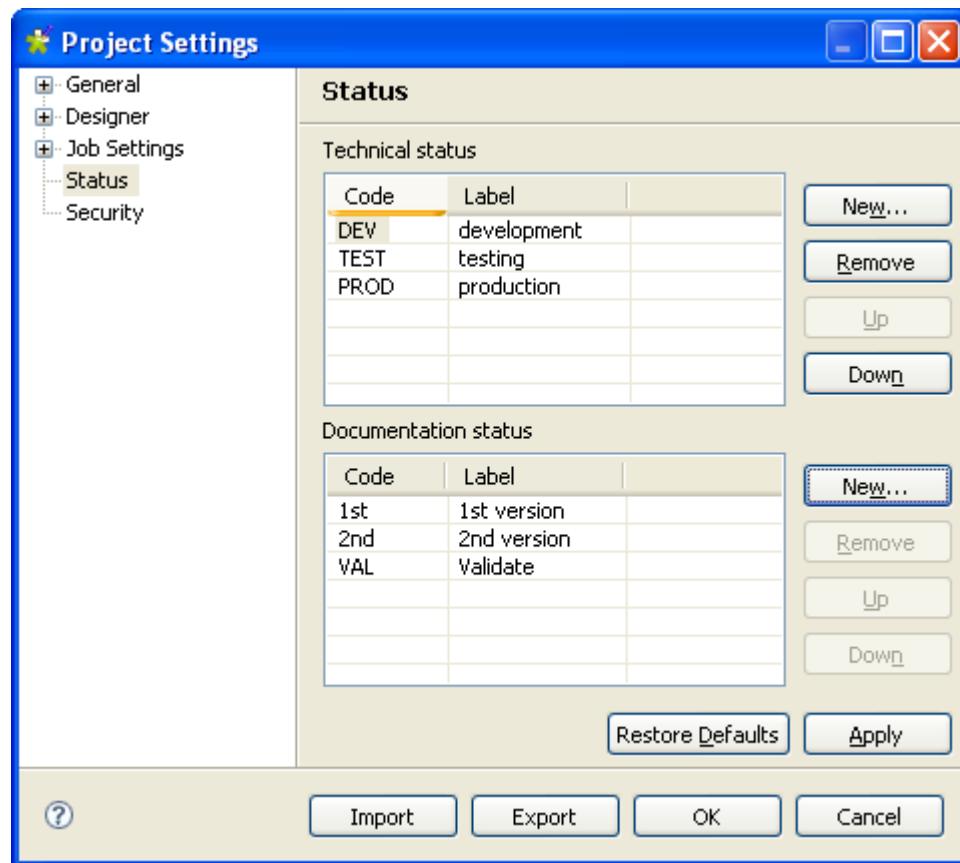
2.6.8. Définir les paramètres des Statuts

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

Pour cela :

1. 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]**.
2. Dans l'arborescence, cliquez sur le nœud **Status** pour paramétrier 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** sauvegarder.

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. 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**.

Pour masquer votre mot de passe :

1. 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]**.
2. Dans l'arborescence, cliquez sur le nœud **Security** pour ouvrir la vue correspondante.
3. Cochez la case **Hide passwords** pour masquer vos mots de passe.



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.

Property Type	<input type="button" value="Repository"/>	DB (MYSQL):demoMysql	<input type="button" value="..."/>
<input type="checkbox"/> Use an existing connection			
Host	"localhost"	* Port	"3306"
Database	"test"	*	
Username	"root"	* Password	****

4. 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 for Data Integration a été spécialement conçu pour permettre une représentation simple et claire des besoins métiers de l'entreprise, ainsi que leur interaction. *Talend Open Studio for Data Integration* 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 for Data Integration*. Pour plus d'informations, consultez [Annexe A, Interface de Talend Open Studio for Data Integration](#).

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 for Data Integration* 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.

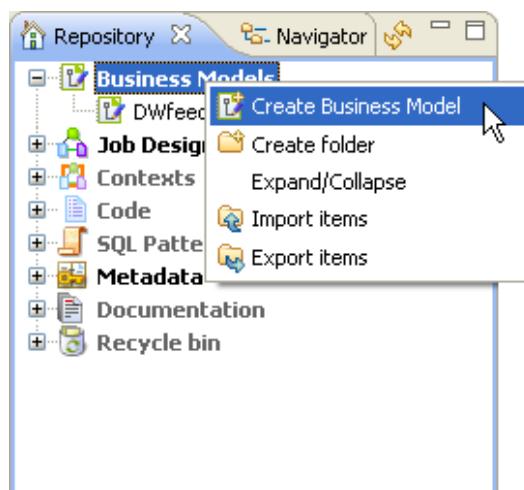
A partir de *Talend Open Studio for Data Integration*, vous disposez de nombreux outils vous permettant de :

- 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. Crédation ou ouverture d'un Business Model

Lancez *Talend Open Studio for Data Integration* suivant la procédure détaillée dans la section [Section 2.2, « Lancement de Talend Open Studio for Data Integration »](#).

Dans la fenêtre principale de *Talend Open Studio for Data Integration*, 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. Créeer un Business Model

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

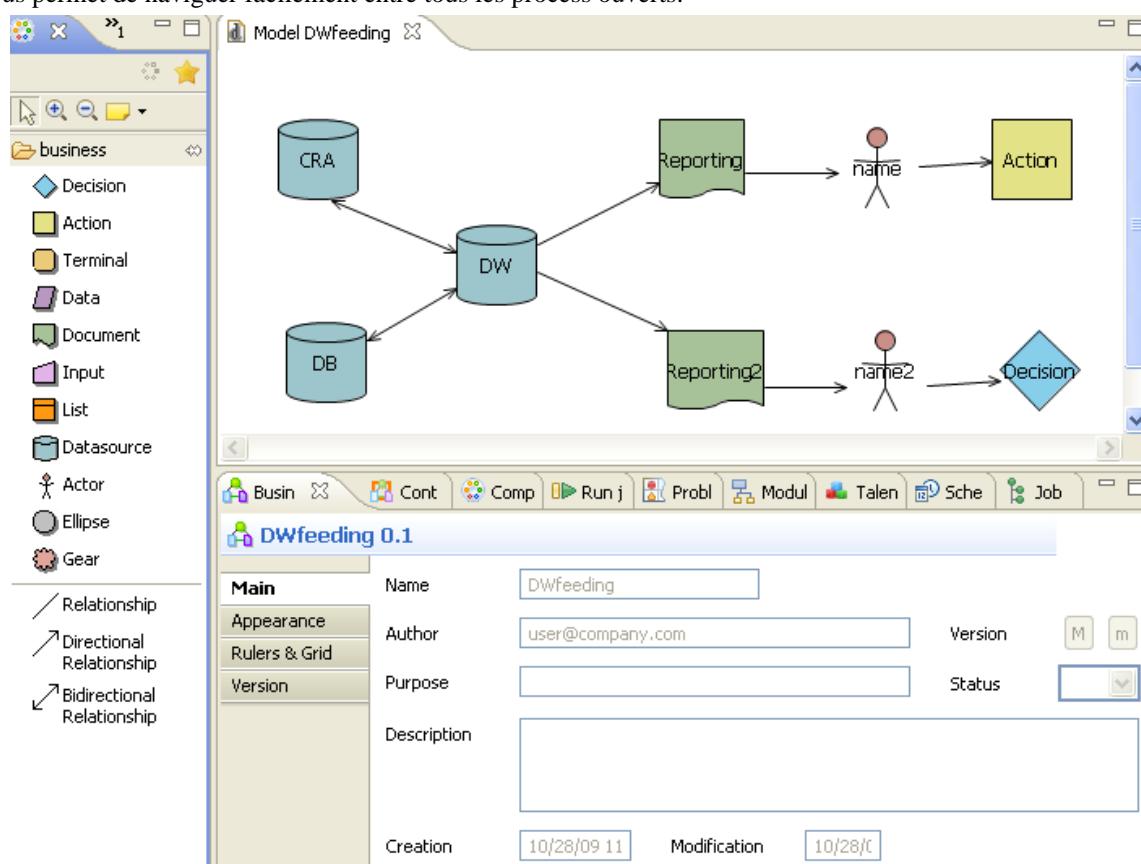
L'assistant de création vous guide dans les étapes de création d'un Business Model.

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 nœud **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.



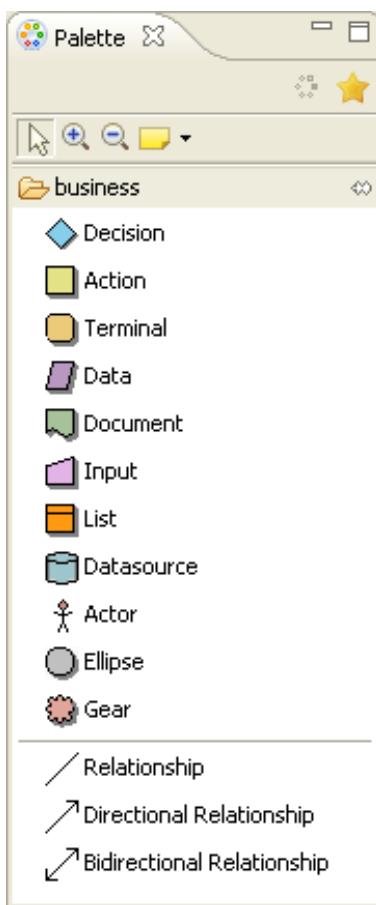
Le **Modeler** est composé des vues suivantes :

- L'éditeur graphique (ou espace de modélisation) de *Talend Open Studio for Data Integration*
- 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 :

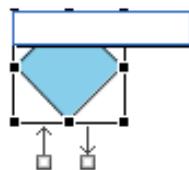


Par exemple, si votre Business Model inclut une étape de décision, sélectionnez la forme losange dans la **Palette**.

 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 : [Section 3.3.2, « Connexions »](#).

Les formes disponibles comprennent :

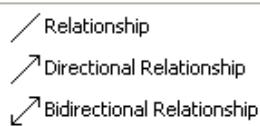
Libellé	Description
Decision	Représente une condition if dans le flux de données. Permet de prendre des mesures contextuelles.
Action	Tout type d'action, notamment, transformation, traduction ou format par exemple.
Terminal	Tout type de terminal de sortie.
Data	Tout type de données numériques ou alphanumériques.
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.

Libellé	Description
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.

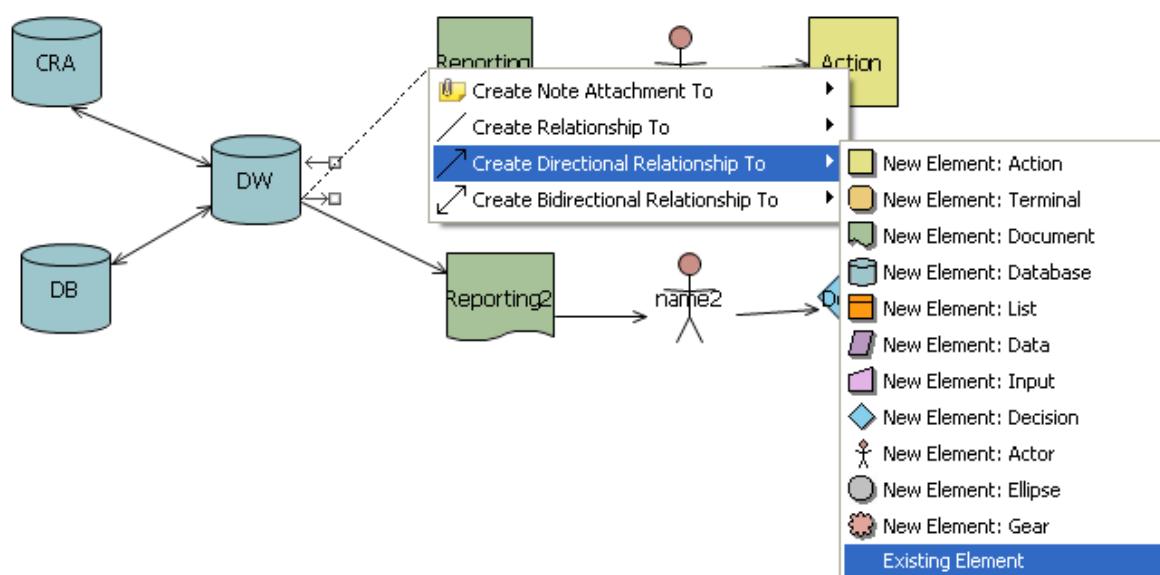
Plusieurs solutions sont offertes :



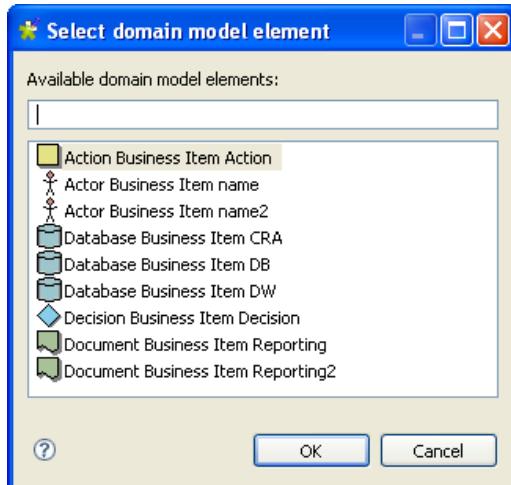
Sélectionnez l'outil **Relationship** de la **Palette**. Puis dans l'éditeur graphique, tirez un lien d'une forme vers une autre pour dessiner la connexion.

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 :

1. Placez simplement votre souris au dessus de la forme source pour faire apparaître les doubles flèches.
2. Sélectionnez la flèche correspondant à la connexion dont vous avez besoin.
3. Faites glisser l'une des flèches vers une zone vide de l'espace de modélisation et relâchez la souris pour afficher le menu contextuel.
4. Sélectionnez la connexion appropriée dans la liste : **Create Relationship To**, **Create Directional Relationship To** ou **Create Bidirectional Relationship To**.
5. Puis sélectionnez l'élément cible de la connexion parmi les éléments proposés.



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



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

La nature de cette connexion peut être décrite à l'aide d'éléments du référentiel de métadonnées (Repository) et sa mise en forme peut être paramétrée dans l'onglet **Business Model**, consultez la section [Section 3.3.4, « Propriétés »](#).

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 : [Section 3.3.3, « Commenter et réarranger un Business Model »](#)

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.

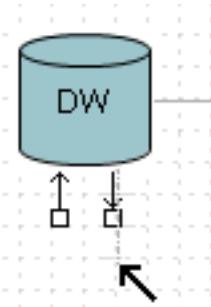
3.3.3.1. Ajouter une note ou du texte libre

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.

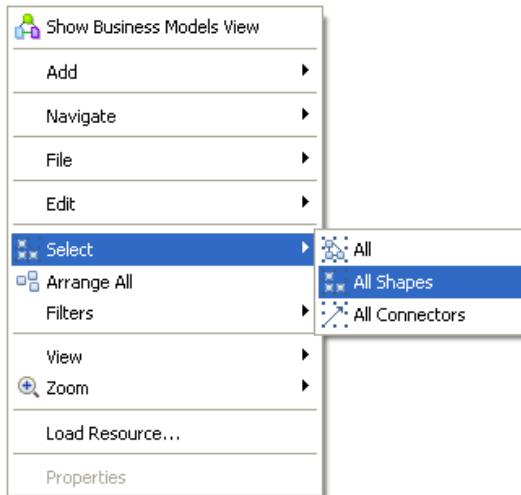


Pour ajouter une note attachée à une forme, sélectionnez **Note attachment** dans la **Palette** puis sélectionnez dans l'espace de modélisation la forme que vous souhaitez annoter. Lorsque la flèche noire apparaît, faites-la glisser vers une zone vide de l'éditeur. Relâchez pour afficher le menu contextuel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

3.3.3.2. 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 [Section 4.2.8.3, « Afficher les onglets/vues de configuration des Jobs »](#).

3.3.4.1. 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 :

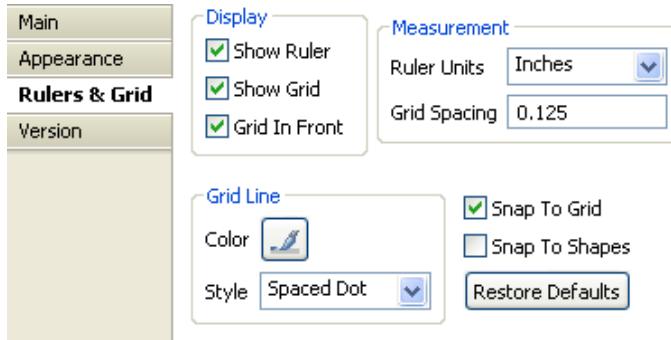
- changer la couleur de la forme
- colorer les bordures
- ajouter du texte
- ajouter un effet dégradé
- ajouter un effet ombré

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.

3.3.4.2. 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.



Propriétés

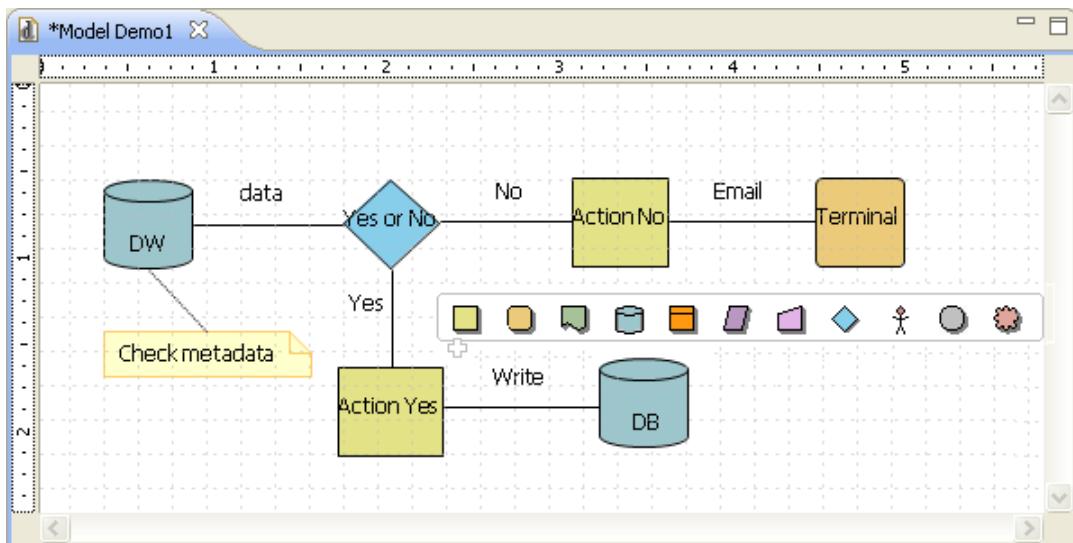
Dans la zone **Display**, cochez la case **Show Ruler** pour afficher la Règle (**Ruler**), ou **Show Grid** pour afficher la Grille (**Grid**) ou les deux. **Grid in front** place la grille devant toutes les autres formes du process.

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**.



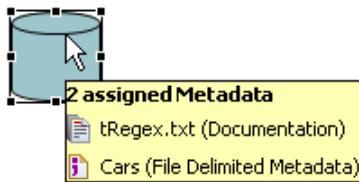
3.3.4.3. Onglet Assignment

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.

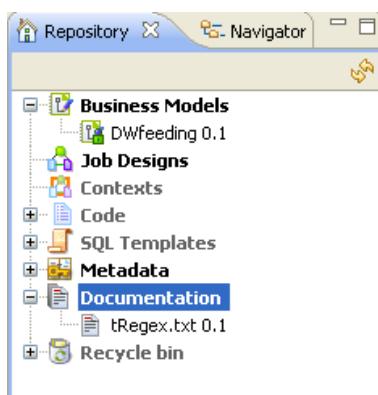


Vous pouvez modifier certaines informations ou lier un commentaire. En outre, si vous mettez à jour des données dans le référentiel, les informations d'affectation sont automatiquement mises à jour.

Pour plus d'informations concernant les affectations, voir [Section 3.4, « Affectation d'éléments du Repository à un Business Model ».](#)

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.

Elément	Description
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 nœud Code .

Pour plus d'informations concernant les éléments du **Repository**, consultez le [Chapitre 7, Gestion des métadonnées](#)

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.

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

2. 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 :

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

Sélectionnez	Pour...
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 for Data Integration* 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 for Data Integration* 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 for Data Integration*.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez [Annexe A, Interface de Talend Open Studio for Data Integration](#).

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 for Data Integration*, 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,
- paramétrier 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 créés dans Talend Open Studio for Data Integration, vous devez installer une JVM Oracle 1.6 ou une version ultérieure (les JVM IBM ne sont pas supportées). Vous pouvez la télécharger sur <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

4.2. Utilisation d'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 la [Section 4.3.1, « Types de connexions »](#) et la [Section 4.2.6, « Définir les propriétés d'un composant »](#).

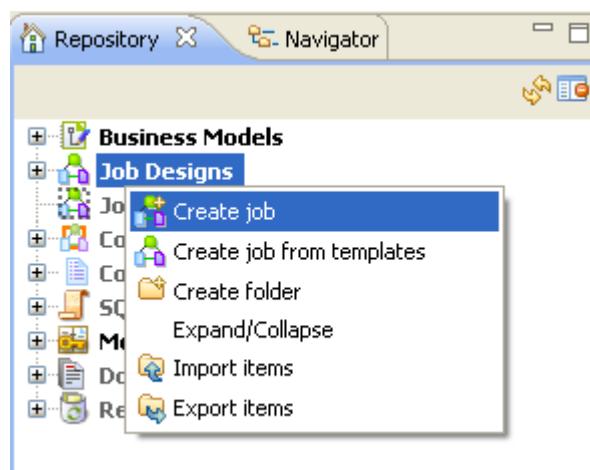
4.2.1. Créez un Job

Talend Open Studio for Data Integration 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.

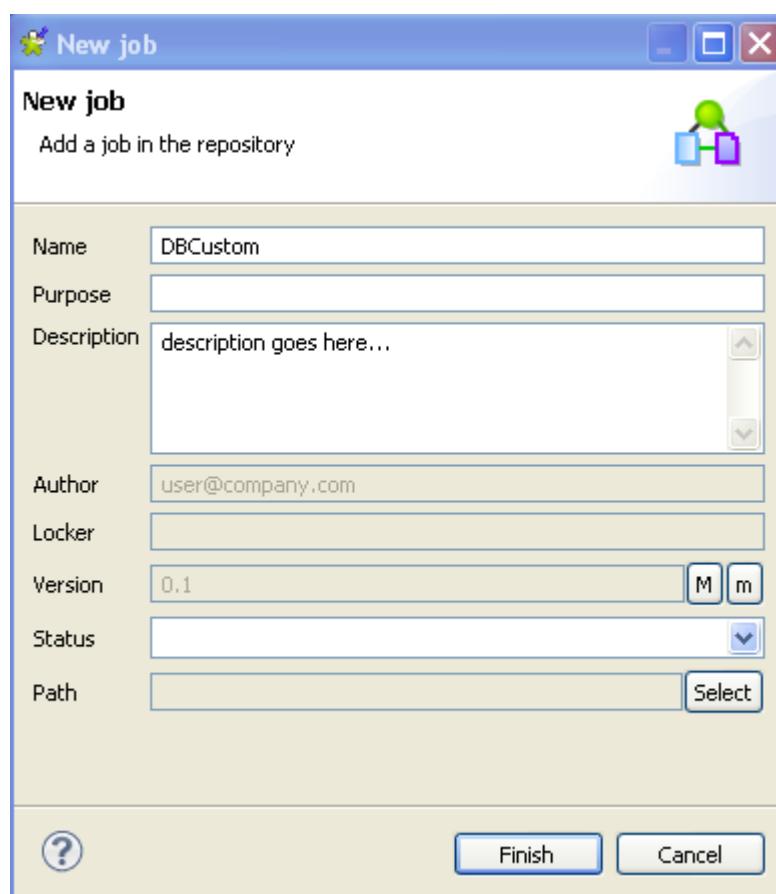
Vous pouvez également créer différents dossiers pour mieux classifier ces Jobs.

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

- Ouvrez *Talend Open Studio for Data Integration* en suivant la procédure détaillée dans la [Section 2.2, « Lancement de Talend Open Studio for Data Integration »](#).
- Dans la vue **Repository**, cliquez-droit sur le nœud **Job Designs**, et sélectionnez **Create job** dans le menu contextuel.



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



- Renseignez les propriétés du Job comme suit :

Champ	Description
Name	Saisissez un nom pour le nouveau Job. Un message s'affiche si vous saisissez des caractères interdits.

Champ	Description
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 la Section 2.6.2, « Gérer les versions » .
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 la [Section 4.2.2, « Déposer des composants dans l'espace de modélisation graphique »](#) et la [Section 4.3.1, « Types de connexions »](#).
- Définissez les propriétés de chaque composant utilisé dans le Job. Pour plus d'informations, consultez la [Section 4.2.6, « Définir les propriétés d'un composant »](#).
- Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter. Pour plus d'informations, consultez la [Section 4.2.7, « Exécuter un Job »](#).

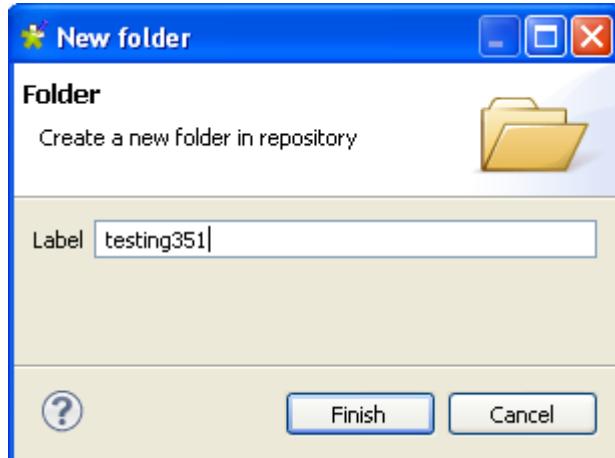
Le Job que vous avez créé s'affiche sous le nœud **Job Designs**, dans la vue **Repository**.

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

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 nœud **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 la [Annexe B, De la théorie à la pratique, exemple de Job](#).

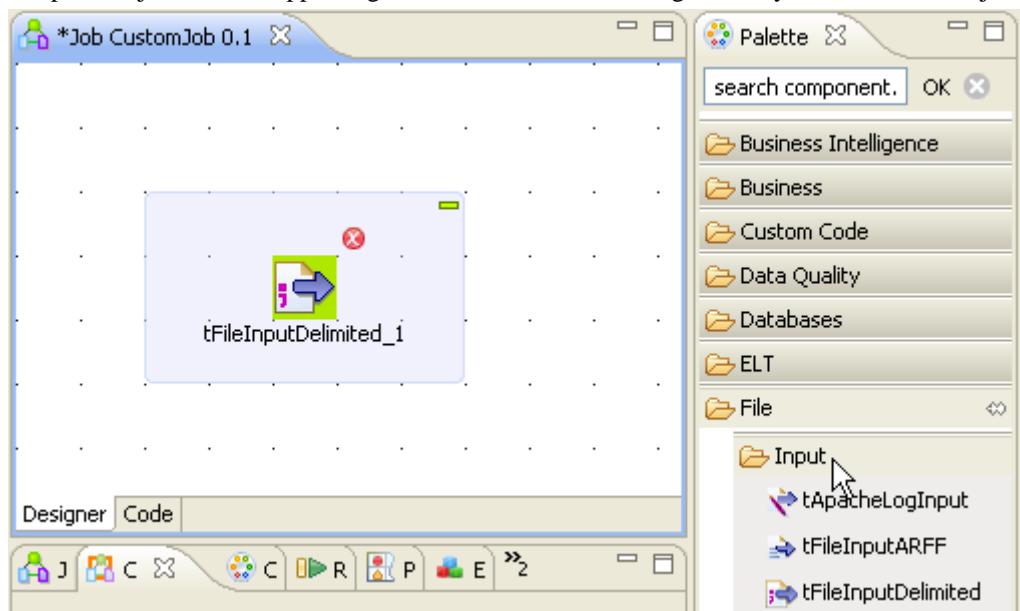
4.2.2. Déposer des composants dans l'espace de modélisation graphique

4.2.2.1. 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 la section intitulée « [Afficher, cacher et déplacer la Palette](#) ».
-  Vous pouvez également déposer une Note de la même façon que vous déposez des composants. Pour plus d'informations, consultez la [Section 4.6.4, « Ajouter une Note au Job »](#).

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 la [Section 4.3.1, « Types de connexions »](#).

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 la [Section 4.6.6.1, « Personnaliser les sous-jobs »](#).

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

 *Vous devrez utiliser le code Java pour votre projet.*

Voir également :

- [Section 4.3.1, « Types de connexions ».](#)
- [Section 4.6.3, « Gérer les icônes d'avertissement/d'erreur sur les composants ».](#)
- [Section 4.2.6, « Définir les propriétés d'un composant ».](#)

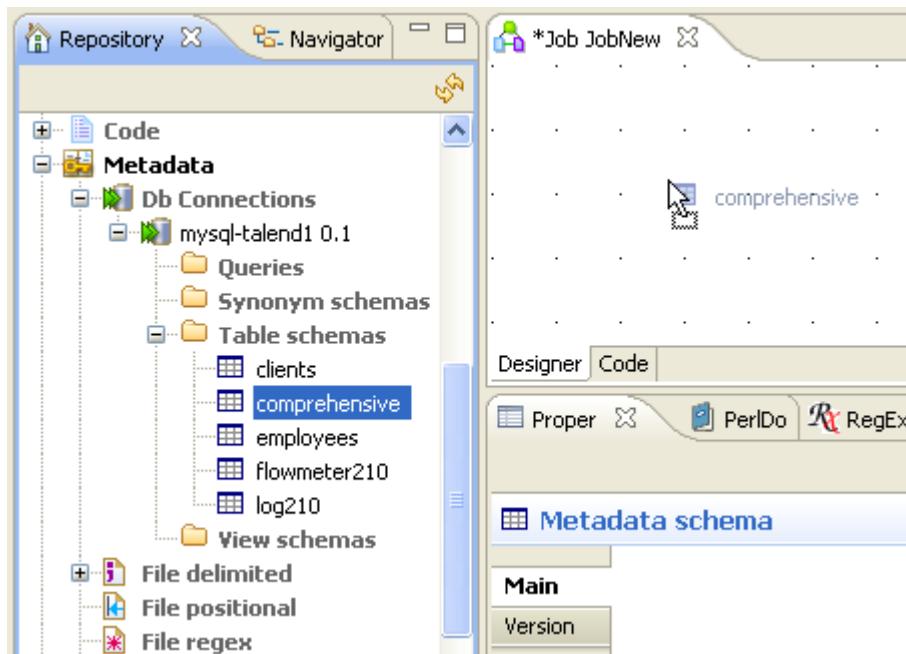
4.2.2.2. Déposer des composants à partir du répertoire Metadata

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 nœud **Metadata**. Différents dossiers sous ce nœud regroupent les connexions créées, qu'elles soient des connexions aux bases de données, aux fichiers ou aux systèmes.

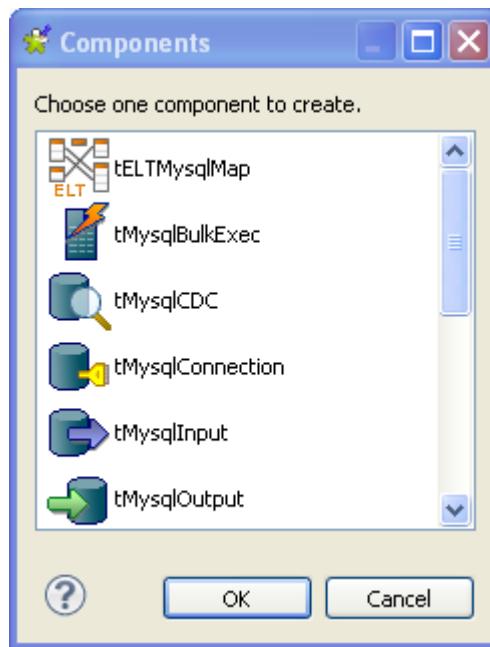
Différents assistants vous permettront de centraliser les connexions et les métadonnées des schémas, dans la vue Repository. Pour plus d'informations concernant les assistants de gestion des métadonnées, consultez la [Section 4.4.1, « Centraliser les métadonnées »](#).

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

1. Développez le nœud **Metadata** du **Repository** ainsi que le(s) dossier(s) où sont stockées les données que vous voulez utiliser.
2. Déposez la connexion/le schéma choisi(e) dans l'éditeur graphique.



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



3. Sélectionnez le composant que vous souhaitez utiliser et cliquez sur **OK**. Le composant sélectionné s'affiche dans l'espace de modélisation graphique.

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

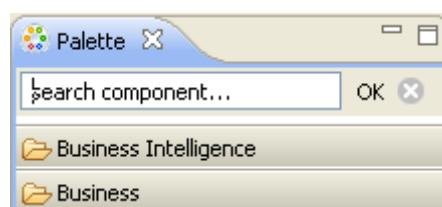
- **Input** : Maintenez la touche **Alt** enfoncée pour déposer l'élément sélectionné dans l'éditeur graphique et ainsi l'ajouter à votre Job Design.
- **Output** : Maintenez la touche **Ctrl** enfoncée pour déposer le composant dans l'éditeur et ainsi l'ajouter à votre Job Design.

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

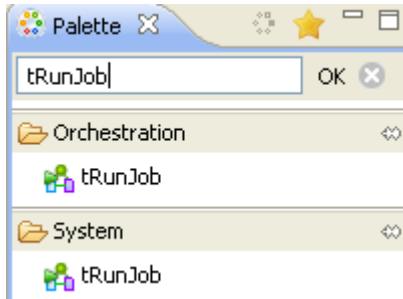
Si vous ne voulez pas parcourir les familles de composants dans la **Palette** pour trouver les composants que vous souhaitez utiliser dans votre Job, vous pouvez chercher le composant directement dans le champ de recherche en haut de la **Palette**.



Pour chercher un composant, procédez comme suit :

1. Cliquez sur le bouton  pour effacer tout texte du champ de recherche.
2. 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 la [Section 4.5.6, « Utiliser les composants tPrejob et tPostjob »](#).

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.

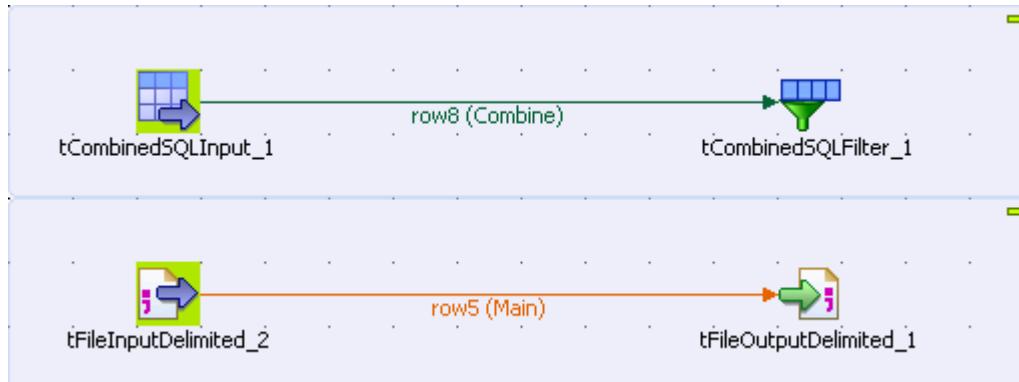
Pour plus d'informations concernant les différents types de connexions disponibles, consultez la [Section 4.3, « Utilisation des connexions »](#).

4.2.5. Cliquer et déposer des composants au milieu d'une connexion de type Row

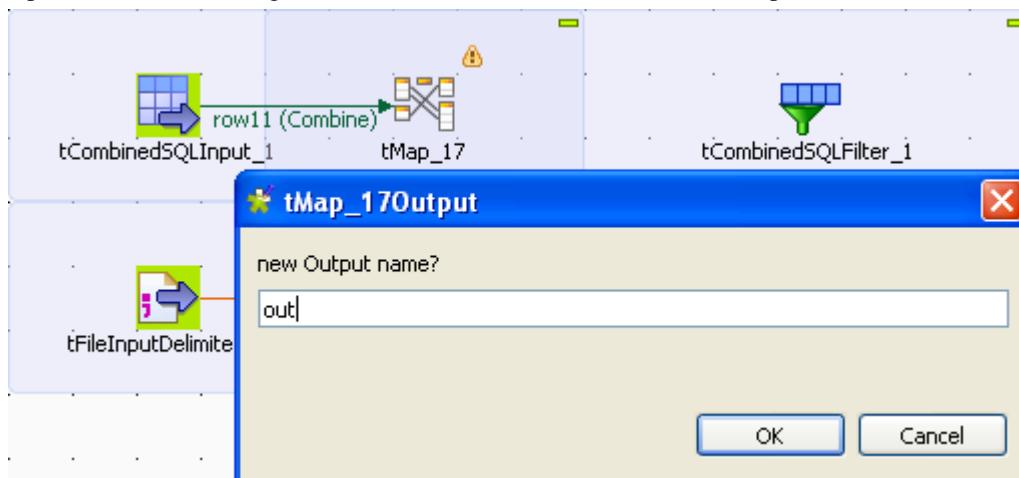
Lorsque vous créez un Job, *Talend Open Studio for Data Integration* vous permet d'insérer un composant au milieu d'une connexion de type **Row > Main**, **Row > Reject** ou de type **Row > Combine**.

Pour ce faire, procédez comme suit :

1. A partir de la **Palette**, cliquez et déposez deux composants Combine et deux composants File (un d'entrée et un de sortie) dans l'espace de modélisation graphique.
2. Connectez les paires de composants en utilisant soit une connexion de type **Row > Main** (ou **Row > Reject**), soit une connexion de type **Row > Reject**.



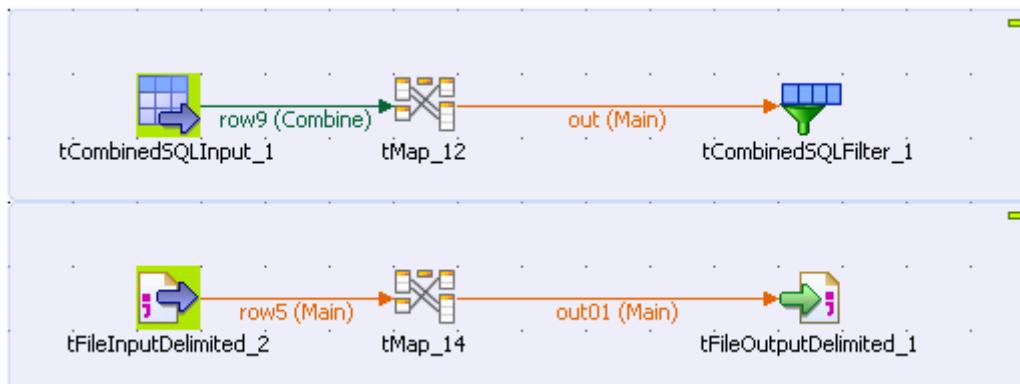
3. Cliquez et déposez le composant que vous souhaitez insérer au milieu de la connexion. La connexion s'affiche en gras puis une boîte de dialogue s'ouvre, vous demandant de saisir un nom pour le lien de sortie.



4. Saisissez un nom puis cliquez sur **OK** pour fermer la boîte de dialogue.

Il est possible qu'un message vous propose de récupérer le schéma du composant cible. Cliquez sur **OK** pour accepter ou sur **No** pour refuser.

Le composant est inséré au milieu de la connexion, désormais divisée en deux.



4.2.6. Définir les propriétés d'un composant

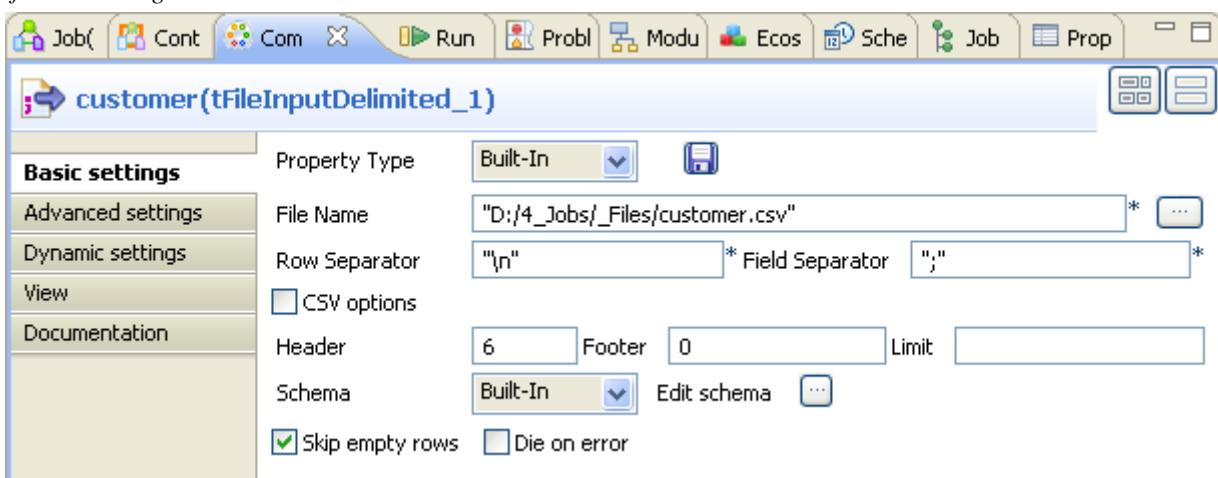
Les propriétés de chacun des composants formant un Job ou un sous-job permettent de paramétriser l'exécution technique du Job actif.

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**.

4.2.6.1. 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 for Data Integration*.



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 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.

Sélectionnez **Repository** dans le champ **Property type** et désignez les métadonnées comportant les paramètres appropriés. Voir également : [Section 4.4.1, « Centraliser les métadonnées »](#).

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

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

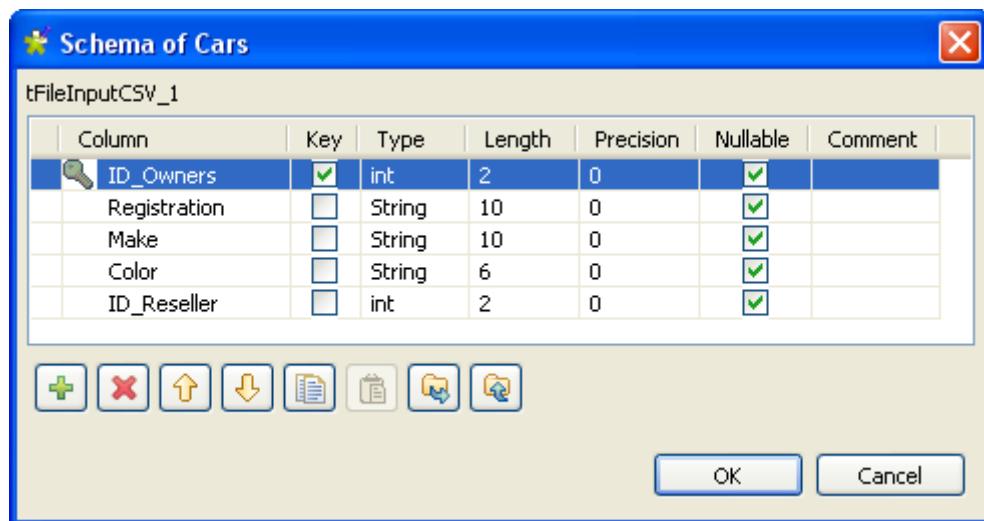
1. Cliquez sur l'icône de disquette. L'assistant de création de métadonnée correspondant à votre composant s'ouvre.
2. Suivez les étapes de l'assistant. Pour plus d'informations sur la création de métadonnées, consultez le [Chapitre 7, Gestion des métadonnées](#).
3. La métadonnée apparaît alors sous le nœud **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.

Paramétriser 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.

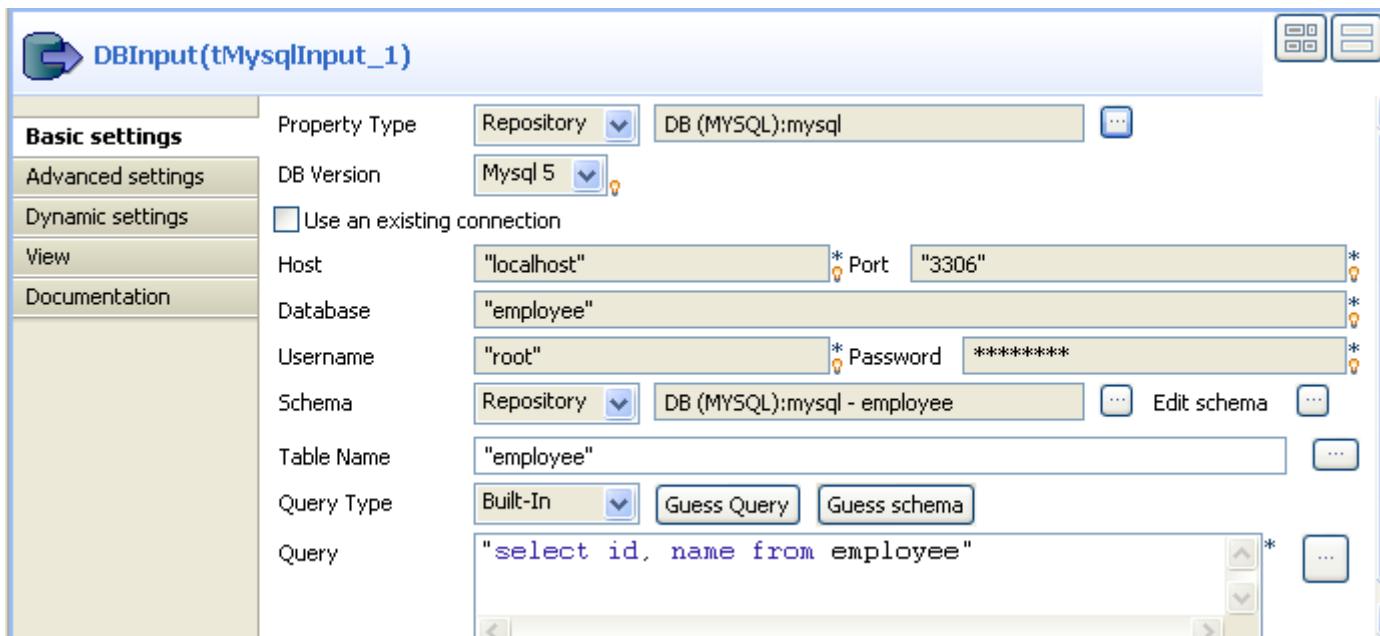


Des informations supplémentaires sont nécessaires. Pour plus d'informations sur les formats de date par exemple, visitez le site : <http://docs.oracle.com/javase/6/docs/api/index.html>

Paramétriser 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 nœud **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.



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 nœud correspondant et sélectionnez l'option correspondante (**Edit connection** ou **Edit file**) dans le menu contextuel.

Voir également : [Section 4.4.1, « Centraliser les métadonnées ».](#)

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**.

1. Placez le curseur de votre souris dans n'importe quel champ de l'onglet **Component**.
2. Appuyez sur **Ctrl+Barre d'espace** pour accéder à la liste des variables.
3. 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"))

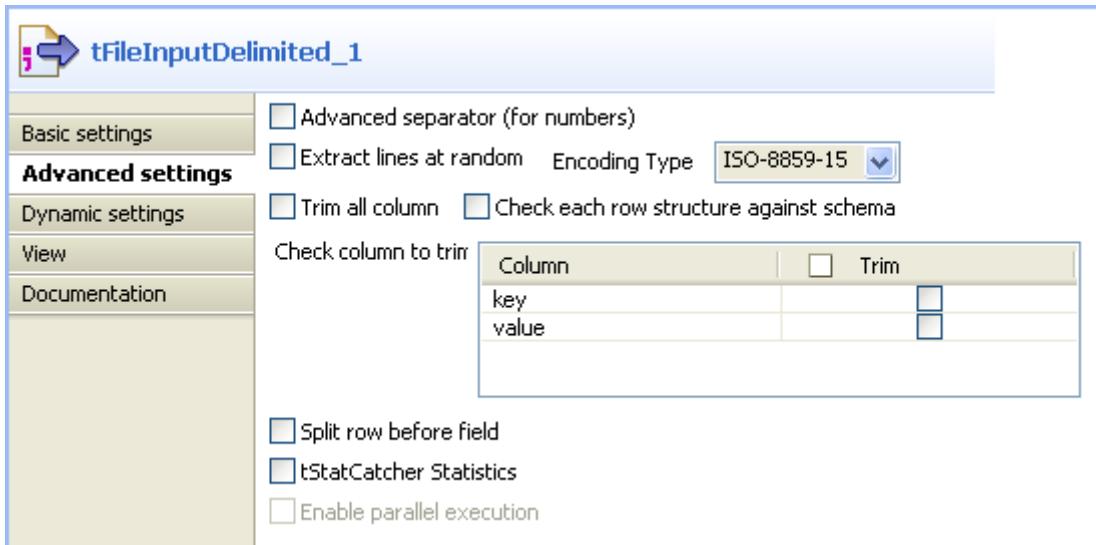
tFileInputDelimited_2.ERROR_MESSAGE
 tFileInputDelimited_2.NB_LINE
tMap_1.ERROR_MESSAGE
 tFileOutputDelimited_1.ERROR_MESSAGE
 tFileOutputDelimited_1.NB_LINE
 tFileOutputDelimited_2.ERROR_MESSAGE
 tFileOutputDelimited_2.NB_LINE
 tFileInputDelimited_3.ERROR_MESSAGE
 tFileInputDelimited_3.NB_LINE
 tFlowMeter_1.ERROR_MESSAGE
 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 : [Section 4.4.2, « Centraliser les Contextes et les Variables ».](#)

4.2.6.2. 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 [Section 4.6.7.1, « Automatiser l'utilisation de statistiques & logs ».](#)

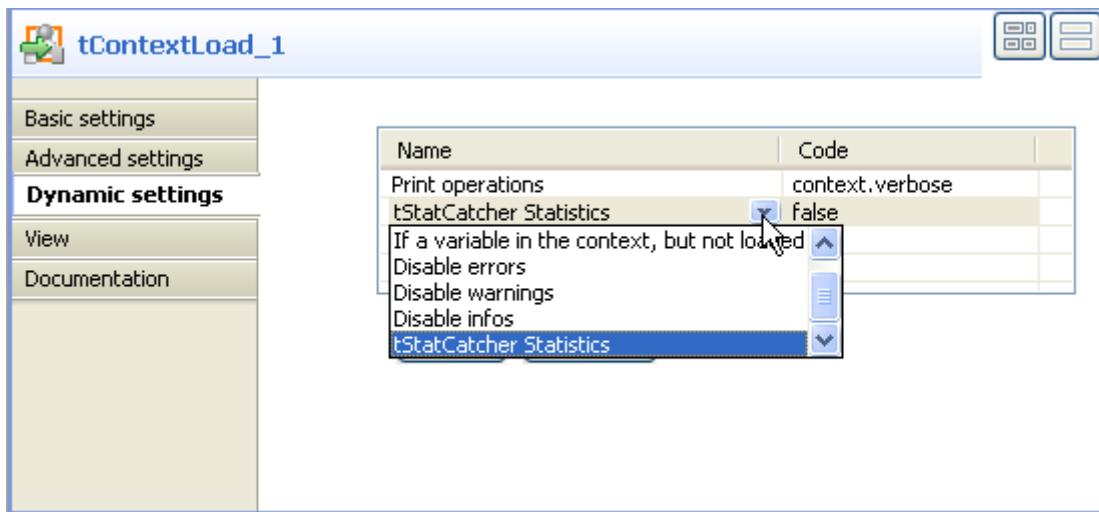
4.2.6.3. Onglet Dynamic settings

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.

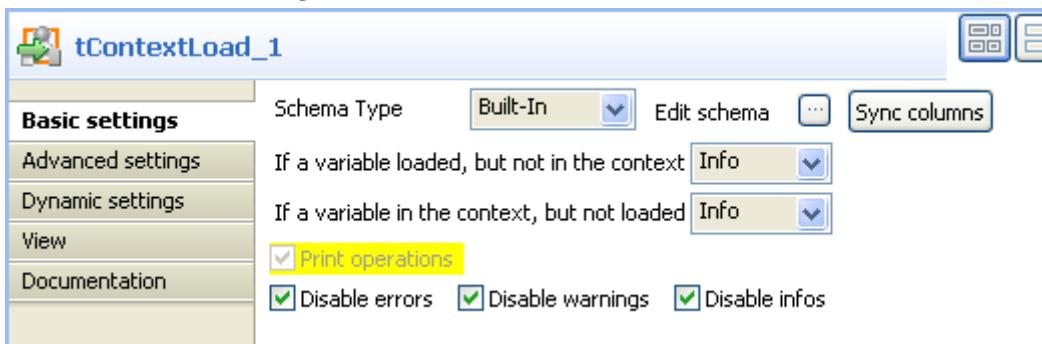


Pour personnaliser ces types de paramètres, notamment les variables de contexte, suivez les étapes qui suivent :

1. Sélectionnez le composant dont les onglets **Basic** et **Advanced settings** contiennent les paramètres que vous souhaitez définir comme variable.
2. Cliquez sur l'onglet **Dynamic settings**
3. Cliquez sur le bouton **[+]** pour créer une nouvelle ligne de paramètres dans le tableau.
4. Cliquez dans la cellule **Name** du paramètre pour afficher la liste des paramètres disponibles. Par exemple : *Print operations*.
5. 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*.

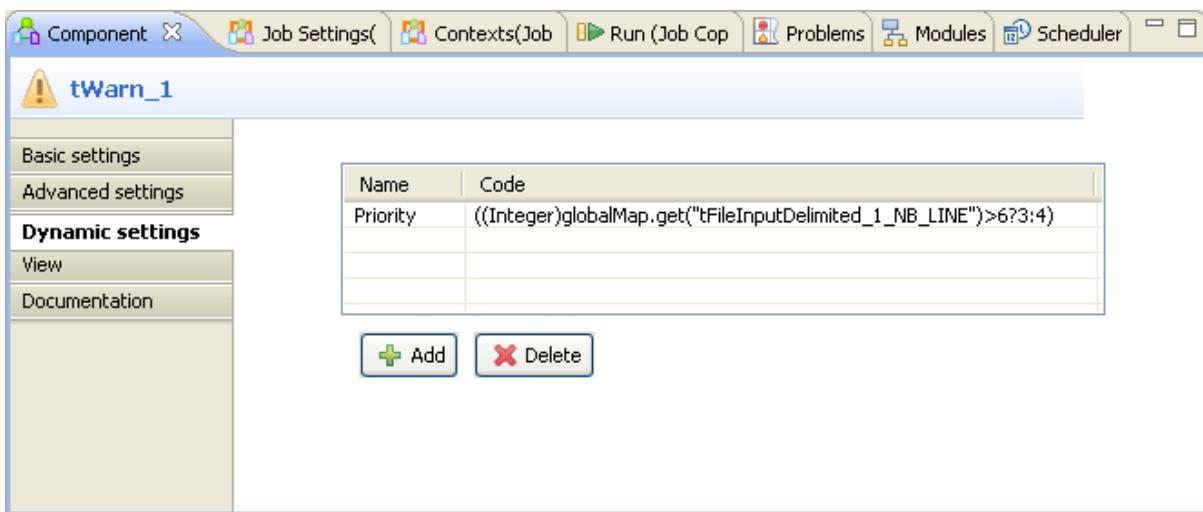
Dans la cellule **Code**, vous pouvez entrer une variable de contexte mais aussi un bout de code Java.

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 [Section 4.4.2.2, « Définir les variables à partir de la vue Contexts »](#).

Vous pouvez aussi utiliser une variable globale ou des bouts de code Java 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.

4.2.6.4. 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.

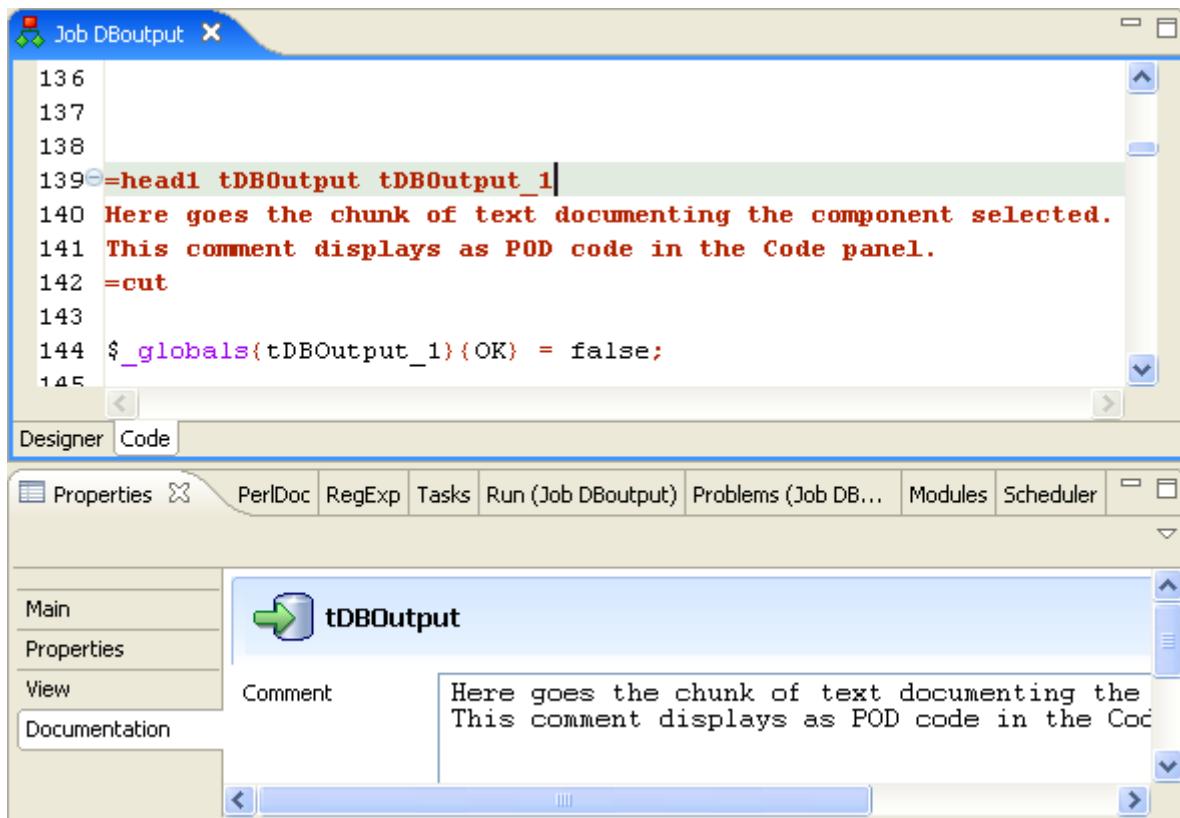
Vous pouvez personnaliser les textes des **Label** et **Hint** à l'aide des balises HTML suivantes :

- 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**.

4.2.6.5. 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.

4.2.7. Exécuter un Job

Vous disposez de plusieurs manières pour exécuter votre Job. Le choix du type d'exécution dépend de l'objectif ainsi que de votre niveau utilisateur.

Si vous avez un niveau utilisateur avancé et souhaitez exécuter votre projet pas à pas, consultez la section [Section 4.2.7.2, « Exécuter un Job en mode Java debug »](#).

Si vous n'avez pas de connaissances avancées en Java et souhaitez exécuter votre Job en mode normal, consultez [Section 4.2.7.1, « Exécuter en mode normal »](#).

4.2.7.1. Exécuter en mode normal

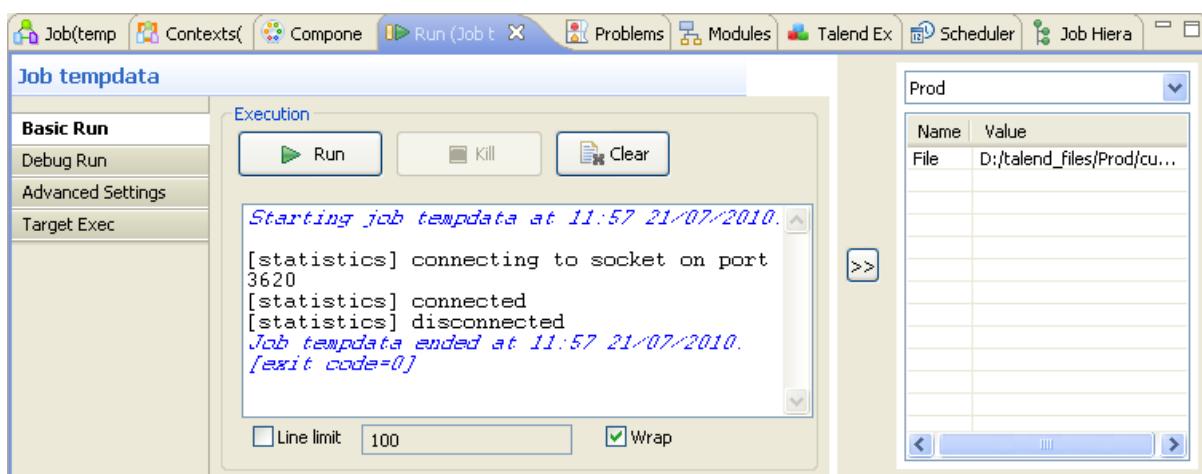
- 💡 Veillez à sauvegarder votre Job avant de l'exécuter, afin que toutes les propriétés puissent être prises en compte.

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

1. Cliquez sur la vue **Run** pour l'ouvrir.
2. Cliquez sur l'onglet **Run Job** pour accéder au mode normal d'exécution.
3. 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 : [Section 4.4.2.2, « Définir les variables à partir de la vue Contexts »](#).

1. Cliquez sur **Run** pour lancer l'exécution.
2. Sur le même panneau, la console de log affiche la progression de l'exécution. La console inclut les messages d'erreurs ainsi que les messages de début et de fin de processus. Elle affiche également le résultat du processus si le composant terminal le prévoit, notamment le **tLogRow**.
3. Pour paramétriser le nombre de lignes à afficher dans la console lors de l'avancement de l'exécution, cochez la case **Line limit** et saisissez une valeur dans le champ.
4. Cochez la case **Wrap** pour activer les retours automatiques à la ligne. Cette case est cochée par défaut. Lorsqu'elle est décochée, une barre de défilement horizontale apparaît, vous permettant de voir la fin des lignes.



Avant d'exécuter à nouveau un Job, vous pouvez vider le contenu de la vue de log, 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 for Data Integration 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.

4.2.7.2. 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 :

1. Cliquez sur l'onglet **Run** pour accéder à cette vue.

2. 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.

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 for Data Integration* ouvre l'éditeur en mode de débogage.

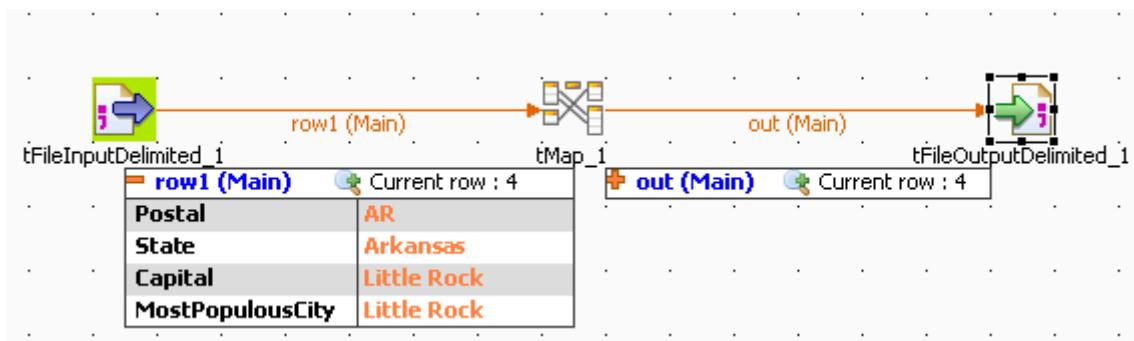
Vous pouvez exécuter le Job pas à pas et vérifier le comportement de chacun des composants ainsi que les valeurs des variables utilisées.

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

4.2.7.3. 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 for Data Integration*.

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



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

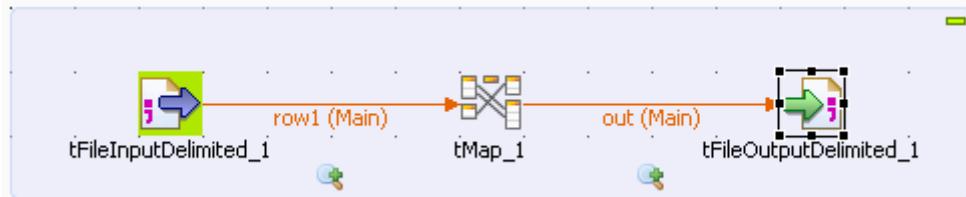
La fonction **Traces** affiche le contenu des lignes traitées dans un tableau.



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

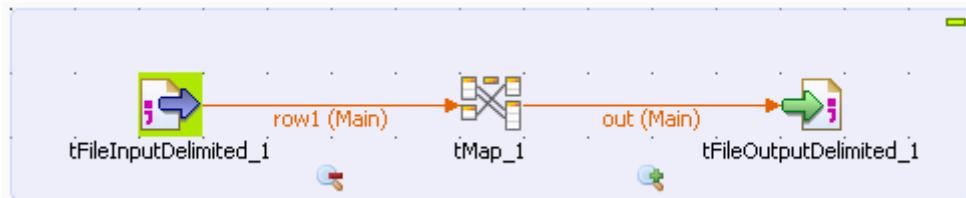
Vous pouvez activer ou désactiver le mode **Traces** ou décider quelles colonnes traitées afficher dans le tableau des traces qui s'affichera dans l'espace de modélisation graphique lors de l'exécution du Job.

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



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

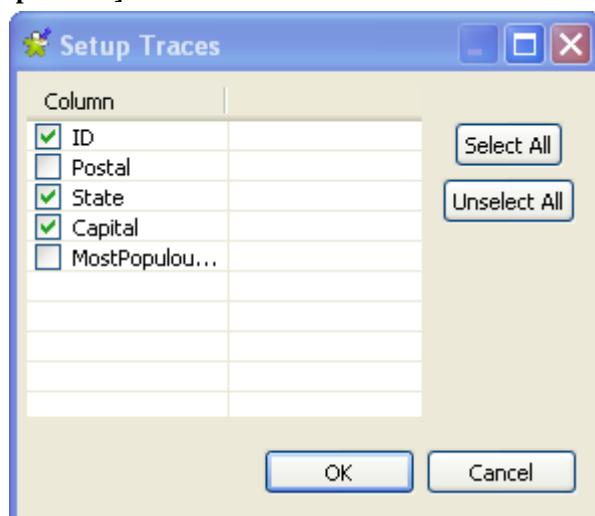
Pour désactiver le mode **Traces** d'un flux de votre Job :



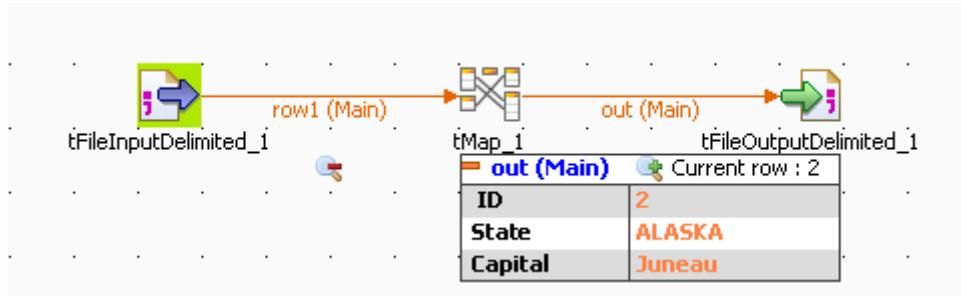
1. Cliquez-droit sur l'icône de **Traces** du flux concerné.
2. Sélectionnez **Disable Traces** dans le menu contextuel. Un signe moins rouge vient remplacer le signe plus vert sur l'icône pour indiquer que le mode **Traces** a été désactivé pour ce flux.

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

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



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



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

Cliquez sur le bouton **Clear** dans l'onglet **Debug Run** pour effacer les statistiques affichées.

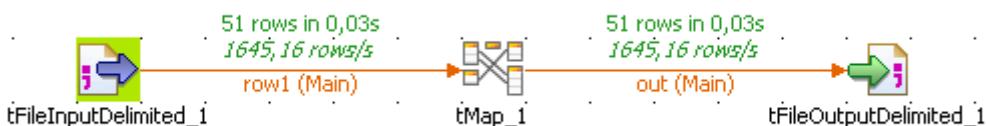
4.2.7.4. Configurer les paramètres d'exécution avancés

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 [la section intitulée « Afficher les Statistiques »](#).
- **Exec time**, cette fonction affiche le temps d'exécution dans la console, à la fin de l'exécution. Pour plus d'informations, consultez [la section intitulée « Afficher la durée d'exécution et d'autres options »](#).
- **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.

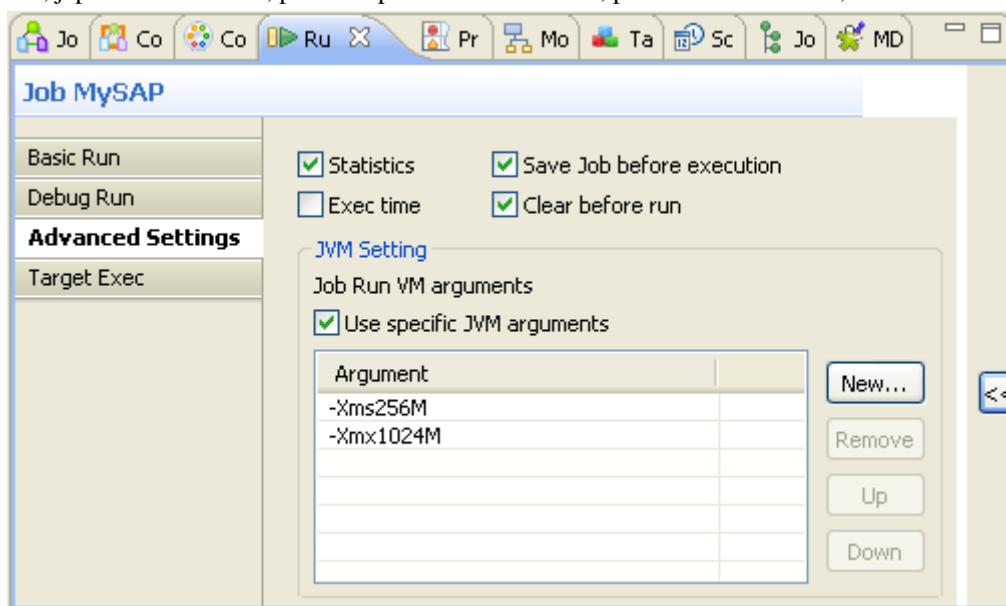
De cette manière vous pouvez vérifier les résultats ou tester votre Job avant de l'envoyer en production.

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 for Data Integration 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 :



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

Cet argument peut être appliqué à toutes vos exécutions de Jobs dans *Talend Open Studio for Data Integration*. Pour plus d'informations concernant l'application d'un argument JVM à toutes les exécutions de Jobs, consultez la [Section 2.5.5, « Préférences d'exécution et de débogage »](#).

4.2.8. Personnaliser l'espace de travail

Lorsque vous utilisez *Talend Open Studio for Data Integration* 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.

4.2.8.1. 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.

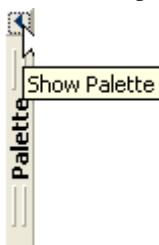
Pour la configuration d'un composant spécifique, consultez le **Guide de Référence des Composants Talend Open Studio**.

Talend Open Studio for Data Integration 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.



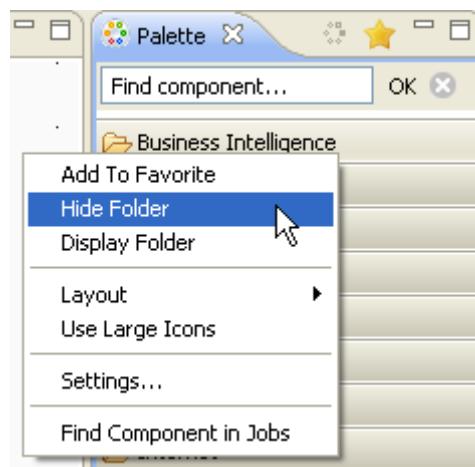
Si vous souhaitez que la **Palette** soit visible en permanence, cliquez sur la flèche de gauche dans le coin supérieur droit de l'éditeur graphique.

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

Vous pouvez aussi détacher la **Palette** de l'espace de modélisation dans *Talend Open Studio for Data Integration*. 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 la section intitulée « [Configurer la Palette favorite](#) ».

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 la [Section 2.6.1, « Personnaliser la Palette »](#).

Pour plus d'informations concernant l'ajout de composants à la **Palette**, à partir de **Talend Exchange** ou développés par vous-même, consultez la [Section 4.5.3, « Télécharger/charger des composants de la communauté »](#) et/ou la [Section 2.5.2, « Composants externes ou utilisateur »](#).

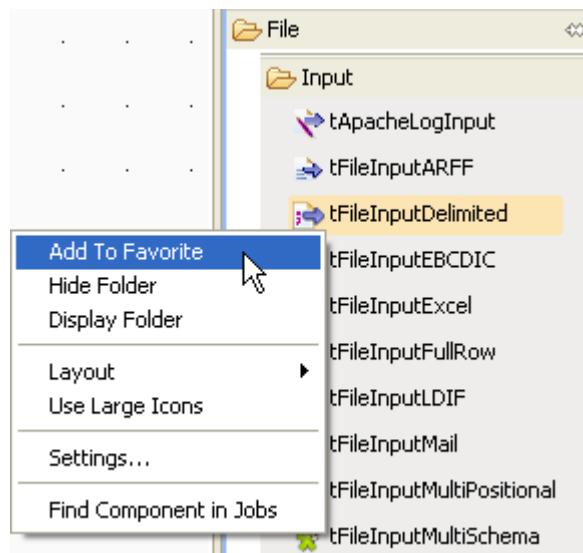
Configurer la Palette favorite

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

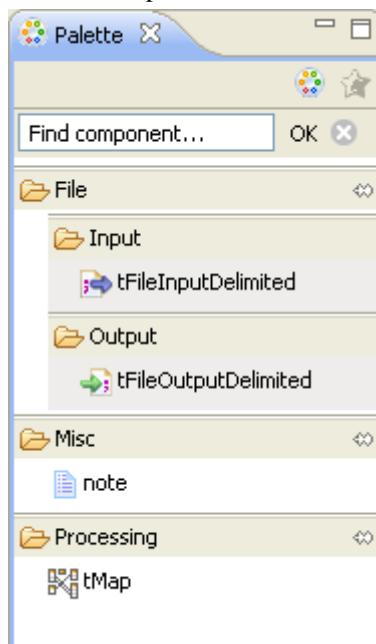
Vous pouvez ajouter et retirer des composants à votre **Palette** favorite de *Talend Open Studio for Data Integration*, afin d'accéder plus rapidement aux composants que vous utilisez le plus souvent.

Pour ce faire :

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



2. Répétez cette action pour tous les composants que vous souhaitez ajouter à votre **Palette** favorite, puis cliquez sur le bouton **Favorite** 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

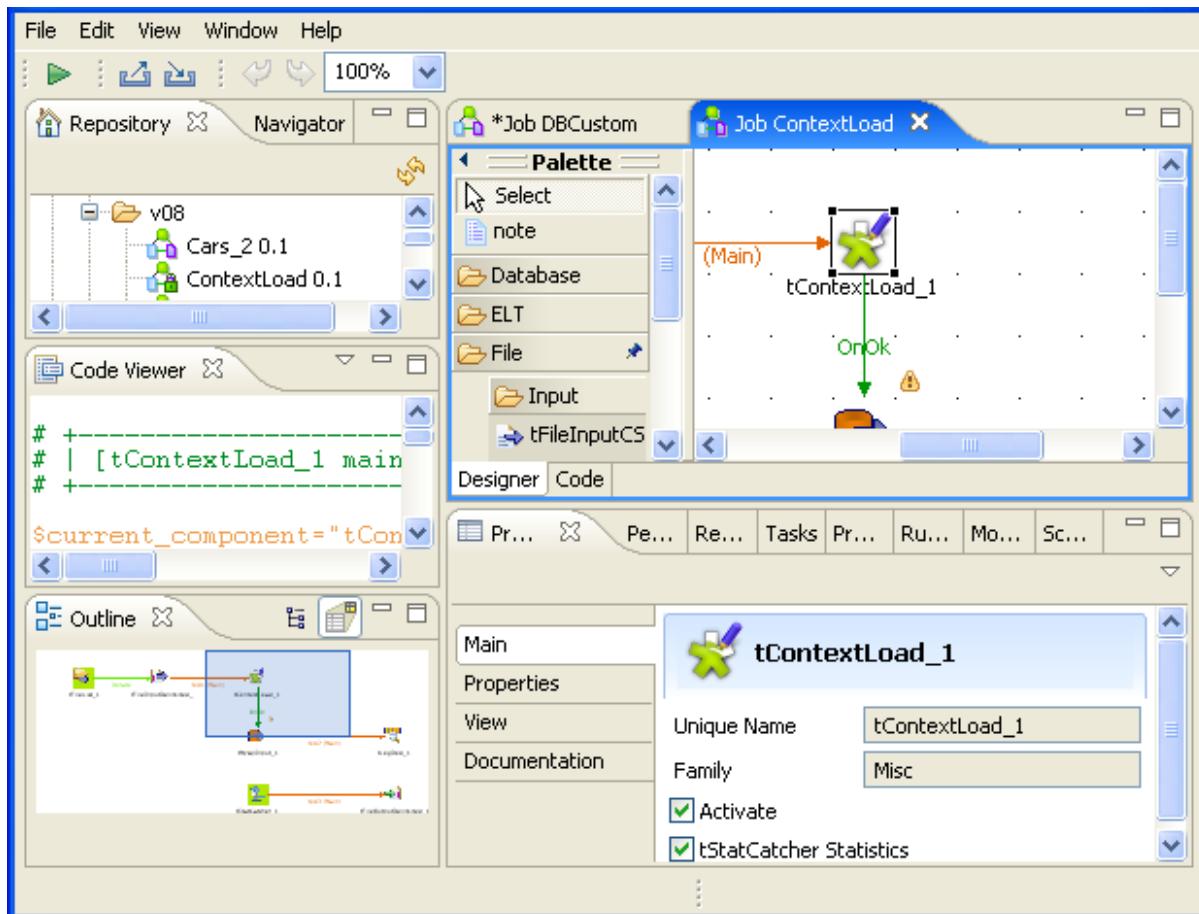
Talend Open Studio for Data Integration vous permet d'ajouter des composants externes à la **Palette** de votre Studio et des utiliser dans vos Jobs.

Pour plus d'informations concernant la création et le développement de composants utilisateurs, consultez notre section wiki [Component creation tutorial section](http://www.talendforge.org/wiki/doku.php?id=component_creation) [http://www.talendforge.org/wiki/doku.php?id=component_creation](en anglais).

Pour plus d'informations concernant le téléchargement de composants utilisateur dans votre Studio, consultez la [Section 2.5.2, « Composants externes ou utilisateur »](#).

4.2.8.2. Réorganiser les onglets d'un Job

Vous pouvez déplacer tous les onglets de *Talend Open Studio for Data Integration* 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.

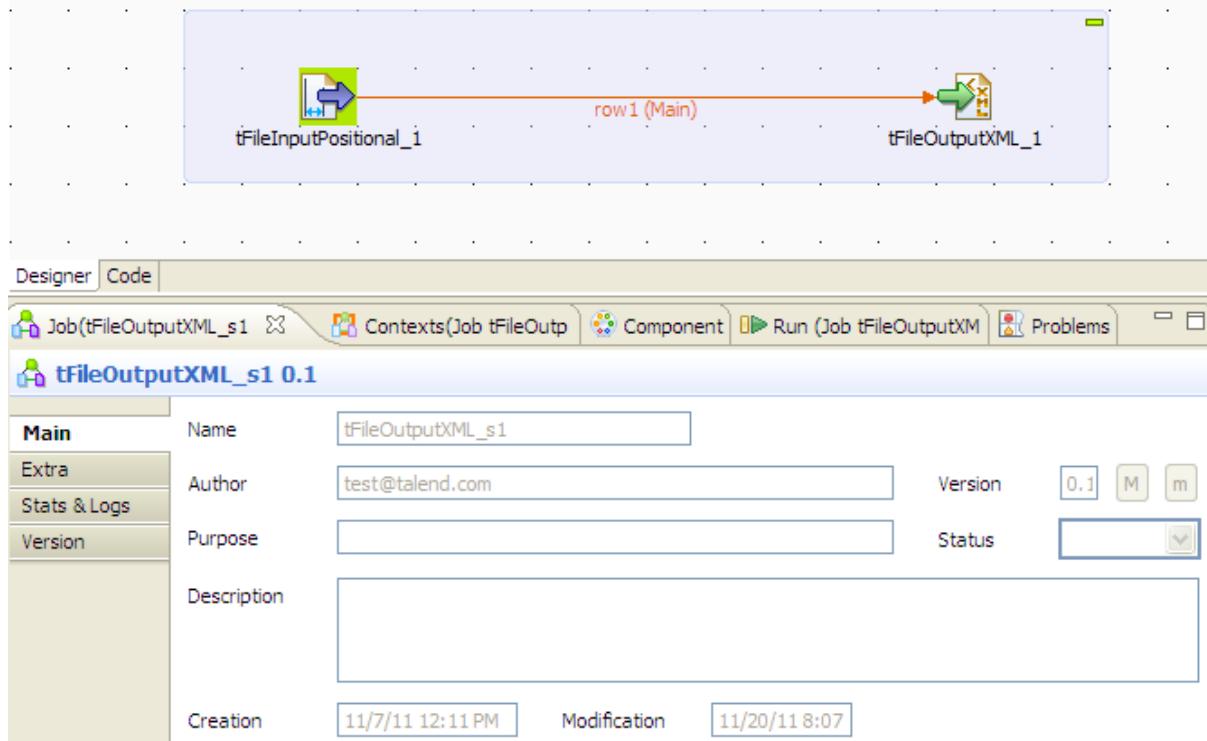
Cliquez sur l'icône minimiser/maximiser (□/□) pour réduire ou agrandir le panneau correspondant. Pour plus d'informations sur comment afficher ou masquer un panneau ou une vue, consultez la [Section 4.2.8.3, « Afficher les onglets/vues de configuration des Jobs »](#).

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 la [Section A.7, « Raccourcis clavier »](#).

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 for Data Integration*.

4.2.8.3. 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**, **Context** et **Talend Exchange** rassemblent toutes les informations relatives aux éléments graphiques sélectionnés dans l'espace de modélisation ou à l'exécution du Job ouvert.

Par défaut, lorsque vous lancez *Talend Open Studio for Data Integration* pour la première fois, l'onglet **Problems** ne s'affichera qu'au moment où vous créerez votre premier Job. Ensuite, cet onglet s'affichera automatiquement.

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

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

Pour une description détaillée concernant ces onglets, consultez la [Section 4.2.8.3, « Afficher les onglets/vues de configuration des Jobs »](#).

4.3. Utilisation des connexions

Dans *Talend Open Studio for Data Integration*, 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.

4.3.1.1. 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.

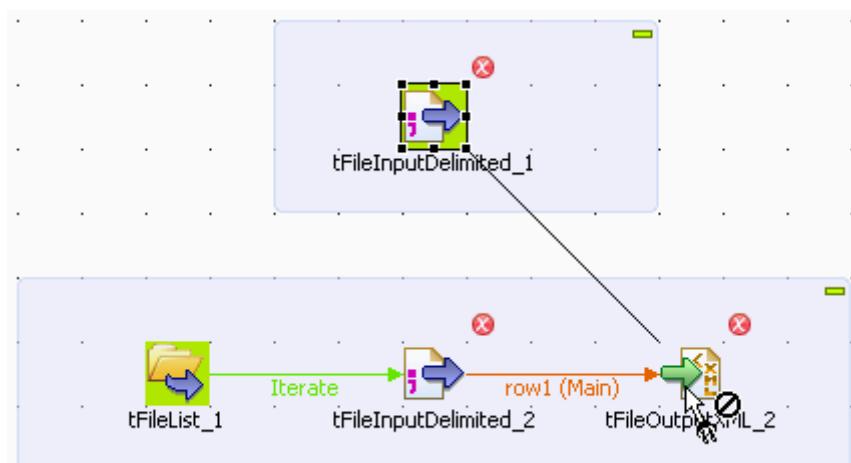
Main

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.



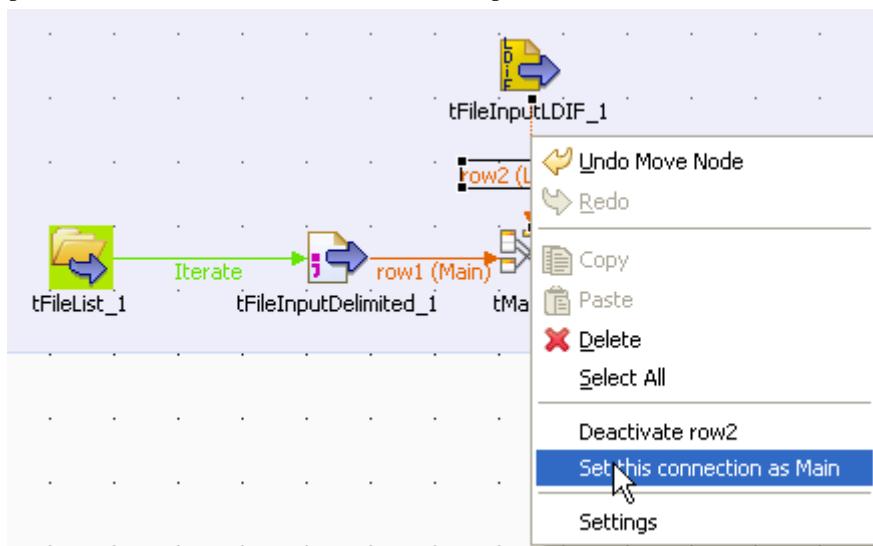
Cliquez droit sur le composant d'entrée et sélectionnez **Row > Main** dans la liste des connexions.

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 la section intitulée « [Job à entrées/sorties multiples](#) ».

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 : [la section intitulée « Job à entrées/sorties multiples »](#).

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 inapproprié, 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 for Data Integration**.

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 inapproprié, valeur Null non définie, format de date erroné, etc.).

Consultez également : [Section 6.2.4.6, « Gestion des erreurs ».](#)

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 : [la section intitulée « Job à entrées/sorties multiples ».](#)

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 [Section 4.5.1, « Mapper des flux de données ».](#)

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

4.3.1.2. 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.

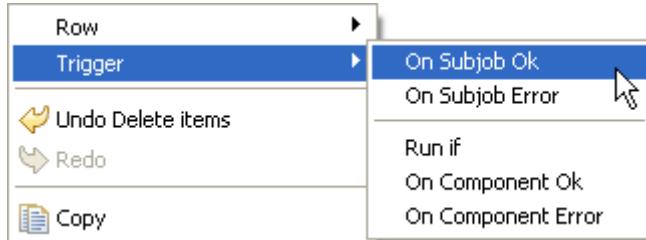
Certains composants ne peuvent fonctionner avec d'autres composants qu'à l'aide d'un lien d'itération, c'est le cas du composant **tFilelist**. Pour plus d'informations concernant la configuration d'une connexion **Iterate**, consultez [Section 4.3.2.2, « Paramètres d'une connexion Iterate »](#).

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

4.3.1.3. Connexion de type Trigger

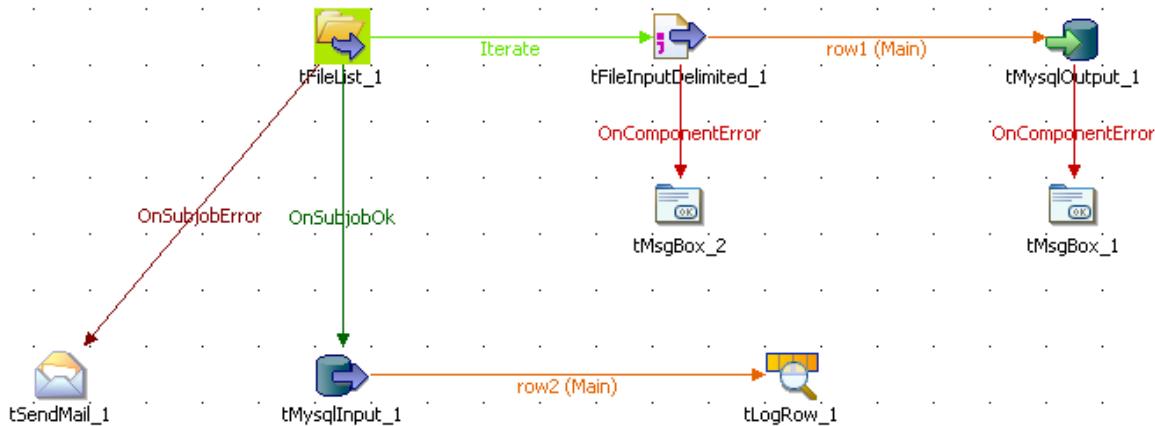
Les connexions de déclenchement, **Trigger**, aident à définir le processus de traitement. Les connexions de type **Trigger** ne servent pas au transfert de données.

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

Voir également : [Section 4.6.2, « Configurer un composant Start ».](#)

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 [Section 4.3.2.3, « Paramètres d'une connexion Trigger ».](#)

4.3.1.4. 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.

4.3.2.1. 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 [la section intitulée « Paramétrer un schéma built-in ».](#)

Column	Key	T...	N...	Date...	L...	P...	D...	C...
id	<input checked="" type="checkbox"/>	In...	<input checked="" type="checkbox"/>		2	0		
CustomerID	<input type="checkbox"/>	St...	<input checked="" type="checkbox"/>		255	0		
CustomerName	<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		
RegisterDate	<input type="checkbox"/>	St...	<input checked="" type="checkbox"/>		50	0		

L'onglet **Advanced settings** vous permet de montrer 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 *Activity Monitoring Console*. Pour plus d'informations concernant *Activity Monitoring Console*, consultez le **Guide utilisateur** de *Activity Monitoring Console*.

Basic settings

Advanced settings

Use input connection name as label

Label: "MyLabel"

Mode: Absolute *

Thresholds

Label	Low end	Top end	Color

[+]

Monitor this connection

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

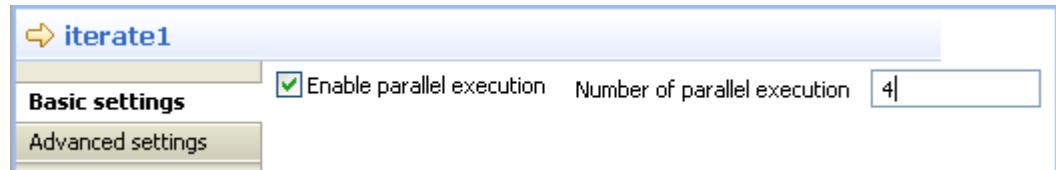
1. Cochez la case **Monitor this connection**.
2. Cochez la case **Use input connection name as label** afin d'utiliser le nom du flux d'entrée comme libellé des données à enregistrer dans le log, ou saisissez le libellé dans le champ **Label**.
3. Dans la liste **Mode**, sélectionnez **Absolute** pour 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**.
4. 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 le **Guide de référence des Composants Talend Open Studio** et le **Guide utilisateur** de *Activity Monitoring Console*.

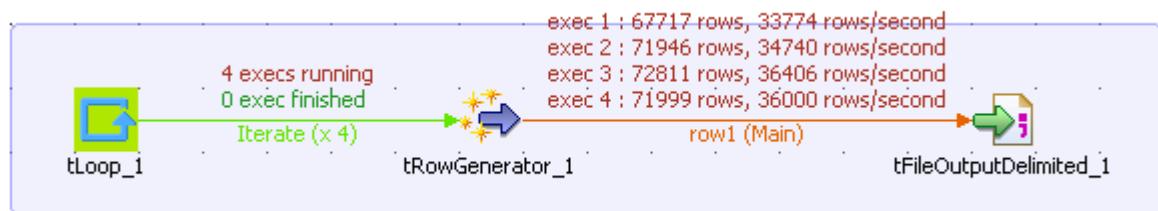
4.3.2.2. Paramètres d'une connexion Iterate

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

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



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

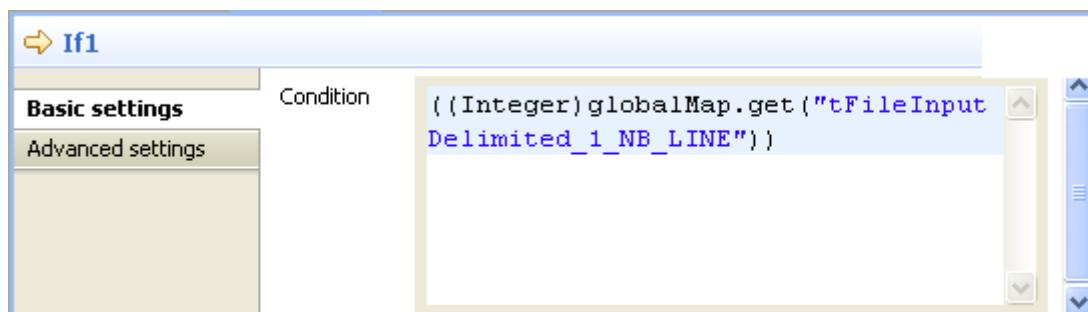


3. Cochez la case **Statistics** de l'onglet **Run**, pour afficher, en temps réel, les exécutions en parallèles dans l'espace de modélisation.

4.3.2.3. 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 les conditions de votre projet en Java. Le raccourci clavier **Ctrl+Barre d'espace** vous donne accès à toutes les variables globales et de contexte.



4.4. Utilisation du Metadata Manager

Talend Open Studio for Data Integration 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 for Data Integration 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

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 [Chapitre 7, Gestion des métadonnées](#).

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 for Data Integration 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 nœud **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**.

4.4.2.1. 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 prédéfinies.

<p>Description: Error Message</p> <p>Global variable, property of component tMap [tMap_1].</p> <p>Type: String</p> <p>Availability: After</p> <p>Variable Name: ((String)globalMap.get("tMap_1_ERROR_MESSAGE"))</p>	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 :

- [Section 4.4.2.4, « Définir les variables à partir de la vue Component ».](#)
- [Section 4.4.2.2, « Définir les variables à partir de la vue Contexts ».](#)

4.4.2.2. 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 [Section 4.4.2.4, « Définir les variables à partir de la vue Component ».](#)

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 for Data Integration*, 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.

Variables			
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



Original order

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.
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 script affiché sera : <code>context.VotreNomDeParametre</code> 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 [Section 4.4.2.4, « Définir les variables à partir de la vue Component »](#).

Pour plus d'informations concernant les variables du **Repository**, consultez [Section 4.4.2.5, « Centraliser les contextes dans le Repository »](#).

Onglet Values as tree

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

Variables				
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 [Section 4.4.2.3, « Configurer les contextes »](#).

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'informations concernant la définition des variables, consultez [Section 4.4.2.4, « Définir les variables à partir de la vue Component »](#) et [Section 4.4.2.5, « Centraliser les contextes dans le Repository »](#).

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, consultez [Section 4.4.2.3, « Configurer les contextes »](#).

Pour plus d'informations concernant la définition des variables, consultez [Section 4.4.2.4, « Définir les variables à partir de la vue Component »](#) et [Section 4.4.2.5, « Centraliser les contextes dans le Repository »](#).

4.4.2.3. Configurer les contextes

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

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



 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.

1. Pour créer un nouveau contexte, cliquez sur **New** dans la boîte de dialogue [**Configure Contexts**].
2. Saisissez le nom de votre nouveau contexte.



3. 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 :

1. Cliquez sur **Edit** dans la boîte de dialogue [**Configure contexts**] et saisissez le nouveau nom.
2. 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 [Section 4.4.2.2, « Définir les variables à partir de la vue Contexts »](#).

4.4.2.4. 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 [Section 4.4.2.2, « Définir les variables à partir de la vue Contexts »](#).

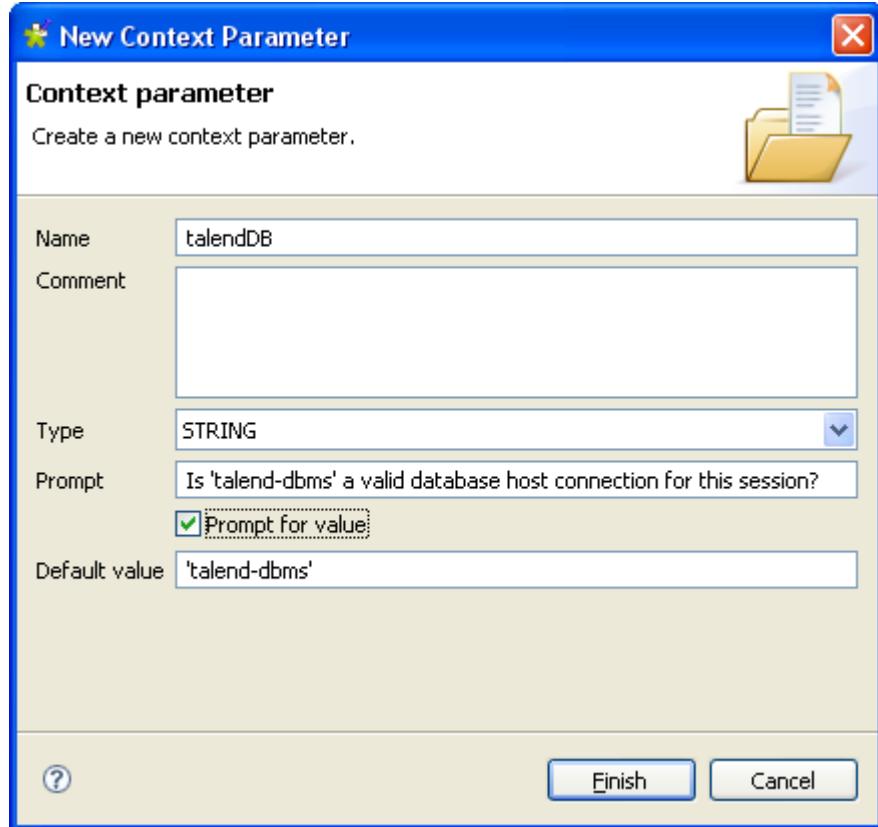
Pour plus d'informations concernant la définition des variables du **Repository**, consultez [Section 4.4.2.5, « Centraliser les contextes dans le Repository »](#).

Création rapide des variables de contexte

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

1. Dans la vue **Component** correspondante, placez votre curseur dans le champ que vous souhaitez paramétriser.

2. Appuyez sur **F5** pour afficher la boîte de dialogue de paramétrage de contexte :



3. Donnez un nom à cette nouvelle variable dans le champ **Name**, renseignez le champ **Comment** et choisissez le **Type**.
4. Saisissez un message **Prompt** à afficher pour confirmer l'utilisation de cette variable lors de l'exécution du Job courant (généralement utilisé comme test). Cochez la case **Prompt for value** pour rendre le champ éditables.
5. 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.
6. Cliquez sur **Finish** pour valider.
7. 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 [Section 4.4.2.3, « Configurer les contextes »](#).

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.

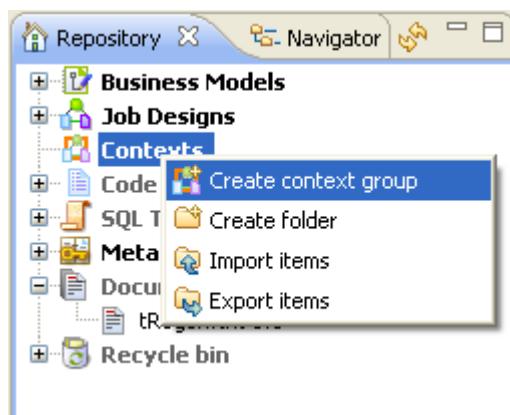
Pour plus d'informations concernant le paramétrage de **StoreSQLQuery**, reportez-vous au **Guide de référence** des *Composants Talend Open Studio*, et plus particulièrement le composant **tDBInput**.

4.4.2.5. 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

1. Cliquez droit sur le nœud **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.

2. Etape 1/2 : renseignez le nom du **context group** que vous êtes en train de créer et ajoutez des informations générales, comme une description, si nécessaire.
3. Cliquez sur **Next** pour passer à l'étape 2/2, qui vous permet de définir les différents contextes et les différentes variables dont vous avez besoin.

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

Procédure 4.1. Définir les variables de contexte à utiliser comme base pour vos autres contextes

1. 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**.

2. Sélectionnez le **Type** de variable dans la liste.

Le **Script code** varie en fonction du type de variable sélectionné. La capture d'écran ci-dessous vous montre le code Java généré.

3. 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 [Section 4.4.2.3, « Configurer les contextes »](#).

4. 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			

5. 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 nœud **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 [Section 7.17, « Exporter une métadonnée en tant que contexte »](#).

4.4.2.6. Appliquer des variables de contexte dans un Job du Repository

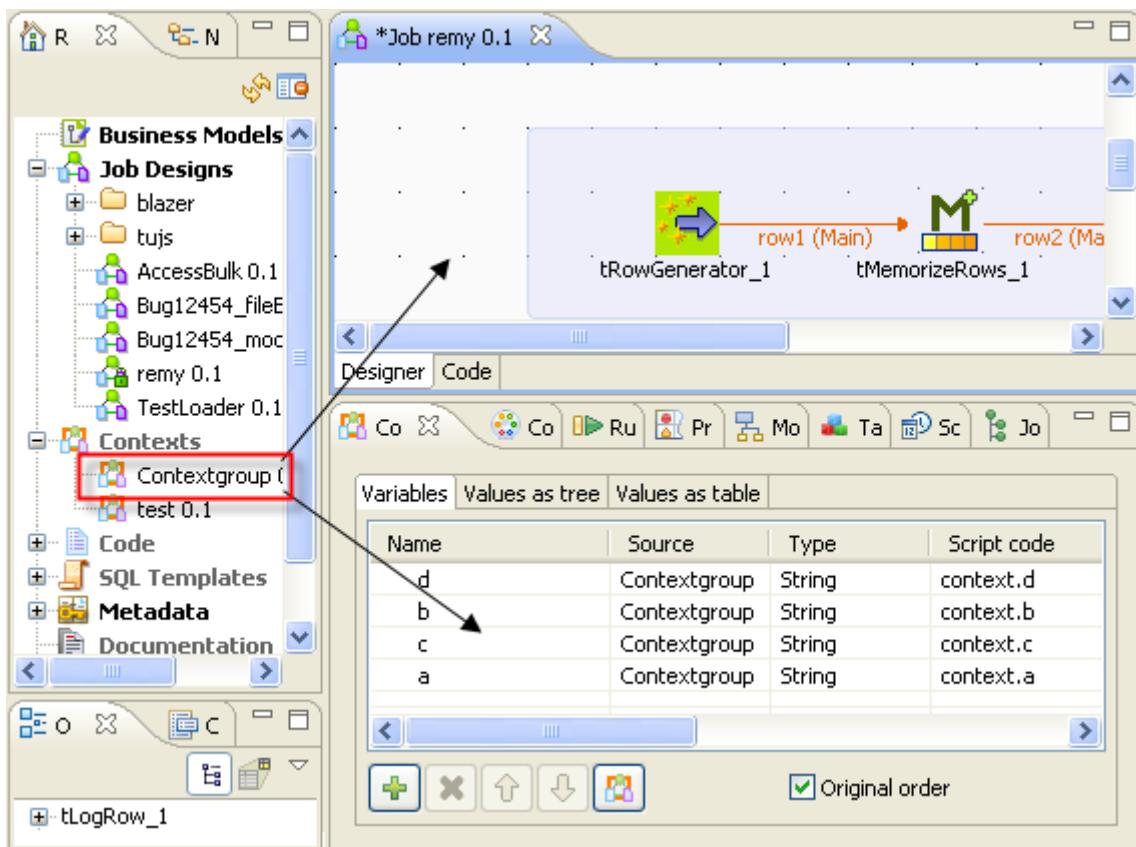
Une fois le groupe de contexte créé et stocké dans le **Repository**, vous pouvez l'appliquer à un Job par deux moyens :

1. Déposez un groupe de contexte. Ainsi, le groupe sera appliqué comme un tout.
2. 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 :

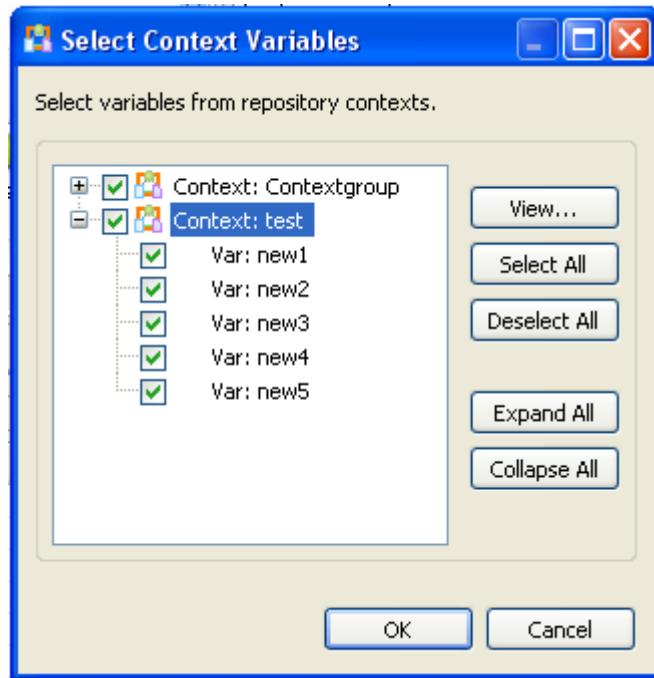
1. Double-cliquez sur le Job auquel jouter un groupe de contexte.
2. Une fois le Job ouvert, glissez le groupe de contexte de votre choix, soit dans l'espace de modélisation graphique du Job, soit dans la vue **Contexts** sous l'espace de travail.



Utiliser l'icône de contexte

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

1. Double-cliquez sur le Job auquel ajouter une variable de contexte.
2. 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.
3. Au bas de la vue **Contexts**, cliquez sur l'icône  afin d'ouvrir l'assistant pour sélectionner les variables de contexte à appliquer.



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



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

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

4.4.2.7. Exécuter un Job dans un contexte défini

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

Context	Target execution								
Test	<table border="1"> <thead> <tr> <th>Name</th><th>Value</th></tr> </thead> <tbody> <tr> <td>host</td><td>'talend-dbms-test'</td></tr> <tr> <td>port</td><td>'8080'</td></tr> <tr> <td>db</td><td>'Talend-test'</td></tr> </tbody> </table>	Name	Value	host	'talend-dbms-test'	port	'8080'	db	'Talend-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**.

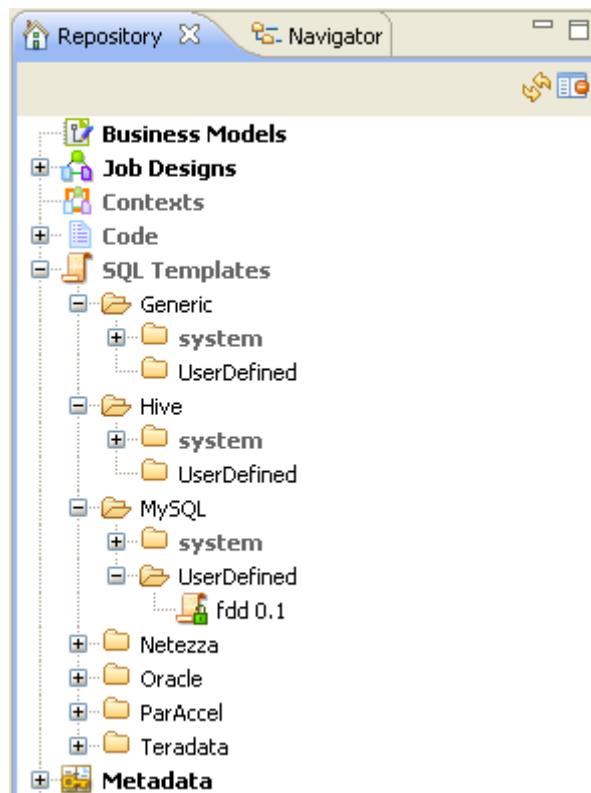
Voir également :

- [Section 4.4.2.2, « Définir les variables à partir de la vue Contexts ».](#)
- [Section 4.4.2.5, « Centraliser les contextes dans le Repository ».](#)

4.4.3. Utiliser les SQL Templates

Talend Open Studio for Data Integration 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 for Data Integration*. 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 for Data Integration*, consultez le [Chapitre 9, SQL Templates](#).

Pour plus d'informations concernant la création de modèles SQL personnalisés et leur utilisation dans un Job, consultez 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 le [Chapitre 6, Mapping de flux de données](#).

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

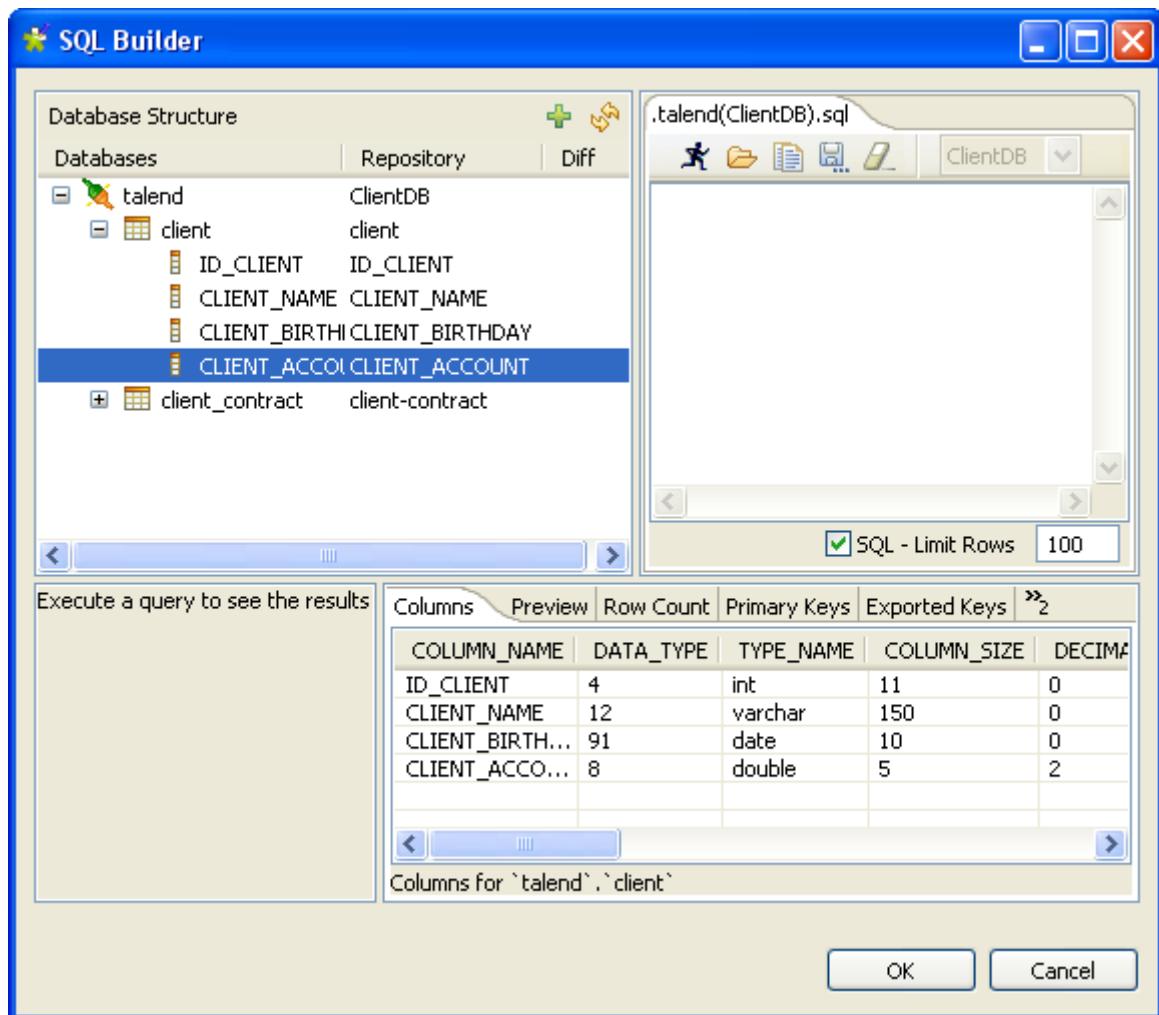
4.5.2. Créer une requête à l'aide de SQLBuilder

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 [...].



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.

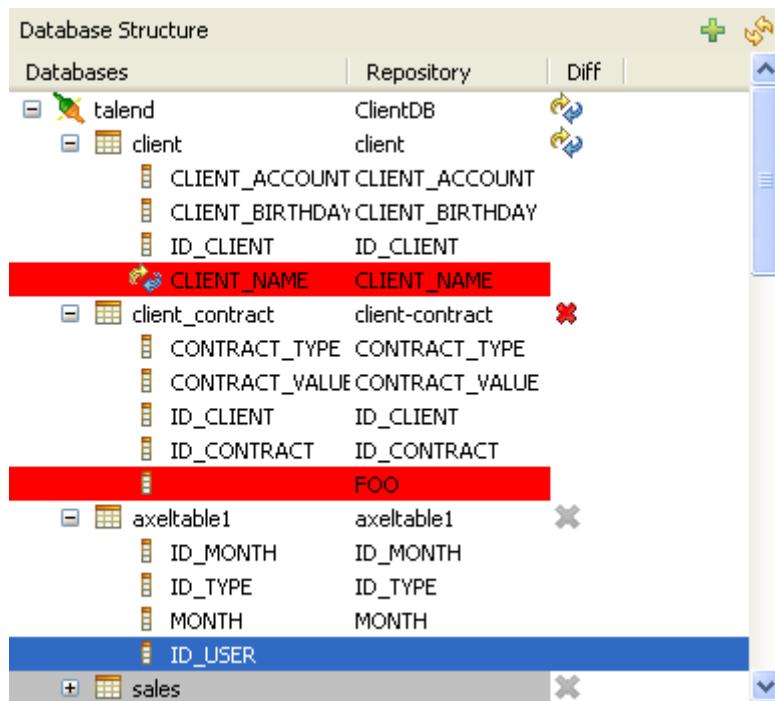
La vue **Schema**, en bas à droite de la fenêtre, fournit une description des colonnes.

4.5.2.1. 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.



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.

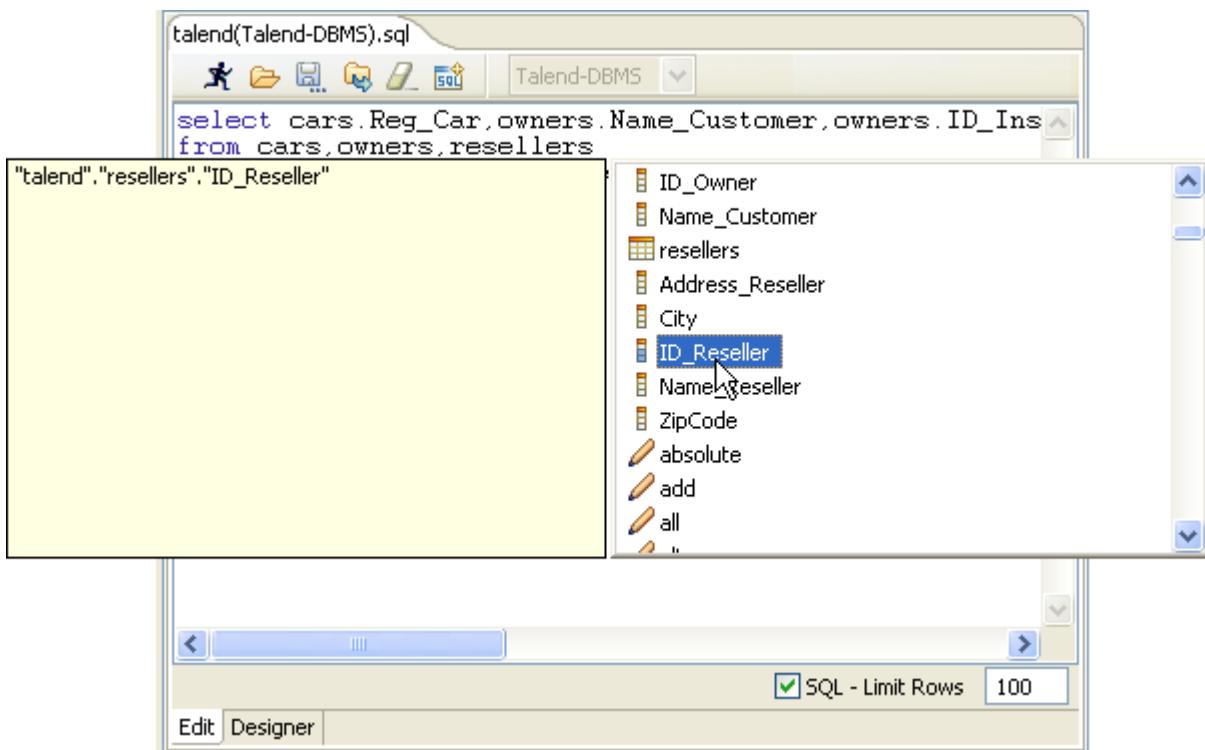
Le surlignage bleu indique que la colonne est présente dans la base de données elle-même mais qu'elle n'est pas présente dans la table de la métadonnée.

4.5.2.2. 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, procédez comme suit :

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

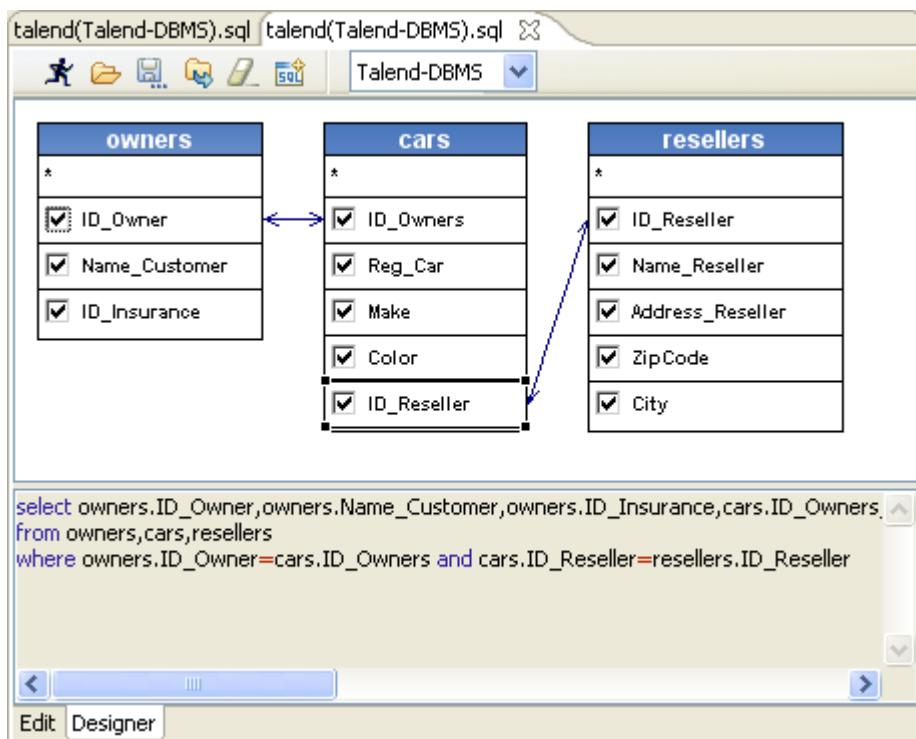


Alternativement, le **Designer** graphique de requête vous permet de manipuler facilement des tables et de générer en temps réel la requête correspondante dans l'onglet **Edit**.

3. Cliquez sur l'onglet **Designer** pour passer du mode **Edit** manuel au mode graphique.
- Certaines déclarations SQL ne peuvent être interprétées par défaut. Décochez la case correspondant aux colonnes que vous souhaitez exclure de la sélection.
4. Si vous avez sélectionné une table, toutes les colonnes sont sélectionnées par défaut. Décochez la case correspondant aux colonnes que vous souhaitez exclure de la sélection.
 5. Dans l'onglet **Designer**, ajoutez d'autres tables en sélectionnant **Add tables** dans le menu contextuel puis en choisissant la table à ajouter dans la liste.

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

Vous pouvez également créer facilement une jointure entre deux tables. Cliquez droit sur les colonnes de la première table à lier et sélectionnez **Equal** dans le menu contextuel. Le champ en sélection est automatiquement lié au champ de la seconde table.



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



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

6. Lorsque votre requête est terminée, exécutez-la en cliquant sur l'icône du curseur.

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 **Results**.

4.5.2.3. 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/charger des composants de la communauté

Talend Open Studio for Data Integration vous permet d'accéder à une liste de composants communautaires dans **Talend Exchange** qui sont compatibles avec votre version de *Talend Open Studio for Data Integration*. Vous

pouvez télécharger ces composants pour les réutiliser dans les Jobs Designs de votre Studio. Depuis **Talend Open Studio for Data Integration**, vous pouvez également charger les composants que vous avez créés vers **Talend Exchange** afin de les partager avec les autres utilisateurs de la Communauté.

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

- composants compatibles dans **Talend Exchange** que vous pouvez télécharger et installer,
- composants que vous avez téléchargés et installés dans une version précédente de *Talend Open Studio for Data Integration*, mais pas encore dans votre version actuelle du Studio,
- composants que vous avez créés, pouvant être dans **Talend Exchange**, pour les partager avec les autres utilisateurs de la Communauté.



Avant de pouvoir télécharger des composants de ou charger vos propres composants vers la Communauté, vous devez vous connecter à **Talend Exchange** depuis votre Studio. Si vous ne vous connectez pas à **Talend Exchange** au lancement du Studio, vous pouvez toujours vous y connecter depuis la page de préférences **Talend Exchange**. Pour plus d'informations, consultez la [Section 2.5.3, « Préférence Exchange »](#).

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 chargement lors de l'installation de certains composants à partir de **Talend Exchange**, que le nom d'un composant soit différent dans la **Palette** de celui dans la vue **Exchange**, ou que vous ne trouviez pas certains composants dans votre **Palette** après une installation apparemment réussie.

4.5.3.1. Installer des composants de la communauté à partir de Talend Exchange

Pour installer des composants de la communauté à partir de **Talend Exchange** dans la **Palette** de votre *Talend Open Studio for Data Integration* actuel :

1. Cliquez sur le lien **Exchange** dans la barre d'outils de *Talend Open Studio for Data Integration* pour ouvrir la vue **Exchange** dans l'espace de travail.

Available Extensions						
Downloaded Extensions		Extension Name	Version	Rating	Author	View
	My Extensions	Facebook Application Insights Component	0.1	★★★★★	saburo	view/download
		tFileOutputDelimitedEx	1.0	★★★★★	Alezis	view/download
		tScriptRules	1.0	★★★★★	walkerca	view/download
		tWorldBank components Demo	0.2	★★★★★	saburo	view/download
		pUpdateMailOffer	V02	★★★★★	PayZen	view/download
		pCreateMailOffer	V02	★★★★★	PayZen	view/download
		tDBFOutput	1.1	★★★★★	BiSi	view/download
		tDBFInput	1.1	★★★★★	BiSi	view/download
		pCreatePayment	V06	★★★★★	PayZen	view/download

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

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

The screenshot shows a software download page for 'tPDFToText'. At the top, there's a back arrow icon. Below it, the title 'tPDFToText' is displayed in bold, followed by 'Version 1.1' and the date '2011-05-17'. A descriptive text box states: 'Convert a PDF to text file. It's possible to extract a delimited area.' To the right, there's a preview image of the software's user interface, which looks like a configuration dialog with various tabs and settings. Below the preview is a 5-star rating with four stars filled. A large green 'Install' button is positioned at the bottom right of the preview area.

User Reviews

[write a review](#)



bien it's good



It works on 4.2.2 You can use it, it works on 4.2.2!



tPDFToText This does not seem compatible with TOS 4.2.2r63143. I have installed it on TOS and it does not generate an output file.

4. Lisez les informations relatives au composant, notamment sa description et les commentaires des utilisateurs, ou écrivez votre propre commentaire, et/ou notez le composant, si vous le souhaitez. Pour plus d'informations concernant les remarques et les notes des composants de la Communauté, consultez la [Section 4.5.3.3, « Commenter et noter un composant de la Communauté »](#).

Si nécessaire, cliquez sur la flèche pointant vers la gauche afin de retourner à la liste des composants.

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

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

Available Extensions	Extension Name	Downloaded Version	Download Date	Install/Update
Downloaded Extensions				
My Extensions	tDBFInput	1.1	2011-11-17	Install
	tDBFOutput	1.1	2011-10-31	Install
	bcLogbackConfig	1.2	2011-10-31	Install
	bcLogbackCatch	1.3	2011-10-31	Install
	tLog4J	1.4	2011-11-17	Install
	tFileOutputDelimitedEx	1.0	2011-11-17	Install
	null	null	2011-11-17	Install
	tScriptRules	1.0	2011-11-17	Install
	BRules	1.1	2011-11-17	Install
	tPDFToText	1.1	2011-11-18	Installed

4.5.3.2. Réinstaller ou mettre à jour des composants de la Communauté

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



Par défaut, lorsque vous êtes connecté à **Talend Exchange**, une boîte de dialogue apparaît pour vous notifier lorsqu'une mise à jour d'un composant de la communauté installé est disponible. Si vous vérifiez souvent les mises à jour des composants communautaires et que vous ne souhaitez pas que cette boîte de dialogue apparaisse à nouveau, vous pouvez la désactiver dans les préférences de **Talend Exchange**. Pour plus d'informations, consultez la [Section 2.5.3, « Préférence Exchange »](#).

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

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

Dans la vue **Downloaded Extensions**, les composants que vous avez installés dans votre version précédente de *Talend Open Studio for Data Integration* mais pas encore dans votre Studio actuel affichent un lien **Install** dans la colonne **Install/Update**. Les composants dont une mise à jour est disponible dans **Talend Exchange** ont un lien **Update**.

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

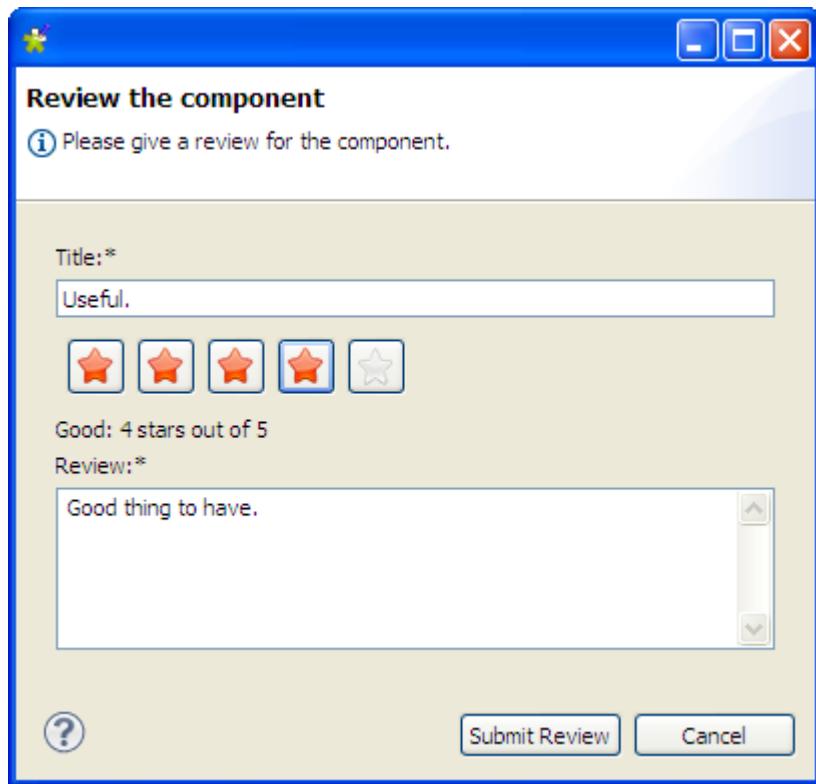
Un indicateur de progression apparaît pour vous montrer l'avancement du téléchargement et de l'installation. Lorsque l'installation est terminée avec succès, la vue **Downloaded Extensions** s'ouvre et affiche le statut du composant, **Installed**.

4.5.3.3. Commenter et noter un composant de la Communauté

Pour commenter et noter un composant de la Communauté :

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

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



- Renseignez les informations requises, un titre et un commentaire, cliquez sur l'une des cinq étoiles afin de noter le composant, puis cliquez sur **Submit Review** afin de soumettre votre commentaire au serveur **Talend Exchange**.

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

4.5.3.4. Charger dans Talend Exchange un composant créé

Vous pouvez créer vos propres composants afin de les utiliser dans vos Jobs, dans *Talend Open Studio for Data Integration* et les charger dans **Talend Exchange** pour les partager avec les autres utilisateurs de **Talend**. Pour plus d'informations concernant la création et le déploiement de vos propres composants dans *Talend Open Studio for Data Integration*, consultez [Section 2.5.2, « Composants externes ou utilisateur »](#).

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

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

Available Extensions				Add New Extension
Downloaded Extensions				
My Extensions	Extension Name	Version	Upload Date	Operation

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

Extension Title: tExEvaluate Initial Version: 1.0

Compatibility:

- All versions
- Version and older: []
- Versions and newer: 5.0
- All versions except: []
- Only these versions: []

Description: tExEvaluate evaluates the execution of a Job.

File: D:\Work\Components\tExEvaluate.zip

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

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

Available Extensions	Add New Extension			
Downloaded Extensions				
My Extensions	Extension Name	Version	Upload Date	Operation
	tExEvaluate	1.0	2011-11-18	

4.5.3.5. Gérer les composants chargés dans Talend Exchange

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

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

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

Initial Version: 2.0

Version:

Compatibility: All versions
 Version and older: _____
 Versions and newer: _____
 All versions except: _____
 Only these versions: _____

File: D:\Work\Components\tExEvaluate.zip

Upload Extension

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

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

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

1. Dans la vue **My Extensions**, cliquez sur l'icône dans la colonne **Operation** du composant dont vous souhaitez modifier les informations, pour ouvrir la page d'édition des informations du composant.

Extension Title: tExEvaluate

Compatibility: All versions
 Version and older: _____
 Versions and newer: 5.1
 All versions except: _____
 Only these versions: _____

Description: tExEvaluate evaluates the execution of a Job.
It requires tExCather to work with it.

Modify Extension

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

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

4.5.4. Installer des modules externes

L'utilisation de certains composants dans *Talend Open Studio for Data Integration* 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 ne sont pas présents par défaut et doivent donc être installés, pour que vous puissiez exécuter correctement les Jobs utilisant ces composants.

- 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.
	Installed	tWebService	geronimo-jaxws_2.1_spec...	Required for using this component.
	Installed	tWebService	wstx-asl-3.2.8.jar	Required for using this component.
	Installed	tWebService	saaj-api-1.3.jar	Required for using this component.
	Installed	tWebService	saaj-impl-1.3.2.jar	Required for using this component.
	Installed	tWebService	jclfs-1.3.11.jar	Required for using this component.
	Not installed	JBoss ESB Export	jbossesb-rosetta.jar	Library needed to export to JBos...
	Not installed	Db connection wizard	sas.core.jar	Required to define SAS connection
	Not installed	Db connection wizard	sas.svc.connection.jar	Required to define SAS connection
	Not installed	Db connection wizard	sas.intranet.javatools.jar	Required to define SAS connection
	Not installed	tHL7Input	hapi-structures-v51-0.6.jar	Required for using this component.
	Not installed	tJasperOutput	poi-3.2-FINAL-20081019.jar	Required for using this component.
	Not installed	tJasperOutputExec	jasperreports-3.1.4.jar	Required for using this component.
	Not installed	tJasperOutputExec	poi-3.2-FINAL-20081019.jar	Required for using this component.
	Not installed	tLDAPAttributesInput	courtinessl.jar	Required for using this component.
	Not installed	tMOMInput	log4j-1.2.13.jar	Required for using this component.
	Not installed	tMOMOutput	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 <i>Talend Open Studio for Data Integration</i> .
	Cette colonne liste les librairies externes ajoutées aux routines créées et sauvegardées dans le dossier des librairies de votre Studio. Pour plus

Colonne	Description
	d'informations, consultez la Section 8.4.3, « Éditer les bibliothèques des routines utilisateur ».
Module	Affiche le nom exact du module.
Description	Explique pourquoi le module/la librairie est requis(e).
Required	La case cochée indique que le module est requis.

Pour installer un module manquant, procédez comme suit :

1. 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.
2. 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

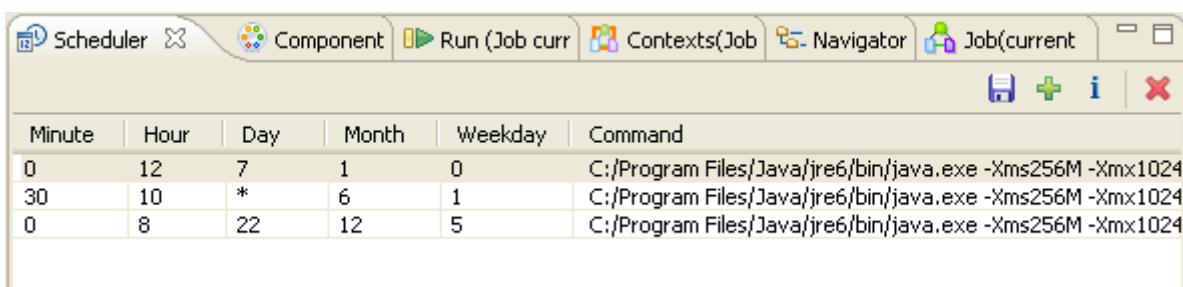
La vue **Scheduler** dans *Talend Open Studio for Data Integration* vous permet de planifier une tâche qui lancera à intervalles réguliers un Job via le programme de planification de tâche (crontab).

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.

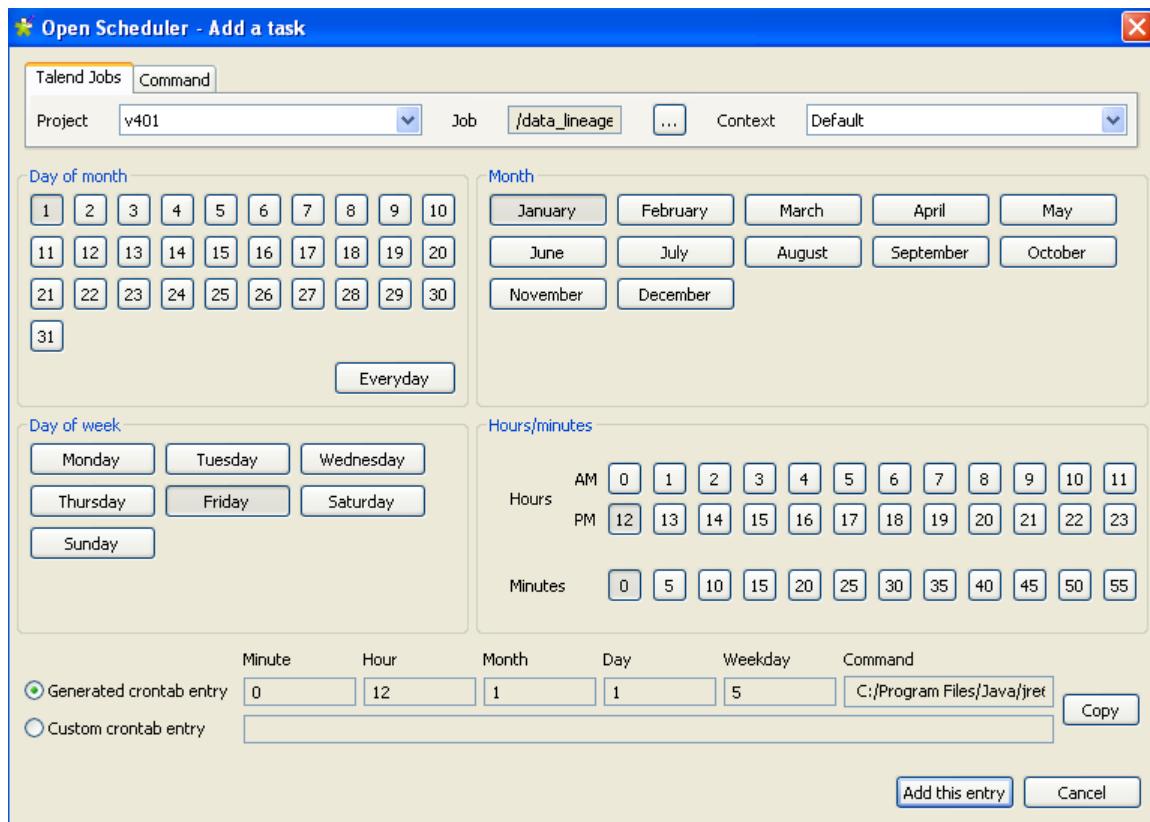


Cette vue est vide si vous n'avez programmé aucune tâche d'exécution de Job. Sinon, elle liste les paramètres de toutes les tâches planifiées.

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.

1. Cliquez sur l'icône  dans le coin supérieur droit de la vue **Scheduler**.

La boîte de dialogue [Open Scheduler] s'ouvre.



2. Dans la liste **Project**, sélectionnez le projet contenant le Job que vous souhaitez lancer périodiquement.
3. Cliquez sur le bouton [...] à côté du champ **Job** et sélectionnez le Job que vous souhaitez lancer à intervalles réguliers.
4. Dans la liste **Context**, s'il y a plusieurs contextes, sélectionnez celui avec lequel vous souhaitez exécuter le Job.
5. 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.

6. 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**.

7. 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.

8. 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.

9. 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.

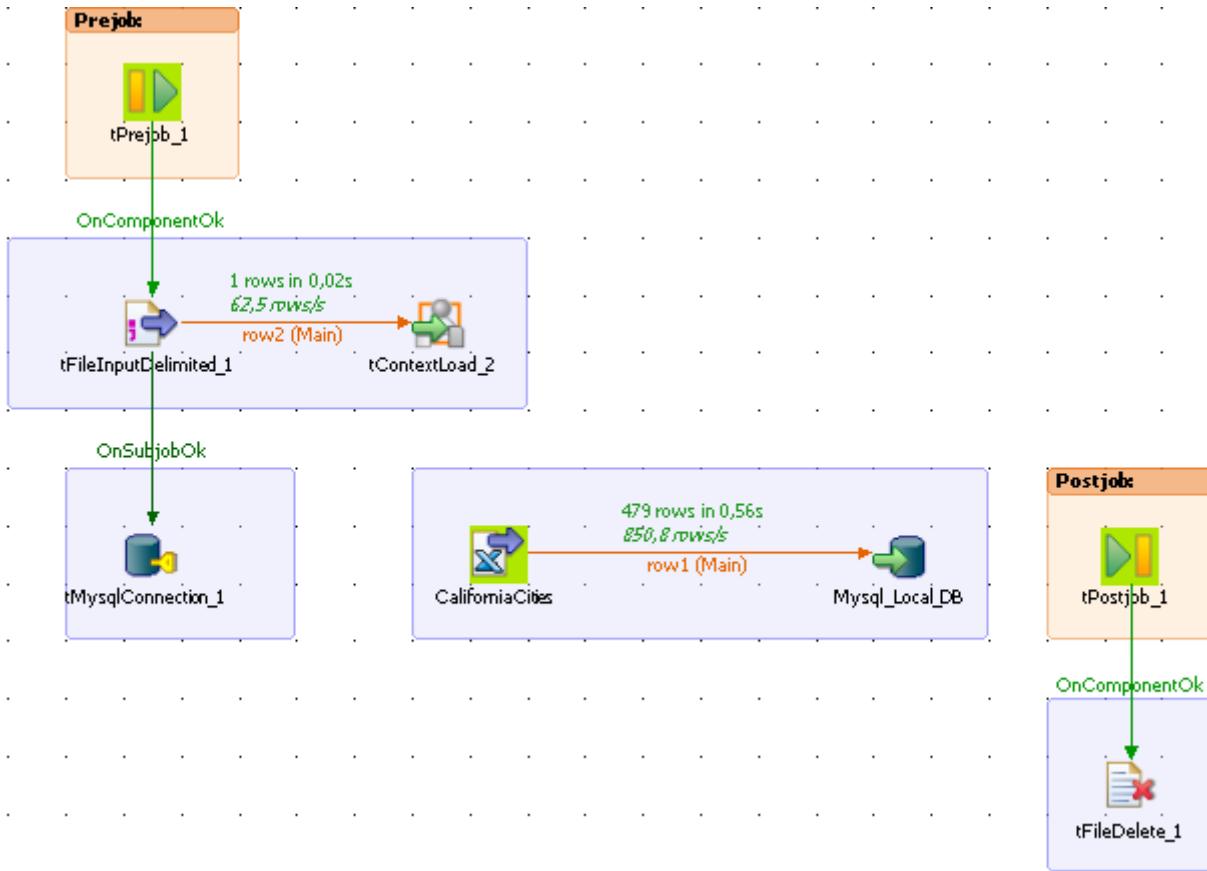
4.5.6. Utiliser les composants tPrejob et tPostjob

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.



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.

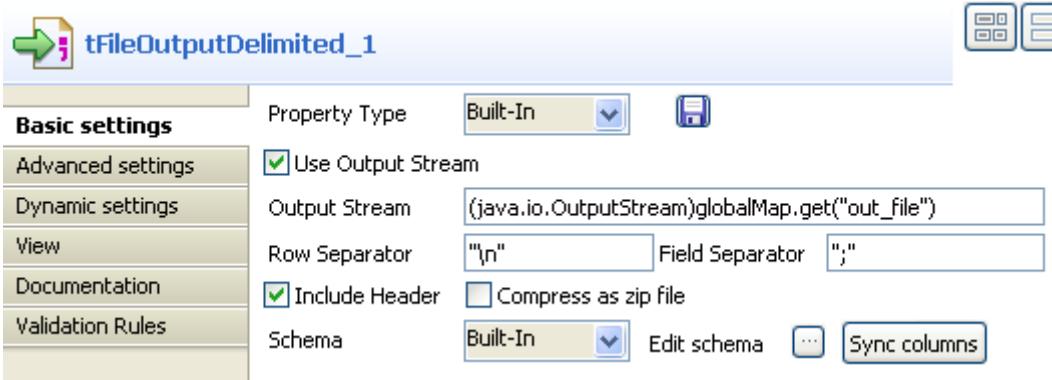
De nombreuses autres tâches collatérales à votre Job pouvant affecter la lisibilité de l'ensemble de votre Job peuvent aussi nécessiter un composant pré-job.

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.

4.5.7. Utiliser la fonctionnalité Use Output Stream

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



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

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



Avant d'utiliser la fonctionnalité de flux de sortie, vous devez avoir ouvert un flux. Pour un exemple détaillé de ce prérequis et de l'utilisation de cette fonctionnalité, consultez [Section B.2, « Utilisation de la fonctionnalité Output Stream »](#). Pour un exemple de Job utilisant cette fonctionnalité, consultez le second scénario du composant **tFileOutputDelimited**, dans le **Guide de référence** des *Composants Talend Open Studio*.

Pour un exemple détaillé de l'utilisation de cette fonctionnalité, consultez [Section B.2, « Utilisation de la fonctionnalité Output Stream »](#).

Pour un exemple de Job utilisant cette fonctionnalité, consultez le composant **tFileOutputDelimited** dans le **Guide de référence** des *Composants Talend Open Studio*.

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.

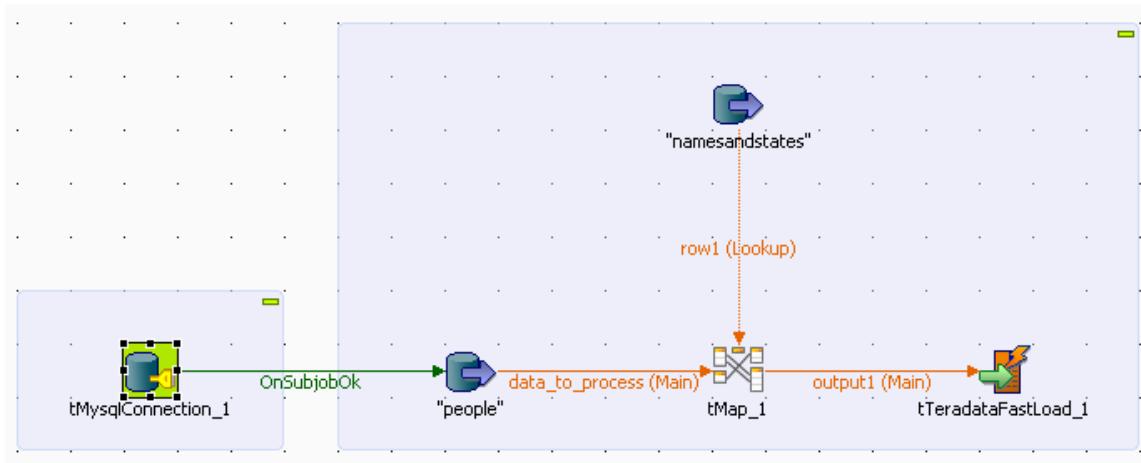
4.6.1. Partager une connexion à une base de données

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**.

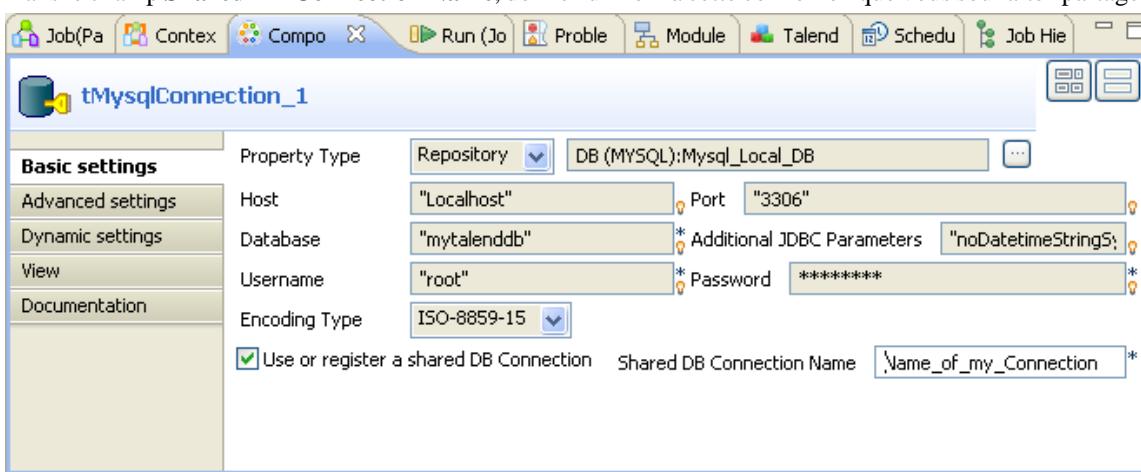
Cette case à cocher a été ajoutée à tous les composants de connexion afin de réduire le nombre d'ouvertures et de fermetures de connexion.

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

1. Cliquez-glissez un composant **tMysqlConnection** (si vous travaillez avec une base de données MySQL).
2. Connectez-le au premier composant de votre Job père.



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



Vous pouvez réutiliser cette connexion dans votre Job fils (et dans tout autre Job utilisant une connexion à cette même base de données).

5. Suivez les mêmes étapes présentées ci-dessus et utilisez le même nom dans le champ **Shared DB Connection Name**.

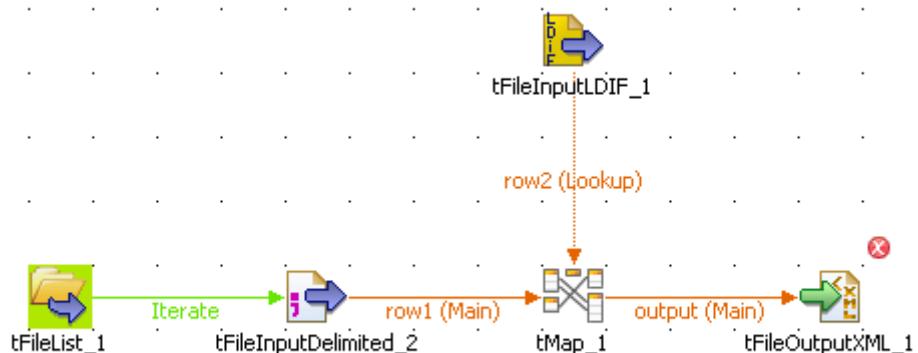
Pour plus d'informations concernant l'utilisation des composants de connexion, consultez le **Guide de référence des Composants Talend Open Studio**.

4.6.2. Configurer un 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 des 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 **Main** en connexion **Lookup**, simplement via un simple clic droit sur la connexion.

Voir également :

- [Section 4.3.1, « Types de connexions ».](#)
- [Section 5.1, « Activation/Désactivation d'un Job ou d'un sous-job ».](#)

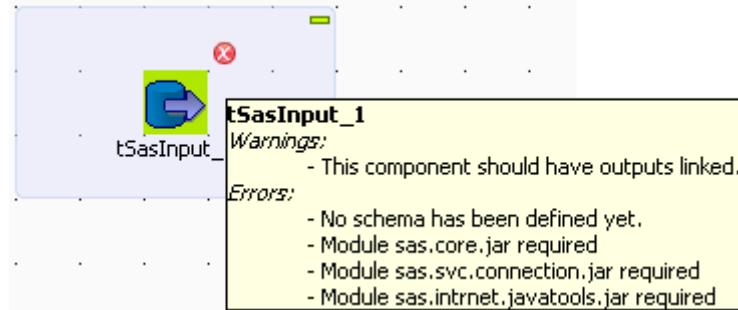
4.6.3. Gérer les icônes d'avertissement/d'erreur sur les composants

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**.

4.6.3.1. 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.

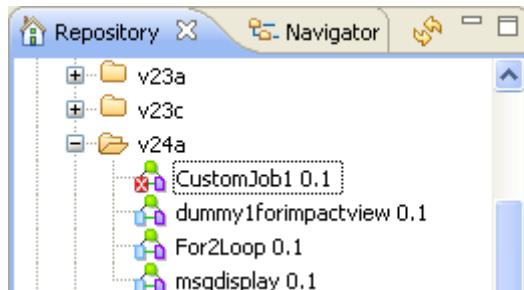
Placez votre souris sur le composant pour faire apparaître les messages d'erreur et d'avertissement. Cette aide contextuelle vous informe sur les données manquantes ou le statut du composant.



 Lorsque les bulles d'aide d'un composant indiquent qu'un module est requis, vous devez installer ce module pour ce composant via la vue **Module**. Cette vue est masquée par défaut. Pour plus d'informations concernant l'installation de modules externes à l'aide de cette vue, consultez la [Section 4.5.4, « Installer des modules externes »](#).

4.6.3.2. 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.

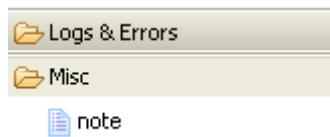


Cliquez sur **Cancel** pour arrêter l'exécution ou sur **Continue** pour continuer l'exécution de votre Job.

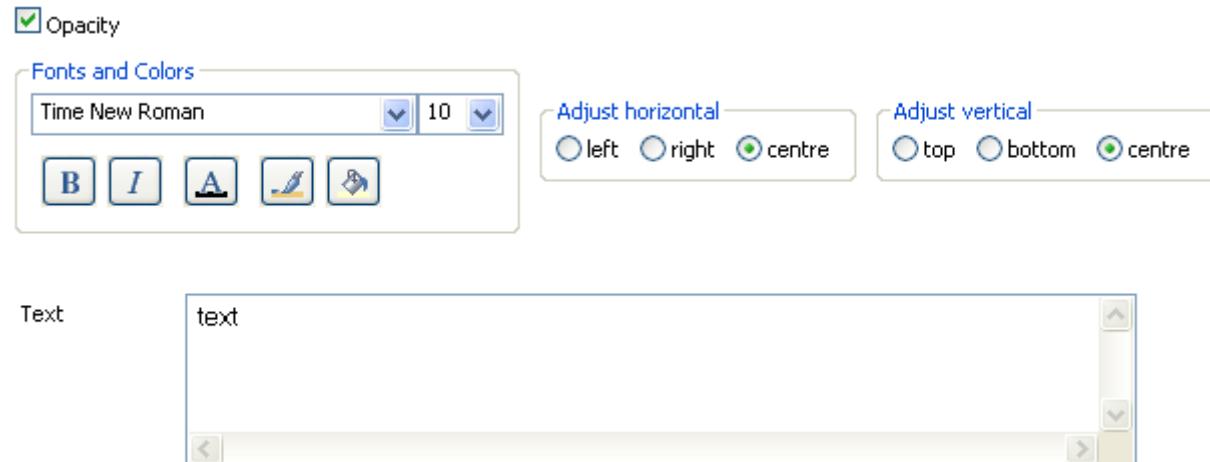
Pour plus d'informations concernant les erreurs dans les composants, consultez la [Section 4.6.3, « Gérer les icônes d'avertissement/d'erreur sur les composants »](#).

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.

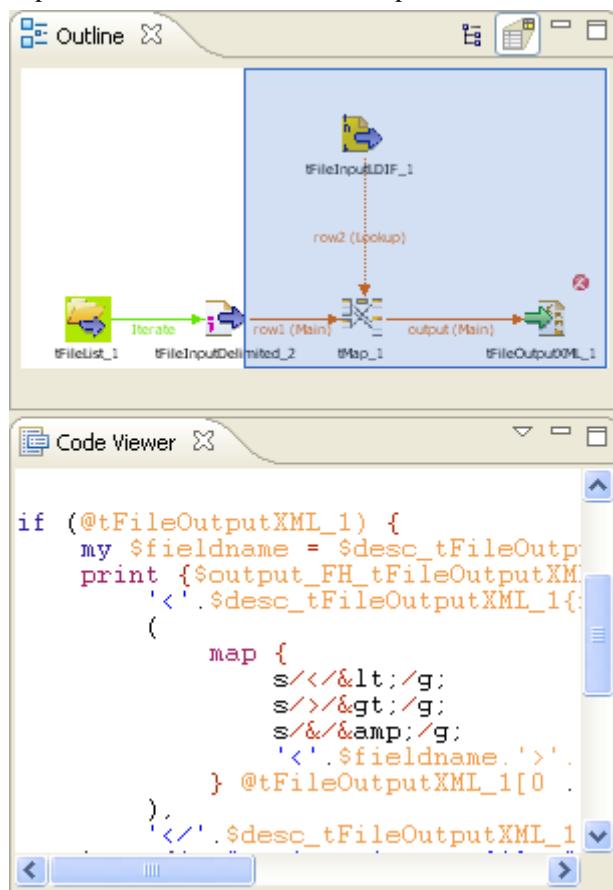
4.6.5. Afficher les onglets Code ou Outline de votre Job

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).

4.6.5.1. 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 nœud d'un composant pour afficher la liste des variables disponibles pour ce composant.

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

4.6.5.2. Code viewer

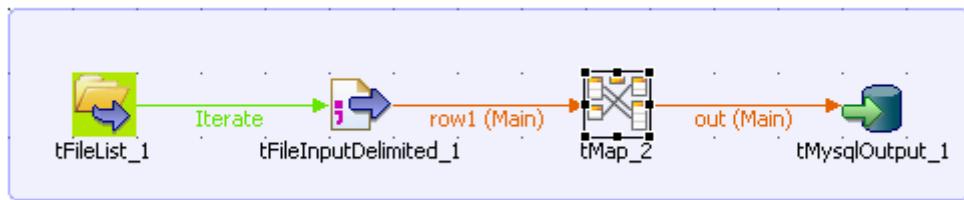
L'onglet **Code viewer** donne un aperçu des lignes de code générées pour le composant sélectionné, sous la vue active du Job Design, ainsi que le menu d'exécution, comprenant les éléments Start, Body et End correspondant aux phases d'exécution de ce composant.

 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.



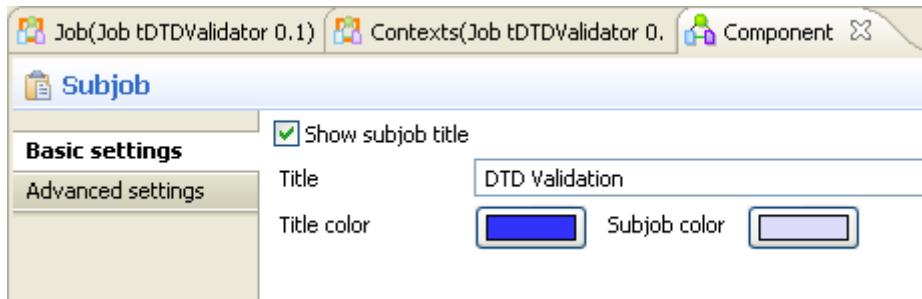
Ce surlignage bleu vous permet de distinguer plus facilement un sous-job d'un autre.

 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 [Section 4.5.6, « Utiliser les composants tPrejob et tPostjob »](#).

4.6.6.1. Personnaliser les sous-jobs

Les 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.

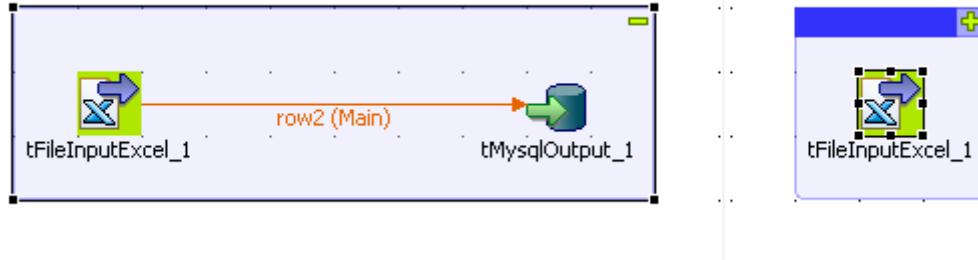
Pour modifier la couleur du titre ou du sous-job :

1. 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.

4.6.6.2. 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.



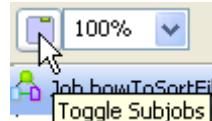
Cliquez sur le signe moins [-] pour réduire le sous-job (capture d'écran de gauche). Quand il est réduit, seul le premier composant du sous-job apparaît.

Cliquez sur le signe [+] pour le restaurer (capture d'écran de droite).

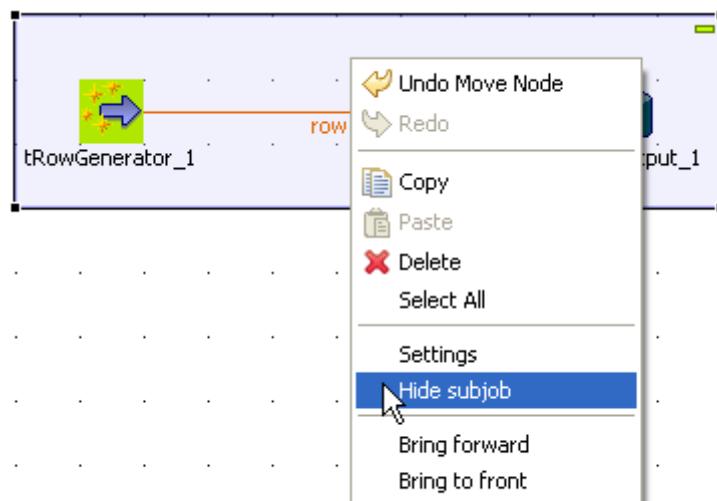
4.6.6.3. Supprimer le surlignage d'un sous-job

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 for Data Integration*.



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 [Section 4.6.7.1, « Automatiser l'utilisation de statistiques & logs »](#).

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 [Section 4.6.7.2, « Fonctions Extra »](#).

4.6.7.1. 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*.

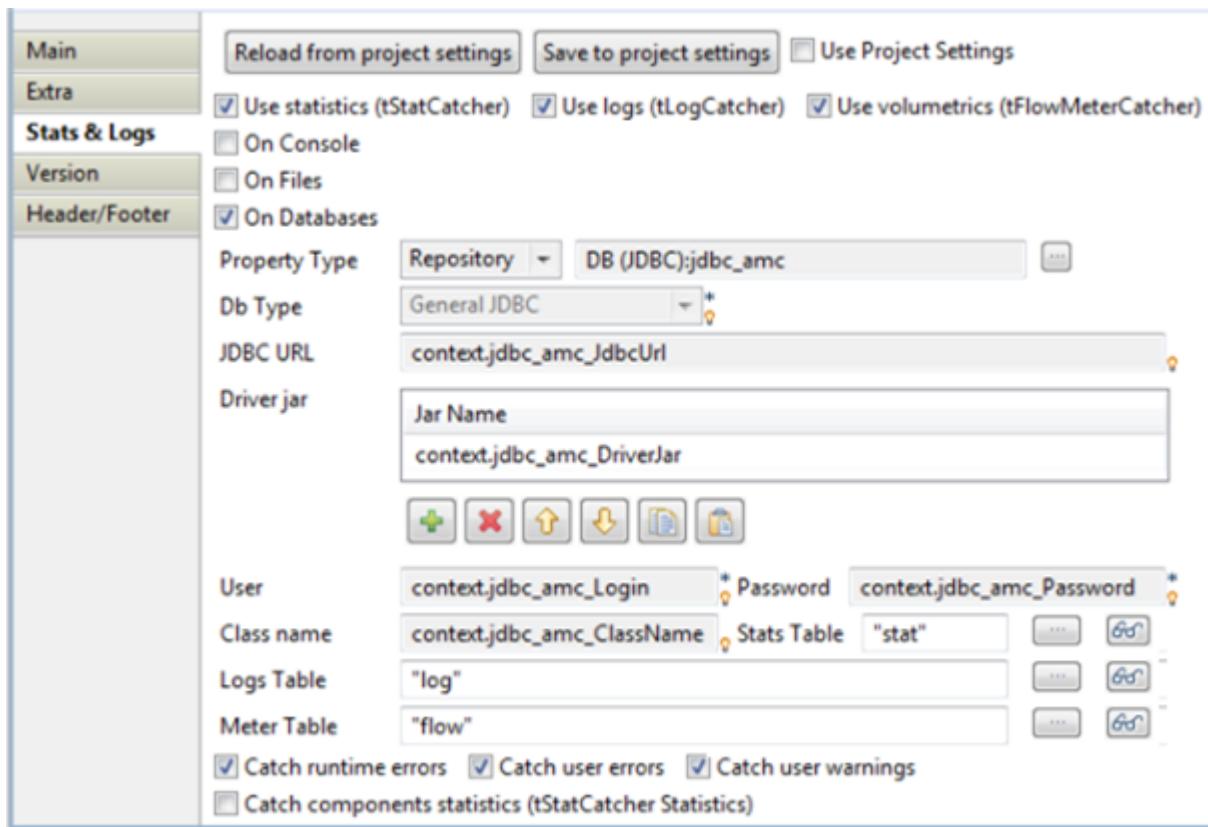
L'onglet **Stats & Logs** est situé dans la vue **Job** sous l'espace de modélisation et vous permet d'éviter de surcharger vos Jobs avec des composants.



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

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

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



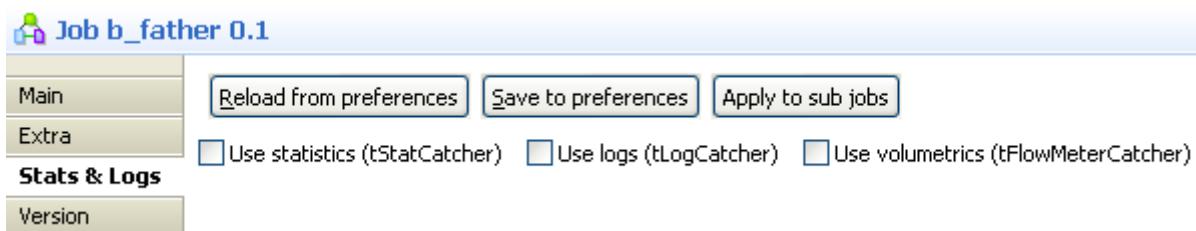
Lorsque vous utilisez une connexion à une base de données, par exemple JDBC, vous devez créer manuellement les tables log, stats et meter.

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



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.

4.6.7.2. 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**.

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

Pour plus d'informations sur les paramètres de contexte, consultez [Section 2.6.6, « Définir les paramètres de contexte »](#).

- 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**.

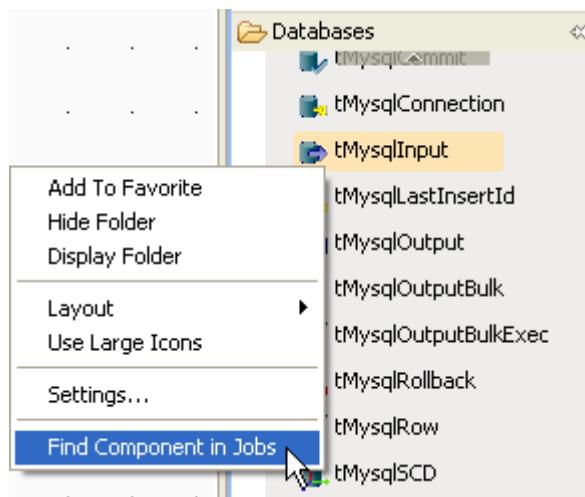
4.6.8. Chercher un Job utilisant un composant particulier



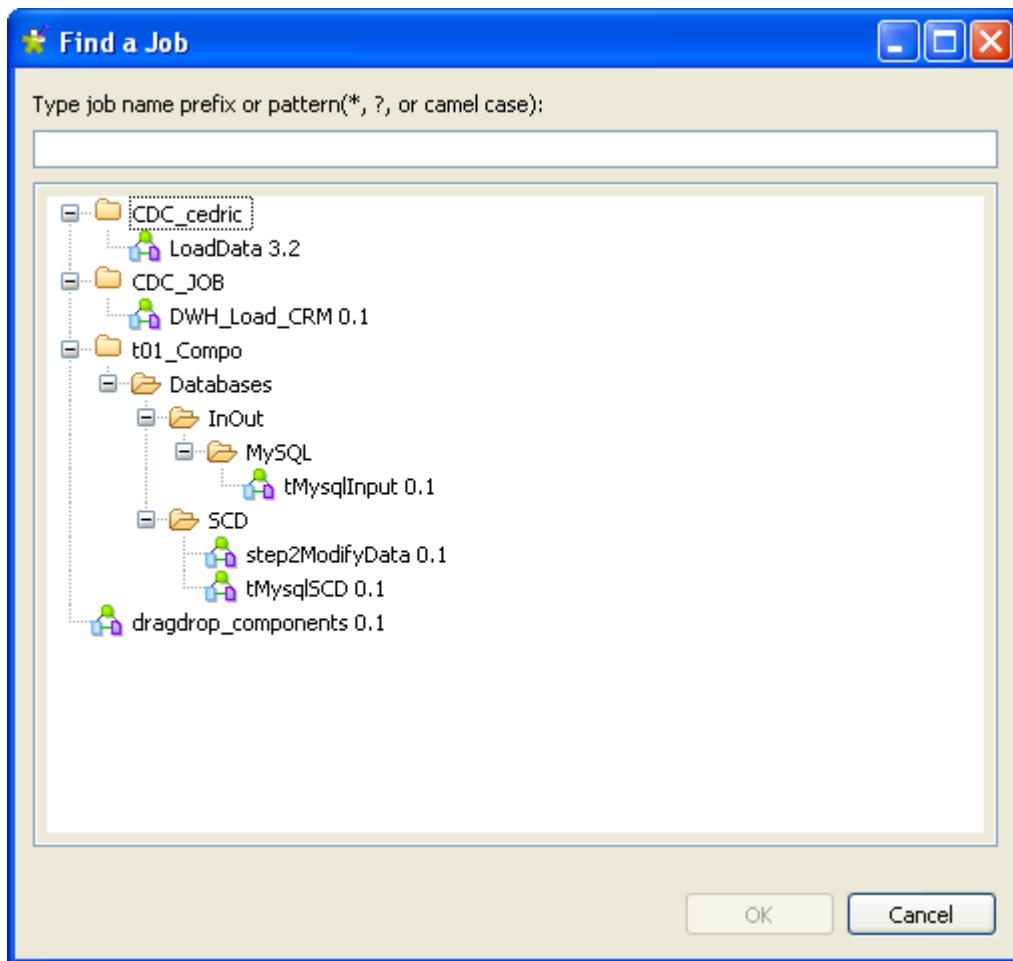
Vous devez ouvrir au moins un Job dans votre Studio afin d'afficher la **Palette** à droite de l'espace de modélisation graphique et lancer la recherche.

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.



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

4.6.9. Configurer les valeurs par défaut dans le schéma d'un composant

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

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

Dans l'exemple suivant, les champs *company* et *city* de certains enregistrements du fichier CSV source sont laissés blancs, comme dans la capture d'écran ci-dessous. Le composant d'entrée lit des données du fichier source et complète les informations manquantes à l'aide des valeurs par défaut dans le schéma, *Talend* et *Paris* respectivement.

```
id;firstName;lastName;company;city;phone
1;Michael;Jackson;IBM;Roma;2323
2;Elisa;Black;Microsoft;London;4499
3;Michael;Dujardin;;;8872
4;Marie;Dolvina;;;6655
5;Jean;Perfide;;;3344
6;Emilie;Taldor;Oracle;Madrid;2266
```

7 ; Anne-Laure ; Paldufier ; Apple ; ; 4422

Pour configurer des valeurs par défaut :

- Double-cliquez sur le composant d'entrée afin d'afficher sa vue **Basic settings**.

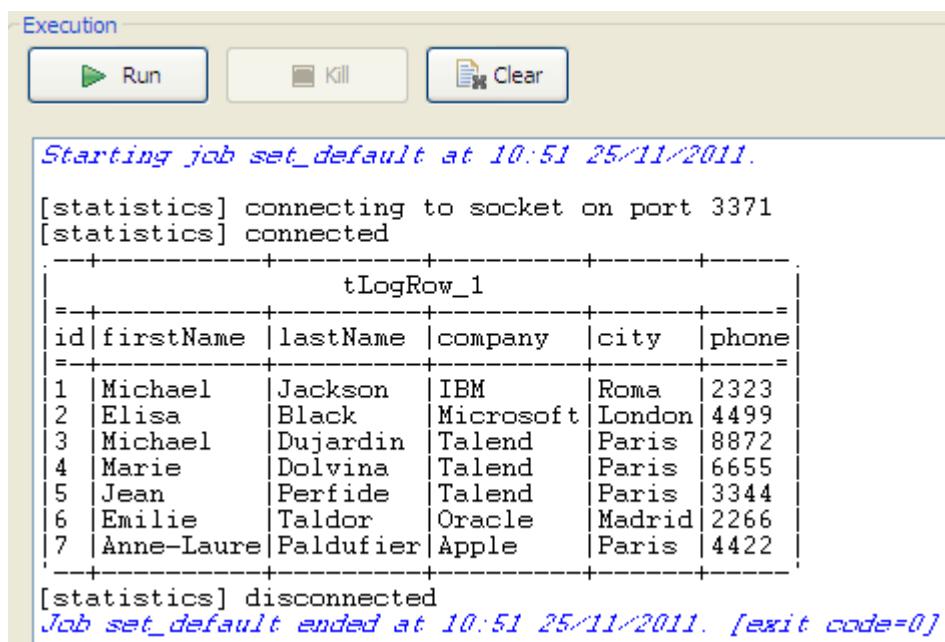
The screenshot shows a Talend job flow with a **tFileInputDelimited_1** component (with ID **tFileInputDelimited_1**) connected to a **tLogRow_1** component. The job is named **Job(set_default 0.1)**. The configuration dialog for **tFileInputDelimited_1** is open, specifically the **Basic settings** tab. The **File name/Stream** field is set to **"D:/Talend/Input/contactInfo.csv"**. In the **Schema** section, the **Repository** dropdown is selected, and the schema is **DELIM:contactInfo - metadata**. A tooltip above the schema dropdown states: "When the input source is a stream or a zip file, footer and random shouldn't be bigger than 0." The **Header** field is set to **1**, and the **Footer** field is set to **0**.

Dans cet exemple, les métadonnées du composant d'entrée sont stockées dans le **Repository**. Pour plus d'informations concernant la création de métadonnées dans le **Repository**, consultez la [Section 4.4.1, « Centraliser les métadonnées »](#).

- Cliquez sur le bouton [...] à côté du champ **Edit schema** et sélectionnez l'option **Change to built-in property** dans la boîte de dialogue afin d'ouvrir l'éditeur du schéma.
- Saisissez *Talend* entre guillemets dans le champ **Default** pour la colonne *company*. Saisissez *Paris* entre guillemets dans le champ **Default** pour la colonne *city* et cliquez sur **OK** pour fermer l'éditeur de schéma.

Column	Key	Type	N..	Dat...	Le...	Pr...	Default	C...
id		Int...	<input checked="" type="checkbox"/>		1	0		
firstName		String	<input checked="" type="checkbox"/>		10	0		
lastName		String	<input checked="" type="checkbox"/>		9	0		
company		String	<input checked="" type="checkbox"/>		9	0	"Talend"	
city		String	<input checked="" type="checkbox"/>		9	0	"Paris"	
phone		Int...	<input checked="" type="checkbox"/>		4	0		

4. Configurez le composant de sortie **tLogRow** pour afficher les résultats d'exécution comme vous le souhaitez et exécutez le Job.



The screenshot shows the Talend Studio Execution view. At the top, there are three buttons: Run (highlighted in blue), Kill, and Clear. Below the buttons, the log output is displayed in a text area:

```
Starting job set_default at 10:51 25/11/2011.  
[statistics] connecting to socket on port 3371  
[statistics] connected  
+-----+-----+-----+-----+  
| id | firstName | lastName | company | city | phone |  
+-----+-----+-----+-----+  
| 1 | Michael | Jackson | IBM | Roma | 2323 |  
| 2 | Elisa | Black | Microsoft | London | 4499 |  
| 3 | Michael | Dujardin | Talend | Paris | 8872 |  
| 4 | Marie | Dolvina | Talend | Paris | 6655 |  
| 5 | Jean | Perfide | Talend | Paris | 3344 |  
| 6 | Emilie | Taldor | Oracle | Madrid | 2266 |  
| 7 | Anne-Laure | Paldufier | Apple | Paris | 4422 |  
+-----+-----+-----+-----+  
[statistics] disconnected  
Job set_default ended at 10:51 25/11/2011. [exit code=0]
```

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

Chapitre 5. Gestion des Jobs d'intégration de données

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

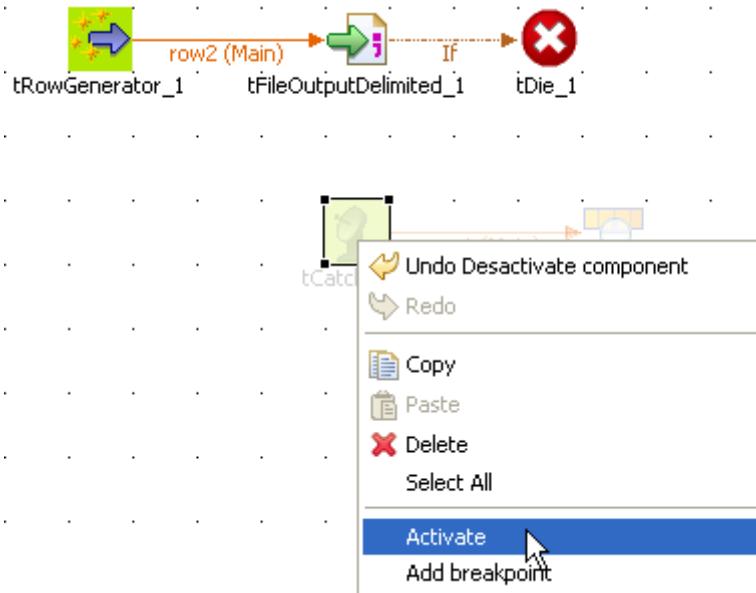
Ces procédures de gestion comprennent l'import et l'export de Jobs et d'éléments entre différents projets ou ordinateurs, ainsi que la programmation de l'exécution et l'exécution de Jobs et leur déploiement sur des serveurs distants, ainsi que la copie des Jobs dans différentes branches SVN.

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez la [Annexe A, Interface de Talend Open Studio for Data Integration](#).

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 : [Section 4.6.2, « Configurer un composant Start ».](#)

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 for Data Integration vous permet d'importer/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 Jobs et ainsi de déployer et d'exécuter les Jobs créés dans le Studio sur n'importe quel serveur.

5.2.1. Import d'éléments

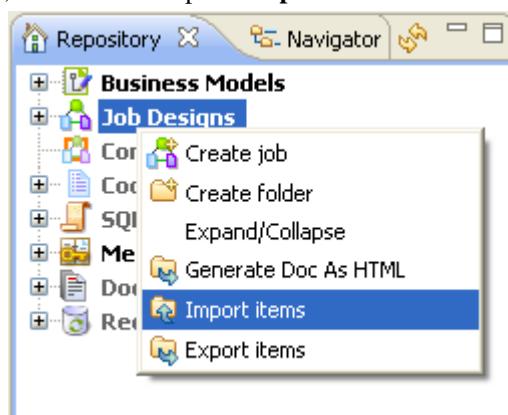
Vous pouvez à présent importer des éléments stockés dans le workspace des versions antérieures de *Talend Open Studio for Data Integration* ou d'un autre projet de votre workspace courant.

Les éléments que vous pouvez importer sont :

- Business Models
- Jobs Designs
- Routines
- Documentation
- Metadata

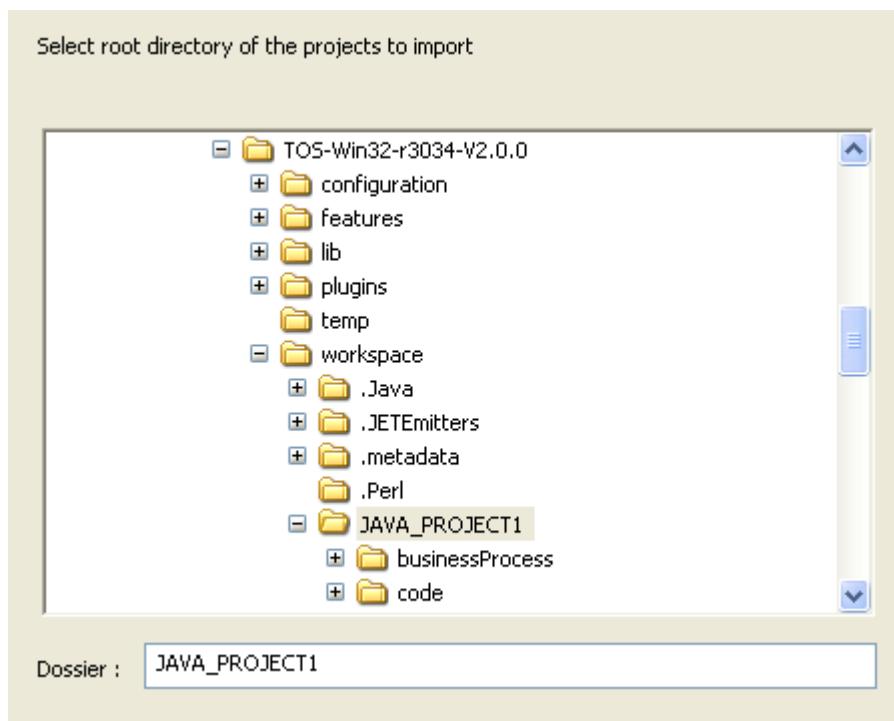
Pour importer des éléments vers un Référentiel, suivez les étapes suivantes :

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

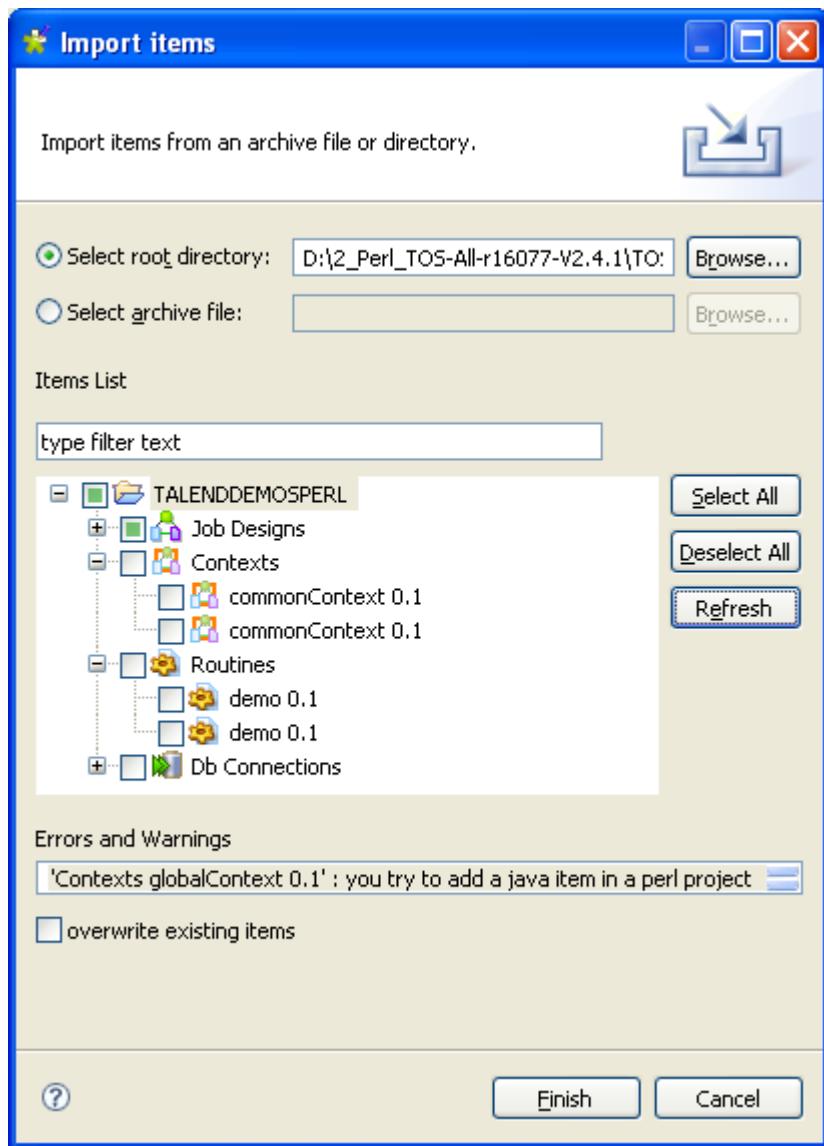


Vous pouvez également cliquer sur l'icône  dans la barre d'outils.

2. Dans la boîte de dialogue qui s'ouvre, sélectionnez 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), utilisez l'option **Select archive file**, sélectionnez le dossier contenant l'archive puis passez à l'étape suivante.
 - Si les éléments à importer sont encore stockés sur le référentiel local, utilisez l'option **Select root directory** et sélectionnez le dossier du projet dans votre système puis passez à l'étape suivante.
3. Sélectionnez le dossier correspondant au **Projet** dans le répertoire **Workspace**. Il correspond au nom du projet que vous créez auparavant.



4. 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**.
5. 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.



6. Cochez la case **overwrite existing items** si vous souhaitez écraser les éléments existants par ceux ayant le même nom et devant être importés. Cela va rafraîchir la liste **Items List**.
7. 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.
8. 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 Jobs vers une archive

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 for Data Integration*.

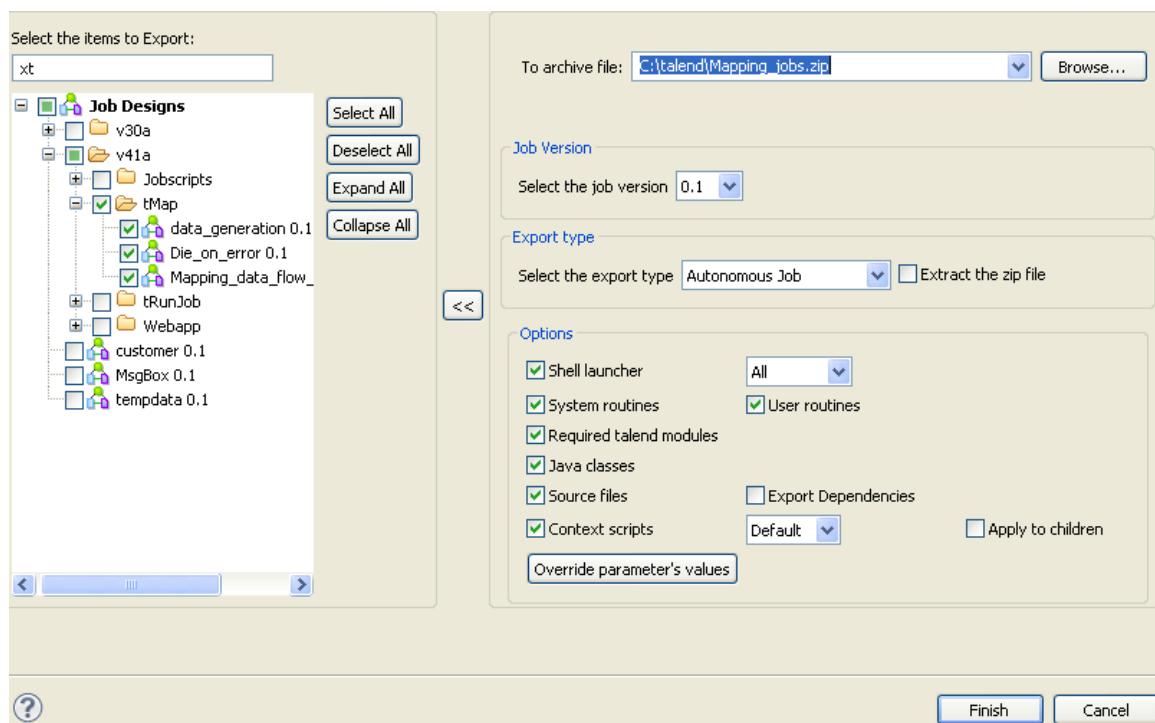
L'export de Job ajoute à l'archive tous les fichiers nécessaires à l'exécution d'un Job, à savoir les fichiers .bat et .sh, ainsi que tous les fichiers de paramètres de contexte ou fichiers liés.

Pour exporter des Jobs, procédez comme suit :

1. Dans l'arborescence **Repository**, cliquez-droit sur le Job que vous voulez exporter et 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 for Data Integration*, 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 for Data Integration*.



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

Option	Description
Shell launcher	Cochez cette case pour exporter les fichiers .bat et/ou .sh nécessaires au lancement du Job exporté. • All : exporte les fichiers .bat et .sh,

Option	Description
	<ul style="list-style-type: none"> Unix : exporte le fichier .sh, Windows : exporte le fichier .bat.
System routines	Cochez cette case pour exporter les routines système.
User routines	Cochez cette case pour exporter les routines utilisateur.
Required Talend modules	Cochez cette case pour exporter les modules Talend .
Java classes	Cochez cette case pour exporter le fichier .java contenant les classes Java générées par le Job lors de sa création.
Source files	Cochez cette case pour exporter les sources utilisés par le Job durant son exécution, notamment les fichiers .item et .properties, ainsi que les sources Java et Talend.  Si vous cochez la case Source files , vous pouvez réutiliser le Job exporté dans un <i>Talend Open Studio for Data Integration</i> installé sur une autre machine. Ces fichiers source sont uniquement utilisés dans <i>Talend Open Studio for Data Integration</i> .
Export Dependencies	Cochez cette case si vous souhaitez exporter les dépendances de votre Job, comme les contextes, les routines, les connexions, etc.
Context script	Cochez cette case pour exporter TOUS les fichiers de paramètres de contexte et pas uniquement ceux sélectionnés dans la liste correspondante.  Pour exporter un contexte, sélectionnez le contexte répondant à vos besoins dans la liste Context script , notamment les fichiers .bat ou .sh contenant les paramètres de contexte appropriés. Vous pouvez ensuite, si vous le souhaitez, éditez les fichiers .bat et .sh pour modifier manuellement le type de contexte.
Apply to children	Cochez cette case si vous souhaitez appliquer le contexte sélectionné dans la liste à tous les Jobs fils.

7. Cliquez sur le bouton **Override parameters' values**, si nécessaire.

Une fenêtre s'ouvre, dans laquelle vous pouvez mettre à jour les paramètres et les valeurs de contexte du Job sélectionné ou ajouter de nouveaux paramètres ou de nouvelles valeurs, selon vos besoins.

8. Cliquez sur **Finish** pour valider les changements effectués, finaliser l'export et fermer la boîte de dialogue.

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



Si le Job à exporter appelle une Routine personnalisée contenant une ou plusieurs classe(s) Java en parallèle avec la classe publique nommée de la même façon que la routine utilisateur, la ou les classe(s) supplémentaire(s) ne sera (seront) pas incluse(s) dans le fichier exporté. Pour exporter ces classes, vous devez les inclure en tant que sous-classes dans la classe ayant le même nom que la routine. Pour plus d'informations concernant les routines utilisateur, consultez [Section 8.4, « Gestion des routines personnalisées »](#). Pour plus d'informations concernant les classes et les sous-classes, consultez les manuels Java correspondants.

5.2.2.1. Exporter un Job en Autonomous Job

Dans le cas d'un export de Plain Old Java Object, si vous souhaitez réutiliser le Job dans *Talend Open Studio for Data Integration* installé sur une autre machine, cochez la case **Source files**. Ces fichiers source (.item et .properties) sont uniquement utilisés dans *Talend Open Studio for Data Integration*.

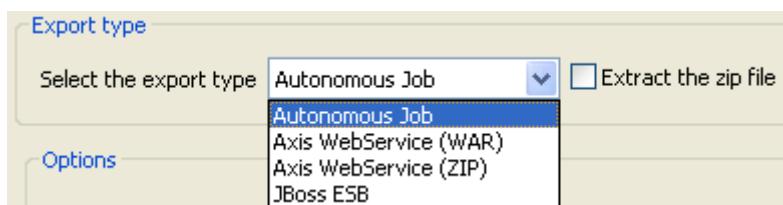
Lorsque que plusieurs contextes sont proposés dans la liste **Context script**, sélectionnez celui qui correspond à vos besoins. Ensuite, cliquez sur le bouton **Override parameters' values**, sous la case **Context script**. Une fenêtre s'ouvre alors et affiche tous les paramètres du contexte sélectionné. 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**.

5.2.2.2. Exporter des Jobs 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
http://localhost:8080/	Saisissez les hôte et port de votre application Web.
/NomAppWeb/	Saisissez le nom de votre application Web.
/services/	Saisissez "services" comme terme d'appel standard pour les services Web.
/NomJob	Saisissez le nom exact du Job que vous souhaitez exécuter.
?method=runJob&args=null	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 la [Section 5.2.2.3, « Exemple d'export de Job en tant que service Web »](#).

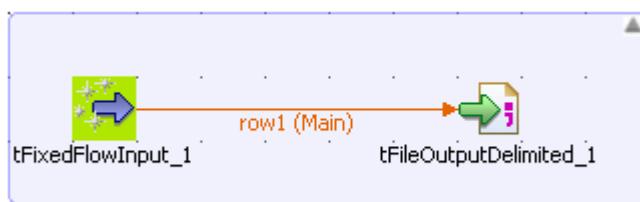
Le composant **tBufferOutput** a été conçu spécialement pour ce type de déploiement. Pour plus d'informations concernant ce composant, consultez le **Guide de référence** des *Composants Talend Open Studio*.

5.2.2.3. Exemple d'export de Job en tant que service Web

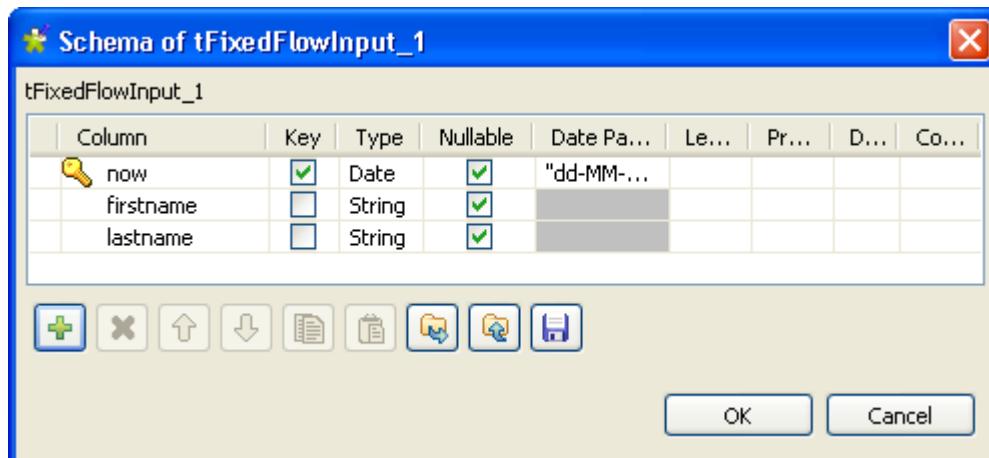
Ce scénario décrit, en premier temps, un simple Job créant un fichier .txt contenant la date en cours, ainsi que des noms et prénoms. En deuxième temps, ce Job est exporté en tant que service Web à partir d'un navigateur Web. Le Job exporté en tant que service Web renverra simplement le "retour d'appel"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

Procédure 5.1. Créer le Job

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



3. Dans l'espace de modélisation, sélectionnez le **tFixedFlowInput** et cliquez sur l'onglet **Component** pour paramétriser les propriétés du **tFixedFlowInput**.
4. 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*.



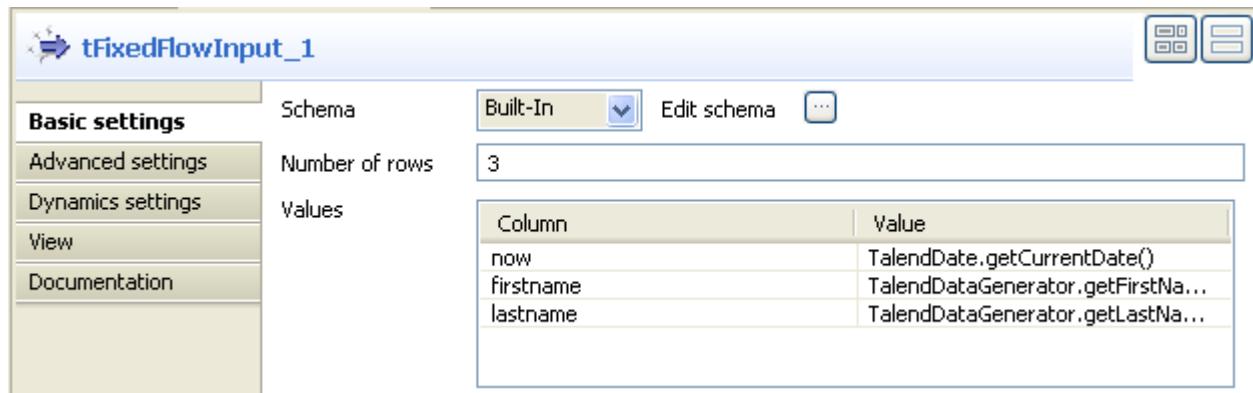
5. Cliquez sur le bouton [+] pour ajouter vos trois lignes de paramètres et définir vos variables. Cliquez sur **OK** pour fermer la boîte de dialogue et acceptez la propagation des modifications lorsque le message de confirmation vous le demande.

Les trois colonnes définies apparaissent dans le tableau **Values** de l'onglet **Basic settings** du composant **tFixedFlowInput**.

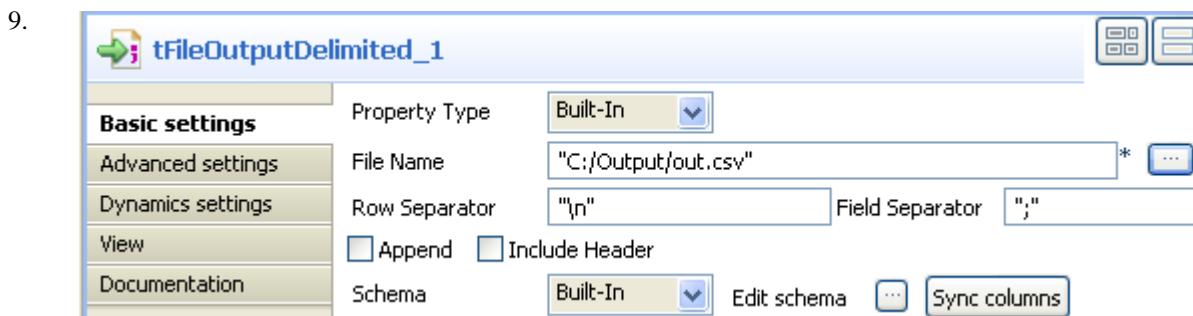
Values	Column	Value
	now	
	firstname	
	lastname	

6. Cliquez sur la cellule **Value** de chaque colonne définie et appuyez sur **Ctrl+Espace** pour accéder à la liste des variables globales et sélectionnez `TalendDate.getCurrentDate()` pour la colonne `now`, `talendDatagenerator.getFirstName` pour la colonne `firstname`, `talendDataGenerator.getLastName` pour la colonne `lastname`.

7. Dans le champ **Number of rows**, saisissez le nombre de lignes à générer.



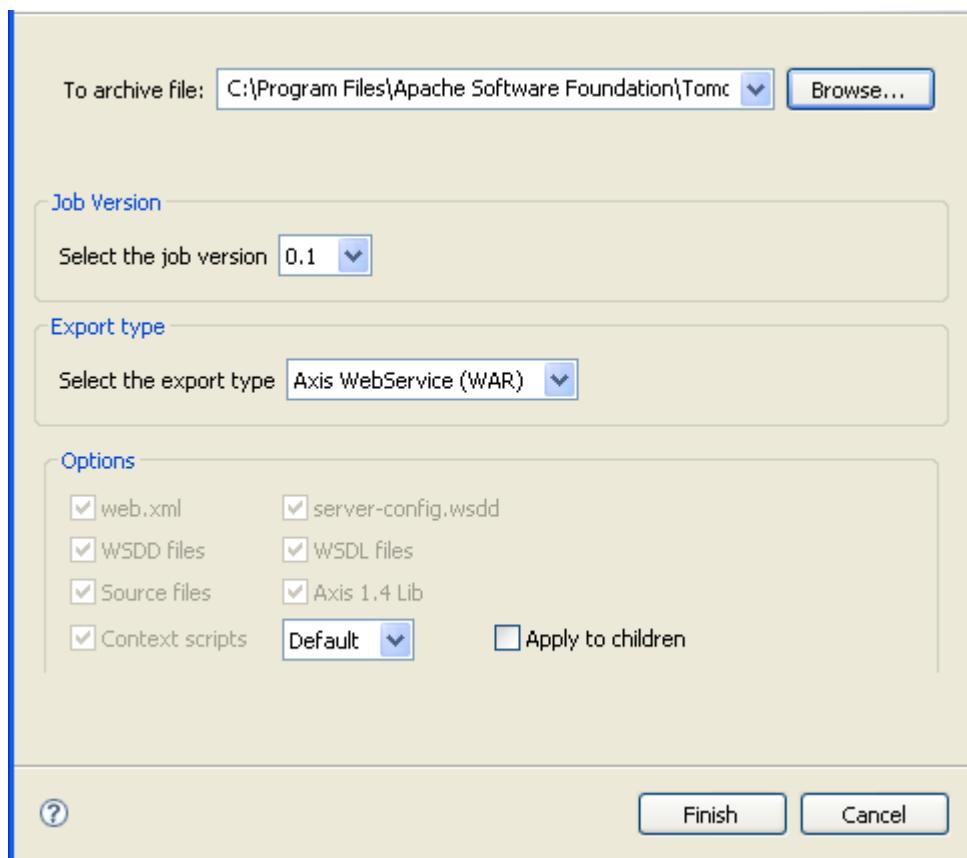
8. Dans l'espace de modélisation, sélectionnez le composant **tFileOutputDelimited**, cliquez sur l'onglet **Component** et, dans le champ **File name**, sélectionnez le fichier de sortie. Paramétrez les autres propriétés du composant, si nécessaire.



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.

Procédure 5.2. Exporter le Job en tant que service Web

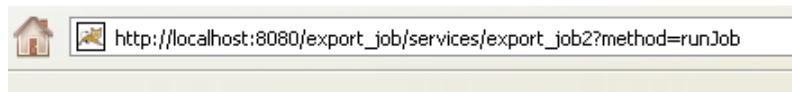
1. Dans la vue **Repository**, cliquez-droit sur le Job que vous venez de créer et sélectionnez **Export Job Scripts**. La boîte de dialogue **Export Job Scripts** apparaît :



2. Dans le champ **To archive file**, cliquez sur le bouton **Browse...** et sélectionnez un répertoire dans lequel enregistrer le fichier archive contenant votre Job.
3. Dans la liste **Job Version**, vous pouvez renseigner la version que vous souhaitez exporter, si vous avez créé plusieurs versions de votre Job.
4. 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.
5. Copiez le dossier War et collez-le dans le répertoire du serveur d'application Tomcat.

Procédure 5.3. Invoquer le Job à partir d'un navigateur Web :

1. 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.



2. 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 **Guide de référence** des *Composants Talend Open Studio*.

5.2.2.4. Exporter un Job en tant que JBoss ESB

Talend Open Studio for Data Integration 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 for Data Integration. Pour plus d'informations concernant la vue Modules, consultez la Section 4.5.4, « Installer des modules externes ».

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 :

1. Dans la zone **Job Version**, sélectionnez la version du Job que vous souhaitez exécuter sur le serveur JBoss ESB.
2. Dans la liste **Select export type** de la zone **Export type**, sélectionnez **JBoss ESB**.
3. 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.
4. 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 for Data Integration* installé sur une autre machine. Ces fichiers source sont utilisés uniquement dans *Talend Open Studio for Data Integration*.*

5. 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.

6. Dans la liste **ESB Export type**, sélectionnez soit *JBoss MQ*, soit *JBoss Messaging*.
7. Dans le champ **Service name**, saisissez le nom du service sur lequel déployer votre Job.
8. Dans le champ **Category**, saisissez le nom de la catégorie du service sur lequel le Job sera déployé.
9. Dans le champ **Message Queue Name**, saisissez le nom de la queue utilisée pour le déploiement du Job.
10. 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.

5.2.2.5. Exporter un Job en tant que Petals ESB

Avec *Talend Open Studio for Data Integration*, 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> [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 for Data Integration* dans Petals ESB. Pour plus d'informations, consultez la [Section 5.2.2.5, « Exporter un Job en tant que Petals ESB »](#). 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 for Data Integration* 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.

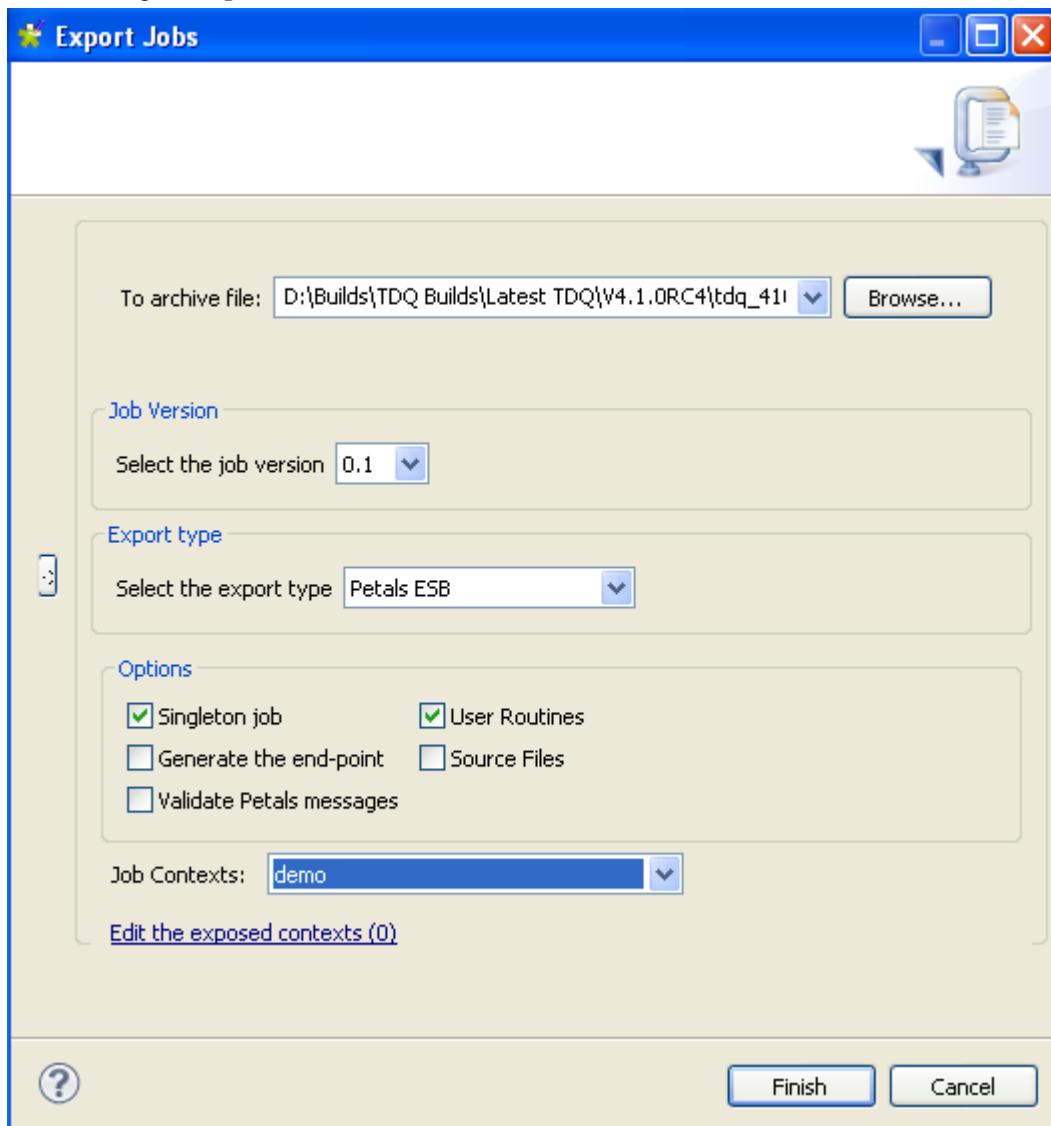
Comment exporter des Jobs vers Petals ESB

Dans la boîte de dialogue [Export Jobs], 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 :

1. Dans la vue **Repository**, cliquez-droit sur le Job que vous souhaitez exporter, puis sélectionnez **Export Job** dans le menu contextuel.

La boîte de dialogue [Export Jobs] s'ouvre.



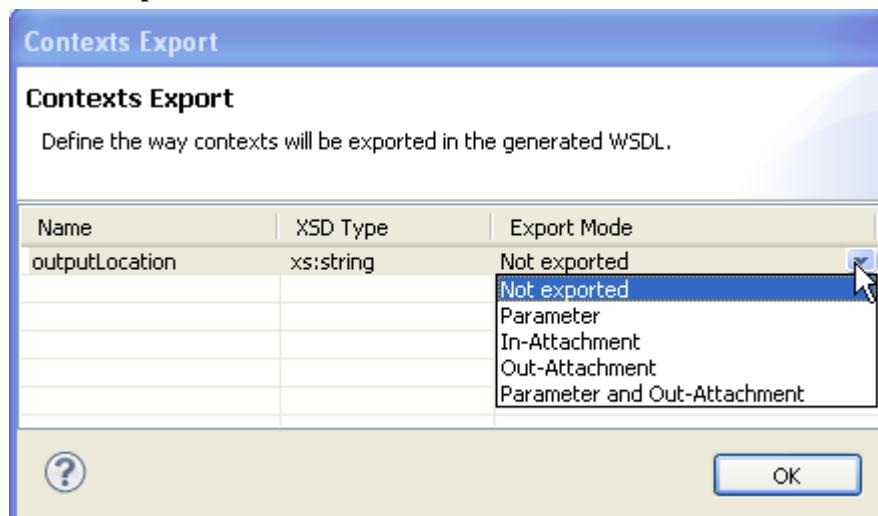
2. Dans le champ **To archive file**, parcourez votre répertoire afin de définir le chemin d'accès à l'archive dans laquelle vous souhaitez exporter le script.
3. Dans la liste **Select the job version**, sélectionnez la version du Job que vous voulez exporter.
4. Dans la liste **Select export type**, 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.
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 End-point .
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.
Out-Attachment	Le contexte sera lu après exécution du Job.

Export Mode	Description
	<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.

7. Cliquez sur **OK** afin de valider votre choix et fermer la boîte de dialogue [**Context Export**].

8. Dans la boîte de dialogue [**Export Jobs**], cliquez sur **Finish**.

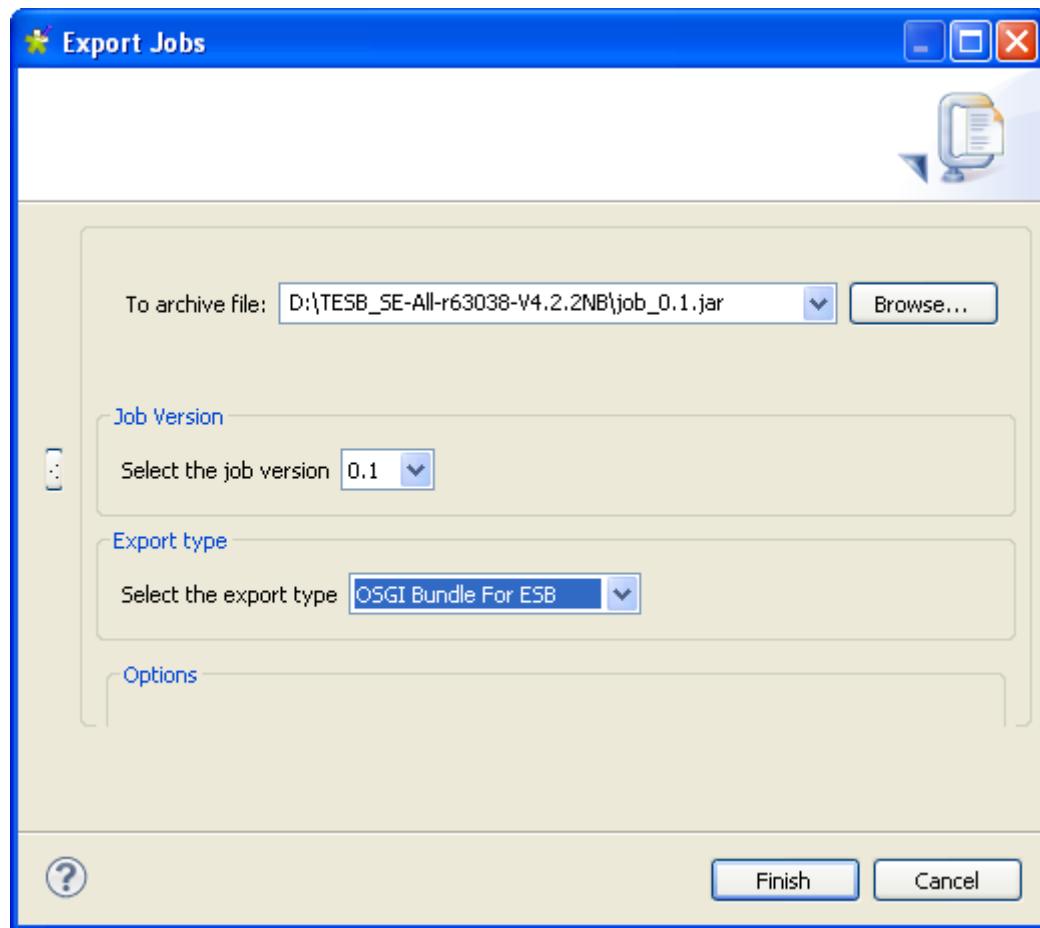
La boîte de dialogue se ferme. Un indicateur de progression s'affiche, afin de montrer le pourcentage d'avancement de l'export.

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.

5.2.2.6. Exporter des Jobs en tant que bundle OSGI pour ESB

Dans la boîte de dialogue [**Export Jobs**], vous pouvez changer le type d'export pour exporter le Job sélectionné en tant que bundle OSGI, afin de déployer votre Job dans le conteneur OSGI de Talend ESB.



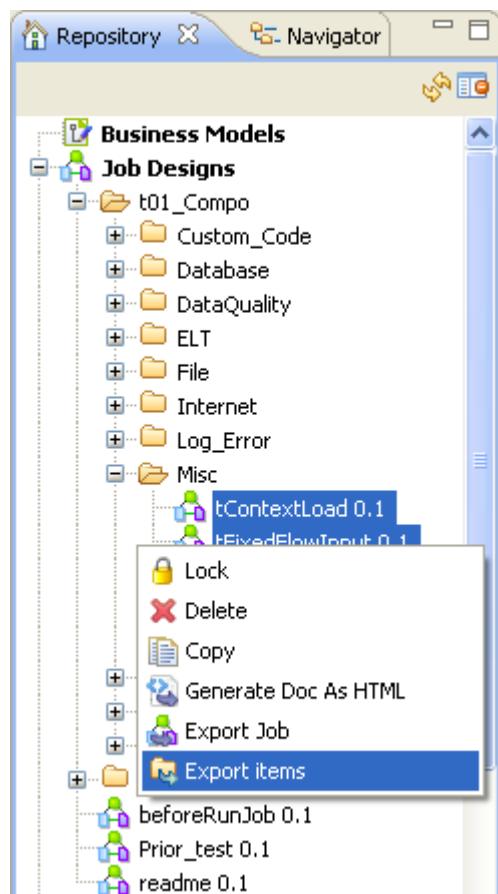
1. Dans la zone **Job Version**, sélectionnez le numéro de version du Job que vous souhaitez exporter si vous en avez créé plus d'une version.
2. Dans la zone **Export Type**, sélectionnez **OSGI Bundle For ESB** afin d'exporter votre Job en tant que bundle OSGI.
L'extension de votre export change automatiquement en *.jar* puisque c'est l'extension attendue par le conteneur OSGI de **Talend ESB**.
3. Cliquez sur le bouton **Browse...** pour spécifier le dossier dans lequel exporter votre Job.
4. Cliquez sur **Finish** afin de l'exporter.

5.2.3. 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.

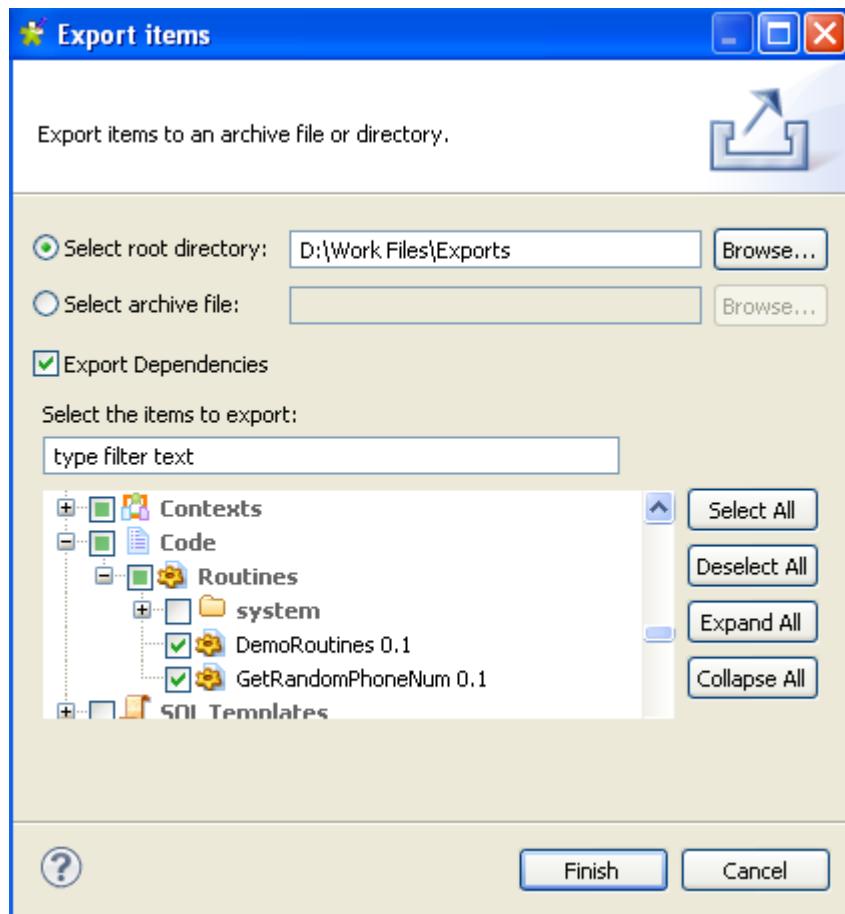
Pour ce faire :

1. Dans le **Repository**, sélectionnez les éléments que vous souhaitez exporter
2. 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.

3. Maintenez la touche **Ctrl** enfoncée et cliquez-droit sur l'élément à exporter, puis sélectionnez **Export items** dans le menu contextuel :



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

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



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



Cochez la case **Export Dependencies** si vous souhaitez définir et exporter les dépendances des routines avec les Jobs que vous exportez. Par défaut, toutes les routines utilisateur sont sélectionnées. Pour plus d'informations, consultez [Section 8.1, « Définition des routines »](#)

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

5.2.4. Changer les paramètres de contexte d'un Job script

Comme expliqué dans les sections [Section 5.2.2, « Export de Jobs vers une archive »](#), vous pouvez éditer les paramètres de contexte :

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

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

Pour...	Paramètre
Changer la <i>valeur1</i> pour le paramètre <i>key1</i>	--context_param key1=valeur1
Changer <i>valeur1</i> et <i>valeur2</i> des paramètres respectifs <i>key1</i> et <i>key2</i>	--context_param key1=valeur1 --context_param key2=valeur2
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 for Data Integration 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.

5.3.1.1. Modifier un élément du Repository

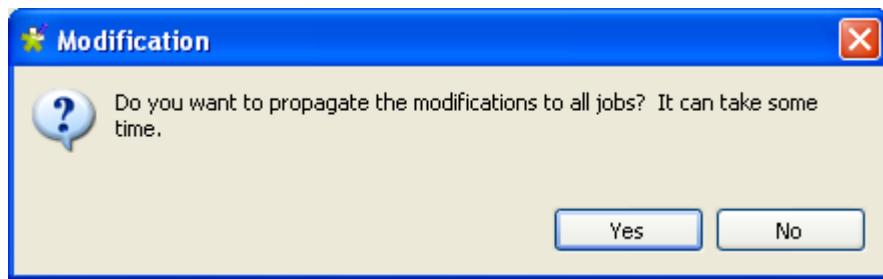
Pour actualiser les paramètres d'un élément du Repository, procédez comme suit :

- Développez le nœud **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 la [Section 5.3.1.3, « Mettre à jour manuellement les Jobs impactés »](#).

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 la [Section 5.3.1.3, « Mettre à jour manuellement les Jobs impactés »](#).

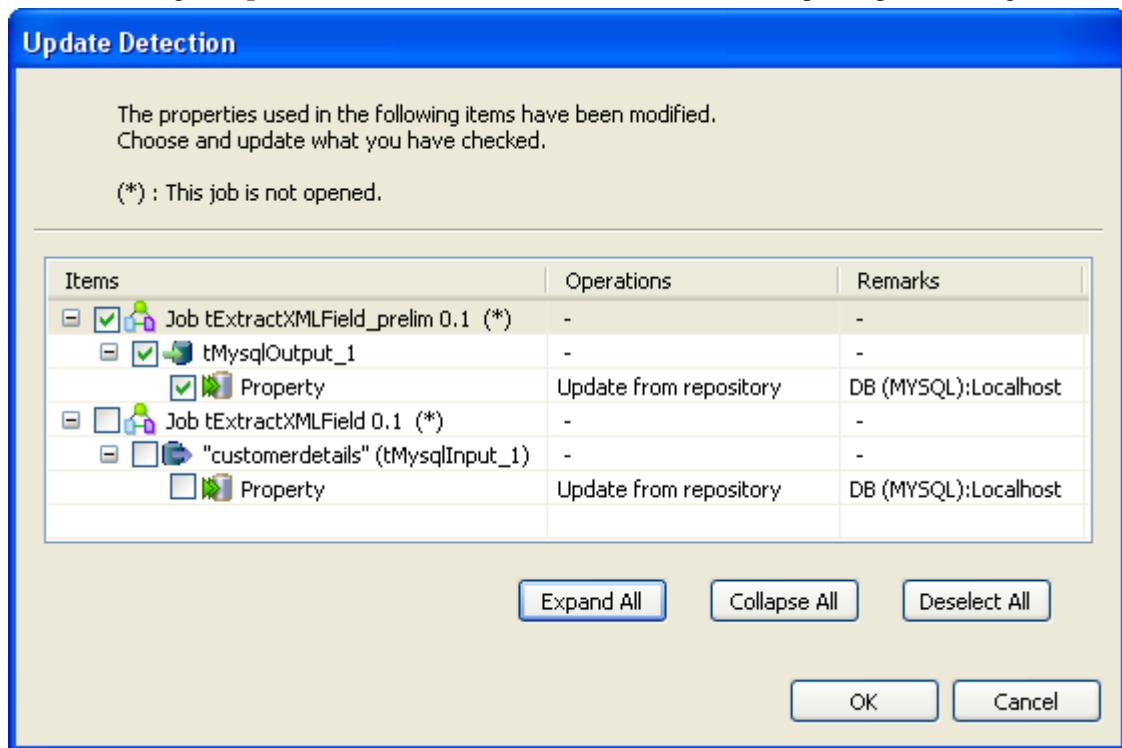
5.3.1.2. Mettre à jour automatiquement les Jobs impactés

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.

Pour mettre à jour les Jobs impactés, procédez comme suit :

- 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.





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 la [Section 5.3.1.3, « Mettre à jour manuellement les Jobs impactés »](#).

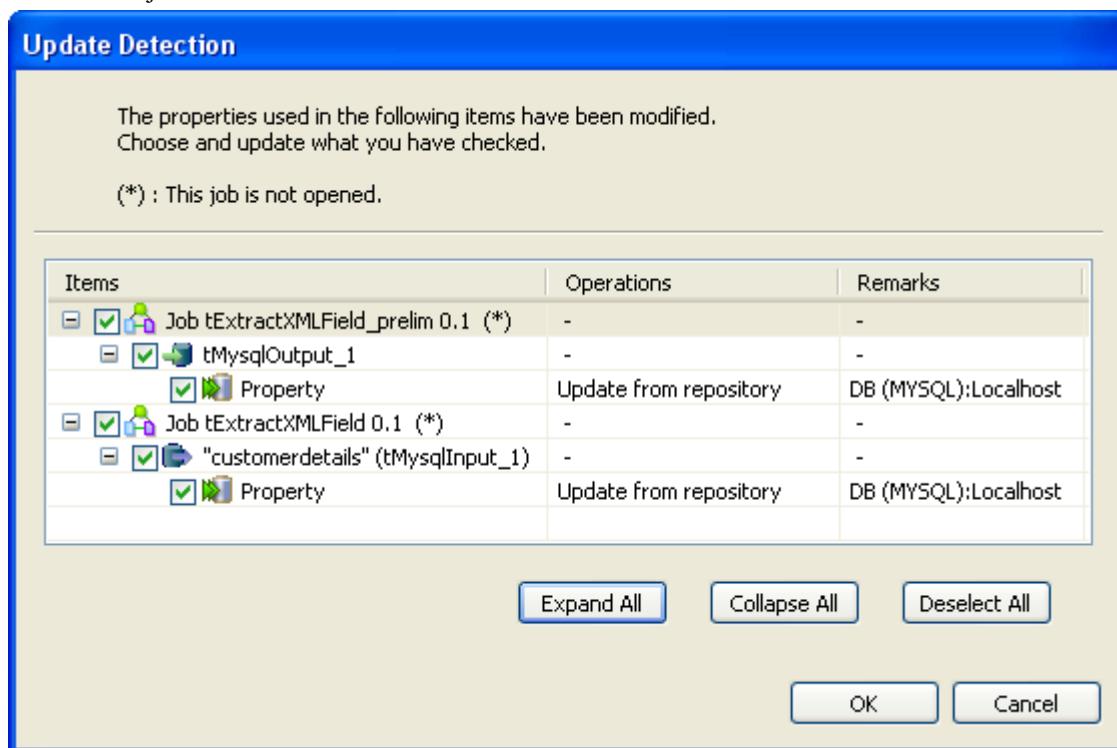
2. Décochez la case correspondant aux Jobs que vous ne souhaitez pas mettre à jour. Vous pouvez les mettre à jour manuellement ultérieurement. Pour plus d'informations, consultez la [Section 5.3.1.3, « Mettre à jour manuellement les Jobs impactés »](#).
3. Cliquez sur **OK** pour fermer la boîte de dialogue et mettre à jour les Jobs sélectionnés.

5.3.1.3. 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 :

1. Dans la vue **Repository**, développez le nœud contenant l'entrée dans laquelle vous souhaitez vérifier les Jobs l'utilisant.
2. Puis cliquez-droit sur cette métadonnée ou ce contexte et sélectionnez l'option **Detect Dependencies** dans le menu contextuel.

Une barre de progression indique le processus de vérification de tous les Jobs utilisant les paramètres de métadonnée ou de contexte modifiés. Une boîte de dialogue s'ouvre et affiche tous les Jobs utilisant cet élément mis à jour.



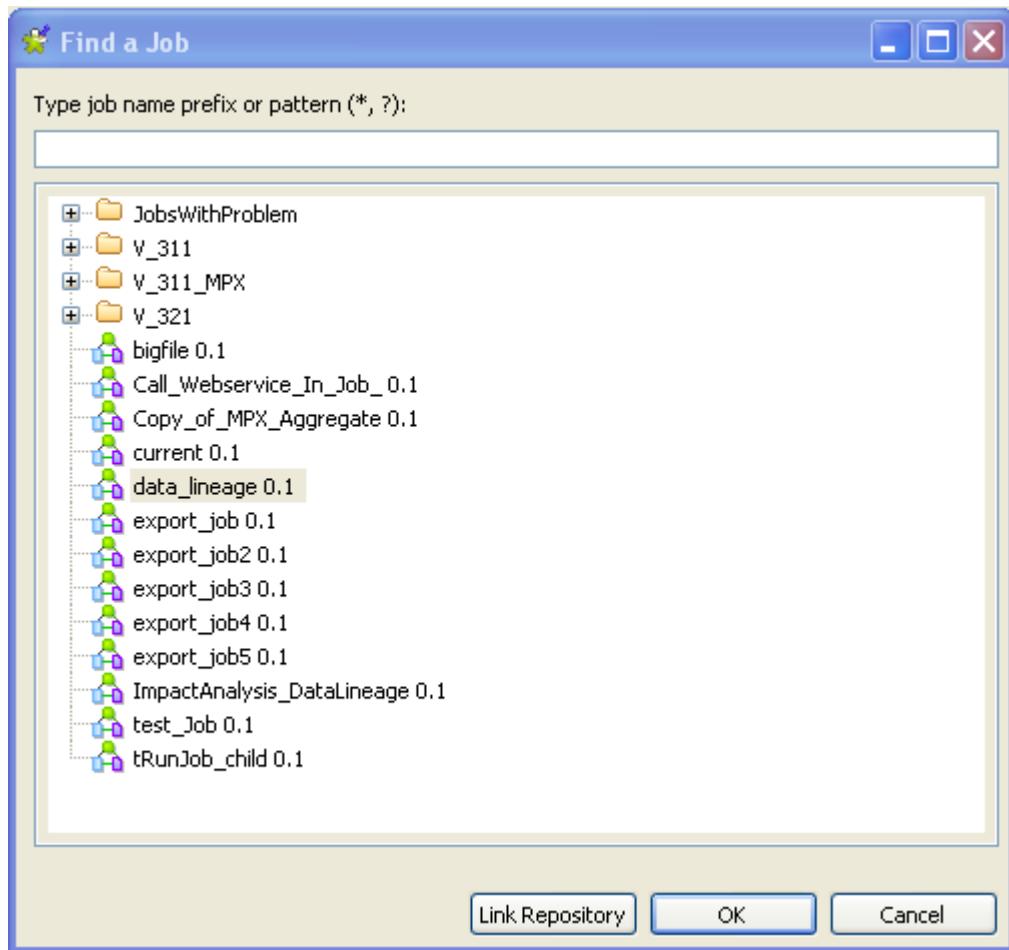
3. Cochez les cases des Jobs à mettre à jour avec les métadonnées ou les paramètres de contexte modifiés et décochez les cases des Jobs que vous ne souhaitez pas mettre à jour.
 4. Cliquez sur **OK** pour valider et fermer la boîte de dialogue.
-  Les Jobs que vous n'aurez pas mis à jour passeront automatiquement 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 for Data Integration* et que vous n'arrivez pas à le retrouver dans la vue **Repository**, cliquez simplement sur le bouton  de la barre d'outils.

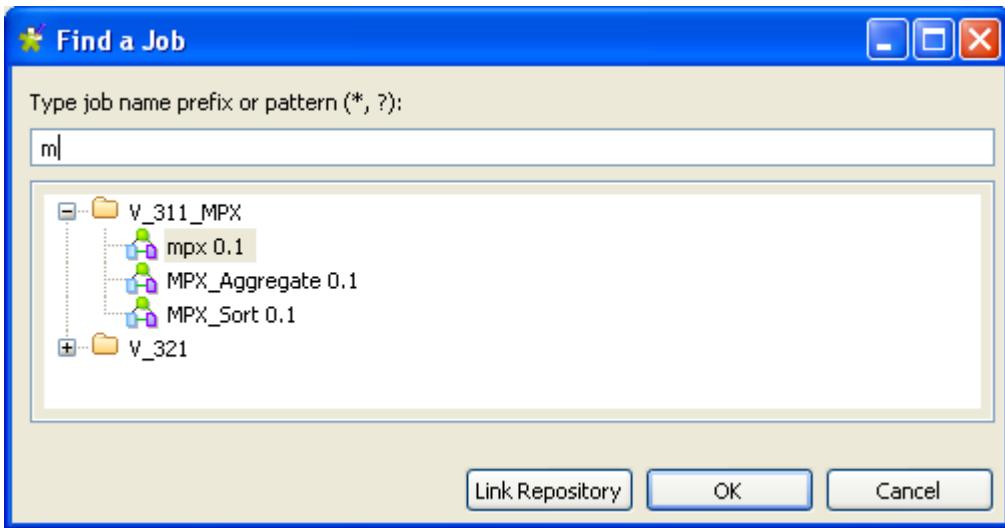
Pour chercher un Job dans la vue **Repository** :

1. Dans la barre d'outils de *Talend Open Studio for Data Integration*, cliquez sur le bouton  pour ouvrir la boîte de dialogue **[Find a Job]** listant automatiquement tous les Jobs créés dans le Studio courant.



2. Saisissez le nom ou une partie du nom du Job dans le champ en haut de la fenêtre.

Lorsque vous commencez à saisir le nom dans le champ, la liste des Jobs est automatiquement mise à jour afin de n'afficher que le ou les Jobs dont le nom correspond aux lettres saisies.



3. Sélectionnez le Job souhaité dans la liste et cliquez sur **Link Repository** pour afficher automatiquement le Job sélectionné dans l'arborescence de la vue **Repository**.
 4. Si nécessaire, cliquez sur **Cancel** pour fermer la boîte de dialogue et cliquez-droit sur le Job sélectionné dans le **Repository** pour effectuer l'une des opérations disponibles dans le menu contextuel.
- Si non, 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 for Data Integration*, 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 :

1. Veillez à fermer votre Job, sinon ses propriétés seront en lecture seule et vous ne pourrez donc pas les modifier.
2. 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.
3. Dans le champ **Version**, cliquez sur le bouton **[M]** pour incrémenter la version majeure et sur le bouton **[m]** pour incrémenter la version mineure.
4. Cliquez sur **Finish** pour valider.



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 :

1. 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.
2. Dans la vue **Repository**, cliquez-droit sur votre Job et sélectionnez **Open another version** dans la liste déroulante.
3. 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.

4. 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 :

1. Dans la vue **Repository**, sélectionnez le Job dont vous souhaitez consulter les versions.
2. Cliquez sur **Job > Version** successivement pour afficher la liste des versions du Job sélectionné .
3. Cliquez-droit sur la version que vous souhaitez consulter.
4. 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 [Section 2.6.2, « Gérer les versions »](#).

5.6. Documentation du Job

Talend Open Studio for Data Integration 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.

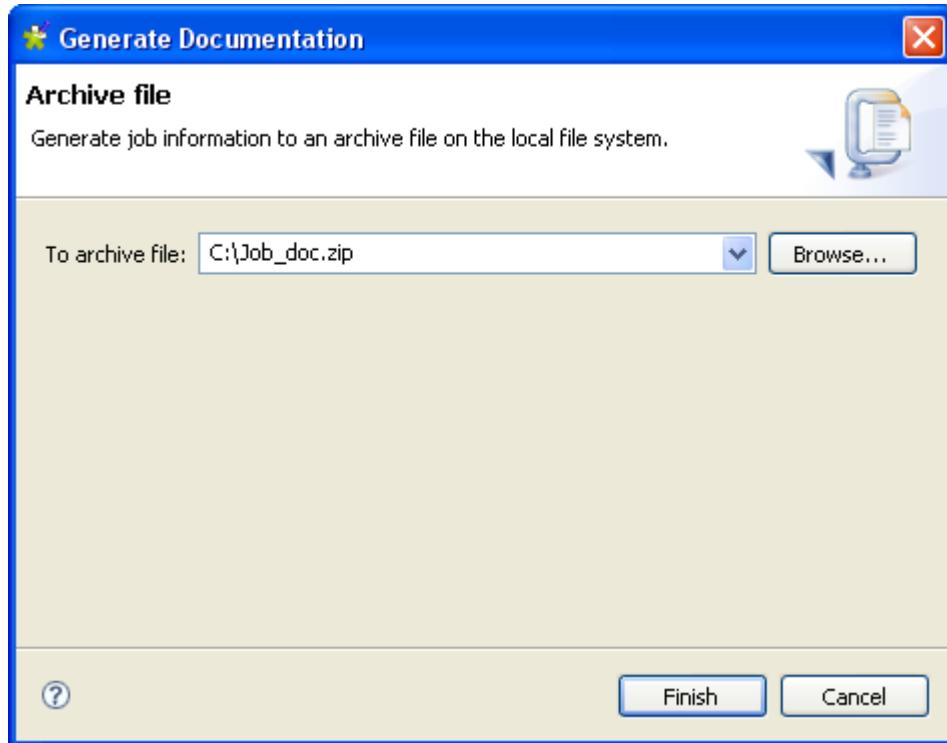
5.6.1. Générer la documentation HTML

Talend Open Studio for Data Integration 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 :

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



3. Renseignez le répertoire dans lequel doit être enregistré la documentation générée.
4. Dans le même champ, renseignez le nom de l'archive regroupant les documents générés.
5. 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

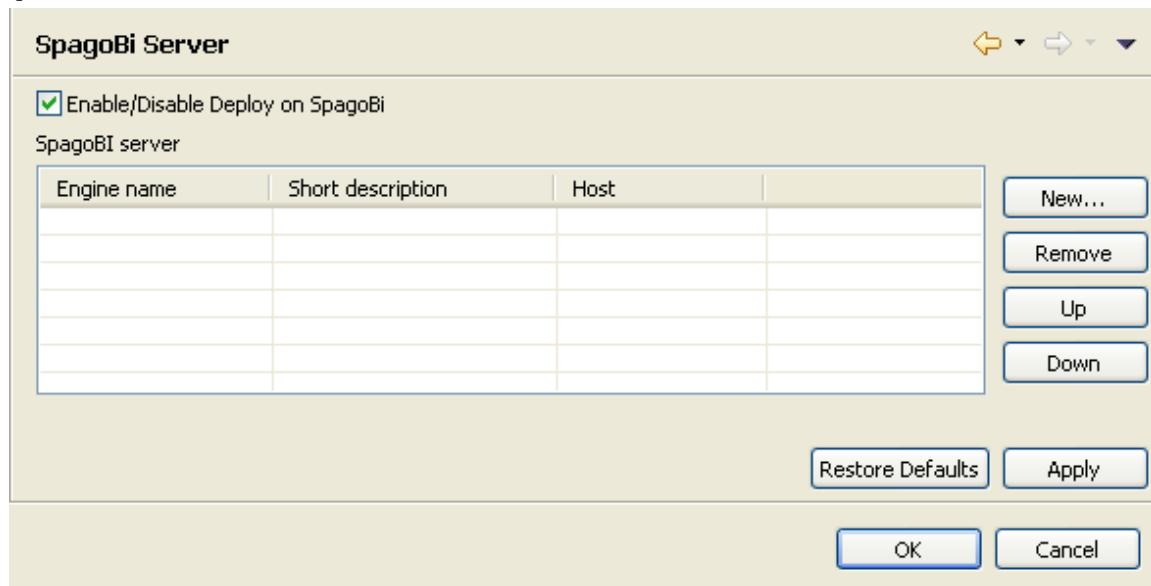
5.7.1. Déploiement d'un Job sur un serveur SpagoBI

A partir de l'interface *Talend Open Studio for Data Integration*, vous pouvez facilement déployer vos Jobs sur un serveur afin de les exécuter à partir de votre administrateur.

5.7.1.1. Créez une nouvelle connexion au serveur SpagoBI

Avant toute chose, renseignez les informations concernant votre serveur simple ou multiple dans *Talend Open Studio for Data Integration*.

1. Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **[Preferences]**.
2. Développez les nœuds **Talend > Import/Export** et sélectionnez **SpagoBI server** pour afficher la vue adéquate.



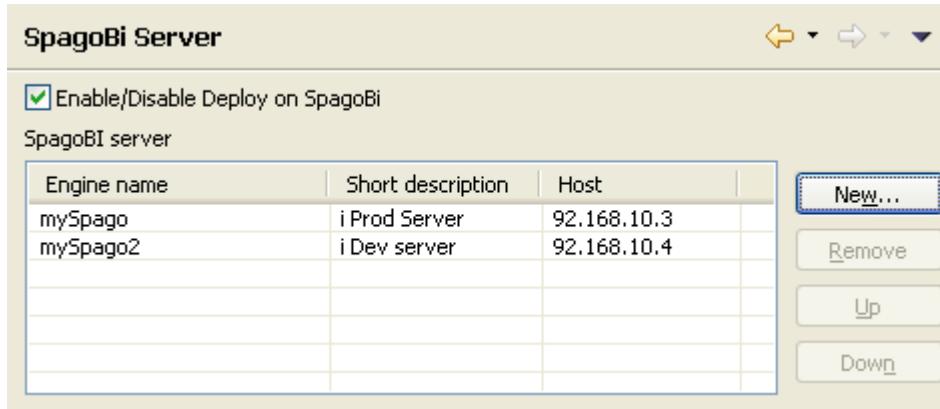
3. Cochez la case **Enable/Disable Deploy on SpagoBI** pour activer le déploiement.
4. Cliquez sur **New** pour ajouter un nouveau serveur à la liste des serveurs.



5. Renseignez les informations du serveur SpagoBI, comme décrit ci-dessous :

Champ	Description
Engine Name	Saisissez le nom interne utilisé dans <i>Talend Open Studio for Data Integration</i> . Ce nom n'est pas utilisé dans le code généré.
Short description	Saisissez une description du serveur que vous êtes en train de créer.
Host	Adresse IP ou nom de l'hôte de la machine sur laquelle SpagoBI fonctionne.
Login	Identifiant requis pour vous connecter au serveur SpagoBI.
Password	Mot de passe pour vous connecter au serveur SpagoBI.

6. Cliquez sur **OK** pour valider ces informations et fermer la boîte de dialogue.



La nouvelle entrée est ajoutée au tableau des serveurs disponibles. Vous pouvez ajouter autant d'entrée que vous le souhaitez.

7. Cliquez sur **Apply** et **OK** pour fermer la boîte de dialogue [**Preferences**].

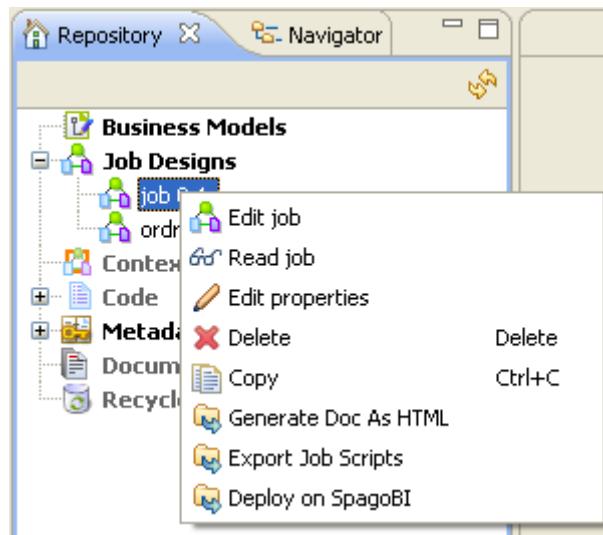
5.7.1.2. Editer ou supprimer un serveur

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.

5.7.1.3. Déployer vos Jobs sur un serveur SpagoBI

Suivez les étapes suivantes pour déployer vos Jobs sur un serveur :



1. A partir du **Repository**, développez le nœud **Job Designs** puis cliquez-droit sur le Job à déployer.
2. Dans la liste déroulante, sélectionnez **Deploy on SpagoBI**.
3. Comme pour tout export de Job script, sélectionnez le nom (**Name**) du Job qui sera exporté et renseignez le champ **To archive file**.
4. Dans le champ **SpagoBI server**, sélectionnez le serveur correspondant dans la liste déroulante.

5. Les champs **Label**, **Name** et **Description** seront pré-remplis avec les propriétés principales définies lors de la création du job.
6. Sélectionnez le contexte adéquat dans la liste.
7. Cliquez sur **OK** lorsque toutes les informations sont renseignées.

Les Jobs sont maintenant déployés sur le serveur SpagoBI sélectionné. Ouvrez votre administrator SpagoBI pour exécuter vos Jobs.

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 les composants **tMap** et **tXMLMap**.

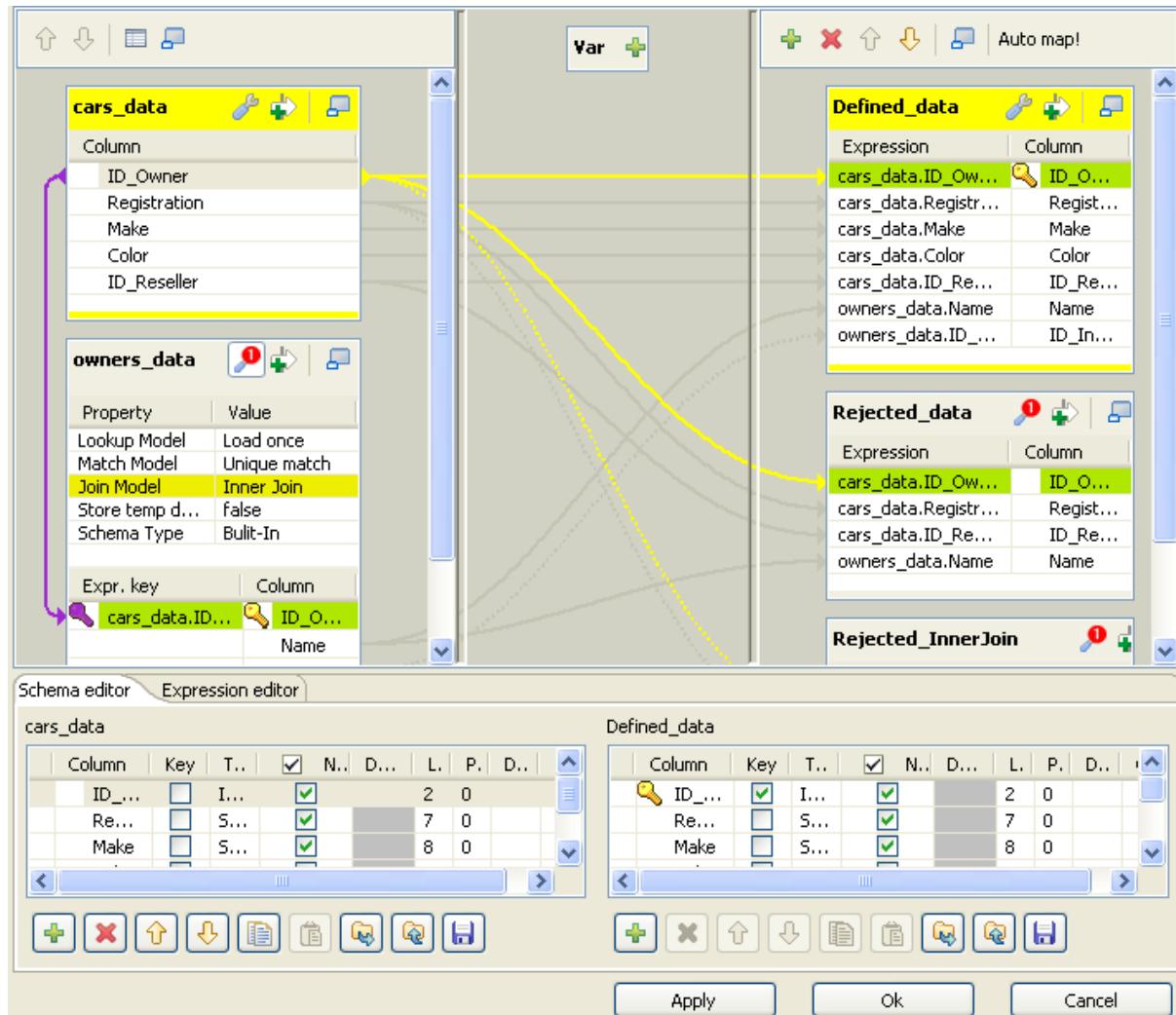
Ce chapitre fournit séparément les informations concernant les principes d'usage de ces composants. Pour plus d'informations, ou pour des cas d'utilisation, consultez le **Guide de Référence** des *Composants Talend Open Studio*.

Avant de commencer un processus d'intégration de données, vous devez vous être familiarisé avec l'interface graphique de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez l'[Annexe A, Interface de Talend Open Studio for Data Integration](#) .

6.1. Interface du tMap et du tXMLMap

Le **tMap** et le **tXMLMap** sont des composants “avancés” qui requièrent 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.



💡 La capture d’écran présente l’interface du **tMap**. Celle du **tXMLMap** a une apparence légèrement différente. Par exemple, en plus des onglets **Schema editor** et **Expression editor** dans la partie inférieure de l’interface, le **tXMLMap** contient un troisième onglet nommé **Tree schema editor**. Pour plus d’informations concernant le composant **tXMLMap**, consultez la [Section 6.3, « Présentation du fonctionnement du tXMLMap »](#).

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 nom des schémas Input/Output dans le **Map Editor** reflète le nom des connexions entrantes et sortantes (connexion de type **Row**).

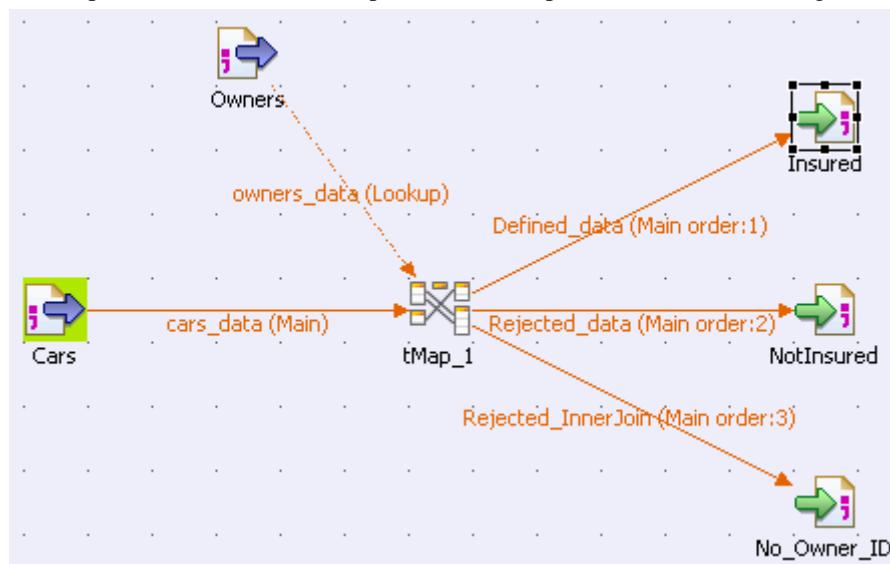
Les sections suivantes présentent séparément le **tMap** et le **tXMLMap**.

6.2. 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 de début ni un composant de fin de process dans un Job Design.



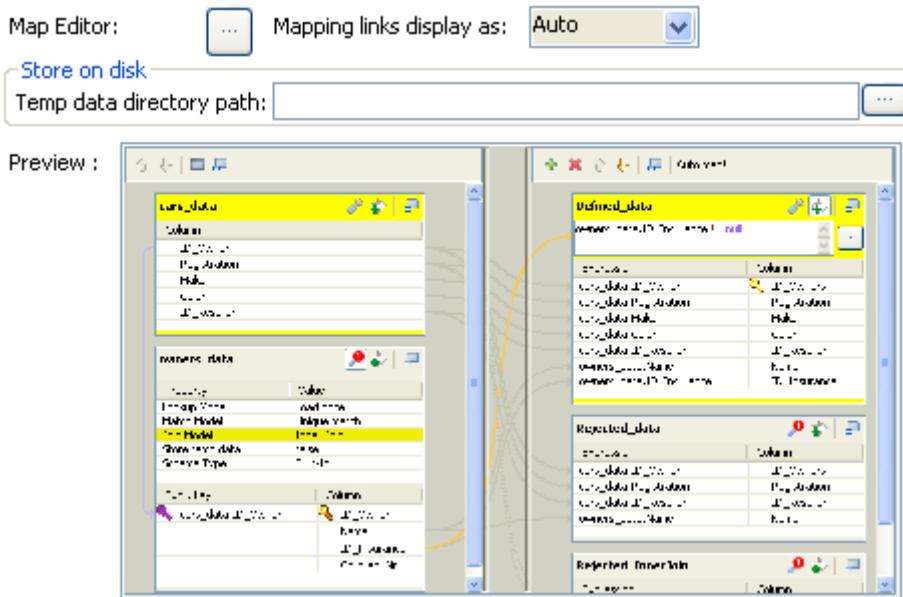
Le **tMap** utilise les connexions entrantes pour pré-remplir les schémas d'entrée dans le **[Map Editor]**. Par conséquent, vous ne pouvez pas créer directement dans le **[Map Editor]**, de nouveaux schémas d'entrée. Par contre, il vous faut mettre en place autant de connexions de type **Row** entrantes que nécessaire, pour que leur schéma s'affiche dans le **[Map Editor]**.

De la même façon, créez autant de connexions de sortie que nécessaire dans l'espace de modélisation. Cependant, vous pouvez définir le type de données en sortie directement dans le **[Map Editor]** via un outil graphique de mapping.

Il ne peut y avoir qu'une seule connexion de type **Main** row. Toute connexion entrante supplémentaire devient automatiquement une connexion **Lookup**. Voir : [Section 4.3.1.1, « Connexion de type Row »](#).

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 une jointure (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 la vue **Component** de l'espace de modélisation.



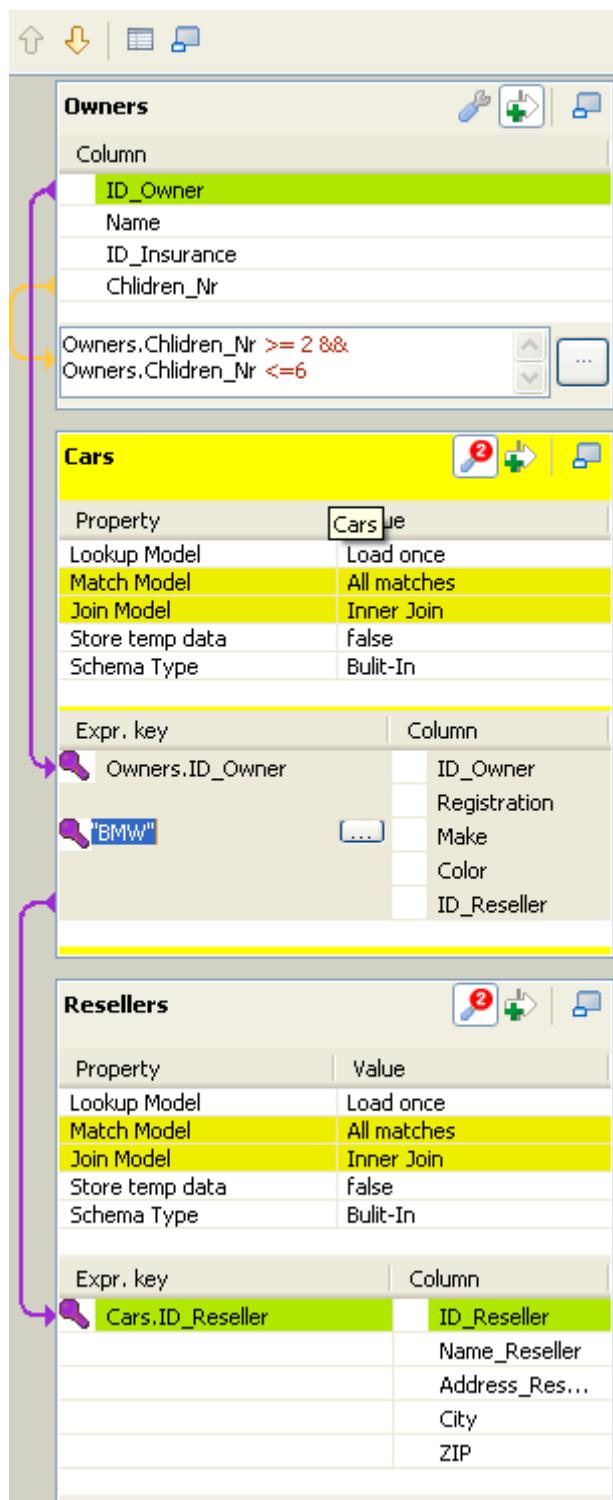
Double-cliquez sur l'icône **tMap** dans l'espace de modélisation ou cliquez sur le bouton [...] à côté du champ **Map Editor** de l'onglet **Basic settings**, dans la vue **Component** du composant **tMap**. L'éditeur de mapping s'ouvre dans une nouvelle fenêtre.

Les sections suivantes vous donnent les informations nécessaires à l'utilisation du composant **tMap** dans vos Job Designs.

6.2.1. Configuration du flux d'entrée dans le Map Editor

L'ordre des tables **Input** (ou schémas) est essentiel. La première table reflète la connexion de flux principal (**Main** row), et pour cette raison, est traitée en priorité dans le composant **tMap**.

Par conséquent, vous ne pouvez pas déplacer 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 les tables secondaires (provenant de connexions **Lookup**), mais vérifiez qu'elles ne sont pas liées par un lien **Join**. Si elles le sont, veillez à ce que ce lien **Join** ne soit pas perdu.

Voir également : [Section 6.2.1.2, « Utiliser les jointures explicites ».](#)

6.2.1.1. 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, à partir du **Map Editor**.

Pour plus d'informations sur la configuration d'un schéma d'un composant, consultez la [Section 4.2.6, « Définir les propriétés d'un composant »](#).

Pour plus d'informations sur la configuration d'un schéma d'entrée dans le **Map Editor**, consultez la [Section 6.2.5, « Configuration des schémas dans le Map Editor »](#).

Contenu des tables Main et Lookup

L'ordre des tables **Input** (ou schémas) est essentiel.

La connexion **Main Row** détermine le contenu de la table **Main**. Ce flux entrant est représenté dans la première table de la zone **Input** du **[Map Editor]**.

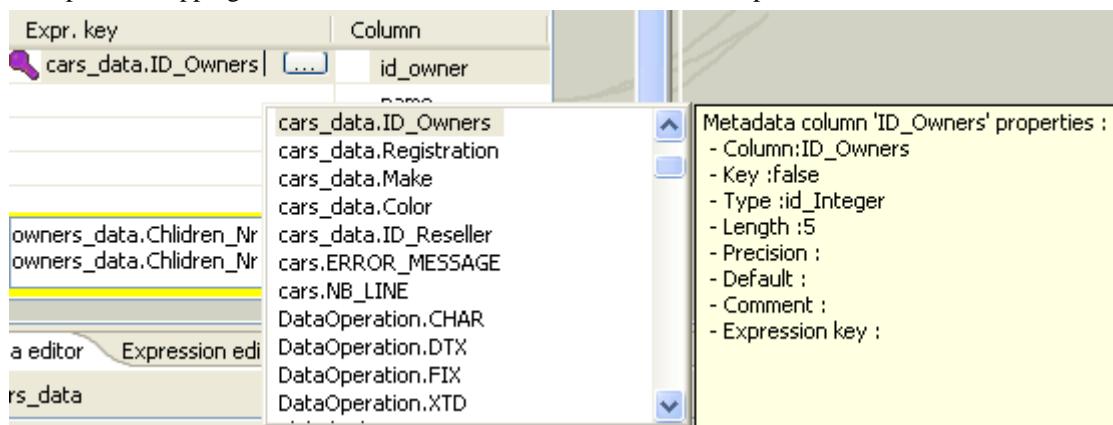
Le contenu des connexions secondaires (**Lookup**) est représenté dans les autres tables apparaissant en dessous de la table **Main**. Si vous n'avez pas encore défini de schéma pour un composant d'entrée, la table correspondante du **[Map Editor]** apparaîtra comme vide.

La clé (**Key**) est également reprise du schéma défini dans les propriétés du composant Input. Cette **Clé** correspond à la clé définie dans le schéma d'entrée, le cas échéant. 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 : [Section 6.2.2, « Mapping de variables »](#)

6.2.1.2. 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.

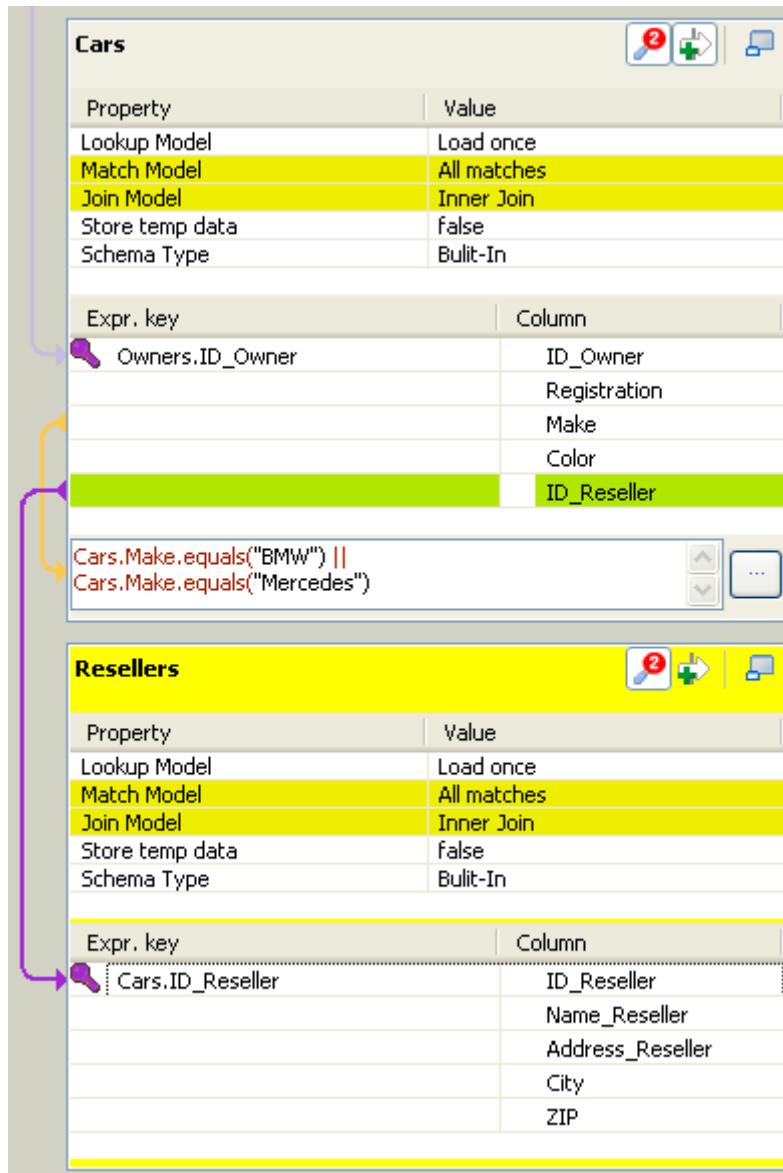
Vous pouvez créer des Joins directs entre la table principale et les tables secondaires. Mais vous pouvez également créer des Joins indirects à partir d'une table principale vers une table secondaire, elle-même reliée à une autre table secondaire. Pour cela, il faut que vous ayez créé une jointure entre 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 6.2.4, « Paramètres Output »](#).

- 💡 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.

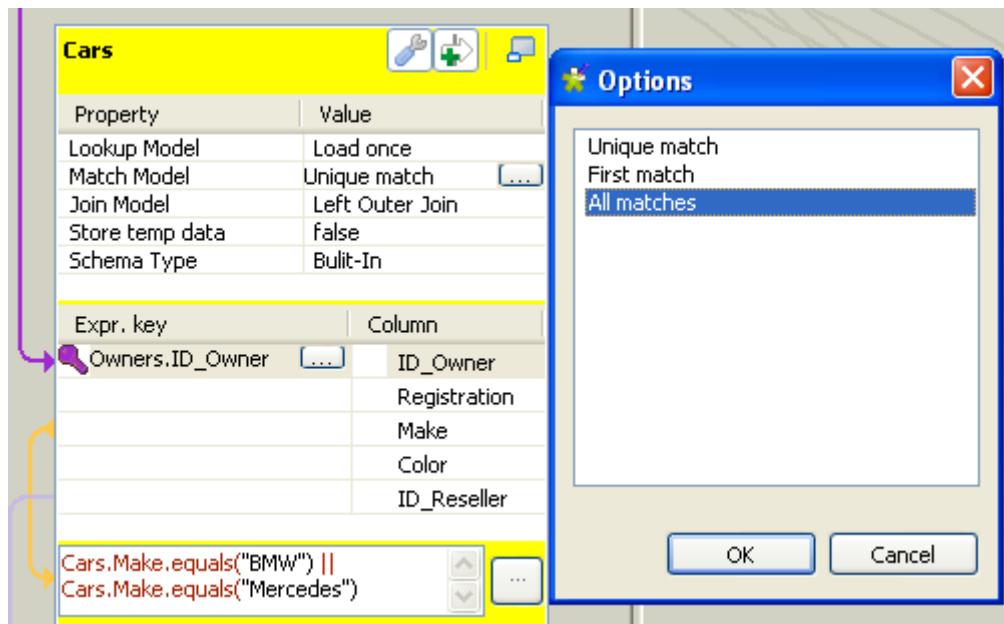
Voir également :

- [Section 6.2.5.2, « Utiliser le Schema editor »](#).
- [Section 6.2.1.3, « Utiliser la fonction Inner Join »](#).

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.

Pour définir le modèle de mise en correspondance de la jointure explicite :

1. Cliquez sur le bouton **tMap settings** en haut de la table à laquelle la jointure est reliée afin d'afficher les propriétés de la table.
2. Cliquez dans le champ **Value** correspondant à **Match Model** puis cliquez sur le bouton [...] qui apparaît afin d'ouvrir la boîte de dialogue [**Options**].
3. Dans la boîte de dialogue [**Options**], double-cliquez sur le modèle souhaité ou sélectionnez-le et cliquez sur **OK** pour valider les paramètres et fermer la boîte de dialogue.



Unique Match

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.

6.2.1.3. Utiliser la fonction Inner Join

La jointure interne **Inner join** est un type particulier de jointure qui se distingue par la façon dont les rejets sont traités.

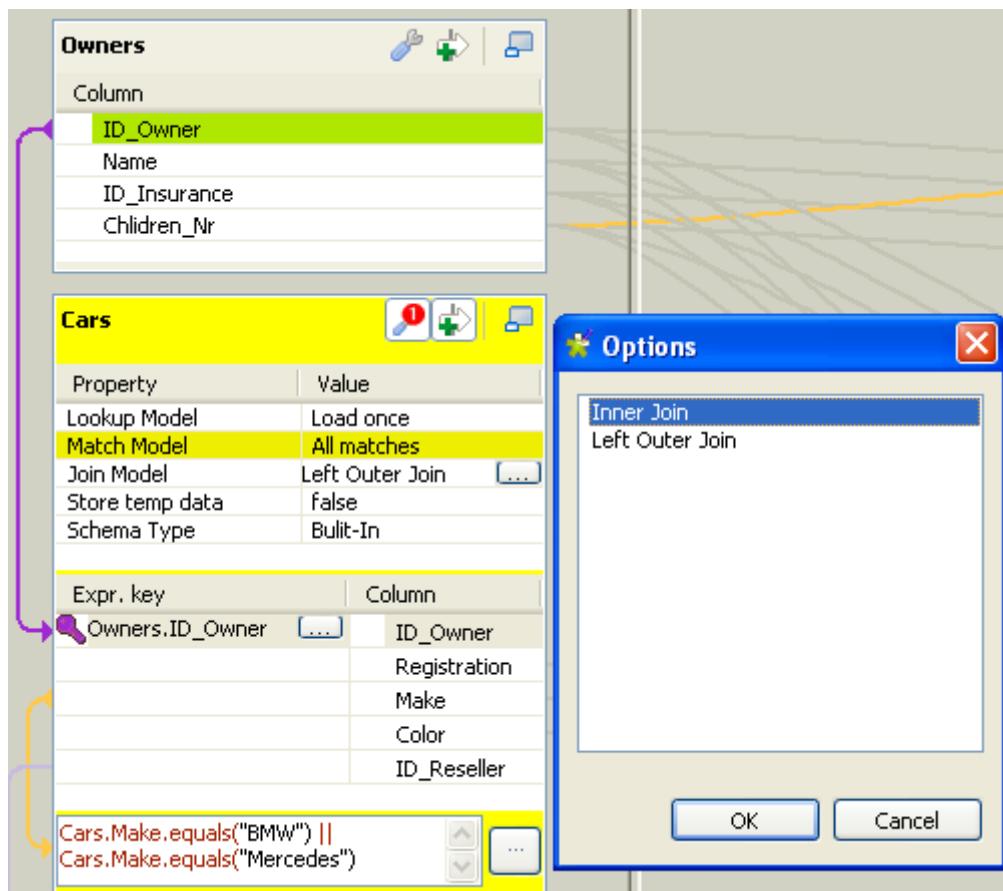
Cette fonction empêche les valeurs nulles de passer dans le flux principal de sortie. Elle permet aussi de faire passer les données rejetées dans une table définie comme table de rejet **Inner Join Reject**.

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

Déposez simplement le nom des colonnes d'une table vers une table subordonnée, afin de créer une relation de jointure 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.

Pour définir le type de jointure explicite :

1. Cliquez sur le bouton **tMap settings** situé en haut de la table de référence vers laquelle pointe la jointure pour afficher les propriétés du tableau.
2. Cliquez dans le champ **Value** correspondant à l'option **Join Model** pour faire apparaître le bouton [...] et cliquez dessus pour ouvrir la boîte de dialogue [**Options**].
3. Dans la boîte de dialogue [**Options**], double-cliquez sur le type de jointure souhaité, ou sélectionnez-le et cliquez sur **OK** pour activer l'option et fermer la boîte de dialogue.



Une table **Inner Join** devrait toujours être associée à une table de rejet **Inner Join Reject**. Pour savoir comment définir une table de sortie comme table **Inner Join Reject**, consultez [Section 6.2.4.4, « Utiliser les Rejets Inner Join »](#).

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

Voir également :

- [Section 6.2.4.4, « Utiliser les Rejets Inner Join ».](#)
- [Section 6.2.1.5, « Filtrer un flux d'entrée ».](#)

6.2.1.4. 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 la [Section 6.2.1.2, « Utiliser les jointures explicites »](#) et la [Section 6.2.1.3, « Utiliser la fonction Inner Join »](#).

6.2.1.5. Filtrer un flux d'entrée

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



Dans la zone de **Filtre**, saisissez les conditions que vous souhaitez appliquer, 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.

6.2.1.6. Retirer des entrées de la table Input

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

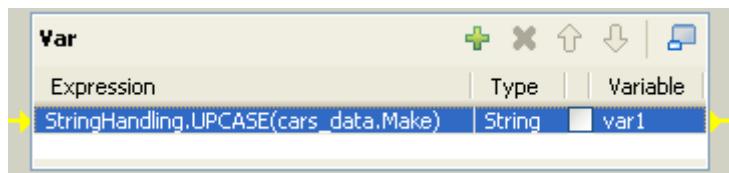
 Vous pouvez enlever des entrées Input d'un schéma dans le [Map Editor], cependant, cette suppression se répercute sur la définition du schéma au niveau des propriétés du composant d'entrée.

6.2.2. Mapping de variables

La table **Var** regroupe toutes les variables de mapping qui peuvent être utilisées à différents endroits du [Map Editor].

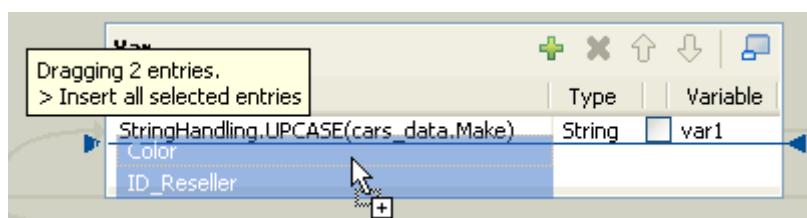
Vous pouvez également utiliser le champ **Expression** de la table **Var** pour réaliser des transformations en langage Java.

Les variables vous permettent de gagner du temps et vous évitez d'avoir à resaisir plusieurs fois les mêmes données.



Il existe plusieurs possibilités pour créer des variables :

- Saisissez librement vos variables en Java. Saisissez les chaînes de caractères entre guillemets simples ou concaténez plusieurs fonctions à l'aide de l'opérateur approprié.
- Ajoutez de nouvelles lignes à la table à l'aide du bouton [+] et retirez des lignes à l'aide du bouton [x]. Puis appuyez sur **Ctrl+Espace** pour récupérer les variables globales et de contexte déjà existantes.
- Déposez une ou plusieurs entrées Input dans la table **Var**.



Selectionnez 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 grise. Maintenez la touche **Ctrl** enfonce 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 Java appropriés. Le point permet la concaténation.

Comment faire pour...	Actions associées
Ecraser une entrée Var par les entrées Input sélectionnées, et les concaténer.	Déposez vers l'entrée Var appropriée qui se met alors en surveillance. 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 surveillance. 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 surveillance, et si nécessaire, de nouvelles lignes sont créées pour contenir les entrées restantes.

6.2.2.1. 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.

6.2.2.2. 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.

6.2.3. 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.

6.2.3.1. Accéder à l'Expression editor

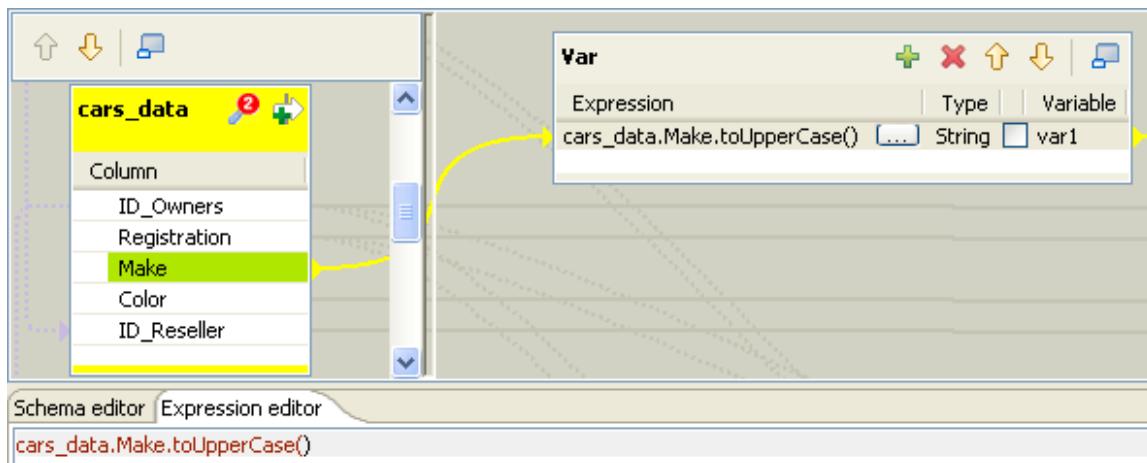
Vous pouvez écrire les expressions dont vous avez besoin pour la transformation de données directement dans la vue **Expression editor** située dans la partie inférieure de l'**Expression editor**, ou vous pouvez ouvrir la boîte de dialogue [**Expression Builder**] dans laquelle vous pouvez écrire les expressions pour la transformation de données.

Pour ouvrir la vue **Expression editor**, procédez comme suit :

1. Double-cliquez sur le composant **tMap** dans votre Job pour ouvrir le **Map Editor**.
2. Dans la partie inférieure de l'éditeur, cliquez sur l'onglet **Expression editor** afin d'ouvrir la vue correspondante.



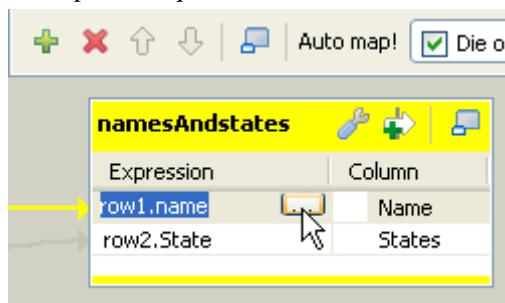
Pour éditer une expression, sélectionnez-la dans le panneau **Input** puis cliquez sur l'onglet **Expression editor** et modifiez l'expression comme nécessaire.



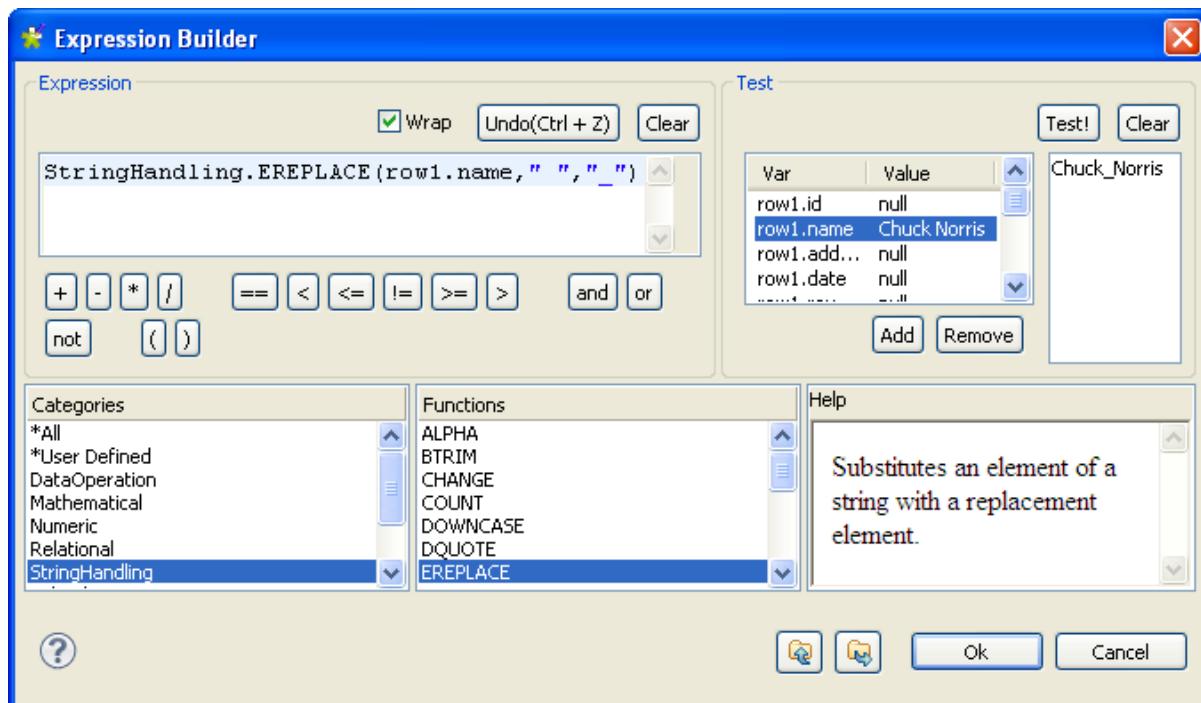
- Saisissez le code Java selon vos besoins. L'expression correspondante dans le panneau output est synchronisée.

 Référez-vous à la documentation Java adéquate pour plus d'informations concernant les fonctions et les opérations.

Pour ouvrir la boîte de dialogue [Expression Builder], 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.

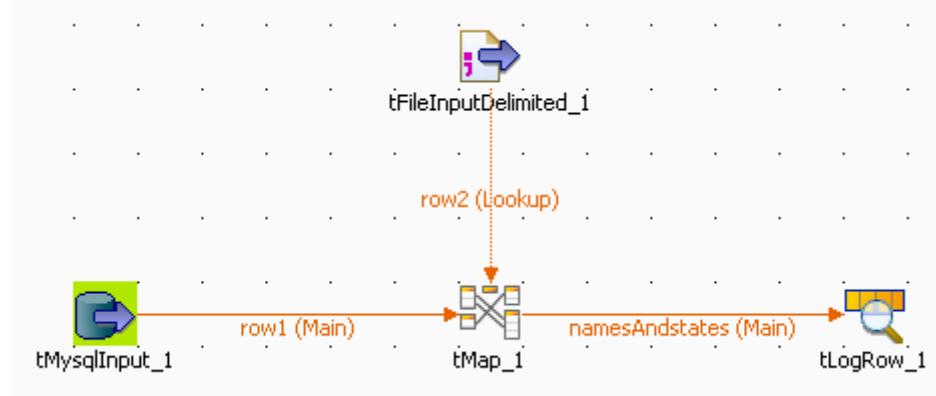


Pour un exemple d'utilisation de l'**Expression editor**, consultez la section suivante.

6.2.3.2. Créer du 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).

L'exemple suivant vous montre l'utilisation de l'**Expression Builder** pour le composant **tMap**.

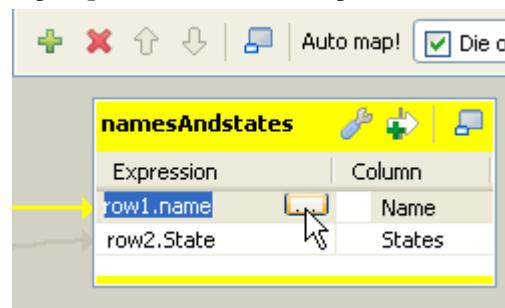


Deux flux d'entrée sont connectés au **tMap**.

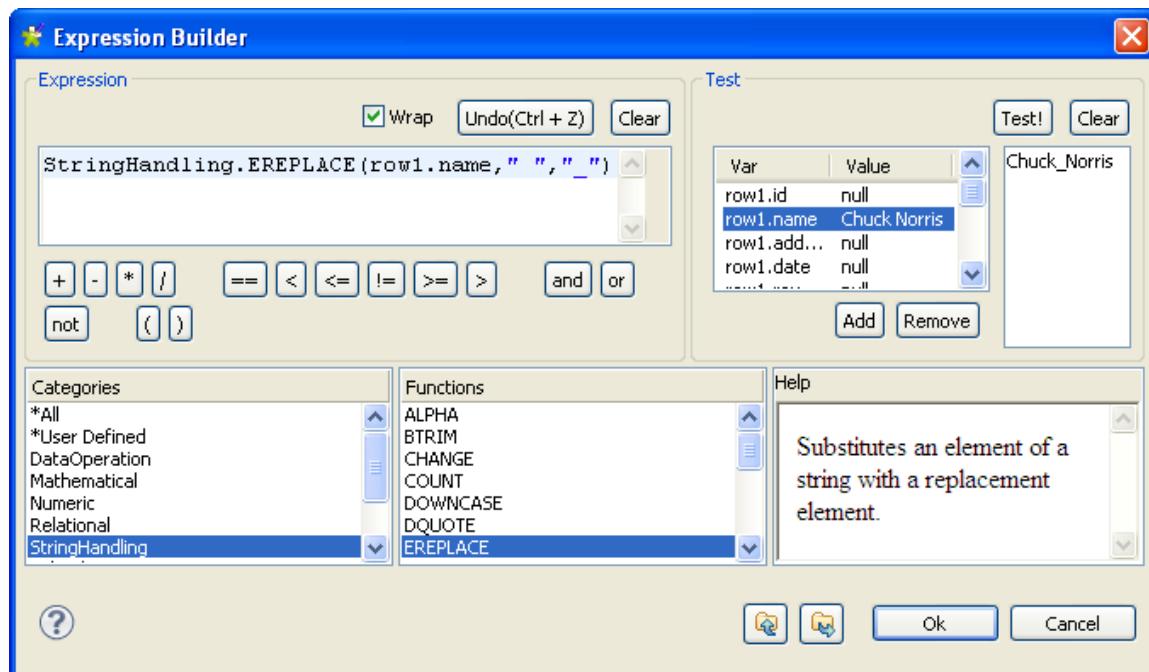
- Un premier flux DB input, contenant une liste de personnes dont le nom et le prénom sont séparés par un espace.
- Un autre flux File input, contenant la liste des états des Etats-Unis, en minuscule.

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 majuscules le nom des états.

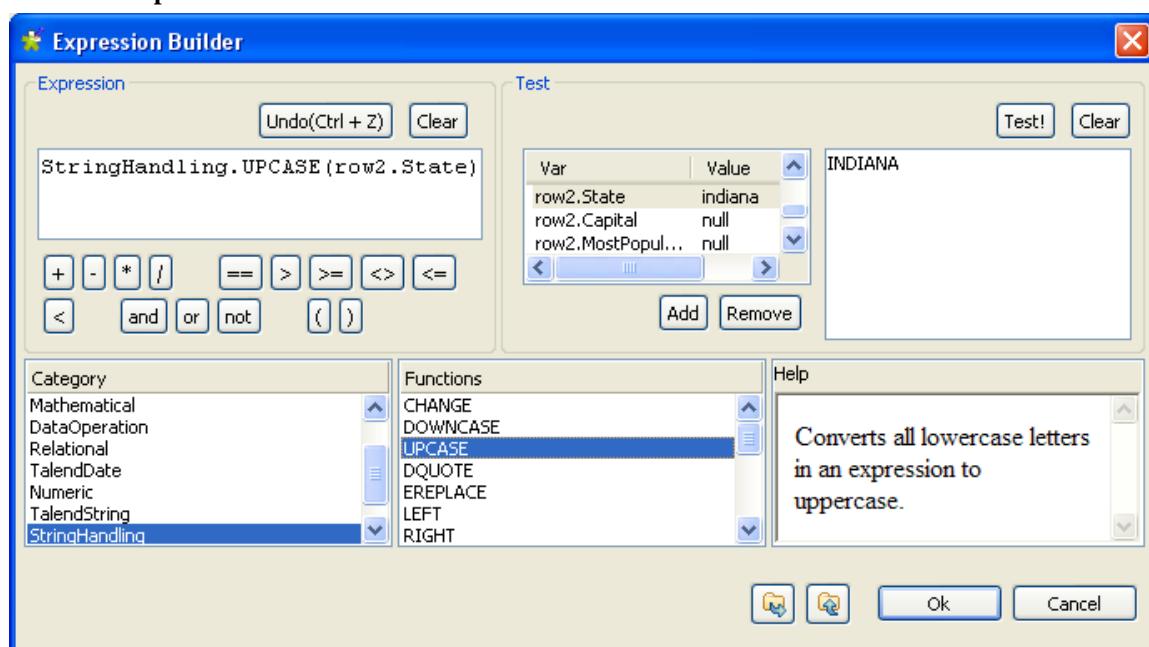
1. Dans le **[Map Editor]**, créez la jointure Inner Join correspondante pour créer le mapping de référence. Pour plus d'informations concernant l'utilisation du **tMap**, consultez le [Chapitre 7, Gestion des métadonnées](#).
2. Déposez la colonne *Name* du flux d'entrée principal Main (*row1*) de la zone **Input** dans la zone **Output**, et la colonne *State* du flux d'entrée secondaire Lookup (*row2*) vers la même zone **Output**.
3. Puis cliquez sur le premier champ **Expression** (*row1.Name*) pour afficher le bouton [...].



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



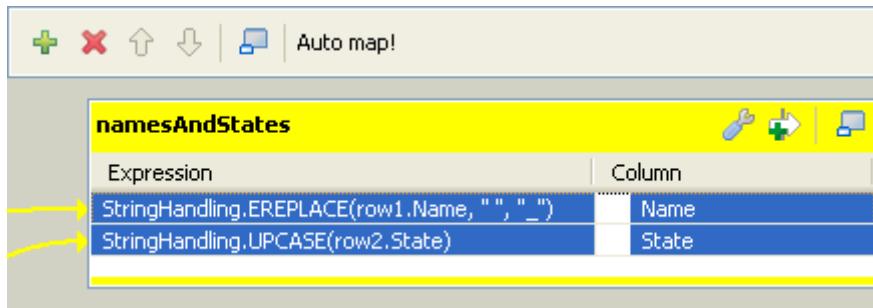
- Dans la liste **Category**, sélectionnez l'action que vous souhaitez effectuer. Pour cet exemple, choisissez **StringHandling**, puis sélectionnez la fonction **EREPLACE** dans la liste **Functions**.
- Dans la zone **Expression**, collez `row1.Name` à la place du texte, pour obtenir : `StringHandling.EREPLACE(row1.Name, " ", "_")`. 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 *Chuck Norris* puis cliquez sur **Test!** et vérifiez que les changements ont bien été effectués, ici : *Chuck_Norris*.
- Cliquez sur **OK** pour valider et 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)`.

9. Vérifiez que la syntaxe de votre code est correcte, en saisissant par exemple *indiana* dans le champ **Value** de la zone **Test**. Cliquez sur **Test!** et le résultat affiché est *INDIANA*. Cliquez sur **OK** pour valider.

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

6.2.4. Paramètres Output

Dans l'espace de modélisation de *Talend Open Studio for Data Integration*, 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]**.

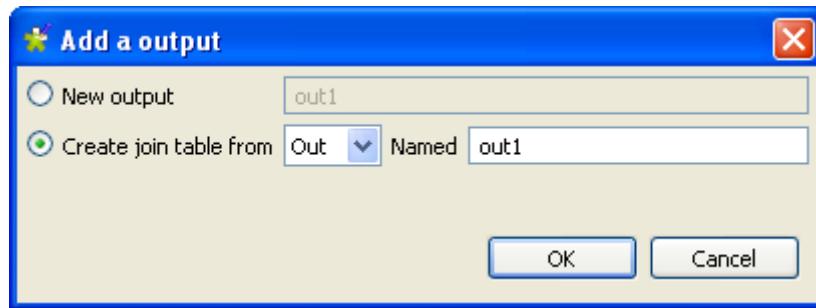
Vous pouvez également ajouter un schéma Output dans votre **[Map Editor]**, à l'aide du signe **[+]** de la barre d'outils de la zone **Output**.

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 **Guide de Référence Talend Open Studio for Data Integration**.



La table de jointure récupère le schéma de la table source.

Lorsque vous cliquez sur le bouton **[+]**, afin d'ajouter un schéma Output ou d'effectuer une jointure entre vos tables de sortie, une boîte de dialogue s'ouvre, vous proposant deux possibilités.



Sélectionnez...	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 Output appropriée.

Appuyez sur **Ctrl** ou **Maj** pour une sélection multiple.

Ou vous pouvez utiliser des expressions de la zone **Var** par glisser-déposer dans la table **Output** avec les données réutilisables appropriées.

Notez que si vous apportez des modifications à la colonne Input du Schema Editor, une boîte de dialogue vous demande de confirmer la propagation des modifications sur toutes les entrées Input/Variable/Output concernées.

Action	Résultat
Glisser-déposer vers des expressions existantes	Concatène l'expression sélectionnée, vers des expressions existantes.
Glisser-déposer vers une ligne d'insertion	Insère une ou plusieurs nouvelle(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.

6.2.4.1. 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 la [Section 6.2.3.2, « Créez du code à l'aide de l'Expression Builder »](#)

6.2.4.2. 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.



Vous pouvez saisir librement vos conditions de filtre à l'aide d'opérateurs et de fonctions Java.

Déposez des expressions d'une zone **Input** ou d'une zone **Var** vers l'entrée de Filtre de la table Output appropriée.



Un lien graphique de couleur orange apparaît. Ajoutez l'opérateur Java nécessaire pour finaliser votre formule de filtre.

Vous pouvez créer plusieurs filtres sur différentes lignes. L'opérateur AND est la conjonction logique de tous les filtres formulés.

6.2.4.3. Utiliser la fonction Rejet

L'option **Reject** définit la nature d'une table Output particulière.

Cette table de rejet rassemble toutes les données qui ne satisfont pas un ou plusieurs des filtres définis dans les tables Output standard. Notez que par table standard sont désignées toutes les tables qui ne sont pas des tables de rejet.

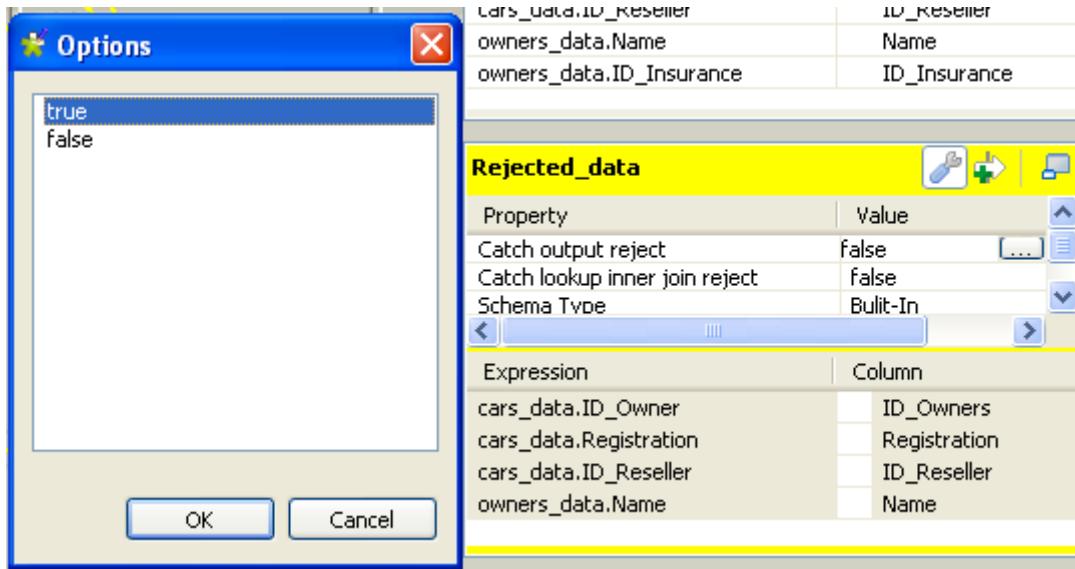
Ainsi les données rejetées des tables de sortie régulières sont regroupées dans une ou plusieurs tables dédiées, vous permettant par conséquent d'identifier les erreurs ou les cas imprévus.

Le principe de rejet (Reject) concatène tous les filtres des tables non-rejet et les définit comme formulation ELSE.

Pour définir une table de sortie comme la partie ELSE des tables régulières :

1. Cliquez sur le bouton **tMap settings** en haut de la table de sortie pour afficher le tableau des propriétés.

2. Cliquez sur le champ **Value** correspondant à l'option **Catch output reject** et cliquez sur le bouton [...] qui apparaît pour ouvrir la boîte de dialogue **[Options]**.
3. Dans la boîte de dialogue **[Options]**, double-cliquez sur **true**, ou sélectionnez-le et cliquez sur **OK** pour activer l'option et fermer la boîte de dialogue.



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.

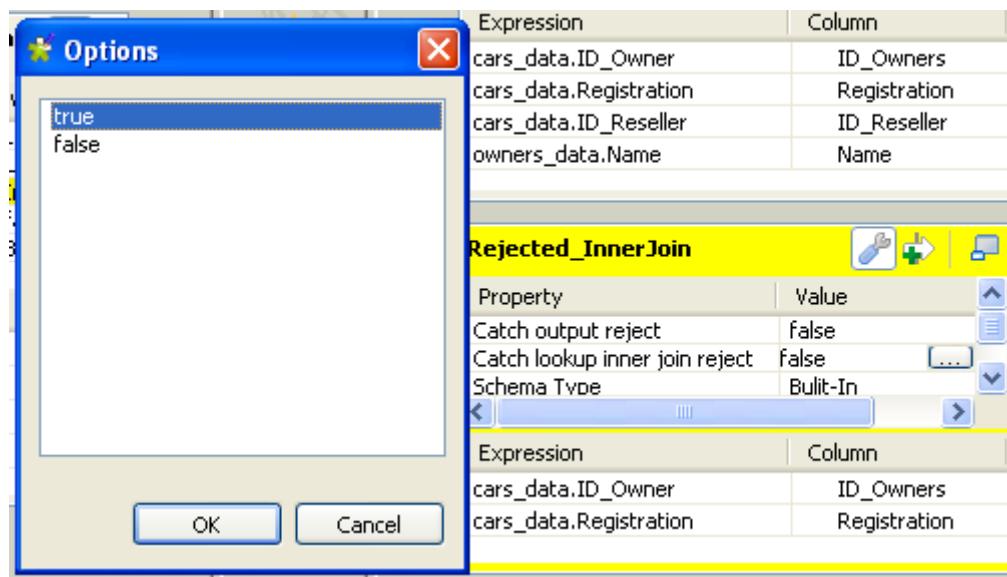
Les données ne sont pas exclusivement traitées vers une seule sortie. Même si une donnée satisfait le filtre de sortie et qu'elle est donc routée vers elle, elle est également vérifiée contre les autres filtres et peut être également routée vers d'autres sorties.

6.2.4.4. 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]**, suivez les étapes suivantes :

1. Cliquez sur le bouton **tMap settings** en haut de la table de sortie pour afficher le tableau des propriétés.
2. Cliquez sur le champ **Value** correspondant à l'option **Catch lookup inner join reject** et cliquez sur le bouton [...] qui apparaît pour ouvrir la boîte de dialogue **[Options]**.
3. Dans la boîte de dialogue **[Options]**, double-cliquez sur **true**, ou sélectionnez-le et cliquez sur **OK** pour activer l'option et fermer la boîte de dialogue.



6.2.4.5. 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.

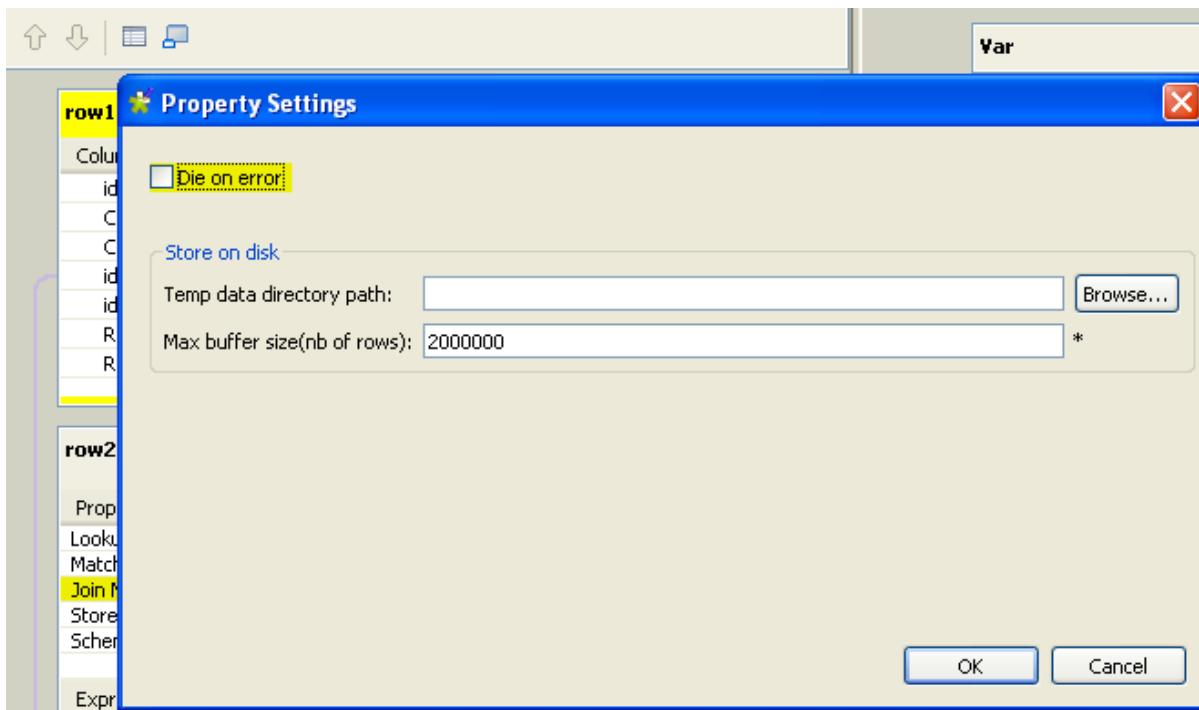
6.2.4.6. Gestion des erreurs

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.

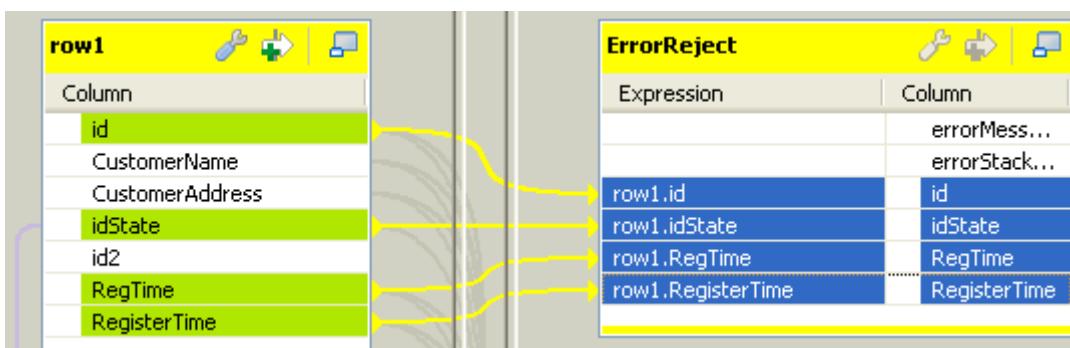
Pour désactiver l'option **Die on error** :

1. Double-cliquez sur le composant **tMap** dans l'espace de modélisation pour ouvrir le **Map Editor**.
2. Cliquez sur le bouton **Property Settings** en haut de la zone input pour ouvrir la boîte de dialogue **[Property Settings]**.
3. Dans la boîte de dialogue **[Property Settings]**, décochez la case **Die on error** et cliquez sur **OK**.

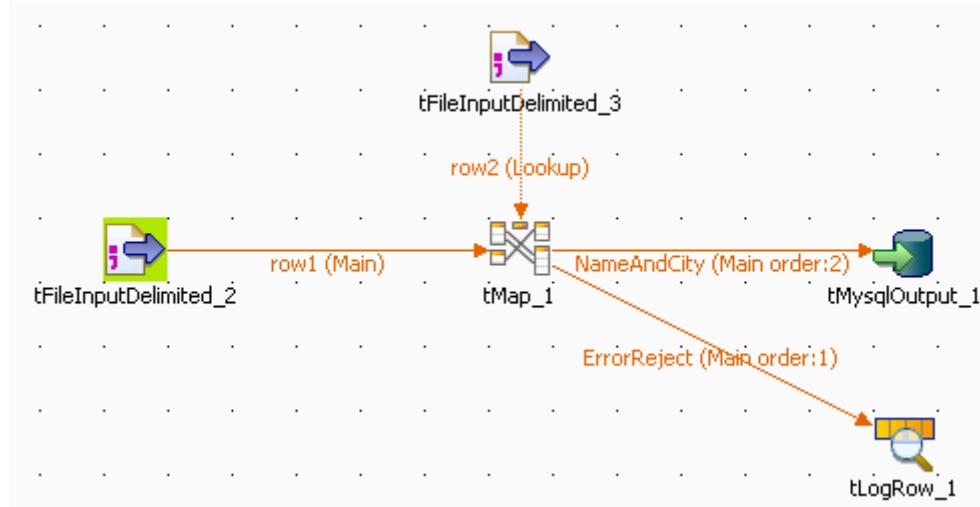


Une nouvelle table appelée **ErrorReject** s'affiche dans l'éditeur du **tMap**, dans la zone de sortie. Cette table de sortie comprend automatiquement deux colonnes : **errorMessage** et **errorStackTrace**, qui retrouvent le message et la trace de la pile de l'erreur rencontrée durant l'exécution du Job. Les erreurs peuvent être des erreurs de format de dates, des null pointer exceptions, des problèmes de conversion, etc.

Vous pouvez également glisser-déposer des colonnes (ici *id* et *date*) des tables d'entrée vers cette table de sortie de rejets. Ces données en erreur peuvent être retrouvées avec le message d'erreur correspondant et être corrigées ensuite.



Une fois la table *ErrorReject* définie, les flux correspondants peuvent être envoyés vers un composant de sortie.



Pour ce faire, dans l'espace de modélisation graphique, cliquez-droit sur le composant **tMap**, sélectionnez **Row > ErrorReject** dans le menu, et cliquez sur le composant de sortie correspondant, ici, le **tLogRow**.

Lorsque vous exécutez le Job, les erreurs sont retournées via le flux **ErrorReject**.

```

Starting job Die_on_error at 17:30 01/09/2010.

java.text.ParseException: Unparseable date: "08 01
1980" | java.lang.RuntimeException:
java.text.ParseException: Unparseable date: "08 01 1980"
    at routines.TalendDate.parseDate(TalendDate.java:503)
    at
doc.die_on_error_0_1.Die_on_error.tFileInputDelimited_2Pro
cess(Die_on_error.java:1409)
    at
doc.die_on_error_0_1.Die_on_error.runJobInTOS(Die_on_error.
java: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 résultat contient le message d'erreur, la trace de sa pile, et les deux colonnes *id* et *date*, glissées et déposées dans la table **ErrorReject**, séparées par une barre verticale “|”.

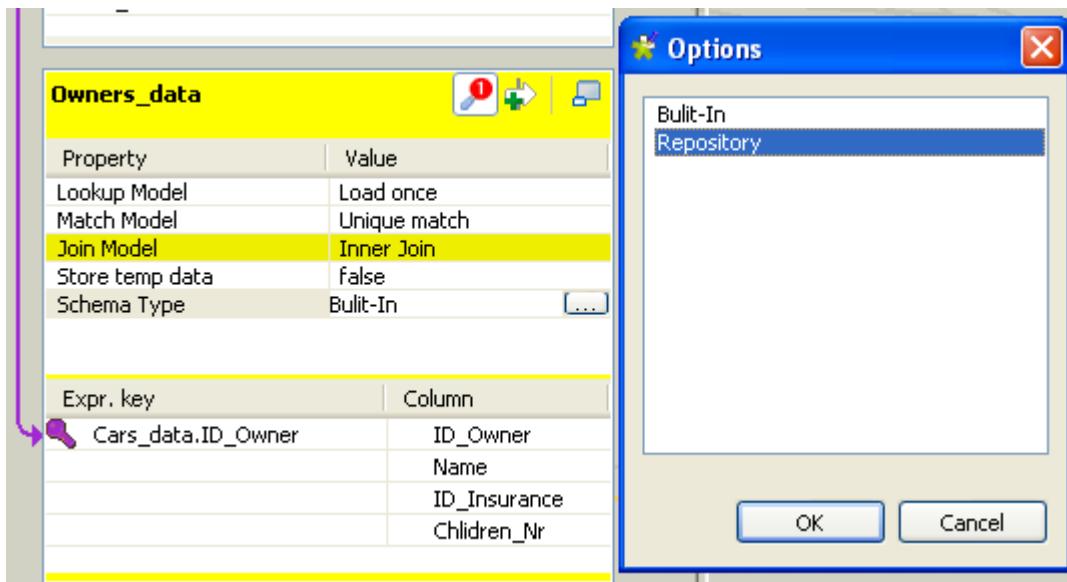
6.2.5. Configuration des schémas dans le Map Editor

Dans le **Map Editor**, vous pouvez définir le type de schéma d'une table comme **Built-In** afin de pouvoir modifier la structure des données dans le panneau **Schema editor**, ou vous pouvez le définir comme **Repository** et récupérer la structure des données depuis le Repository. Par défaut, le type du schéma est défini comme **Built-In** pour toutes les tables.

6.2.5.1. Récupérer la structure d'un schéma depuis le Repository

Pour récupérer la structure d'un schéma de la table sélectionnée dans le **Repository** :

1. Cliquez sur le bouton **tMap Settings** en haut de la table pour afficher le tableau des propriétés.
2. Cliquez sur le champ **Value** correspondant à l'option **Schema Type**, et cliquez sur le bouton [...] qui apparaît pour ouvrir la boîte de dialogue **[Options]**.



3. Dans la boîte de dialogue **[Options]**, double-cliquez sur **Repository**, ou sélectionnez-le et cliquez sur **OK** pour fermer la boîte de dialogue et afficher une nouvelle option **Schema Id** en dessus du champ **Schema Type** dans le tableau des propriétés.



Si vous fermez le **Map Editor** maintenant sans spécifier le schéma à utiliser, le type du schéma redevient **Built-In**.

4. Cliquez sur le champ **Value** de l'option **Schema Id**, et cliquez sur le bouton [...] qui apparaît dans la boîte de dialogue **[Repository Content]**.
5. Dans la boîte de dialogue **[Repository Content]**, sélectionnez votre schéma comme vous le faites pour n'importe quel autre composant et cliquez sur **OK**.

Le champ **Value** de l'option **Schema Id** est renseigné avec le schéma que vous venez de sélectionner, et le schéma de cette table affiché dans le panneau **Schema editor** passe en lecture seule.

The screenshot shows the Talend Map Editor's Schema editor interface. At the top, there's a properties panel for the 'Owners_data' component. Below it, two tables are shown: one for 'Expr. key' and another for 'Column'. The 'Column' table lists four columns: ID_Owner, Name, ID_Insurance, and Children_Nr. At the bottom, a larger table provides detailed schema information for these columns.

Column	Key	Type	Length	Precision	Nullable	Comment
ID_Owner	<input checked="" type="checkbox"/>	int	2	0	<input checked="" 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"/>	
Children_Nr	<input type="checkbox"/>	int	2	0	<input checked="" type="checkbox"/>	

- 💡 Si vous changez le type du schéma d'une table subordonnée contenant une jointure de **Built-In à Repository**, cette jointure sera perdue.
- 💡 Les modifications apportées au schéma d'une table à partir du **Map Editor** sont automatiquement répercutées sur le schéma du composant correspondant, relié au **tMap**.

6.2.5.2. Utiliser le Schema editor

Le **Schema Editor** fournit les informations concernant les champs d'une table sélectionnée. Lorsque le type du schéma est défini comme **Built-in**, vous pouvez modifier ce schéma à partir du **Schema Editor**.

The screenshot shows the Schema editor for the 'owners_data' component. It displays a table with columns: ID_Owner, First Name, Name, and ID_Insurance. The 'First Name' column is highlighted. Below the table is a toolbar with several icons for managing columns.

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.
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 n'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).

6.2.6. Résolution des problèmes de mémoire lors de l'utilisation du tMap

Lorsque vous devez traiter un nombre important de données, par exemple, de nombreuses colonnes, différents types de colonnes ou lignes, votre système peut rencontrer des problèmes de mémoire insuffisante empêchant votre Job de s'exécuter correctement, et plus particulièrement lorsque vous utilisez un composant **tMap** pour effectuer des transformations.

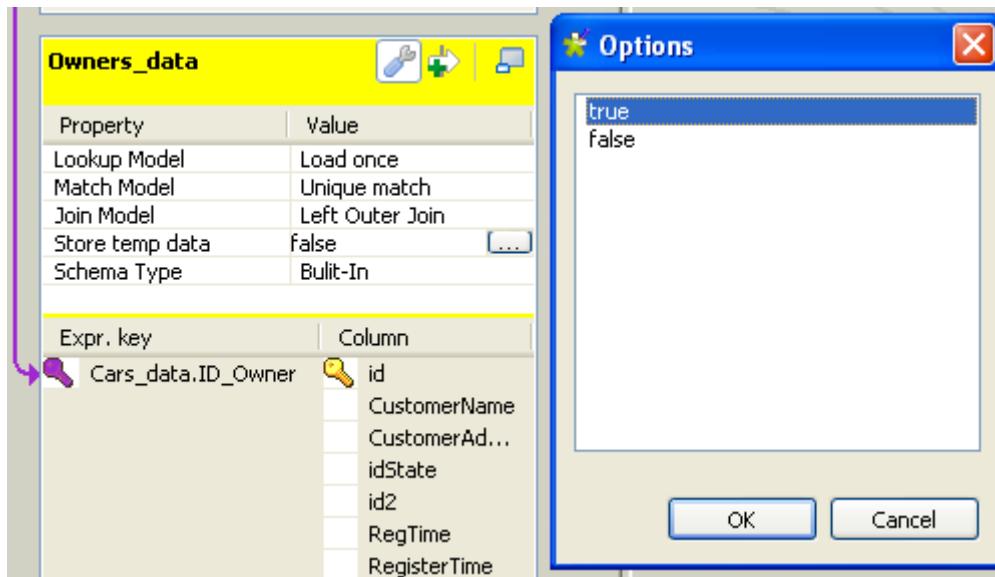
Une option (uniquement disponible en Java pour le moment) a été ajoutée au composant **tMap**, pour utiliser moins de mémoire lors du traitement des données de référence (lookup). En effet, au lieu de stocker les données de référence dans la mémoire système et ainsi en atteindre les limites, l'option **Store temp data** vous permet de stocker les données de référence dans un dossier temporaire sur votre disque dur.

Cette option peut être sélectionnée sur la table Lookup de la zone **Input** (à gauche) de l'éditeur du **tMap**, le **[Map Editor]**.

Pour activer l'option **Store temp data**, suivez les étapes suivantes :

1. Double-cliquez sur le composant **tMap** de votre Job pour lancer le **[Map Editor]**.
2. Dans le panneau de gauche, correspondant aux sources **Input** et **Lookup**, cliquez sur la table **Lookup** affichant les données de Lookup que vous souhaitez charger sur le disque pour ne pas encombrer la mémoire.
3. Cliquez sur le bouton **tMap settings** pour afficher le tableau des propriétés.
4. Cliquez sur le champ **Value** correspondant à l'option **Store temp data**, et cliquez sur le bouton [...] pour ouvrir la boîte de dialogue **[Options]**.

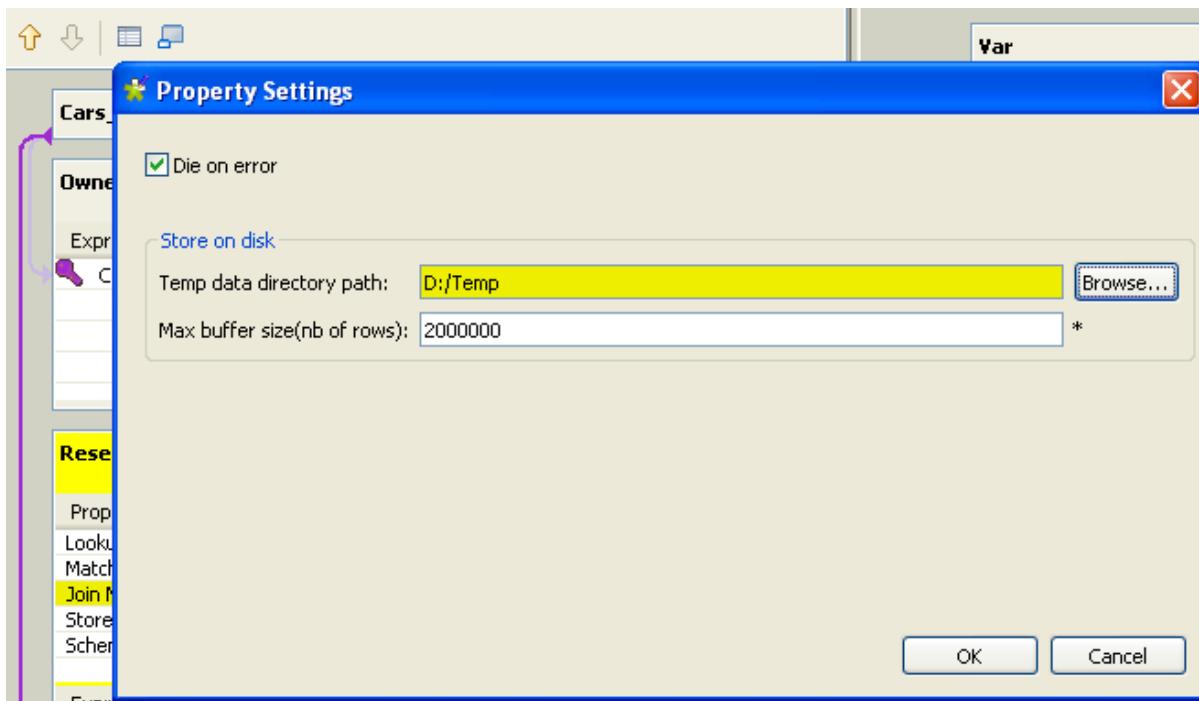
5. Dans la boîte de dialogue **[Options]**, double-cliquez sur **true**, ou sélectionnez-le et cliquez sur **OK** pour activer l'option et fermer la boîte de dialogue.



Pour que cette option soit totalement active, vous devez définir le répertoire de votre disque dans lequel les données seront stockées, et la taille du buffer, à savoir le nombre de lignes de données stockées dans chaque fichier temporaire. Vous pouvez spécifier le répertoire de stockage temporaire des données et la taille du buffer à la fois à partir du **Map Editor** et de la vue **Component** du **tMap**.

Pour paramétriser le répertoire de stockage temporaire et la taille du buffer à partir du **Map Editor** :

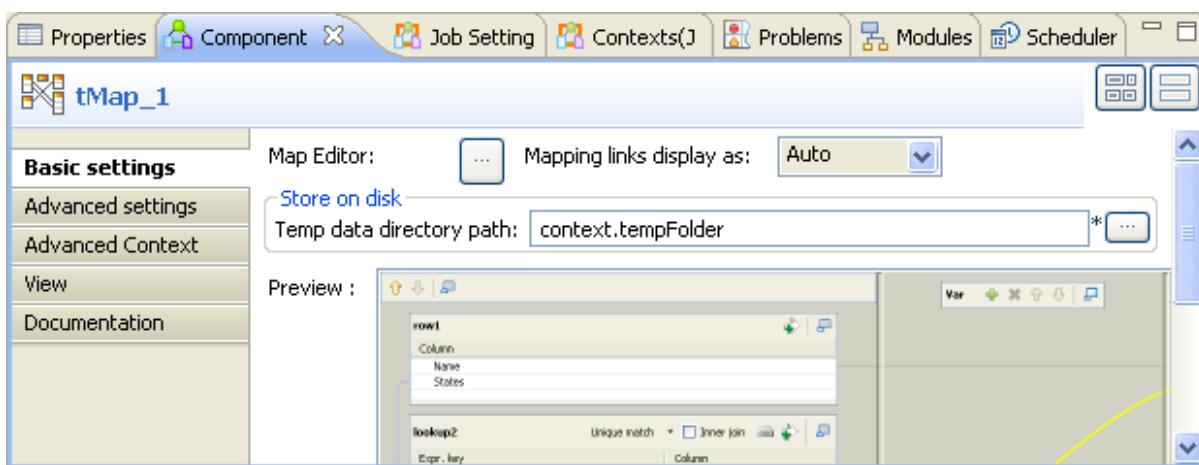
1. Cliquez sur le bouton **Property Settings** en haut de la zone Input pour ouvrir la boîte de dialogue **[Property Settings]**.
2. Dans la boîte de dialogue **[Property Settings]**, renseignez le chemin d'accès complet vers le dossier dans lequel les données temporaire seront stockées dans le champ **Temp data directory path**.
3. Dans le champ **Max buffer size (nb of rows)**, spécifiez le nombre maximum de lignes contenues dans chaque fichier temporaire. La valeur par défaut est de **2 000 000** lignes.
4. Cliquez sur **OK** pour valider les paramètres et fermer la boîte de dialogue **[Property Settings]**.



Pour paramétriser le répertoire de stockage temporaire du composant **tMap** sans avoir à ouvrir le **Map Editor** :

1. Cliquez sur le composant **tMap** dans l'espace de modélisation pour le sélectionner, puis sélectionnez la vue **Component** pour afficher l'onglet **Basic settings**.
2. Dans la zone **Store on disk**, renseignez le chemin d'accès complet vers le dossier dans lequel les données temporaires seront stockées dans le champ **Temp data directory path**.

Vous pouvez aussi utiliser une variable de contexte à l'aide du raccourci **Ctrl+Espace** si vous avez configuré une variable dans un groupe de contexte dans le **Repository**. Pour plus d'informations sur les contextes, consultez la [Section 4.4.2, « Centraliser les Contextes et les Variables »](#)



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 (Nb of rows)** afin de séparer les données stockées sur le disque en autant de fichiers que nécessaire.

6.2.7. 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.

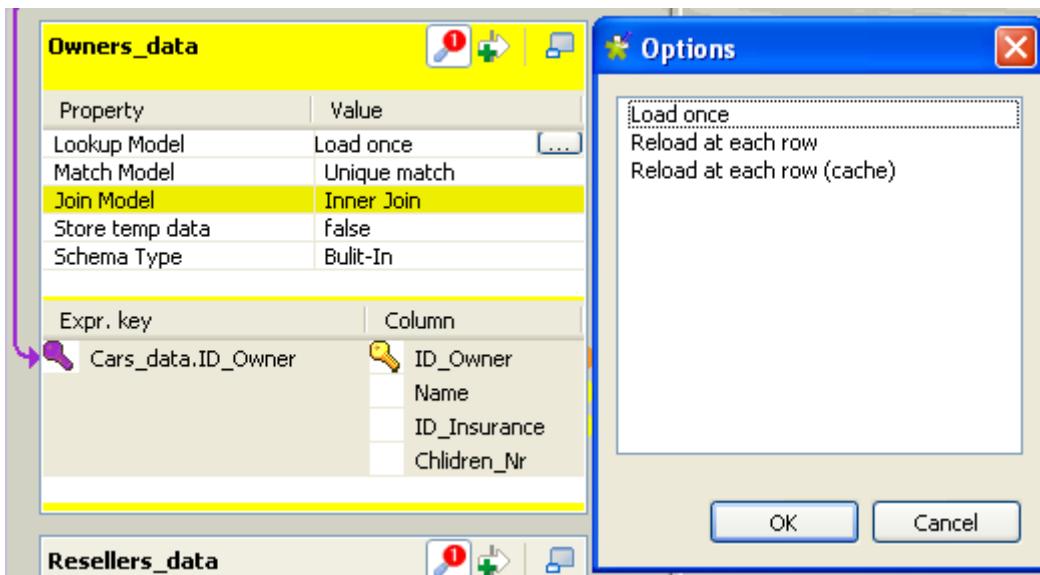
- **Load once** : Paramètre par défaut. Sélectionnez cette option pour charger la totalité du flux de référence avant de traiter le flux principal. Sélectionnez cette option si un nombre important de données de votre flux principal est sollicité par votre flux de référence (lookup), ou si vos données 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 principal 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 configurer le mode de chargement du flux de référence :

1. Cliquez sur le bouton **tMap settings** en haut de la table de référence (lookup) pour afficher le tableau des propriétés.
2. Cliquez sur le champ **Value** correspondant à l'option **Lookup Model**, et cliquez sur le bouton [...] pour ouvrir la boîte de dialogue **[Options]**.
3. Dans la boîte de dialogue **[Options]**, double-cliquez sur le mode de chargement souhaité, ou sélectionnez-le et cliquez sur **OK** pour activer l'option et fermer la boîte de dialogue.



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 **Guide de référence des Composants Talend Open Studio**.

6.3. Présentation du fonctionnement du tXMLMap

 Avant de lire les sections suivantes, il est recommandé de lire les sections précédentes concernant le **tMap** afin d'avoir les connaissances des base du composant **Talend** de mapping.

Le **tXMLMap** est conçu pour utiliser le type de données **Document** afin de traiter des données XML, en cas de transformation mélangeant souvent des données hiérarchiques (XML) et des données plates. Il vous permet d'ajouter autant de flux d'entrée et de sortie que nécessaire dans un éditeur graphique de mapping dans lequel vous pouvez effectuez des opérations, comme :

- multiplexage et démultiplexage des données,
- transformation des données sur tout type de champs, particulièrement sur le type **Document**,
- mise en correspondance de données via différents modes, par exemple le mode **Unique match** (consultez la [Section 6.2.1.2, « Utiliser les jointures explicites »](#)),
- construction d'arbre XML des côtés d'entrée et de sortie,
- jointures Inner Join et Left Outer Join (consultez la [Section 6.2.1.3, « Utiliser la fonction Inner Join »](#)),
- flux lookup entre tous types de sources de données, XML ou plates à l'aide de modèles comme **Load once** (consultez la [Section 6.2.7, « Gestion des références »](#)),
- concaténation et inversion de champs,
- filtrage de champs à l'aide de contraintes,
- gestion des rejets de données.

Comme pour le **tMap**, un éditeur de mapping est requis pour configurer ces opérations. Pour ouvrir cet éditeur, vous pouvez double-cliquer sur l'icône du **tXMLMap** dans l'espace de modélisation graphique ou bien cliquer sur le bouton [...] à côté du **Map Editor** dans la vue **Basic settings** du composant **tXMLMap**.

Les composants **tXMLMap** et **tMap** ont un fonctionnement quasiment identique. Ainsi, les sections qui suivent ne couvriront que le fonctionnement spécifique au **tXMLMap**, à savoir le traitement des données hiérarchiques.

Les opérations relatives aux données hiérarchiques sont :

- l'utilisation du type **Document**. Pour plus d'informations, consultez la [Section 6.3.1, « Utiliser le type Document pour créer l'arborescence XML »](#).
- la gestion des données XML de sortie. Pour plus d'informations, consultez la [Section 6.3.2, « Définir le mode de sortie »](#).
- l'édition de schémas d'arborescence XML. Pour plus d'informations, consultez la [Section 6.3.3, « Utiliser le Tree schema editor »](#).

Les sections suivantes reprennent plus en détails les diverses fonctionnalités du **tXMLMap**.



A la différence du **tMap**, le **tXMLMap** ne possède pas l'option **Store temp data** pour stocker les données temporaires dans un répertoire de votre disque. Pour plus d'informations concernant cette option du **tMap**, consultez la [Section 6.2.6, « Résolution des problèmes de mémoire lors de l'utilisation du tMap »](#).

6.3.1. Utiliser le type Document pour créer l'arborescence XML

Le type de données **Document** correspond parfaitement au concept de données structurées de type XML. Lorsque vous devez utiliser la structure XML pour mapper le flux d'entrée ou de sortie, ou les deux, utilisez ce type. Vous pouvez importer des arborescences XML de plusieurs sources XML et de les éditer directement dans son éditeur de mapping, afin de vous éviter de les éditer manuellement un à un.

6.3.1.1. Configurer le type Document

Le type de données **Document** est un des types de données fournis par **Talend**. Le type **Document** se sélectionne lorsque vous définissez le schéma correspondant à vos données dans le panneau **Schema editor**. Pour plus d'informations sur le **Schema editor**, consultez la [Section 6.2.5.2, « Utiliser le Schema editor »](#).

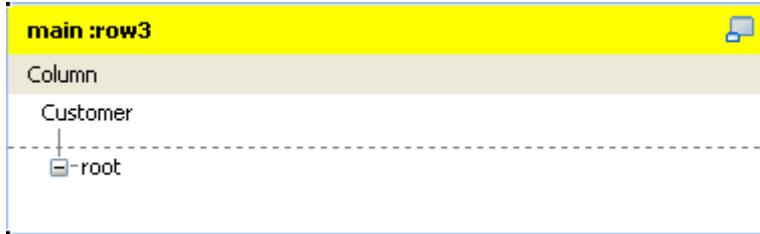
La capture d'écran ci-dessous montre un exemple de flux d'entrée, *Customer*, de type **Document**. Afin de le reproduire dans le **Map editor**, cliquez sur le bouton [...] pour ajouter une ligne du côté de l'entrée du schéma, renommez-la et sélectionnez **Document** dans la liste des types.

The screenshot shows the Talend Schema editor interface. At the top, there are three tabs: 'Schema editor' (selected), 'Tree schema editor', and 'Expression editor'. Below the tabs is a table with columns: Column, Key, Type, N.., Date Pattern ..., Length, Precision, Def..., and Comm... . A row in the table is selected for the 'Customer' column, and the 'Type' dropdown is open, showing a list of options: short | Short, String, List, Dynamic, and Document. The 'Document' option is highlighted. At the bottom of the editor are several toolbar icons.

Dans la plupart des cas, le tXMLMap récupère le schéma du composant précédent ou suivant, par exemple, d'un **tFileInputXML** ou d'un scénario ESB, d'un **tESBProviderRequest**. Cela permet d'éviter les efforts manuels lors de la définition du type **Document** dans le flux XML à traiter. Cependant, pour continuer à modifier la structure XML ainsi que le contenu d'une ligne de type **Document**, vous devez utiliser le **Map editor**.

-  Un flux **Document** comporte une arborescence XML personnalisée et représente un seul champ du schéma, pouvant contenir plusieurs champs de types différents. Pour plus d'informations concernant la configuration d'un schéma, consultez [Section 4.2.6.1, « Onglet Basic Settings »](#).

Une fois une ligne de données définie comme type **Document** dans la table du flux de données correspondante, une arborescence XML de base est créée automatiquement pour ce flux de données.



Cette figure vous montre un exemple du flux d'entrée XML, *Customer*. A partir de la racine XML créée, vous pouvez développer l'arborescence XML qui vous intéresse.

Pour ce faire, vous devez :

1. importer l'arborescence XML personnalisée à partir de l'un des types de sources suivants :
 - des fichiers XML (consultez la [Section 6.3.1.2, « Importer une structure XML à partir de fichiers XML »](#)) ;
 - des connexions vers des fichiers XML créées et centralisées dans les métadonnées du **Repository** dans le Studio (consultez la [Section 6.3.1.3, « Importer une structure XML à partir du Repository »](#)). Pour plus d'informations sur la création de connexions à des fichiers XML dans le Repository, consultez la [Section 7.8, « Configurer un Schéma File XML »](#).
2. définir l'élément de boucle pour l'arborescence XML que vous créez. A cette étape, vous devez envisager les situations suivantes :
 - Si vous avez créé plusieurs arborescences XML, vous devez définir un élément de boucle pour chacune d'entre elles.
 - Si vous importez l'arborescence XML du Repository, l'élément de boucle aura déjà été défini. Cependant, vous pouvez toujours le réinitialiser. Pour plus d'informations, consultez la [Section 6.3.1.4, « Définir ou réinitialiser un élément de boucle pour une structure XML créée »](#).

Si nécessaire, vous pouvez continuer à modifier l'arborescence XML à l'aide des options fournies dans le menu contextuel. Le tableau suivant présente les opérations que vous pouvez effectuer via ces options.

Options	Opérations
Create Sub-element et Create Attribute	Ajoute des éléments ou des attributs à l'arborescence XML importée. Consultez également la Section 6.3.1.5, « Ajouter un sous-élément ou un attribut à une structure XML » .
Set a namespace	Ajoute et gère des espaces de nommage donnés dans la structure XML importée. Consultez également la Section 6.3.1.7, « Gérer un espace de nommage » .
Delete	Supprime un élément ou un attribut. Consultez également la Section 6.3.1.6, « Supprimer un élément ou un attribut depuis la structure XML » .
Rename	Renomme un élément ou un attribut.
As loop element	Définit ou réinitialise un élément de boucle.

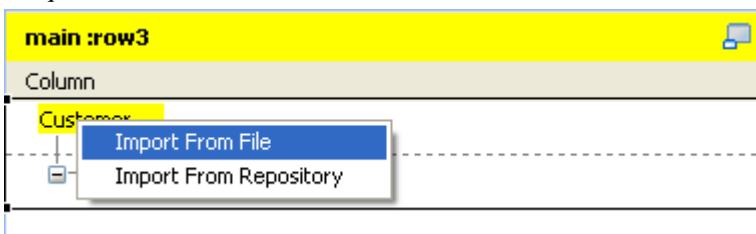
Options	Opérations
As group element	Dans la structure XML de sortie, définit un élément comme group element . Consultez également la Section 6.3.1.8, « Grouper les données de sortie »
As aggregate element	Dans la structure XML de sortie, Définit un élément aggregate . Consultez également la Section 6.3.1.9, « Agréger les données de sortie »

Les sections suivantes présentent de manière détaillée les processus de création d'arborescence XML.

6.3.1.2. Importer une structure XML à partir de fichiers XML

Pour importer une structure XML à partir d'un fichier, suivez la procédure suivante :

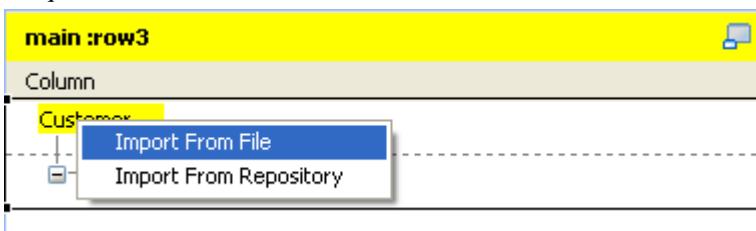
1. Dans la table d'entrée correspondante, cliquez-droit sur le nom de la colonne pour ouvrir le menu contextuel. Dans cet exemple, cliquez-droit sur la colonne *Customer*.



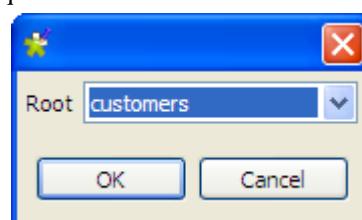
2. Dans le menu contextuel, sélectionnez l'option **Import From File**.
3. Dans la boîte de dialogue, parcourez votre système jusqu'au fichier XML que vous souhaitez utiliser pour fournir la structure XML, puis double-cliquez sur le fichier.

Pour importer la structure XML d'un fichier XSD, procédez comme suit :

1. Dans la table d'entrée correspondante, cliquez-droit sur le nom de la colonne pour ouvrir le menu contextuel. Dans cet exemple, cliquez-droit sur la colonne *Customer*.



2. Dans ce menu, sélectionnez **Import From File**.
3. Dans la boîte de dialogue, parcourez votre système jusqu'au fichier XSD que vous souhaitez utiliser pour fournir la structure XML, puis double-cliquez sur le fichier.
4. Dans la boîte de dialogue qui apparaît, sélectionnez un élément racine dans la liste **Root** pour être la racine de votre arborescence XML puis cliquez sur **OK**.





La racine de l'arborescence XML est adaptable :

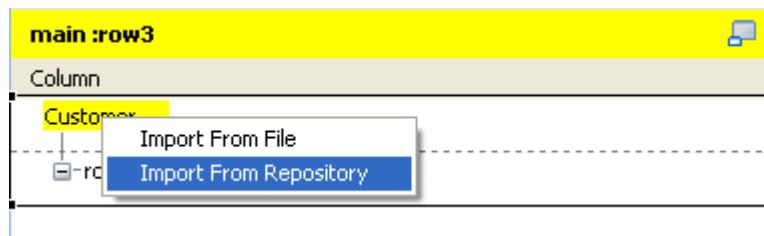
- Lorsque vous importez une structure XML d'entrée ou de sortie à partir d'un fichier XSD, vous pouvez choisir l'élément racine de votre arborescence XML.
- Une fois qu'une structure XML a été importée, la balise **root** est automatiquement renommée avec le nom de la source XML. Pour modifier le nom de la racine manuellement, vous devez utiliser l'éditeur de schéma. Pour plus d'informations concernant cet éditeur, consultez la [Section 6.3.3, « Utiliser le Tree schema editor »](#).

Puis spécifiez l'élément de boucle de cette structure XML. Pour plus d'informations sur le paramétrage de l'élément de boucle, consultez la [Section 6.3.1.4, « Définir ou réinitialiser un élément de boucle pour une structure XML créée »](#).

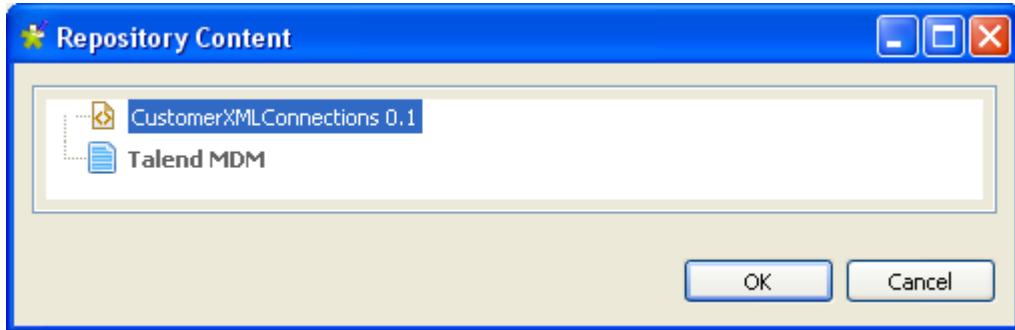
6.3.1.3. Importer une structure XML à partir du Repository

Pour importer une XML à partir d'une connexion du Repository, suivez la procédure suivante :

1. Dans la table d'entrée correspondante, cliquez-droit sur le nom de la colonne pour ouvrir le menu contextuel. Dans cet exemple, cliquez-droit sur la colonne *Customer*.



2. Dans le menu contextuel, sélectionnez l'option **Import From Repository**.
3. Dans la boîte de dialogue qui s'ouvre alors, sélectionnez la connexion XML ou MDM souhaitée pour importer la structure XML correspondante.



Cette figure vous montre un exemple de connexion XML importée du **Repository**.



Pour importer une arborescence XML du **Repository**, la connexion XML correspondante doit déjà avoir été créée. Pour plus d'informations concernant la création d'une connexion à un fichier XML dans le **Repository**, consultez la [Section 7.8, « Configurer un Schéma File XML »](#).

4. Cliquez sur **OK** pour valider la sélection.

La structure XML est créée et une boucle est automatiquement définie car la boucle avait déjà été spécifiée lors de la création de la métadonnée de connexion XML.

6.3.1.4. Définir ou réinitialiser un élément de boucle pour une structure XML créée

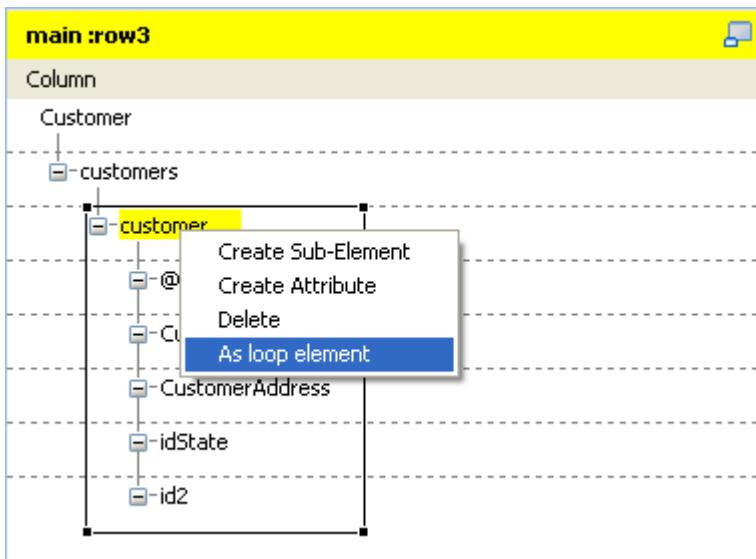
Vous devez définir un élément boucle pour tous les flux de données XML n'ayant pas d'élément boucle déjà défini. S'ils en ont déjà un, vous devez réinitialiser l'élément de boucle lorsque c'est nécessaire.



Un et seulement un élément boucle est nécessaire pour chaque flux de données XML dans le Map editor.

Pour définir un élément boucle, faites comme suit :

1. Dans la structure XML créée, cliquez-droit sur l'élément que vous souhaitez définir comme élément boucle. Par exemple, vous souhaitez définir le nœud *Customer* comme élément boucle.



2. Dans le menu contextuel, sélectionnez **As loop element** afin de définir l'élément sélectionné comme élément boucle.

Une fois ceci effectué, l'élément sélectionné se voit ajouter le texte suivant : **loop : true**.



Si vous fermez le **Map Editor** sans avoir défini les éléments de boucle nécessaires, comme expliqué



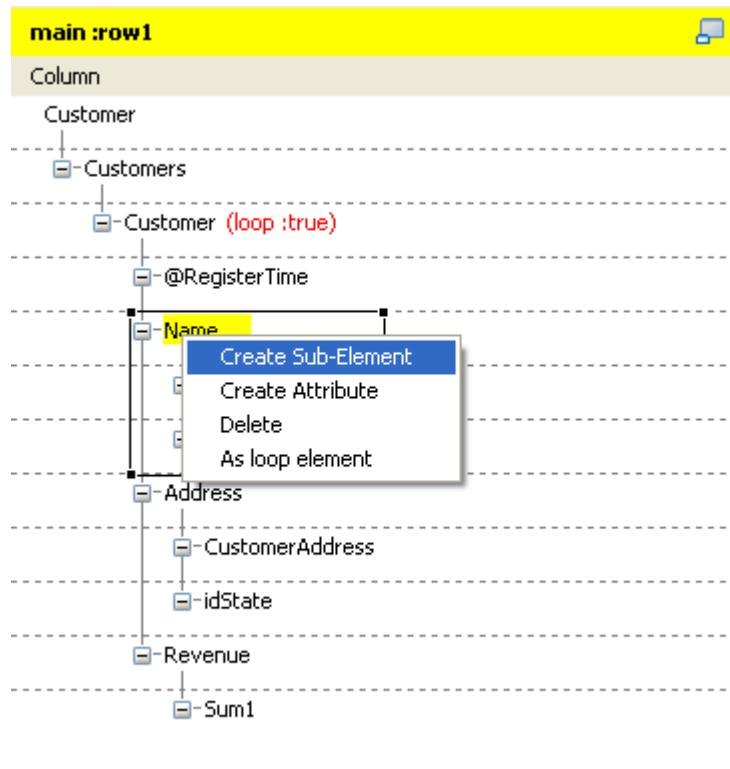
plus tôt dans ce scénario, une icône d'erreur apparaît dans le coin supérieur droit du composant **tXMLMap**, dans l'espace de modélisation graphique.

6.3.1.5. Ajouter un sous-élément ou un attribut à une structure XML

Dans la structure XML, vous pouvez ajouter manuellement un sous-élément ou un attribut à la racine ou à l'un des éléments.

Pour effectuer une de ces opérations, procédez comme suit :

1. Dans la structure que vous souhaitez modifier, cliquez-droit sur l'élément sur lequel vous souhaitez ajouter un sous-élément ou un attribut et sélectionnez **Create Sub-Element** ou **Create Attribute** en fonction de ce que vous souhaitez ajouter.



2. Dans l'assistant [Create New Element], saisissez le nom du sous-élément ou de l'attribut à ajouter.

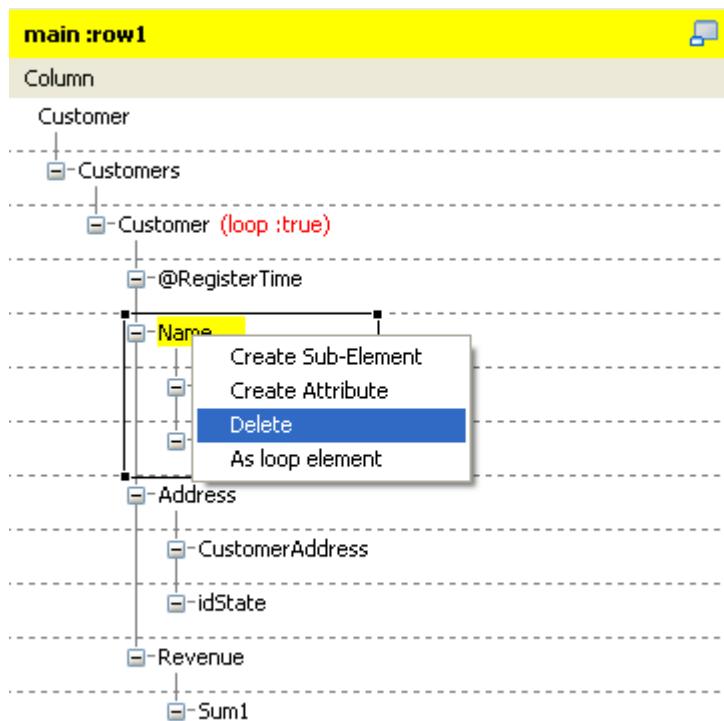


3. Cliquez OK pour valider la création du nouvel élément. Ce nouveau sous-élément ou attribut apparaît alors dans la structure XML.

6.3.1.6. Supprimer un élément ou un attribut depuis la structure XML

Pour supprimer un élément ou un attribut d'une structure XML existante, procédez comme suit :

1. Dans l'arborescence XML que vous souhaitez modifier, cliquez-droit sur l'élément ou l'attribut que vous souhaitez supprimer.



2. Dans le menu contextuel, sélectionnez **Delete**.

L'élément ou l'attribut sélectionné est supprimé, ainsi que tous les sous-éléments ou attributs qu'il contenait.

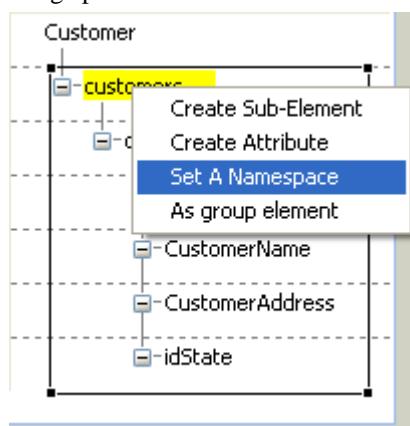
6.3.1.7. Gérer un espace de nommage

Vous pouvez définir et modifier un espace de nommage pour chacun des éléments des arborescences XML de ces flux d'entrée ou de sortie.

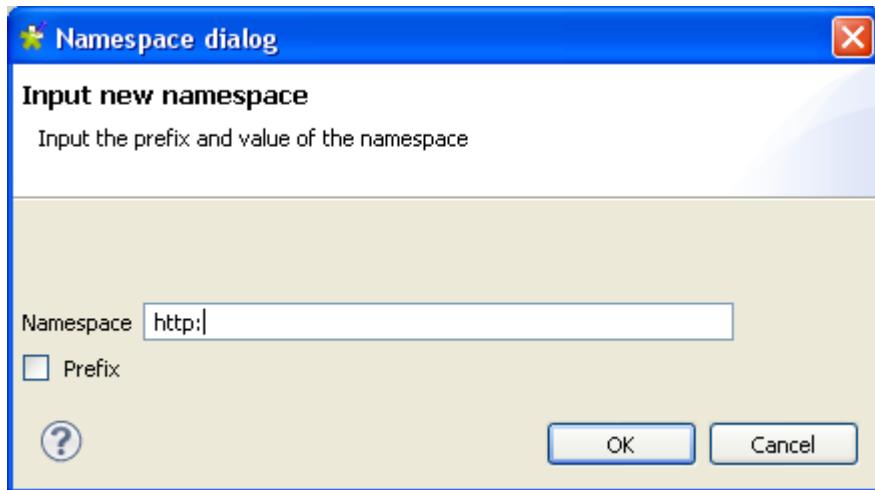
Définir un espace de nommage

Pour définir un espace de nommage :

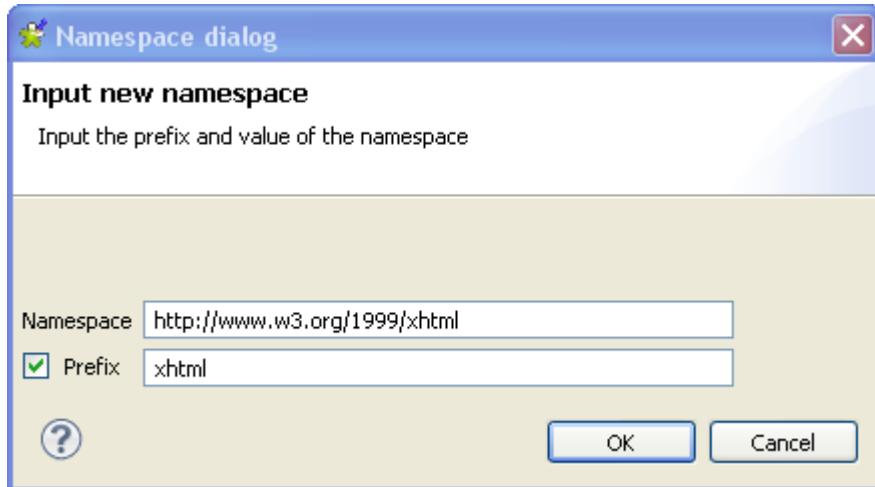
1. Dans la structure XML du flux d'entrée ou de sortie que vous souhaitez modifier, cliquez-droit sur l'élément auquel vous souhaitez associer un espace de nommage. Par exemple, dans l'arborescence XML de *Customer*, vous devez créer un espace de nommage pour le nœud racine.



2. Dans le menu contextuel, sélectionnez **Set A Namespace** et l'assistant [**Namespace dialog**] s'ouvre.
3. Dans cet assistant, saisissez l'URI à utiliser.



4. Si vous devez ajouter un préfixe à cet espace de nommage, cochez la case **Prefix** dans l'assistant et saisissez le préfixe à utiliser. Dans cet exemple, cochez la case et saisissez *xhtml*.

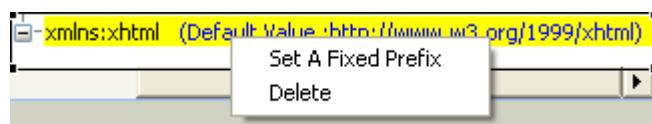


5. Cliquez sur **OK** pour valider l'espace de nommage.

Modifier la valeur par défaut d'un espace de nommage

Pour modifier la valeur par défaut d'un espace de nommage, procédez comme suit :

1. Dans l'arborescence XML contenant l'espace de nommage à modifier, cliquez-droit sur l'espace de nommage pour ouvrir le menu contextuel.



2. Dans le menu, sélectionnez **Set A Fixed Prefix** pour ouvrir l'assistant correspondant.
3. Saisissez la nouvelle valeur dans cet assistant.
4. Cliquez sur **OK** pour valider cette modification.

Supprimer un espace de nommage

Pour supprimer un espace de nommage, procédez comme suit :

1. Dans l'arborescence XML contenant l'espace de nommage, cliquez-droit sur cet espace de nommage pour ouvrir le menu contextuel.



2. Dans le menu, cliquez sur **Delete** pour valider cette suppression.

6.3.1.8. Grouper les données de sortie

Le composant **tXMLMap** utilise un "**group element**" pour regrouper les données de sortie selon certaines conditions données. Cela vous permet d'entourer de balises "**group element**" les éléments répondant à la condition.

Pour définir un **group element**, deux restrictions doivent être respectées :

1. le nœud racine ne peut être défini en tant que **group element** ;
2. le **group element** doit être défini sur l'élément dont le sous-élément de boucle dépend directement.



L'option de **group element** est visible à partir du moment où vous avez défini un élément de boucle. Cette option est également invisible lorsqu'un élément n'est pas autorisé à être un élément de groupe.

Une fois le **group element** défini, tous ses sous-éléments sauf la boucle sont utilisés comme conditions pour regrouper les données de sortie.

Vous devez créer avec soin l'arborescence XML pour une utilisation optimale d'un **group element** donné. Pour plus d'informations concernant l'utilisation d'un **group element**, consultez le **tXMLMap** dans le **Guide de référence des Composants Talend Open Studio**.



Le **tXMLMap** propose des éléments "**group**" et "**aggregate**" afin de classer les données dans la structure XML. Lorsque vous gérez une ligne de données XML, la différence de comportement est la suivante :

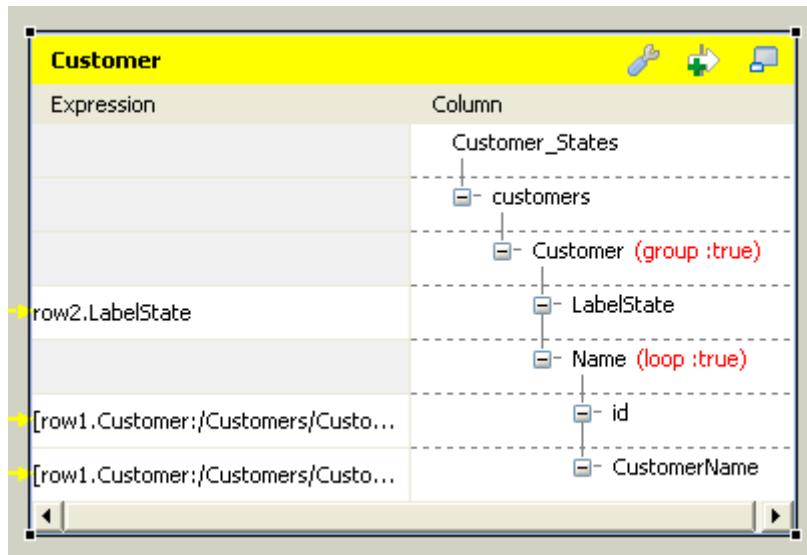
- L'élément "**group**" traite toujours les données en un seul flux.
- L'élément "**aggregate**" sépare ce flux en différents flux XML complets.

Définir un group element

Pour définir un **group element**, procédez comme suit :

1. Dans l'arborescence XML de la sortie dans le **Map editor**, cliquez-droit sur l'élément que vous voulez définir comme **as group element**.
2. Dans le menu contextuel qui s'ouvre, sélectionnez **As group element**.

Cet élément de la sélection devient le **group element**. La capture d'écran vous montre un exemple de l'arborescence XML avec le **group element**.



Révoquer un group element défini

Pour révoquer un **group element** défini, procédez comme suit :

1. Dans l'arborescence XML de la sortie dans le **Map editor**, cliquez-droit sur l'élément que vous avez défini comme **group element**.
2. Dans le menu contextuel, sélectionnez **Remove group element**.

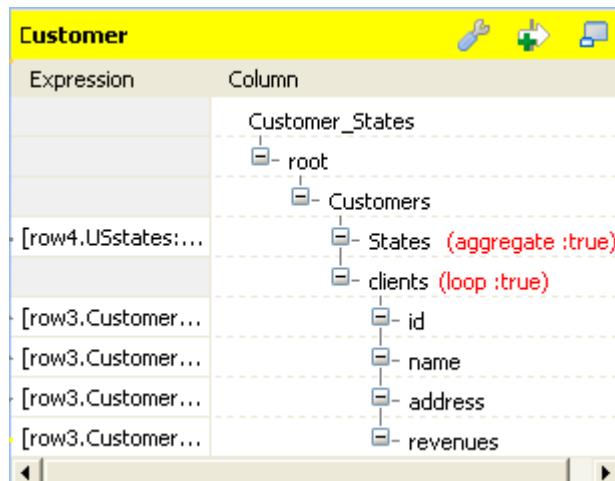
Le **group element** défini est révoqué.

6.3.1.9. Agréger les données de sortie

Avec le **tXMLMap**, vous pouvez définir autant d'éléments "**aggregate**" que nécessaire dans l'arborescence XML de sortie afin de classer les données XML. Ce composant écrit les données classées, chaque classification devenant un flux XML complet.

1. Pour définir un élément en tant qu'élément "**aggregate**", cliquez-droit sur l'élément concerné dans l'arborescence XML du côté droit du **Map editor**, puis, dans le menu contextuel, sélectionnez **As aggregate element**.

Cet élément devient l'élément "**aggregate**". Un texte lui est ajouté : **aggregate : true**.



2. Pour révoquer la définition de l'élément "**aggregate**", cliquez-droit sur cet élément puis sélectionnez **Remove aggregate element**, dans le menu contextuel.

 Pour définir un élément "**aggregate**", assurez-vous que cet élément n'a pas d'enfant et que la fonctionnalité **All in one** est désactivée. L'option **As aggregate element** est disponible dans le menu contextuel si les deux conditions sont respectées. Pour plus d'informations concernant la fonctionnalité **All in one**, consultez la [Section 6.3.2.1, « Ecrire dans un document les éléments en sortie »](#).

Pour un exemple d'utilisation de l'élément "**aggregate**" avec le **tXMLMap**, consultez le [Guide de référence Composants Talend Open Studio](#).

 Le composant **tXMLMap** propose les éléments "**group**" et "**aggregate**" pour classer les données dans une structure XML. Lors de la gestion d'une ligne de données (flux XML complet), la différence de comportement est la suivante :

- L'élément "**group**" traite toujours les données dans un flux unique.
- L'élément "**aggregate**" sépare ce flux en différents flux XML complets.

6.3.2. Définir le mode de sortie

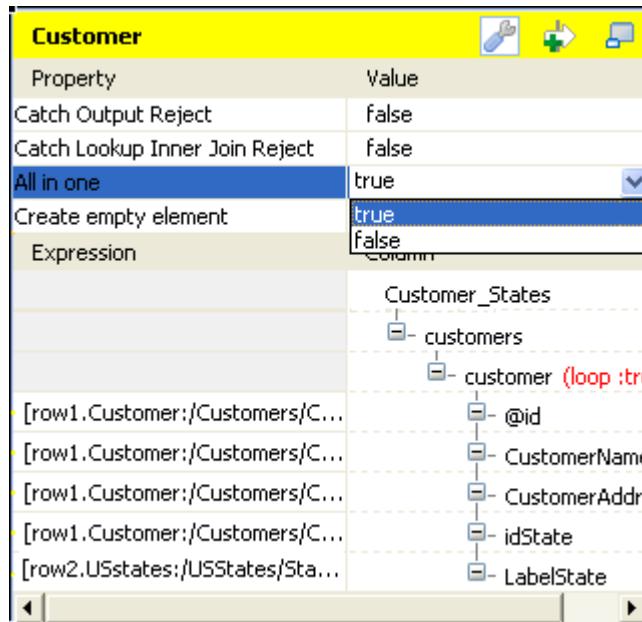
Pour définir le mode de sortie des données de type **Document**, vous devez regrouper tous les éléments XML en un seul flux XML, et, lorsqu'un élément vide existe, choisir où les écrire. Ainsi, vous ne modifiez pas la structure de l'arborescence XML créée.

6.3.2.1. Ecrire dans un document les éléments en sortie

Sauf si vous utilisez l'élément "**aggregate**" qui sépare toujours un flux XML, vous devez être capable de déterminer si un flux XML est écrit en un seul flux ou dans des flux séparés, à l'aide de la fonctionnalité **All in one** de l'éditeur du **tXMLMap**.

Pour ce faire, dans la partie droite du **Map editor**, procédez comme suit :

1. Cliquez sur l'icône de clé anglaise afin d'ouvrir le panneau de paramétrage du mapping. La capture d'écran suivante vous montre un exemple.



2. Cliquez sur le champ **All in one** et, dans la liste déroulante, sélectionnez **true** ou **false** afin de choisir si le flux XML doit être écrit en un flux unique.

- Si vous sélectionnez **true**, les données XML sont écrites en un seul flux. Dans cet exemple, le flux unique s'affiche comme suit :

```
Starting job tXMLMap at 10:16 09/11/2011.

[statistics] connecting to socket on port 3643
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="1"><CustomerName>Griffith Paving
and
Sealcoatin</CustomerName><CustomerAddress>talend@apres91</C
ustomerAddress><idState>7</idState><LabelState>Connecticut<
/LabelState></customer><customer
id="56"><CustomerName>Glenn Oaks Office
Supplies</CustomerName><CustomerAddress>1859 Green Bay
Rd.</CustomerAddress><idState>7</idState><LabelState>Conne
cticut</LabelState></customer><customer
id="2"><CustomerName>Bill's Dive
Shop</CustomerName><CustomerAddress>511 Maple Ave. Apt.
1B</CustomerAddress><idState>35</idState><LabelState>Ohio<
/LabelState></customer><customer id="61"><CustomerName>DBN
Bank</CustomerName><CustomerAddress>456 Grossman
Ln.</CustomerAddress><idState>35</idState><LabelState>Ohio<
/LabelState></customer><customer
id="63"><CustomerName>Pivot Point
College</CustomerName><CustomerAddress>1547 Knolwood
Rd.</CustomerAddress><idState>9</idState><LabelState>Flori
da</LabelState></customer></customers>
[statistics] disconnected
Job tXMLMap ended at 10:16 09/11/2011. [exit code=0]
```

La structure de ce flux est la suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
    <customer id="1">
        <CustomerName>Griffith Paving and Sealcoatin</CustomerName>
        <CustomerAddress>talend@apres91</CustomerAddress>
        <idState>7</idState>
        <LabelState>Connecticut</LabelState>
    </customer>
    <customer id="56">
        <CustomerName>Glenn Oaks Office Supplies</CustomerName>
        <CustomerAddress>1859 Green Bay Rd.</CustomerAddress>
        <idState>7</idState>
        <LabelState>Connecticut</LabelState>
    </customer>
    <customer id="2">
        <CustomerName>Bill's Dive Shop</CustomerName>
        <CustomerAddress>511 Maple Ave. Apt. 1B</CustomerAddress>
        <idState>35</idState>
        <LabelState>Ohio</LabelState>
    </customer>
    <customer id="61">
        <CustomerName>DBN Bank</CustomerName>
        <CustomerAddress>456 Grossman Ln.</CustomerAddress>
        <idState>35</idState>
        <LabelState>Ohio</LabelState>
    </customer>
    <customer id="63">
        <CustomerName>Pivot Point College</CustomerName>
        <CustomerAddress>1547 Knolwood Rd.</CustomerAddress>
        <idState>9</idState>
        <LabelState>Florida</LabelState>
    </customer>
</customers>
```

- Si vous sélectionnez **false**, les données XML sont écrites dans des flux séparés, chaque boucle représentant un flux. Dans cet exemple, les flux sont les suivants :

```
Starting job tXMLMap at 10:25 09/11/2011.

[statistics] connecting to socket on port 4036
[statistics] connected
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="1"><CustomerName>Griffith Paving
and
Sealcoatin</CustomerName><CustomerAddress>talend@apres91</C
ustomerAddress><idState>7</idState><LabelState>Connecticut<
/LabelState></customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="56"><CustomerName>Glenn Oaks
Office Supplies</CustomerName><CustomerAddress>1859 Green
Bay
Rd.</CustomerAddress><idState>7</idState><LabelState>Conne
cticut</LabelState></customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="2"><CustomerName>Bill's Dive
Shop</CustomerName><CustomerAddress>511 Maple Ave. Apt.
1B</CustomerAddress><idState>35</idState><LabelState>Ohio<
/LabelState></customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="61"><CustomerName>DBN
Bank</CustomerName><CustomerAddress>456 Grossman
Ln.</CustomerAddress><idState>35</idState><LabelState>Ohio<
/LabelState></customer></customers>
<?xml version="1.0" encoding="UTF-8"?>
<customers><customer id="63"><CustomerName>Pivot Point
College</CustomerName><CustomerAddress>1547 Knolwood
Rd.</CustomerAddress><idState>9</idState><LabelState>Flori
da</LabelState></customer></customers>
```

Chaque flux contient une structure XML complète. La structure du premier flux, par exemple est la suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
    <customer id="1">
        <CustomerName>Griffith Paving and Sealcoatin</CustomerName>
        <CustomerAddress>talend@apres91</CustomerAddress>
        <idState>7</idState>
        <LabelState>Connecticut</LabelState>
    </customer>
</customers>
```



La fonctionnalité **All in one** est désactivée si vous utilisez l'élément "**aggregate**". Pour plus d'informations concernant l'élément "**aggregate**", consultez la [Section 6.3.1.9, « Agréger les données de sortie »](#).

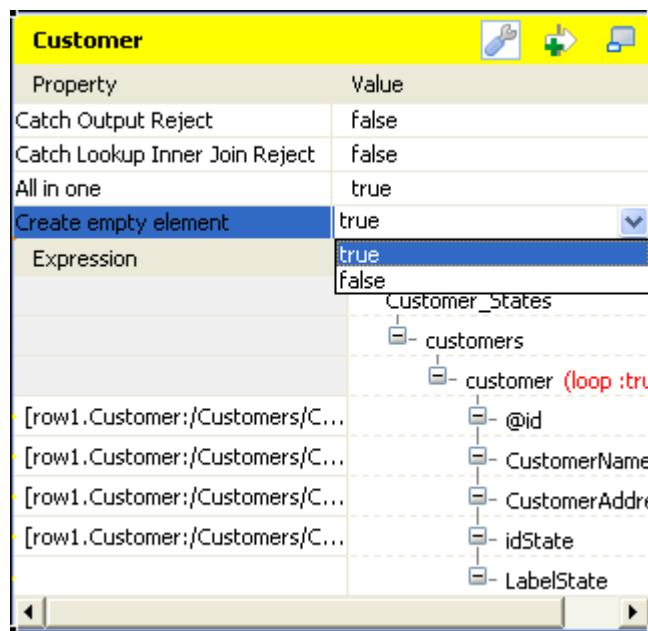
6.3.2.2. Gérer les éléments vides dans le Map editor

Il peut être nécessaire de créer des éléments de sortie vides durant le processus de transformation des données en un flux XML, par exemple quand le **tXMLMap** est associé à un **tWriteXMLField** créant des éléments vides ou lorsqu'il n'y a pas de colonne associée à un certain noeud XML dans le flux de données XML de sortie.

Vous pouvez garder ces éléments vides dans l'arborescence XML selon vos besoins, même si vous ne souhaitez pas les écrire en sortie.

Le composant **tXMLMap** vous permet de définir le booléen pour la création de l'élément vide. pour ce faire, dans la partie droite du **Map editor**, effectuez les opérations suivantes :

1. Cliquez sur la clé anglaise afin d'ouvrir le panneau de paramétrage du mapping.



- Dans le panneau, cliquez sur le champ **Create empty element** et, dans la liste déroulante, sélectionnez **true** ou **false** afin de choisir d'écrire ou non l'élément vide.
 - Si vous sélectionnez **true**, l'élément vide est créé dans le flux XML de sortie et écrit, par exemple, comme suit `<customer><LabelState/></customer>`.
 - Si vous sélectionnez **false**, l'élément vide n'est pas écrit.

6.3.3. Utiliser le Tree schema editor

En plus des vues **Schema editor** et **Expression editor** proposées par l'éditeur du **tMap**, l'éditeur du composant **tXMLMap** fournit la vue **Tree schema editor** permettant de modifier le schéma de l'arborescence XML des flux d'entrée et de sortie.

Pour accéder au **Tree schema editor**, cliquez sur l'onglet correspondant en bas de l'éditeur.

XPath	Key	Type	Nullable	Pattern
row1.Customer:/Cust...		String		

XPath
Customer.newColumn:/customers
Customer.newColumn:/customers/customer
Customer.newColumn:/customers/customer/@id
Customer.newColumn:/customers/customer/Custo
Customer.newColumn:/customers/customer/Custo
Customer.newColumn:/customers/customer/Custo

Le schéma de l'arborescence XML du flux d'entrée apparaît à gauche de l'onglet **Tree schema editor**, et celui du flux de sortie apparaît à droite.

Le tableau suivant fournit des informations concernant cet éditeur de schéma.

Métadonnée	Description
XPath	Le chemin absolu pointant sur chaque élément ou attribut de l'arborescence XML et éditez le nom de l'élément ou attribut correspondant.

Métadonnée	Description
Key	La clé indique si la valeur de la clé d'expression devrait être utilisée pour retrouver les données d'une autre table à travers un lien Join. Si cette case est décochée, la relation Join est désactivée.
Type	Type de données : String, Integer, Document, etc.  Cette colonne est obligatoire.
Nullable	Indique si les valeurs de champs nulles sont autorisées.
Pattern	Indique si un modèle est utilisé pour les données de type Date.

 Les schémas d'entrée et de sortie sont indépendants les uns des autres. Vous pouvez par exemple changer le libellé d'une colonne de sortie sans que le libellé d'entrée ne change.

Cependant, toute modification apportée au schéma est immédiatement reportée sur le schéma correspondant dans la zone Input ou Output correspondante dans l'éditeur du **tXMLMap**, mais aussi au niveau des propriétés elles-mêmes des composants Input et Output concerné.

Pour des scénarios détaillés concernant les nombreuses opérations que vous pouvez effectuer à l'aide du **tXMLMap**, consultez le **Guide de référence** des *Composants Talend Open Studio*.

Chapitre 7. Gestion des métadonnées

Dans *Talend Open Studio for Data Integration*, 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 for Data Integration*. Pour plus d'informations, consultez l'[Annexe A, Interface de Talend Open Studio for Data Integration](#).

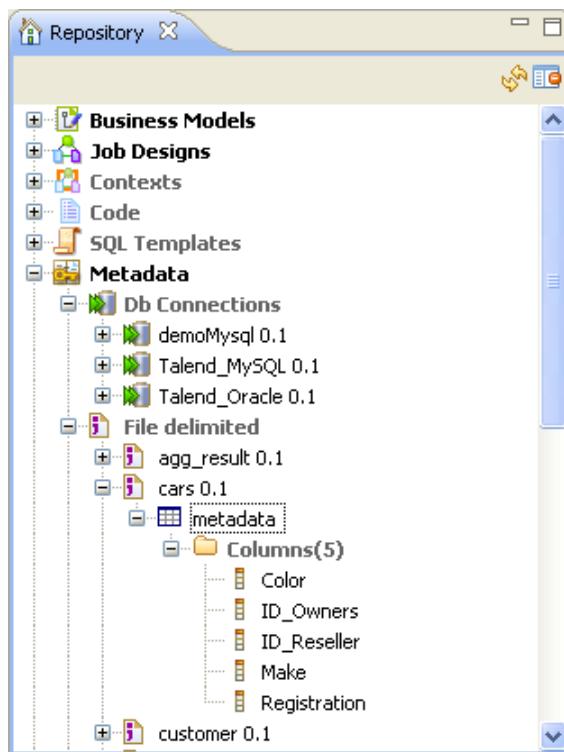
7.1. Objectifs

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 paramétriser 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 for Data Integration*.

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 for Data Integration*, il est possible de créer, entre autres :

- des connexions à des bases de données,
- un schéma JDBC,
- une connexion SAS,
- un schéma de type Fichier (File),
- un schéma LDAP,
- un schéma Salesforce,
- un schéma générique,
- une connexion MDM,
- un schéma WSDL (Services Web),
- une connexion FTP,

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 nœud **Metadata** de la vue **Repository**.

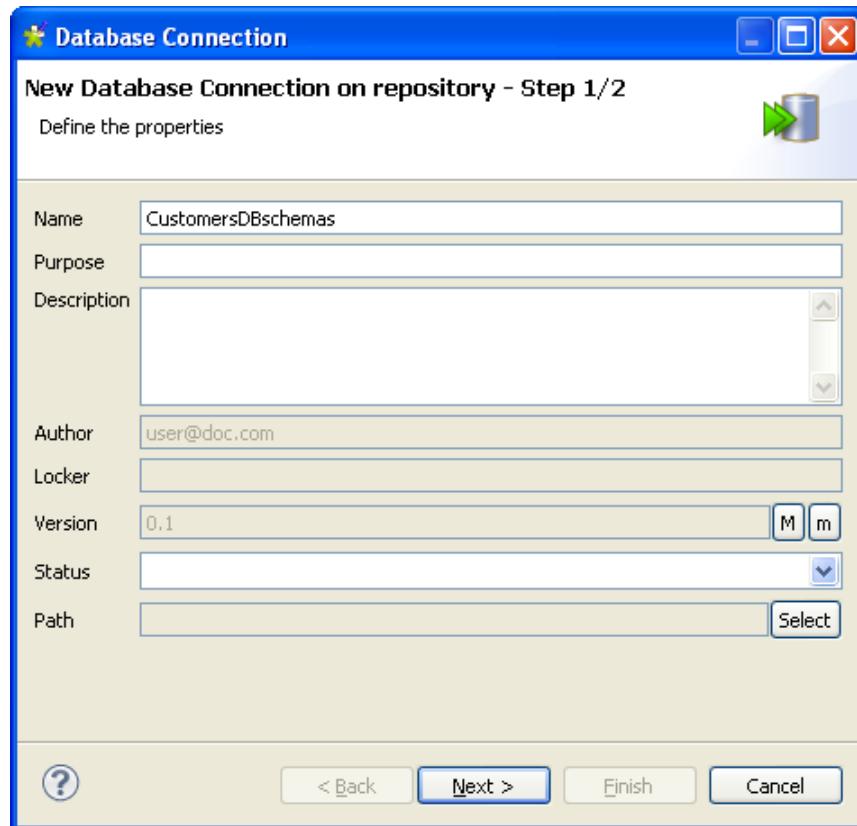


Vous pouvez également configurer cette connexion DB en cliquant sur l'icône dans la vue **Basic settings** de tous les composants DB input et output.

7.2.1. Etape 1 : Propriétés générales

Pour créer une connexion à une base de données, suivez la procédure décrite ci-dessous. Développez le nœud **Metadata** dans la vue **Repository**, cliquez-droit sur **Db Connections** puis sélectionnez **Create connection** dans le menu contextuel.

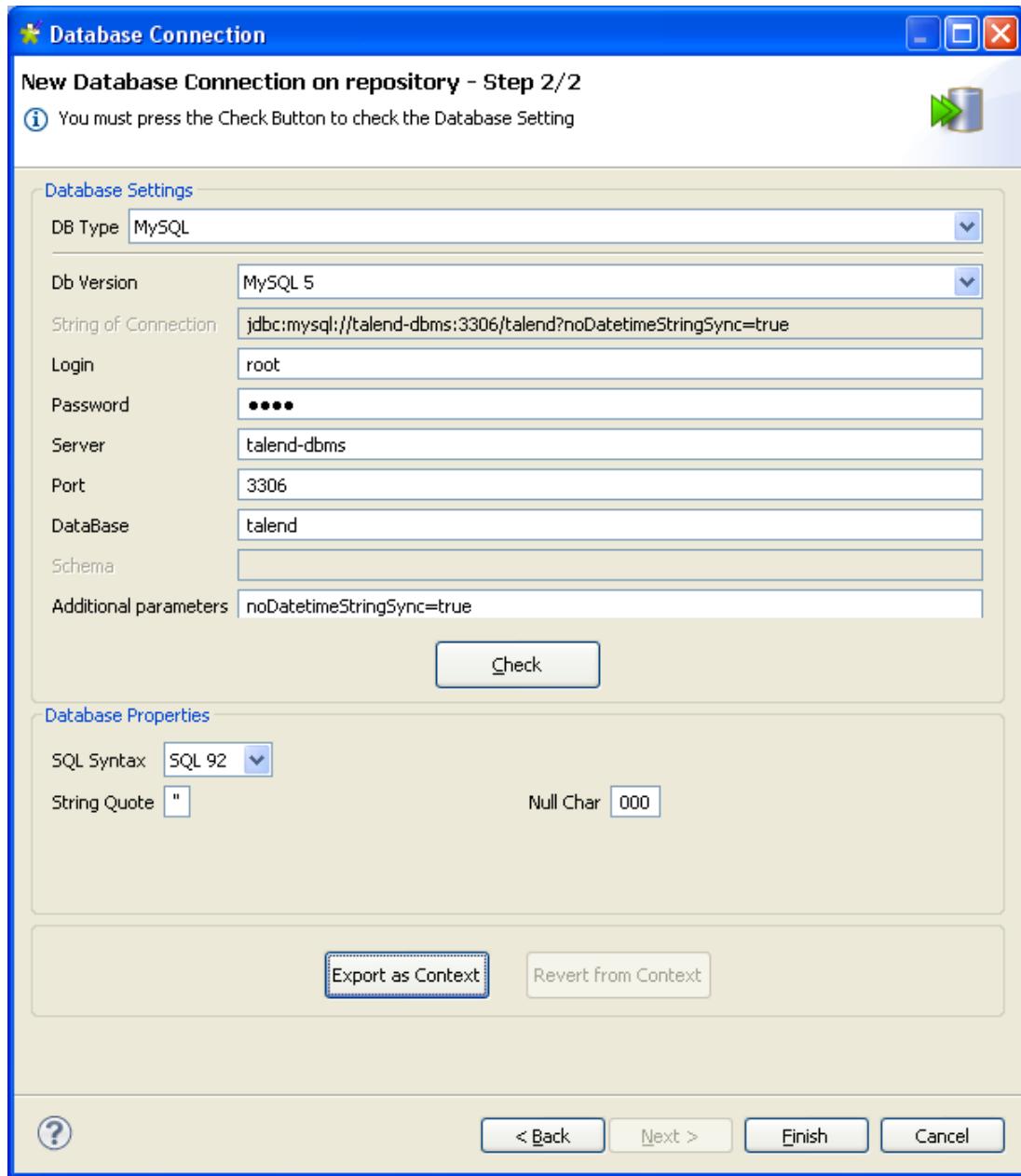
Un assistant de connexion s'ouvre. Renseignez les propriétés générales du schéma, notamment les champs **Name** et **Description**. Le champ **Status** est un champ personnalisé que vous pouvez définir dans les Préférences (**Window > Preferences**).



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 **DB 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.



 Lors de la création d'une connexion à des bases de données telles que AS400, HSQDB, Informix, MsSQL, MySQL, Oracle, Sybase, ou Teradata, vous pouvez spécifier des propriétés de connexion supplémentaires dans le champ **Additional parameters** de la zone **Database settings**.

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

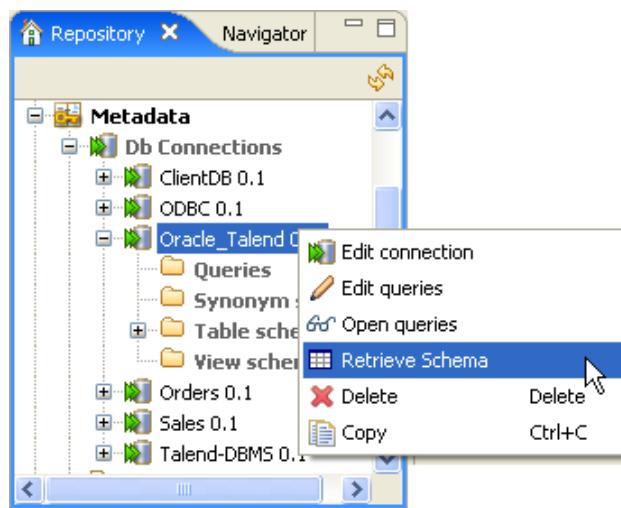
 Pour pouvoir retrouver les schémas de toutes les tables dans la base de données :

-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.



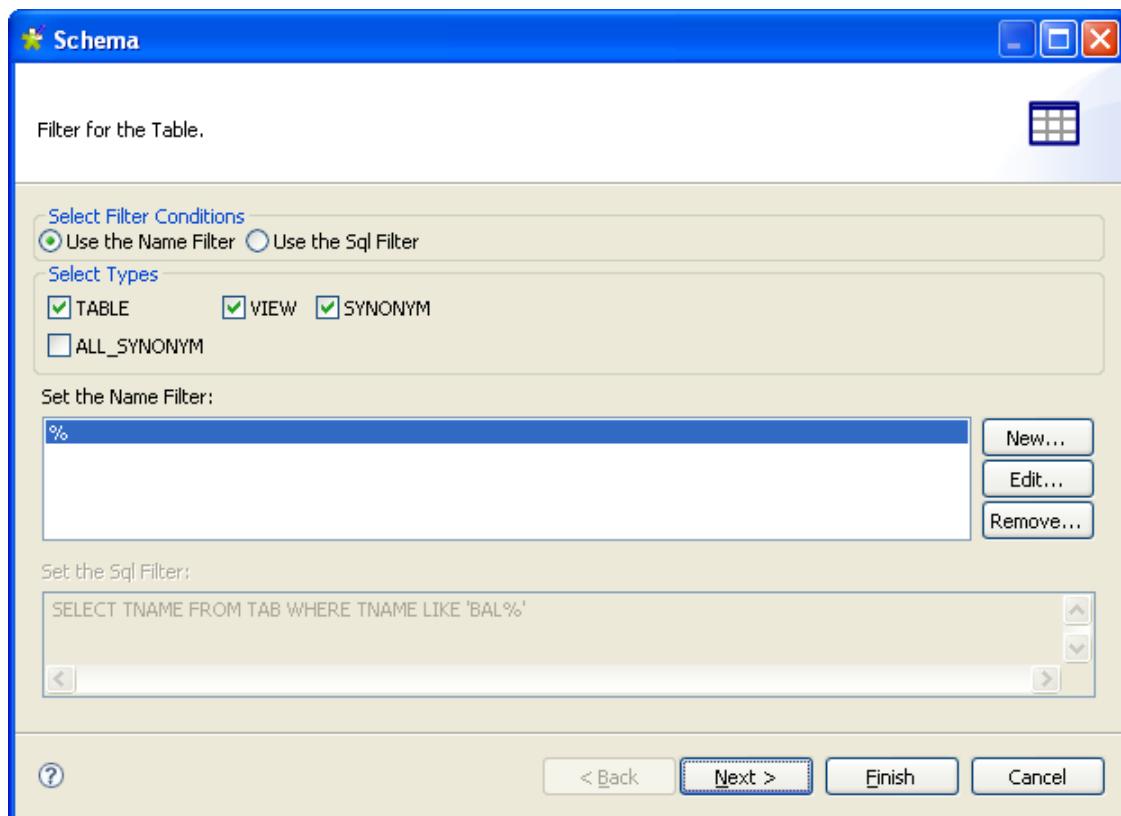
A présent, cliquez droit sur la connexion nouvellement créée et sélectionnez **Retrieve schema** dans le menu contextuel afin de charger le schéma de la table, dans la connexion établie.

- 💡 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 à la base de données.

- 💡 Pour l'instant, l'option des synonymes fonctionne avec Oracle, DB2 et MSSQL uniquement.



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**.

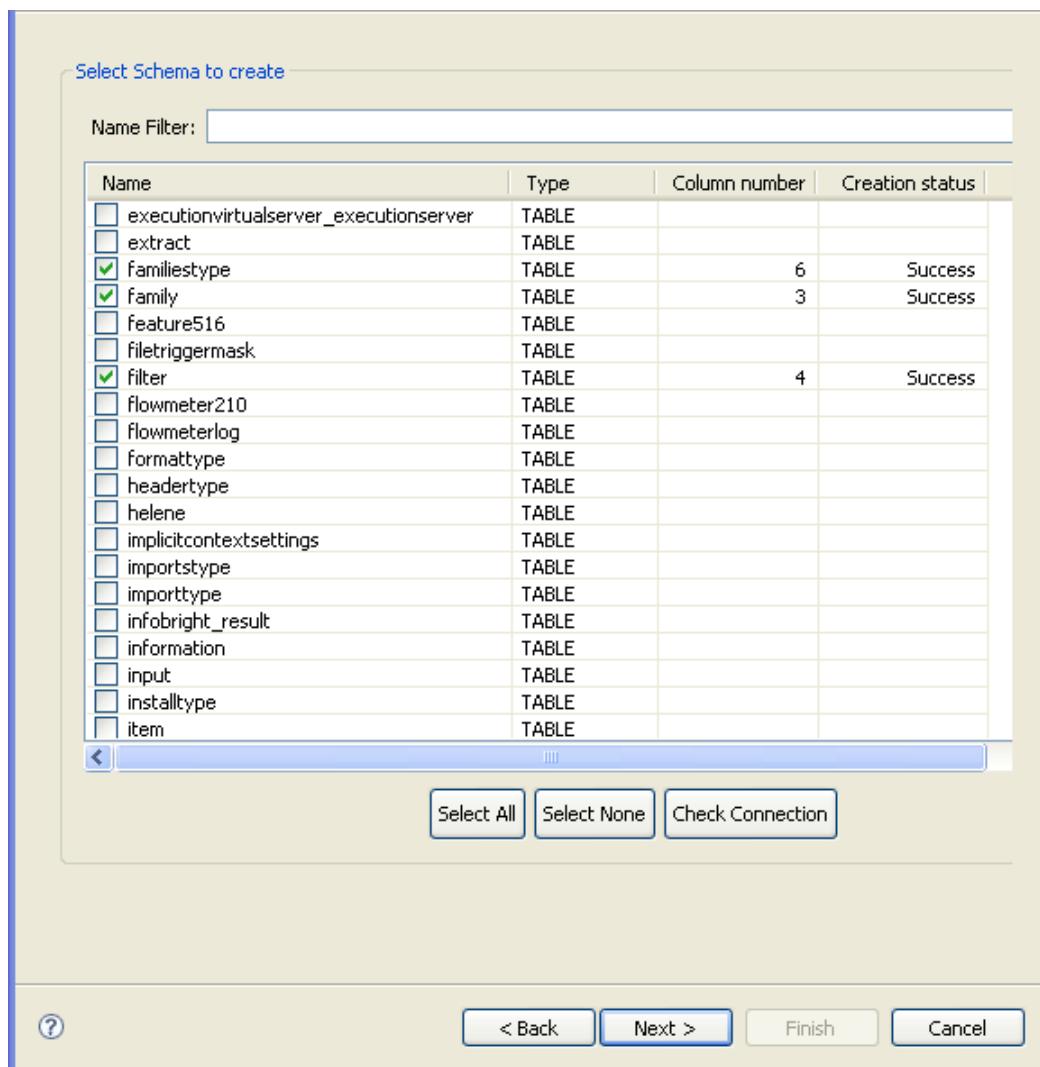
Pour filtrer les objets d'une base de données par leur nom, observez les étapes suivantes :

- 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.

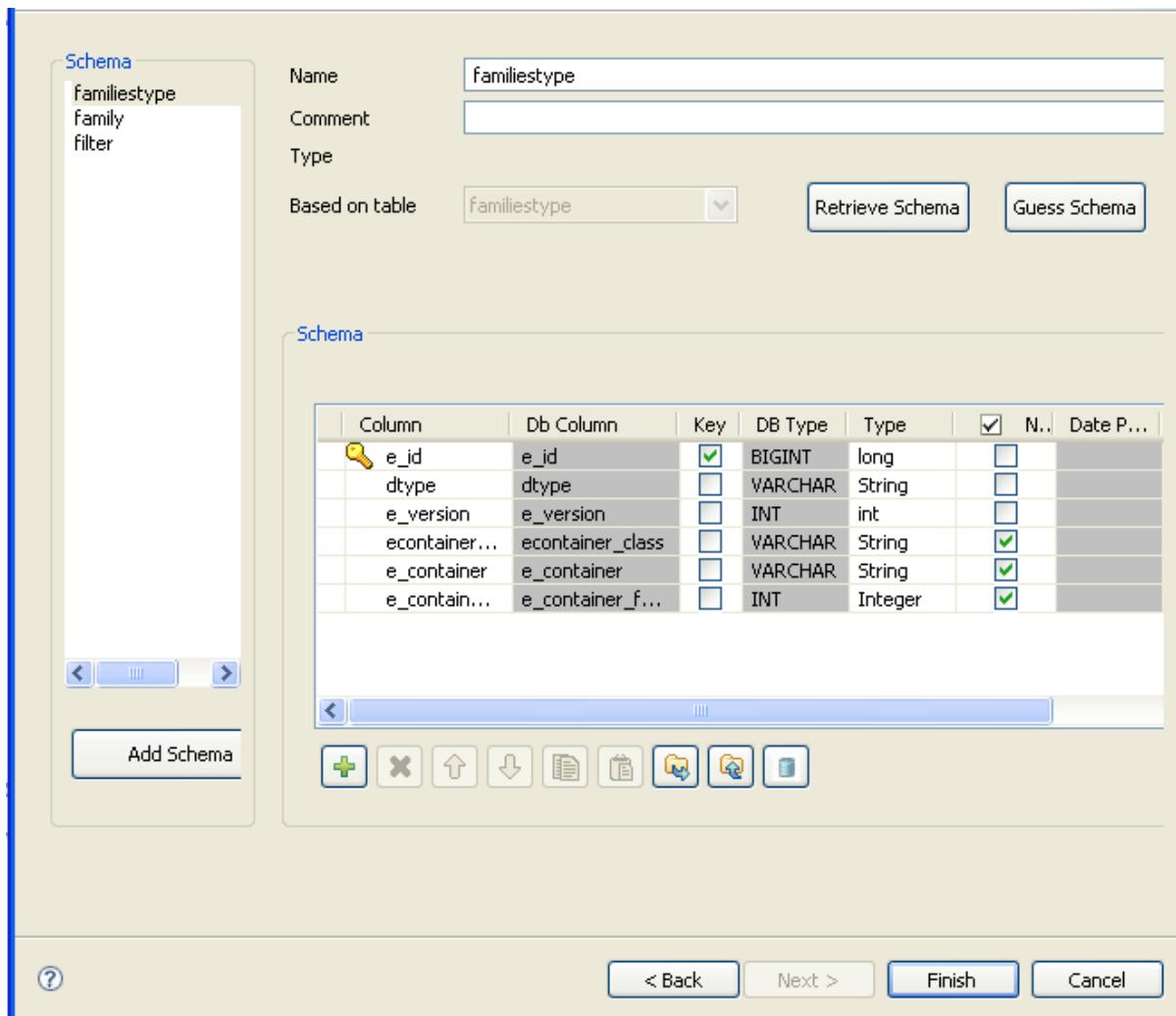
Pour filtrer les objets à l'aide d'une requête SQL :

- 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://docs.oracle.com/javase/6/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 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.

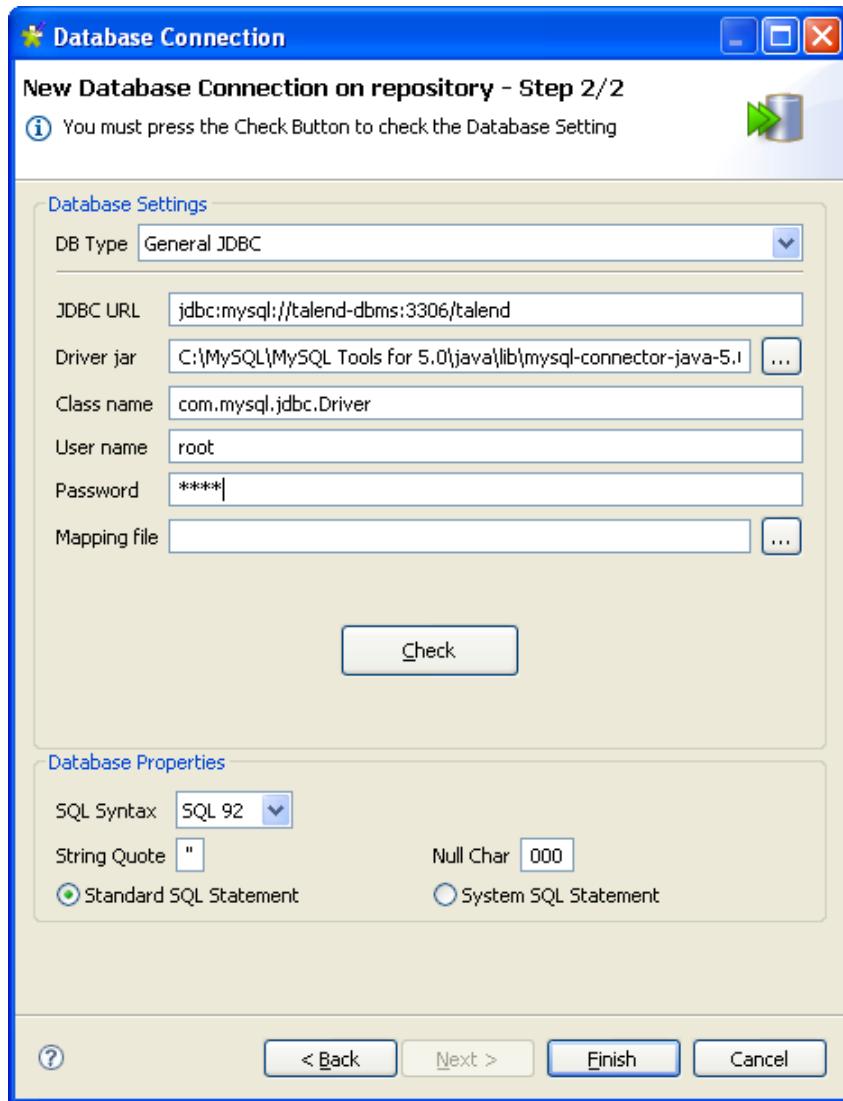
7.3.1. Etape 1 : Propriétés générales

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

Pour plus d'informations, consultez [Section 7.2.1, « Etape 1 : Propriétés générales »](#).

7.3.2. 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 :

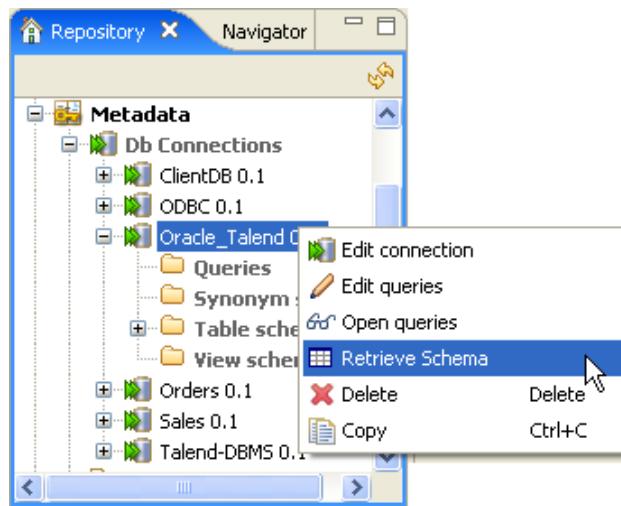
- Dans le champ **JDBC URL**, renseignez l'URL de connexion au serveur SGBD.
- Dans le champ **Driver jar**, sélectionnez le pilote jar permettant de valider votre connexion à la base de données.
- Dans le champ **Class name**, renseignez la classe principale du pilote permettant de communiquer avec 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. Par exemple : le type VARCHAR pour une base de données correspond au type String en Java.

 Les fichiers de mapping sont au format XML et sont définis dans le menu **Window > Preferences** sous le nœud **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.

7.3.3. 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://docs.oracle.com/javase/6/docs/api/index.html>.

7.3.4. 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.

7.4. Configurer une connexion SAS

Talend Open Studio for Data Integration vous permet de configurer une connexion à un système distant SAS.

7.4.1. 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.

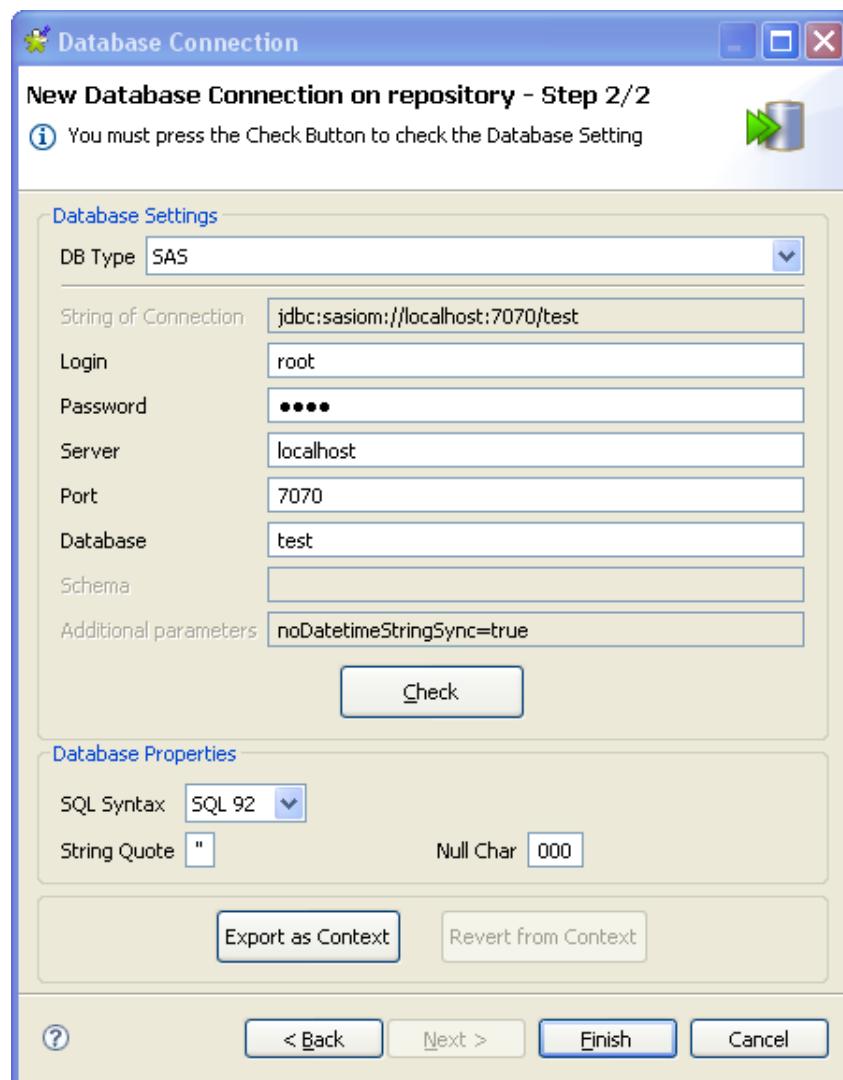
7.4.2. Etape 1 : Propriétés générales

1. Dans l’arborescence **Repository** de *Talend Open Studio for Data Integration*, développez le nœud **Metadata** puis cliquez droit sur **DB Connection**.
2. Sélectionnez **Create connection** dans le menu contextuel pour ouvrir l’assistant [**Database Connection**].
3. 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 [Section 7.2.1, « Etape 1 : Propriétés générales »](#).

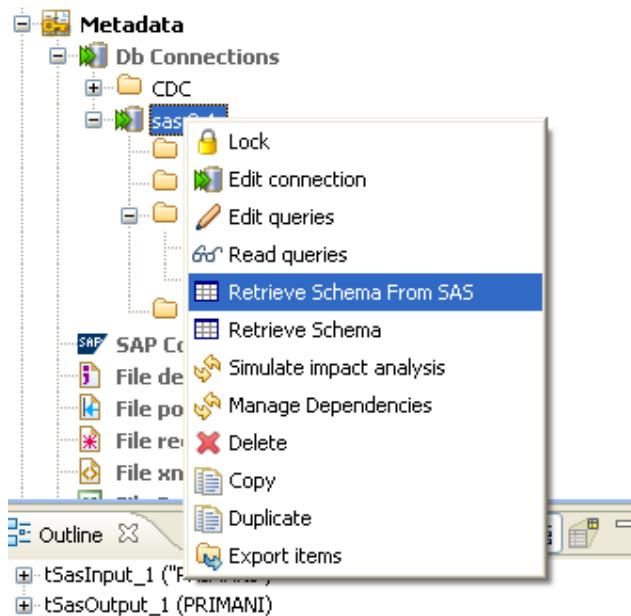
7.4.3. Etape 2 : Connexion

- 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 nœud **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.



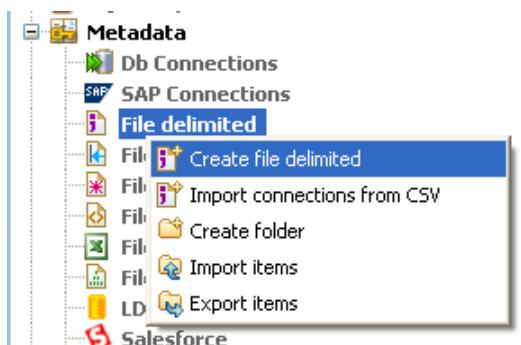
- Cliquez droit sur la connexion SAS nouvellement créée et sélectionnez **Retrieve Schema From SAS** dans le menu contextuel pour afficher tous les schémas de la base de données dans le sous-dossier **Table schemas**.

7.5. Configurer un schéma File Delimited

Les métadonnées de type File Delimited peuvent être utilisées pour définir les propriétés des composants **tInput-FileDelimited** et **tInputFileExcel** 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 (Expressions 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.

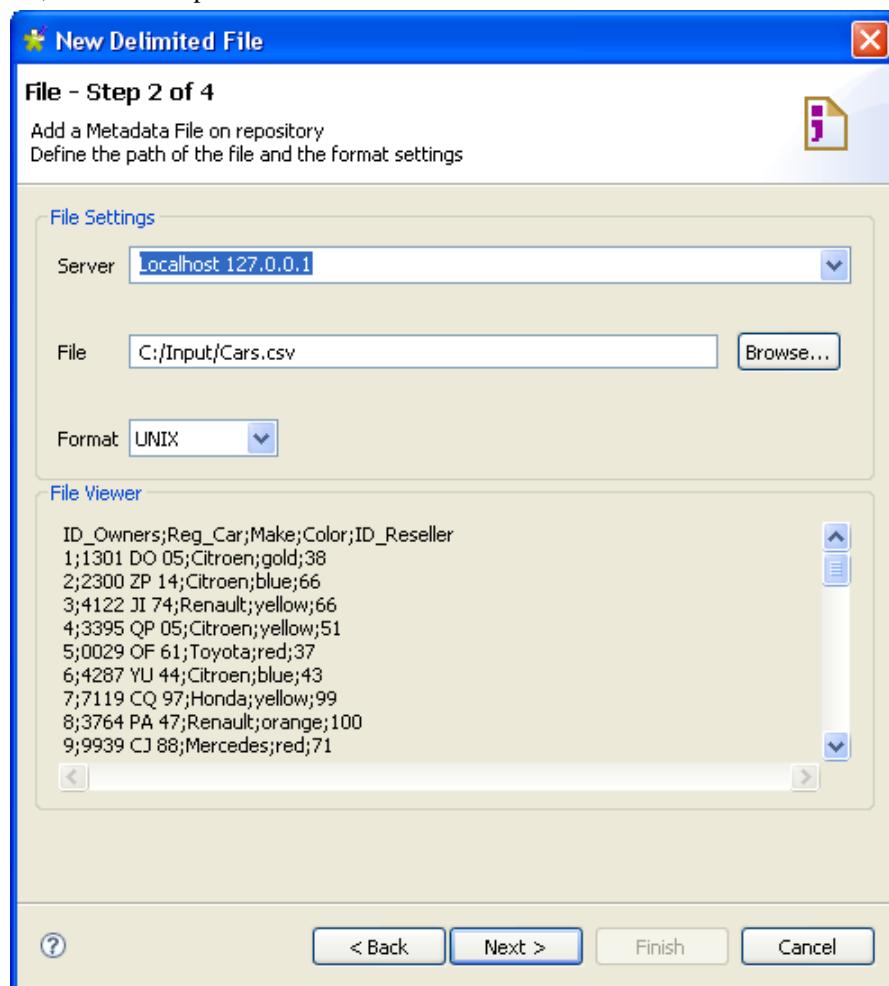
7.5.1. Etape 1 : Propriétés générales

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

Pour plus d'informations, consultez [Section 7.2.1, « Etape 1 : Propriétés générales »](#), de la section *Configurer une connexion DB*.

7.5.2. 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**.



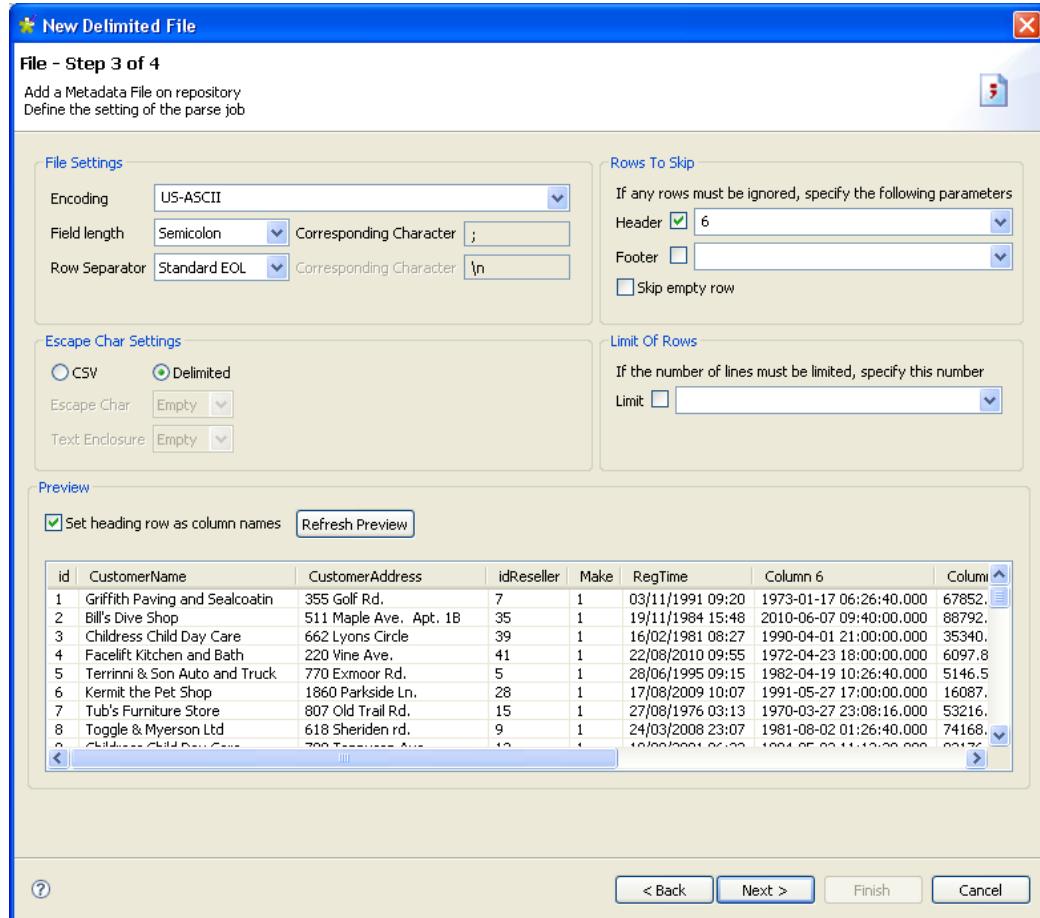
Sélectionnez le **Format** du système d'exploitation dans lequel le fichier a été créé. Cette information est utilisée pour pré-remplir les champs des étapes suivantes. Si la liste ne propose pas le format approprié, ignorez ce champ.

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

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

7.5.3. 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.

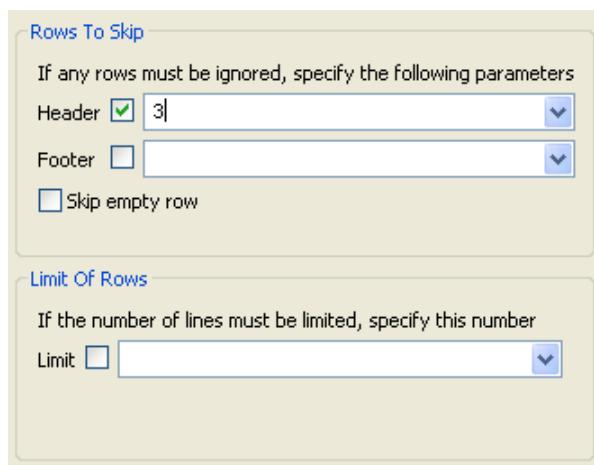


Paramétrez l'**Encoding**, ainsi que les séparateurs de champs et de lignes (**Field separator** et **Row separator**) dans les paramètres Delimited File.



En fonction de votre type de fichier (csv ou delimited), vous pouvez également paramétrier 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.

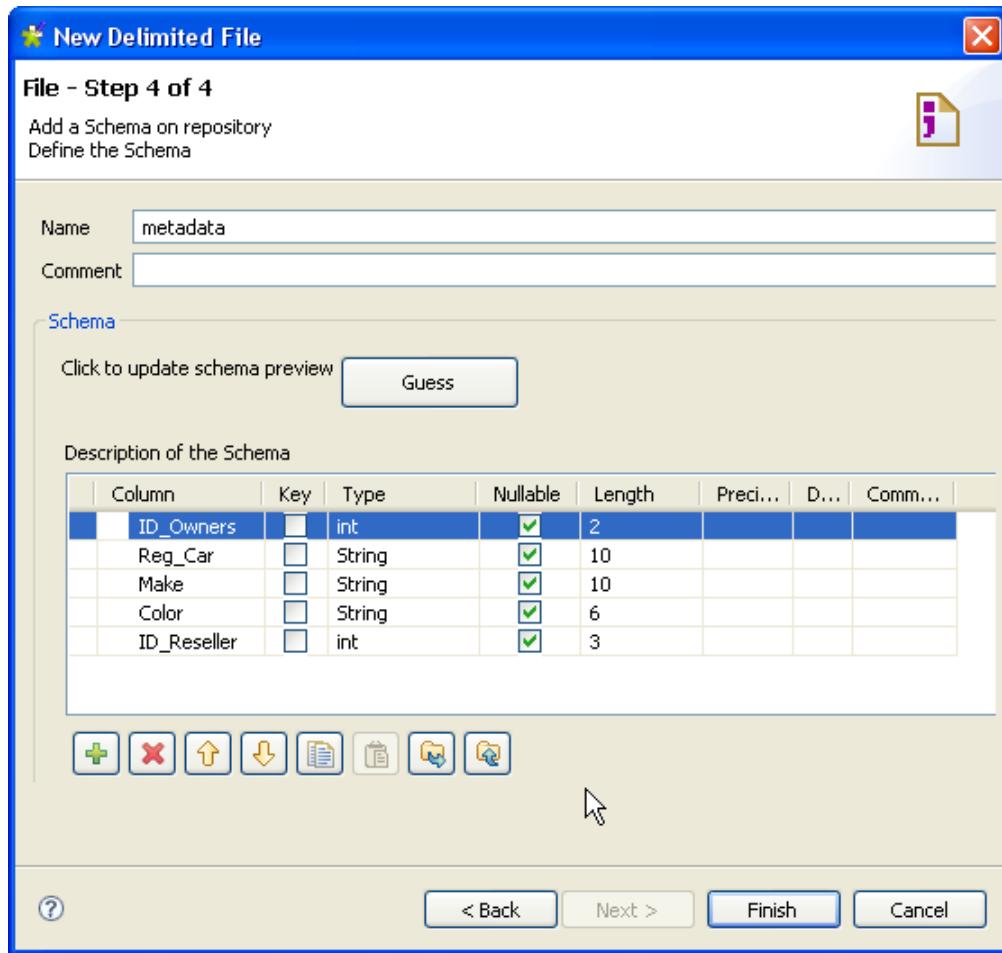
Cochez la case **Set heading row as column names** pour transformer la première ligne analysée en libellés des colonnes du schéma. Notez que le nombre de lignes d'en-tête à ignorer (champ **Rows To Skip**) est incrémenté de 1.

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	KWW2844
3	2580 TT 77	Renault	orange	1	bouhnan	BNU9147
4	1722 VE 11	Citroen	silver	10	kitcha	TEU0920

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

7.5.4. Etape 4 : Schéma final

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



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 nœud de la connexion **File Delimited** appropriée.

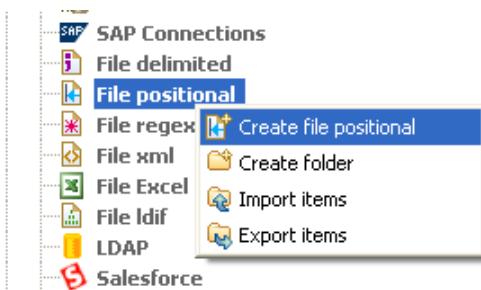
Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d'informations concernant l'action de déposer des métadonnées de composants dans l'espace de modélisation graphique, consultez la [Section 4.2.2.2, « Déposer des composants à partir du répertoire Metadata »](#).

Pour plus d'informations concernant l'action de déposer des métadonnées de composants dans l'espace de modélisation graphique, consultez la [Section 4.2.2.2, « Déposer des composants à partir du répertoire Metadata »](#).

7.6. 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**.

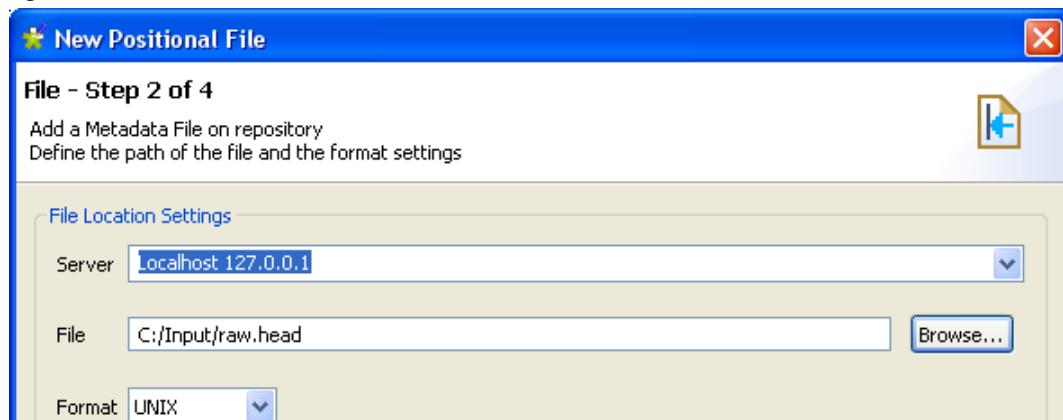
7.6.1. Etape 1 : Propriétés générales

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

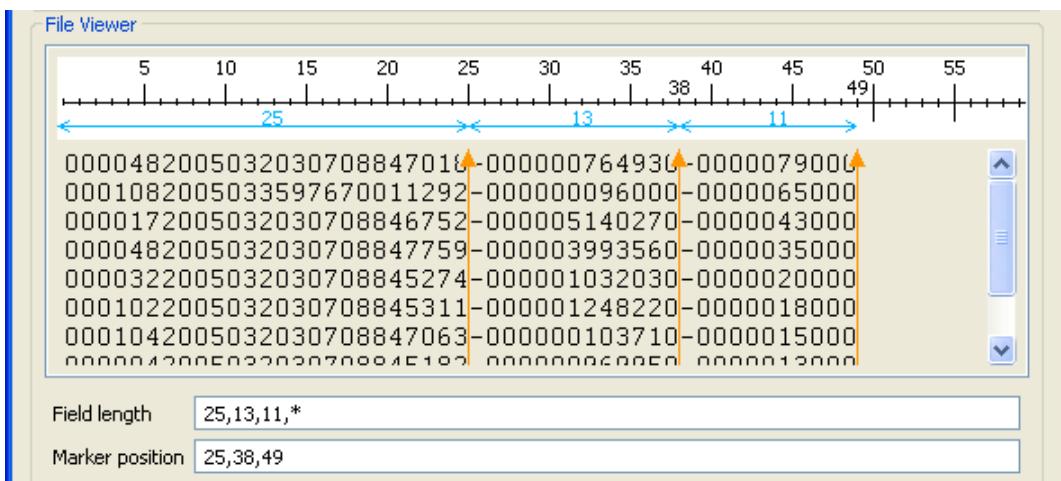
7.6.2. Etape 2 : Connexion et chargement du fichier

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.

7.6.3. 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**.

7.6.4. 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.

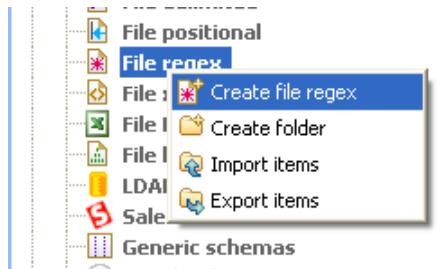
Vous pouvez personnaliser le nom de la métadonnée (par défaut, metadata) et apporter des modifications à l’aide de la barre d’outils.

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

Le nouveau schéma apparaît dans l’arborescence du **Repository**, sous le noeud de la connexion **File positional** 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.

7.7. 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**.

7.7.1. Étape 1 : Propriétés générales

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

7.7.2. Étape 2 : Chargement du fichier

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

File Settings	
Server	localhost 127.0.0.1
File	C:/Users/fbensaid/Desktop/regex.txt <input type="button" value="Browse..."/>
Format	WINDOWS
File Viewer	
<pre>firstname=Michael lastname=Jackson age=42 firstname=Elisa lastname=Do Brasil age=28 firstname=Michel lastname=Dujardin age=26 firstname=Robert lastname=Partou age=56 firstname=Diana lastname=Melbourne age=34 firstname=Marie lastname=Dolvina age=46 firstname=Jean lastname=Perfide age=59 firstname=Emilia lastname=Taldor age=21 firstname=Anne-Laure lastname=Paldufier age=67</pre>	

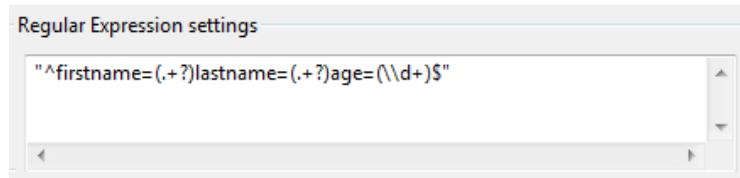
Comme pour la création d'un schéma Delimited File, le format est demandé pour pré-remplir certains champs des étapes suivantes. Si le système d'exploitation de création du fichier n'est pas proposé dans la liste, ignorez ce champ.

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

7.7.3. Étape 3 : Définition du schéma

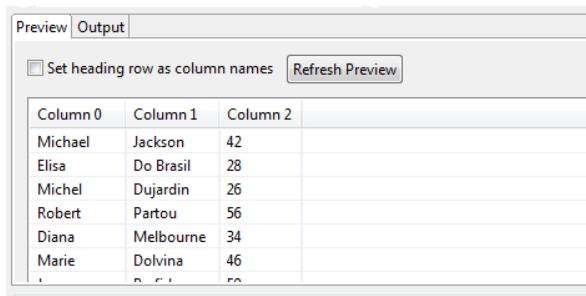
Cette étape ouvre la fenêtre de paramétrage du schéma. Comme pour les autres schémas de fichier, vous pouvez affiner la définition du schéma en précisant les séparateurs de champs et de lignes (**Field separator** et **Row separator**), le nombre de lignes d'en-tête, etc.

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



Veillez à utiliser les bon guillemets (simples ou doubles) au moment de rédiger le code.

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.

7.7.4. Étape 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 nœud 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 la [Section 4.2.2.2, « Déposer des composants à partir du répertoire Metadata »](#).

7.8. 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 7.8.1, « Configurer un schéma XML pour un fichier d'entrée \(Input\) »](#).

Si vous souhaitez écrire un fichier XML, consultez la [Section 7.8.2, « Configurer un schéma XML pour un fichier de sortie \(Output\) »](#).

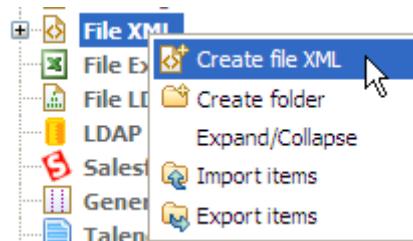
7.8.1. Configurer un schéma XML pour un fichier d'entrée (Input)

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 [Section 7.8.2, « Configurer un schéma XML pour un fichier de sortie \(Output\) »](#).

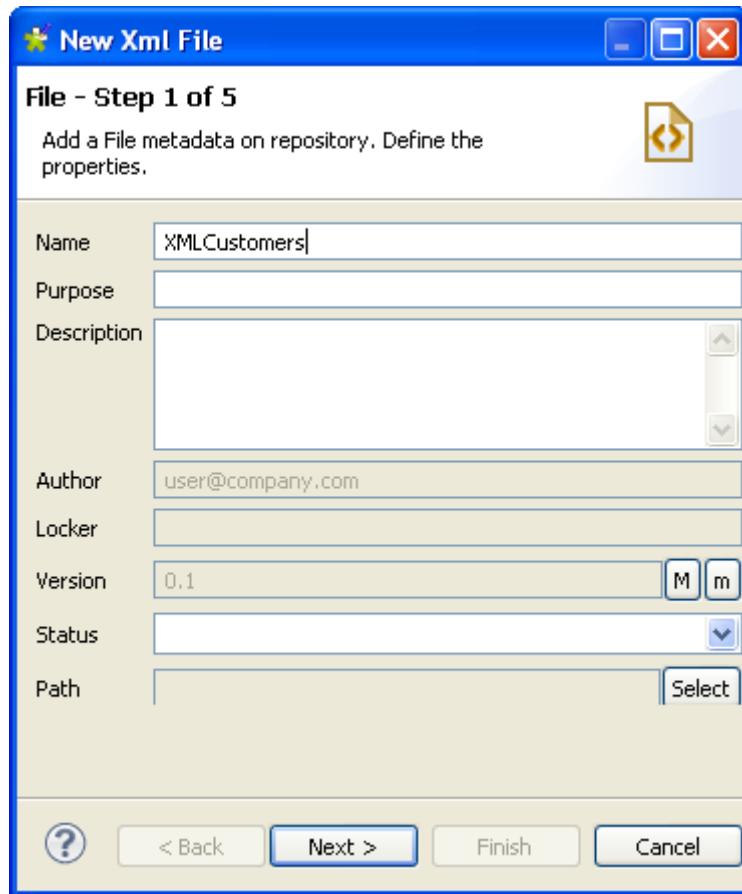
7.8.1.1. Étape 1 : Propriétés générales

Dans cette étape, vous allez définir les propriétés générales du fichier, telles que le nom (champ **Name**), l'objectif (champ **Purpose**) et une **Description**.

1. Dans la vue **Repository**, développez le nœud **Metadata**.
2. Cliquez-droit sur **File XML**, et sélectionnez **Create XML File** dans le menu contextuel.



3. Remplissez les informations généraiques du schéma, comme son nom (**Name**) et sa **Description**.

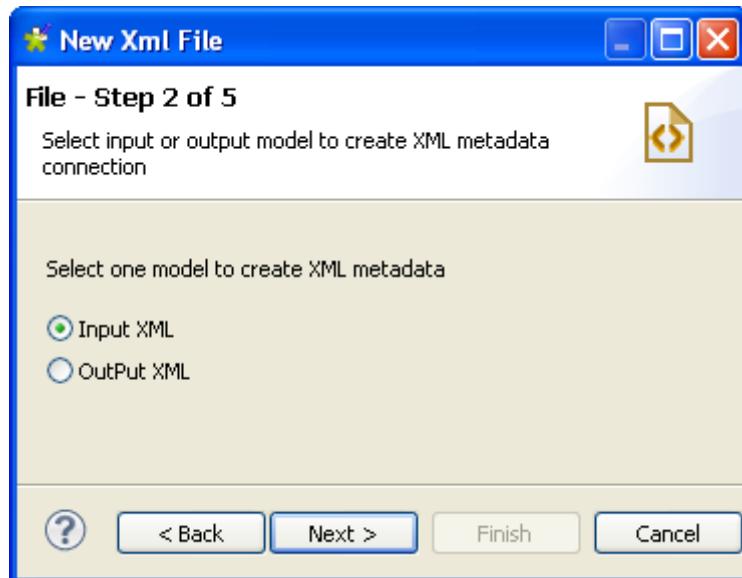


4. Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

7.8.1.2. Étape 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.

1. Dans la boîte de dialogue, sélectionnez **Input XML**.



2. Cliquez sur **Next** pour continuer.

7.8.1.3. Étape 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 ou XSD sélectionné.



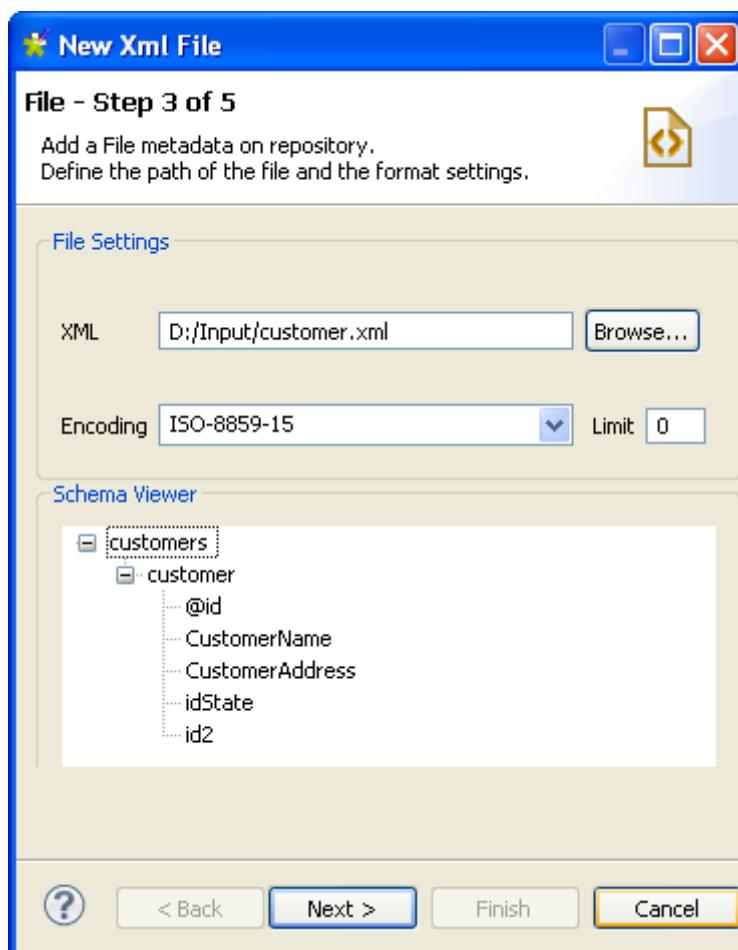
Si vous chargez un fichier XSD,

- les données seront sauvegardées dans le **Repository** et les métadonnées ne seront pas affectées par la suppression ou le déplacement du fichier.
- vous pouvez choisir un élément comme étant la racine de votre arborescence XML.

Pour charger un fichier XML, procédez comme suit :

1. 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.

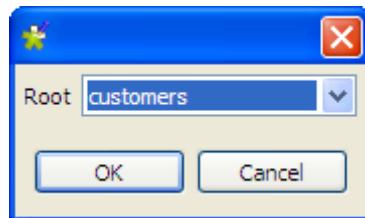


2. Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement
3. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez *0* si vous souhaitez l'effectuer sur toutes les colonnes.

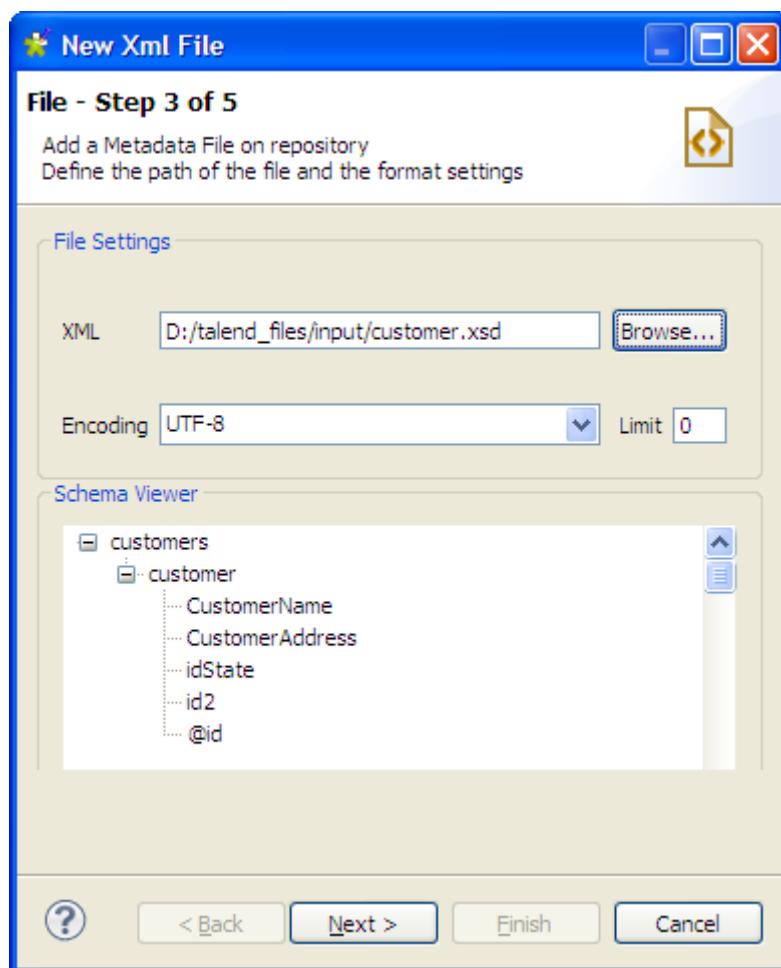
4. Cliquez sur **Next** pour continuer et définir les paramètres du schéma.

Pour charger un fichier XSD, procédez comme suit :

1. Cliquez sur **Browse...** et parcourez votre système jusqu'au répertoire du fichier XSD à charger. Ou bien saisissez le chemin d'accès au fichier.
2. Dans la boîte de dialogue qui apparaît, sélectionnez un élément dans la liste **Root**, afin d'en faire la racine de votre arborescence XML, puis cliquez sur **OK**.



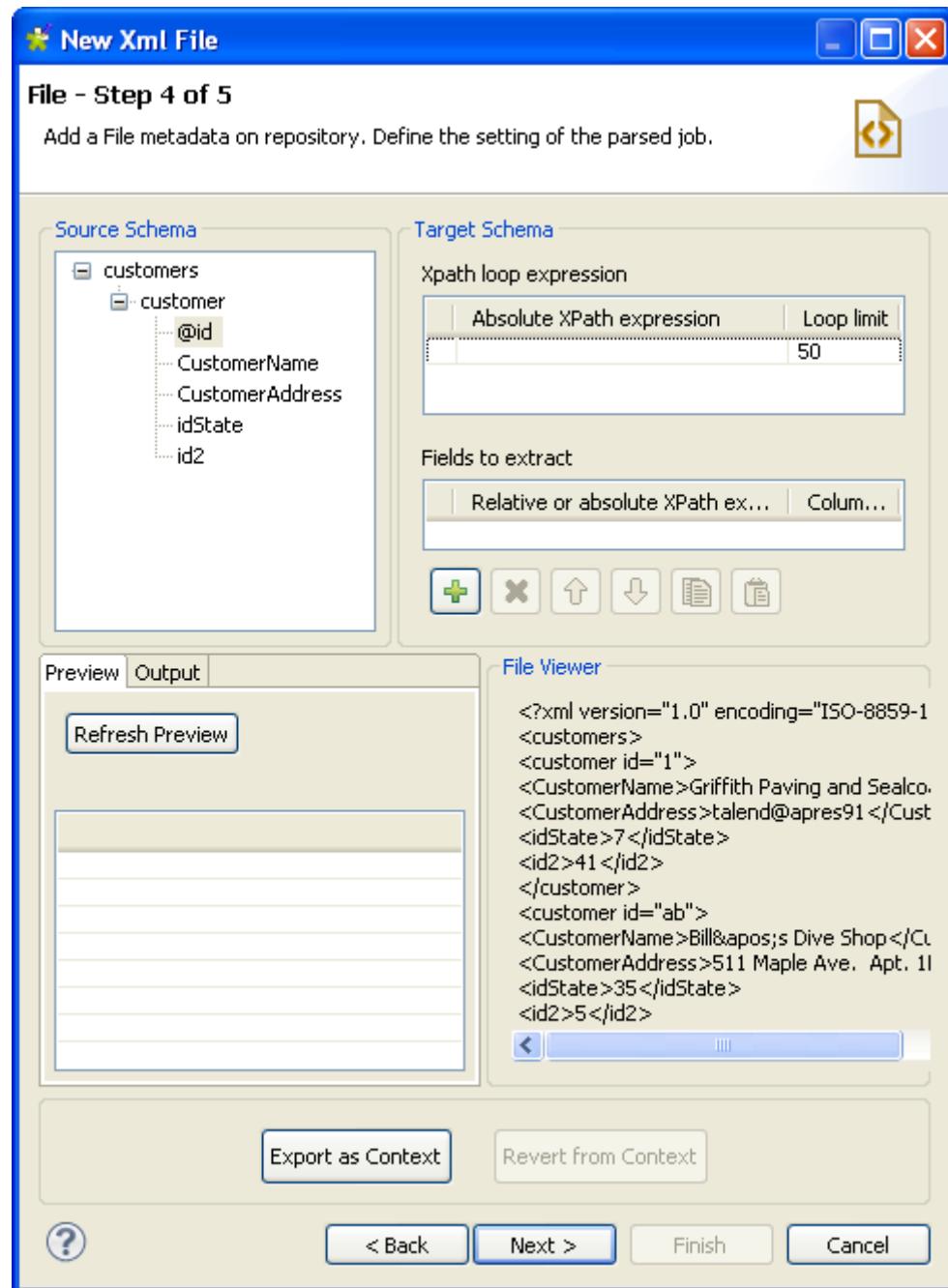
La zone **Schema Viewer** affiche un aperçu de la structure XML. Vous pouvez développer et visualiser chaque niveau de la structure XML du fichier.



3. Dans le champ **Encoding**, saisissez le type d'encodage si le système ne le détecte pas automatiquement.
4. Dans le champ **Limit**, saisissez le nombre de colonnes sur lesquelles exécuter la requête XPath, ou saisissez *0* si vous souhaitez l'exécuter sur tous les colonnes.
5. Cliquez sur **Next** afin de configurer les paramètres du schéma.

7.8.1.4. Étape 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	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible, ainsi que des données d'entrée des colonnes sélectionnées, affichées dans l'ordre défini. 💡 La fonctionnalité d'aperçu n'est pas disponible si vous chargez un fichier XSD.

Vue	Description
File viewer	Fenêtre d'aperçu des données brutes.

Tout d'abord, définissez la boucle XPath ainsi que le nombre maximum d'itérations à effectuer. Pour ce faire :

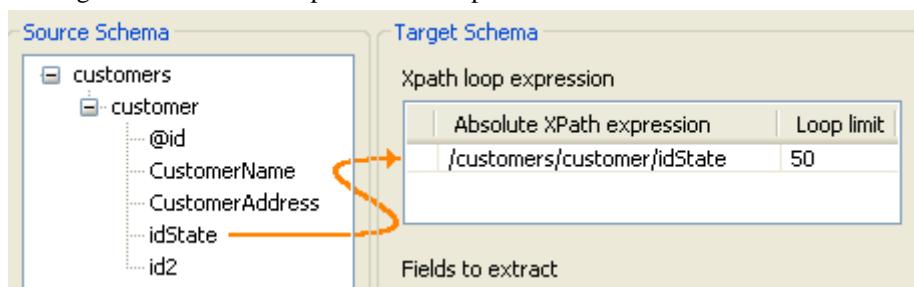
1. Peuplez le champ **Xpath loop expression**, avec l'expression XPath absolue du nœud qui fait l'objet de l'itération. Vous pouvez le faire de deux manières :

- Saisissez l'expression XPath absolue du nœud 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 nœud sélectionné de l'arborescence **Source Schema** dans le champ **Absolute XPath expression**.

Une flèche orange relie le nœud à l'expression correspondante.



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

2. Saisissez une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de nœuds à itérer.

Il est ensuite nécessaire de définir les champs à extraire. Pour ce faire :

1. Glissez-déposez les nœuds appropriés de la vue **Source Schema** vers le champ **Relative or absolute XPath expression**.



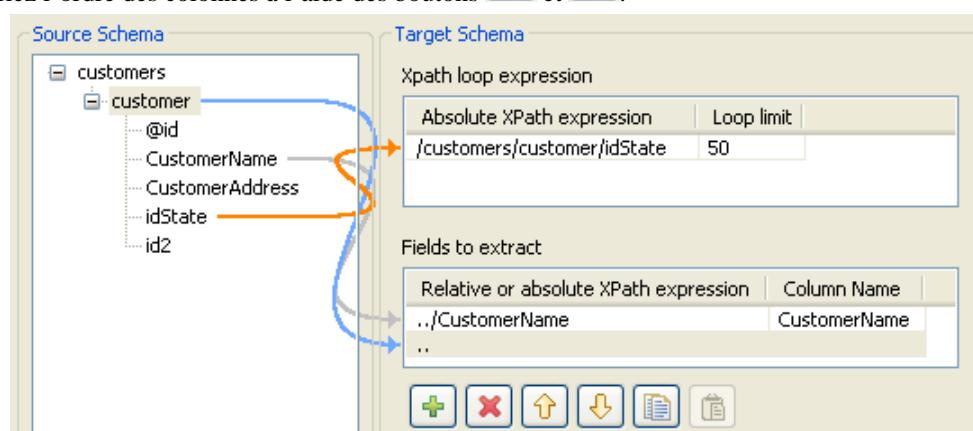
Vous pouvez sélectionner plusieurs nœuds à déposer dans le tableau, en appuyant sur **Ctrl** ou **Maj**, et en cliquant sur les nœuds adéquats. La flèche reliant un nœud sélectionné dans la zone **Source Schema** à celui déposé dans la zone **Fields to extract** est bleue. Les autres liens sont gris.



2. 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.



L'aperçu n'est pas disponible si vous chargez un fichier XSD.

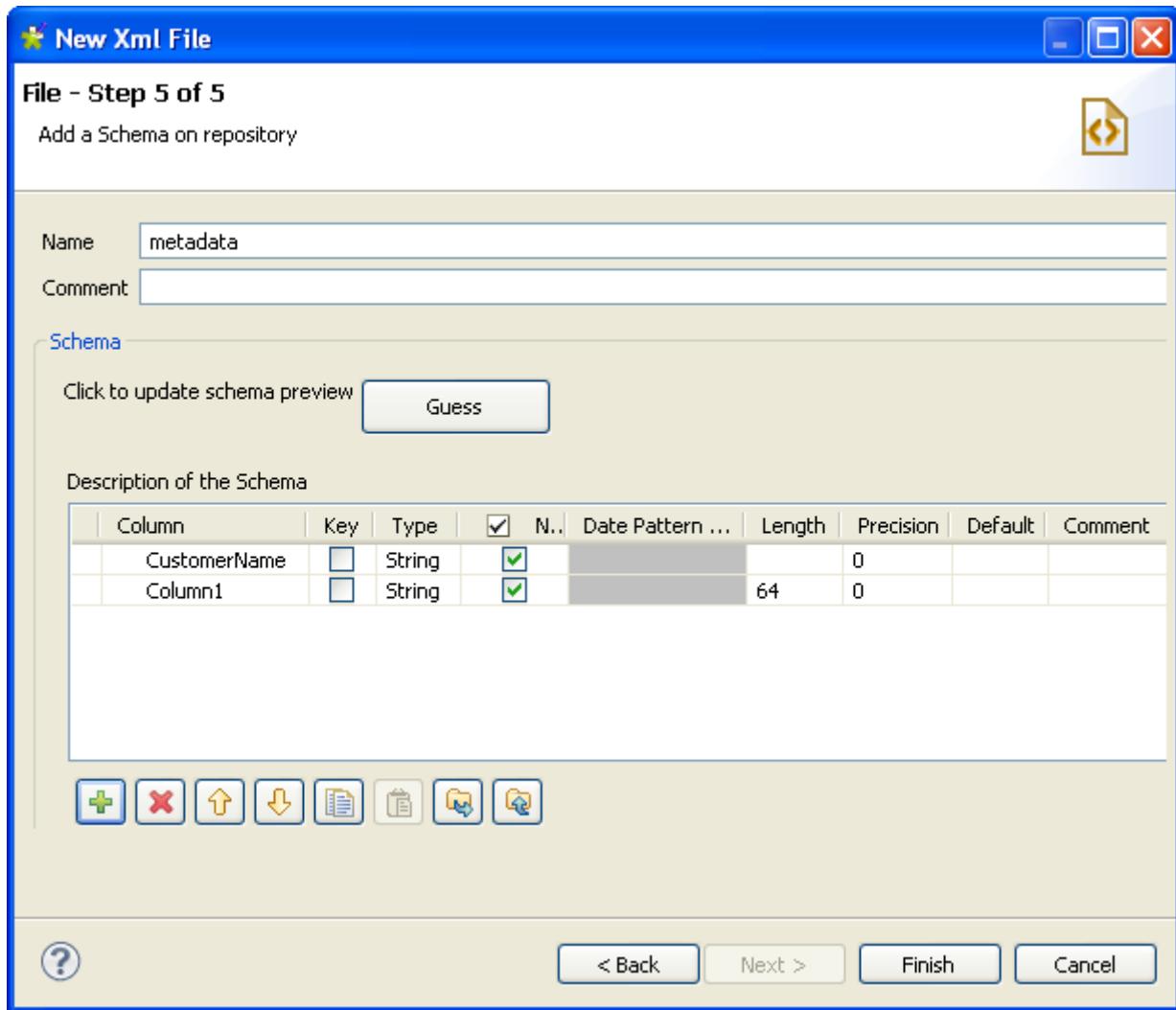
CustomerName	CustomerAddress
Griffith Paving and Sealcoatin	talend@apres91
Bill's Dive Shop	511 Maple Ave, Apt. 1B
Childress Child Day Care	662 Lyons Circle

```
<?xml version="1.0" encoding="UTF-8"?>
<customers>
  <customer id="1">
    <CustomerName>Griffith Paving and Sealcoatin</CustomerName>
    <CustomerAddress>talend@apres91</CustomerAddress>
    <idState>7</idState>
    <id2>2011</id2>
  </customer>
  <customer id="2">
    <CustomerName>Bill's Dive Shop</CustomerName>
```

- Cliquez sur **Next** pour finaliser le schéma final.

7.8.1.5. Étape 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**.



1. Ajoutez ou supprimez une colonne à l'aide des boutons et .
2. Modifiez l'ordre des colonnes à l'aide des boutons et .
3. Cliquez sur **Finish**.

Le nouveau schéma s'affiche dans le **Repository** sous le nœud **File xml** correspondant.

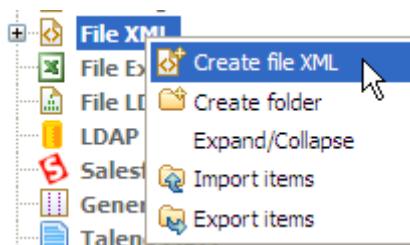
7.8.2. Configurer un schéma XML pour un fichier de sortie (Output)

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 la [Section 7.8.1, « Configurer un schéma XML pour un fichier d'entrée \(Input\) »](#).

7.8.2.1. Étape 1 : Propriétés générales

Dans cette étape, vous allez définir les propriétés générales du fichier, telles que le nom (champ **Name**), l'objectif (champ **Purpose**) et une **Description**.

1. Dans la vue **Repository**, développez le nœud **Metadata**.
2. Cliquez-droit sur **File XML** et sélectionnez **Create file XML** dans le menu contextuel.



3. Saisissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.

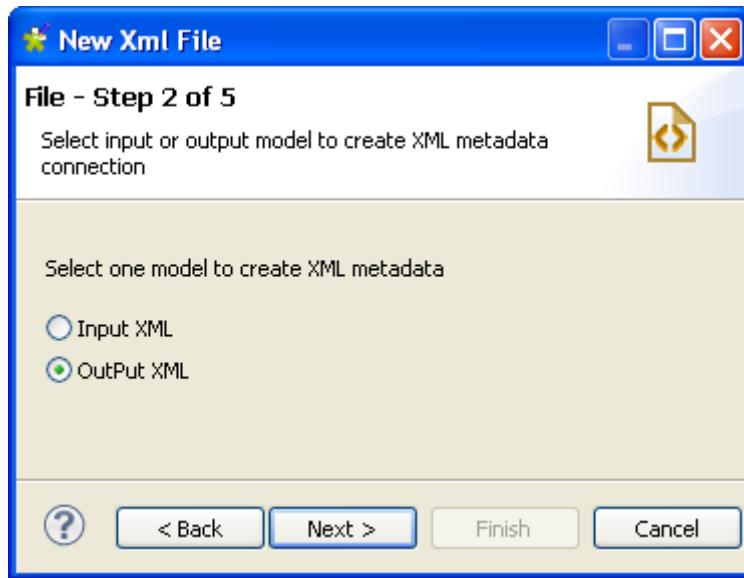
Name	NewXML_Output		
Purpose			
Description			
Author	user@company.com		
Locker			
Version	0.1	M	m
Status			
Path		Select	
<input style="width: 20px; height: 20px; vertical-align: middle;" type="button" value="?"/> <input style="width: 100px; height: 20px; vertical-align: middle;" type="button" value="< Back"/> <input style="width: 100px; height: 20px; vertical-align: middle;" type="button" value="Next >"/> <input style="width: 100px; height: 20px; vertical-align: middle;" type="button" value="Finish"/> <input style="width: 100px; height: 20px; vertical-align: middle;" type="button" value="Cancel"/>			

4. Cliquez sur **Next** pour définir le type de schéma.

7.8.2.2. Étape 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**.

1. Dans la boîte de dialogue, sélectionnez **Output XML**.



2. Cliquez sur **Next** pour définir le fichier de sortie.

7.8.2.3. Étape 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, comme le **tAdvancedFileOutputXML**. Dans cet exemple, créez le fichier en chargeant un fichier existant XML ou XSD.



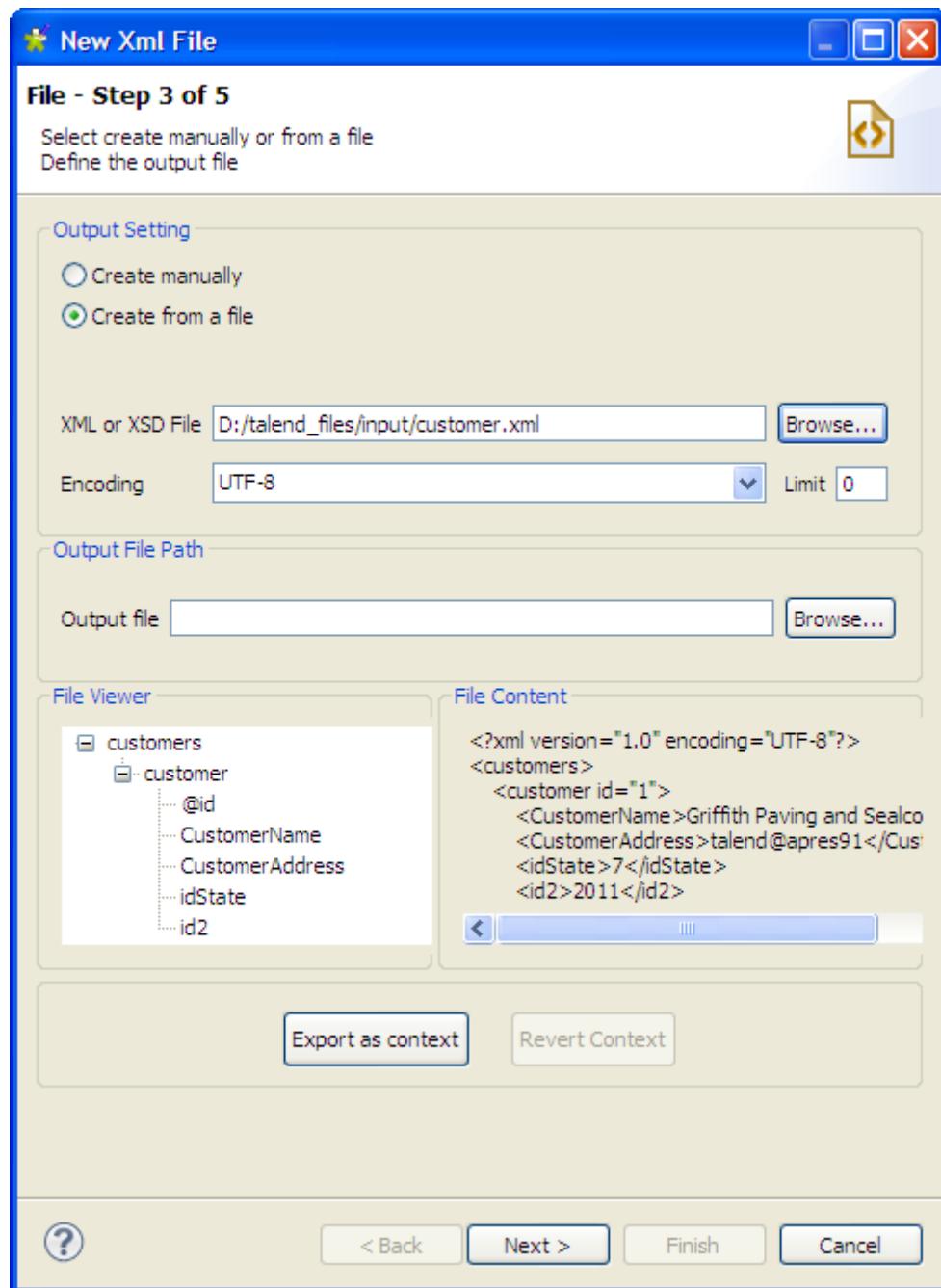
Si vous chargez un fichier XSD,

- les données seront sauvegardées dans le **Repository** et les métadonnées ne seront pas affectées par la suppression ou le déplacement du fichier.
- vous pouvez choisir un élément comme étant la racine de votre arborescence XML.

Pour créer le schéma XML de sortie à partir d'un fichier XML, procédez comme suit :

1. Sélectionnez l'option **Create from a file**.
2. Cliquez sur le bouton **Browse...** à côté du champ **XML or XSD File**, parcourez votre système jusqu'au fichier XML duquel la structure doit être appliquée au fichier de sortie et double-cliquez sur le fichier.

La zone **File Viewer** affiche l'aperçu de la structure XML et la zone **File Content** affiche au maximum les cinquante premières lignes du fichier.

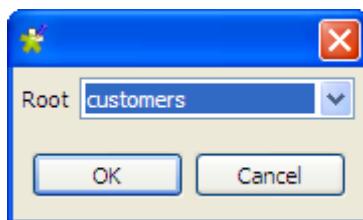


3. Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement.
4. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez *0* si vous souhaitez l'effectuer sur toutes les colonnes.
5. Dans la zone **Output File Path**, dans le champ **Output file**, renseignez le chemin d'accès à votre fichier de sortie. Si le fichier n'existe pas, il sera créé, lors de l'utilisation, dans un Job, du composant **tAdvancedFileOutputXML**. Si le fichier existe déjà, il sera écrasé.
6. Cliquez sur **Next** pour définir le schéma.

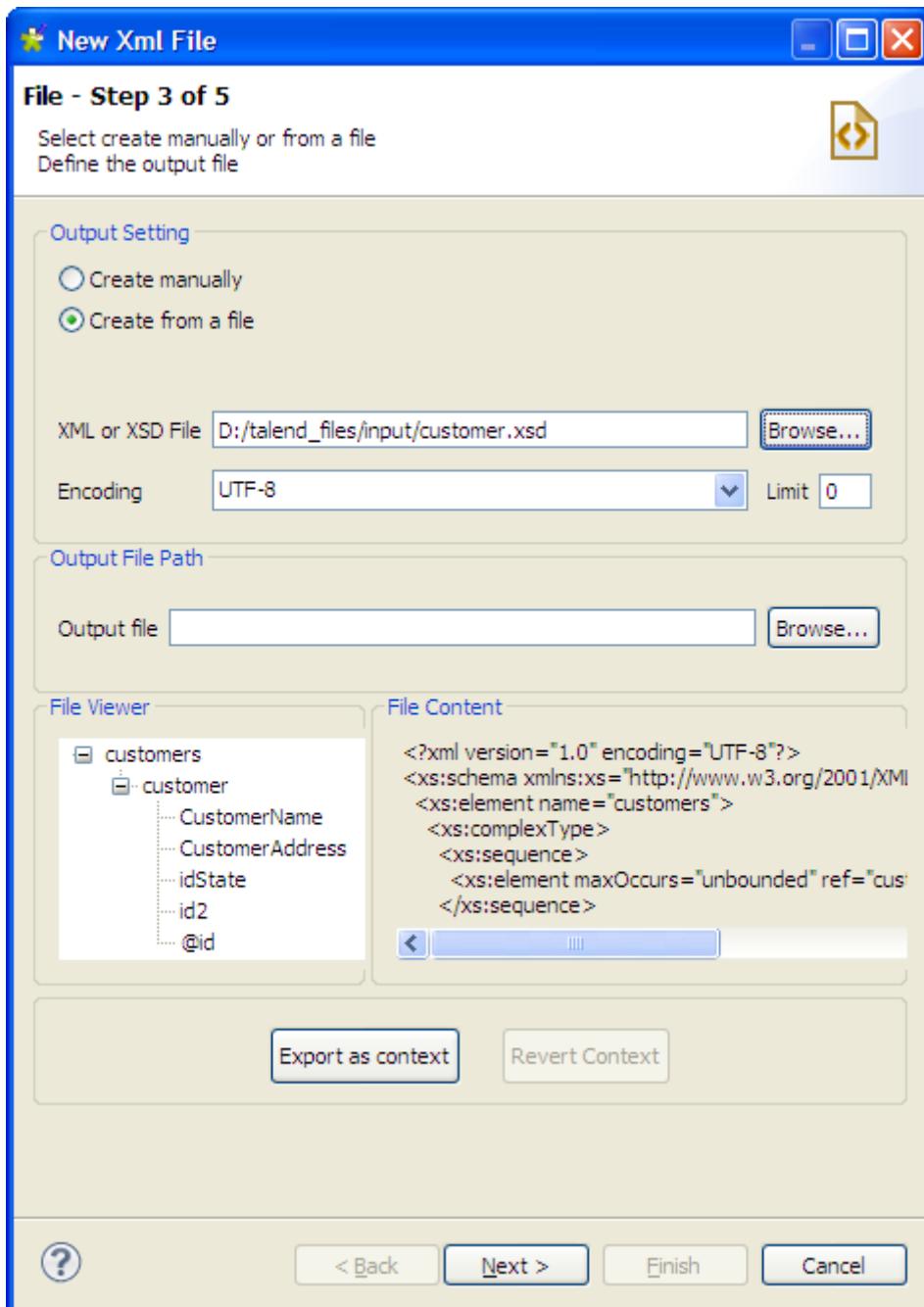
Pour créer le schéma XML de sortie à partir d'un fichier XSD, procédez comme suit :

1. Sélectionnez l'option **Create from a file**.
2. Cliquez sur le bouton **Browse...** à côté du champ **XML or XSD File**, parcourez votre système jusqu'au fichier XSD duquel la structure doit être appliquée au fichier de sortie et double-cliquez sur le fichier.

3. Dans la boîte de dialogue qui apparaît, sélectionnez un élément de la liste **Root**, afin d'en faire la racine de votre arborescence XML et cliquez sur **OK**.



La zone **File Viewer** affiche l'aperçu de la structure XML et la zone **File Content** affiche au maximum les cinquante premières lignes du fichier.



4. Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement.

5. Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez *0* si vous souhaitez l'effectuer sur toutes les colonnes.
6. Dans la zone **Output File Path**, dans le champ **Output file**, renseignez le chemin d'accès à votre fichier de sortie. Si le fichier n'existe pas, il sera créé, lors de l'utilisation, dans un Job, du composant **tAdvancedFileOutputXML**. Si le fichier existe déjà, il sera écrasé.
7. Cliquez sur **Next** pour définir le schéma.

7.8.2.4. Étape 4 : Définition du schéma

Lorsque les opérations précédentes sont terminées, les colonnes de la zone **Linker Source** sont automatiquement mappées vers celles correspondantes dans la zone **Linker Target**, comme le montrent les flèches bleues. Vous pouvez personnaliser les mappings lors de la configuration du schéma.

Dans cette étape, configurez le schéma de sortie. Le tableau suivant décrit comment faire :

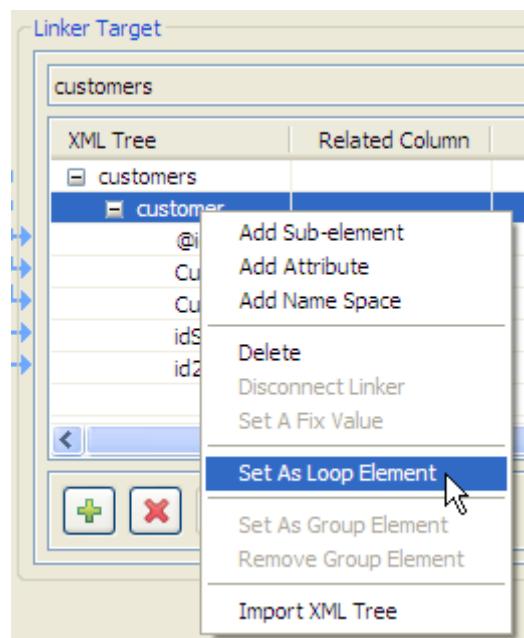
Pour...	Effectuer...
Définir un élément de boucle	<p>Dans la zone Linker Target, cliquez-droit sur l'élément qui vous intéresse et sélectionnez Set As Loop Element dans le menu contextuel.</p> <p> Cette opération est obligatoire pour définir un élément sur lequel effectuer une boucle.</p>
Définir un group element	<p>Dans la zone Linker Target, cliquez-droit sur l'élément qui vous intéresse et sélectionnez Set As Group Element dans le menu contextuel.</p> <p> Vous pouvez configurer l'élément parent de l'élément de boucle en tant que group element à condition que l'élément parent ne soit pas la racine de l'arborescence XML.</p>
Créer un élément fils pour un élément	<p>Dans la zone Linker Target,</p> <ul style="list-style-type: none"> • Cliquez-droit sur l'élément qui vous intéresse et sélectionnez Add Sub-element dans le menu contextuel, saisissez un nom pour le sous-élément dans la boîte de dialogue qui apparaît, puis cliquez sur OK, ou • Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton  en bas, sélectionnez Create as sub-element dans la boîte de dialogue qui apparaît, puis cliquez sur OK. Ensuite, saisissez un nom pour le sous-élément dans la boîte de dialogue suivante et cliquez sur OK.
Créer un attribut pour un élément	<p>Dans la zone Linker Target,</p> <ul style="list-style-type: none"> • Cliquez-droit sur l'élément qui vous intéresse et sélectionnez Add Attribute dans le menu contextuel, saisissez un nom pour l'attribut dans la boîte de dialogue qui apparaît et cliquez sur OK, ou • Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton  en bas, sélectionnez Create as attribute dans la boîte de dialogue qui apparaît, puis cliquez sur OK. Ensuite, saisissez un nom pour l'attribut dans la boîte de dialogue suivante et cliquez sur OK.
Créer un espace de nommage pour un élément	<p>Dans la zone Linker Target,</p> <ul style="list-style-type: none"> • Cliquez-droit sur l'élément qui vous intéresse et sélectionnez Add Name Space dans le menu contextuel, saisissez un nom pour l'espace de nommage dans la boîte de dialogue qui apparaît et cliquez sur OK, ou

Pour...	Effectuer...
	<ul style="list-style-type: none"> Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton en bas, sélectionnez Create as name space dans la boîte de dialogue qui apparaît, puis cliquez sur OK. Ensuite, saisissez un nom pour l'espace de nommage dans la boîte de dialogue suivante et cliquez sur OK.
Supprimer un ou plusieurs éléments/attributs/espace de nommage	<p>Dans la zone Linker Target,</p> <ul style="list-style-type: none"> Cliquez-droit sur les éléments/attributs/espace de nommage qui vous intéressent et sélectionnez Delete dans le menu contextuel, ou Sélectionnez les éléments/attributs/espace de nommage qui vous intéressent et cliquez sur le bouton en bas, ou Sélectionnez les éléments/attributs/espace de nommage qui vous intéressent et appuyez sur la touche Suppr. <p> Supprimer un élément supprime également ses enfants, s'il en a.</p>
Ajuster l'ordre d'un ou plusieurs élément(s)	Dans la zone Linker Target , sélectionnez l'élément qui vous intéressent et cliquez sur les boutons et , afin de déplacer vers le haut ou vers le bas, respectivement, l'élément sélectionné.
Configurer une valeur statique pour un élément/attribut/espace de nommage	<p>Dans la zone Linker Target, cliquez-droit sur l'élément/l'attribut/l'espace de nommage qui vous intéresse et sélectionnez Set A Fix Value dans le menu contextuel.</p> <p></p> <ul style="list-style-type: none"> La valeur que vous avez configurée va remplacer toute valeur récupérée du flux d'entrée dans votre Job. Vous pouvez configurer une valeur statique pour un élément fils de l'élément de boucle uniquement, à condition que l'élément n'ait pas lui-même d'enfants et pas de mapping source-cible sur lui.
Créer un mapping source-cible	<p>Sélectionnez la colonne qui vous intéresse, dans la zone Linker Source, déposez-la sur le noeud qui vous intéresse de la zone Linker Target et sélectionnez Create as sub-element of target node, Create as attribute of target node, ou Add linker to target node selon vos besoins, dans la boîte de dialogue qui apparaît, puis cliquez sur OK.</p> <p>Si vous choisissez une option qui n'est pas permise pour le noeud cible, un message d'avertissement apparaît et l'opération échoue.</p>
Supprimer un mapping source-cible	Dans la zone Linker Target , cliquez-droit sur le noeud qui vous intéresse et sélectionnez Disconnect Linker dans le menu contextuel.
Créer une arborescence XML à partir d'un autre fichier XML ou XSD	Cliquez-droit sur tout élément du schéma dans la zone Linker Target et sélectionnez Import XML Tree dans le menu contextuel afin de charger un autre fichier XML ou XSD. Ensuite, vous devez créer manuellement les mappings source-cible et définir à nouveau le schéma de sortie.

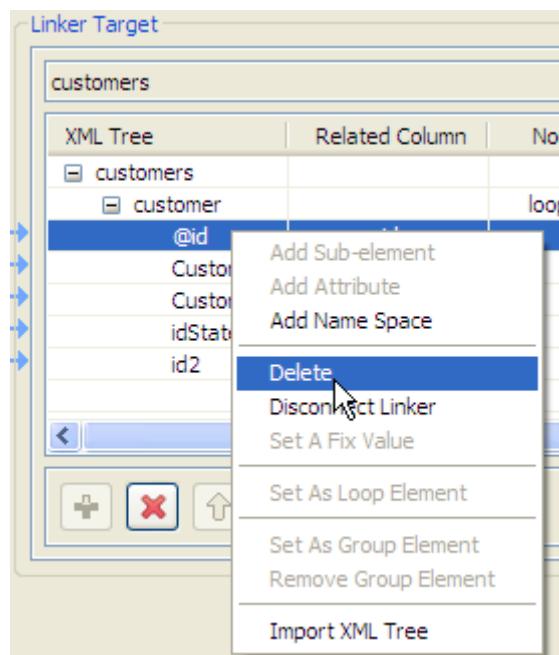
Vous pouvez sélectionner et déposer plusieurs champs à la fois, en utilisant les touches **Ctrl** ou **Maj** pour effectuer une sélection multiple et rendre le mapping plus rapide. Vous pouvez également effectuer une sélection multiple lors des opérations de clic-droit.

Dans cet exemple, le schéma de sortie est basé sur le fichier chargé *customer.xml*. Configurez une boucle sur l'élément *customer* et définissez le noeud *id* comme élément fils, plutôt que comme attribut dans le fichier chargé. Pour ce faire :

1. Dans la zone **Linker Target**, cliquez-droit sur l'élément *customer*, puis, dans le menu contextuel, sélectionnez **Set As Loop Element**.

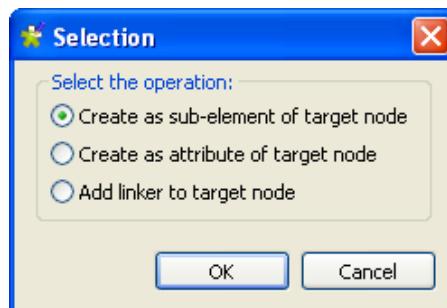


2. Cliquez-droit sur le nœud *id* et sélectionnez **Delete** dans le menu contextuel.



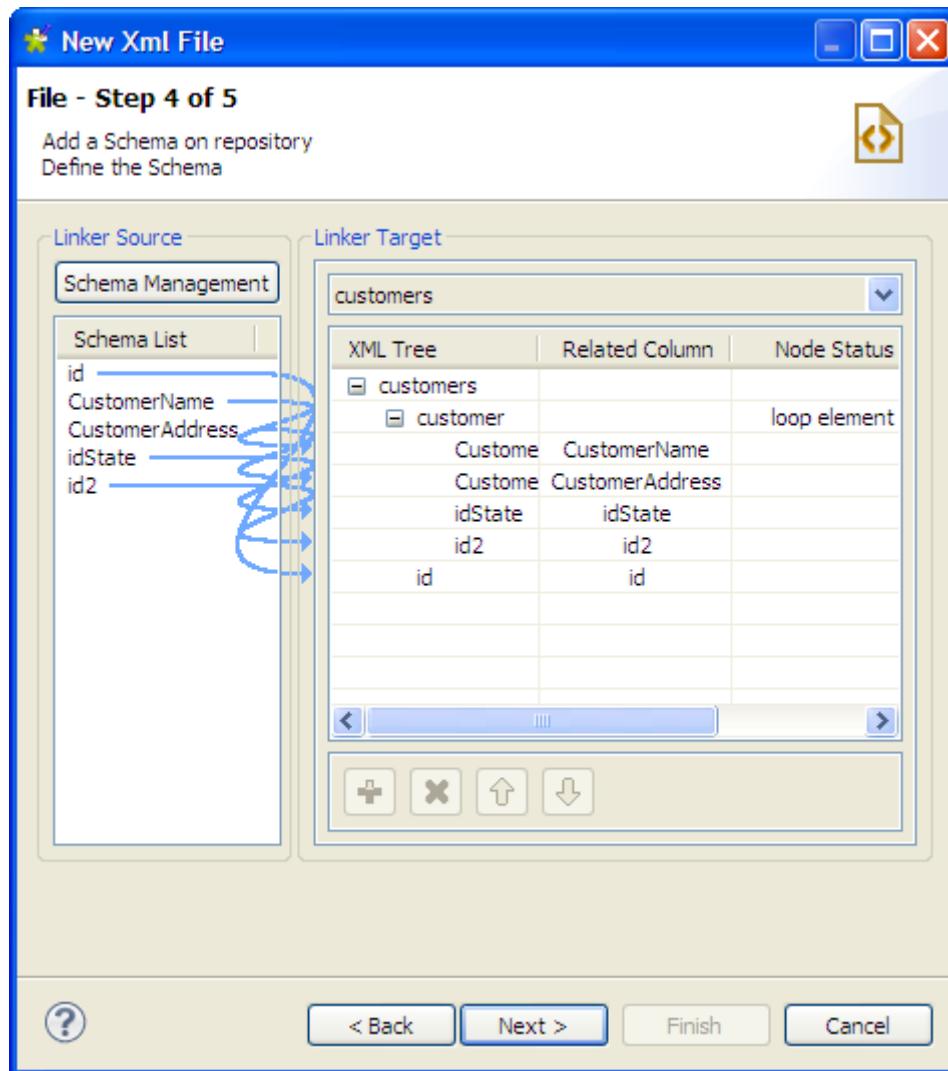
3. Sélectionnez la colonne *id*, dans la zone **Linker Source**, et déposez-la dans l'élément *customer* de la zone **Linker Target**.

La boîte de dialogue **[Selection]** s'ouvre et vous demande de définir la relation source-cible.



4. Sélectionnez l'option **Create as sub-element of target node** et cliquez sur **OK** pour valider votre choix.

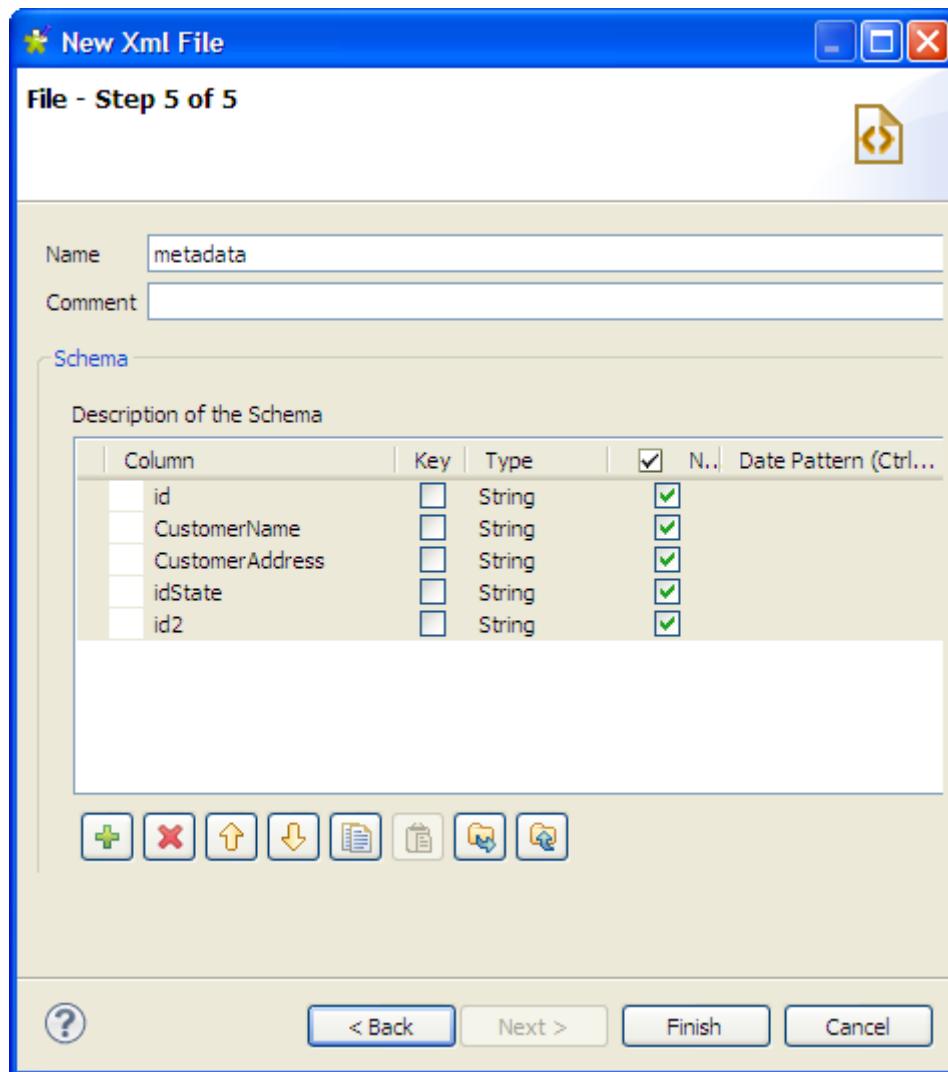
Une flèche bleue relie alors les deux colonnes.



5. Cliquez sur **Next** pour passer à l'étape suivante.

7.8.2.5. Étape 5 : Schéma final

L'étape 5 affiche le schéma final généré.



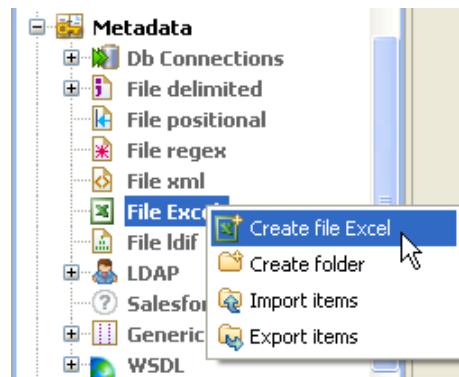
Vous pouvez personnaliser le nom de la métadonnée dans le champ **Name** (par défaut *metadata*), ajouter un commentaire dans le champ **Comment** et apporter des modifications à l'aide de la barre d'outils, par exemple :

- 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 nœud **File xml** correspondant.

7.9. 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**.

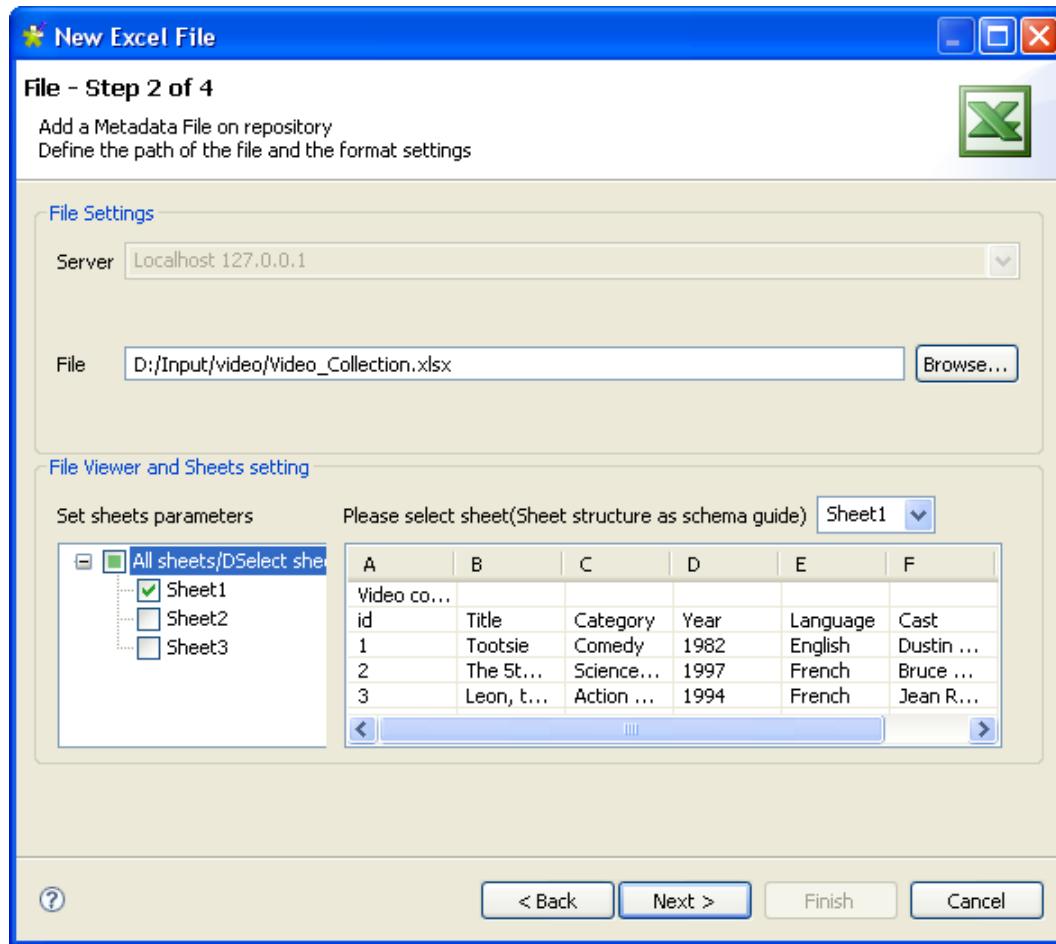
7.9.1. Étape 1 : Propriétés générales

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

7.9.2. Étape 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.

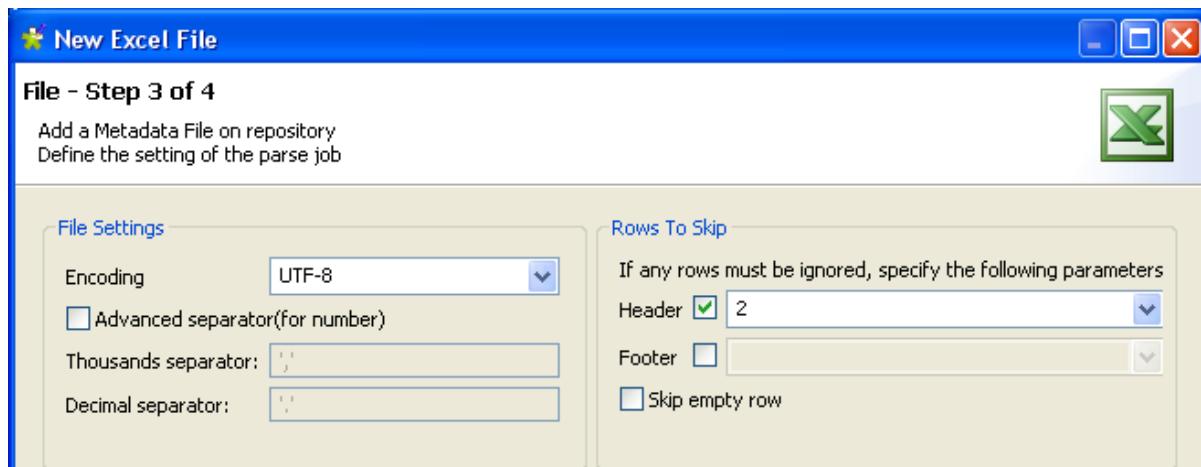
Dans la liste **Set sheets parameters**, cochez la case à gauche de la feuille Excel à charger. Par défaut, l'aperçu affiche le contenu de la première feuille Excel.

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.

7.9.3. Étape 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.

Metadata column setting First column: <input type="text" value="1"/> Last column: <input type="text"/>	Limit Of Rows If the number of lines must be limited, specify this number Limit <input type="text"/>
---	---

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.

Preview

Set heading row as column names

A	B	C	D	E	F
Video collect...					
id	Title	Category	Year	Language	Cast
1	Tootsie	Comedy	1982	English	Dustin Hoffman, Jessica Lange, Sydney Pollack
2	The 5th Element	Science Fiction	1997	French	Bruce Willis, Gary Oldman, Milla Jovovich
3	Leon, the Professional	Action drama	1994	French	Jean Reno, Gary Oldman, Natalie Portman

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.

7.9.4. Étape 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.

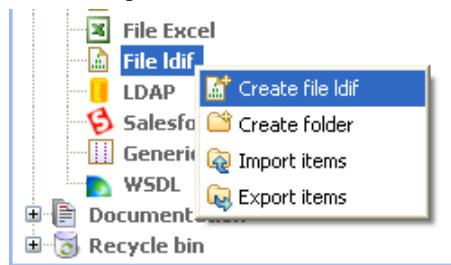
Vous pouvez personnaliser le nom de la métadonnée (par défaut, *metadata*) et apporter des modifications à l'aide de la barre d'outils.

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.

7.10. 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**.

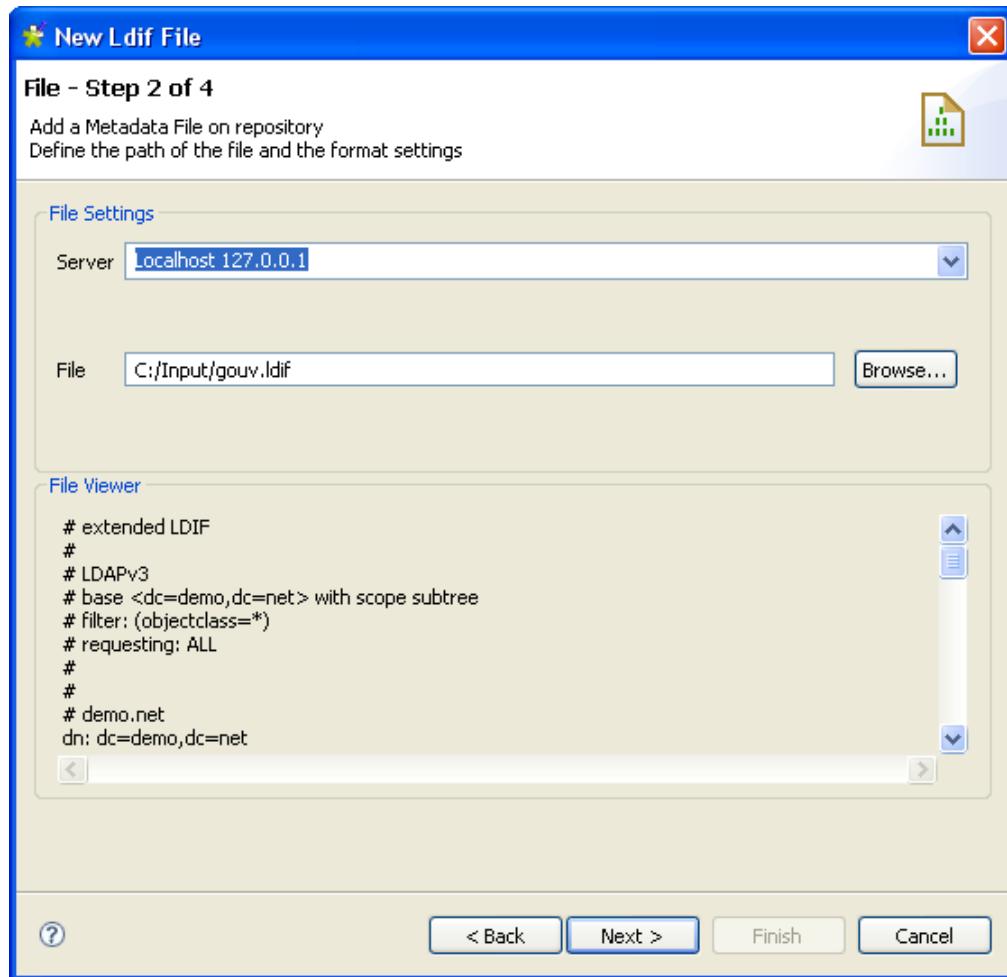
Assurez-vous que les modules requis pour l'utilisation de cette fonctionnalité sont bien installés. Pour plus d'informations, visitez <http://talendforge.org/wiki/doku.php>

7.10.1. Étape 1 : Propriétés générales

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

7.10.2. Étape 2 : Chargement du fichier

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

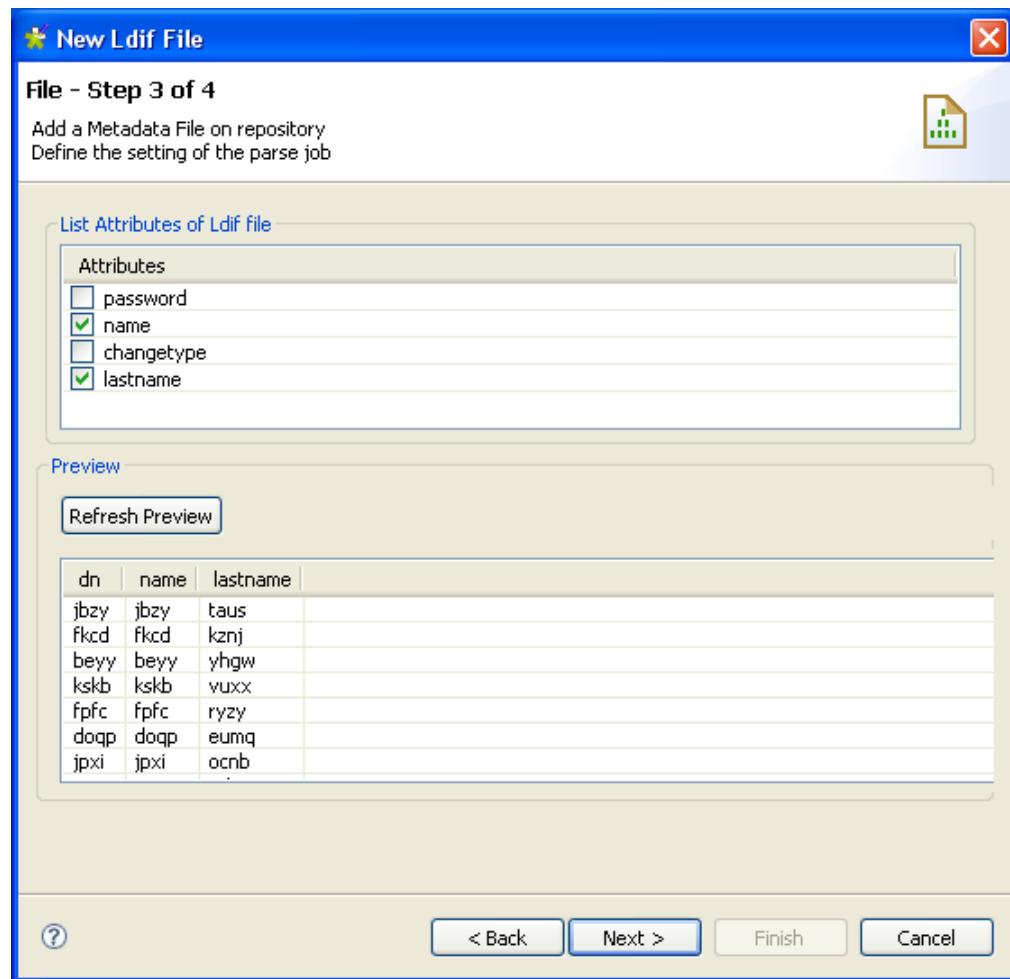


 La fonctionnalité de connexion à un serveur distant n'est pour l'instant pas opérationnelle pour la récupération de fichier LDIF.

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

7.10.3. Étape 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.

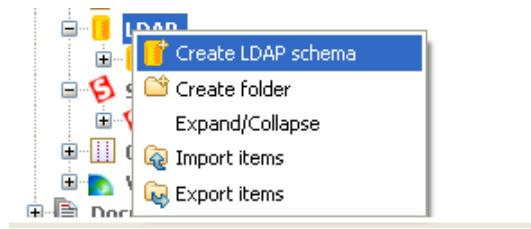
7.10.4. Étape 4 : Schéma final

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 nœud de connexion LDIF approprié.

7.11. Configurer un schéma LDAP

Dans le **Repository**, cliquez droit sur le nœud **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.

7.11.1. Étape 1 : Propriétés générales

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

Pour plus d'informations, consultez [Section 7.2.1, « Étape 1 : Propriétés générales »](#).

7.11.2. Étape 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

A screenshot of the 'File - Step 2 of 5' configuration window. It shows a 'Network Parameter' section with three fields: 'Hostname' set to 'Your-LDAP-IP', 'Port' set to '389', and 'Encryption method' set to 'LDAP'. Below the fields is a note: 'Click the button to check connection status.' followed by a 'Check Network Parameter' button. At the top right, there is a small icon of a document with a purple 'i' symbol.

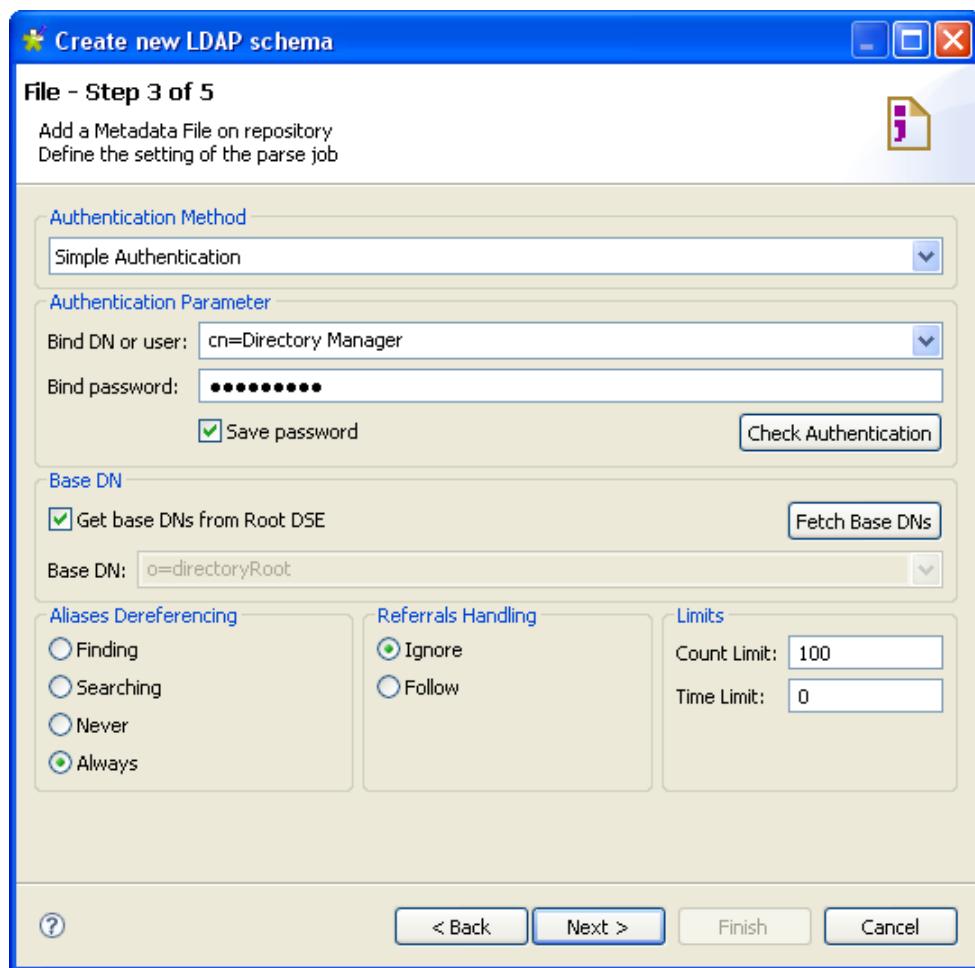
Vérifiez votre connexion à l'aide de bouton **Check Network Parameter**, puis cliquez sur le bouton **Next**.

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.

7.11.3. Étape 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	<p>Bind DN or User : connectez-vous en fonction de la méthode d'authentification LDAP choisie.</p> <p>Bind password : renseignez le mot de passe.</p> <p>Save password : cochez cette case pour enregistrer les informations d'authentification.</p>
Get Base DN from Root DSE / Base DN	<p>Chemin d'accès à l'arborescence de l'utilisateur.</p> <p>Le bouton Fetch Base DNs redirige le DN automatiquement à la racine.</p>
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>

Champ	Description
	Finding : cette option déréférence les alias uniquement lors de la résolution du nom.
Referral Handling	Redirection des requêtes utilisateur : Ignore ne tient pas compte des redirections des requêtes. Follow tient compte des redirections des requêtes.
Limit	Limite le nombre d'enregistrements lus

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

7.11.4. Étape 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.

The screenshot shows the 'Create new LDAP schema' dialog box, specifically Step 4 of 5. The main interface includes:

- List attributes of LDAP Schema:** A list of attributes with checkboxes. 'uid' is checked, while 'description', 'userpassword', 'sn', and 'cn' are unchecked.
- Filter:** A text input field containing the filter expression: `(&(objectClass=*))`.
- Preview:** A table showing a sample of data extracted from the schema. The columns are uid, mail, givenname, and telephonenumber. The data rows are:

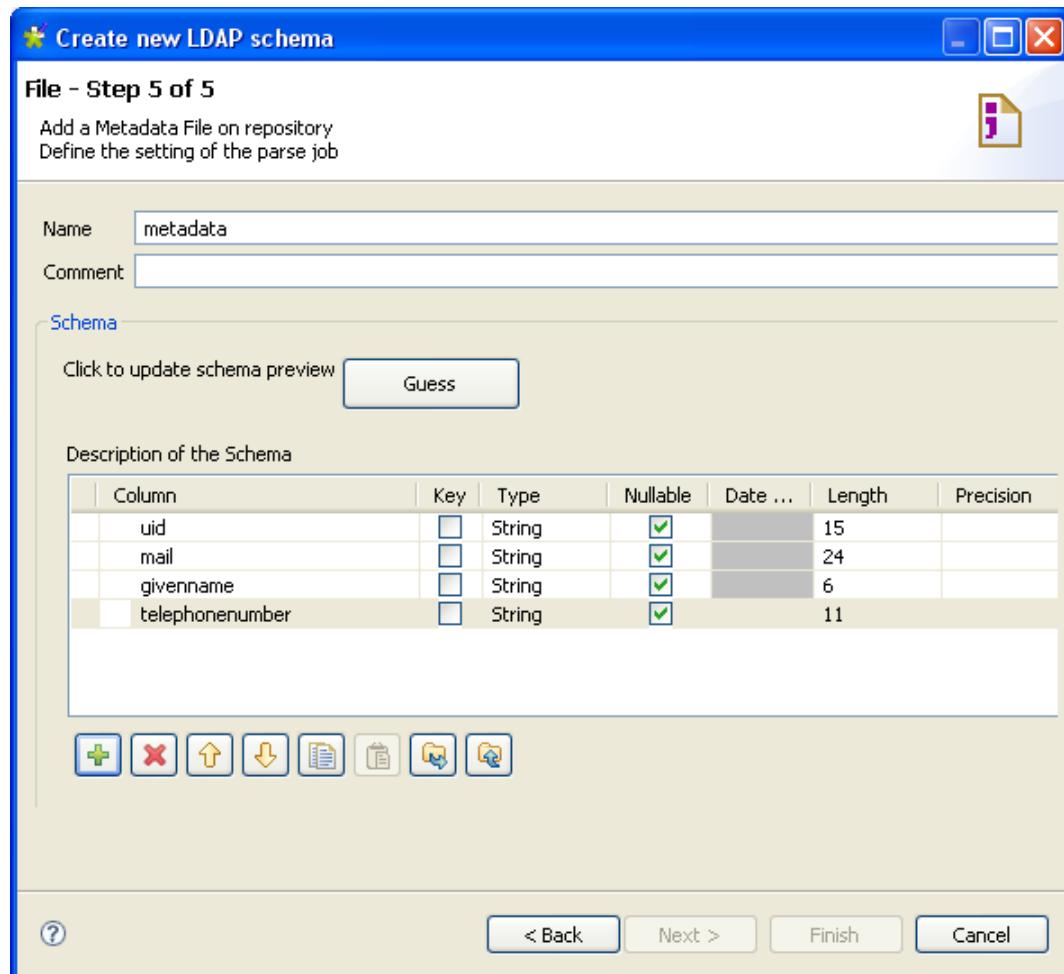
uid	mail	givenname	telephonenumber
PIERRE DUPONT	Pierre.Dupont@talend.com	PIERRE	00149684750
PIERRE DUPON...	mhirt78@talend.com	PIERRE	
mhirt			
greg			
..			
- Buttons at the bottom:** '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

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

Puis cliquez sur **Next** pour continuer.

7.11.5. Étape 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.

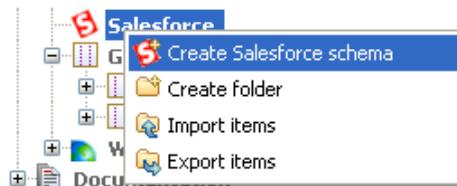


Si l'annuaire LDAP ayant donné lieu au schéma a changé, cliquez sur le bouton **Guess** pour générer le schéma une nouvelle fois. Notez que les modifications que vous avez apportées au schéma seront perdues si vous cliquez sur le bouton **Guess**.

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

7.12. Configurer un schéma Salesforce

Dans le **Repository**, cliquez-droit sur le nœud **Salesforce** dans l'arborescence, puis sélectionnez **Create Salesforce schema** dans le menu contextuel.



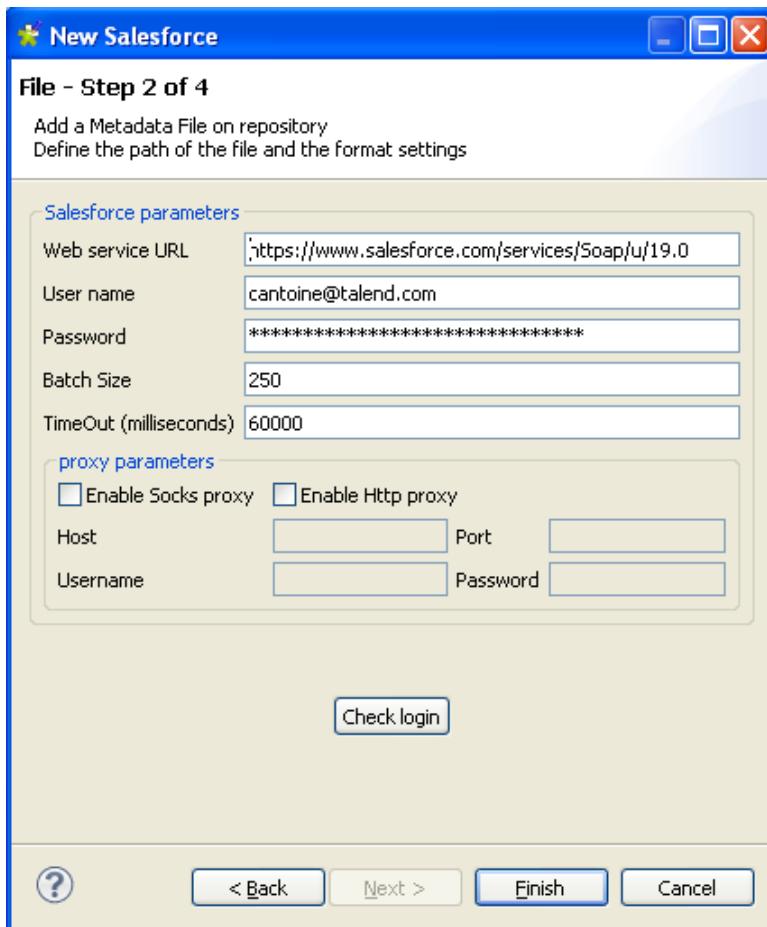
Procédez de la même manière que pour toute autre connexion de métadonnées.

7.12.1. Étape 1 : Propriétés générales

Renseignez les informations générales du schéma, notamment son nom dans le champ **Name** et sa **Description**.

7.12.2. Étape 2 : Connexion au compte Salesforce

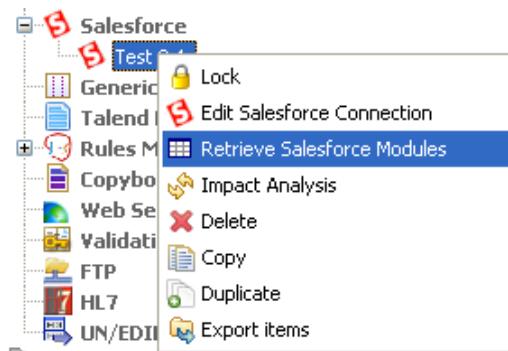
L'adresse du service Web Salesforce apparaît par défaut dans le champ **Web service URL**.



- Dans les champs **User name** et **Password**, saisissez votre identifiant et mot de passe de connexion au service Web Salesforce.
- Cliquez sur **Check login** pour vérifier que votre connexion fonctionne correctement.
- Cliquez sur **Finish** pour fermer l'assistant.

7.12.3. Étape 3 : Récupérer les modules Salesforce

Cette étape récupère les modules Salesforce à partir de la connexion que vous avez définie dans l'étape 2.



Dans la vue **Repository**, développez le nœud **Connection** et cliquez-droit sur la connexion définie dans l'étape 2 puis sélectionnez **Retrieve Salesforce Modules** dans le menu contextuel.

Dans la zone **Select Schema to create**, vous pouvez réduire la sélection des schémas affichés à l'aide d'un filtre. Saisissez dans le champ **Name Filter** le nom du schéma que vous souhaitez retrouver. Pour récupérer plus de modules, cochez la case des schémas respectifs.

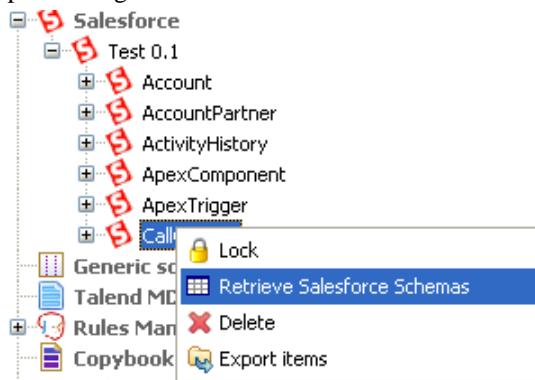
Name	Type	Column number	Creation status
<input checked="" type="checkbox"/> Account	Modules	43	Success
<input type="checkbox"/> AccountContactRole	Modules		
<input type="checkbox"/> AccountFeed	Modules		
<input type="checkbox"/> AccountHistory	Modules		
<input checked="" type="checkbox"/> AccountPartner	Modules	13	Success
<input type="checkbox"/> AccountShare	Modules		
<input checked="" type="checkbox"/> ActivityHistory	Modules	28	Success
<input type="checkbox"/> AdditionalNumber	Modules		
<input type="checkbox"/> AggregateResult	Modules		
<input type="checkbox"/> ApexClass	Modules		
<input checked="" type="checkbox"/> ApexComponent	Modules	14	Success
<input type="checkbox"/> ApexLog	Modules		
<input type="checkbox"/> ApexPage	Modules		
<input checked="" type="checkbox"/> ApexTrigger	Modules	23	Success
<input type="checkbox"/> Approval	Modules		
<input type="checkbox"/> Asset	Modules		
<input type="checkbox"/> AssetFeed	Modules		
<input type="checkbox"/> AssignmentRule	Modules		
<input type="checkbox"/> AsyncApexJob	Modules		
<input type="checkbox"/> Attachment	Modules		

Buttons at the bottom: Select All, Select None, Check Connection, Finish, Cancel.

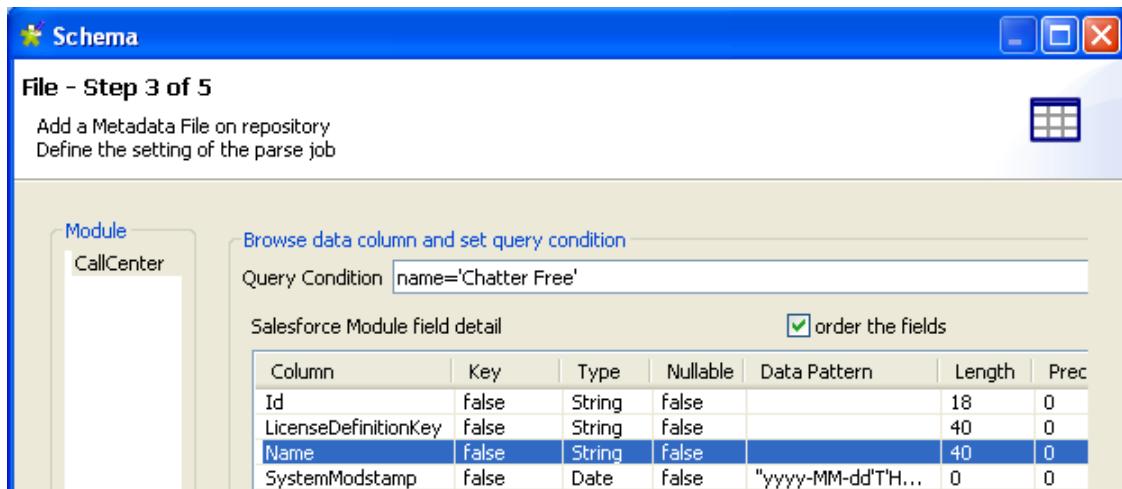
Cliquez sur le bouton **Check Connection** afin de vérifier le statut de création puis cliquez sur **Finish** pour sauvegarder les modules que vous avez récupérés.

7.12.4. Étape 4 : Récupérer les schémas Salesforce

Cette étape ouvre la fenêtre de paramétrage du schéma en fonction du module récupéré dans l'étape 3.



Cliquez-droit sur le module récupéré précédemment et sélectionnez **Retrieve Salesforce Schemas** dans le menu contextuel.



Dans la zone **Browse data column and set query condition**, vous pouvez filtrer les données affichées. Pour cela, saisissez le nom de 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. Décochez la case **order the fields** pour les lister de manière aléatoire.

The 'Preview' tab displays a table of retrieved data from the Salesforce module. The columns are Id, LicenseDefinitionKey, Name, and SystemModstamp. The data is as follows:

Id	LicenseDefinitionKey	Name	SystemModstamp
100700000009iZKAAY	CSN_User	Chatter Free	2010-12-04T18:18:49.000Z
100700000009RacAAE	SFDC	Salesforce	2011-03-09T03:54:47.000Z

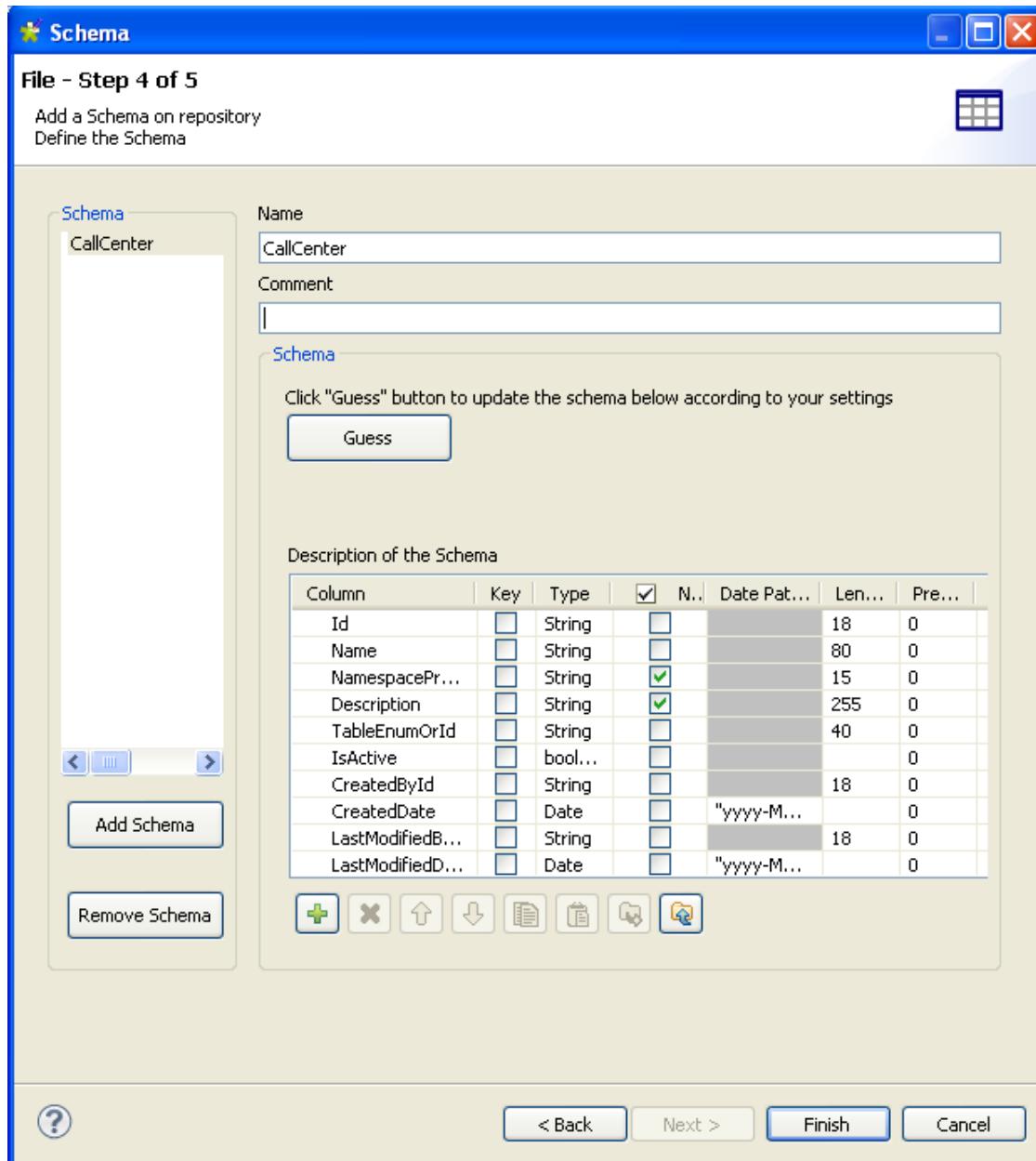
Cliquez sur **Refresh Preview** si vous avez saisi une requête, afin que l'aperçu soit mis à jour. Par défaut, l'aperçu montre toutes les colonnes de l'objet sélectionné.

Puis cliquez sur **Next** pour continuer.

7.12.5. Étape 5 : Schéma final

L'étape 5 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 récupérer 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 nœud **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 la [Section 4.2.2.2, « Déposer des composants à partir du répertoire Metadata »](#).

7.13. Configurer un schéma générique

Talend Open Studio for Data Integration 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**.

7.13.1. 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.

7.13.2. 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 nœud **Generic Schemas** correspondant.

7.14. Configurer une connexion MDM

Talend Open Studio for Data Integration vous permet de centraliser les informations d'une ou plusieurs connexion(s) MDM sous le nœud **Metadata** de la vue **Repository**. Vous pouvez utiliser l'une de ces connexions établies pour vous connecter au serveur MDM.



Vous pouvez également paramétrer une connexion MDM en cliquant sur l'icône de la vue **Basic settings** des composants **tMDMInput** et **tMDMOutput**. Pour plus d'informations, consultez le **Guide de référence des Composants Talend Open Studio**.

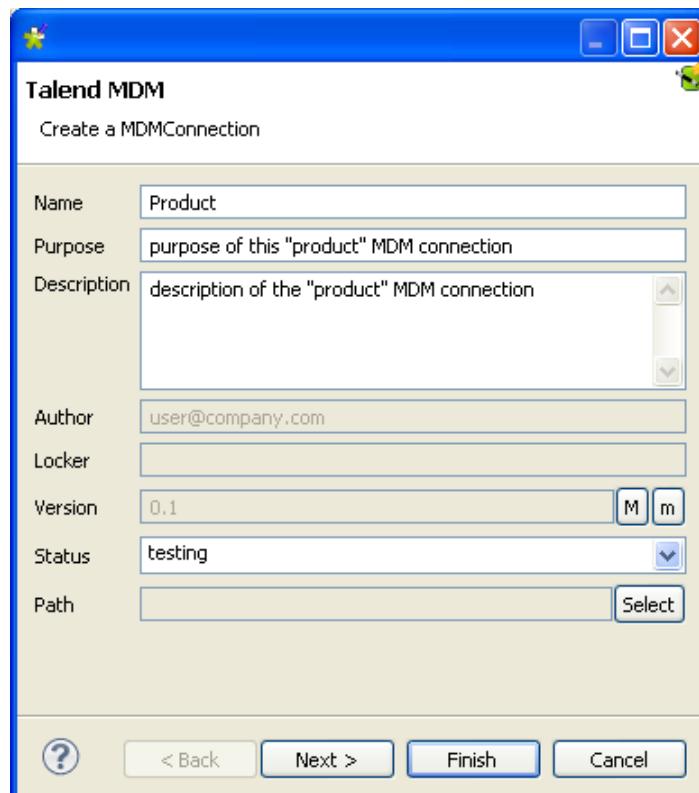
Selon l'option que vous sélectionnez, l'assistant vous permet de créer soit un schéma XML **Input** (d'entrée), soit un schéma XML **Output** (de sortie), soit un schéma XML **Receive** (de réception). Plus tard, dans un Job **Talend**, le composant **tMDMInput** utilise le schéma d'entrée défini pour 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.

7.14.1. Étape 1 : Définir la connexion

Pour établir une connexion MDM, procédez comme suit :

- Dans la vue **Repository**, développez le nœud **Metadata** et cliquez-droit sur **Talend MDM**.
- Sélectionnez **Create MDM** dans le menu contextuel.

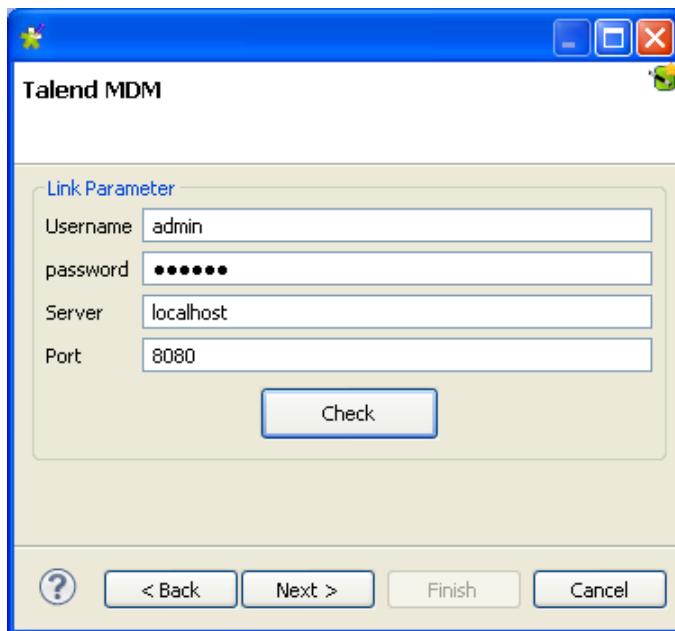
L'assistant de connexion s'affiche.



- Remplissez les champs **Name**, **Purpose** et **Description**. Le champ **Status** est un champ personnalisé pouvant être défini. Pour plus d'informations, consultez la [Section 2.6.8, « Définir les paramètres des Statuts »](#).

Étape 1 : Définir la connexion

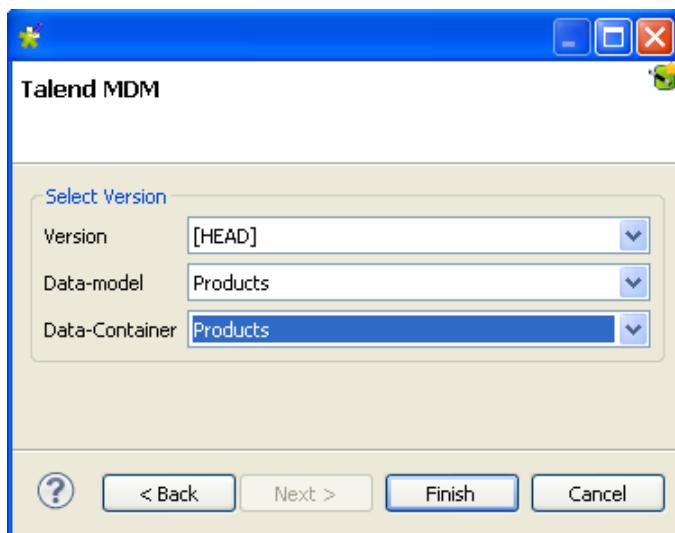
- Cliquez sur **Next** afin de passer à l'étape suivante.



- 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.

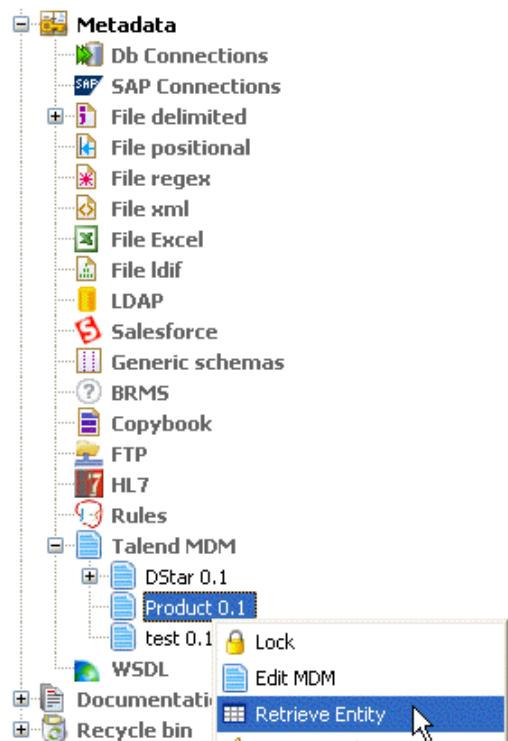
Une boîte de dialogue s'ouvre pour confirmer que votre connexion a été créée avec succès.

- 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.

7.14.2. Étape 2 : Définir un schéma MDM

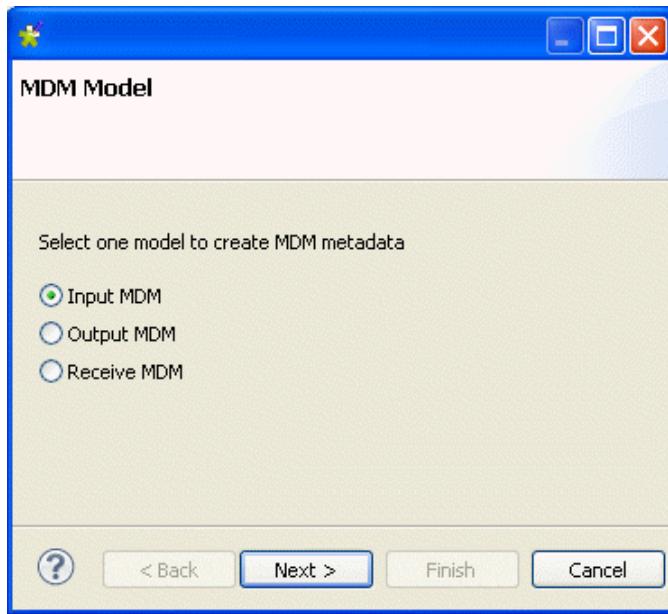
7.14.2.1. Définir un schéma d'entrée MDM

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 la [Section 7.14.2.2, « Définir un schéma de sortie MDM »](#).

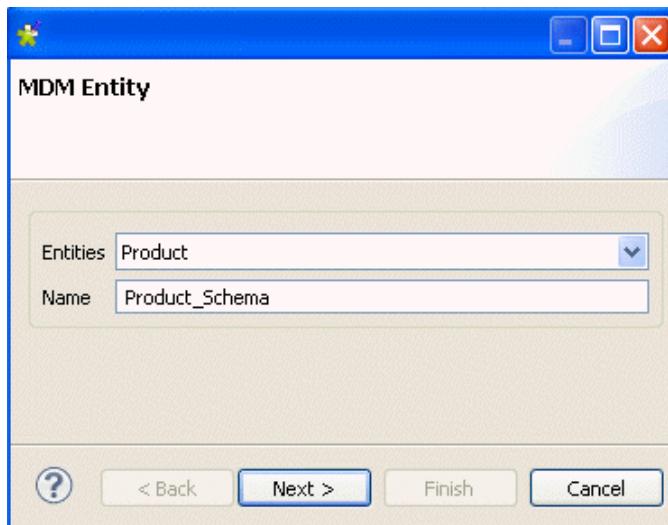
Pour définir les valeurs à retrouver à partir d'une ou plusieurs entité(s) liée(s) à une connexion MDM procédez comme suit :

- 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.

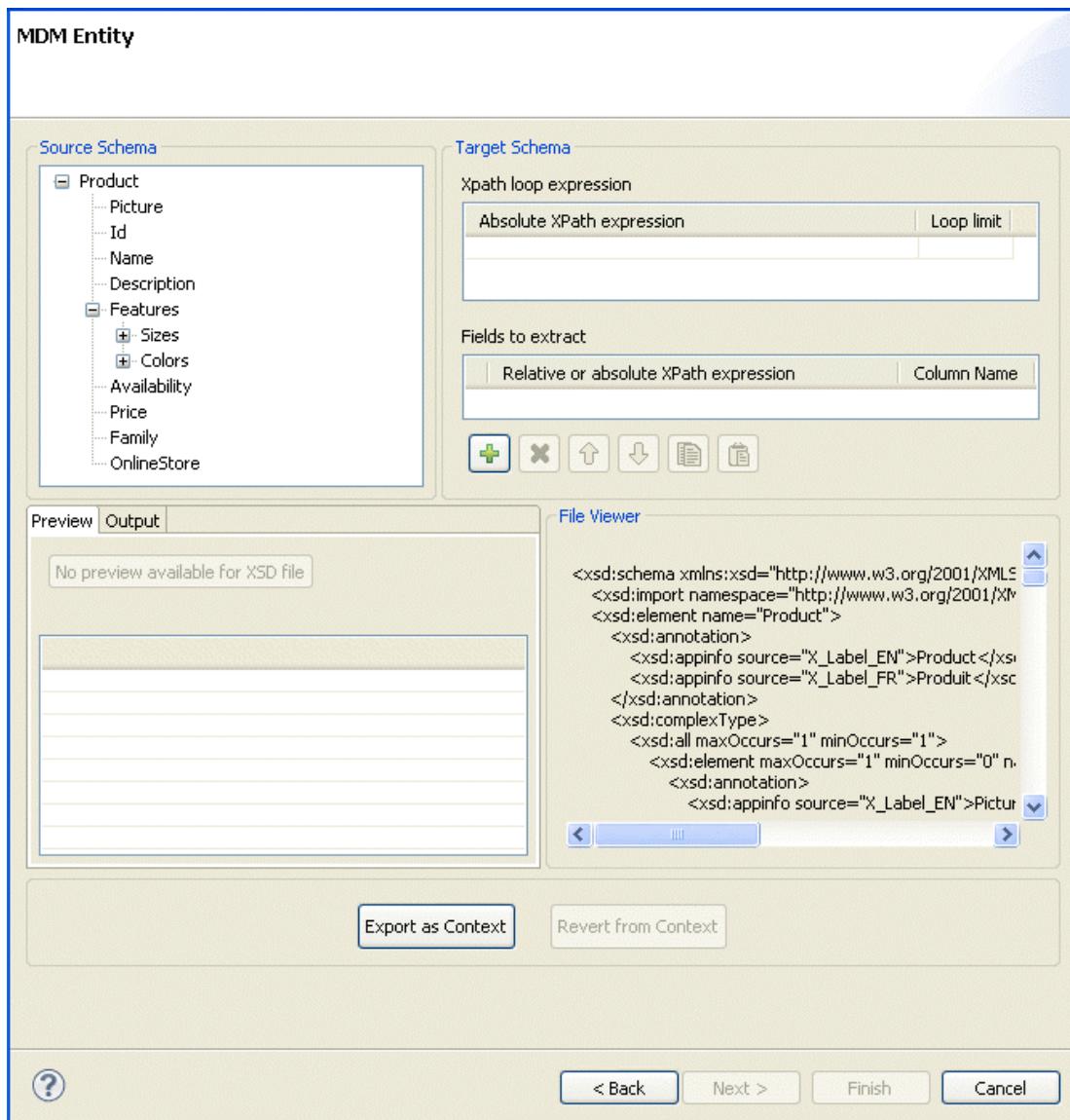


- Dans le champ Entities, sélectionnez l'entité métier (schéma XML) à partir de laquelle vous souhaitez récupérer les valeurs.

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 :

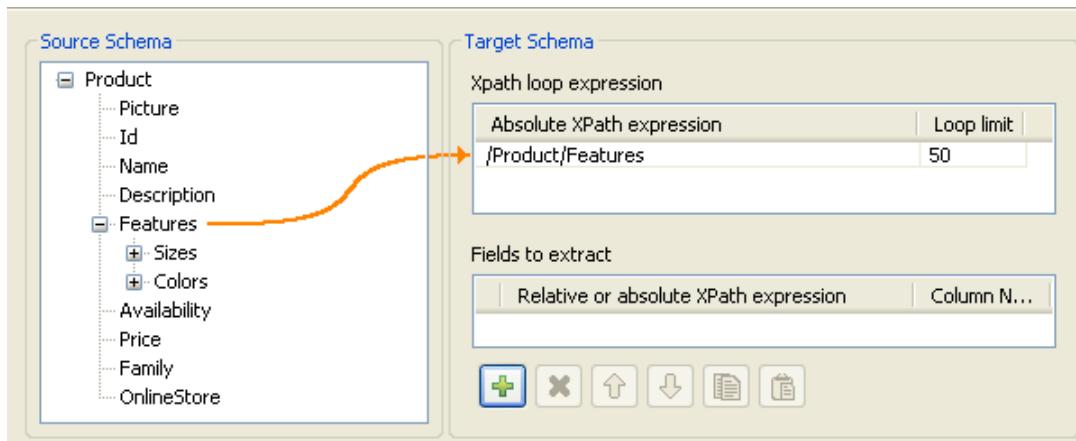
Panneau	Description
Source Schema	Arborescence de l'entité chargée.
Target schema	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible.
File viewer	Visionneur des données brutes.

- Dans la zone **Xpath loop expression**, saisissez l'expression absolue du chemin XPath vers le nœud de la structure XML sur lequel appliquer l'itération. Vous pouvez également glisser le nœud du schéma source dans le champ Xpath du schéma cible. Le lien s'affiche en orange.



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.

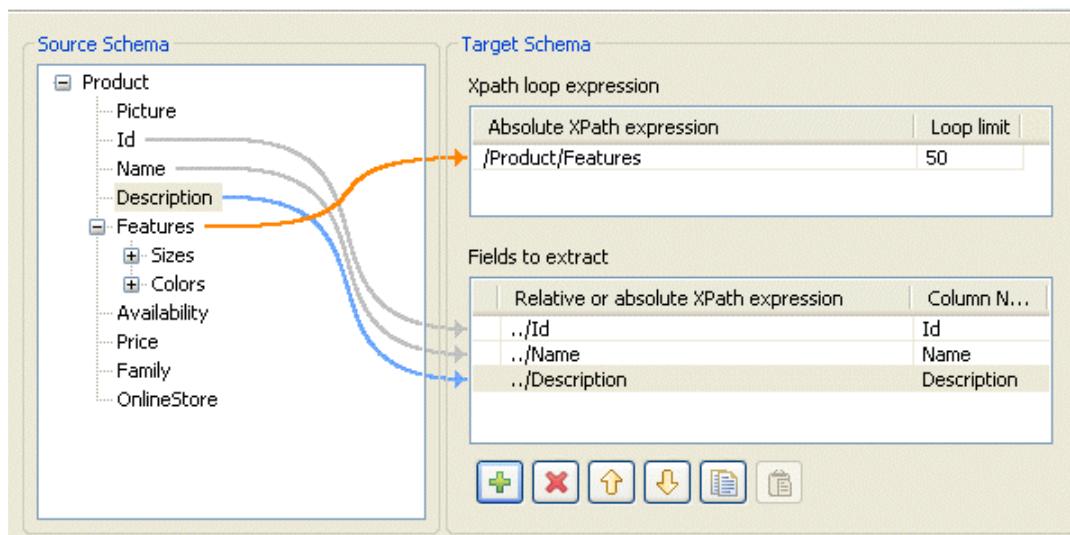


Dans la capture d’écran ci-dessus, utilisez *Features* comme élément sur lequel effectuer une boucle, puisqu’il est répété dans l’entité *Product* :

```
<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.

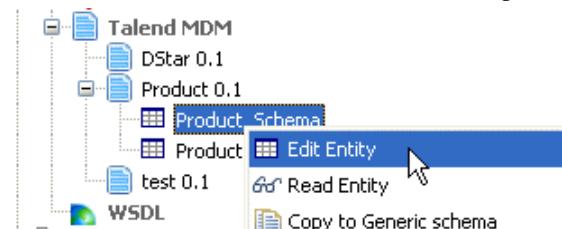
- Afin de définir les champs à extraire, glissez le noeud correspondant du schéma source dans le champ **Relative or absolute XPath expression**.



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 nœuds, 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.

Column	Key	Type	N..	Date Pattern (Ctrl...)	Length	Precision	Default	Commer
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.

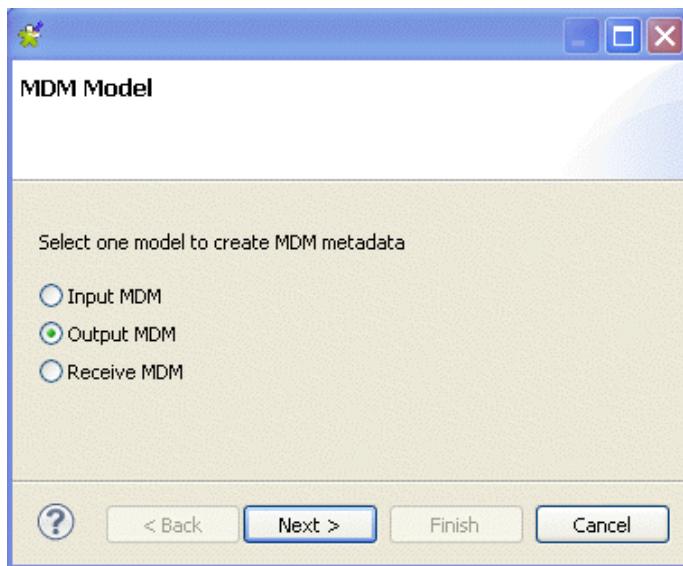
7.14.2.2. Définir un schéma de sortie MDM

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 la [Section 7.14.2.1, « Définir un schéma d'entrée MDM »](#).

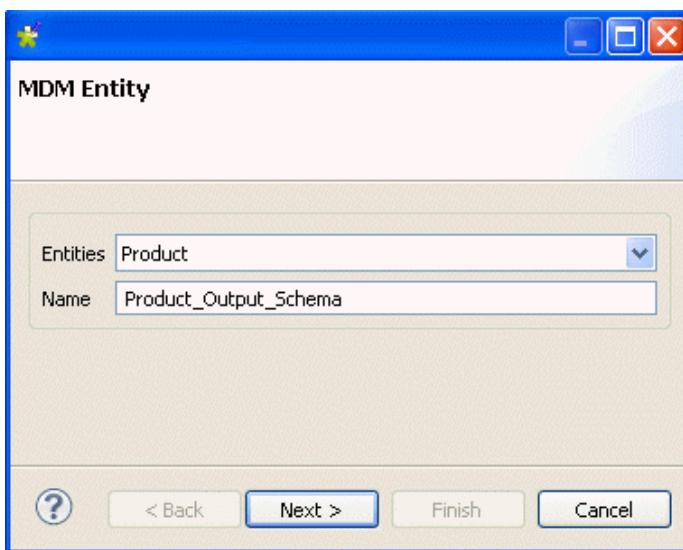
Pour définir les valeurs à écrire dans une ou plusieurs entité(s) liée(s) à une connexion MDM spécifique, procédez comme suit :

- 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.

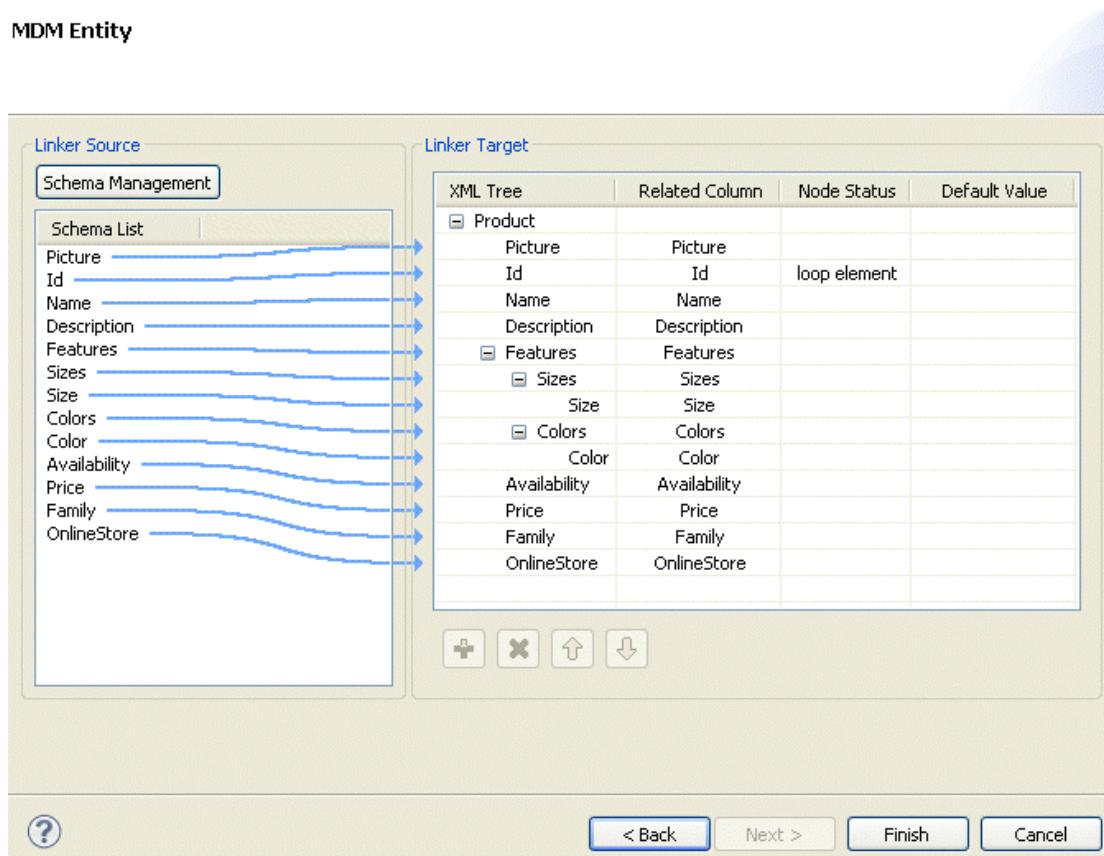


- Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) dans laquelle vous souhaitez écrire les valeurs.

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.

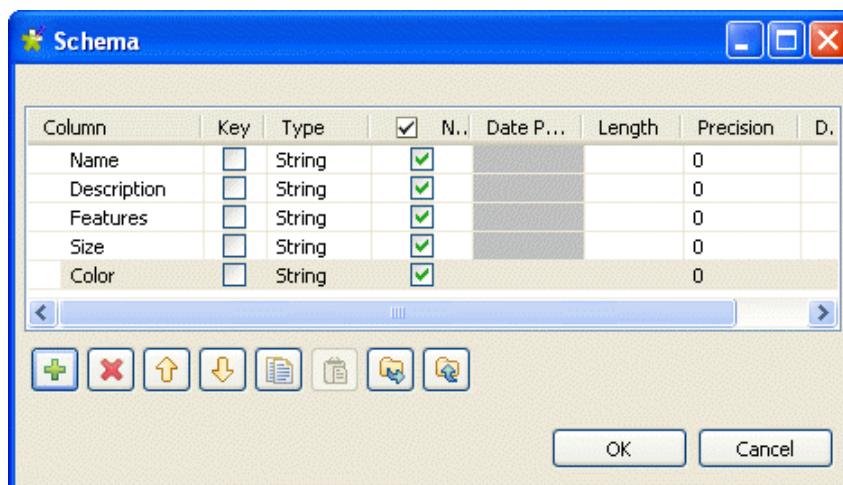
- Cliquez sur **Next** pour procéder à l'étape suivante.



💡 Un schéma identique à celui de l'entité sélectionnée est automatiquement créé dans le panneau **Linker Target**, et les colonnes sont mappées automatiquement du panneau source au panneau cible. L'assistant définit automatiquement l'élément Id comme étant l'élément de boucle. Vous pouvez toujours choisir d'effectuer une boucle sur un autre élément.

Vous pouvez définir les paramètres à prendre en compte lors de la définition du schéma XML.

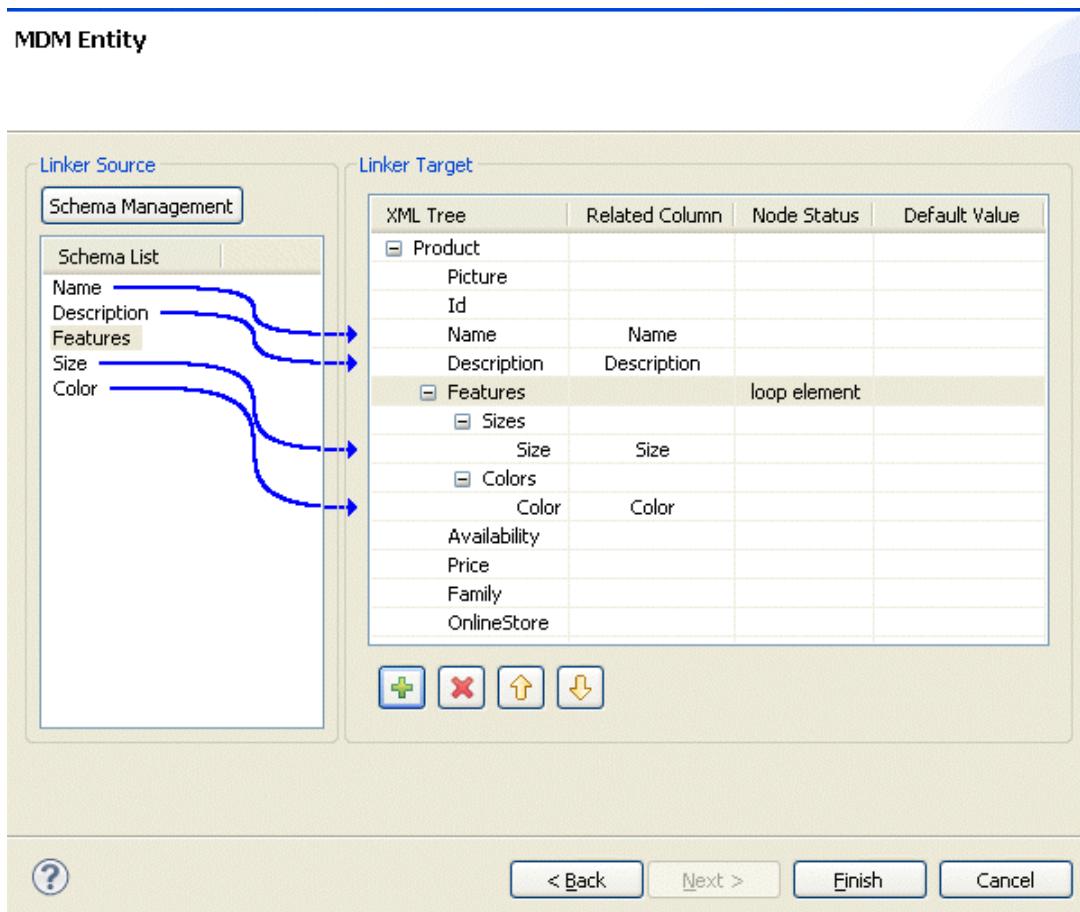
- Cliquez sur **Schema Management** pour ouvrir une boîte de dialogue.
- Effectuez les modifications nécessaires à la définition du schéma XML que vous souhaitez écrire dans l'entité sélectionnée.



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**.



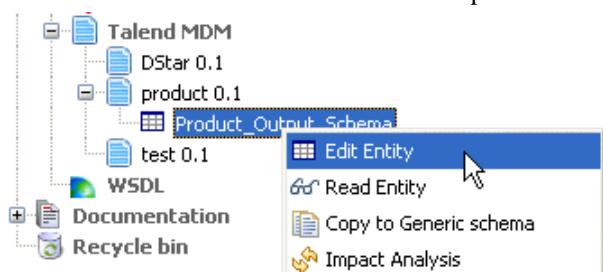
- Dans le panneau **Linker Target**, cliquez-droit sur l'élément que vous souhaitez définir comme élément de boucle et sélectionnez **Set as loop element**. Cela limitera l'itération à un ou plusieurs noeud(s).

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**.



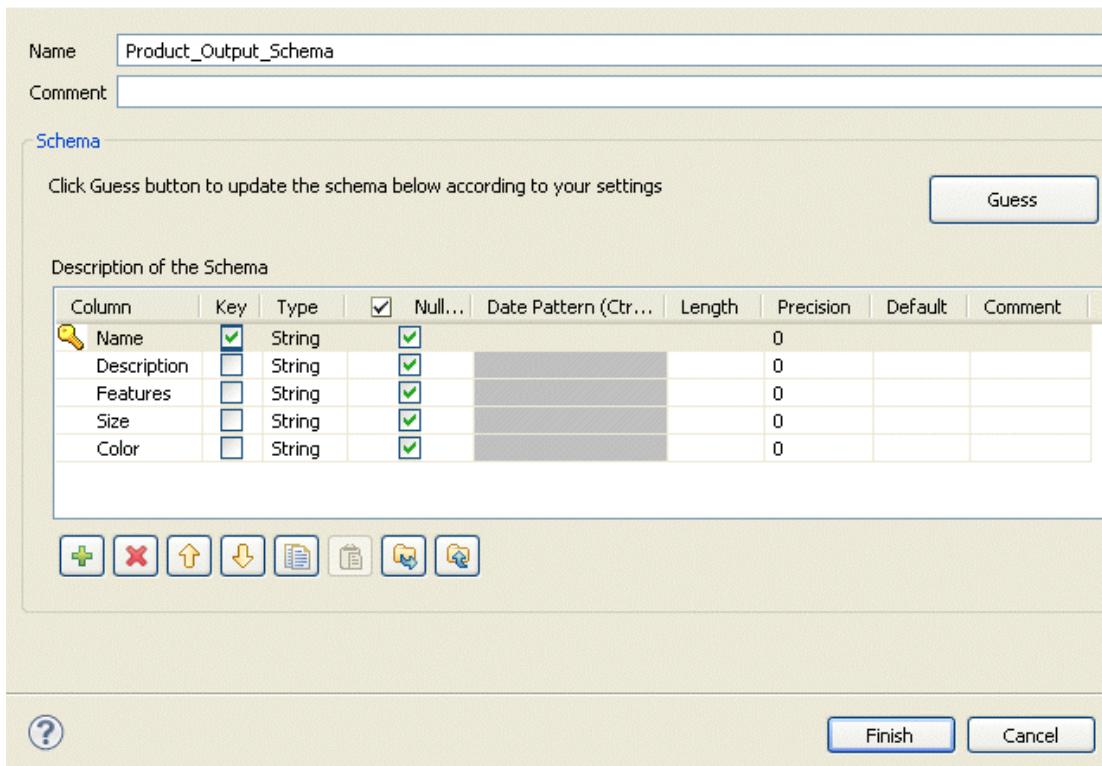
Pour modifier le schéma créé, procédez comme suit :

- Dans la vue **Repository**, développez les nœuds **Metadata** et **Talend MDM** puis parcourez le référentiel jusqu'au schéma que vous souhaitez modifier.

Étape 2 : Définir un schéma MDM

- 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** 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.

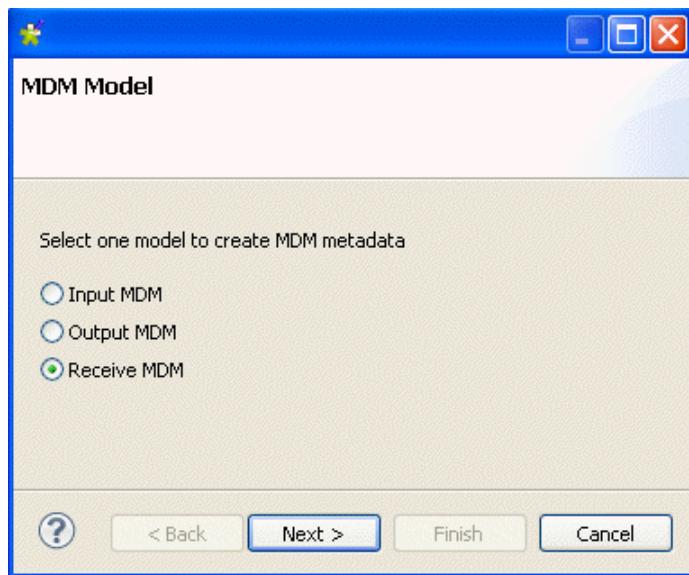
7.14.2.3. Définir un schéma de réception MDM

Cette section décrit la définition d'un schéma MDM XML de réception basé sur une connexion MDM.

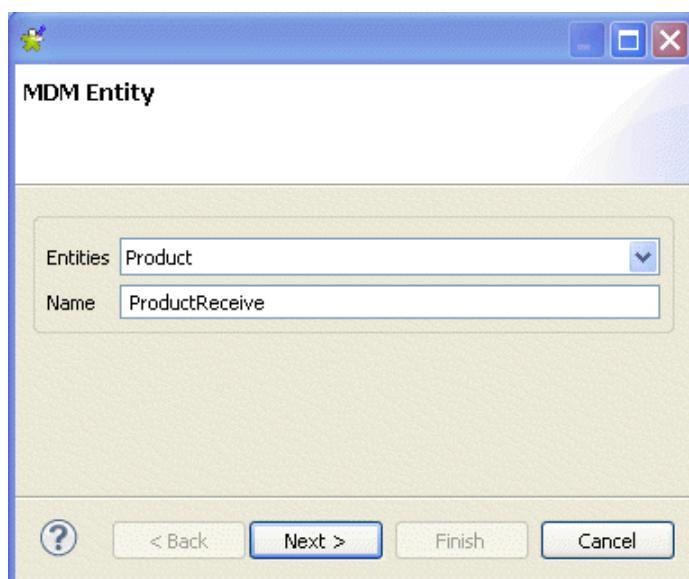
Pour définir le schéma XML que vous souhaitez recevoir selon une connexion MDM spécifique, procédez comme suit :

- Dans la vue **Repository**, développez le nœud **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.

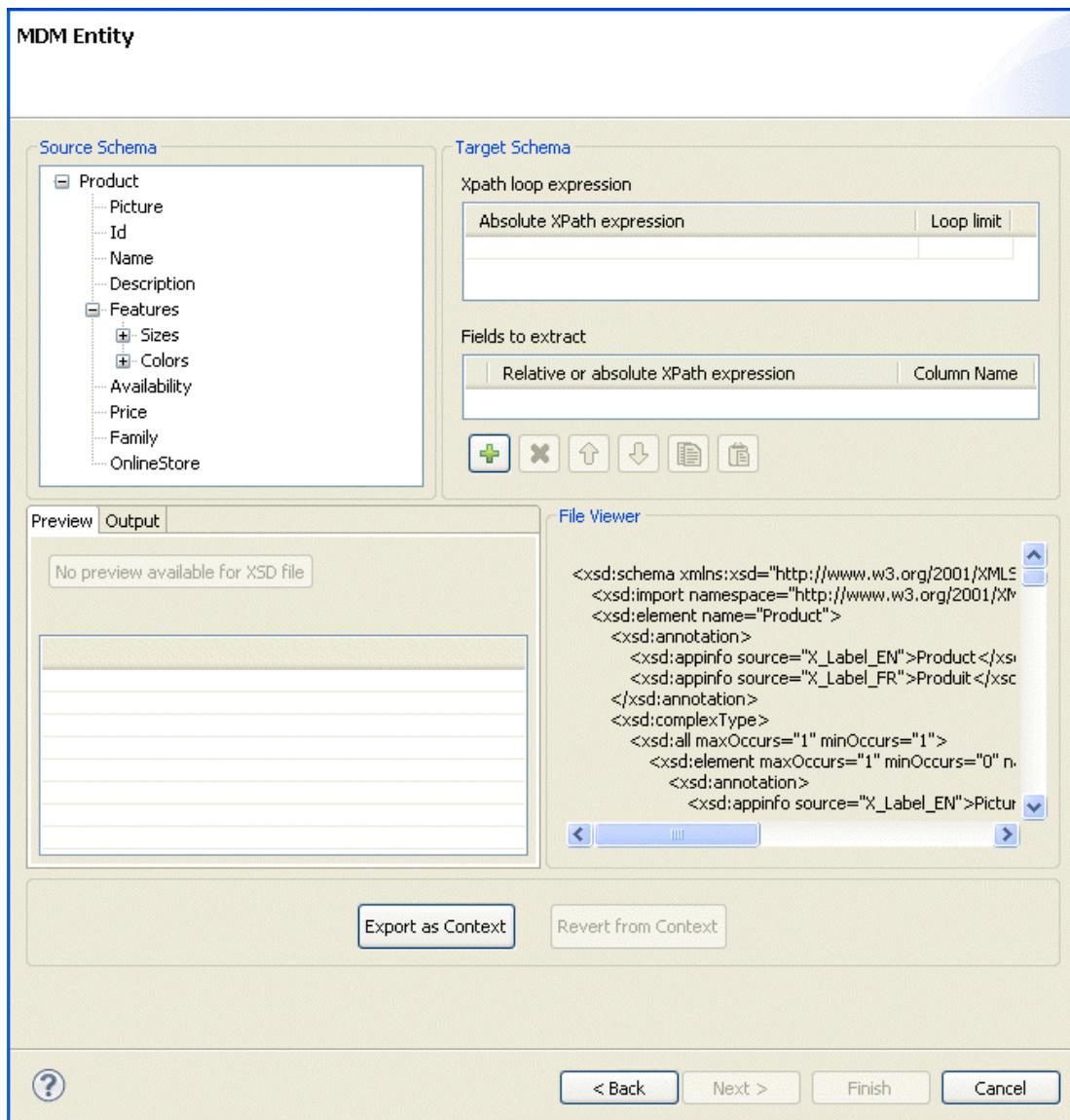


- Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) selon laquelle vous souhaitez recevoir le schéma XML.

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.



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 :

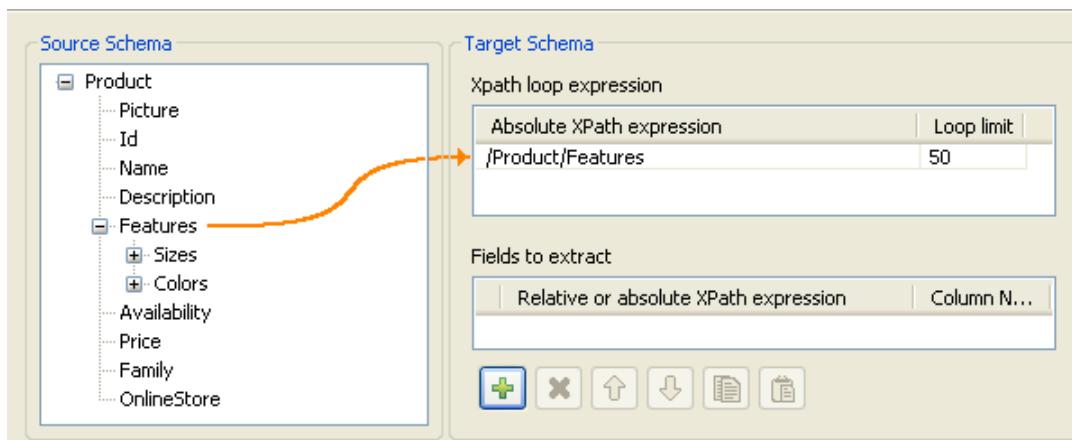
Panneau	Description
Source Schema	Arborescence de l'entité chargée.
Target schema	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible.
File viewer	Visionneur des données brutes.

- Dans la zone **Xpath loop expression**, saisissez l'expression absolue du chemin XPath vers le nœud de la structure XML sur lequel appliquer l'itération. Vous pouvez également glisser le nœud du schéma source dans le champ Xpath du schéma cible. Le lien s'affiche en orange.



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.

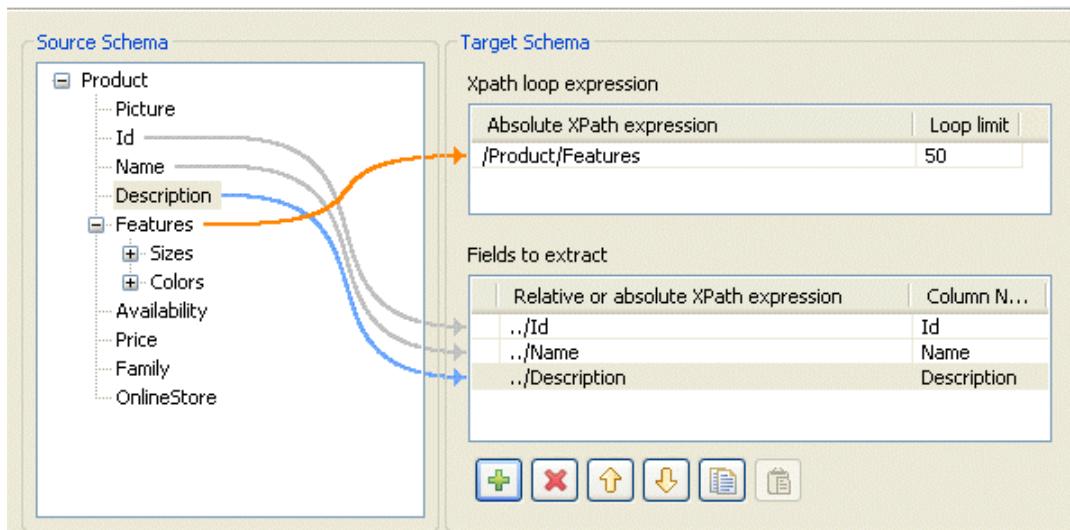


Dans la capture d’écran ci-dessus, utilisez *Features* comme élément sur lequel effectuer une boucle, puisqu’il est répété dans l’entité *Product* :

```
<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>
```

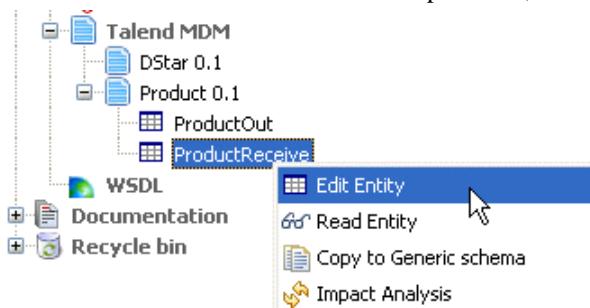
Ce faisant, le composant **tMDMReceive** utilisant la connexion MDM va créer une nouvelle ligne pour chaque élément Feature rencontré.

- Afin de définir les champs à extraire, glissez le nœud 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 nœuds, 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.

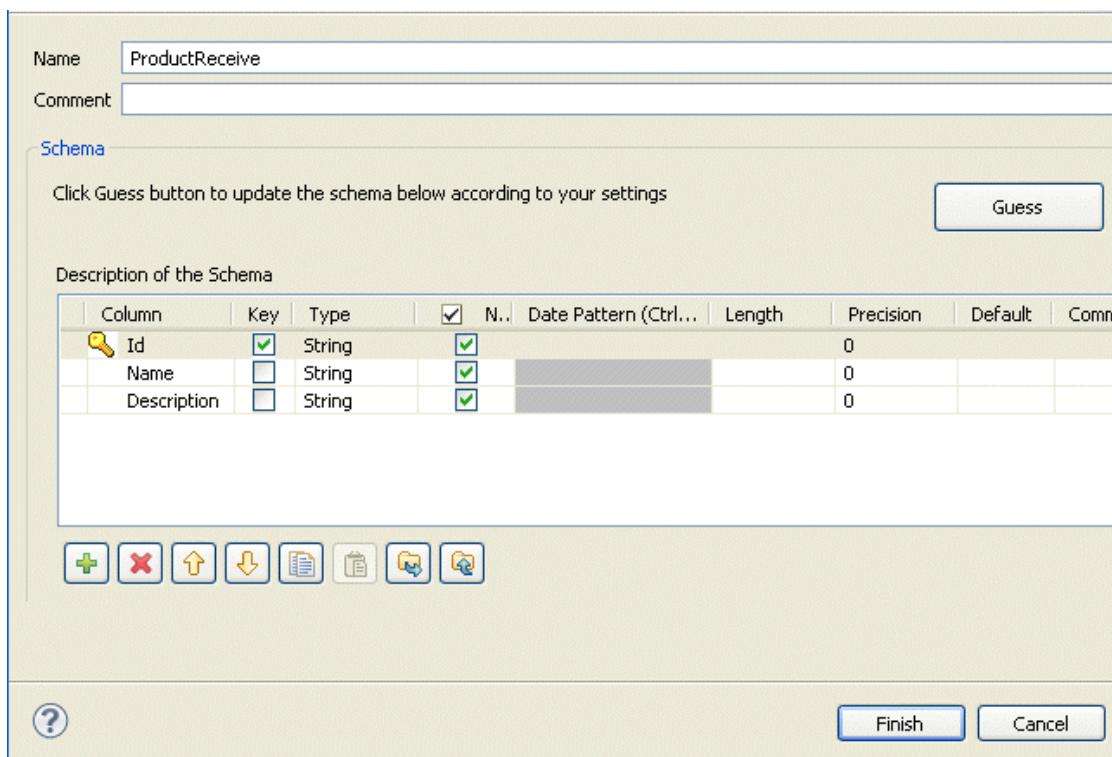
Le nouveau schéma créé s'affiche sous la connexion MDM correspondante, dans la vue **Repository**.



Pour modifier le schéma créé, procédez comme suit :

- Dans la vue **Repository**, développez les nœuds **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.



- 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.

7.15. Configurer un schéma Web Service

Talend Open Studio for Data Integration vous permet de stocker vos connexions Web Service dans le **Repository**.

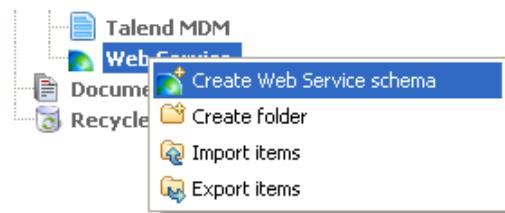
7.15.1. Configurer un schéma simple

Cette section décrit la définition d'un schéma Web Service.

7.15.1.1. Étape 1 : Propriétés générales

Dans cette étape, vous allez définir les propriétés générales du fichier.

- Dans la vue **Repository**, développez le nœud **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**.

The dialog box is titled 'Create new Web Service schema' and 'File - Step 1 of 4'. It contains the following fields:

- Name: SimpleGetWeather
- Purpose: (empty)
- Description: (empty)
- Author: user@company.com
- Locker: (empty)
- Version: 0.1 (with M and m buttons)
- Status: (empty)
- Path: (empty) with a 'Select' button

At the bottom are buttons: '?', '< Back', 'Next >', 'Finish', and 'Cancel'.

- Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

7.15.1.2. Étape 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

Need authentication?

User Password

Use http proxy

host Port

user password

Method

Time Out(Second)

Parameters

Value
Melbourne

Add 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.

Preview Output

Refresh Preview Web Service Explorer

GetWeather |
Sunny

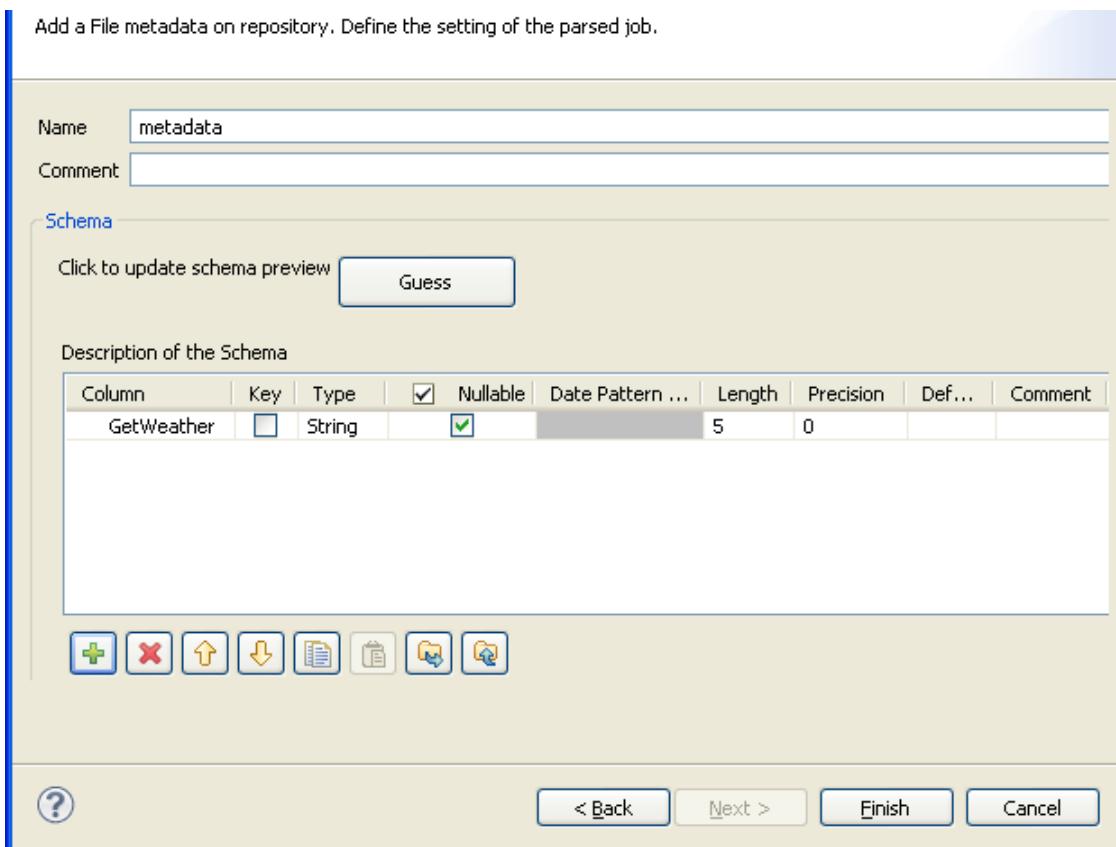
Export as Context Revert from Context

< Back Next > Finish Cancel

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

7.15.1.3. Étape 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.



- 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 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**.

7.16.1. Étape 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 la [Section 2.5, « Configuration des préférences de Talend Open Studio for Data Integration »](#).
- Cliquez sur **Next** une fois terminé. L'étape suivante vous permet de renseigner les données de connexion au serveur FTP.

7.16.2. Étape 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 nœud des connexions aux serveurs FTP, dans la vue **Repository**.

Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d'informations concernant l'action de déposer des métadonnées dans l'espace de modélisation graphique, consultez la [Section 4.2.2.2, « Déposer des composants à partir du répertoire Metadata »](#).

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.

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 D, Routines système](#).

Avant de commencer un processus métier, vous devez vous être familiarisé avec l'interface graphique de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez l'[Annexe A, Interface de Talend Open Studio for Data Integration](#).

8.1. Définition des routines

Les routines sont des fonctions Java 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 permettent de faire remonter les erreurs Java directement dans le Studio, facilitant ainsi l'identification et la résolution des problèmes au cours du développement de vos processus d'intégration avec **Talend**.

Les routines peuvent être de deux types :

- 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 la [Section 8.2, « Accès aux routines système »](#).

Pour plus d'informations concernant la création des routines utilisateur, consultez la [Section 8.4.1, « Créer des routines personnalisées »](#).



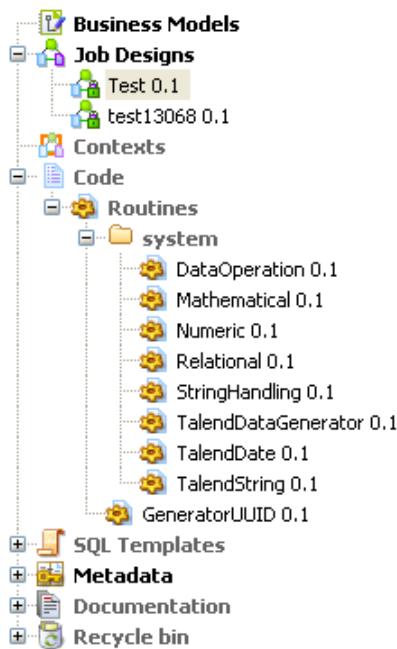
Vous pouvez également définir les dépendances des routines dans des Jobs. Pour ce faire, cliquez-droit sur un Job dans la vue **Repository** et sélectionnez **Set up routine dependencies**. Dans la boîte de dialogue qui s'ouvre, toutes les routines sont définies par défaut. Vous pouvez utiliser la barre d'outils afin de supprimer les routines, si nécessaire.

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.

- Appuyez sur **Ctrl + O** dans la routine ouverte.

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 D, Routines système](#).

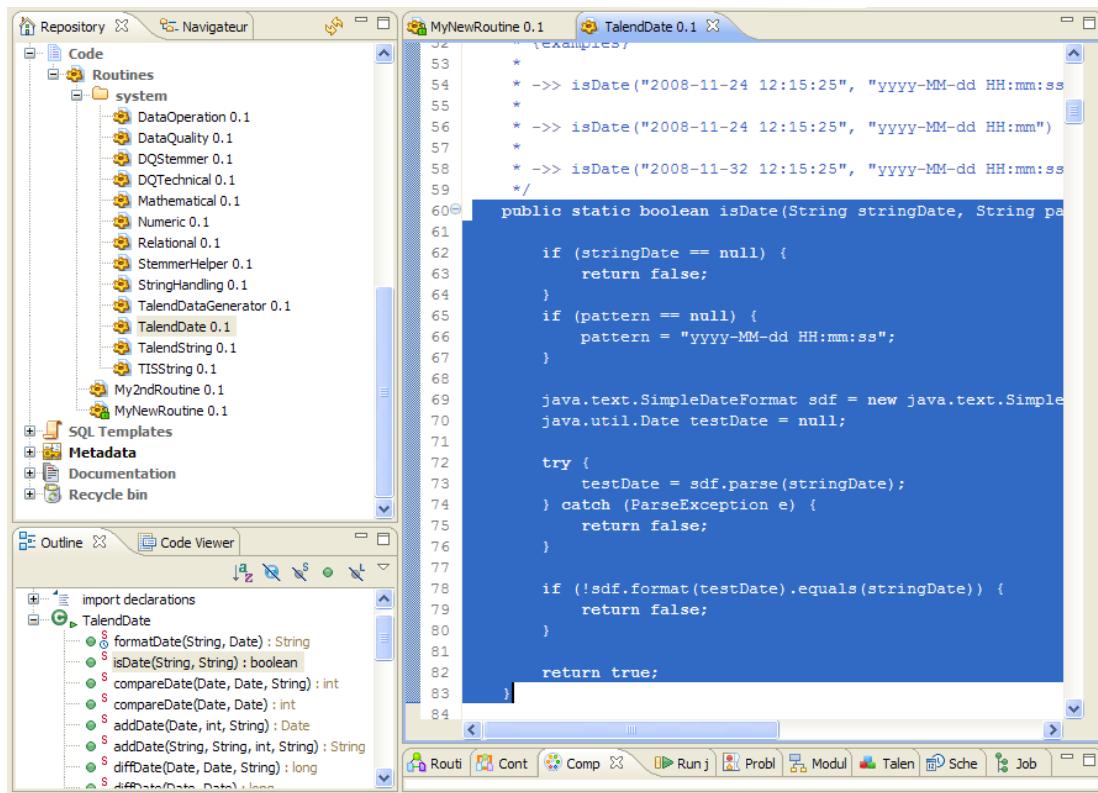
8.3. Personnalisation des routines système

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 :

- Créez d'abord une routine utilisateur en suivant les étapes décrites dans la [Section 8.4.1, « Créez des routines personnalisées »](#). 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 for Data Integration 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.

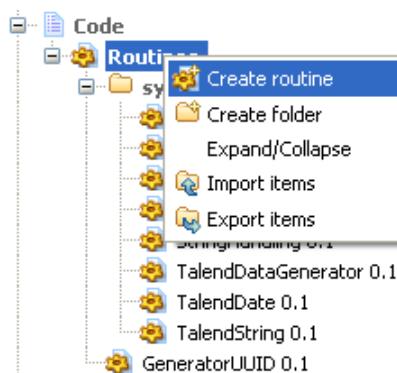
8.4.1. Créer des routines personnalisées

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.

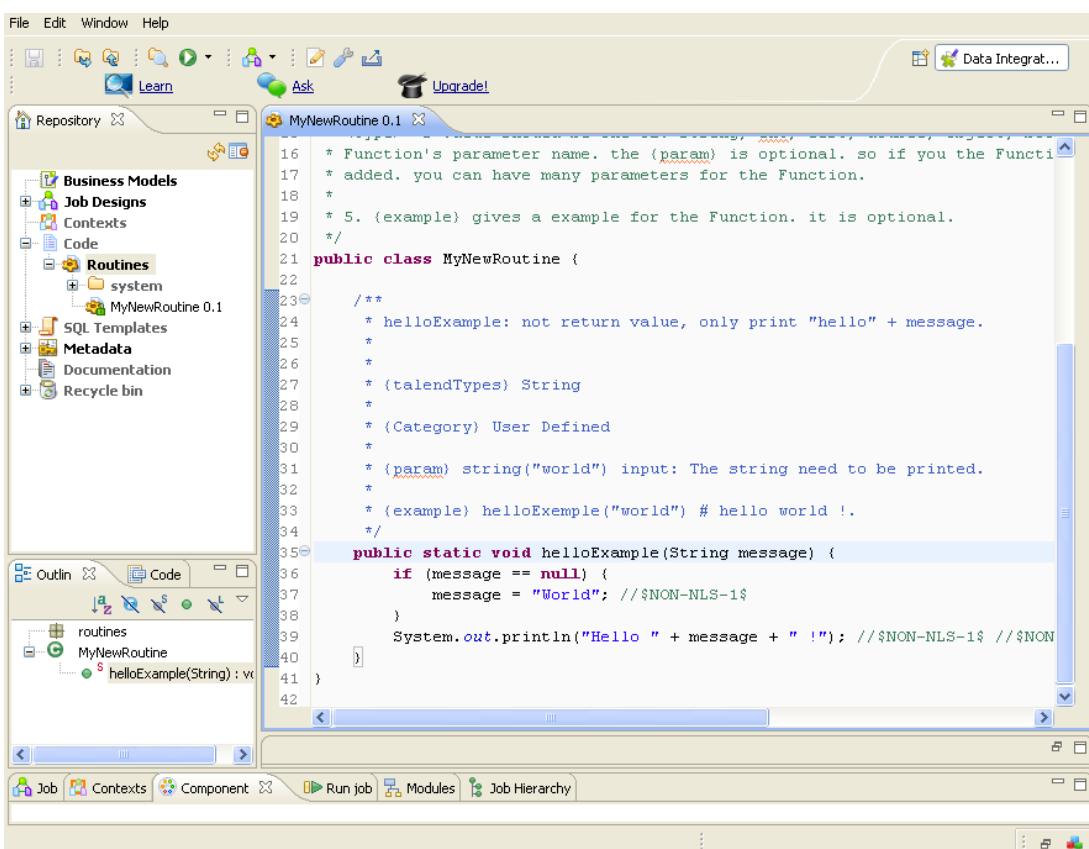
Pour créer un nouvelle routine :

- 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.



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 + " !");
```

- daptez 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 la [Section 8.3, « Personnalisation des routines système »](#).

8.4.2. Éditer 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 [Section 8.3, « Personnalisation des routines système »](#).

8.4.3. Éditer les bibliothèques des routines utilisateur

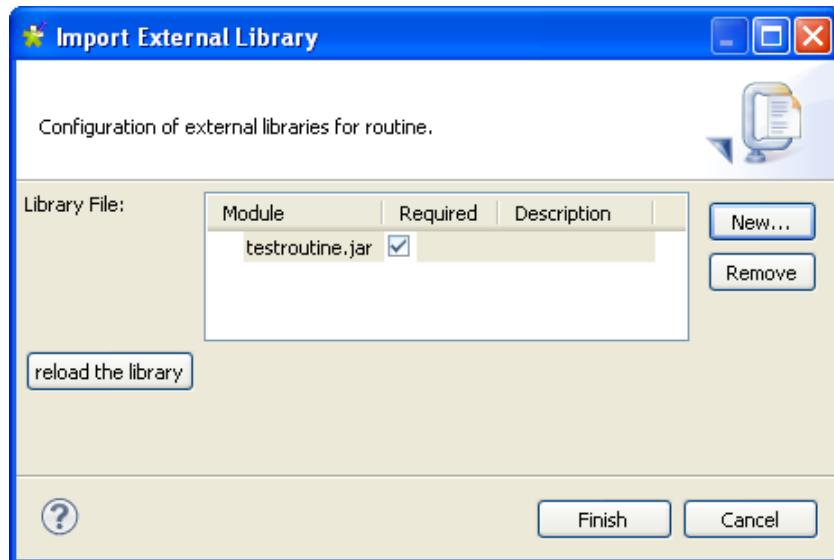
Vous pouvez éditer la bibliothèque de chaque routine utilisateur en important des fichiers .jar externes pour la routine sélectionnée. Ces fichiers externes seront affichés, comme les modules, dans la vue **Modules** de votre Studio. Pour plus d'informations concernant la vue **Modules**, consultez [Section 4.5.4, « Installer des modules externes »](#).

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 :

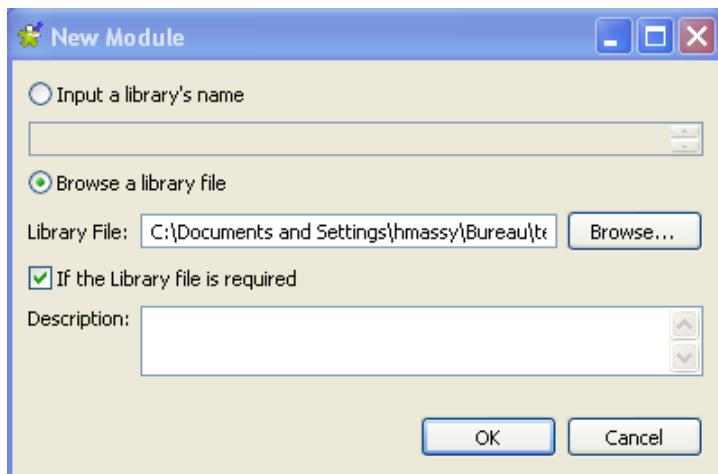
- Dans la vue **Repository**, développez les nœuds **Code > Routines**.
- Cliquez-droit sur la routine utilisateur dont vous souhaitez éditer la bibliothèque, puis sélectionnez **Edit Routine Library**.

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.

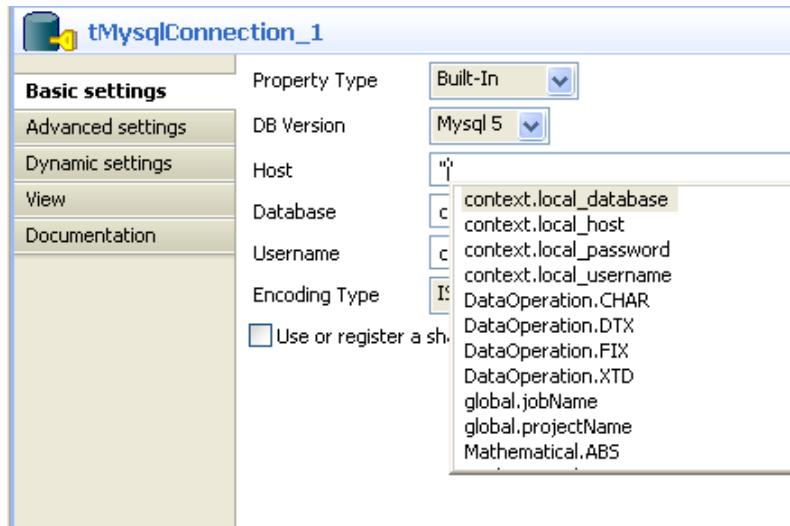
Pour plus d'informations concernant la vue **Modules**, consultez la [Section 4.5.4, « Installer des modules externes »](#).

8.5. Appel d'une routine à partir d'un Job

Prérequis : Pour pouvoir exécuter votre routine, vous devez avoir créé au minimum un Job. Pour plus d'informations sur la création d'un Job, consultez la [Section 4.2.1, « Créer un Job » du Guide Utilisateur Talend Open Studio for Data Integration](#).

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 suivi 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

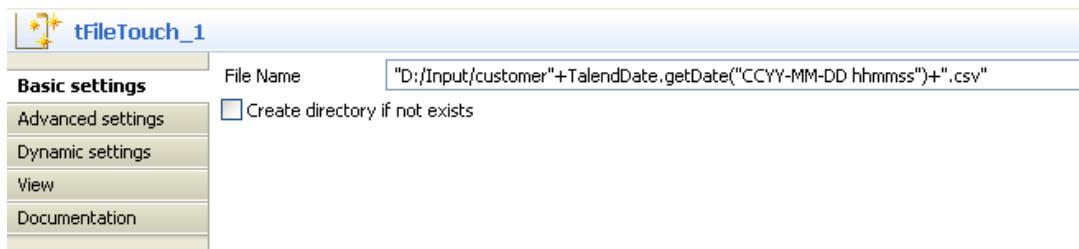
Le scénario suivant illustre l'utilisation d'une routine. Le Job se compose d'un composant unique qui appelle une routine système.



- 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.
- 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 **TalendDate** 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 appelée *GetDate*.



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 for Data Integration*. Pour plus d'informations, consultez la [Annexe A, Interface de Talend Open Studio for Data Integration](#).

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 for Data Integration*, 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 [Section 4.3.1, « Types de connexions »](#).

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 for Data Integration* fournit de nombreux modèles SQL pour simplifier les tâches les plus communes. Il comprend également un éditeur SQL vous permettant de personnaliser ou de créer vos propres modèles SQL, afin de répondre à vos besoins.

Ces modèles SQL sont utilisés avec les composants de la famille **Talend** ELT, tels que le **tSQLTemplate**, le **tSQLTemplateFilterColumns**, le **tSQLTemplateCommit**, le **tSQLTemplateFilterRows**, le **tSQLTemplateRollback**, le **tSQLTemplateAggregate** et le **tSQLTemplateMerge**. Ces composants exécutent les instructions SQL sélectionnées. 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 for Data Integration fournit les types suivants de modèles SQL, sous le nœud **SQL templates** de la vue **Repository** :

- Modèles SQL **system** : ils sont classés selon le type de base de données.
- Modèles SQL personnalisés : ce sont les modèles que vous avez créés ou adaptés des modèles existants.

Des informations plus précises concernant les modèles SQL sont présentées dans les sections suivantes.

Pour plus d'informations concernant les composants de la famille ELT, consultez le **Guide de Référence** des composants de *Talend Open Studio for Data Integration*.



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 for Data Integration 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 prédefinis.

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.

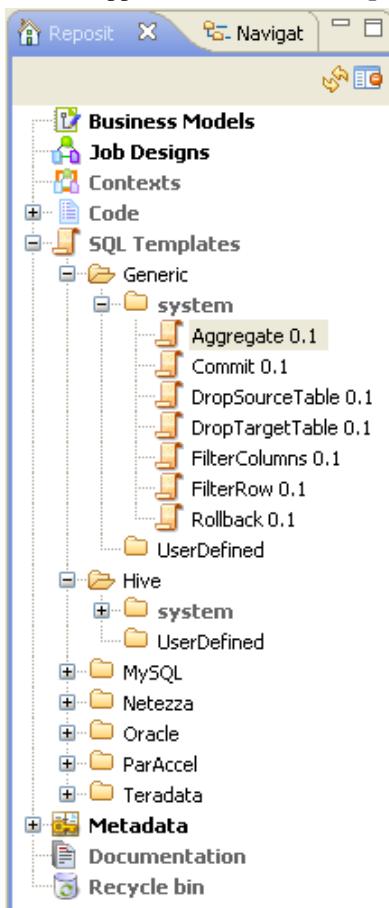
Le tableau ci-dessous présente ces types ainsi que les informations liées à chacun.

Nom	Fonction	Composants associés	Paramètres requis dans le composant
<i>Aggregate</i>	Réalise l'agrégation (sum, average, count, etc.) d'un ensemble de données.	tSQLTemplateAggregate	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.	tSQLTemplate tSQLTemplateAggregate tSQLTemplateCommit tSQLTemplateFilterColumns tSQLTemplateFilterRows tSQLTemplateMerge tSQLTemplateRollback	Null
<i>Rollback</i>	Envoie une instruction Rollback au SGBDR.	tSQLTemplate tSQLTemplateAggregate tSQLTemplateCommit tSQLTemplateFilterColumns tSQLTemplateFilterRows tSQLTemplateMerge tSQLTemplateRollback	Null
<i>DropSourceTable</i>	Supprime une source table.	tSQLTemplate tSQLTemplateAggregate tSQLTemplateFilterColumns tSQLTemplateFilterRows	Nom de la table (lors de l'utilisation du tSQLTemplate) Nom de la table source
<i>DropTargetTable</i>	Supprime une table cible.	tSQLTemplateAggregate tSQLTemplateFilterColumns tSQLTemplateFilterRows	Nom de la table cible
<i>FilterColumns</i>	Sélectionne et extrait un ensemble de données dans des colonnes données du SGBDR.	tSQLTemplateAggregate tSQLTemplateFilterColumns tSQLTemplateFilterRows	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.	tSQLTemplateFilterRows	Nom de la table cible (et schéma)

				Nom de la table source (et schéma)
				Conditions
<i>MergeInsert</i>	Insère des enregistrements d'une table source vers une table cible.	tSQLTemplateMerge	tSQLTemplateCommit	Nom de la table cible (et schéma)
				Nom de la table source (et schéma)
				Conditions
<i>MergeUpdate</i>	Met à jour la table cible avec les enregistrements de la table source.	tSQLTemplateMerge	tSQLTemplateCommit	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 nœud **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 for Data Integration*.



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 nœud **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.

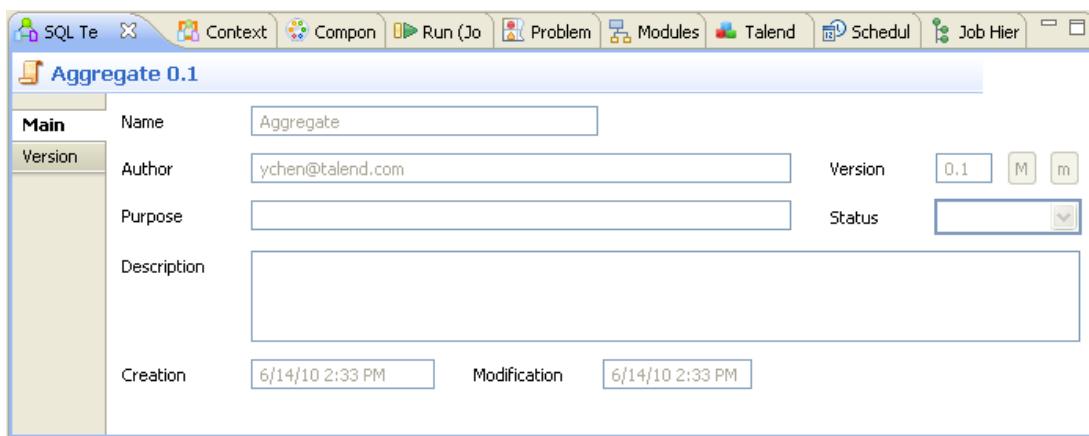
```

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 [Section 9.2, « Définition des SQL Templates Talend »](#).

- A chaque fois que vous cliquez sur un modèle SQL ou que vous ouvrez un modèle SQL, la vue des propriétés correspondante à ce modèle s'affiche en bas du Studio. Par exemple, vous cliquez le modèle *aggregate* ou vous l'ouvrez, ses propriétés seront présentées comme suit :

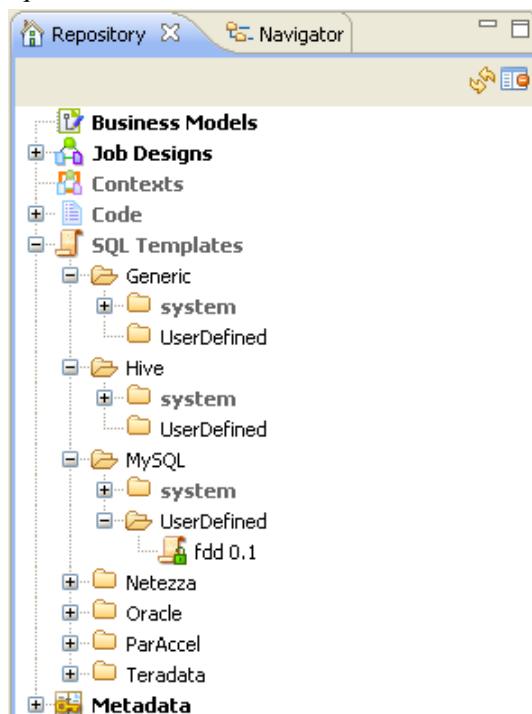


Pour plus d'informations concernant les types de modèles SQL, consultez [Section 9.3.1, « Les types de SQL Templates système »](#).

Pour plus d'informations concernant l'utilisation des modèles SQL avec les composants associés, consultez [Section 4.4.3, « Utiliser les SQL Templates »](#).

9.3.3. Créer des SQL Templates personnalisés

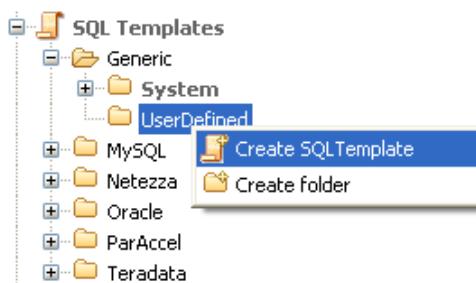
Comme la transformation que vous devez effectuer en mode ELT peut dépasser les possibilités des modèles SQL, *Talend Open Studio for Data Integration* vous permet de développer vos propres modèles SQL, à condition de respecter quelques règles d'écriture. Ces modèles SQL sont stockés dans les dossiers **User-defined**, regroupés selon la base de données dans laquelle ils seront utilisés.



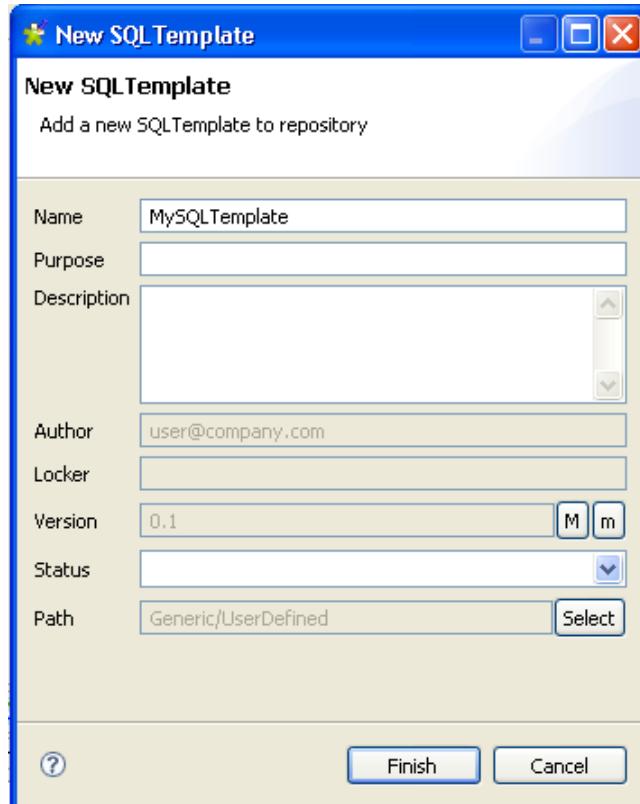
Pour plus d'informations concernant les règles d'écriture des modèles SQL, consultez l'[Annexe C, Règles d'écriture des SQL Templates](#).

Pour créer un modèle SQL personnalisé :

- Dans la vue **Repository**, développez le nœud **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 nœud **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 **Guide de Référence Talend Open Studio for Data Integration**.

9.3.4. Exemple d'utilisation des SQL Templates système

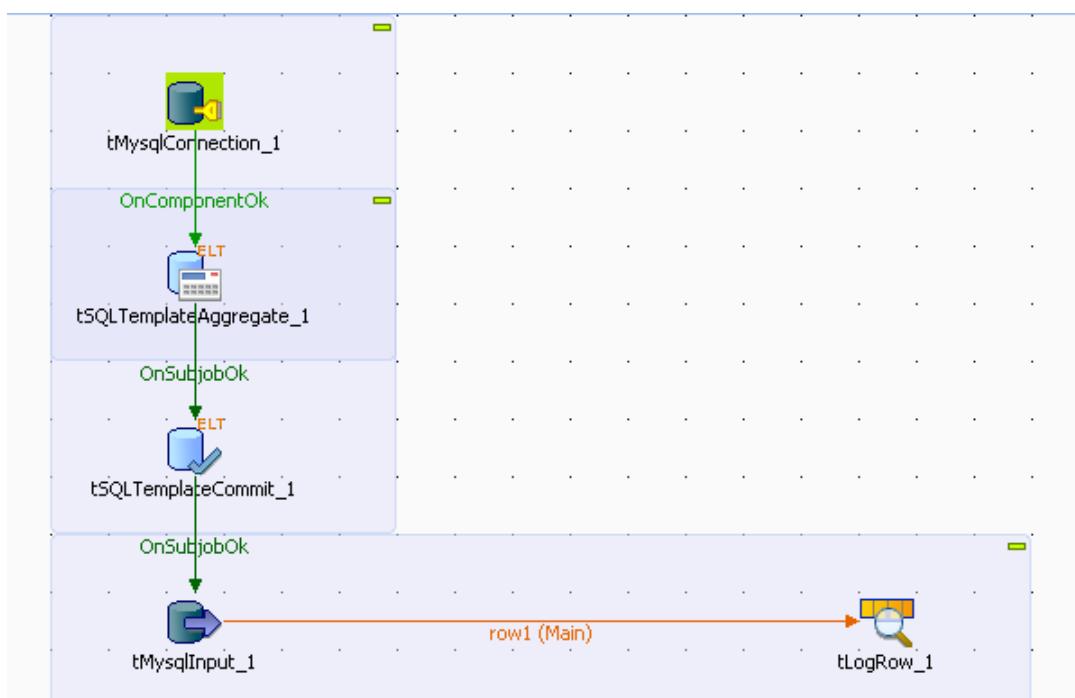
Puisque beaucoup d'instructions SQL sont communes, standardisées, *Talend Open Studio for Data Integration* vous permet de bénéficier de plusieurs modèles SQL système.

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 :

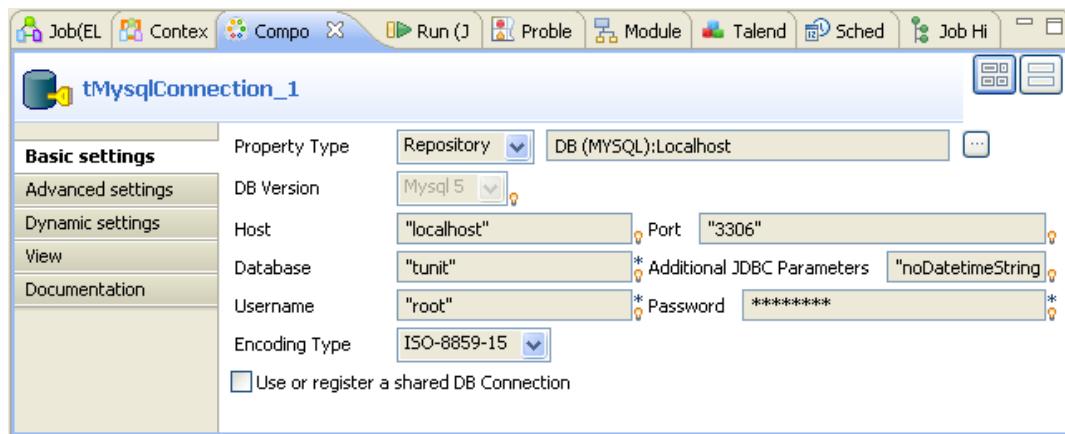
- ouvre une connexion à une base de données MySQL,
- collecte des données regroupées par valeurs spécifiques à partir d'une table de la base de données et écrit des données agrégées dans une table cible de la base de données,
- supprime la table source d'où proviennent les données agrégées,
- lit la 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 :

- A partir de la **Palette**, glissez les composants suivants dans l'espace de modélisation graphique : **tMysqlConnection**, **tSQLTemplateAggregate**, **tSQLTemplateCommit**, **tMysqlInput**, et **tLogRow**.
- Cliquez-droit sur le **tMysqlConnection** et dans le menu contextuel, sélectionnez **Trigger > OnComponentOk** afin de lier le **tMysqlConnection** au **tSQLTemplateAggregate**.
- Connectez le **tSQLTemplateAggregate**, le **tSQLTemplateCommit** et le **tMysqlInput** à l'aide de liens **On-ComponentOk**.
- Connectez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Main > Row**.

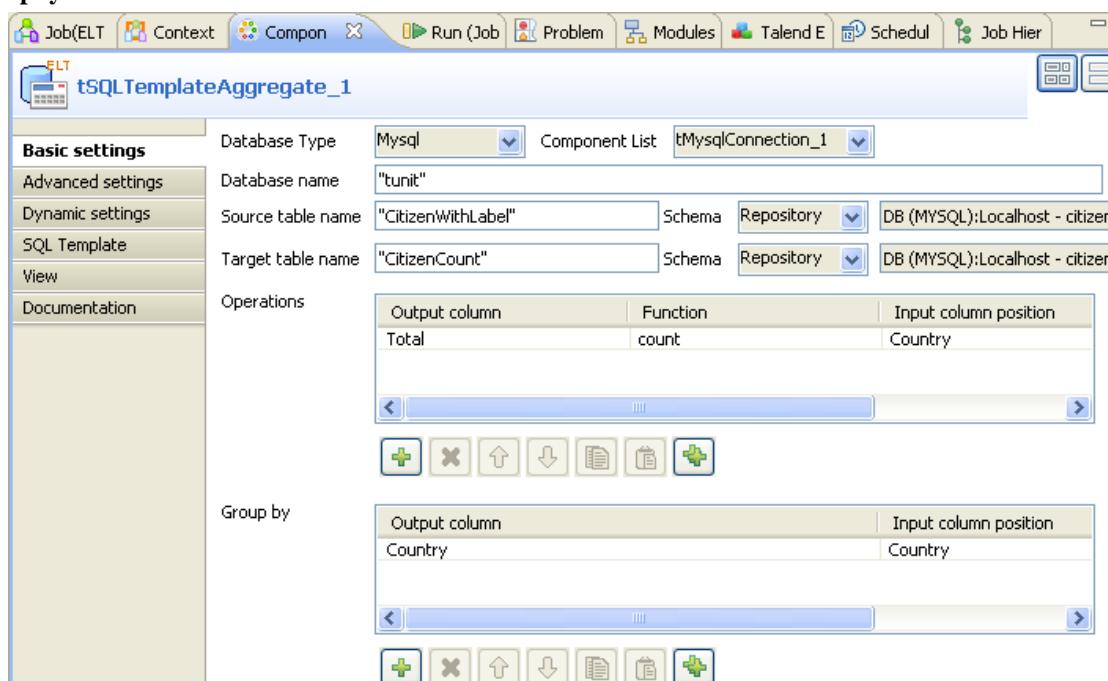


- Dans l'espace de modélisation graphique, sélectionnez le composant **tMysqlConnection** et cliquez sur l'onglet **Component** pour définir ses propriétés.
- Dans la vue **Basic settings**, configurez manuellement les paramètres de connexion.



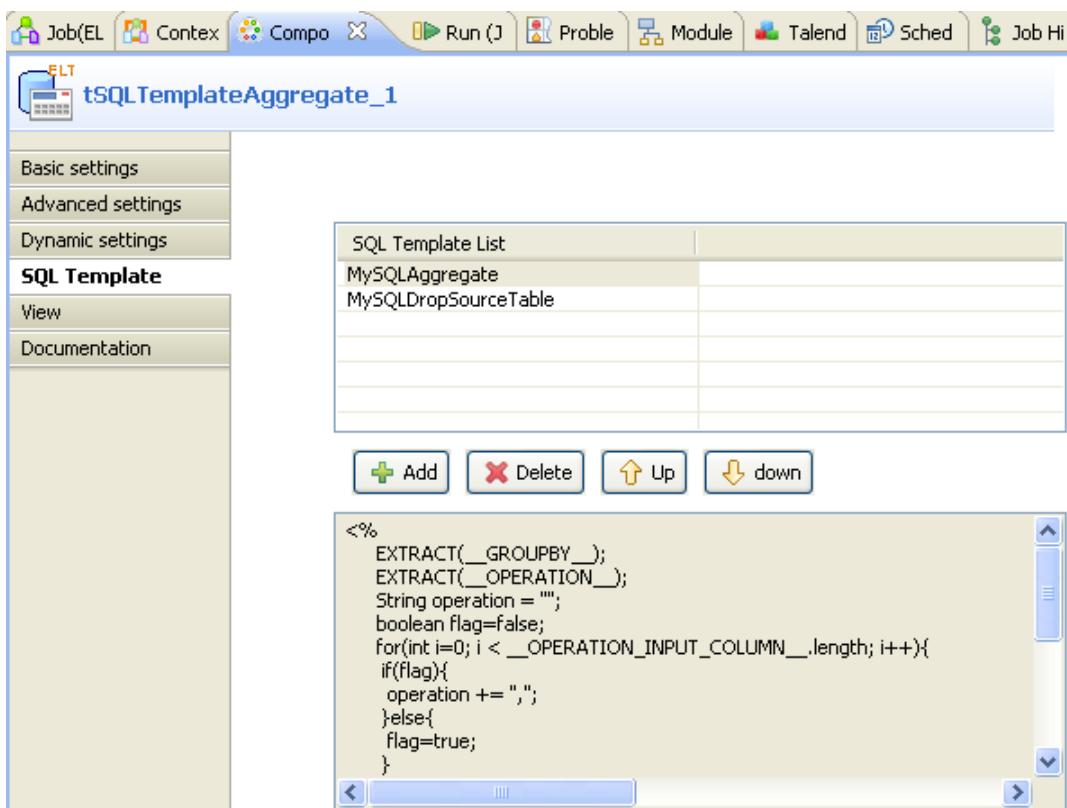
- Dans l'espace de modélisation graphique, sélectionnez le **tSQLTemplateAggregate** et cliquez sur l'onglet **Component** pour définir ses propriétés (**Basic settings**).
- Dans la liste **Database type**, sélectionnez la base de données correspondante.
- Dans la liste **Component list**, sélectionnez le composant de connexion adéquat, si plus d'une connexion est ouverte dans le Job.
- Saisissez le nom de la base de données, de la table source et de la table cible 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.

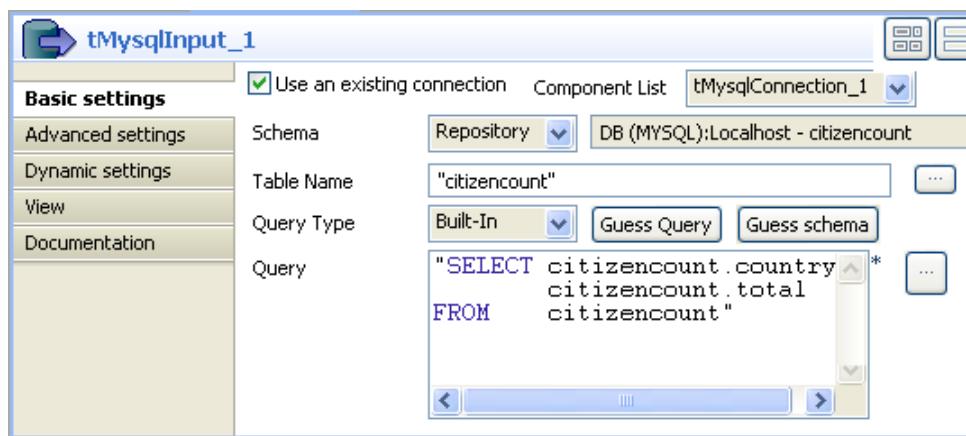


- Dans le tableau **Operations**, cliquez sur le bouton [+] afin d'ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** pour sélectionner la colonne de sortie qui contiendra les données comptées.
- Cliquez sur la ligne **Function**, puis sélectionnez l'opération à effectuer.
- 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 **tSQLTemplateCommit** 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 EITYudong 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 EITYudong 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.

Annexe A. Interface de *Talend Open Studio for Data Integration*

Cette annexe décrit l’interface de *Talend Open Studio for Data Integration*.

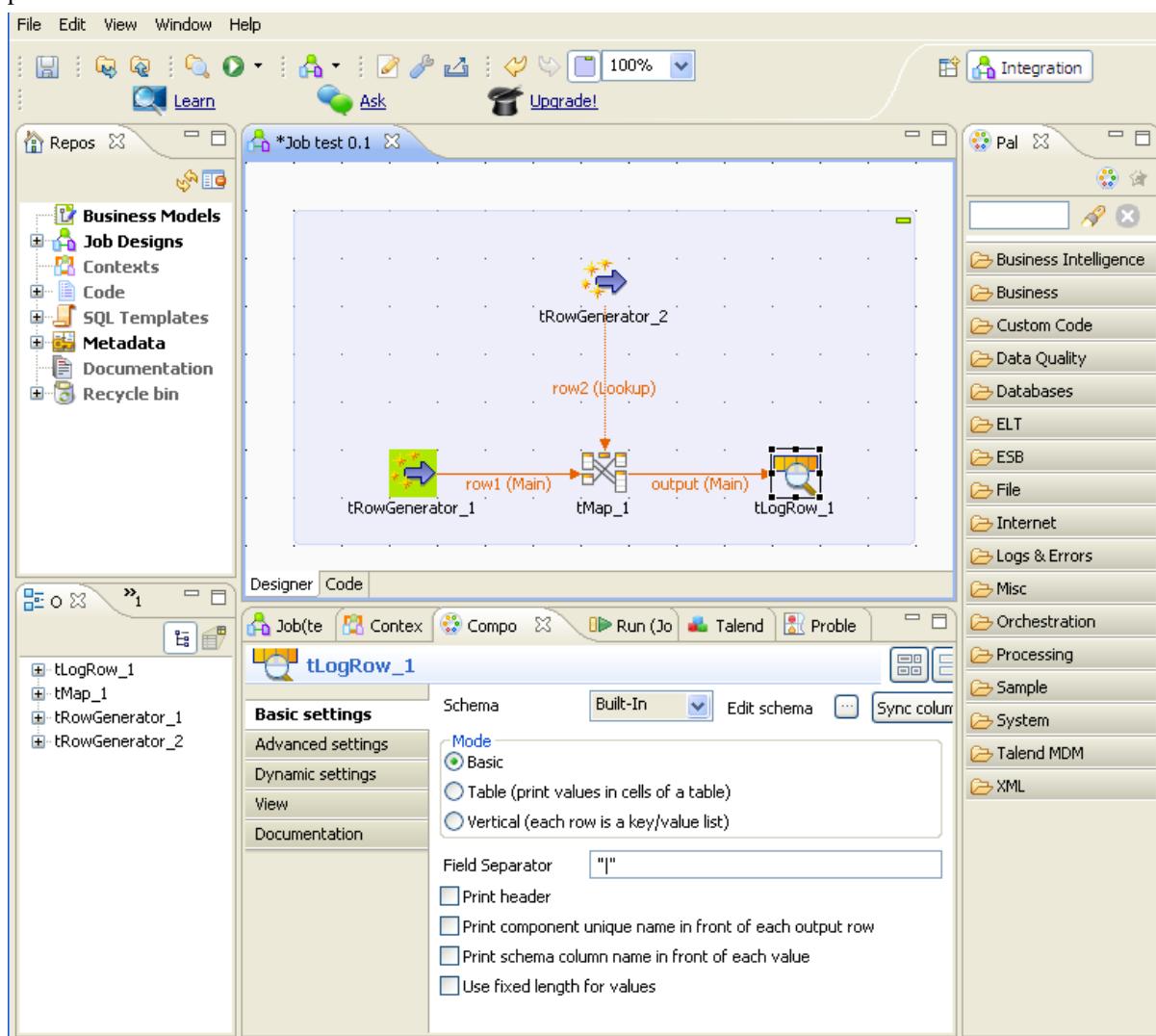
A.1. Fenêtre principale

La fenêtre principale de *Talend Open Studio for Data Integration* 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 for Data Integration* 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 for Data Integration*, 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 for Data Integration*, 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 for Data Integration*

Les menus de *Talend Open Studio for Data Integration* 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 for Data Integration*.

Le tableau ci-dessous décrit les menus et sous-menus disponible dans la barre de menu de *Talend Open Studio for Data Integration*.



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 la Section 3.6, « Enregistrement d'un Business Model »
	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 la Section 2.6, « Personnalisation des paramètres du projet » .
	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.
Edit	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.

Menu	Sous-menu	Description
	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).
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 paramétrer les préférences de <i>Talend Open Studio for Data Integration</i> . Pour plus d'informations concernant les préférences, consultez la Section 2.5, « Configuration des préférences de Talend Open Studio for Data Integration » .
Help	Welcome	Ouvre une page d'accueil contenant des liens vers la documentation de <i>Talend Open Studio for Data Integration</i> et vers les sites pratiques de Talend .
	Help Contents	Ouvre l'aide en ligne d'Eclipse.
	About Talend Open Studio for Data Integration	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 le(s) plug-in(s) -des informations détaillées concernant les fonctionnalités de <i>Talend Open Studio for Data Integration</i> .
	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 installées ou de chercher de nouvelles fonctionnalités à installer. Manage Configuration... : Ouvre une boîte de dialogue [Product Configuration] dans laquelle gérer la configuration de <i>Talend Open Studio for Data Integration</i> .

A.2.2. Barre d'outils de *Talend Open Studio for Data Integration*

La barre d'outils vous permet un accès rapide aux fonctions les plus courantes de *Talend Open Studio for Data Integration*.



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.

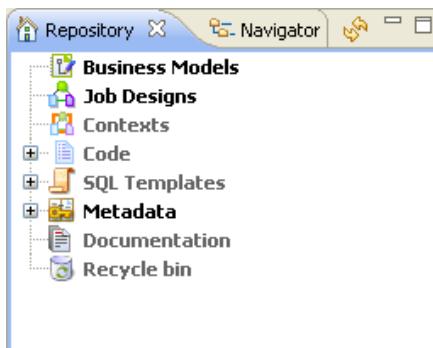
Nom	Icône	Description
Save		Enregistre le Job Design courant.
Save as		Enregistre en tant que nouveau Job.
Export items		Exporte les éléments du Repository dans un fichier archive, pour les déployer en dehors de <i>Talend Open Studio for Data Integration</i> . Si vous souhaitez importer les éléments exportés dans une version plus récente de <i>Talend Open Studio for Data Integration</i> ou d'un autre ordinateur, assurez-vous d'avoir inclus 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 courante de <i>Talend Open Studio for Data Integration</i> . Pour plus d'informations, consultez la Section 5.2.1, « Import d'éléments » .
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 la Section 4.2.7, « Exécuter un Job » .
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 la Section 2.6, « Personnalisation des paramètres du projet » .
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 la Section 2.4.6, « Exporter un projet » .

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 bouton vous permet de mettre à jour l'arborescence et d'y inclure les dernières modifications.

Sauvegardez toutes vos données dans les divers dossiers projets (Business Models et Job Designs) et de métadonnées (Routines, DB/File connections, toute documentation significative, etc.).

Le tableau ci-dessous décrit les nœuds de la vue **Repository**.

Nœud	Description
Business Models	Toutes les représentations graphiques des processus métier d'un projet sont regroupées sous le nœud Business Models . Double-cliquez sur le nom pour l'ouvrir dans l'espace de modélisation. Pour plus d'informations, consultez le Chapitre 3, Conception d'un Business Model .
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 le Chapitre 4, Conception d'un Job d'intégration de données .
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 Section 4.4.2, « Centraliser les Contextes et les Variables » .
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 le Chapitre 7, Gestion des métadonnées .
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 Section 4.4.3, « Utiliser les SQL Templates » .
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 le Chapitre 7, Gestion des métadonnées .
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 Section 5.6.1, « Générer la documentation HTML » .
Recycle bin	Recycle bin , la Corbeille rassemble tous les éléments supprimés à partir de tout dossier du Repository .

Nœud	Description
	<p> Les éléments supprimés sont toujours présents dans vos fichiers système, dans la Corbeille, jusqu'à ce que vous cliquez-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

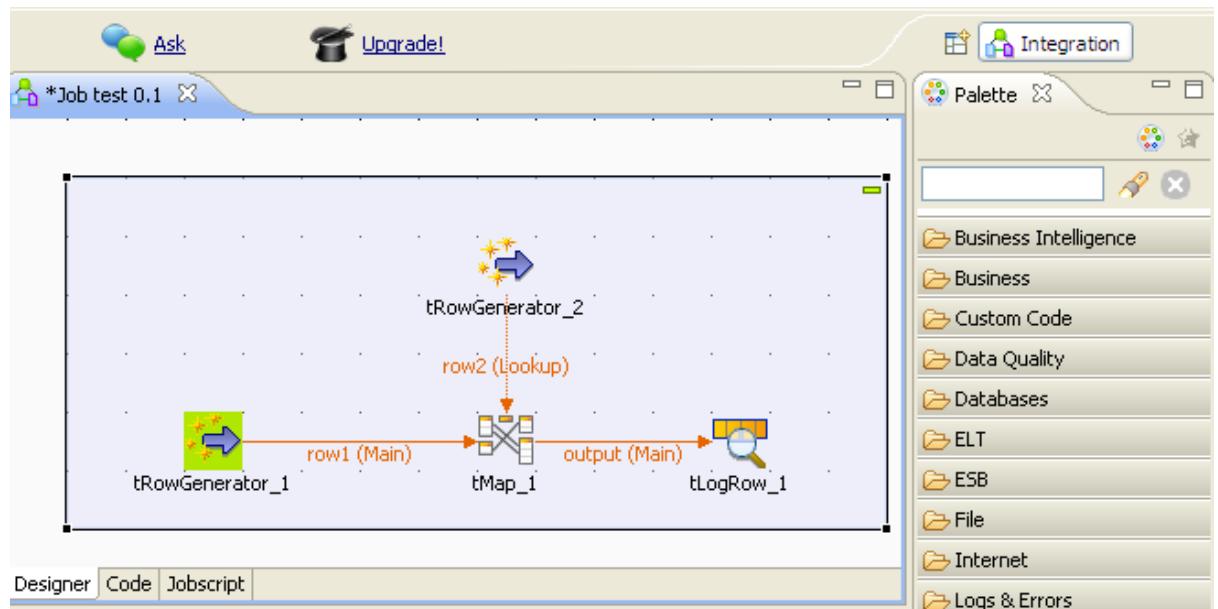
L'espace de modélisation graphique de *Talend Open Studio for Data Integration* vous permet de concevoir graphiquement les Business Models (processus métier) mais également les Job Designs (processus techniques).

Pour plus d'informations, consultez la [Section 3.2, « Crédit ou ouverture d'un Business Model »](#) et la [Section 4.2.1, « Créer un Job »](#).

Pour les Business Models et les Job Designs : les Job Designs et Business Models ouverts sont organisés par un système d'onglets au-dessus de cet espace de modélisation.

Pour les Job Designs uniquement : Sous cet espace de modélisation, plusieurs onglets sont également disponibles :

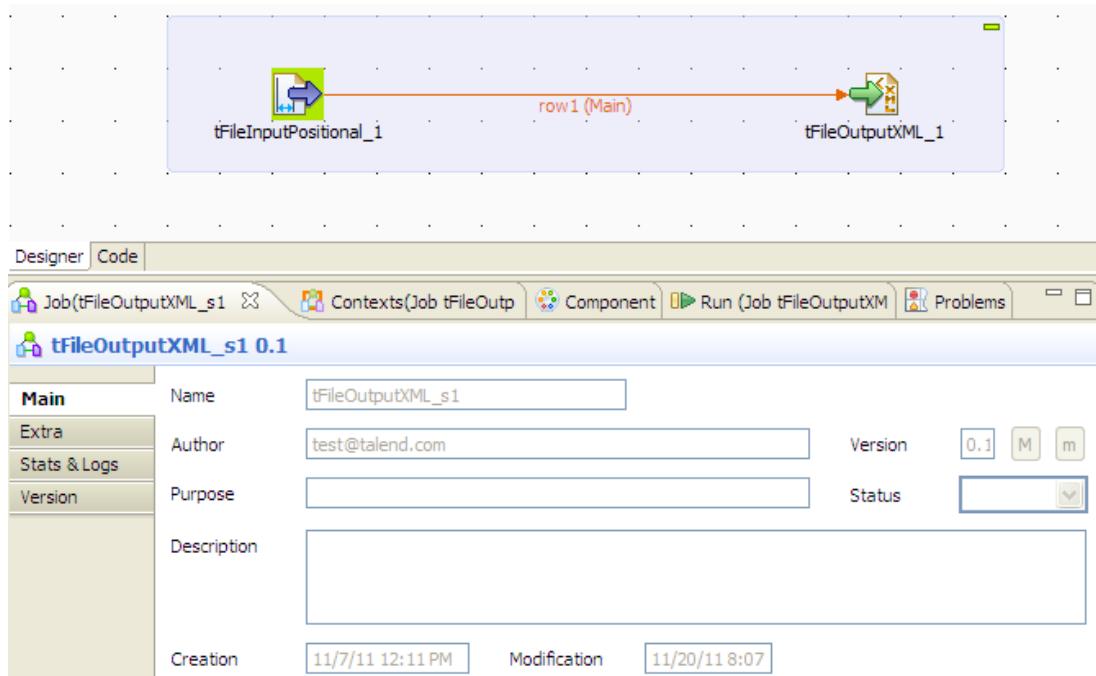
- l'onglet **Designer** : c'est l'onglet ouvert par défaut lors de la création d'un Job. Il affiche le Job en mode graphique ;
 - l'onglet **Code** : il permet de visualiser le code et met en évidence les éventuelles erreurs de langage ;
-  Les avertissements sont indiqués en jaune tandis que les erreurs sont indiquées en rouge.



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. 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 nœud 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 Section 4.2.6, « Définir les propriétés d'un composant » .
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 Section 4.2.7, « Exécuter un Job » .
Error Log	Cette vue est généralement utilisée 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 Java est en cours d'exécution.

Onglet	Description
	L'onglet Error Log est masqué par défaut. Comme pour tout autre onglet, cliquez sur Window > Show views , puis développez le nœud 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 Section 4.5.4, « Installer des modules externes » .
Scheduler	Cette vue vous permet de planifier une tâche qui exécutera périodiquement le Job sélectionné via le programme crontab. Pour plus d'informations, consultez Section 4.5.5, « Lancer un Job périodiquement » .
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 Section 4.6.7.2, « Fonctions Extra »
	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 le Guide de Référence des 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 Section 4.6.7.1, « Automatiser l'utilisation de statistiques & logs »
	Onglet Version Cet onglet regroupe les différentes versions du Job ouvert, ainsi que leurs date et heure de création et de modification.
Talend change	Ex- Cette vue vous donne accès dans Talend Exchange à une liste de composants compatibles avec votre version de <i>Talend Open Studio for Data Integration</i> 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.

Onglet	Description
	Pour plus d'informations, consultez Section 4.5.3, « Télécharger/charger des composants de la communauté ».
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 Section 4.6.3, « Gérer les icônes d'avertissement/d'erreur sur les composants ».
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 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.6. 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 la [Section 4.6.5, « Afficher les onglets Code ou Outline de votre Job ».](#)

A.7. Raccourcis clavier

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

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

Raccourci	Pour	Contexte
Ctrl + Shift + F3	Synchroniser les templates javajet.	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	Arrêter le Job courant.	Dans la vue Run Job
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

Annexe B. De la théorie à la pratique, exemple de Job

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

B.1. Exemple de Job comprenant un tMap

B.1.1. Présentation du scénario

Pour illustrer le fonctionnement de *Talend Open Studio for Data Integration*, 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).

B.1.1.1. 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 for Data Integration* comprend les colonnes suivantes :

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

B.1.1.2. 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.1.1.3. 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
Aliso Viejo	Orange
Anaheim	Orange
Arcadia	Los Angeles

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

B.1.2. 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.1.2.1. Etape 1 : Création du Job, définition des données d'entrée, lecture du fichier

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

Dans le champ **User Email**, saisissez votre e-mail 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 for Data Integration*.

Cliquez ensuite sur **OK** pour lancer le Studio. Puis cliquez sur le lien **Start using Talend Open Studio for Data Integration 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 for Data Integration*.

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

- Le **Business Modeler** : Pour plus d'informations, consultez la [Section 3.3, « Modélisation d'un Business Model »](#).
- Le **Job Designer** : Pour plus d'informations, consultez la [Section 4.2.1, « Créer un Job »](#).
- Le **Metadata Manager** : Pour plus d'informations, consultez la [Section 4.4.1, « Centraliser les métadonnées »](#).

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 nœud **File Delimited**, sélectionnez l'option **Create file delimited**. L'assistant spécifique aux fichiers délimités s'ouvre :

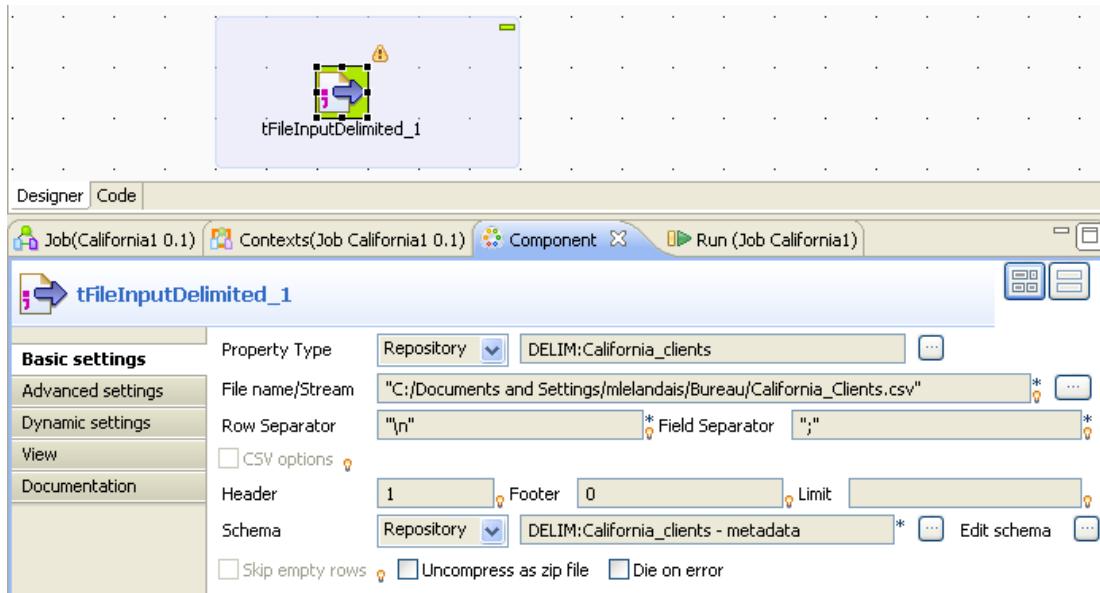
- A l'étape 1, seul le champ **Name** est obligatoire : saisissez le nom **California_clients** et passez à l'étape suivante.
- A l'étape 2, sélectionnez le fichier d'entrée (*California_Clients.csv*) via le bouton **Browse....** Un extrait du fichier apparaît immédiatement dans la zone **File viewer** 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 for Data Integration*) 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.



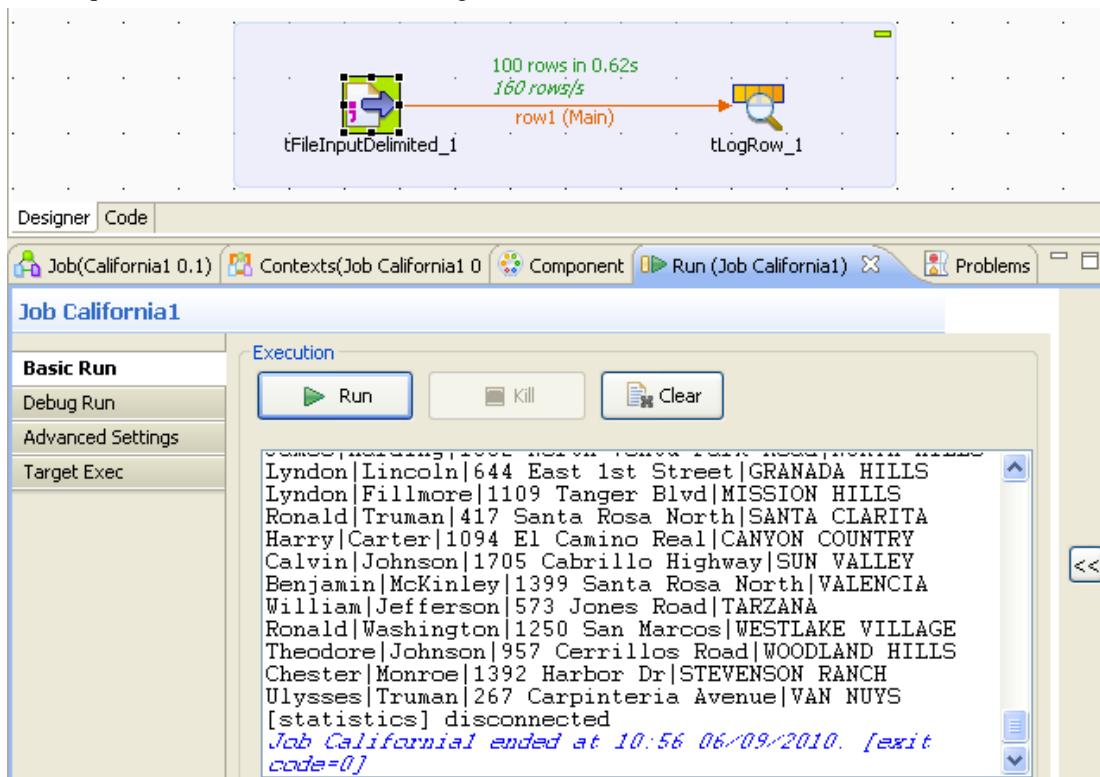
A cette étape, terminez votre flux en envoyant tout simplement les données lues dans le fichier d'entrée vers une sortie standard (StdOut).

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.1.2.2. 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.

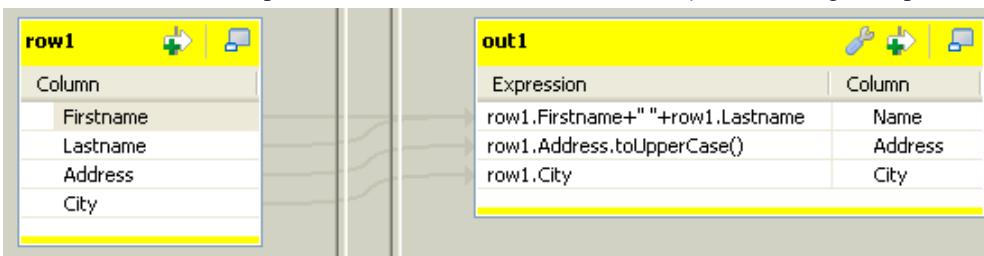
Supprimez la connexion reliant vos deux composants via un clic-droit sur cette connexion et en sélectionnant l'option **Delete**. Puis placez le **tMap** entre les deux autres composants et reliez-le au composant d'entrée comme vous l'avez fait précédemment.

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 rétro-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.

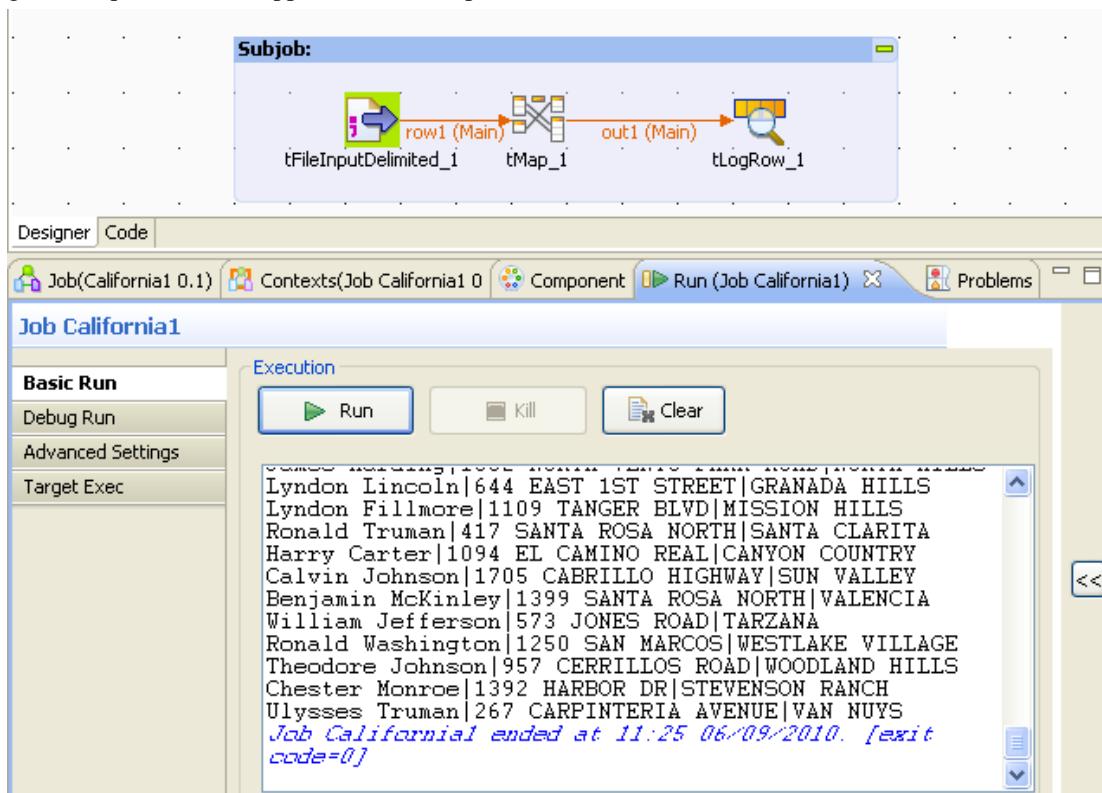
Puis supprimez la colonne *LastName* de la table *out1*, et augmentez la longueur des colonnes restantes. Pour cela, cliquez sur l'onglet **Schema Editor** situé en bas de l'éditeur du **[Map Editor]** et procédez comme suit :

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.

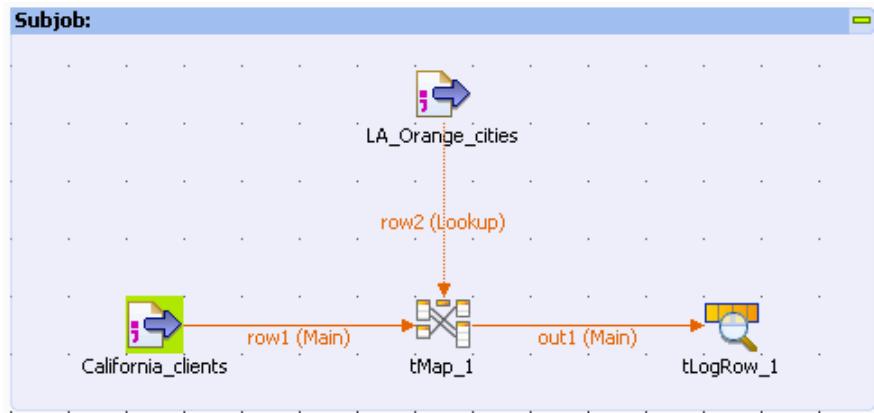
B.1.2.3. Etape 3 : Définition du fichier de référence, mapping des données de référence, sélection du mode Inner Join

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.

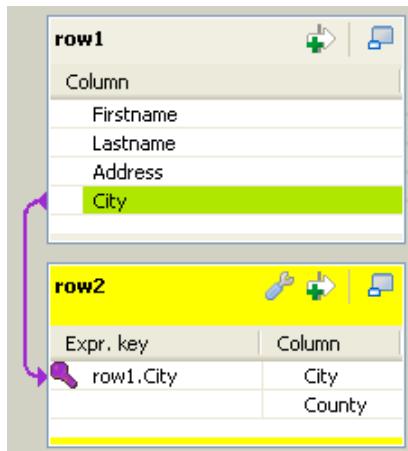
Reliez ce composant au **tMap**.



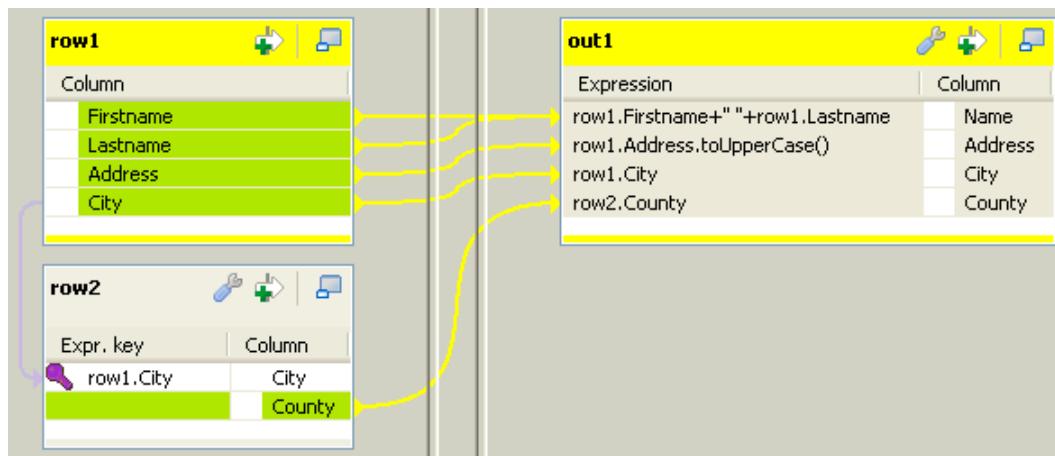
Double-cliquez de nouveau sur le composant **tMap** pour ouvrir son interface. Notez que la table de référence (*row2*) correspondant au fichier *LosAngelesandOrangeCounties.txt*, apparaît à gauche de la fenêtre dans la zone **Input** d'entrée, juste en dessus de votre flux d'entrée principal (*row1*).

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.

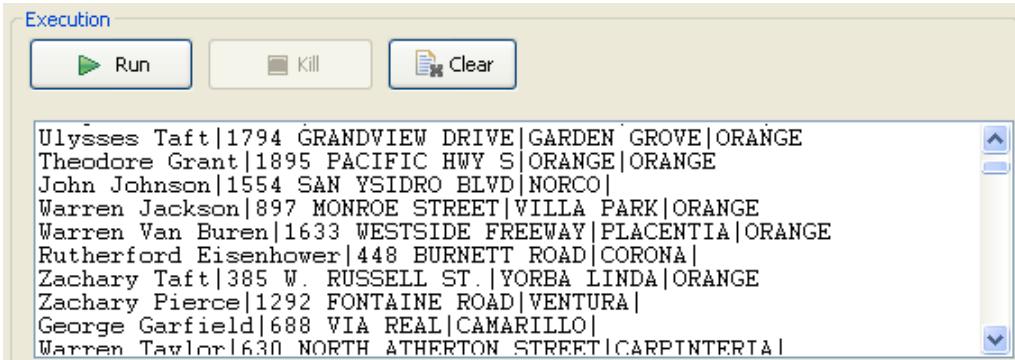


Maintenant, vous pouvez utiliser la colonne *County* de la table de référence dans la table de sortie (*out1*).



Enfin, cliquez sur le bouton **OK** pour valider les modifications et exécutez ce nouveau job.

La sortie suivante s'affichera dans la console :



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.1.2.4. Etape 4 : Sortie vers une table MySQL

Votre Job fonctionne à merveille. Pour le finaliser, redirigez le flux de sortie vers une table MySQL.

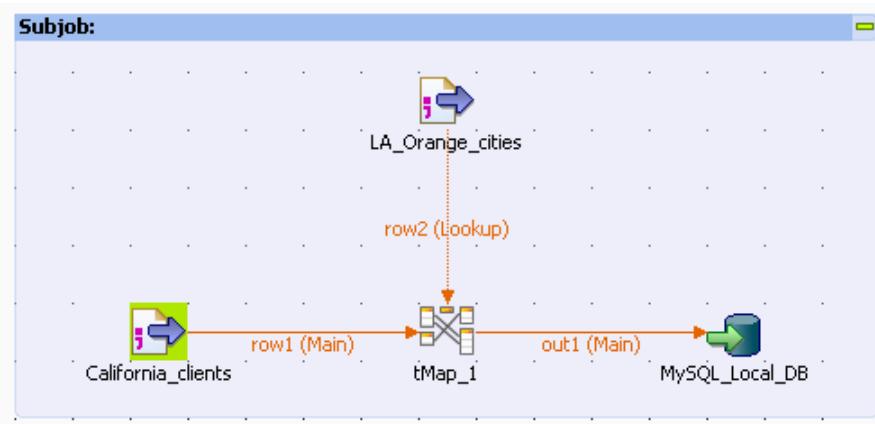
Pour cela, créez tout d'abord une métadonnée décrivant la connexion à la base de données MySQL. Double-cliquez sur **DemoMySQL** dans le répertoire **Metadata > MySQL** du Référentiel (à condition que vous ayez bien importé le projet **Demo**) pour lancer l'assistant Metadata.

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 450 (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**.

B.2. Utilisation de la fonctionnalité Output Stream

B.2.1. Présentation du scénario

Le scénario suivant a pour objectif de montrer comment utiliser la fonctionnalité de flux de sortie dans un certain nombre de composants, afin d'améliorer considérablement les performances en sortie.

Dans ce scénario, un fichier .csv prédéfini contenant des informations client est chargé dans une table d'une base de données. Les données chargées sont sélectionnées à l'aide d'un composant **tMap** et écrites dans un fichier de sortie local, ainsi que dans la console, via la fonctionnalité **Output stream**.

B.2.1.1. Données d'entrée

Le fichier d'entrée, dont les données seront chargées dans la table de la base de données, contient des informations clients variées.

La structure du fichier appelée **Schema** dans *Talend Open Studio for Data Integration* comprend les colonnes suivantes :

- *id* (Type : Integer)
- *CustomerName* (Type : String)
- *CustomerAge* (Type : Integer)
- *CustomerAddress* (Type : String)
- *CustomerCity* (Type : String)
- *RegisterTime* (Type : Date)

B.2.1.2. Données de sortie

Le composant **tMap** est utilisé pour sélectionner les colonnes *id*, *CustomerName* et *CustomerAge* dans les données d'entrée. Les données sélectionnées sont écrites en sortie via la fonctionnalité de flux de sortie.

Les données attendues en sortie doivent avoir la structure suivante :

- *id* (Type : Integer)
- *CustomerName* (Type : String)
- *CustomerAge* (Type : Integer)

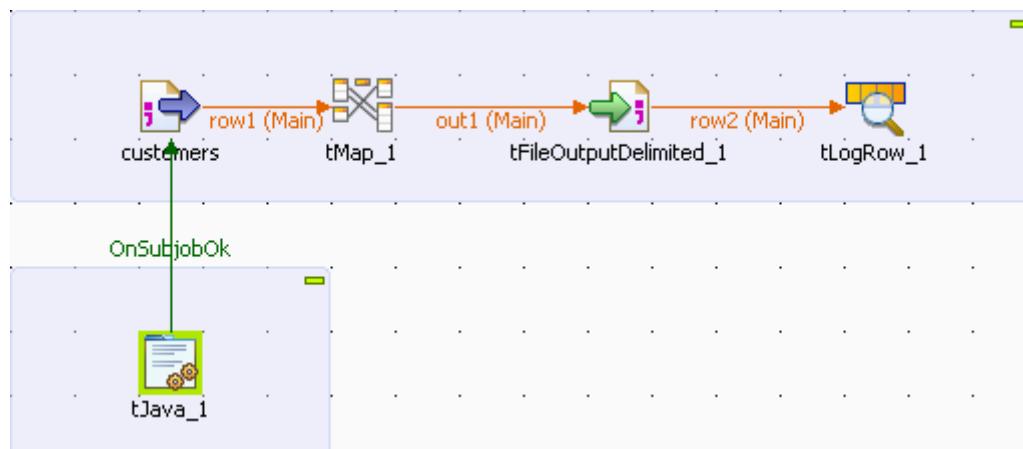
Ces trois colonnes proviennent des colonnes des données d'entrée.

B.2.2. Crédation du Job

Pour créer ce Job, vous devez effectuer les quatre étapes suivantes :

1. Crédation du Job, configuration du schéma d'entrée et lecture du fichier d'entrée selon le schéma défini.
2. Définition de la commande activant la fonctionnalité de flux de sortie.
3. Mapping des données via le composant **tMap**.
4. Ecriture en sortie du flux de données sélectionné.

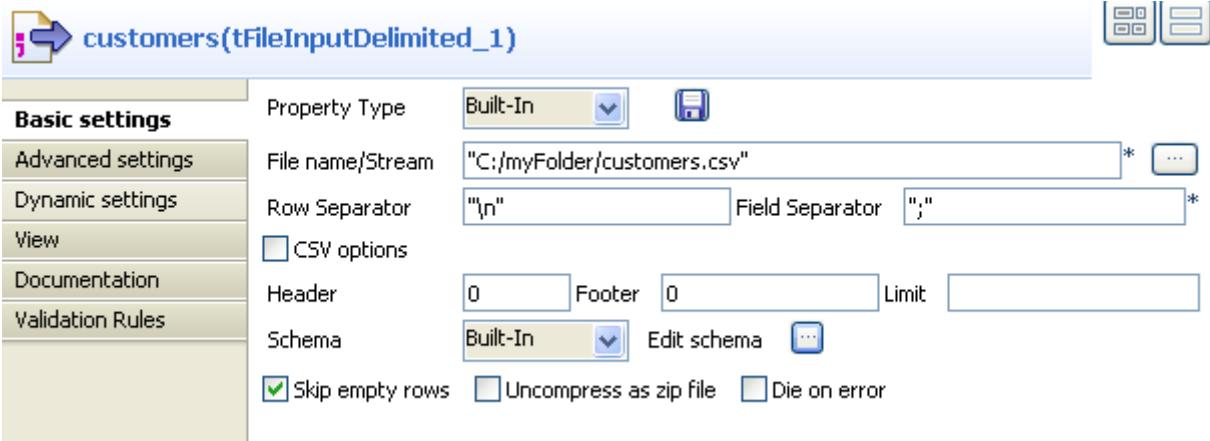
Vous pouvez voir le Job terminé dans la capture d'écran ci-dessous. Pour consulter les instructions détaillées relatives à la création du Job, lisez les sections suivantes.



B.2.2.1. Étape 1 : Lire les données d'entrée d'un fichier local

Utilisez le composant **tFileInputDelimited** pour lire le fichier *customers.csv* contenant les données d'entrée. Ce composant se trouve dans la famille **File/Input** de la **Palette**. Cliquez sur ce puis déposez-le dans l'espace de modélisation graphique.

1. Double-cliquez sur le composant **tFileInputDelimited** afin d'ouvrir sa vue **Basic settings** et définir ses propriétés de base.



2. Cliquez sur le bouton [...] à côté du champ **File name/Stream** et parcourez votre système jusqu'à votre fichier d'entrée. Vous pouvez également saisir manuellement le chemin d'accès à ce fichier.
3. Cliquez sur **Edit schema** pour ouvrir une boîte de dialogue dans laquelle configurer la structure du fichier d'entrée.
4. Cliquez six fois sur le bouton [+] pour ajouter six colonnes, puis, dans la colonne **Type**, sélectionnez **Integer** pour les colonnes *id* et *CustomerAge*, **String** pour les colonnes *CustomerName*, *CustomerAddress* et *CustomerCity*. Sélectionnez **Date** pour la colonne *RegisterTime*.

Column	Key	Type	N.	Date Patt...	Length	Pre...	D...	Co...
<i>id</i>		Integer	<input checked="" type="checkbox"/>					
<i>CustomerName</i>		String	<input checked="" type="checkbox"/>					
<i>CustomerAge</i>		Integer	<input checked="" type="checkbox"/>					
<i>CustomerAddress</i>		String	<input checked="" type="checkbox"/>					
<i>CustomerCity</i>		String	<input checked="" type="checkbox"/>					
<i>RegisterTime</i>		Date	<input checked="" type="checkbox"/>	"dd-MM-yyyy"				

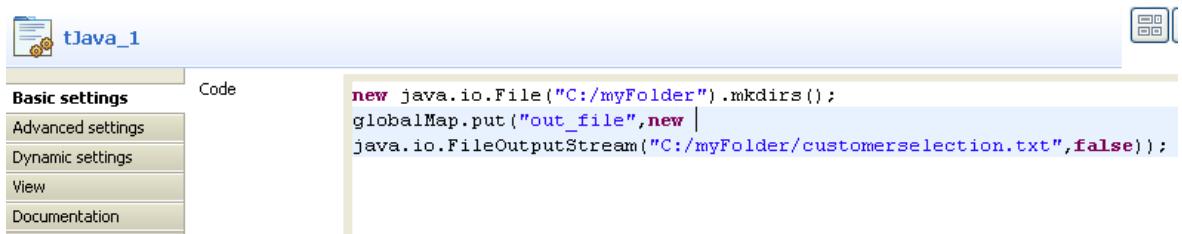
5. Cliquez sur **OK** pour fermer la boîte de dialogue.

B.2.2.2. Étape 2 : Configurer la commande pour activer la fonctionnalité Output Stream

Utilisez le **tJava** pour définir la commande de création d'un fichier de sortie et un répertoire contenant le fichier de sortie.

Pour ce faire, déposez un composant **tJava** dans l'espace de modélisation graphique.

1. Double-cliquez sur le **tJava** pour ouvrir sa vue **Basic settings** et définir ses propriétés.



2. Dans le champ **Code**, saisissez la commande suivante :

```

new java.io.File("C:/myFolder").mkdirs();

globalMap.put("out_file",new
java.io.FileOutputStream("C:/myFolder/customerselection.txt",false));

```

 La commande saisie ci-dessus crée un nouveau répertoire *C:/myFolder* pour sauvegarder le fichier de sortie *customerselection.txt*. Vous pouvez personnaliser la commande selon vos besoins.

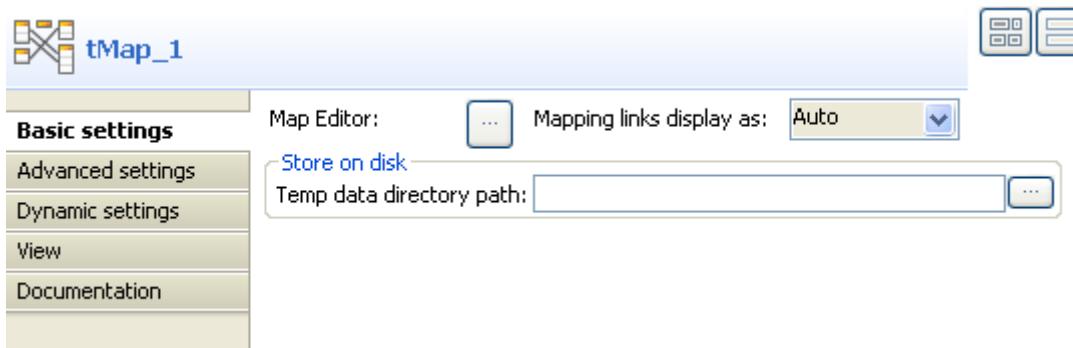
3. Reliez le **tJava** au **tFileInputDelimited** à l'aide d'un lien **Trigger > On Subjob Ok**. Cela déclenche le **tJava** lorsque le sous-job commençant par le **tFileInputDelimited** est correctement exécuté.



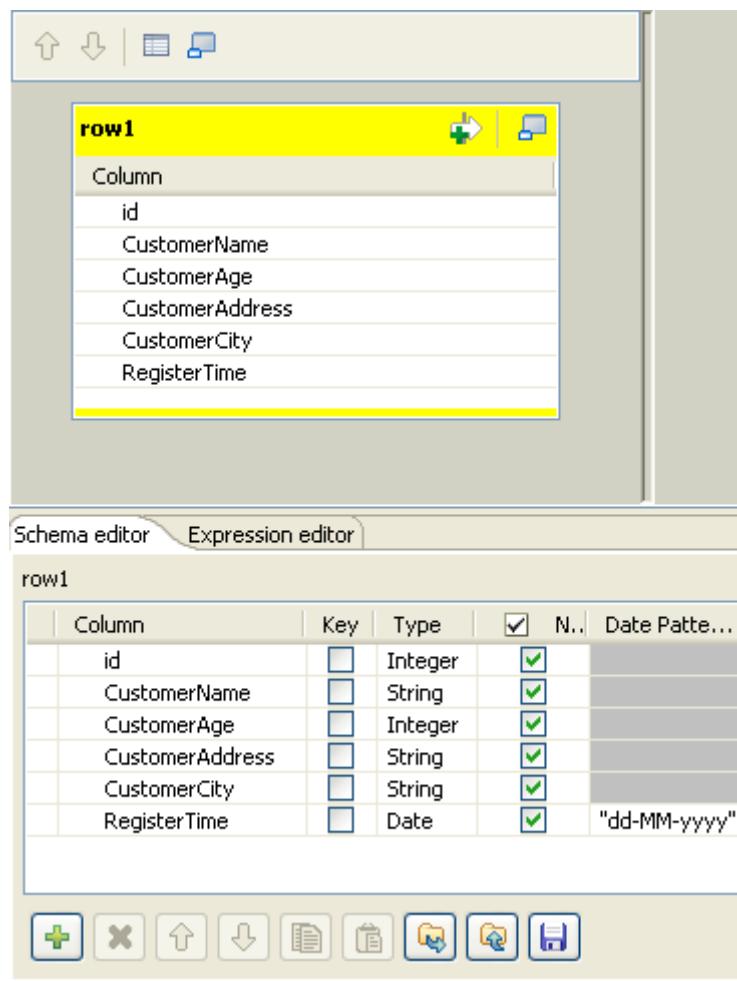
B.2.2.3. Étape 3 : Mapper les données à l'aide du composant **tMap**

Déposez un composant **tMap** dans l'espace de modélisation graphique.

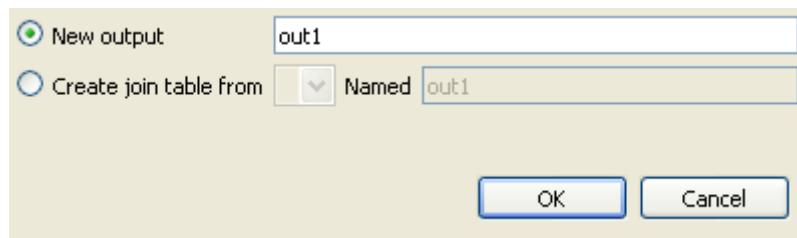
1. Double-cliquez sur le **tMap** afin d'afficher sa vue **Basic settings** et configurer ses propriétés de base.



2. Cliquez sur le bouton [...] à côté du **Map Editor** pour ouvrir une boîte de dialogue dans laquelle configurer le mapping.
3. Cliquez sur le bouton [+] de gauche pour ajouter six colonnes au schéma d'entrée, ces colonnes devant être *id*, *CustomerName*, *CustomerAge*, *CustomerAddress*, *CustomerCity*, *RegisterTime*.

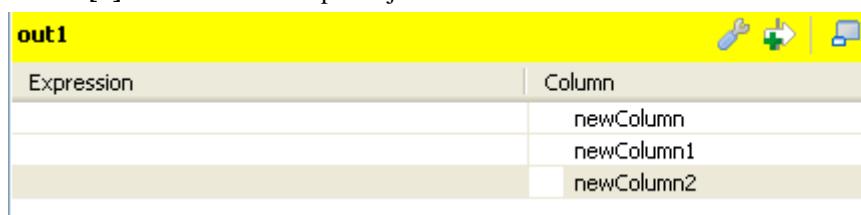


4. Cliquez sur le bouton [+] à droite pour ajouter un schéma de sortie.

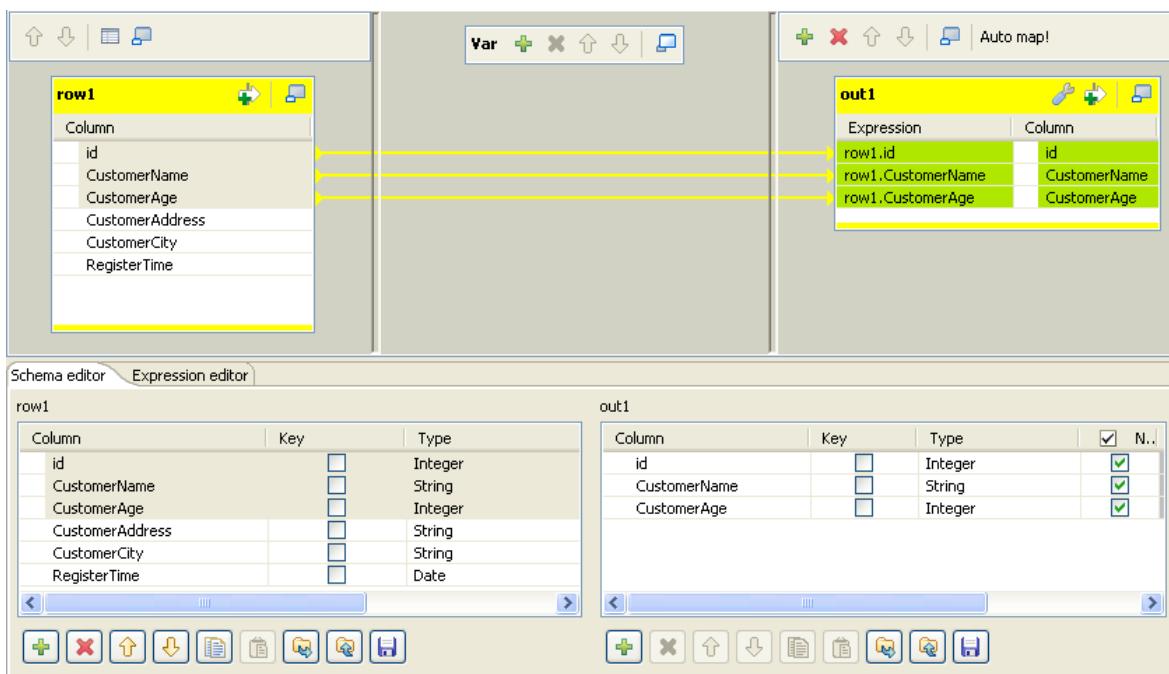


5. Sélectionnez **New output** et cliquez sur **OK** pour sauvegarder votre schéma de sortie. Le schéma de sortie est vide.

6. Cliquez sur le bouton [+] sous la table *out1* pour ajouter trois colonnes aux données de sortie.



7. Déposez les colonnes *id*, *CustomerName* et *CustomerAge* de la gauche à la droite, dans leurs lignes respectives.



8. Cliquez sur **OK** pour sauvegarder les paramètres.

B.2.2.4. Étape 4 : Ecrire en sortie le flux de données sélectionné

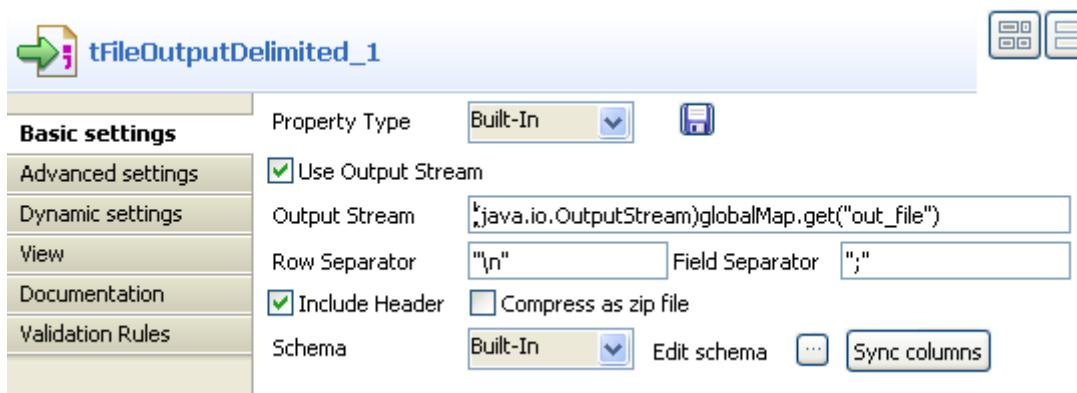
Déposez un composant **tFileOutputDelimited** dans l'espace de modélisation graphique.

1. Double-cliquez sur le **tFileOutputDelimited** afin d'ouvrir sa vue **Basic settings** et configurer ses propriétés de base.
2. Cochez la case **Use Output Stream** pour activer le champ **Output Stream** et saisissez dans le champ **Output Stream** la commande suivante :

```
( java.io.OutputStream)globalMap.get( "out_file" )
```



Vous pouvez personnaliser la commande dans le champ **Output Stream** en appuyant sur les touches **Ctrl+Espace** pour utiliser l'autocomplétion et sélectionner des commandes built-in. Vous pouvez également saisir manuellement votre commande. Dans ce scénario, la commande utilisée dans le champ **Output Stream** appelle la classe `java.io.OutputStream` pour écrire le flux de données filtré dans un fichier local, spécifié dans la zone **Code** du **tJava**.

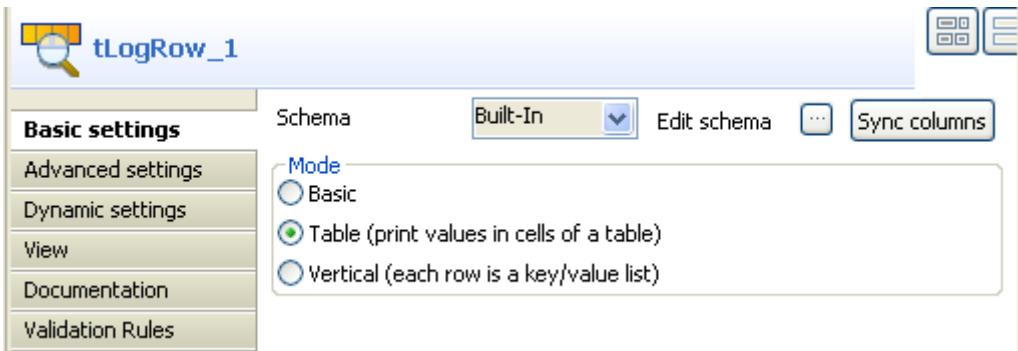


3. Reliez le **tFileInputDelimited** au **tMap** à l'aide d'un lien **Row > Main** puis reliez le **tMap** au **tFileOutputDelimited** à l'aide du lien **Row > out1**, défini dans le **Map Editor** du **tMap**.

4. Cliquez sur le bouton **Sync columns** pour récupérer le schéma du composant précédent.

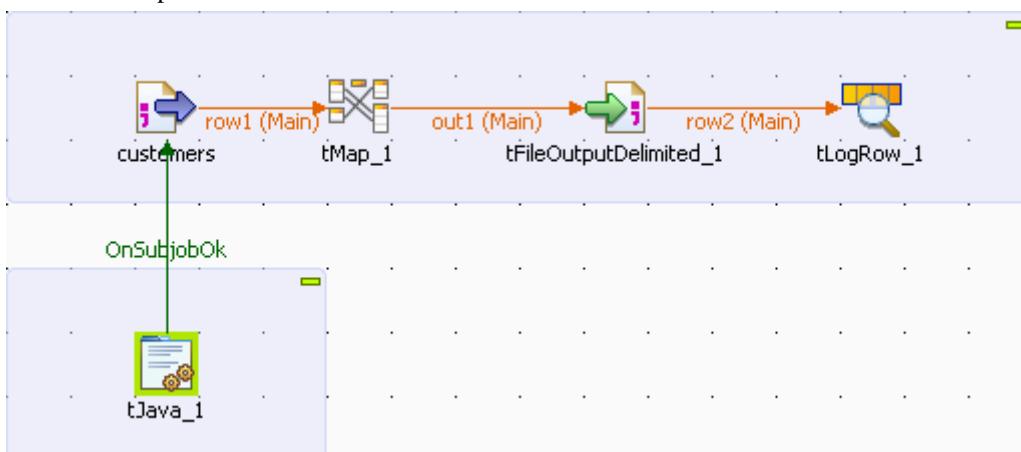
Pour écrire dans la console les données sélectionnées, ajoutez un composant **tLogRow** (de la famille **Logs & Errors**) dans le Job.

1. Déposez le **tLogRow** dans l'espace de modélisation graphique.
2. Double-cliquez sur ce composant afin d'ouvrir sa vue **Basic settings**.
3. Sélectionnez l'option **Table** dans la zone **Mode**.



4. Reliez le **tFileOutputDelimited** au **tLogRow** à l'aide d'un lien **Row > Main**.
5. Cliquez sur **Sync columns** pour récupérer le schéma défini dans le composant précédent.

Ce Job est maintenant prêt à être exécuté.



Appuyez sur les touches **Ctrl+S** afin de sauvegarder votre Job et appuyez sur **F6** pour l'exécuter.

Le contenu de données sélectionnées s'affiche dans la console.

```

Starting job OutputStream at 17:31 19/10/2011.

[statistics] connecting to socket on port 4059
[statistics] connected
+-----+
| tLogRow_1 |
+-----+
| id | CustomerName | CustomerAge |
+-----+
| 10001 | Warren | 67 |
| 10002 | Woodrow | 68 |
| 10003 | Grover | 77 |
| 10004 | Abraham | 74 |
| 10005 | Chester | 78 |
| 10006 | Calvin | 63 |
| 10007 | Zachary | 53 |
| 10008 | Chester | 36 |
| 10009 | Chester | 60 |
| 10010 | Woodrow | 57 |
+-----+
[statistics] disconnected
Job OutputStream ended at 17:31 19/10/2011. [exit code=0]

```

Les données sélectionnées sont également écrites dans le fichier local *customerselection.txt*.

customerselection.txt		
1	id;CustomerName;CustomerAge	
2	10001;Warren;67	
3	10002;Woodrow;68	
4	10003;Grover;77	
5	10004;Abraham;74	
6	10005;Chester;78	
7	10006;Calvin;63	
8	10007;Zachary;53	
9	10008;Chester;36	
10	10009;Chester;60	
11	10010;Woodrow;57	

Pour un exemple de Job utilisant cette fonctionnalité, consultez le composant **tFileOutputDelimited** du **Guide de référence** des *Composants Talend Open Studio*.

Pour consulter les principes de la fonctionnalité **Use Output Stream**, consultez la [Section 4.5.7, « Utiliser la fonctionnalité Use Output Stream »](#).

Annexe C. 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 for Data Integration* 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 `<%...%>` 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 **Guide de référence des Composants**.

Pour plus d'informations concernant la syntaxe `<%...%>`, consultez la [Section C.3, « La syntaxe <%...%> »](#).

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 [Section C.4, « La syntaxe <%=...%> »](#).

Pour plus d'informations concernant la syntaxe <%=...%>, consultez la section précédente.

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 la [Section 4.2.6.1, « Onglet Basic Settings »](#).

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 **tSQLTemplateAggregate**.

Vous pouvez accéder à ces paramètres de transformation, naturellement plus flexibles et complexes, de deux façons différentes.

- L'approche </ . . . />.

</ . . . /> est l'une des syntaxes utilisées par les modèles SQL. Cette approche nécessite souvent du code dur pour chaque paramètre à extraire.

Par exemple, un nouveau paramètre est créé par l'utilisateur et nommé *NEW_PROPERTY*. Si vous souhaitez y accéder en utilisant </*NEW_PROPERTY*/>, le code ci-dessous est requis.

```
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.

Annexe D. 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 8, Gestion des routines](#).

Avant de commencer un processus métier (Job), vous devez vous être familiarisé avec l'interface graphique de *Talend Open Studio for Data Integration*. Pour plus d'informations, consultez la [Annexe A, Interface de Talend Open Studio for Data Integration](#) .

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 <code>true</code> si l'ordre alphabétique est vérifié, et <code>false</code> , dans le cas inverse.	StringHandling.ALPHA("chaîne à vérifier")
IS_ALPHA	vérifie si l'expression ne contient que des caractères alphabétiques. Retourne le booléen <code>true</code> si c'est le cas, et <code>false</code> dans le cas inverse.	StringHandling.IS_ALPHA("chaîne à vérifier")
CHANGE	remplace un élément d'une chaîne de caractères par l'élément de remplacement défini et retourne la nouvelle chaîne.	StringHandling.CHANGE("chaîne à vérifier", "chaîne à remplacer", "chaîne de remplacement")
COUNT	retourne le nombre d'occurrences d'une sous-chaîne dans une chaîne de caractères.	StringHandling.COUNT("chaîne à vérifier", "chaîne à compter")
DOWNCASE	convertit toutes les majuscules d'une expression en minuscules et retourne la nouvelle chaîne.	StringHandling.DOWNCASE("chaîne à convertir")

Routine	Description	Syntaxe
<i>UPCASE</i>	convertit toutes les minuscules d'une expression en majuscules et retourne la nouvelle chaîne.	StringHandling.UPCASE("chaîne à convertir")
<i>DQUOTE</i>	entoure une expression de guillemets doubles.	StringHandling.DQUOTE("chaîne à traiter")
<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.	StringHandling.INDEX("chaîne à vérifier", "sous-chaîne recherchée")
<i>LEFT</i>	retourne une sous-chaîne correspondant aux n premiers caractères d'une chaîne de caractères.	StringHandling.LEFT("chaîne à vérifier", nombre de caractères)
<i>RIGHT</i>	retourne une sous-chaîne correspondant aux n derniers caractères d'une chaîne de caractères.	StringHandling.RIGHT("chaîne à vérifier", nombre de caractères)
<i>LEN</i>	retourne la longueur d'une chaîne de caractères.	StringHandling.LEN("chaîne à vérifier")
<i>SPACE</i>	retourne une chaîne faite du nombre de caractères vides indiqué.	StringHandling.SPACE(nombre d'espaces à créer)
<i>SQUOTE</i>	entoure une expression de guillemets simples.	StringHandling.SQUOTE("chaîne à traiter")
<i>STR</i>	retourne un caractère répété le nombre de fois indiqué.	StringHandling.STR('caractère à générer', nombre de répétition)
<i>TRIM</i>	supprime les espaces et les tabulations en début et fin d'une chaîne de caractères et retourne la nouvelle chaîne.	StringHandling.TRIM("chaîne à traiter")
<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.	StringHandling.BTRIM("chaîne à traiter")
<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.	StringHandling.FTRIM("chaîne à traiter")

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 routinel at 11:45 23/02/2010.  
[statistics] connecting to socket on port 3892  
[statistics] connected  
false  
[statistics] disconnected  
Job routinel 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 routinel at 15:47 24/02/2010.  
[statistics] connecting to socket on port 4027  
[statistics] connected  
0  
6  
11  
-1  
[statistics] disconnected  
Job routinel 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 [Section 8.3, « Personnalisation des routines système »](#).

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(TalendDateGenerator.getFirstName());
System.out.println(TalendDateGenerator.getLastName());
System.out.println(TalendDateGenerator.getUsCity());
System.out.println(TalendDateGenerator.getUsState());
System.out.println(TalendDateGenerator.getUsStateId());
System.out.println(TalendDateGenerator.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>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>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>diffDate</i>	retourne la différence entre deux dates, en nombre de jours, mois ou années selon le paramètre de comparaison spécifié.	TalendDate.diffDate(Date1(), Date2(), "format de données à comparer - ex yyyy")

Routine	Description	Syntaxe
<code>diffDateFloor</code>	retourne la différence entre deux dates, en nombre d'années, mois, jours, heures, minutes, secondes ou millisecondes selon le paramètre de comparaison spécifié.	<code>TalendDate.diffDateFloor(Date1(), Date2(), "format de données à comparer - ex MM")</code>
<code>formatDate</code>	retourne une expression de type date formatée selon le modèle date spécifié.	<code>TalendDate.formatDate("format de date - ex : YYYY-MM-dd HH:mm:ss ", Date() à formater)</code>
<code>formatDateLocale</code>	transforme une date en une chaîne de caractères de type date/heure selon le modèle et la locale spécifiés. Retourne la nouvelle date formatée.	<code>TalendDate.formatDateLocale("format cible", java.util.Date date, "code de la langue ou du pays")</code>
<code>getCurrentDate</code>	retourne la date courante. Aucun paramètre d'entrée n'est attendu.	<code>TalendDate.getCurrentDate()</code>
<code>getDate</code>	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.	<code>TalendDate.getDate("Format de chaîne de caractère - ex : CCYY-MM-DD")</code>
<code>getFirstDayOfMonth</code>	change le jour d'une date en premier jour du mois courant et retourne la nouvelle date.	<code>TalendDate.getFirstDayMonth(Date)</code>
<code>getLastDayOfMonth</code>	change le jour d'une date en dernier jour du mois courant et retourne la nouvelle date.	<code>TalendDate.getLastDayMonth(Date)</code>
<code>getPartOfDay</code>	retourne une partie d'une date dans le format spécifié. Cette chaîne peut contenir des chaînes de caractères fixes, ou des variables liées à la date.	<code>TalendDate.getPartOfDay("Chaîne de caractères indiquant la partie de la date à récupérer", "Chaîne de caractères au format date à parser")</code>
<code>getRandomDate</code>	retourne une date aléatoire, au format ISO.	<code>TalendDate.getRandomDate("Chaîne de caractère de type Date, de ", String maxDate)</code>
<code>isDate</code>	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.	<code>TalendDate.isDate(Date() à vérifier, "format de date cible - ex : YYYY-MM-dd HH:mm:ss")</code>
<code>parseDate</code>	transforme une chaîne de caractères en Date. Retourne une date formatée en standard.	<code>TalendDate.parseDate("format date de la chaîne à parser", "Chaîne de caractères au format date à parser")</code>
<code>parseDateLocale</code>	parse une chaîne de caractères correspondant à un modèle spécifié, et en extrait une date. Retourne une date formatée selon la locale spécifiée.	<code>TalendDate.parseDateLocale("format date de la chaîne à parser", "Chaîne de caractères au format date à parser", "code de la langue ou du pays")</code>
<code> setDate</code>	modifie une partie de la date en entrée par un entier spécifié et basé sur le format spécifié.	<code>TalendDate.setDate(Date, entier n, "format de la donnée à changer - ex : yyyy")</code>

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 qu'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 retournée :

```
Starting job routine1 at 11:58 01/03/2010.  
Sat Oct 20 19:00:59 CET 1979  
Job routine1 ended at 11:58 01/03/2010. [exit code=0]
```

D.5.6. Exemple de récupération d'une partie d'une Date

Vous pouvez tester simplement la routine **getPartOfDay**, à l'aide d'un composant **tJava**, pour récupérer une partie d'une date, par exemple :

```
Date D=TalendDate.parseDate("dd-MM-yyyy HH:mm:ss", "13-10-2010 12:23:45");  
  
System.out.println(D.toString());  
System.out.println(TalendDate.getPartOfDay("DAY_OF_MONTH", D));  
System.out.println(TalendDate.getPartOfDay("MONTH", D));  
System.out.println(TalendDate.getPartOfDay("YEAR", D));  
System.out.println(TalendDate.getPartOfDay("DAY_OF_YEAR", D));  
System.out.println(TalendDate.getPartOfDay("DAY_OF_WEEK", D));
```

Dans cet exemple, sont retournés le jour du mois (DAY_OF_MONTH), le mois (MONTH), l'année (YEAR), le numéro de jour de l'année (DAY_OF_YEAR) et le numéro de jour de la semaine (DAY_OF_WEEK). Toutes les données retournées sont de type numérique.

```
Starting job routine at 10:52 17/12/2010.  
[statistics] connecting to socket on port 3565  
[statistics] connected  
Wed Oct 13 12:23:45 CEST 2010  
13  
9  
2010  
286  
4  
[statistics] disconnected  
Job routine ended at 10:52 17/12/2010. [exit code=0]
```



Dans la console de la vue **Run**, la chaîne de caractères qui renvoie aux mois (MONTH) va de 0 à 11 : 0 correspondant à janvier, et 11 correspondant à décembre.

D.5.7. Exemple de formatage de la Date courante

Vous pouvez tester simplement la routine **getDate**, à l'aide d'un composant **tJava**, pour récupérer la date courante et la formater selon un modèle spécifié, par exemple :

```
System.out.println(TalendDate.getDate("CCYY-MM-DD"));
```

La date courante est retournée selon le format spécifié (optionnel) :

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>checkCDATAFor XML</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)
<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]
```