



Connecting the
Data-Driven Enterprise >



Talend Open Studio for Big Data

Guide utilisateur

7.0.1M2

Convient à la version 7.0.1M2. Annule et remplace toute version antérieure de ce guide.

Date de publication : 20 décembre 2017

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/>

Mentions légales

Talend est une marque déposée de Talend, Inc.

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

Licence applicable

Le logiciel décrit dans cette documentation est soumis à la Licence Apache, Version 2.0 (la "Licence"). Vous ne pouvez utiliser ce logiciel que conformément aux dispositions de la Licence. Vous pouvez obtenir une copie de la Licence sur <http://www.apache.org/licenses/LICENSE-2.0.html> (en anglais). Sauf lorsqu'explicitement prévu par la loi en vigueur ou accepté par écrit, le logiciel distribué sous la Licence est distribué "TEL QUEL", SANS GARANTIE OU CONDITION D'AUCUNE SORTE, expresse ou implicite. Consultez la Licence pour connaître la terminologie spécifique régissant les autorisations et les limites prévues par la Licence.

Ce produit comprend les logiciels développés par AOP Alliance (standards Java/J2EE AOP), ASM, Amazon, AntLR, Apache ActiveMQ, Apache Ant, Apache Avro, Apache Axiom, Apache Axis, Apache Axis 2, Apache Batik, Apache CXF, Apache Cassandra, Apache Chemistry, Apache Common Http Client, Apache Common Http Core, Apache Commons, Apache Commons Bcel, Apache Commons JXPath, Apache Commons Lang, Apache Datafu, Apache Derby Database Engine and Embedded JDBC Driver, Apache Geronimo, Apache HCatalog, Apache Hadoop, Apache Hbase, Apache Hive, Apache HttpClient, Apache HttpComponents Client, Apache JAMES, Apache Log4j, Apache Lucene Core, Apache Neethi, Apache Oozie, Apache POI, Apache Parquet, Apache Pig, Apache PiggyBank, Apache ServiceMix, Apache Sqoop, Apache Thrift, Apache Tomcat, Apache Velocity, Apache WSS4J, Apache WebServices Common Utilities, Apache Xml-RPC, Apache Zookeeper, Box Java SDK (V2), CSV Tools, Cloudera HTrace, ConcurrentLinkedHashMap for Java, Couchbase Client, DataNucleus, DataStax Java Driver for Apache Cassandra, Ehcache, Ezmorph, Ganymed SSH-2 for Java, Google APIs Client Library for Java, Google Gson, Groovy, Guava : Google Core Libraries for Java, H2 Embedded Database and JDBC Driver, Hector : client Java haut niveau pour Apache Cassandra, Hibernate BeanValidation API, Hibernate Validator, HighScale Lib, HsqlDB, Ini4j, JCLOUDS, JDO-API, JLine, JSON, JSR 305: Annotations for Software Defect Detection in Java, JUnit, Jackson Java JSON-processor, Java API for RESTful Services, Java Agent for Memory Measurements, Jaxb, Jaxen, JetS3T, Jettison, Jetty, Joda-Time, Json Simple, LZ4 : Extremely Fast Compression algorithm, LightCouch, MetaStuff, Metrics API, Metrics Reporter Config, Microsoft Azure SDK pour Java, Mondrian, MongoDB Java Driver, Netty, Ning Compression codec for LZF encoding, OpenSAML, Paraccel JDBC Driver, Parboiled, PostgreSQL JDBC Driver, Protocol Buffers - Google's data interchange format, Resty : client simple HTTP REST pour Java, Riak Client, Rocoto, SDSU Java Library, SL4J : Simple Logging Facade for Java, SQLite JDBC Driver, Scala Lang, Simple API for CSS, Snappy for Java a fast compressor/decompresser, SpyMemCached, SshJ, StAX API, StAXON - JSON via StAX, Super SCV, The Castor Project, The Legion of the Bouncy Castle, Twitter4J, Uuid, W3C, bibliothèques Windows Azure Storage pour Java, Woden, Woodstox : High-performance XML processor, Xalan-J, Xerces2, XmlBeans, XmlSchema Core, Xmlsec - Apache Santuario, YAML parser et émetteur pour Java, Zip4J, atinject, dropbox-sdk-java : bibliothèque Java pour l'API Dropbox Core, google-guice. Fournis sous leur licence respective.

Table des matières

Préface	vii
1. Informations générales	vii
1.1. Objectif	vii
1.2. Public visé	vii
1.3. Conventions typographiques	vii
2. Remarques et Support	vii
chapitre 1. Intégration de données :	
principes et concepts	1
chapitre 2. Introduction aux solutions	
Big Data de Talend	3
2.1. Hadoop et studio Talend	4
2.2. Architecture fonctionnelle des solutions Big Data de Talend	4
chapitre 3. Travailler avec les projets	7
3.1. Créer un projet	8
3.2. Importer un projet Démo	9
3.3. Importer un projet	10
3.4. Exporter un projet	12
chapitre 4. Conception d'un Business Model	15
4.1. Objectifs	16
4.2. Création d'un Business Model	16
4.3. Modélisation d'un Business Model	18
4.3.1. Formes (Shapes)	19
4.3.2. Connexions	20
4.3.3. Commenter et réarranger un Business Model	22
4.3.4. Propriétés	24
4.4. Affectation d'éléments du Repository à un Business Model	27
chapitre 5. Conception de Jobs	29
5.1. Qu'est-ce qu'un Job ?	30
5.2. Prise en main d'un Job simple	30
5.2.1. Créer un Job	30
5.2.2. Ajouter des composants au Job	32
5.2.3. Connecter les composants	36
5.2.4. Configurer les composants	37
5.2.5. Exécuter le Job	39
5.3. Utilisation des composants	40
5.3.1. Ajouter un composant entre deux composants reliés	40
5.3.2. Définir les propriétés des composants	44
5.3.3. Chercher un Job utilisant un composant particulier	53
5.3.4. Configurer les valeurs par défaut dans le schéma d'un composant dans un Job	54
5.3.5. Utiliser des composants Camel dans une Route	56
5.4. Utilisation des connexions dans un Job	57
5.4.1. Types de connexions	58
5.4.2. Définir les paramètres des connexions	62
5.5. Utiliser les contextes et les variables	64
5.5.1. Définir les variables de contexte pour un Job	65
5.5.2. Centraliser les variables de contexte dans le Repository	70
5.5.3. Appliquer des variables de contexte du référentiel dans un Job	77
5.5.4. Utiliser des variables dans un Job	79
5.5.5. Exécuter un Job dans un contexte défini	80
5.5.6. Utiliser une variable de contexte pour déboguer	81
5.6. Gestion des Jobs : Utilisation avancée	81
5.6.1. Mapper des flux de données	81
5.6.2. Créer une requête à l'aide de SQLBuilder	81
5.6.3. Télécharger/charger des composants de la communauté	86
5.6.4. Utiliser les composants tPrejob et tPostjob	92
5.6.5. Utiliser la fonctionnalité Use Output Stream	93
5.7. Gestion des Jobs : Autres fonctions	93
5.7.1. Utiliser les dossiers	94
5.7.2. Partager une connexion à une base de données	94
5.7.3. Gérer les icônes d'avertissement/d'erreur sur les composants	95
5.7.4. Ajouter une Note au Job	97
5.7.5. Afficher les onglets Code ou Outline de votre Job	98
5.7.6. Gérer l'affichage d'un sous-job	99
5.7.7. Paramétrage des options dans la vue Job	101
chapitre 6. Gestion des Jobs	105
6.1. Activation/Désactivation d'un composant ou d'un sous-job	106
6.1.1. Activer ou désactiver un composant	106
6.1.2. Activer ou désactiver un sous-job	106
6.1.3. Activer ou désactiver tous les sous-jobs liés	107
6.2. Import/export d'éléments et construction de Jobs	107
6.2.1. Import d'éléments	107
6.2.2. Construction de Jobs	111
6.2.3. Export d'éléments	118
6.2.4. Changer les paramètres de contexte dans des Jobs	120
6.3. Gestion des éléments du Repository	121
6.3.1. Gérer les mises à jour des éléments du Repository	121
6.4. Recherche d'un Job dans le référentiel	124
6.5. Gestion des versions d'un Job	125
6.6. Documentation du Job	127
6.6.1. Générer la documentation HTML	127
6.6.2. Mettre la documentation à jour	128
6.7. Exécution des Jobs	128
6.7.1. Exécuter un Job en mode normal	128
6.7.2. Exécuter un Job en mode Java debug	129
6.7.3. Exécuter un Job en mode Traces Debug	130
6.7.4. Configurer les paramètres d'exécution avancés	131
6.7.5. Moniturer l'utilisation des ressources de la JVM durant une exécution de Job	133
6.7.6. Déploiement d'un Job sur un serveur SpagoBI	134
6.8. Utilisation de la parallélisation pour optimiser les performances des Jobs	137
6.8.1. Exécuter plusieurs sous-jobs en parallèle	137
6.8.2. Lancer des itérations parallèles pour lire des données	138

chapitre 7. Gestion des Jobs	141
7.1. Exécution d'un Job via Oozie	142
7.1.1. Configurer les informations de la connexion HDFS	142
7.1.2. Exécuter un Job sur le serveur HDFS	144
7.1.3. Planifier les exécutions d'un Job	145
7.1.4. Monitorer le statut des exécutions de Job	146
chapitre 8. Mapping de flux de données	149
8.1. Interfaces de mapping	150
8.2. Présentation du fonctionnement du tMap	151
8.2.1. Configuration du flux d'entrée dans le Map Editor	152
8.2.2. Mapping de variables	159
8.2.3. Utilisation des expressions	161
8.2.4. Paramètres Output	168
8.2.5. Configuration des schémas dans le Map Editor	173
8.2.6. Activation de la conversion automatique des types de données	177
8.2.7. Résolution des problèmes de mémoire lors de l'utilisation du tMap	179
8.2.8. Gestion des références	181
8.3. Présentation du fonctionnement du tXMLMap	184
8.3.1. Utiliser le type Document pour créer l'arborescence XML	185
8.3.2. Définir le mode de sortie	196
8.3.3. Utiliser le Tree schema editor	201
chapitre 9. Mapping de flux Big Data	203
9.1. Interface du tPigMap	204
9.2. Présentation du fonctionnement du tPigMap	205
9.2.1. Configurer les opérations de jointure	205
9.2.2. Capturer les enregistrements rejetés	206
9.2.3. Editer les expressions	207
9.2.4. Configurer une fonction Pig utilisateur	210
9.2.5. Définir une fonction personnalisée Pig à l'aide du panneau UDF	211
chapitre 10. Gestion des métadonnées dans l'intégration de données	215
10.1. Objectifs	216
10.2. Centraliser des métadonnées de base de données	217
10.2.1. Configurer une connexion à une base de données	217
10.2.2. Récupérer les schémas de table	220
10.3. Centraliser des métadonnées JDBC	225
10.3.1. Configurer une connexion JDBC	225
10.3.2. Récupérer des schémas de table	227
10.4. Centraliser des métadonnées SAS	229
10.4.1. Configurer une connexion SAS	229
10.4.2. Récupérer des schémas de table	231
10.5. Centraliser des métadonnées File Delimited	232
10.6. Configurer un schéma File Positional	238
10.7. Centraliser des métadonnées File Regex	243
10.8. Centraliser des métadonnées d'un fichier XML	247
10.8.1. Configurer les métadonnées pour un fichier d'entrée (Input) XML	248
10.8.2. Configurer les métadonnées pour un fichier de sortie (Output) XML	257
10.9. Centraliser des métadonnées d'un fichier Excel	267
10.10. Centraliser les métadonnées d'un fichier LDIF	272
10.11. Centraliser les métadonnées d'un fichier JSON	277
10.11.1. Configurer les métadonnées JSON pour un fichier d'entrée	277
10.11.2. Configurer les métadonnées JSON pour un fichier de sortie	285
10.12. Centraliser les métadonnées d'une connexion LDAP	293
10.13. Centraliser les métadonnées Azure Storage	298
10.14. Centraliser les métadonnées Marketo	302
10.15. Centraliser les métadonnées d'une connexion Salesforce	306
10.16. Centraliser les métadonnées Snowflake	310
10.17. Configurer un schéma générique	314
10.17.1. Configurer un schéma générique à partir de rien	315
10.17.2. Configurer un schéma générique à partir d'un fichier XML	318
10.17.3. Enregistrer le schéma d'un composant en tant que schéma générique	320
10.18. Centraliser les métadonnées MDM ...	321
10.18.1. Définir la connexion	322
10.18.2. Définir un schéma MDM	324
10.19. Centraliser les métadonnées Web Service	340
10.19.1. Configurer un schéma simple	340
10.19.2. Configurer un schéma avancé	345
10.20. Centraliser une connexion FTP	353
10.21. Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion	356
10.21.1. Exporter des informations de connexion en tant que variables de contexte	356
10.21.2. Utiliser les variables d'un groupe de contextes existant afin de configurer une connexion	361
10.22. Comment utiliser les métadonnées centralisées dans un Job	366
chapitre 11. Gestion des métadonnées pour Talend Big Data	369
11.1. Gérer les métadonnées NoSQL	370
11.1.1. Centraliser les métadonnées de Cassandra	371
11.1.2. Centraliser les métadonnées de MongoDB	376
11.1.3. Centraliser les métadonnées de Neo4j	381
11.2. Gérer les métadonnées Hadoop	386
11.2.1. Centraliser une connexion Hadoop	387
11.2.2. Connexion à une distribution Hadoop personnalisée	400

11.2.3. Centraliser des métadonnées	
HBase	402
11.2.4. Centraliser les métadonnées	
MapR-DB	409
11.2.5. Centraliser des métadonnées	
HCatalog	411
11.2.6. Centraliser des métadonnées	
HDFS	416
11.2.7. Centraliser des métadonnées	
Hive	422
11.2.8. Centraliser une connexion	
Oozie	429
11.2.9. Configuration des propriétés réutilisables de Hadoop	432
chapitre 12. Gestion des routines	437
12.1. Définition des routines	438
12.2. Accès aux routines système	438
12.3. Personnalisation des routines système	439
12.4. Gestion des routines personnalisées	440
12.4.1. Crée des routines personnalisées	440
12.4.2. Modifier des routines	442
12.4.3. Modifier les bibliothèques des routines utilisateur	442
12.5. Appel d'une routine à partir d'un Job	444
12.6. Cas d'utilisation : créer un fichier à la date du jour	444
chapitre 13. SQL Templates	447
13.1. Qu'est-ce que l'ELT ?	448
13.2. Définition des SQL Templates Talend	448
13.3. Gérer les SQL Templates Talend	448
13.3.1. Les types de SQL Templates système	449
13.3.2. Accéder aux SQL Templates système	450
13.3.3. Crée des SQL Templates personnalisés	451
13.3.4. Exemple d'utilisation des SQL Templates système	453
annexe A. Personnalisation du Studio Talend et de ses préférences	459
A.1. Personnalisation des paramètres du projet	460
A.1.1. Configuration du niveau de conformité du compilateur	460
A.1.2. Personnaliser les modèles de scripts de build Maven	461
A.1.3. Personnaliser la Palette	464
A.1.4. Mapping de type	465
A.1.5. Gérer les versions	465
A.1.6. Gérer les statuts	467
A.1.7. Personnaliser les paramètres du Job	468
A.1.8. Configurer les informations de logs	468
A.1.9. Définir les paramètres de contexte	470
A.1.10. Appliquer les paramètres du projet	471
A.1.11. Définir les paramètres des Statuts	471
A.1.12. Paramètres de sécurité	473
A.2. Personnaliser l'espace de travail	473
A.2.1. Modifier la disposition et les paramètres de la Palette	474
A.2.2. Réorganiser les onglets d'un Job	477
A.2.3. Afficher les onglets/vues de configuration des Jobs	478
A.3. Filtrage des entrées listées dans la vue Repository	478
A.3.1. Filtrer avec le nom du Job	479
A.3.2. Filtrer avec l'utilisateur	480
A.3.3. Filtrer avec le statut du Job	482
A.3.4. Sélectionner les nœuds du Repository à afficher	483
A.4. Configuration des préférences du Studio Talend	485
A.4.1. Interpréteur Java (Talend)	485
A.4.2. Préférences du Designer (Talend > Appearance)	485
A.4.3. Définir le dossier de composants utilisateur (Talend > Components)	486
A.4.4. Définir les préférences de composants spécifiques (Talend > Components)	487
A.4.5. Préférences de la documentation (Talend > Documentation)	488
A.4.6. Préférence Exchange (Talend > Exchange)	489
A.4.7. Ajouter du code par défaut (Talend > Import/Export)	489
A.4.8. Préférences de langue (Talend > Internationalization)	490
A.4.9. Préférences de la Palette (Talend > Palette Settings)	490
A.4.10. Préférences de Performance (Talend > Performance)	491
A.4.11. Préférences d'exécution et de débogage (Talend > Run/Debug)	492
A.4.12. Afficher des caractères spéciaux dans les colonnes de schémas (Talend > Specific settings) ..	494
A.4.13. Préférences des schémas (Talend > Specific Settings)	494
A.4.14. Préférences du SQL Builder (Talend > Specific Settings)	496
A.4.15. Préférences du collecteur de données d'utilisation (Talend > Usage Data Collector)	496
annexe B. Routines système	499
B.1. Routines de type Numeric	500
B.1.1. Exemple de création séquentielle	500
B.1.2. Exemple de conversion décimale implicite	500
B.2. Routines de type Relational	501
B.3. Routines de type StringHandling	501
B.3.1. Exemple de vérification de tri alphabétique	503
B.3.2. Exemple de vérification de type alphabétique	504
B.3.3. Exemple de remplacement de chaîne	504
B.3.4. Exemple de vérification de chaîne	504
B.3.5. Exemple de calcul de longueur de chaîne	505
B.3.6. Exemple de nettoyage d'espaces inutiles	505
B.4. Routines de type TalendDataGenerator	505
B.4.1. Exemple de génération de données factices	506
B.5. Routines de type TalendDate	506
B.5.1. Exemple de formatage d'une Date	508

B.5.2. Exemple de vérification d'une	
Date	508
B.5.3. Exemple de comparaison de	
Dates	509
B.5.4. Exemple de configuration de	
Date	509
B.5.5. Exemple de parsage de Date	509
B.5.6. Exemple de récupération d'une	
partie d'une Date	510
B.5.7. Exemple de formatage de la	
Date courante	510
B.6. Routines de type TalendString	511
B.6.1. Exemple de formatage XML	
d'une chaîne	511
B.6.2. Exemple de trimming d'une	
chaîne	511
B.6.3. Exemple de désaccentuation	
d'une chaîne	512
B.7. Routines TalendStringUtil	512
annexe C. Règles d'écriture des SQL	
Templates	515
C.1. Instructions SQL	516
C.2. Lignes de commentaire	516
C.3. La syntaxe <% . . . %>	516
C.4. La syntaxe <%= . . . %>	517
C.5. La syntaxe </ . . . />	517
C.6. Le code pour accéder aux éléments du	
schéma des composants	518
C.7. Le code pour accéder aux propriétés de	
la matrice du composant	518

Préface

1. Informations générales

1.1. Objectif

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

Les informations contenues dans ce document s'appliquent à la version **7.0.1M2** de *Talend Open Studio for Big Data*.

1.2. Public visé

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



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.
- Les commandes sont surlignées en gris clair.

2. 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 site Web **Talend Community** (en anglais) disponible à l'adresse :

<https://community.talend.com/>



Chapitre 1. Intégration de données : principes et concepts

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.).

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



Chapitre 2. Introduction aux solutions Big Data de Talend

Les collections de données des entreprises sont de plus en plus volumineuses et complexes, particulièrement à l'ère d'Internet. Il est de plus en plus difficile de traiter des ensembles de données si volumineux et si complexes, à l'aide des outils de gestion traditionnels disponibles. Pour répondre à cette difficulté, une nouvelle plateforme d'outils "Big Data" a été créée, afin de gérer de manière cohérente de très grandes quantités de données, comme la plateforme Apache Hadoop Big Data Platform.

Construit sur les solutions d'intégration de données de **Talend**, les solutions Big Data de **Talend** fournissent un outil puissant permettant aux utilisateurs d'accéder aux données volumineuses, de les transformer, déplacer et de les synchroniser, en tirant parti de la plateforme Apache Hadoop Big Data et en facilitant l'utilisation de cette plateforme.

2.1. Hadoop et studio Talend

Lorsque des spécialistes en informatique parlent de Big Data, ils font généralement référence à des ensembles de données si volumineux et si complexes qu'ils ne peuvent être traités par les outils traditionnels de gestion de données. Ces grands volumes de données sont produits pour de nombreuses raisons. Des flux de données peuvent être générés automatiquement (rapports, logs, vidéosurveillance, etc.) ou peuvent résulter d'analyses détaillées du comportement des clients (données relatives à la consommation), des recherches scientifiques (le grand collisionneur de hadrons) ou de la consolidation de différentes sources de données.

Ces référentiels de données, contenant souvent des pétabytes et exabytes de données, sont difficiles à analyser, car les systèmes traditionnels de bases de données ne sont pas assez puissants. Les Big Data doivent être analysées dans des environnements massivement parallèles, dans lesquels la puissance de calcul est répartie entre des milliers d'ordinateurs et les résultats transférés dans un emplacement central.

La plateforme Open source Hadoop est devenue la plus utilisée pour l'analyse de données volumineuses. Ce système de fichiers distribué divise les informations en plusieurs blocs de données et répartit ces blocs dans différents systèmes du réseau (cluster Hadoop). En répartissant cette puissance de calcul, Hadoop assure un haut niveau de disponibilité et de redondance. Un "nœud maître" gère le stockage de fichiers ainsi que les requêtes.

Hadoop est une plateforme de calcul très puissante permettant de travailler avec des données volumineuses. Elle accepte les requêtes externes, les répartit dans des ordinateurs individuels dans le cluster puis les exécute en parallèle sur les nœuds individuels. Les résultats sont retournés vers un emplacement central, où ils peuvent être analysés.

Cependant, afin de tirer parti des avantages de Hadoop, les analystes de données doivent trouver un moyen de charger les données dans Hadoop et de les extraire de ce système Open source. C'est là qu'intervient le studio **Talend**.

Construit sur les solutions d'intégration de données de **Talend**, le studio permet aux utilisateurs de gérer facilement des Big Data en tirant parti de Hadoop, de ses bases de données ou de ses technologies, notamment HBase, HCatalog, HDFS, Hive, Oozie et Pig.

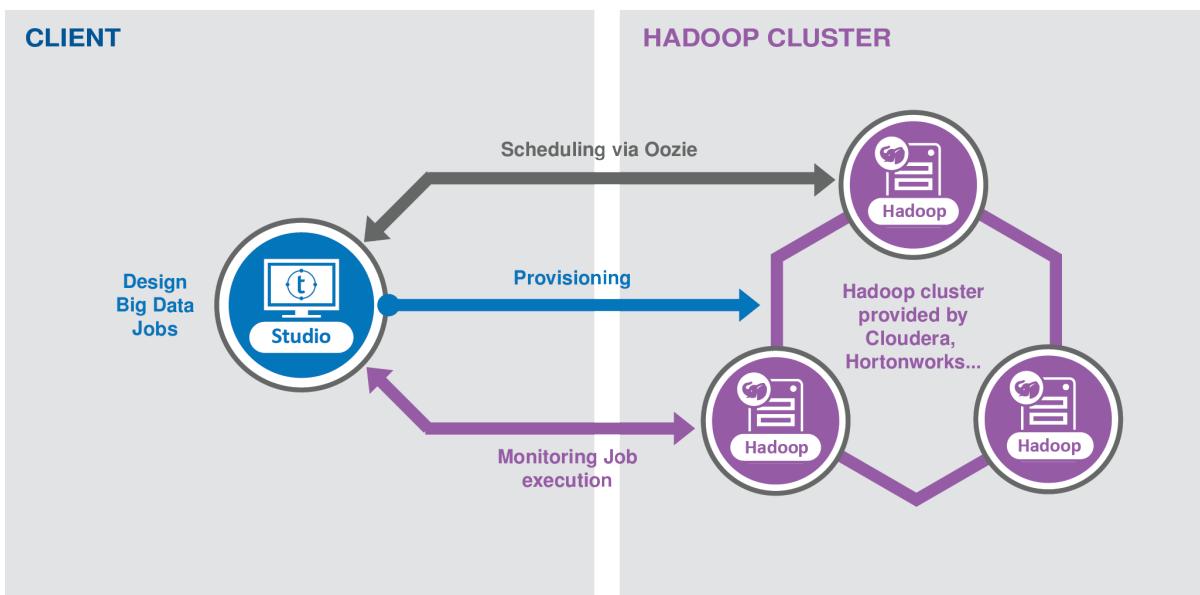
Le studio **Talend** est un environnement de développement graphique permettant les interactions avec des sources et des cibles Big Data, sans nécessité d'apprendre ou d'écrire du code. Une fois qu'une connexion Big Data est configurée, le code sous-jacent est automatiquement généré et peut être déployé en tant que service, exécutable ou Job standalone s'exécutant nativement dans votre cluster Big Data - HDFS, Pig, HCatalog, HBase, Sqoop ou Hive.

Les solutions Big Data de **Talend** fournissent un support complet de toutes les plateformes principales de Big Data. Les composants Big Data de **Talend** fonctionnent avec les distributions majeures de Hadoop, notamment Cloudera, Greenplum, Hortonworks et MapR. **Talend** offre un support clé en main d'un grand nombre de plateformes Big Data des principaux vendeurs d'appliance, comme Greenplum, Netezza, Teradata et Vertica.

2.2. Architecture fonctionnelle des solutions Big Data de Talend

L'architecture fonctionnelle des solutions de Big Data de **Talend** est un modèle architectural identifiant les fonctions, les différentes interactions et les ressources informatiques nécessaires. L'architecture globale isole les différentes fonctionnalités et les schématisé sous forme de blocs fonctionnels.

Le diagramme suivant illustre les principaux blocs fonctionnels concernant la gestion des Big Data dans le studio.



Les trois différents types de blocs fonctionnels sont définis comme suit :

- au moins un studio à partir duquel vous pouvez créer des Jobs Big Data tirant parti de la plateforme Apache Hadoop afin de gérer de grands volumes de données. Ces Jobs peuvent être exécutés localement ou déployés, planifiés et exécutés sur une grille (Grid) Hadoop via le système d'ordonnancement Oozie de workflows intégré dans le studio.
- un système d'ordonnancement de workflows intégré dans le studio, à travers lequel vous pouvez déployer, planifier et exécuter des Jobs Big Data dans une grille Hadoop et montrer le statut d'exécution, ainsi que les résultats des Jobs.
- Une grille (Grid) Hadoop indépendante du système **Talend** pour gérer d'importants ensembles de données.



Chapitre 3. Travailler avec les projets

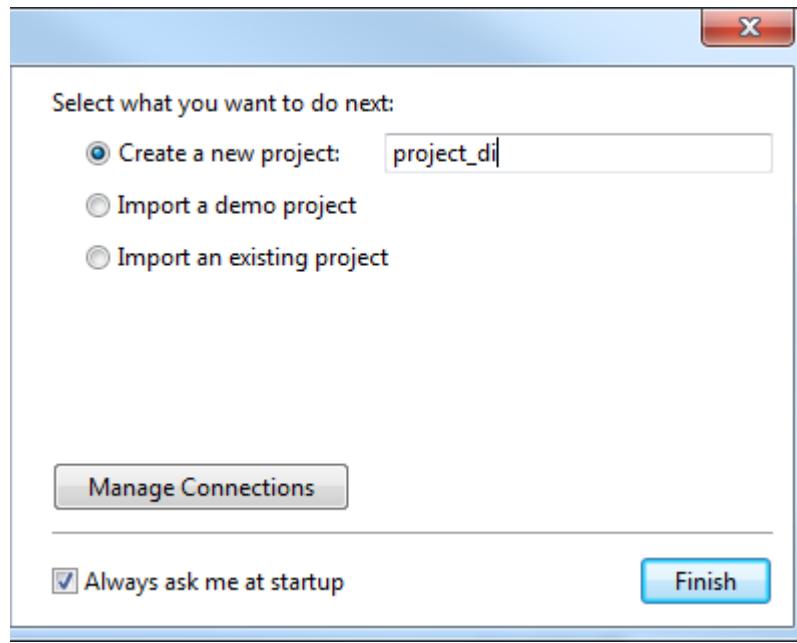
Lorsque vous lancez votre *Studio Talend* et avant de commencer un Business Model, un Job d'intégration de données, ou n'importe quelle autre tâche, vous devez d'abord créer ou importer un projet.

Ce chapitre explique comment créer, importer, exporter, supprimer et travailler sur des projets dans *Studio Talend*. Pour le lancement et la prise en main du *Studio Talend*, consultez le Guide de prise en main.

3.1. Créer un projet

Pour créer un projet lors du premier démarrage du Studio, procédez comme suit :

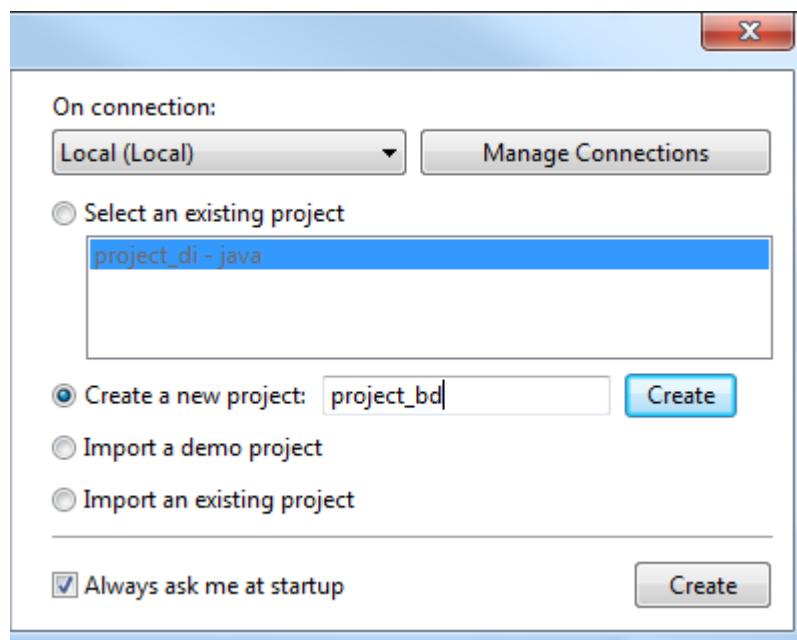
1. Lancez le *Studio Talend*.
2. Dans la fenêtre de connexion, sélectionnez l'option **Create a new project** et saisissez un nom de projet dans le champ.



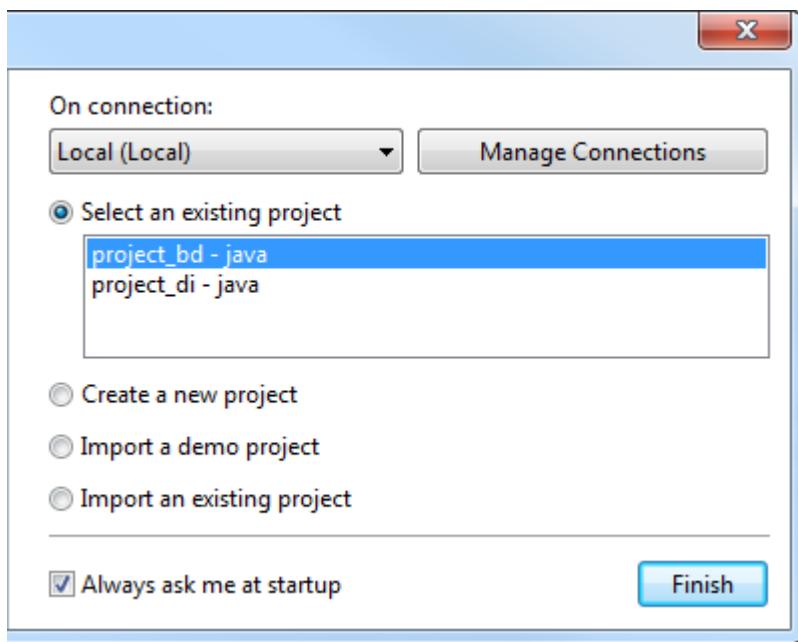
3. Cliquez sur **Finish** pour créer le projet et l'ouvrir dans le Studio.

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

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



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



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

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

3.2. Importer un projet Démo

Talend fournit plusieurs projets démo que vous pouvez importer dans votre *Studio Talend*. Les projets démo disponibles dépendent de votre licence et peuvent inclure des Jobs, des Services de données ou des analyses de profiling prêt(e)s à l'emploi, vous permettant de comprendre les fonctionnalités des différents composants **Talend**.

Vous pouvez importer le projet démo depuis la page de connexion à votre studio en tant que projet séparé, ou depuis la perspective **Integration** dans votre projet courant.

Pour importer un projet démo en tant que projet séparé dans le studio :

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



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

3. Dans la boîte de dialogue qui s'ouvre, saisissez un nom pour le projet démo que vous souhaitez importer et cliquez sur **Finish**.

Une barre s'affiche, indiquant la progression de l'opération.

4. Dans la fenêtre de connexion, sélectionnez dans la liste le projet démo importé et cliquez sur **Finish** pour ouvrir ce projet dans le studio.

Toutes les données du projet démo sont importées dans le studio, dans différentes dossiers du référentiel, notamment les fichiers d'entrée et les métadonnées de connexion nécessaires à l'exécution des exemples de la démo.

Pour importer le projet démo dans votre projet courant dans le studio :

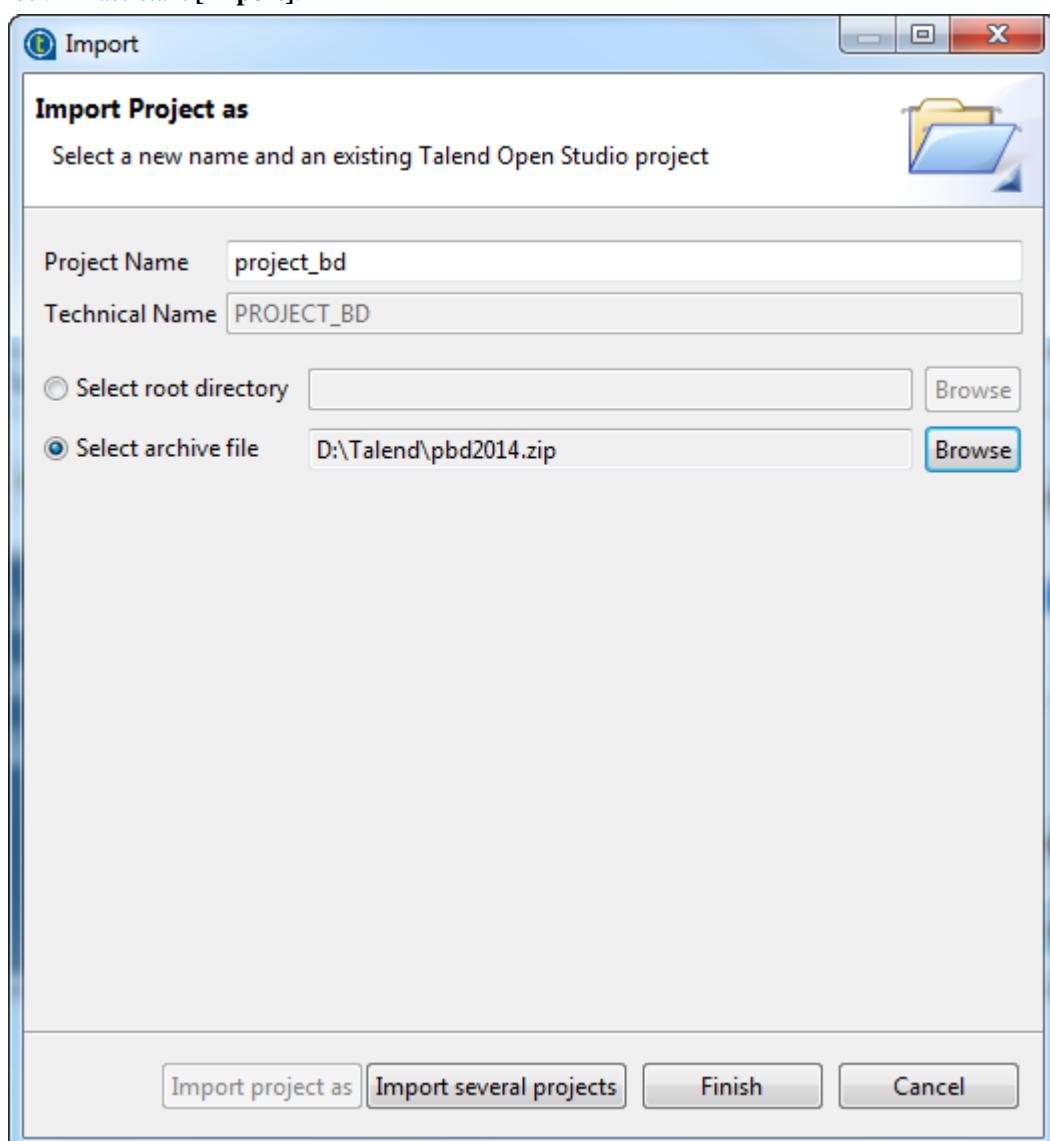
1. Lancez votre studio et, dans la perspective **Integration**, cliquez sur l'icône  dans la barre d'outils.
2. Dans la boîte de dialogue qui s'ouvre, sélectionnez le projet démo à importer et cliquez sur **Finish**.
Une barre s'affiche, indiquant la progression de l'opération, puis un message de confirmation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

3.3. Importer un projet

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

Pour importer un projet, procédez comme suit :

1. Dans la fenêtre de connexion du Studio, sélectionnez **Import an existing project** puis cliquez sur **Select** pour ouvrir l'assistant **[Import]**.

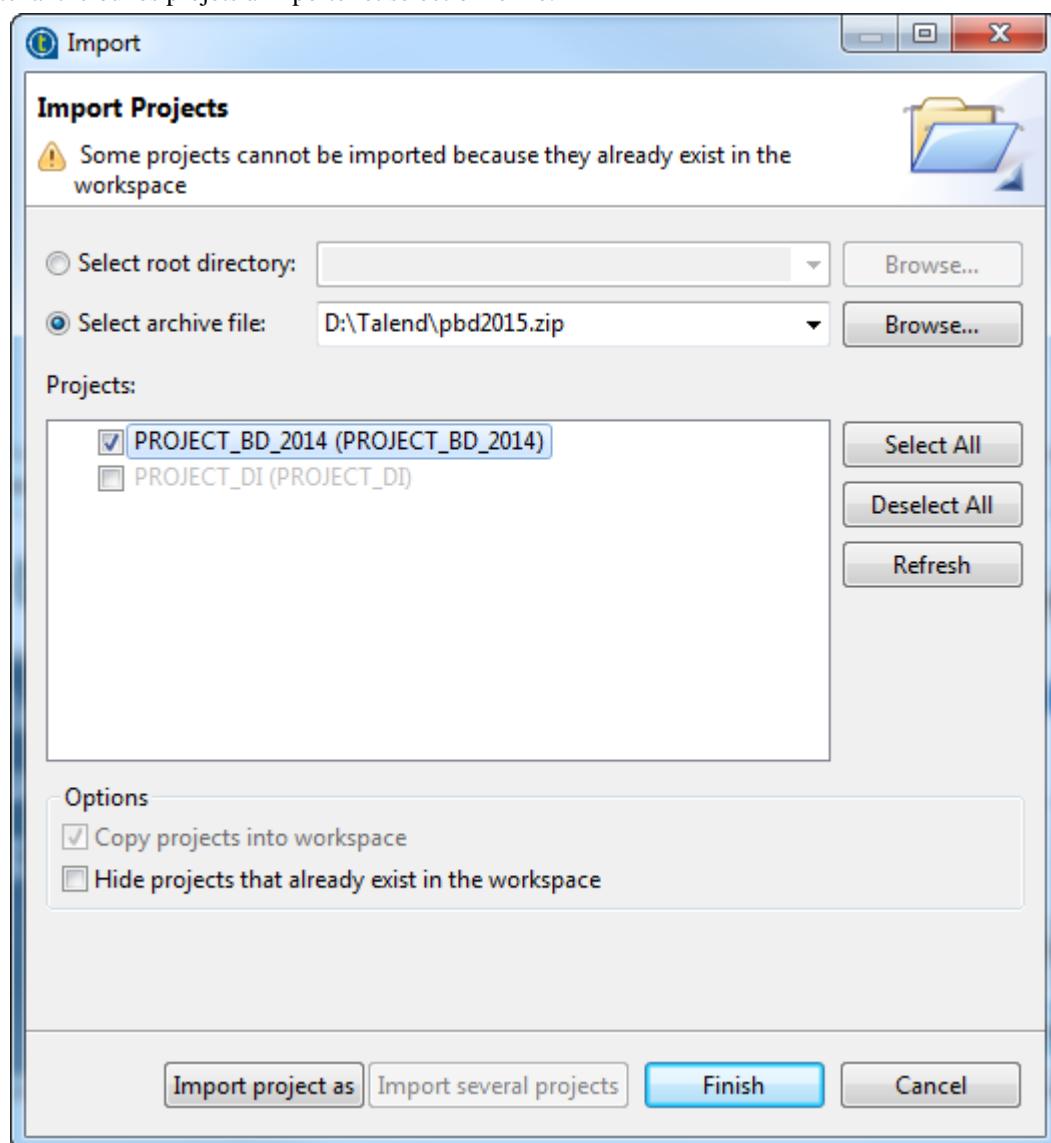


2. Cliquez sur le bouton **Import project as** et saisissez un nom pour votre nouveau projet dans le champ **Project Name**.

3. Cliquez sur **Select root directory** ou **Select archive file** selon la source de laquelle vous souhaitez importer.
4. Cliquez sur **Browse...** pour sélectionner le répertoire Workspace/le fichier d'archive du dossier du projet spécifique. Par défaut, le Workspace sélectionné est celui qui correspond à la version de votre Studio. Parcourez votre système jusqu'au répertoire Workspace du Studio contenant les projets à importer.
5. Cliquez sur **Finish** pour valider l'opération et retourner à la fenêtre de connexion.

Pour importer simultanément différents projets, procédez comme suit :

1. Dans la fenêtre de connexion au Studio, sélectionnez **Import an existing project** puis cliquez sur **Select** pour ouvrir l'assistant [**Import**].
2. Cliquez sur **Import several projects**.
3. Cliquez sur **Select root directory** ou **Select archive file** en fonction du type de source à partir de laquelle vous souhaitez faire l'import : respectivement une répertoire ou un fichier archive.
4. 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 du *Studio Talend* ou le fichier archive contenant le ou les projets à importer et sélectionnez-le.



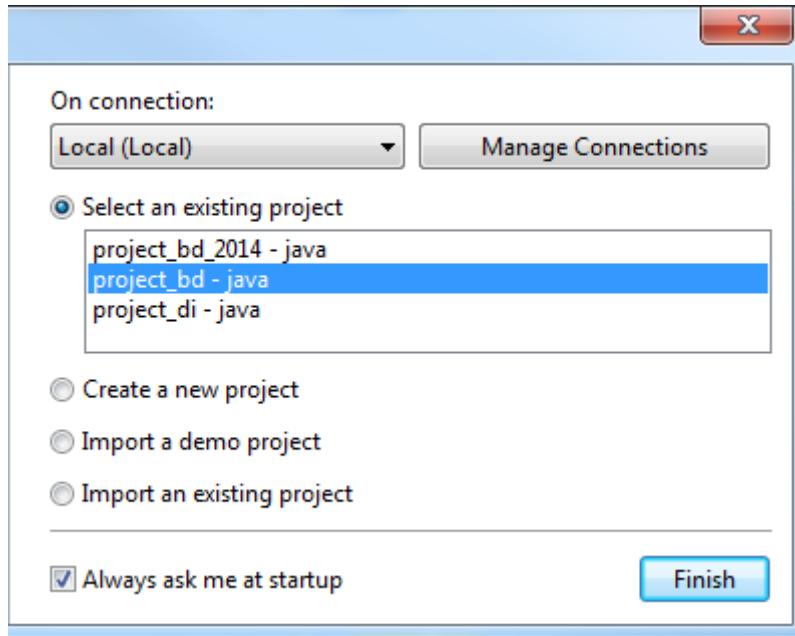
5. Cochez la case **Copy projects into workspace** pour faire une copie des projets importés dans le nouveau Workspace. Cette option est disponible lorsque vous importez plusieurs projets depuis un répertoire racine.



Si vous ne souhaitez pas déplacer les projets de leur emplacement initial vers le nouveau Workspace, décochez cette case. Il est recommandé de conserver une version de sauvegarde.

6. Cochez la case **Hide projects that already exist in the workspace** pour masquer les projets existants dans la liste **Projects**. Cette option est disponible uniquement lorsque vous importez plusieurs projets.
7. Dans la liste **Projects**, sélectionnez les projets à importer et cliquez sur **Finish** pour valider.

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



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

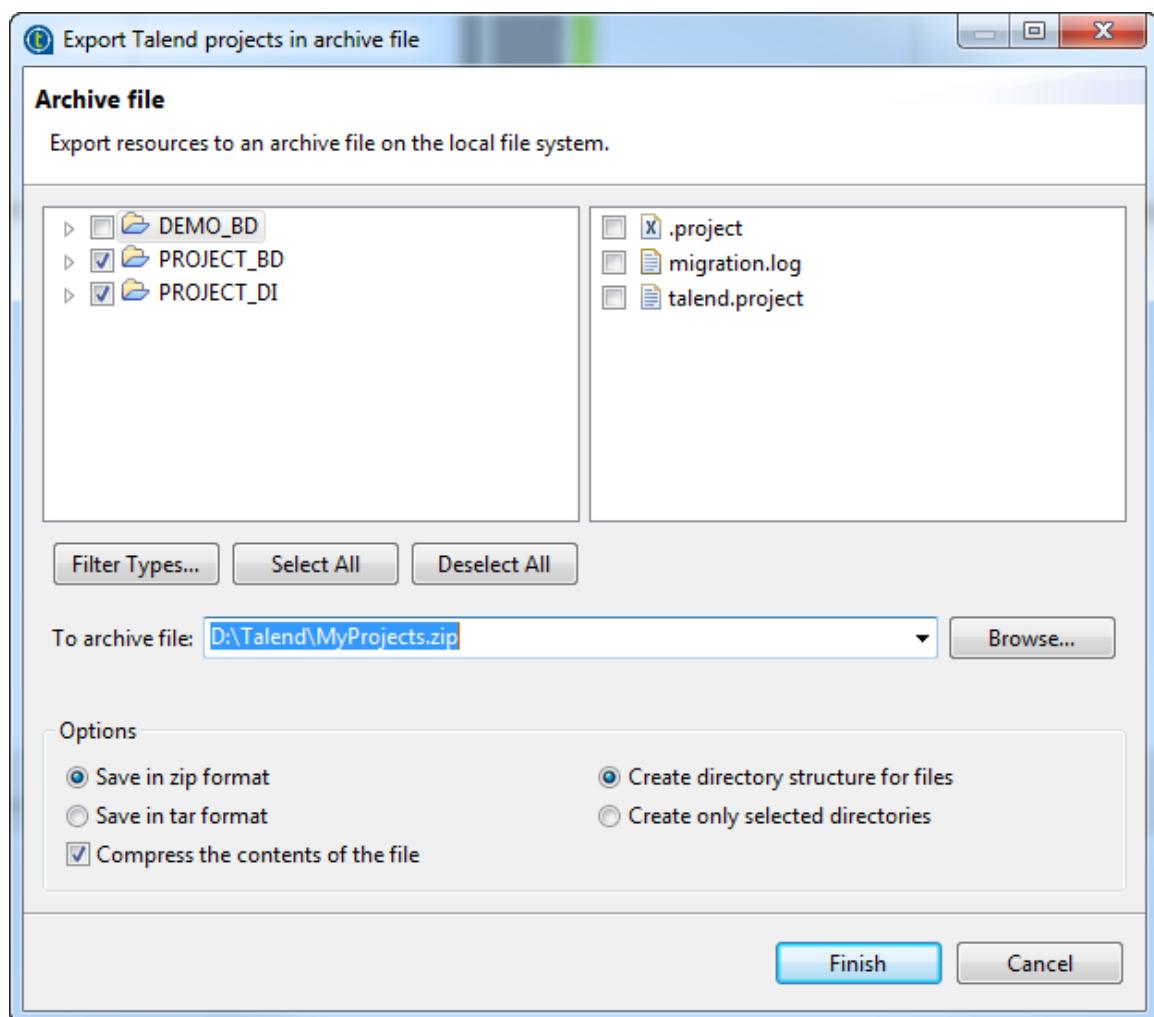


Une fenêtre d'initialisation du générateur de code peut apparaître lors du lancement de l'application. Il est recommandé d'attendre que l'initialisation se termine.

3.4. Exporter un projet

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

1. 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.



Chapitre 4. Conception d'un Business Model

La perspective **Integration** du *Studio Talend* a été spécialement conçu pour permettre une représentation simple et claire des besoins métiers de l'entreprise, ainsi que leur interaction. Le *Studio Talend* 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.

4.1. 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 goulets 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 la perspective **Integration** du *Studio Talend* par de multiples formes et liens disponibles dans la **Palette**.

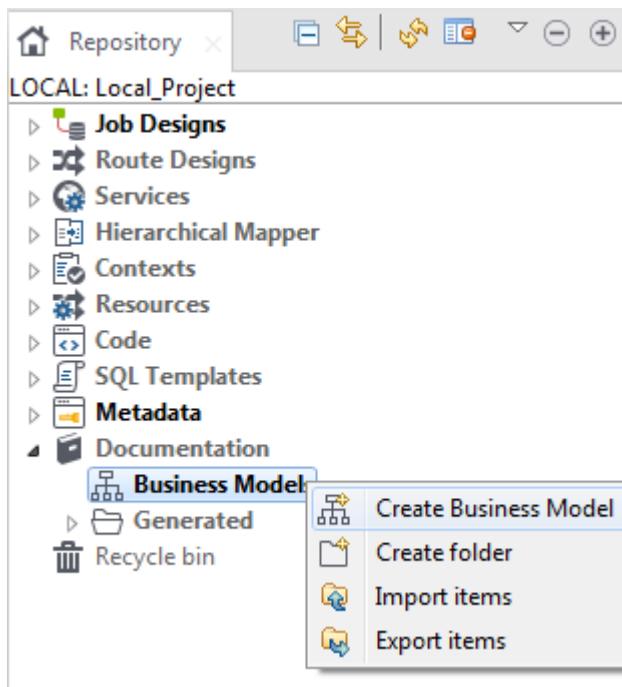
Toutes les formes et tous les liens peuvent être décrits dans leurs propriétés à l'aide des paramètres de documentation et de formats que vous définissez dans le référentiel de métadonnées, ou directement dans votre Business Model.

A partir de la perspective **Integration** du *Studio Talend*, vous disposez de nombreux outils vous permettant de :

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

4.2. Crédit d'un Business Model

1. Dans la fenêtre principale de la perspective **Integration** du *Studio Talend*, cliquez-droit sur le nœud **Business Models** et sélectionnez **Create Business Model**.



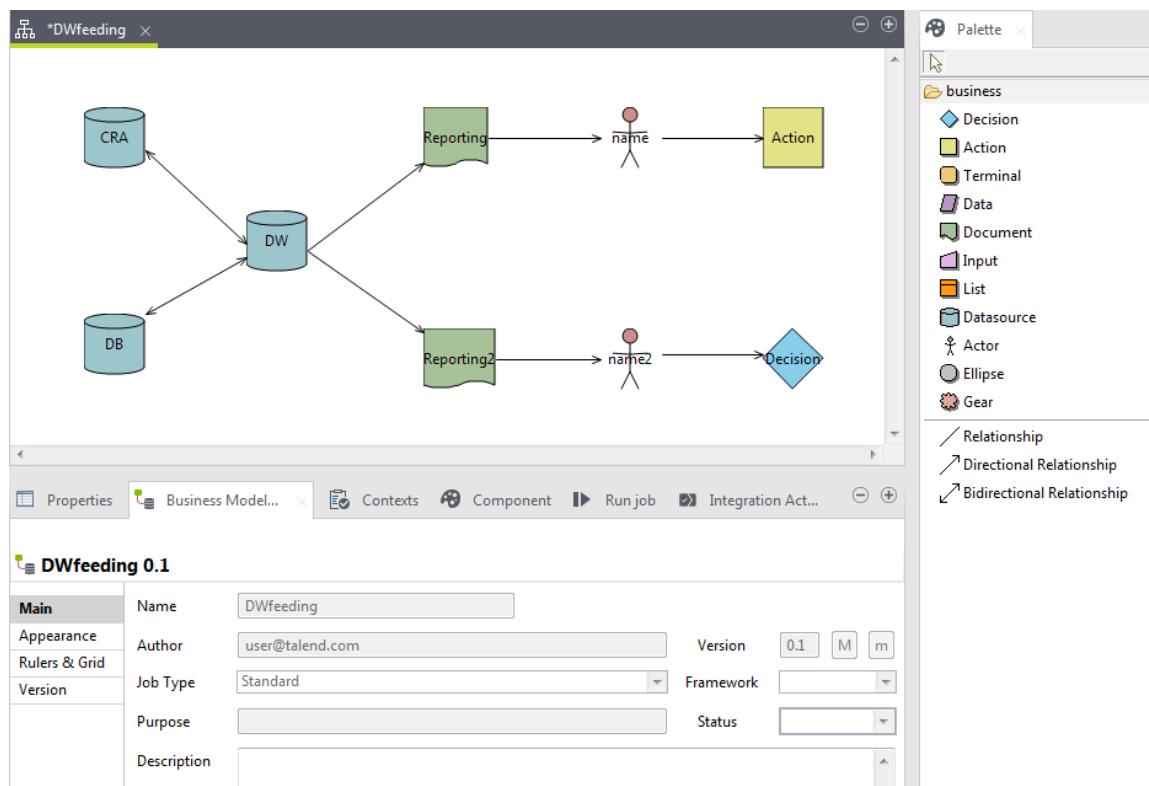
L'assistant de création vous guide à travers les étapes de création d'un nouveau Business Model.

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

Champ	Description
Name	Nom du nouveau Business Model. Un message s'affiche si vous saisissez des caractères interdits.
Purpose	Objectif du Business Model ou toute information utile concernant son utilisation.
Description	Description du Business Model.
Author	Champ en lecture seule affichant par défaut le nom de l'utilisateur courant.
Locker	Champ en lecture seule affichant par défaut le nom de l'utilisateur courant ayant verrouillé le Job. Ce champ est vide lorsque vous créez un Business Model et contient des données uniquement lorsque vous modifiez les propriétés d'un Business Model existant.
Version	Champ en lecture seule. Vous pouvez incrémenter manuellement la version en utilisant les boutons M (majeure) et m (mineure).
Status	Liste dans laquelle sélectionner le statut du Business Model en cours de création.
Path	Liste dans laquelle sélectionner le dossier dans lequel créer le Business Model.

3. Le **Modeler** s'ouvre dans l'espace de modélisation graphique vide.

Vous pouvez créer autant de modèles que vous le souhaitez et tous les ouvrir.



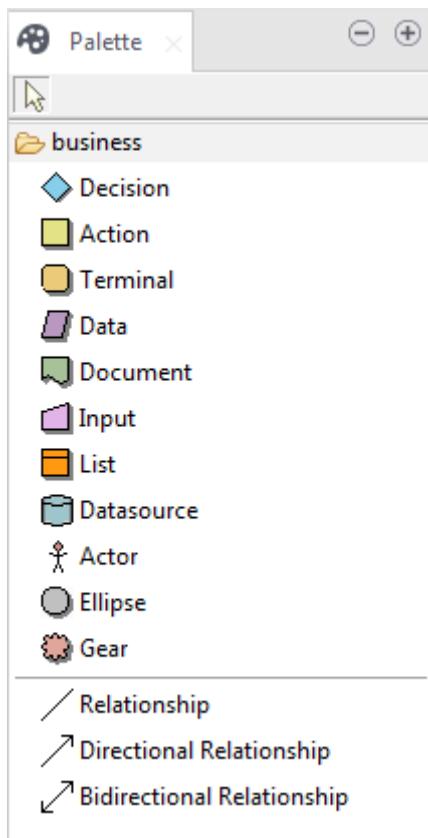
Le **Modeler** se compose des panneaux suivants :

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

4.3. 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 bibliothèques de formes.

4.3.1. Formes (Shapes)

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

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



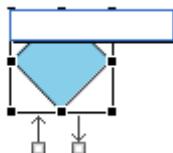
Par exemple, si votre Business Model inclut une étape de décision, sélectionnez la forme losange dans la **Palette**.



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

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

La forme déposée est entourée d'un cadre noir dont les points d'angle vous permettent de la redimensionner selon vos besoins.



En outre, une zone de saisie bleue vous permet d'apposer un libellé sur la forme en sélection. Donnez un nom parlant qui vous permettra ensuite d'identifier rapidement le rôle de cette forme dans le process.

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

Voir également : [Connexions](#).

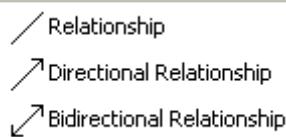
Les formes disponibles comprennent :

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

4.3.2. 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.

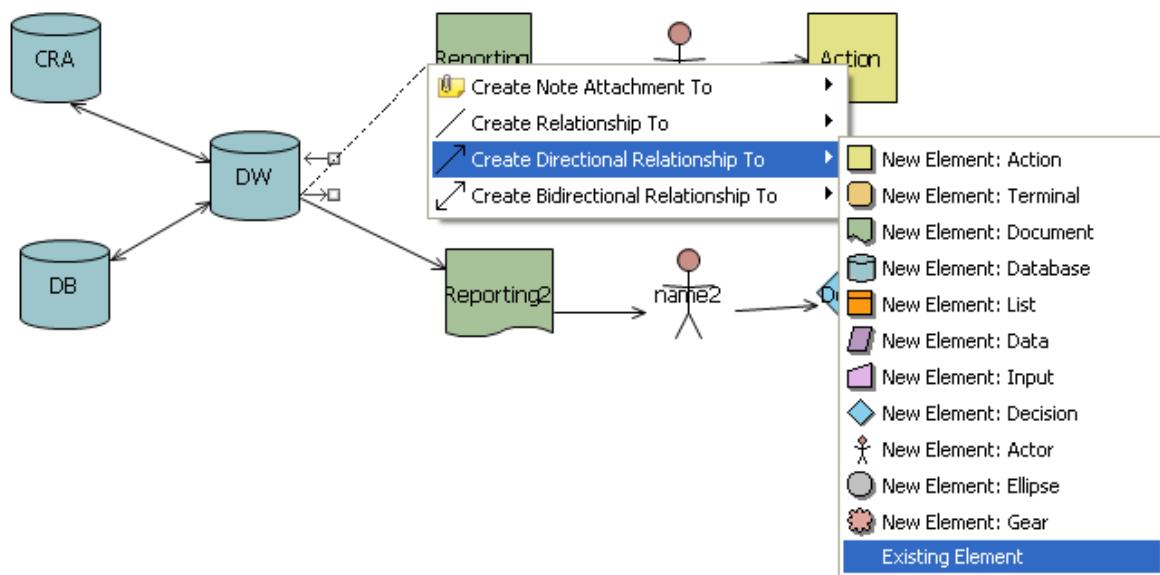
Plusieurs solutions sont offertes :



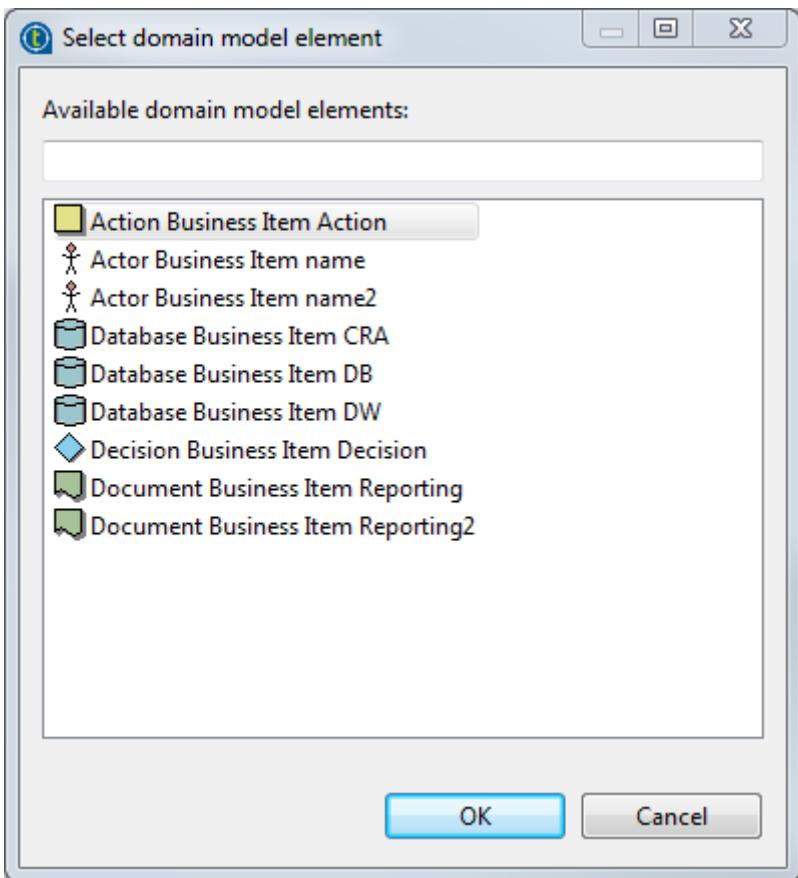
Sélectionnez l'outil **Relationship** de la **Palette**. Puis dans l'éditeur graphique, tirez un lien d'une forme vers une autre pour dessiner la connexion.

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 [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 : [Commenter et réarranger un Business Model](#)

4.3.3. 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.

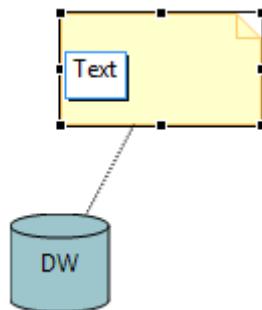
4.3.3.1. Ajouter une note ou du texte libre

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.

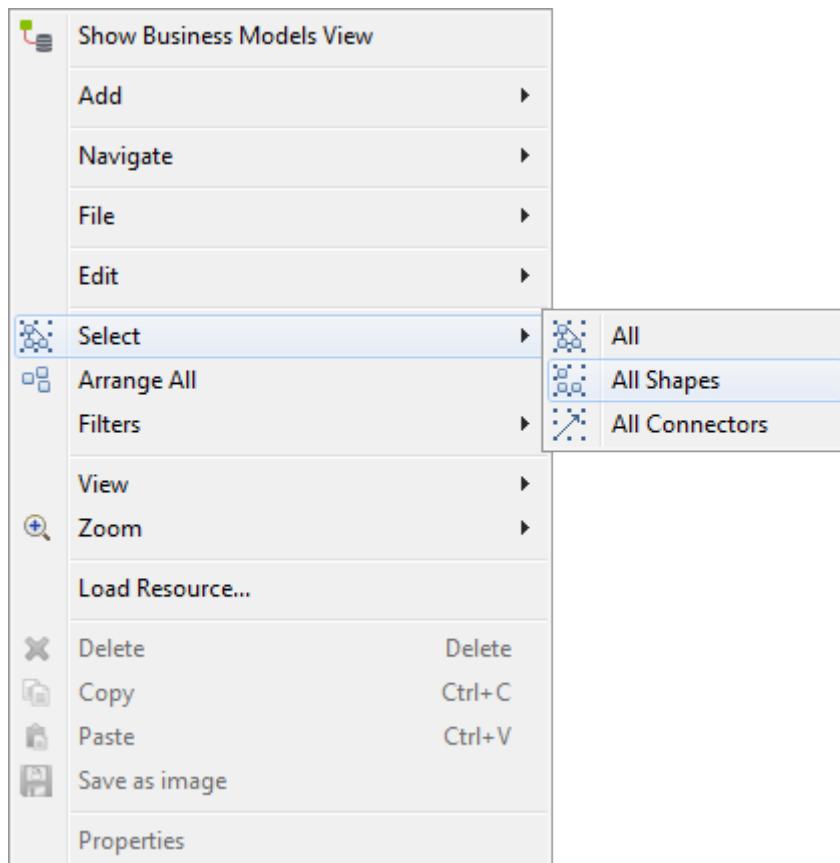


Pour ajouter une note attachée à une forme, sélectionnez **Note attachment** dans la **Palette** puis sélectionnez dans l'espace de modélisation la forme que vous souhaitez annoter. Lorsque la flèche noire apparaît, faites-la glisser vers une zone vide de l'éditeur. Relâchez pour afficher le menu contextuel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

4.3.3.2. 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.

4.3.4. 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 [Afficher les onglets/vues de configuration des Jobs](#).

4.3.4.1. 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 :

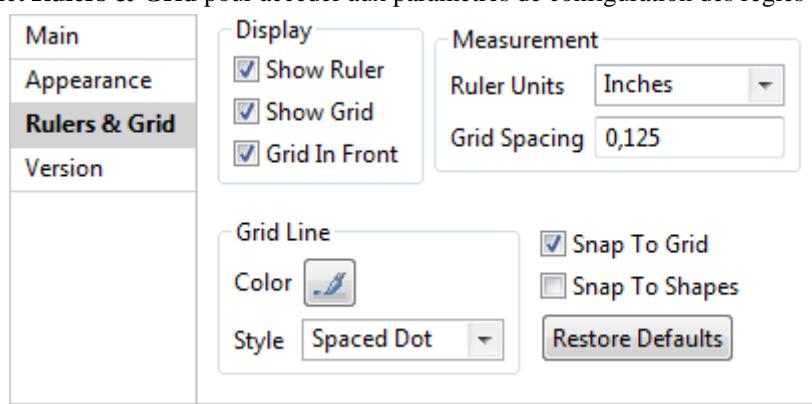
- changer la couleur de la forme
- colorer les bordures
- ajouter du texte
- ajouter un effet dégradé
- ajouter un effet ombré

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.

4.3.4.2. 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.



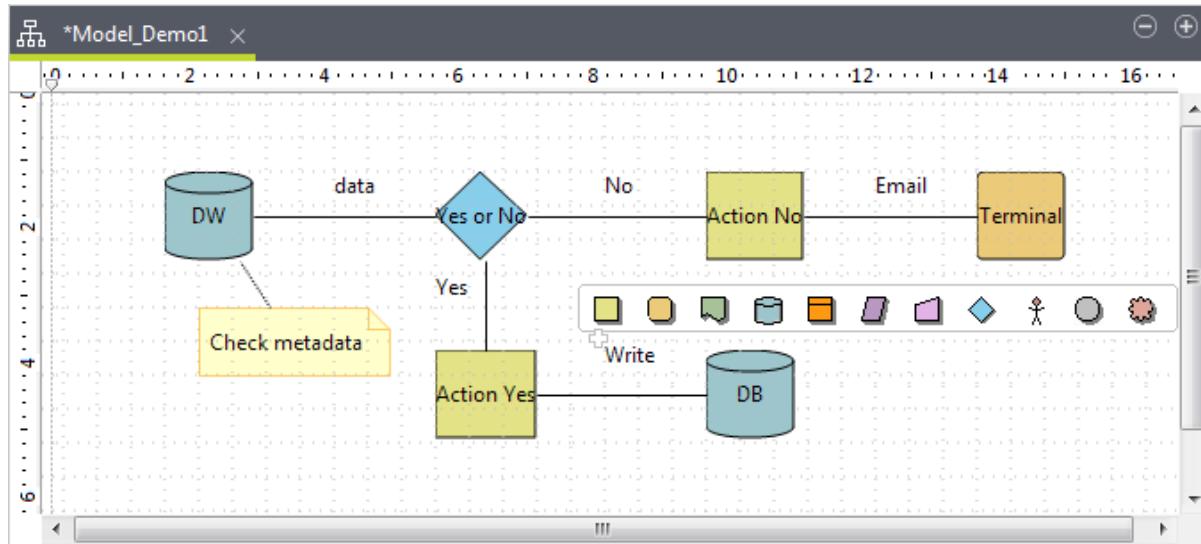
Dans la zone **Display**, cochez la case **Show Ruler** pour afficher la Règle (**Ruler**), ou **Show Grid** pour afficher la Grille (**Grid**) ou les deux. **Grid in front** place la grille devant toutes les autres formes du process.

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**.



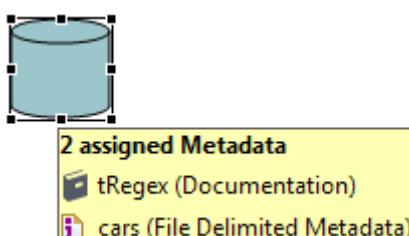
4.3.4.3. Onglet Assignment

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	
File Delimited Metadata	cars	

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

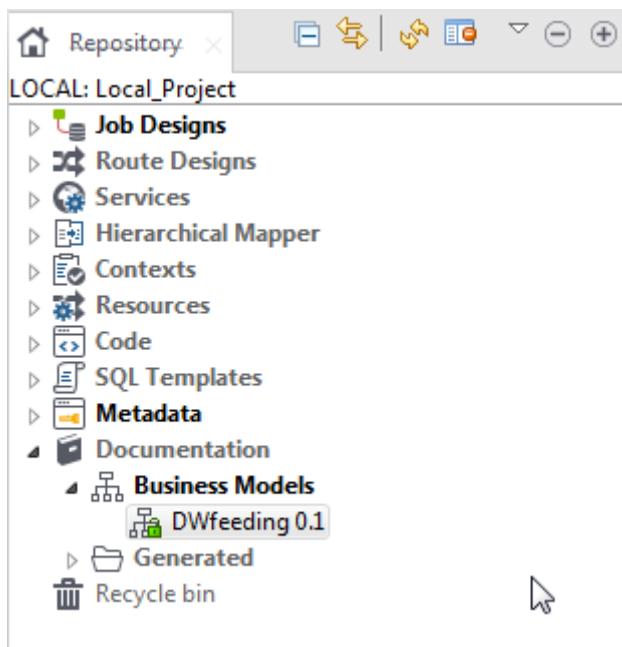


Vous pouvez modifier certaines informations ou lier un commentaire. En outre, si vous mettez à jour des données dans le référentiel, les informations d'affectation sont automatiquement mises à jour.

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

4.4. 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 :

Élément	Description
Job Designs	Si la représentation graphique d'un Job est disponible, elle peut être réutilisée comme métadonnée du Business Model actif.
Metadata	Toute donnée descriptive conservée dans le référentiel peut être affectée à un objet de la vue. Il peut s'agir par exemple des coordonnées de connexion à une base de données.
Business Models	Vous pouvez réutiliser comme métadonnées tout Business Model conservé dans le référentiel de ce projet.

Élément	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 [*Gestion des métadonnées dans l'intégration de données*](#)



Chapitre 5. Conception de Jobs

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

Les Jobs d'intégration de données que vous modélisez dans le *Studio Talend* 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 du *Studio Talend*.

5.1. Qu'est-ce qu'un Job ?

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 le *Studio Talend*, vous pouvez :

- modéliser des actions d'intégration de données grâce à la bibliothèque de composants techniques,
- changer les paramètres par défaut des composants et même créer de nouveaux composants ou familles de composants qui correspondent au mieux à vos besoins,
- 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 le *Studio Talend*, vous devez installer une JVM Oracle 1.8 ou une version ultérieure (les JVM IBM ne sont pas supportées). Vous pouvez la télécharger sur <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

5.2. Prise en main d'un Job simple

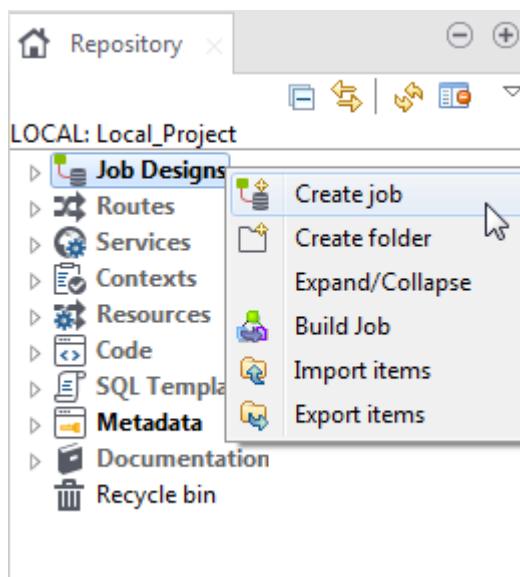
Cette section présente un exemple pas à pas qui montre comment créer un Job, ajouter des composants, les configurer et exécuter le Job. Ce Job se nomme *A_Basic_Job*. Il lit un fichier texte, affiche son contenu dans la console **Run** et écrit les données dans un autre fichier texte.

5.2.1. Créer un Job

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

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

1. Dans la vue **Repository** de la perspective **Integration**, cliquez-droit sur le nœud **Job Designs** et sélectionnez **Create job** dans le menu contextuel.



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

Name	A_Basic_Job
Purpose	This Job aims to create a Basic Job.
Description	This Job is a simple design to present the basic functionalities and configuration of a Talend Job.
Author	user@talend.com
Locker	
Version	0.1
Status	
Path	Select
<input type="button" value="Finish"/> <input type="button" value="Cancel"/>	

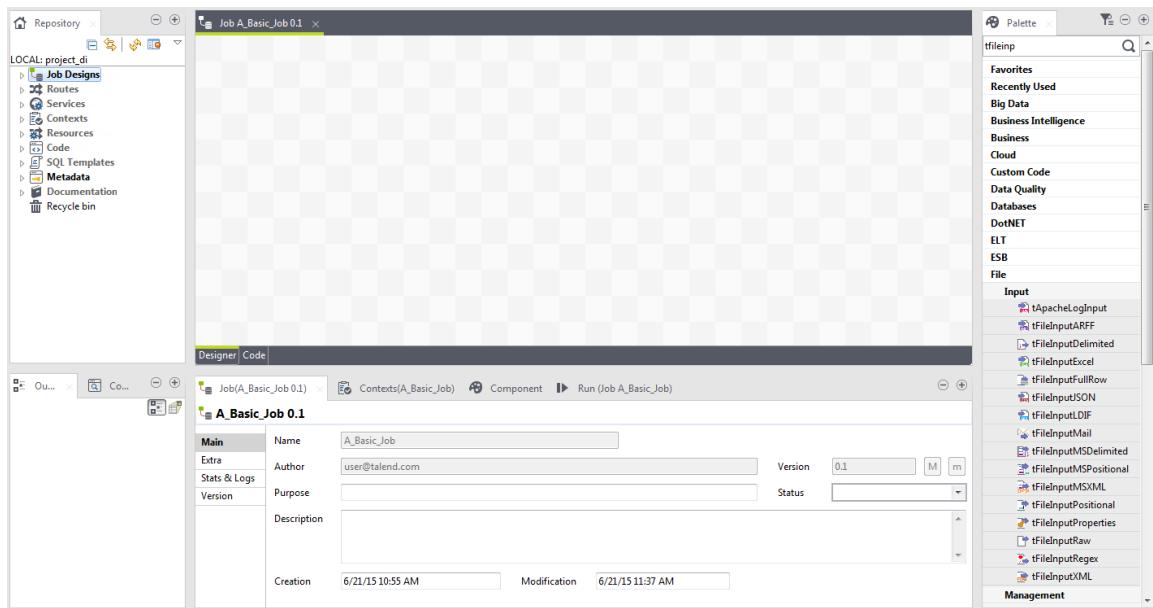
2. Renseignez les propriétés du Job comme montré dans la capture d'écran.

Les champs correspondent aux propriétés suivantes :

Champ	Description
Name	Saisissez un nom pour le nouveau Job. Un message s'affiche si vous saisissez des caractères interdits.
Purpose	Saisissez toute information que vous jugerez utile concernant l'utilisation du Job.

Champ	Description
Description	Si nécessaire, saisissez une description contenant toute information permettant de décrire ce que fait le Job et comment il le fait.
Author	Ce champ est en lecture seule car il reprend par défaut l'identifiant de l'utilisateur courant.
Locker	Ce champ est en lecture seule car il reprend par défaut l'identifiant de l'utilisateur ayant verrouillé le Job courant. Ce champ est vide lorsque vous créez un Job, et n'a des données que lorsque vous éditez les propriétés d'un Job existant.
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 Gestion des versions d'un Job .
Status	Sélectionnez dans la liste le statut du Job que vous créez.
Path	Sélectionnez le dossier dans lequel vous souhaitez créer le Job.

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

5.2.2. Ajouter des composants au Job

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

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

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

Cette section décrit les trois premières méthodes. Pour plus d'informations sur la manière de déposer un composant depuis le nœud **Metadata**, consultez le [Gestion des métadonnées dans l'intégration de données](#) et le [Gestion des métadonnées pour Talend Big Data](#).

5.2.2.1. Déposer le premier composant depuis la Palette

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

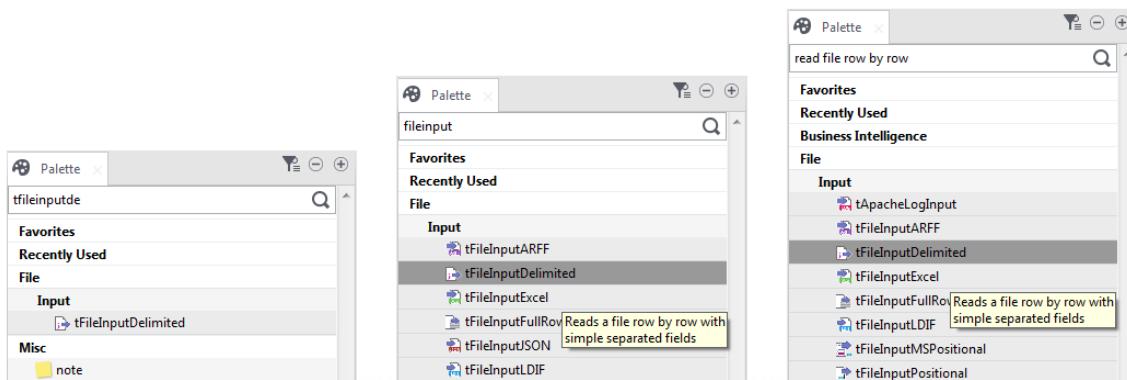
Pour déposer un composant depuis la **Palette**, procédez comme suit :

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

Les mots-clés peuvent être le nom complet ou partiel du composant ou une phrase décrivant sa fonctionnalité, si vous ne connaissez pas son nom, par exemple, *tfileinputde*, *fileinput*, ou *read file row by row*. Si la langue de votre Studio est le français, vous pouvez saisir vos mots-clés en français. La **Palette** affiche uniquement les familles dans lesquelles le composant peut être trouvé. Si la **Palette** n'est pas affichée dans le Studio, consultez [Afficher, cacher et déplacer la Palette](#).



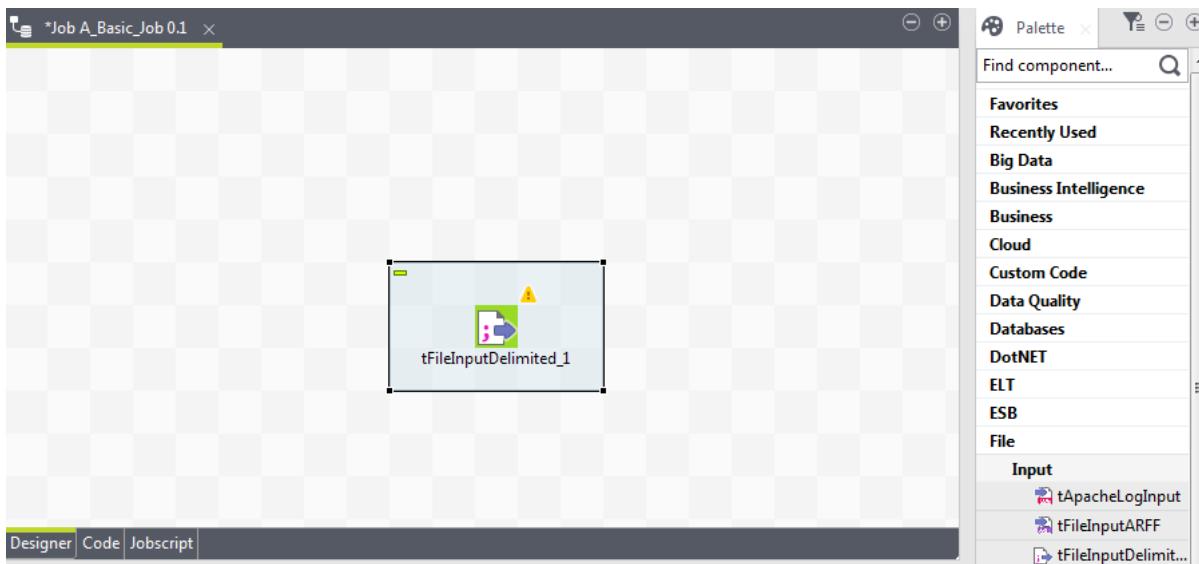
Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case **Also search from Help when performing a component searching** dans la fenêtre des **Preferences > Palette Settings**. Pour plus d'informations, consultez [Préférences de la Palette \(Talend > Palette Settings\)](#).



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

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

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



5.2.2.2. Ajouter le deuxième composant en saisissant son nom dans l'espace de modélisation graphique

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

Prérequis : Assurez-vous d'avoir coché la case **Enable Component Creation Assistant** dans les préférences du Studio. Pour plus d'informations, consultez la [Comment utiliser les métadonnées centralisées dans un Job](#).

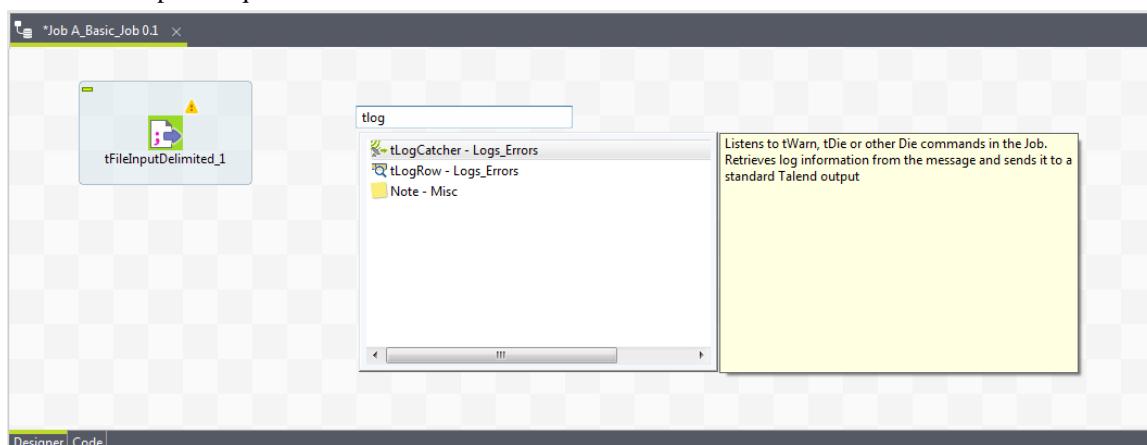
Pour ajouter un composant directement dans l'espace de modélisation graphique, procédez comme suit :

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



Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case **Also search from Help when performing a component searching** dans la fenêtre des **Preferences > Palette Settings**. Pour plus d'informations, consultez [Préférences de la Palette \(Talend > Palette Settings\)](#).

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



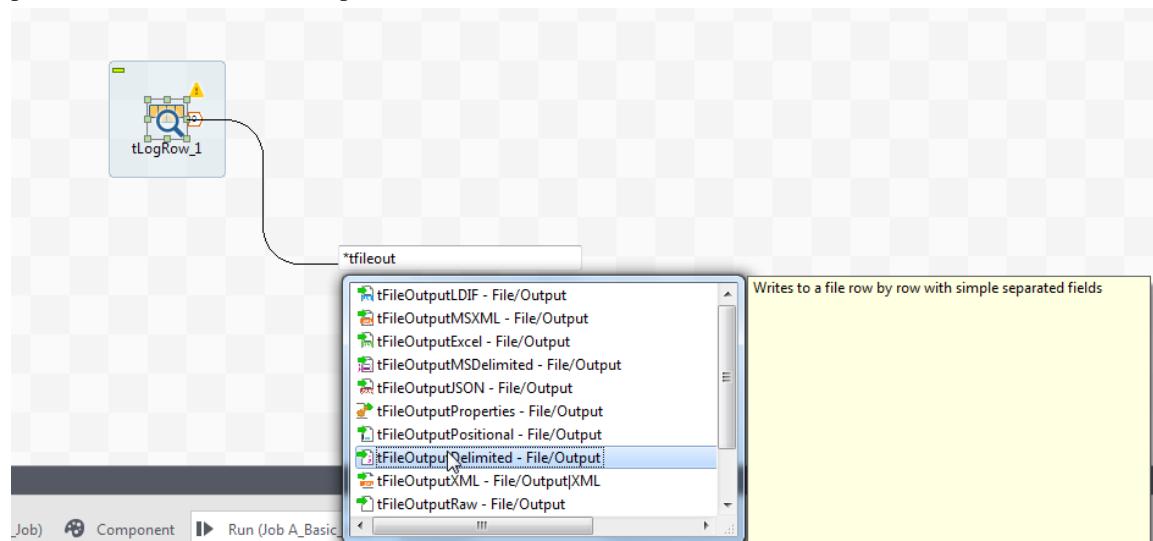
2. Double-cliquez sur le composant que vous souhaitez utiliser, **tLogRow** dans cet exemple, afin de l'ajouter dans l'espace de modélisation graphique.

5.2.2.3. Ajouter un composant de sortie en le glissant depuis un composant d'entrée

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

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

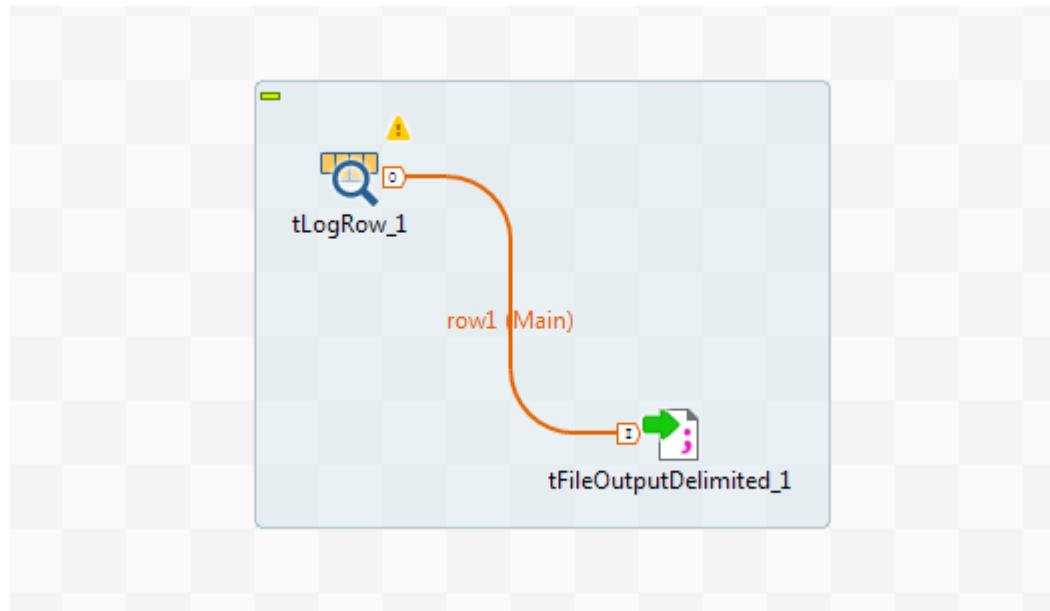
Un champ textuel et une liste de composants s'affichent. La liste des composants affiche tous les composants pouvant être connectés au composant d'entrée.



3. Pour réduire les résultats de la recherche, saisissez dans le champ le nom du composant à ajouter, ses premières lettres ou une phrase décrivant sa fonctionnalité si vous ne connaissez pas son nom, puis double-cliquez sur le composant de votre choix, le **tFileOutputDelimited** dans cet exemple, dans la liste de composants, pour l'ajouter à l'espace de modélisation graphique. Le nouveau composant est automatiquement connecté au composant d'entrée, le **tLogRow**, à l'aide d'un lien **Row > Main**.



Pour utiliser une phrase descriptive comme mots-clés pour une recherche floue, assurez-vous de cocher la case **Also search from Help when performing a component searching** dans la fenêtre des **Preferences > Palette Settings**. Pour plus d'informations, consultez [Préférences de la Palette \(Talend > Palette Settings\)](#).



5.2.3. Connecter les composants

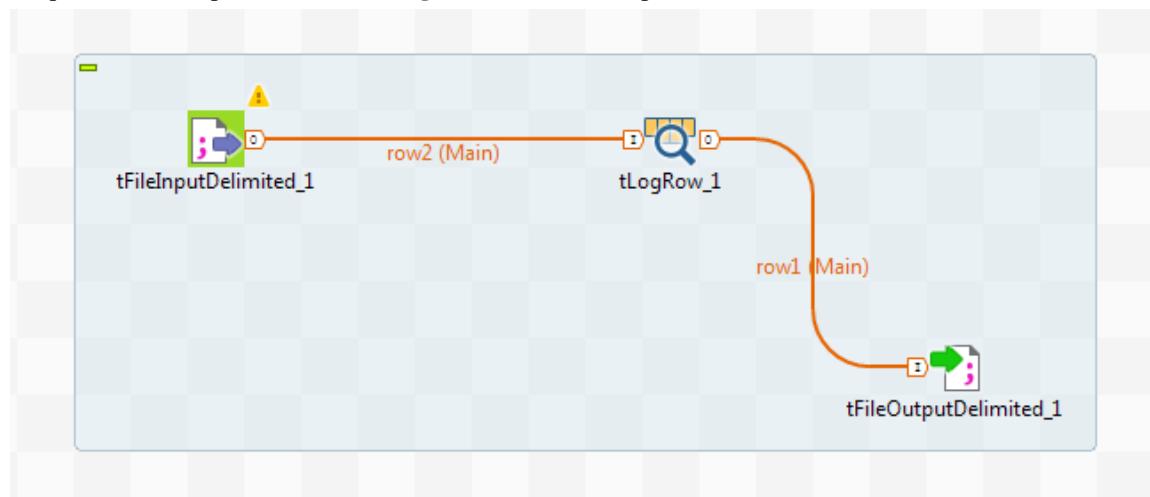
Maintenant que les composants sont ajoutés dans l'espace de modélisation graphique, ils doivent être connectés entre eux. Lorsque des composants sont connectés, ils forment un sous-job. Les Jobs sont composés d'un ou plusieurs sous-jobs effectuant différentes opérations.

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

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

Méthode par clics

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



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

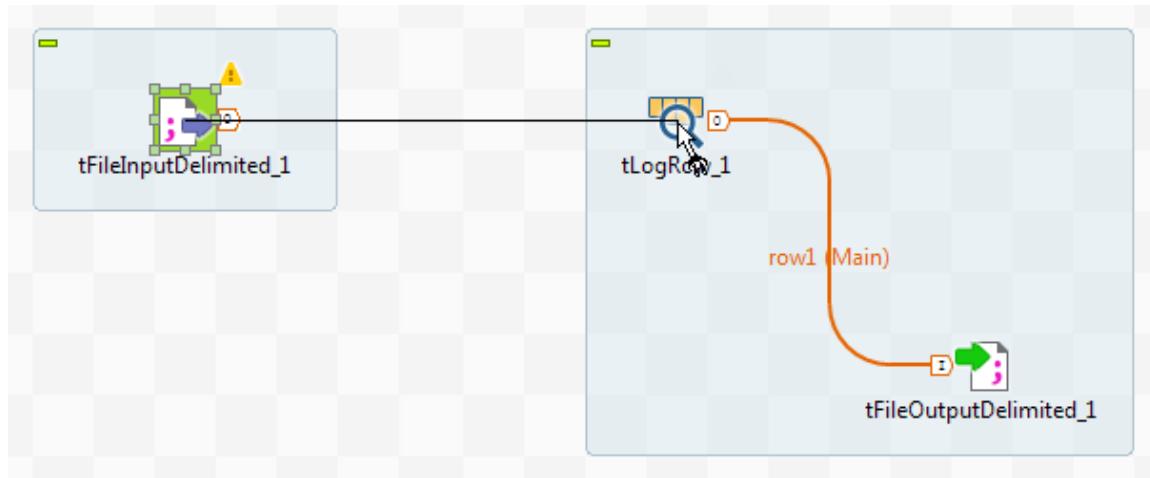


Selon la nature et le rôle des composants que vous souhaitez connecter, différents types de liens sont disponibles. Seules les connexions autorisées sont listées dans le menu contextuel.

Méthode par glisser-déposer

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

Un lien **Row > Main** est automatiquement créé entre ces deux composants.



Cette méthode requiert moins d'opérations mais elle ne fonctionne uniquement avec les connexions de type **Row : Main, Lookup, Output, Filter et Reject**, selon la nature et le rôle des composants que vous connectez.

Vous pouvez également déposer des composants au milieu d'une connexion de type **Row**. Pour plus d'informations, consultez [Ajouter un composant entre deux composants reliés](#).

Pour plus d'informations concernant l'utilisation des différents types de connexions, consultez [Utilisation des connexions dans un Job](#).

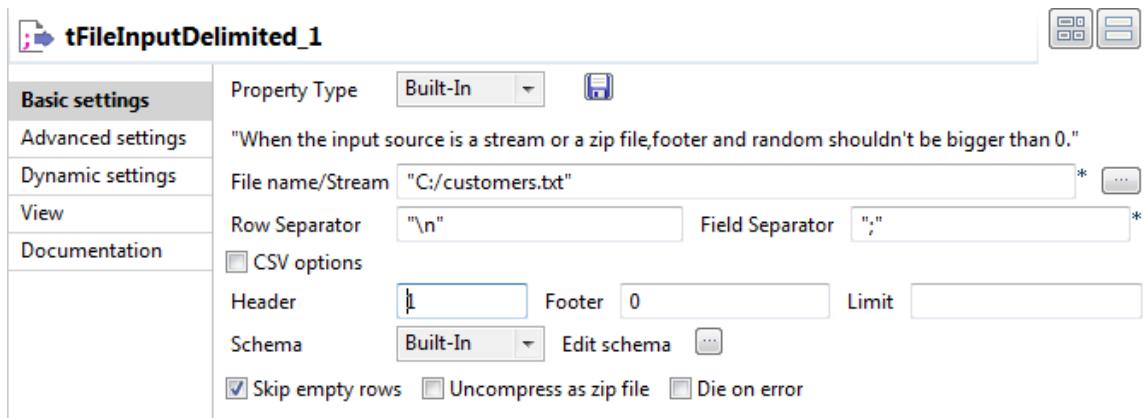
5.2.4. Configurer les composants

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

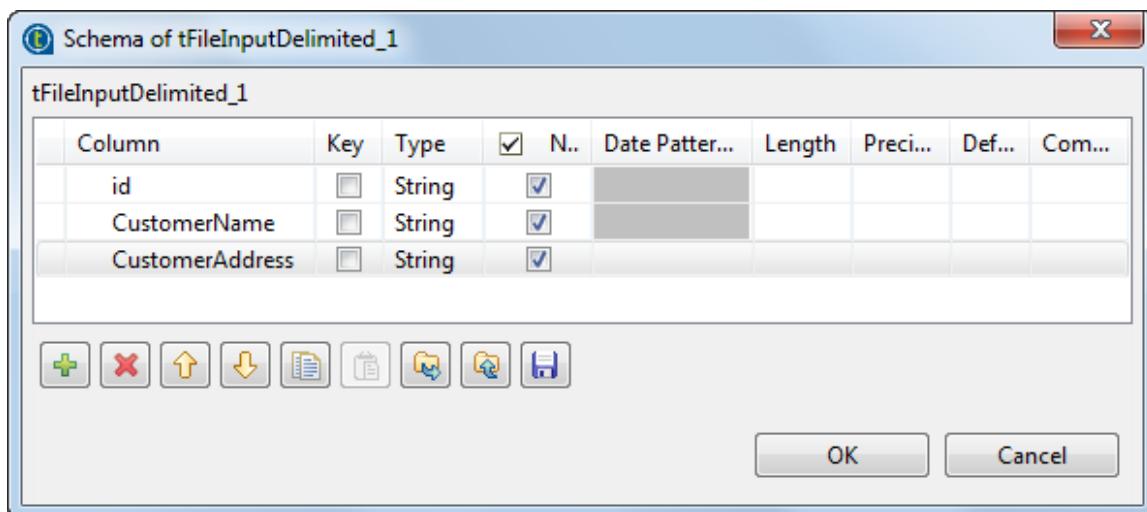
Pour plus de détails concernant les propriétés des composants, consultez [Définir les propriétés des composants](#).

Configurer le composant **tFileInputDelimited**

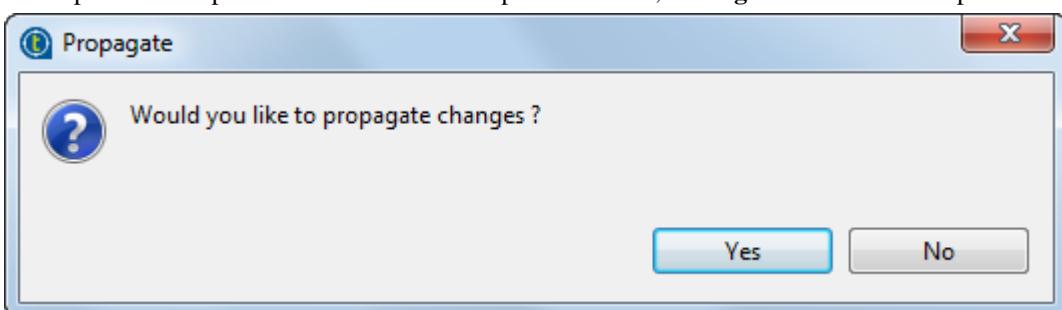
1. Double-cliquez sur le **tFileInputDelimited** afin d'ouvrir sa vue **Basic settings**.



2. Cliquez sur le bouton [...] à côté du champ **File Name/Stream**.
3. Parcourez votre système ou saisissez le chemin vers le fichier d'entrée, *customers.txt* dans cet exemple.
4. Dans le champ **Header**, saisissez *I*.
5. Cliquez sur le bouton [...] à côté du champ **Edit schema**.
6. Dans l'éditeur de schéma qui s'ouvre, cliquez trois fois sur le bouton [+] afin d'ajouter trois colonnes.
7. Nommez ces colonnes *id*, *CustomerName* et *CustomerAddress* respectivement puis cliquez sur **OK** afin de fermer l'éditeur.



8. Dans la boîte de dialogue qui s'ouvre, cliquez sur **OK** afin d'accepter la propagation des changements.
Cela vous permet de copier le schéma créé au composant suivant, le **tLogRow** dans cet exemple.

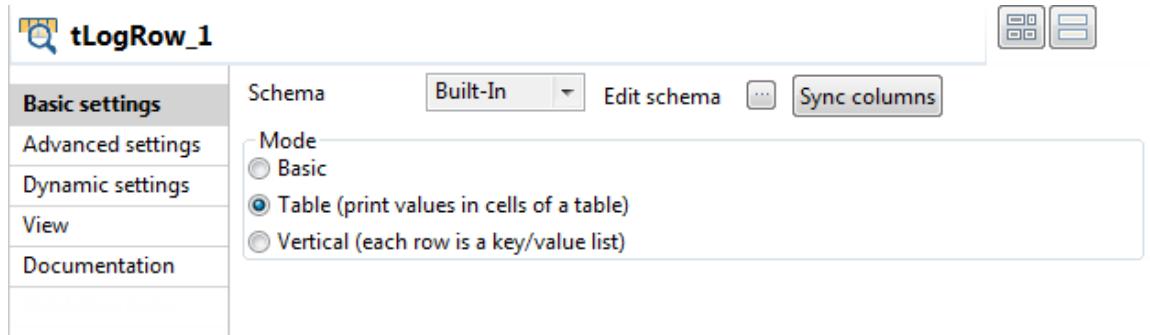


Configurer le composant **tLogRow**

1. Double-cliquez sur le **tLogRow** afin d'ouvrir sa vue **Basic settings**.

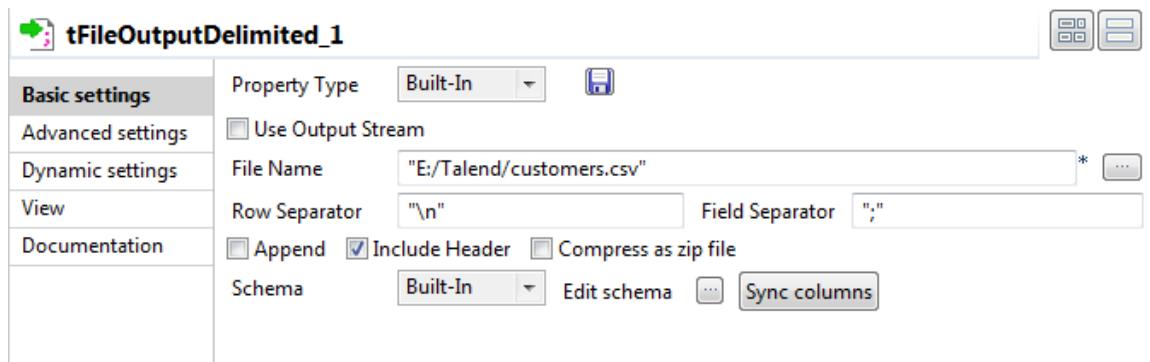
- Dans la zone **Mode**, sélectionnez **Table (print values in cells of a table)**.

Ainsi, le contenu du fichier *customers.txt* est affiché sous forme de tableau afin d'être plus facilement lisible.



Configurer le composant tFileOutputDelimited

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



- Cliquez sur le bouton [...] à côté du champ **File Name**.
- Parcourez votre système ou saisissez le chemin d'accès à votre fichier de sortie, *customers.csv* dans cet exemple.
- Cochez la case **Include Header**.
- Si nécessaire, cliquez sur le bouton **Sync columns** afin de récupérer le schéma du composant d'entrée.

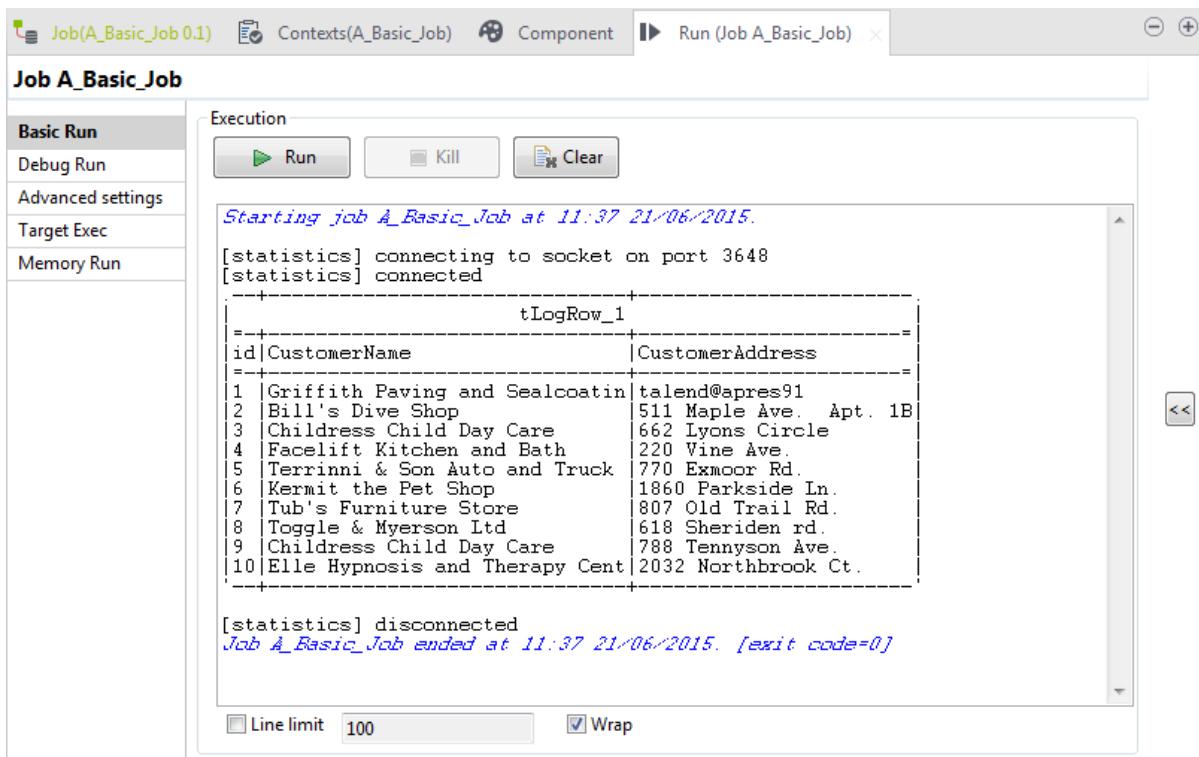
5.2.5. Exécuter le Job

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

Pour ce faire, procédez comme suit :

- Appuyez sur **Ctrl+S** afin de sauvegarder votre Job.
- Ouvrez l'onglet **Run** et cliquez sur **Run** afin d'exécuter le Job.

Le fichier est lu ligne par ligne, les champs extraits sont affichés dans la console **Run** et écrits dans le fichier de sortie spécifié.



5.3. Utilisation des composants

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

- [Ajouter un composant entre deux composants reliés](#)
- [Définir les propriétés des composants](#)
- [???](#)
- [Chercher un Job utilisant un composant particulier](#)
- [Configurer les valeurs par défaut dans le schéma d'un composant dans un Job](#)

5.3.1. Ajouter un composant entre deux composants reliés

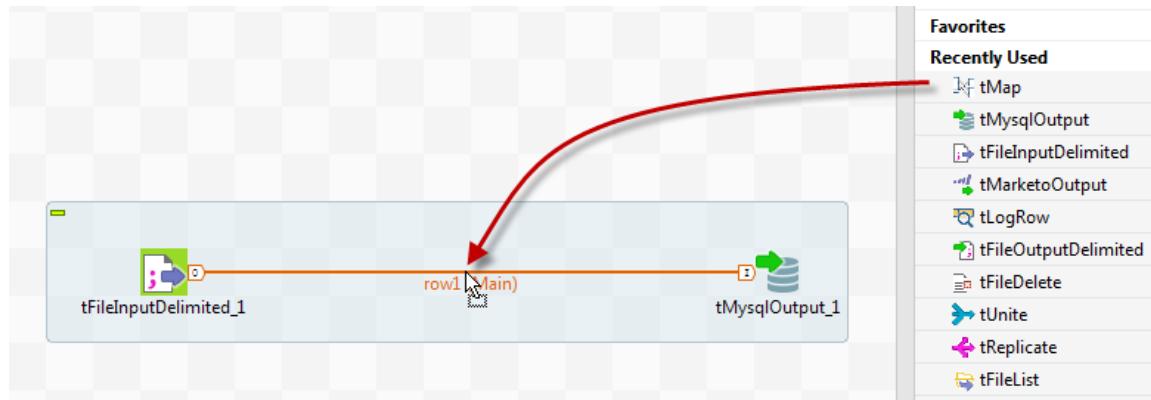
Lorsque vous créez un Job, vous pouvez insérer un composant entre deux composants reliés à l'aide d'un lien **Row**, à condition que le nouveau composant puisse servir de composant intermédiaire entre les deux autres.

Les exemples ci-dessous montrent différentes manières d'insérer un **tMap** entre un **tFileInputDelimited** et un **tMysqlOutput** reliés à l'aide d'un lien **Row > Main**. Pour relier des composants dans un Job, consultez [Connecter les composants](#). Pour plus d'informations sur les différents types de liens, consultez [Types de connexions](#).

Glisser-déposer les composants sur le lien depuis la Palette

1. Depuis la **Palette**, localisez et sélectionnez le **tMap**.

- Glissez-déposez le composant sur le lien **Row**.

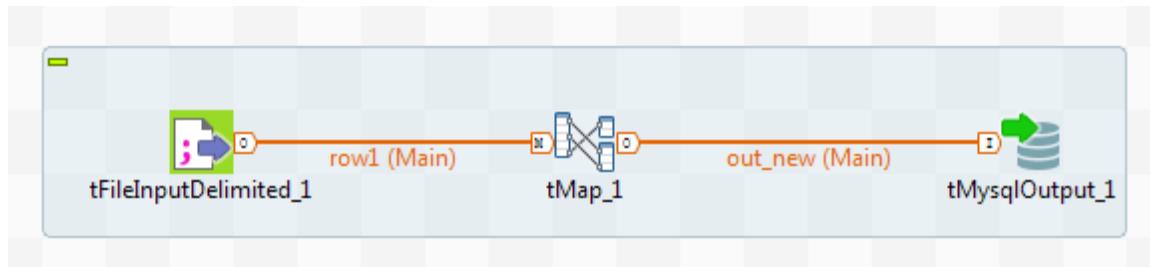


S'il vous est demandé de nommer la connexion de sortie du nouveau composant ajouté, ce qui est le cas avec un composant **tMap**, saisissez un nom et cliquez sur **OK** pour fermer la boîte de dialogue.



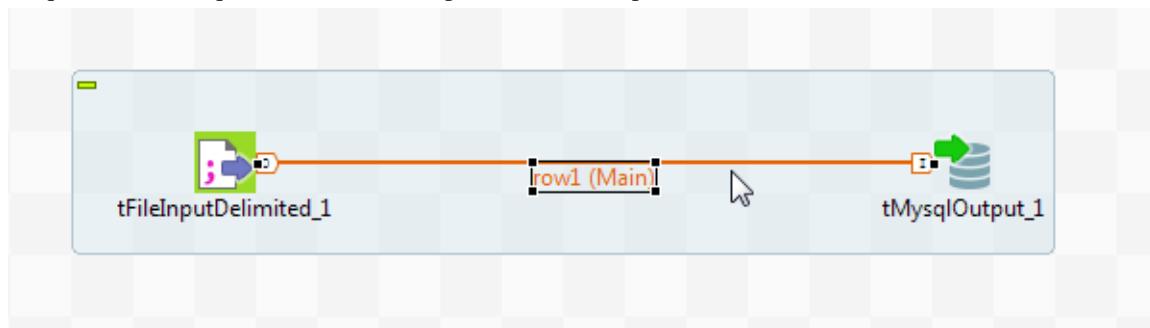
Il vous sera peut-être demandé de récupérer le schéma du composant cible. Dans ce cas, cliquez sur **OK** pour accepter ou cliquez sur **No** pour refuser.

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

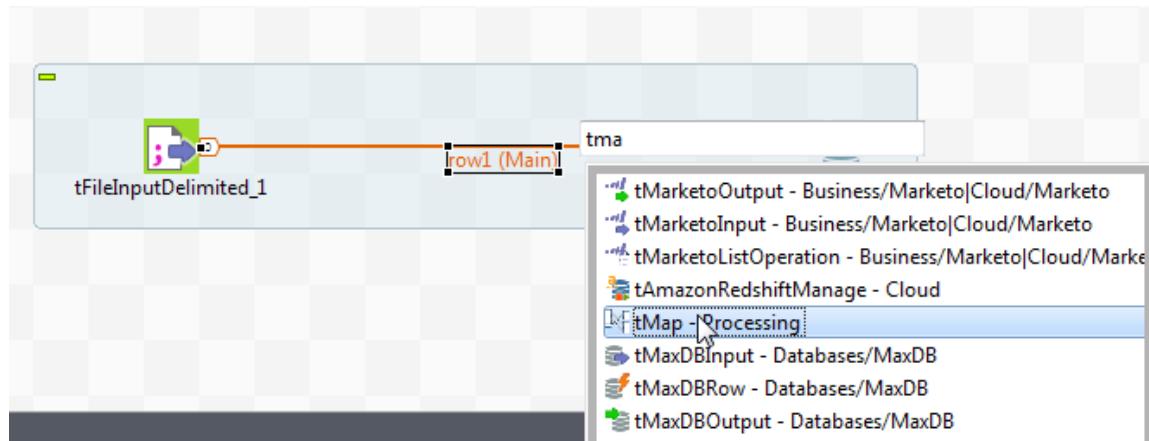


Ajouter le composant en saisissant son nom sur le lien

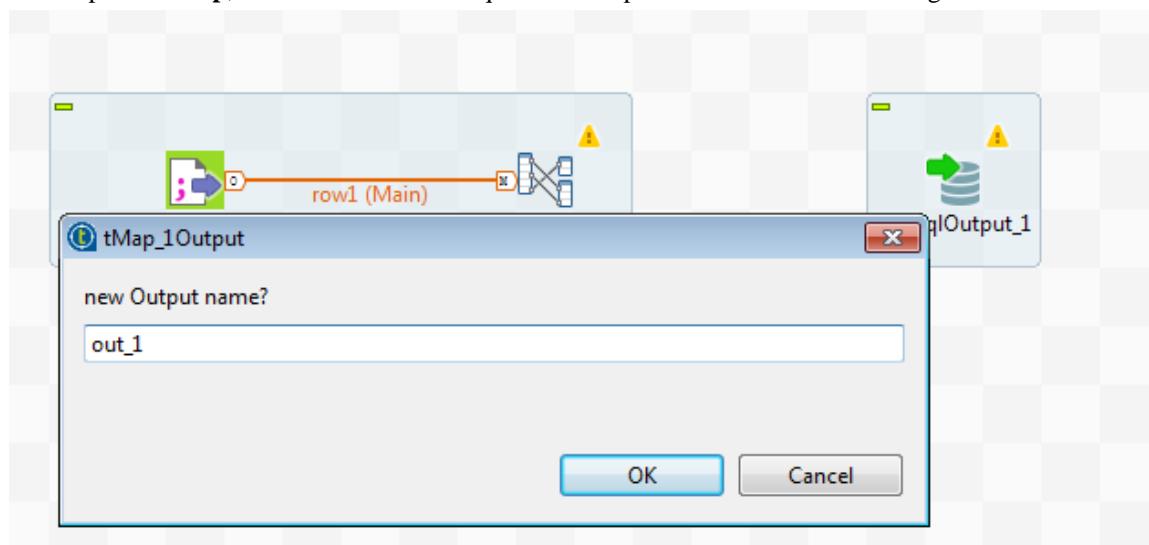
- Cliquez sur le lien qui relie les deux composants existants pour le sélectionner.



- Saisissez le nom du nouveau composant que vous souhaitez ajouter, le **tMap** dans cet exemple, puis double-cliquez sur le composant dans la liste de suggestions pour l'ajouter au lien.

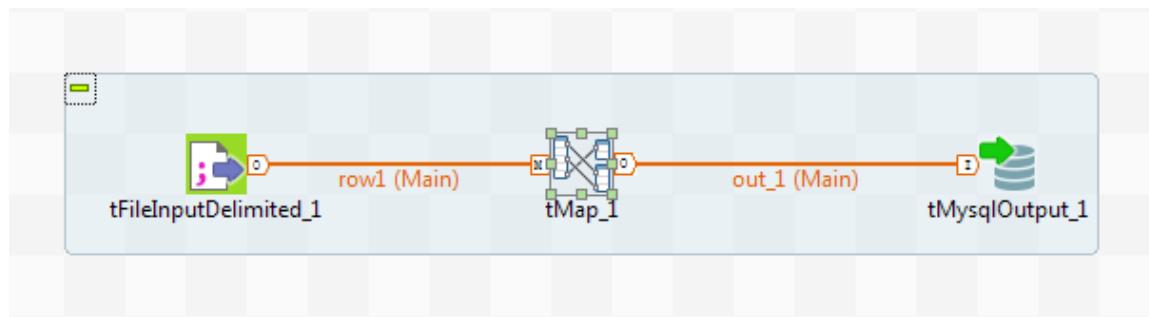


3. S'il vous est demandé de nommer la connexion de sortie du nouveau composant ajouté, ce qui est le cas avec un composant **tMap**, saisissez un nom et cliquez sur **OK** pour fermer la boîte de dialogue.



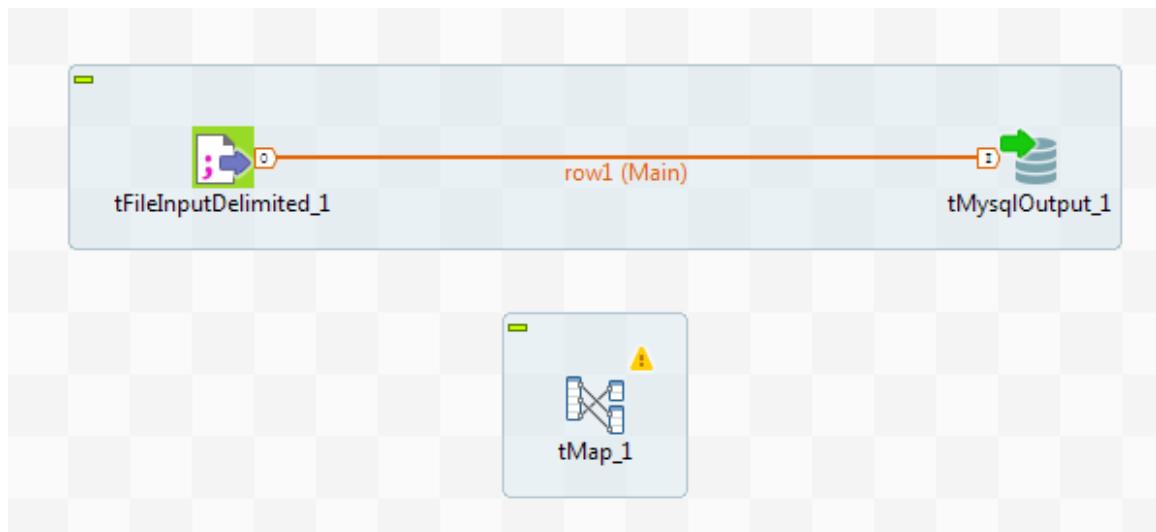
Il vous sera peut-être demandé de récupérer le schéma du composant cible. Dans ce cas, cliquez sur **OK** pour accepter ou cliquez sur **No** pour refuser.

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

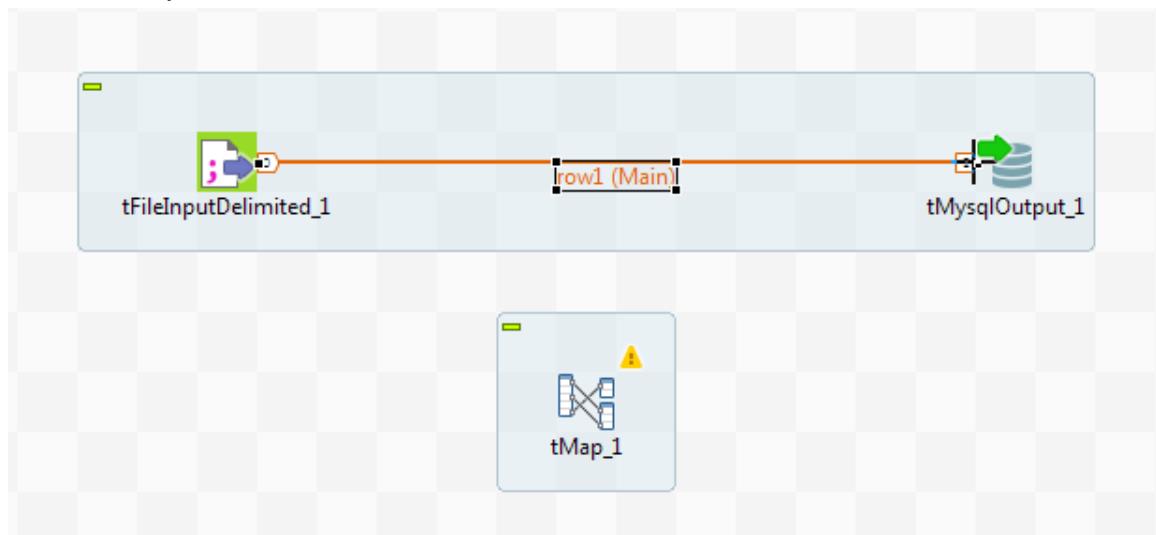


Ajouter le composant dans l'espace de modélisation graphique et déplacer le lien existant

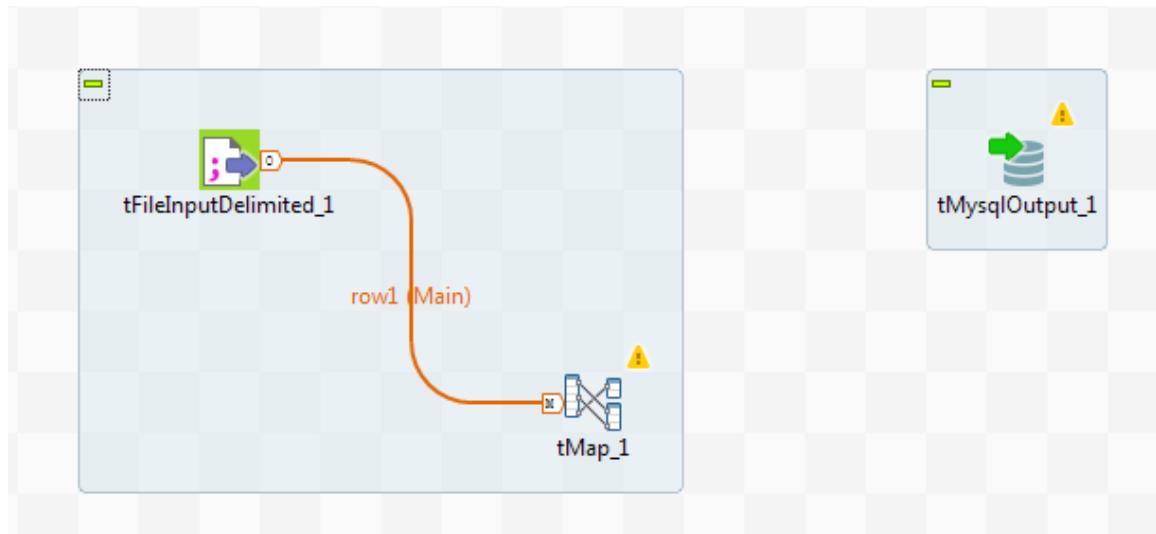
1. Ajoutez le nouveau composant, le **tMap** dans cet exemple, dans l'espace de modélisation graphique en faisant un glisser-déposer depuis la **Palette** ou en saisissant son nom dans l'espace de modélisation graphique.



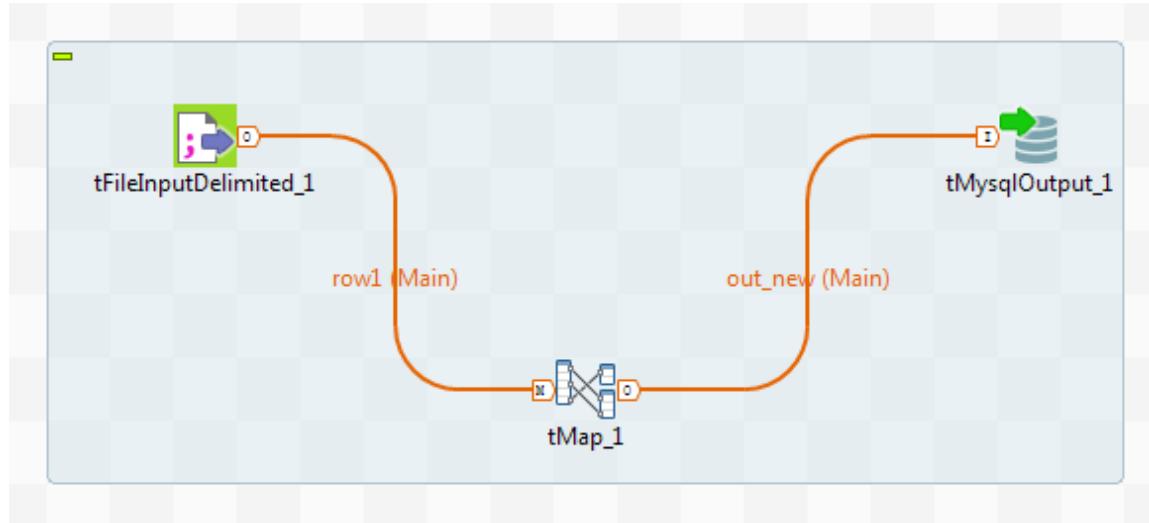
2. Sélectionnez le lien et déplacez le curseur de votre souris vers la fin du lien jusqu'à ce que celui-ci prenne la forme d'un symbole +.



3. Glissez-déposez le lien du composant **tMysqlOutput** au composant **tMap**.



4. Reliez le composant **tMap** au **tMysqlOutput** à l'aide d'un lien **Row > Main**.



5.3.2. Définir les propriétés des composants

Les propriétés de chacun des composants dans un 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.

5.3.2.1. Onglet Basic settings

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

Basic settings	Property Type	Built-In
Advanced settings	"When the input source is a stream or a zip file, footer and random shouldn't be bigger than 0."	
Dynamic settings	File name/Stream: "C:/customer.csv"	
View	Row Separator	"\n"
Documentation	Field Separator	";"
Validation Rules	Header	6
	Footer	0
	Schema	Built-In
	<input checked="" type="checkbox"/> Skip empty rows <input type="checkbox"/> Uncompress as zip file <input type="checkbox"/> Die on error	

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

Certains composants sont paramétrés à l'aide de code ou de fonctions. Veillez à bien utiliser le code Java dans les propriétés en Java.

Pour les composants **File** (Fichier) et **Database** (Base de données) dans un Job, vous pouvez centraliser leurs propriétés dans les métadonnées stockées dans le dossier **Metadata** du **Repository**. Dans l'onglet **Basic settings**,

vous pouvez paramétrer vous même les propriétés en sélectionnant l'option **Built-In** dans la liste déroulante **Property Type** ou vous pouvez utiliser les propriétés stockées dans le dossier **Metadata** 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 : [Gestion des métadonnées dans l'intégration de données](#).

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 :

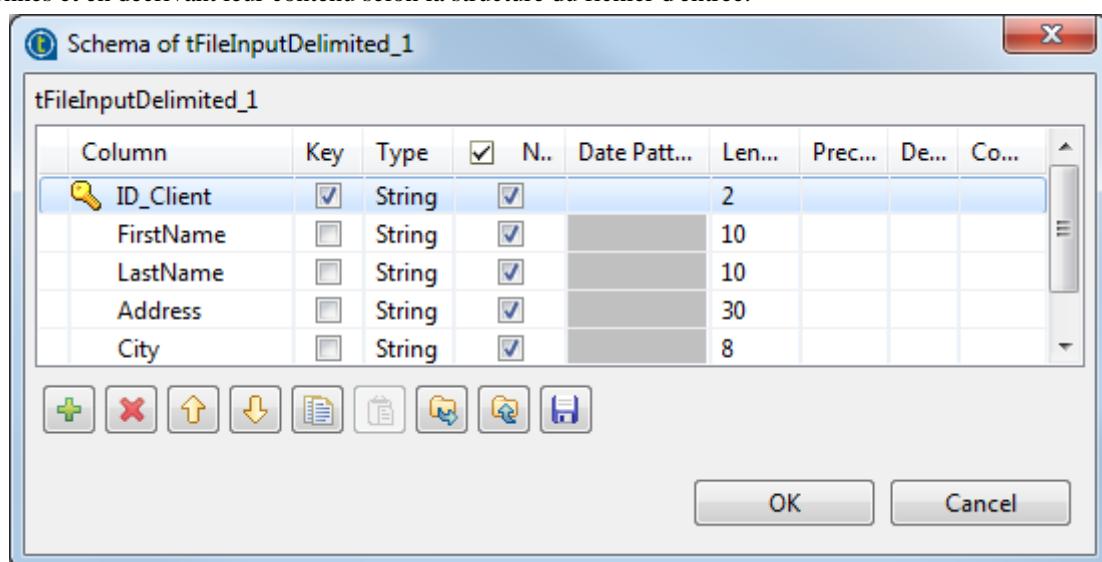
1. Cliquez sur l'icône de disquette. L'assistant de création de métadonnée correspondant à votre composant s'ouvre.
2. Suivez les étapes de l'assistant. Pour plus d'informations sur la création de métadonnées, consultez [Gestion des métadonnées dans l'intégration de données](#).
3. La métadonnée apparaît 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 dans un Job

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

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.



Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

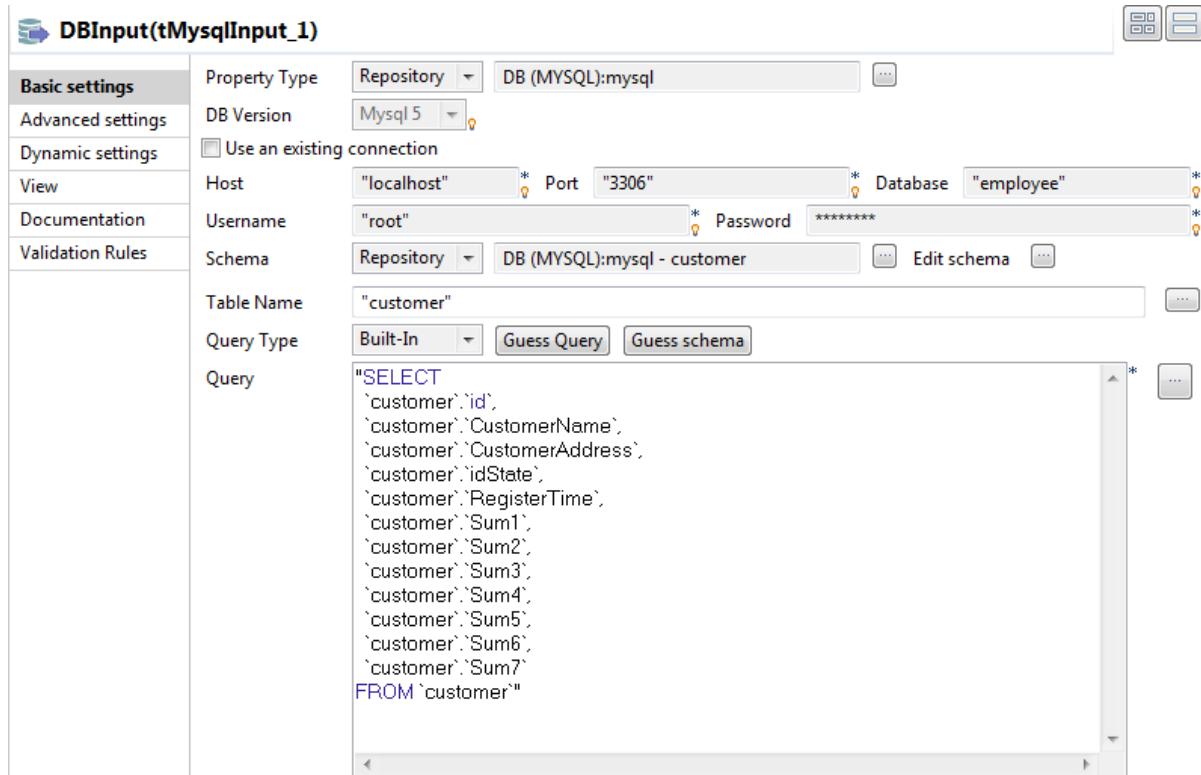
Vous pouvez récupérer le schéma ponctuel d'un fichier d'entrée pour alimenter le schéma de sortie. Pour ce faire, cliquez sur le bouton **Sync columns** dans les propriétés du composant Output.

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

Paramétriser un schéma du Repository dans un Job

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

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 pouvez également utiliser partiellement un schéma du référentiel. Pour plus d'informations, consultez [Utiliser partiellement un schéma du Repository dans un Job](#).



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

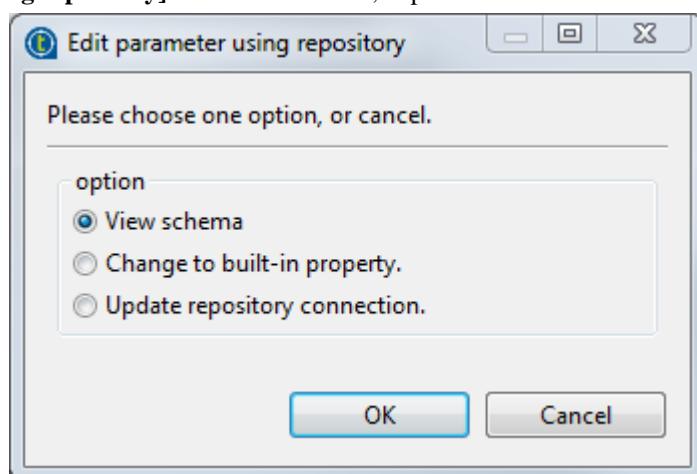
Consultez également [Gestion des métadonnées dans l'intégration de données](#).

Utiliser partiellement un schéma du Repository dans un Job

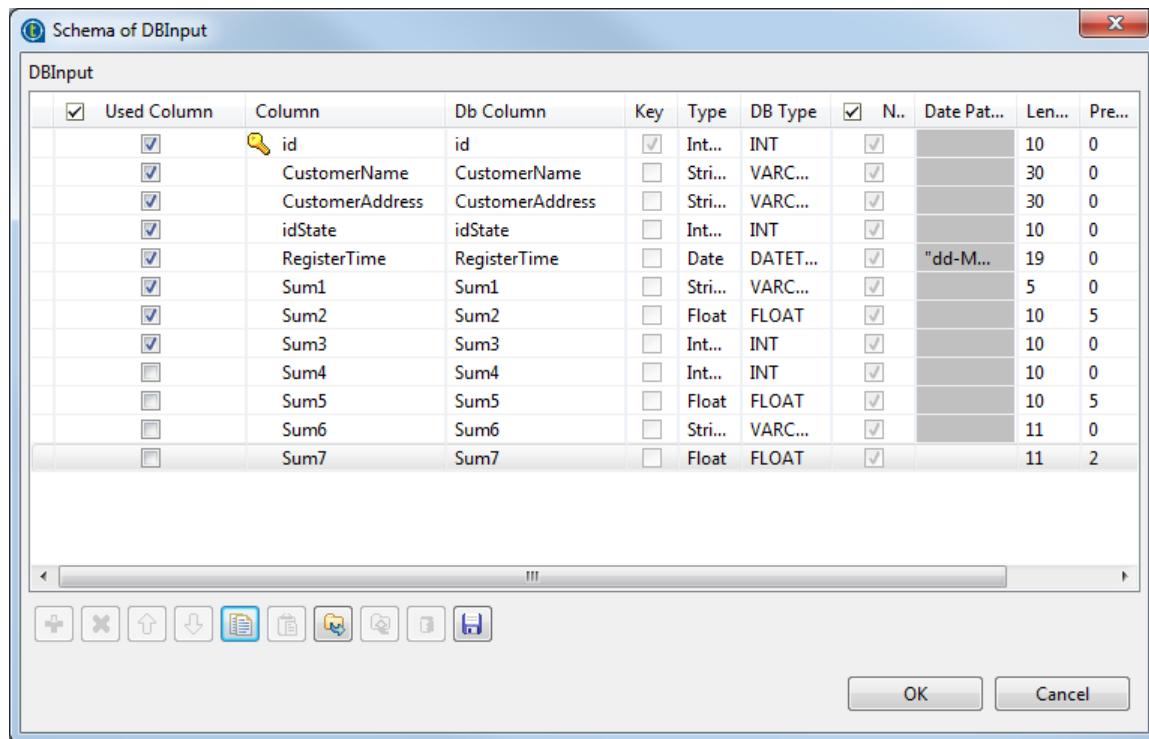
Lorsque vous utilisez un schéma du référentiel, si vous ne souhaitez pas utiliser toutes les colonnes prédefinies, vous pouvez sélectionner des colonnes précises, sans modifier le schéma et le passer en mode Built-In.

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

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



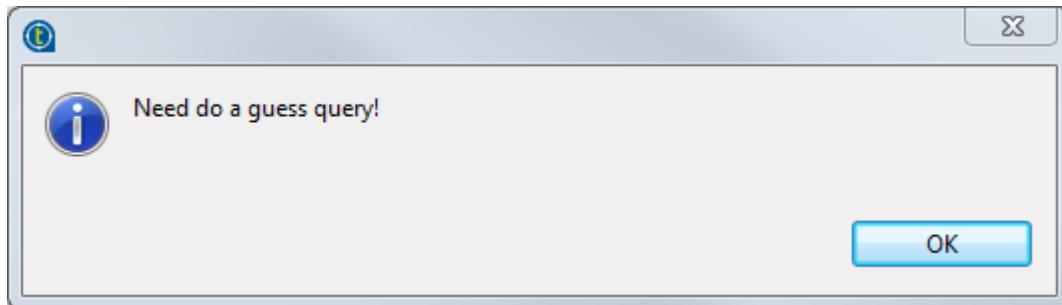
2. Cliquez sur **OK**. La boîte de dialogue **[Schema]** s'ouvre et affiche toutes les colonnes du schéma. La case **Used Column** devant chaque colonne indique si la colonne est utilisée.
3. Sélectionnez les colonnes à utiliser.



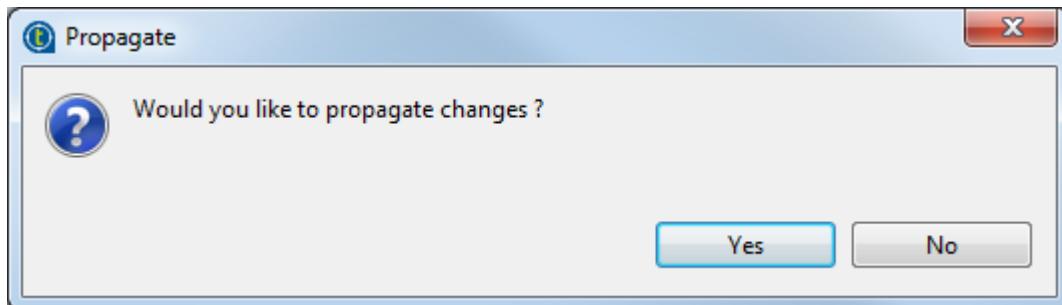
- Cliquez sur **OK**. Un message s'ouvre et vous demande de cliquer sur le bouton **Guess Query**.



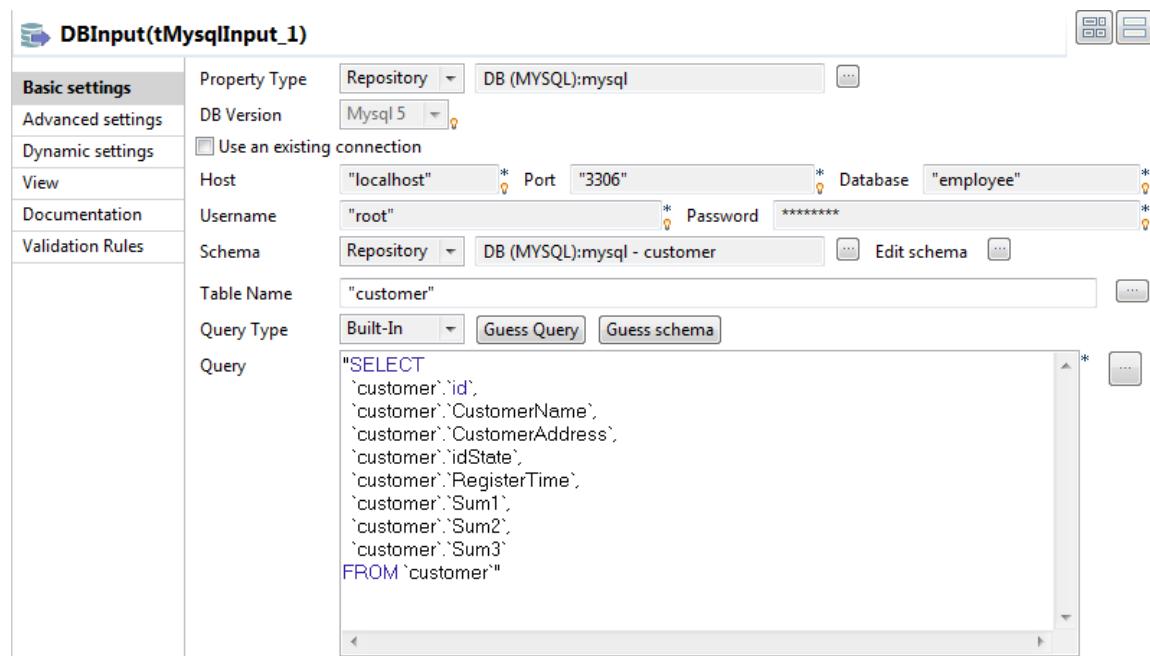
L'opération **Guess Query** est nécessaire uniquement pour les métadonnées de base de données.



- Cliquez sur **OK** pour fermer le message. La fenêtre **[Propagate]** apparaît. Cliquez sur **Yes** pour propager les modifications.



- Dans la zone **Basic settings**, cliquez sur **Guess Query**. Le nom des colonnes sélectionnées s'affiche dans la zone **Query** comme attendu.

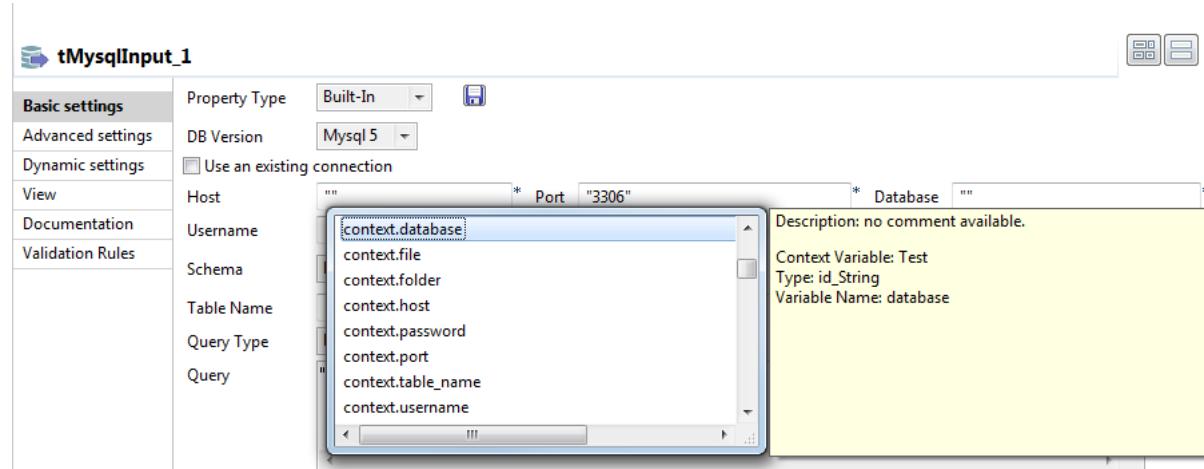


Pour plus d'informations concernant la définition d'un schéma dans le référentiel, consultez [Paramétriser un schéma du Repository dans un Job](#).

Afficher un champ de manière dynamique (Ctrl+Espace)

Dans tous les champs de propriétés de vos composants, vous pouvez accéder à la liste des variables globales et de contexte et afficher de manière dynamique les valeurs contenues dans chaque champ à l'aide du raccourci clavier **Ctrl+Barre d'espace**.

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.

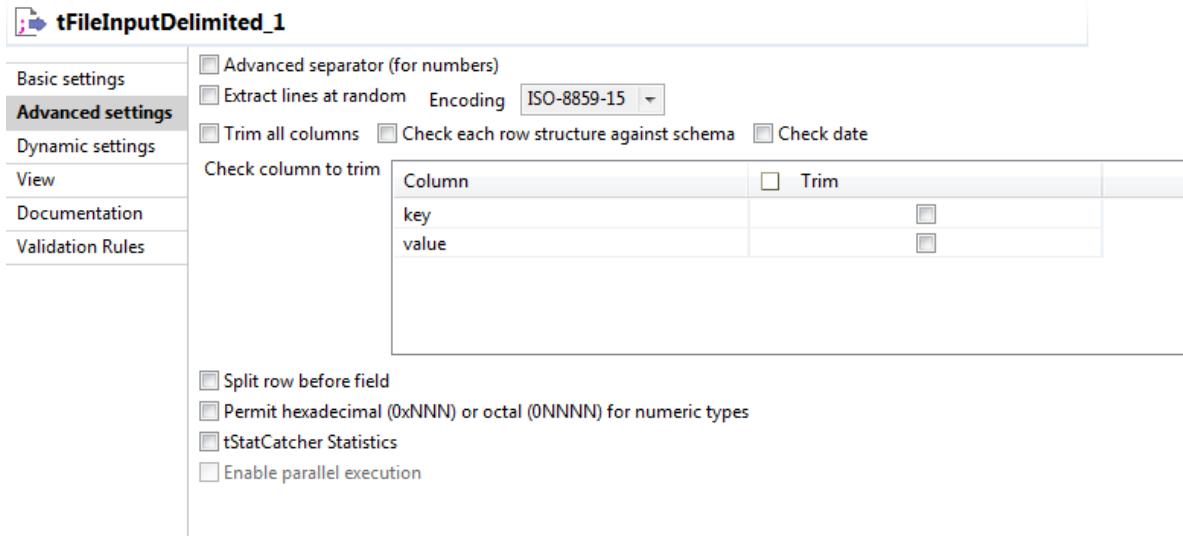


Les paramètres affichés peuvent être : des messages d'erreur, le nombre de lignes traitées, etc. La liste varie en fonction des composants sélectionnés ou du contexte sur lequel vous travaillez.

Voir également : [Utiliser les contextes et les variables](#).

5.3.2.2. Onglet Advanced settings

Certains composants, notamment les composants **File** et **Database** dans les Jobs, offrent de nombreuses fonctions avancées.



Le contenu de l'onglet **Advanced settings** change en fonction du composant sélectionné.

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

Mesurer les flux de données

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

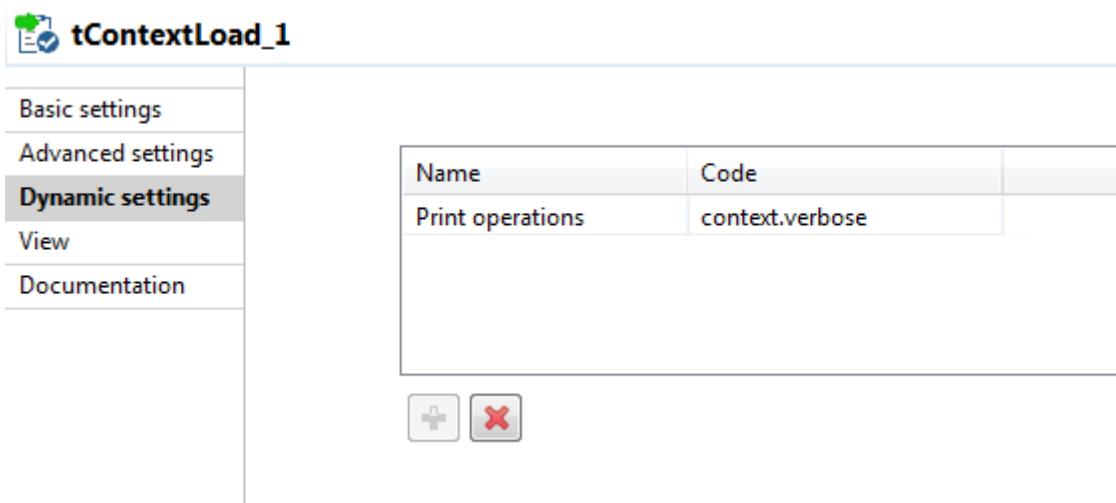
5.3.2.3. Onglet Dynamic settings des composants dans un Job

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

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

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

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



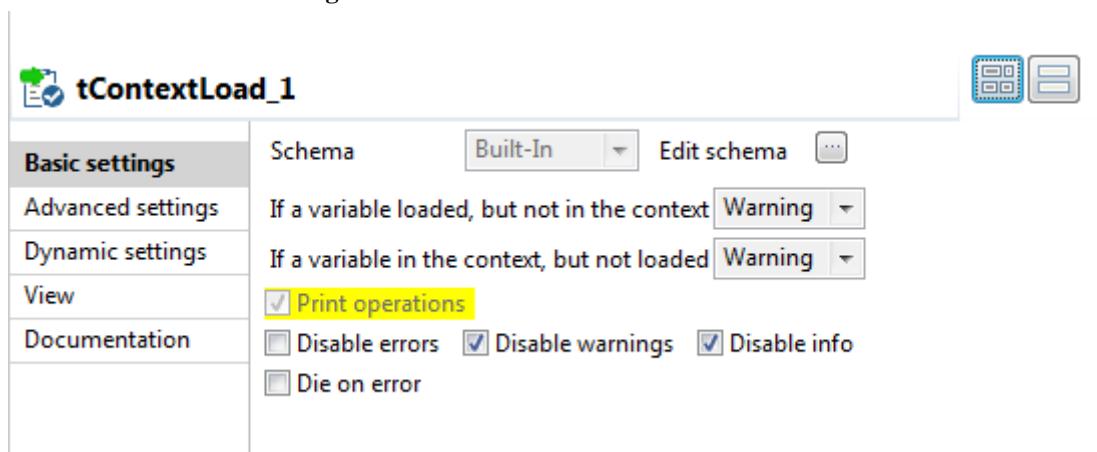
Pour personnaliser ces types de paramètres, notamment les variables de contexte, suivez les étapes qui suivent :

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 **Contexts**. Pour plus d'informations concernant la définition des variables de contexte, consultez [Définir les variables de contexte dans la vue Contexts](#).

Pour comprendre comment définir un paramètre dynamique, consultez le scénario *Lire des données dans des bases de données MySQL à l'aide de connexions dynamiques basées sur les variables de contexte* du composant **tMysqlInput** ainsi que le scénario du **tContextLoad** à l'adresse <https://help.talend.com>.

5.3.2.4. 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.

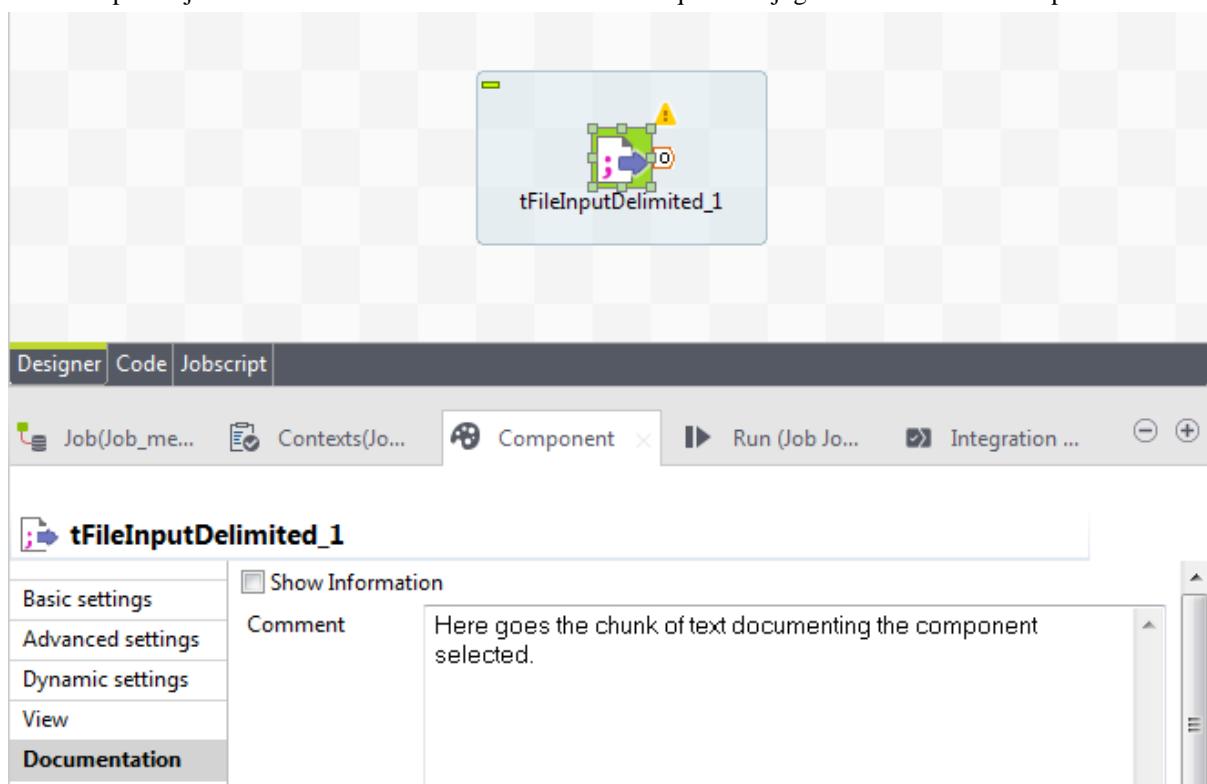
Vous pouvez personnaliser les textes des **Label** et **Hint** à l'aide des balises HTML suivantes :

- 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**.

5.3.2.5. 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.

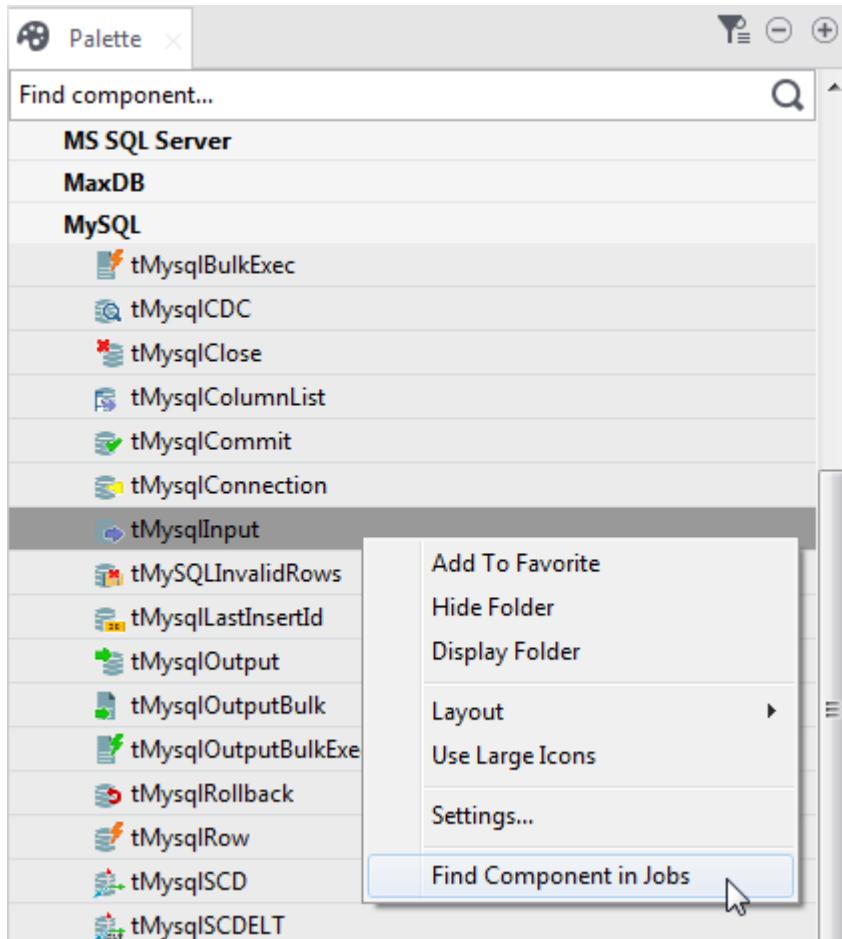
5.3.3. Chercher un Job utilisant un composant particulier



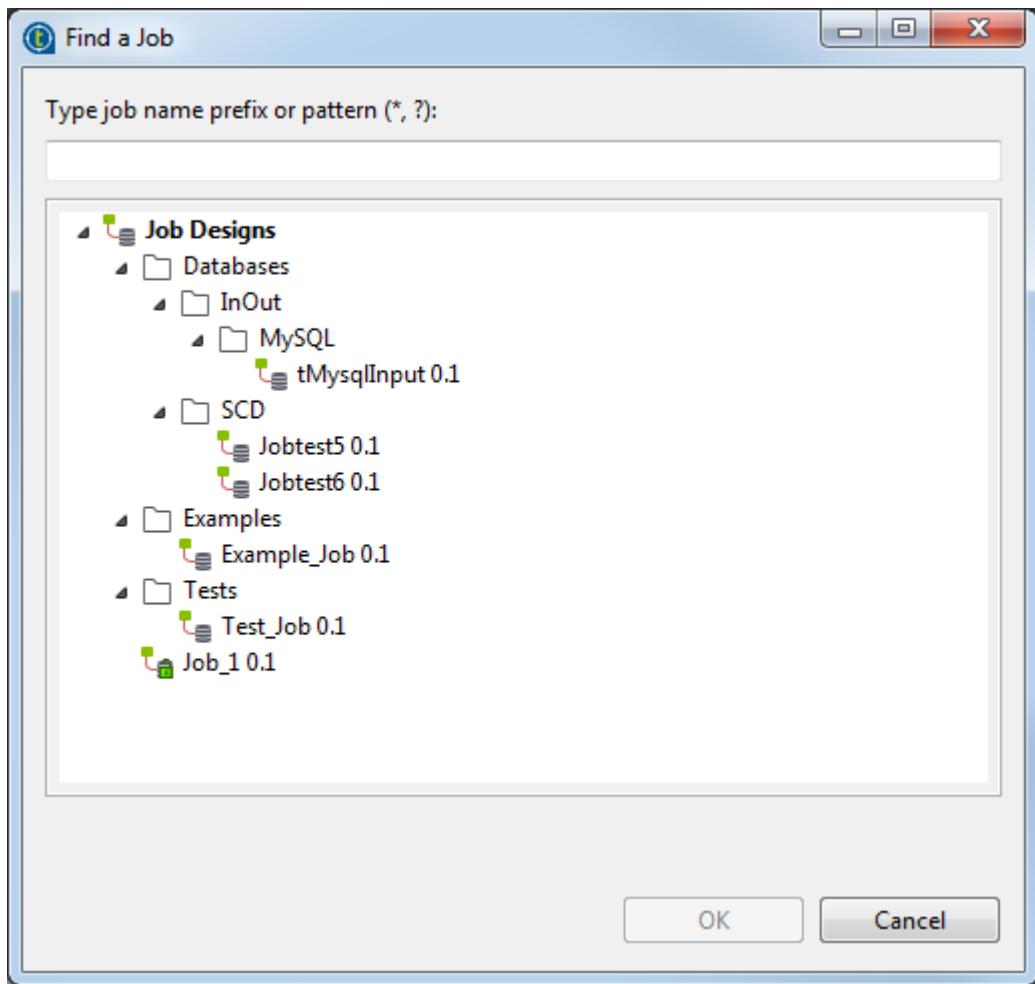
Vous devez ouvrir au moins un Job dans votre Studio afin d'afficher la **Palette** à droite de l'espace de modélisation graphique et lancer la recherche.

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

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



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



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

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

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

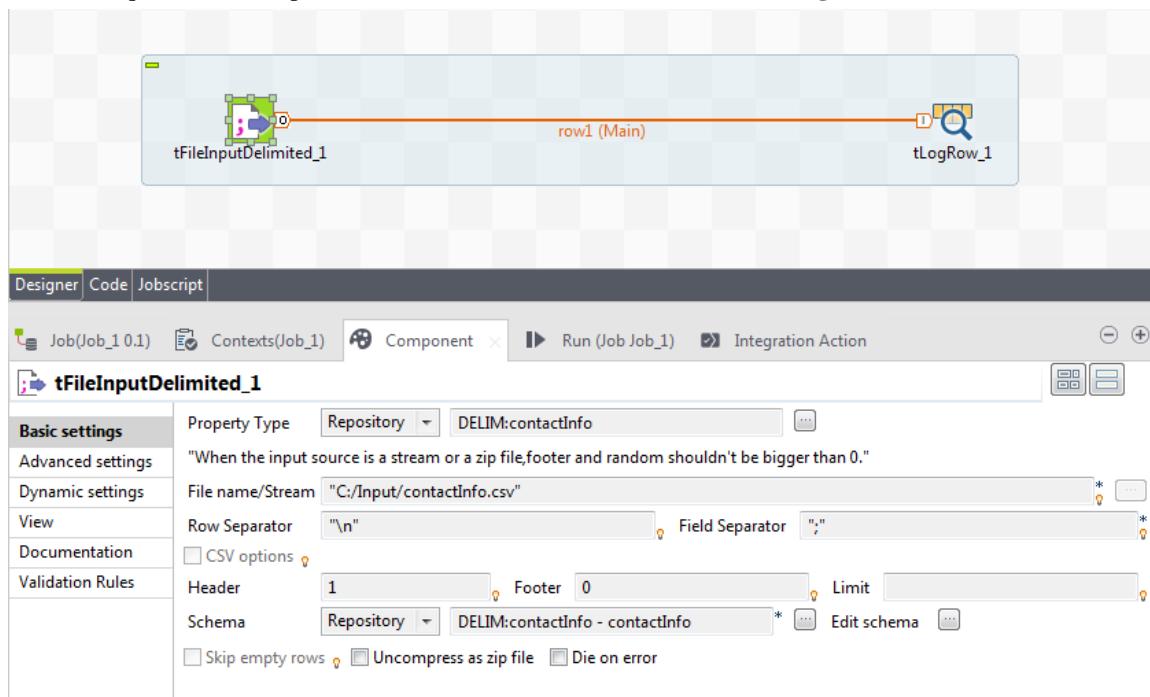
 Pour le moment, seuls les **tFileInputDelimited**, **tFileInputExcel** et **tFixedFlowInput** supportent les valeurs par défaut dans le schéma.

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
```

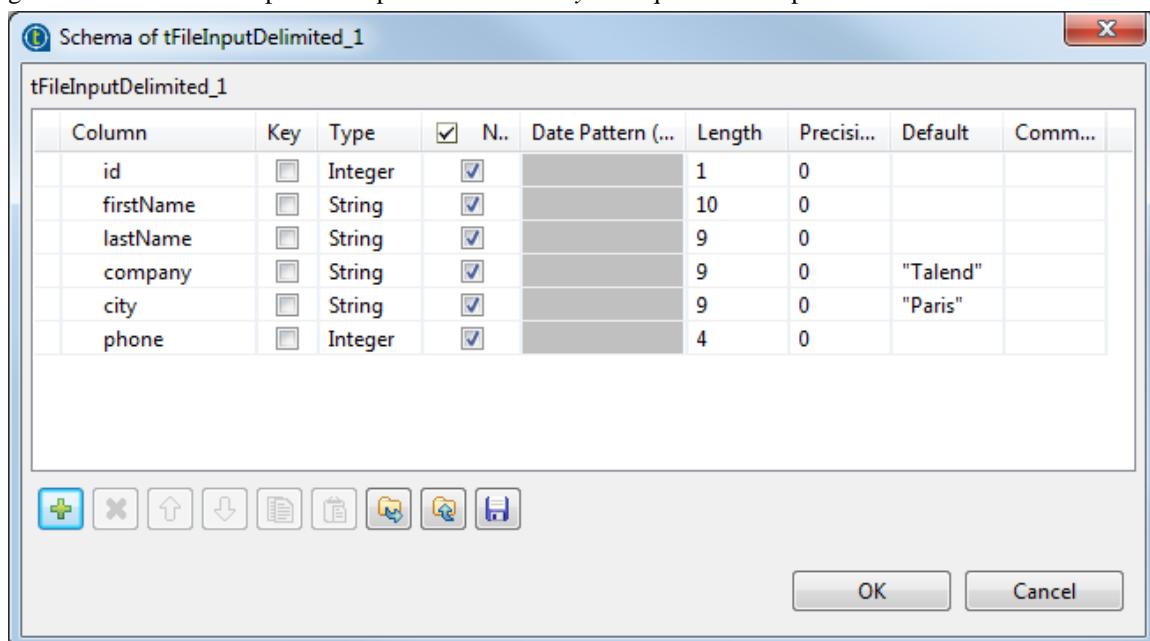
Pour configurer des valeurs par défaut :

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



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 [Gestion des métadonnées dans l'intégration de données](#).

- 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.



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

```

Starting job Job_1 at 15:57 09/06/2015.

[statistics] connecting to socket on port 3454
[statistics] connected
---+-----+-----+-----+-----+
      tLogRow_1
---+-----+-----+-----+-----+
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 Job_1 ended at 15:57 19/08/2015. [exit code=0]

```

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

5.3.5. Utiliser des composants Camel dans une Route

De nombreux composants Camel sont supportés mais non inclus directement dans la **Palette**, comme camel-mina, camel-http4, par exemple. Ces composants Camel requièrent le **cMessagingEndpoint** pour fonctionner et leurs bibliothèques doivent être ajoutées à la Route. Toutes les versions des composants Camel publiées avant le *Studio Talend* sont supportées.

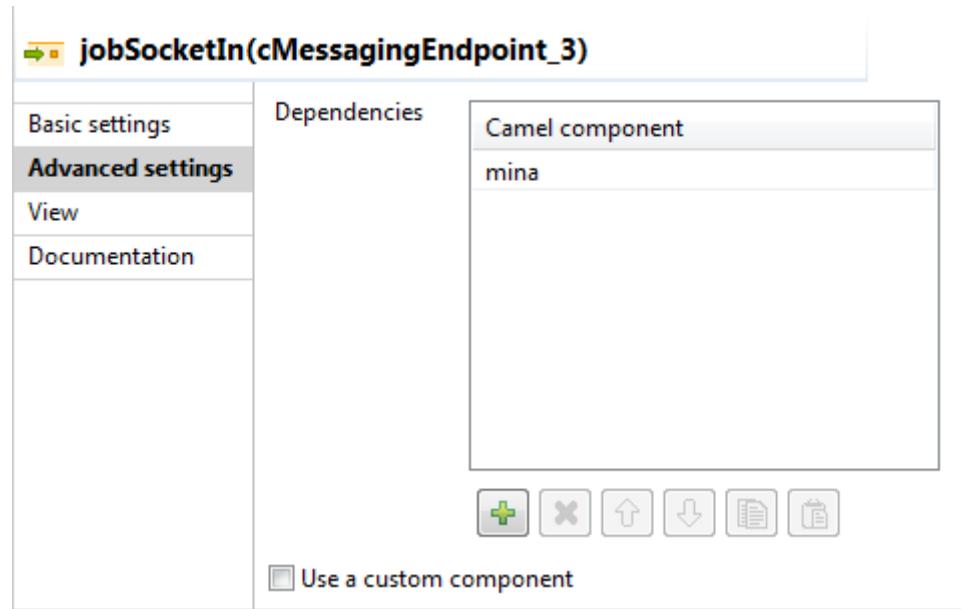


Le processus d'utilisation d'un composant Camel externe dans une Route est différent de l'utilisation d'un composant, externe ou personnalisé, dans un Job. Pour plus d'informations concernant l'utilisation de composants externes ou utilisateur dans un Job, consultez [Définir le dossier de composants utilisateur \(Talend > Components\)](#).

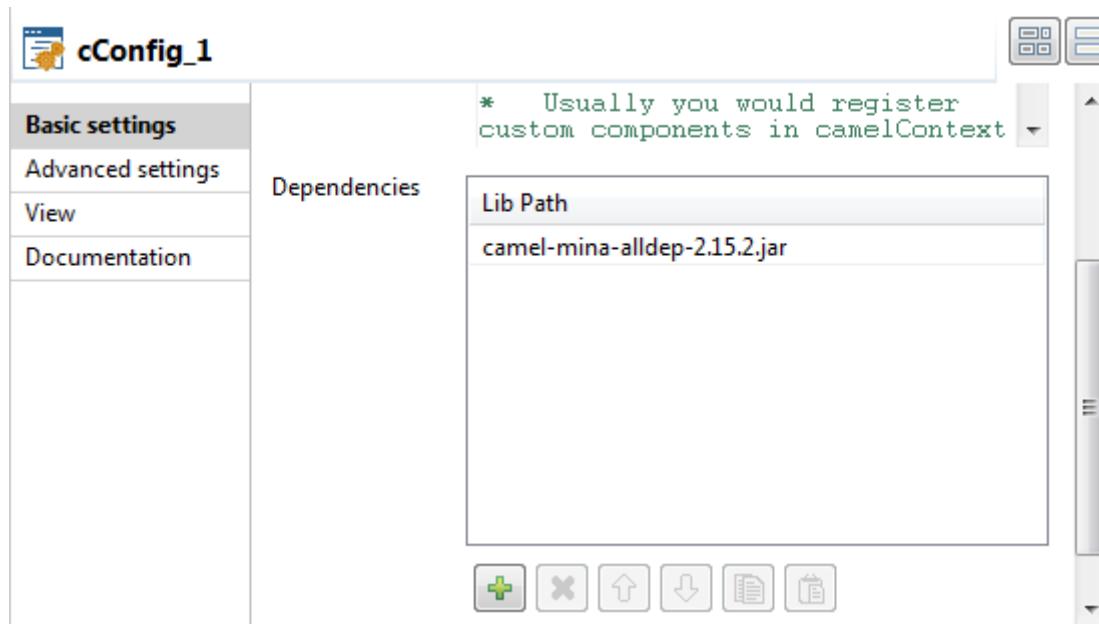
La capture d'écran suivante illustre l'utilisation du composant camel-mina.



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



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



Pour un exemple d'utilisation réelle des composants Camel, consultez *Création d'une Route communiquant avec un Job d'intégration* sur <https://help.talend.com>.

5.4. Utilisation des connexions dans un Job

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

5.4.1. 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 toutes les connexions disponibles pour le composant sélectionné.

Les sections suivantes décrivent tous les types de connexions disponibles.

5.4.1.1. Connexion de type Row

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

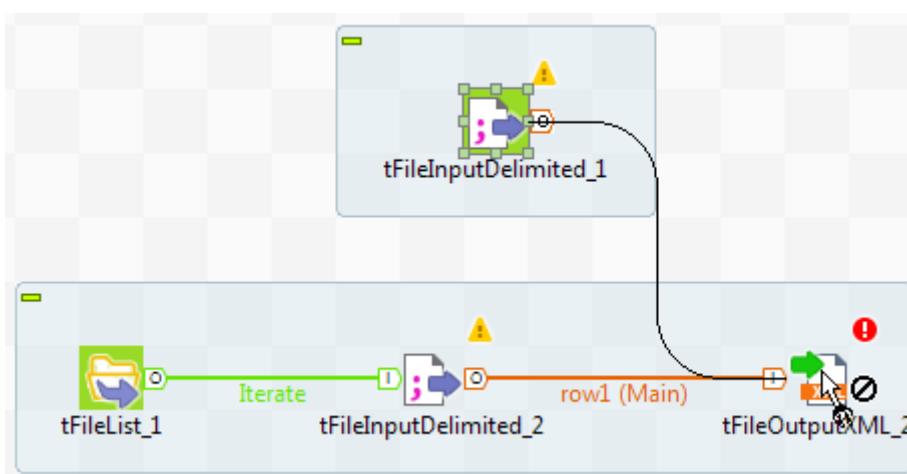
Main

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 **Row > Main**. Une seule connexion entrante de type **Row** est possible par composant. La seconde connexion **Row** se transforme en connexion **Lookup** (de référence) automatiquement.



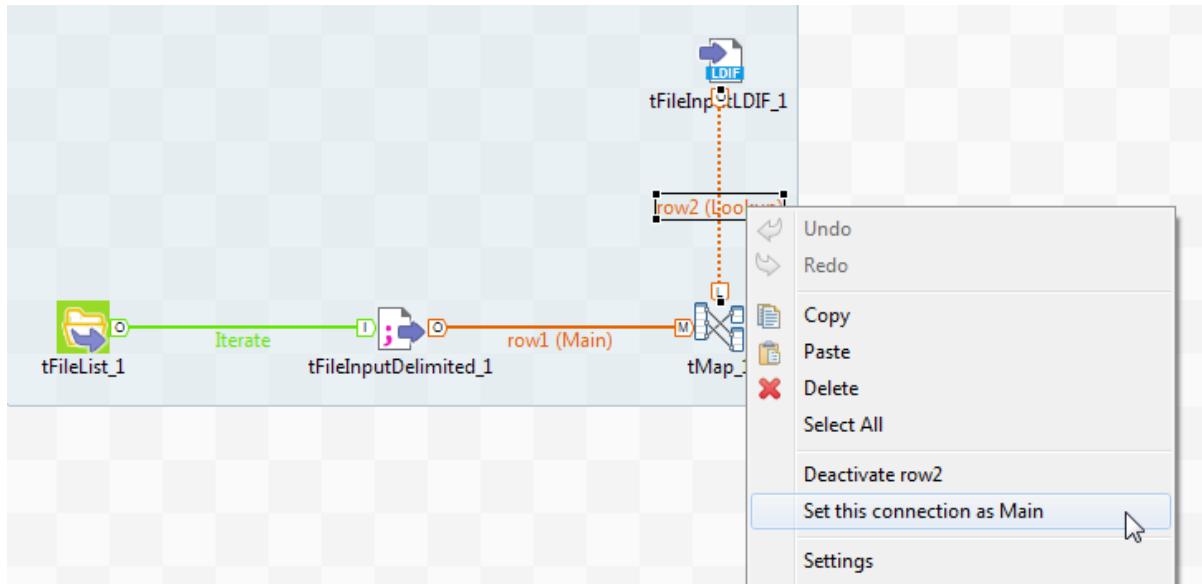
Cliquez-droit sur le composant d'entrée et sélectionnez **Row > Main** dans la liste des connexions.

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

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

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**.

Consultez également : [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, cette connexion est activée lorsque l'option **Die on error** est désactivée.

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 : [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.

Consultez également : [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 [Mapping de flux de données](#).

Combine

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

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

5.4.1.2. 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'une seule connexion 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.

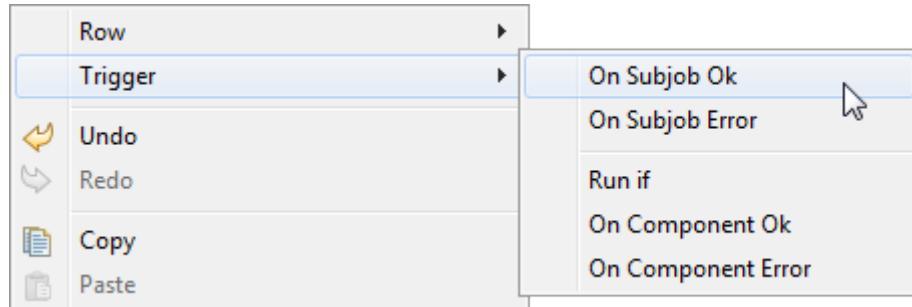
Certains composants ne peuvent fonctionner avec d'autres composants qu'à l'aide d'une connexion d'itération, c'est le cas du composant **tFilelist**. Pour plus d'informations concernant la configuration d'une connexion **Iterate**, consultez [Paramètres d'une connexion Iterate](#).

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

5.4.1.3. Connexion de type Trigger

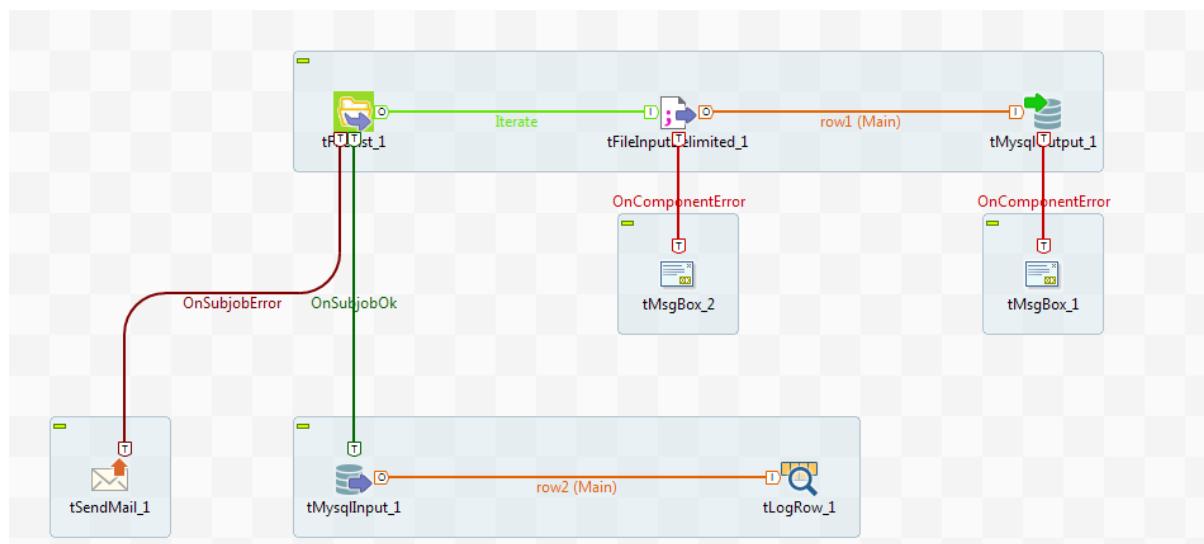
Les connexions de déclenchement, **Trigger**, aident à définir le processus de traitement. Les connexions de type **Trigger** ne servent pas au transfert de données.

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

Consultez également : [???](#).

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'information concernant la configuratoin d'une connexion **Run if**, consultez [Paramètres d'une connexion Run if](#).

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

5.4.1.4. Connexion de type Link

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

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

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

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

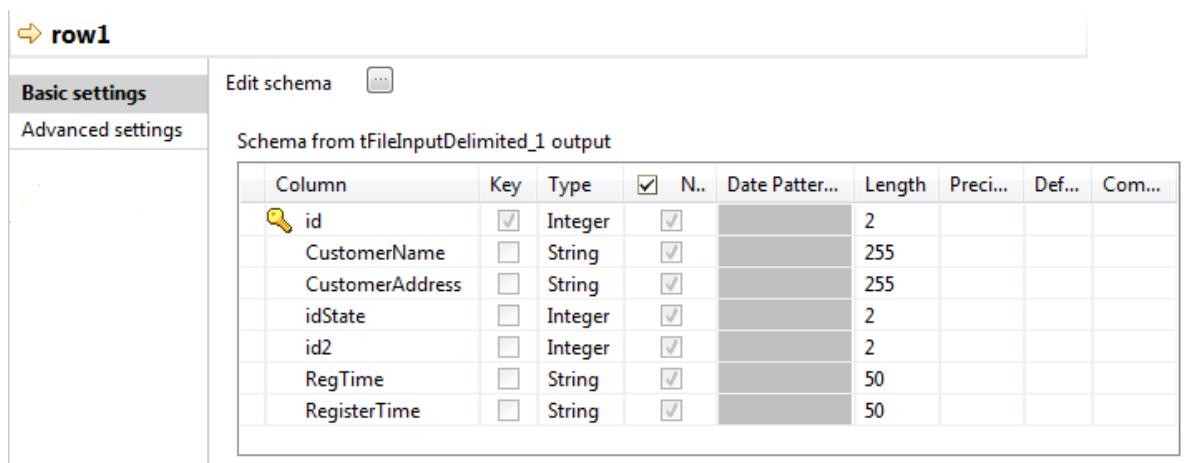
En effet, le nom de la connexion étant utilisé dans la formulation de la requête SQL générée par le composant ELT **tMap**, un même nom ne devrait jamais être utilisé deux fois.

5.4.2. 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.

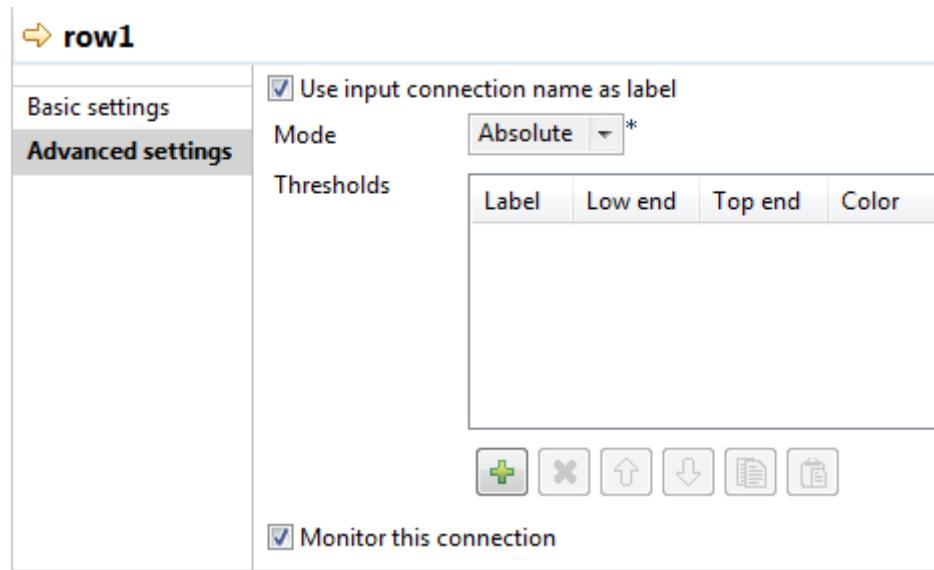
5.4.2.1. Paramètres d'une connexion Row

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**. Pour plus d'informations, consultez [Paramétriser un schéma built-in dans un Job](#).



Column	Key	Type	N..	Date Patter...	Length	Preci...	Def...	Com...
id	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2			
CustomerName	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255			
CustomerAddress	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255			
idState	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2			
id2	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2			
RegTime	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50			
RegisterTime	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50			

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 *Talend Activity Monitoring Console* (disponible avec les produits **Talend** en souscription).



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

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

Consultez également la documentation du **tFlowMeterCatcher** à l'adresse <https://help.talend.com>.

5.4.2.2. Paramètres d'une connexion **Iterate**

Lorsque vous configurez une connexion **Iterate**, vous activez les itérations parallèles. Pour plus d'informations, consultez [Lancer des itérations parallèles pour lire des données](#).

5.4.2.3. Paramètres d'une connexion **Trigger**

Paramètres d'une connexion **Run if**

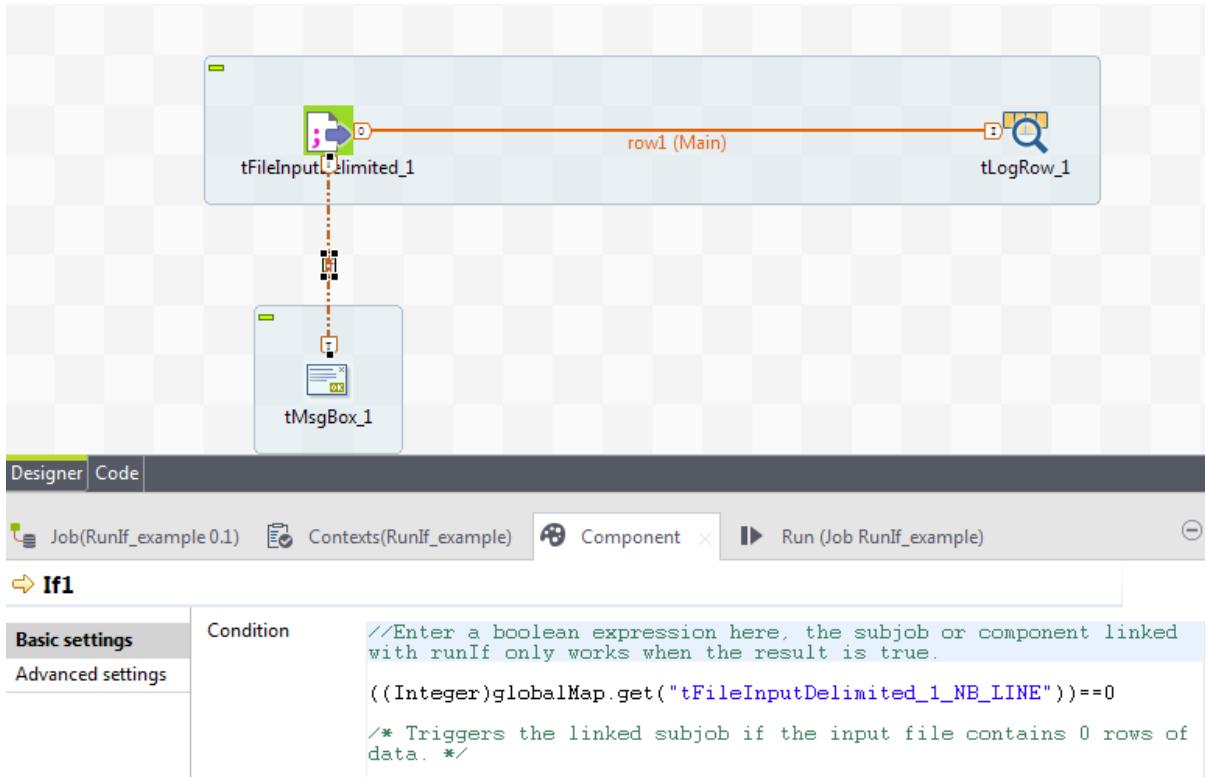
Cliquez sur la connexion pour afficher les propriétés de votre connexion **If**, puis définissez les conditions de votre projet en Java.

Vous pouvez utiliser des variables dans votre condition. Le raccourci clavier **Ctrl+Barre d'espace** vous donne accès à toutes les variables globales et de contexte. Pour plus d'informations, consultez [Utiliser des variables dans un Job](#).



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

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



- Créez un Job et déposez les trois composants suivants dans l'espace de modélisation graphique : un **tFileInputDelimited**, un **tLogRow** et un **tMsgBox**.
 - Reliez les composants comme suit :
 - Cliquez-droit sur le **tFileInputDelimited**, sélectionnez **Row > Main** dans le menu contextuel et cliquez sur le composant **tLogRow**.
 - Cliquez-droit sur le composant **tFileInputDelimited**, sélectionnez **Trigger > Run if** dans le menu contextuel et cliquez sur le **tMsgBox**.
 - Configurez le **tFileInputDelimited** afin qu'il lise un fichier contenant zéro ligne de données.
 - Sélectionnez la connexion **Run if** entre le **tFileInputDelimited** et le **tMsgBox** et cliquez sur sa vue **Component**. Dans le champ **Condition**, dans l'onglet **Basic settings**, appuyez sur les touches **Ctrl+Espace** pour accéder à la liste des variables et sélectionnez la variable **NB_LINE** du composant **tFileInputDelimited**. Modifiez la condition comme suit :
- ```
((Integer)globalMap.get("tFileInputDelimited_1_NB_LINE"))==0
```
- Cliquez sur la vue **Component** du composant **tMsgBox** et saisissez un message, "No data is read from the file" par exemple, dans le champ **Message**.
  - Sauvegardez et exécutez le Job. Vous devriez consulter le message défini dans le composant **tMsgBox**.

## 5.5. Utiliser les contextes et les variables

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

Il est possible que vous souhaitiez gérer différemment les types d'exécution de votre Job selon les conditions d'utilisation, connues comme contextes (*Prod* et *Test* dans l'exemple ci-dessous). Par exemple, vous pouvez intégrer dans votre cycle de validation plusieurs phases de test avant que votre Job soit mis en production.

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**.

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

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

Cette section décrit comment créer des contextes et des variables ainsi que définir les valeurs des paramètres de contexte. Pour un exemple de chargement dynamique des paramètres de contexte à l'aide du composant **tContextLoad**, consultez la documentation du **tContextLoad** à l'adresse <https://help.talend.com>. Pour un exemple de chargement dynamique de paramètres de contexte à l'aide de la fonctionnalité de chargement implicite des contextes, consultez *Utilisation de la fonctionnalité de chargement implicite des contextes* à l'adresse <https://help.talend.com>.

## 5.5.1. Définir les variables de contexte pour un Job

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

- Dans la vue **Contexts** du Job. Consultez [Définir les variables de contexte dans la vue Contexts](#).
- Avec la touche F5 à partir de la vue **Component** d'un composant. Consultez [Définir les variables à partir de la vue Component](#).

### 5.5.1.1. Définir les variables de contexte dans la vue Contexts

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

La vue **Contexts** montre toutes les variables qui ont été définies dans le Job courant, ainsi que les variables de contexte importées dans le Job courant.

|   | Name                  | Type     | Comment                     | Prod             |                          | Test        |                                     |
|---|-----------------------|----------|-----------------------------|------------------|--------------------------|-------------|-------------------------------------|
|   |                       |          |                             | Value            |                          | Value       |                                     |
| 1 | FILENAME              | File     | CSV file for car sales info | cars.csv         | <input type="checkbox"/> | cars.csv    | <input type="checkbox"/>            |
| 2 | TalendDB (from repos) |          |                             |                  |                          |             |                                     |
| 3 | host                  | String   |                             | "192.168.30.110" | <input type="checkbox"/> | "localhost" | <input type="checkbox"/>            |
| 4 | port                  | String   |                             | "3308"           | <input type="checkbox"/> | "3306"      | <input type="checkbox"/>            |
| 5 | database              | String   |                             | "prod_db"        | <input type="checkbox"/> | "test"      | <input type="checkbox"/>            |
| 6 | username              | String   |                             | "prod_user"      | <input type="checkbox"/> | "root"      | <input type="checkbox"/>            |
| 7 | password              | Password |                             | *****            | <input type="checkbox"/> | *****       | <input type="checkbox"/>            |
| 8 | table_name            | String   |                             | "addresses"      | <input type="checkbox"/> | "testtable" | <input checked="" type="checkbox"/> |

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

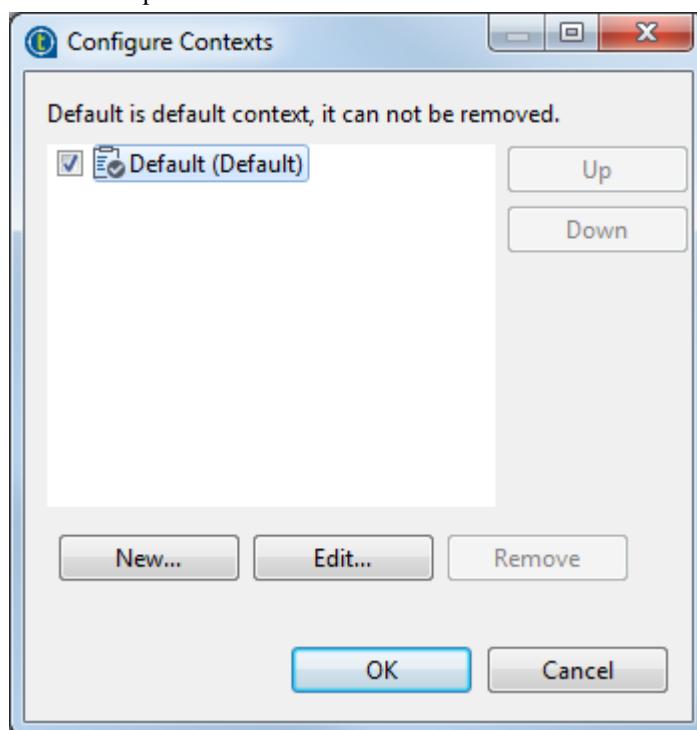
- Créer et gérer les contextes built-in.
- Créer, éditer et supprimer les variables built-in.
- Réorganiser les variables de contexte.
- Ajouter les variables de contexte built-in dans le **Repository**.
- Importer les variables à partir d'une source contexte du **Repository**.
- Éditer les variables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

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

## Définir les contextes

1. Ouvrez le Job dans l'espace de modélisation graphique.
2. Sélectionnez **Window > Show view > Talend > Contexts** pour ouvrir la vue **Contexts**, dans la perspective **Integration**.
3. Sélectionnez la vue **Contexts** et cliquez sur le bouton **[+]** dans le coin supérieur droit de la vue.

La boîte de dialogue **[Configure Contexts]** s'ouvre. Un contexte appelé *Default* a été créé et défini par le système comme étant le contexte par défaut.



4. Cliquez sur le bouton **[+]** dans le coin supérieur droit.

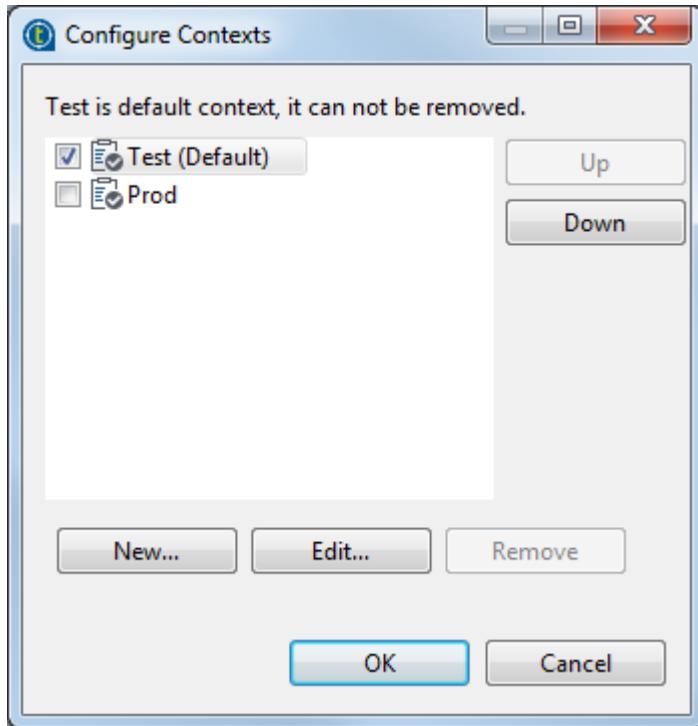
La boîte de dialogue **[Configure Contexts]** s'ouvre et un contexte nommé *Default* est créé par défaut.

5. Dans la boîte de dialogue ouverte, cliquez sur **New...** et saisissez *Test* dans la boîte de dialogue **[New Context]**, puis cliquez sur **OK**.
6. Cochez la case à côté du contexte que vous souhaitez configurer comme contexte par défaut.

Vous pouvez également configurer le contexte par défaut en sélectionnant le nom du contexte dans la liste **Default context environment**, dans la vue **Contexts**.

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

Dans cet exemple, configurez *Test* comme contexte par défaut et déplacez-le vers le haut.



- Cliquez sur **OK** pour valider votre définition de contexte et fermer la boîte de dialogue.

Les nouveaux contextes s'affichent dans la table des variables de contexte, dans la vue **Contexts**.

- Répétez les étapes ci-dessus pour ajouter autant de contextes que nécessaire.

Si vous ne souhaitez pas définir les valeur de chaque nouveau contexte de zéro, vous pouvez créer le premier contexte et définir ses valeurs, car, lorsque vous créez un nouveau contexte, tous les paramètres du contexte par défaut sont copiés dans le nouveau contexte. Vous pouvez ensuite modifier les valeurs du nouveau contexte selon vos besoins.

## Définir les variables

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

|   | Name       | Type     | Comment | Test        |                                     | Prod        |                  |
|---|------------|----------|---------|-------------|-------------------------------------|-------------|------------------|
|   |            |          |         | Value       | Prompt                              | Value       | Prompt           |
| 1 | host       | String   |         | "localhost" | <input type="checkbox"/>            | host?       | "192.168.30.110" |
| 2 | port       | String   |         | "3306"      | <input type="checkbox"/>            | port?       | "3308"           |
| 3 | database   | String   |         | "test"      | <input type="checkbox"/>            | database?   | "prod_db"        |
| 4 | username   | String   |         | "root"      | <input type="checkbox"/>            | username?   | "prod_user"      |
| 5 | password   | Password |         | *****       | <input type="checkbox"/>            | password?   | *****            |
| 6 | table_name | String   |         | "testtable" | <input checked="" type="checkbox"/> | table_name? | "addresses"      |
|   |            |          |         |             |                                     |             |                  |

+
-
↑
↓
✖
✖

Default context environment Test ▾

- Cliquez dans le champ **Name** et saisissez le nom de la variable que vous créez.

Nommez la première variable *host* dans cet exemple.

3. Dans la liste **Type**, sélectionnez le type de la variable.
4. Si nécessaire, cliquez dans le champ **Comment** et saisissez un commentaire décrivant la variable.
5. Cliquez dans le champ **Value** et saisissez la valeur de la variable sous chaque contexte.

Lorsque vous cliquez dans le champ **Value**, il peut être légèrement différent pour les différents types de variables, ses fonctions également :

| Type                                                               | Contenu du champ                                                                                                                        | Valeur par défaut |
|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>String</b> (type par défaut)                                    | Champ de texte modifiable                                                                                                               | null              |
| <b>Boolean</b>                                                     | Liste déroulante avec deux options : <b>true</b> et <b>false</b>                                                                        |                   |
| <b>Character, Double, Integer, Long, Short, Object, BigDecimal</b> | Champ de texte modifiable                                                                                                               |                   |
| <b>Date</b>                                                        | Champ de texte modifiable, avec un bouton pour ouvrir la boîte de dialogue <b>[Select Date &amp; Time]</b> .                            |                   |
| <b>File</b>                                                        | Champ de texte modifiable, avec un bouton pour ouvrir la boîte de dialogue <b>[Open]</b> et sélectionner le fichier.                    |                   |
| <b>Directory</b>                                                   | Champ de texte modifiable, avec un bouton pour ouvrir la boîte de dialogue <b>[Browse for Folder]</b> et sélectionner le dossier.       |                   |
| <b>List of Value</b>                                               | Champ de texte modifiable, avec un bouton pour ouvrir la boîte de dialogue <b>[Configure Values]</b> pour créer et configurer la liste. | (Empty)           |
| <b>Password</b>                                                    | Champ de texte modifiable ; le texte saisi s'affiche de manière chiffrée.                                                               |                   |



*Il est recommandé d'entourer les valeurs des variables de type String type par des guillemets doubles, afin d'éviter de possibles erreurs durant l'exécution du Job.*

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

Vous pouvez masquer/afficher une colonne **Prompt** de la table en cliquant sur le triangle noir pointant vers la droite/gauche, respectivement, à côté du nom du contexte concerné.

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

- *port*, de type **String**,
- *database*, de type **String**,
- *username*, de type **String**,
- *password*, de type **Password**,
- *table\_name*, de type **String**.

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

Vous pouvez également ajouter une variable de contexte built-in dans le **Repository** afin de pouvoir la réutiliser dans d'autres Jobs. Pour plus d'informations, consultez [Ajouter une variable de contexte built-in dans le Repository](#).

## Voir également :

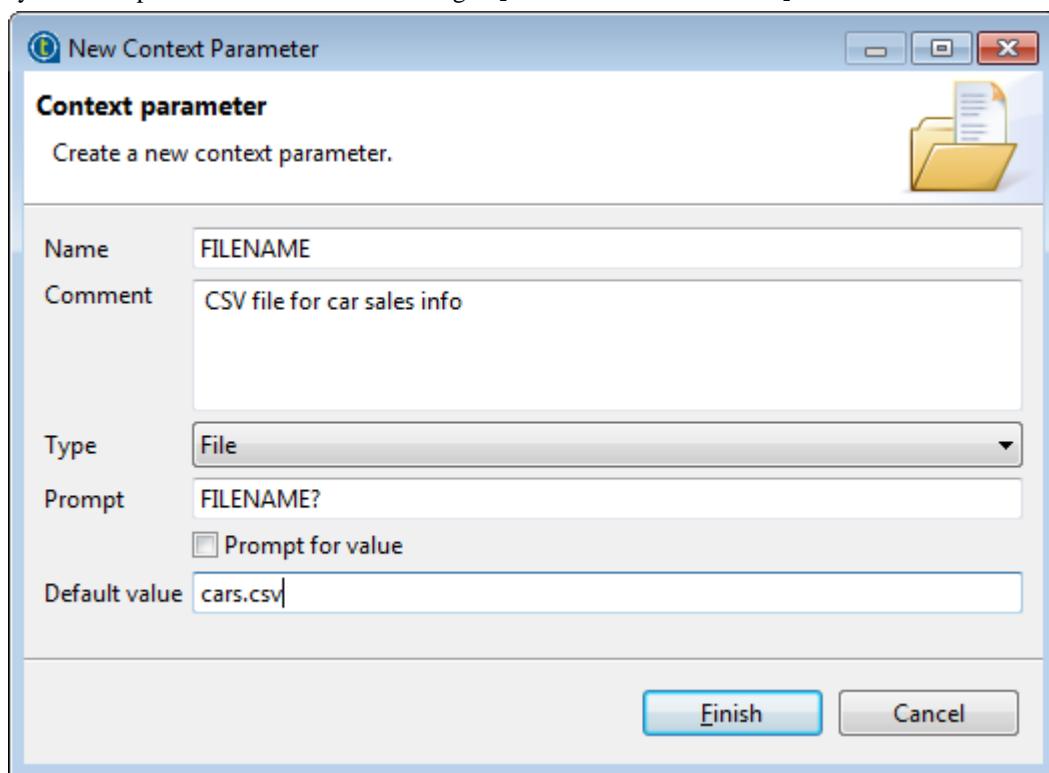
- [Définir les variables à partir de la vue Component](#)
- [Centraliser les variables de contexte dans le Repository](#)

- Utiliser des variables dans un Job
- Exécuter un Job dans un contexte défini

### 5.5.1.2. Définir les variables à partir de la vue Component

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

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



3. Donnez un nom à cette nouvelle variable dans le champ **Name**, renseignez le champ **Comment** si nécessaire 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), puis cochez la case **Prompt for value** pour afficher le message **Prompt** et le champ de valeur éditable lors de l'exécution.
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 [Définir les contextes](#).

### Voir également :

- [Définir les variables de contexte dans la vue Contexts](#)
- [Centraliser les variables de contexte dans le Repository](#)
- [Utiliser des variables dans un Job](#)
- [Exécuter un Job dans un contexte défini](#)

## 5.5.2. Centraliser les variables de contexte dans le Repository

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

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

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

### 5.5.2.1. Créer un groupe de contexte et définir les variables de contexte

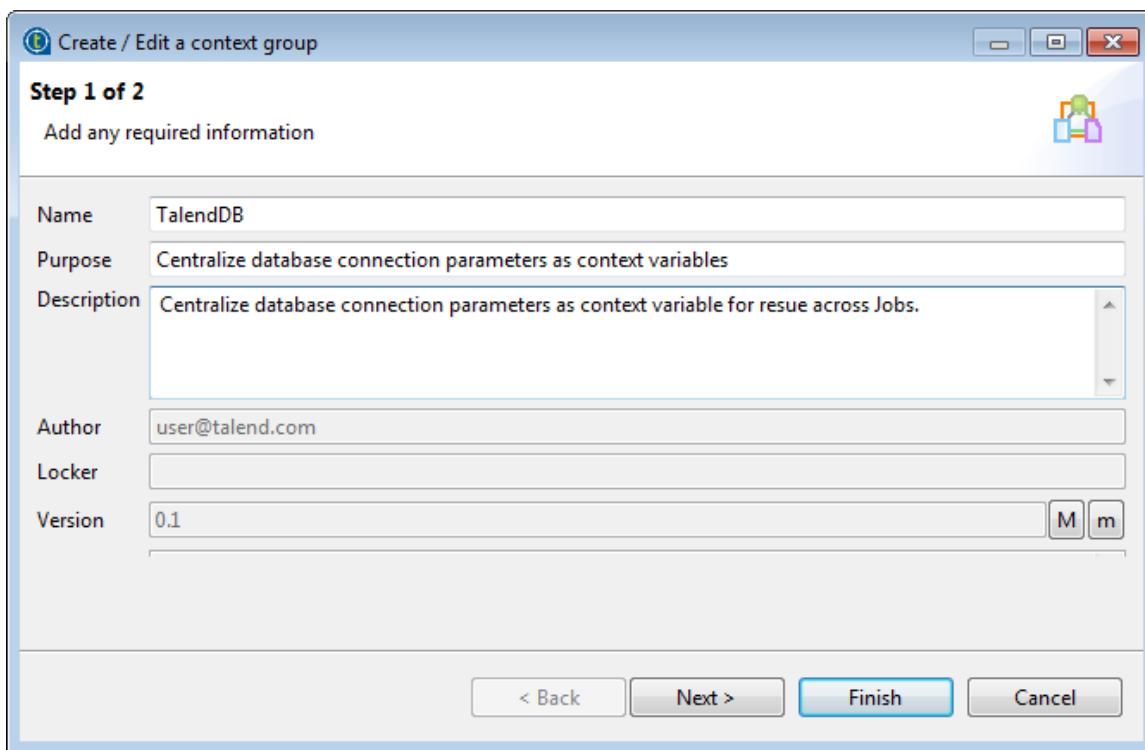
L'exemple suivant montre comment utiliser l'assistant **[Create/Edit a context group]** pour créer un groupe de contexte appelé *TalendDB* contenant deux contextes (appelés *Prod* et *Test*) et pour définir un série de variables (*host*, *port*, *database*, *username*, *password* et *table\_name*) sous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

#### Créez le groupe de contexte et les contextes

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

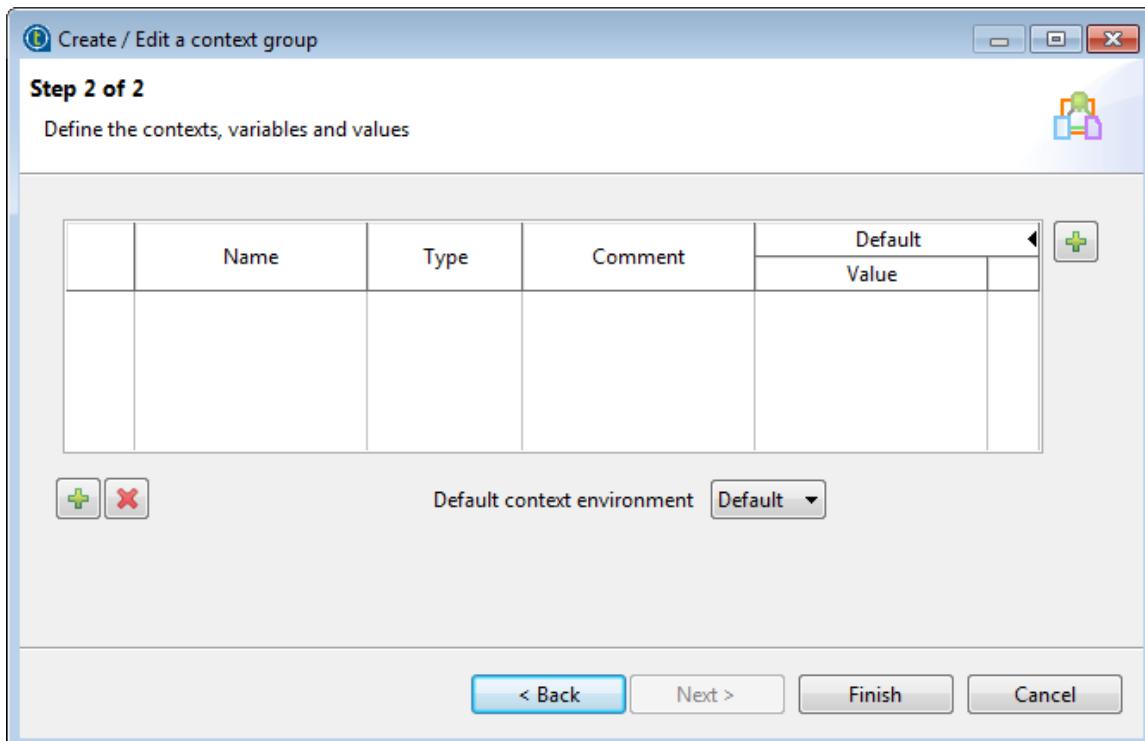
Un assistant en deux étapes s'ouvre pour vous aider à définir les différents contextes et leurs paramètres.

2. Dans la première étape, saisissez un nom pour le groupe de contexte à créer (*TalendDB* dans cet exemple) et ajoutez toutes les informations générales, telles que la description, si elles sont requises. Les informations que vous fournissez dans le champ **Description** apparaissent dans une info-bulle lorsque vous bougez le curseur de votre souris au-dessus du groupe de contexte dans le **Repository**.

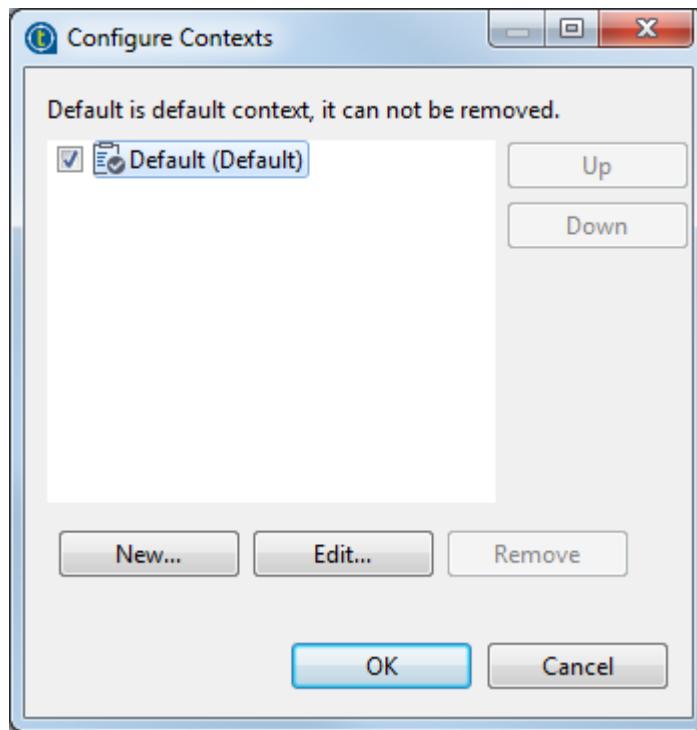


3. Cliquez sur **Next** pour passer à la seconde étape qui vous permet de définir les différents contextes et variables dont vous avez besoin.

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



4. Cliquez sur le bouton [+] dans le coin supérieur droit de l'assistant pour définir les contextes. La boîte de dialogue **[Configure Contexts]** s'ouvre.



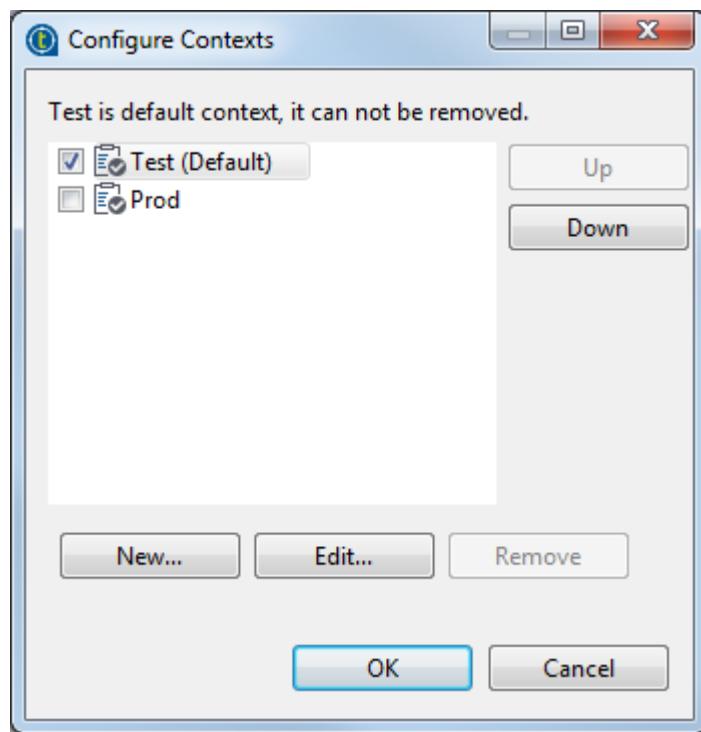
5. Sélectionnez le contexte **Default**, cliquez sur le bouton **Edit...** et saisissez *Prod* dans la boîte de dialogue **[Rename Context]** qui s'ouvre pour renommer le contexte *Default* en *Prod*.

Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.

6. Cliquez sur le bouton **New...** et saisissez *Test* dans la boîte de dialogue **[New Context]**. Cliquez ensuite sur **OK** pour fermer la boîte de dialogue.
7. Cochez la case précédant le contexte que vous souhaitez définir comme contexte par défaut. Vous pouvez également définir le contexte par défaut en sélectionnant le nom du contexte dans la liste **Default context environment** dans l'assistant.

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

Dans cet exemple, définissez *Test* comme contexte par défaut et déplacez-le vers le haut.



- Cliquez sur **OK** pour valider le contexte que vous avez défini et fermer la boîte de dialogue [**Configure Contexts**].

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

|  | Name | Type | Comment | Test<br>Value | Prod<br>Value |
|--|------|------|---------|---------------|---------------|
|  |      |      |         |               |               |

Default context environment **Test**

## Définir les variables de contexte

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

3. Dans la liste **Type**, sélectionnez le type de la variable correspondant au champ de composant dans lequel cette variable sera utilisée. Dans cet exemple, il s'agit du type **String** pour la variable *host*.
4. Si nécessaire, cliquez dans le champ **Comment** et saisissez un commentaire pour décrire la variable.
5. Cliquez dans le champ **Value** et saisissez la valeur de la variable sous chaque contexte.

Selon le type de variable, le champ **Value** différera légèrement lorsque vous cliquerez dedans et fonctionnera de manière différente :

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



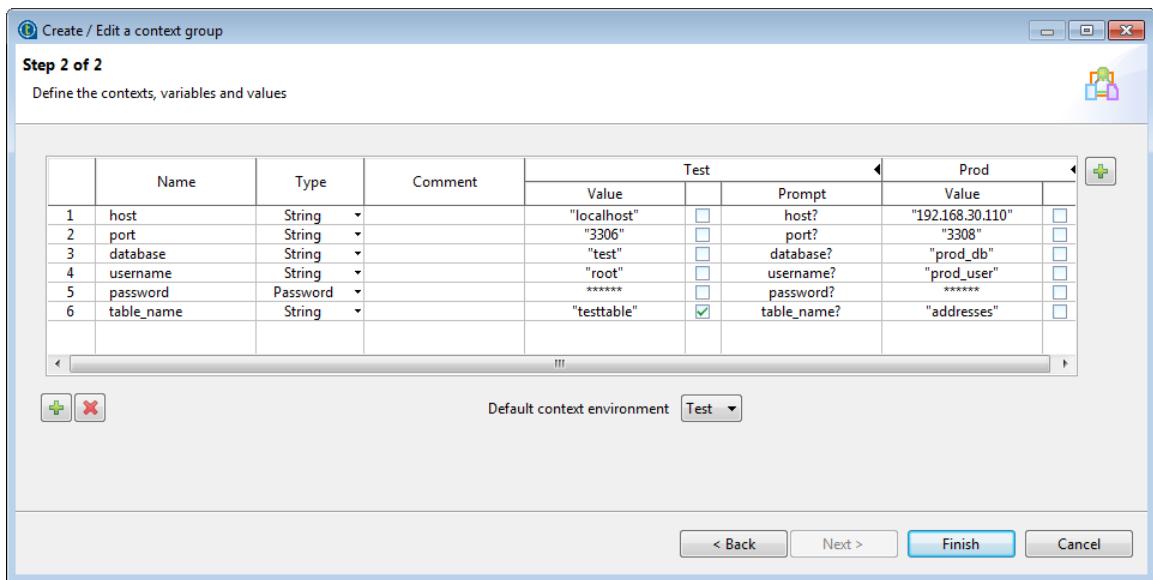
*Il est recommandé que d'entourer les valeurs des variables de type **String** par des guillemets doubles afin d'éviter d'éventuelles erreurs lors de l'exécution du Job.*

6. Si nécessaire, cochez la case à côté de la variable qui vous intéresse et saisissez le message **Prompt** dans le champ **Prompt** correspondant. Cela vous permet de voir le message **Prompt** pour la valeur de la variable et de l'éditer lors de l'exécution.

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

7. Répétez les étapes décrites ci-dessus pour définir toutes les variables dans cet exemple.

- *port*, de type **String**,
- *database*, de type **String**,
- *username*, de type **String**,
- *password*, de type **Password**,
- *table\_name*, de type **String**



Toutes les variables créées et leurs valeurs sous les différents contextes sont affichées dans la table et sont prêtes à être utilisées dans votre Job. Vous pouvez continuer d'édition des variables si nécessaire.

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

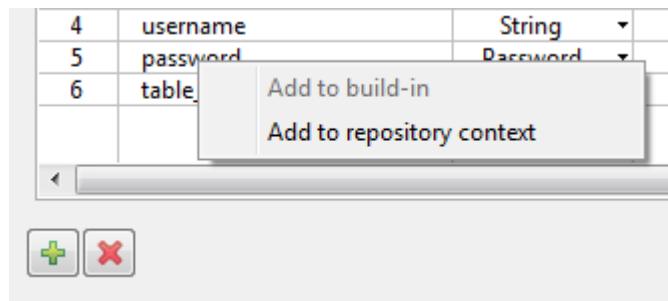
### Voir également :

- [Ajouter une variable de contexte built-in dans le Repository](#).
- [Créer un contexte à partir d'une métadonnée](#).
- [Appliquer des variables de contexte du référentiel dans un Job](#).
- [Définir les variables de contexte pour un Job](#).
- [Utiliser des variables dans un Job](#).
- [Exécuter un Job dans un contexte défini](#).

### 5.5.2.2. Ajouter une variable de contexte built-in dans le Repository

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

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



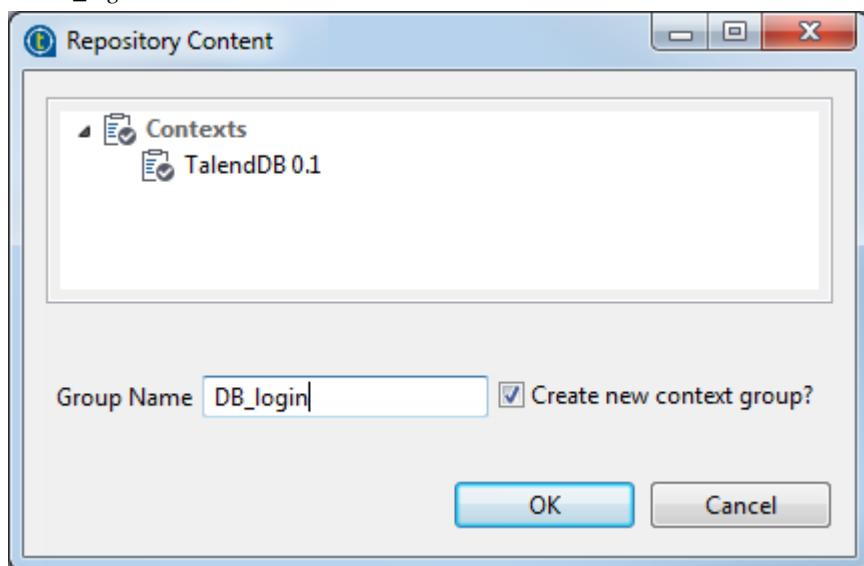
2. Dans la boîte de dialogue, effectuez l'une des actions suivantes :

- pour ajouter votre variable de contexte à un nouveau groupe de contexte, sélectionnez **Create new context group** et saisissez un nom pour le nouveau groupe de contexte dans le champ **Group Name**, puis cliquez sur **OK**.
- pour ajouter une variable de contexte à un groupe de contexte déjà existant, sélectionnez le groupe de contexte et cliquez sur **OK**.



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

Dans cet exemple, ajoutez la variable de contexte *password* définie dans un Job à un nouveau groupe de contexte appelé *DB\_login*.



La variable de contexte est ajoutée au groupe de contexte de votre choix dans le **Repository**, avec les contextes built-in définis. Elle apparaît en tant que variable de contexte stockée dans le **Repository** dans la vue **Contexts**.

|   | Name                       | Type     | Comment | Test        |             |                  | Prod        |  |  |
|---|----------------------------|----------|---------|-------------|-------------|------------------|-------------|--|--|
|   |                            |          |         | Value       | Prompt      | Value            | Prompt      |  |  |
| 1 | host                       | String   |         | "localhost" | host?       | "192.168.30.110" | host?       |  |  |
| 2 | port                       | String   |         | "3306"      | port?       | "3308"           | port?       |  |  |
| 3 | database                   | String   |         | "test"      | database?   | "prod_db"        | database?   |  |  |
| 4 | username                   | String   |         | "root"      | username?   | "prod_user"      | username?   |  |  |
| 5 | DB_login (from repository) |          |         | *****       | password?   | *****            | password?   |  |  |
| 6 | password                   | Password |         | "testtable" | table_name? | "addresses"      | table_name? |  |  |
| 7 | table_name                 | String   |         |             |             |                  |             |  |  |

## Voir également :

- [Créer un groupe de contexte et définir les variables de contexte.](#)

- *Créer un contexte à partir d'une métadonnée.*
- *Appliquer des variables de contexte du référentiel dans un Job.*
- *Définir les variables de contexte pour un Job .*
- *Utiliser des variables dans un Job.*
- *Exécuter un Job dans un contexte défini.*

### 5.5.2.3. Créer un contexte à partir d'une métadonnée

Lorsque vous créez ou éditez une connexion de métadonnées (à l'aide d'un assistant de métadonnées File ou DB), vous avez la possibilité de sauvegarder les paramètres de connexion en tant que variables de contexte dans un groupe de contexte nouvellement créé sous le nœud **Contexts** du **Repository**. Pour ce faire, complétez vos détails de connexion et cliquez sur le bouton **Export as context** dans la seconde étape de l'assistant.

Pour plus d'informations concernant cette fonctionnalité, consultez *Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion.*

### 5.5.3. Appliquer des variables de contexte du référentiel dans un Job

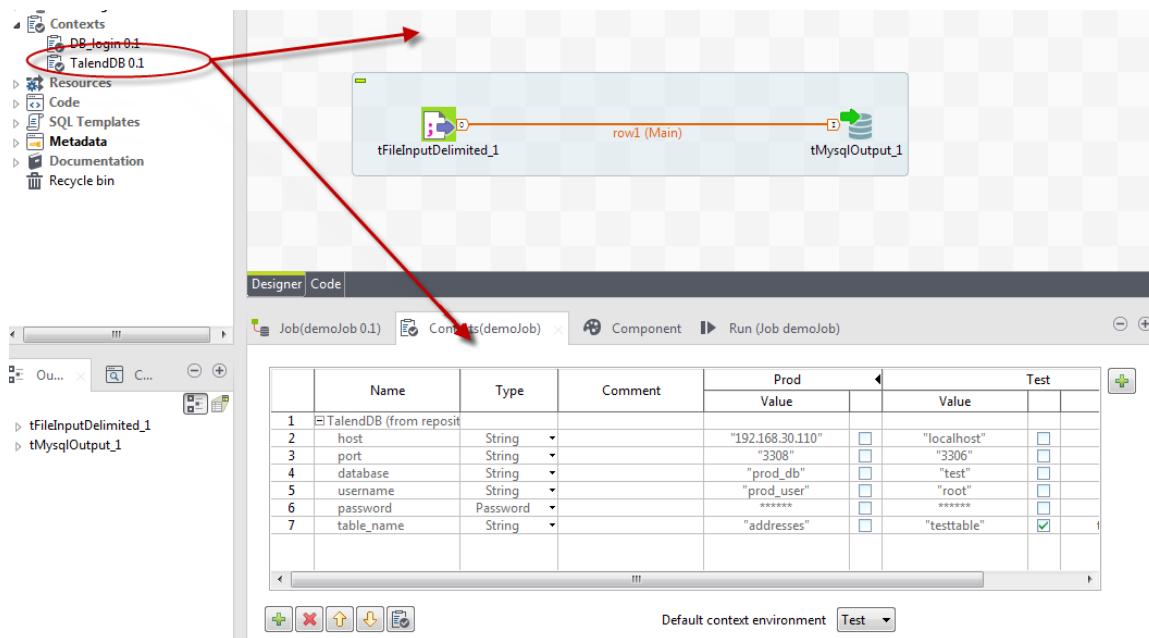
Une fois le groupe de contexte créé et stocké dans le **Repository**, vous pouvez l'appliquer à un Job par deux moyens :

- Déposez un groupe de contexte. Ainsi, le groupe sera appliqué comme un tout. Consultez *Déposer un groupe de contexte dans un Job* pour plus de détails.
- Utilisez le bouton  . De cette manière, les variables d'un groupe de contexte pourront être appliquées séparément. Consultez *Appliquer des variables de contexte dans un Job à l'aide du bouton contexte* pour plus de détails.

#### 5.5.3.1. 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 ajouter 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, soit dans la vue **Contexts** sous l'espace de travail.



La vue **Contexts** montre tous les contextes et toutes les variables du groupe. Vous pouvez :

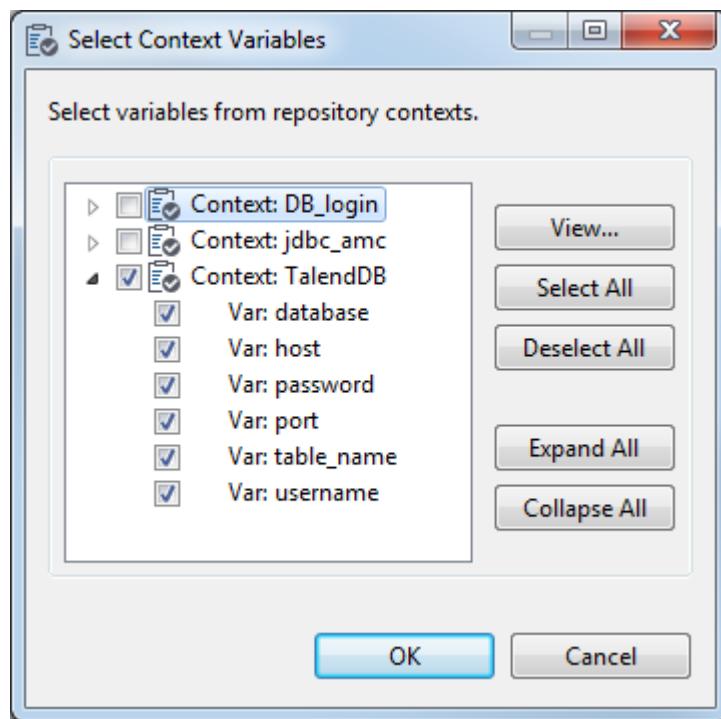
- éditer les contextes en cliquant sur le bouton **[+]** dans le coin supérieur droit de la vue **Contexts**.
- supprimer tout le groupe ou n'importe quelle variable en sélectionnant le nom de groupe ou la variable et en cliquant sur le bouton **[X]**.
- sauvegarder n'importe quelle variable de contexte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

### 5.5.3.2. Appliquer des variables de contexte dans un Job à l'aide du bouton contexte



Pour utiliser le bouton 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 le bouton afin d'ouvrir l'assistant pour sélectionner les variables de contexte à appliquer.



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



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

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

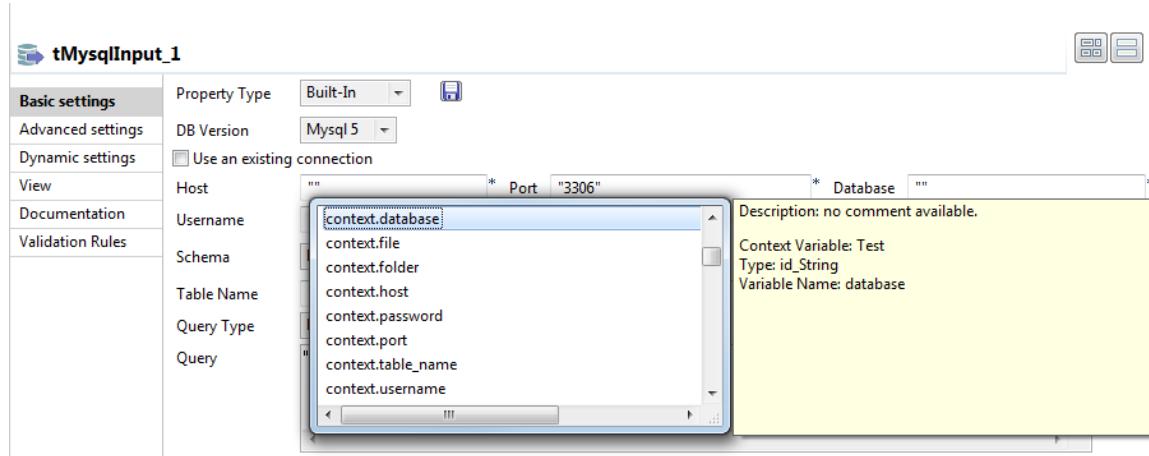
La vue **Contexts** montre le groupe de contexte et les variables de contexte sélectionnées. Vous pouvez éditer les contextes en cliquant sur le bouton **[+]** dans le coin supérieur droit de la vue **Contexts**. Vous pouvez également supprimer tout le groupe ou n'importe quelle variable en sélectionnant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

## 5.5.4. Utiliser des variables dans un Job

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

- Dans la vue **Component** correspondante, placez votre curseur de souris dans le champ que vous souhaitez paramétrier et appuyez sur **Ctrl+Espace** pour afficher la liste complète de toutes les variables globales et de contexte définies ou appliquées dans votre Job.

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



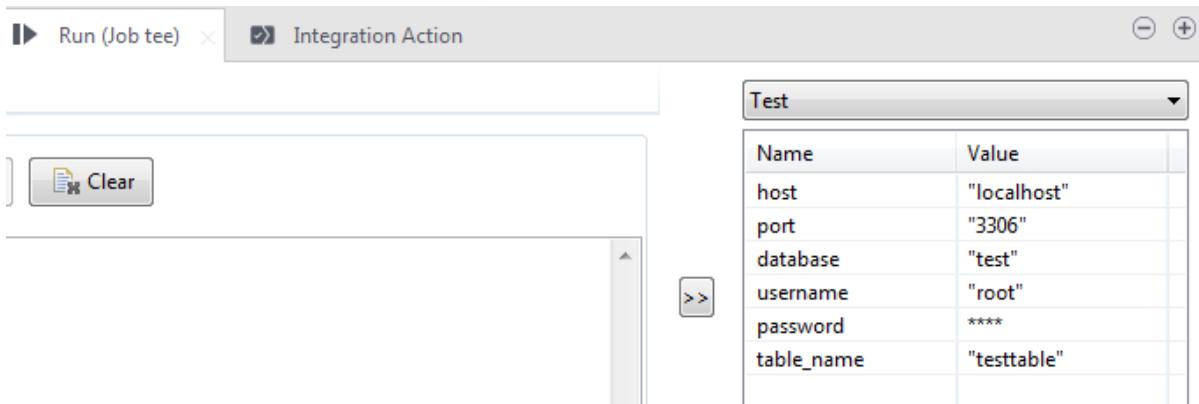
2. Double-cliquez sur la variable de votre choix pour la renseigner dans le champ.

#### Voir également :

- [Définir les variables de contexte pour un Job .](#)
- [Centraliser les variables de contexte dans le Repository.](#)
- [Appliquer des variables de contexte du référentiel dans un Job.](#)
- [Exécuter un Job dans un contexte défini.](#)

## 5.5.5. Exécuter un Job dans un contexte défini

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



Cliquez sur l'onglet **Run** et, dans la zone **Context**, sélectionnez le contexte approprié parmi ceux que vous avez créés.

Si vous n'avez pas créé de contexte, seul le contexte par défaut, **Default**, est proposé dans la liste.

Toutes les variables de contexte que vous avez créées pour le contexte en sélection, ainsi que leurs valeurs associées apparaissent dans un tableau.

Pour rendre permanent une modification de valeur de variable, vous devez la changer dans la vue **Context** si votre variable est de type built-in ou sélectionnez un autre groupe de contexte du **Repository**.

#### Voir également

- [Définir les variables de contexte pour un Job .](#)

- *Centraliser les variables de contexte dans le Repository.*
- *Appliquer des variables de contexte du référentiel dans un Job.*
- *Utiliser des variables dans un Job.*

## 5.5.6. Utiliser une variable de contexte pour déboguer

**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**, consultez le scénario du composant **tDBInput** à l'adresse <https://help.talend.com>.

## 5.6. 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.

### 5.6.1. Mapper des flux de données

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

Pour plus d'informations concernant les principes d'utilisation de ce composant, consultez [Mapping de flux de données](#).

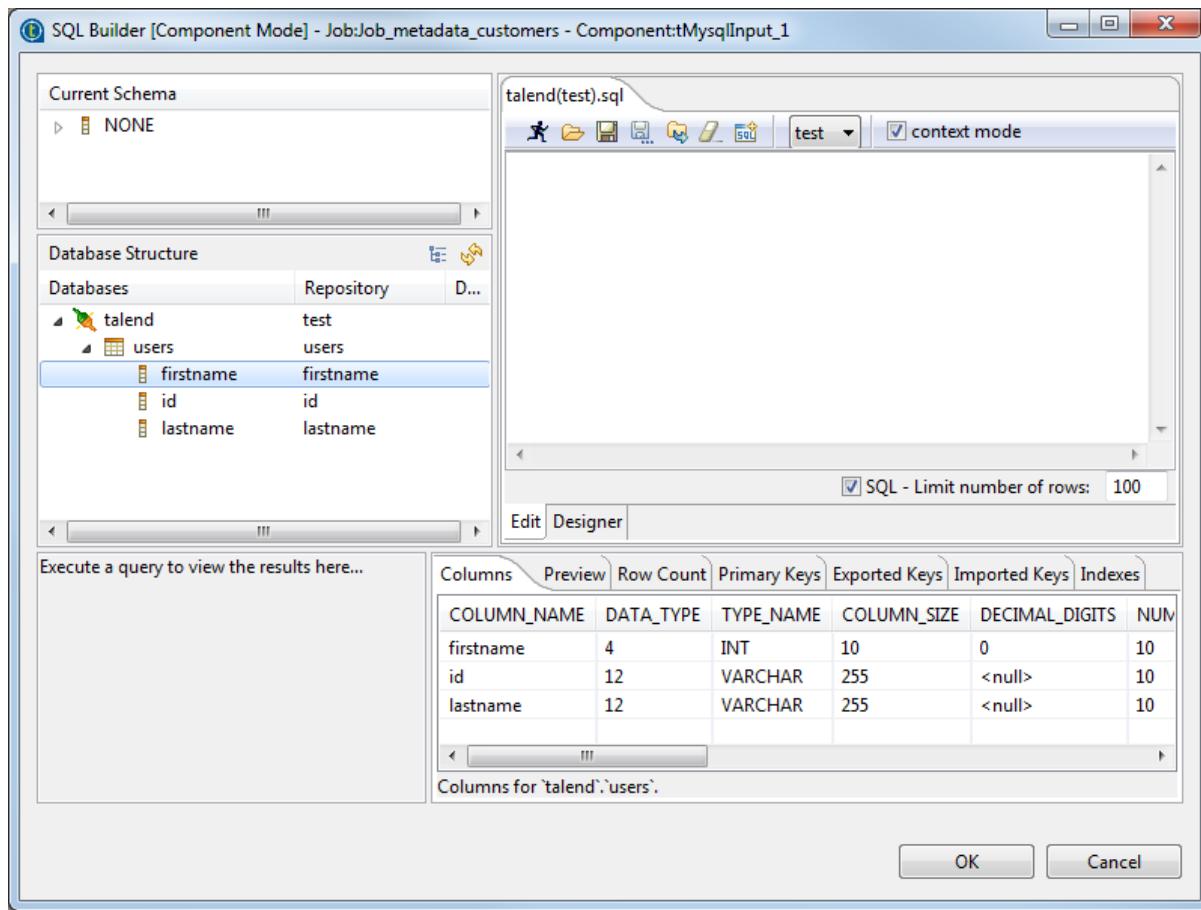
### 5.6.2. Créez une requête à l'aide de SQLBuilder

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

Vous pouvez créer une requête avec SQLbuilder que votre schéma de table soit stocké dans le 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 :

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

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

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

### 5.6.2.1. 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 elle-même.

 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.

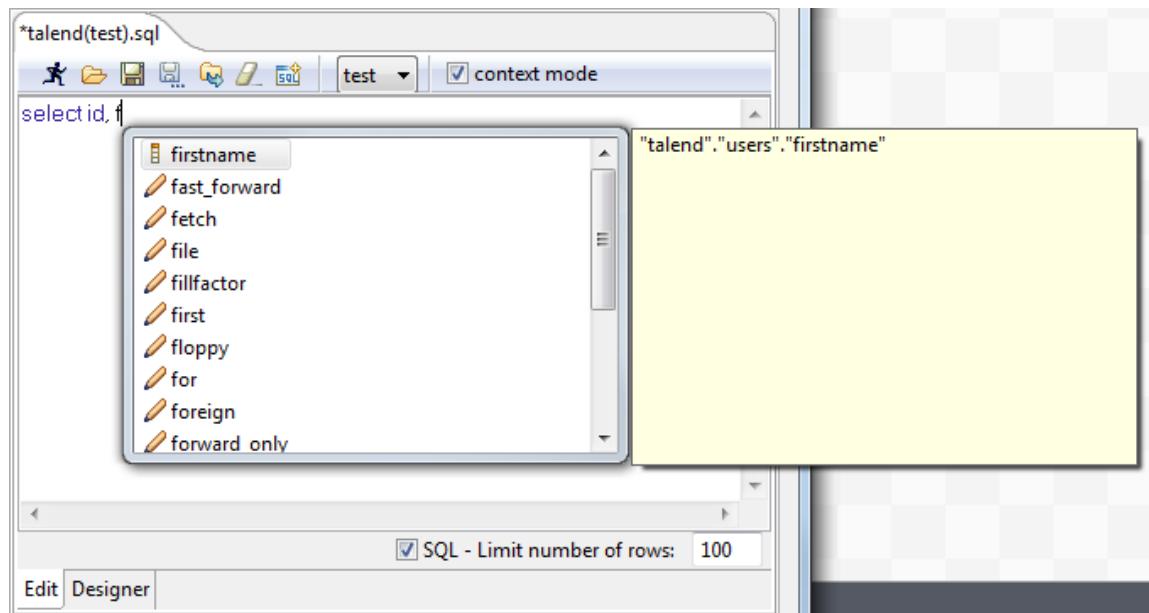
Le surlignage bleu indique que la colonne est présente dans la base de données elle-même mais qu'elle n'est pas présente dans la table de la métadonnée.

### 5.6.2.2. Créer une requête

L'éditeur de requête est un système multi-onglet 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**.

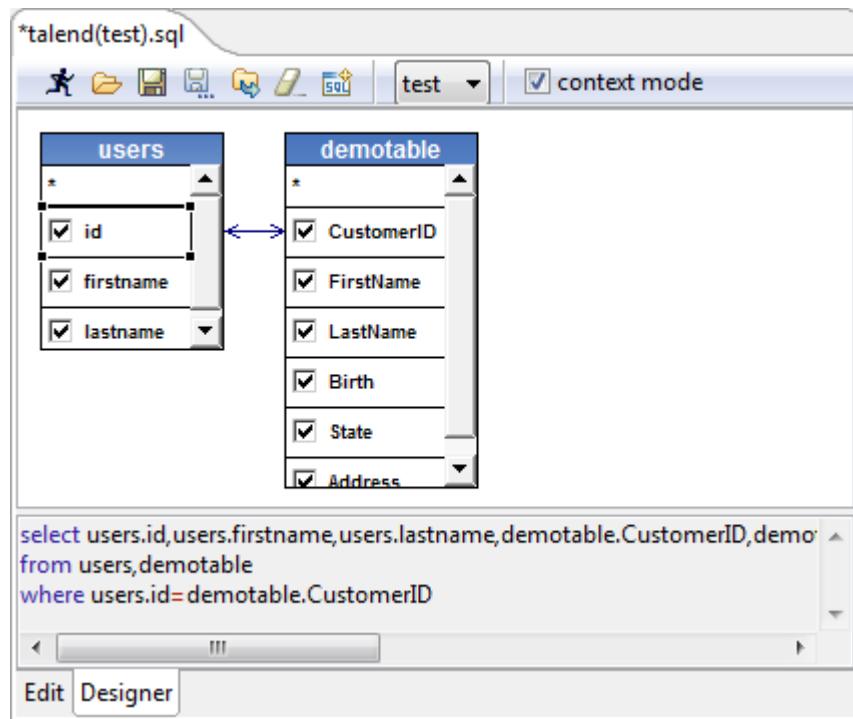
- Cliquez sur l'onglet **Designer** pour passer du mode **Edit** manuel au mode graphique.

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

- Si vous avez sélectionné une table, toutes les colonnes sont sélectionnées par défaut. Décochez la case correspondant aux colonnes que vous souhaitez exclure de la sélection.
- Dans l'onglet **Designer**, ajoutez d'autres tables en sélectionnant **Add tables** dans le menu contextuel puis en choisissant la table à ajouter dans la liste.

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

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



La requête SQL correspondante s'affiche également dans la vue d'aperçu. Sinon, 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 courrier.

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

Les résultats de la requête active s'affichent dans la vue **Results**.

7. Si nécessaire, vous pouvez cocher la case **context mode** pour conserver l'instruction de requête originale et la personnaliser dans la zone **Query** du composant. Pour plus d'exemples, si un paramètre de contexte est utilisé dans l'instruction de requête, vous ne pouvez l'exécuter en cliquant sur le bouton dans la barre d'outils.
8. Cliquez sur **OK**. L'instruction de requête est automatiquement chargée dans la zone **Query** du composant.

### 5.6.2.3. Stocker une requête dans le Repository

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

Dans l'éditeur SQL Builder, cliquez sur **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.

## 5.6.3. Télécharger/charger des composants de la communauté

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

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

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

Notez que l'approche expliquée dans cette section doit être utilisée uniquement pour les composants susmentionnés.



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

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.

### 5.6.3.1. 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 actuel :

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

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

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.

**tPDFToText**

Version 1.1  
2011-05-17

Convert a PDF to text file. It's possible to extract a delimited area.

bien it's good  
It works on 4.2.2 You can use it, it works o

#### User Reviews

[write a review](#)

- bien it's good
- It works on 4.2.2 You can use it, it works o
- n 4.2.2!
- tPDFToText This does not seem compatible with TOS 4.2.2r63143. I have installed it on TOS and it does n ot 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 [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    | <a href="#">Install</a> |
|                       | tDBFOoutput            | 1.1                | 2011-10-31    | <a href="#">Install</a> |
|                       | bcLogbackConfig        | 1.2                | 2011-10-31    | <a href="#">Install</a> |
|                       | bcLogbackCatch         | 1.3                | 2011-10-31    | <a href="#">Install</a> |
|                       | tLog4J                 | 1.4                | 2011-11-17    | <a href="#">Install</a> |
|                       | tFileOutputDelimitedEx | 1.0                | 2011-11-17    | <a href="#">Install</a> |
|                       | null                   | null               | 2011-11-17    | <a href="#">Install</a> |
|                       | tScriptRules           | 1.0                | 2011-11-17    | <a href="#">Install</a> |
|                       | BRules                 | 1.1                | 2011-11-17    | <a href="#">Install</a> |
|                       | tPDFToText             | 1.1                | 2011-11-18    | Installed               |

### 5.6.3.2. 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 *Studio Talend* ou installer la version mise à jour du *Studio Talend* 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 [Préférence Exchange \(Talend > Exchange\)](#).

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 du *Studio Talend* 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**.

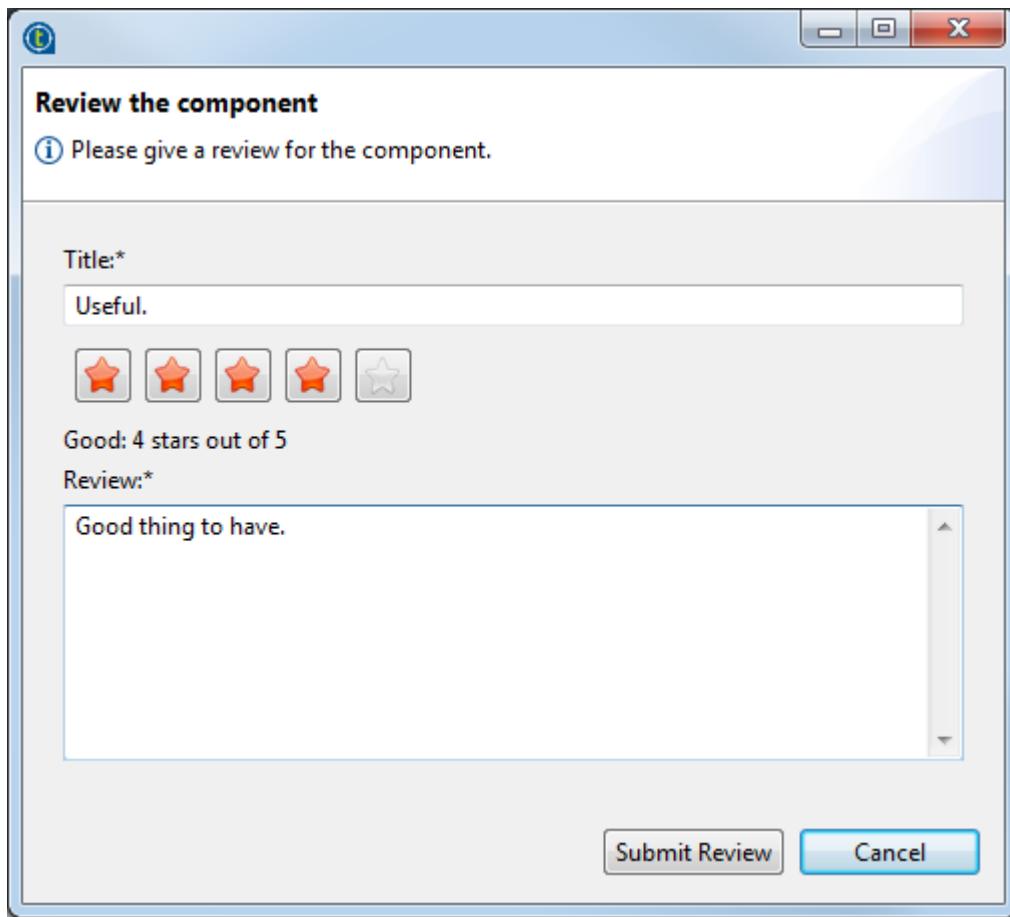
2. Cliquez sur le lien **Install** ou **Update** du composant qui vous intéresse afin de commencer l'installation.

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**.

### 5.6.3.3. 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.
2. 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.

#### 5.6.3.4. Charger dans Talend Exchange un composant créé

Vous pouvez créer vos propres composants afin de les utiliser dans vos Jobs, dans le *Studio Talend* et les charger dans **Talend Exchange** pour les partager avec les autres utilisateurs de **Talend**. Pour plus d'informations concernant la création et le déploiement de vos propres composants dans le *Studio Talend*, consultez [Définir le dossier de composants utilisateur \(Talend > Components\)](#).

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 | Downloaded Extensions | Add New Extension |         |             |           |
|----------------------|-----------------------|-------------------|---------|-------------|-----------|
| 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.

**Add New Extension**

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: C:\Work\Components\tExEvaluate.zip

**Add Extension**

- 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 | Extension Name | Version    | Upload Date |                   |
| My Extensions        | tExEvaluate           | 1.0            | 2011-11-18 |             |                   |

### 5.6.3.5. 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

Compatibility:

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

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

**Upload Extension**

- 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 :

- 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**

- 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**.

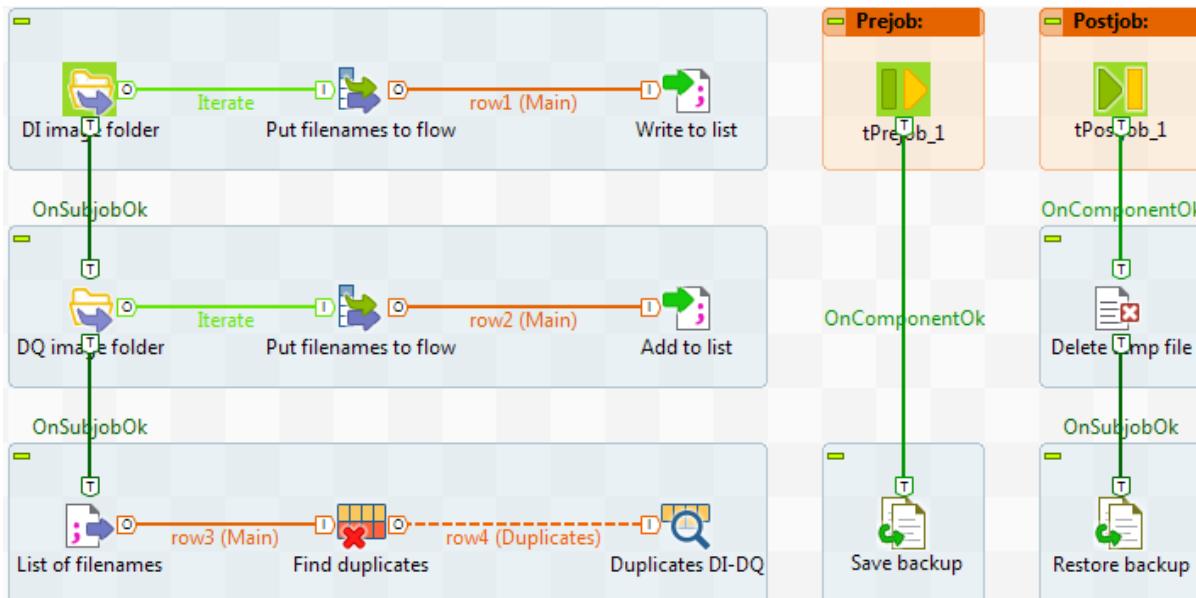
## 5.6.4. Utiliser les composants tPrejob et tPostjob

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

- 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.

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 les composants **tPrejob** et **tPostjob**, déposez-les simplement dans l'espace de modélisation graphique comme vous le feriez avec d'autres composants et connectez-les à ceux-ci à l'aide d'une connexion **Trigger**. Un carré orange affiche les parties pré et post-job qui sont des types différents de sous-jobs.

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.

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

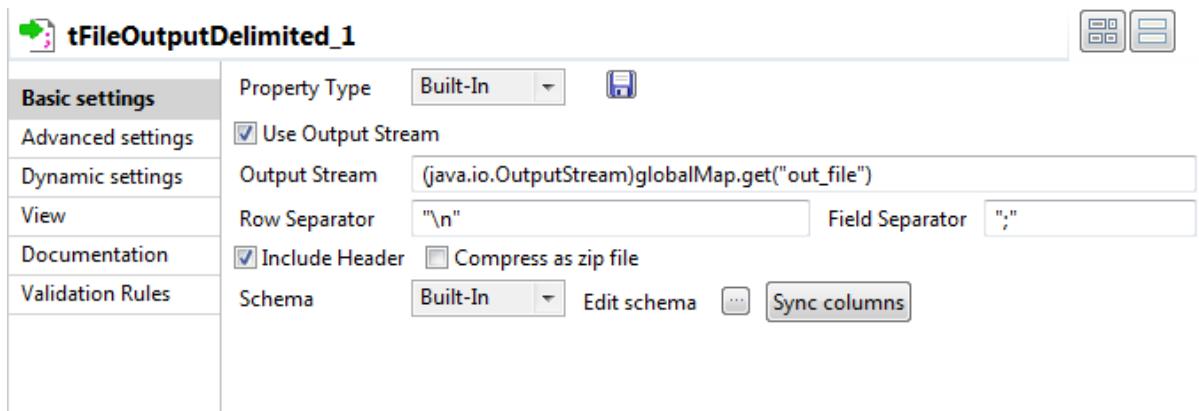
- la suppression des fichiers temporaires créés durant l'exécution du Job principal,

- fermer la connexion à une base de données ou à un service extérieur,
- toute tâche devant être exécutée même si le sous-job précédent a échoué.

Pour un exemple d'utilisation des composants **tPrejob** et **tPostjob**, consultez le scénario du composant **tPrejob** à l'adresse <https://help.talend.com>.

## 5.6.5. 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 l'utilisation de cette fonctionnalité, consultez *Utilisation de la fonctionnalité Use Output Stream* à l'adresse <https://help.talend.com>.

## 5.7. Gestion des Jobs : Autres fonctions

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

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

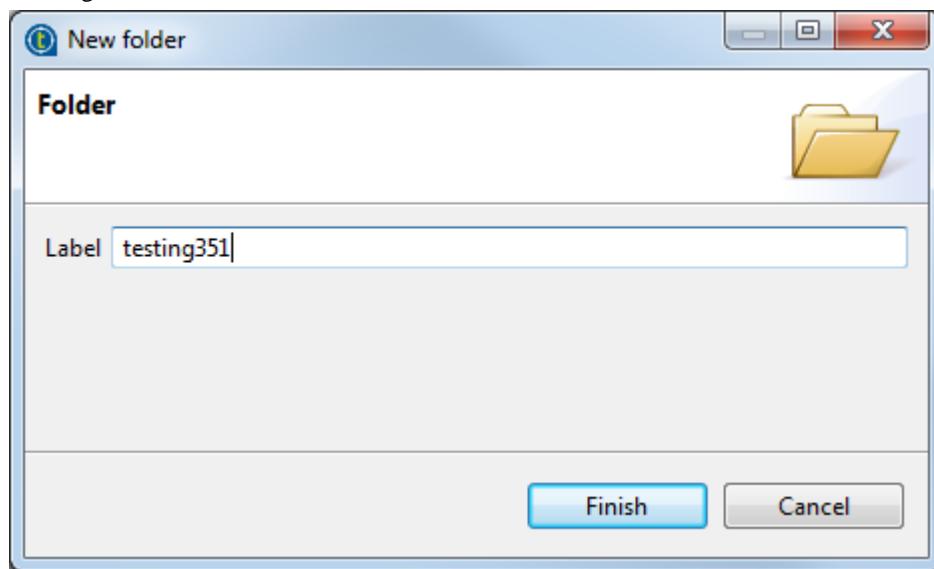
## 5.7.1. Utiliser les dossiers

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

Pour créer un dossier, procédez comme suit :

- Dans la vue **Repository** de la perspective **Integration**, cliquez-droit sur **Job Designs** et sélectionnez **Create Folder** dans le menu contextuel.

La boîte de dialogue **[New Folder]** s'ouvre.



- 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.

## 5.7.2. Partager une connexion à une base de données

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

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

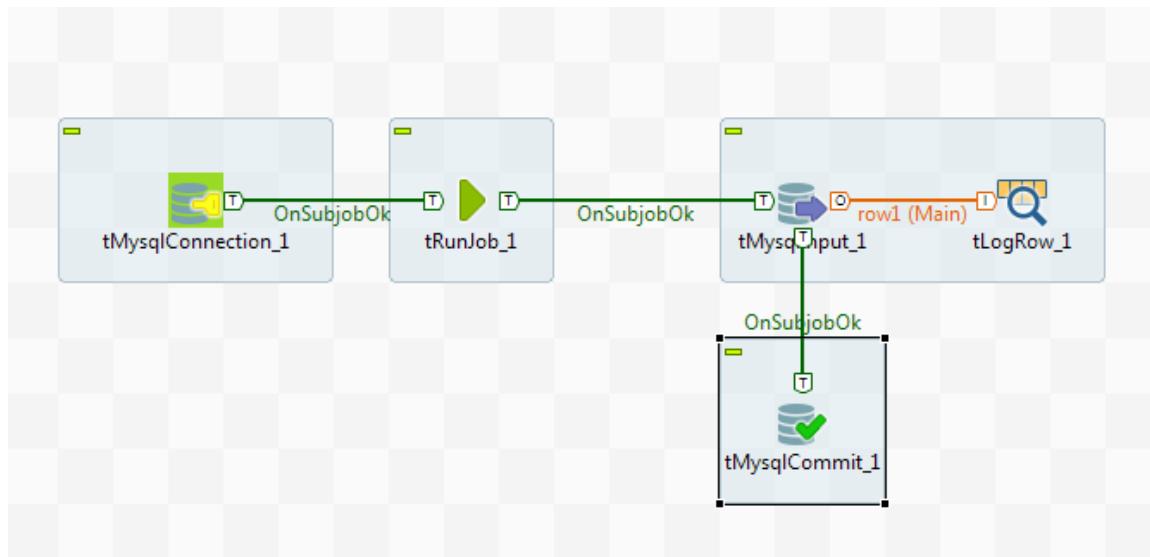


*L'option **Use or register a shared DB Connection** de tous les composants de connexion aux bases de données est incompatible avec les options **Use dynamic job** et **Use an independent process to run subjob** du composant **tRunJob**. Utiliser une connexion partagée à une base de données avec un **tRunJob**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

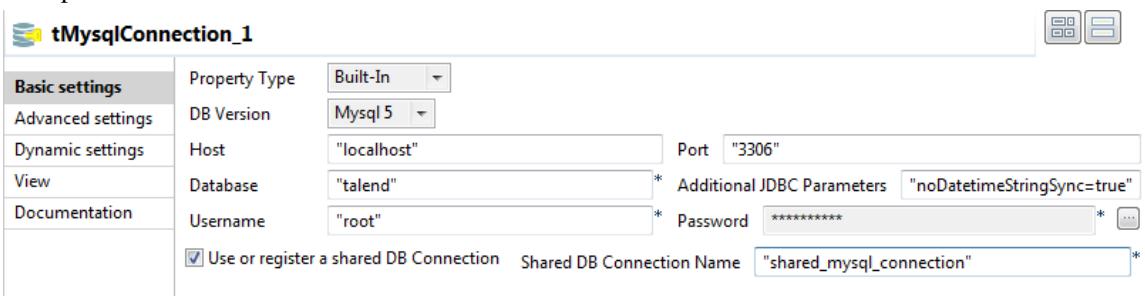
Considérons, par exemple, deux Jobs liés (un Job parent et un Job fils) devant se connecter à votre bases de données MySQL distante. Pour utiliser dans deux Jobs une connexion partagée à une base de données, procédez comme suit :

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

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



3. Dans la vue **Basic settings** du composant **tMysqlConnection** qui va s'exécuter en premier, saisissez les informations de connexion à la base de données, si la connexion à la base de données n'est pas stockée dans le **Repository**.
4. Cochez la case **Use or register a shared DB Connection** et saisissez un nom pour la connexion dans le champ **Shared DB Connection Name**.



Vous pouvez réutiliser cette connexion dans votre Job fils.

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

 Parmi les différents Jobs partageant une même connexion à une base de données, vous devez configurer les informations de connexion uniquement dans le premier Job ouvrant la connexion à la base de données.

Pour un scénario complet, consultez le scénario du composant **tMysqlConnection** qui montre comment partager une connexion entre différents Jobs à l'adresse <https://help.talend.com>.

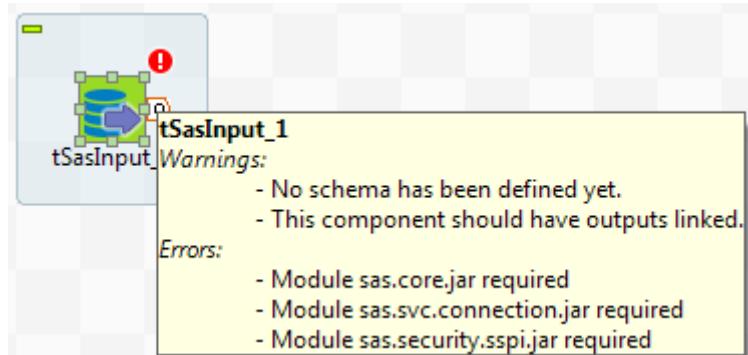
### 5.7.3. Gérer les icônes d'avertissement/d'erreur sur les composants

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**.

### 5.7.3.1. 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.

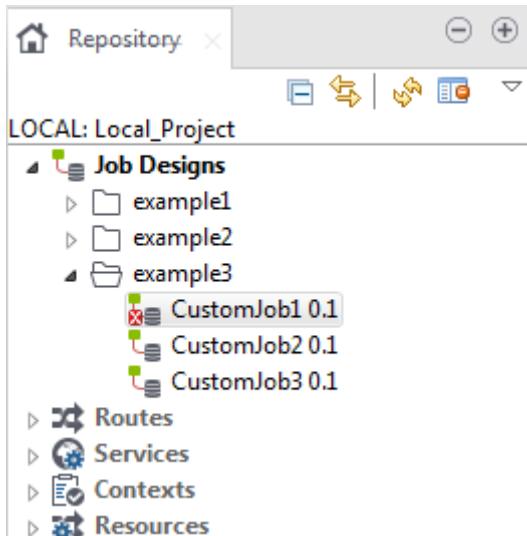
Placez votre souris sur le composant pour faire apparaître les messages d'erreur et d'avertissement. Cette aide contextuelle vous informe sur les données manquantes ou le statut du composant.



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 le *Guide d'installation et de migration Talend*.

### 5.7.3.2. 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 s'affiche sur l'icône du Job à côté de son nom dans le **Repository**.



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

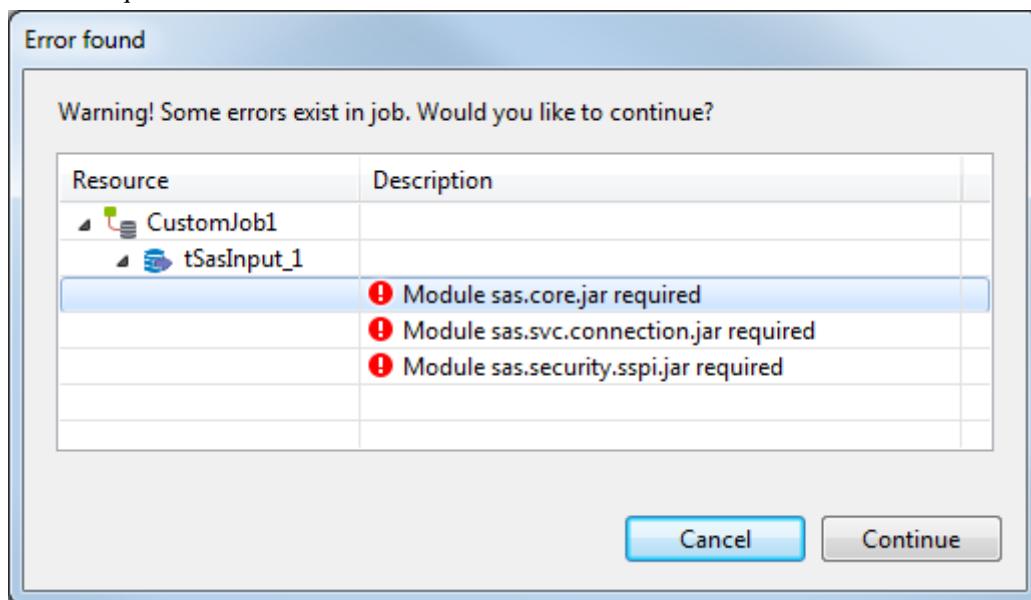
La compilation ou la génération de code a uniquement lieu dans les cas suivants :

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

- l'enregistrement du Job.

L'icône d'erreur s'affiche lorsque vous effectuez l'une de ces actions.

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



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

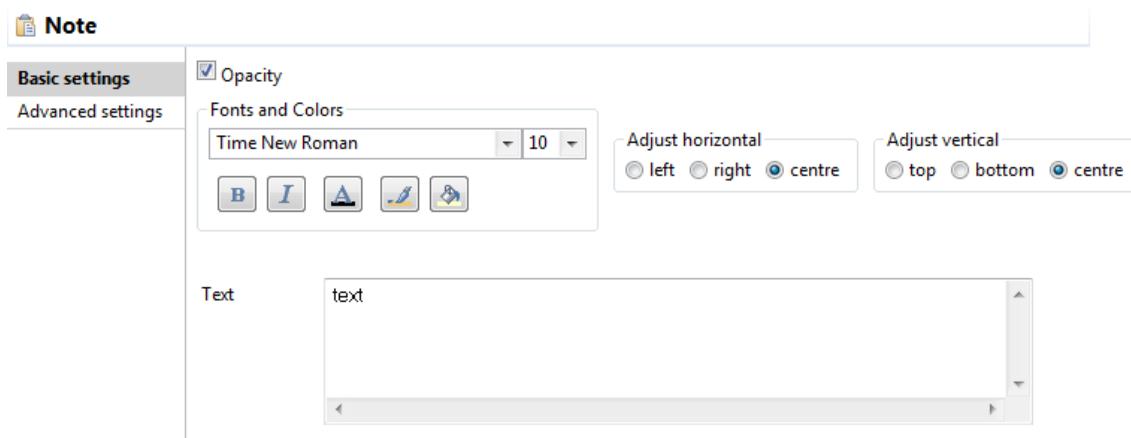
Pour plus d'informations concernant les erreurs dans les composants, consultez [Gérer les icônes d'avertissement/erreur sur les composants](#).

## 5.7.4. 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.

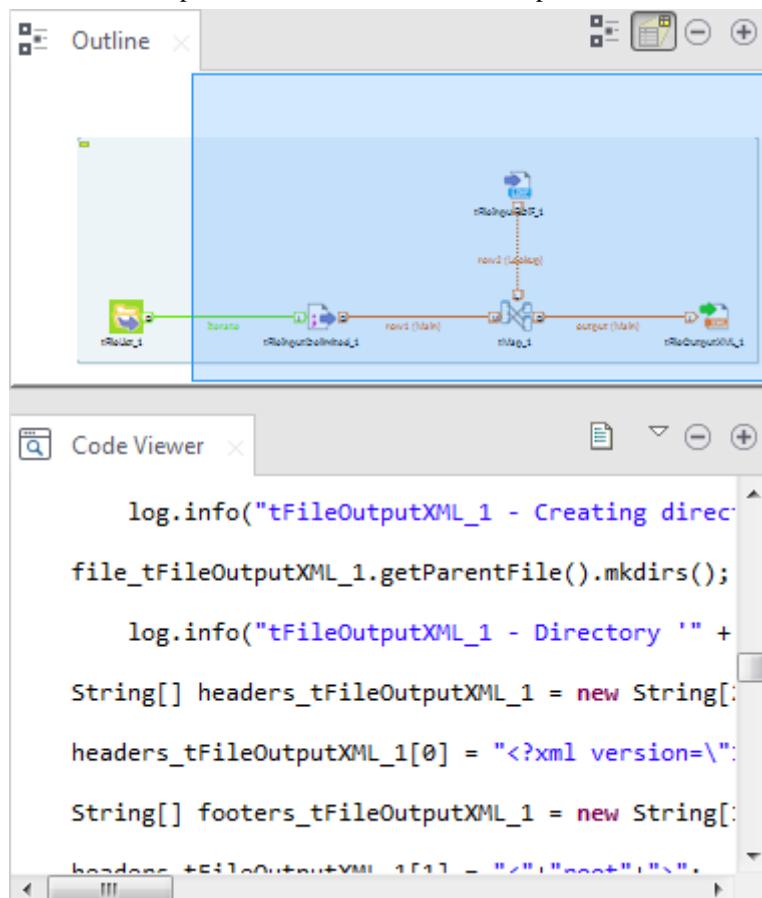
## **5.7.5. Afficher les onglets Code ou Outline de votre Job**

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

### **5.7.5.1. Outline**

L'onglet **Outline** 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.

### 5.7.5.2. 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.

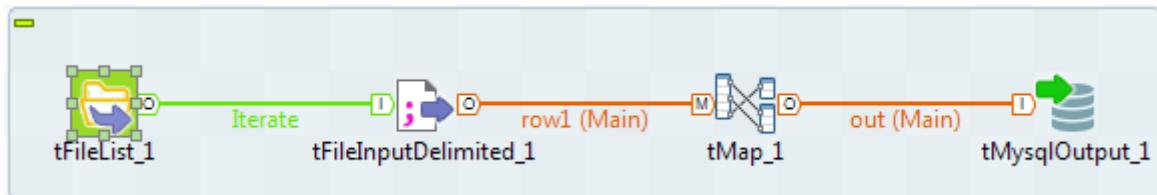


Cette vue ne concerne que le code produit dans un Job Design. En effet, aucun code n'est généré à partir des Business Models.

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.

### 5.7.6. 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.

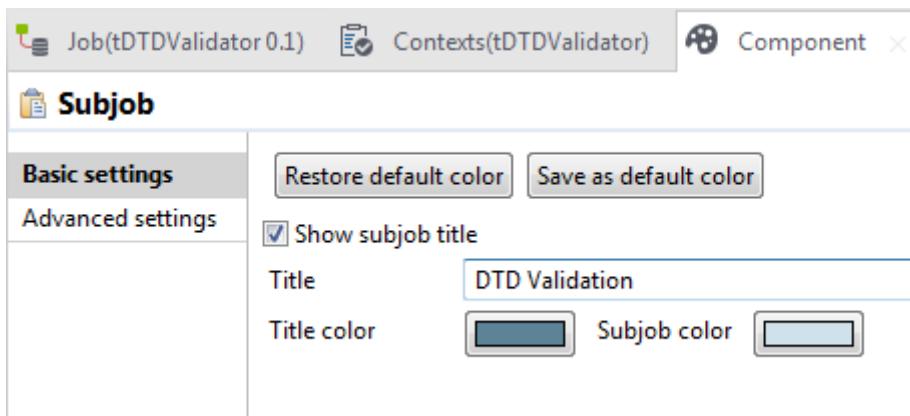


Un Job peut être composé d'un seul sous-job. Les rectangles oranges correspondent aux pré-jobs et aux post-jobs, qui sont différents des sous-jobs.

Pour plus d'informations sur les pré-jobs et les post-jobs, consultez [Utiliser les composants tPrejob et tPostjob](#).

### 5.7.6.1. 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**.



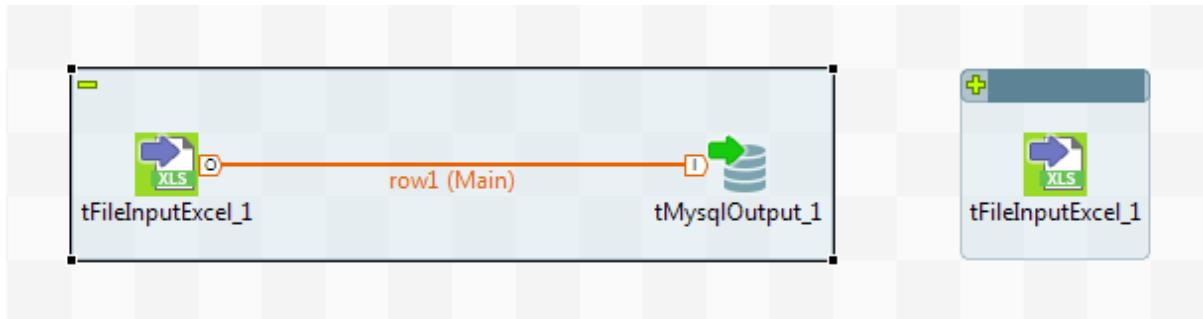
Dans la vue **Basic settings**, cochez la case **Show subjob title** si vous souhaitez ajouter un titre au sous-job et dans le champ **Title**, saisissez le titre.

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]**.
2. Sélectionnez la couleur que vous souhaitez. Par défaut, le titre est bleu et le sous-job bleu transparent.

### 5.7.6.2. 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.



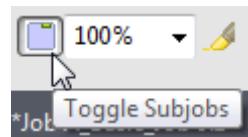
Cliquez sur le signe moins [-] pour réduire le sous-job (capture d'écran de gauche). Quand il est réduit, seul le premier composant du sous-job apparaît.

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

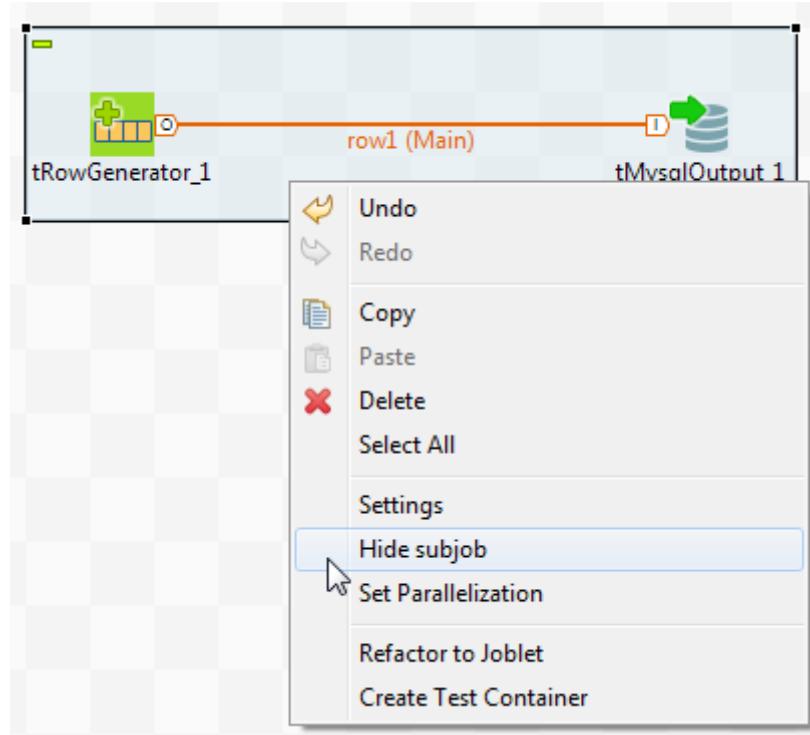
### 5.7.6.3. Supprimer le surlignage d'un sous-job

Si vous ne souhaitez pas 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 du *Studio Talend*.



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



## 5.7.7. Paramétrage des options dans la vue Job

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

L'onglet **Stats & Logs** vous permet d'utiliser de manière automatique les fonctions **Stats & Logs** et la fonction d'automatisation des paramètres de Contexte. Pour plus d'informations, consultez [Automatiser l'utilisation des 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 [Fonctions de l'onglet Extra](#).

### 5.7.7.1. Automatiser l'utilisation des 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.

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.

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 paramètres de projets en cliquant sur le bouton **Save to project settings**. Sinon, vous pouvez accéder à ces paramètres via **File > Edit project settings > 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.

### 5.7.7.2. Fonctions de l'onglet Extra

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

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

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.

Pour un exemple de chargement dynamique de paramètres de contexte à l'aide de la fonctionnalité de chargement implicite des contextes (Implicit Context Load), consultez *Utilisation de la fonctionnalité de chargement implicite des contextes* à l'adresse <https://help.talend.com>.

- 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 [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**.





## Chapitre 6. Gestion des Jobs

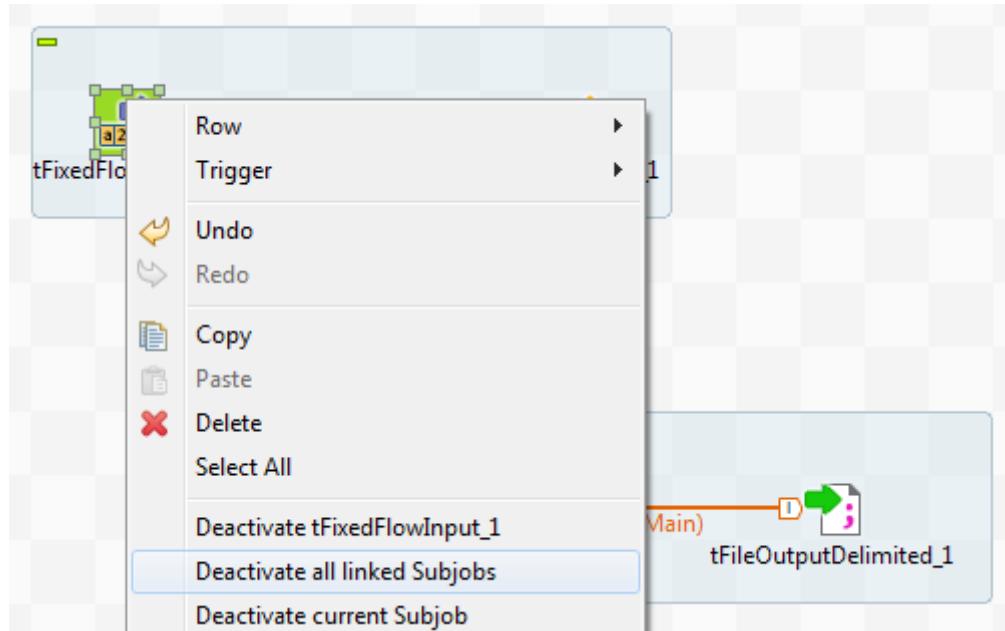
Ce chapitre décrit les procédures de gestion des Jobs créés dans le *Studio Talend*, ainsi que des autres éléments inclus dans un projet, par exemple les routines ou les métadonnées.

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 des Jobs.

## 6.1. Activation/Désactivation d'un composant ou d'un sous-job

Vous pouvez activer ou désactiver un sous-job directement connecté au composant sélectionné. Vous pouvez également activer ou désactiver uniquement un composant ainsi que tous les sous-jobs liés à un composant de début. Le composant de début est le composant déclencheur du Job et a un arrière-plan vert. Pour plus d'informations sur les composants de début, consultez [???](#).

Lorsqu'un composant ou sous-job est désactivé, vous ne pouvez pas créer ou modifier de liens à partir ou vers ce dernier. De plus, à l'exécution, aucun code n'est généré pour le composant, le sous-job désactivé.



### 6.1.1. Activer ou désactiver un composant

Pour activer ou désactiver un composant, procédez comme suit :

1. Cliquez-droit sur le composant que vous souhaitez activer ou désactiver, le **tFixedFlowInput** par exemple.
2. Sélectionnez l'option correspondant à l'action que vous souhaitez effectuer :
  - **Activate tFixedFlowInput\_1** si vous souhaitez l'activer.
  - **Deactivate tFixedFlowInput\_1** si vous souhaitez le désactiver.

### 6.1.2. Activer ou désactiver un sous-job

Pour activer ou désactiver un sous-job, procédez comme suit :

1. Cliquez-droit sur n'importe quel composant formant le sous-job.
2. Sélectionnez l'option correspondant à l'action que vous souhaitez effectuer :

- **Activate current Subjob** si vous souhaitez l'activer.
- **Deactivate current Subjob** si vous souhaitez le désactiver.

### 6.1.3. Activer ou désactiver tous les sous-jobs liés

Pour activer ou désactiver tous les sous-jobs liés, procédez comme suit :

1. Cliquez-droit sur le composant de début.
2. Sélectionnez l'option correspondant à l'action que vous souhaitez effectuer :
  - **Activate all linked Subjobs** si vous souhaitez les activer.
  - **Deactivate all linked Subjobs** si vous souhaitez les désactiver.

## 6.2. Import/export d'éléments et construction de Jobs

Le *Studio Talend* 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 de construire des Jobs et ainsi de déployer et d'exécuter les Jobs créés dans le Studio sur n'importe quel serveur.

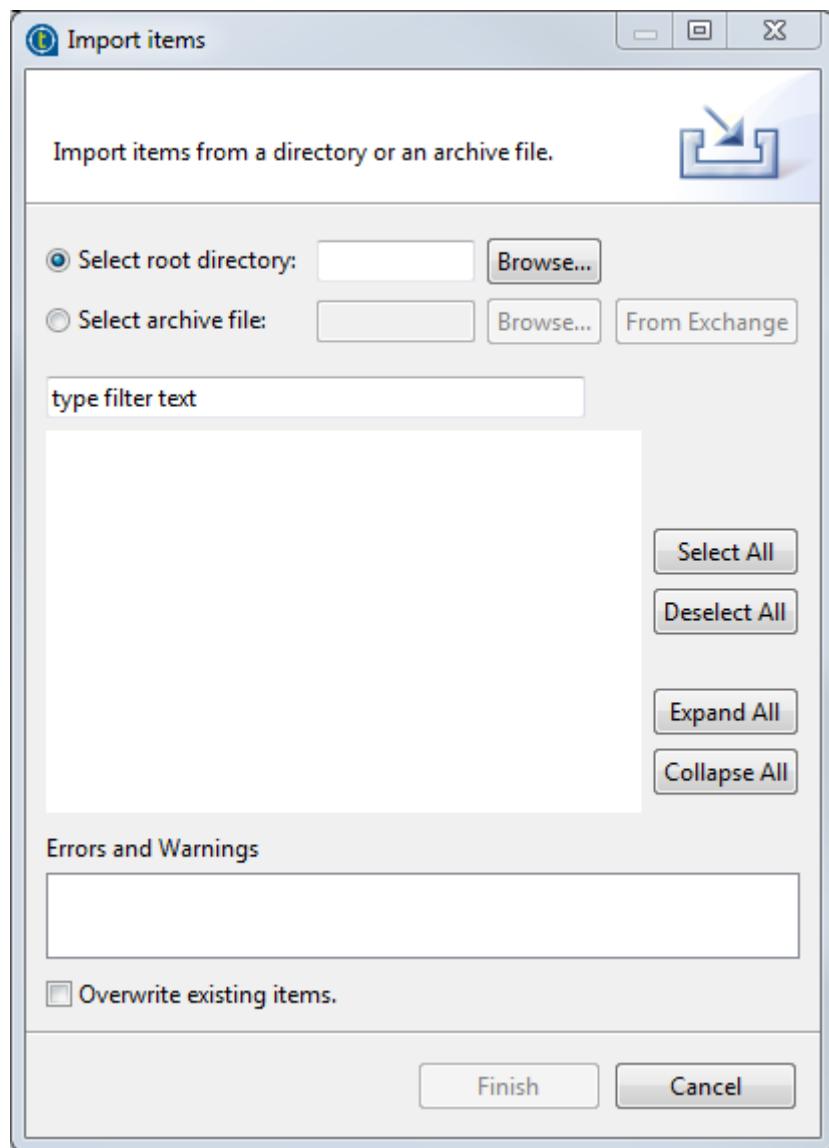
### 6.2.1. Import d'éléments

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

Les éléments que vous pouvez importer sont :

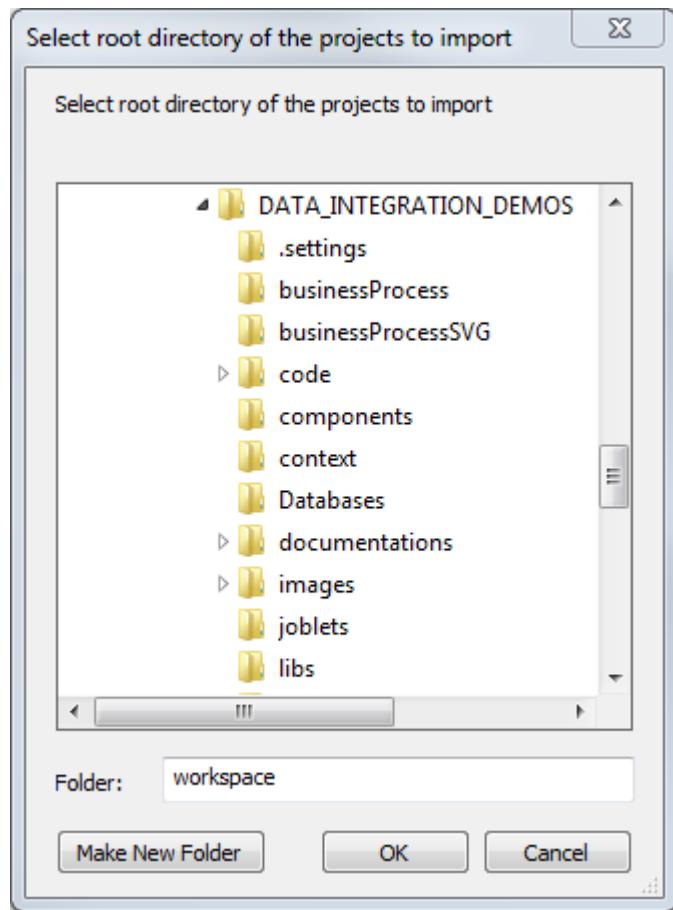
- Business Models
- Jobs Designs
- Routines
- Documentation
- Metadata

Pour importer des éléments, cliquez-droit sur n'importe quelle entrée du **Repository**, notamment **Job Designs** ou **Business Models** puis, dans le menu contextuel, sélectionnez l'option **Import Items**. Vous pouvez également cliquer directement sur le bouton  de la barre d'outils pour ouvrir la boîte de dialogue **[Import items]** puis sélectionner une option d'import.



Pour importer les éléments stockés dans un répertoire local, procédez comme suit :

1. Dans la boîte de dialogue **[Import items]**, sélectionnez l'option **Select root directory**.
2. Cliquez sur **Browse** afin de parcourir l'arborescence de votre workspace jusqu'au dossier portant le même nom que votre projet.



3. Si vous souhaitez importer des éléments en particulier, tels que des **Jobs Designs**, vous pouvez sélectionner le dossier correspondant, par exemple le dossier **Process** où sont stockés tous les Jobs du projet. Si vous n'avez que des Business Models à importer, sélectionnez le dossier **BusinessProcess** puis cliquez sur **OK**.

Cependant, si votre projet contient différents types d'éléments (Business Models, Jobs Designs, Métadonnées, Routines...), il est recommandé de sélectionner le dossier du projet afin d'importer tous les éléments en une fois puis de cliquer sur **OK**.

4. Si nécessaire, cochez la case **overwrite existing items** afin d'écraser les éléments existants par ceux ayant le même nom et devant être importés. Cela rafraîchit la liste **Items List**.
5. A partir de la liste **Items List**, qui contient tous les éléments valides pouvant être exportés, vous pouvez choisir les éléments à importer en cochant les cases correspondantes.
6. Cliquez sur **Finish** pour valider l'import.

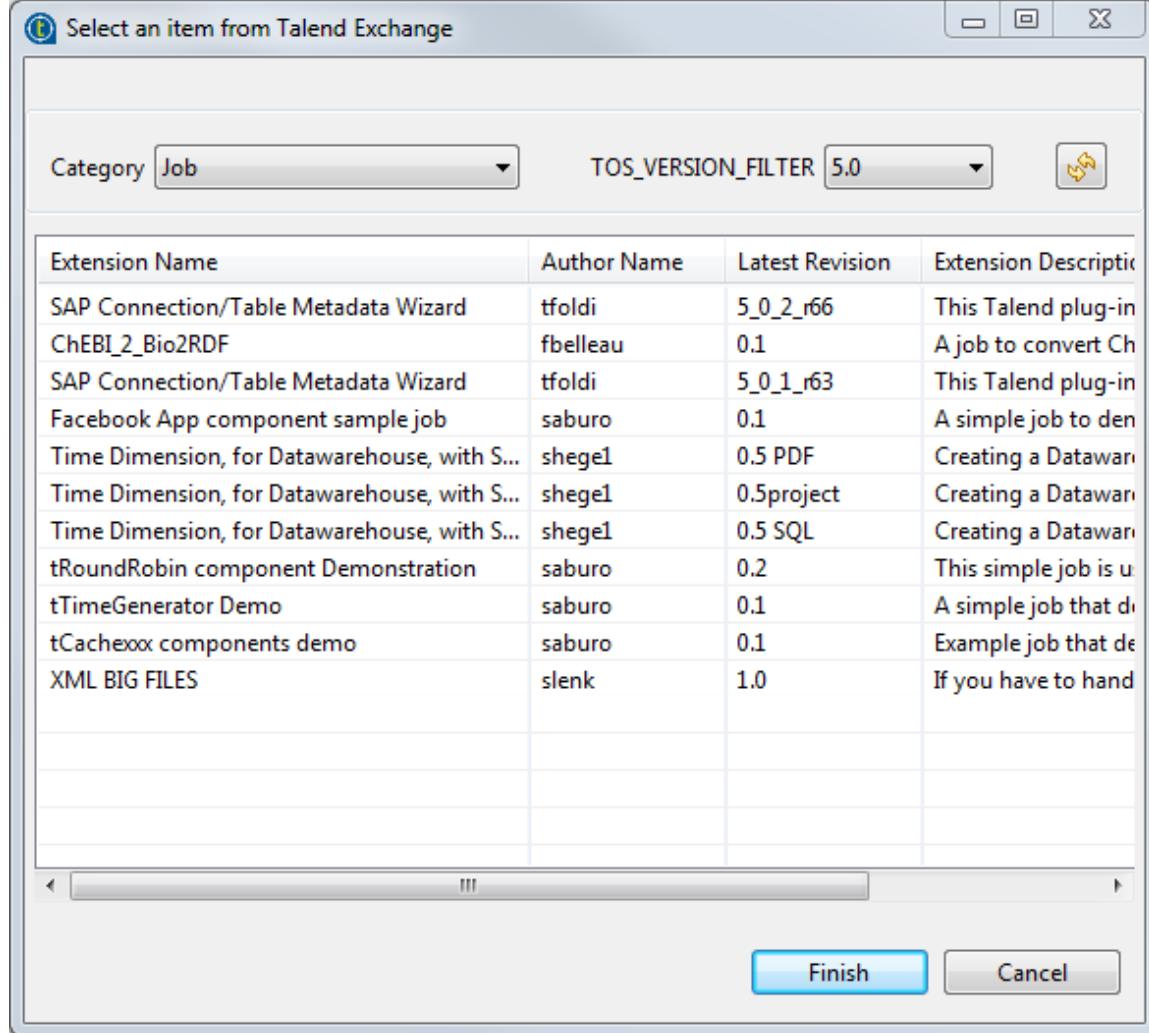
Pour importer des éléments à partir d'une archive (fichiers source et scripts inclus), procédez comme suit :

1. Dans la boîte de dialogue [**Import items**], sélectionnez l'option **Select archive file**.
2. Parcourez votre système jusqu'au fichier archive à importer puis cliquez sur **Open**.
3. Si nécessaire, cochez la case **overwrite existing items** afin d'écraser les éléments existants par ceux ayant le même nom et devant être importés. Cela rafraîchit la liste **Items List**.
4. A partir de la liste **Items List**, qui contient tous les éléments valides pouvant être importés, vous pouvez choisir les éléments à importer en cochant les cases correspondantes.
5. Cliquez sur **Finish** afin de valider l'import.

Pour importer des éléments **Talend Exchange**, procédez comme suit :

1. Dans la boîte de dialogue **[Import items]**, sélectionnez l'option **Select archive file**. Cliquez ensuite sur le bouton **BrowseTalendExchange** afin d'ouvrir la boîte de dialogue **[Select an item from Talend Exchange]**.
2. Dans la liste **Category**, sélectionnez la catégorie à importer puis, dans la liste **TOS\_VERSION\_FILTER**, sélectionnez la version souhaitée.

Une barre de progression s'affiche indiquant que les extensions sont en cours de téléchargement. Une fois le téléchargement terminé, les extensions correspondant à la catégorie et à la version sélectionnées s'affichent dans la boîte de dialogue.



3. Dans la liste, sélectionnez l'extension que vous souhaitez importer.

Cliquez sur **Finish** pour fermer la boîte de dialogue.

4. Si nécessaire, cochez la case **overwrite existing items** afin d'écraser les éléments existants par ceux ayant le même nom et devant être importés. Cela rafraîchit la liste **Items List**.
5. A partir de la liste **Items List**, qui contient tous les éléments valides pouvant être importés, vous pouvez choisir les éléments à importer en cochant les cases correspondantes.
6. Cliquez sur **Finish** afin de valider l'import.



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.

## 6.2.2. Construction de Jobs

La fonction **Build Job** vous permet de déployer et d'exécuter un Job à partir de n'importe quel serveur, indépendamment du *Studio Talend*.

En exécutant des scripts de build générés depuis des modèles définis dans les paramètres du projet, la fonctionnalité de construction de Job ajoute une archive tous les fichiers nécessaires à l'exécution d'un Job, y compris les fichiers .bat et .sh, ainsi que tous les fichiers de paramètres de contextes ou fichiers liés.



Votre *Studio Talend* fournit un ensemble de scripts de construction par défaut. Vous pouvez personnaliser ces modèles afin qu'ils répondent à vos besoins. Pour plus d'informations, consultez [Personnaliser les modèles de scripts de build Maven](#).

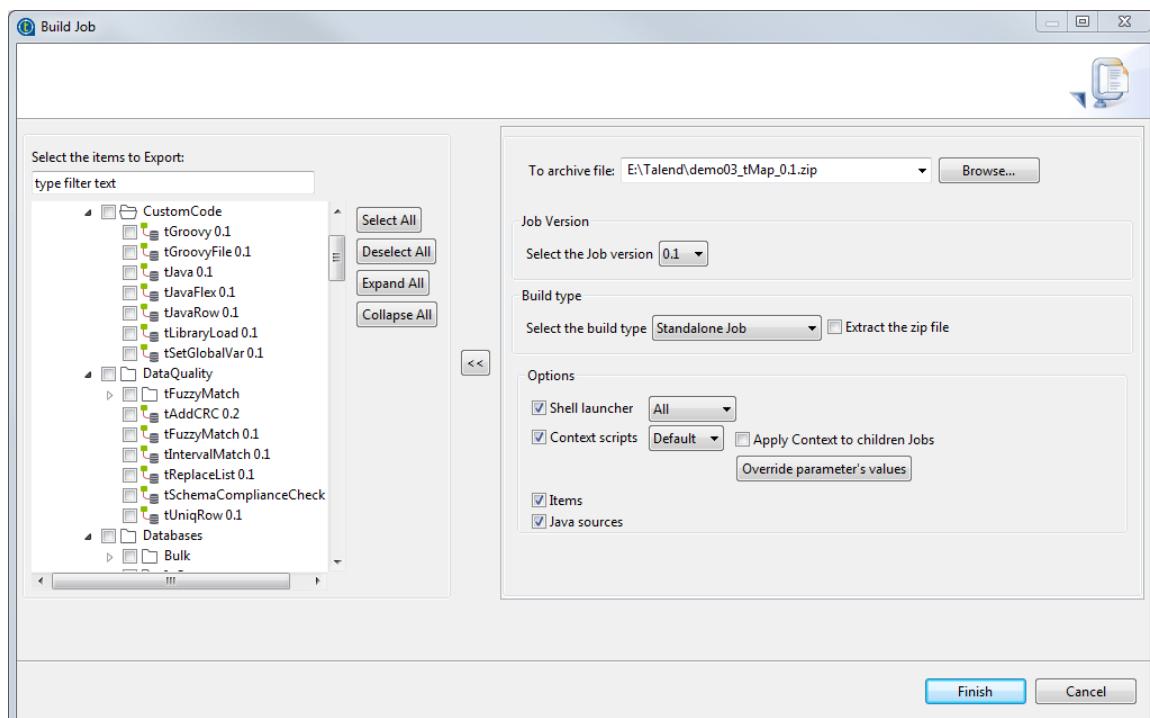
Par défaut, lorsqu'un Job est construit, tous les fichiers Jars requis sont inclus dans la commande .bat ou .sh. Pour un Job complexe impliquant de nombreux Jars, le nombre de caractères dans la commande batch peut dépasser la longueur autorisée par certains systèmes d'exploitation. Pour éviter l'échec de l'exécution de la commande batch à cause de cette limitation, avant de construire votre Job, cliquez sur **Window > Preferences**. Sélectionnez **Talend > Import/Export** et cochez la case **Add classpath jar in exported jobs**, afin de regrouper les Jars dans un fichier *classpath.jar* ajouté au Job construit.

Pour construire des Jobs, procédez comme suit :

1. Dans l'arborescence **Repository**, cliquez-droit sur le Job que vous voulez construire et sélectionnez **Build Job** pour ouvrir la boîte de dialogue [**Build Job**].



Vous pouvez directement afficher ou cacher, dans la boîte de dialogue [**Build Job**], l'arborescence de tous les Jobs ayant été créés dans le *Studio Talend*, 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 le *Studio Talend*.



2. Dans le champ **To archive file**, cliquez sur le bouton **Browse...** et parcourez jusqu'au répertoire dans lequel vous souhaitez sauvegarder votre Job construit.
3. Dans la zone **Job Version**, sélectionnez la version du Job que vous souhaitez construire, si vous en avez créé plusieurs versions.

4. Dans la liste **Build Type**, sélectionnez le type de construction. Les différents types sont : **Standalone Job**, **Axis WebService (WAR) (Deprecated)**, **Axis WebService (ZIP) (Deprecated)** et **OSGI Bundle For ESB**.

Si le Job de services de données inclut le composant **tRESTClient** ou le **tESBConsumer** et qu'aucun Service Registry, Service Locator ou Service Activity Monitor n'est activé pour ce composant, le Job de services de données peut être construit en tant que bundle OSGI pour ESB ou en tant que Job Standard. Si le Service Registry, le Service Locator ou le Service Activity Monitor est activé, le Job de services de données incluant le **tRESTClient** ou le **tESBConsumer** ne peut être construit qu'en tant que bundle OSGI pour ESB.

5. Cochez la case **Extract the zip file**, si vous souhaitez extraire automatiquement le fichier dans le répertoire cible.
6. Dans la zone **Options**, sélectionnez le type de fichiers que vous souhaitez ajouter à l'archive. Les options correspondant aux fichiers nécessaires à l'exécution du Job sont cochées par défaut, vous pouvez néanmoins les décocher en fonction de ce que vous souhaitez construire.

| Option                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Shell launcher</b>    | Cochez cette case pour exporter les fichiers .bat et/ou .sh nécessaires au lancement du Job construit. <ul style="list-style-type: none"> <li>• <b>All</b> : exporte les fichiers .bat et .sh,</li> <li>• <b>Unix</b> : exporte le fichier .sh,</li> <li>• <b>Windows</b> : exporte le fichier .bat.</li> </ul>                                                                                                                                                                                                                                                           |
| <b>Context scripts</b>   | Cochez cette case pour exporter TOUS les fichiers de paramètres de contexte et pas uniquement ceux sélectionnés dans la liste correspondante. <p> Pour exporter un contexte, sélectionnez le contexte répondant à vos besoins dans la liste <b>Context scripts</b>, 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.</p> |
| <b>Apply to children</b> | Cochez cette case si vous souhaitez appliquer le contexte sélectionné dans la liste à tous les Jobs fils.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Items</b>             | Cochez cette case pour exporter les sources utilisées par le Job durant son exécution, y compris les fichiers .item et .properties, ainsi que les sources Java et <b>Talend</b> . <p> Si vous cochez la case <b>Items</b> ou <b>Source files</b>, vous pouvez réutiliser le Job construit dans un <i>Studio Talend</i> installé sur une autre machine. Ces fichiers source sont uniquement utilisés dans le <i>Studio Talend</i>.</p>                                                  |
| <b>Java sources</b>      | Cochez cette case pour exporter le fichier .java contenant les classes Java générées par le Job lors de sa création.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

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

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

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

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



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

### 6.2.2.1. Construire un Job en tant que Job standalone

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

Lorsque que plusieurs contextes sont proposés dans la liste **Context script**, sélectionnez celui qui correspond à vos besoins. Ensuite, cliquez sur le bouton **Override parameters' values**, sous la case **Context 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.

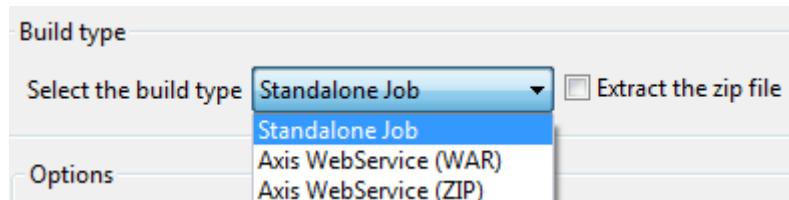
Tous les fichiers de paramètres de contexte sont exportés et pas uniquement celui que vous avez sélectionné dans la liste.



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

### 6.2.2.2. Construire un Job en tant que WebService

Dans la boîte de dialogue **[Build Job]**, vous pouvez modifier le type de construction afin de construire 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.               |

| Paramètres de l'URL      | Description                                          |
|--------------------------|------------------------------------------------------|
| ?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 de construction de Job en tant que service Web et de déploiement de ce Job à partir d'un navigateur Web, consultez [Exemple de construction de Job en tant que service Web](#).

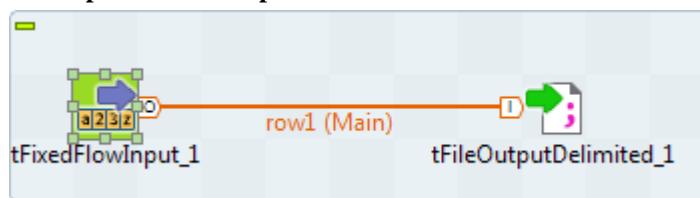
Le composant **tBufferOutput** a été conçu spécialement pour ce type de déploiement. Pour plus d'informations concernant ce composant, consultez la documentation du composant **tBufferOutput**.

### 6.2.2.3. Exemple de construction de Job en tant que service Web

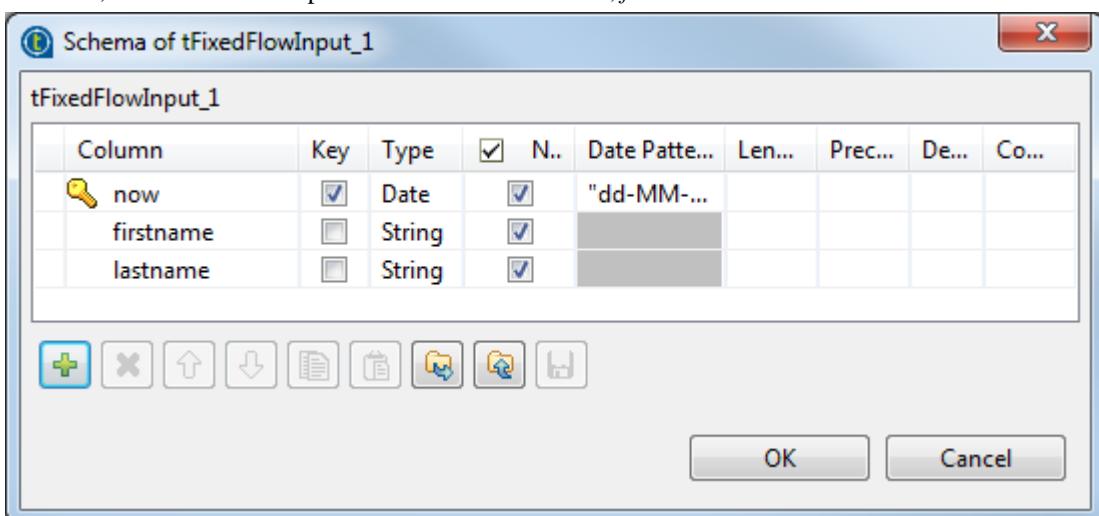
Ce scénario décrit, dans un premier temps, un simple Job créant un fichier .txt contenant la date en cours, ainsi que des noms et prénoms. Dans un deuxième temps, ce Job est construit en tant que service Web à partir d'un navigateur Web. Le Job construit en tant que service Web renverra simplement le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

#### Créer le Job

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



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



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

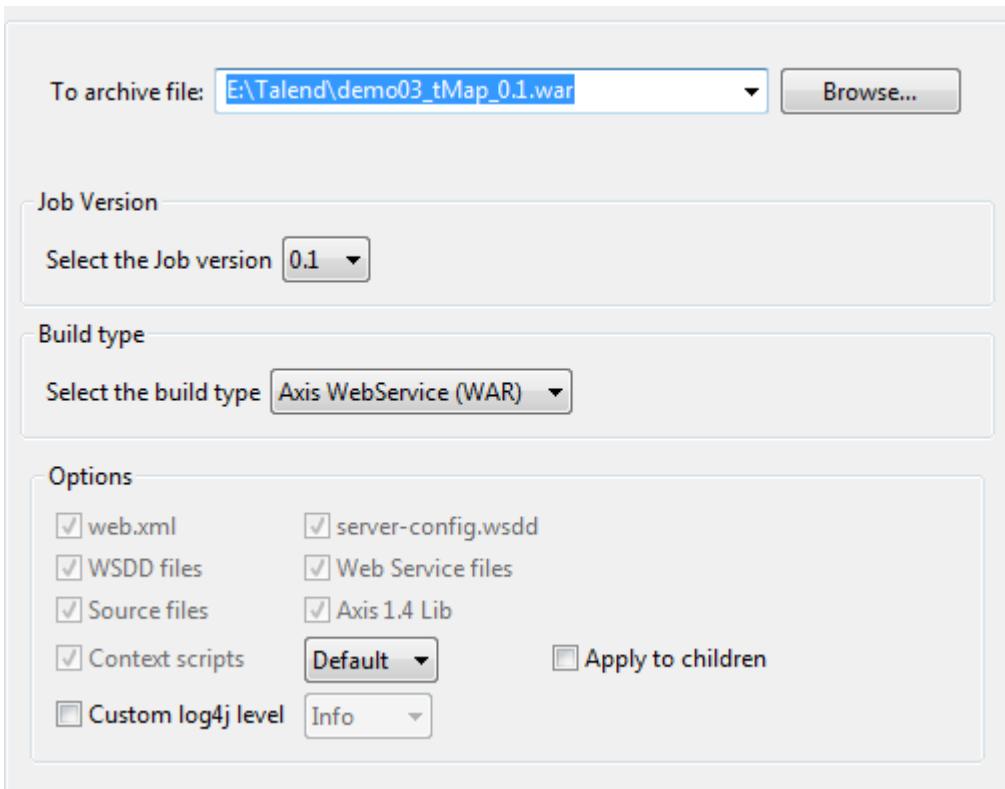
| Column    | Value                              |
|-----------|------------------------------------|
| now       | TalendDate.getCurrentDate()        |
| firstname | TalendDataGenerator.getFirstName() |
| lastname  | TalendDataGenerator.getLastName()  |

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.

## Construire le Job en tant que service Web

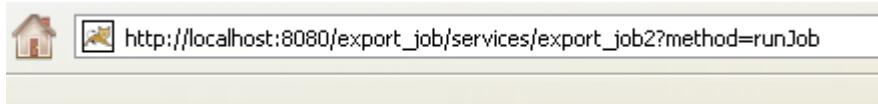
1. Dans la vue **Repository**, cliquez-droit sur le Job que vous venez de créer et sélectionnez **Build Job**. La boîte de dialogue **[Build Job]** apparaît :



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 construire, si vous avez créé plusieurs versions de votre Job.
4. Dans la liste **Build type**, sélectionnez le type de construction que vous souhaitez utiliser dans votre application Web (WAR pour cet exemple) et cliquez sur **Finish**. La boîte de dialogue [Build Job] disparaît.
5. Copiez le dossier War et collez-le dans le répertoire du serveur d'application Tomcat.

### 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

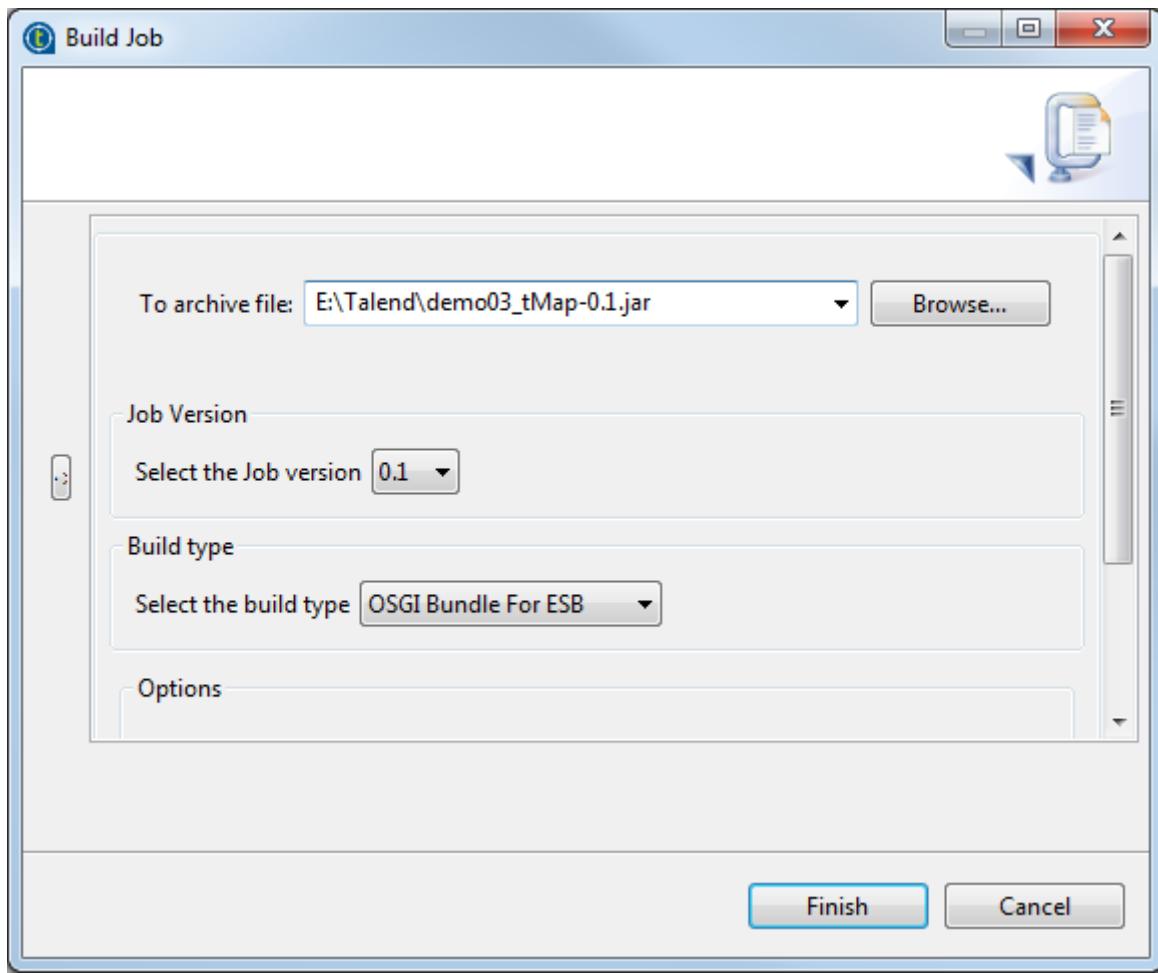
```
- <soapenv:Envelope>
 - <soapenv:Body>
 - <runJobReturn xsi:type="ns1:runJobReturn">
 - <ns1:item xsi:type="ns1:ArrayOf_xsd_string">
 <ns1:item xsi:type="xsd:string">0</ns1:item>
 </ns1:item>
 </runJobReturn>
 </soapenv:Body>
</soapenv:Envelope>
```

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 de construction d'un Job en tant que service Web via le composant **tBufferOutput**, consultez la documentation du composant **tBufferOutput**.

#### 6.2.2.4. Construire un Job en tant que bundle OSGI pour ESB

Dans la boîte de dialogue **[Build Job]**, vous pouvez changer le type de construction pour construire le Job sélectionné en tant que bundle OSGI, afin de déployer votre Job dans votre conteneur **Talend ESB Container**.



1. Dans la zone **Job Version**, sélectionnez le numéro de version du Job que vous souhaitez construire si vous en avez créé plus d'une version.
2. Dans la zone **Build type**, sélectionnez **OSGI Bundle For ESB** afin de construire votre Job en tant que bundle OSGI.

L'extension de votre construction change automatiquement en *.jar* puisque c'est l'extension attendue par **Talend ESB Container**.

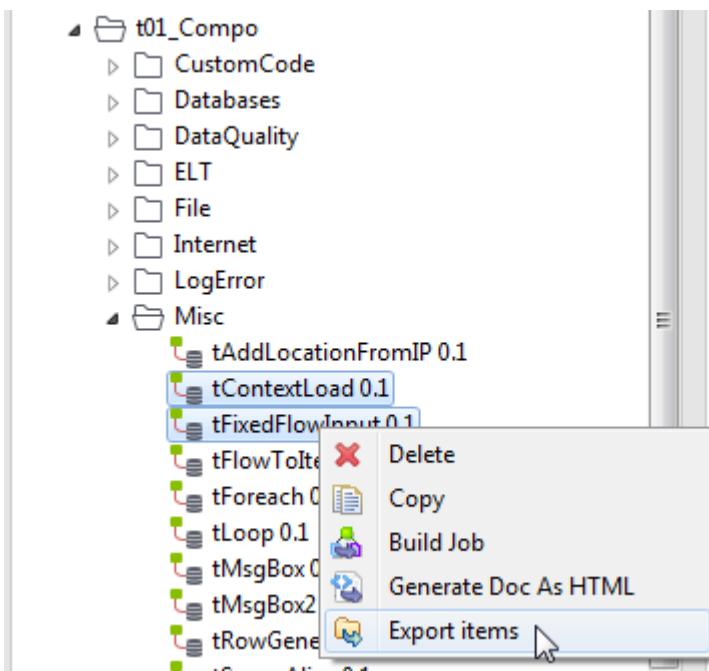
3. Cliquez sur le bouton **Browse...** pour spécifier le dossier dans lequel construire votre Job.
4. Cliquez sur **Finish** afin de le construire.

### 6.2.3. 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.

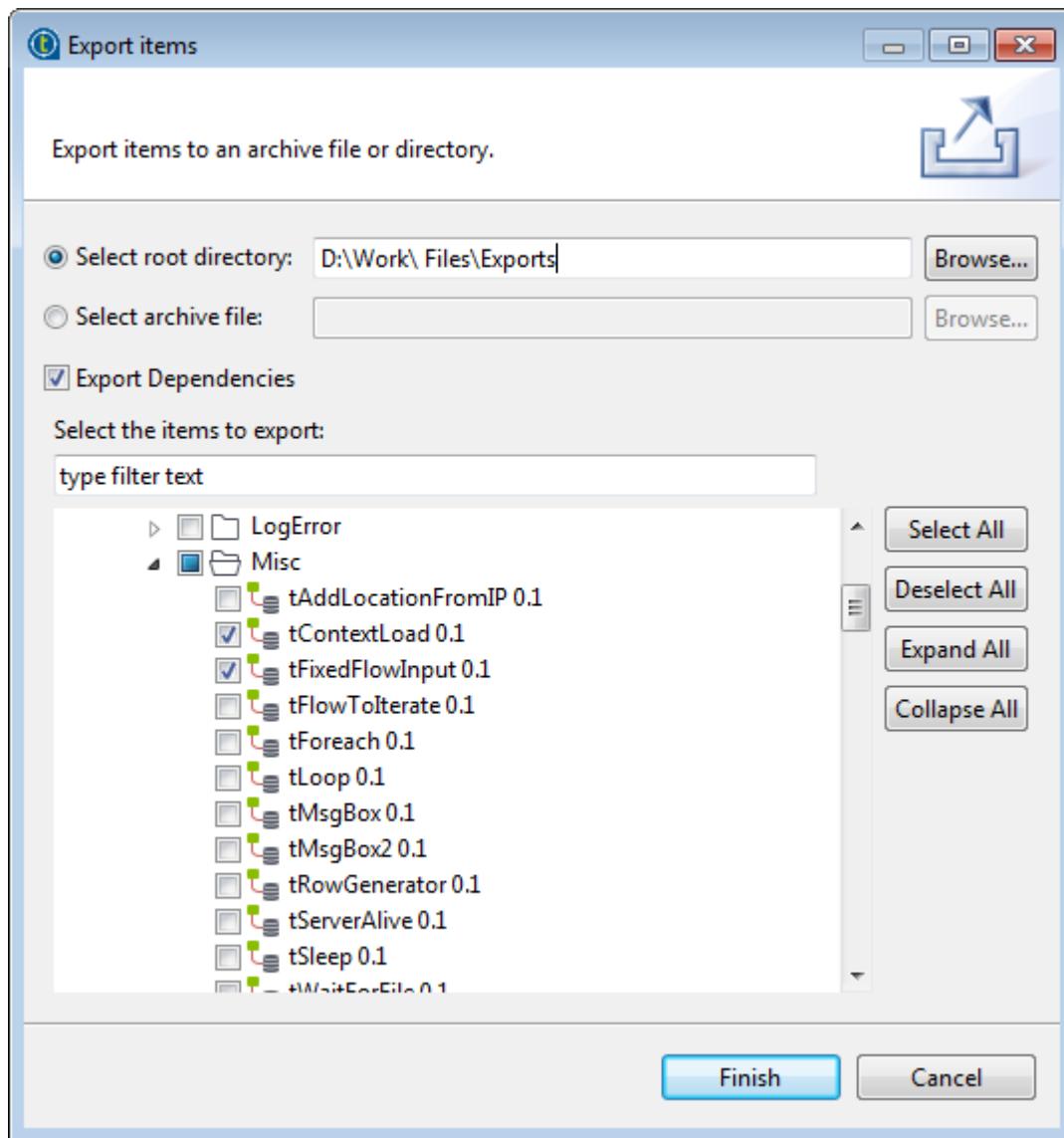
Pour ce faire :

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 connexion à la base de données et pas uniquement la table souhaitée, afin que le processus d'export se déroule correctement.*

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



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

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



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



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

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

## 6.2.4. Changer les paramètres de contexte dans des Jobs

Comme expliqué dans [Construction de Jobs](#), vous pouvez éditer les paramètres de contexte :

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

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

| Pour...                                                                                       | Paramètre                                                              |
|-----------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Changer la <i>valeur1</i> pour le paramètre <i>key1</i>                                       | <code>--context_param key1=valeur1</code>                              |
| Changer <i>valeur1</i> et <i>valeur2</i> des paramètres respectifs <i>key1</i> et <i>key2</i> | <code>--context_param key1=valeur1 --context_param key2=valeur2</code> |
| Changer une valeur contenant des caractères d'espace comme dans les chemins d'accès           | <code>--context_param key1="chemin d'accès"</code>                     |

## 6.3. Gestion des éléments du Repository

Le *Studio Talend* vous permet d'éditer les éléments centralisés dans le Repository et de mettre à jour les Jobs qui utilisent ces éléments.

### 6.3.1. Gérer les mises à jour des éléments du Repository

Vous pouvez mettre à jour les paramètres des métadonnées, des contextes centralisés à chaque fois dans la vue **Repository**, afin d'actualiser la connexion à la base de données ou par exemple les détails du groupe de contexte.

Lorsque vous modifiez l'un des paramètres d'une entrée de la vue **Repository**, tous les Jobs utilisant cette entrée seront impactés par cette modification. C'est pourquoi le système vous proposera de propager ces modifications à tous les Jobs utilisant l'entrée du 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.

#### 6.3.1.1. Modifier un élément du Repository

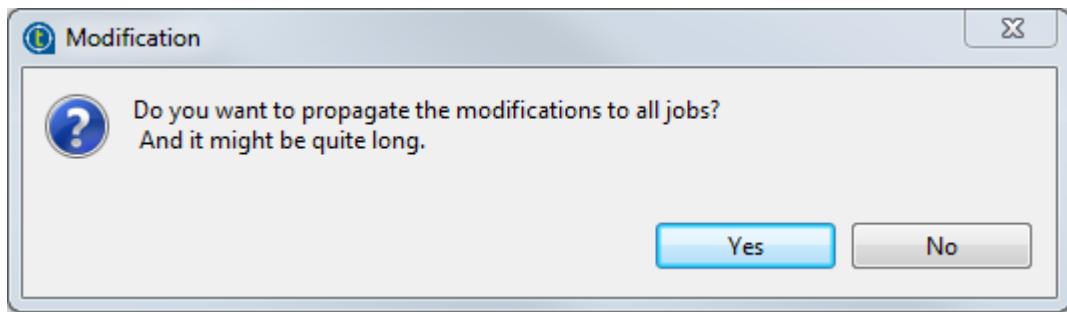
Pour actualiser les paramètres d'un élément du Repository, procédez comme suit :

1. Développez le nœud **Metadata, Contexts** dans la vue **Repository** et parcourez jusqu'à l'entrée correspondante que vous voulez actualiser.
2. Cliquez-droit sur l'entrée et sélectionnez dans le menu contextuel l'option **edit** correspondante.

L'assistant respectif apparaît, vos permettant d'éditer chaque étape de la définition des paramètres d'entrée.

Pour mettre à jour les paramètres d'entrée, vous devez propager les modifications de plusieurs Jobs ou de tous les Jobs utilisant cette entrée.

Une boîte de dialogue apparaît automatiquement à la fin de votre mise à jour ou de vos modifications, quand vous cliquez sur le bouton **Finish** dans l'assistant.



- Cliquez sur **Yes** (Oui) pour fermer la boîte de dialogue et exécuter les modifications apportées à tous les Jobs concernés. Pour plus d'informations concernant la première méthode de propagation des modifications, consultez [Mettre à jour manuellement les Jobs impactés](#).

Cliquez sur **No** (Non) si vous voulez fermer la boîte de dialogue sans propager les changements. Cela vous permettra de propager manuellement, une par une, les modifications apportées aux Jobs concernés. Pour plus d'informations concernant la deuxième méthode de propagation des modifications, consultez [Mettre à jour manuellement les Jobs impactés](#).

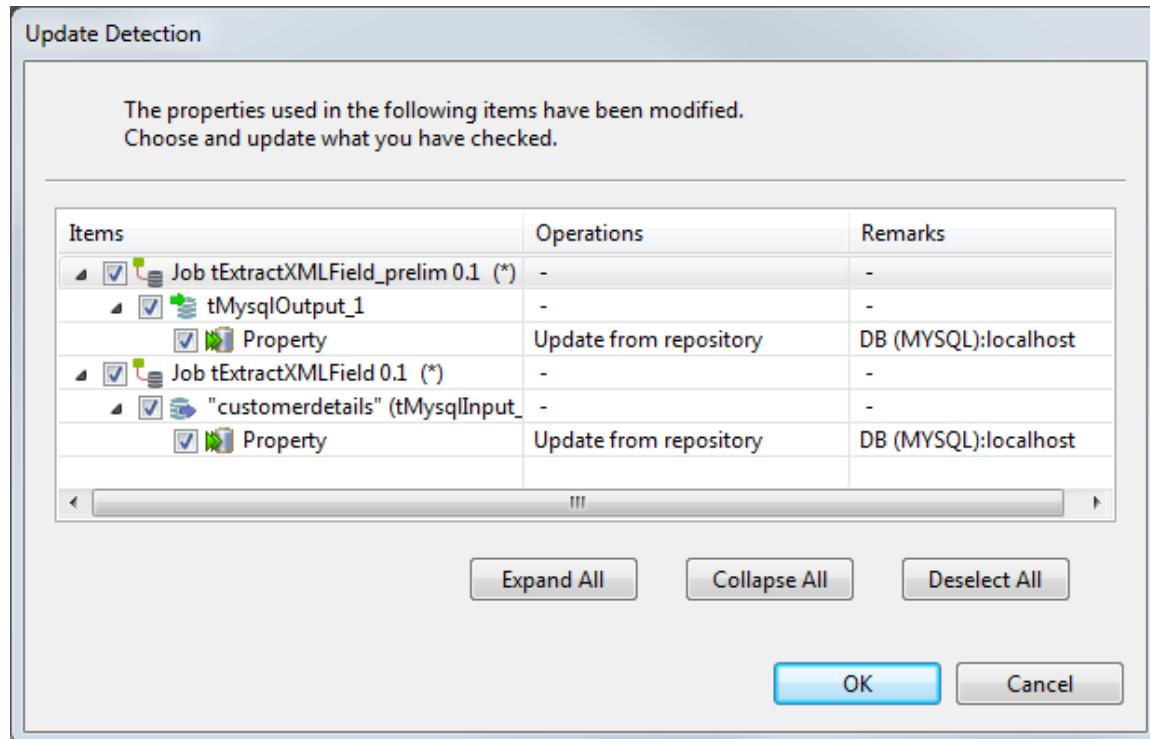
### 6.3.1.2. Mettre à jour automatiquement les Jobs impactés

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 [Mettre à jour manuellement les Jobs impactés](#).

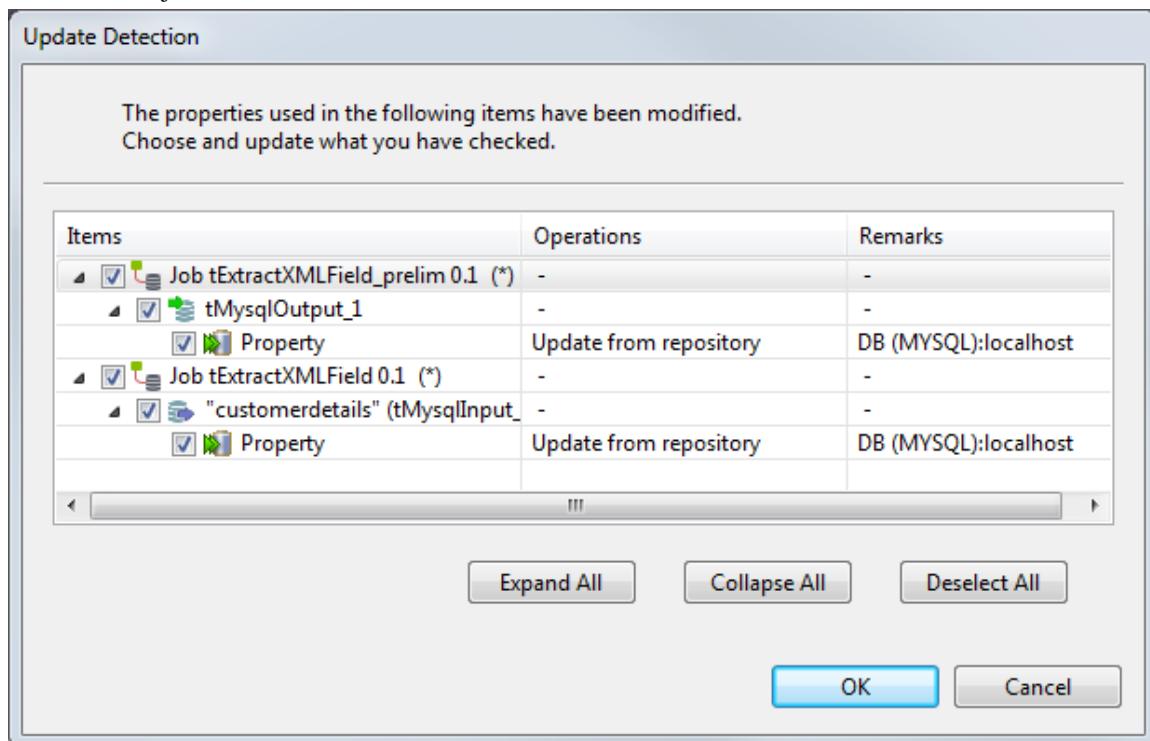
2. DÉCOchez la case correspondant aux Jobs que vous ne souhaitez pas mettre à jour. Vous pouvez les mettre à jour manuellement ultérieurement. Pour plus d'informations, consultez [Mettre à jour manuellement les Jobs impactés](#).
3. Cliquez sur **OK** pour fermer la boîte de dialogue et mettre à jour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

### 6.3.1.3. 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.



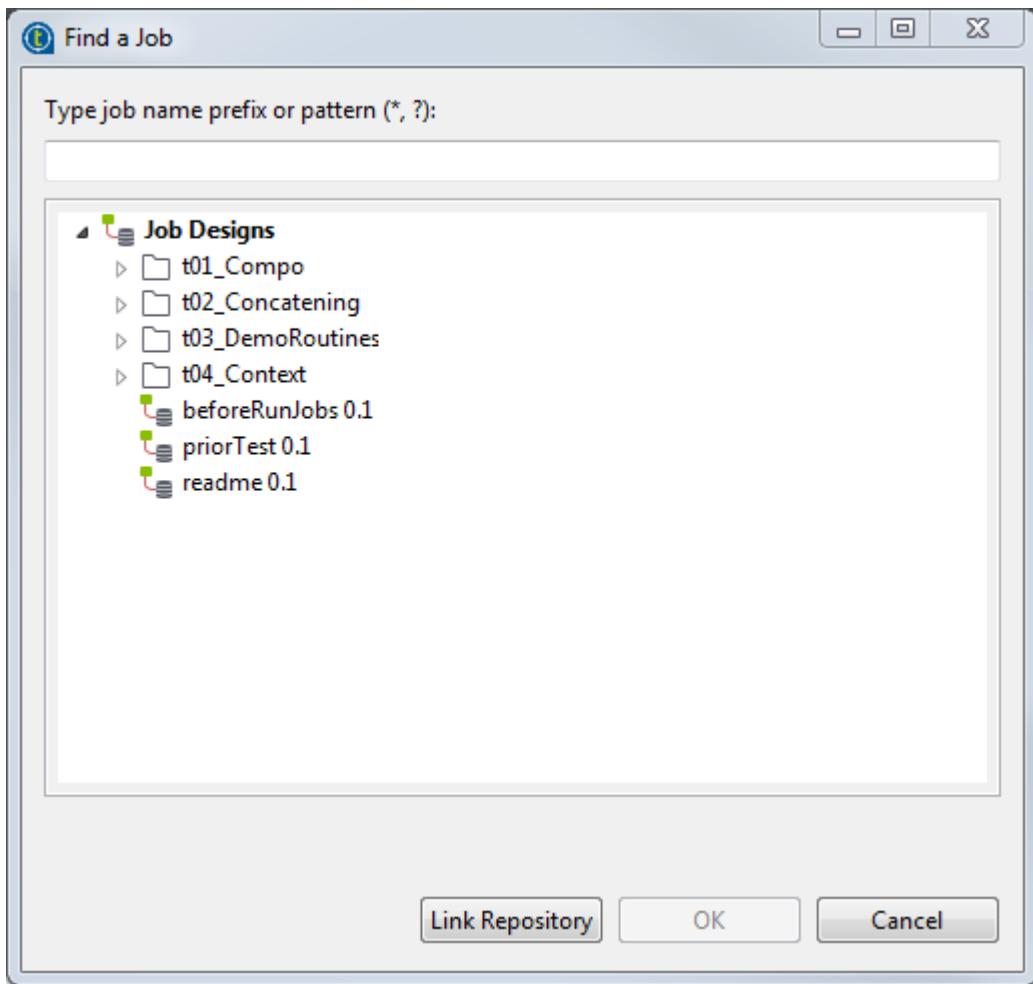
Les Jobs que vous n'aurez pas mis à jour passeront automatiquement en **Built-in**, puisque le lien vers le **Repository** ne pourra pas être maintenu. Cependant, les paramètres resteront les mêmes que ceux définis avant les modifications.

## 6.4. Recherche d'un Job dans le référentiel

Si vous souhaitez ouvrir un Job précis dans le *Studio Talend* et que vous n'arrivez pas à le retrouver dans la vue **Repository**, cliquez simplement sur le bouton  de la barre d'outils.

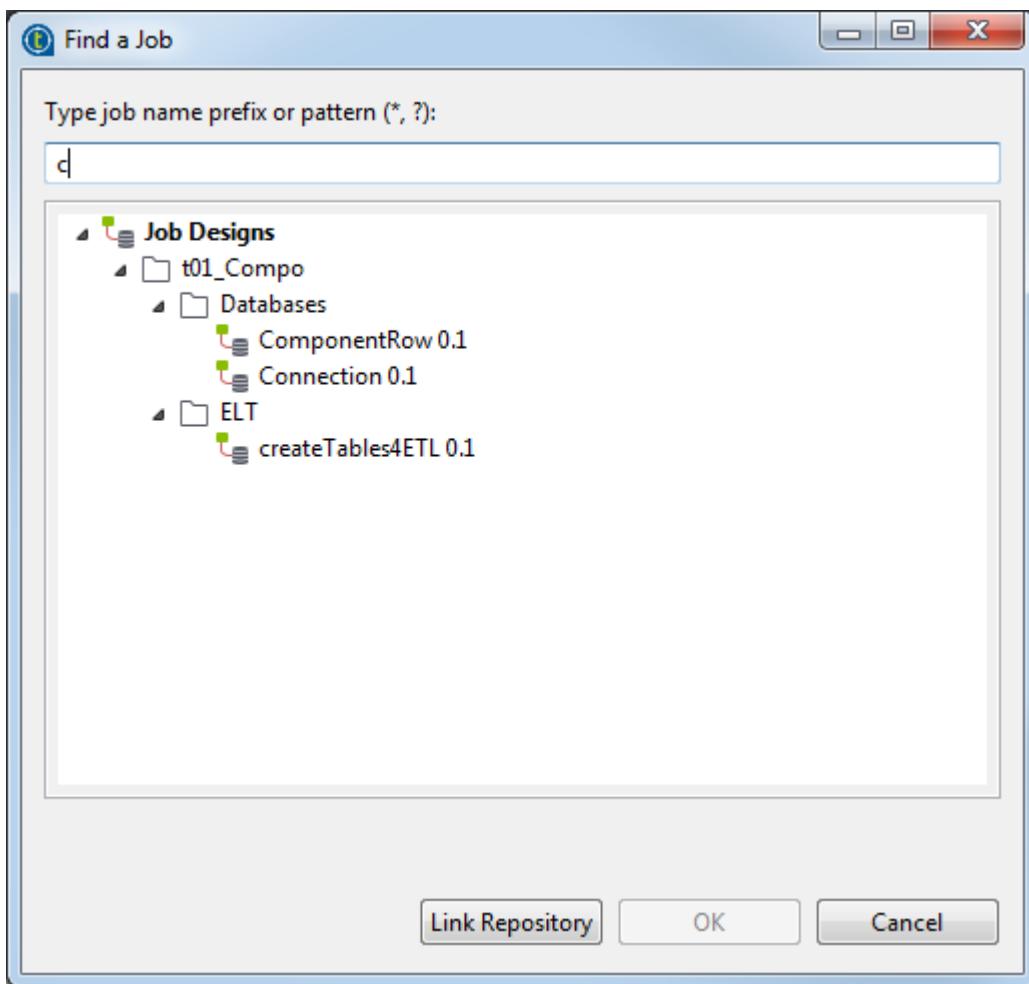
Pour chercher un Job dans la vue **Repository** :

1. Dans la barre d'outils du *Studio Talend*, 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 Job(s) dont le nom correspond aux lettres saisis.



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

## 6.5. Gestion des versions d'un Job

Lorsque vous créez un Job dans le *Studio Talend*, son numéro de version par défaut est 0.1 : 0 correspond à la version majeure et 1 à la version mineure.

Vous pouvez créer autant de versions d'un même Job que vous le souhaitez. Pour ce faire :

1. Veillez à fermer votre Job s'il est ouvert dans l'espace de modélisation graphique. Autrement, ses propriétés seront en lecture seule et vous ne pourrez donc pas les modifier.
2. Cliquez-droit sur le libellé du Job dans le **Repository** et sélectionnez **Edit properties** à partir du 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 du Job 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. Autrement 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** à partir de 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 le Job actif et incrémenter sa version simultanément, en cliquant sur **File>Save As...** et indiquer une nouvelle version dans la boîte de dialogue **[Save As]**.



Donner un nouveau nom à votre Job n'écrase pas la version actuellement ouverte, mais l'enregistre en tant que nouveau Job avec la même version que le Job actuel ou une nouvelle version si vous en spécifiez une.

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. Dans le panneau des onglets de configuration, cliquez sur l'onglet **Job** et sélectionnez **Version** 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...                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Edit Job</b>             | ouvrir la dernière version du Job .<br><br> Cette option n'est disponible que lorsque vous sélectionnez la dernière version du Job .                                                                                                                         |
| <b>Read job</b>             | consulter le Job en lecture seule.                                                                                                                                                                                                                                                                                                              |
| <b>Open Job Hierarchy</b>   | consulter la hiérarchie du Job.                                                                                                                                                                                                                                                                                                                 |
| <b>Edit properties</b>      | éditer les propriétés du Job .<br><br><b>Note :</b> le Job ne doit pas être ouvert dans l'espace de modélisation graphique, sinon il sera en lecture seule.<br><br> Cette option n'est disponible que lorsque vous sélectionnez la dernière version du Job . |
| <b>Run job</b>              | exécuter le Job.                                                                                                                                                                                                                                                                                                                                |
| <b>Generate Doc As HTML</b> | générer le détail des informations du Job .<br><br> Cette option n'est disponible que pour un Job d'intégration et non pour une Route.                                                                                                                       |

Vous pouvez également gérer la version de plusieurs Jobs et/ou métadonnées simultanément, ainsi que les Jobs et leurs dépendances et/ou leur(s) Job(s) fils, dans les paramètres des projets. Pour plus d'informations, consultez [Gérer les versions](#).

## 6.6. Documentation du Job

Le *Studio Talend* 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.

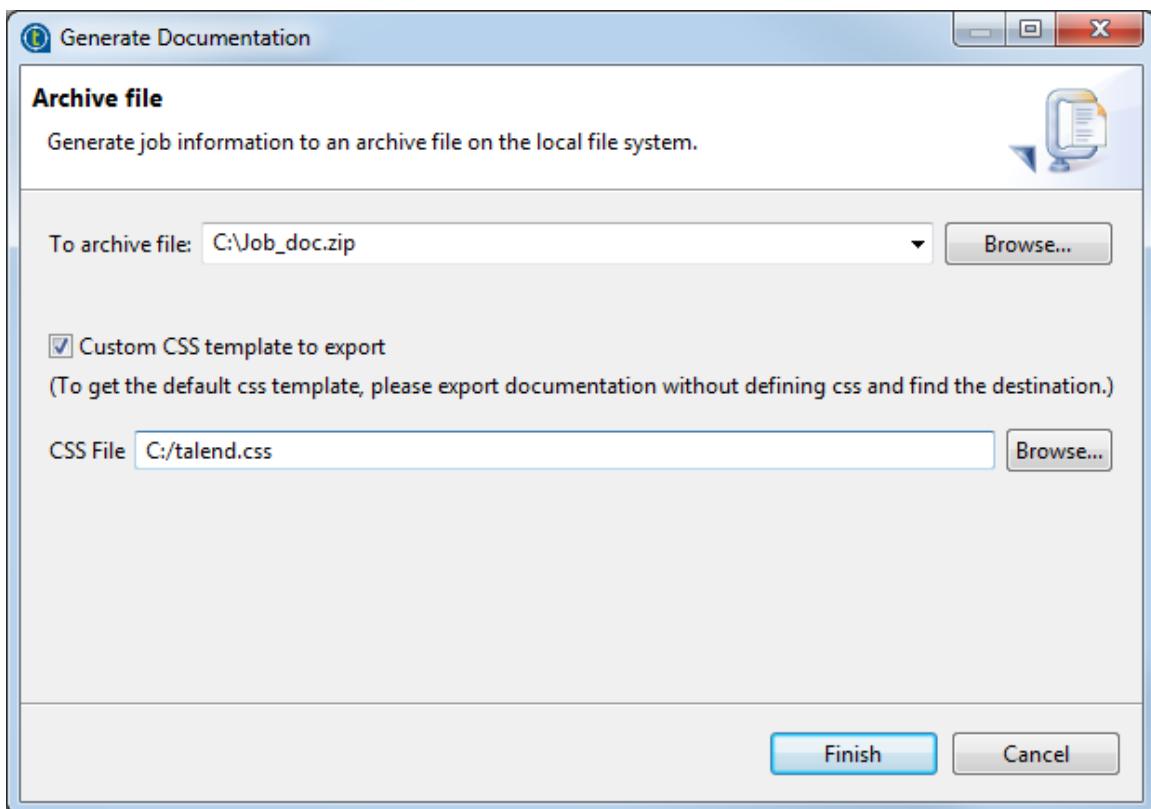
### 6.6.1. Générer la documentation HTML

Le *Studio Talend* vous permet de générer de la documentation détaillée au format HTML des Jobs sélectionnés dans le **Repository** de votre Studio, dans la perspective **Integration**. 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 éléments pour générer une documentation multiple.
2. Sélectionnez **Generate Doc as HTML** dans le menu contextuel.



3. Renseignez le répertoire dans lequel doit être enregistré la documentation générée.
4. Dans le même champ, renseignez le nom de l'archive regroupant les documents générés.
5. Si vous souhaitez utiliser un fichier CSS, cochez la case **Use CSS file as a template to export** afin d'activer le champ **CSS File**.

6. Dans le champ **CSS File**, parcourez votre système ou saisissez le chemin vers le fichier CSS utilisé.
7. Cliquez sur **Finish** pour valider.

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

## 6.6.2. 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**.

## 6.7. Exécution des Jobs

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

Cette section présente :

- *Exécuter un Job en mode normal.*
- *Exécuter un Job en mode Java debug.*
- *Exécuter un Job en mode Traces Debug.*
- *Configurer les paramètres d'exécution avancés.*
- *Monitorer l'utilisation des ressources de la JVM durant une exécution de Job*
- *Déploiement d'un Job sur un serveur SpagoBI.*

Pour savoir comment exécuter un Job sur un serveur Hadoop via **Oozie scheduler**, consultez [Gestion des Jobs](#).

### 6.7.1. Exécuter un Job en mode normal



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

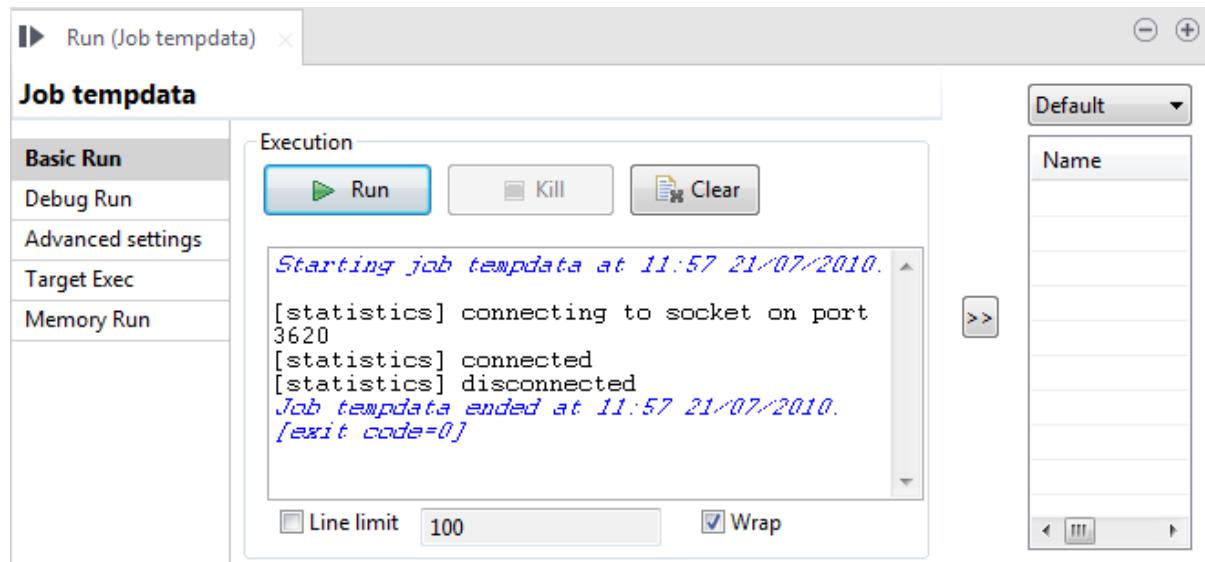
Pour exécuter votre Job en mode normal, procédez comme suit :

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 l'exécution de ce Job. 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 : [Utiliser les contextes et les variables](#).

1. Cliquez sur **Run** pour lancer l'exécution.
2. Sur le même panneau, la console de log affiche la progression de l'exécution. La console inclut les messages d'erreurs ainsi que les messages de début et de fin de processus.

3. Pour 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 s'affiche, vous permettant de voir la fin des lignes.



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

Le *Studio Talend* offre d'autres fonctionnalités informatives, notamment **Statistics** et **Traces**, qui facilitent la supervision du Job ainsi que le travail de débogage. Pour plus d'informations, consultez les sections suivantes.

## 6.7.2. 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.

Dans un premier temps, il est recommandé 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 de pause s'affiche à gauche du composant dans l'éditeur graphique.

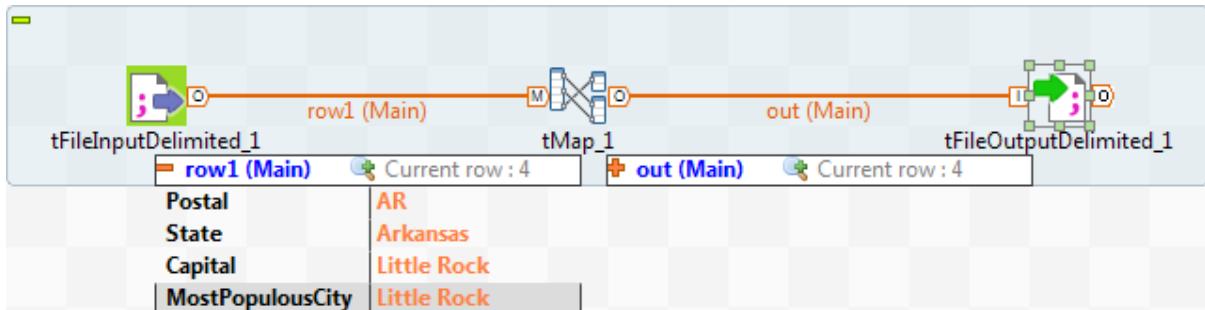
Pour passer en mode Debug, cliquez sur le bouton **Debug** dans le panneau **Run Job**. Le *Studio Talend* ouvre l'éditeur en mode de débogage.

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

### 6.7.3. 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 la perspective **Integration** du *Studio Talend*.

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



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

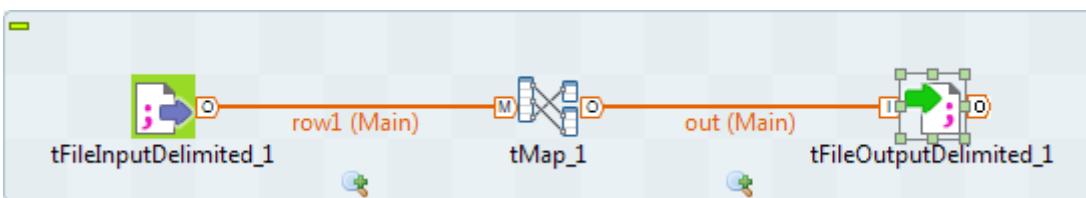
La fonction **Traces** affiche le contenu des lignes traitées dans un tableau.



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

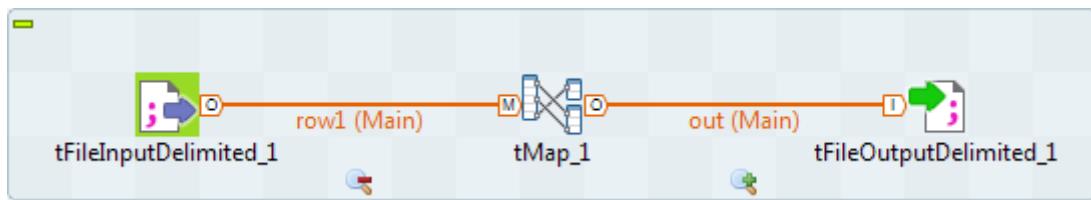
Vous pouvez activer ou désactiver le mode **Traces** ou décider quelles colonnes traitées afficher dans le tableau des traces qui s'affichera dans l'espace de modélisation graphique lors de l'exécution du Job.

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**.

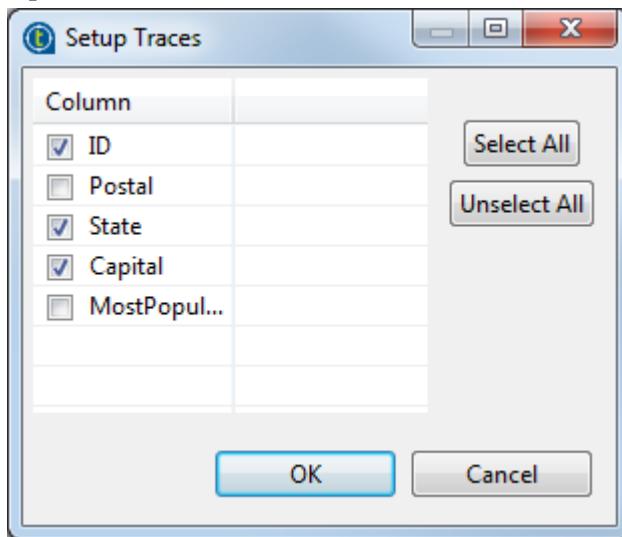
Pour désactiver le mode **Traces** d'un flux de votre Job :



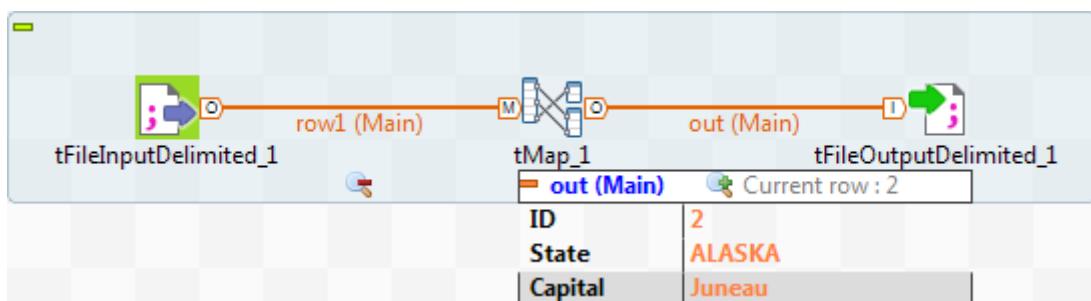
1. Cliquez-droit sur l'icône de **Traces** du flux concerné.
2. Sélectionnez **Disable Traces** dans le menu contextuel. Un signe moins rouge vient remplacer le signe plus vert sur l'icône pour indiquer que le mode **Traces** a été désactivé pour ce flux.

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.

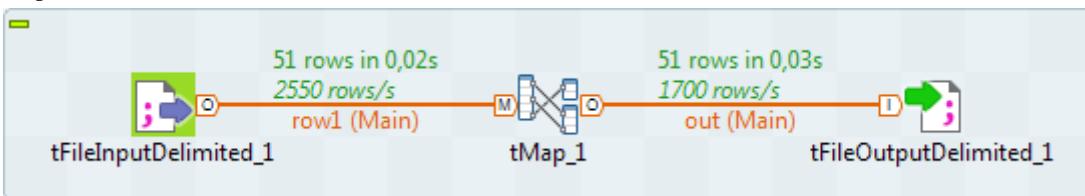
## 6.7.4. Configurer les paramètres d'exécution avancés

Dans l'onglet **Advanced settings** de la vue **Run**, plusieurs paramètres d'exécution avancés sont disponibles pour rendre l'exécution des Jobs plus pratique :

- **Statistics**, cette fonction affiche le taux de performance de traitement. Pour plus d'informations, consultez [Afficher les Statistiques](#).
- **Exec time**, cette fonction affiche le temps d'exécution dans la console, à la fin de l'exécution. Pour plus d'informations, consultez [Afficher la durée d'exécution et d'autres options](#).
- **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. Pour un exemple concernant l'utilisation de cette fonction, consultez [Afficher des caractères spéciaux dans la console](#).

### 6.7.4.1. 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 le Job 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**.

### 6.7.4.2. Afficher la durée d'exécution et d'autres options

Pour afficher le temps total d'exécution, cochez la case **Exec time** avant d'exécuter votre Job.

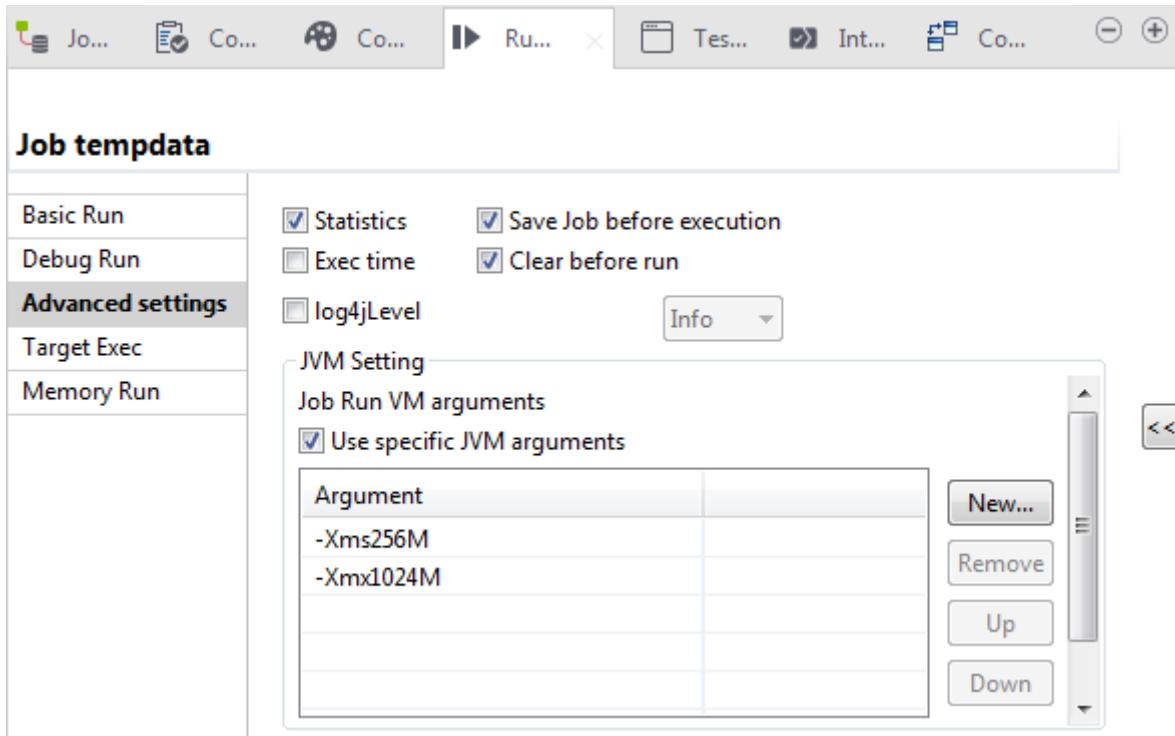
De cette manière vous pouvez vérifier les résultats ou tester votre Job avant de l'envoyer en production.

Vous pouvez aussi nettoyer votre espace de modélisation avant chaque exécution de Job 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**.

### 6.7.4.3. Afficher des caractères spéciaux dans la console

Le *Studio Talend* 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 le *Studio Talend*. Pour plus d'informations concernant l'application d'un argument JVM à toutes les exécutions de Jobs, consultez [Préférences d'exécution et de débogage \(Talend > Run/Debug\)](#).

### 6.7.5. Montrer l'utilisation des ressources de la JVM durant une exécution de Job

L'onglet **Memory Run** de la vue **Run** de votre *Studio Talend* vous permet de montrer en temps réel l'utilisation des ressources de la JVM durant l'exécution d'un Job, notamment la consommation de mémoire et l'utilisation de CPU hôte. Cela vous permet de prendre des décisions lorsque l'utilisation des ressources est trop importante et que votre *Studio Talend* subit une perte de performance. Vous pouvez augmenter la taille de la mémoire allouée à la JVM, arrêter des Jobs qui n'ont pas forcément à s'exécuter, etc.

Pour montrer l'utilisation des ressources de la JVM lors de l'exécution d'un Job, procédez comme suit :

1. Ouvrez votre Job.

Dans la vue **Run**, cliquez sur l'onglet **Memory Run**.

- Cliquez sur **Run** pour exécuter le Job.

Vous pouvez cliquez sur le bouton **Run** de l'onglet **Memory Run** pour monitorer l'utilisation des ressources de la JVM par votre Job à tout moment, même après avoir lancé son exécution dans l'onglet **Basic Run**.

La console du Studio affiche des graphiques montrant l'utilisation des ressources de la JVM et du processeur, respectivement, durant l'exécution du Job. Des messages d'avertissement s'affichent en rouge dans la zone **Job execution information** lorsque les seuils sont atteints.



- Pour voir des informations concernant les ressources utilisées à un moment donné de l'exécution, placez votre curseur sur ce point, dans le graphique correspondant. Selon le graphique sur lequel vous placez votre curseur, vous pouvez voir les informations relatives à la taille du tas, le seuil des 90 % et des 70 % du tas, ou l'usage du CPU à un moment donné.
- Pour exécuter le ramasse-miettes (**Garbage Collector**) durant un intervalle de temps spécifié, cochez la case **With Garbage Collector pace set to** et sélectionnez un intervalle en secondes. Le ramasse-miettes s'exécute automatiquement lors des périodes indiquées.

Pour exécuter le ramasse-miettes immédiatement, cliquez sur le bouton **Trigger GC**.

- Pour exporter les informations de log dans un fichier texte, cliquez sur le bouton **Export** et sélectionnez un fichier pour sauvegarder le log.
- Pour arrêter le Job, cliquez sur le bouton **Kill**.

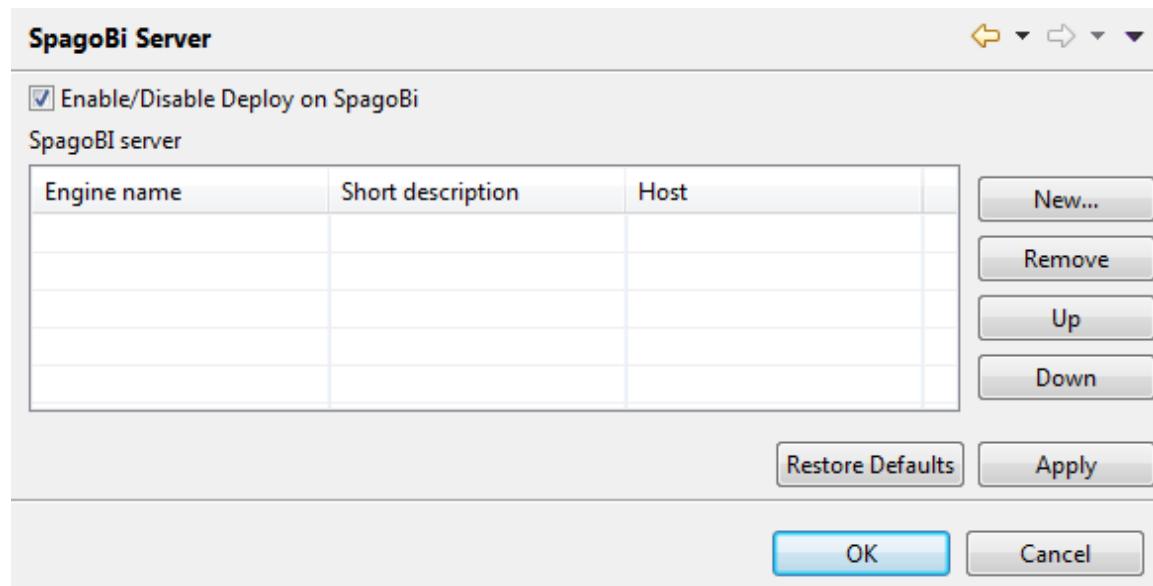
## 6.7.6. Déploiement d'un Job sur un serveur SpagoBI

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

### 6.7.6.1. Créez une nouvelle connexion au serveur SpagoBI

Avant toute chose, renseignez les informations concernant votre serveur simple ou multiple dans le *Studio Talend*.

- Cliquez sur le menu **Window > Preferences** pour ouvrir la boîte de dialogue **[Preferences]**.
- 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                                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------------------------------|
| <b>Engine Name</b>       | Saisissez le nom interne utilisé dans le <i>Studio Talend</i> . Ce nom n'est pas utilisé dans le code généré. |
| <b>Short description</b> | Saisissez une description du serveur que vous êtes en train de créer.                                         |
| <b>Host</b>              | Adresse IP ou nom de l'hôte de la machine sur laquelle SpagoBI fonctionne.                                    |
| <b>Login</b>             | Identifiant requis pour vous connecter au serveur SpagoBI.                                                    |
| <b>Password</b>          | Mot de passe pour vous connecter au serveur SpagoBI.                                                          |

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

The screenshot shows the 'SpagoBI Server' preferences dialog. At the top left is a checked checkbox labeled 'Enable/Disable Deploy on SpagoBI'. Below it is a section titled 'SpagoBI server' containing a table with three rows:

| Engine name | Short description | Host        |
|-------------|-------------------|-------------|
| mySpago     | i Prod Server     | 92.168.10.3 |
| mySpago2    | i Dev Server      | 92.168.10.4 |

On the right side of the dialog are four buttons: 'New...', 'Remove', 'Up', and 'Down'.

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

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

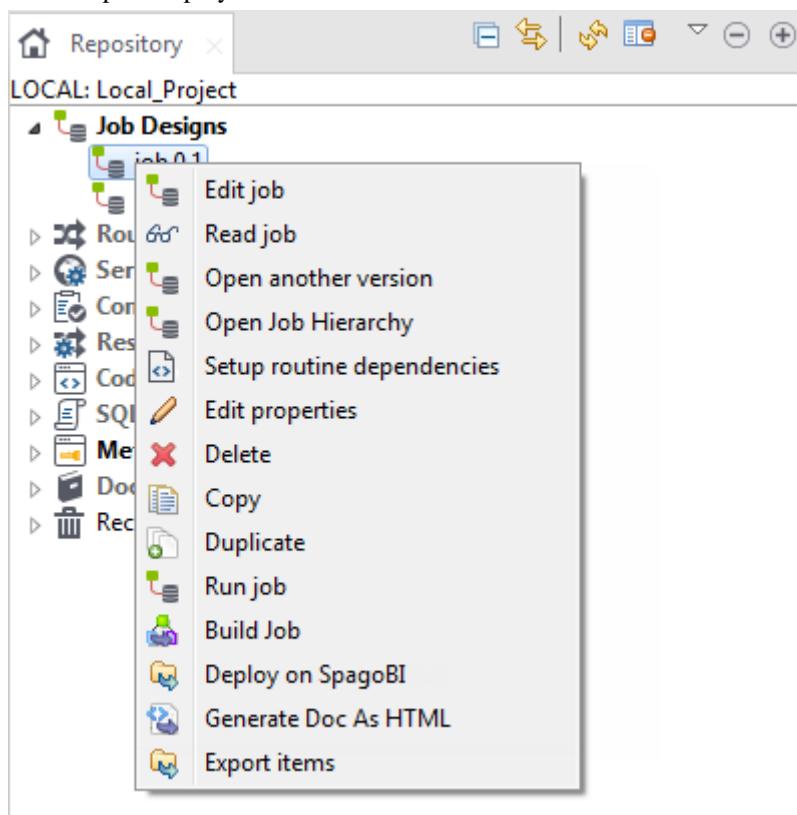
### 6.7.6.2. Modifier ou supprimer un serveur SpagoBI

Pour supprimer une entrée obsolète, sélectionnez l'entrée dans le tableau, puis cliquez sur le bouton **Remove**.

Puis, si nécessaire, créez une nouvelle entrée contenant les informations mises à jour.

### 6.7.6.3. 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 administrateur SpagoBI pour exécuter vos Jobs.

## 6.8. Utilisation de la parallélisation pour optimiser les performances des Jobs

La parallélisation, en termes de Jobs **Talend**, signifie accomplir différents processus techniques via des exécutions parallèles. Lorsqu'il est bien conçu, un processus technique parallélisé peut être exécuté en moins de temps.

Le studio **Talend** vous permet d'implémenter différents types de parallélisation selon les circonstances. Les circonstances peuvent être :

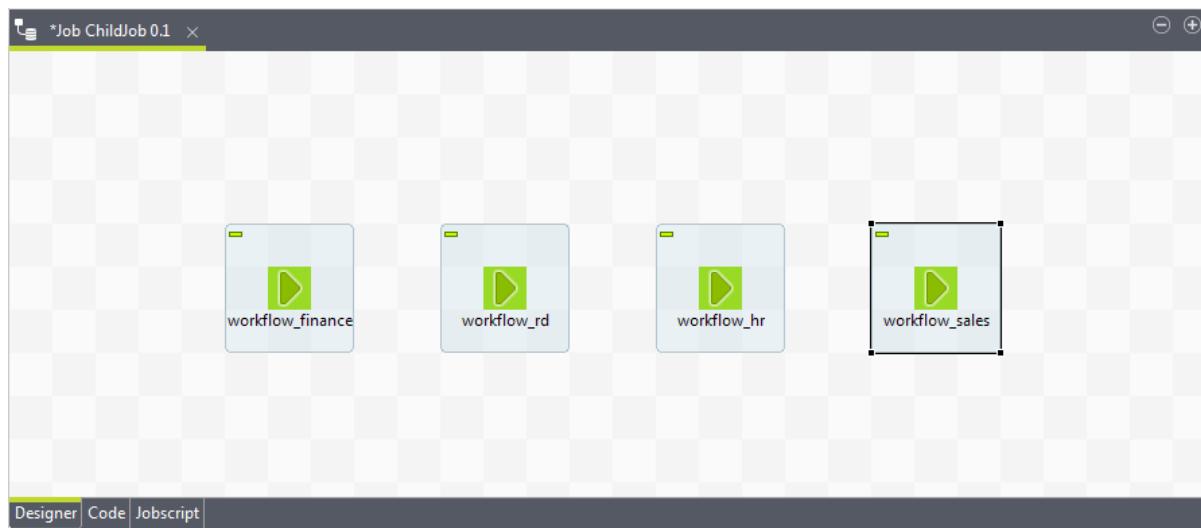
1. Exécution parallèle de plusieurs sous-jobs. Pour plus d'informations, consultez [\*Exécuter plusieurs sous-jobs en parallèle\*](#).
2. Itérations parallèles pour lecture de données. Pour plus d'informations, consultez [\*Lancer des itérations parallèles pour lire des données\*](#).

La parallélisation est une fonctionnalité avancée et nécessite des connaissances basiques concernant les Jobs **Talend**, notamment comment créer et exécuter un Job ou un sous-job, comment utiliser des composants et comment utiliser les différents types de liens reliant les composants ou les Jobs. Si vous ne possédez pas ces connaissances, consultez [\*Conception de Jobs\*](#).

### 6.8.1. Exécuter plusieurs sous-jobs en parallèle

La fonctionnalité **Multi thread execution** vous permet d'exécuter en parallèle différents sous-jobs actifs dans l'espace de modélisation.

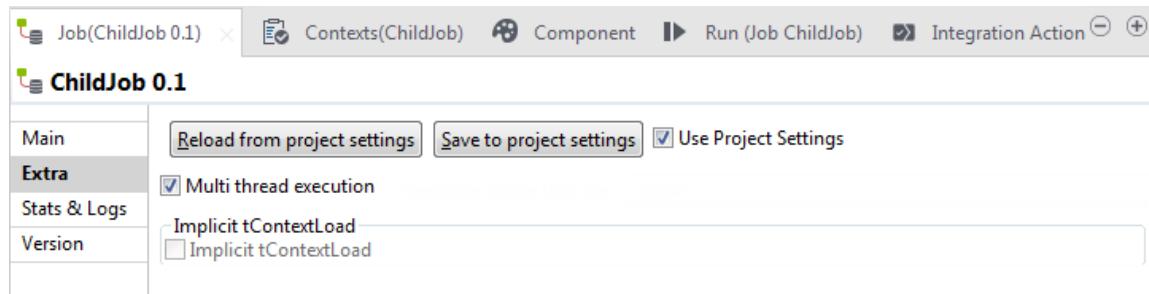
Comme expliqué dans la section précédente, un Job ouvert dans l'espace de modélisation graphique peut contenir plusieurs sous-jobs. Vous pouvez personnaliser leur ordre d'exécution à l'aide des liens de déclenchement (**Trigger**), comme le lien **OnSubjobOk**. Cependant, il est possible de lancer les sous-jobs n'ayant aucune dépendance entre eux en même temps. Par exemple, l'image suivante présente quatre sous-jobs au sein d'un même Job sans aucune dépendance entre eux.



Plusieurs composants **tRunJob** sont utilisés dans cet exemple. Chacun appelle le sous-job qu'il représente. Pour plus d'informations concernant le **tRunJob**, consultez la documentation de ce composant à l'adresse <https://help.talend.com>.

Une fois le Job ouvert dans l'espace de modélisation graphique, procédez comme suit pour exécuter les sous-jobs en parallèle :

1. Cliquez sur l'onglet **Job** puis sur l'onglet **Extra**.



2. Cochez la case **Multi thread execution** pour activer l'exécution parallèle.

Cette fonctionnalité est optimale lorsque le nombre de processus (en général un sous-job compte pour un processus) ne dépasse pas le nombre de processeurs de l'ordinateur utilisé pour ces exécutions parallèles.

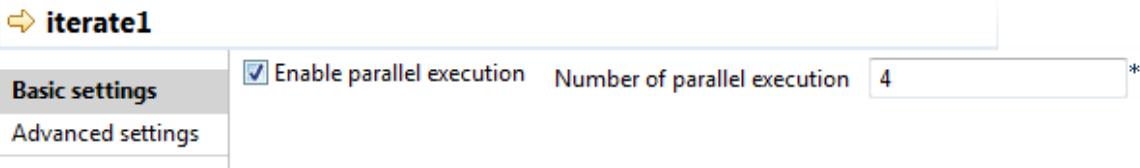
Pour un cas d'utilisation de cette fonctionnalité pour exécuter des Jobs en parallèle, consultez *Utilisation de la fonctionnalité Exécution en multi thread pour exécuter des Jobs en parallèle* à l'adresse <https://help.talend.com>.

## 6.8.2. Lancer des itérations parallèles pour lire des données

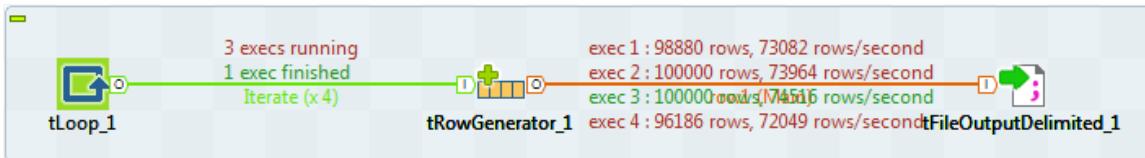
Une connexion **Iterate** parallélisée permet au composant recevant des données de la connexion de lire ces données en parallèle.

Procédez comme suit pour configurer les itérations parallèles :

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



Lorsque vous exécutez votre Job, le nombre d'itérations parallèles est distribué sur les processeurs disponibles.



3. Cochez la case **Statistics** de la vue **Run** afin d'afficher les exécutions parallèles en temps réel dans l'espace de modélisation graphique.

Cette fonctionnalité est particulièrement utile lorsque vous devez utiliser le lien **Iterate** afin de passer les variables de contexte à un sous-job. Dans ce cas, les variables doivent être lues en parallèle dans le sous-job et les processus générés par le sous-job sont exécutés simultanément alors l'aide de ces variables.





## Chapitre 7. Gestion des Jobs

Ce chapitre introduit les procédures de gestion de Jobs dans votre studio **Talend**, permet de tirer parti de la plateforme Big Data de Hadoop et de travailler avec des ensembles de données volumineux. Pour des procédures générales de création, exécution et gestion de Jobs **Talend** d'intégration de données, consultez [\*Conception de Jobs\*](#).

## 7.1. Exécution d'un Job via Oozie

Votre studio **Talend** fournit un **Oozie scheduler**, une fonctionnalité qui vous permet de planifier les exécutions d'un Job que vous avez créé ou de l'exécuter directement sur un serveur distant Hadoop Distributed File System (HDFS) et de moniturer le statut d'exécution de votre Job. Plus plus d'informations concernant Apache Oozie et Hadoop, consultez <http://oozie.apache.org/> et <http://hadoop.apache.org/> (en anglais).

 Si la vue **Oozie scheduler** ne s'affiche pas, cliquez sur **Window > Show view** et sélectionnez **Talend Oozie** dans la boîte de dialogue **[Show view]** afin de l'afficher dans la zone des onglets de configuration.

### 7.1.1. Configurer les informations de la connexion HDFS

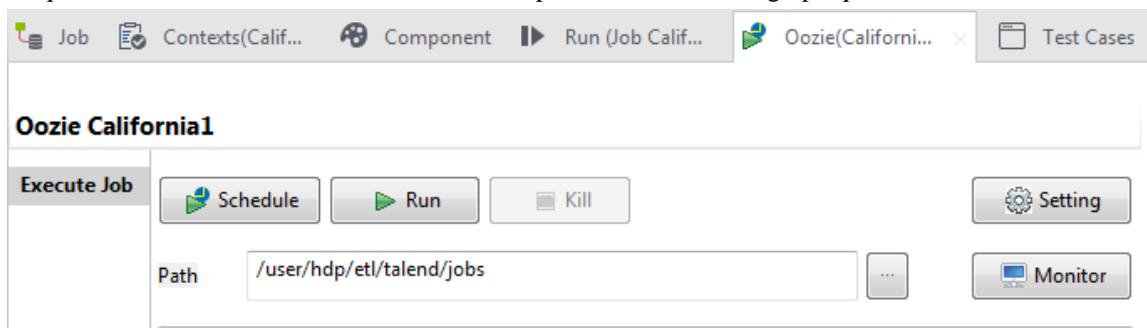
**Talend** Oozie vous permet de planifier les exécutions des Jobs créés dans le studio.

Avant d'exécuter ou de planifier les exécutions d'un Job sur un serveur HDFS, vous devez d'abord configurer les détails de la connexion HDFS, dans la vue **Oozie scheduler**, puis spécifier l'emplacement où sera déployé votre Job.

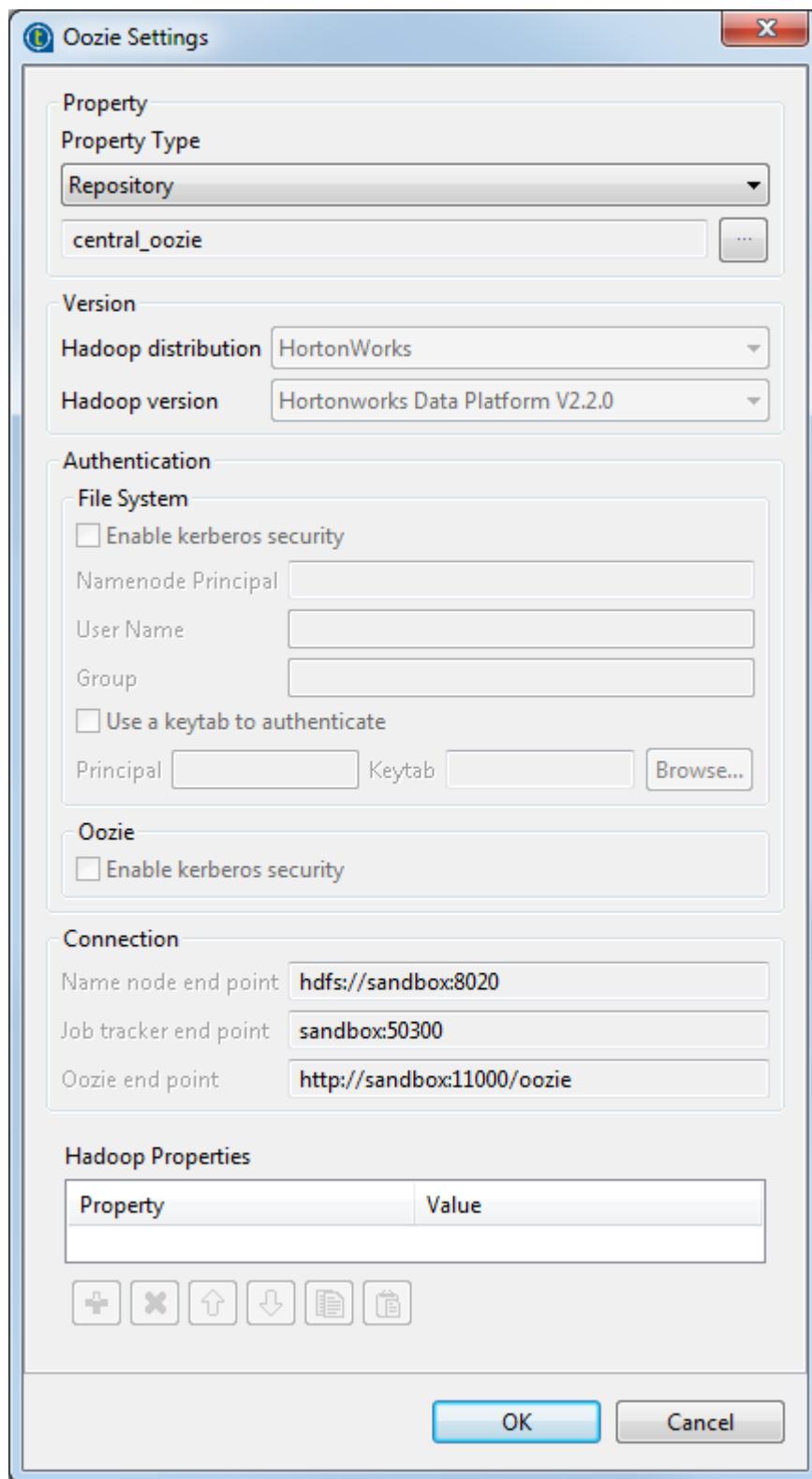
#### 7.1.1.1. Définir les détails de la connexion HDFS dans la vue Oozie scheduler

Afin de configurer les détails de la connexion HDFS dans la vue **Oozie scheduler** procédez comme suit :

1. Cliquez sur la vue **Oozie scheduler** en bas de l'espace de modélisation graphique.



2. Cliquez sur **Setting** pour ouvrir la boîte de dialogue de la connexion.



Les paramètres de connexion affichés ci-dessus sont donnés à titre d'exemple.

- Si vous avez configuré la connexion Oozie dans le référentiel, comme expliqué dans [Centraliser une connexion Oozie](#), vous pouvez la réutiliser.

Pour ce faire, sélectionnez **Repository** dans la liste **Property type**, cliquez sur le bouton [...] pour ouvrir la boîte de dialogue **[Repository Content]** et sélectionnez la connexion à Oozie à utiliser.

- Sinon, renseignez les informations dans les champs correspondants, comme expliqué dans le tableau ci-dessous.

| Champ/Option                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Hadoop distribution</b>      | Distribution Hadoop à laquelle vous connecter. Cette distribution héberge le système de fichiers HDFS à utiliser. Si vous sélectionnez <b>Custom</b> pour vous connecter à une distribution Hadoop personnalisée, cliquez sur le bouton [...] pour ouvrir la boîte de dialogue <b>[Import custom definition]</b> . Dans cette boîte de dialogue, importez les fichiers Jar requis par la distribution personnalisée.<br><br>Pour plus d'informations, consultez la <a href="#">Connexion à une distribution Hadoop personnalisée</a> .                                                                                                                                                                                                                                                               |
| <b>Hadoop version</b>           | Version de la distribution Hadoop distribution à laquelle vous connecter. Cette liste disparaît si vous sélectionnez <b>Custom</b> dans la liste <b>Hadoop distribution</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Enable kerberos security</b> | Si vous accédez au cluster Hadoop fonctionnant avec la sécurité Kerberos, cochez cette case, puis saisissez le Principal Name de Kerberos pour le NameNode dans le champ affiché. Cela vous permet d'utiliser votre nom d'utilisateur pour vous authentifier, en les comparant aux informations stockées dans Kerberos.<br><br>Cette case est disponible ou non selon la distribution Hadoop à laquelle vous vous connectez.                                                                                                                                                                                                                                                                                                                                                                         |
| <b>User Name</b>                | Nom d'utilisateur.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Name node end point</b>      | URI du NameNode, le cœur du système de fichier HDFS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Job tracker end point</b>    | URI du nœud Job Tracker, qui sous-traite les tâches MapReduce dans des nœuds spécifiques du cluster.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Oozie end point</b>          | URI de la console Web d'Oozie, pour le monitoring de l'exécution du Job.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Hadoop Properties</b>        | Si vous devez utiliser une configuration personnalisée pour la distribution d'Hadoop à utiliser, renseignez cette table avec la ou les propriété(s) à personnaliser. Lors de l'exécution, les propriétés personnalisées écrasent celles par défaut utilisées par le studio pour son moteur Hadoop.<br><br>Pour plus d'informations concernant les propriétés requises par Hadoop, consultez la documentation de Apache Hadoop à l'adresse suivante : <a href="http://hadoop.apache.org">http://hadoop.apache.org</a> (en anglais) ou la documentation de la distribution Hadoop que vous utilisez.<br><br> Les paramètres configurés dans cette table sont effectifs dans le Job pour lequel ils ont été définis. |

Une fois configuré le chemin de déploiement dans la vue **Oozie scheduler**, vous pouvez planifier les exécutions de votre Job, ou l'exécuter immédiatement sur le serveur HDFS.

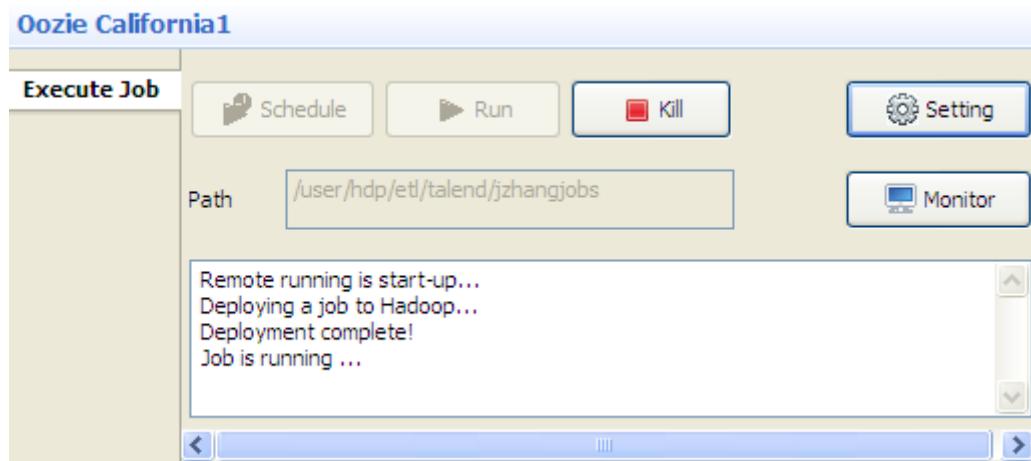
## 7.1.2. Exécuter un Job sur le serveur HDFS

Pour exécuter un Job sur le serveur HDFS, procédez comme suit :

1. Dans le champ **Path**, dans l'onglet **Oozie scheduler**, saisissez le chemin où déployer votre Job sur le serveur HDFS.
2. Cliquez sur le bouton **Run** pour commencer le déploiement et l'exécution du Job sur le serveur HDFS.

Les données de votre Job sont zippées, envoyées et déployées sur le serveur HDFS selon les paramètres de connexion du serveur et sont automatiquement exécutées. Selon votre connectivité, l'opération peut prendre du temps. La console affiche le statut du déploiement et de l'exécution du Job.

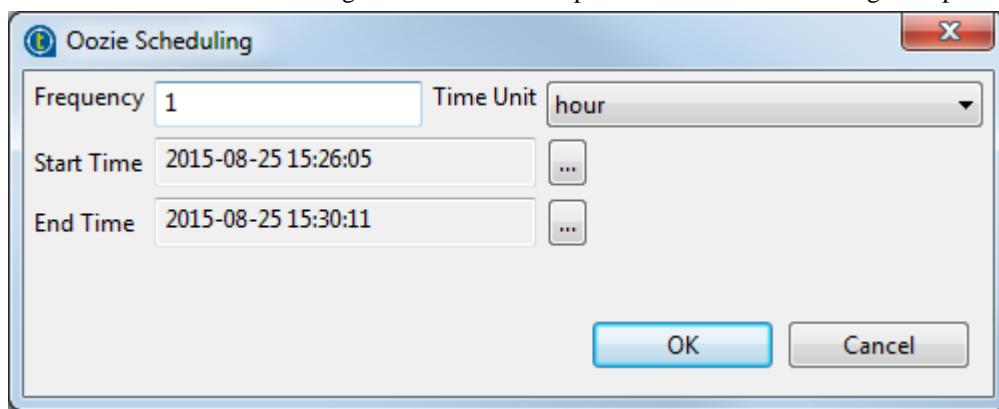
Pour arrêter l'exécution du Job avant sa fin, cliquez sur le bouton **Kill**.



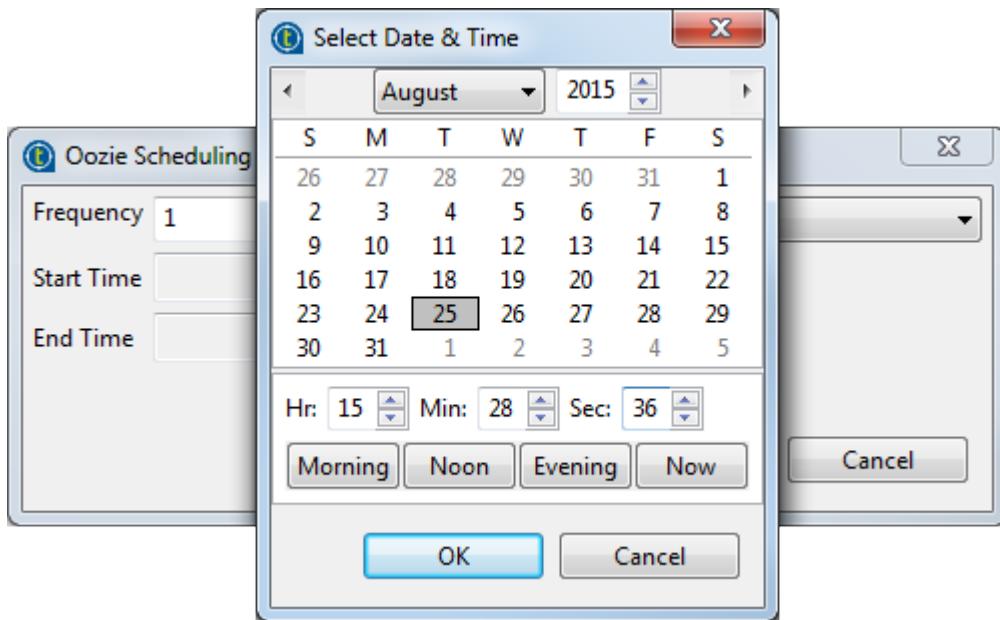
### 7.1.3. Planifier les exécutions d'un Job

La fonctionnalité **Oozie scheduler** intégrée dans le studio **Talend** vous permet de planifier les exécutions de votre Job sur le serveur HDFS. Votre Job est exécuté selon la fréquence définie, durant le laps de temps configuré. Pour configurer la planification du Job, procédez comme suit :

1. Dans le champ **Path** de l'onglet **Oozie scheduler**, saisissez le chemin où déployer votre Job sur le serveur HDFS, si le chemin de déploiement n'est pas encore défini.
2. Cliquez sur le bouton **Schedule** de l'onglet **Oozie scheduler** pour ouvrir la boîte de dialogue de planification.

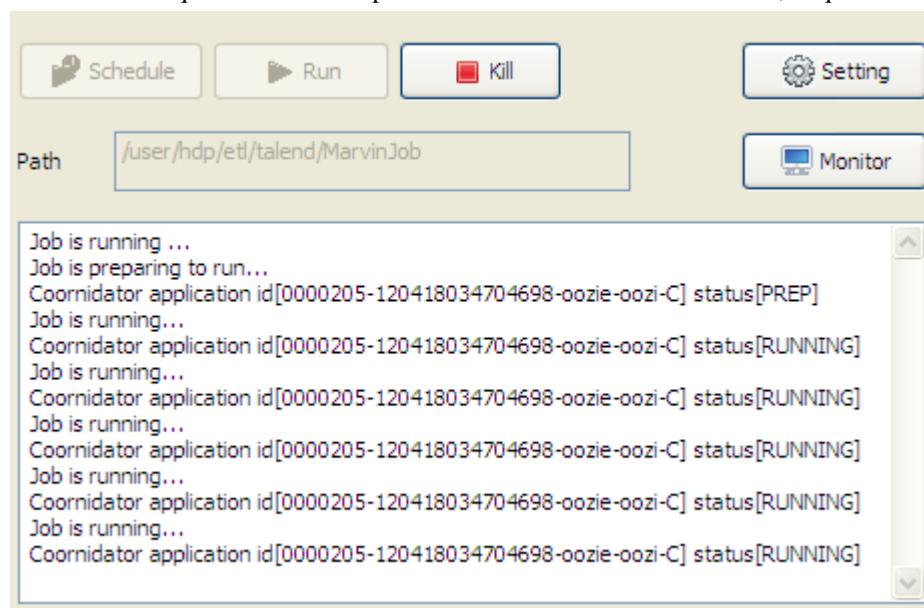


3. Dans le champ **Frequency**, saisissez un entier et sélectionnez une unité de temps dans la liste **Time Unit** afin de configurer la fréquence d'exécution du Job.
4. Cliquez sur le bouton [...] à côté du champ **Start Time** pour ouvrir la boîte de dialogue **Select Date & Time** et sélectionnez la date, l'heure, la minute et la seconde. Cliquez sur **OK** pour configurer l'heure de début de l'exécution du Job. De la même manière, configurez l'heure de fin d'exécution du Job.



5. Cliquez sur **OK** pour fermer la boîte de dialogue et commencer les exécutions planifiées de votre Job.

Le Job s'exécute automatiquement selon les paramètres définis. Pour arrêter le Job, cliquez sur **Kill**.



## 7.1.4. Montrer le statut des exécutions de Job

Pour montrer le statut et les résultats des exécutions de Job, cliquez sur le bouton **Monitor** de l'onglet **Oozie scheduler**. L'URI de l'endpoint Oozie s'ouvre dans votre navigateur Web, affichant les informations d'exécution des Jobs sur le serveur HDFS.

| Oozie Web Console                       |             |                             |                           |                                 |                 |                               |                               |                         |  |  |
|-----------------------------------------|-------------|-----------------------------|---------------------------|---------------------------------|-----------------|-------------------------------|-------------------------------|-------------------------|--|--|
| Workflow Jobs                           |             | Coordinator Jobs            | Bundle Jobs               | System Info                     | Instrumentation |                               |                               |                         |  |  |
| <a href="#">All Jobs</a>                |             | <a href="#">Active Jobs</a> | <a href="#">Done Jobs</a> | <a href="#">Custom Filter ▾</a> |                 |                               |                               |                         |  |  |
| Job Id                                  | Name        | Status                      | Run                       | User                            | Group           | Created                       | Started                       | Last Modified           |  |  |
| 31 0000184-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:59:39 GMT | Thu, 26 Apr 2012 08:59:39 GMT | Thu, 26 Apr 2012 09:00: |  |  |
| 32 0000183-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:58:54 GMT | Thu, 26 Apr 2012 08:58:54 GMT | Thu, 26 Apr 2012 08:59: |  |  |
| 33 0000182-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:52:14 GMT | Thu, 26 Apr 2012 08:52:14 GMT | Thu, 26 Apr 2012 08:52: |  |  |
| 34 0000181-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:44:49 GMT | Thu, 26 Apr 2012 08:44:49 GMT | Thu, 26 Apr 2012 08:45: |  |  |
| 35 0000180-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:43:53 GMT | Thu, 26 Apr 2012 08:43:53 GMT | Thu, 26 Apr 2012 08:44: |  |  |
| 36 0000179-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:41:07 GMT | Thu, 26 Apr 2012 08:41:07 GMT | Thu, 26 Apr 2012 08:41: |  |  |
| 37 0000178-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:40:02 GMT | Thu, 26 Apr 2012 08:40:02 GMT | Thu, 26 Apr 2012 08:40: |  |  |
| 38 0000177-120418034704698-oozie-oozi-W | MarvinJob   | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:39:13 GMT | Thu, 26 Apr 2012 08:39:13 GMT | Thu, 26 Apr 2012 08:39: |  |  |
| 39 0000175-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:39:12 GMT | Thu, 26 Apr 2012 08:39:12 GMT | Thu, 26 Apr 2012 08:39: |  |  |
| 40 0000174-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:37:35 GMT | Thu, 26 Apr 2012 08:37:35 GMT | Thu, 26 Apr 2012 08:38: |  |  |
| 41 0000173-120418034704698-oozie-oozi-W | MarvinJob   | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:37:14 GMT | Thu, 26 Apr 2012 08:37:14 GMT | Thu, 26 Apr 2012 08:37: |  |  |
| 42 0000172-120418034704698-oozie-oozi-W | MarvinJob   | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:36:48 GMT | Thu, 26 Apr 2012 08:36:48 GMT | Thu, 26 Apr 2012 08:37: |  |  |
| 43 0000170-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:33:43 GMT | Thu, 26 Apr 2012 08:33:43 GMT | Thu, 26 Apr 2012 08:34: |  |  |
| 44 0000169-120418034704698-oozie-oozi-W | California1 | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:33:04 GMT | Thu, 26 Apr 2012 08:33:04 GMT | Thu, 26 Apr 2012 08:33: |  |  |
| 45 0000168-120418034704698-oozie-oozi-W | California1 | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:32:38 GMT | Thu, 26 Apr 2012 08:32:38 GMT | Thu, 26 Apr 2012 08:33: |  |  |
| 46 0000166-120418034704698-oozie-oozi-W | California1 | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:29:11 GMT | Thu, 26 Apr 2012 08:29:11 GMT | Thu, 26 Apr 2012 08:29: |  |  |
| 47 0000164-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:25:58 GMT | Thu, 26 Apr 2012 08:25:58 GMT | Thu, 26 Apr 2012 08:26: |  |  |
| 48 0000163-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:17:35 GMT | Thu, 26 Apr 2012 08:17:35 GMT | Thu, 26 Apr 2012 08:17: |  |  |
| 49 0000162-120418034704698-oozie-oozi-W | ghdf        | SUCCEEDED                   | 0                         | hdp                             | users           | Thu, 26 Apr 2012 08:12:24 GMT | Thu, 26 Apr 2012 08:12:24 GMT | Thu, 26 Apr 2012 08:12: |  |  |

Pour afficher les informations détaillées d'un Job particulier, cliquez sur l'un des champs du Job, pour ouvrir une page séparée affichant les détails du Job.

**Job (Name: MarvinJob/JobId: 0000216-120418034704698-oozie-oozi-W)**

| Job Info                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Job Definition | Job Configuration | Job Log |            |                             |                        |        |            |           |         |                                            |           |      |    |     |                             |                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-------------------|---------|------------|-----------------------------|------------------------|--------|------------|-----------|---------|--------------------------------------------|-----------|------|----|-----|-----------------------------|------------------------|
| <b>Job Id:</b> 0000216-120418034704698-oozie<br><b>Name:</b> MarvinJob<br><b>App Path:</b> hdfs://192.168.0.81/user/hdp/etl/<br><b>Run:</b> 0<br><b>Status:</b> SUCCEEDED<br><b>User:</b> hdp<br><b>Group:</b> users<br><b>Create Time:</b> Thu, 26 Apr 2012 10:17:45 GMT<br><b>Nominal Time:</b><br><b>Start Time:</b> Thu, 26 Apr 2012 10:17:45 GMT<br><b>Last Modified:</b> Thu, 26 Apr 2012 10:18:09 GMT<br><b>End Time:</b> Thu, 26 Apr 2012 10:18:09 GMT |                |                   |         |            |                             |                        |        |            |           |         |                                            |           |      |    |     |                             |                        |
| <b>Actions</b> <table border="1"> <thead> <tr> <th>Action Id</th> <th>Name</th> <th>Type</th> <th>Status</th> <th>Transition</th> <th>StartTime</th> <th>EndTime</th> </tr> </thead> <tbody> <tr> <td>1 0000216-120418034704698-oozie-oozi-W@...</td> <td>MarvinJob</td> <td>java</td> <td>OK</td> <td>end</td> <td>Thu, 26 Apr 2012 10:17:4...</td> <td>Thu, 26 Apr 2012 10:18</td> </tr> </tbody> </table>                                                   |                |                   |         | Action Id  | Name                        | Type                   | Status | Transition | StartTime | EndTime | 1 0000216-120418034704698-oozie-oozi-W@... | MarvinJob | java | OK | end | Thu, 26 Apr 2012 10:17:4... | Thu, 26 Apr 2012 10:18 |
| Action Id                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Name           | Type              | Status  | Transition | StartTime                   | EndTime                |        |            |           |         |                                            |           |      |    |     |                             |                        |
| 1 0000216-120418034704698-oozie-oozi-W@...                                                                                                                                                                                                                                                                                                                                                                                                                     | MarvinJob      | java              | OK      | end        | Thu, 26 Apr 2012 10:17:4... | Thu, 26 Apr 2012 10:18 |        |            |           |         |                                            |           |      |    |     |                             |                        |





## Chapitre 8. Mapping de flux de données

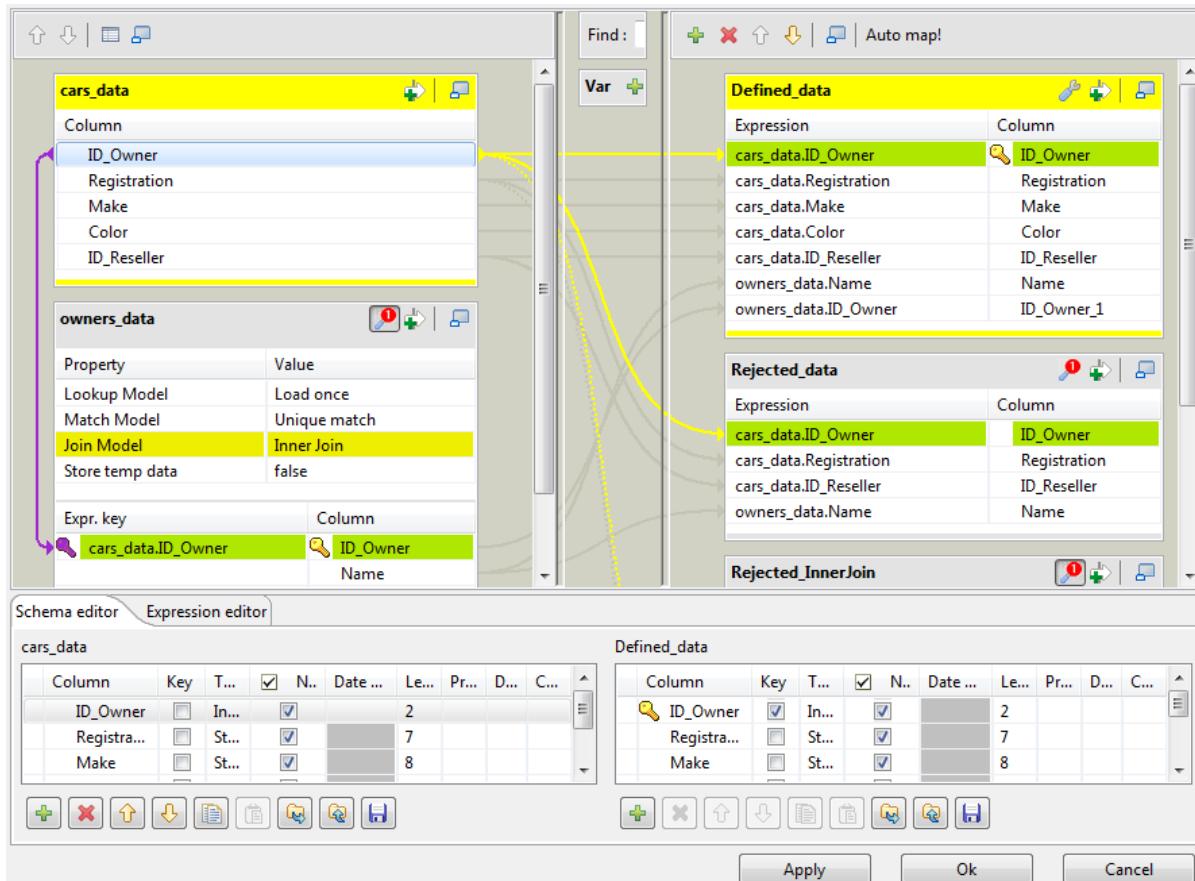
Le moyen le plus courant de gérer des flux multiples en entrée et en sortie, ainsi que les transformations et redirection des données est d'utiliser les composants de mapping.

Ce chapitre explique la théorie derrière l'utilisation des composants de mapping, en prenant comme exemple les composants typiques auxquels vous pouvez vous référer afin d'utiliser les autres composants de mapping.

## 8.1. Interfaces de mapping

Les composants de mapping 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]** contenues 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 des autres composants de mapping a une apparence légèrement différente. Par exemple, en plus des onglets **Schema editor** et **Expression editor** dans la partie inférieure de l'interface, le **tXMLMap** contient un troisième onglet nommé **Tree schema editor**. Pour plus d'informations concernant le composant **tXMLMap**, consultez [Présentation du fonctionnement du tXMLMap](#).

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.
- Le panneau de recherche (**Search panel**), au-dessus de la zone **Variable**. Il permet chercher dans l'éditeur des colonnes ou des expressions contenant le texte saisi dans le champ **Find**.
- La zone **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**).

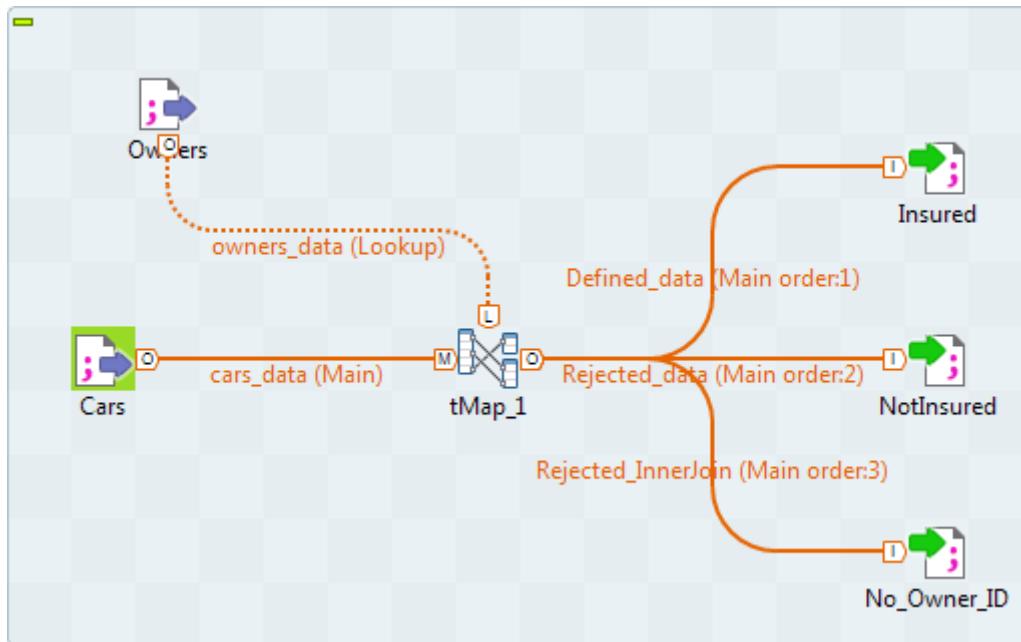
Les sections suivantes présentent séparément les différents composants de mapping, pouvant chacun mapper des flux natures différentes.

## 8.2. 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.

Comme toutes ces opérations de transformation et/ou routage sont réalisées par le **tMap**, ce composant ne peut être ni un composant de début ni un composant de fin de process dans un Job Design.



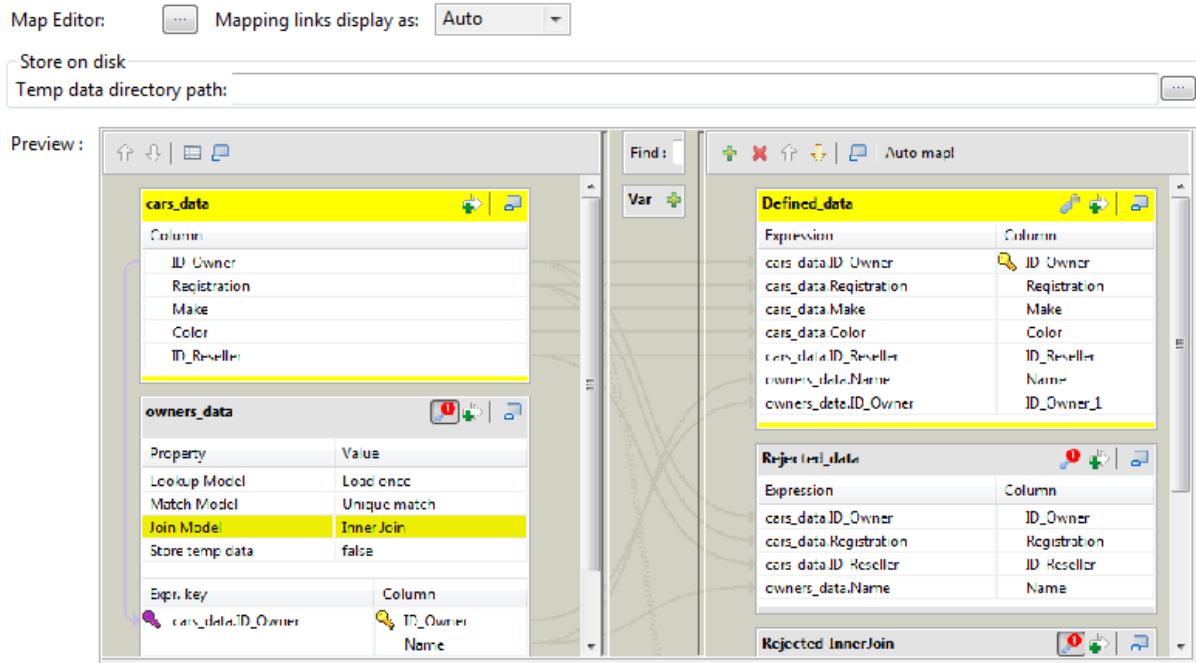
Le **tMap** utilise les connexions entrantes pour pré-remplir les schémas d'entrée dans le **[Map Editor]**. Par conséquent, vous ne pouvez pas créer directement dans le **[Map Editor]**, de nouveaux schémas d'entrée. Par contre, il vous faut mettre en place autant de connexions de type **Row** entrantes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

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

Il ne peut y avoir qu'une seule connexion de type **Main** row. Toute connexion entrante supplémentaire devient automatiquement une connexion **Lookup**. Voir : [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.



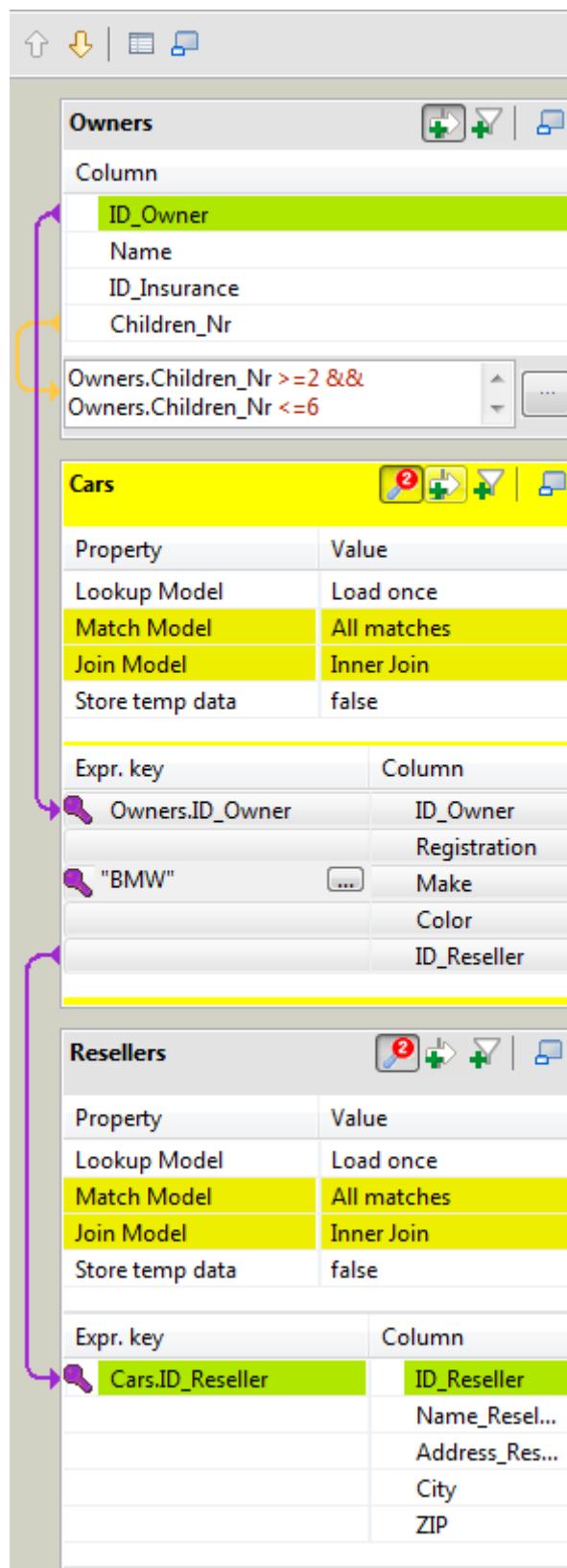
Double-cliquez sur l'icône **tMap** dans l'espace de modélisation ou cliquez sur le bouton [...] à côté du champ **Map Editor** de l'onglet **Basic settings**, dans la vue **Component** du composant **tMap**. L'éditeur de mapping s'ouvre dans une nouvelle fenêtre.

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

## 8.2.1. 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 : [Utiliser les jointures explicites](#).

### 8.2.1.1. 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 [Définir les propriétés des composants](#).

Pour plus d'informations sur la configuration d'un schéma d'entrée dans le **Map Editor**, consultez [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]**.

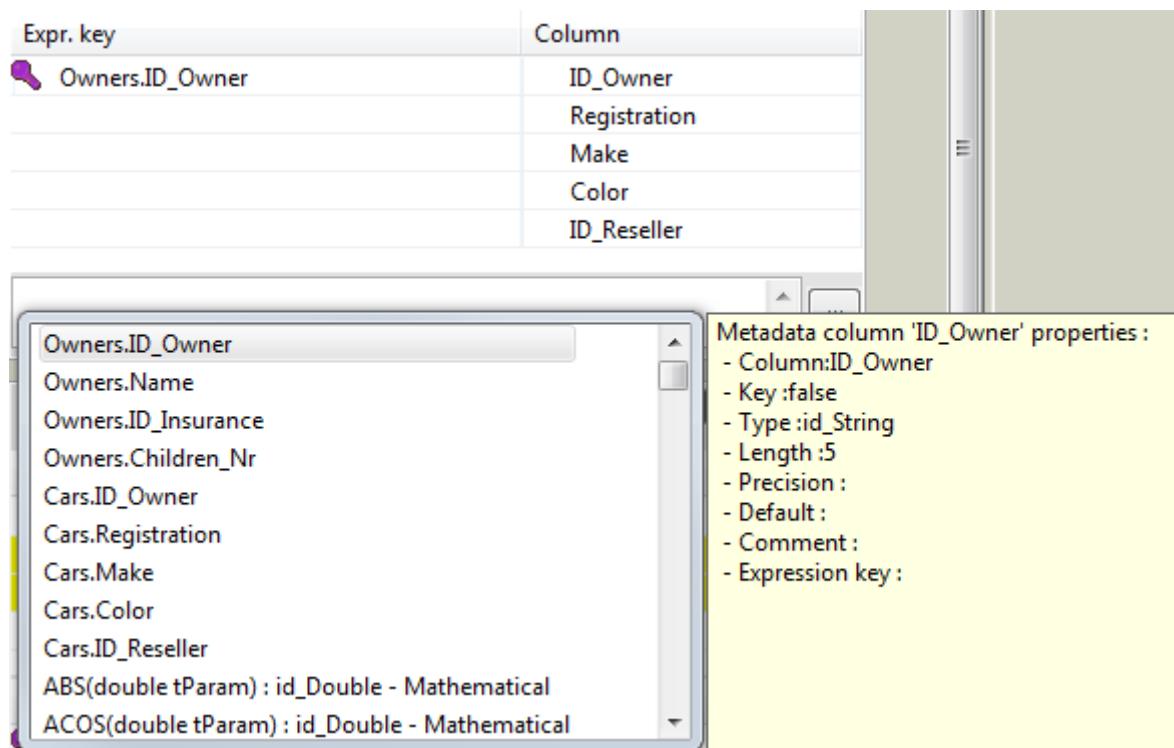
Le contenu des connexions secondaires (**Lookup**) est représenté dans les autres tables apparaissant en dessous de la table **Main**. Si vous n'avez pas encore défini de schéma pour un composant d'entrée, la table correspondante du **[Map Editor]** apparaîtra comme vide.

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 : [Mapping de variables](#)

### 8.2.1.2. 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 créent automatiquement une clé qui aura la même fonction qu'une clé de hachage, c'est-à-dire d'accélérer le processus de recherche.

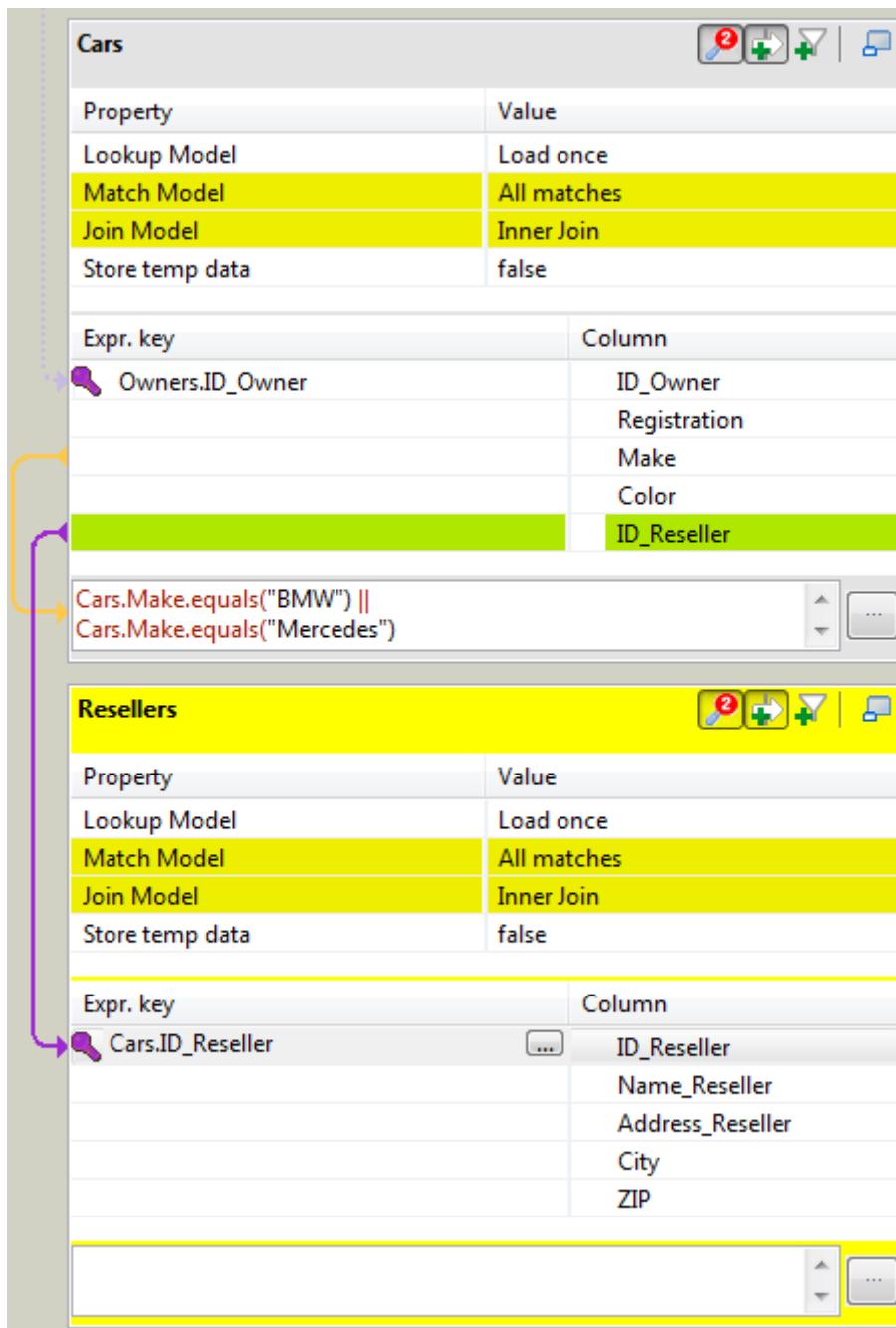
Vous pouvez créer des Joins directs entre la table principale et les tables secondaires. Mais vous pouvez également créer des Joins indirects à partir d'une table principale vers une table secondaire, elle-même reliée à une autre table secondaire. Pour cela, il faut que vous ayez créé une jointure entre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

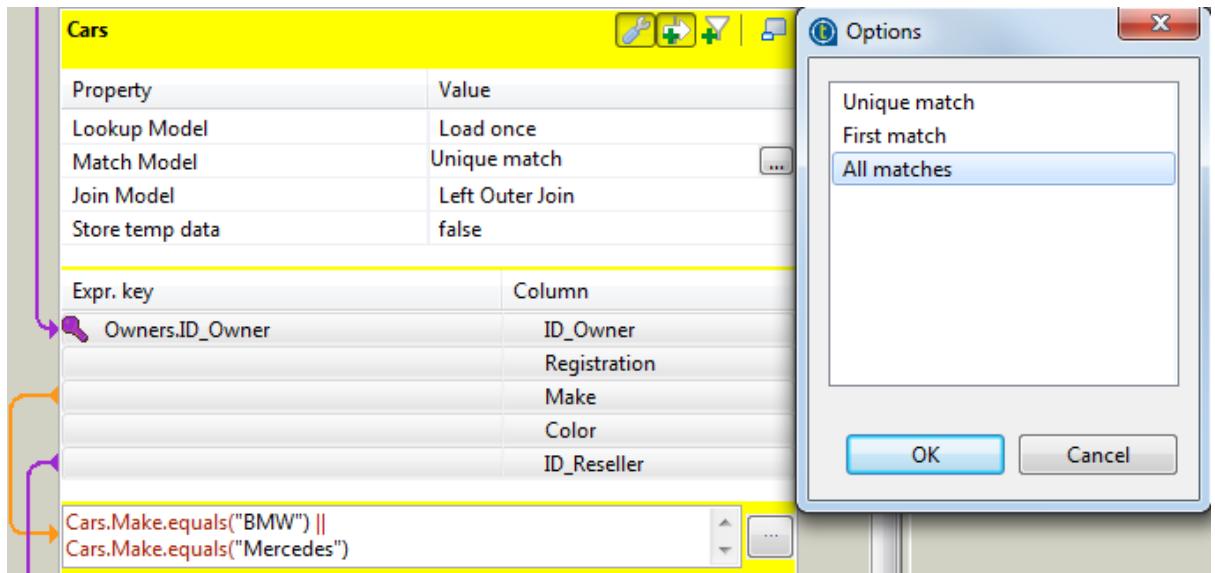
Voir également :

- [Utiliser le Schema editor](#) .
- [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.

### 8.2.1.3. 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.

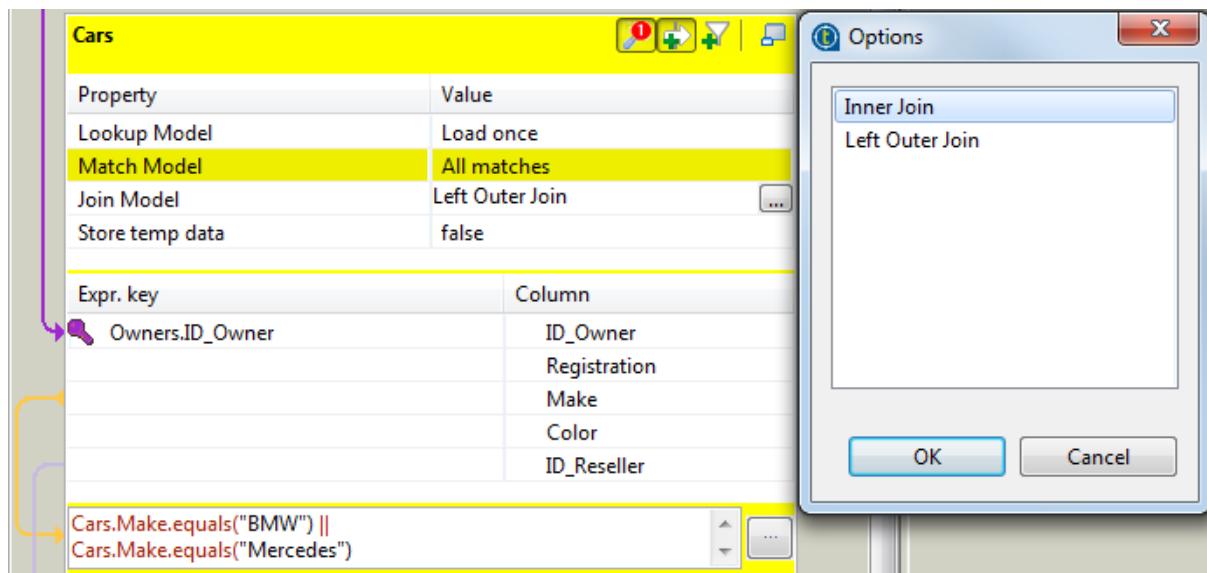
Cette fonction empêche les valeurs nulles de passer dans le flux principal de sortie. Elle permet aussi de faire passer les données rejetées dans une table définie comme table de rejet **Inner Join Reject**.

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 [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 :

- [Utiliser les Rejets Inner Join](#).
- [Filtrer un flux d'entrée](#).

## 8.2.1.4. 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**.

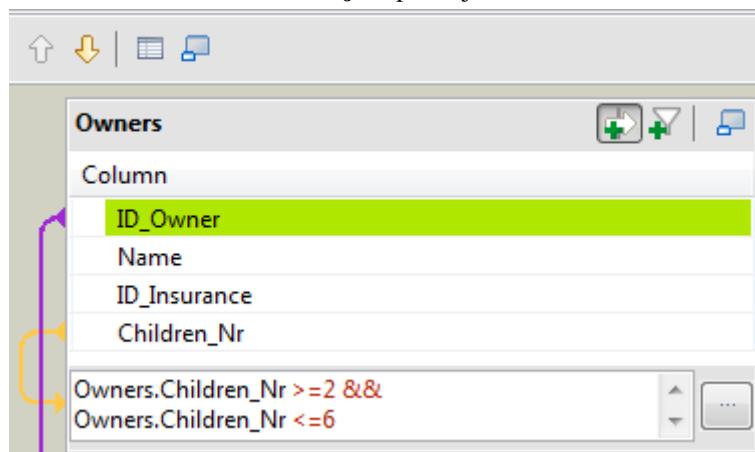
La sortie correspond au produit cartésien des deux tables (ou plus, selon vos besoins).



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 [Utiliser les jointures explicites](#) et la [Utiliser la fonction Inner Join](#).

## 8.2.1.5. 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.

## 8.2.1.6. 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.

## 8.2.2. 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 évitent d'avoir à ressaïsir plusieurs fois les mêmes données.

| Var                              |        |          |
|----------------------------------|--------|----------|
| Expression                       | Type   | Variable |
| StringHandling.UPCASE(Cars.Make) | String | var1     |

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**.

| Var                              |        |          |
|----------------------------------|--------|----------|
| Expression                       | Type   | Variable |
| StringHandling.UPCASE(Cars.Make) | String | var1     |
| Color                            |        |          |
| ID_Reseller                      |        |          |

Sélectionnez une entrée dans la zone **Input** ou appuyez sur la touche **Maj** pour effectuer une sélection multiple.

Appuyez sur la touche **Ctrl** pour sélectionner des entrées dispersées dans une table ou pour sélectionner des entrées de diverses tables. Lorsque vous sélectionnez plusieurs entrées, la première sélection peut prendre une couleur grisée. Maintenez la touche **Ctrl** enfoncée pour glisser-déposer toutes les entrées en une fois. Une bulle d'aide contextuelle vous indique le nombre d'entrées sélectionnées.

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

| Comment faire pour...                                                                                                                     | Actions associées                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Insérer toutes les entrées sélectionnées comme variables séparées.                                                                        | Faites simplement un glisser-déposer vers la table Var. Des flèches latérales vous indiquent où la nouvelle entrée Var peut être insérée. Chacune des Input est insérée dans une cellule séparée.                                                      |
| Concaténer toutes les entrées sélectionnées avec une entrée Var existante.                                                                | 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. |
| Ecraser une entrée Var par les entrées Input sélectionnées, et les concaténer.                                                            | Déposez vers l'entrée Var appropriée qui se met alors en surbrillance. Puis appuyez sur <b>Ctrl</b> et relâchez. Toutes les entrées sélectionnées sont concaténées et écrasent la variable sélectionnée.                                               |
| Concaténer les entrées Input sélectionnées, avec plusieurs entrées Var mises en surbrillance. Créer de nouvelles lignes en cas de besoin. | Déposez vers des Var existantes puis appuyez sur <b>Maj</b> . Les premières entrées sont concaténées avec les entrées Var en surbrillance, et si nécessaire, de nouvelles lignes sont créées pour contenir les entrées restantes.                      |

### 8.2.2.1. 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.

### 8.2.2.2. 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.

### 8.2.3. Utilisation des expressions

Toutes les déclarations d'expressions (**Input**, **Var** ou **Output**) et de filtres peuvent être visualisées et modifiées directement dans les champs des expressions, dans l'éditeur d'expression.

#### 8.2.3.1. Accéder à l'Expression editor

Cet éditeur fournit le confort d'une fenêtre dédiée pour écrire des fonctions ou des commandes de transformation.

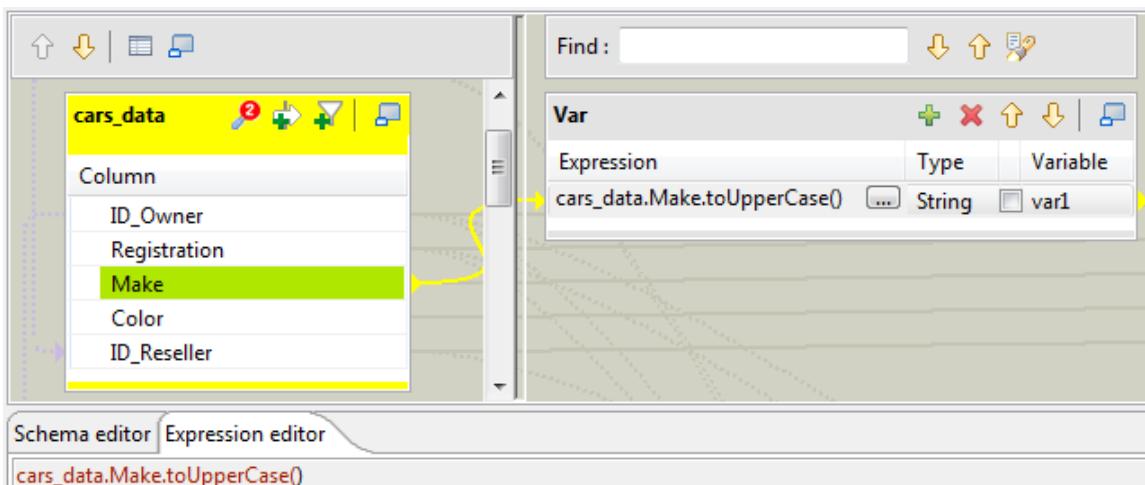
Vous pouvez écrire les expressions dont vous avez besoin pour la transformation de données directement dans la vue **Expression editor** située dans la partie inférieure de l'**Expression editor**.

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.



3. 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.

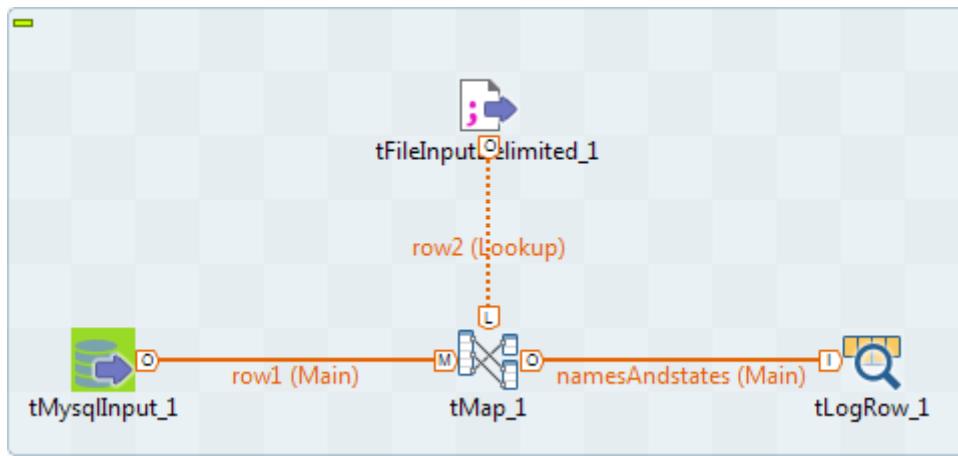
## 8.2.3.2. Créer du code à l'aide de l'Expression Builder

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 rédiger ce code (en Java), connu sous le nom d'expressions.

À l'aide de l'éditeur d'expressions du **tMap**, vous pouvez modifier l'expression pour une colonne d'entrée, une colonne de sortie ou une variable, ou modifier les expressions pour plusieurs colonnes de sortie en même temps.

### Modifier des expressions individuelles

L'exemple suivant vous montre l'utilisation de l'**Expression Builder** pour modifier deux expressions individuelles.



Dans cet exemple, deux flux d'entrée sont connectés au **tMap**.

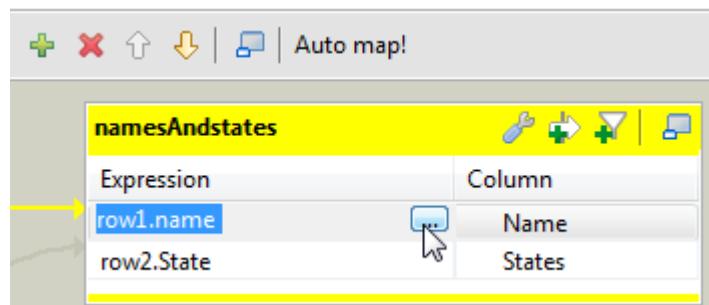
- Un premier flux DB input, contenant une liste de personnes dont le nom et le prénom sont séparés par un espace.
- Un autre flux File input, contenant la liste des états des États-Unis, en minuscule.

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

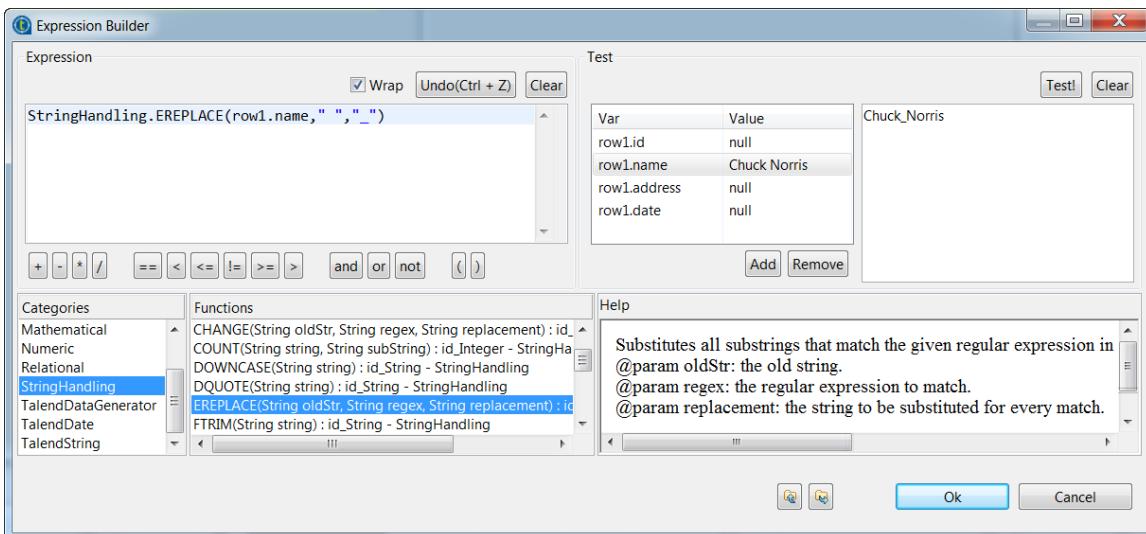
1. Dans le **[Map Editor]**, créez la jointure Inner Join correspondante pour créer le mapping de référence.

Pour plus d'informations concernant l'utilisation du **tMap**, consultez [Gestion des métadonnées dans l'intégration de données](#).

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. Cliquez sur le premier champ **Expression** (*row1.Name*) puis cliquez sur le bouton [...] qui s'affiche à côté de l'expression.



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

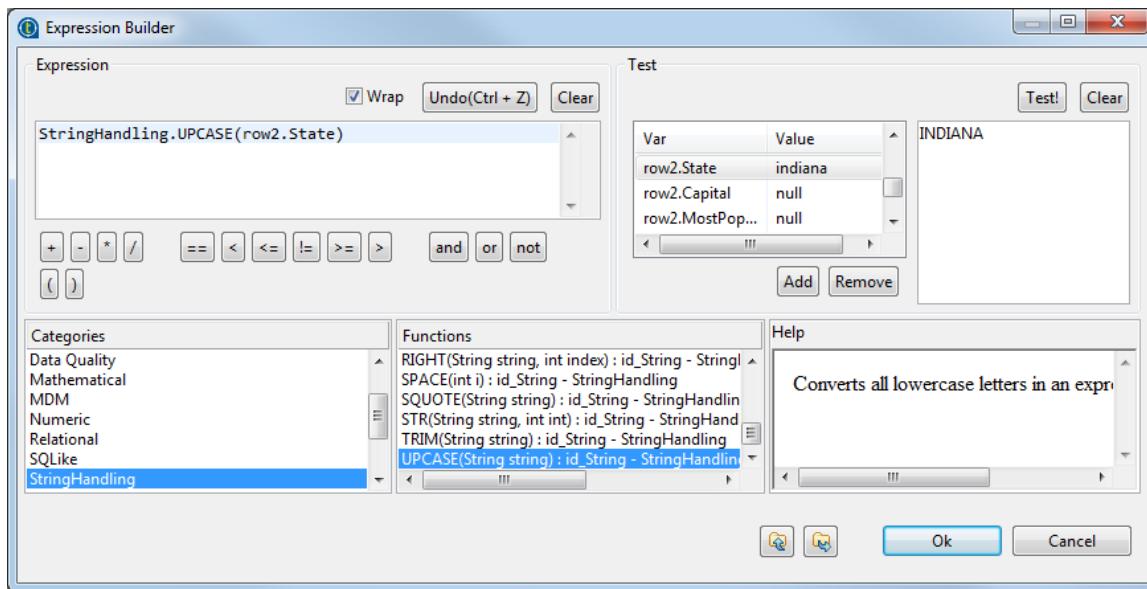


4. Dans la liste **Category**, sélectionnez l'action que vous souhaitez effectuer. Pour cet exemple, choisissez **StringHandling**, puis sélectionnez la fonction **EREPLACE** dans la liste **Functions**.
5. Dans la zone **Expression**, collez *row1.Name* à la place du texte, pour obtenir : **StringHandling.EREPLACE(row1.Name, " ", "\_")**. Cette expression remplacera l'espace par un tiret souligné dans la chaîne de caractères fournie.

Notez que les fonctions **CHANGE** et **EREPLACE** de la catégorie **StringHandling** sont utilisées pour substituer toutes les sous-chaînes de caractères correspondant à l'expression régulière donnée dans l'ancienne chaîne de caractères, par le remplacement donné et retourner une nouvelle chaîne de caractères. Leurs trois paramètres sont :

- **oldStr** : l'ancienne chaîne de caractères.
- **newStr** : l'expression régulière avec laquelle effectuer la correspondance.
- **replacement** : la chaîne de caractères à substituer pour chaque correspondance.

6. Vérifiez votre code en renseignant le champ **Value** correspondant dans la zone **Test**, par exemple saisissez *Chuck Norris* puis cliquez sur **Test!** et vérifiez que les changements ont bien été effectués, ici : *Chuck\_Norris*
7. Cliquez sur **OK** pour valider et répétez la même opération pour la deuxième colonne (*State*).
8. Dans le **tMap** de sortie, sélectionnez l'expression *row2.State* et cliquez sur le bouton [...] pour ouvrir de nouveau l'**Expression builder**.



Cette fois, la fonction StringHandling à utiliser est UPCASE. L'expression complète est : `StringHandling.UPCASE(row2.State)`.

- 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**.

| Expression                                                | Column |
|-----------------------------------------------------------|--------|
| <code>StringHandling.EREPLACE(row1.name, " ", "_")</code> | Name   |
| <code>StringHandling.UPCASE(row2.State)</code>            | States |

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 :

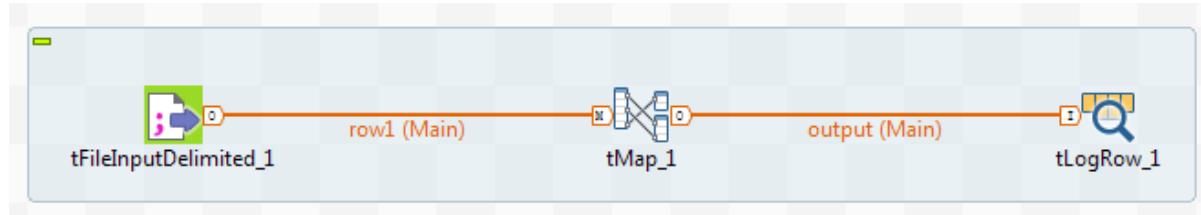
*Starting job NamesAndStates at 10:02 10/10/2007.*

| 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 |

## Configurer des expressions pour plusieurs colonnes de sortie simultanément

Le **tMap** vous permet de définir le comportement de transformation de plusieurs colonnes de sortie en même temps.

En utilisant un Job simple de transformation, l'exemple suivant vous montre comment définir les expressions sur plusieurs colonnes, à la manière d'un lot, dans le **tMap**.



Voici le contenu du fichier CSV d'entrée utilisé dans cet exemple :

```

id;firstname;lastname;city;state
1; Andrew; Adams; Madison; Rhode Island
2; Andrew; Garfield; Saint Paul; Colorado
3; Woodrow; Eisenhower; Juneau; New Hampshire
4; Woodrow; Jackson; Denver; Maine
5; Lyndon; Buchanan; Pierre; Kentucky
6; Bill; Tyler; Helena; New York
7; George; Adams; Oklahoma City; Alaska
8; Ulysses; Garfield; Santa Fe; Massachusetts
9; Thomas; Coolidge; Charleston; Mississippi
10; John; Polk; Carson City; Louisiana

```

Dans cet exemple, dans toutes les colonnes de sortie de type String, les espaces en début et fin de chaîne seront supprimés et les noms de famille et d'États seront passés en majuscules.

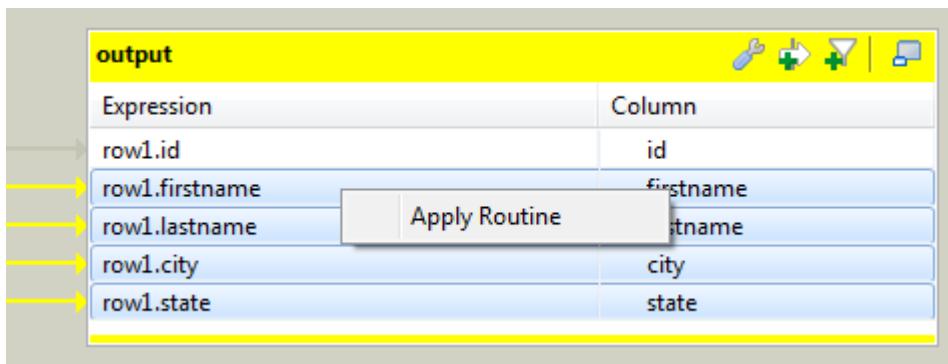
1. Dans l'éditeur de mapping, effectuez les mappings de l'entrée vers la sortie.

| Column    | Key | Type    | ✓ N..                               | Date Patt... | Len... | Prec... | De... |
|-----------|-----|---------|-------------------------------------|--------------|--------|---------|-------|
| id        |     | Integer | <input checked="" type="checkbox"/> |              |        |         |       |
| firstname |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |
| lastname  |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |
| city      |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |
| state     |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |

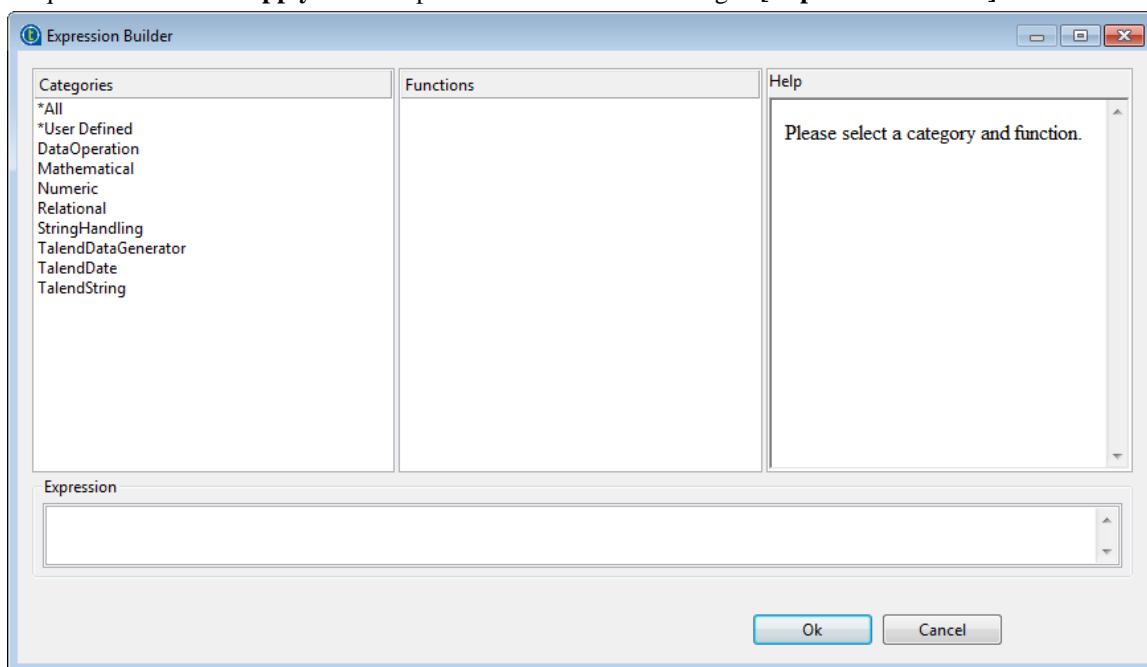
  

| Column    | Key | Type    | ✓ N..                               | Date Patt... | Len... | Prec... | De... |
|-----------|-----|---------|-------------------------------------|--------------|--------|---------|-------|
| id        |     | Integer | <input checked="" type="checkbox"/> |              |        |         |       |
| firstname |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |
| lastname  |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |
| city      |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |
| state     |     | String  | <input checked="" type="checkbox"/> |              |        |         |       |

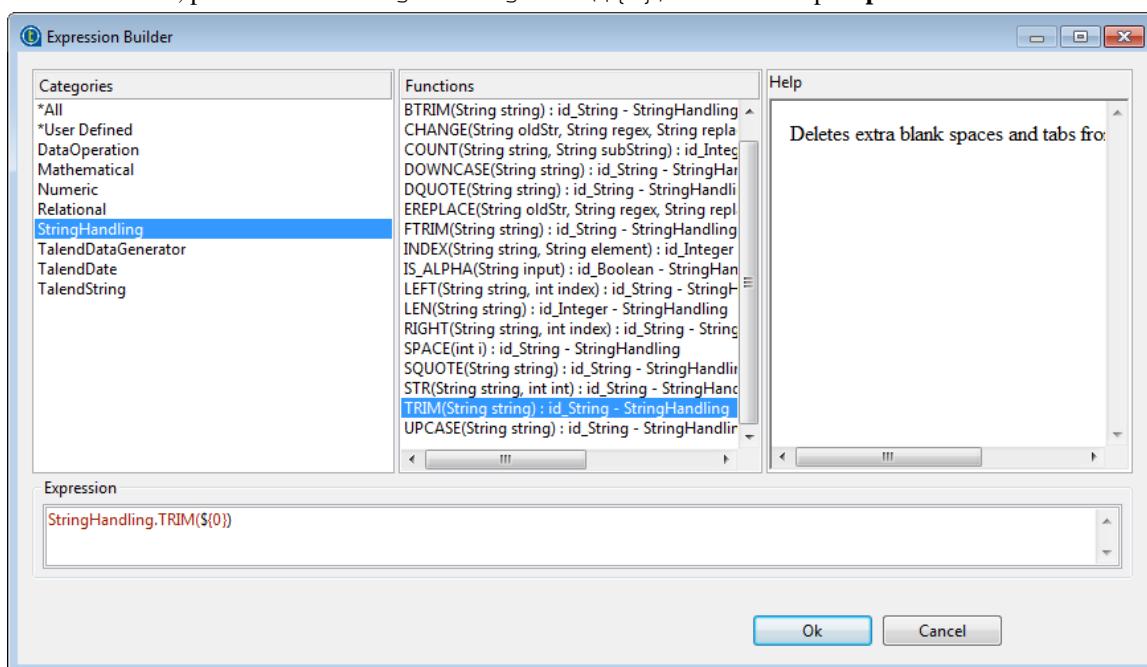
2. Sélectionnez les colonnes de type String dans la table de sortie, respectivement *firstname*, *lastname*, *city* et *state* dans cet exemple, puis cliquez-droit sur la sélection, afin d'afficher le bouton **Apply Routine**.



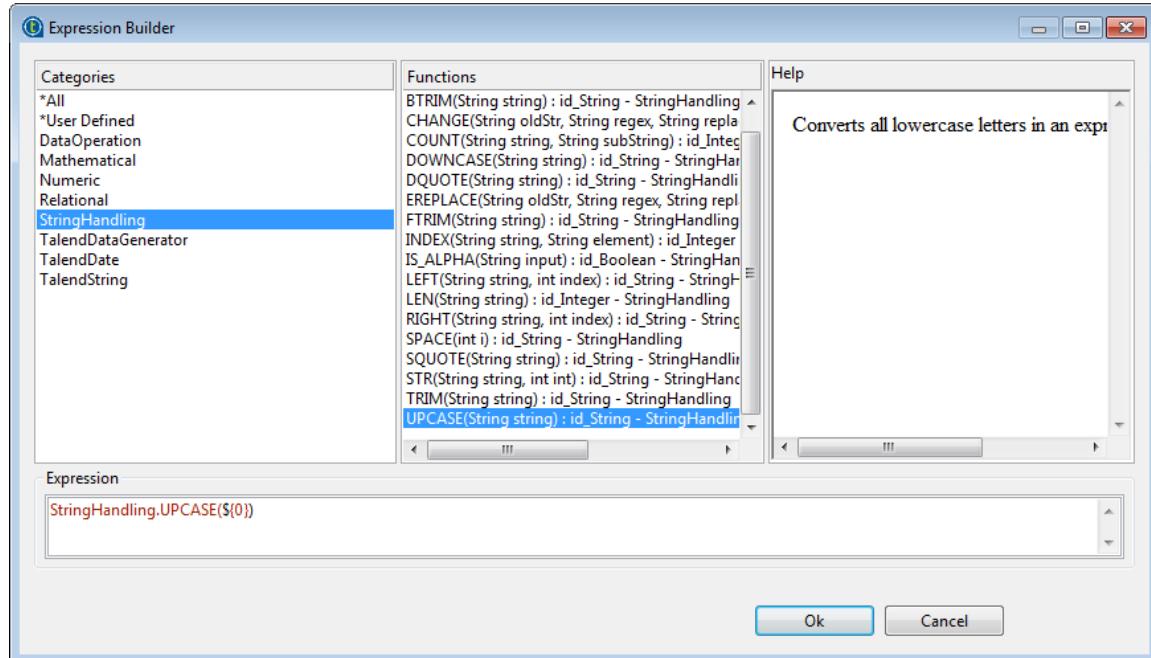
3. Cliquez sur le bouton **Apply Routine** pour ouvrir la boîte de dialogue [Expression Builder].



4. Sélectionnez **StringHandling** dans la zone **Categories**, puis double-cliquez sur la fonction **TRIM** dans la zone **Functions**, pour obtenir **StringHandling.TRIM(\${0})** dans le champ **Expression**.



5. Cliquez sur **OK** pour fermer la boîte de dialogue [**Expression Builder**].
6. Sélectionnez les colonnes *lastname* et *state* dans la table de sortie, dans l'éditeur de mapping, cliquez-droit sur la sélection puis cliquez sur le bouton **Apply Routine** pour ouvrir la boîte de dialogue [**Expression Builder**].
7. Sélectionnez **StringHandling** dans la zone **Categories**, puis double-cliquez sur la fonction **UPPERCASE** dans la zone **Functions** pour obtenir la fonction **StringHandling.UPCASE(\${0})** dans le champ **Expression**.



8. Cliquez sur **OK** pour fermer la boîte de dialogue [**Expression Builder**].

Les expressions de ces colonnes ressemblent à ceci :

| output                                                     |           |
|------------------------------------------------------------|-----------|
| Expression                                                 | Column    |
| row1.id                                                    | id        |
| StringHandling.TRIM(row1.firstname )                       | firstname |
| StringHandling.UPCASE(StringHandling.TRIM(row1.lastname )) | lastname  |
| StringHandling.TRIM(row1.city )                            | city      |
| StringHandling.UPCASE(StringHandling.TRIM(row1.state ))    | state     |

Les fonctions seront exécutées au cours du traitement du flux. La sortie se présente comme suit, dans cet exemple.

Execution

**Run**    **Kill**    **Clear**

```
[statistics] connected
+-----+-----+-----+
| tLogRow_1
+-----+-----+-----+
| id | firstname | lastname | city | state
+-----+-----+-----+
| 1 | Andrew | ADAMS | Madison | RHODE ISLAND
| 2 | Andrew | GARFIELD | Saint Paul | COLORADO
| 3 | Woodrow | EISENHOWER | Juneau | NEW HAMPSHIRE
| 4 | Woodrow | JACKSON | Denver | MAINE
| 5 | Lyndon | BUCHANAN | Pierre | KENTUCKY
| 6 | Bill | TYLER | Helena | NEW YORK
| 7 | George | ADAMS | Oklahoma City | ALASKA
| 8 | Ulysses | GARFIELD | Santa Fe | MASSACHUSETTS
| 9 | Thomas | COOLIDGE | Charleston | MISSISSIPPI
| 10 | John | POLK | Carson City | LOUISIANA
+-----+-----+-----+
[statistics] disconnected
```

## 8.2.4. Paramètres Output

Dans l'espace de modélisation du *Studio Talend*, la création d'une connexion Row à partir du composant **tMap** vers des composants en sortie, a pour effet d'ajouter les schémas Output correspondants dans la zone **Output** du **[Map Editor]**.

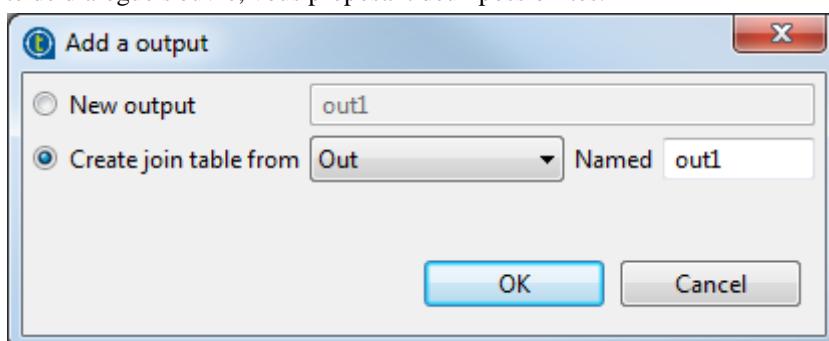
Vous pouvez également ajouter un schéma Output dans votre **[Map Editor]**, à l'aide du signe **[+]** de la barre d'outils de la zone **Output**.

Vous pouvez aussi effectuer une jointure entre vos tables de sortie. La jointure sur les tables de sortie permet de traiter les flux séparément, mais de les unifier en sortie.



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...                                                                                                                                                                                                                |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>New output</b>             | Ajouter une table de sortie.                                                                                                                                                                                           |
| <b>Create join table from</b> | 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 <b>Named</b> , 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 + <b>Ctrl</b>                   | Remplace les expressions en surbrillance par les expressions sélectionnées.                                            |
| Glisser-déposer + <b>Maj</b>                    | Ajoute les champs sélectionnés à toutes les expressions en surbrillance. Insère de nouvelles lignes si nécessaire.     |
| Glisser-déposer + <b>Ctrl + Maj</b>             | 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.

### 8.2.4.1. Créer des expressions complexes

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

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

Pour plus d'informations concernant l'Expression Builder, consultez [Créer du code à l'aide de l'Expression Builder](#)

### 8.2.4.2. Utiliser les filtres

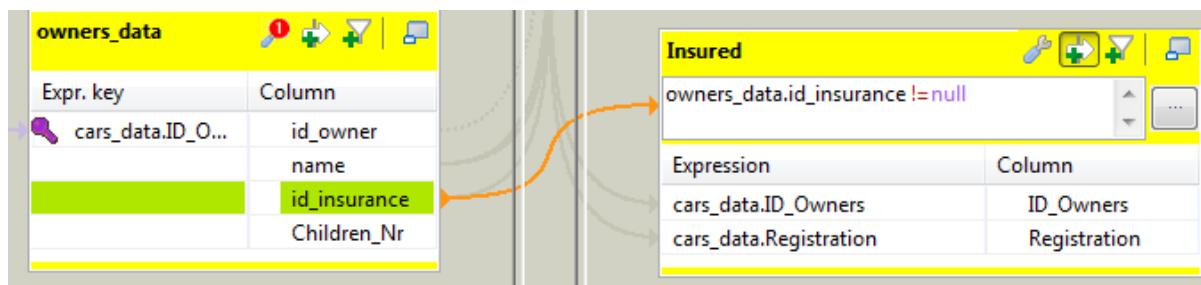
Les conditions de filtre vous permettent de sélectionner les champs et de les envoyer vers les sorties appropriées.

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.

### 8.2.4.3. Utiliser la fonction Rejet

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

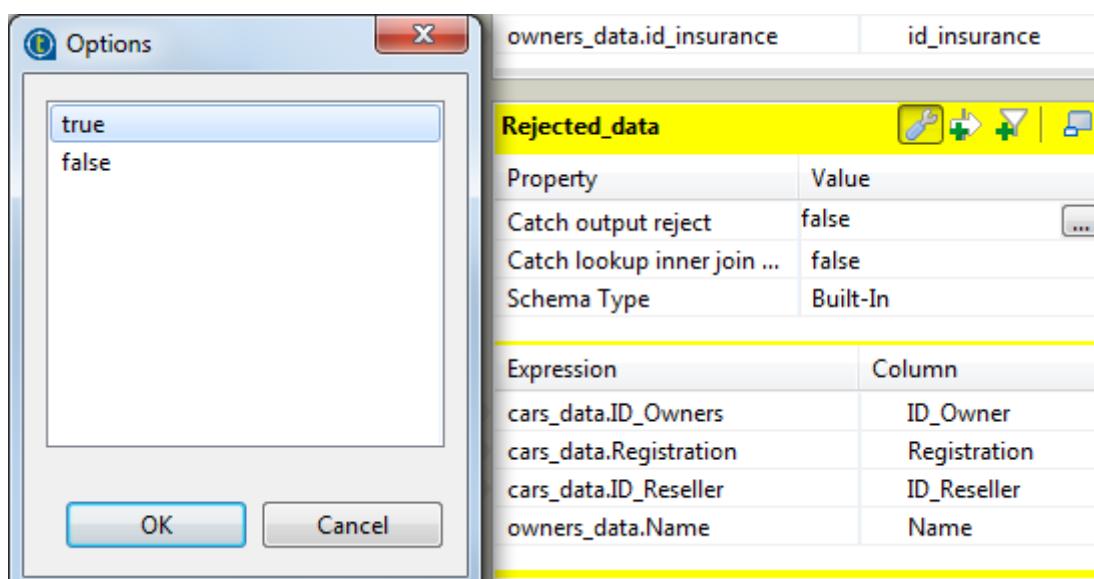
Cette table de rejet rassemble toutes les données qui ne satisfont pas un ou plusieurs des filtres définis dans les tables Output standard. Notez que par table standard sont désignées toutes les tables qui ne sont pas des tables de rejet.

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.

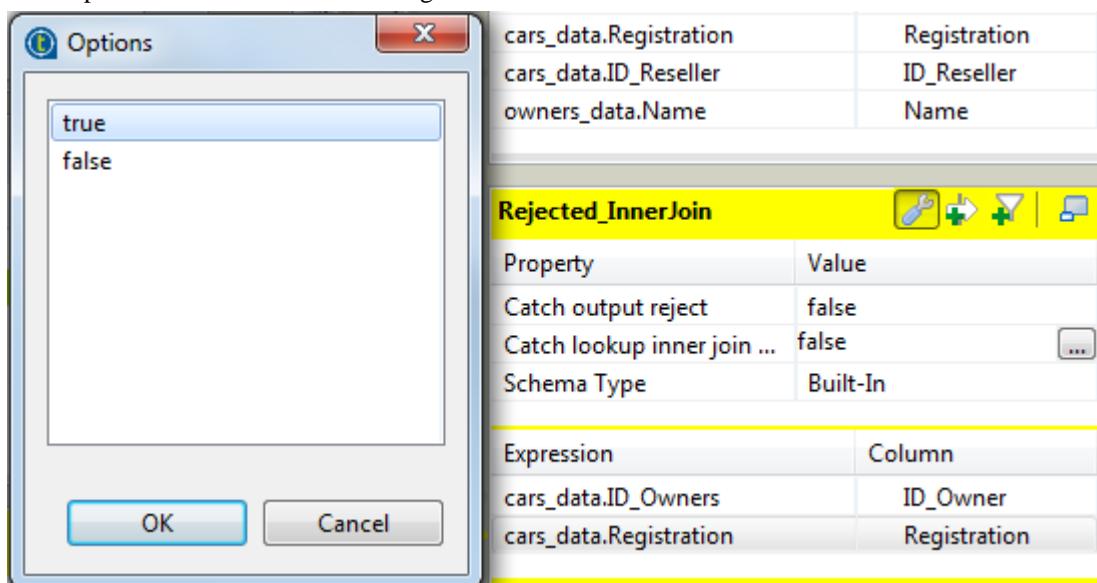
Les données ne sont pas exclusivement traitées vers une seule sortie. Même si une donnée satisfait le filtre de sortie et qu'elle est donc routée vers elle, elle est également vérifiée contre les autres filtres et peut être également routée vers d'autres sorties.

#### 8.2.4.4. 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.



#### 8.2.4.5. 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.

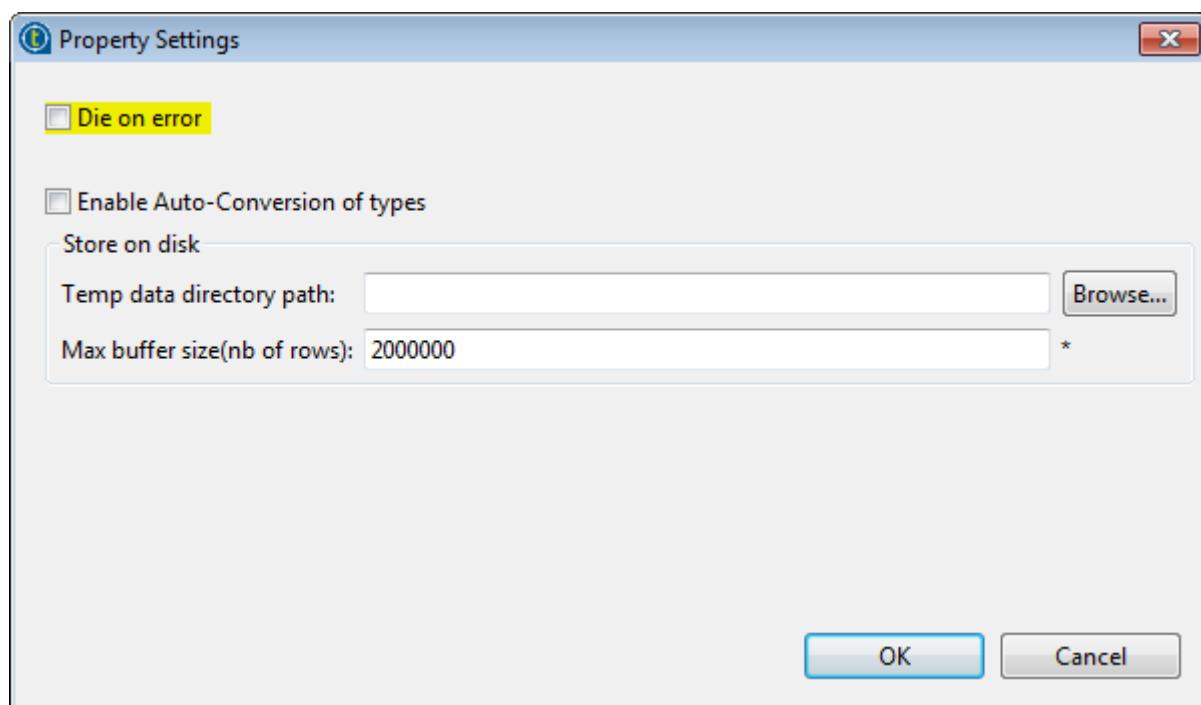
## 8.2.4.6. Gestion des erreurs

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.

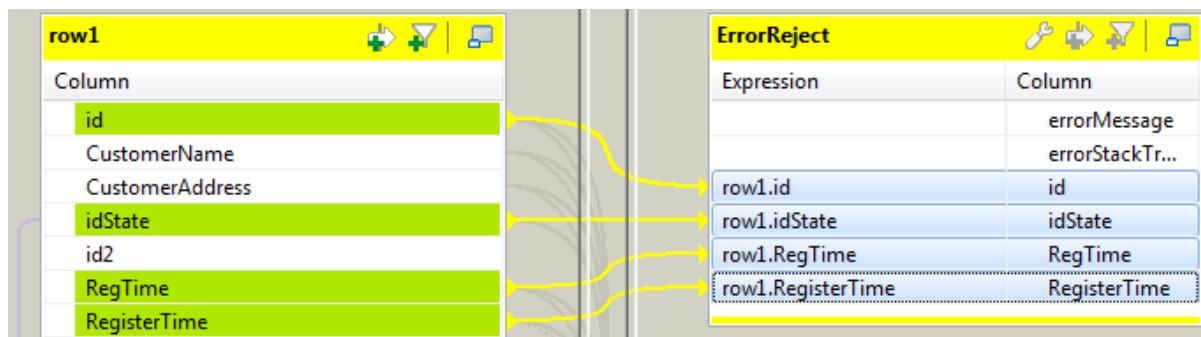
Pour désactiver l'option **Die on error** :

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**.

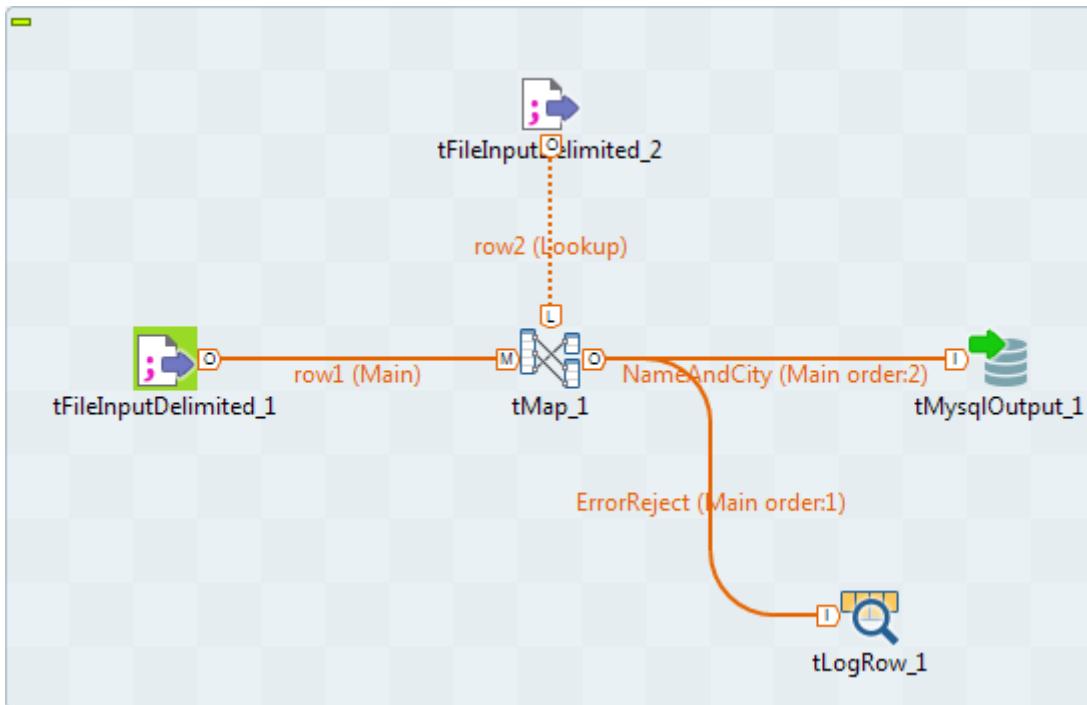


Une nouvelle table appelée **ErrorReject** s'affiche dans l'éditeur du **tMap**, dans la zone de sortie. Cette table de sortie comprend automatiquement deux colonnes : **errorMessage** et **errorStackTrace**, qui retrouvent le message et la trace de la pile de l'erreur rencontrée durant l'exécution du 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 "|".

## 8.2.5. Configuration des schémas dans le Map Editor

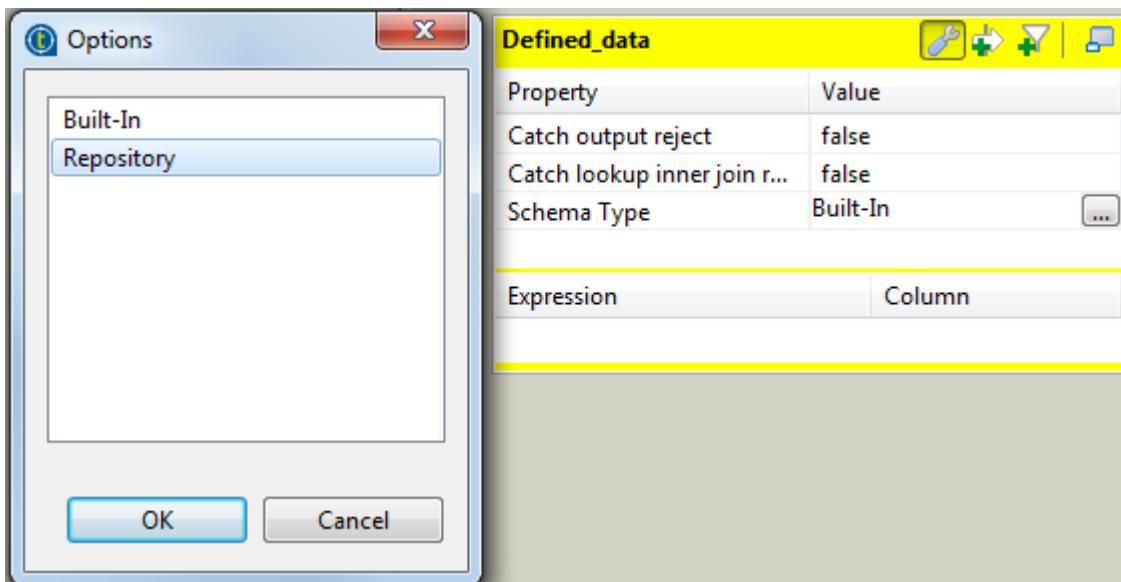
Dans le **Map Editor**, vous pouvez définir le type de schéma d'une table comme **Built-In** afin de pouvoir modifier la structure des données dans le panneau **Schema editor**, ou vous pouvez le définir comme **Repository** et récupérer

la structure des données depuis le Repository. Par défaut, le type du schéma est défini comme **Built-In** pour toutes les tables.

### 8.2.5.1. 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 displays the 'Defined\_data' configuration in the Map Editor. The top part is a table with properties:

| Property                       | Value                   |
|--------------------------------|-------------------------|
| Catch output reject            | false                   |
| Catch lookup inner join reject | false                   |
| Schema Type                    | Repository              |
| Schema Id                      | DELIM:owners - metadata |

The bottom part shows the mapping between an expression and columns:

| Expression | Column       |
|------------|--------------|
|            | ID_Owner     |
|            | Name         |
|            | ID_Insurance |
|            | Children_Nr  |

Below this, a detailed view of the schema table is shown:

| Column      | Key                      | T... | <input checked="" type="checkbox"/> | N... | Date ... | L... | Pr... | D... | C... |
|-------------|--------------------------|------|-------------------------------------|------|----------|------|-------|------|------|
| ID_Owner    | <input type="checkbox"/> | I... | <input checked="" type="checkbox"/> |      |          | 2    | 0     |      |      |
| Name        | <input type="checkbox"/> | S... | <input checked="" type="checkbox"/> |      |          | 19   | 0     |      |      |
| ID_Insur... | <input type="checkbox"/> | S... | <input checked="" type="checkbox"/> |      |          | 3    | 0     |      |      |
| Childre...  | <input type="checkbox"/> | I... | <input checked="" type="checkbox"/> |      |          | 2    | 0     |      |      |



*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**.

### 8.2.5.2. Rechercher des colonnes de schéma

Le filtre sur les colonnes du schéma du **tMap** vous permet de rechercher rapidement, en une fois, une colonne d'entrée ou de sortie parmi les centaines de colonnes.

L'exemple suivant vous montre comment trouver des colonnes contenant la chaîne de caractères "customer" dans la table de sortie, dans l'éditeur de mapping.

- Ouvrez l'éditeur de mapping et cliquez sur le bouton en haut de la table, pour ouvrir la zone de filtre.

| Expression            | Column           |
|-----------------------|------------------|
| row2.id               | id               |
| row1.CustomerName     | customerName     |
| row2.age              | age              |
| row1.CustomerAddress  | customerAddress  |
| row2.city             | city             |
| row1.idState          | idState          |
| row2.carModel         | carModel         |
| row2.madeIn           | madeIn           |
| row2.dealer           | dealer           |
| row2.shopName         | shopName         |
| row2.discount         | discount         |
| row2.totalPrice       | totalPrice       |
| row2.salesManager     | salesManager     |
| row2.insuranceCompany | insuranceCompany |
| row2.insureTvinc      | insureTvinc      |

2. Dans la zone de filtre, saisissez votre recherche, *customer* dans cet exemple.

Lorsque vous commencez à saisir, la table affiche les colonnes correspondant aux caractères.

| Expression                  | Column                 |
|-----------------------------|------------------------|
| row1.CustomerName           | customerName           |
| row1.CustomerAddress        | customerAddress        |
| row2.customerServiceHistory | customerServiceHistory |

### 8.2.5.3. 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**.

| Column       | Key                                 | Type     | <input checked="" type="checkbox"/> | N.. | Date Patter... | Length | Preci... | De... | Com... |
|--------------|-------------------------------------|----------|-------------------------------------|-----|----------------|--------|----------|-------|--------|
| ID_Owner     | <input checked="" type="checkbox"/> | Integ... | <input checked="" type="checkbox"/> |     |                | 2      | 0        |       |        |
| First N      | <input type="checkbox"/>            | String   | <input type="checkbox"/>            |     |                | 19     | 0        |       |        |
| ID_Insurance | <input type="checkbox"/>            | String   | <input checked="" type="checkbox"/> |     |                | 3      | 0        |       |        |
| Children_Nr  | <input type="checkbox"/>            | Integ... | <input checked="" type="checkbox"/> |     |                | 2      | 0        |       |        |

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                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Column</b>    | Nom de colonne tel que défini dans le schéma du <b>[Map Editor]</b> et dans le schéma du composant Input ou Output correspondant.                                                                          |
| <b>Key</b>       | 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. |
| <b>Type</b>      | Type de données : Chaîne de caractères (String) ou entier (Integer).                                                                                                                                       |
|                  | Cette colonne est obligatoire dans la version Java.                                                                                                                                                        |
| <b>Length</b>    | -1 indique qu'aucune valeur de longueur n'a été définie dans le schéma.                                                                                                                                    |
| <b>Precision</b> | Précise le nombre de chiffres, à droite de la virgule.                                                                                                                                                     |
| <b>Nullable</b>  | Décochez cette case si vous ne souhaitez pas autoriser les valeurs de champs nulles.                                                                                                                       |
| <b>Default</b>   | Indique les valeurs par défaut, si elles sont définies.                                                                                                                                                    |
| <b>Comment</b>   | 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.

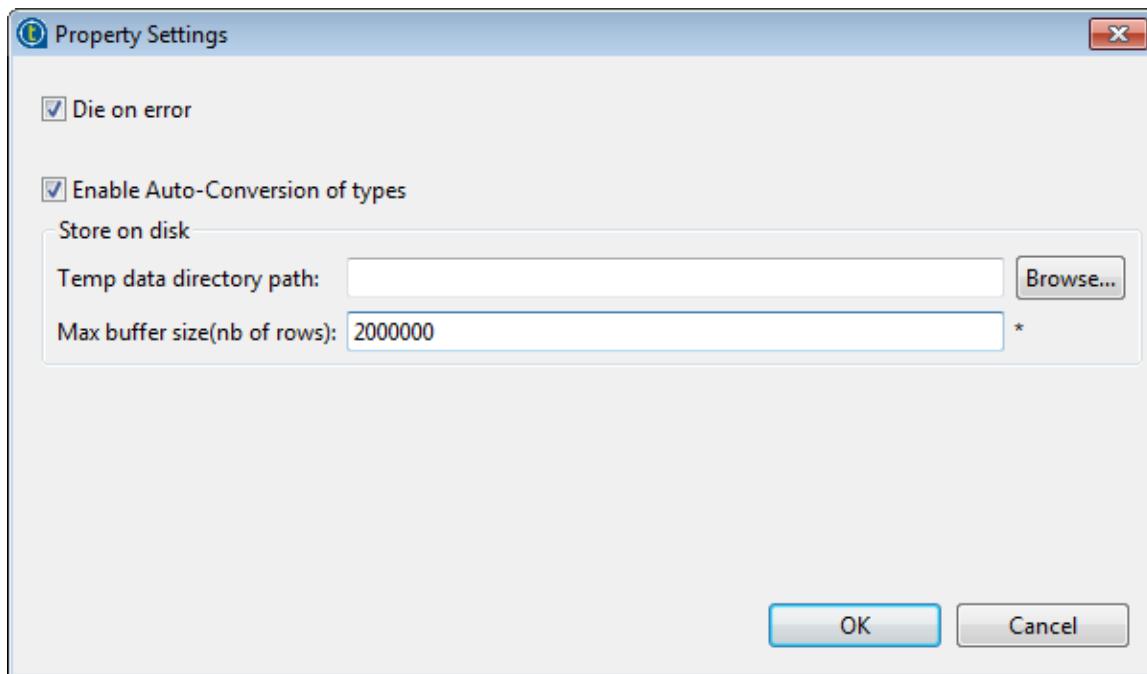
Un arrière-plan rouge s'affiche lorsqu'un caractère non valide a été saisi. La plupart des caractères spéciaux sont interdits pour réduire les risques de mauvaise interprétation dans le code. Les caractères autorisés sont : les minuscules et les majuscules, les chiffres (à l'exception du caractère de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

## 8.2.6. Activation de la conversion automatique des types de données

Lors du traitement des flux de données avec le **tMap**, si les colonnes d'entrée et de sortie dans le mapping sont de différents types de données, des erreurs de compilation peuvent survenir pendant l'exécution du Job. L'option **Enable Auto-Conversion of types** dans le **tMap** permet d'éviter ce type d'erreur.

Pour activer cette fonctionnalité dans un **tMap** dans un Job :

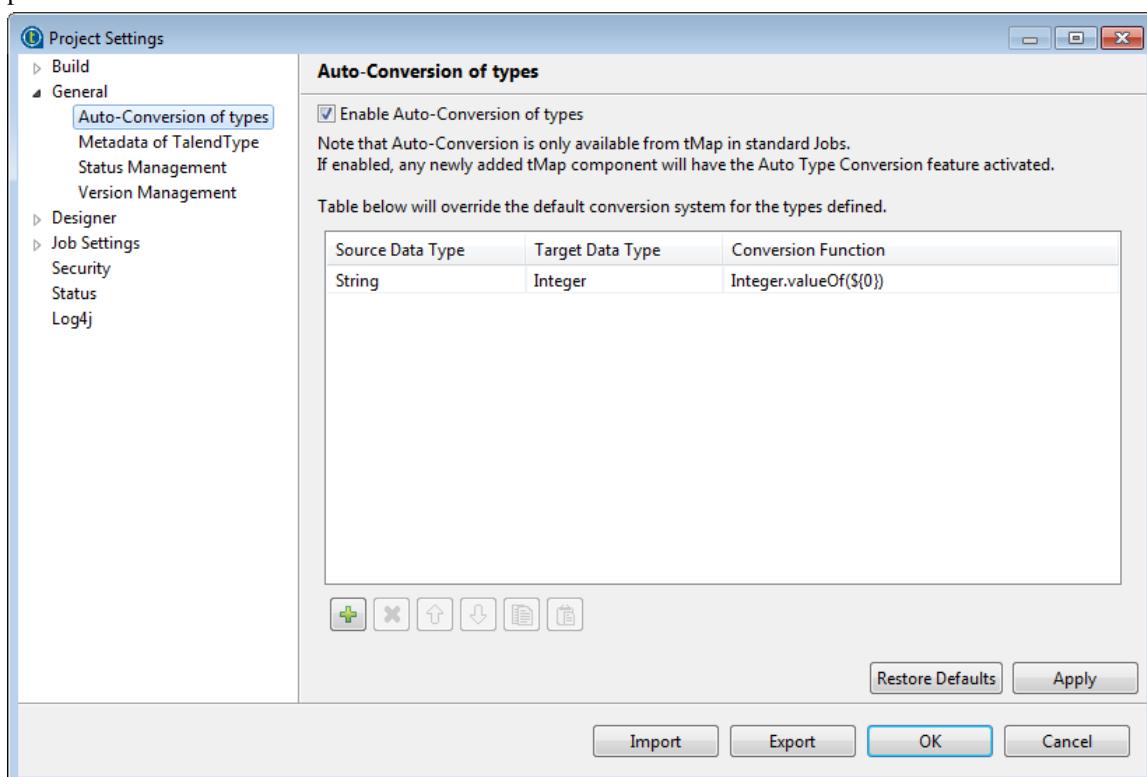
1. Cliquez sur le bouton en haut de l'éditeur de mapping pour ouvrir la boîte de dialogue **[Property Settings]**.
2. Cochez la case **Enable Auto-Conversion of types** puis cliquez sur **OK**.



Vous pouvez activer l'option de conversion automatique au niveau du projet afin que cette fonctionnalité soit activée dans tous les composants **tMap** ajoutés par la suite au projet.

Si nécessaire, vous pouvez également définir des règles de conversion pour écraser le comportement de conversion par défaut du **tMap**.

1. Dans la barre d'outils de la fenêtre principale du Studio, cliquez sur ou sur **File > Edit Project Properties** dans la barre du menu afin d'ouvrir la boîte de dialogue **[Project Settings]**.
2. Dans l'arborescence de la boîte de dialogue, développez **General** et sélectionnez **Auto-Conversion of types** pour ouvrir cette vue.



3. Cochez la case **Enable Auto-Conversion of types** afin d'activer la fonctionnalité de conversion automatique du type pour tous les composants **tMap** ajoutés par la suite au projet.
4. Si nécessaire, cliquez sur le bouton **[+]** pour ajouter une ligne et sélectionnez les types source et cible. Définissez ensuite une fonction Java pour la conversion du type de données afin de créer une règle de conversion écrasant le comportement par défaut de conversion du **tMap** pour les données correspondant à la règle.

Vous pouvez appuyer sur les touches **Ctrl+Espace** dans le champ **Conversion Function** pour accéder à une liste de fonctions Java disponibles.

Vous pouvez créer autant de règles de conversion que vous le souhaitez.

5. Cliquez sur **Apply** pour appliquer vos changements, puis sur **OK** afin de fermer la boîte de dialogue.

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

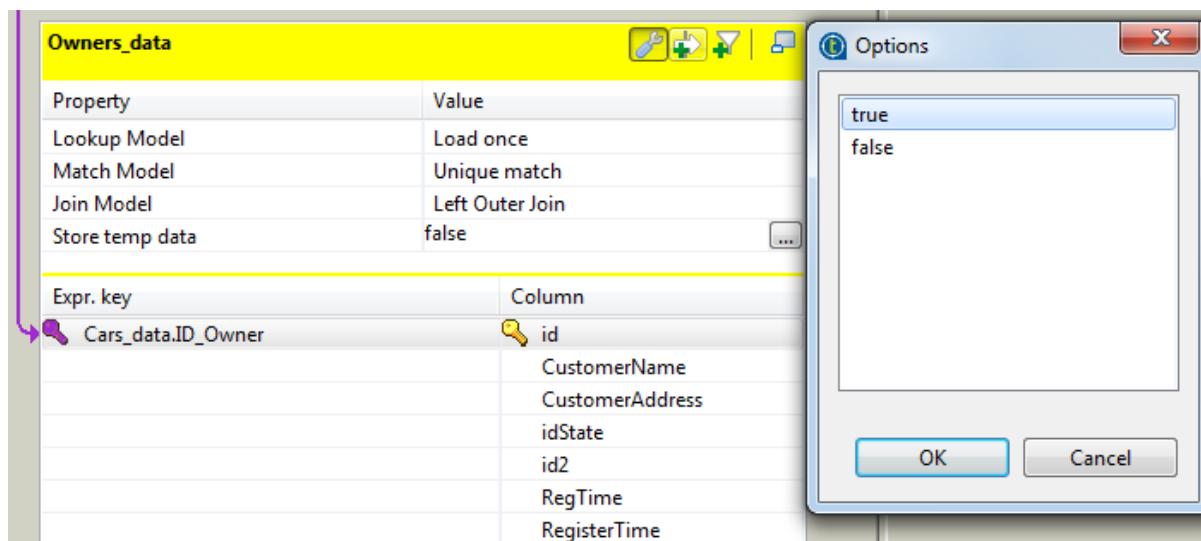
Lorsque vous devez traiter un nombre important de données, par exemple, de nombreuses colonnes, différents types de colonnes ou lignes, votre système peut rencontrer des problèmes de mémoire insuffisante empêchant votre Job de s'exécuter correctement, et plus particulièrement lorsque vous utilisez un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

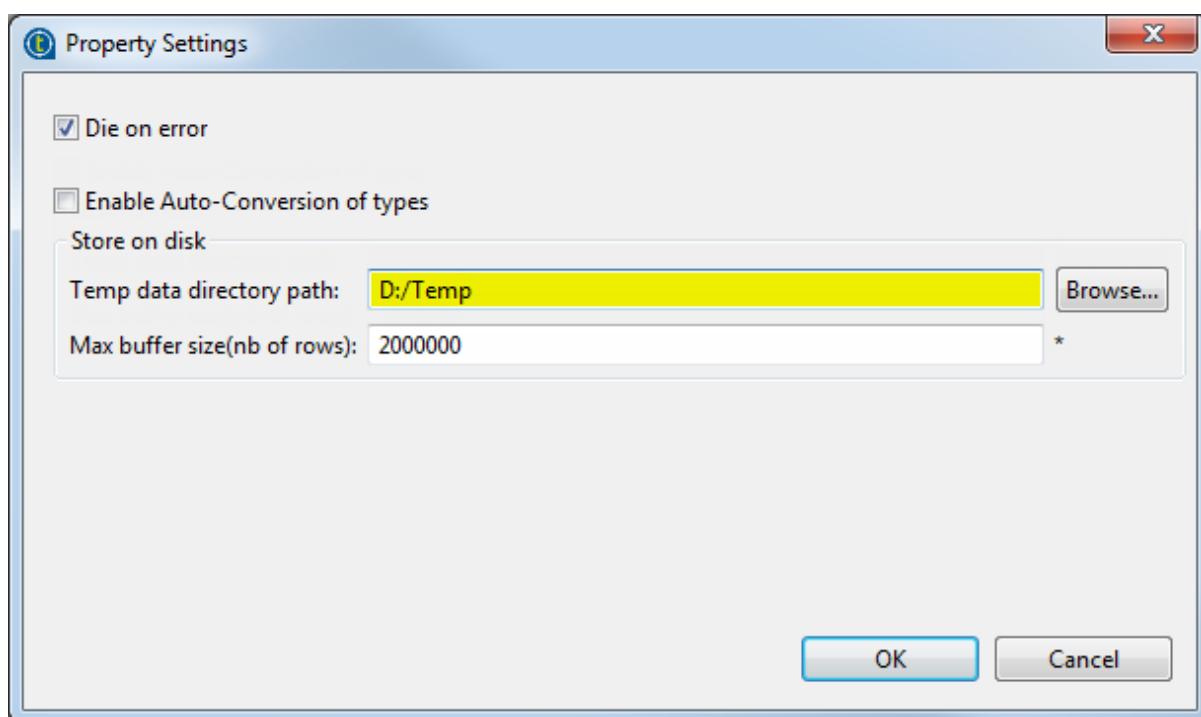
1. Double-cliquez sur le composant **tMap** de votre Job pour lancer le **[Map Editor]**.
2. Dans le panneau de gauche, correspondant aux sources **Input** et **Lookup**, cliquez sur la table **Lookup** affichant les données de Lookup que vous souhaitez charger sur le disque pour ne pas encombrer la 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 paramétrer le répertoire de stockage temporaire et la taille du buffer à partir du **Map Editor** :

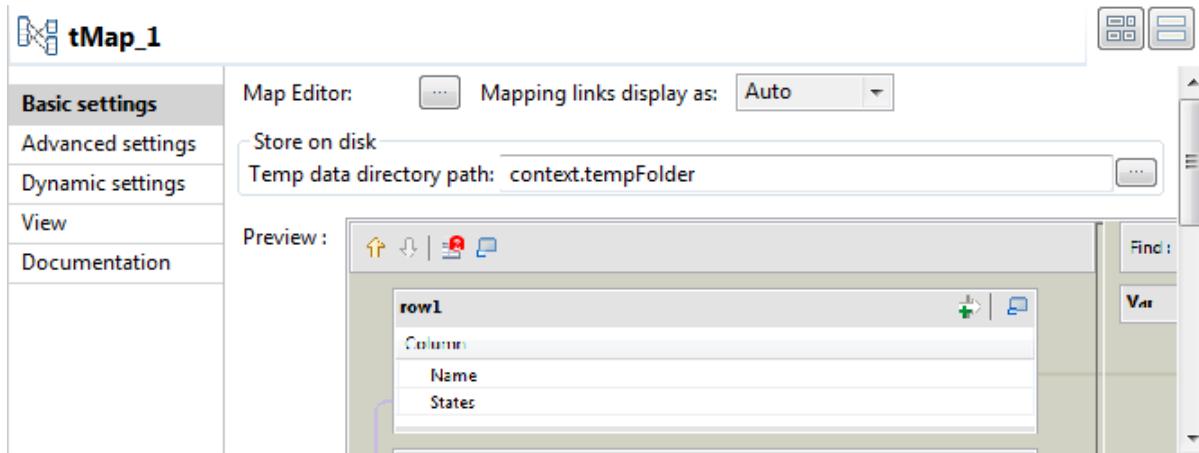
1. Cliquez sur le bouton **Property Settings** en haut de la zone Input pour ouvrir la boîte de dialogue [**Property Settings**].
2. Dans la boîte de dialogue [**Property Settings**], renseignez le chemin d'accès complet vers le dossier dans lequel les données temporaire seront stockées dans le champ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



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

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

Vous pouvez aussi utiliser une variable de contexte à l'aide du raccourci **Ctrl+Espace** si vous avez configuré une variable dans un groupe de contexte dans le **Repository**. Pour plus d'informations sur les contextes, consultez [Utiliser 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.



Étant donné que l'écriture du flux principal sur le disque nécessite le stockage de données, l'ordre des lignes en sortie peut ne pas être respecté.

Dans la vue **Advanced settings**, vous pouvez aussi ajouter un buffer, au besoin. Pour cela, renseignez le champ **Max. buffer size (Nb of rows)** afin de séparer les données stockées sur le disque en autant de fichiers que nécessaire.

## 8.2.8. Gestion des références

Lorsque vous implémentez une jointure (notamment **Inner Join** et **Left Outer Join**) dans un **tMap**, sur différentes sources de données, il y a toujours un flux principal (Main) et un ou plusieurs flux de référence (Lookup) connectés au **tMap**. Tous les enregistrements du flux de référence doivent être chargés avant de traiter chaque enregistrement du flux principal. Trois types de modèles de chargement de Lookup sont fournis, correspondants à différents besoins métier et besoins en performances : **Load once**, **Reload at each row** et **Reload at each row (cache)**.

- **Load once** : charge une fois (une fois seulement) tous les enregistrements du flux de référence, soit dans la mémoire, soit dans un fichier local, avant de traiter chaque enregistrement du flux principal, si l'option **Store temp data** est configurée à **true**. Cette configuration est celle par défaut et est l'option recommandé si vous avez un jeu d'enregistrements volumineux à traiter dans le flux principal, à l'aide d'une jointure entre les deux flux.
- **Reload at each row** : charge tous les enregistrements du flux de référence pour chaque enregistrement du flux principal. Généralement, cette option augmente le temps d'exécution du Job, à cause du chargement répété des enregistrements du flux de référence à chaque enregistrement du flux principal. Cependant, cette option est recommandée dans les situations suivantes :
  - le flux de données de référence est constamment mis à jour et vous souhaitez charger les dernières données de référence pour chaque enregistrement du flux principal, pour obtenir les données les plus récentes après exécution de la jointure ;

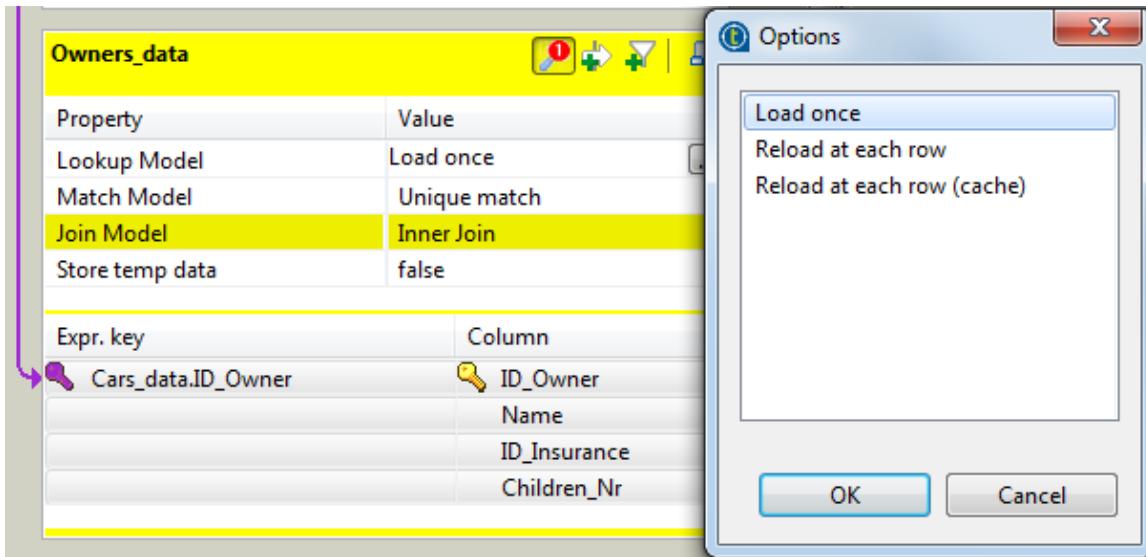
- les données sont peu nombreuses dans le flux principal et le flux de référence contient de nombreuses données d'une table de base de données. Dans ce cas, l'utilisation de l'option **Load once** peut causer une erreur de mémoire *OutOfMemory*. Vous pouvez utiliser des paramètres variables dynamiques comme une clause WHERE pour mettre à jour le flux de référence à la volée durant le chargement, avant traitement de la jointure du flux principal. Par exemple, consultez [Recharger des données à chaque ligne](#).
- Reload at each row (cache)** : fonctionne comme le modèle **Reload at each row**, tous les enregistrements du flux de référence sont chargés pour chaque enregistrement du flux principal. Cependant, ce modèle ne peut être utilisé avec l'option **Store temp data on disk**. Les données de référence sont en cache dans la mémoire et, lorsqu'un nouveau chargement survient, seuls les enregistrements qui n'existent pas déjà dans le cache seront chargés, pour éviter deux fois les mêmes enregistrements. Cette option permet d'optimiser le temps de traitement et d'améliorer les performances de traitement du composant **tMap**. Notez que vous ne pouvez pas utiliser les options **Reload at each row (cache)** et **Store temp data** en même temps.

Notez que, lorsque votre référence est une table de base de données, la bonne pratique est d'ouvrir la connexion au début du Job, afin d'optimiser les performances.

### 8.2.8.1. Configurer le mode de chargement d'un flux de référence

Pour configurer le mode de chargement d'un flux de référence :

- Cliquez sur le bouton **tMap settings** en haut à droite de la table de référence, pour afficher les propriétés de la table.
- Cliquez sur le champ **Value** correspondant à l'option **Lookup Model**, puis cliquez sur le bouton [...] pour ouvrir la boîte de dialogue **[Options]**.



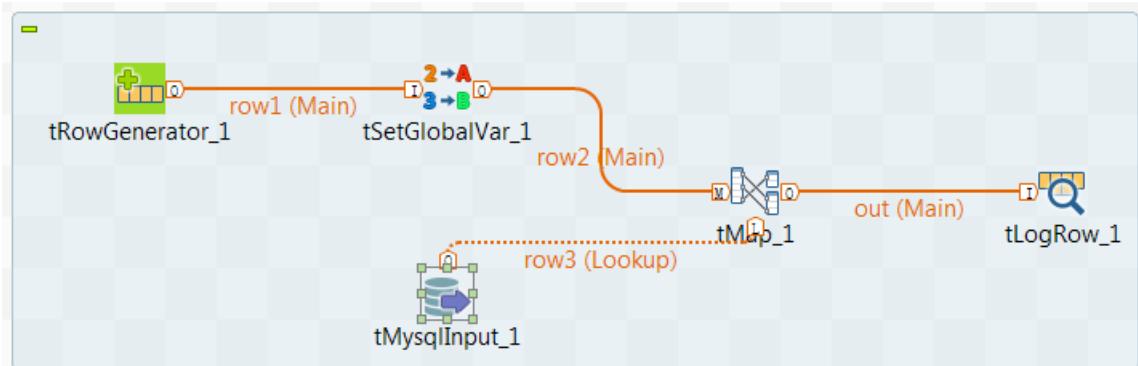
- Dans la boîte de dialogue **[Options]** double-cliquez sur le mode de chargement souhaité, ou sélectionnez-le et cliquez sur **OK** pour activer l'option et fermer la boîte de dialogue.

Pour un exemple d'utilisation de ces options, consultez la documentation associée au composant **tMap**.

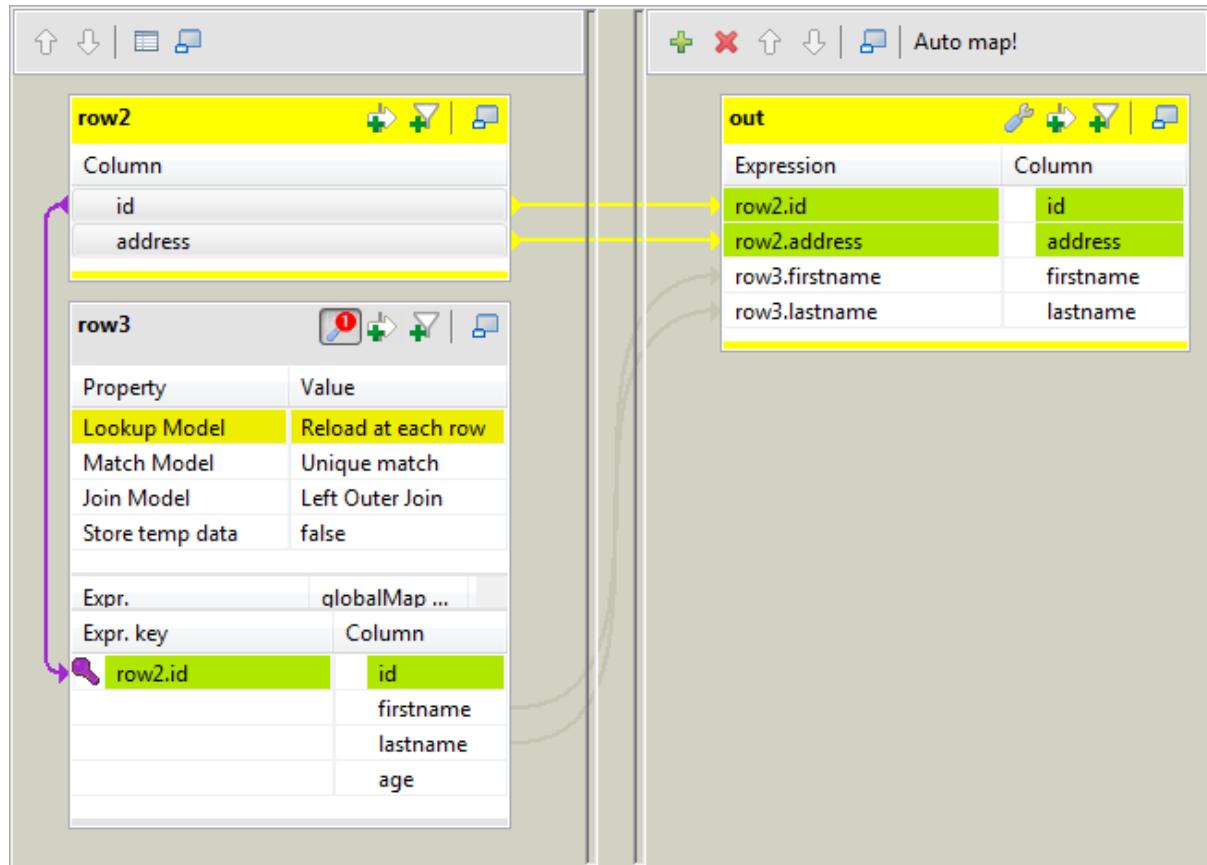
### 8.2.8.2. Recharger des données à chaque ligne

L'option **Reload at each row** est utilisée pour charger tous les enregistrements d'un flux Lookup pour chaque enregistrement du flux principal.

Lorsque le flux principal contient moins de lignes que le flux de référence (par exemple, avec un ratio de 1000 ou plus) et que l'entrée de référence est un composant de base de données, l'avantage de cette approche est qu'elle permet de gérer des données de références de plus en plus nombreuses, puisque vous pouvez exécuter des requêtes par rapport aux données du flux principal dans le composant de base de données, pour sélectionner uniquement les données de référence correspondant à chaque enregistrement du flux principal. Par exemple, ici, avec des données de référence d'une base de données MySQL.

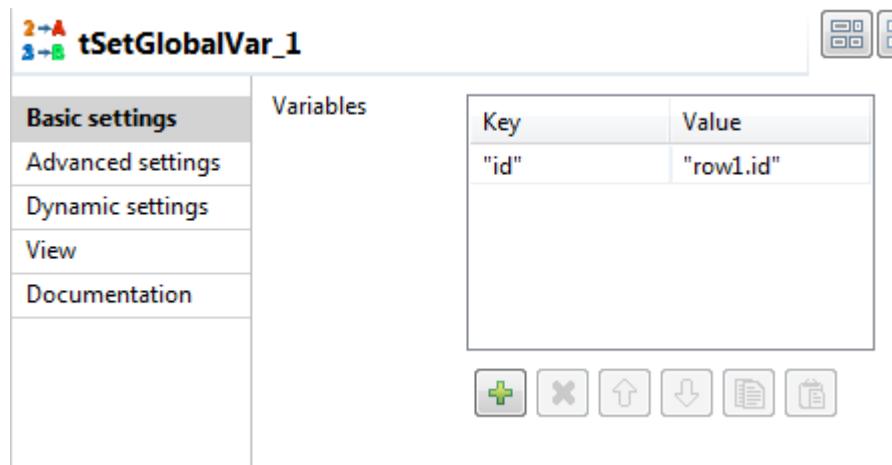


Les schémas du flux principal, du flux de référence et du flux de sortie se présentent comme suit :

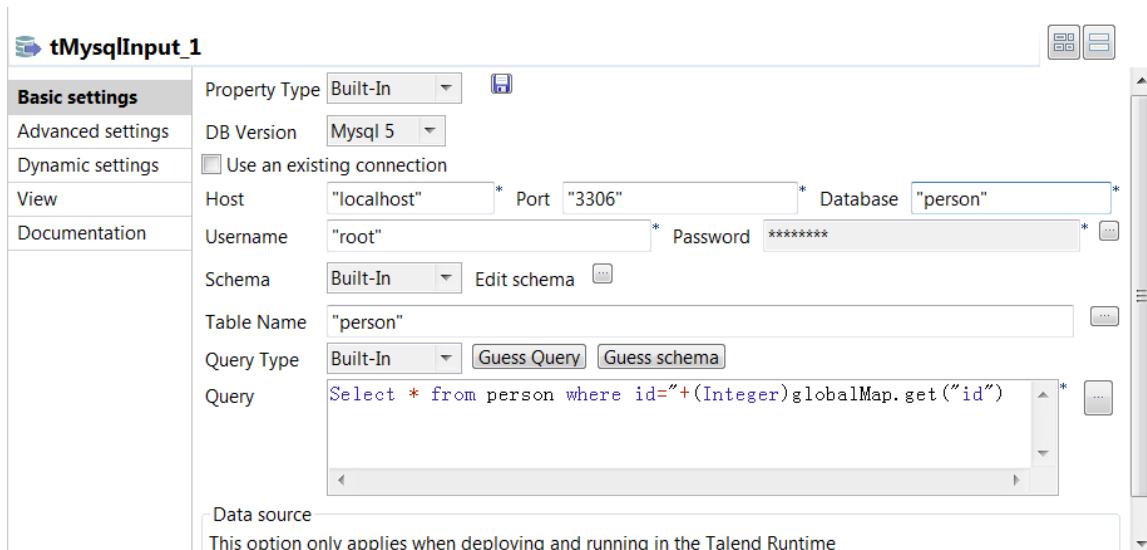


Dans la base de données MySQL, vous pouvez sélectionner uniquement les données correspondant aux valeurs de la colonne *id* du flux principal. Pour ce faire, procédez comme suit :

1. Double-cliquez sur le composant **tSetGlobalVar** pour ouvrir sa vue **Component**.



2. Cliquez sur le bouton [+] pour ajouter une ligne et, dans la colonne **Key**, saisissez *id*. Dans la colonne **Value**, saisissez *row1.id*.
3. Double-cliquez sur le tMysqlInput pour ouvrir sa vue Component.



4. Dans le champ **Query**, saisissez la requête pour sélectionner les données correspondant à la colonne *id* du flux principal. Dans cet exemple, la requête se présente comme suit :

```
Select * from person where id='+(Integer)globalMap.get("id")'
```

Pour plus d'informations sur les composants utilisés dans cet exemple, consultez la documentation associée.

## 8.3. 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 [Utiliser les jointures explicites](#)),
- construction automatisée d'arbre XML des côtés d'entrée et de sortie,
- jointures Inner Join et Left Outer Join (consultez [Utiliser la fonction Inner Join](#)),
- flux lookup entre tous types de sources de données, XML ou plates à l'aide de modèles comme **Load once** (consultez [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 [Utiliser le type Document pour créer l'arborescence XML](#).
- la gestion des données XML de sortie. Pour plus d'informations, consultez [Définir le mode de sortie](#).
- l'édition de schémas d'arborescence XML. Pour plus d'informations, consultez [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 [Résolution des problèmes de mémoire lors de l'utilisation du tMap](#).

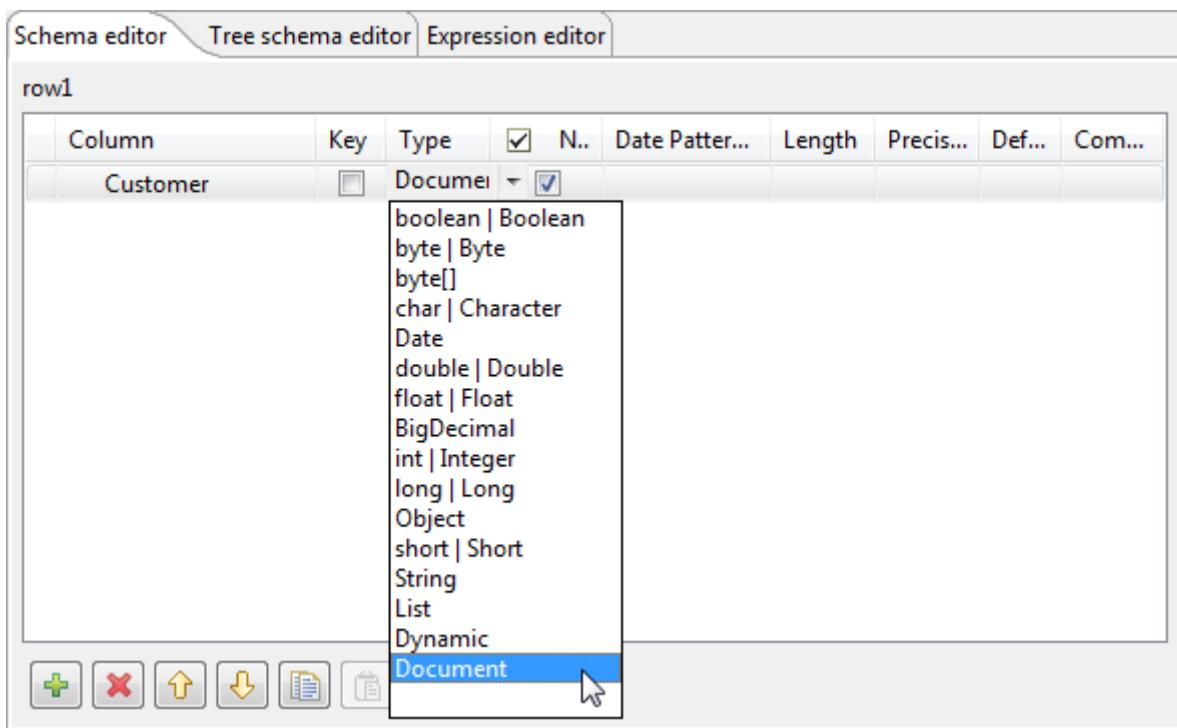
## 8.3.1. Utiliser le type Document pour créer l'arborescence XML

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.

### 8.3.1.1. 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 [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.



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 [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 refléter les détails de la structure. L'arborescence représente le nombre d'éléments minimum requis pour une arborescence XML valide, dans le **tXMLMap** :

- L'élément **root** (racine) : il est l'élément requis par une arborescence XML à traiter et, si nécessaire, il est la base d'une arborescence XML plus sophistiquée.
- L'élément **loop** (de boucle) : il détermine l'élément sur lequel s'effectue l'itération, afin de lire les données hiérarchiques dans une arborescence XML. Par défaut, l'élément racine est défini comme l'élément de boucle.



Cette capture d'écran 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 sources de type :
  - fichiers XML ou XSD (consultez [Importer une structure XML à partir de fichiers XML et XSD](#)).



Lorsque vous importez un fichier XSD, vous créez la structure XML que décrit le fichier XSD.

- Connexions vers des fichiers XML créées et centralisées dans les métadonnées du **Repository** dans le Studio (consultez [Importer une structure XML à partir du Repository](#)). Pour plus d'informations sur la création de connexions à des fichiers XML dans le Repository, consultez [Centraliser des métadonnées d'un fichier XML](#).



Si nécessaire, vous pouvez développer manuellement l'arborescence XML qui vous intéresse, à l'aide des options fournies dans le menu contextuel.

## 2. Définir l'élément de boucle pour l'arborescence XML que vous créez.

- 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 [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                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Create Sub-element et Create Attribute</b> | Ajoute des éléments ou des attributs à l'arborescence XML importée. Consultez également <a href="#">Ajouter un sous-élément ou un attribut à une structure XML</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Set a namespace</b>                        | Ajoute et gère des espaces de nommage donnés dans la structure XML importée. Consultez également <a href="#">Gérer un espace de nommage</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>Delete</b>                                 | Supprime un élément ou un attribut. Consultez également <a href="#">Supprimer un élément ou un attribut depuis la structure XML</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Rename</b>                                 | Renomme un élément ou un attribut.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>As loop element</b>                        | Définit ou réinitialise un élément de boucle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>As optional loop</b>                       | Cette option est disponible uniquement sur l'élément de boucle que vous avez défini.<br><br>Lorsque l'élément correspondant existe dans le fichier source, l'élément de boucle facultatif fonctionne de la même manière qu'un élément de boucle normal. Sinon, il définit automatiquement son élément père comme élément de boucle. En l'absence d'élément père dans le fichier source, il prend l'élément du plus haut niveau hiérarchique, avant l'élément racine. Cependant, dans des cas d'utilisation réelle, avec des différences entre l'arborescence XML et la structure du fichier source, il est recommandé d'adapter l'arborescence XML au fichier source, pour améliorer les performances. |
| <b>As group element</b>                       | Dans la structure XML de sortie, définit un élément comme <b>group element</b> . Consultez également <a href="#">Grouper les données de sortie</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>As aggregate element</b>                   | Dans la structure XML de sortie, Définit un élément <b>aggregate</b> . Consultez également <a href="#">Agréger les données de sortie</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Add Choice</b>                             | Configure l'élément <b>Choice</b> . Tous ses éléments fils développés seront contenus dans cette déclaration. Cet élément provient des concepts XSD. Il permet au <b>tXMLMap</b> d'exécuter la fonction de l'élément <b>Choice</b> XSD afin de lire ou d'écrire un flux Document.<br><br>Lorsque le <b>tXMLMap</b> traite un élément <b>Choice</b> , les éléments contenus dans sa déclaration ne sont pas écrits en sortie, à moins que leurs expressions de mapping soient définies de manière appropriée.<br><br>Le composant <b>tXMLMap</b> déclare automatiquement tout élément <b>Choice</b> défini dans le fichier XSD importé.                                                                 |
| <b>Set as Substitution</b>                    | Configure l'élément <b>Substitution</b> afin de spécifier l'élément substituable pour un élément <b>Head</b> donné, défini dans le XSD correspondant. L'élément de substitution permet au <b>tXMLMap</b> d'exécuter la fonction de l'élément <b>Substitution</b> XSD afin de lire ou d'écrire un flux Document.                                                                                                                                                                                                                                                                                                                                                                                        |

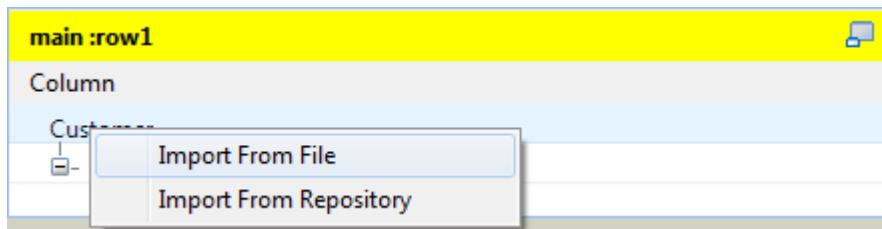
| Options | Opérations                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <p>Lorsque le <b>tXMLMap</b> traite un élément de substitution, les éléments contenus dans sa déclaration ne sont pas écrits en sortie, à moins que leurs expressions de mapping soient définies de manière appropriée.</p> <p> Le composant <b>tXMLMap</b> déclare automatiquement tout élément <b>Substitution</b> défini dans le fichier XSD importé.</p> |

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

### 8.3.1.2. Importer une structure XML à partir de fichiers XML et XSD

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

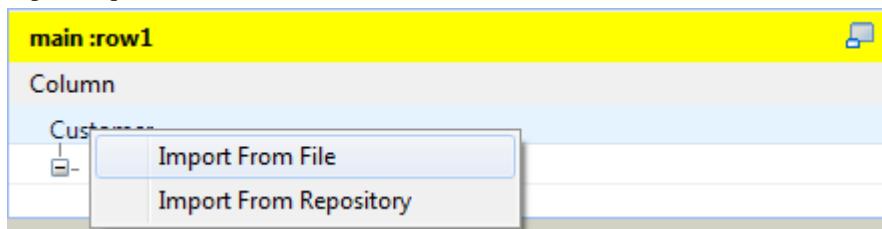
1. Dans la table d'entrée correspondante, cliquez-droit sur le nom de la colonne pour ouvrir le menu contextuel. Dans cet exemple, cliquez-droit sur la colonne *Customer*.



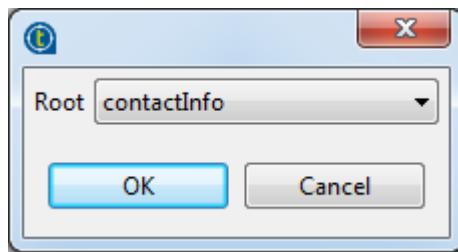
2. Dans le menu contextuel, sélectionnez l'option **Import From File**.
3. Dans la boîte de dialogue, parcourez votre système jusqu'au fichier XML que vous souhaitez utiliser pour fournir la structure XML, puis double-cliquez sur le fichier.

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**. L'arborescence XML décrite par le fichier XSD importée est établie.



La racine de l'arborescence XML est adaptable :

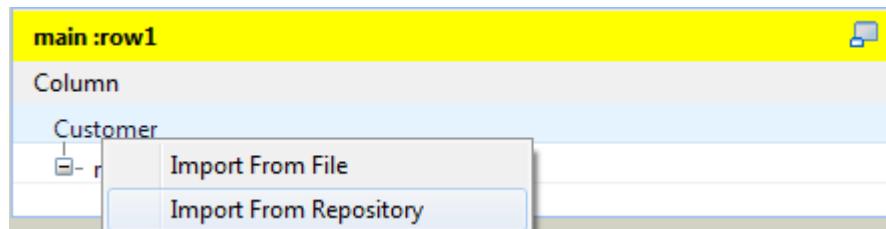
- Lorsque vous importez une structure XML d'entrée ou de sortie à partir d'un fichier XSD, vous pouvez choisir l'élément racine de votre arborescence XML.
- Une fois qu'une structure XML a été importée, la balise **root** est automatiquement renommée avec le nom de la source XML. Pour modifier le nom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

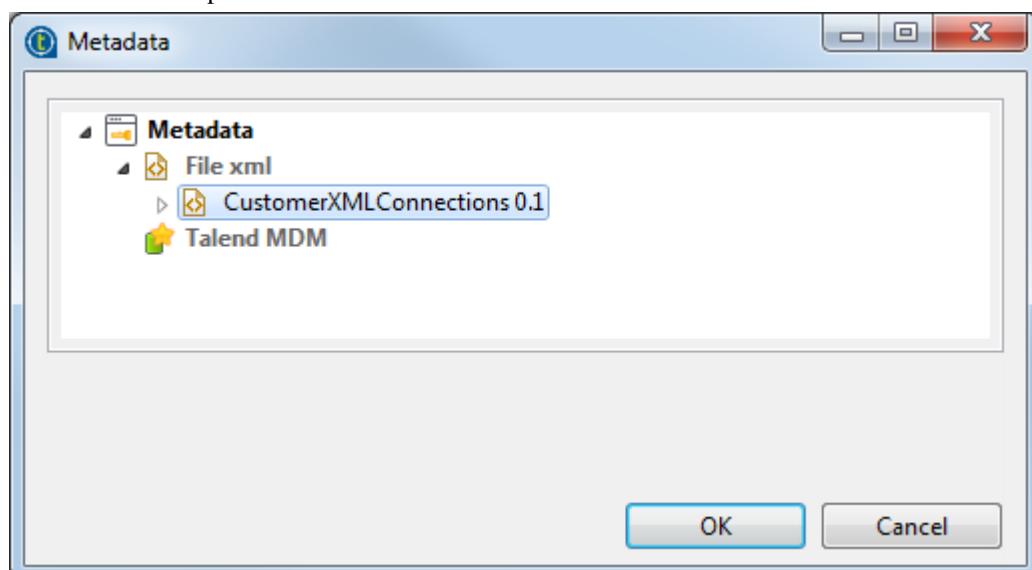
### 8.3.1.3. Importer une structure XML à partir du Repository

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 [Centraliser des métadonnées d'un fichier 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.

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

Vous devez définir un minimum un élément de 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.

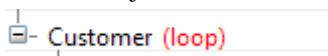
Pour définir ou réinitialiser un élément de boucle, procédez comme suit :

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



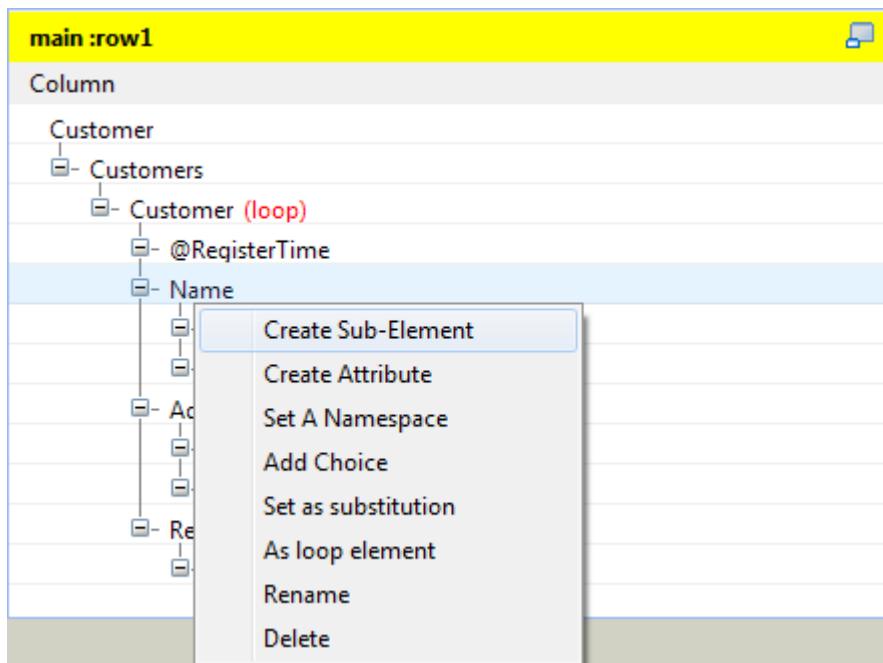
Si vous fermez le **Map Editor** sans avoir défini les éléments de boucle nécessaires, comme expliqué plus tôt dans ce scénario, l'élément racine est automatiquement défini comme élément de boucle.

### 8.3.1.5. Ajouter un sous-élément ou un attribut à une structure XML

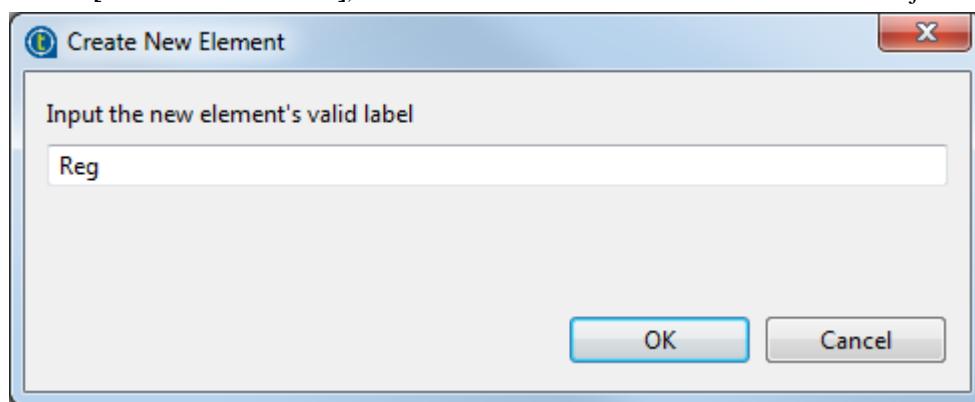
Dans la structure XML, vous pouvez ajouter manuellement un sous-élément ou un attribut à la racine ou à l'un des éléments.

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.

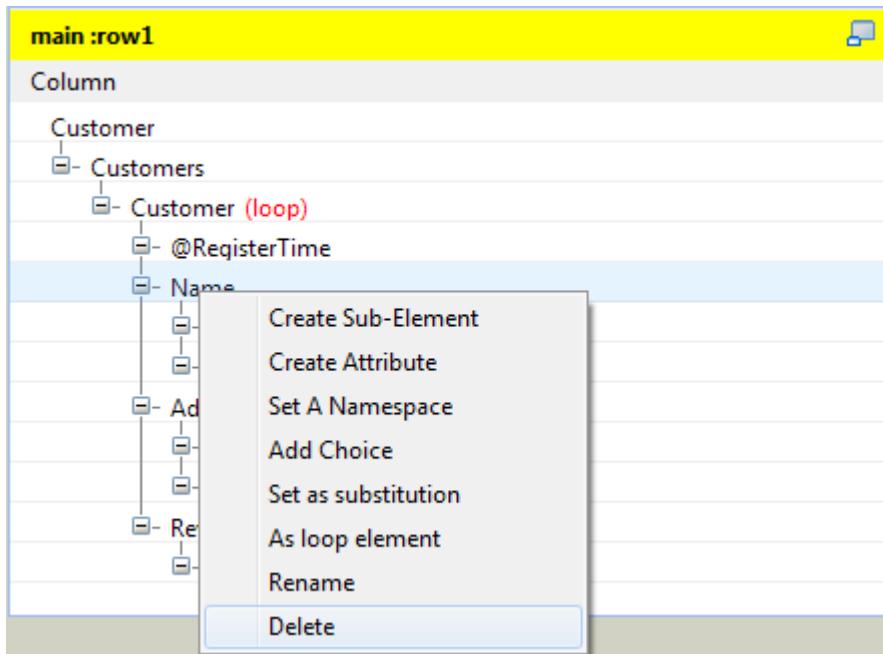


3. Cliquez **OK** pour valider la création du nouvel élément. Ce nouveau sous-élément ou attribut apparaît alors dans la structure XML.

### 8.3.1.6. 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.

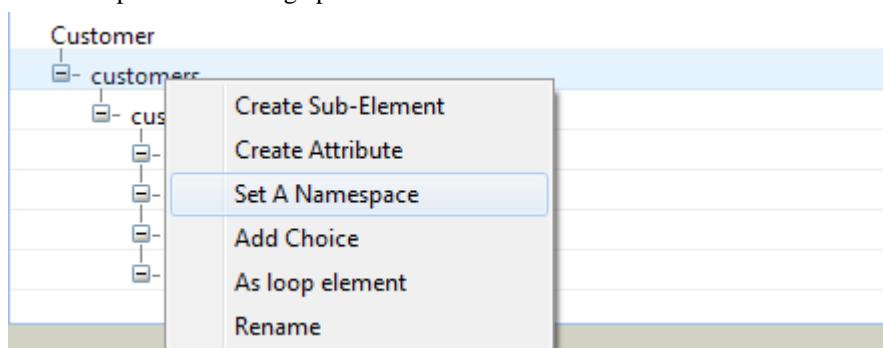
### 8.3.1.7. Gérer un espace de nommage

Vous pouvez définir et modifier un espace de nommage pour chacun des éléments des arborescences XML de ces flux d'entrée ou de sortie.

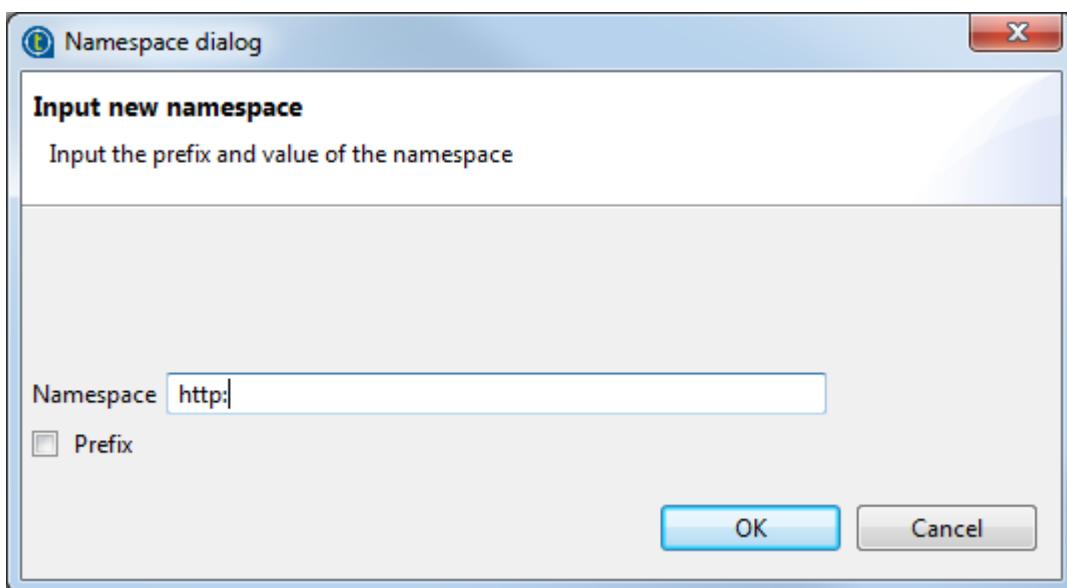
#### Définir un espace de nommage

Pour définir un espace de nommage :

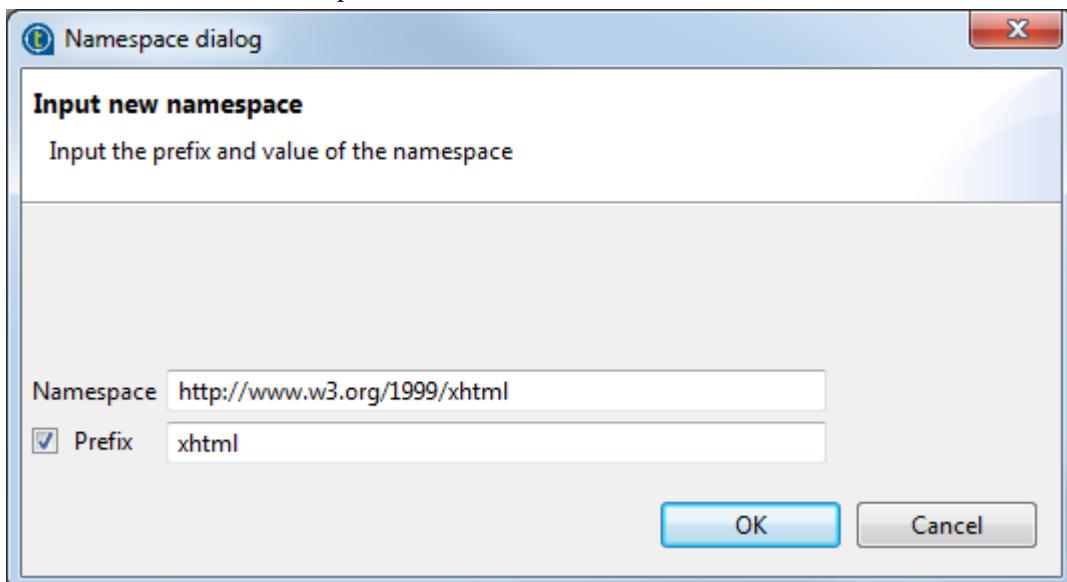
1. Dans la structure XML du flux d'entrée ou de sortie que vous souhaitez modifier, cliquez-droit sur l'élément auquel vous souhaitez associer un espace de nommage. Par exemple, dans l'arborescence XML de *Customer*, vous devez créer un espace de nommage pour le nœud racine.



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*.

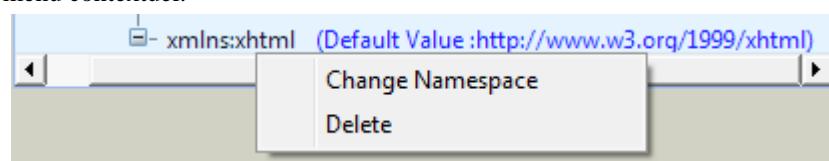


5. Cliquez sur **OK** pour valider l'espace de nommage.

## Modifier la valeur par défaut d'un espace de nommage

Pour modifier la valeur par défaut d'un espace de nommage, procédez comme suit :

- Dans l'arborescence XML contenant l'espace de nommage à modifier, cliquez-droit sur l'espace de nommage pour ouvrir le menu contextuel.



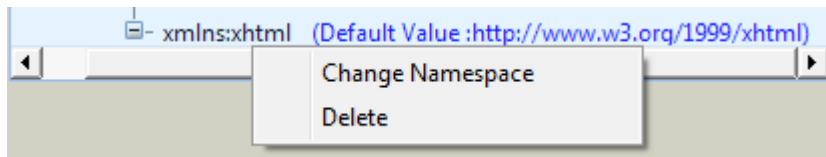
- Dans le menu, sélectionnez **Change Namespace** pour ouvrir l'assistant correspondant.

3. Saisissez la nouvelle valeur dans cet assistant.
4. Cliquez sur **OK** pour valider cette modification.

## Supprimer un espace de 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.

### 8.3.1.8. 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** à l'adresse <https://help.talend.com>.



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 :

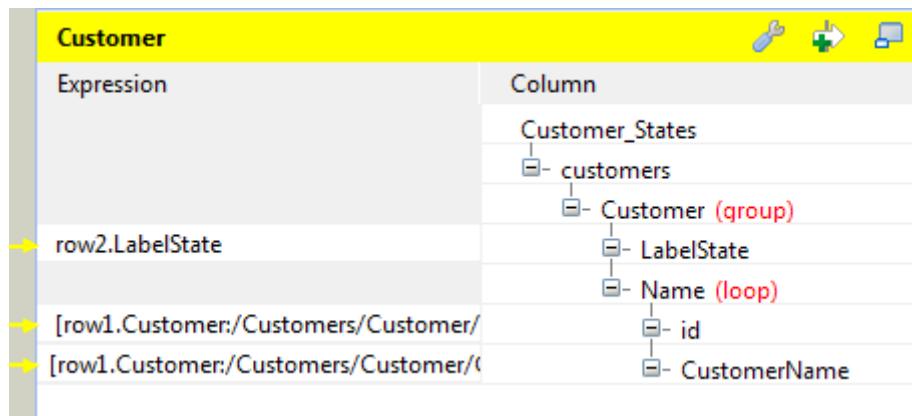
- L'élément "**group**" traite toujours les données en un seul flux.
- L'élément "**aggregate**" sépare ce flux en différents flux XML complets.

## 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**.

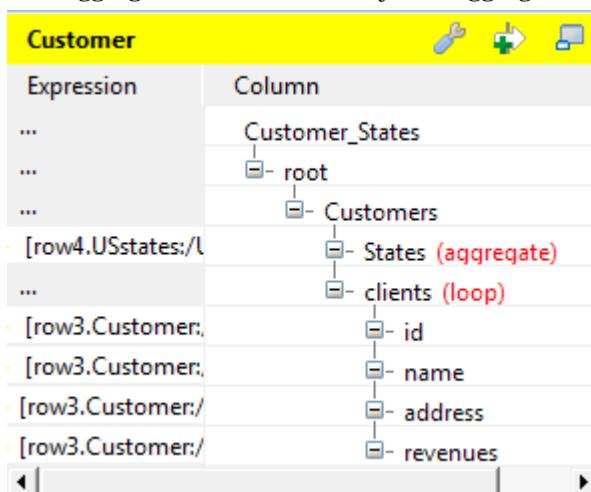
Le **group element** défini est révoqué.

### 8.3.1.9. 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**.



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 [Ecrire dans un document les éléments en sortie](#).

Pour un exemple d'utilisation de l'élément "aggregate" avec le **tXMLMap**, consultez la documentation du **tXMLMap** à l'adresse <https://help.talend.com>.

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.

### **8.3.2. 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.

#### **8.3.2.1. 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.

| Customer                    |                                                                                                                                                                                                                                                                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Property                    | Value                                                                                                                                                                                                                                                                                                           |
| Catch Output Reject         | false                                                                                                                                                                                                                                                                                                           |
| Catch Lookup Inner Join ... | false                                                                                                                                                                                                                                                                                                           |
| All in one                  | true                                                                                                                                                                                                                                                                                                            |
| Create empty element        | true                                                                                                                                                                                                                                                                                                            |
| Expression                  | <p>Col</p> <p>Customer_States</p> <ul style="list-style-type: none"> <li>  customers</li> <li>    customer (loop)           <ul style="list-style-type: none"> <li>      @id</li> <li>      CustomerName</li> <li>      CustomerAddress</li> <li>      idState</li> <li>      LabelState</li> </ul> </li> </ul> |
| ...                         |                                                                                                                                                                                                                                                                                                                 |
| ...                         |                                                                                                                                                                                                                                                                                                                 |
| ...                         |                                                                                                                                                                                                                                                                                                                 |
| [row1.Customer:/Cu          |                                                                                                                                                                                                                                                                                                                 |
| [row1.Customer:/Cu          |                                                                                                                                                                                                                                                                                                                 |
| [row1.Customer:/Cu          |                                                                                                                                                                                                                                                                                                                 |
| [row1.Customer:/Cu          |                                                                                                                                                                                                                                                                                                                 |
| [row2.USstates:/USsta       |                                                                                                                                                                                                                                                                                                                 |

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 [Agréger les données de sortie](#).

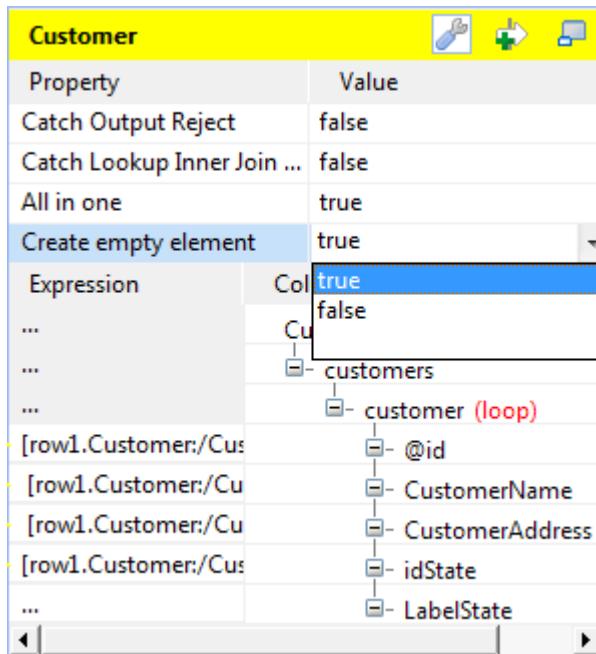
### 8.3.2.2. Gérer les éléments vides dans le Map editor

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 nœud XML dans le flux de données XML de sortie.

Vous pouvez garder ces éléments vides dans l'arborescence XML selon vos besoins, même si vous ne souhaitez pas les écrire en sortie.

Le composant **tXMLMap** vous permet de définir le booléen pour la création de l'élément vide. pour ce faire, dans la partie droite du **Map editor**, effectuez les opérations suivantes :

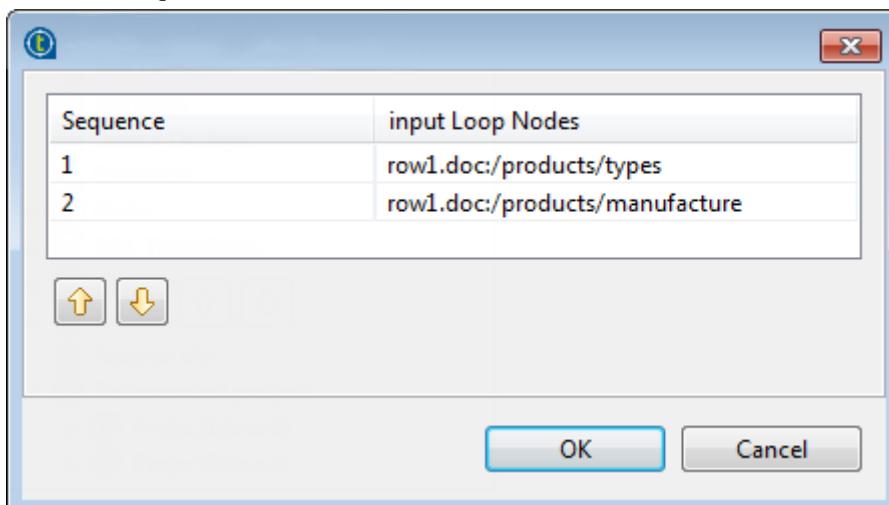
1. Cliquez sur la clé anglaise afin d'ouvrir le panneau de paramétrage du mapping.



2. Dans le panneau, cliquez sur le champ **Create empty element** et, dans la liste déroulante, sélectionnez **true** ou **false** afin de choisir d'écrire ou non l'élément vide.
  - Si vous sélectionnez **true**, l'élément vide est créé dans le flux XML de sortie et écrit, par exemple, comme suit <customer><LabelState/></customer>.
  - Si vous sélectionnez **false**, l'élément vide n'est pas écrit.

### 8.3.2.3. Définir la séquence des différents boucles en entrée

Si un élément de boucle, ou si le flux de données plat reçoit des mappings d'un ou plusieurs élément(s) de boucle du flux d'entrée, vous devez définir la séquence des boucles en entrée. Le premier élément de boucle de cette séquence sera la boucle primaire. Le processus de transformation relatif à cette séquence effectuera d'abord une boucle sur cet élément, afin que les données écrites en sortie soient triées selon la valeur de cet élément.



Par exemple, dans cette capture d'écran, l'élément *types* est la boucle primaire et les données écrites en sortie seront triées selon la valeur de cet élément.

```

<types>
 <type>DELL123</type>
 <manufacture_id>manu_1</manufacture_id>
</types>
<types>
 <type>DELL123</type>
 <manufacture_id>manu_2</manufacture_id>
</types>
<types>
 <type>DELL456</type>
 <manufacture_id>manu_1</manufacture_id>
</types>
<types>
 <type>DELL456</type>
 <manufacture_id>manu_2</manufacture_id>
</types>
<types>
 <type>HP123</type>
 <manufacture_id>manu_1</manufacture_id>
</types>
<types>
 <type>HP123</type>
 <manufacture_id>manu_2</manufacture_id>
</types>
<types>
 <type>HP456</type>
 <manufacture_id>manu_1</manufacture_id>
</types>
<types>
 <type>HP456</type>
 <manufacture_id>manu_2</manufacture_id>
</types>
</manufactures>

```

Dans le cas d'une réception de plusieurs éléments de boucle en entrée, un bouton [...] apparaît à côté de l'élément recevant la boucle, ou dans le cas de données plates, apparaît en haut de la table représentant le flux de données plat. Pour définir la séquence des boucles, procédez comme suit :

1. Cliquez sur le bouton [...] pour ouvrir la fenêtre de configuration de séquence comme dans la capture d'écran présentée précédemment dans cette section.
2. Utilisez les boutons de montée ou de descente afin de définir l'ordre de votre séquence.

### 8.3.3. 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.

The screenshot shows the 'Tree schema editor' interface. On the left, under 'row1', there is a table with columns: XPath, Key, Type, Nullable, and Pattern. The 'XPath' column contains several entries like 'row1.Customer:/Customers'. On the right, under 'Customer', there is another table with the same columns, and its 'XPath' column also contains entries related to customer data.

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
<b>XPath</b>	Le chemin absolu pointant sur chaque élément ou attribut de l'arborescence XML et éditez le nom de l'élément ou attribut correspondant.
<b>Key</b>	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.
<b>Type</b>	Type de données : String, Integer, Document, etc.  💡 Cette colonne est obligatoire.
<b>Nullable</b>	Indique si les valeurs de champs nulles sont autorisées.
<b>Pattern</b>	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.



## Chapitre 9. Mapping de flux Big Data

Lors du développement de processus ETL pour Big Data, il est courant de mapper des données d'une ou plusieurs source(s) vers des données stockées dans le système cible.

**Talend** fournit des composants de mapping optimisés pour l'environnement Hadoop, afin de mapper visuellement les flux d'entrée de données et de sortie.

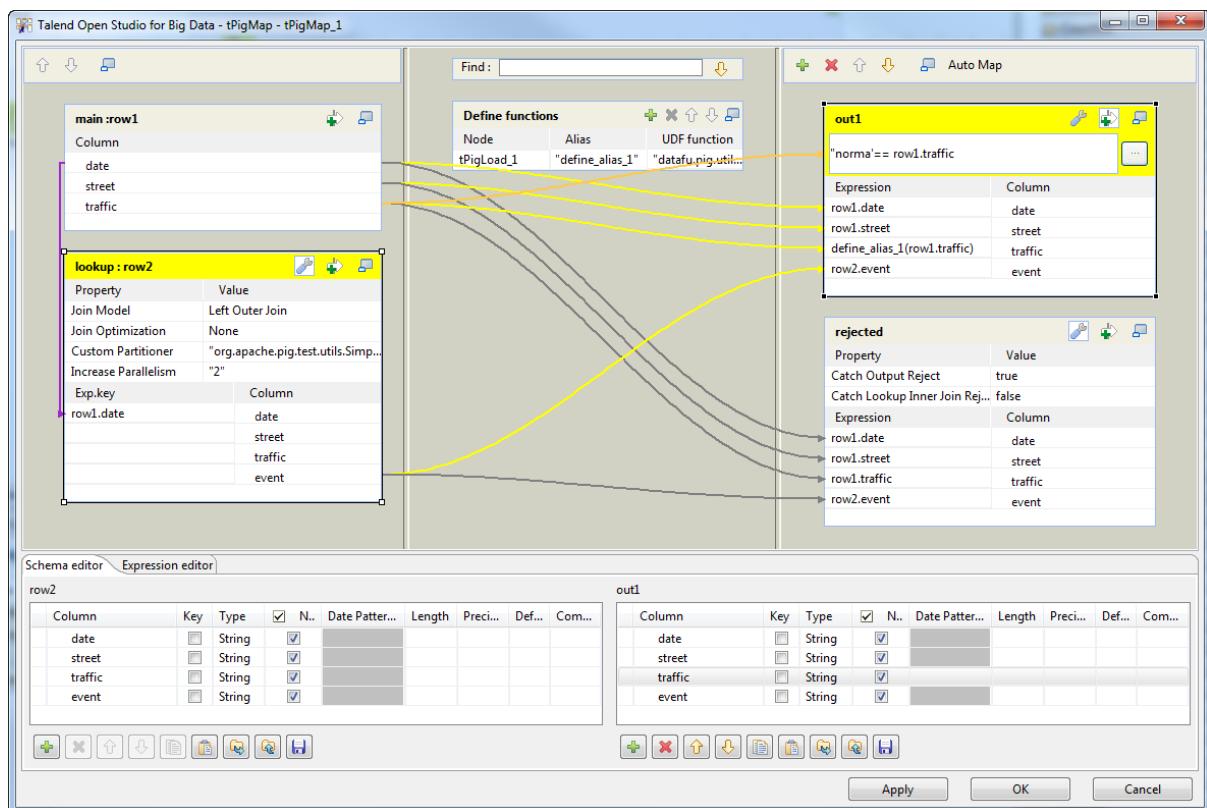
## 9.1. Interface du tPigMap

Pig est une plateforme utilisant un langage de script permettant de définir des flux de données. Elle permet de programmer des opérations étape par étape pour transformer des données à l'aide de Pig Latin, nom du langage utilisé par Pig.

Le **tPigMap** est un composant avancé mappant les flux de données d'entrée et de sortie gérés par un processus Pig (un ensemble de composants Pig). Il requiert un **tPigLoad** pour lire des données du système source et un **tPigStoreResult** pour écrire de données dans une source cible. A partir de ce processus basique composé d'un **tPigLoad**, d'un **tPigMap** et d'un **tPigStoreResult**, vous pouvez développer visuellement un processus Pig très complexe en utilisant d'autres composants Pig autour du **tPigMap**. Comme ces composants génèrent du code Pig, le Job développé est alors optimisé pour un environnement Hadoop.

Vous devez utiliser un éditeur de mapping pour configurer le **tPigMap**. Ce **Map Editor** est un outil "tout en un" vous permettant de définir tous les paramètres nécessaires au mapping, aux transformations et au routage vos flux de données via une interface graphique pratique.

Vous pouvez minimiser et restaurer le **Map Editor** et toutes les tables du **Map Editor** à l'aide des icônes de la fenêtre.



Le **Map Editor** se compose de différentes zones :

- La zone d'entrée (**Input**), à gauche de l'éditeur. Elle offre une représentation graphique de tous les flux de données (**Main** et **Lookup**). Les données sont regroupées dans plusieurs colonnes des schémas Input. Notez que le nom de la table reflète le nom du lien Main ou Lookup dans l'espace de modélisation graphique du Job.
- La zone de sortie (**Output**), à droite de l'éditeur. Elle permet de mapper des données et des champs des tables d'entrée vers les lignes de sortie correspondantes.
- Le panneau de recherche (**Search panel**), en haut, au centre de l'éditeur. Il vous permet de chercher dans l'éditeur des colonnes ou des expressions contenant le texte saisi dans le champ **Find**.
- Le panneau **Define functions**, situé sous le panneau de recherche, vous permet de définir les fonctions personnalisées Pig (User-Defined Functions, UDF) à charger par le(s) composant(s) d'entrée et à appliquer aux

données de sortie spécifiques. Pour plus d'informations, consultez [Définir une fonction personnalisée Pig à l'aide du panneau UDF](#).

- Les deux panneaux du bas sont les descriptions des schémas d'entrée et de sortie. L'onglet **Schema editor** offre une vue du schéma, comprenant les colonnes des tables d'entrée et de sortie dans leur panneau respectif.
- L'onglet **Expression editor** est l'outil d'édition pour toutes les clés d'expression des données d'entrée/de sortie ou les conditions de filtre.

Le nom des tables d'entrée/de sortie dans le **Map Editor** reflète le nom des flux entrant(s) et sortant(s) (liens **Row**).

Cet éditeur **Map Editor** est un éditeur **Talend** de mapping typique, comme celui du **tMap**. Afin de comprendre complètement le fonctionnement d'un composant de mapping basique, il est recommandé de lire le chapitre de référence décrivant comment le studio **Talend** mappe les flux de données, dans [Mapping de flux de données](#).

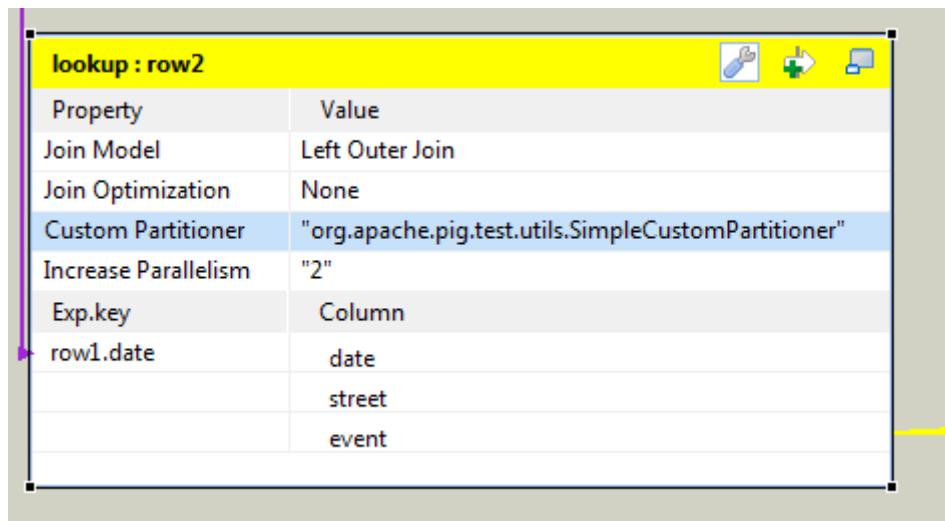
## 9.2. Présentation du fonctionnement du tPigMap

Vous pouvez mapper des flux de données simplement en glissant-déposant des colonnes de la zone d'entrée à la zone de sortie d'un **tPigMap**. Fréquemment, vous pouvez effectuer des opérations plus complexes, comme éditer un filtre, configurer une jointure ou utiliser une fonction personnalisée pour Pig. Dans cette situation le, **tPigMap** fournit un ensemble varié d'options à configurer et génère le code Pig correspondant afin de répondre à vos besoins.

Les sections suivantes présentent les différentes options.

### 9.2.1. Configurer les opérations de jointure

Du côté de l'entrée, vous pouvez afficher le panneau utilisé pour les paramètres de jointure en cliquant sur l'icône de la table.

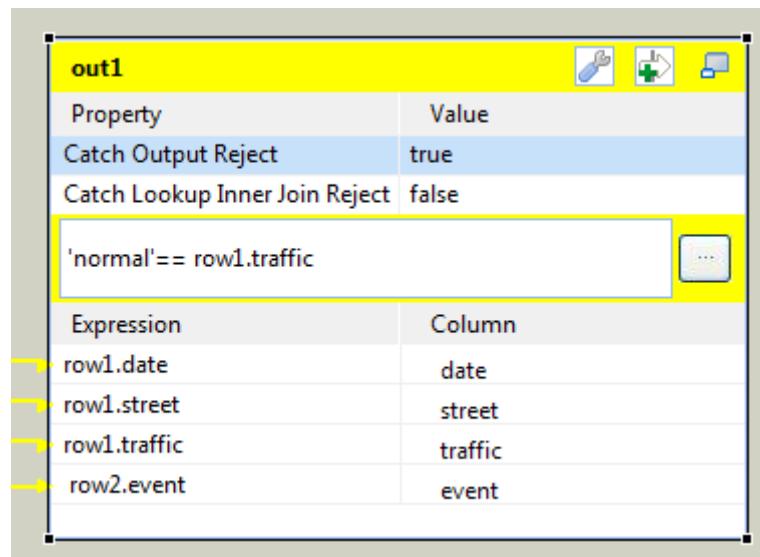


Propriétés du flux Lookup	Valeur
Join Model	Inner Join ;

Propriétés du flux Lookup	Valeur
	<p><b>Left Outer Join</b> ;</p> <p><b>Right Outer Join</b> ;</p> <p><b>Full Outer Join</b>.</p> <p>L'option de jointure par défaut est <b>Left Outer Join</b> lorsque vous n'activez pas les paramètres des options en affichant le panneau. Ces options effectuent une jointure de deux flux ou plus, selon la valeur des champs communs.</p> <p>Lorsque plusieurs tables de référence (lookup) nécessitent une jointure, le flux d'entrée principal active la jointure à partir du premier flux Lookup et utilise le résultat pour effectuer une jointure sur la deuxième, etc, de la même manière, jusqu'à ce que le dernier flux Lookup ait une jointure.</p>
<i>Join Optimization</i>	<p><b>None</b> ;</p> <p><b>Replicated</b> ;</p> <p><b>Skewed</b> ;</p> <p><b>Merge</b>.</p> <p>L'option de jointure par défaut est <b>None</b> lorsque vous n'activez pas les paramètres des options en affichant le panneau. Ces options sont utilisées pour effectuer des opérations de jointure plus efficaces. Par exemple, si vous utilisez le parallélisme des différentes tâches Reduce, la jointure <b>Skewed</b> peut être utilisée pour rééquilibrer les données à traiter, si elles sont réparties de manière asymétrique.</p> <p>Chacune de ces options est sujette aux contraintes présentées dans la documentation Apache concernant Pig Latin.</p>
<i>Custom Partitioner</i>	<p>Saisissez le nom du Partitioner Hadoop à utiliser pour contrôler le partitionnement des clés des map-sorties intermédiaires. Par exemple, saisissez, entre guillemets doubles,</p> <pre>org.apache.pig.test.utils.SimpleCustomPartitioner</pre> <p>pour utiliser le Partitioner <i>SimpleCustomPartitioner</i>. Le fichier Jar de ce Partitioner doit avoir été enregistré dans la table <b>Register jar</b> dans la vue <b>Advanced settings</b> du composant <b>tPigLoad</b> lié au <b>tPigMap</b> à utiliser.</p> <p>Pour plus d'informations concernant le code de ce <i>SimpleCustomPartitioner</i>, consultez la documentation Apache concernant Pig Latin.</p>
<i>Increase Parallelism</i>	Saisissez le nombre de tâches Reduce pour les Jobs Hadoop MapReduce générés par Pig. Pour plus d'informations concernant les fonctionnalités de parallélisation, consultez la documentation Apache concernant Pig Latin.

## 9.2.2. Capturer les enregistrements rejetés

Du côté de la sortie, les options suivantes deviennent disponibles lorsque vous affichez le panneau utilisé pour configurer les options de sortie, en cliquant sur le bouton  de la table.



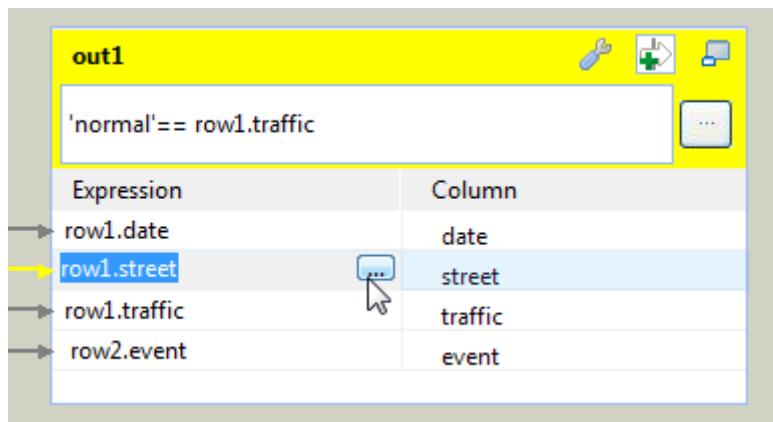
Propriétés du flux de sortie	Valeur
<i>Catch Output Reject</i>	True ; False. Cette option, une fois activée, vous permet de capturer les enregistrements rejetés par un filtre défini dans la zone appropriée.
<i>Catch Lookup Inner Join Reject</i>	True ; False. Cette option, une fois activée, vous permet de capturer les enregistrements rejetés, par la jointure Inner Join effectuée sur les flux d'entrée.

## 9.2.3. Editer les expressions

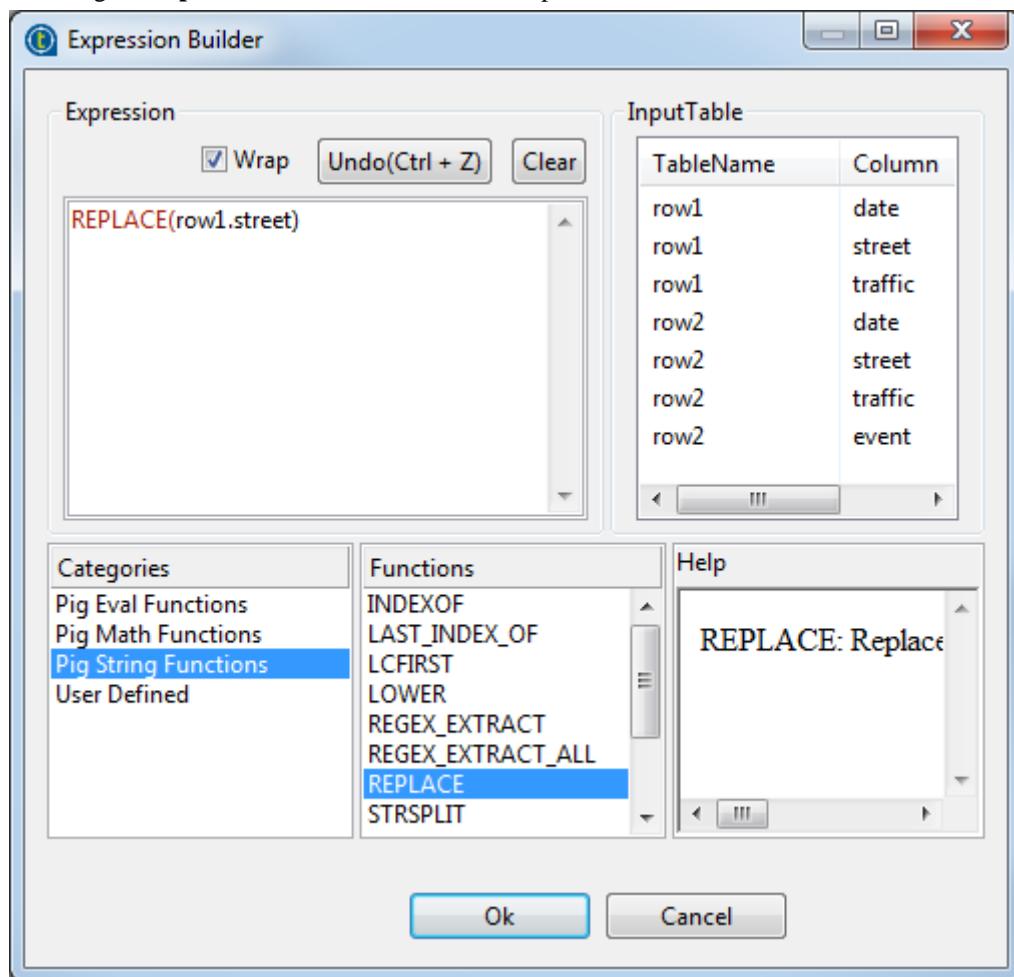
Des deux côtés, vous pouvez éditer toutes les clés d'expression des données d'entrée/sortie ou des conditions de filtre en utilisant Pig Latin. Cela vous permet de filtrer ou diviser une relation sur la base de ces conditions. Pour plus d'informations concernant Pig Latin, consultez la documentation Apache concernant Pig, notamment *Pig Latin Basics* et *Pig Latin Reference Manual*.

Vous pouvez écrire les expressions nécessaires à la transformation des données, directement dans la vue **Expression editor** située dans la partie inférieure de l'éditeur. Sinon, ouvrez la boîte de dialogue [**Expression Builder**] dans laquelle vous pouvez écrire les expressions de transformation des données.

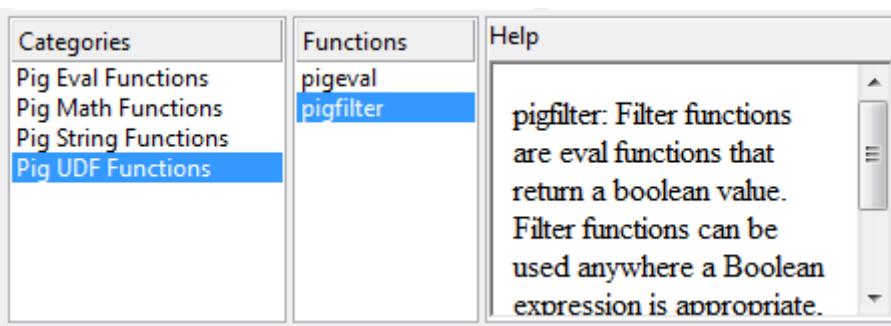
Pour ouvrir la boîte de dialogue [**Expression Builder**], cliquez sur le bouton à côté de l'expression à ouvrir dans les panneaux représentant le(s) flux Lookup ou le(s) flux de sortie du **Map Editor**.



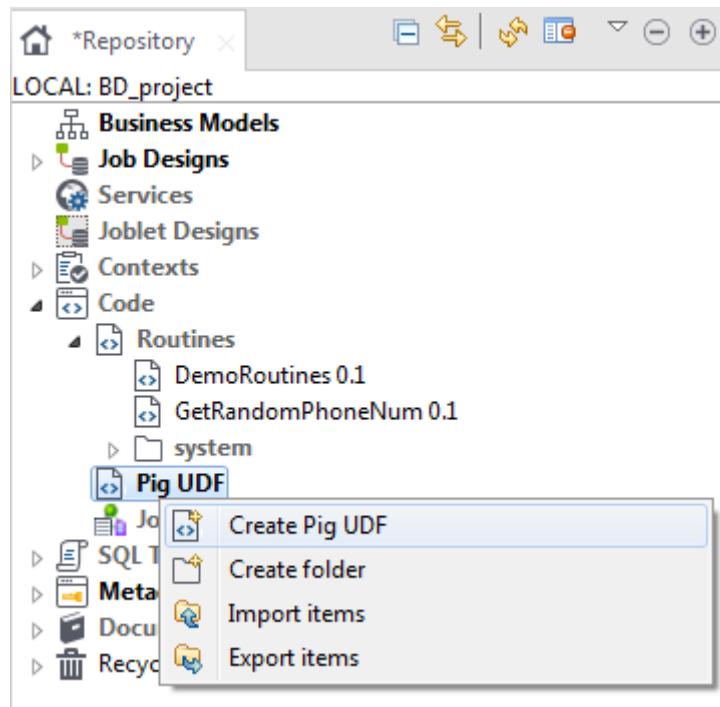
La boîte de dialogue [Expression Builder] s'ouvre sur l'expression sélectionnée.



Si vous avez créé une fonction Pig utilisateur (Pig UDF) dans le studio, une option **Pig UDF Functions** apparaît automatiquement dans la liste **Categories**. Vous pouvez la sélectionner pour éditer l'expression de mapping à utiliser.



Vous devez utiliser l'élément **Pig UDF** sous le nœud **Code** de la vue **Repository** pour créer une fonction Pig UDF. Même si vous devez savoir écrire une fonction Pig à l'aide de Pig Latin, une fonction Pig UDF est créée de la même manière qu'une routine **Talend**.

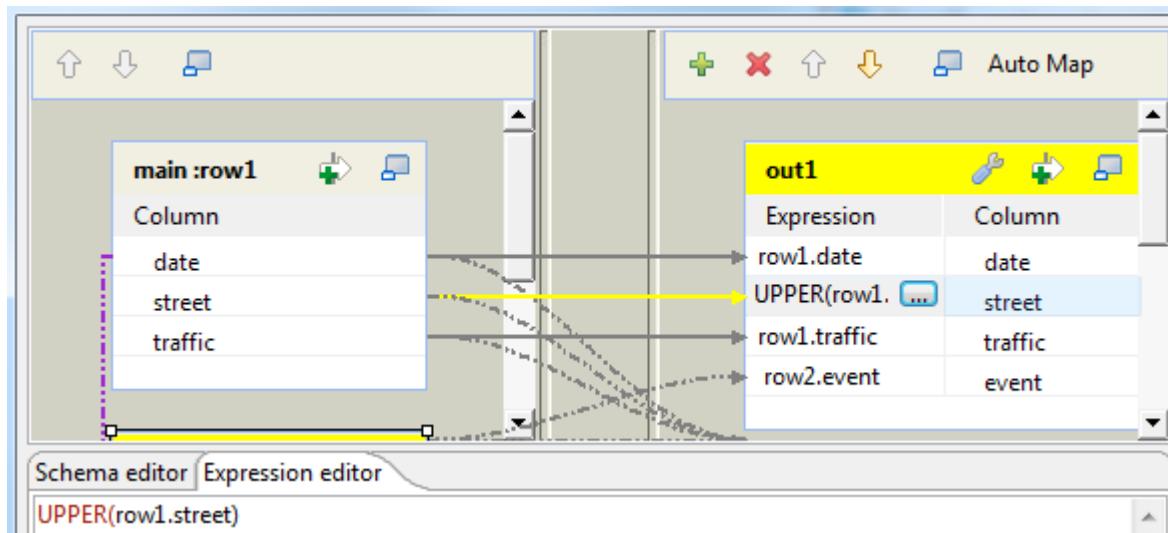


Notez que, suivant la licence que vous utilisez, votre **Repository** peut ne pas avoir la même apparence que dans la capture d'écran ci-dessus.

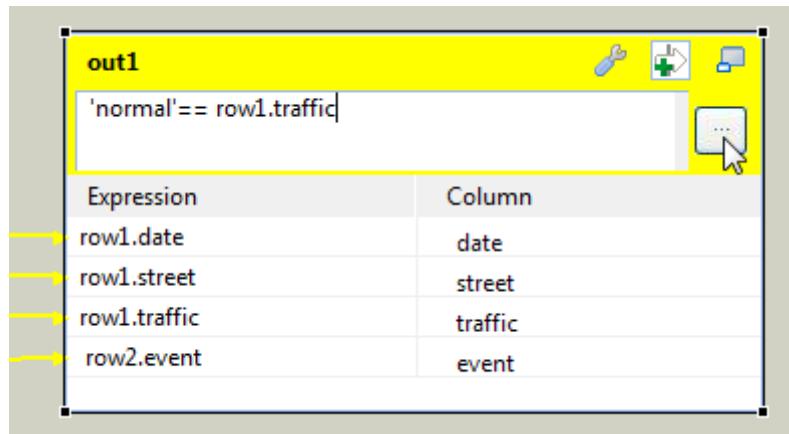
Pour plus d'informations concernant les routines, consultez [Gestion des routines](#).

Pour ouvrir la vue **Expression editor**,

1. Dans la partie inférieure de l'éditeur, cliquez sur l'onglet **Expression editor** pour ouvrir la vue correspondante.
2. Cliquez sur la colonne pour laquelle vous souhaitez paramétrier les expressions et modifiez ces expressions dans la vue **Expression editor**.



Si vous devez paramétriser des conditions de filtre pour un flux d'entrée ou de sortie, cliquez sur le bouton puis éditez les expressions dans la zone affichée ou en utilisant la vue **Expression editor** ou dans la boîte de dialogue **[Expression Builder]**.

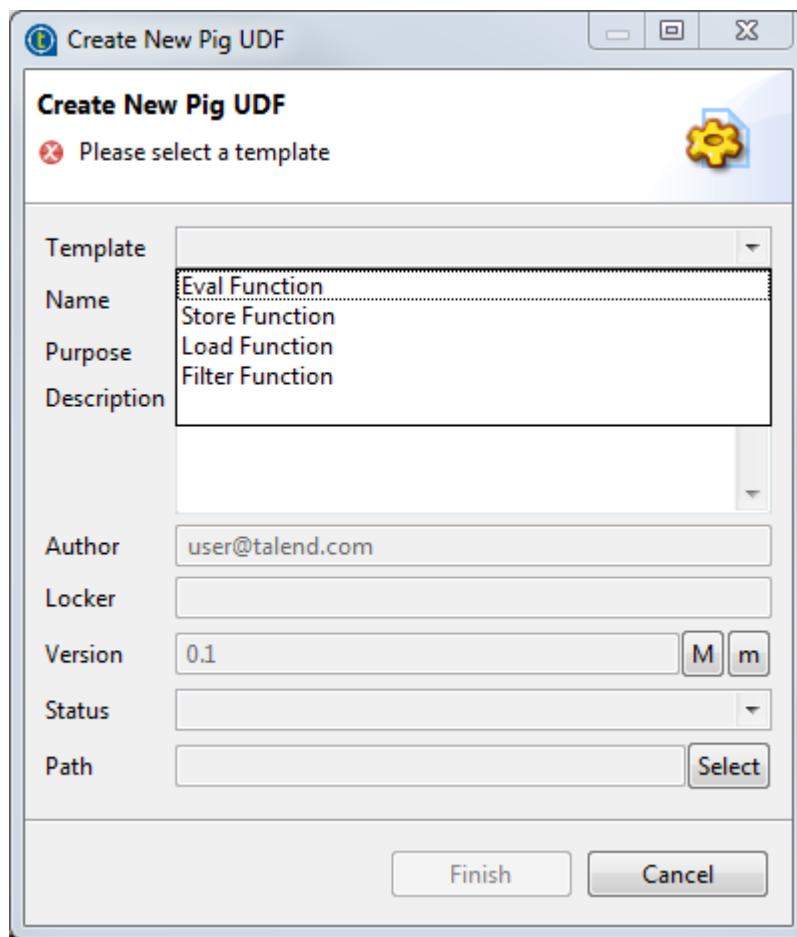


## 9.2.4. Configurer une fonction Pig utilisateur

Comme expliqué dans la section précédente, vous pouvez créer une fonction Pig utilisateur (Pig UDF). Cette dernière apparaît ensuite automatiquement dans la liste **Categories** de la vue **Expression Builder**.

1. Cliquez-droit sur le sous-nœud **Pig UDF**, situé sous le nœud **Code** de la vue **Repository**. Sélectionnez ensuite **Create Pig UDF** dans le menu contextuel.

L'assistant **[Create New Pig UDF]** s'ouvre.



2. Dans la liste **Template**, sélectionnez le type de fonction Pig UDF que vous souhaitez créer. Le studio se base sur votre choix afin de vous fournir un modèle correspondant pour vous aider à développer un Pig UDF répondant à vos besoins.
3. Complétez les autres champs de l'assistant.
4. Cliquez sur **Finish** pour valider vos changements. Le modèle de Pig UDF s'ouvre dans l'espace de modélisation graphique.
5. Saisissez votre code dans le modèle.

Une fois votre code saisi, le Pig UDF apparaît automatiquement dans la liste **Categories** de la vue **Expression Builder** du **tPigMap** et est prêt à être utilisé.

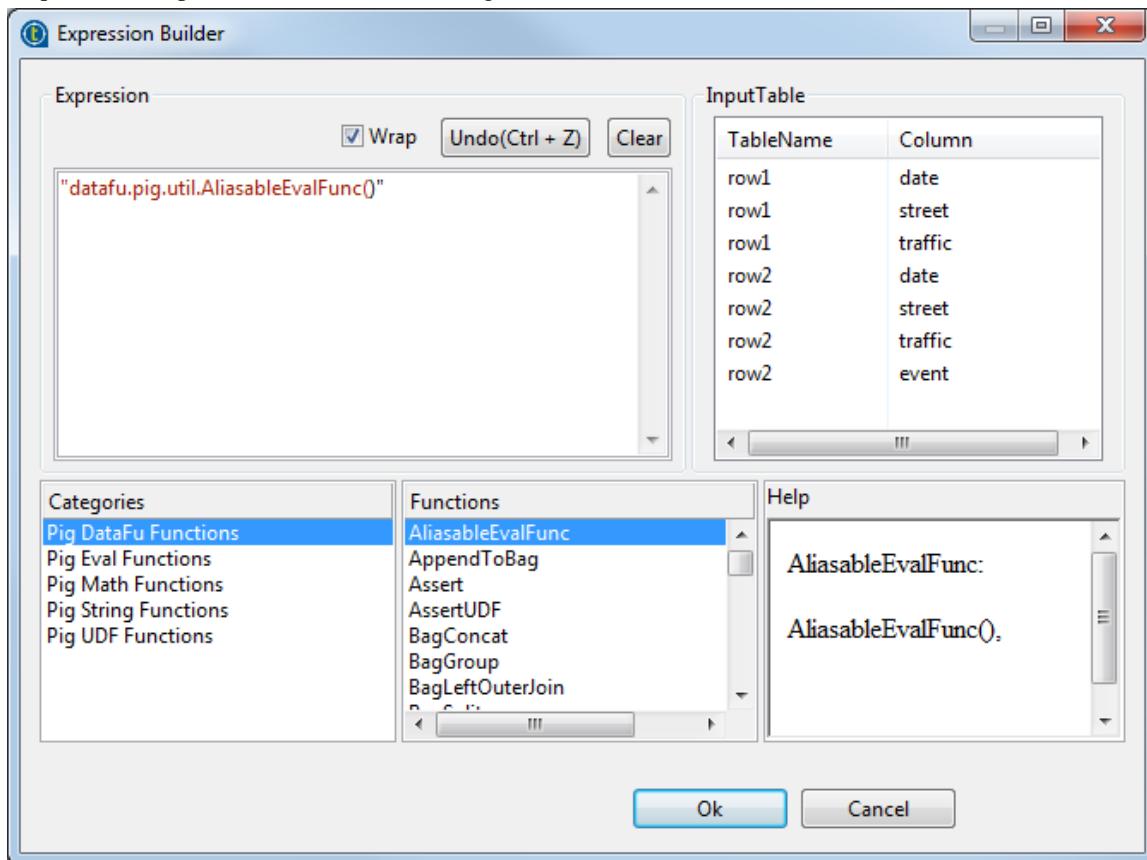
## 9.2.5. Définir une fonction personnalisée Pig à l'aide du panneau UDF

Grâce au panneau UDF du **tPigMap**, vous pouvez facilement définir des fonctions personnalisées Pig, notamment celles nécessitant un alias, comme les fonctions Apache DataFu Pig, à charger avec les données d'entrée.

1. Dans la table **Define functions** de l'éditeur **Map Editor**, cliquez sur le bouton pour ajouter une ligne. Les champs **Node** et **Alias** sont automatiquement renseignés avec les paramètres par défaut.

Define functions		
Node	Alias	UDF function
tPigLoad_1	"define_alias_1"	

2. Si nécessaire, cliquez dans le champ **Node** et sélectionnez, dans la liste déroulante, le composant **tPigLoad** à utiliser pour charger la fonction personnalisée que vous définissez.
3. Si vous souhaitez que votre fonction personnalisée ait un alias différent de celui proposé, saisissez votre alias dans le champ **Alias**, entre guillemets doubles.
4. Cliquez dans le champ **UDF function** puis sur le bouton  pour ouvrir la boîte de dialogue [**Expression Builder**].
5. Sélectionnez une catégorie dans la liste **Categories**. La liste **Functions** affiche toutes les fonctions disponibles de la catégorie sélectionnée.
6. Dans la liste **Functions**, double-cliquez sur la fonction de votre choix à ajouter dans la zone **Expression** et cliquez sur **OK** pour fermer la boîte de dialogue.



La fonction sélectionnée s'affiche dans le champ **UDF function**.

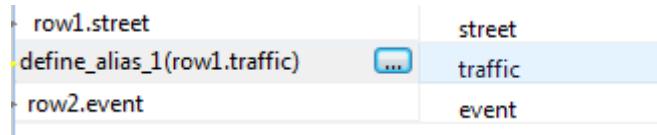
Define functions		
Node	Alias	UDF function
tPigLoad_1	"define_alias_1"	"datafu.pig.util.AliasableEvalFunc()

Une fois vos fonctions personnalisées définies dans la table, la table **Define functions** du composant **tPigLoad** spécifié est automatiquement synchronisée. De la même manière, une fois que vous avez défini vos fonctions personnalisées dans un **tPigLoad** connecté, cette table est automatiquement synchronisée.

Après définition d'une fonction personnalisée dans le panneau, vous pouvez l'utiliser dans une expression en :

- glissant-déposant son alias dans l'expression cible et éditant l'expression selon vos besoins,
- ouvrant la boîte de dialogue [Expression Builder] comme expliqué dans [Editer les expressions](#), en sélectionnant **User Defined** dans la liste **Category**, en double-cliquant sur l'alias de la fonction personnalisée dans la liste **Functions** et en l'ajoutant comme expression, puis en modifiant l'expression selon vos besoins.

Cela fait, l'alias est affiché dans l'expression, à la place de la fonction.







## Chapitre 10. Gestion des métadonnées dans l'intégration de données

Dans le *Studio Talend*, les métadonnées permettent de définir, de décrire ou de documenter les données traitées par le logiciel.

Dans la perspective **Integration** du studio, le dossier **Metadata** stocke les informations réutilisables concernant les fichiers, les bases de données et/ou les systèmes dont vous avez besoin pour créer vos Jobs.

Différents assistants vous permettent de conserver ces informations et de les réutiliser ultérieurement pour définir les paramètres de connexion des composants d'entrée ou de sortie correspondants. Vous pouvez également stocker les schémas des données dans votre studio.

Les procédures des assistants diffèrent légèrement selon le type de connexion choisi.

Ce chapitre vous décrit les procédures de création et de gestion des différentes métadonnées du **Repository** pouvant être utilisées dans tous vos Jobs. Pour plus d'informations sur l'utilisation des métadonnées du **Repository**, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Avant de commencer un processus de gestion de données, vous devez vous être familiarisé avec l'interface graphique de votre studio. Pour plus d'informations, consultez l'annexe traitant les éléments de l'interface graphique.

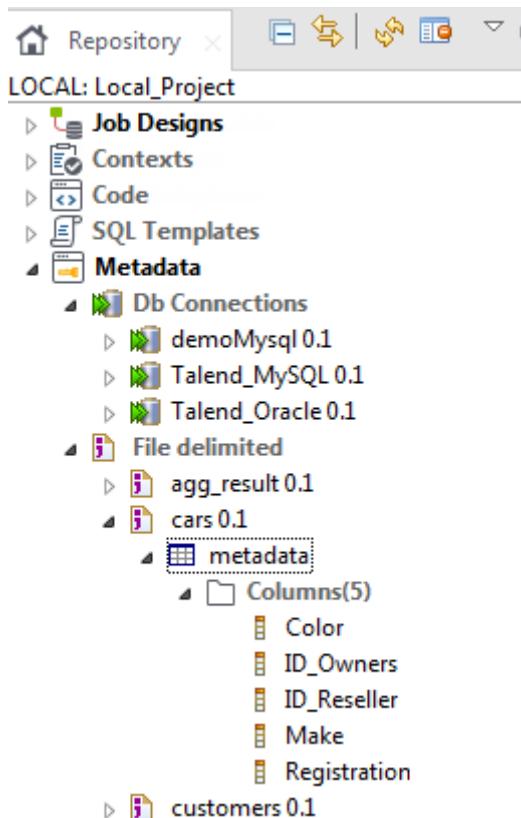
## 10.1. Objectifs

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 création 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" de façon centralisée dans le *Studio Talend*.

Les procédures des différents assistants diffèrent en fonction du type de connexion choisi.

Cliquez sur **Metadata** dans le **Repository** pour développer l'arborescence. Chaque dossier **Connection** contient les différentes connexions que vous avez créées.



Grâce au *Studio Talend*, 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.

## 10.2. Centraliser des métadonnées de base de données

Si vous vous connectez régulièrement à une base de données, vous pouvez centraliser les informations de connexion à cette base de données sous le nœud **Metadata** de la vue **Repository**.

Cette procédure est composée de deux tâches séparées mais proches :

1. La configuration d'une connexion à une base de données,
2. La récupération des schémas de table.

Les sections suivantes décrivent dans le détail comment effectuer ces tâches.

**Prérequis :** Le *Studio Talend* requiert l'installation de bibliothèques Java tierces ou de pilotes de bases de données spécifiques (fichiers *.jar*) afin de se connecter à des sources et des cibles. En raison de restrictions liées aux licences, il se peut que **Talend** ne puisse pas intégrer certaines bibliothèques ou pilotes. Dans ce cas, l'assistant de connexion présenté dans les sections suivantes affiche les informations nécessaires pour vous aider à identifier et à installer les bibliothèques ou les pilotes en question. Pour plus d'informations, consultez le *Guide d'installation et de migration Talend*.

### 10.2.1. Configurer une connexion à une base de données

Pour créer une connexion à une base de données à partir de zéro, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **Db Connections** puis sélectionnez **Create connection** dans le menu contextuel afin d'ouvrir l'assistant de connexion à une base de données.

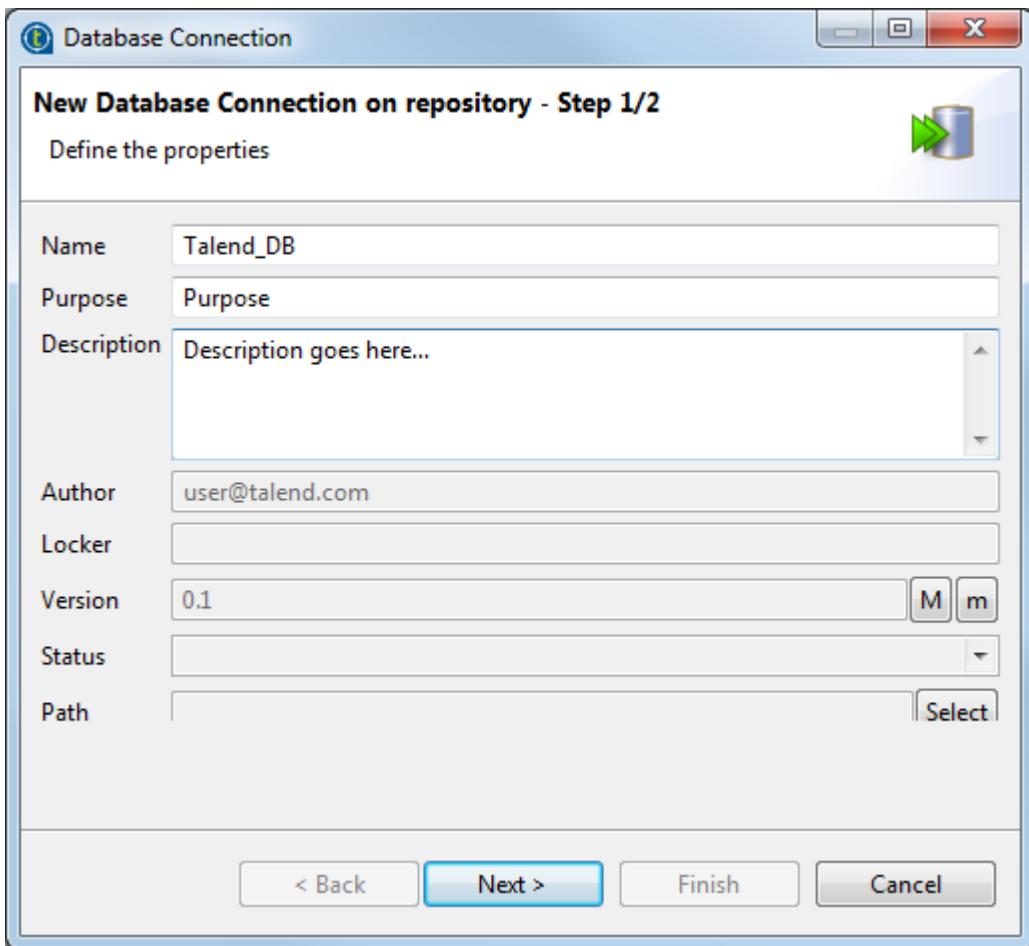
Pour centraliser les paramètres de connexion à une base de données que vous avez définis dans un Job, cliquez sur l'icône  dans la vue **Basic settings** du composant de base de données correspondant. Sa propriété **Property Type** doit être **Built-in** afin d'ouvrir l'assistant de configuration de connexion à une base de données.

Pour modifier une connexion à une base de données existante, cliquez-droit sur la connexion dans le **Repository** puis sélectionnez **Edit connection** afin d'ouvrir l'assistant de configuration de connexion.

Définissez ensuite les propriétés et les paramètres généraux de la connexion dans l'assistant.

#### 10.2.1.1. Définir les propriétés générales

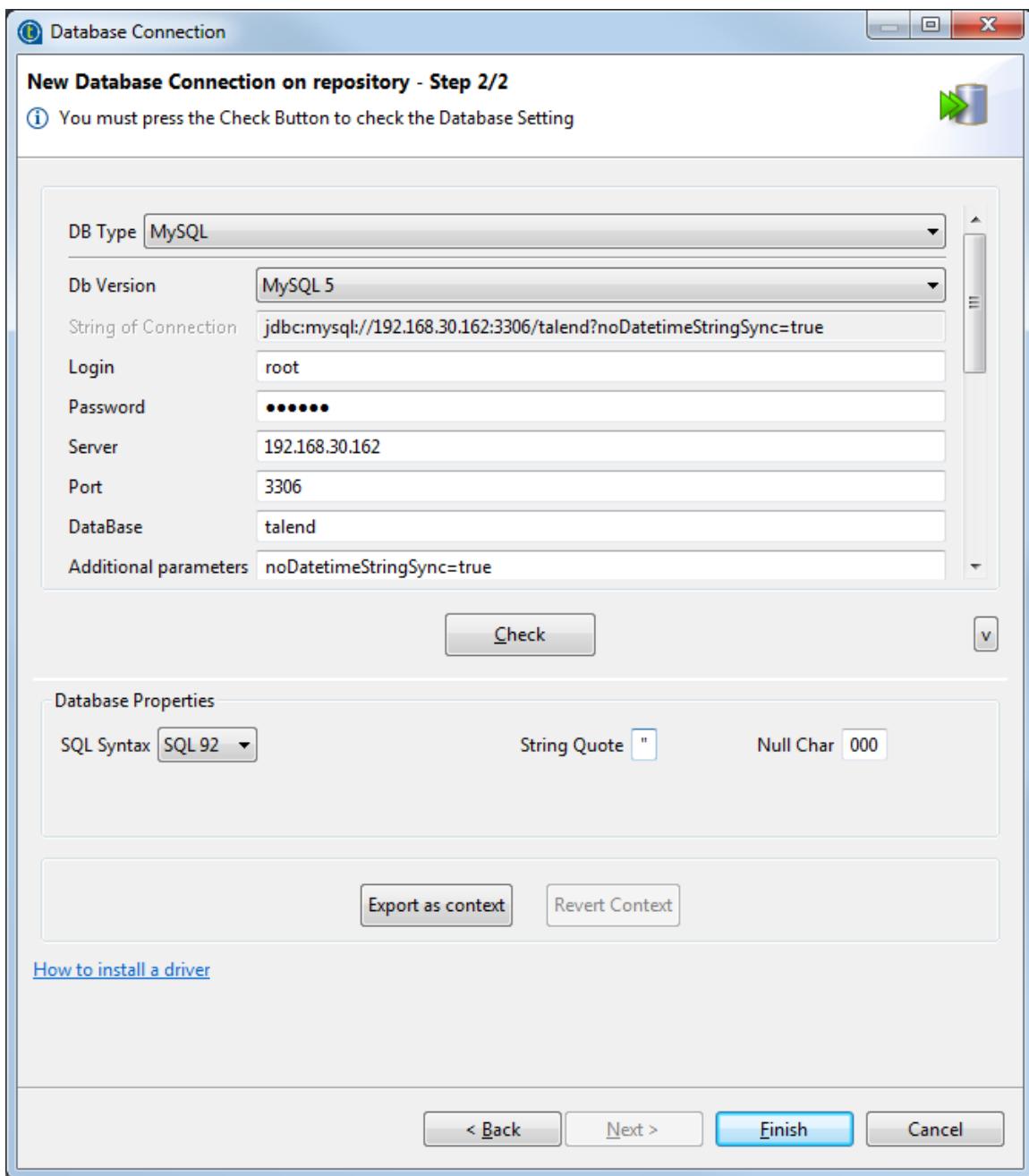
1. Dans l'assistant de connexion qui s'ouvre, donnez un nom à votre connexion dans le champ **Name**. Ce nom apparaît comme nom de la connexion à la base de données sous le nœud **Metadata** du **Repository**.



2. Renseignez les champs facultatifs **Purpose** et **Description**. Les informations que vous saisissez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous déplacez le pointeur de votre souris sur la connexion.
3. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
4. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **Db connections**. Ce dossier contient la connexion à la base de données créée. Notez que vous ne pouvez pas sélectionner de dossier pendant que vous éditez une connexion à une base de données existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.
5. Cliquez sur **Next** lorsque vous avez terminé. L'étape suivante nécessite que vous saisissez ou que vous éditiez des données de connexion à une base de données.

### 10.2.1.2. Définir des paramètres de connexion

1. Sélectionnez le type de base de données auquel vous souhaitez vous connecter et renseignez les détails de connexion. Les champs que vous devez compléter varient selon le type de base de données sélectionné.



Lors de la création d'une connexion à des bases de données telles que AS/400, HSQDB, Informix, Microsoft SQL, MySQL, Oracle, SAP HANA, Sybase, ou Teradata, vous pouvez spécifier des propriétés de connexion supplémentaires dans le champ **Additional parameters** de la zone **Database settings**.

À partir de la version 6.0 du *Studio Talend*, à cause de limitations liées à Java 8, ODBC n'est plus supporté pour les connexions à la base de données Access. Le seul type de pilote supporté est JDBC.

À cause de ces limitations liées à Java 8, vous ne pouvez créer de connexion ODBC générique ou Microsoft SQL Server (ODBC) dans le *Studio Talend* version 6.0 et supérieures, à moins d'importer des connexions précédemment créées dans une ancienne version du *Studio Talend* - dans ce cas, vous pouvez créer des connexions ODBC génériques et Microsoft SQL Server (ODBC) mais elles ne fonctionneront qu'avec Java 7.

Pour une connexion à Microsoft SQL Server (JDBC), quand **Microsoft** est sélectionné dans la liste **Db Version**, vous devez télécharger le pilote Microsoft JDBC pour les serveurs SQL sur [Microsoft Download Center](#) (en anglais). Vous devez ensuite décompresser le fichier zip téléchargé, choisir un fichier .jar dans le dossier décompressé, selon votre version de JRE, renommer le fichier .jar en *mssql-jdbc.jar* et l'installer manuellement. Pour plus d'informations relatives au choix du fichier .jar, consultez la configuration système requise sur [Microsoft Download Center](#) (en anglais).

Si vous souhaitez vous connecter à Hive, il est recommandé d'utiliser l'une des solutions Big Data de Talend.



*Si vous créez une connexion MSSQL, afin de pouvoir récupérer les schémas des tables de la base de données, assurez-vous de :*

- saisir *dbo* dans le champ **Schema** si vous vous connectez à MSSQL 2000,
- retirer *dbo* du champ **Schema** si vous vous connectez à MSSQL 2005/2008.

2. Vérifiez votre connexion à l'aide de la fonction **Check**.

En cas d'échec de la connexion, une fenêtre s'affiche, indiquant que la connexion a échoué. Depuis cette fenêtre, vous pouvez cliquer sur **Details** pour obtenir des informations supplémentaires.

Si l'erreur est due à une bibliothèque ou un pilote (fichier *.jar*) manquant, ouvrez la vue **Details**. Dans cette vue, des informations supplémentaires s'affichent et vous permettent d'installer la bibliothèque ou le pilote manquant.

Le studio fournit plusieurs approches afin d'automatiser l'installation. Pour plus d'informations, consultez le chapitre concernant l'installation de modules externes dans le *Guide d'installation et de migration Talend*.

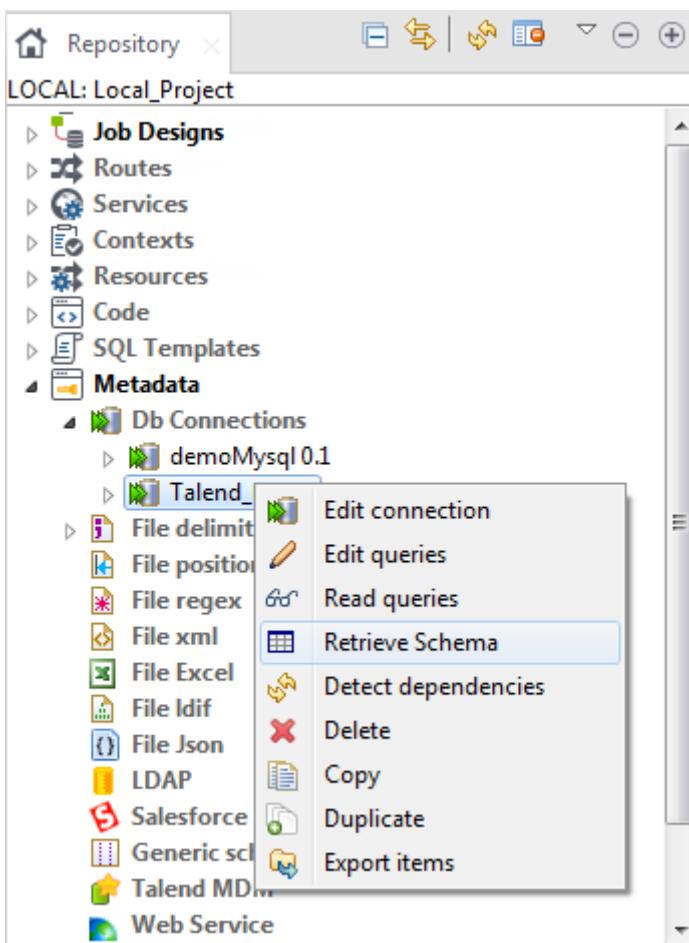
3. Si nécessaire, renseignez les propriétés de base de données dans la zone **Database Properties**. La première phase de configuration d'une connexion à une base de données est terminée. Cliquez sur **Finish** pour valider.

La connexion à la base de données nouvellement créée est accessible depuis le **Repository** et contient plusieurs sous-dossiers, dont **Queries** pour les requêtes SQL et **Table schemas** qui regroupe tous les schémas relatifs à cette connexion, une fois le schéma de la table récupéré.

Vous pouvez à présent déposer la connexion à la base de données dans l'espace de modélisation graphique en tant que composant, afin de réutiliser les détails de connexion à la base de données dans votre Job.

## 10.2.2. Récupérer les schémas de table

Pour récupérer les schéma de table à partir d'une connexion à une base de données définie, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve schema** dans le menu contextuel.

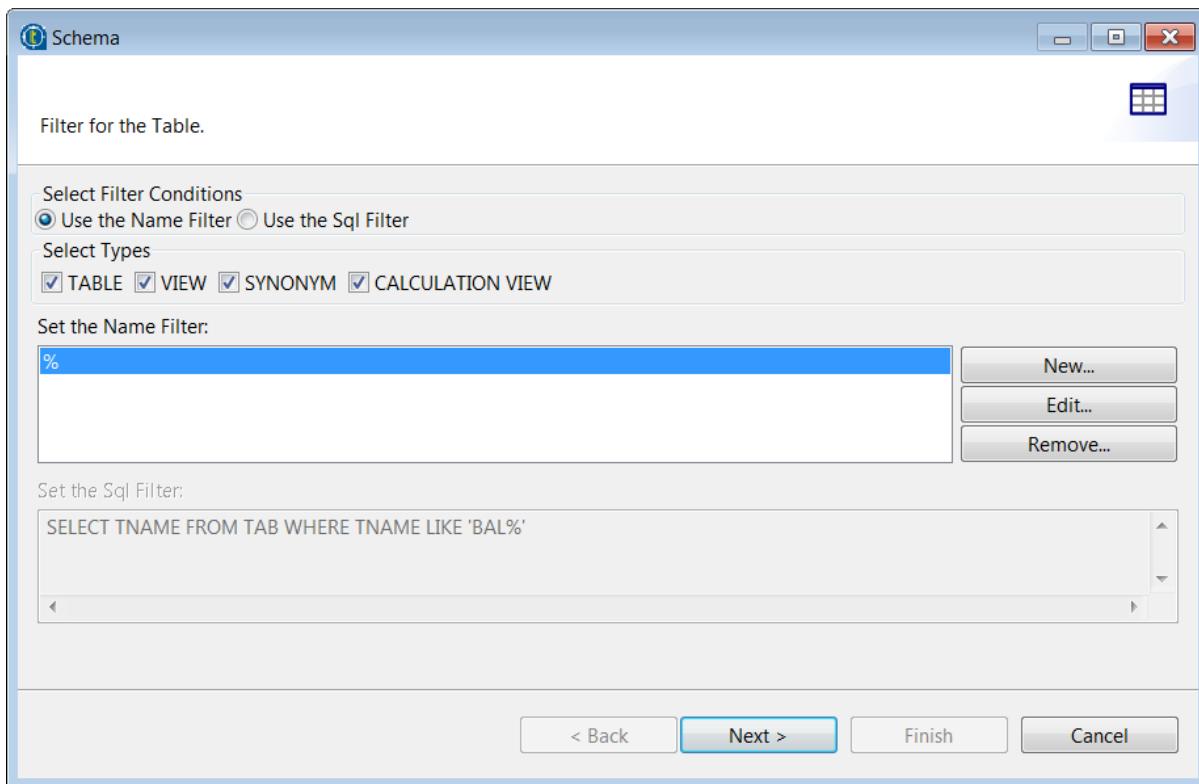


Un message d'erreur s'affiche s'il n'y a aucune table à retrouver dans la base de données sélectionnée ou si vous n'avez pas les droits d'accès à cette base de données.

Un nouvel assistant 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.



### 10.2.2.1. Filtrer les objets de votre base de données

Dans la zone **Select Filter Conditions**, vous pouvez filtrer les tables de votre base de données soit par leur nom à l'aide d'un assistant dans le champ **Set the Name Filter**, soit à l'aide d'une requête SQL dans le champ **Use the Sql Filter**.

Pour filtrer les tables d'une base de données par leur nom, observez les étapes suivantes :

1. Dans la zone **Select Filter Conditions**, sélectionnez l'option **Use the Name Filter**.
2. Dans la zone **Select Types**, cochez la/les case(s) du/des objet(s) de la base de données que vous voulez filtrer ou afficher.

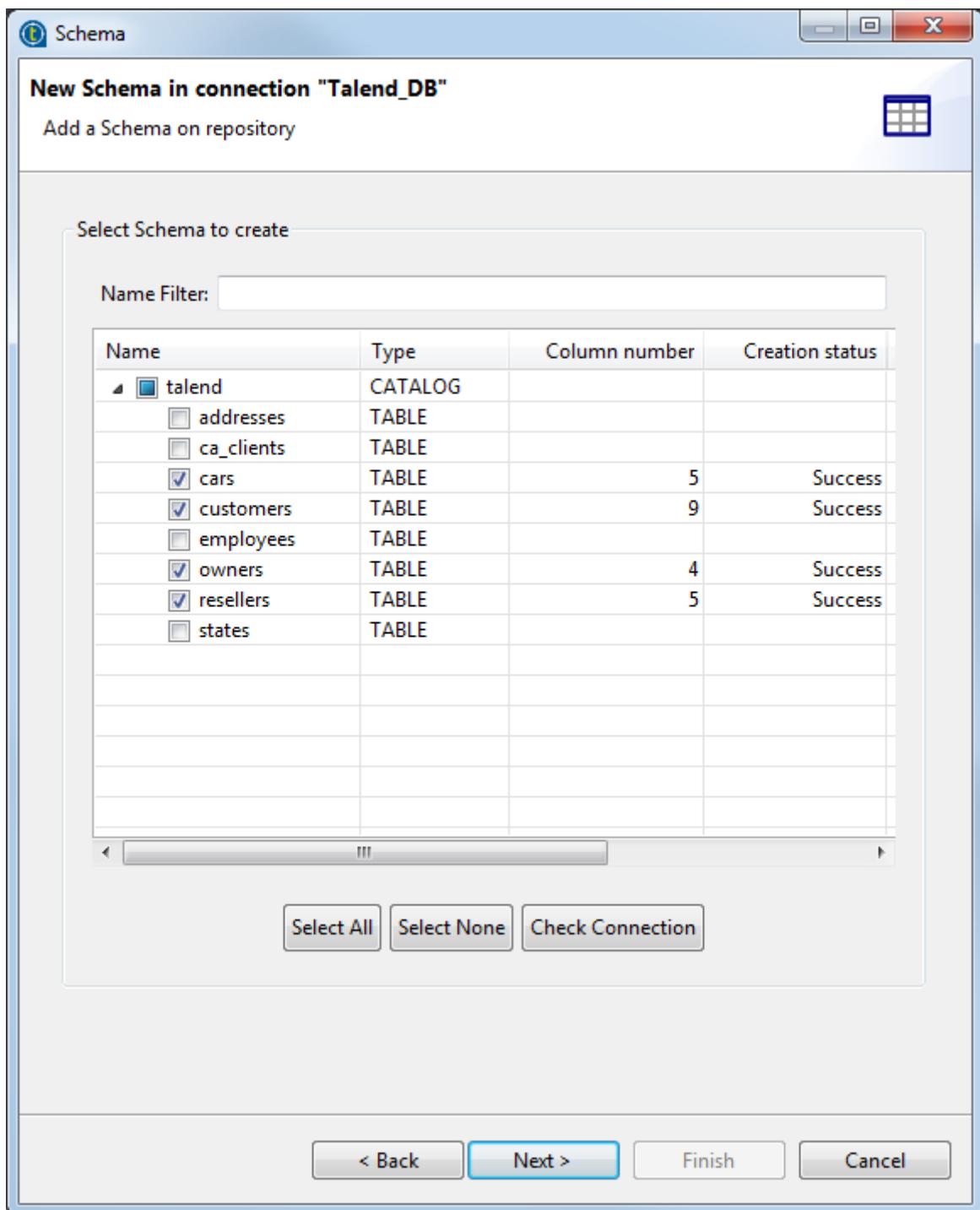


Les options disponibles peuvent varier en fonction de la base de données sélectionnée.

3. Dans la zone **Set the Name Filter**, cliquez sur le bouton **Edit...** pour ouvrir la boîte de dialogue [**Edit Filter Name**].
4. Dans la boîte de dialogue, saisissez le filtre souhaité. Par exemple, si vous souhaitez récupérer les objets dont le nom commence par "A", saisissez le filtre "A%" ou si vous souhaitez récupérer les objets dont le nom termine par "type", saisissez "%type".
5. Cliquez sur **OK** pour fermer la boîte de dialogue.
6. Cliquez sur **Next** pour afficher la vue suivante de l'assistant, qui présente la liste des objets filtrés.

Pour filtrer les objets à l'aide d'une requête SQL :

1. Dans la zone **Select Filter Conditions**, sélectionnez l'option **Use Sql Filter**.
2. Dans le champ **Set the Sql Filter**, saisissez la requête SQL souhaitée.
3. Cliquez sur **Next** pour afficher la vue suivante de l'assistant, qui présente la liste des objets filtrés.



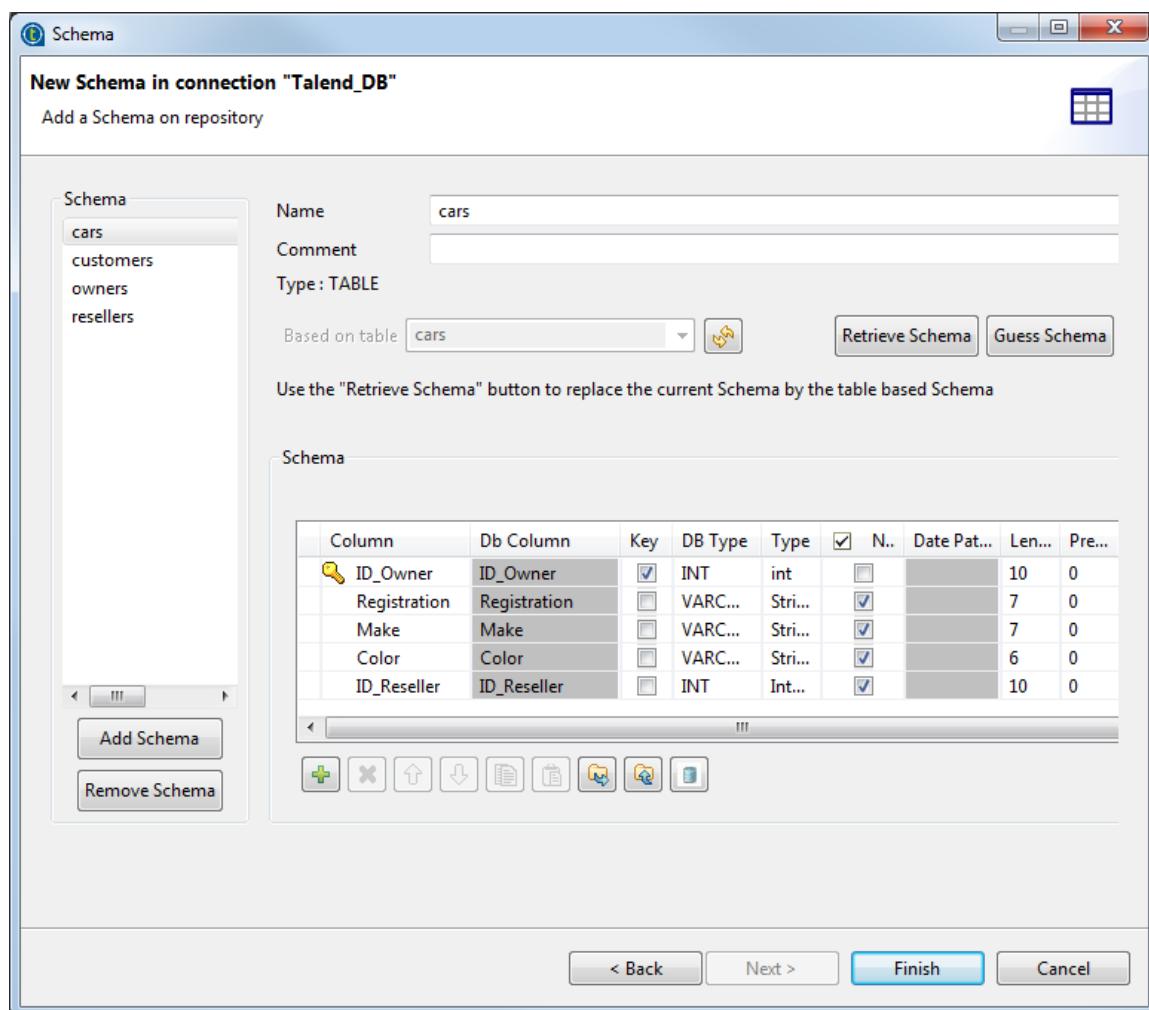
### 10.2.2.2. Sélectionner les tables et définir les schémas de table

Une fois que vous avez filtré la liste des objets de base de données (tables, vues et synonymes), procédez comme suit afin de charger les schémas des objets souhaités dans votre **Repository** :

1. Sélectionnez un ou plusieurs objets à partir de la liste puis cliquez sur le bouton **Next** pour ouvrir la vue suivante de l'assistant, où vous pourrez consulter les schémas des objets sélectionnés.



Si aucun schéma n'est visible dans la liste, cliquez sur le bouton **Check connection** en dessous de la liste pour vérifier l'état de la connexion de base de données.



## 2. Modifiez les schémas si besoin.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- **Object** : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- **List** : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- **Document** : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.



*Si la table de la base de données source contient une valeur par défaut étant une fonction ou une expression et non une chaîne de caractères, assurez-vous de supprimer les guillemets simples entourant la valeur par défaut du schéma final, s'il y en a, afin d'éviter des résultats inattendus lors de la création de tables de bases de données à l'aide de ce schéma.*

*Pour plus d'informations, consultez Vérification des valeurs par défaut dans un schéma récupéré à l'adresse <https://help.talend.com>.*

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 de 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 dans le sous-dossier **Table schemas** sous le nœud correspondant à votre base de données.

Vous pouvez à présent déposer n'importe quel schéma de table d'une connexion à une base de données depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant de base de données. Vous pouvez également le déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

## 10.3. Centraliser des métadonnées JDBC

Pour centraliser des métadonnées basées sur des tables de base de données dans une connexion JDBC, sous le nœud **Metadata** du **Repository**, la procédure est composée de deux tâches séparées mais proches :

1. La configuration d'une connexion JDBC,
2. La récupération des schémas de table.

Les sections suivantes décrivent dans le détail comment effectuer ces tâches.

### 10.3.1. Configurer une connexion JDBC

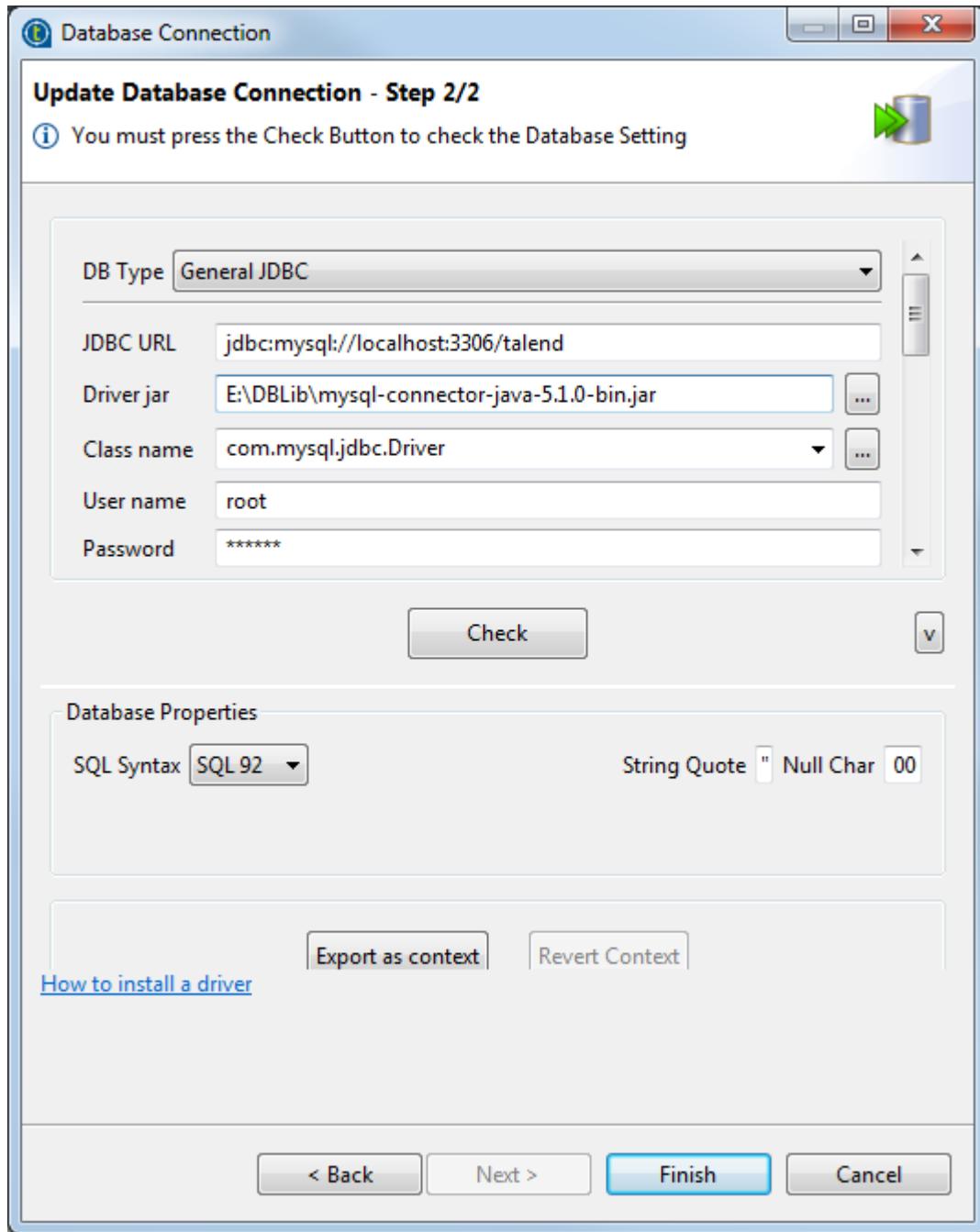
1. Pour créer une connexion JDBC à partir de zéro, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **Db Connections** puis sélectionnez **Create connection** dans le menu contextuel afin d'ouvrir l'assistant de connexion à une base de données.

Pour centraliser les paramètres de connexion à une base de données que vous avez définis dans une connexion JDBC d'un Job, cliquez sur l'icône  dans la vue **Basic settings** du composant de base de données correspondant. Sa propriété **Property Type** doit être **Built-in** afin d'ouvrir l'assistant de configuration de connexion à une base de données.

Pour modifier une connexion JDBC existante, cliquez-droit sur la connexion dans le **Repository** puis sélectionnez **Edit connection** afin d'ouvrir l'assistant de configuration de connexion.

2. Renseignez les informations génériques du schéma, notamment les champs **Name** et **Description** puis cliquez sur **Next** afin de passer à l'étape de définition des détails de connexion.

Pour plus d'informations, consultez la section sur la définition des propriétés générales dans [Configurer une connexion à une base de données](#).

3. Sélectionnez **General JDBC** dans la liste **DB type**.

## 4. Renseignez les informations requises pour la connexion comme suit :

- Dans le champ **JDBC URL**, renseignez l'URL de connexion au serveur SGBD.
- Dans le champ **Driver 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 votre nom et mot de passe utilisateur.
- Dans le champ **Mapping File**, renseignez le mapping permettant de faire la corrélation entre le type de données de la base de données et le type de données en Java. Pour ce faire, cliquez sur le bouton

afin d'ouvrir une boîte de dialogue et, depuis la zone **Mapping list**, sélectionnez le fichier de mapping correspondant au type de base de données auquel vous vous connectez.

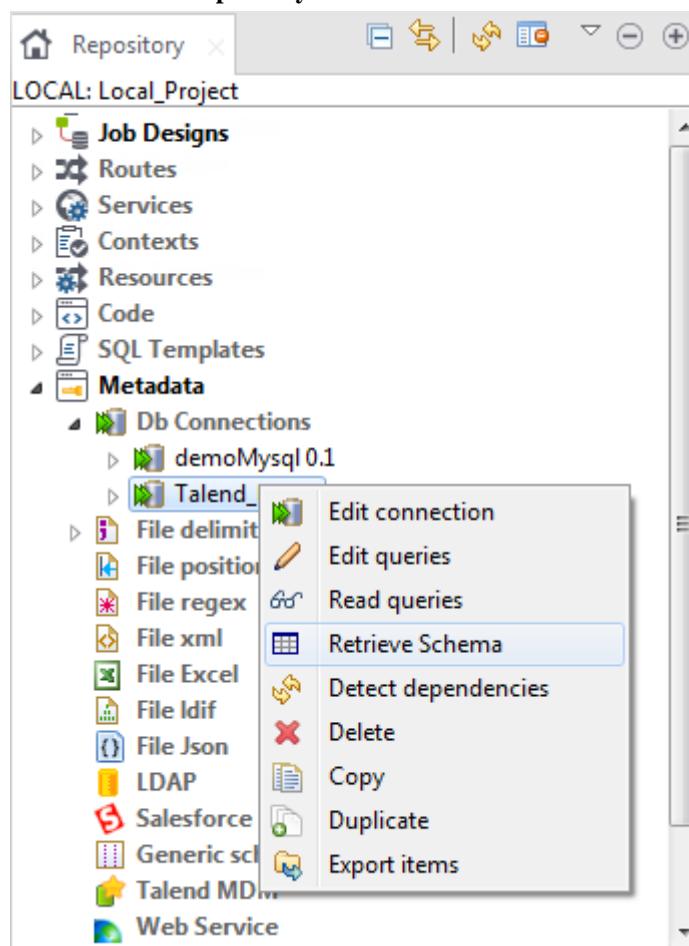
 Les fichiers de mapping sont au format XML et sont gérés dans le menu **Window > Preferences > Talend > Specific Settings > Metadata of TalendType**. Pour plus d'informations, consultez [Mapping de type](#).

5. Vérifiez votre connexion à l'aide de la fonction **Check**.
6. Saisissez, si nécessaire, les informations relatives aux propriétés de la base de données. Cliquez sur **Finish** pour fermer l'assistant de configuration de connexion.

La connexion JDBC nouvellement créée est accessible depuis le **Repository** et contient plusieurs sous-dossiers dont **Queries** pour les requêtes SQL et **Table schemas** qui regroupe tous les schémas relatifs à cette connexion une fois ces derniers récupérés.

### 10.3.2. Récupérer des schémas de table

1. Pour récupérer des schémas de table depuis la connexion de base de données que vous venez de configurer, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve schema** dans le menu contextuel.



Dans l'assistant qui s'ouvre, vous pouvez filtrer et afficher différents objets (tables, vues et synonymes) de votre connexion à la base de données, sélectionner des tables et définir des schémas de table.

2. Configurez le filtre sur les objets de bases de données selon vos besoins. Pour plus de détails, consultez [Filtrer les objets de votre base de données](#).

Cliquez sur **Next** pour ouvrir une vue listant les objets de bases de données que vous avez filtrés. La liste montre toutes les bases de données avec toutes leurs tables présentes dans la connexion de base de données qui correspondent aux conditions de filtre que vous avez définies.

Si aucune base de données n'est visible dans la liste, cliquez sur **Check connection** afin de vérifier la connexion à la base de données.

3. Sélectionnez une ou plusieurs tables dans la liste pour les charger sur votre système de fichier Référentiel. Vos schémas de **Repository** sont basés sur ces tables.
4. Cliquez sur **Next**. Dans la fenêtre suivante, quatre panneaux de configuration vous permettent de définir les schémas à créer. Modifiez les schémas si nécessaire.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.



*Si la table de la base de données source contient une valeur par défaut étant une fonction ou une expression et non une chaîne de caractères, assurez-vous de supprimer les guillemets simples entourant la valeur par défaut du schéma final, s'il y en a, afin d'éviter des résultats inattendus lors de la création de tables de bases de données à l'aide de ce schéma.*

*Pour plus d'informations, consultez Vérification des valeurs par défaut dans un schéma récupéré à l'adresse <https://help.talend.com>.*

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.

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.

Quand vous avez apporté les changements souhaités, cliquez sur **Finish** pour terminer la création du schéma de base de données. Tous les schémas ainsi créés apparaissent dans le sous-dossier **Table schemas** du nœud correspondant à votre base de données.

Vous pouvez à présent déposer n'importe quel schéma de table d'une connexion à une base de données depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant de base de données. Vous pouvez également le déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

## 10.4. Centraliser des métadonnées SAS

Si vous devez souvent vous connecter à un système distant SAS, vous pouvez centraliser les informations de connexion dans le **Repository**.

Pour centraliser les métadonnées d'information d'une connexion SAS dans le **Repository**, vous devez au préalable :

1. Configurer une connexion SAS,
2. Récupérer les schémas de base de données.

### Prérequis :

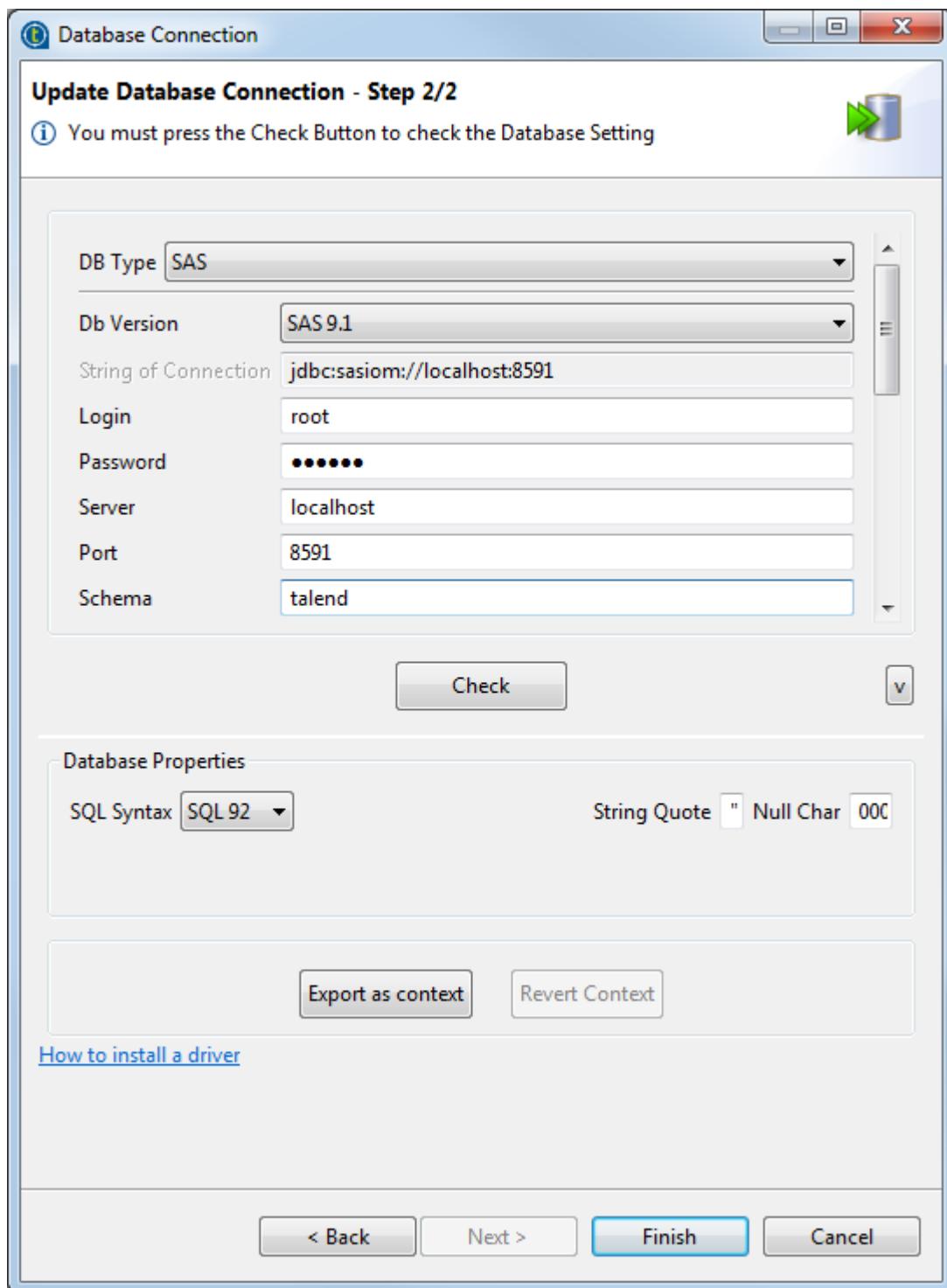
- Le *Studio Talend* requiert l'installation de bibliothèques Java tierces ou de pilotes de bases de données spécifiques (fichiers *.jar*) afin de se connecter à des sources et des cibles. En raison de restrictions liées aux licences, il se peut que **Talend** ne puisse pas intégrer certaines bibliothèques ou pilotes. Dans ce cas, l'assistant de connexion présenté dans les sections suivantes affiche les informations nécessaires pour vous aider à identifier et à installer les bibliothèques ou les pilotes en question. Pour plus d'informations, consultez le *Guide d'installation et de migration Talend*.
- Avant d'effectuer la procédure suivante pour configurer votre connexion SAS, assurez-vous de récupérer vos métadonnées depuis le serveur SAS et de les exporter au format XML.

### 10.4.1. Configurer une connexion SAS

1. Dans l'arborescence du **Repository** du *Studio Talend*, cliquez-droit sur **DB Connection** sous le nœud **Metadata** et sélectionnez **Create connection** dans le menu contextuel afin d'ouvrir l'assistant [**Database Connection**].
2. Renseignez les propriétés générales de la connexion, comme le nom (**Name**) et la description (**Description**) puis cliquez sur le bouton **Next** pour passer à la vue suivante de l'assistant afin de définir les détails de la connexion.

Pour plus d'informations, consultez la section sur la définition des propriétés générales dans [Configurer une connexion à une base de données](#).

3. Dans le champ **DB Type** de l'assistant [**Database Connection**], sélectionnez l'option **SAS** et renseignez les champs qui suivent avec vos informations de connexion SAS.

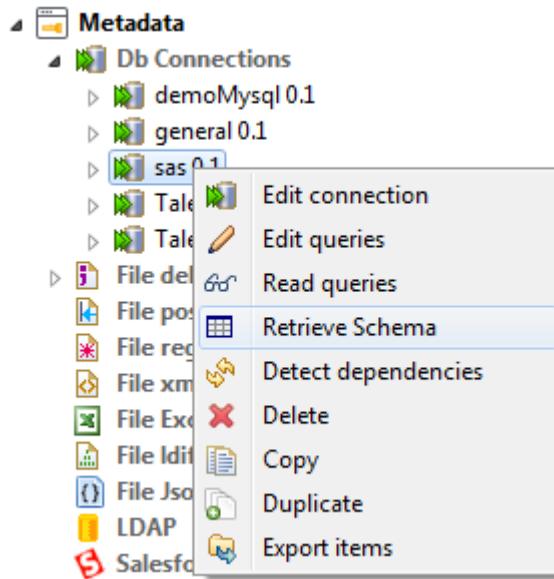


4. Si nécessaire, vérifiez votre connexion à l'aide de la fonction **Check**.
5. Si nécessaire également, renseignez les propriétés de base de données dans la zone **Database Properties**.
6. Cliquez sur **Finish** pour valider vos modifications et fermer l'assistant.

La connexion à la base de données nouvellement créée est accessible depuis le **Repository** sous le nœud **DB Connections**. Elle contient plusieurs sous-dossiers dont **Table schemas** qui regroupe tous les schémas relatifs à cette connexion une fois ces derniers récupérés.

## 10.4.2. Récupérer des schémas de table

1. Cliquez-droit sur la connexion SAS nouvellement créée et sélectionnez **Retrieve Schema** dans le menu contextuel.



Dans l'assistant qui s'ouvre, vous pouvez filtrer et afficher différents objets (tables, vues) de votre connexion à la base de données, sélectionner des tables et définir des schémas de table.

2. Filtrez les objets de base de données selon vos besoins, sélectionnez une ou plusieurs tables et modifiez les tables de schéma si nécessaire. Pour plus de détails, consultez [Récupérer les schémas de table](#).

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

Quand vous avez apporté les changements souhaités, vous pouvez déposer n'importe quel schéma de table de la connexion SAS depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également le déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

## 10.5. Centraliser des métadonnées File Delimited

Si vous devez souvent lire et/ou écrire des données dans des fichiers délimités, vous pouvez centraliser les métadonnées de ces fichiers dans le **Repository** afin de les réutiliser facilement. Les métadonnées de type File Delimited peuvent être utilisées pour définir les propriétés des composants **tInputFileDelimited**, **tFileOutputDelimited** et **t\*OutputBulk**.

 La création de schémas de fichier est similaire pour tous les types de fichiers : Delimited (délimités), Positional (positionnels), Regex (Expressions régulières), XML ou LDIF.

A la différence de l'assistant de connexion à une base de données, l'assistant [**New Delimited File**] regroupe la connexion au fichier et la définition du schéma, en une seule procédure de quatre étapes.

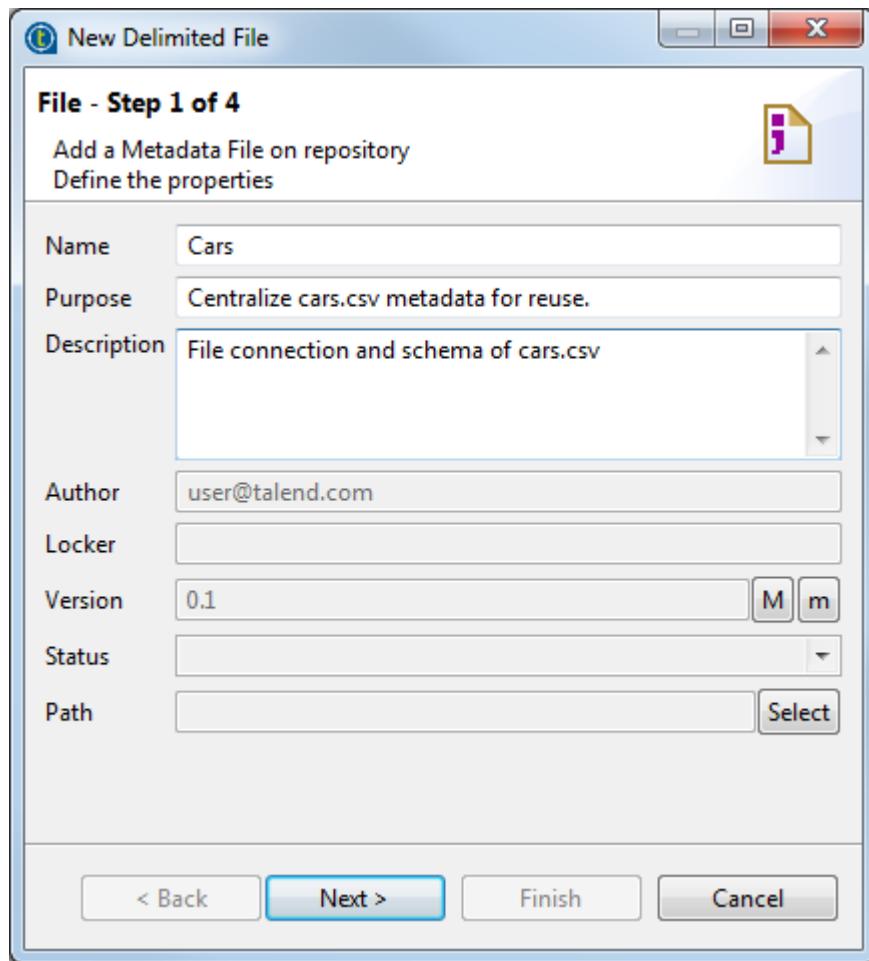
Pour créer une connexion à un fichier délimité à partir de zéro, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **File Delimited** puis sélectionnez **Create file delimited** dans le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser une connexion à un fichier que vous avez définie dans un Job ainsi que son schéma, cliquez sur l'icône  dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

Définissez ensuite les propriétés générales et le schéma du fichier dans l'assistant.

### Définir les propriétés générales

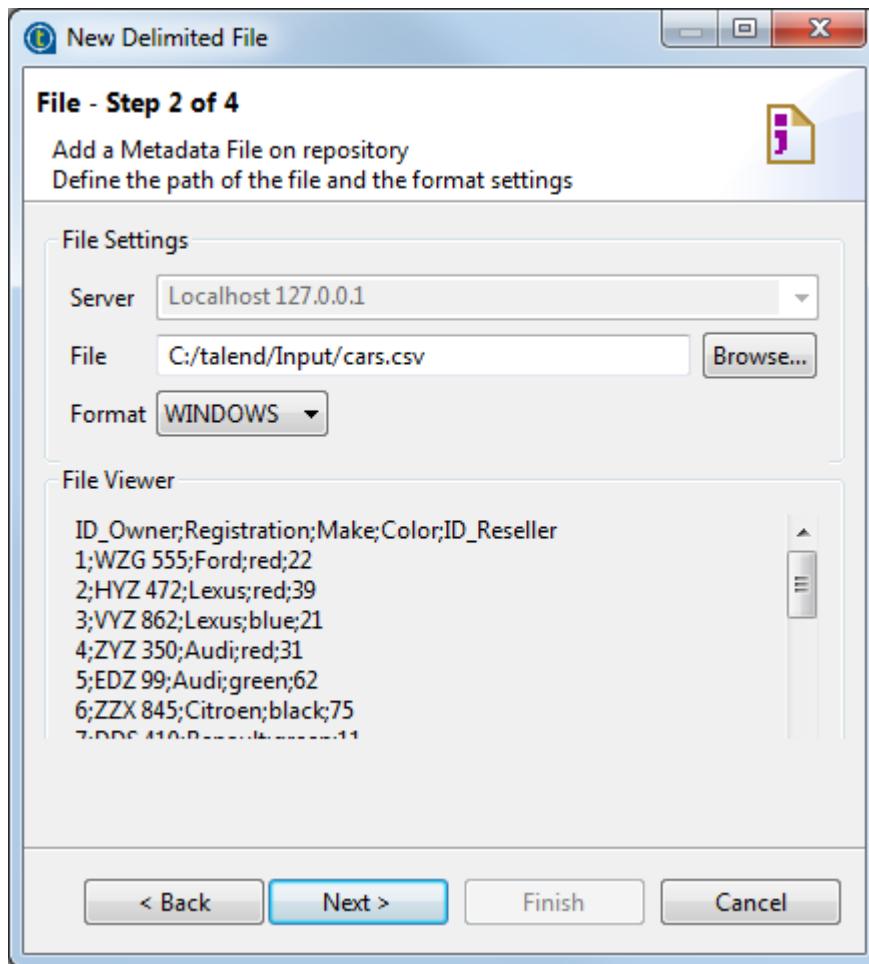
1. Dans l'assistant de configuration des métadonnées du fichier, renseignez le champ **Name**, qui est obligatoire, et les champs **Purpose** et **Description** si vous le souhaitez. Les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File delimited**. Ce dossier contient la connexion au fichier juste créé. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous éditez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.
4. Cliquez sur **Next >** lorsque vous avez défini les propriétés générales.

### Définir le chemin et le format du fichier

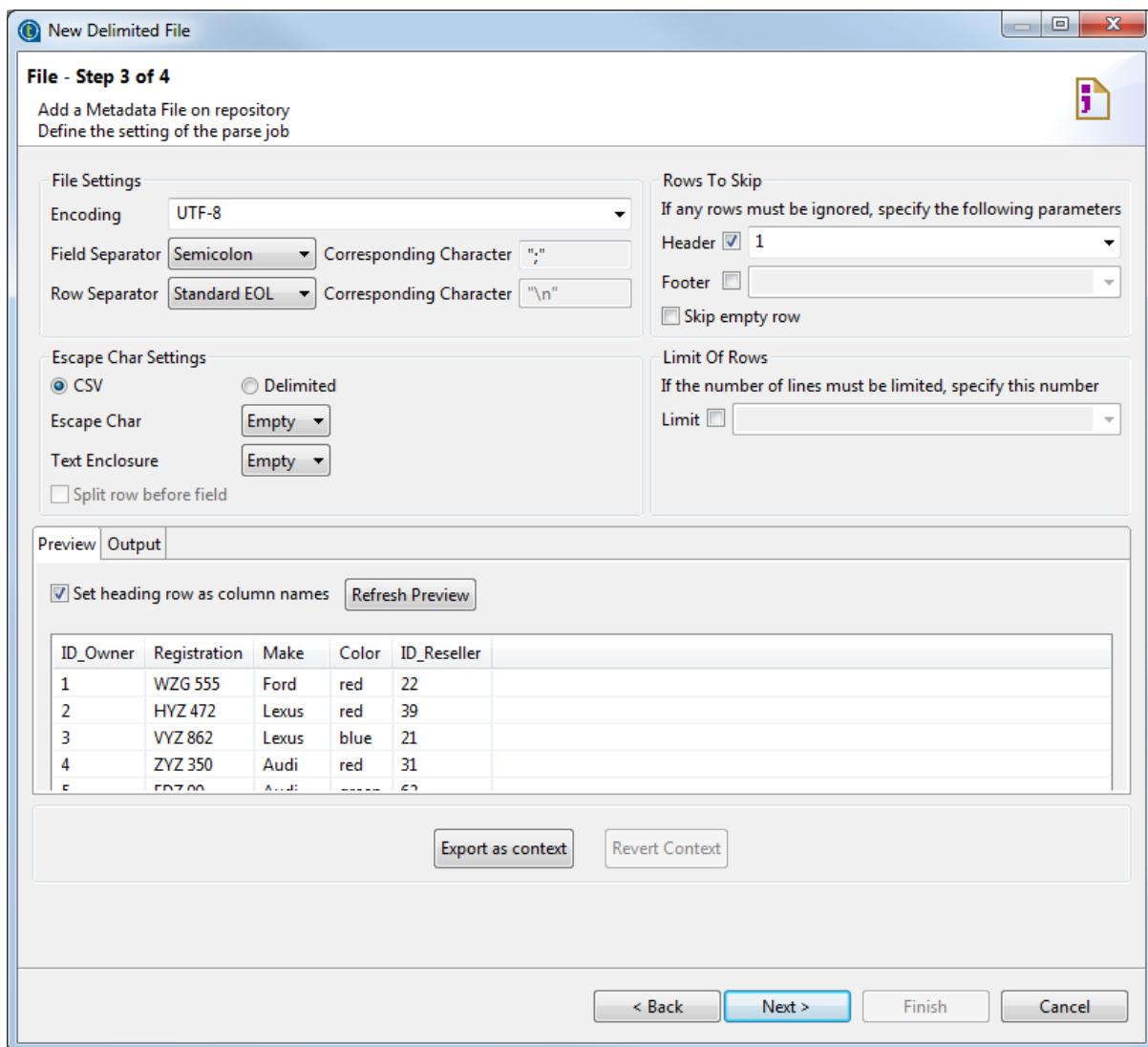
1. Cliquez sur le bouton **Browse...** pour rechercher votre fichier, dans votre hôte local ou un hôte LAN.



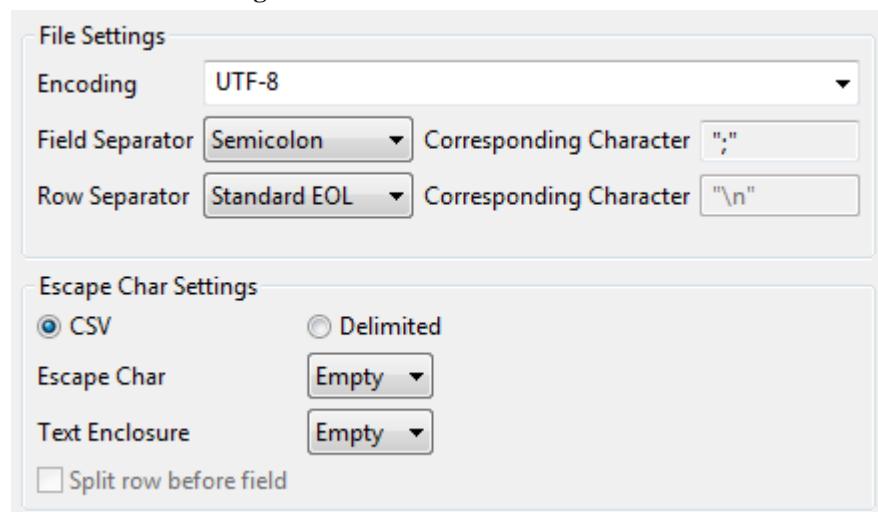
2. Sélectionnez le **Format** du système d'exploitation dans lequel le fichier a été créé. Cette information est utilisée pour pré-remplir les champs des étapes suivantes. Si la liste ne propose pas le format approprié, ignorez ce champ.
3. Le **File viewer** donne un instantané du fichier chargé. Vérifiez la cohérence du fichier, la présence d'un en-tête et plus généralement la structure du fichier.
4. Cliquez sur **Next** pour passer à l'étape suivante.

## Définir le parsing du fichier

Dans cette vue, vous pouvez affiner différents paramètres de votre fichier afin que le schéma soit correctement récupéré.

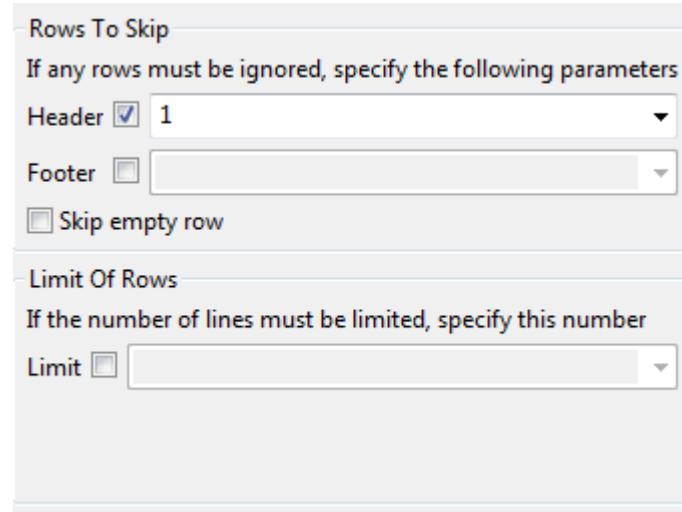


- Paramétrez le type d'encodage (**Encoding**) et les séparateurs de champs et de lignes (**Field separator** et **Row separator**) dans la zone **File Settings**.



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

3. Si l'aperçu du fichier indique un message d'en-tête, excluez 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.



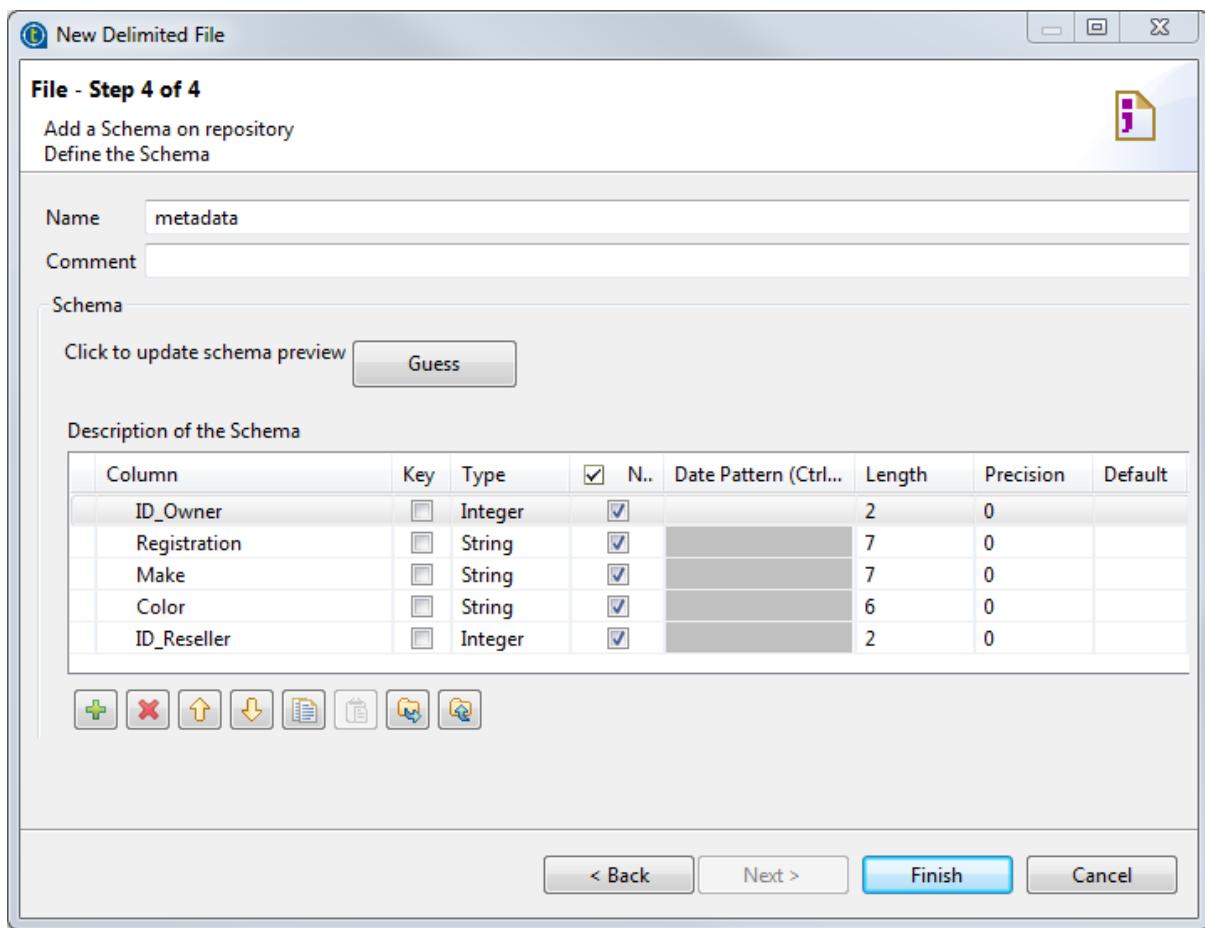
4. Le champ **Limit of Rows** vous permet de restreindre l'étendue du fichier qui fait l'analyse. Si nécessaire, cochez la case **Limit** et définissez ou sélectionnez le nombre de lignes souhaitées.
5. Dans le panneau **File Preview**, vous pouvez visualiser l'impact de vos paramétrages.
6. Cochez la case **Set heading row as column names** pour transformer la première ligne analysée en libellés des colonnes du schéma. Notez que le nombre de lignes d'en-tête à ignorer (champ **Rows To Skip**) est incrémenté de 1.

		Preview	Output
<input checked="" type="checkbox"/> Set heading row as column names		<input type="button" value="Refresh Preview"/>	
ID_Owner	Registration	Make	Color
1	WZG 555	Ford	red
2	HYZ 472	Lexus	red
3	VYZ 862	Lexus	blue
4	ZYZ 350	Audi	red
5	EDZ 99	Audi	green
6	ZZX 845	Citroen	black

7. Cliquez sur **Refresh** dans le panneau d'aperçu, afin que les modifications de paramétrage prennent effet.
8. Cliquez sur **Next** pour accéder à l'étape suivante permettant de vérifier et de personnaliser le schéma de fichier généré.

## Vérifier et personnaliser le schéma du fichier

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



Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
  - List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
  - Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
1. Si le fichier délimité sur lequel est basé le schéma a été modifié, utilisez le bouton **Guess** pour générer le schéma à nouveau. Notez que si vous personnalisez le schéma, la fonctionnalité **Guess** ne retiendra pas ces modifications.
  2. 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 à présent déposer n'importe quelle connexion à un fichier ou n'importe quel schéma depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file delimited** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.6. Configurer un schéma File Positional

Si vous devez souvent lire et/ou écrire des données dans des fichiers positionnels, vous pouvez centraliser les métadonnées de ces fichiers dans le **Repository** afin de les réutiliser facilement. Les métadonnées de type File Positional peuvent être utilisées pour définir les propriétés des composants **tInputFilePositional**, **tFileOutputPositional** et **tFileInputMSPositional**.

L'assistant **[New Positional File]** regroupe la connexion au fichier et la définition du schéma en une seule procédure de quatre étapes.

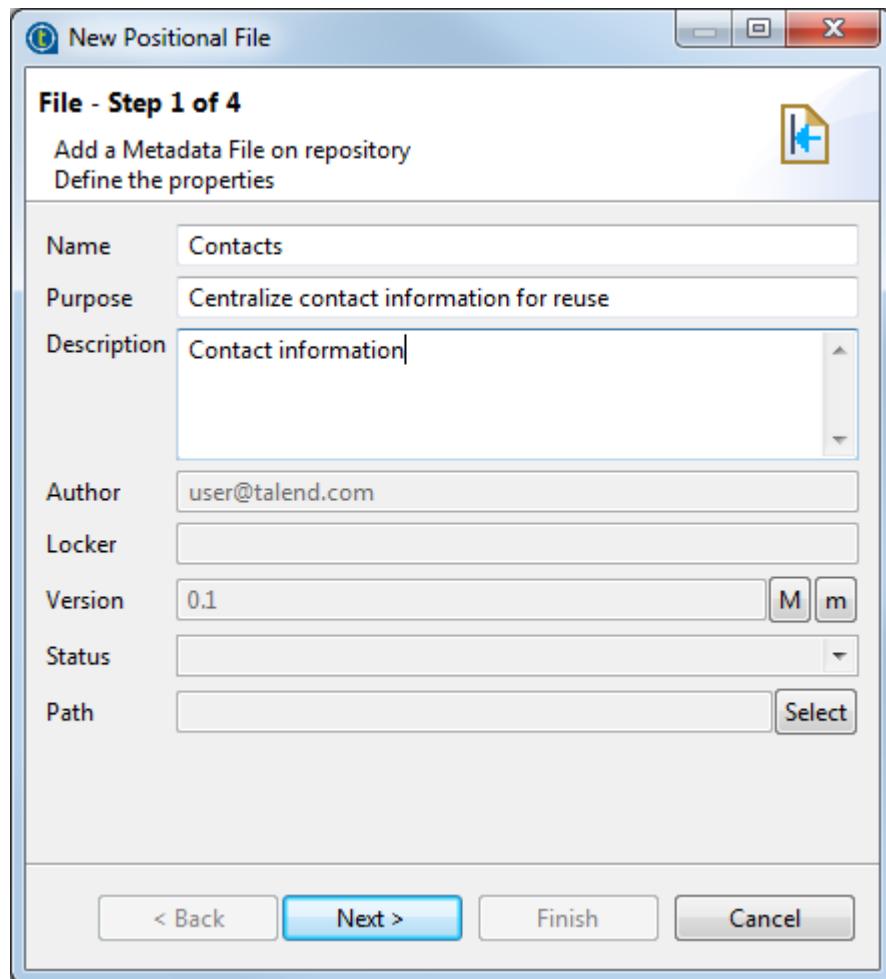
Pour créer une connexion à un fichier délimité à partir de zéro, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **File Positional** puis sélectionnez **Create file positional** dans le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser une connexion à un fichier que vous avez définie dans un Job ainsi que son schéma, cliquez sur l'icône dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

Définissez ensuite les propriétés générales et le schéma du fichier dans l'assistant.

### Définir les propriétés générales

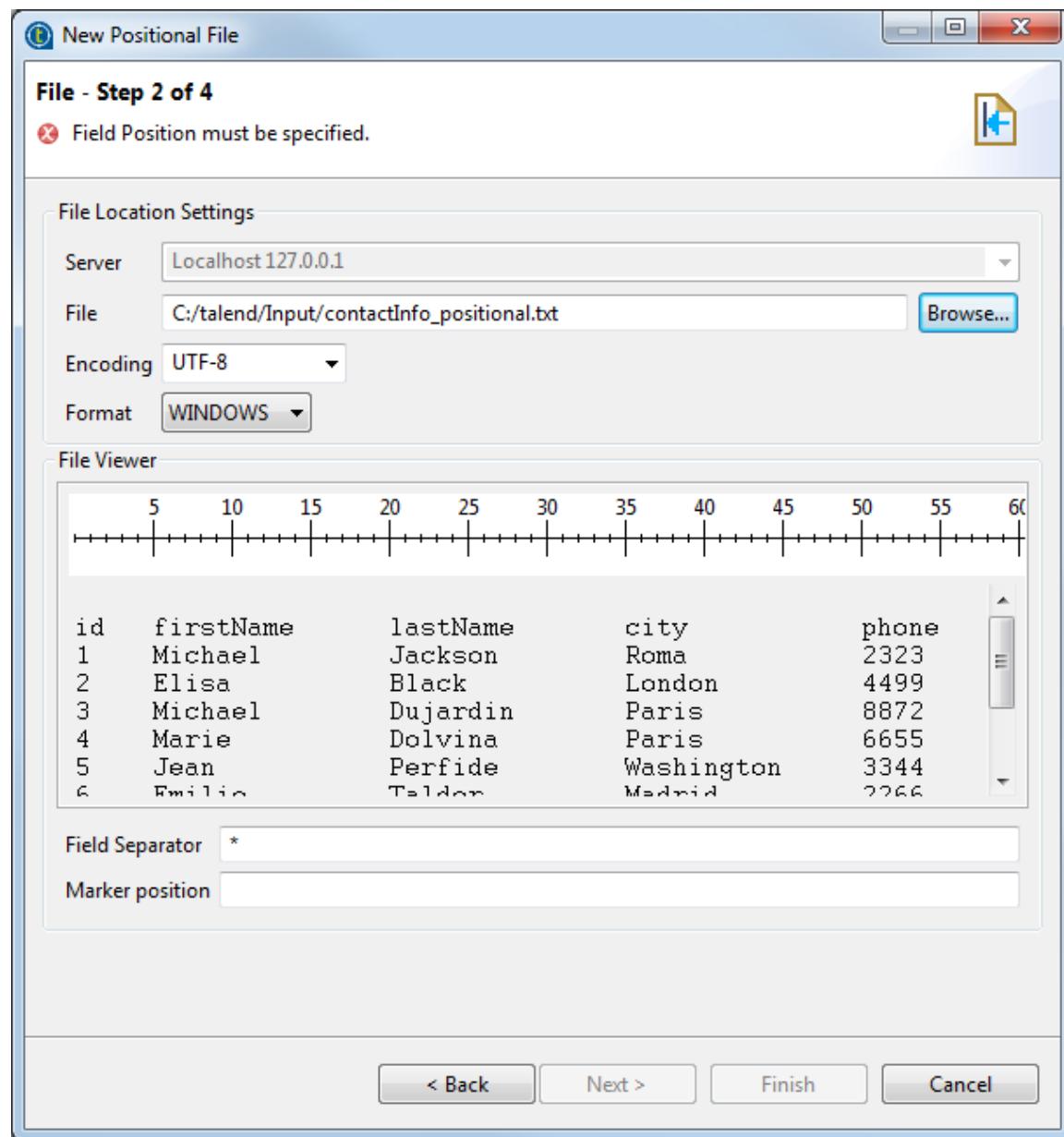
1. Dans l'assistant de configuration des métadonnées du fichier, renseignez le champ **Name**, qui est obligatoire et les champs **Purpose** et **Description** si vous le souhaitez. Les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File positional**. Ce dossier contient la connexion au fichier créée. Notez que vous ne pouvez pas sélectionner de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.
4. Cliquez sur **Next** lorsque vous avez configuré les propriétés générales.

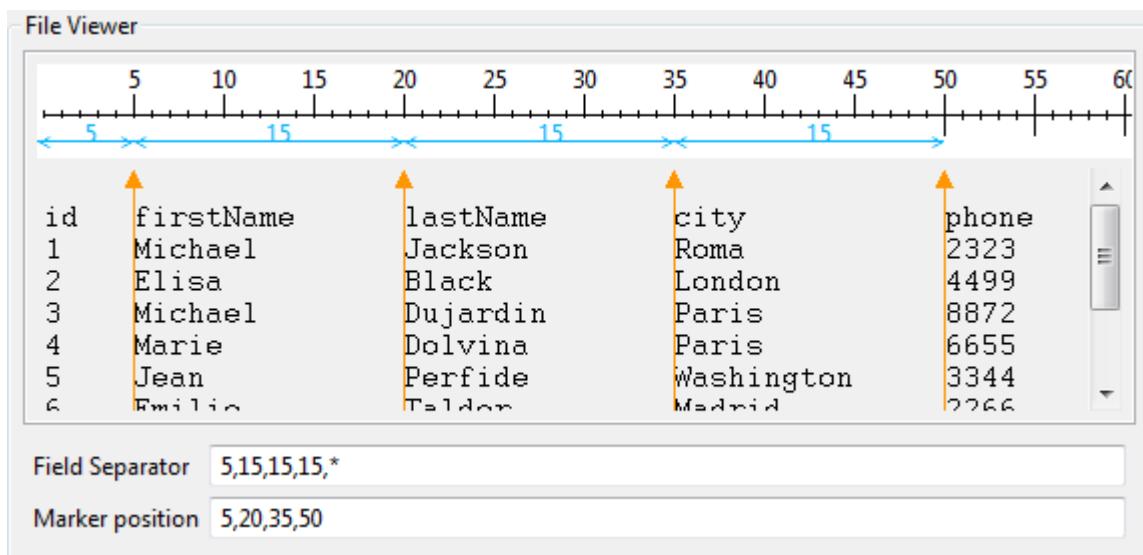
### Définir le chemin, le format et les marqueurs de position du fichier

1. Sélectionnez le type d'encodage (**Encoding**) et le format du système d'exploitation (**Format**) dans lequel le fichier a été créé. Ces informations sont utilisées afin de pré-remplir les champs suivants. Si le système d'exploitation souhaité n'est pas proposé dans la liste, ignorez ce champ.



Le fichier est chargé et la zone **File Viewer** donne un instantané du fichier et vous permet de placer les marqueurs de position.

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



Les champs **Field Separator** et **Marker position** sont automatiquement remplis par une série de chiffres séparés par des virgules.

Les chiffres dans le champ **Field Separator** représentent le nombre de caractères entre les séparateurs, c'est-à-dire la longueur des colonnes dans le fichier chargé. L'astérisque symbolise tous les caractères restants jusqu'à la fin de la ligne, à partir du séparateur précédent. Vous pouvez changer ces chiffres afin de spécifier la longueur des colonnes de façon précise.

Le champ **Marker position** indique la position exacte de chaque marqueur sur la règle, en nombre de caractères. Vous pouvez affiner la position exacte du marqueur en saisissant la valeur exacte de position.

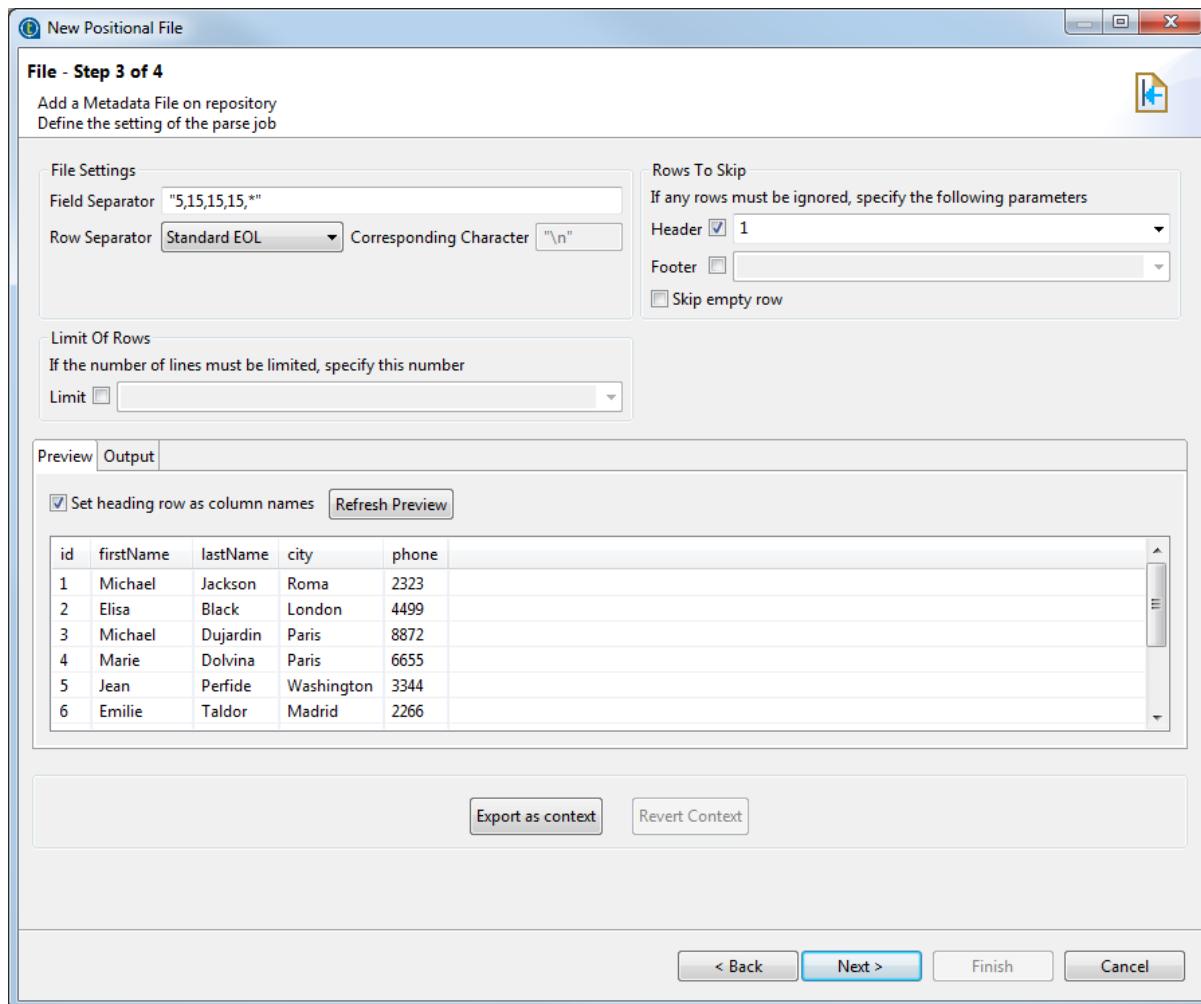
Afin de déplacer un marqueur, cliquez sur une flèche et glissez-la vers sa nouvelle position. Pour retirer un marqueur, cliquez sur sa flèche et faites-la glisser vers la règle, jusqu'à ce qu'une icône apparaisse.

3. Cliquez sur **Next** pour continuer.

### Configurer les paramètres de parsing du fichier

Dans cette vue, vous pouvez définir les paramètres de parsing du fichier afin que le schéma du fichier soit correctement récupéré.

A ce stade, l'aperçu affiche les colonnes du fichier par la position de marqueurs.



1. Définissez les séparateurs de champs et de lignes dans la zone **File Settings**.
  - Si nécessaire, vous pouvez modifier les chiffres dans le champ **Field Separator** afin de spécifier la longueur des colonnes de façon précise.
  - Si le séparateur de lignes de votre fichier n'est pas le caractère de fin de ligne standard, sélectionnez **Custom String** dans la liste **Row Separator** et spécifiez le caractère dans le champ **Corresponding Character**.
2. Si votre fichier possède des en-têtes à exclure du contenu des données, cochez la case **Header** dans la zone **Rows To Skip** et configurez le nombre de lignes à ignorer dans le champ correspondant. De plus, si vous savez que le fichier contient des informations de pied de page, cochez la case **Footer** et définissez le nombre de lignes à ignorer.
3. La zone **Limit of Rows** vous permet de restreindre l'étendue du fichier à analyser. Si nécessaire, cochez la case **Limit** et définissez ou sélectionnez le nombre de lignes souhaité.
4. Si le fichier contient des libellés de colonne, cochez la case **Set heading row as column names** afin de transformer la première ligne en libellés de colonne. Notez que le nombre de lignes d'en-tête à ignorer est incrémenté de 1.
5. Cliquez sur le bouton **Refresh Preview** dans le panneau de prévisualisation (**Preview**) afin que les paramètres prennent effet. Un aperçu du résultat s'affiche dans le panneau **Preview**.
6. Cliquez sur **Next** afin passer à l'étape suivante pour vérifier et personnaliser le schéma de fichier généré.

## Vérifier et personnaliser le schéma du fichier

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 remplace les astérisques.

1. Renommez le schéma (par défaut, *metadata*) et modifiez les colonnes du schéma comme souhaité.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

2. Pour générer à nouveau le schéma du fichier positionnel, cliquez sur le bouton **Guess**. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur **Guess**.

3. Une fois vos changements effectués, cliquez sur **Finish** pour fermer l'assistant.

Le nouveau schéma apparaît dans l'arborescence du **Repository**, sous le nœud de la connexion **File positional** appropriée. Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file positional** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.7. Centraliser des métadonnées File Regex

Les schémas de fichier Regex servent à manipuler les fichiers composés d'expressions régulières, notamment les fichiers log. Si vous devez souvent vous connecter à un fichier Regex, vous pouvez centraliser les informations de connexion et le schéma de ce fichier dans le **Repository** afin de les réutiliser facilement.

L'assistant [**New RegEx File**] regroupe la connexion au fichier et la définition du schéma en une seule procédure de quatre étapes.



Cette procédure nécessite une connaissance avancée de la syntaxe des expressions régulières.

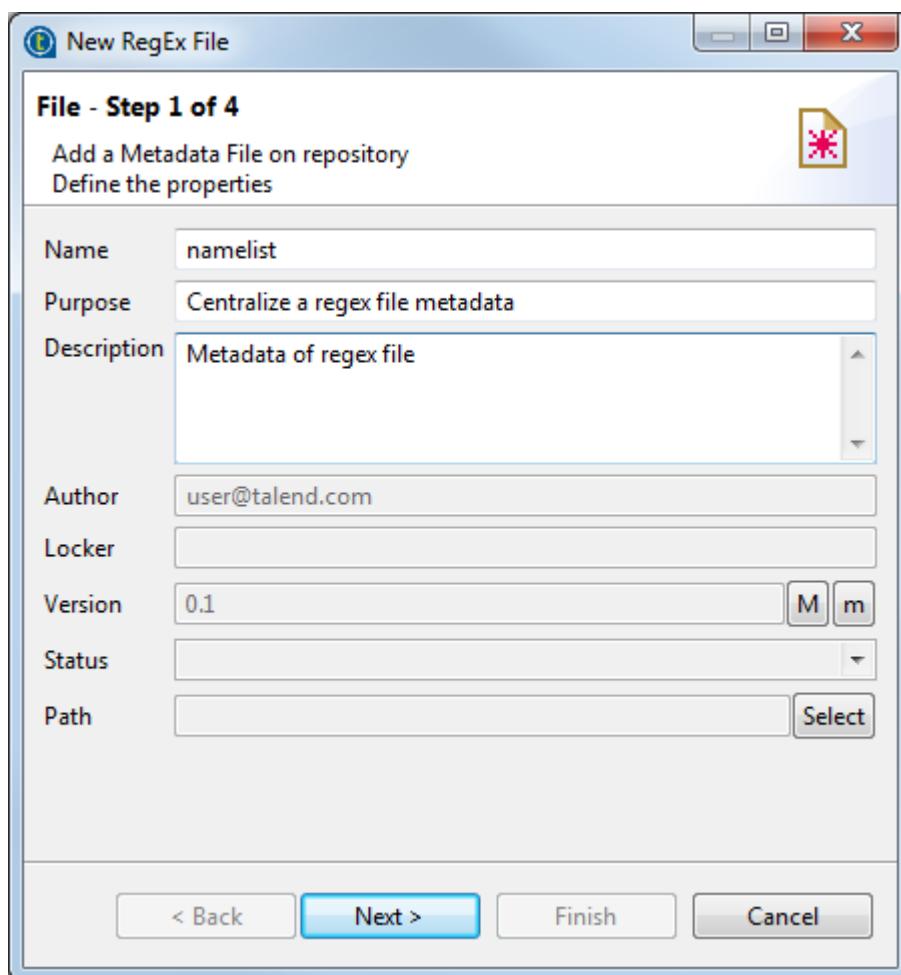
Pour créer une connexion à un fichier Regex à partir de zéro, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **File Regex** puis sélectionnez **Create file regex** dans le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser une connexion à un fichier que vous avez définie dans un Job ainsi que son schéma, cliquez sur l'icône  dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

Définissez ensuite les propriétés générales et le schéma du fichier dans l'assistant.

## Définir les propriétés générales

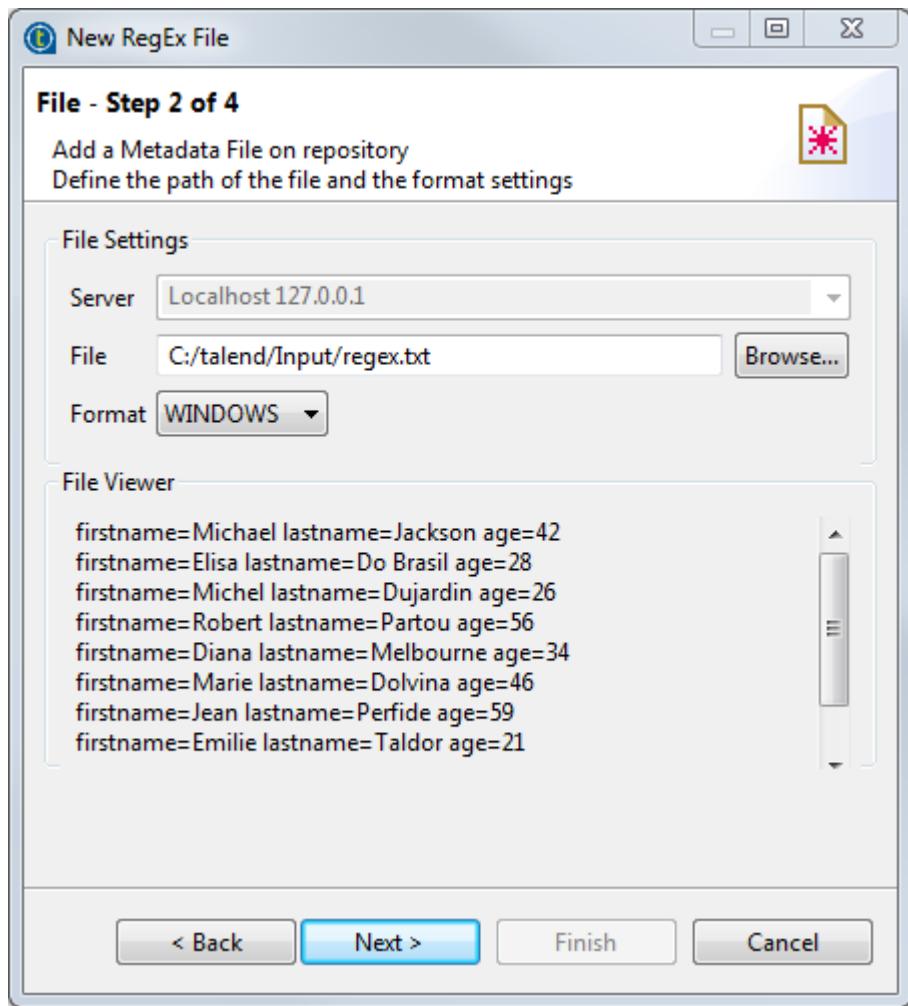
1. Dans l'assistant de configuration des métadonnées du fichier, renseignez le champ **Name**, qui est obligatoire et les champs **Purpose** et **Description** si vous le souhaitez. Les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File regex**. Ce dossier contient la connexion au fichier créée. Notez que vous ne pouvez pas sélectionner de dossier pendant que vous modifiez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.
4. Cliquez sur **Next** lorsque vous avez défini les propriétés générales.

## Configurer le chemin et le format du fichier

1. Cliquez sur le bouton **Browse...** pour rechercher votre fichier, dans votre hôte local ou un hôte LAN.
2. 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.



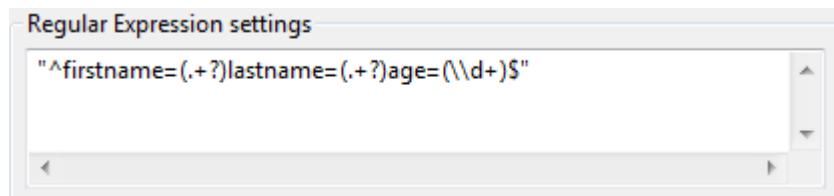
Le **File viewer** donne un instantané du fichier chargé.

3. Cliquez sur **Next** pour définir la structure du schéma.

## Définir les paramètres de parsing du fichier

Dans cette vue, vous pouvez affiner différents paramètres de votre fichier afin que le schéma soit correctement récupéré.

1. Définissez les séparateurs de champs et de lignes dans la zone **File Settings**.
  - Si nécessaire, vous pouvez modifier les chiffres dans le champ **Field Separator** afin de spécifier la longueur des colonnes de façon précise.
  - Si le séparateur de lignes de votre fichier n'est pas le caractère de fin de ligne standard, sélectionnez **Custom String** dans la liste **Row Separator** et spécifiez le caractère dans le champ **Corresponding Character**.
2. Dans le panneau de paramétrage des expressions régulières, **Regular Expression settings**, saisissez l'expression régulière utilisée pour délimiter le fichier.



Veuillez à utiliser les bons guillemets (simples ou doubles) au moment de rédiger le code.

3. Si votre fichier possède des en-têtes à exclure du contenu des données, cochez la case **Header** dans la zone **Rows To Skip** et définissez le nombre de lignes à ignorer dans le champ correspondant. De plus, si vous savez que le fichier contient des informations de pied de page, cochez la case **Footer** et définissez le nombre de lignes à ignorer.
4. La zone **Limit of Rows** vous permet de restreindre l'étendue du fichier à analyser. Si nécessaire, cochez la case **Limit** et définissez ou sélectionnez le nombre de lignes souhaité.
5. Si le fichier contient des libellés de colonne, cochez la case **Set heading row as column names** afin de transformer la première ligne en libellés de colonne. Notez que le nombre de lignes d'en-tête à ignorer est incrémenté de 1.
6. Cliquez ensuite sur **Refresh preview** pour que les modifications soient prises en compte. Le bouton change d'apparence en **Stop** jusqu'à ce que l'aperçu soit rafraîchi.

Preview			Output	
			<input type="checkbox"/> Set heading row as column names	Refresh Preview
<b>Column 0</b> <b>Column 1</b> <b>Column 2</b>				
Michael	Jackson	42		
Elisa	Do Brasil	28		
Michel	Dujardin	26		
Robert	Partou	56		
Diana	Melbourne	34		
Marie	Dolvina	46		

7. Cliquez sur **Next** afin de passer à l'étape suivante pour vérifier et personnaliser le schéma de fichier Regex généré.

## Vérifier et personnaliser le schéma du fichier

1. Renommez le schéma (par défaut, *metadata*) et modifiez les colonnes du schéma comme souhaité.  
Assurez-vous que le type de données dans la colonne **Type** est correctement défini.  
Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).  
Les types de données **Talend** les plus utilisés sont les suivants :
  - Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
  - List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.

- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
2. Pour restaurer ou mettre à jour le schéma du fichier Regex, cliquez sur **Guess**. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur **Guess**.
  3. Une fois vos changements effectués, cliquez sur **Finish** pour fermer l'assistant.

Le nouveau schéma apparaît dans l'arborescence du **Repository**, sous le nœud de la connexion **File regex** appropriée. Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file regex** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.8. Centraliser des métadonnées d'un fichier XML

Si vous devez souvent vous connecter à un fichier XML, vous pouvez utiliser l'assistant **[New Xml File]** pour centraliser les informations de connexion et le schéma de ce fichier dans le **Repository** afin de les réutiliser facilement.

Selon l'option sélectionnée, cet assistant vous permet de créer une connexion en lecture (Input) ou une connexion en écriture (Output). Dans un Job, les composants **tFileInputXML** et **tExtractXMLField** utilisent la connexion en lecture créée pour lire des fichiers XML. Dans un Job, le composant **tAdvancedFileOutputXML** utilise le schéma d'écriture créé pour écrire un fichier XML, s'il n'existe pas, ou y ajouter des informations, s'il existe déjà.

Si vous souhaitez lire un fichier XML, consultez [Configurer les métadonnées pour un fichier d'entrée \(Input\) XML](#).

Si vous souhaitez écrire un fichier XML, consultez [Configurer les métadonnées pour un fichier de sortie \(Output\) XML](#).

Pour créer une connexion à un fichier XML à partir de zéro, développez le nœud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **File XML** puis sélectionnez **Create file XML** dans le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser une connexion à un fichier que vous avez définie dans un Job ainsi que son schéma, cliquez sur l'icône  dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

Définissez ensuite les propriétés générales et le schéma du fichier dans l'assistant.

## 10.8.1. Configurer les métadonnées pour un fichier d'entrée (Input) XML

Cette section décrit la définition et le chargement d'une connexion à un fichier d'entrée (Input) XML. Pour définir et charger un fichier de sortie (Output), consultez [Configurer les métadonnées pour un fichier de sortie \(Output\) XML](#).

### Définir les propriétés générales

Dans cette étape, vous allez définir les propriétés générales des métadonnées, telles que le nom (champ **Name**), l'objectif (champ **Purpose**) et une **Description**.

1. Dans l'assistant de configuration des métadonnées du fichier, renseignez le champ **Name**, qui est obligatoire, et les champs **Purpose** et **Description** si vous le souhaitez. Les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



Lorsque vous saisissez les propriétés générales des métadonnées à créer, vous devez définir le type de connexion, entrée (input) ou sortie (output). Il est donc conseillé de saisir des informations susceptibles de vous aider à distinguer les schémas d'entrée et de sortie.

Field	Value
Name	ContactInfo_input
Purpose	Centralize contactInfo.xml metadata for reuse.
Description	Input connection and schema of contact information.
Author	user@talend.com
Locker	
Version	0.1
Status	
Path	

2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte

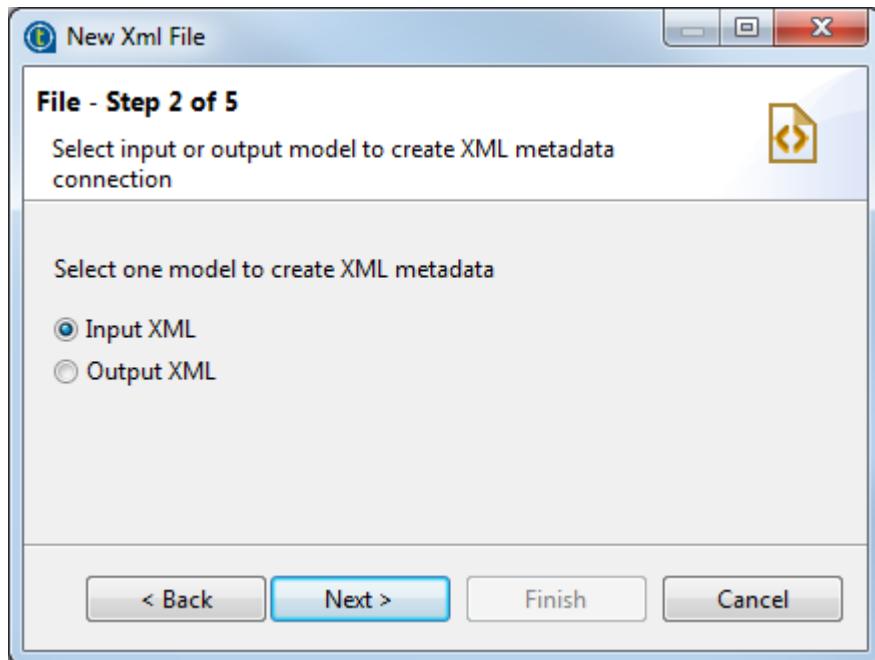
de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.

3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File XML**. Ce dossier contient la connexion au fichier juste créé. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous éditez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.
4. Cliquez sur **Next** pour sélectionner le type de métadonnées.

## Sélectionner le type de métadonnées (Input)

Dans cette étape, vous allez définir le type de métadonnées, 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.

## Charger un 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 (XML Schema Definition) sélectionné.

Le fichier XML d'entrée utilisé pour expliquer cette étape contient des informations de contact. Sa structure se présente comme suit :

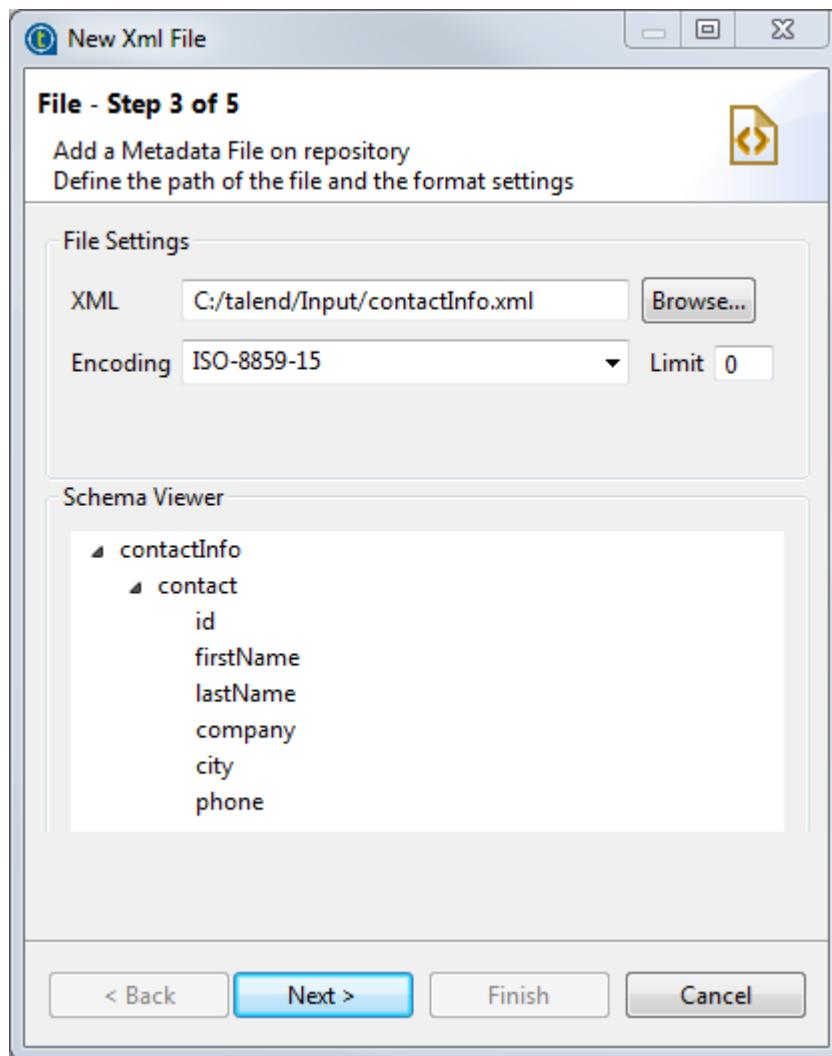
```
<contactInfo>
 <contact>
 <id>1</id>
 <firstName>Michael</firstName>
 <lastName>Jackson</lastName>
 <company>Talend</company>
 <city>Paris</city>
 <phone>2323</phone>
 </contact>
 <contact>
 <id>2</id>
 <firstName>Elisa</firstName>
```

```
<lastName>Black</lastName>
<company>Talend</company>
<city>Paris</city>
<phone>4499</phone>
</contact>
...
</contactInfo>
```

Pour charger un fichier XML, procédez comme suit :

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.

## Charger un fichier XSD

Cette procédure décrit comment charger un fichier XSD pour obtenir une structure XML. Pour charger un fichier XML, consultez [Charger un fichier Input](#).

Un fichier XSD est utilisé pour décrire le schéma des fichiers XML. La structure et les types d'éléments peuvent être décrits à l'aide du XSD suivant, utilisé comme XSD d'entrée d'exemple dans cette section.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xs:element name="contactInfo">
 <xs:complexType>
 <xs:sequence>
 <xs:element maxOccurs="unbounded" ref="contact"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="contact">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="id"/>
 <xs:element ref="firstName"/>
 <xs:element ref="lastName"/>
 <xs:element ref="company"/>
 <xs:element ref="city"/>
 <xs:element ref="phone"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="id" type="xs:integer"/>
 <xs:element name="firstName" type="xs:NCName"/>
 <xs:element name="lastName" type="xs:NCName"/>
 <xs:element name="company" type="xs:NCName"/>
 <xs:element name="city" type="xs:NCName"/>
 <xs:element name="phone" type="xs:integer"/>
</xs:schema>
```

Pour plus d'informations concernant les schémas XML, consultez <http://www.w3.org/XML/Schema> (en anglais).

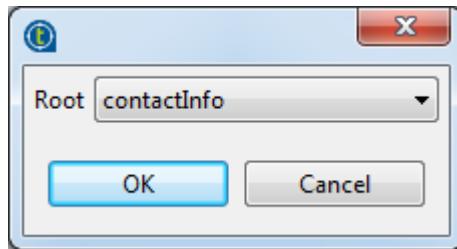


Lorsque vous chargez un fichier XSD,

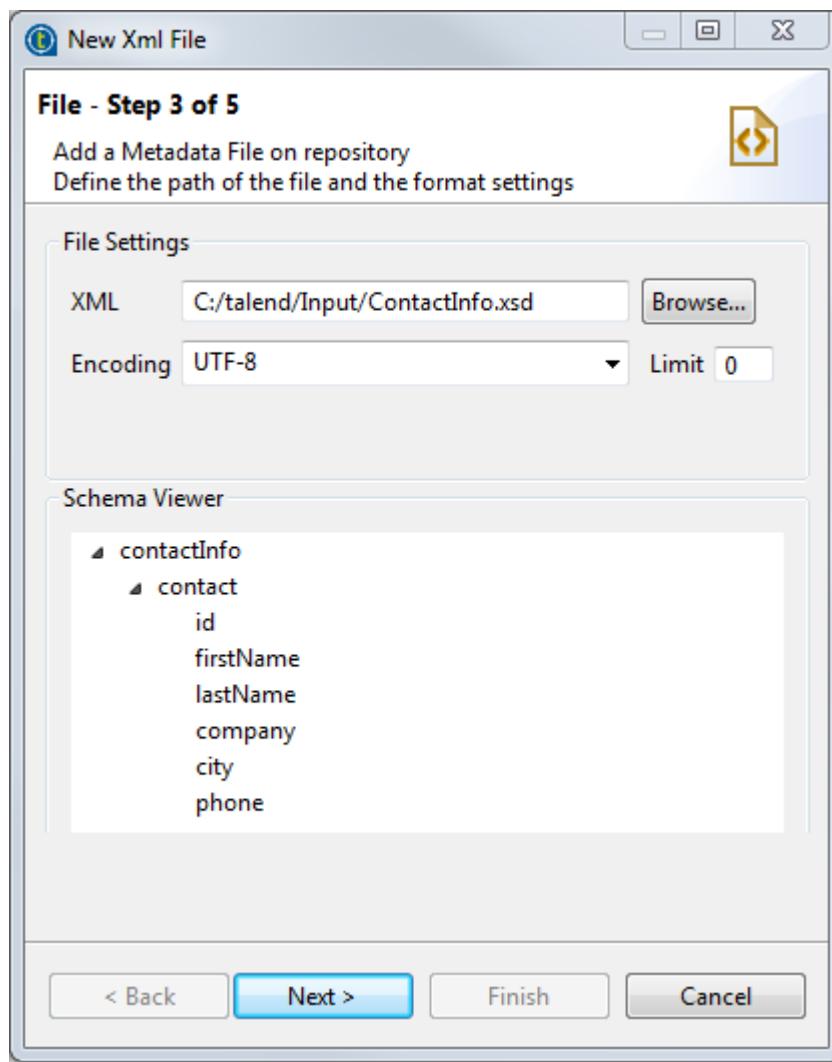
- les données sont sauvegardées dans le **Repository**. Les métadonnées ne seront pas affectées par une suppression ou un déplacement du fichier.
- vous pouvez choisir un élément en tant que racine de votre arborescence XML.

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**.



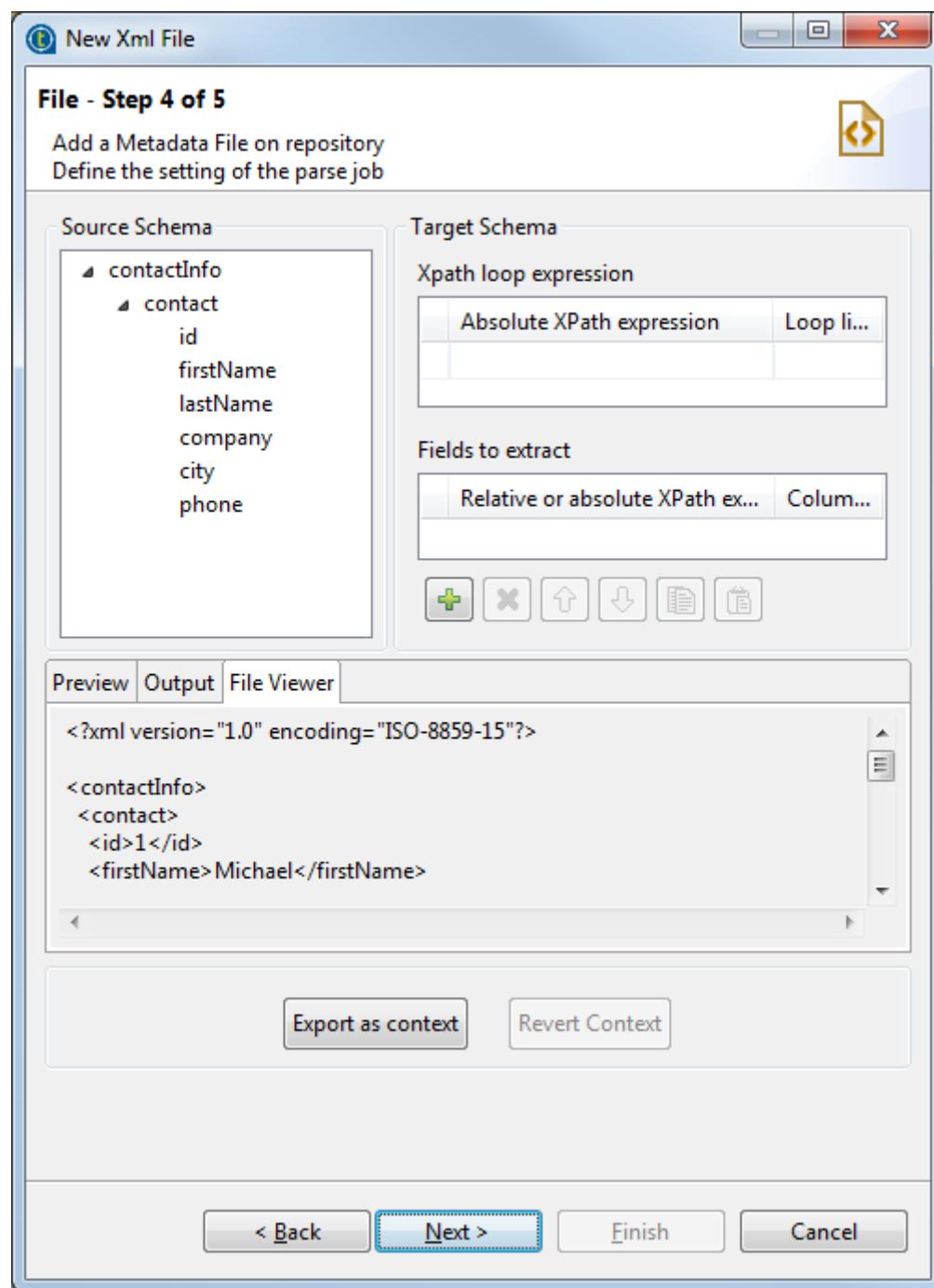
La zone **Schema Viewer** affiche un aperçu de la structure XML. Vous pouvez développer et visualiser chaque niveau de la structure XML du fichier.



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.

## Définir un 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 :

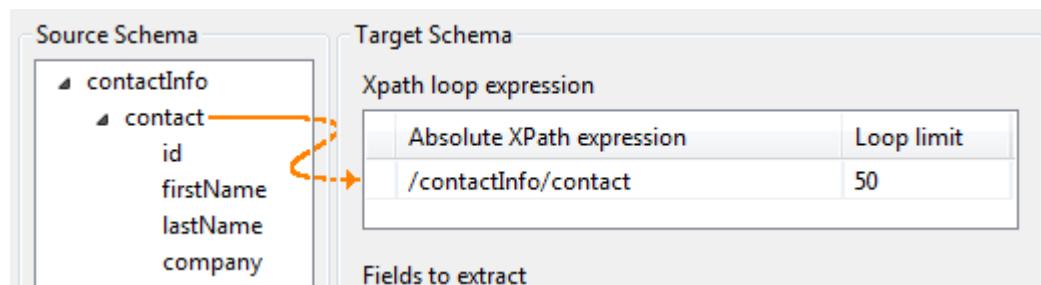
Vue	Description
<b>Source Schema</b>	Arborescence du fichier XML.
<b>Target Schema</b>	Informations d'extraction et d'itération.
<b>Preview</b>	Aperçu du schéma cible, ainsi que des données d'entrée des colonnes sélectionnées, affichées dans l'ordre défini.  💡 La fonctionnalité d'aperçu n'est pas disponible si vous chargez un fichier XSD.
<b>File viewer</b>	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 :

- 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.
- Déposez le nœud sélectionné de l'arborescence **Source Schema** dans le champ **Absolute XPath expression**.

Une flèche orange relie le noeud à l'expression correspondante.

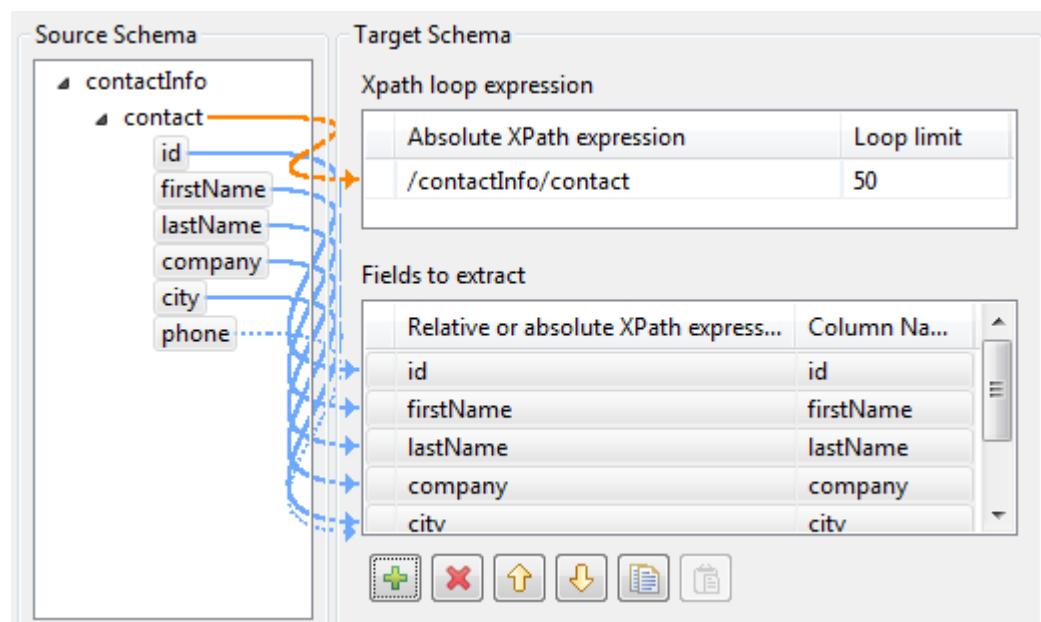


Le champ **Xpath loop expression** est obligatoire.

- Saisissez une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de nœuds à itérer ou saisissez **-1** si vous souhaitez l'effectuer sur toutes les colonnes.
- Définissez les champs à extraire en déposant 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.



- 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 **[x]**.

- Modifiez l'ordre des colonnes à l'aide des boutons et .
5. Dans le champ **Column name**, nommez les libellés des colonnes qui s'afficheront dans la vue d'aperçu du schéma (**Schema preview**).
  6. Cliquez sur **Refresh preview** pour afficher l'aperçu du schéma. Les champs apparaissent ainsi dans l'aperçu du schéma dans l'ordre imposé.



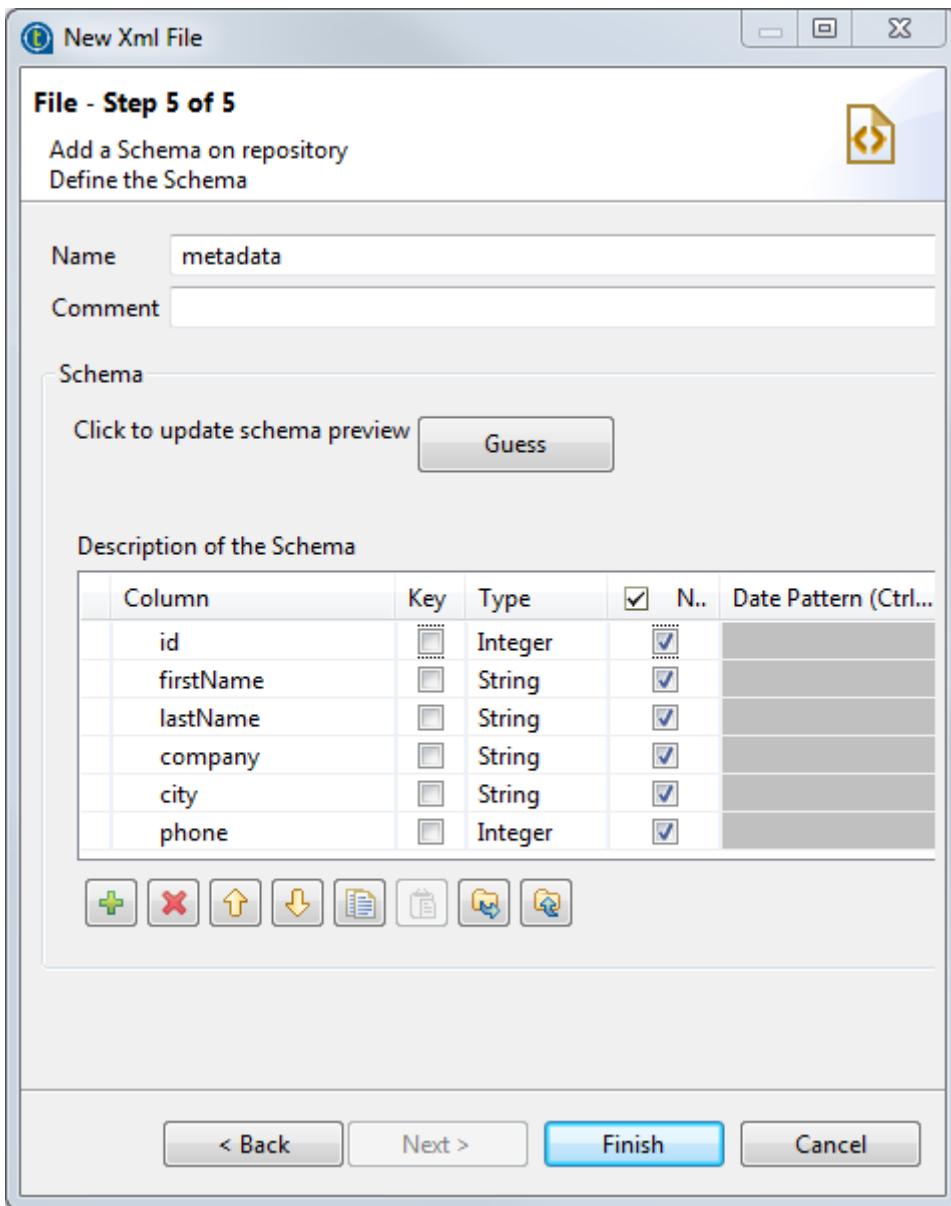
L'aperçu n'est pas disponible si vous chargez un fichier XSD.

	id	firstName	lastName	company	city	phone
1	Michael	Jackson	Talend	Paris	2323	
2	Elisa	Black	Talend	Paris	4499	
3	Michael	Dujardin	Talend	Paris	8872	
4	Marie	Dolvina	Talend	Paris	6655	
5	Jean	Perfide	Talend	Paris	3344	
6	Emilie	Taldor	Talend	Paris	2266	
7	Anne-Laure	Paldufier	Talend	Paris	4422	

7. Cliquez sur **Next** pour vérifier et éditer le schéma final.

## Finaliser le schéma du fichier

Le schéma généré affiche les colonnes sélectionnées dans le fichier XML et vous permet de personnaliser ce schéma.



- Si nécessaire, renommez la métadonnée dans le champ **Name**, (*metadata*, par défaut), ajoutez un commentaire dans le champ **Comment** et procédez à d'autres modifications, par exemple :

- Définissez les colonnes en éditant les champs correspondants.
- Ajoutez ou supprimez une colonne à l'aide des boutons [+] et [x].
- Modifiez l'ordre des colonnes à l'aide des boutons et .

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

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

- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
  - Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
2. Si le fichier XML sur lequel se base le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma.
  3. Cliquez sur **Finish**. La nouvelle connexion au fichier, ainsi que son schéma, s'affichent sous le nœud **File XML** du **Repository**.

Vous pouvez à présent déposer n'importe quelle connexion à un fichier ou n'importe quel schéma depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant **tFileDialogXML** ou **tExtractXMLField**. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file xml** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.8.2. Configurer les métadonnées pour un fichier de sortie (Output) XML

Cette section décrit la définition et le chargement d'une connexion à un fichier de sortie (Output) XML. Pour définir et charger un schéma XML d'entrée (Input), consultez [Configurer les métadonnées pour un fichier d'entrée \(Input\) XML](#).

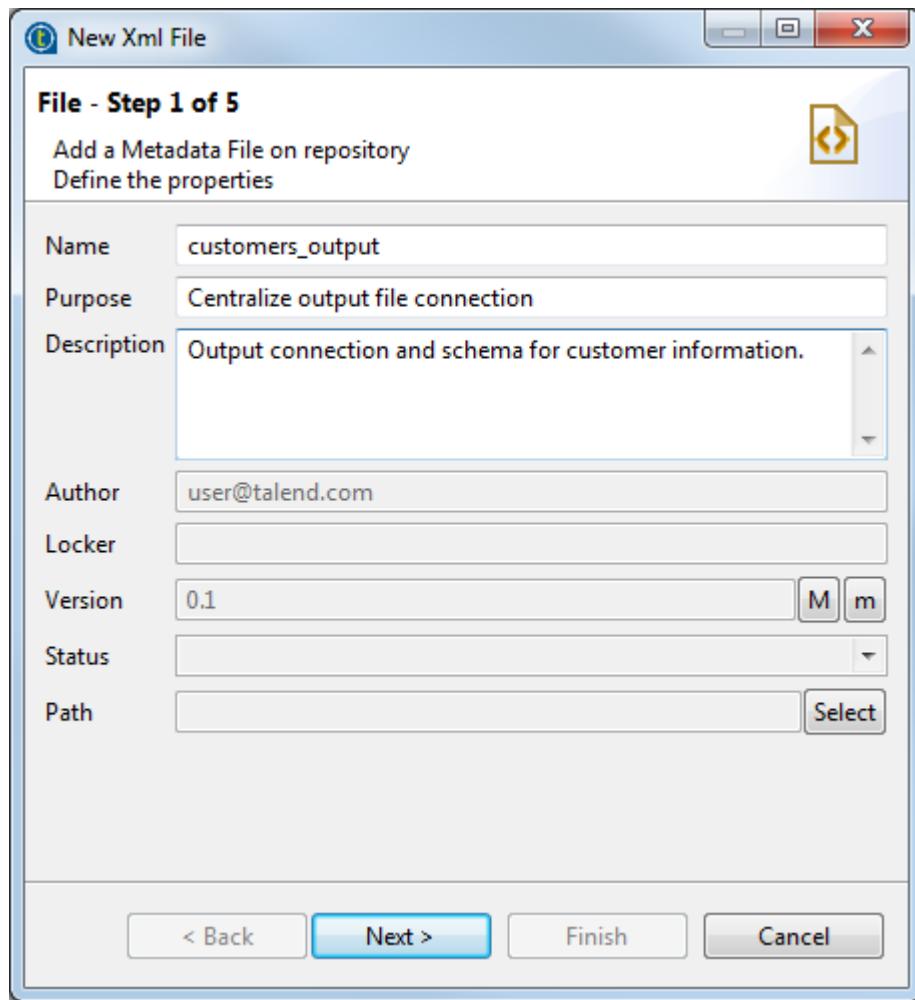
### Définir les propriétés générales

Dans cette étape, vous allez définir les propriétés générales des métadonnées, telles que le nom (champ **Name**), l'objectif (champ **Purpose**) et une **Description**.

1. Dans l'assistant de configuration des métadonnées du fichier, renseignez le champ **Name**, qui est obligatoire, et les champs **Purpose** et **Description** si vous le souhaitez. Les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



Lorsque vous saisissez les propriétés générales des métadonnées à créer, vous devez définir le type de connexion, entrée (input) ou sortie (output). Il est donc conseillé de saisir des informations susceptibles de vous aider à distinguer les schémas d'entrée et de sortie.

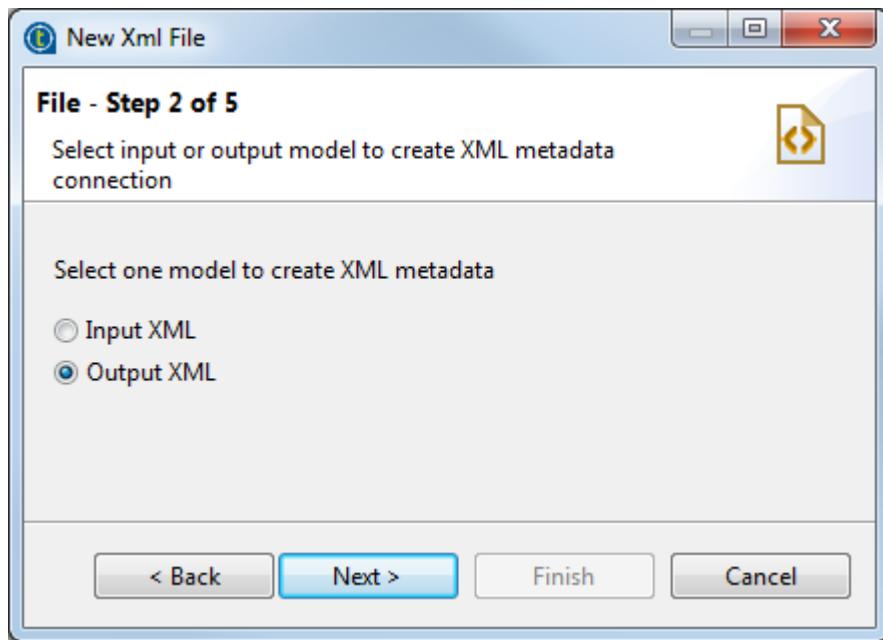


2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
3. Si nécessaire également, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File XML**. Ce dossier contient la connexion au fichier juste créé. Notez que vous ne pouvez pas sélectionnez de dossier pendant que vous éditez une connexion existante. Vous pouvez cependant déposer une connexion dans un nouveau dossier quand vous le souhaitez.
4. Cliquez sur **Next** pour sélectionner le type de métadonnées.

### Sélectionner le type de métadonnées (Output)

Dans cette étape, vous allez définir le type de métadonnées, 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, à partir d'un fichier XML, XSD ou à partir de rien.

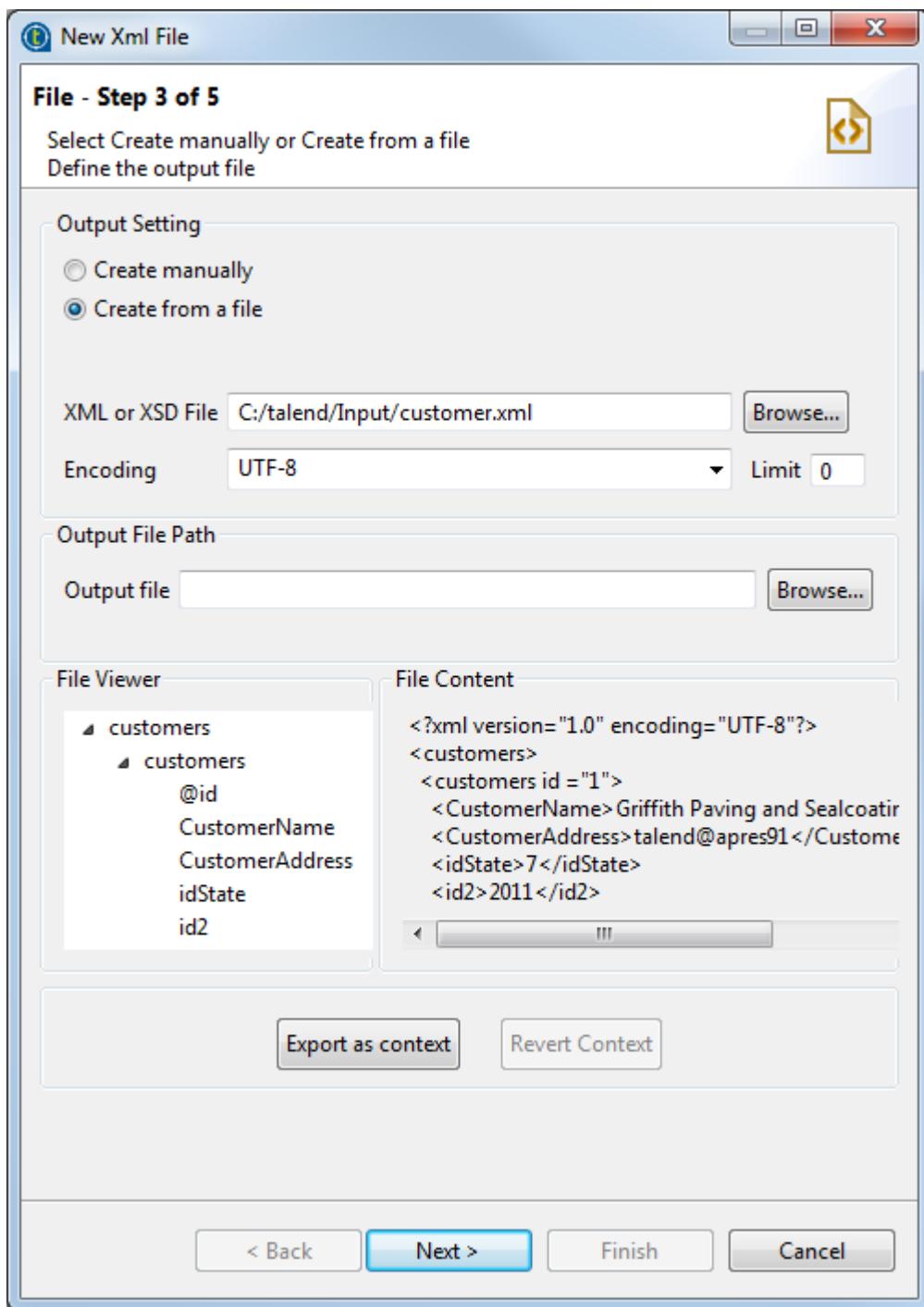
### Définir la structure du fichier de sortie à l'aide d'un fichier XML existant

Vous pouvez choisir de créer votre fichier manuellement ou de le créer à partir d'un fichier existant. Vous pouvez choisir de le créer manuellement (en sélectionnant **Create manually**), mais vous devrez alors configurer vous-même votre schéma, vos colonnes sources et vos colonnes cibles à l'étape 4 de l'assistant. Le fichier est créé lors de l'utilisation dans un Job d'un composant de sortie, comme le **tAdvancedFileOutputXML**.

Dans cet exemple, créez le fichier en chargeant un fichier existant XML ou XSD.

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.

## Définir la structure du fichier de sortie à l'aide d'un fichier XSD

Cette procédure décrit comment définir la structure d'un fichier XML de sortie à partir d'un fichier XSD. Pour définir la structure XML à partir d'un fichier XML, consultez [Définir la structure du fichier de sortie à l'aide d'un fichier XML existant](#).



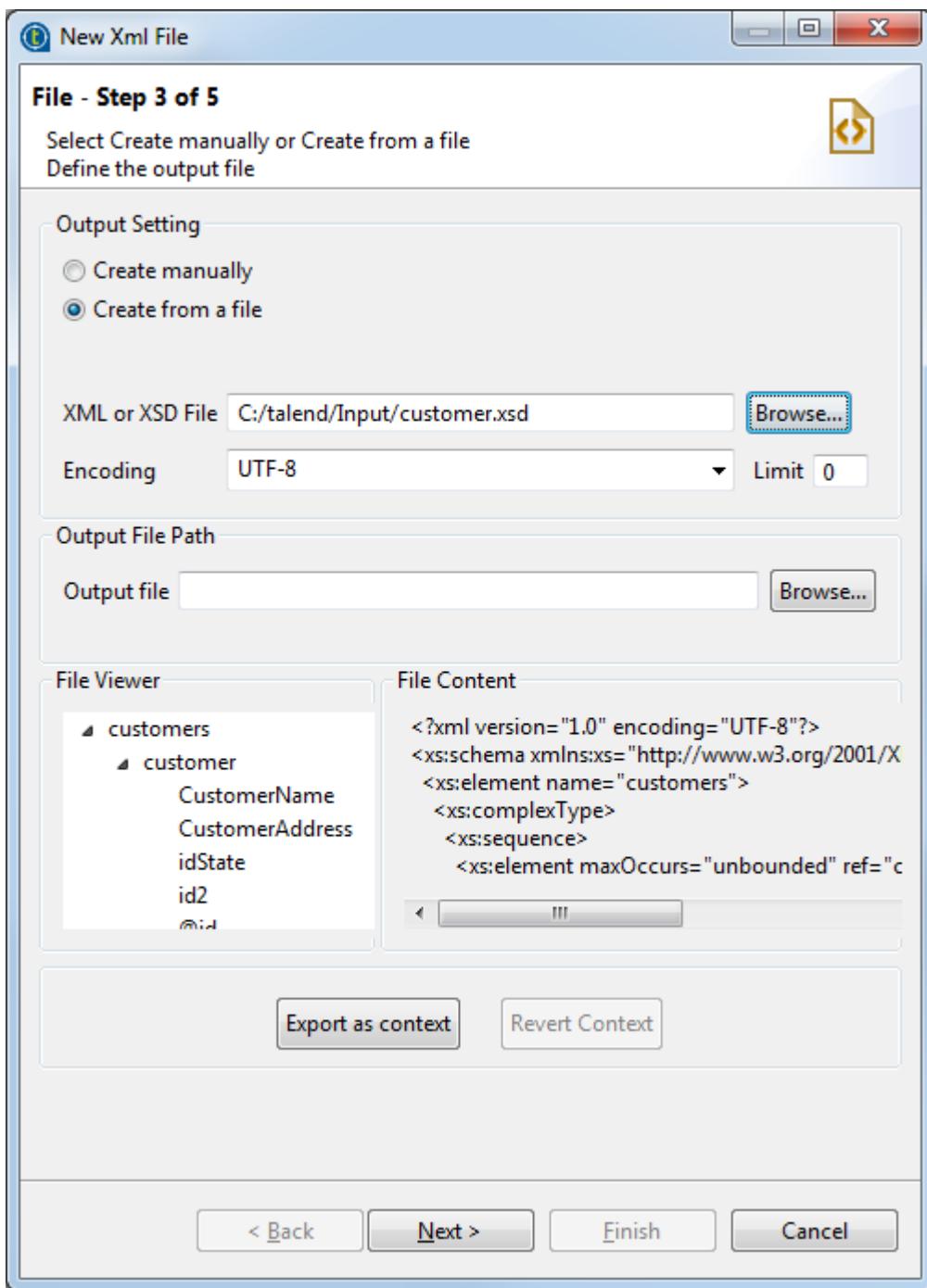
Lorsque vous chargez un fichier XSD,

- les données sont sauvegardées dans le **Repository** et la métadonnées n'est pas affectée par la suppression ou le déplacement d'un fichier.
- vous pouvez choisir un élément comme racine de votre arborescence XML.

Pour créer la structure XML de sortie à partir d'un fichier XSD, procédez comme suit :

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**.

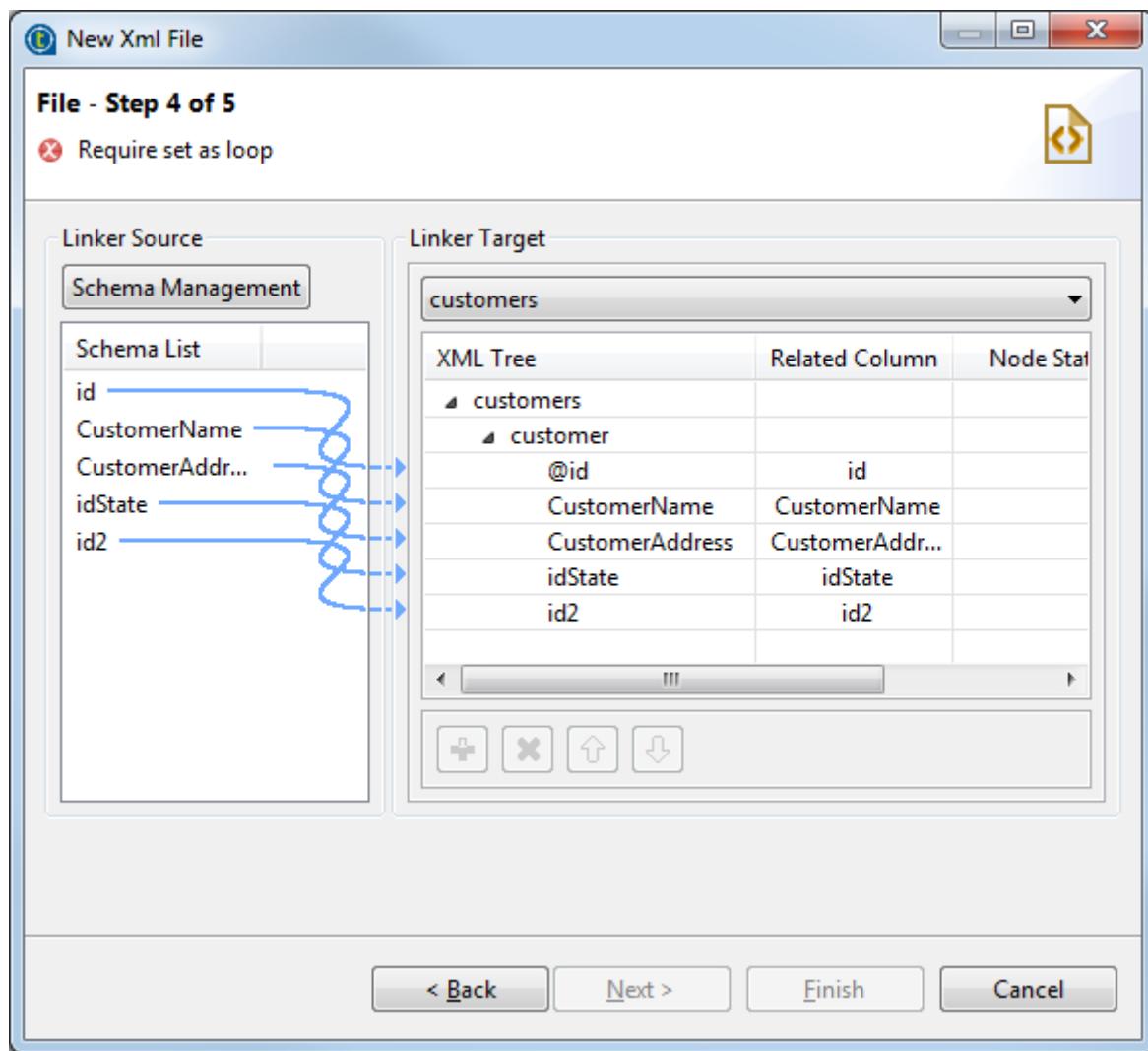
La zone **File Viewer** affiche l'aperçu de la structure XML et la zone **File Content** affiche au maximum les cinquante premières lignes du fichier.



- Renseignez le champ **Encoding** si le système ne l'a pas détecté automatiquement.
- Dans le champ **Limit**, définissez le nombre de colonnes sur lesquelles effectuer la requête XPath, ou saisissez *0* si vous souhaitez l'effectuer sur toutes les colonnes.
- Dans la zone **Output File Path**, dans le champ **Output file**, renseignez le chemin d'accès à votre fichier de sortie. Si le fichier n'existe pas, il sera créé, lors de l'utilisation, dans un Job, du composant **tAdvancedFileOutputXML**. Si le fichier existe déjà, il sera écrasé.
- Cliquez sur **Next** pour définir le schéma.

## 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.



Dans cette étape, configuez le schéma de sortie. Le tableau suivant décrit comment faire :

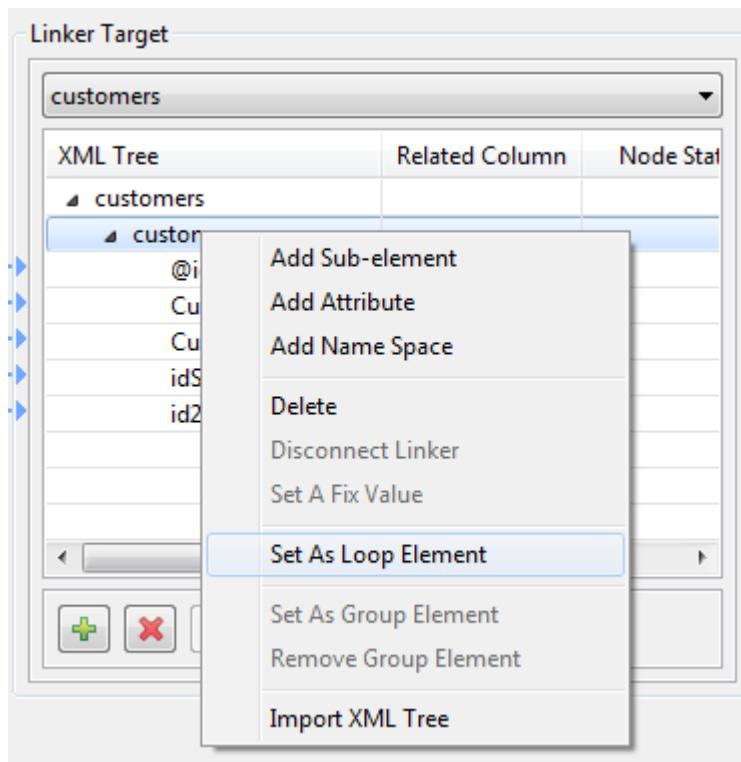
Pour...	Effectuer...
Créer un schéma à partir de zéro ou éditer les colonnes du schéma source à passer au schéma cible	Dans la zone <b>Linker Source</b> , cliquez sur le bouton <b>Schema Management</b> afin d'ouvrir l'éditeur de schéma.
Définir un élément de boucle	Dans la zone <b>Linker Target</b> , cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Set As Loop Element</b> dans le menu contextuel.  Cette opération est obligatoire pour définir un élément sur lequel effectuer une boucle.
Définir un <b>group element</b>	Dans la zone <b>Linker Target</b> , cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Set As Group Element</b> dans le menu contextuel.  Vous pouvez configurer l'élément parent de l'élément de boucle en tant que <b>group element</b> à condition que l'élément parent ne soit pas la racine de l'arborescence XML.
Créer un élément fils pour un élément	Dans la zone <b>Linker Target</b> , <ul style="list-style-type: none"> <li>Cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Add Sub-element</b> dans le menu contextuel, saisissez un nom pour le sous-élément dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>.</li> <li>Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton <b>[+]</b> en bas, sélectionnez <b>Create as sub-element</b> dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>. Ensuite, saisissez un nom pour le sous-élément dans la boîte de dialogue suivante et cliquez sur <b>OK</b>.</li> </ul>

Pour...	Effectuer...
Créer un attribut pour un élément	<p>Dans la zone <b>Linker Target</b>,</p> <ul style="list-style-type: none"> <li>Cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Add Attribute</b> dans le menu contextuel, saisissez un nom pour l'attribut dans la boîte de dialogue qui apparaît et cliquez sur <b>OK</b>.</li> <li>Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton <b>[+]</b> en bas, sélectionnez <b>Create as attribute</b> dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>. Ensuite, saisissez un nom pour l'attribut dans la boîte de dialogue suivante et cliquez sur <b>OK</b>.</li> </ul>
Créer un espace de nommage pour un élément	<p>Dans la zone <b>Linker Target</b>,</p> <ul style="list-style-type: none"> <li>Cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Add Name Space</b> dans le menu contextuel, saisissez un nom pour l'espace de nommage dans la boîte de dialogue qui apparaît et cliquez sur <b>OK</b>.</li> <li>Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton <b>[+]</b> en bas, sélectionnez <b>Create as name space</b> dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>. Ensuite, saisissez un nom pour l'espace de nommage dans la boîte de dialogue suivante et cliquez sur <b>OK</b>.</li> </ul>
Supprimer un ou plusieurs éléments/attributs/espaces de nommage	<p>Dans la zone <b>Linker Target</b>,</p> <ul style="list-style-type: none"> <li>Cliquez-droit sur les éléments/attributs/espaces de nommage qui vous intéressent et sélectionnez <b>Delete</b> dans le menu contextuel.</li> <li>Sélectionnez les éléments/attributs/espaces de nommage qui vous intéressent et cliquez sur le bouton <b>[x]</b> en bas.</li> <li>Sélectionnez les éléments/attributs/espaces de nommage qui vous intéressent et appuyez sur la touche <b>Suppr</b>.</li> </ul> <p> Supprimer un élément supprime également ses enfants, s'il en a.</p>
Ajuster l'ordre d'un ou plusieurs élément(s)	<p>Dans la zone <b>Linker Target</b>, sélectionnez l'élément qui vous intéressent et cliquez sur les boutons  et </p> <ul style="list-style-type: none"> <li>La valeur que vous avez configurée va remplacer toute valeur récupérée du flux d'entrée dans votre Job.</li> <li>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.</li> </ul>
Créer un mapping source-cible	<p>Sélectionnez la colonne qui vous intéresse, dans la zone <b>Linker Source</b>, déposez-la sur le noeud qui vous intéresse de la zone <b>Linker Target</b> et sélectionnez <b>Create as sub-element of target node</b>, <b>Create as attribute of target node</b>, ou <b>Add linker to target node</b> selon vos besoins, dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>.</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	<p>Dans la zone <b>Linker Target</b>, cliquez-droit sur le noeud qui vous intéresse et sélectionnez <b>Disconnect Linker</b> dans le menu contextuel.</p>
Créer une arborescence XML à partir d'un autre fichier XML ou XSD	<p>Cliquez-droit sur tout élément du schéma dans la zone <b>Linker Target</b> et sélectionnez <b>Import XML Tree</b> 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.</p>



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.

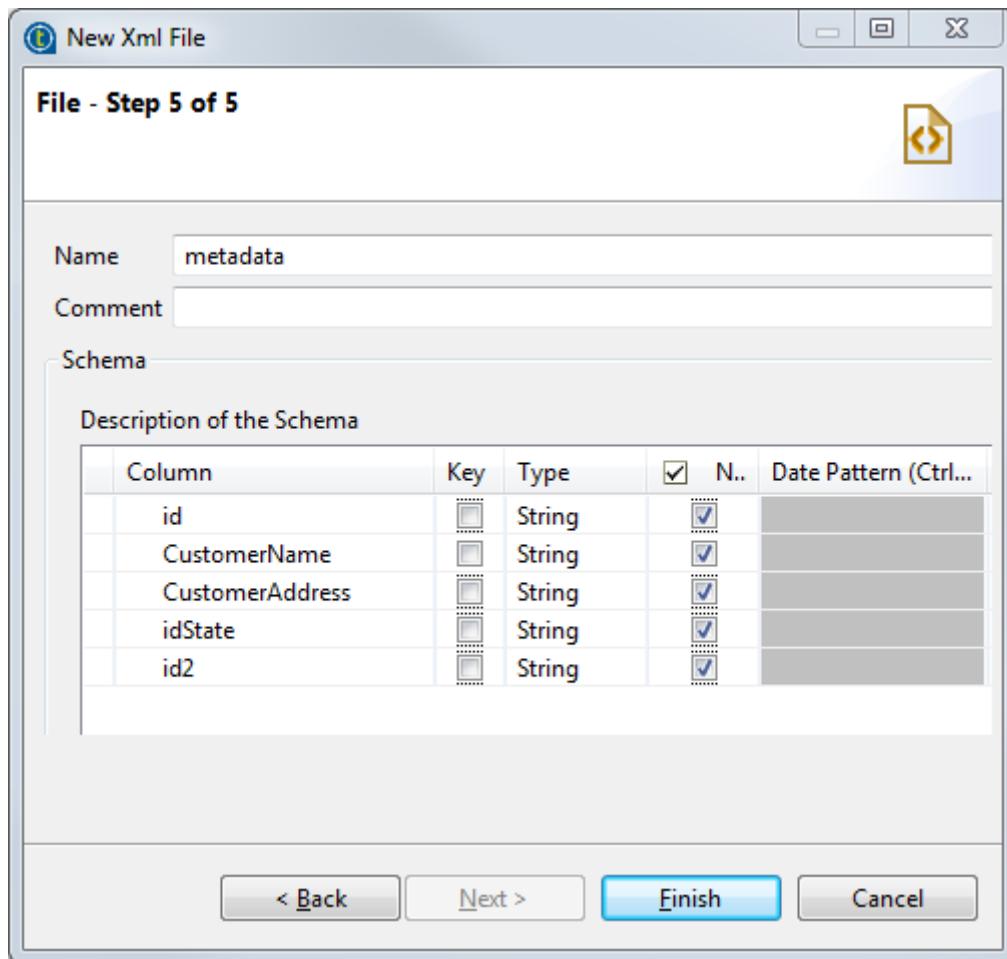
1. Dans la zone **Linker Target**, cliquez-droit sur l'élément sur lequel vous souhaitez effectuer une boucle, puis, dans le menu contextuel, sélectionnez **Set As Loop Element**.



2. Définissez les autres propriétés de fichier Output, puis cliquez sur **Next** pour passer à l'étape suivante.

### Finaliser le schéma

L'étape 5 de l'assistant affiche le schéma final généré et vous permet de le modifier.



- Vous pouvez personnaliser le nom de la métadonnée dans le champ **Name** (par défaut *metadata*), ajouter un commentaire dans le champ **Comment** et apporter d'autres modifications, par exemple :

- Si nécessaire, définissez à nouveau les colonnes en éditant les champs appropriés.
- Ajoutez ou supprimez une colonne à l'aide des boutons **[+]** et **[x]**.
- Modifiez l'ordre des colonnes à l'aide des boutons et .

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

2. Si le fichier XML sur lequel se base le schéma a été modifié, cliquez sur le bouton **Guessafin** de générer le schéma à nouveau. Notez que si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.
3. 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.

Vous pouvez à présent déposer n'importe quelle connexion à un fichier ou n'importe quel schéma depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant **tAdvancedFileOutputXML**. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file xml** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.9. Centraliser des métadonnées d'un fichier Excel

Si vous devez souvent écrire des données dans et/ou lire des données depuis une feuille de calcul Excel, vous pouvez centraliser les informations de connexion et le schéma de ce fichier dans le **Repository** afin de les réutiliser facilement. Ainsi, vous n'aurez pas à définir ces métadonnées manuellement pour chaque composant à chaque fois que vous souhaitez utiliser ce fichier.

Vous pouvez centraliser une connexion à un fichier Excel depuis un fichier Excel existant ou depuis les propriétés de ce fichier définies dans un Job.

Pour centraliser une connexion ainsi que le schéma d'un fichier Excel, développez le nœud **Metadata** du **Repository**, cliquez-droit sur **File Excel** et sélectionnez **Create file Excel** depuis le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser un schéma ainsi qu'une connexion à un fichier que vous avez définis dans un Job, cliquez sur l'icône  dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

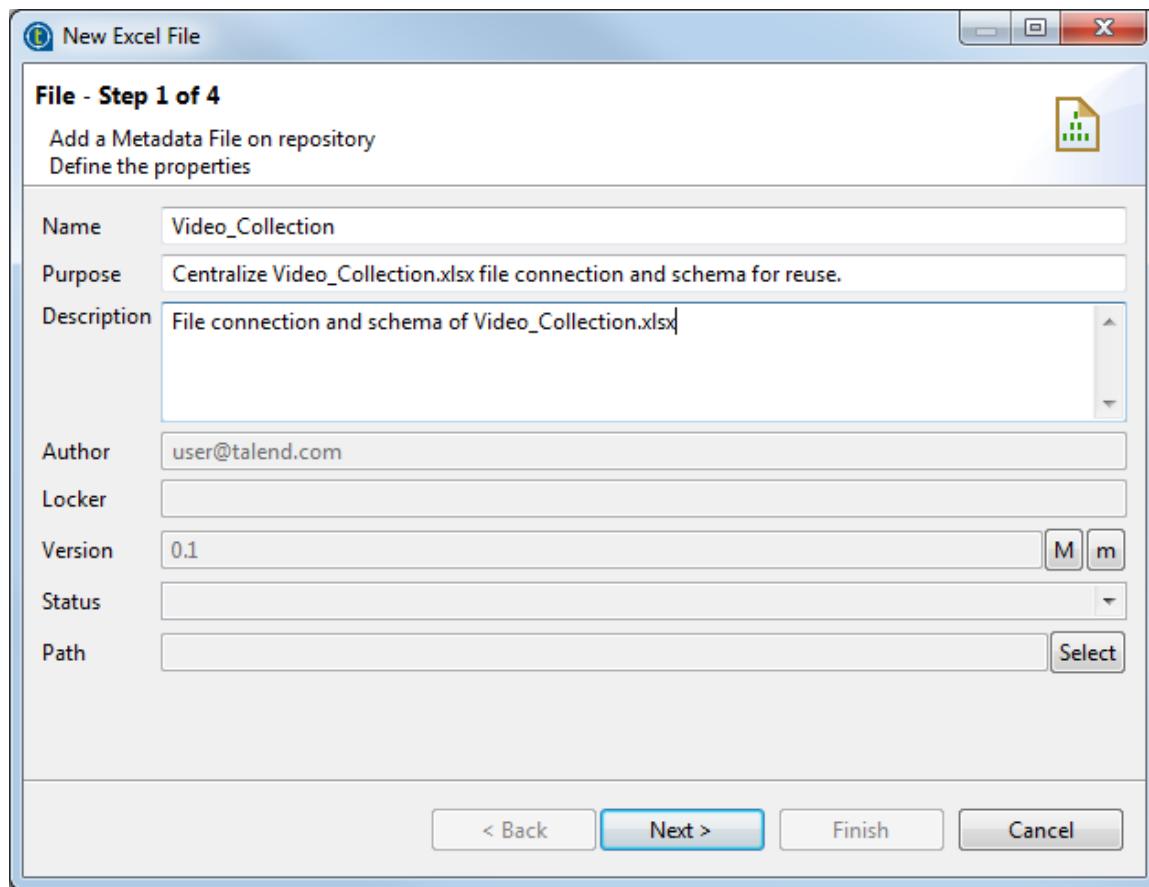
Suivez ensuite les étapes suivantes dans l'assistant afin de :

- Définir les informations qui identifient la connexion au fichier. Pour plus d'informations, consultez [Définir les propriétés générales](#).
- Charger le fichier. Pour plus d'informations, consultez [Charger le fichier](#).
- Analyser le fichier afin de récupérer son schéma. Pour plus d'informations, consultez [Analyser le fichier](#).
- Finaliser le schéma du fichier. Pour plus d'informations, consultez [Finaliser le schéma](#).

### Définir les propriétés générales

1. Dans l'assistant de configuration des métadonnées, renseignez le champ **Name**, qui est obligatoire, et les champs **Purpose** et **Description** si vous le souhaitez. Les informations que vous renseignez dans le

Le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.

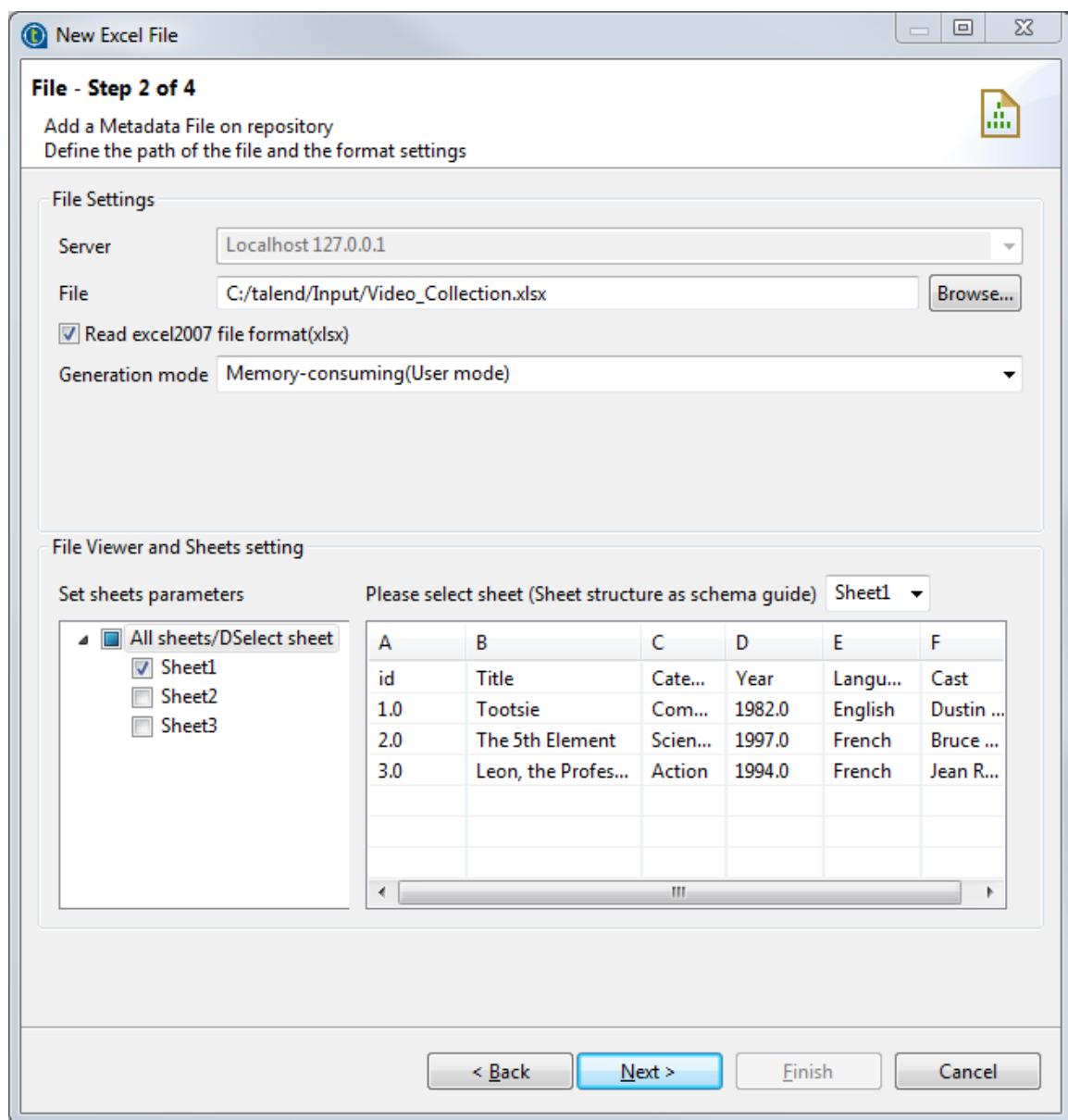


2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File Excel**. Ce dossier contient la connexion au fichier juste créée.
4. Cliquez sur **Next** pour passer aux paramètres du fichier.

## Charger le fichier

1. Cliquez sur le bouton **Browse...** afin de parcourir votre système vers le fichier ou saisissez son chemin dans le champ **File**.

Passez cette étape si vous sauvegardez la connexion à un fichier Excel définie dans un composant car son chemin est déjà défini dans le champ **File**.



2. Si le fichier chargé provient de Excel 2007, assurez-vous que la case **Read excel2007 file format(xlsx)** soit cochée.
3. Par défaut, le mode **Memory-consuming (User mode)** est sélectionné. Cependant, si le fichier .xlsx chargé est volumineux, sélectionnez **Less memory consumed for large excel(Event mode)** dans la liste **Generation mode** pour éviter les erreurs de mémoire.
4. Dans la zone **File viewer and sheets settings**, sélectionnez la (les) feuille(s) que vous souhaitez utiliser.
  - Dans la liste **Please select sheet**, sélectionnez la feuille dont vous souhaitez avoir un aperçu. La table d'aperçu affiche le contenu de la feuille sélectionnée.

Par défaut, la table affiche la première feuille du fichier.

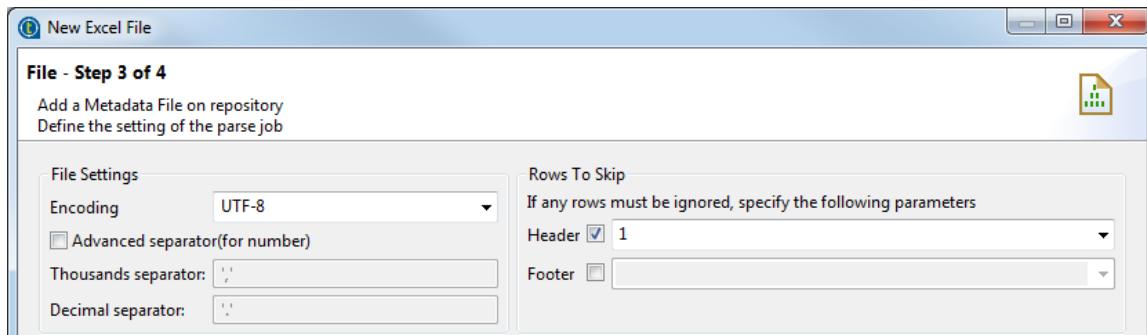
  - Dans la liste **Set sheets parameters**, cochez la case à côté de la (les) feuille(s) que vous souhaitez charger.

Si vous sélectionnez plus d'une feuille, le schéma final sera une combinaison des structures de toutes les feuilles sélectionnées.
5. Cliquez sur **Next** pour continuer.

## Analyser le fichier

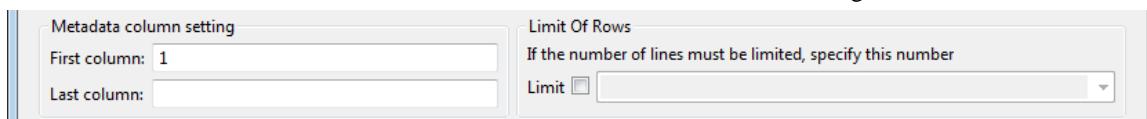
Dans cette étape, vous pouvez définir les différents paramètres de votre fichier afin que le schéma soit correctement récupéré.

1. Selon votre fichier Excel, spécifiez l'encodage (dans le champ **Encoding**), le séparateur avancé pour les nombres (dans le champ **Advanced separator (for number)**) ainsi que le nombre de lignes à ignorer en en-tête (**Header**) ou en pied de page (**Footer**).



2. Si nécessaire, vous pouvez définir précisément le nombre de colonnes à lire à l'aide des champs **First column** et **Last column**. Par exemple, si vous souhaitez ignorer la première colonne car elle ne contient pas de données à utiliser, saisissez 2 dans le champ **First column** afin de définir la deuxième colonne du schéma comme étant la première.

Afin de récupérer le schéma d'un fichier Excel, vous n'avez pas besoin d'analyser toutes les lignes du fichier, en particulier lorsque le fichier à charger est volumineux. Afin de limiter le nombre de lignes à analyser, cochez la case **Limit** dans la zone **Limit Of Rows** et définissez le nombre de lignes désirées.



3. Si le fichier Excel a une ligne d'en-tête, cochez la case **Set heading row as column names** afin de prendre en compte le nom des en-têtes. Cliquez sur **Refresh** afin de voir le résultat des changements précédents dans la table d'aperçu.

1	Tootsie	Comedy	1982	English	Dustin Hoffman, Jessica Lange, Sydney
2	The 5th Element	Science fiction	1997	French	Bruce Willis, Gary Oldman, Milla Jovovitch
3	Leon, the Professor	Action drama	1994	French	Jean Reno, Gary Oldman, Nathalie Portman

4. Cliquez sur **Next** pour continuer.

## Finaliser le schéma

La dernière étape de l'assistant affiche le schéma final généré et vous permet de personnaliser le schéma selon vos besoins.

Remarquez que tous les caractères du fichier pouvant être mal interprétés par le programme par la suite sont remplacés par un caractère neutre. Par exemple, les tirets bas remplacent les astérisques.

Column	Key	Type	N.	Date Pattern (Ctrl...)	Length	Precision	Default
id	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		1	0	
Title	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		22	0	
Category	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		15	0	
Year	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		4	0	
Language	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		7	0	
Cast	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		42	0	

- Si besoin, renommez le schéma (par défaut *metadata*) et laissez un commentaire.

Vous pouvez également personnaliser le schéma selon vos besoins : vous pouvez ajouter, supprimer ou déplacer les colonnes du schéma, exporter le schéma dans un fichier XML ou remplacer le schéma en important une définition de schéma depuis un fichier XML à l'aide de la barre d'outils.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.

- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
2. Si le fichier Excel sur lequel est basé le schéma a changé, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur **Guess**.
  3. Cliquez sur **Finish**. Le nouveau schéma apparaît dans le **Repository**, sous le nœud **File Excel** correspondant.

Vous pouvez à présent déposer la connexion au fichier, ou son schéma, depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file Excel** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.10. Centraliser les métadonnées d'un fichier LDIF

Les fichiers LDIF sont des fichiers annuaires distincts par leurs attributs. Si vous devez souvent lire un fichier LDIF, vous pouvez centraliser les informations de connexion et les attributs de ce fichier dans le **Repository** afin de les réutiliser facilement. Ainsi, vous n'aurez pas à définir ces métadonnées manuellement pour chaque composant à chaque fois que vous souhaitez utiliser ce fichier.

Vous pouvez centraliser une connexion à un fichier LDIF depuis un fichier LDIF existant ou depuis les propriétés de ce fichier définies dans un Job.

Pour centraliser une connexion ainsi que le schéma d'un fichier LDIF, développez le nœud **Metadata** du **Repository**, cliquez-droit sur **File ldif** et sélectionnez **Create file ldif** depuis le menu contextuel afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour centraliser un schéma ainsi qu'une connexion à un fichier que vous avez définis dans un Job, cliquez sur l'icône  dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

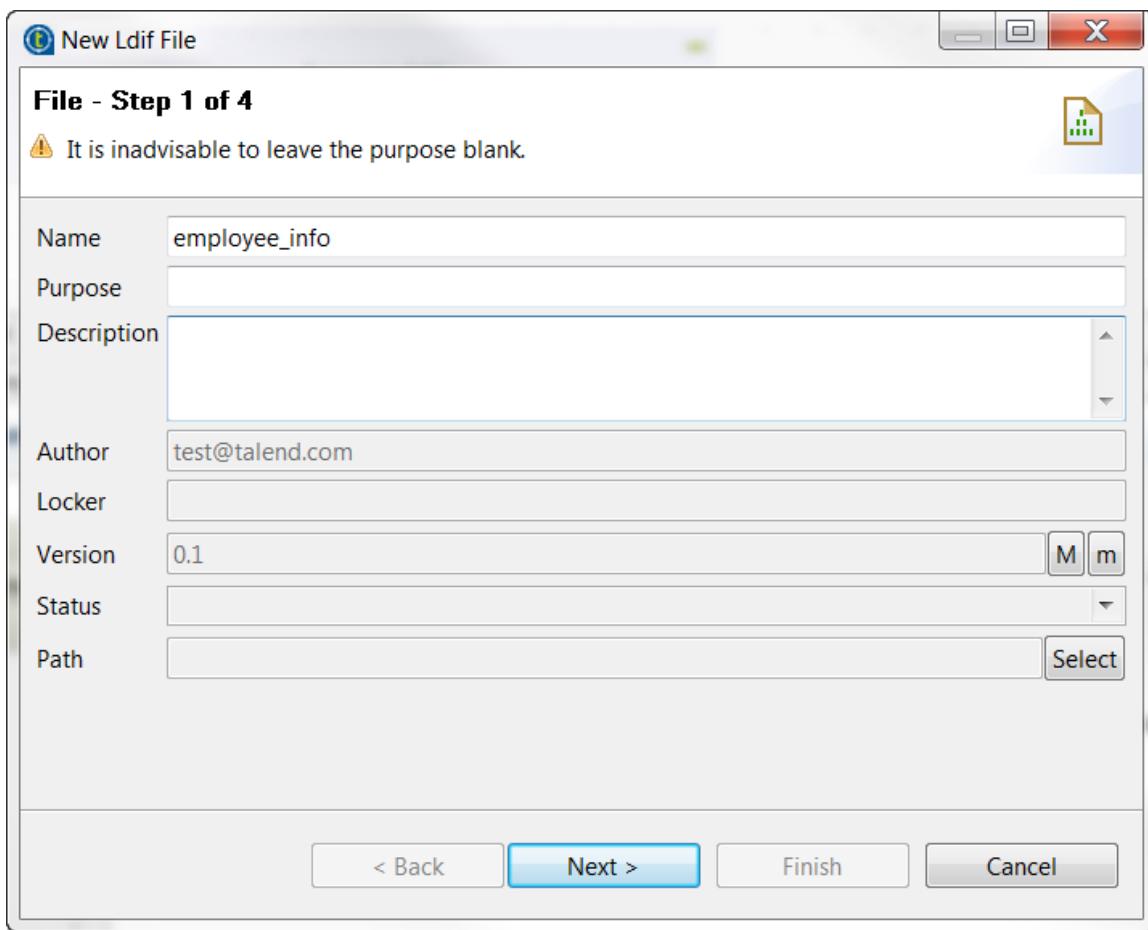
Suivez ensuite les étapes suivantes dans l'assistant :



*Assurez-vous que les modules tiers requis pour l'utilisation de cette fonctionnalité sont bien installés. Pour plus d'informations, consultez le Guide d'installation et de migration Talend.*

1. Afin d'identifier la métadonnées LDIF, renseignez les informations génériques, notamment **Name**, **Purpose** et **Description**.

Le champ **Name** est obligatoire et les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.

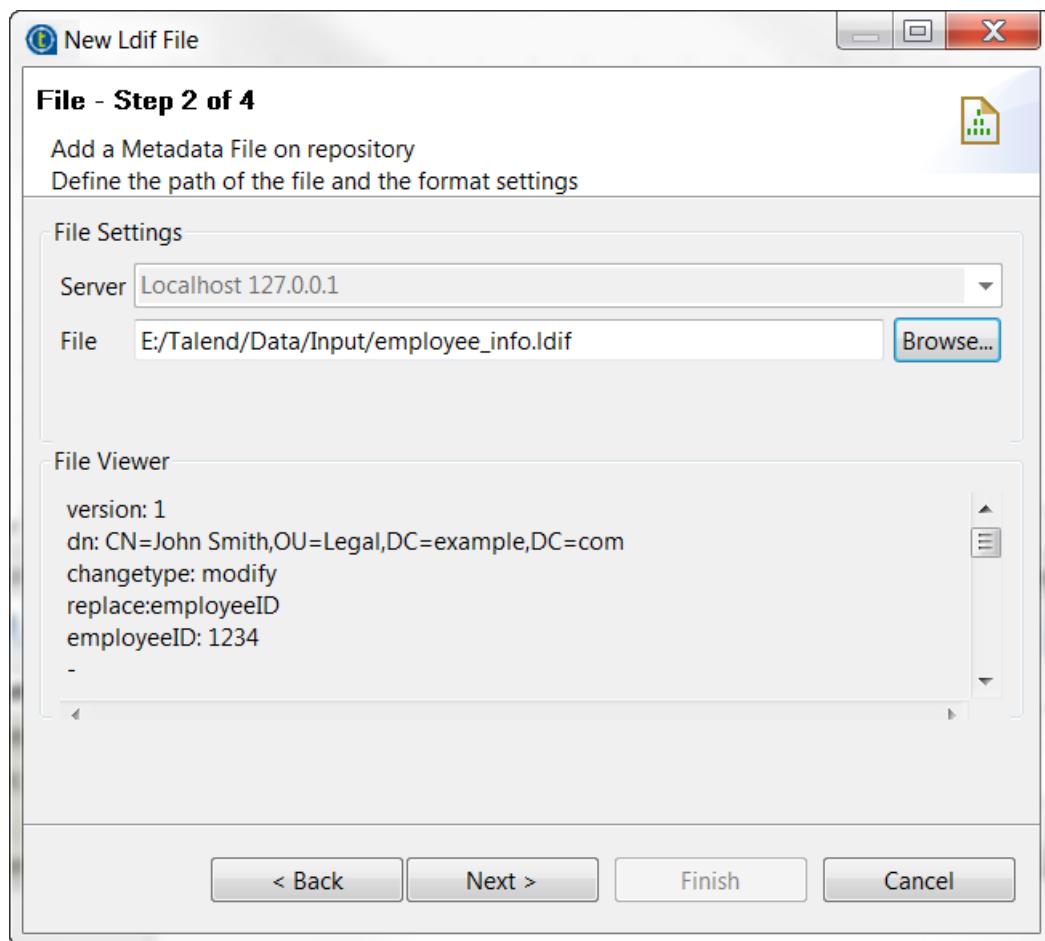


2. Si nécessaire, définissez la version et le statut de la connexion dans les champs **Version** et **Status**, respectivement. Vous pouvez également gérer la version et le statut d'un élément du **Repository** dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#), respectivement.
3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File Ldif**. Ce dossier contient la connexion au fichier juste créée.

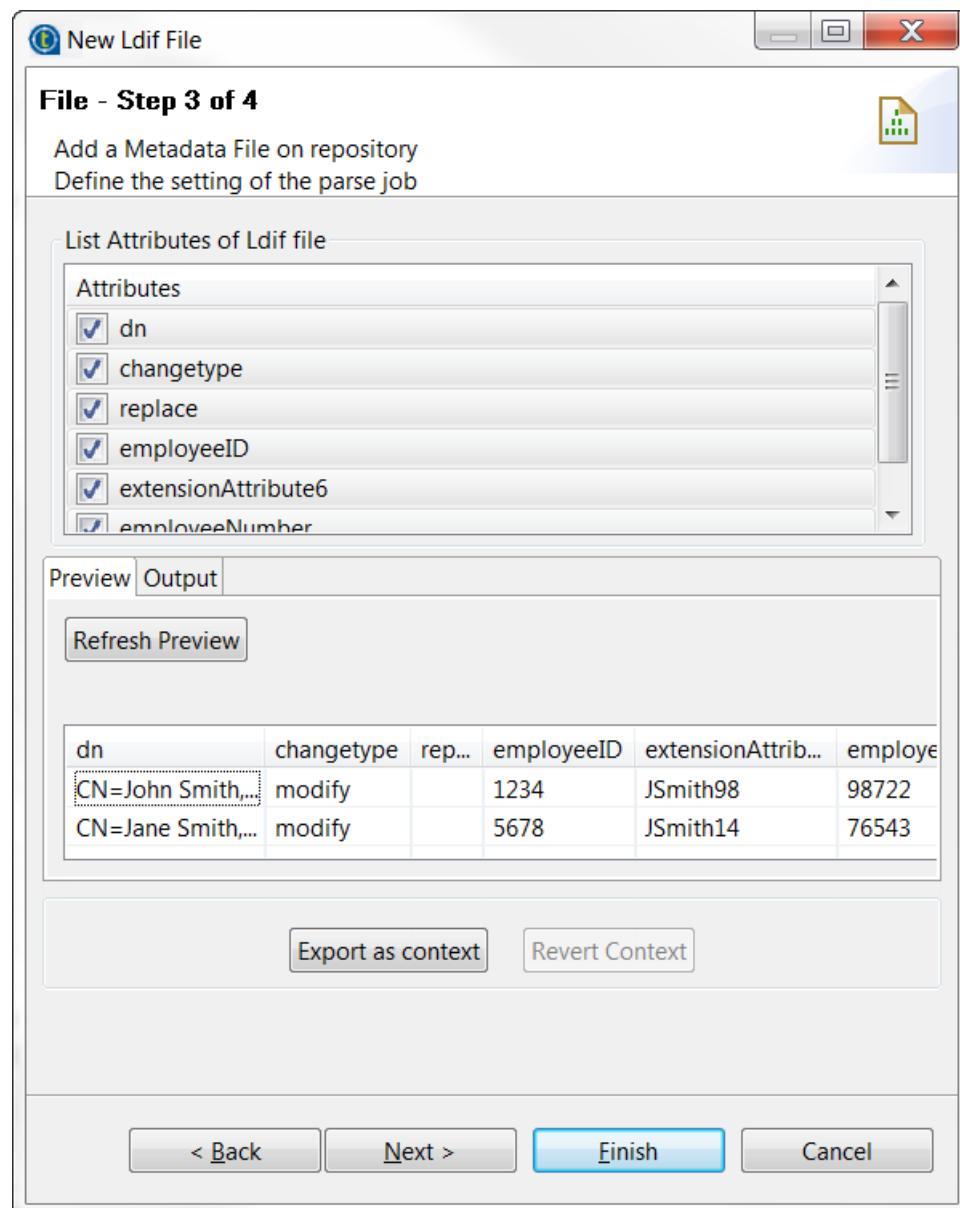
Cliquez sur **Next** pour passer aux paramètres du fichier.

4. Cliquez sur le bouton **Browse...** afin de parcourir votre système vers le fichier ou saisissez son chemin dans le champ **File**.

Passez cette étape si vous sauvegardez la connexion à un fichier LDIF définie dans un composant car son chemin est déjà défini dans le champ **File**.

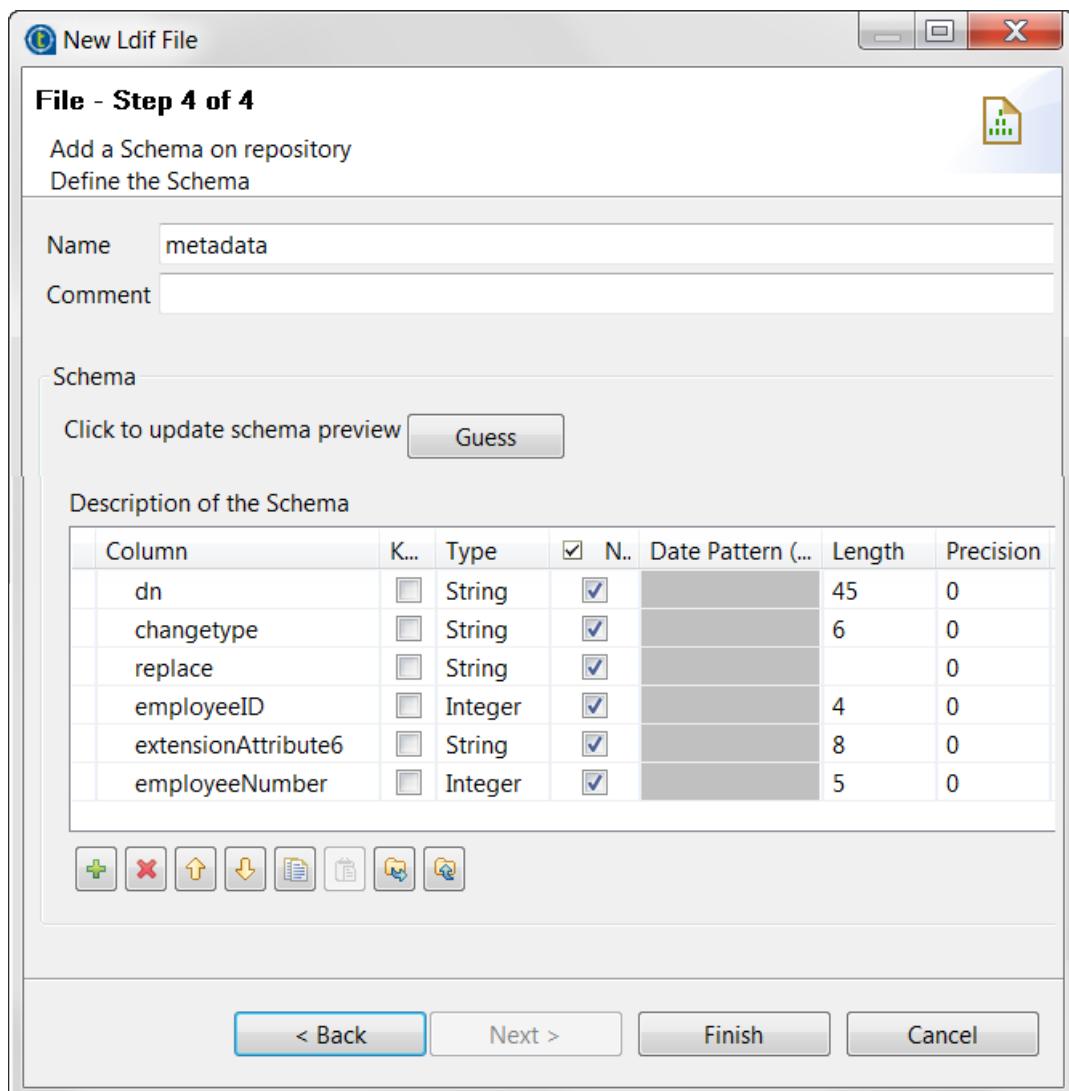


5. La zone **File Viewer** affiche les 50 premières lignes du fichier. Cliquez sur **Next** pour continuer.
  6. Dans la liste d'attributs du fichier chargé, sélectionnez les attributs que vous souhaitez utiliser et cliquez sur **Refresh Preview** afin d'obtenir un aperçu des attributs sélectionnés.
- Cliquez sur **Next** pour passer à la finalisation du schéma.



7. Si besoin, personnalisez le schéma généré. Vous pouvez :

- Renommer le schéma (par défaut *metadata*) et laisser un commentaire.
- Ajouter, supprimer ou déplacer les colonnes du schéma, exporter le schéma dans un fichier XML ou remplacer le schéma en important une définition de schéma depuis un fichier XML à l'aide de la barre d'outils.



Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
  - List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
  - Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
8. Si le fichier LDIF sur lequel est basé le schéma a changé, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez cependant que toute modification ou personnalisation du schéma n'est pas conservée lorsque vous cliquez sur **Guess**.
  9. Cliquez sur **Finish**. Le nouveau schéma apparaît dans le **Repository**, sous le noeud **File ldif** correspondant.

Vous pouvez à présent déposer la connexion au fichier, ou son schéma, depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit file Idif** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.11. Centraliser les métadonnées d'un fichier JSON

Si vous utilisez fréquemment un fichier JSON, vous pouvez utiliser l'assistant **[New Json File]** pour centraliser la connexion au fichier, vos instructions de requêtes XPath et la structure des données dans le **Repository** afin de les réutiliser facilement.

Selon l'option sélectionnée, l'assistant vous permet de créer une connexion à un fichier d'entrée ou de sortie. Dans un Job, les composants **tFileInputJSON** et **tExtractJSONFields** utilisent le schéma d'entrée créé pour lire des fichiers ou des champs JSON et le **tWriteJSONField** utilise le schéma de sortie créé pour écrire un champ JSON pouvant être sauvegardé dans un fichier par le **tFileOutputJSON** ou extrait par le **tExtractJSONFields**.

Pour plus d'informations concernant la configuration des métadonnées JSON d'entrée, consultez [Configurer les métadonnées JSON pour un fichier d'entrée](#).

Pour plus d'informations concernant la configuration des métadonnées JSON de sortie, consultez [Configurer les métadonnées JSON pour un fichier de sortie](#).

Dans la vue **Repository**, développez le nœud **Metadata**, cliquez-droit sur **File JSON** et sélectionnez **Create JSON Schema** dans le menu contextuel pour ouvrir l'assistant **[New Json File]**.

### 10.11.1. Configurer les métadonnées JSON pour un fichier d'entrée

Cette section décrit comment définir et charger une connexion à un fichier et charger un schéma JSON pour un fichier d'entrée. Pour définir une connexion à un fichier et un schéma JSON de sortie, consultez [Configurer les métadonnées JSON pour un fichier de sortie](#).

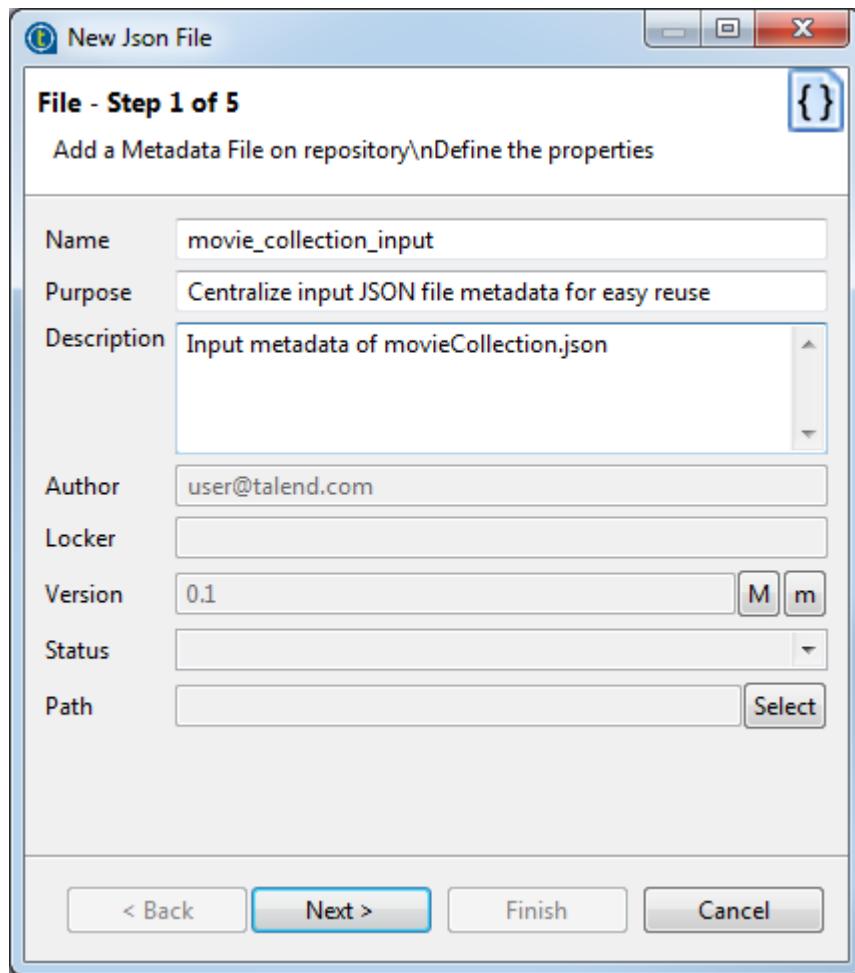
#### Définir les propriétés générales

1. Dans cet assistant, définissez les propriétés générales du fichier, afin d'identifier les métadonnées du fichier JSON, telles que le nom (champ **Name**), l'objectif (champ **Purpose**) et une **Description**.

Le champ **Name** est obligatoire et les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



Dans cette étape, il est recommandé de saisir les informations vous permettant de distinguer vos connexions d'entrée et de sortie, puisque l'étape d'après vous demande de choisir entre les deux.



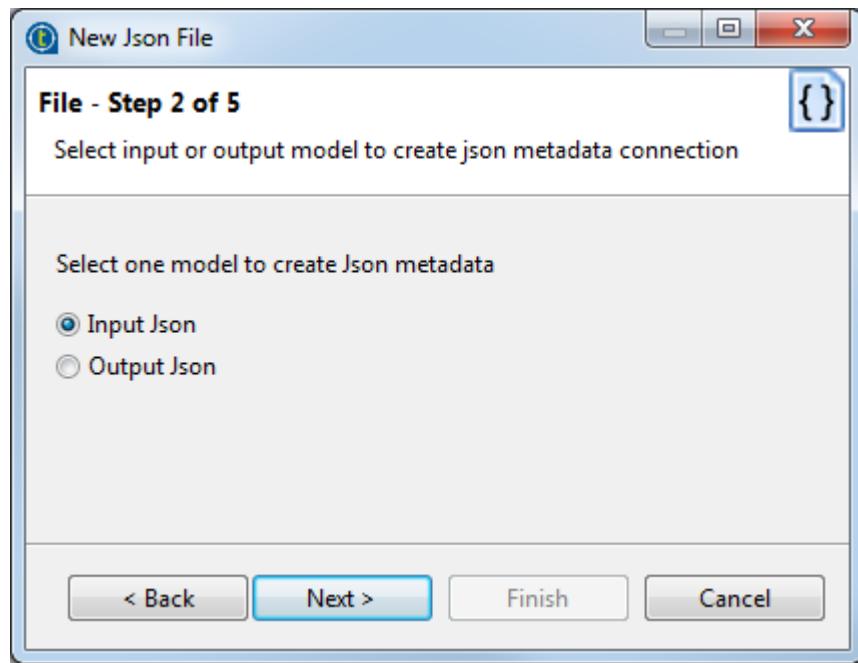
2. Si nécessaire, configurez le numéro de version et le statut, respectivement dans les champs **Version** et **Status**.

Vous pouvez également gérer la version et le statut d'un élément du référentiel dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#) respectivement.

3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File Json** pour contenir la nouvelle connexion au fichier créée.
4. Cliquez sur **Next** pour sélectionner le type de métadonnées.

## Configurer le type de métadonnées et charger le fichier d'entrée

1. Dans la boîte de dialogue, sélectionnez **Input Json** et cliquez sur **Next** pour passer à l'étape suivant et charger le fichier d'entrée.



2. Dans la liste **Read By**, sélectionnez le type de requête pour lire le fichier source JSON.

- **JsonPath** : lit les données JSON selon une requête JsonPath.

Ce type est le type de requêtes par défaut recommandé pour lire des données JSON afin de gagner en performance et d'éviter les problèmes que vous pouvez rencontrer lors de la lecture de données JSON à partir d'une requête XPath.

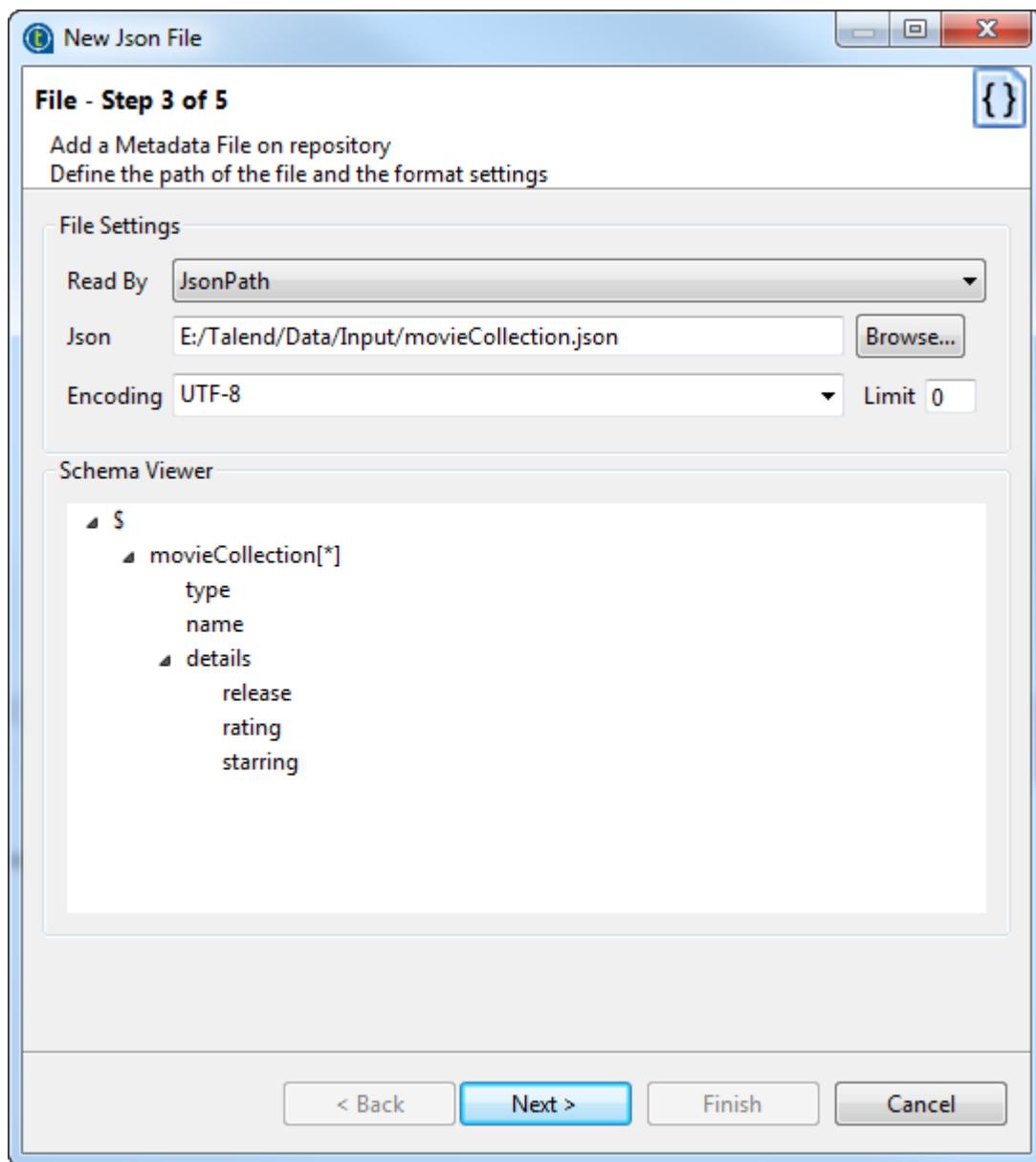
- **Xpath** : lit les données JSON selon une requête XPath.

3. Cliquez sur **Browse...** et parcourez votre répertoire jusqu'au fichier JSON à charger. Vous pouvez également saisir le chemin d'accès complet ou l'URL du fichier JSON.

Dans cet exemple, le fichier JSON d'entrée contient ceci :

```
{
 "movieCollection": [
 {
 "type": "Action Movie",
 "name": "Brave Heart",
 "details": {
 "release": "1995",
 "rating": "5",
 "starring": "Mel Gibson"
 }
 },
 {
 "type": "Action Movie",
 "name": "Edge of Darkness",
 "details": {
 "release": "2010",
 "rating": "5",
 "starring": "Mel Gibson"
 }
 }
]
}
```

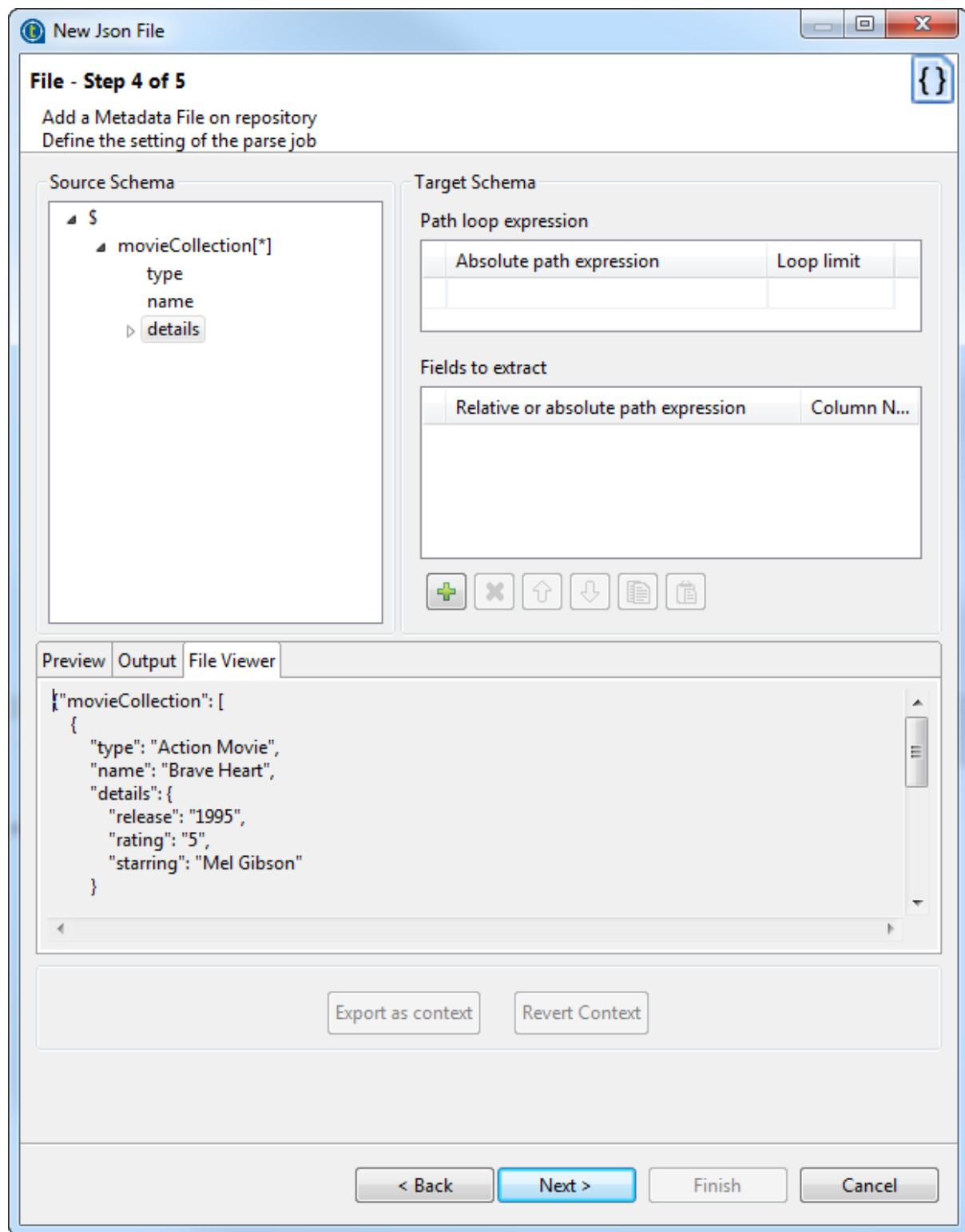
La zone **Schema Viewer** affiche une prévisualisation de la structure JSON. Vous pouvez développer et visualiser chaque niveau de la structure du fichier JSON.



4. Saisissez le type d'encodage dans le champ **Encoding** si le système ne le détecte pas automatiquement.
5. Dans le champ **Limit**, définissez le nombre de niveaux dans la profondeur hiérarchique JSON, profondeur à laquelle vous souhaitez limiter la requête JsonPath ou XPath. Saisissez 0 pour ne pas configurer de limite.  
Configurer une valeur inférieure à 5 pour ce paramètre peut empêcher l'assistant de s'arrêter, dans le cas d'un fichier JSON volumineux.
6. Cliquez sur **Next** afin de configurer les paramètres du schéma.

## Définir le schéma

Dans cette étape, renseignez les paramètres du schéma.



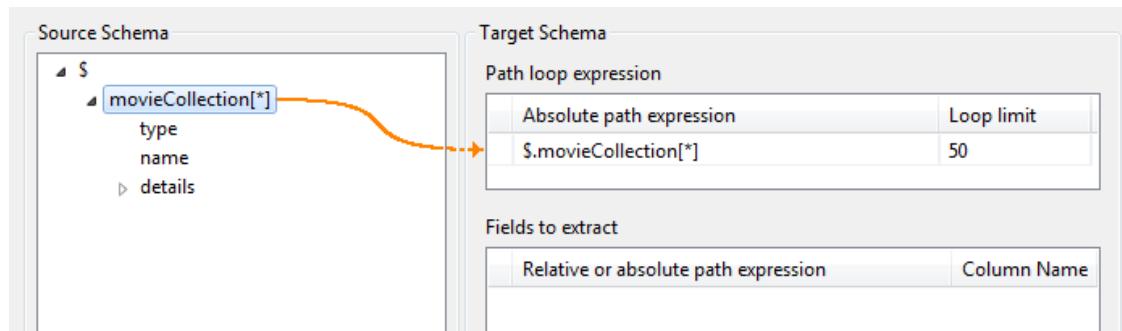
La fenêtre de définition du schéma est composée de quatre vues :

Vue	Description
<b>Source Schema</b>	Arborescence du fichier JSON.
<b>Target Schema</b>	Informations d'extraction et d'itération.
<b>Preview</b>	Aperçu du schéma cible, ainsi que des données d'entrée des colonnes sélectionnées, affichées dans l'ordre défini.
<b>File Viewer</b>	Aperçu des données du fichier JSON.

- Alimentez le champ **Path loop expression**, avec l'expression absolue JsonPath ou Xpath, selon le type de requête sélectionné, du nœud qui fait l'objet de l'itération. Vous pouvez le faire de deux manières :

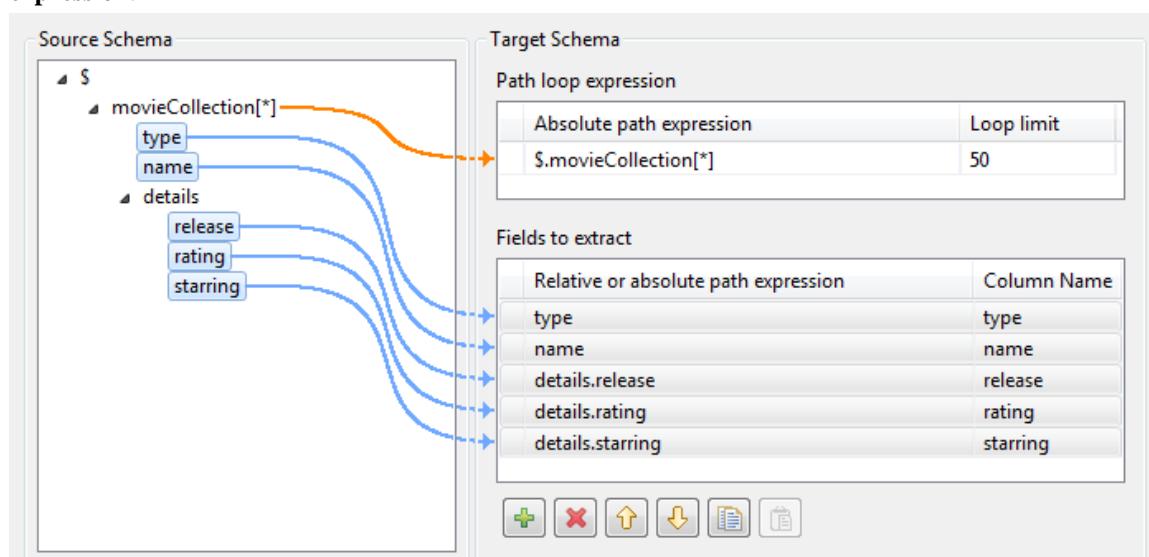
- Saisissez l'expression JsonPath ou Xpath absolue du nœud qui fait l'objet de l'itération. Vous pouvez saisir l'expression entière ou appuyer sur **Ctrl+Espace** pour utiliser la liste d'autocomplétion.
- Déposez le nœud sélectionné de l'arborescence **Source Schema** dans le champ **Absolute path expression** de la table **Path loop expression**.

Une flèche orange relie le nœud à l'expression correspondante.



Le champ **Path loop expression** est obligatoire.

- Saisissez une limite de boucle dans le champ **Loop limit** pour restreindre le nombre de noeuds à itérer.
- Glissez-déposez les nœuds appropriés de la vue **Source Schema** vers le champ **Relative or absolute path expression**.



Vous pouvez sélectionner plusieurs nœuds à déposer dans le tableau, en appuyant sur **Ctrl** ou **Maj**, et en cliquant sur les nœuds adéquats.

- 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 **[x]**.
  - Modifiez l'ordre des colonnes à l'aide des boutons et .

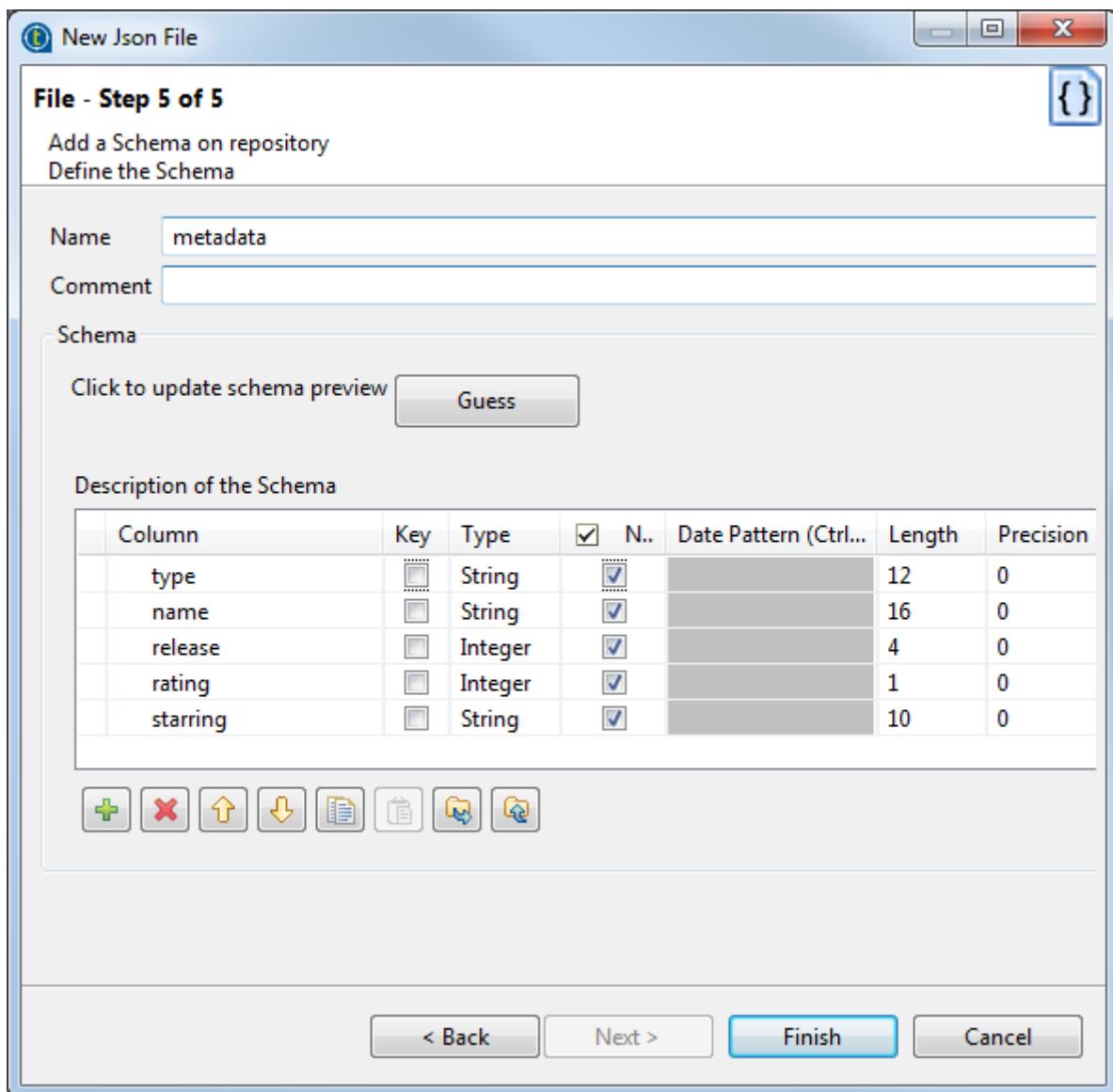
5. Si vous souhaitez que schéma contienne des noms de colonnes différents de ceux récupérés dans le fichier d'entrée, saisissez dans la colonne **Column name**, le libellé des colonnes à afficher dans la zone **Preview** du schéma.
6. Cliquez sur **Refresh Preview** pour afficher l'aperçu du schéma. Les champs apparaissent ainsi dans l'aperçu du schéma dans l'ordre imposé.

type	name	release	rating	starring
Action Movie	Brave Heart	1995	5	Mel Gibson
Action Movie	Edge of Darkness	2010	5	Mel Gibson

7. Cliquez sur **Next** pour finaliser le schéma.

## Finaliser le schéma

La dernière étape de l'assistant affiche le schéma généré et vous permet de le personnaliser selon vos besoins.



- Si nécessaire, renommez le schéma (par défaut, *metadata*) et saisissez un commentaire.

Au besoin, personnalisez le schéma : ajoutez, supprimez ou déplacez des colonnes, exportez le schéma vers un fichier XML, ou remplacez le schéma en important le fichier XML de définition de schéma à l'aide de la barre d'outils.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

2. Si le fichier JSON sur lequel est basé le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.
3. Cliquez sur **Finish**. La nouvelle connexion au fichier, avec son schéma, s'affiche sous le noeud **File Json** de la vue **Repository**.

Vous pouvez glisser-déposer la connexion au fichier ou son schéma de la vue **Repository** dans l'espace de modélisation graphique, en tant que composant **tFileInputJSON** ou **tExtractJSONFields**, ou sur un composant existant afin de réutiliser les métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit JSON** pour ouvrir l'assistant de configuration du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.11.2. Configurer les métadonnées JSON pour un fichier de sortie

Cette section décrit comment définir des métadonnées JSON pour un fichier de sortie. Pour définir les métadonnées JSON pour un fichier d'entrée, consultez [Configurer les métadonnées JSON pour un fichier d'entrée](#).

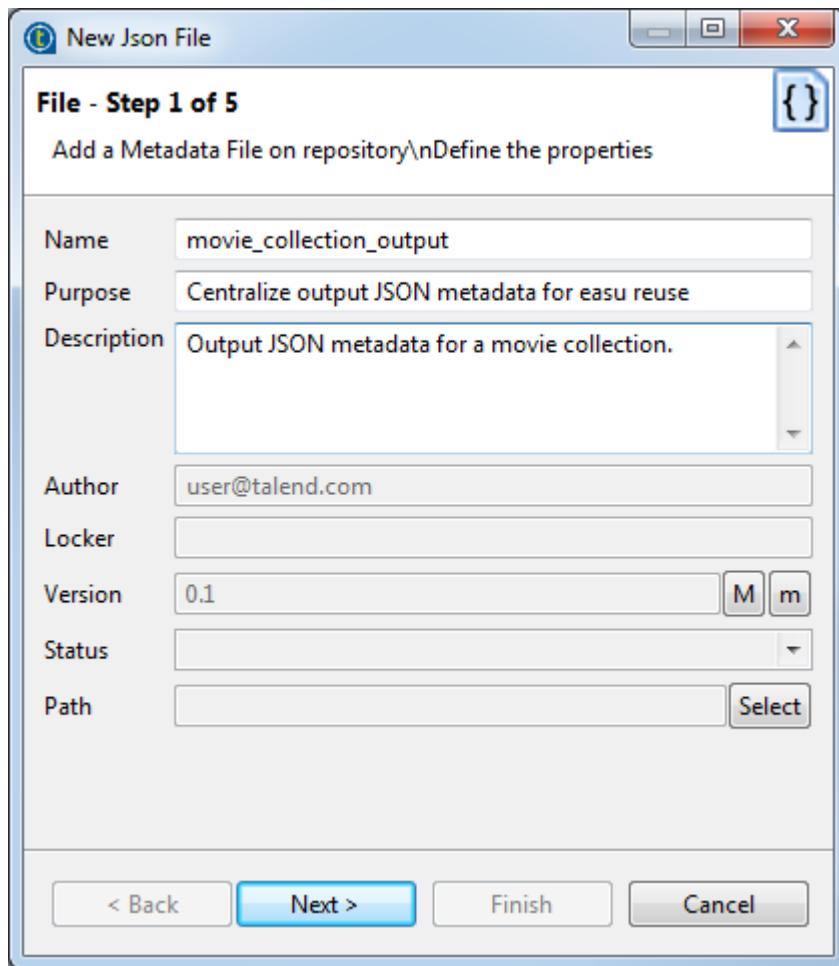
### Définir les propriétés générales

1. Dans l'assistant, renseignez les informations générales dans les champs correspondants afin d'identifier les métadonnées du fichier JSON, notamment les champs **Name**, **Purpose** et **Description**.

Le champ **Name** est obligatoire et les informations contenues dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion.



Dans cette étape, il est recommandé de saisir les informations vous permettant de distinguer vos connexions d'entrée et de sortie, puisque l'étape d'après vous demande de choisir entre les deux.



- Si nécessaire, configurez le numéro de version et le statut, respectivement dans les champs **Version** et **Status**.

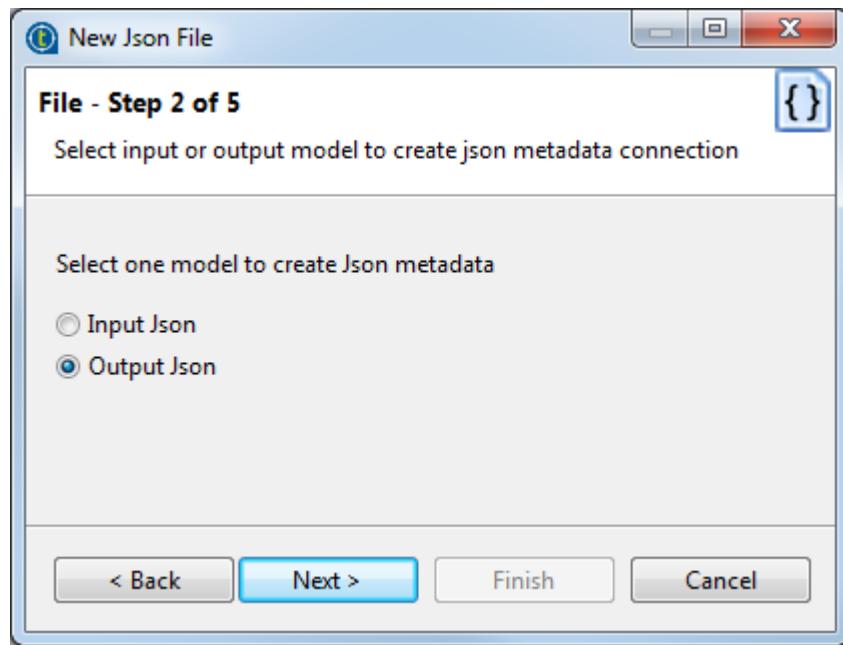
Vous pouvez également gérer la version et le statut d'un élément du référentiel dans la boîte de dialogue **[Project Settings]**. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#) respectivement.

- Au besoin, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **File Json** pour contenir votre nouvelle connexion à un fichier.
- Cliquez sur **Next** pour configurer le type de métadonnées.

## Configurer le type de métadonnées et charger le fichier JSON modèle

Dans cette étape, vous allez définir le type de schéma, Input ou Output. Dans cet exemple, sélectionnez le type Output.

- Dans la boîte de dialogue, sélectionnez **Output JSON**, cliquez sur **Next** pour passer à l'étape suivante de l'assistant.



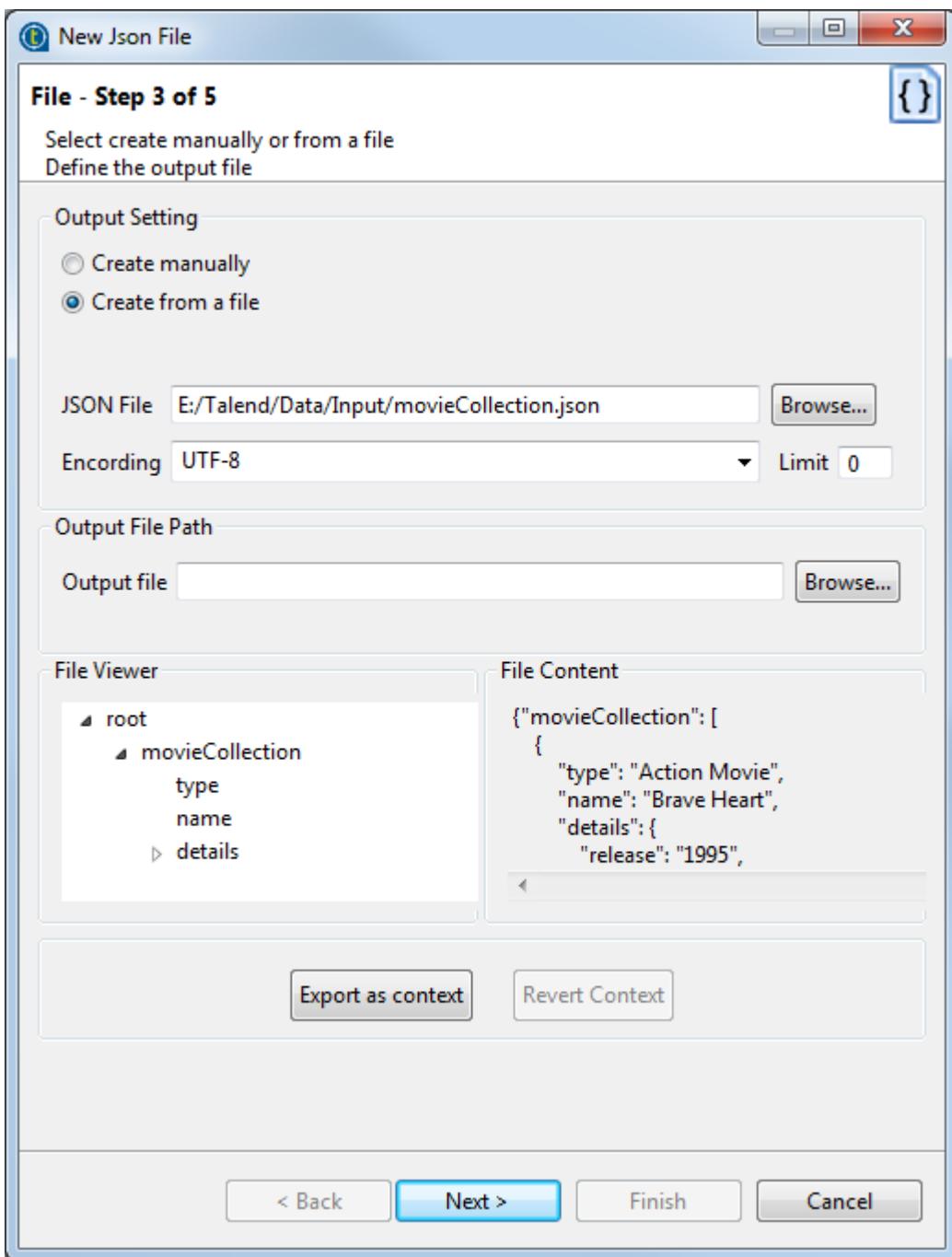
- Choisissez de créer vos métadonnées manuellement ou à partir d'un fichier modèle JSON existant.

Si vous choisissez de le créer manuellement (en sélectionnant **Create manually**), vous devez configurer vous-même votre schéma, vos colonnes sources et vos colonnes cibles. Le fichier sera créé lors de l'utilisation dans un Job d'un composant de sortie, comme le **tWriteJSONField**. Dans cet exemple, créez le schéma de sortie en chargeant un fichier JSON existant.

Dans cet exemple, créez les métadonnées de sortie en chargeant un fichier JSON existant. Sélectionnez donc l'option **Create from a file**.

- Cliquez sur le bouton **Browse...** à côté du champ **JSON File**, parcourez votre système jusqu'au fichier JSON dont la structure va être appliquée au fichier/champ de sortie JSON et double-cliquez sur le fichier. Sinon, vous pouvez saisir le chemin complet ou l'URL pointant vers le fichier JSON modèle.

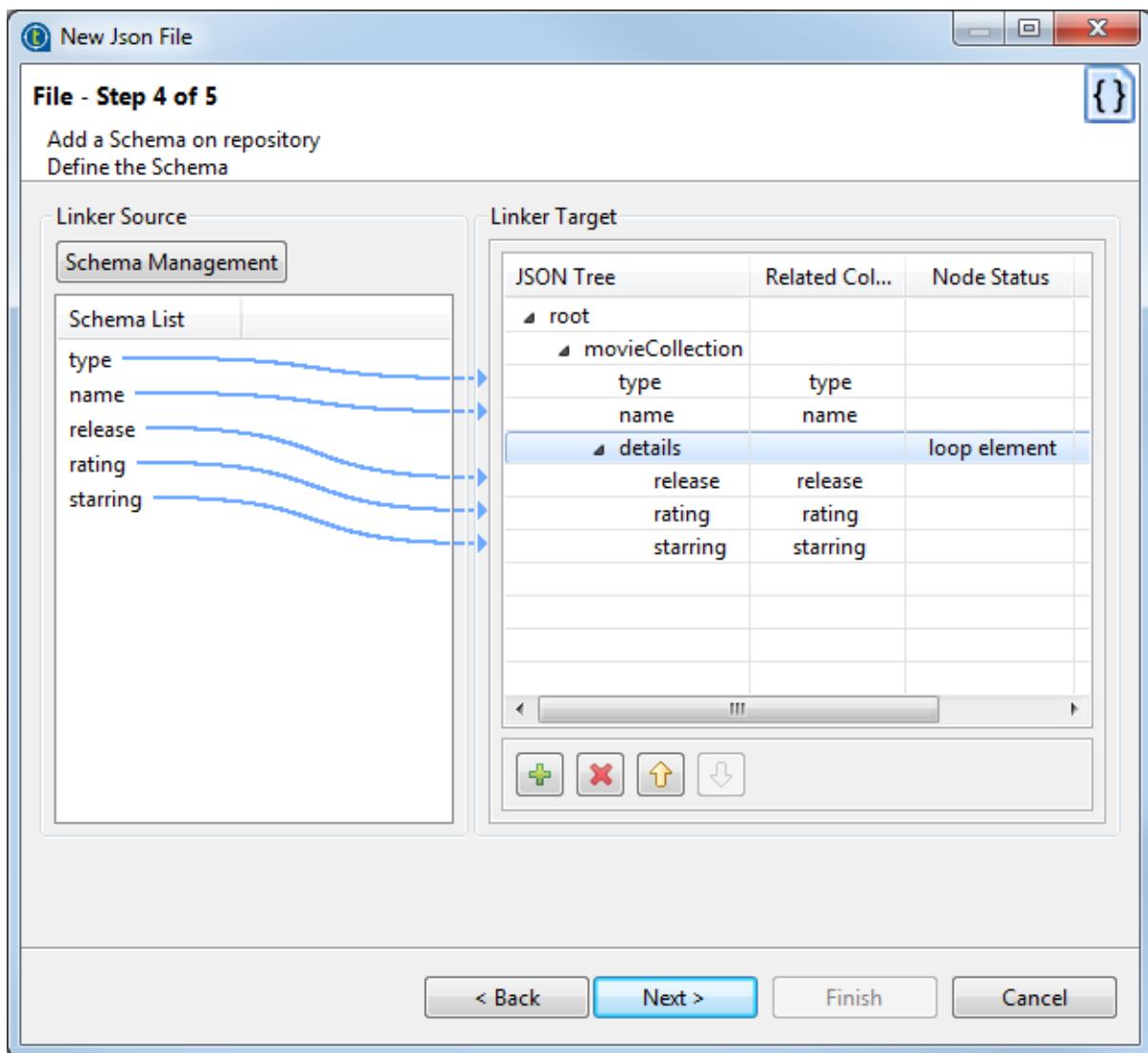
La zone **File Viewer** affiche un aperçu de la structure JSON et la zone **File Content** affiche au maximum les cinquante premières lignes du fichier.



4. Renseignez le champ **Encoding** si le système n'a pas détecté automatiquement l'encodage.
5. Dans le champ **Limit**, définissez le nombre de niveaux dans la profondeur hiérarchique JSON, profondeur à laquelle vous souhaitez limiter la requête JsonPath ou XPath. Saisissez 0 pour ne pas configurer de limite.  
Configurer une valeur inférieure à 5 pour ce paramètre peut empêcher l'assistant de s'arrêter, dans le cas d'un fichier JSON volumineux.
6. Vous pouvez également, de manière facultative, spécifier un chemin vers un fichier de sortie.
7. Cliquez sur **Next** pour définir le schéma.

## Définir le schéma

Lorsque les opérations précédentes sont terminées, les colonnes de la zone **Linker Source** sont automatiquement mappées vers celles correspondantes dans la zone **Linker Target**, comme le montrent les flèches bleues.



Dans cette étape, configuez le schéma de sortie. Le tableau suivant décrit comment faire :

Pour...	Effectuer...
Définir un élément de boucle	<p>Dans la zone <b>Linker Target</b>, cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Set As Loop Element</b> dans le menu contextuel.</p> <p> Cette opération est obligatoire pour définir un élément sur lequel effectuer une boucle.</p>
Définir un élément de groupe	<p>Dans la zone <b>Linker Target</b>, cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Set As Group Element</b> dans le menu contextuel.</p> <p> Vous pouvez configurer l'élément parent de l'élément de boucle en tant que <b>group element</b> à condition que l'élément parent ne soit pas la racine de l'arborescence JSON.</p>
Créer un élément fils pour un élément	<p>Dans la zone <b>Linker Target</b>,</p> <ul style="list-style-type: none"> <li>• Cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Add Sub-element</b> dans le menu contextuel, saisissez un nom pour le sous-élément dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>.</li> <li>• Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton <b>[+]</b> en bas, sélectionnez <b>Create as sub-element</b> dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>. Ensuite, saisissez un nom pour le sous-élément dans la boîte de dialogue suivante et cliquez sur <b>OK</b>.</li> </ul>
Créer un attribut pour un élément	Dans la zone <b>Linker Target</b> ,

Pour...	Effectuer...
	<ul style="list-style-type: none"> <li>Cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Add Attribute</b> dans le menu contextuel, saisissez un nom pour l'attribut dans la boîte de dialogue qui apparaît et cliquez sur <b>OK</b>.</li> <li>Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton <b>[+]</b> en bas, sélectionnez <b>Create as attribute</b> dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>. Ensuite, saisissez un nom pour l'attribut dans la boîte de dialogue suivante et cliquez sur <b>OK</b>.</li> </ul>
Créer un espace de noms pour un élément	<p>Dans la zone <b>Linker Target</b>,</p> <ul style="list-style-type: none"> <li>Cliquez-droit sur l'élément qui vous intéresse et sélectionnez <b>Add Name Space</b> dans le menu contextuel, saisissez un nom pour l'espace de nommage dans la boîte de dialogue qui apparaît et cliquez sur <b>OK</b>.</li> <li>Sélectionnez l'élément qui vous intéresse, cliquez sur le bouton <b>[+]</b> en bas, sélectionnez <b>Create as name space</b> dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>. Ensuite, saisissez un nom pour l'espace de nommage dans la boîte de dialogue suivante et cliquez sur <b>OK</b>.</li> </ul>
Supprimer un ou plusieurs élément(s)/attribut(s)/espace(s) de noms	<p>Dans la zone <b>Linker Target</b>,</p> <ul style="list-style-type: none"> <li>Cliquez-droit sur les éléments/attributs/espace de nommage qui vous intéressent et sélectionnez <b>Delete</b> dans le menu contextuel.</li> <li>Sélectionnez les éléments/attributs/espace de nommage qui vous intéressent et cliquez sur le bouton <b>[x]</b> en bas.</li> <li>Sélectionnez les éléments/attributs/espace de nommage qui vous intéressent et appuyez sur la touche <b>Suppr.</b></li> </ul> <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 <b>Linker Target</b> , sélectionnez l'élément qui vous intéressent et cliquez sur les boutons  et  <ul style="list-style-type: none"> <li>La valeur que vous avez configurée va remplacer toute valeur récupérée du flux d'entrée dans votre Job.</li> <li>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.</li> </ul>
Créer un mapping source-cible	<p>Sélectionnez la colonne qui vous intéresse, dans la zone <b>Linker Source</b>, déposez-la sur le nœud qui vous intéresse de la zone <b>Linker Target</b> et sélectionnez <b>Create as sub-element of target node</b>, <b>Create as attribute of target node</b>, ou <b>Add linker to target node</b> selon vos besoins, dans la boîte de dialogue qui apparaît, puis cliquez sur <b>OK</b>.</p> <p>Si vous choisissez une option qui n'est pas permise pour le nœud cible, un message d'avertissement apparaît et l'opération échoue.</p>
Supprimer un mapping source-cible	Dans la zone <b>Linker Target</b> , cliquez-droit sur le nœud qui vous intéresse et sélectionnez <b>Disconnect Linker</b> dans le menu contextuel.
Créer une structure JSON à partir d'un autre fichier JSON	Cliquez-droit sur tout élément du schéma dans la zone <b>Linker Target</b> et sélectionnez <b>Import XML Tree</b> dans le menu contextuel afin de charger un autre fichier JSON. Ensuite, vous devez créer manuellement les mappings source-cible et définir à nouveau le schéma de sortie.



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 la zone **Linker Target**, cliquez-droit sur l'élément que vous souhaitez définir comme élément de boucle et sélectionnez **Set As Loop Element** dans le menu contextuel.

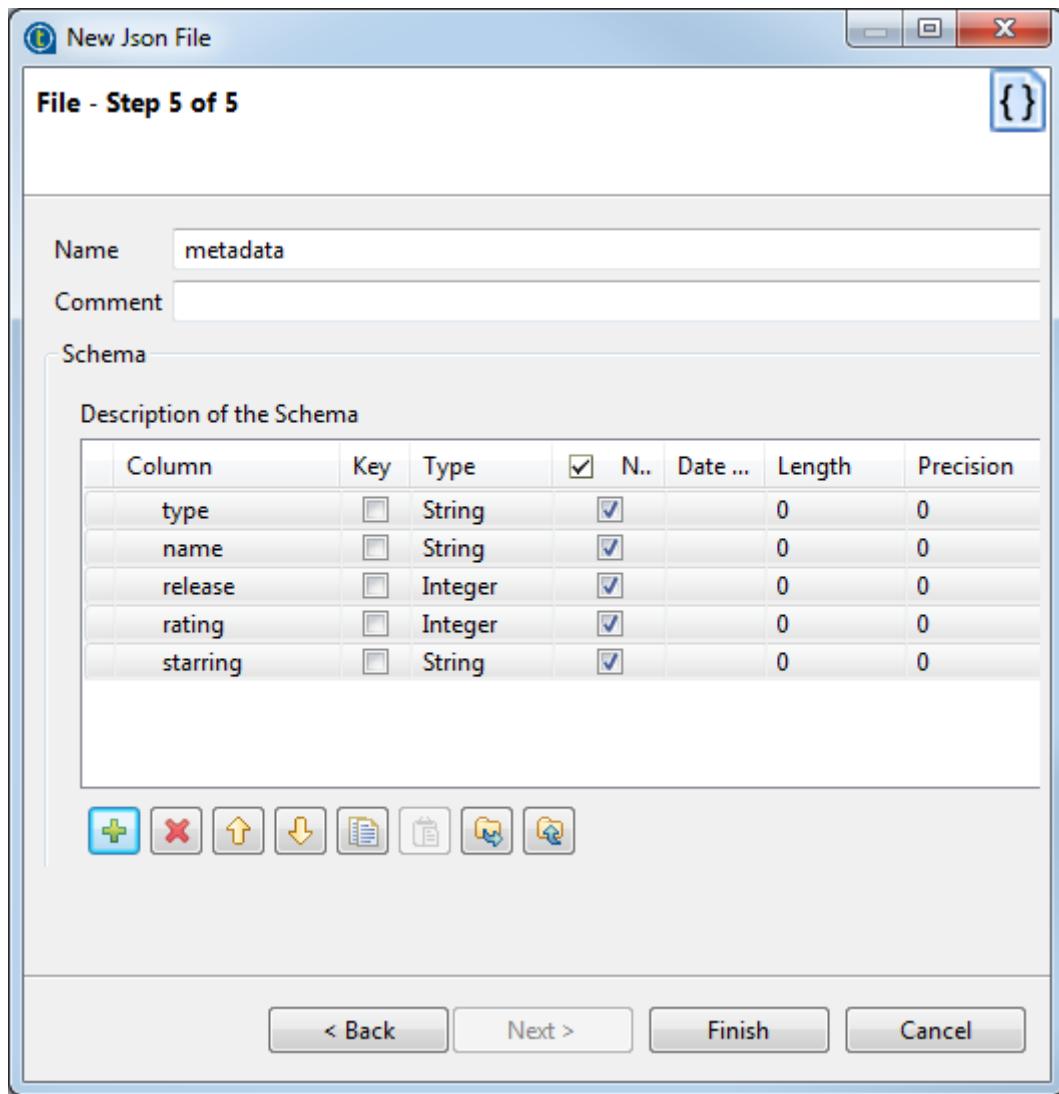
Dans cet exemple, configurez la boucle sur l'élément *details*.

The screenshot shows the 'Linker Target' interface with a JSON tree structure. The tree has nodes for 'root', 'movieCollector', 'type', 'name', and 'details'. The 'details' node is selected, and a context menu is open over it. The menu items include: Add Sub-element, Add Attribute, Add Name Space, Delete, Disconnect Linker, Set A Fix Value, Set As Loop Element (which is highlighted with a blue selection bar), Set As Group Element, Remove Group Element, and Import JSON Tree. At the bottom of the interface, there are four small icons: a green plus sign, a red minus sign, an orange up arrow, and a blue down arrow.

2. Personnalisez le mapping, si nécessaire.
3. Cliquez sur **Next** pour finaliser le schéma.

## Finaliser le schéma

La dernière étape de l'assistant affiche le schéma final généré et vous permet de le personnaliser selon vos besoins.



- Si nécessaire, renommez le schéma (par défaut, *metadata*) et saisissez un commentaire.

Au besoin, personnalisez le schéma : ajoutez, supprimez ou déplacez des colonnes, exportez le schéma vers un fichier XML, ou remplacez le schéma en important le fichier XML de définition de schéma à l'aide de la barre d'outils.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

2. Si le fichier JSON sur lequel est basé le schéma a été modifié, cliquez sur le bouton **Guess** afin de générer à nouveau le schéma. Notez que, si vous avez personnalisé le schéma, la fonctionnalité **Guess** ne retient pas ces modifications.
3. Cliquez sur **Finish**. La nouvelle connexion au fichier, avec son schéma, s'affiche sous le noeud **File Json** de la vue **Repository**.

Vous pouvez glisser-déposer la connexion au fichier ou son schéma de la vue **Repository** dans l'espace de modélisation graphique, en tant que composant **tFileInputJSON** ou **tExtractJSONFields**, ou sur un composant existant afin de réutiliser les métadonnées. Pour plus d'informations sur l'utilisation des métadonnées centralisées, consultez [Comment utiliser les métadonnées centralisées dans un Job](#) et [Paramétriser un schéma du Repository dans un Job](#).

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit JSON** pour ouvrir l'assistant de configuration du fichier.

Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.12. Centraliser les métadonnées d'une connexion LDAP

Si vous devez souvent accéder à un répertoire LDAP, vous pouvez centraliser les informations de connexion à un serveur LDAP dans le **Repository** afin de les réutiliser facilement.

Vous pouvez créer une connexion LDAP depuis un répertoire LDAP disponible ou à partir des paramètres définis dans un Job.

Afin de créer une connexion depuis un répertoire LDAP disponible, développez le noeud **Metadata** du **Repository**. Ensuite, cliquez-droit sur **LDAP** puis sélectionnez **Create LDAP schema** dans le menu contextuel afin d'ouvrir l'assistant [**Create new LDAP schema**].

Pour centraliser une connexion à un serveur LDAP que vous avez définie dans un Job ainsi que son schéma, cliquez sur l'icône  dans la vue **Basic settings** du composant correspondant afin d'ouvrir l'assistant de configuration des métadonnées du fichier. Notez que **Built-in** doit être sélectionné dans la liste **Property Type**.

Contrairement à l'assistant de connexion DB, l'assistant LDAP regroupe à la fois la connexion au serveur et la définition du schéma dans une procédure en cinq étapes.

### Définir les propriétés générales

1. Dans cet assistant, définissez les propriétés générales du fichier, afin d'identifier les métadonnées de la connexion LDAP, telles que le nom (champ **Name**), l'objectif (champ **Purpose**) et une **Description**.

Le champ **Name** est obligatoire et les informations que vous renseignez dans le champ **Description** apparaissent en tant qu'info-bulle lorsque vous placez le pointeur de votre souris sur la connexion LDAP.

2. Si nécessaire, configurez le numéro de version et le statut, respectivement dans les champs **Version** et **Status**. Vous pouvez également gérer la version et le statut d'un élément du référentiel dans la boîte de dialogue [**Project Settings**]. Pour plus d'informations, consultez [Gérer les versions](#) et [Gérer les statuts](#) respectivement.

3. Si nécessaire, cliquez sur le bouton **Select** à côté du champ **Path** afin de sélectionner un dossier sous le nœud **LDAP** pour contenir la nouvelle connexion LDAP créée.
4. Cliquez sur **Next** pour définir les détails de connexion.

## Définir la connexion au serveur

1. 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 below to check the connection.

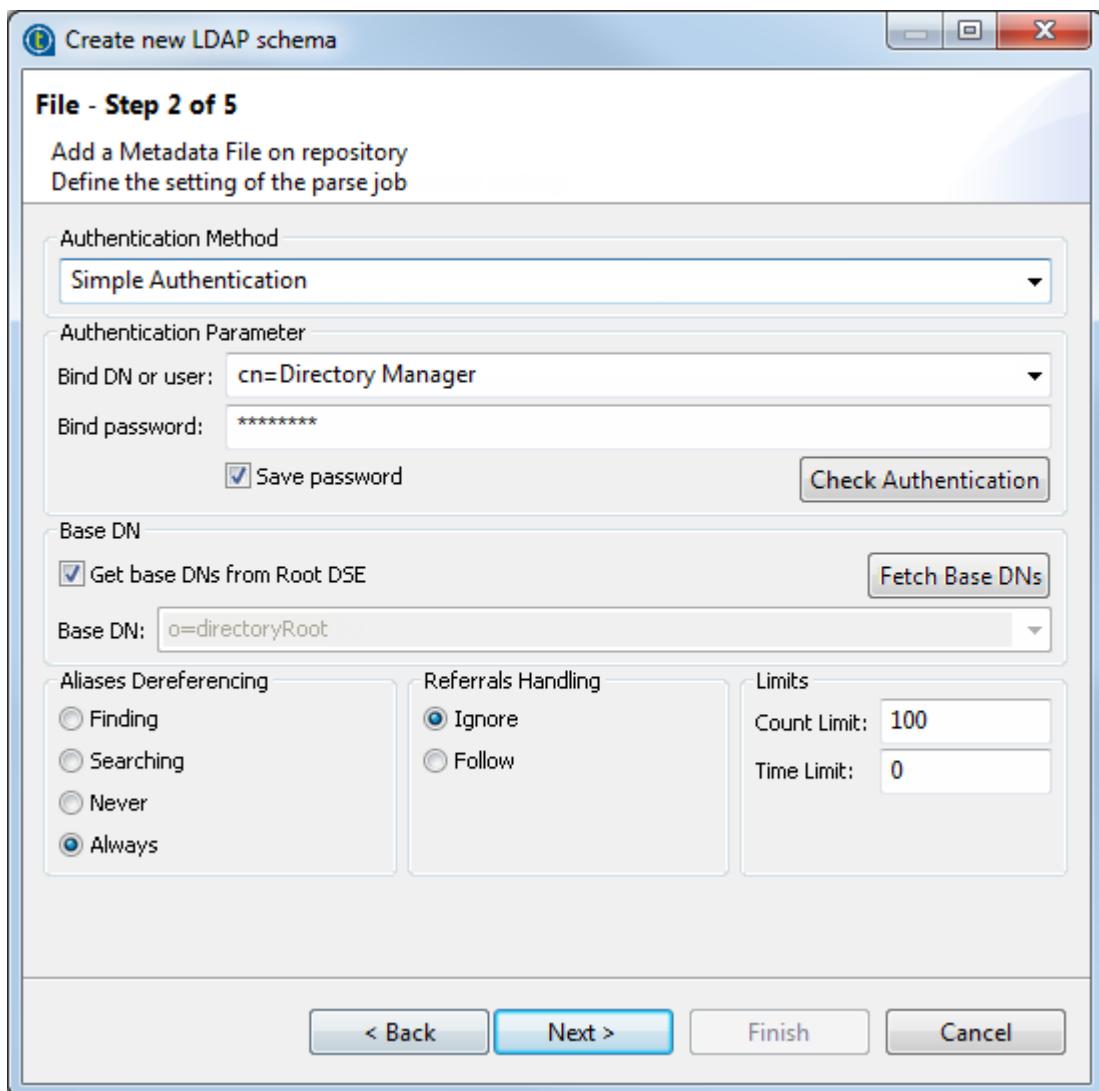
**Check Network Parameter**

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

2. Vérifiez votre connexion à l'aide de bouton **Check Network Parameter**, puis cliquez sur le bouton **Next**.
3. Cliquez sur **Next** pour valider cette étape et continuer.

## Configurer les paramètres d'accès à LDAP

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



Champ	Description
<b>Authentication method</b>	Pour une <b>Simple authentication</b> , renseignez le champ <b>Authentication Parameters</b> . <b>Anonymous authentication</b> ne nécessite aucun paramètre d'authentification.
<b>Authentication Parameters</b>	<b>Bind DN or User</b> : connectez-vous en fonction de la méthode d'authentification LDAP choisie.  <b>Bind password</b> : renseignez le mot de passe.  <b>Save password</b> : cochez cette case pour enregistrer les informations d'authentification.
<b>Get Base DN from Root DSE / Base DN</b>	Chemin d'accès à l'arborescence de l'utilisateur.  Le bouton <b>Fetch Base DNs</b> redirige le DN automatiquement à la racine.
<b>Alias Dereferencing</b>	<b>Never</b> permet l'amélioration des performances de recherche si vous êtes sûr qu'aucun alias n'est déréférencé. Par défaut, utilisez <b>Always</b> . <b>Always</b> : les alias sont toujours déréférencés.  <b>Never</b> : les alias ne jamais sont déréférencés.  <b>Searching</b> : cette option déréférence les alias uniquement après la résolution du nom.  <b>Finding</b> : cette option déréférence les alias uniquement lors de la résolution du nom.
<b>Referral Handling</b>	Redirection des requêtes utilisateur :

Champ	Description
	<b>Ignore</b> ne tient pas compte des redirections des requêtes. <b>Follow</b> tient compte des redirections des requêtes.
<b>Limit</b>	Limite le nombre d'enregistrements lus

2. Cliquez sur **Check authentication** pour vérifier vos droits d'accès.
3. Cliquez sur **Fetch Base DNs** pour conserver le DN et cliquez sur **Next** pour continuer.
4. Si des bibliothèques tierces requises pour la configuration de la connexion LDAP sont manquantes, un assistant d'installation de modules externes s'affiche. Installez les bibliothèques requises en suivant les étapes de l'assistant. Pour plus d'informations sur l'installation des modules tiers, consultez le *Guide d'installation et de migration Talend*.

## Définir le schéma

1. 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 title bar says 'File - Step 4 of 5'. The main area has two sections: 'List attributes of LDAP Schema' and 'Filter'. In the 'List attributes of LDAP Schema' section, there is a checkbox for 'Attributes' and several checkboxes for attributes: 'description', 'userpassword', 'uid', 'sn', and 'cn'. The 'uid' checkbox is checked. In the 'Filter' section, there is a text input field containing the filter '(&(objectClass=\*))'. Below these sections is a 'Preview' section with a 'Refresh Preview' button. A preview table shows four columns: 'uid', 'mail', 'givenname', and 'telephonenumber'. The data rows are:

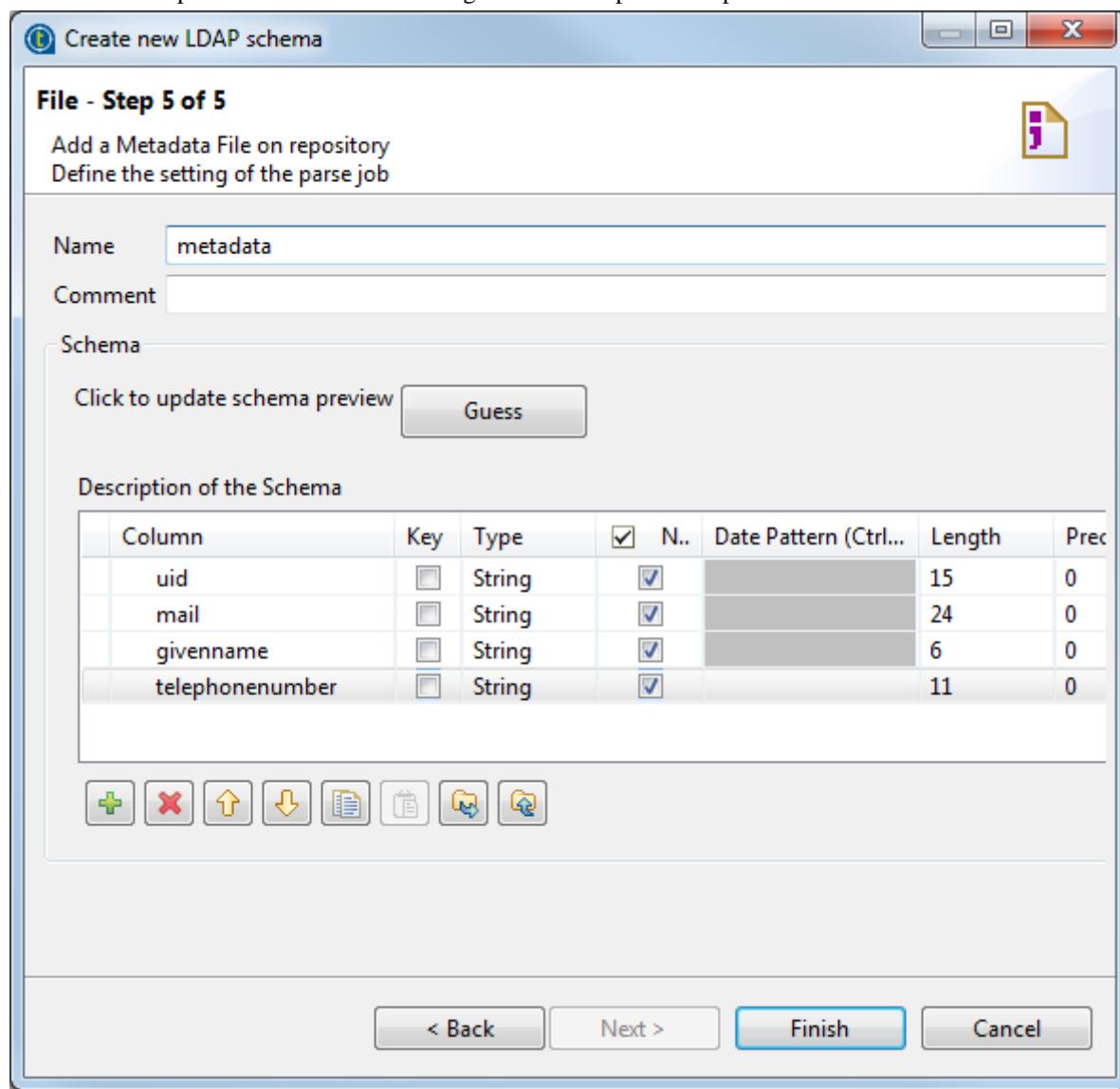
uid	mail	givenname	telephonenumber
PIERRE DUPONT	Pierre.Dupont@talend.com	PIERRE	00149684750
PIERRE DUPON...	mhirt78@talend.com	PIERRE	
mhirt			
greg			

At the bottom of the dialog box are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

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

## Finaliser le schéma

Cette dernière étape affiche le schéma LDAP généré et vous permet de personnaliser le schéma final.



- Si nécessaire, renommez le schéma (par défaut, *metadata*) dans le champ **Name** et saisissez un commentaire dans le champ **Comment**. Vous pouvez apporter d'autres modifications telles que :

- Définir à nouveau les colonnes en éditant les champs appropriés.
- Ajouter ou supprimer une colonne à l'aide des boutons et .
- Modifier l'ordre des colonnes à l'aide des boutons et .

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
  - List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
  - Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.
2. Si l'annuaire LDAP ayant donné lieu au schéma a changé, cliquez sur le bouton **Guess** pour générer le schéma une nouvelle fois. Notez que les modifications que vous avez apportées au schéma seront perdues si vous cliquez sur le bouton **Guess**.
3. Cliquez sur **Finish**. Le nouveau schéma s'affichera dans le **Repository**, sous le nœud LDAP correspondant.

Vous pouvez à présent déposer n'importe quelle connexion à un fichier ou n'importe quel schéma depuis le **Repository** dans l'espace de modélisation graphique en tant que nouveau composant. Vous pouvez également la déposer sur un composant afin de réutiliser ses métadonnées.

Pour modifier une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Edit LDAP schema** afin d'ouvrir l'assistant de configuration des métadonnées du fichier.

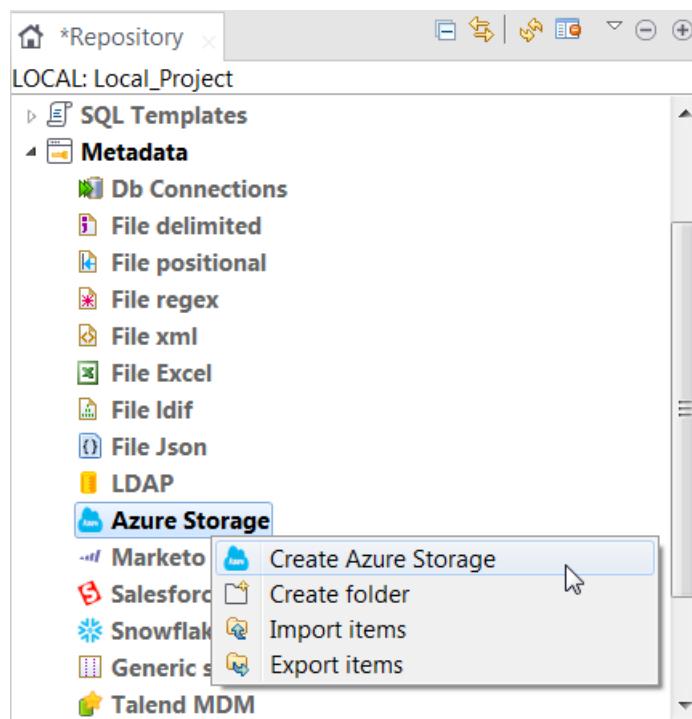
Pour ajouter un nouveau schéma à une connexion à un fichier existante, cliquez-droit sur la connexion dans le **Repository** et sélectionnez **Retrieve Schema** dans le menu contextuel.

Pour éditer un schéma de fichier existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

## 10.13. Centraliser les métadonnées Azure Storage

Vous pouvez utiliser l'assistant de création de métadonnées Azure Storage fourni par le *Studio Talend* pour configurer rapidement une connexion à Azure Storage et récupérer le schéma des conteneurs, des files et des tables.

1. Dans le **Repository**, développez le nœud **Metadata**, puis cliquez-droit sur le nœud **Azure Storage** dans l'arborescence. Dans le menu contextuel, sélectionnez **Create Azure Storage** pour ouvrir l'assistant [Azure Storage].



2. Dans la boîte de dialogue **Azure Storage Connection Settings**, renseignez (ou mettez à jour, si nécessaire) les informations des champs listés dans le tableau suivant.

**Azure Storage Connection Settings**

Fill in fields to configure connection.

Name	<input type="text" value="azurestorage_connection"/>	*
Account Name	<input type="text" value=""/>	
Account Key	<input type="text" value="*****"/>	*
Protocol	<input type="button" value="HTTPS"/>	*
<input type="checkbox"/> Use Azure Shared Access Signature		
<input type="button" value="Test connection"/>		
<input type="button" value="Export as context"/> <input type="button" value="Revert Context"/>		
<input type="button" value="&lt; Back"/> <input type="button" value="Next &gt;"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>		

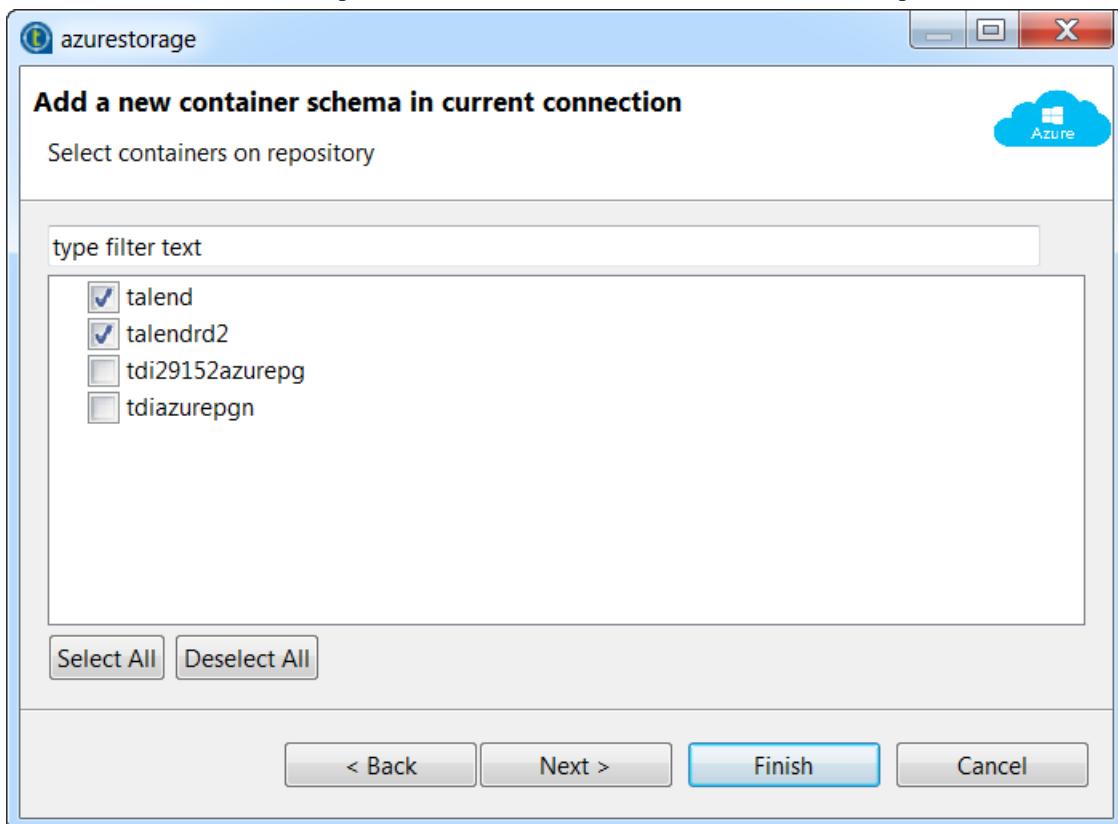
Champ	Description
Name	Saisissez le nom de la connexion à créer.
Account Name	Saisissez le nom du compte de stockage auquel vous devez accéder. Vous pouvez trouver un nom de compte de stockage dans le dashboard <b>Manage Access Keys</b> du système Microsoft Azure Storage à utiliser.

Champ	Description
Account Key	Saisissez la clé associée au compte de stockage auquel vous devez accéder. Deux clés sont disponibles pour chaque compte. Par défaut, n'importe laquelle peut être utilisée pour accéder au compte.
Protocol	Sélectionnez le protocole de la connexion à créer.
Use Azure Shared Access Signature	Cochez cette case pour utiliser une signature d'accès partagé, afin d'accéder aux ressources de stockage sans avoir besoin d'une clé associée au compte. Dans le champ <b>Azure Shared Access Signature</b> affiché, saisissez votre signature d'accès partagé, entre guillemets doubles. Pour plus d'informations, consultez <a href="#">Using Shared Access Signatures (SAS)</a> (en anglais).

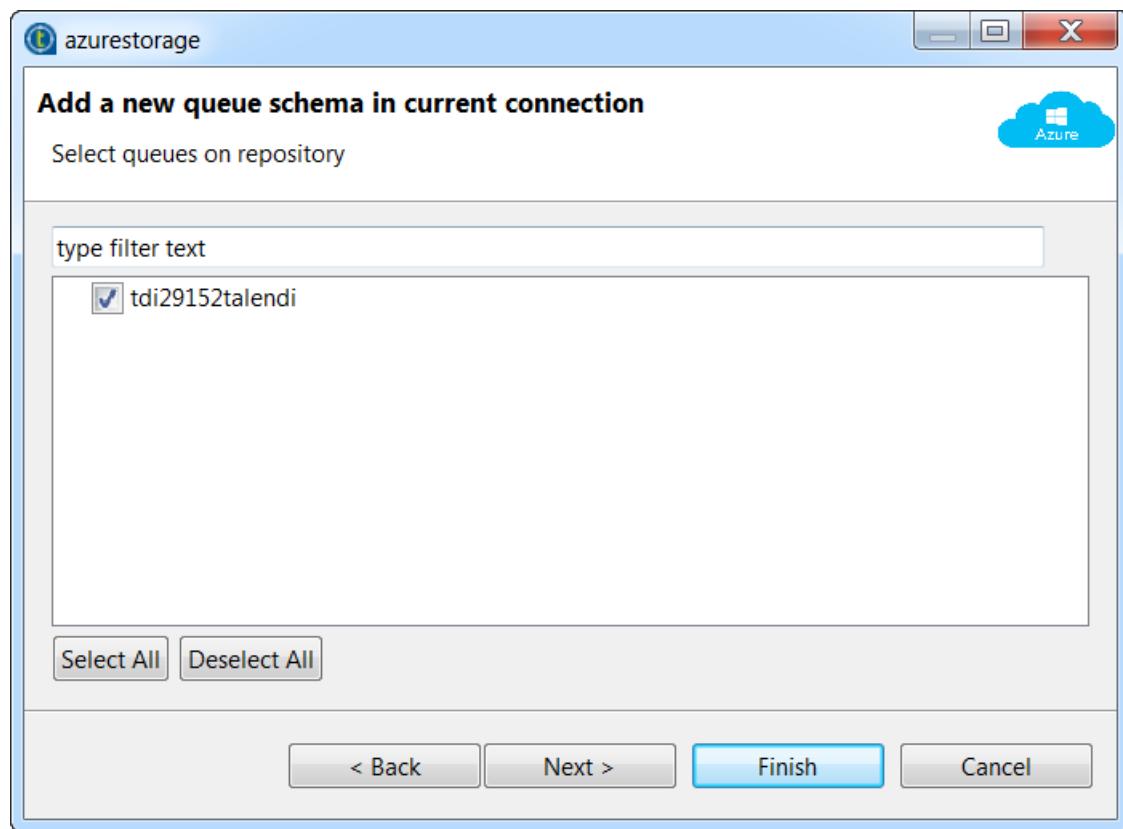
3. Cliquez sur **Test connection** pour vérifier la configuration.

Lorsque les informations de connexion fournies sont correctes, un message indiquant le succès de la vérification de la connexion s'affiche. Cliquez sur **OK** pour fermer la boîte de dialogue. Le bouton **Next** n'est plus grisé.

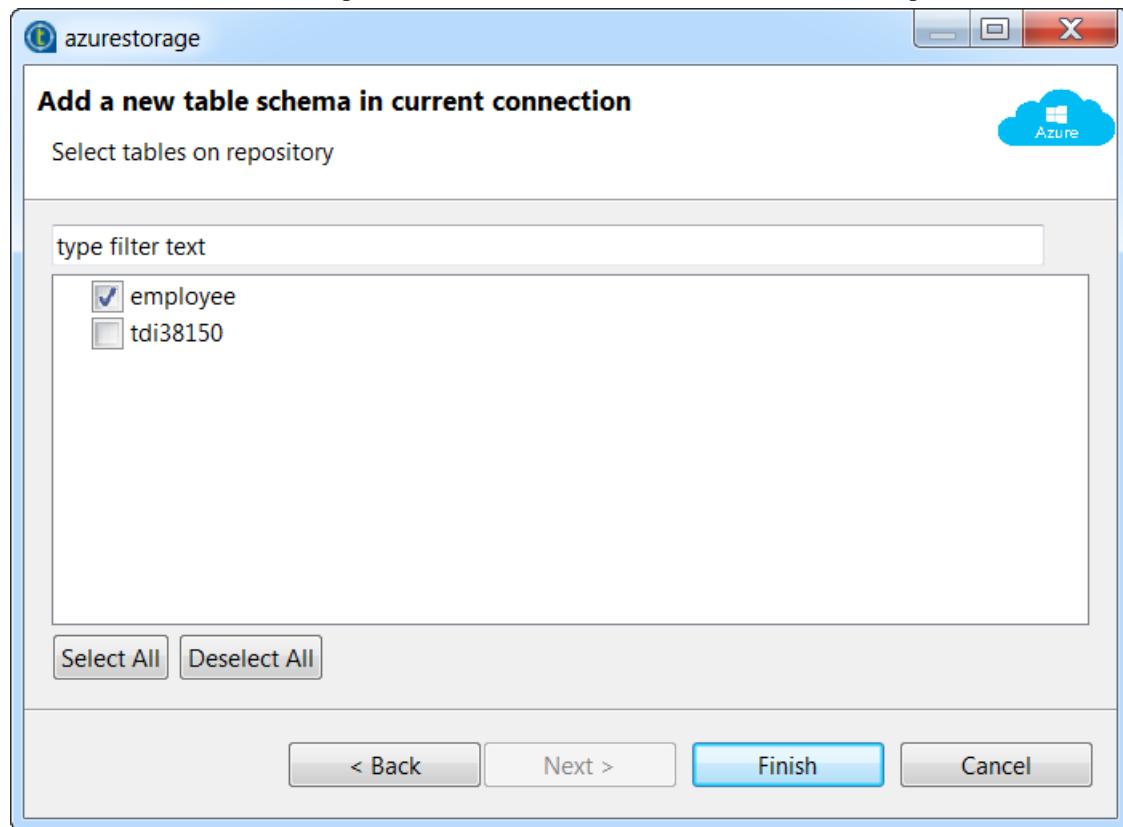
4. Cliquez sur **Next**, puis dans la boîte de dialogue **[Add a new container schema in current connection]** qui s'affiche, cochez les cases correspondant au(x) conteneur(s) dont vous souhaitez récupérer le schéma.



5. Cliquez sur **Next**, puis dans la boîte de dialogue **[Add a new queue schema in current connection]** qui s'affiche, cochez les cases correspondant à la ou aux file(s) dont vous souhaitez récupérer le schéma.

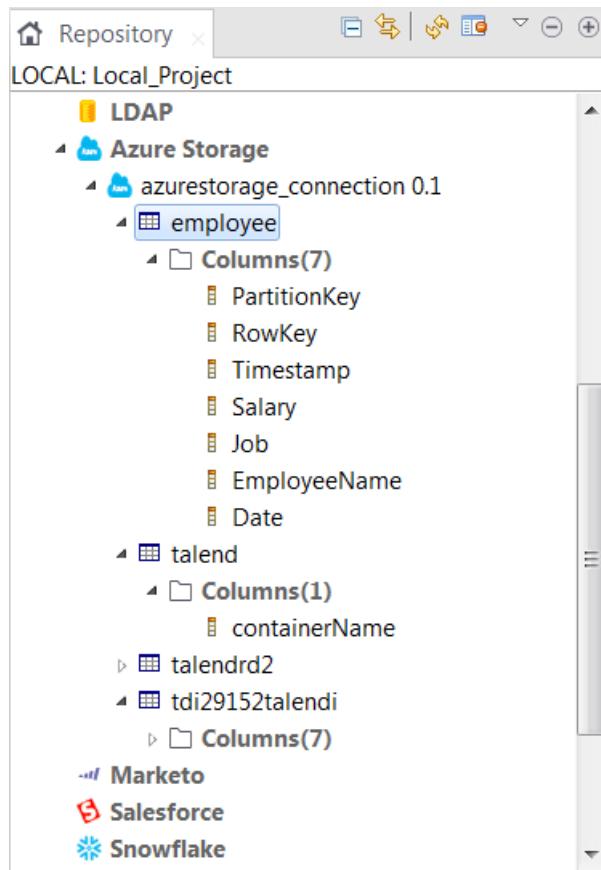


6. Cliquez sur **Next**, puis dans la boîte de dialogue [**Add a new table schema in current connection**] qui s'affiche, cochez les cases correspondant à la ou aux table(s) dont vous souhaitez récupérer le schéma.



7. Cliquez sur **Finish** pour finaliser la procédure.

La nouvelle connexion à Azure Storage créée s'affiche sous le nœud **Azure Storage** du **Repository**, de même que les schémas des conteneurs, files et tables sélectionné(e)s.



Vous pouvez maintenant ajouter un composant Azure Storage dans l'espace de modélisation graphique, en glissant-déposant la connexion à Azure Storage créée ou n'importe quel(le) conteneur/file/table récupéré(e) depuis le **Repository**, afin de réutiliser la connexion et/ou les informations du schéma. Pour plus d'informations sur le glisser-déposer de métadonnées dans l'espace de modélisation graphique, consultez [Comment utiliser les métadonnées centralisées dans un Job](#). Pour plus d'informations sur les composants Azure Storage, consultez la documentation relative aux composants Azure Storage.

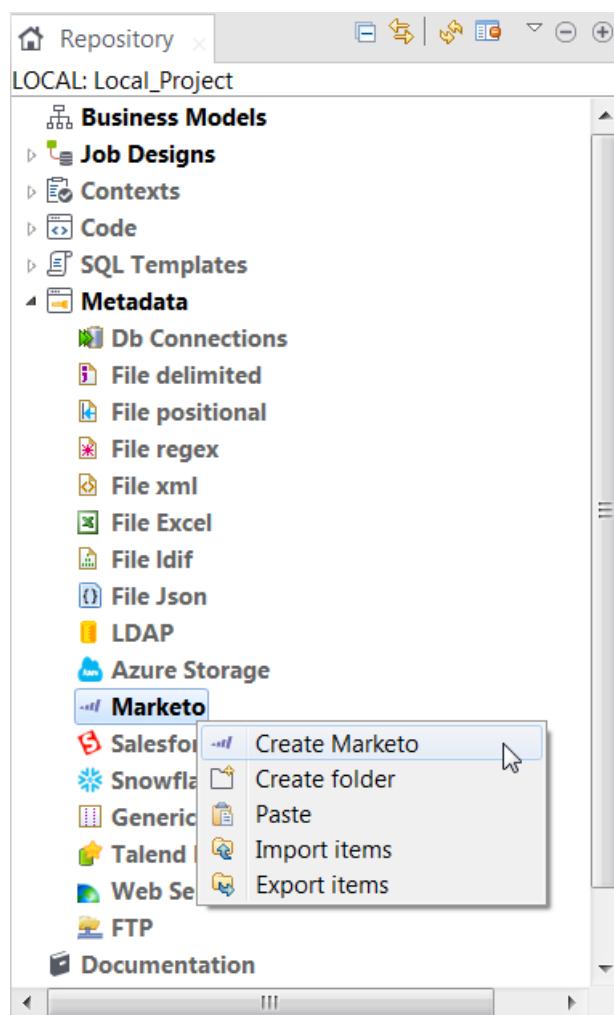
Pour modifier la métadonnée de connexion à Azure Storage créée, cliquez-droit sur le nœud de la connexion dans le **Repository**, puis dans le menu contextuel, sélectionnez **Edit Azure Storage** pour ouvrir l'assistant de configuration de la métadonnée.

Pour modifier le schéma d'un(e) conteneur/file/table, cliquez-droit sur le nœud du conteneur ou de la file/table dans le **Repository**, puis dans le menu contextuel, sélectionnez **Edit Schema** pour ouvrir l'assistant de mise à jour du schéma.

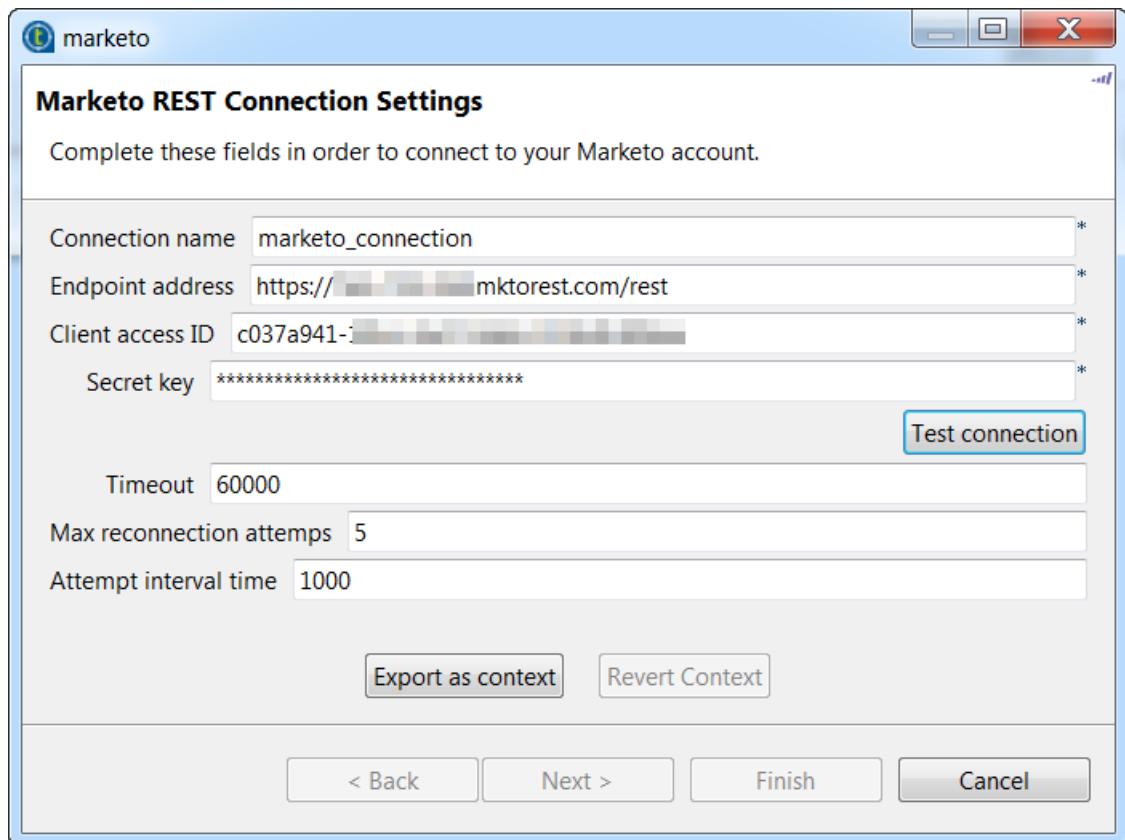
## 10.14. Centraliser les métadonnées Marketo

Vous pouvez utiliser l'assistant de création de métadonnées Marketo fourni par le *Studio Talend* pour configurer rapidement une connexion à Marketo et récupérer le schéma des objets personnalisés via l'API REST.

1. Dans le **Repository**, développez le nœud **Metadata**, puis cliquez-droit sur le nœud **Marketo** dans l'arborescence. Dans le menu contextuel, sélectionnez **Create Marketo** pour ouvrir l'assistant [Marketo].



2. Dans la boîte de dialogue **Marketo REST Connection Settings**, renseignez (ou mettez à jour, si nécessaire) les informations des champs listés dans le tableau suivant.

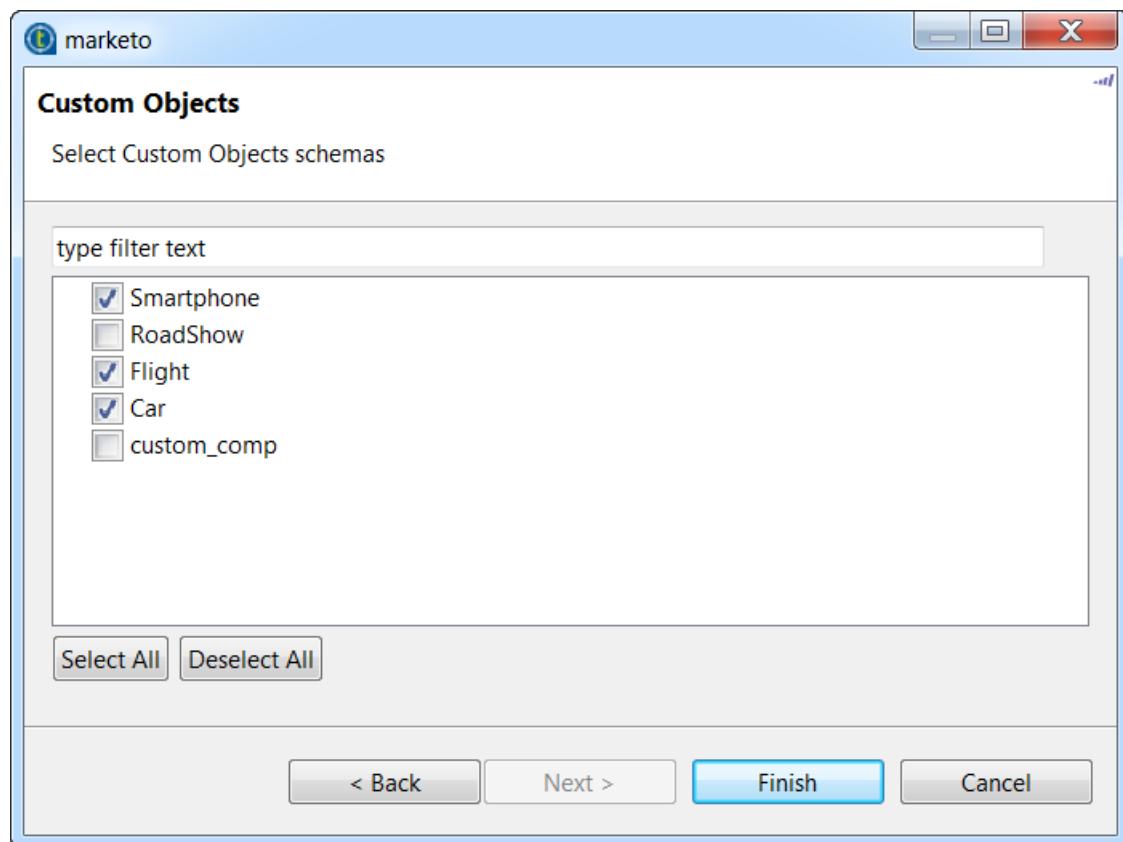


Champ	Description
<b>Connection name</b>	Saisissez le nom de la connexion à créer.
<b>Endpoint address</b>	Saisissez l'URL de l'endpoint de l'API du service Web Marketo. Vous pouvez trouver l'URL de l'endpoint de l'API dans le panneau <b>Admin &gt; Web Services</b> de Marketo.
<b>Client access ID</b>	Saisissez l'identifiant utilisateur permettant d'accéder au service Web Marketo.
<b>Secret key</b>	Saisissez la clé de cryptage (Secret du client) permettant d'accéder au service Web Marketo.
<b>Timeout</b>	Saisissez en millisecondes le délai de connexion au service Web Marketo, avant suspension de la requête.
<b>Max reconnection attempts</b>	Spécifiez le nombre maximal de tentatives de connexion au service Web Marketo, avant d'abandonner.
<b>Attempt interval time</b>	Saisissez en millisecondes le délai entre les tentatives de connexion successives.

3. Cliquez sur **Test connection** pour vérifier la configuration.

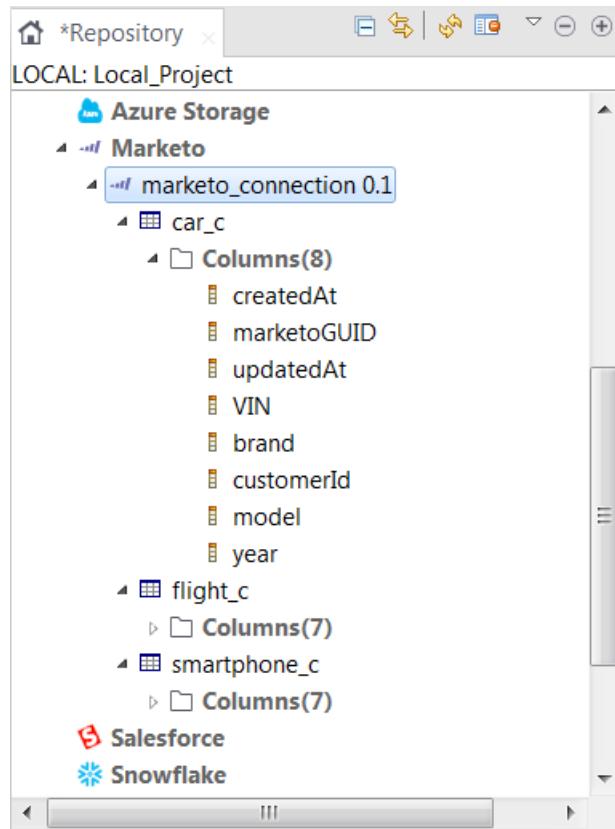
Lorsque les informations de connexion fournies sont correctes, un message indiquant le succès de la vérification de la connexion s'affiche. Cliquez sur **OK** pour fermer la boîte de dialogue. Le bouton **Next** n'est plus grisé.

4. Cliquez sur **Next** pour passer à l'étape suivante afin de sélectionner les objets personnalisés souhaités.



5. Cochez les cases correspondant aux objets personnalisés dont vous souhaitez récupérer le schéma, puis cliquez sur **Finish**.

La nouvelle connexion à Marketo créée s'affiche sous le nœud **Marketo** du **Repository**, de même que les schémas des objets personnalisés sélectionnés.



Vous pouvez maintenant ajouter un composant Marketo dans l'espace de modélisation graphique, en glissant-déposant la connexion à Marketo créée ou n'importe quelle table récupérée depuis le **Repository**, afin de réutiliser la connexion et/ou les informations du schéma. Pour plus d'informations sur le glisser-déposer de métadonnées dans l'espace de modélisation graphique, consultez [Comment utiliser les métadonnées centralisées dans un Job](#). Pour plus d'informations sur les composants Marketo, consultez la documentation relative aux composants Marketo.

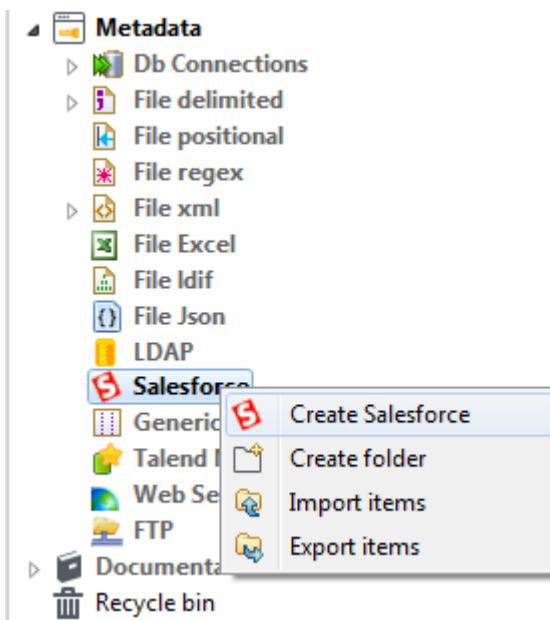
Pour modifier la métadonnée de connexion à Marketo créée, cliquez-droit sur le nœud de la connexion dans le **Repository**, puis dans le menu contextuel, sélectionnez **Edit Marketo** pour ouvrir l'assistant de configuration de la métadonnée.

Pour modifier le schéma d'une table, cliquez-droit sur le nœud de la table dans le **Repository**, puis dans le menu contextuel, sélectionnez **Edit Schema** pour ouvrir l'assistant de mise à jour du schéma.

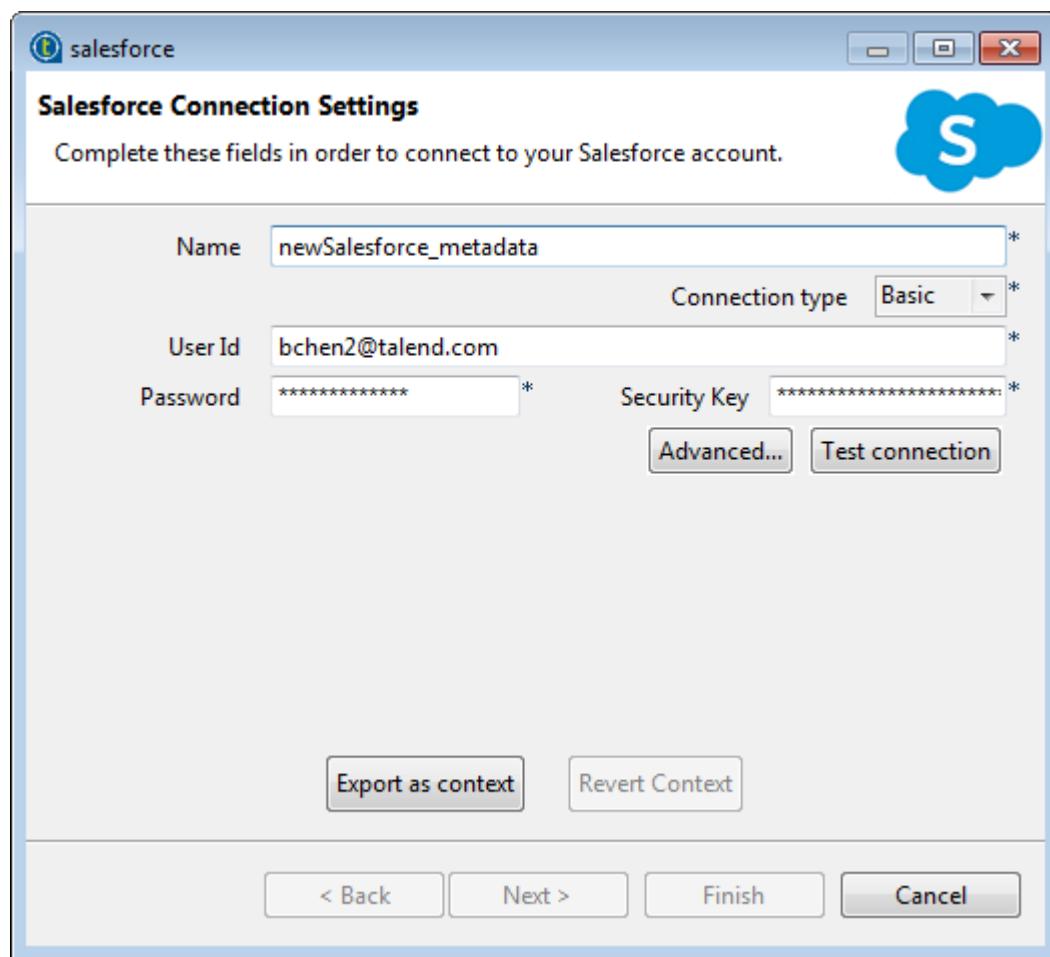
## 10.15. Centraliser les métadonnées d'une connexion Salesforce

Vous pouvez utiliser l'assistant de création de métadonnées Salesforce fourni par le *Studio Talend* pour configurer rapidement une connexion à un système Salesforce et ainsi réutiliser vos métadonnées Salesforce dans plusieurs Jobs.

1. Dans le **Repository**, développez le nœud **Metadata**, cliquez-droit sur le nœud **Salesforce** dans l'arborescence, puis sélectionnez **Create Salesforce** dans le menu contextuel pour ouvrir l'assistant [Salesforce].

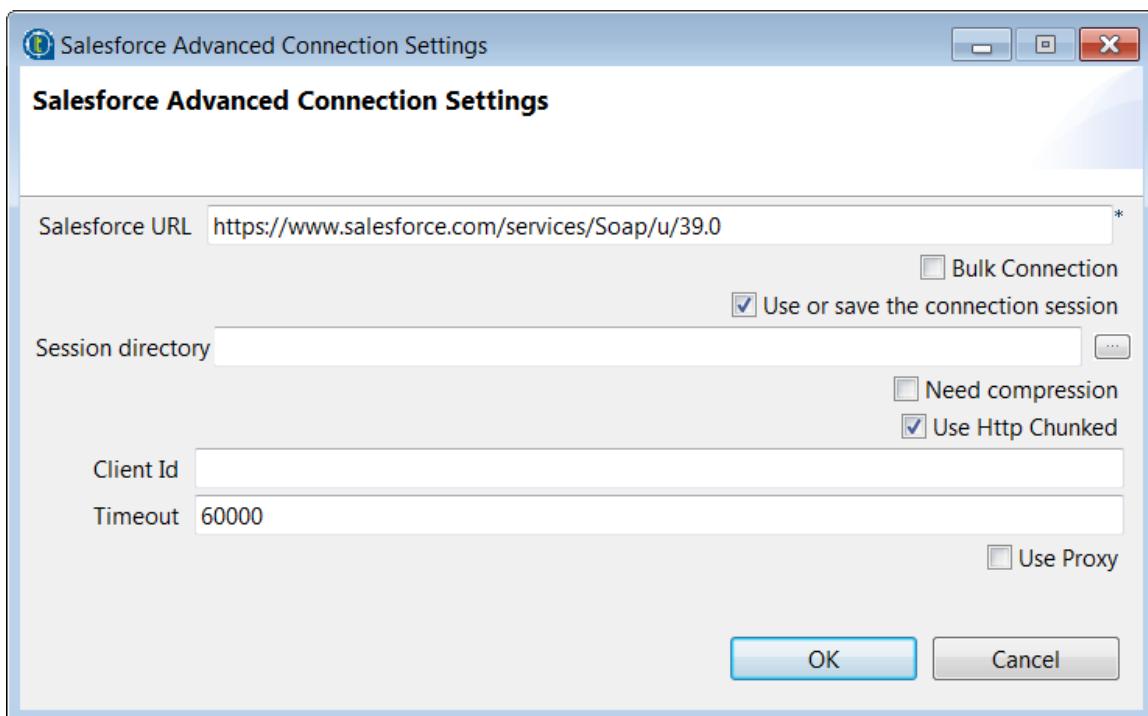


2. Saisissez le nom de votre connexion dans le champ **Name**, sélectionnez **Basic** ou **OAuth** dans la liste **Connection type**, puis renseignez les informations de la connexion selon le type de connexion que vous avez sélectionné.



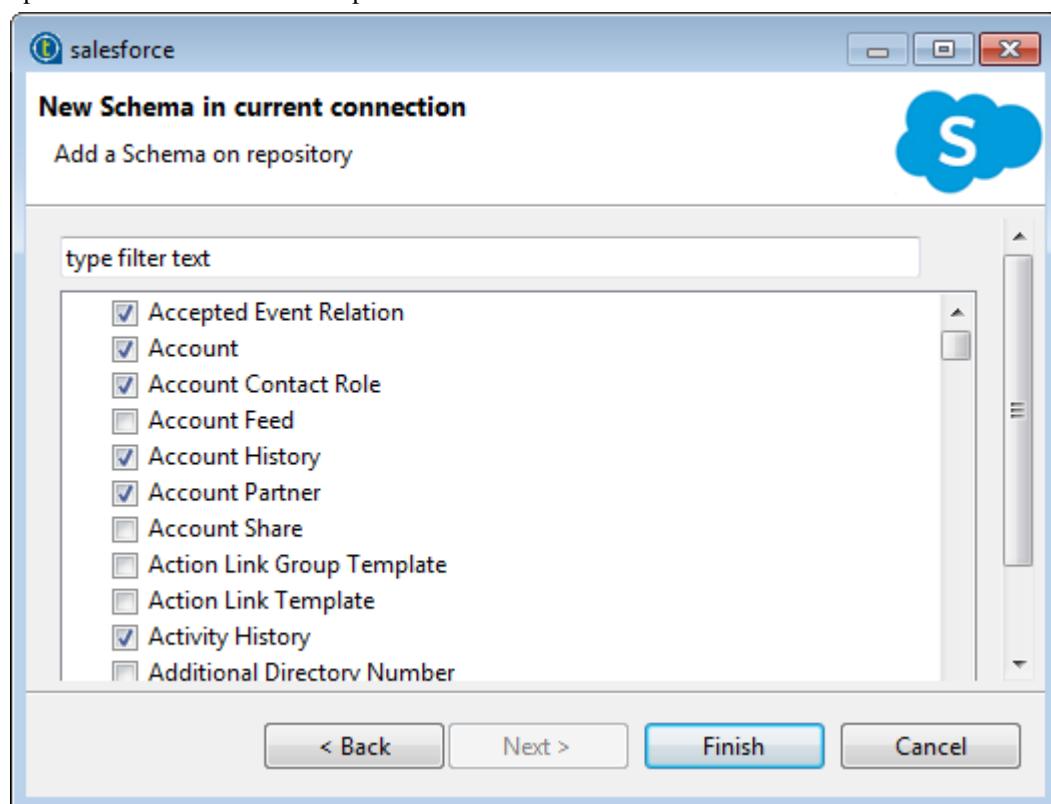
- Si l'option **Basic** est sélectionnée, vous devez spécifier les informations suivantes :
- **User Id** : identifiant de l'utilisateur de Salesforce.

- **Password** : mot de passe associé à l'identifiant de l'utilisateur.
  - **Security Key** : jeton de sécurité.
  - Si l'option **OAuth** est sélectionnée, vous devez spécifier les informations suivantes :
    - **Client Id et Client Secret** : Consumer key/Consumer Secret OAuth, disponibles dans la zone **OAuth Settings** de l'application connectée que vous avez créée sur Salesforce.com.
    - **Callback Host et Callback Port** : URL de rappel pour l'authentification OAuth. Cette URL (l'hôte et le port) est définie durant la création d'une application connectée et sera affichée dans la zone **OAuth Settings** de l'application connectée.
    - **Token File** : chemin d'accès au fichier de jeton stockant le jeton de rafraîchissement utilisé pour obtenir le jeton d'accès sans autorisation.
3. Si nécessaire, cliquez sur **Advanced...** pour ouvrir la boîte de dialogue [**Salesforce Advanced Connection Settings**], effectuez les tâches suivantes puis cliquez sur **OK** :
- saisissez l'URL du Service Web de Salesforce requise pour se connecter au système Salesforce.
  - cochez la case **Bulk Connection** si vous devez utiliser la fonction de traitement des données de masse.
  - cochez la case **Use or save the connection session** et, dans le champ **Session directory** qui s'affiche, spécifiez le chemin d'accès au fichier de sessions de la connexion utilisé ou à sauvegarder.
- Ce fichier de sessions peut être partagé par différents Jobs afin de récupérer une session de connexion tant que l'ID correct de l'utilisateur est fourni par le composant. Ainsi, vous n'avez pas besoin de vous connecter au serveur pour récupérer la session.
- Lorsqu'une session expirée est détectée, si les informations de connexion correctes (ID utilisateur, mot de passe, clé de sécurité) sont fournies, le composant se connecte au serveur afin de récupérer les informations de la nouvelle session et mettre à jour le fichier de sessions de connexion.
- Cette case est disponible uniquement lorsque l'option **Basic** est sélectionnée dans la liste déroulante **Connection type**.
- cochez la case **Need compression** pour activer la compression des messages SOAP, ce qui peut augmenter les niveaux de performance.
  - cochez la case **Trace HTTP message** pour écrire en sortie les interactions HTTP dans la console.
- Cette option est disponible si la case **Bulk Connection** est cochée.
- cochez la case **Use HTTP Chunked** pour utiliser le mécanisme de transfert des données HTTP morcelées.
- Cette option est indisponible lorsque la case **Bulk Connection** est cochée.
- saisissez l'identifiant d'un utilisateur réel dans le champ **Client Id** afin de différencier les utilisateurs d'un même compte et mot de passe pour accéder au site Web de Salesforce.
  - Renseignez le champ **Timeout** avec la valeur de délai d'attente de connexion à Salesforce, en millisecondes.

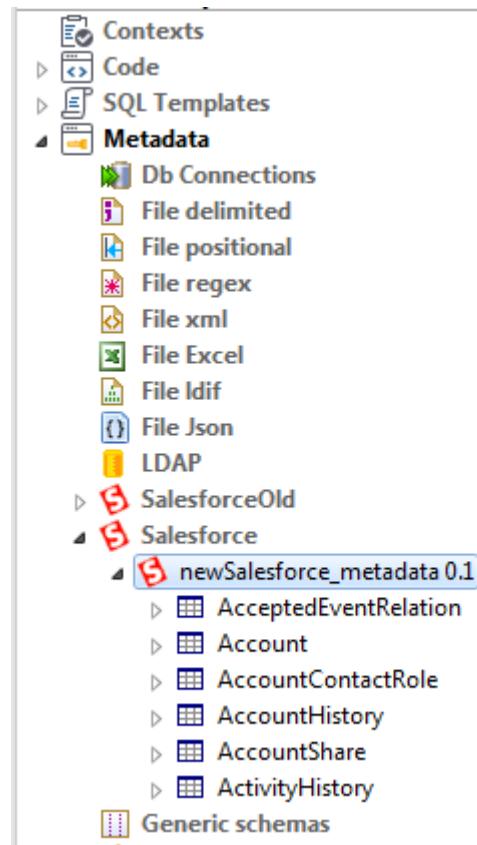


4. Cliquez sur **Test connection** pour vérifier les paramètres de connexion et lorsque le message indiquant le succès de la vérification de la connexion s'affiche, cliquez sur **OK** pour confirmer. Cliquez ensuite sur **Next** pour passer à l'étape suivante afin de sélectionner les modules dont vous souhaitez récupérer le schéma.
5. Cochez les cases correspondant aux modules souhaités puis cliquez sur **Finish** pour récupérer les schémas des modules sélectionnés.

Vous pouvez saisir du texte en filtre pour réduire votre sélection.



La nouvelle connexion à Salesforce créée s'affiche sous le nœud **Salesforce du Repository**, de même que les schémas des modules sélectionnés.



Vous pouvez à présent glisser-déposer la connexion à Salesforce ou n'importe lequel de ses schémas depuis le **Repository** dans l'espace de modélisation graphique. Dans la boîte de dialogue qui s'ouvre, vous pouvez choisir un composant Salesforce à utiliser dans votre Job. Vous pouvez également déposer la connexion à Salesforce ou un de ses schémas sur un composant existant pour réutiliser les informations de connexion ou des métadonnées dans ce composant. Pour plus d'informations sur le glisser-déposer de métadonnées dans l'espace de modélisation graphique, consultez [Comment utiliser les métadonnées centralisées dans un Job](#).

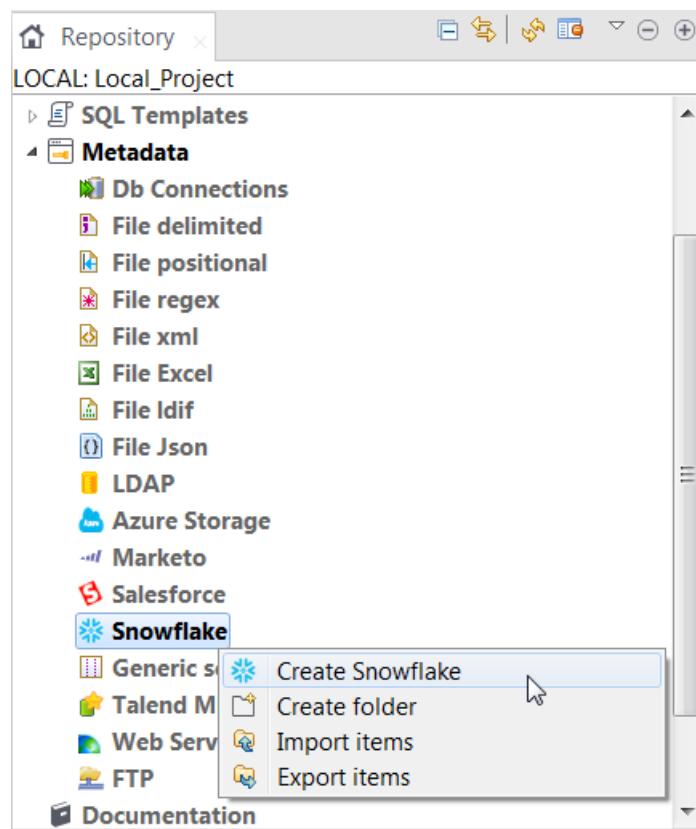
Pour modifier la métadonnée Salesforce, cliquez-droit sur le schéma dans le **Repository**, puis sélectionnez **Edit Salesforce** pour ouvrir l'assistant de configuration de la métadonnée du fichier.

Pour modifier un schéma Salesforce existant, cliquez-droit sur le schéma dans le **Repository** et sélectionnez **Edit Schema** dans le menu contextuel.

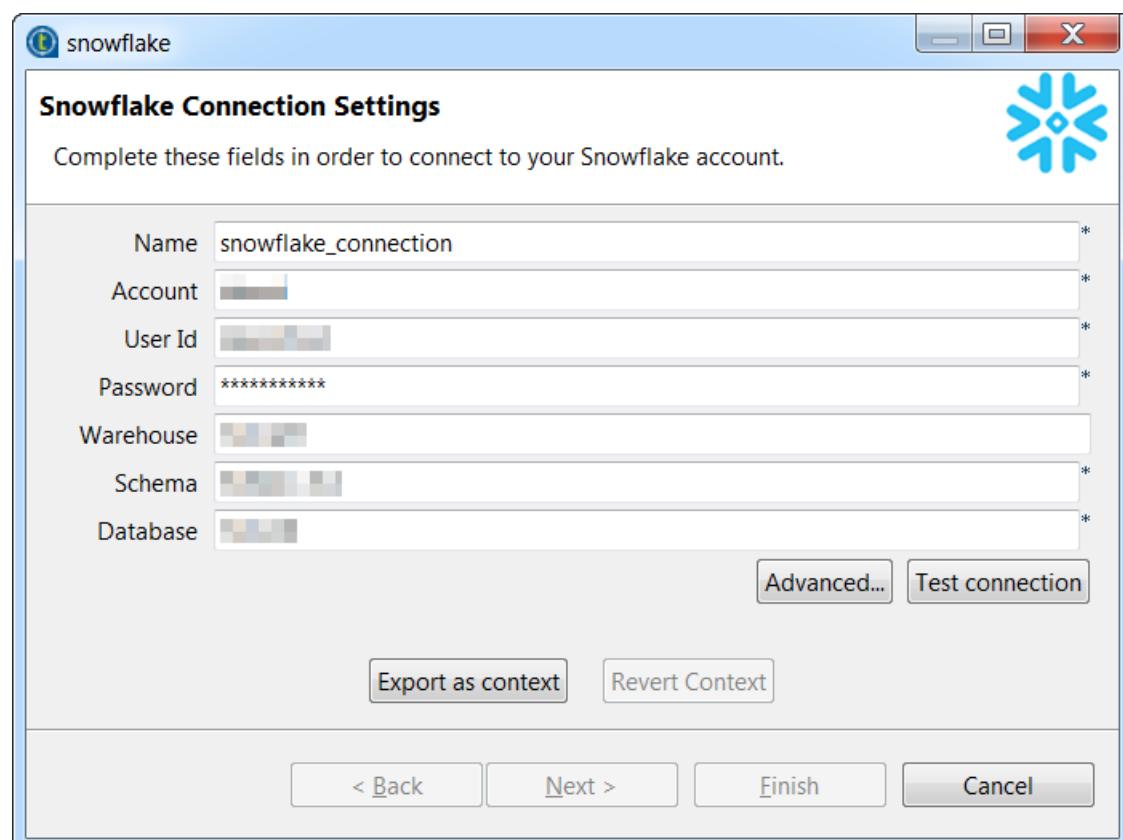
## 10.16. Centraliser les métadonnées Snowflake

Vous pouvez utiliser l'assistant de création de métadonnées Snowflake fourni par le *Studio Talend* pour configurer rapidement une connexion à Snowflake et récupérer le schéma des tables souhaitées.

1. Dans le **Repository**, développez le nœud **Metadata**, puis cliquez-droit sur le nœud **Snowflake** dans l'arborescence. Dans le menu contextuel, sélectionnez **Create Snowflake** pour ouvrir l'assistant [Snowflake].

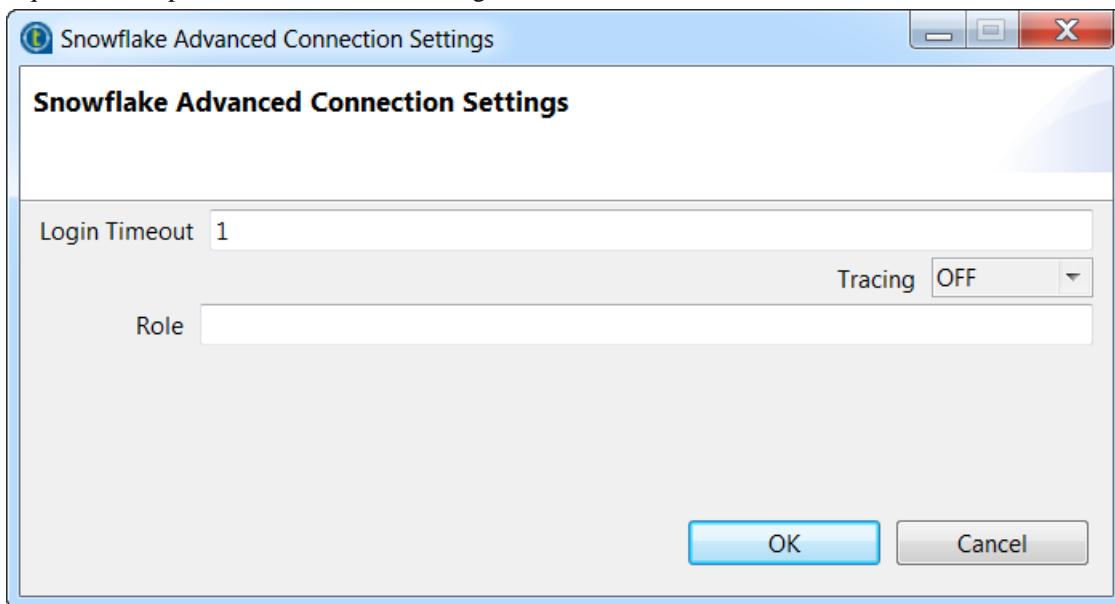


2. Dans la boîte de dialogue **Snowflake Connection Settings**, renseignez les informations des champs listés dans le tableau suivant.



Champ	Description
Name	Saisissez le nom de la connexion à créer.
Account	Saisissez le nom du compte qui vous a été assigné par Snowflake.
User Id	Saisissez votre identifiant défini dans Snowflake via le paramètre <i>LOGIN_NAME</i> de Snowflake. Pour plus d'informations, contactez l'administrateur de votre système Snowflake.
Password	Saisissez le mot de passe associé à l'identifiant de l'utilisateur.
Warehouse	Saisissez le nom de l'entrepôt Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.
Schema	Saisissez le nom du schéma de la base de données à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.
Database	Saisissez le nom de la base de données Snowflake à utiliser. Ce nom est sensible à la casse et est normalement en lettres capitales dans Snowflake.

3. Cliquez sur **Advanced...**, puis dans la boîte de dialogue [**Snowflake Advanced Connection Settings**] qui s'affiche, renseignez ou mettez à jour les valeurs des propriétés avancées listées dans le tableau suivant. Cliquez sur **OK** pour fermer la boîte de dialogue.

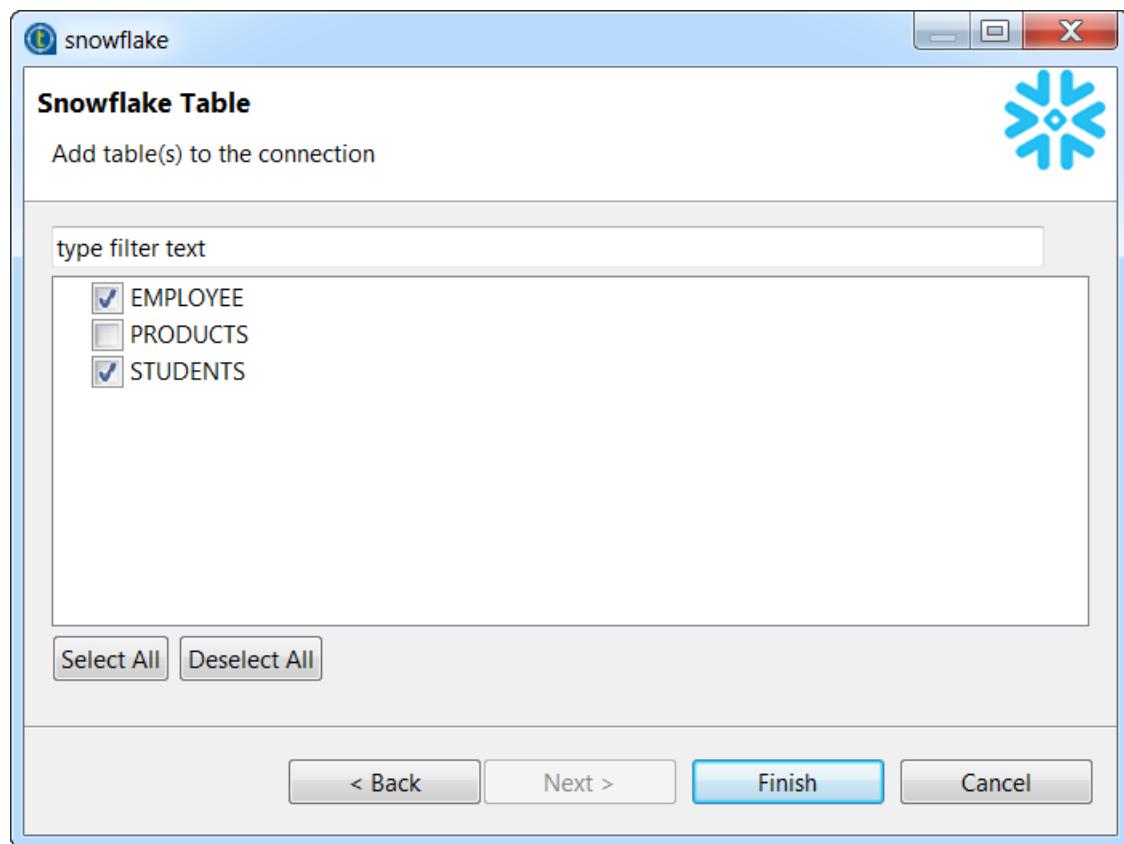


Propriété	Description
Login Timeout	Spécifiez le délai d'attente d'une réponse lors de la connexion à Snowflake, avant de retourner une erreur.
Tracing	Sélectionnez le niveau de log pour le pilote JDBC de Snowflake. Si cette option est activée, un log standard Java est généré.
Role	Saisissez le rôle de contrôle des accès par défaut à utiliser pour initialiser la session Snowflake.  Ce rôle doit déjà exister et doit avoir été assigné à l'ID de l'utilisateur que vous utilisez pour vous connecter à Snowflake. Si vous laissez ce champ vide, le rôle PUBLIC est automatiquement assigné. Pour plus d'informations concernant le modèle de contrôle des accès de Snowflake, consultez la documentation Snowflake à l'adresse suivante <a href="#">Understanding the Access Control Model</a> (en anglais).

4. Cliquez sur **Test connection** pour vérifier la configuration.

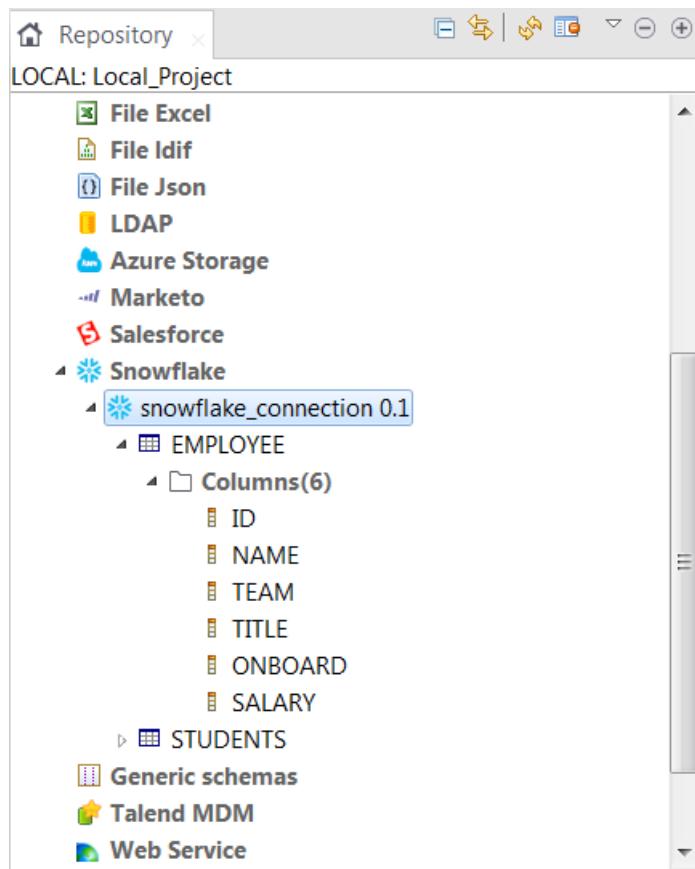
Lorsque les informations de connexion fournies sont correctes, un message indiquant le succès de la vérification de la connexion s'affiche. Cliquez sur **OK** pour fermer la boîte de dialogue. Le bouton **Next** n'est plus grisé.

5. Cliquez sur **Next** pour passer à l'étape suivante afin de sélectionner les tables souhaitées.



6. Cochez les cases correspondant aux tables dont vous souhaitez récupérer le schéma, puis cliquez sur **Finish**.

La nouvelle connexion à Snowflake créée s'affiche sous le nœud **Snowflake du Repository**, de même que les schémas des tables sélectionnées.



Vous pouvez maintenant ajouter un composant Snowflake dans l'espace de modélisation graphique, en glissant-déposant la connexion à Snowflake créée ou n'importe quelle table récupérée depuis le **Repository**, afin de réutiliser la connexion et/ou les informations du schéma. Pour plus d'informations sur le glisser-déposer de métadonnées dans l'espace de modélisation graphique, consultez [Comment utiliser les métadonnées centralisées dans un Job](#). Pour plus d'informations sur les composants Snowflake, consultez la documentation relative aux composants Snowflake.

Pour modifier la métadonnée de connexion à Snowflake créée, cliquez-droit sur le nœud de la connexion dans le **Repository**, puis dans le menu contextuel, sélectionnez **Edit Snowflake** pour ouvrir l'assistant de configuration de la métadonnée.

Pour modifier le schéma d'une table, cliquez-droit sur le nœud de la table dans le **Repository**, puis dans le menu contextuel, sélectionnez **Edit Schema** pour ouvrir l'assistant de mise à jour du schéma.

## 10.17. Configurer un schéma générique

Le *Studio Talend* vous permet de créer un schéma afin de l'utiliser dans vos Jobs, si aucun des assistants de métadonnées ne correspond à vos besoins ou si vous n'avez pas de fichier source duquel prendre le schéma.

Vous pouvez créer un schéma générique :

- à partir de rien. Pour plus de détails, consultez [Configurer un schéma générique à partir de rien](#),
- à partir d'un fichier XML de définition de schéma. Pour plus de détails, consultez [Configurer un schéma générique à partir d'un fichier XML](#),
- et à partir du schéma défini dans un composant. Pour plus de détails, consultez [Enregistrer le schéma d'un composant en tant que schéma générique](#).

Pour utiliser le schéma générique sur un composant, utilisez l'une des méthodes suivantes :

- Sélectionnez **Repository** dans la liste **Schema** de la vue **Basic settings** du composant.

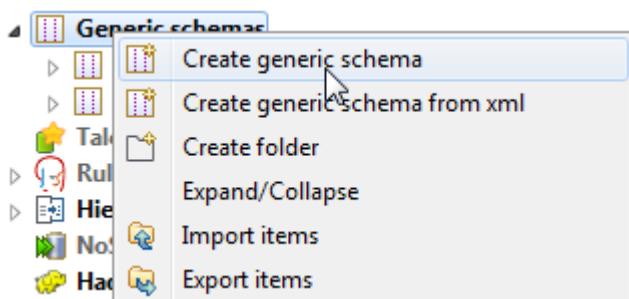
Cliquez sur le bouton [...] pour ouvrir la boîte de dialogue [**Repository Content**] et sélectionnez le schéma correspondant sous le nœud **Generic schemas**. Cliquez sur **OK**.

- Sélectionnez le nœud **metadata** du schéma générique depuis le **Repository** et déposez-le sur un composant.

## 10.17.1. Configurer un schéma générique à partir de rien

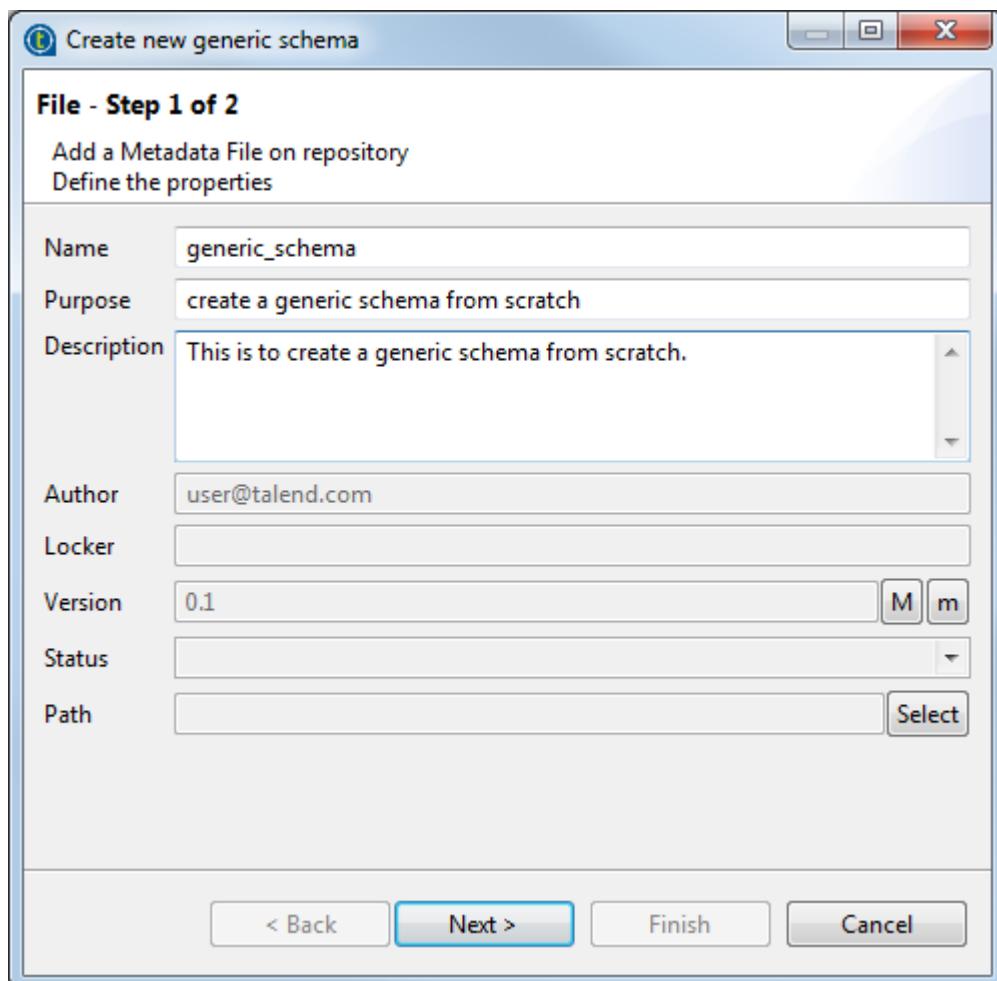
Pour créer un schéma générique à partir de rien, procédez comme suit :

1. Sous le nœud **Metadata**, cliquez-droit sur **Generic schemas** dans la vue **Repository** et sélectionnez **Create generic schema**.



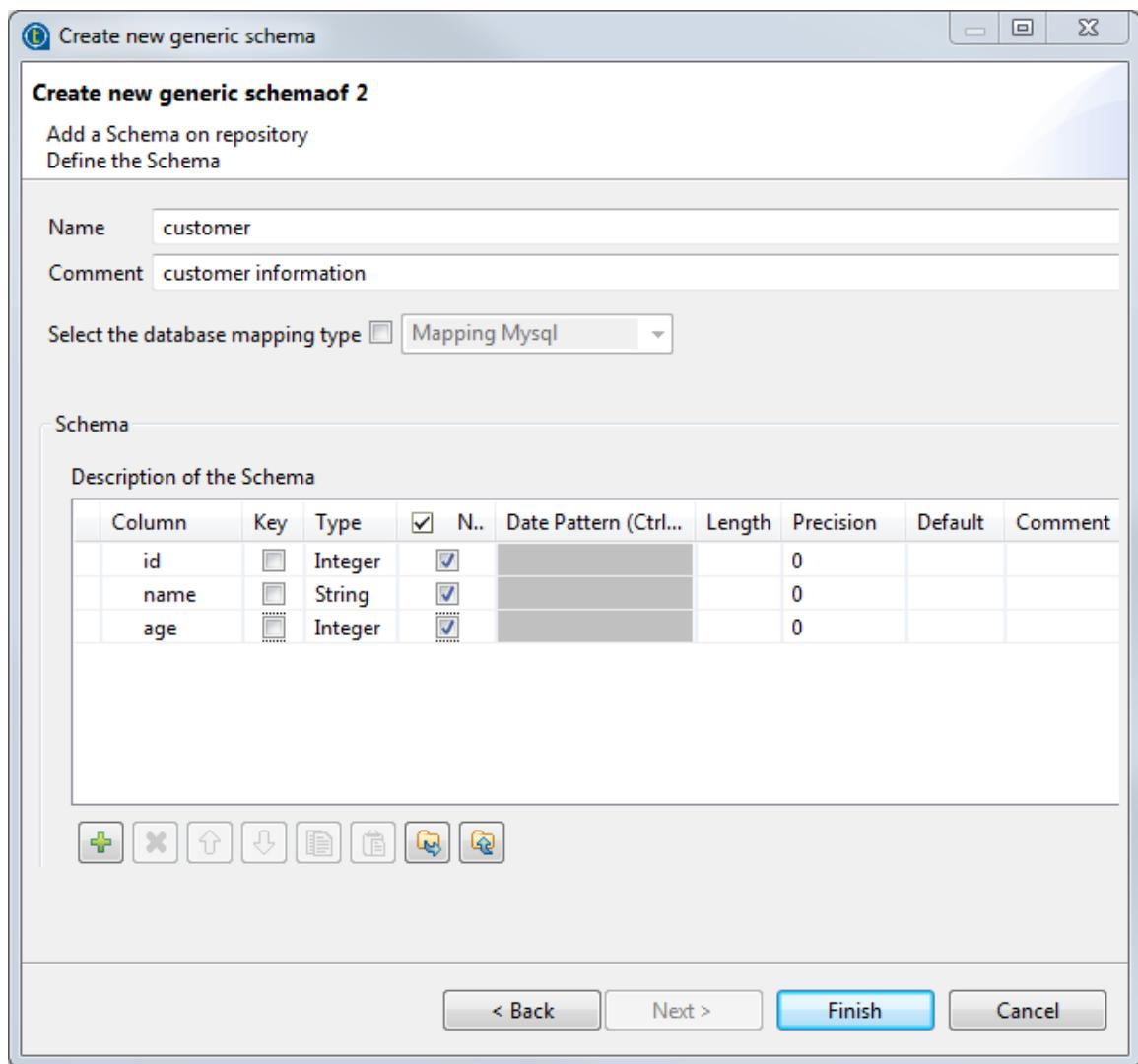
2. Dans l'assistant de création de schéma qui apparaît, renseignez les propriétés génériques du schéma, comme son nom (dans le champ **Name**) et sa **Description**. Le champ **Status** (pour le statut) est un champ personnalisable. Pour plus d'informations concernant ce champ, consultez [Définir les paramètres des Statuts](#).

Cliquez sur **Next** pour continuer.



3. Nommez votre schéma ou utilisez le nom par défaut (metadata) et ajoutez un commentaire, si nécessaire. Personnalisez la structure du schéma dans le panneau **Schema** selon vos besoins.

La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma. Vous pouvez également exporter le schéma courant en tant que fichier XML ou importer un schéma depuis un fichier XML. Il doit être un schéma exporté depuis le Studio, afin de remplacer le schéma courant.



Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

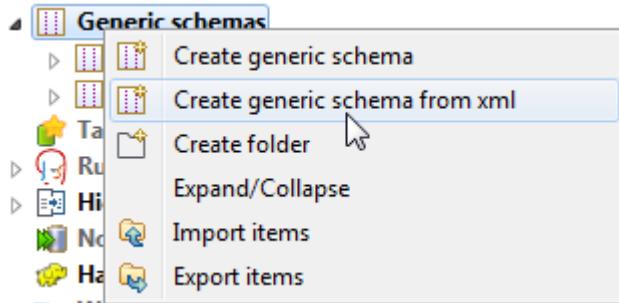
4. Cliquez sur **Finish** pour terminer la création du schéma générique. Le schéma créé s'affiche sous le noeud **Generic schemas**.

## 10.17.2. Configurer un schéma générique à partir d'un fichier XML

 Le fichier XML source à partir duquel vous pouvez créer un schéma générique doit être un schéma exporté depuis le Studio ou depuis un fichier XML ayant la même structure. Il ne peut cependant être exporté depuis un autre type de fichier XML.

Pour créer un schéma générique à partir d'un fichier source XML, procédez comme suit :

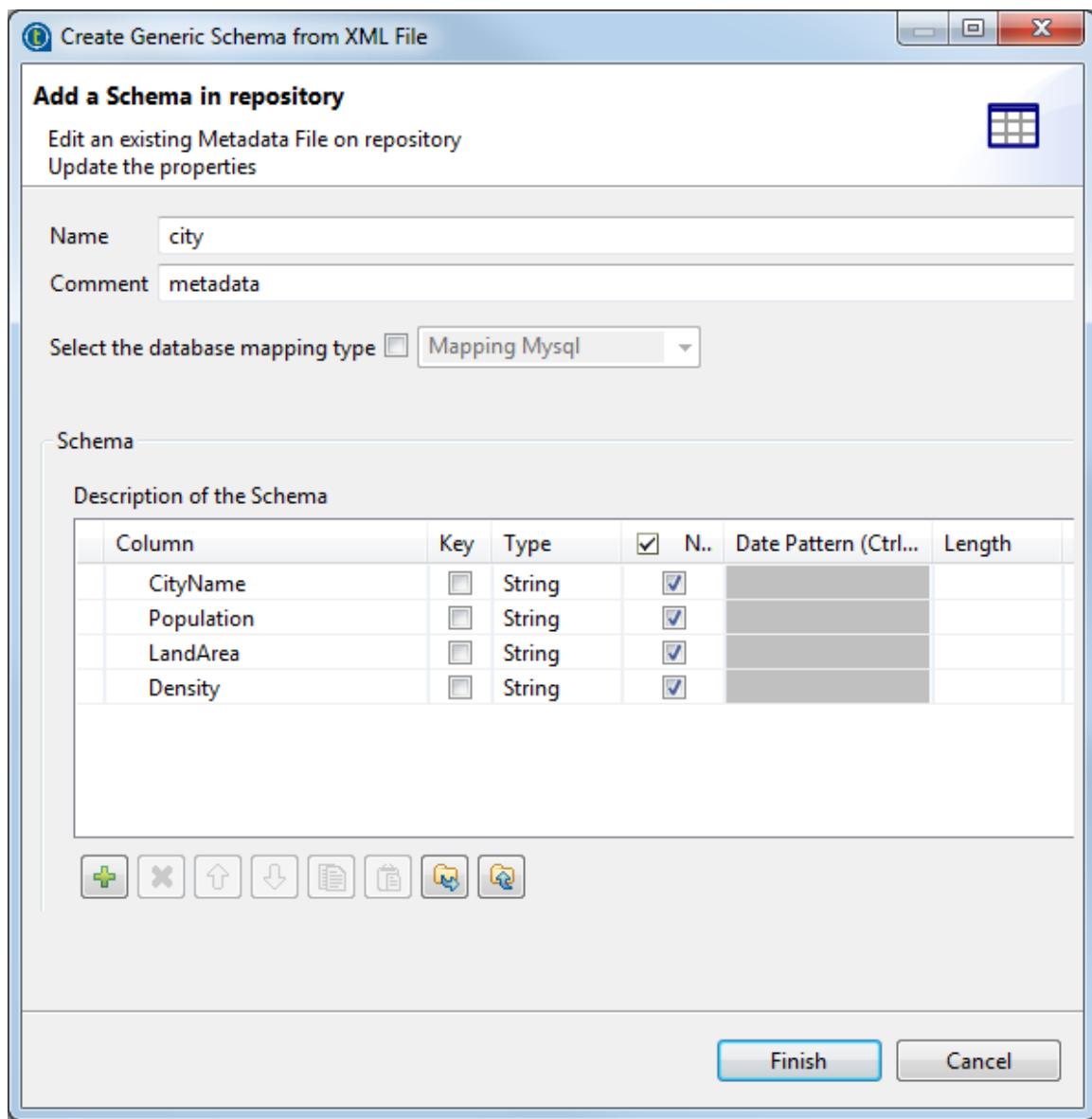
1. Cliquez-droit sur **Generic schemas** dans la vue **Repository** et sélectionnez **Create generic schema from xml**.



2. Dans la boîte de dialogue qui apparaît, sélectionnez le fichier source XML duquel prendre le schéma et cliquez sur **Open**.
3. Dans l'assistant de création de schéma qui apparaît, renseignez le champ **Name** en saisissant un nom pour le schéma ou utilisez celui par défaut (metadata) et saisissez un commentaire, dans le champ **Comment**, si nécessaire.

La structure du schéma du fichier source s'affiche dans le panneau **Schema**. Vous pouvez personnaliser les colonnes du schéma selon vos besoins.

La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes de votre schéma. Vous pouvez également exporter le schéma courant en tant que fichier XML ou importer un schéma depuis un fichier XML. Il doit être un schéma exporté depuis le Studio, afin de remplacer le schéma courant.



Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

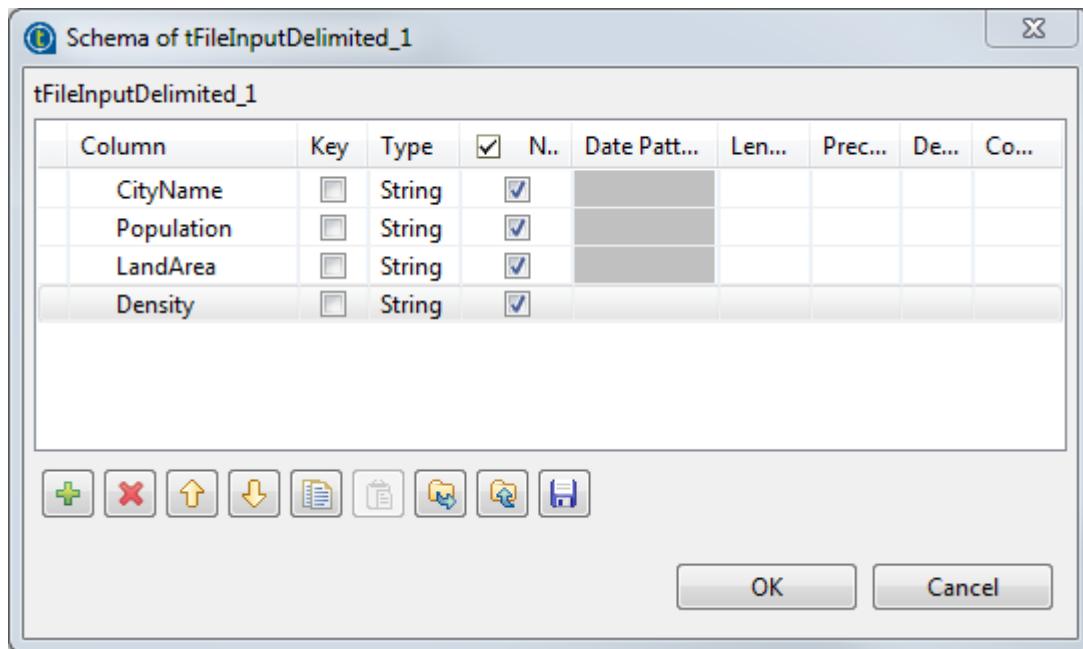
- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

4. Cliquez sur **Finish** pour terminer la création du schéma générique. Le schéma créé s'affiche sous le nœud **Generic schemas**.

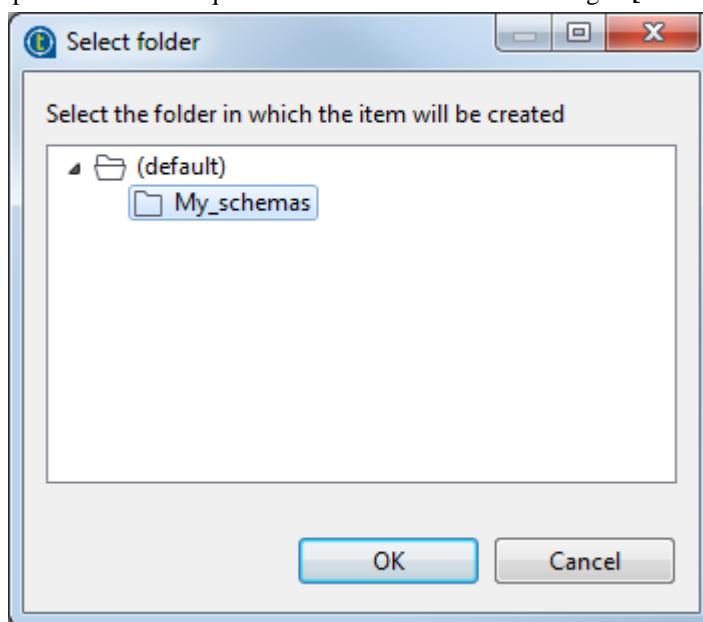
### 10.17.3. Enregistrer le schéma d'un composant en tant que schéma générique

Vous pouvez créer un schéma générique en enregistrant le schéma défini pour un composant. Pour ce faire, procédez comme suit :

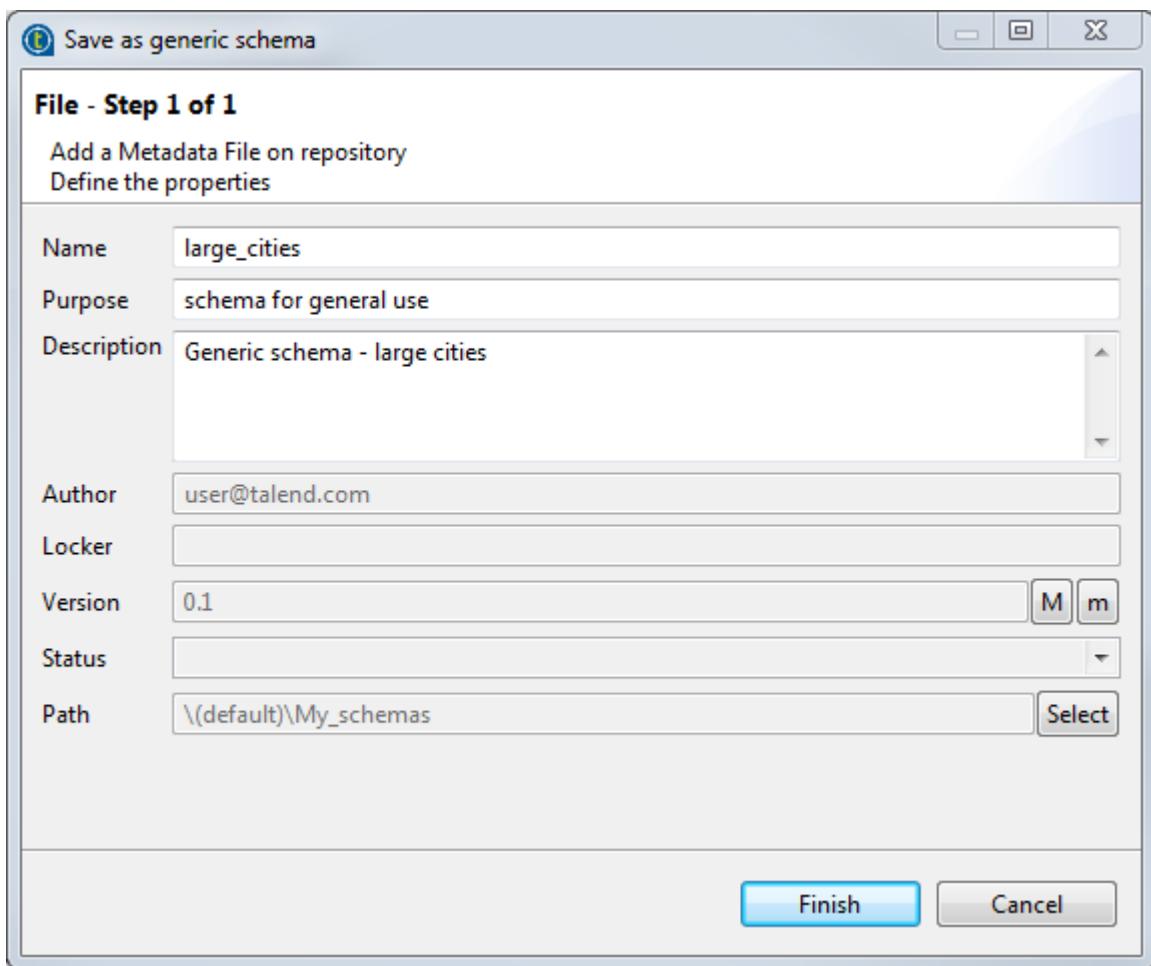
- Ouvrez la vue **Basic settings** du composant ayant le schéma à partir duquel vous souhaitez créer un schéma générique. Cliquez sur le bouton [...] à côté du champ **Edit schema** afin d'ouvrir la boîte de dialogue **[Schema]**.



- Cliquez sur l'icône représentant une disquette afin d'ouvrir la boîte de dialogue **[Select folder]**.



- Si nécessaire, sélectionnez un dossier puis cliquez sur **OK** afin de fermer la boîte dialogue et ouvrir l'assistant de création **[Save as generic schema]**.



- Renseignez le champ **Name** (obligatoire) ainsi que les autres champs (si nécessaire). Cliquez sur **Finish** pour sauvegarder le schéma. Fermez ensuite la boîte de dialogue **[Schema]** ouverte depuis la vue **Basic settings** du composant.

Le schéma est sauvegardé dans le dossier sélectionné sous le nœud **Generic schemas** dans le **Repository**.

## 10.18. Centraliser les métadonnées MDM

Le *Studio Talend* vous permet de centraliser les informations d'une ou plusieurs connexion(s) MDM sous le nœud **Metadata** de la vue **Repository**. Vous pouvez utiliser l'une de ces connexions établies pour vous connecter au serveur MDM.



Vous pouvez également paramétriser une connexion MDM en cliquant sur l'icône de la vue **Basic settings** des composants **tMDMInput** et **tMDMOutput**. Pour plus d'informations, consultez le **tMDMInput** et le **tMDMOutput** à l'adresse <https://help.talend.com>.

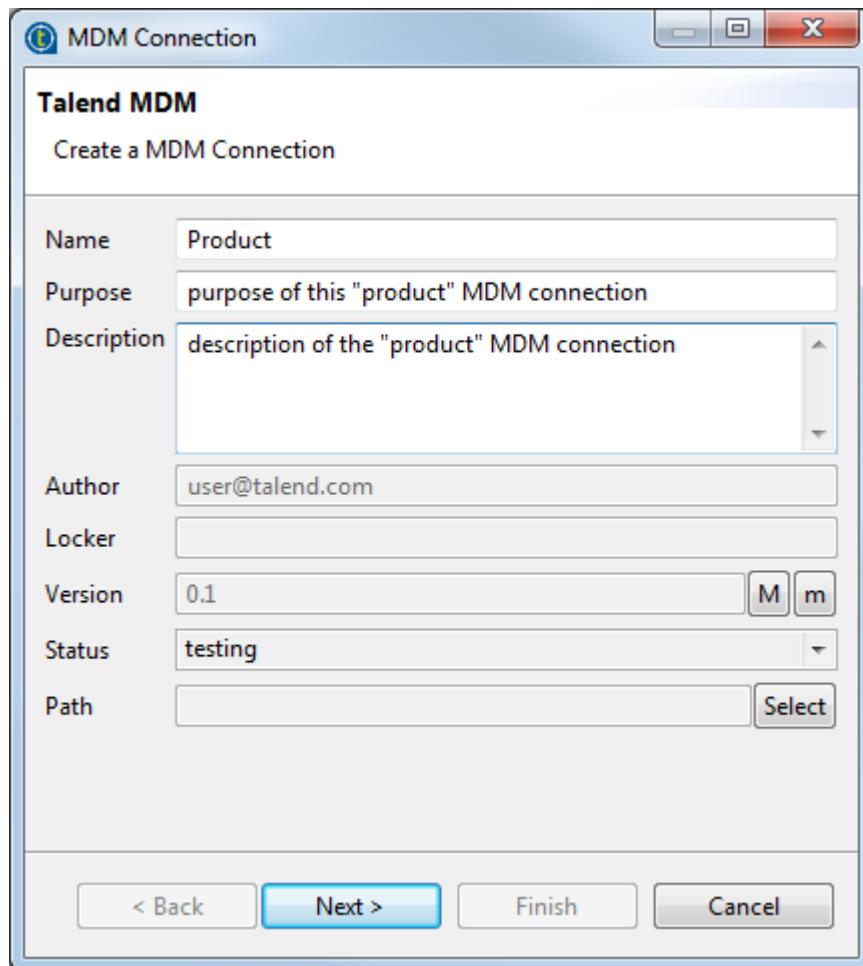
Selon l'option que vous sélectionnez, l'assistant vous permet de créer soit un schéma **XML Input** (d'entrée), soit un schéma **XML Output** (de sortie), soit un schéma **XML Receive** (de réception). Plus tard, dans un Job **Talend**, le composant **tMDMInput** utilise le schéma d'entrée défini pour 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.

## 10.18.1. Définir la connexion

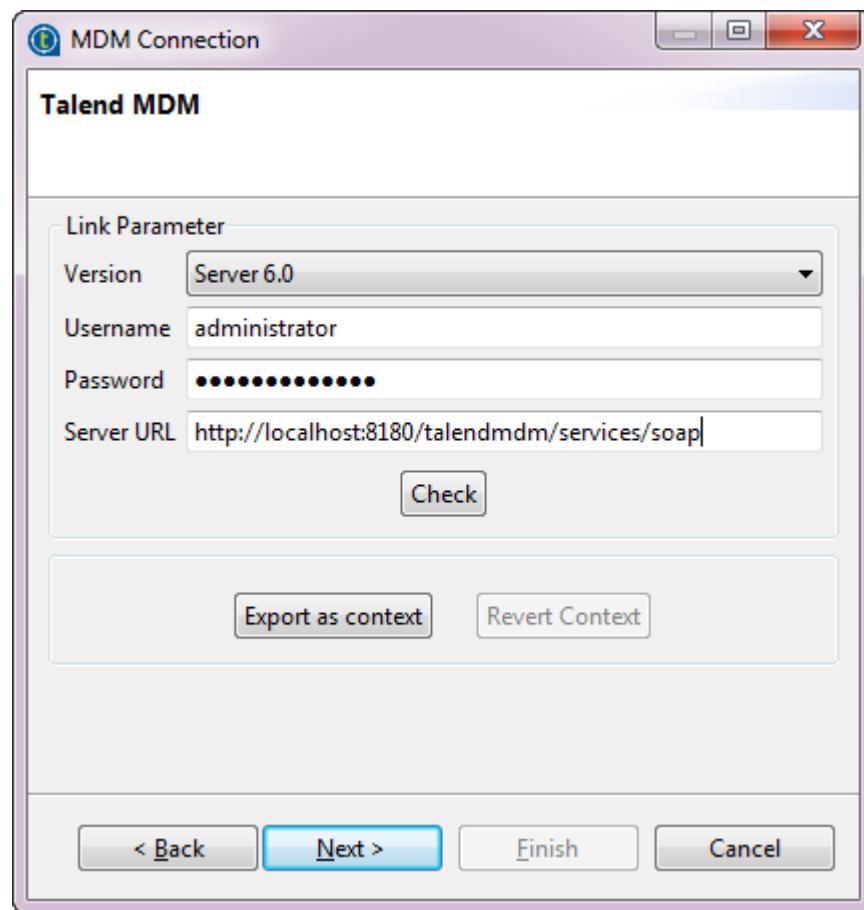
Pour établir une connexion MDM, procédez comme suit :

1. Dans la vue **Repository**, développez le nœud **Metadata** et cliquez-droit sur **Talend MDM**.
2. Sélectionnez **Create MDM Connection** dans le menu contextuel.

L'assistant de connexion s'affiche.



3. Renseignez les champs **Name**, **Purpose** et **Description**. Le champ **Status** est un champ personnalisé pouvant être défini. Pour plus d'informations, consultez [Définir les paramètres des Statuts](#).
4. Cliquez sur **Next** afin de passer à l'étape suivante.



5. Dans la liste **Version**, sélectionnez la version du serveur MDM à laquelle vous connecter.



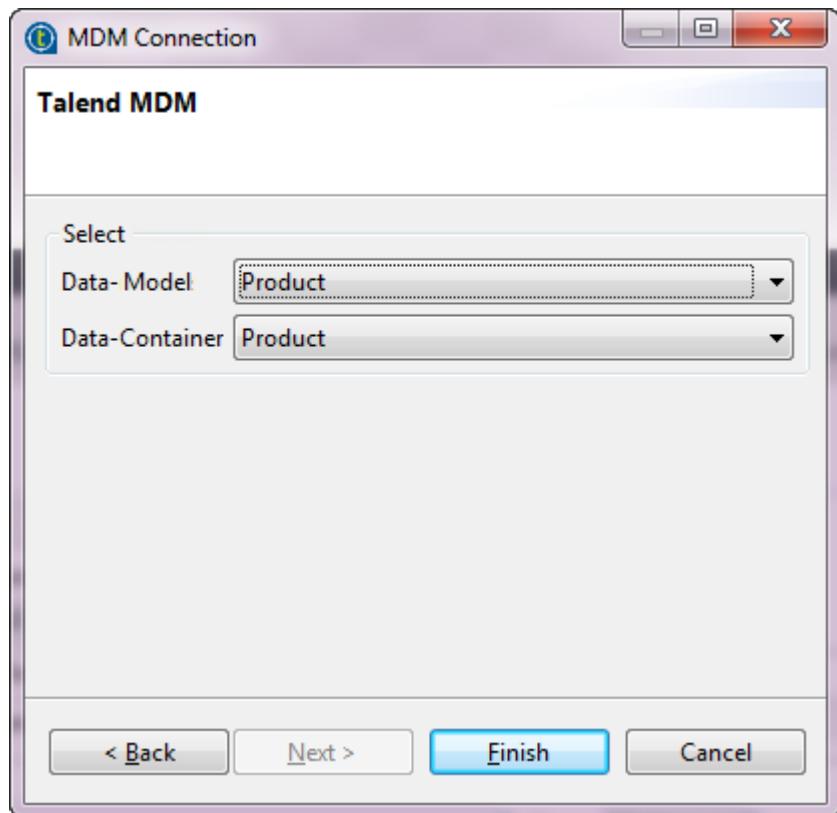
La valeur par défaut dans le champ **Server URL** varie selon l'option sélectionnée dans la liste **Version**.

6. Remplissez les informations de connexion au serveur MDM, y compris les informations d'authentification, puis cliquez sur **Check** afin de vérifier la connexion que vous avez créée.

Une boîte de dialogue s'ouvre pour confirmer que votre connexion a été créée avec succès. Cliquez sur **OK** pour fermer la boîte de dialogue.

Si nécessaire, vous pouvez cliquer sur **Export as context** pour exporter les paramètres de connexion **Talend MDM** dans un nouveau groupe de contextes dans le **Repository** ou réutiliser des variables d'un groupe de contextes pour paramétriser la connexion de la métadonnée. Pour plus d'informations, consultez [Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion](#).

7. Cliquez sur **Next** afin de passer à l'étape suivante.



8. Dans la liste **Data-Model**, sélectionnez le modèle de données par rapport auquel vous souhaitez valider les données maître.
9. Dans la liste **Data-Container**, sélectionnez le conteneur de données où sont stockées les données maître auxquelles vous souhaitez accéder.
10. Cliquez sur **Finish** afin de valider vos changements et fermer la boîte de dialogue.

La connexion nouvellement créée s'affiche sous le nœud **Talend MDM** dans le dossier **Metadata** dans la vue **Repository**.

Vous devez maintenant récupérer le schéma XML des entités métier liées à cette connexion MDM.

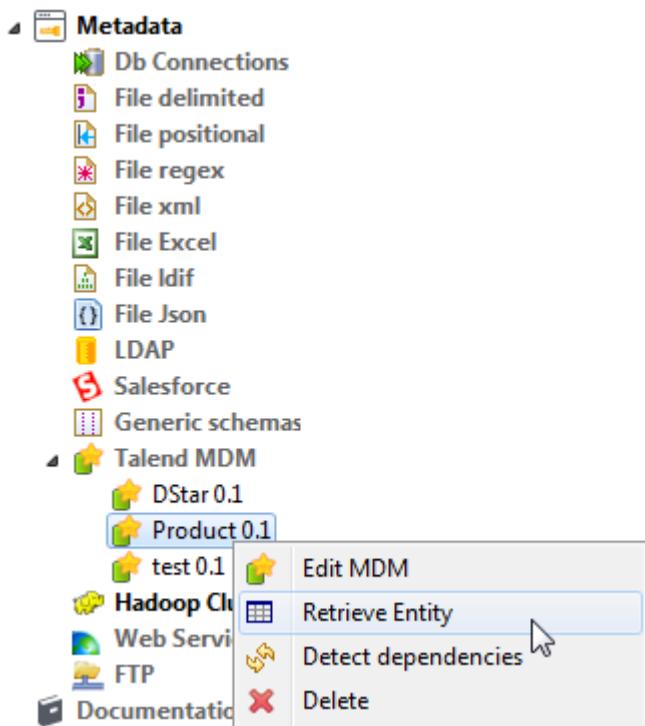
## 10.18.2. Définir un schéma MDM

### 10.18.2.1. Définir un schéma d'entrée MDM

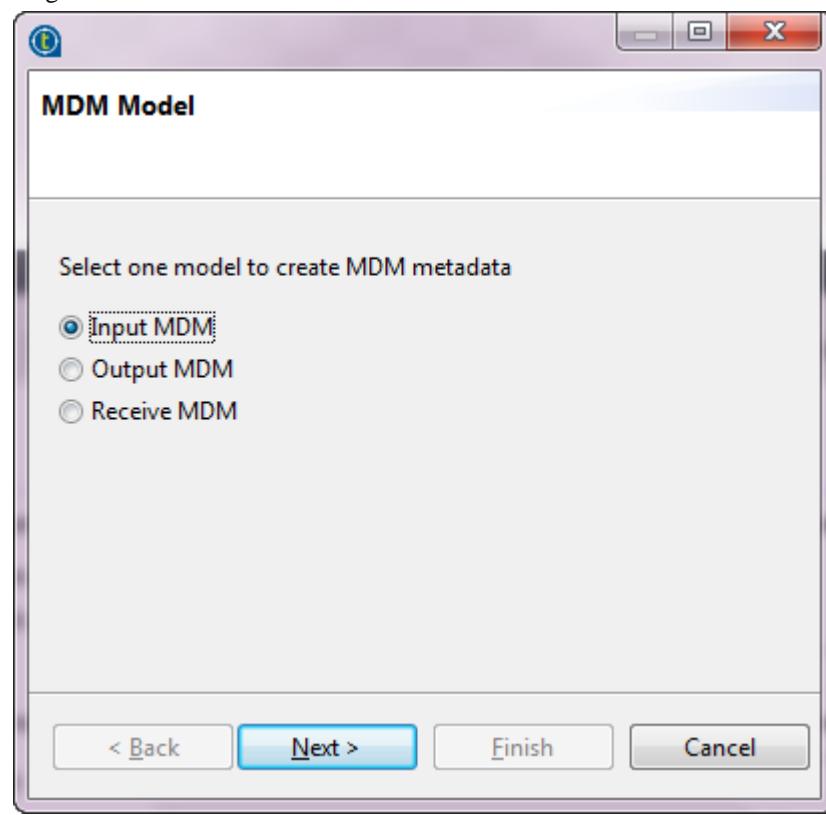
Cette section décrit la définition et le chargement d'un schéma MDM XML d'entrée. Pour définir et charger un schéma MDM XML de sortie, consultez [Définir un schéma de sortie MDM](#).

Pour définir les valeurs à retrouver à partir d'une ou plusieurs entité(s) liée(s) à une connexion MDM procédez comme suit :

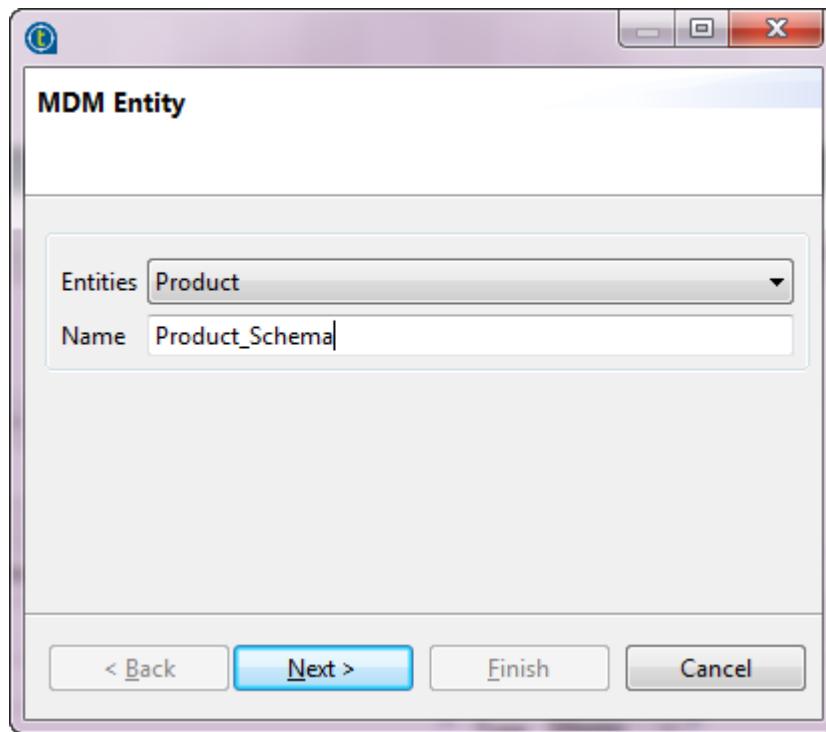
1. Dans la vue **Repository**, développez le nœud **Metadata** puis cliquez-droit sur la connexion MDM pour laquelle vous souhaitez récupérer les valeurs de l'entité/des entités.
2. Sélectionnez **Retrieve Entity** dans le menu contextuel.



Une boîte de dialogue s'affiche.



3. Sélectionnez l'option **Input MDM** afin de charger un schéma XML d'entrée, puis cliquez sur **Next** afin de passer à l'étape suivante.



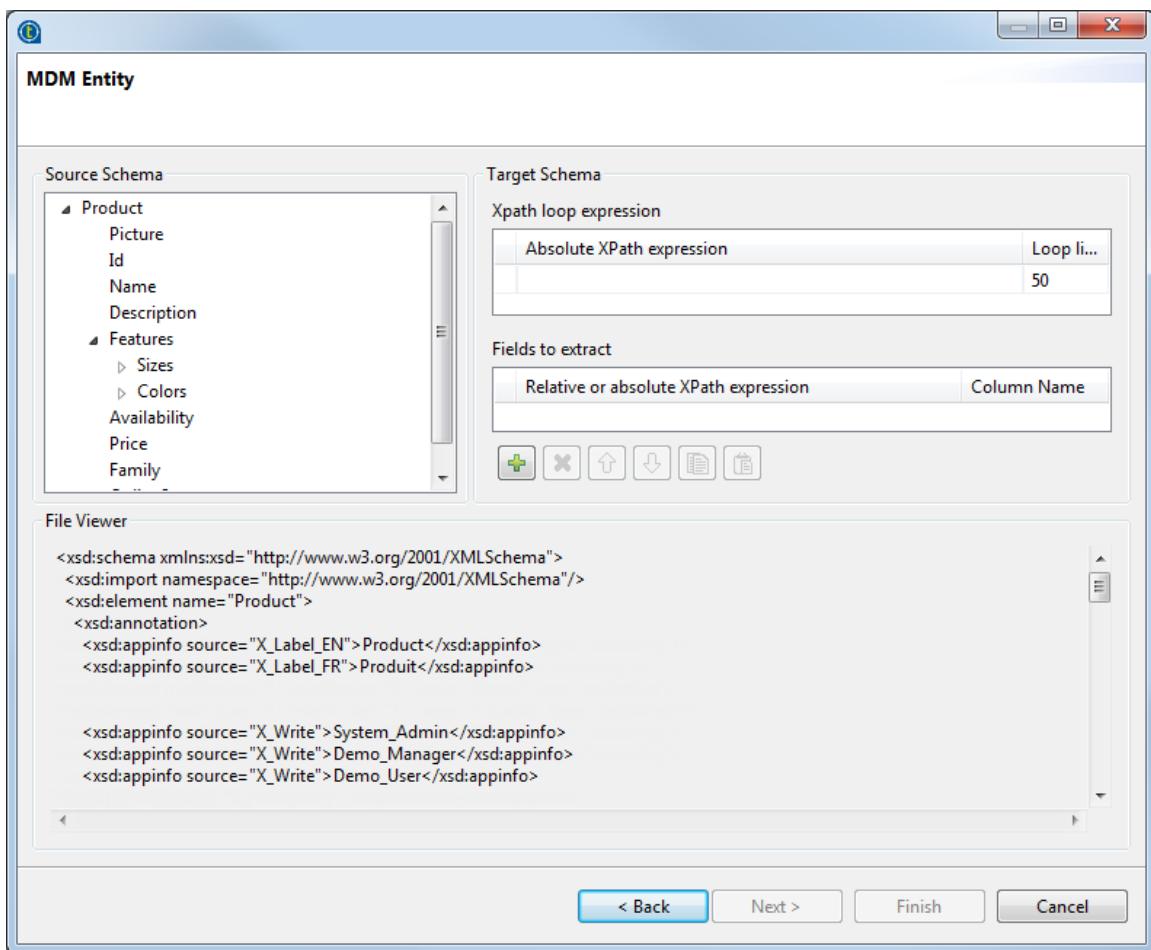
4. Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) à partir de laquelle vous souhaitez récupérer les valeurs.

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.

5. 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 :

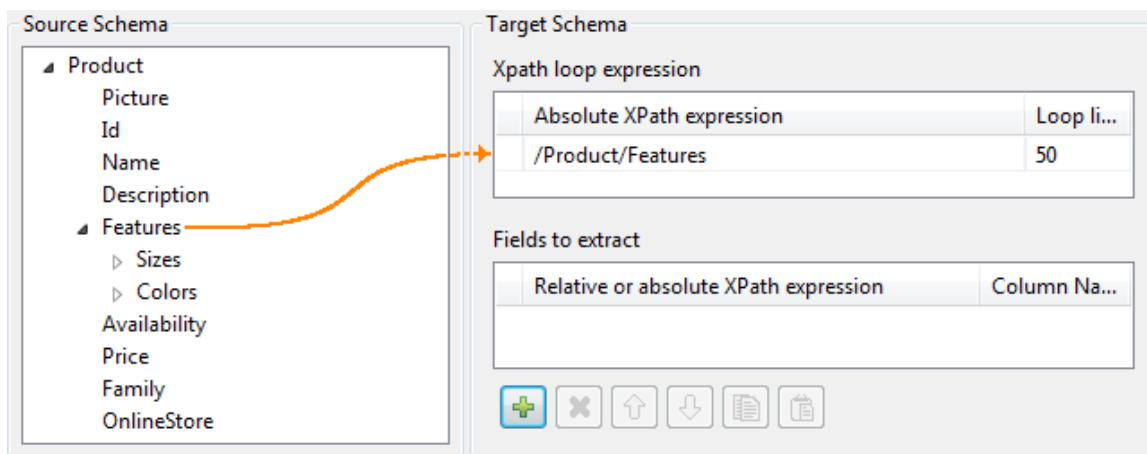
Panneau	Description
Source Schema	Arborescence de l'entité chargée.
Target schema	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible.
File viewer	Visionneur des données brutes.

6. 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.

7. Au besoin, définissez une limite (**Loop limit**) afin de restreindre l'itération à un certain nombre de nœuds.

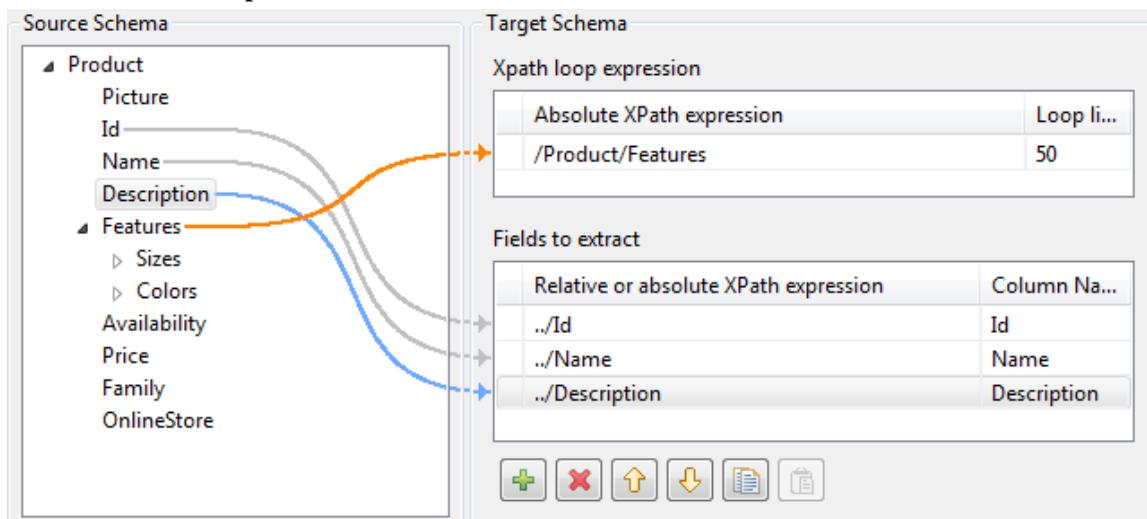


Dans la capture d'écran ci-dessus, utilisez *Features* comme élément sur lequel effectuer une boucle, puisqu'il est répété dans l'entité *Product* :

```
<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.

8. 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.

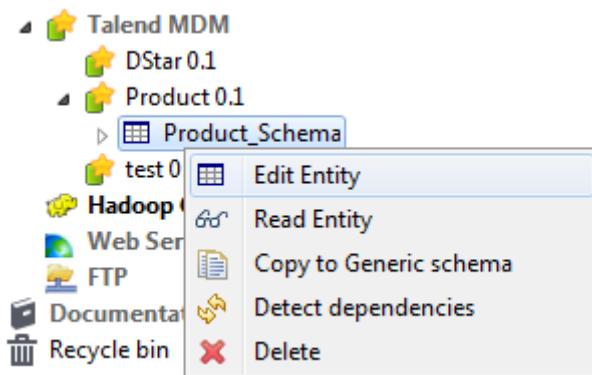
9. Au besoin, saisissez un nom pour chaque colonne récupérée, dans le champ **Column name**.



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.

10. Cliquez sur **Finish** afin de valider vos modifications et fermer la boîte de dialogue.

Le schéma nouvellement créé s'affiche sous le nœud **Talend MDM** correspondant dans la vue **Repository**.



Pour modifier le schéma créé, procédez comme suit :

1. Dans la vue **Repository**, développez les nœuds **Metadata** puis **Talend MDM** et parcourez votre répertoire jusqu'au schéma que vous souhaitez modifier.
2. Cliquez-droit sur le nom du schéma puis sélectionnez **Edit Entity** dans le menu contextuel.

Une boîte de dialogue s'ouvre.

Column	Key	Type	N.	Date Pattern (Ctrl...)	Length	Precision	Default	Comment
Id	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Name	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			
Description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		0			

3. Modifiez le schéma comme vous le souhaitez. Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.

Assurez-vous que le type de données dans la colonne **Type** est correctement défini.

Pour plus d'informations concernant les types de données Java, à savoir le modèle de date, consultez [Java API Specification](#) (en anglais).

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

- Object : est un type de données **Talend** générique qui permet le traitement des données sans tenir compte de leur contenu, par exemple, un fichier de données non supporté peut être traité à l'aide d'un composant **tFileInputRaw** en spécifiant qu'il comporte un type de données Object.
- List : est une liste d'éléments de type primitifs, séparés par un espace, dans une définition de Schéma XML, définis à l'aide de l'élément xsd:list.
- Document : est un type de données permettant le traitement d'un document XML en entier sans tenir compte de son contenu.

4. Cliquez sur **Finish** afin de fermer la boîte de dialogue.

La connexion MDM d'entrée (**tMDMInput**) est maintenant prête à être déposée dans l'un de vos Jobs.

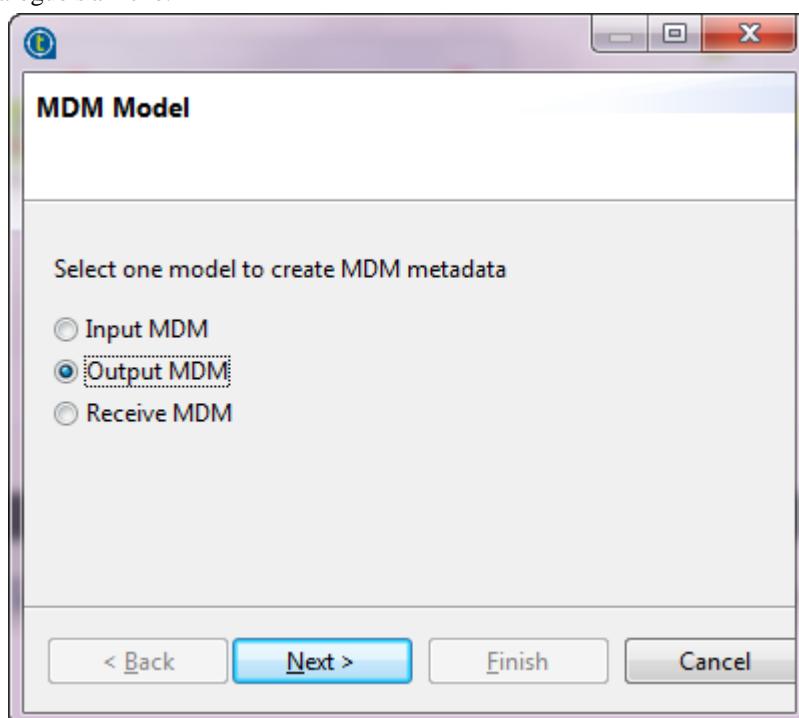
### 10.18.2.2. Définir un schéma de sortie MDM

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 [Définir un schéma d'entrée MDM](#).

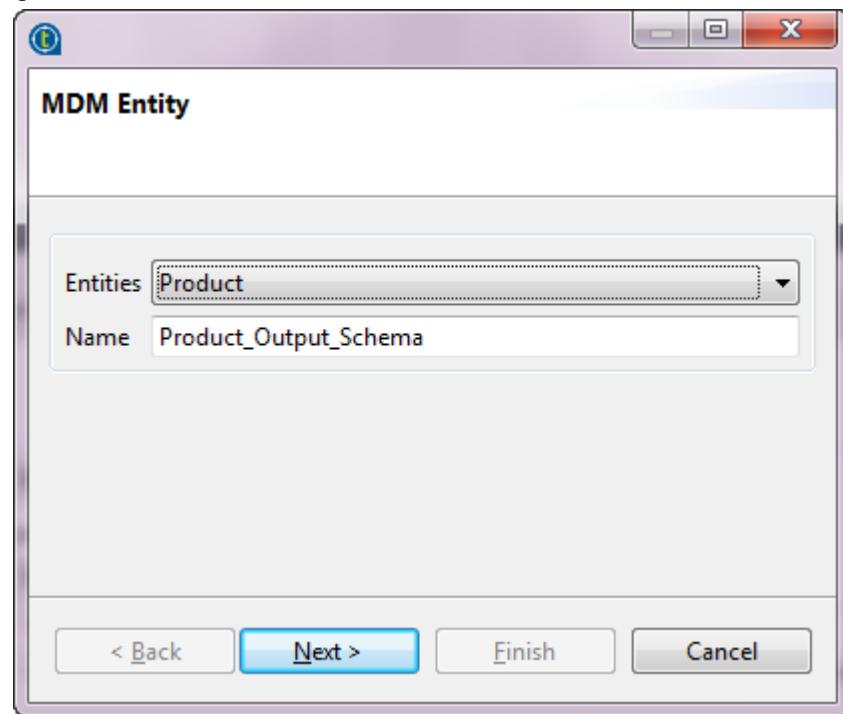
Pour définir les valeurs à écrire dans une ou plusieurs entité(s) liée(s) à une connexion MDM spécifique, procédez comme suit :

1. Dans la vue **Repository**, développez le nœud **Metadata** et cliquez-droit sur la connexion MDM pour laquelle vous souhaitez écrire les valeurs de l'entité.
2. Sélectionnez **Retrieve Entity** dans le menu contextuel.

Une boîte de dialogue s'affiche.



3. Sélectionnez l'option **Output MDM** afin de définir un schéma XML de sortie, puis cliquez sur **Next** pour procéder à l'étape suivante.



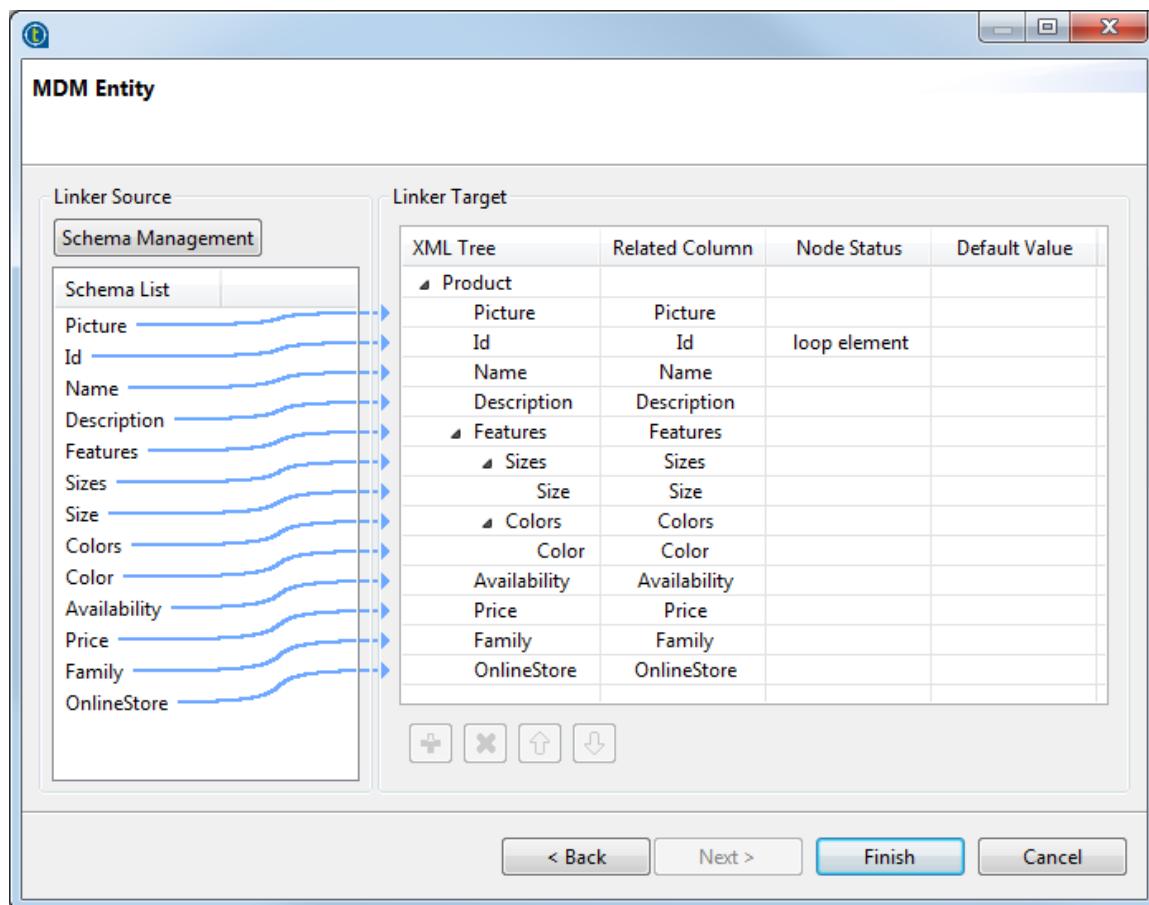
4. Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) dans laquelle vous souhaitez écrire les valeurs.

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.

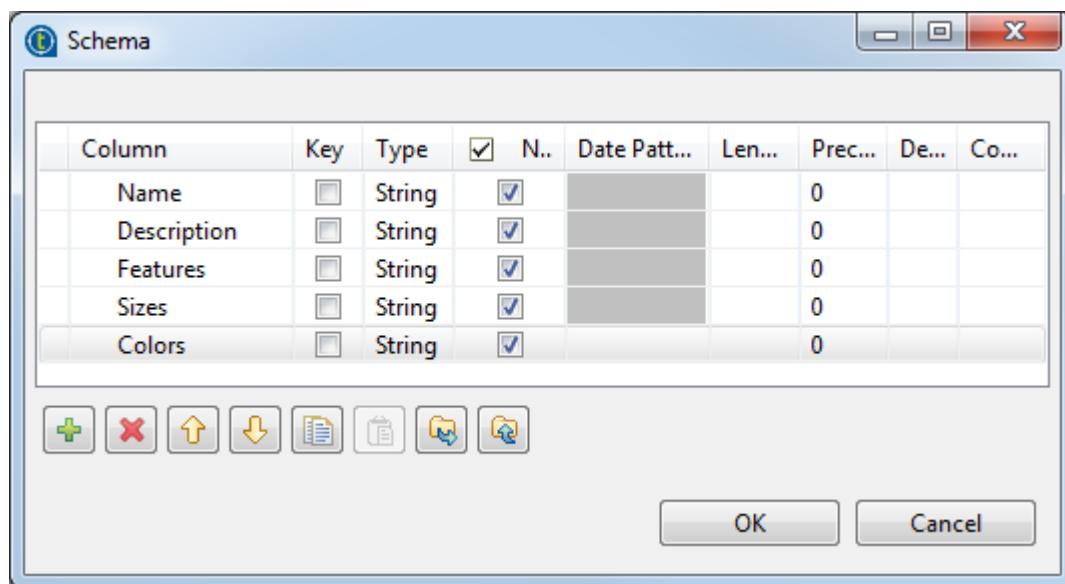
5. Cliquez sur **Next** pour procéder à l'étape suivante.



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.

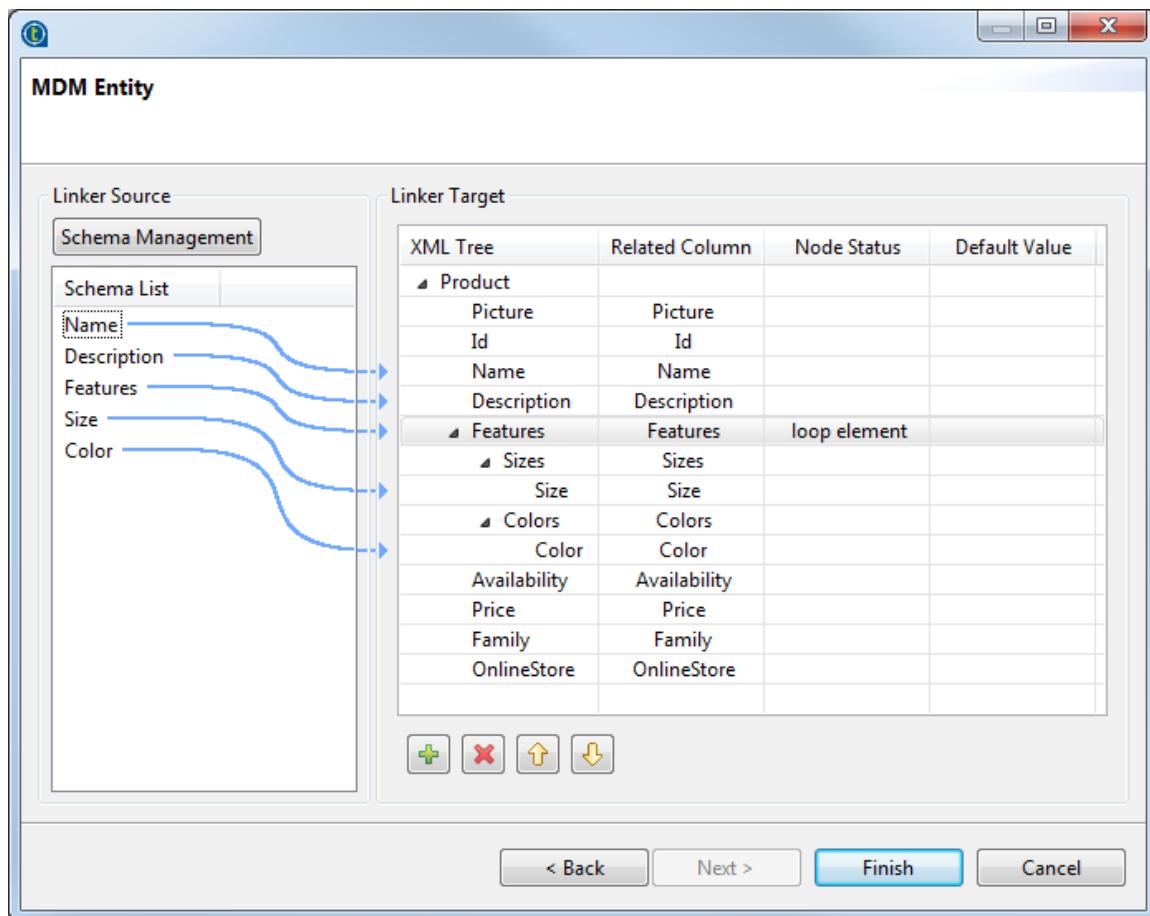
6. Cliquez sur **Schema Management** pour ouvrir une boîte de dialogue.
7. Effectuez les modifications nécessaires à la définition du schéma XML que vous souhaitez écrire dans l'entité sélectionnée.



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.

8. Cliquez sur **OK** pour fermer la boîte de dialogue.

Le schéma défini s'affiche dans la colonne **Schema list**.



9. 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 nœud(s).

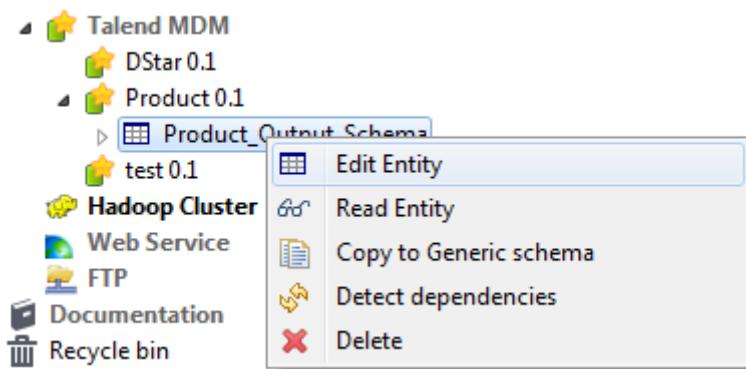
Ce faisant, le composant **tMDMOutput** utilisant cette connexion MDM créera une nouvelle ligne pour chaque élément ayant une "feature" différente.



Vous pouvez prioriser l'ordre des champs à écrire, en sélectionnant le champ et en utilisant les flèches montante et descendante.

10. 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 :

1. Dans la vue **Repository**, développez les nœuds **Metadata** et **Talend MDM** puis parcourez le référentiel jusqu'au schéma que vous souhaitez modifier.
2. Cliquez-droit sur le nom du schéma et sélectionnez **Edit Entity** dans le menu contextuel.

Une boîte de dialogue s'affiche.

Column	Key	Type	N.	Date Pattern (Ctrl...)	Length	Precision	Default	Comment
Name	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>			0		
Description	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			0		
Features	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			0		
Size	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			0		
Color	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			0		

3. Modifiez le schéma comme vous le souhaitez.

Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.

4. 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.

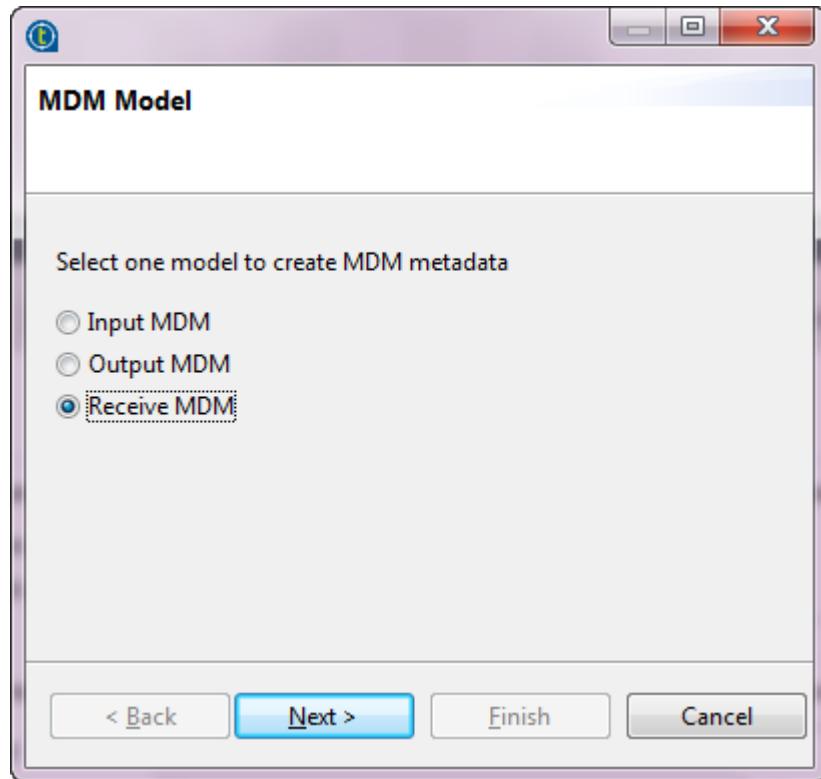
### 10.18.2.3. Définir un schéma de réception MDM

Cette section décrit la définition d'un schéma MDM XML de réception basé sur une connexion MDM.

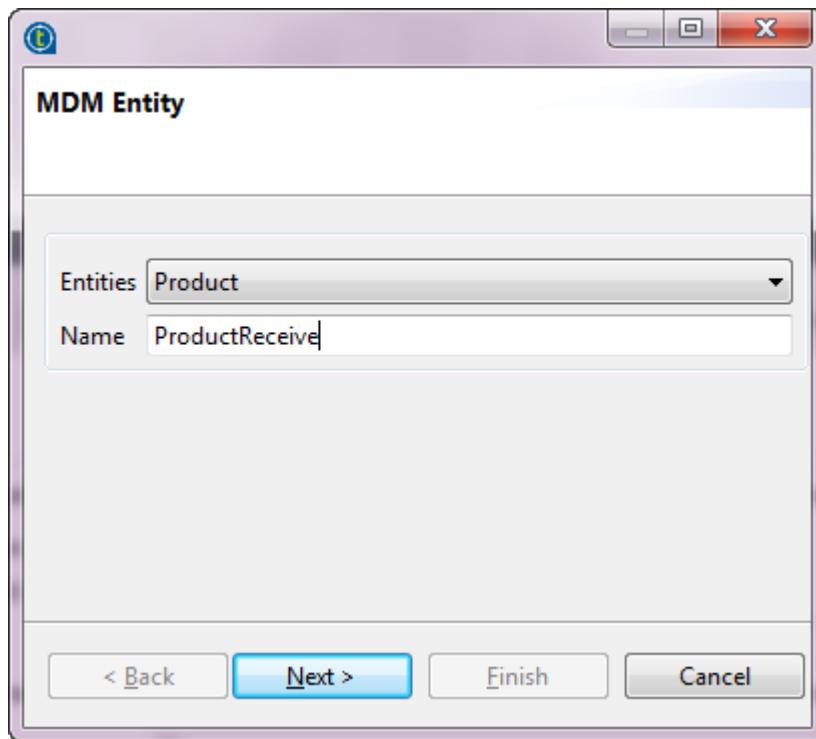
Pour définir le schéma XML que vous souhaitez recevoir selon une connexion MDM spécifique, procédez comme suit :

1. Dans la vue **Repository**, développez le nœud **Metadata** puis cliquez-droit sur la connexion MDM pour laquelle vous souhaitez récupérer les valeurs de l'entité.
2. Sélectionnez **Retrieve Entity** dans le menu contextuel.

Une boîte de dialogue s'ouvre.



3. Sélectionnez l'option **Receive MDM** afin de définir un schéma XML de réception puis cliquez sur **Next** pour passer à l'étape suivante.



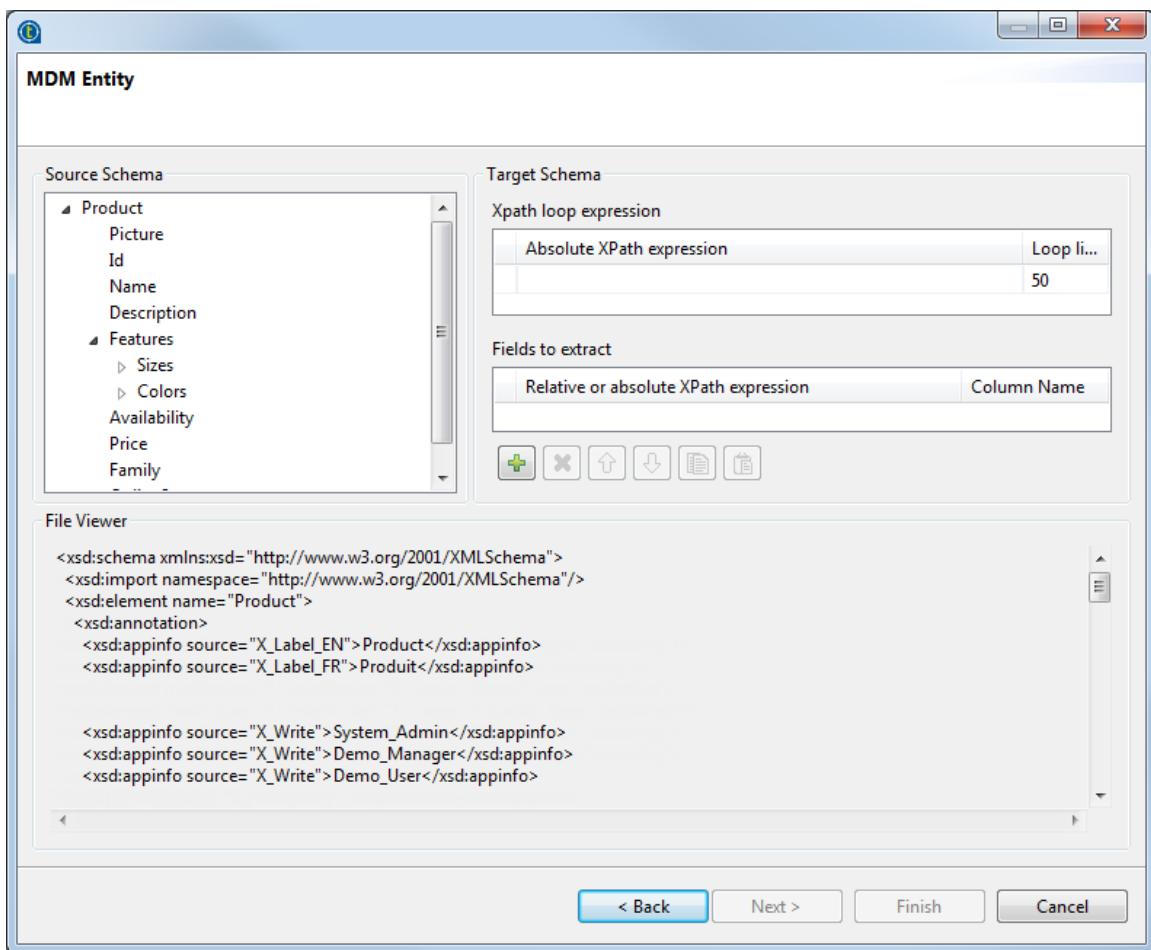
4. Dans le champ **Entities**, sélectionnez l'entité métier (schéma XML) selon laquelle vous souhaitez recevoir le schéma XML.

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.

5. 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 :

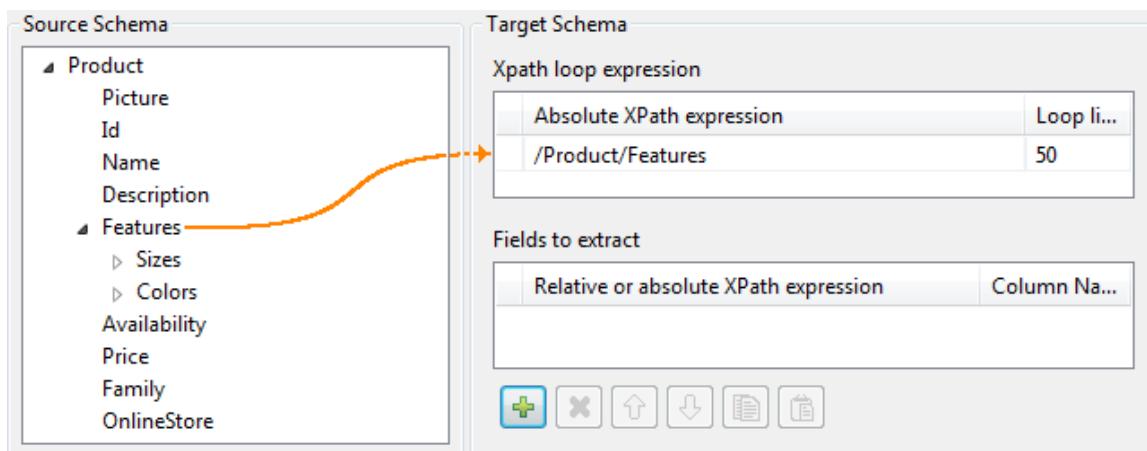
Panneau	Description
Source Schema	Arborescence de l'entité chargée.
Target schema	Informations d'extraction et d'itération.
Preview	Aperçu du schéma cible.
File viewer	Visionneur des données brutes.

6. 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.

7. Au besoin, définissez une limite (**Loop limit**) afin de restreindre l'itération à un certain nombre de nœuds.

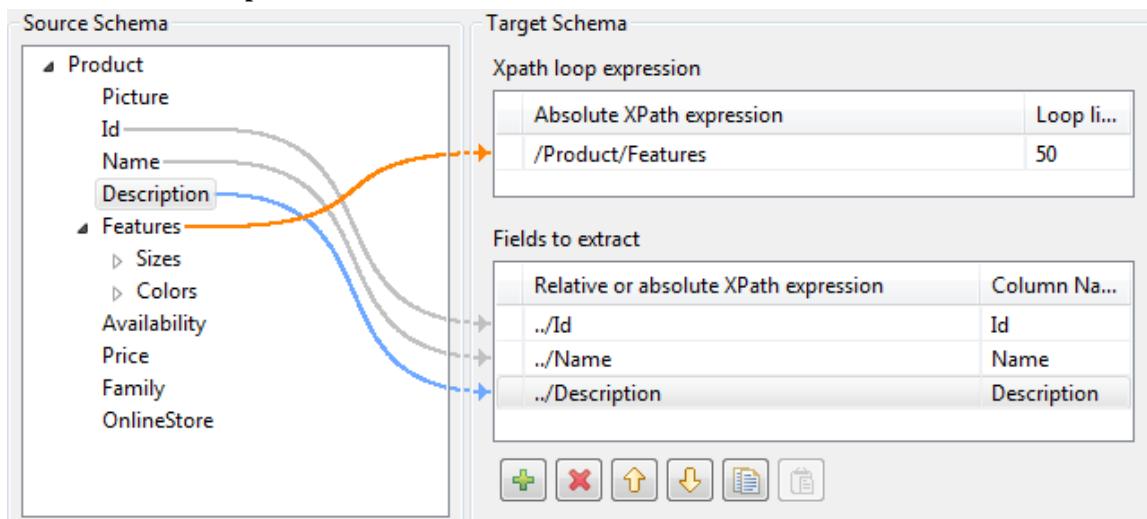


Dans la capture d'écran ci-dessus, utilisez *Features* comme élément sur lequel effectuer une boucle, puisqu'il est répété dans l'entité *Product* :

```
<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.

8. 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.

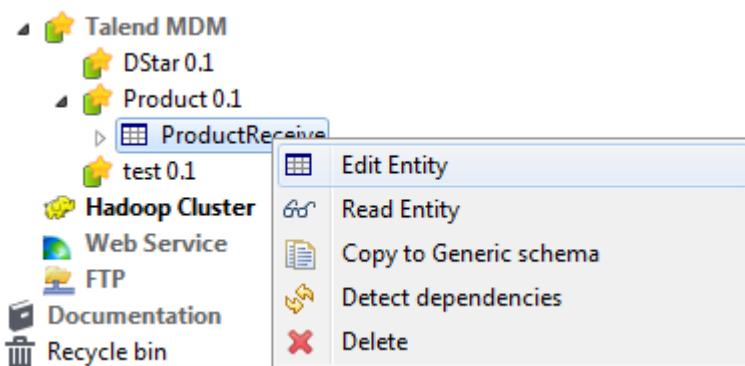
9. Au besoin, saisissez un nom pour chaque colonne reçue, dans le champ **Column name**.



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.

10. 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 :

1. Dans la vue **Repository**, développez les nœuds **Metadata** et **Talend MDM** puis parcourez le référentiel jusqu'au schéma que vous souhaitez modifier.
2. Cliquez-droit sur le nom du schéma et sélectionnez **Edit Entity** dans le menu contextuel.

Une boîte de dialogue s'affiche.

Column	Key	Type	Date Pattern (Ctrl...)	Length	Precision	Default	Comment
<b>Id</b>	<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		

Buttons at the bottom: +, -, Up, Down, Save, Cancel, < Back, Next >, Finish.

3. Modifiez le schéma comme vous le souhaitez. Vous pouvez changer le nom du schéma selon vos besoins, et vous pouvez également personnaliser la structure du schéma dans le panneau correspondant. La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.
4. 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.

## 10.19. Centraliser les métadonnées Web Service

Si vous accédez souvent à un service Web depuis *Talend Open Studio for Big Data*, vous pouvez stocker vos connexions Web Service dans le **Repository**.

Cet assistant intègre deux assistants en un seul. En effet, selon vos besoins, vous pouvez créer un schéma simple (**Simple WSDL**), ou un schéma avancé (**Advanced WebService**).

 L'étape 1, dans laquelle vous devez renseigner les propriétés générales du futur schéma, précède l'étape de sélection du type **Simple** ou **Advanced**. Veillez donc à différencier vos schémas simples de vos schémas avancés, en saisissant des noms que vous pourrez reconnaître.

Si vous souhaitez créer un schéma simple, consultez [Configurer un schéma simple](#).

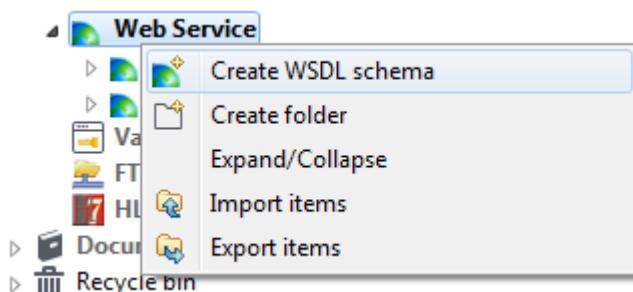
Si vous souhaitez créer un schéma avancé, consultez [Configurer un schéma avancé](#).

### 10.19.1. Configurer un schéma simple

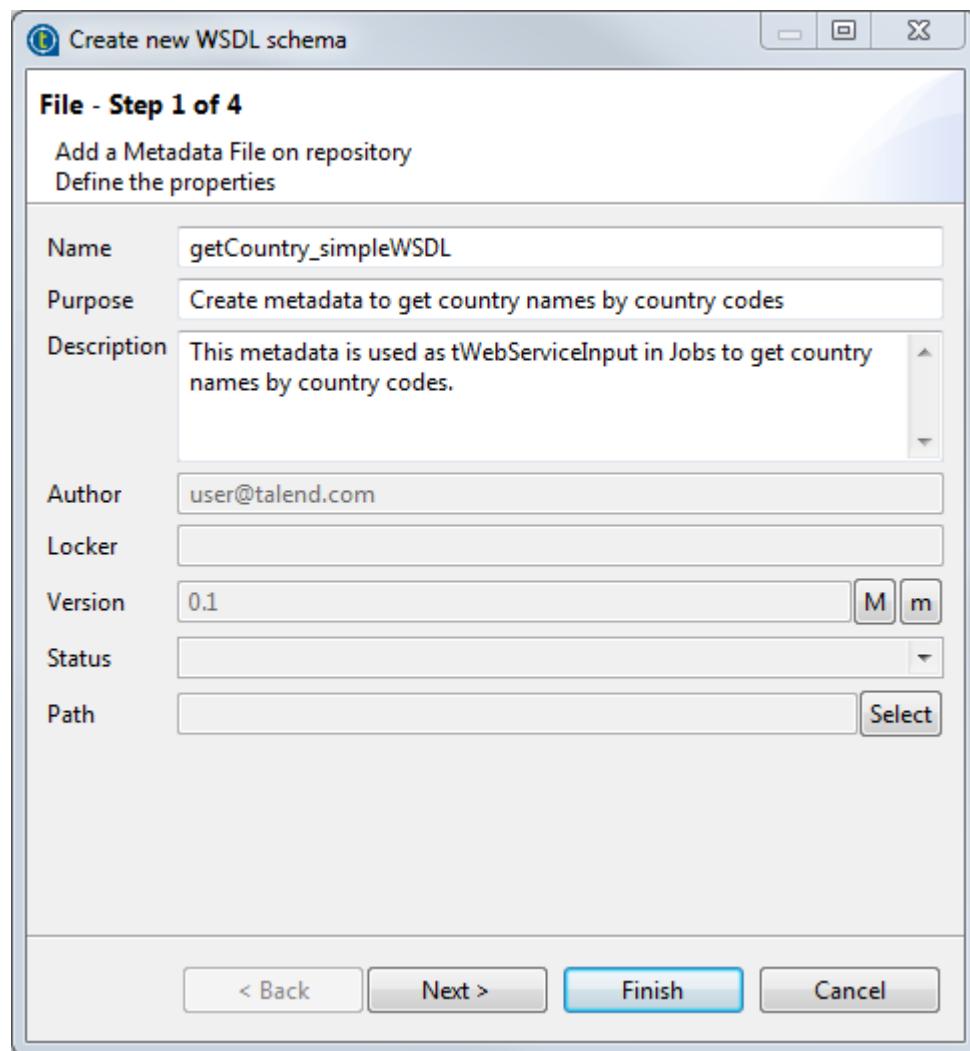
Cette section décrit la définition d'un schéma Web Service simple (**Simple WSDL**). Pour définir un schéma avancé (**Advanced Web Service**), consultez [Configurer un schéma avancé](#).

#### Définir les propriétés générales

1. Dans la vue **Repository**, développez le nœud **Metadata**.
2. Cliquez-droit sur **Web Service**, et sélectionnez **Create WSDL** dans le menu contextuel.



3. Remplissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.



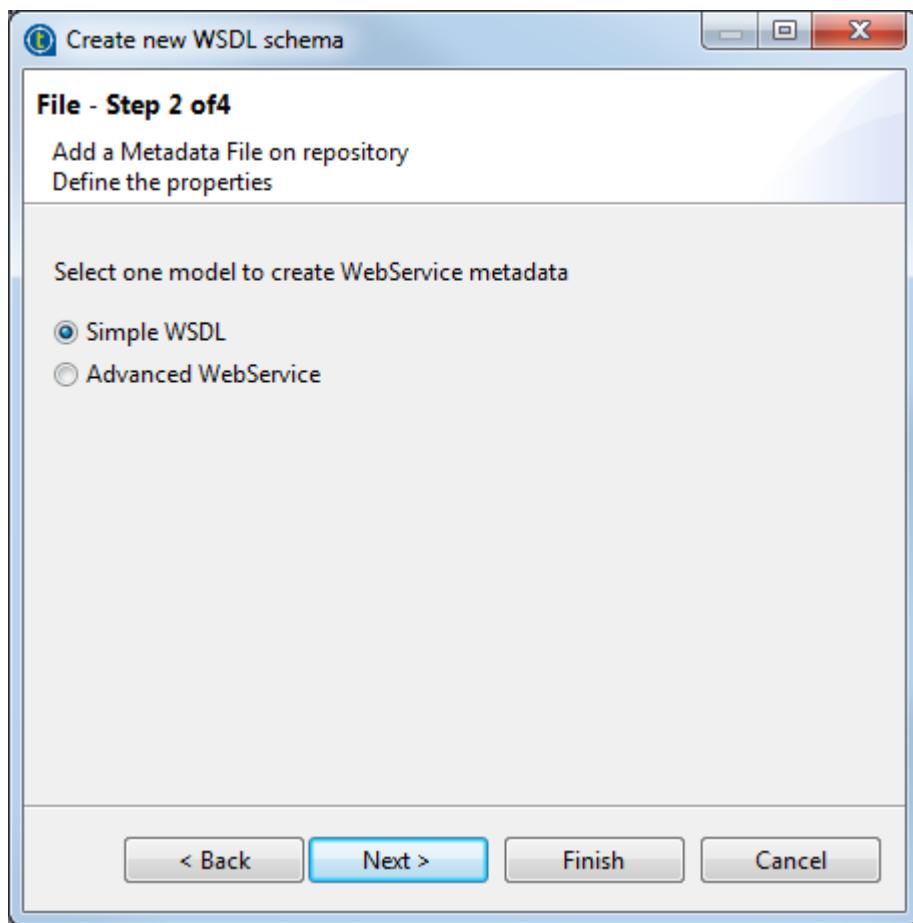
4. Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

### Sélectionner le type de schéma (Simple)

Dans cette étape, vous allez définir le type de schéma, **Simple** ou **Advanced**. Dans cet exemple, sélectionnez le type **Simple**.

1. Dans la boîte de dialogue, sélectionnez **Simple WSDL**.

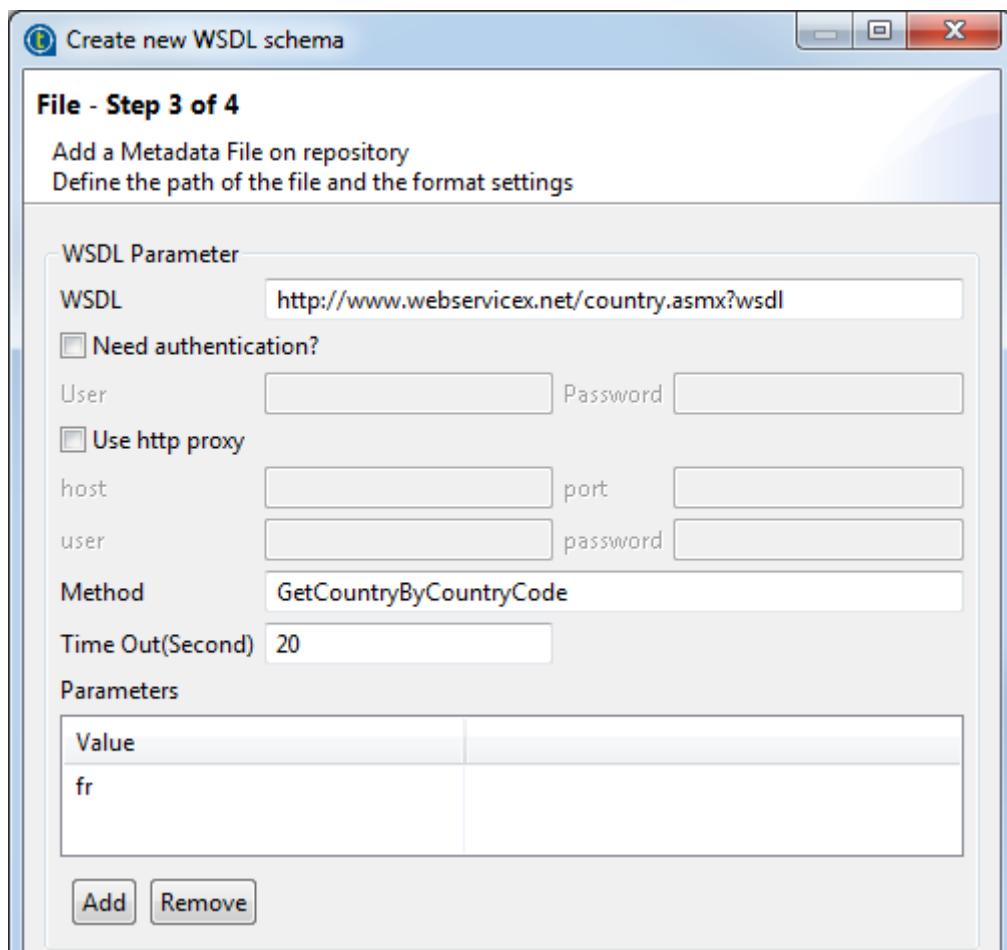
2.



Cliquez sur **Next** pour continuer.

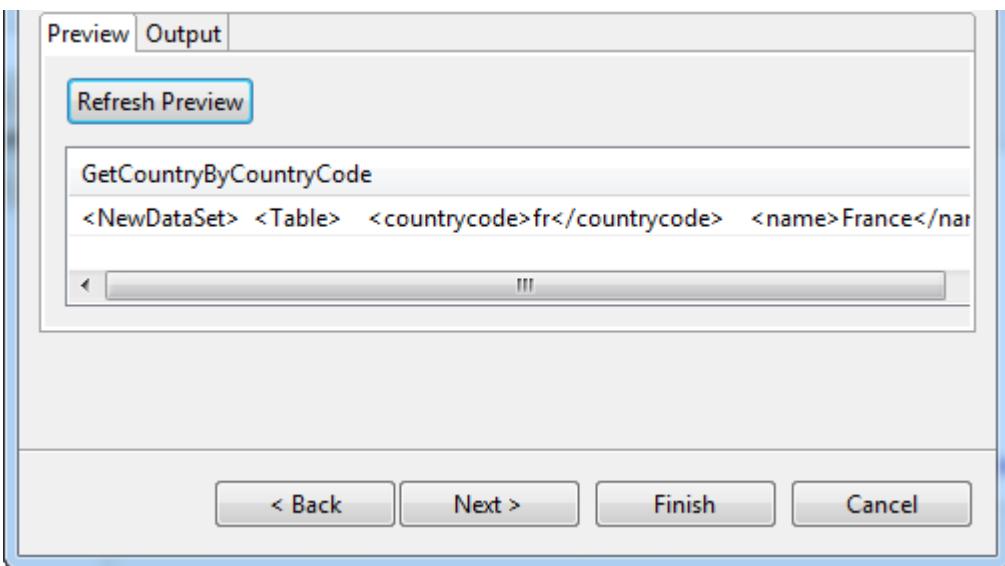
### Définir l'URI et la méthode

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



Dans la zone **Web Service Parameter** :

1. Dans le champ **WSDL**, renseignez l'URI qui transmettra les valeurs souhaitées, *http://www.webservicex.net/country.asmx?wsdl* dans cet exemple.
2. Si nécessaire, cochez la case **Need authentication?** puis saisissez respectivement votre identifiant et votre mot de passe, dans les champs **User** et **Password**.
3. Si vous utilisez un proxy http, cochez la case **Use http proxy**, et renseignez les propriétés (respectivement l'hôte, le port, l'identifiant et le mot de passe).
4. Dans le champ **Method**, définissez la méthode en la nommant, *GetCountryByCountryCode* dans cet exemple.
5. Dans le tableau **Value**, ajoutez ou supprimez autant de valeurs que vous souhaitez, en cliquant sur le bouton **Add** (Ajouter) ou **Remove** (Supprimer).
6. Puis cliquez sur **Refresh Preview** pour vérifier que les paramètres sont corrects.



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

## Finaliser le schéma

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

**Create new WSDL schema**

**File - Step 4 of 4**

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

Name	metadata														
Comment															
<b>Schema</b>															
Click to update schema preview															
Guess															
<b>Description of the Schema</b>															
<table border="1"> <thead> <tr> <th>Column</th> <th>Key</th> <th>Type</th> <th><input checked="" type="checkbox"/></th> <th>N..</th> <th>Date Pattern (Ctrl...)</th> <th>Len</th> </tr> </thead> <tbody> <tr> <td>Result</td> <td><input type="checkbox"/></td> <td>String</td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> <td>184</td> </tr> </tbody> </table>		Column	Key	Type	<input checked="" type="checkbox"/>	N..	Date Pattern (Ctrl...)	Len	Result	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			184
Column	Key	Type	<input checked="" type="checkbox"/>	N..	Date Pattern (Ctrl...)	Len									
Result	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>			184									
<input type="button" value="&lt; Back"/> <input type="button" value="Next &gt;"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/>															

- 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**.

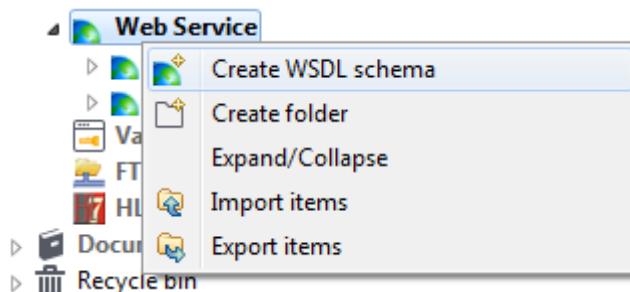
Le nouveau schéma s'affiche dans le **Repository** sous le nœud **Web Service** approprié. Vous pouvez à présent le déposer dans l'espace de modélisation graphique en tant que composant **tWebServiceInput** dans votre Job.

## 10.19.2. Configurer un schéma avancé

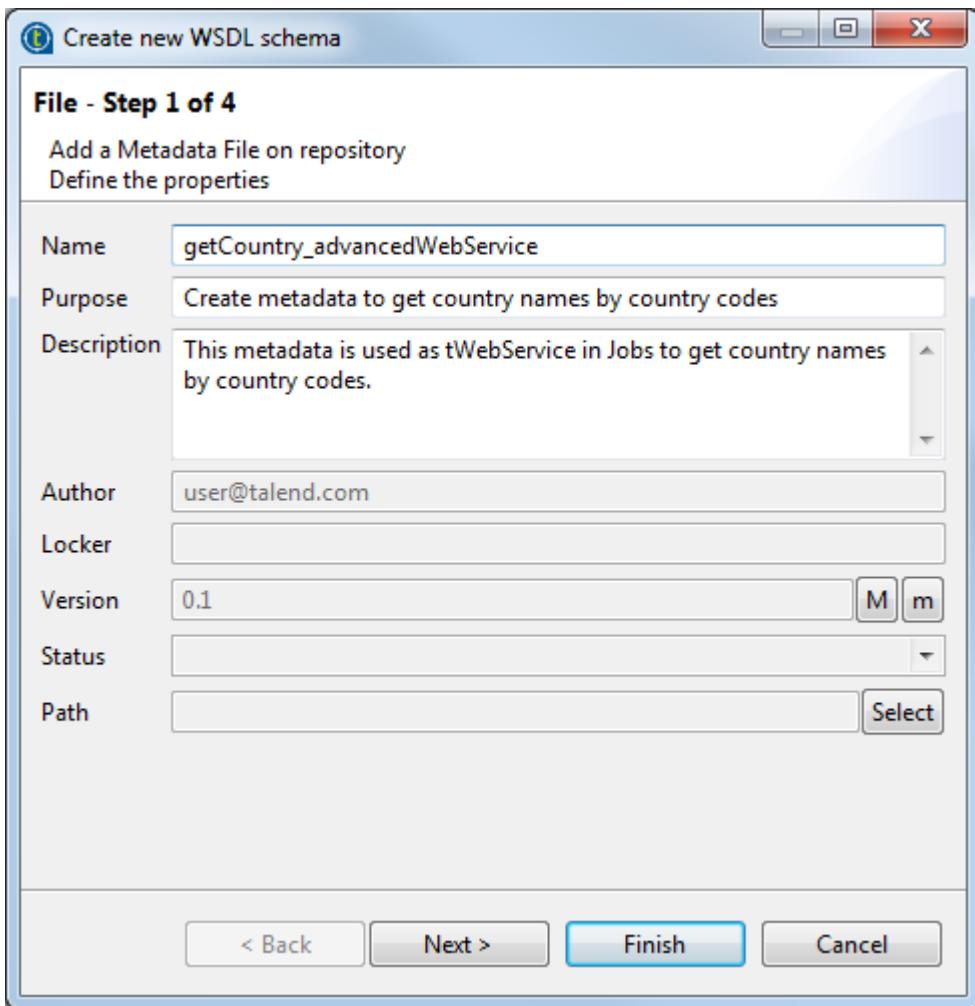
Cette section décrit la définition d'un schéma Web Service avancé (**Advanced WebService**). Pour définir un schéma simple (**Simple WSDL**), consultez [Configurer un schéma simple](#).

### Définir les propriétés générales

1. Dans la vue **Repository**, développez le nœud **Metadata**.
2. Cliquez-droit sur **Web Service**, et sélectionnez **Create WDSL schema** dans le menu contextuel.



3. Remplissez les informations génériques du schéma, comme son nom (**Name**) et sa **Description**.

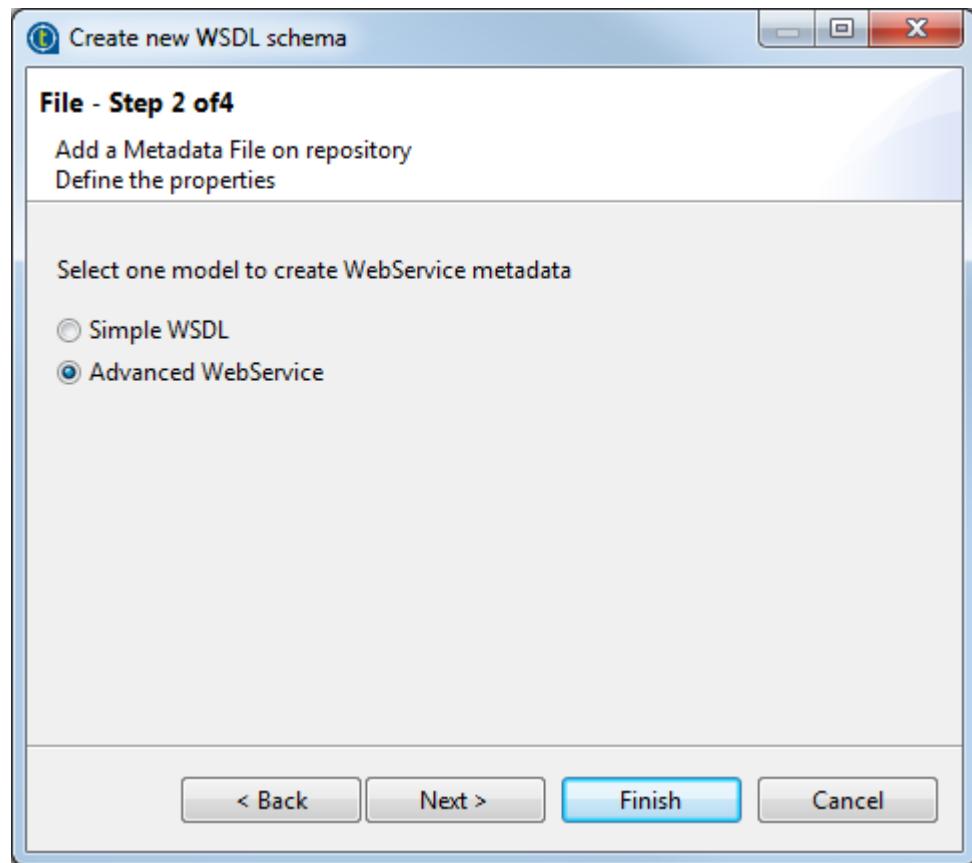


4. Cliquez sur **Next** pour passer à l'étape suivante et sélectionner le type de schéma.

### Sélectionner le type de schéma (Advanced)

Dans cette étape, vous allez définir le type de schéma, **Simple** ou **Advanced**. Dans cet exemple, sélectionnez le type **Advanced**.

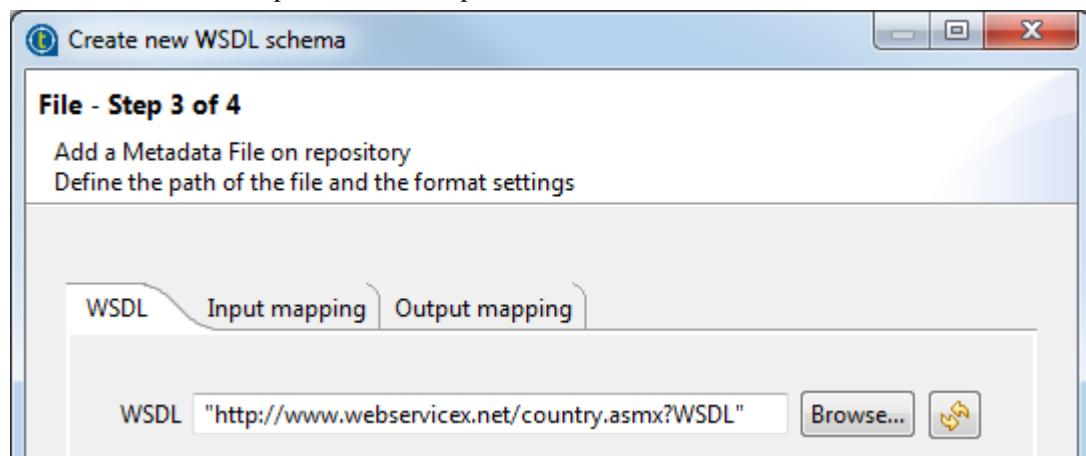
1. Dans la boîte de dialogue, sélectionnez **Advanced WebService**.



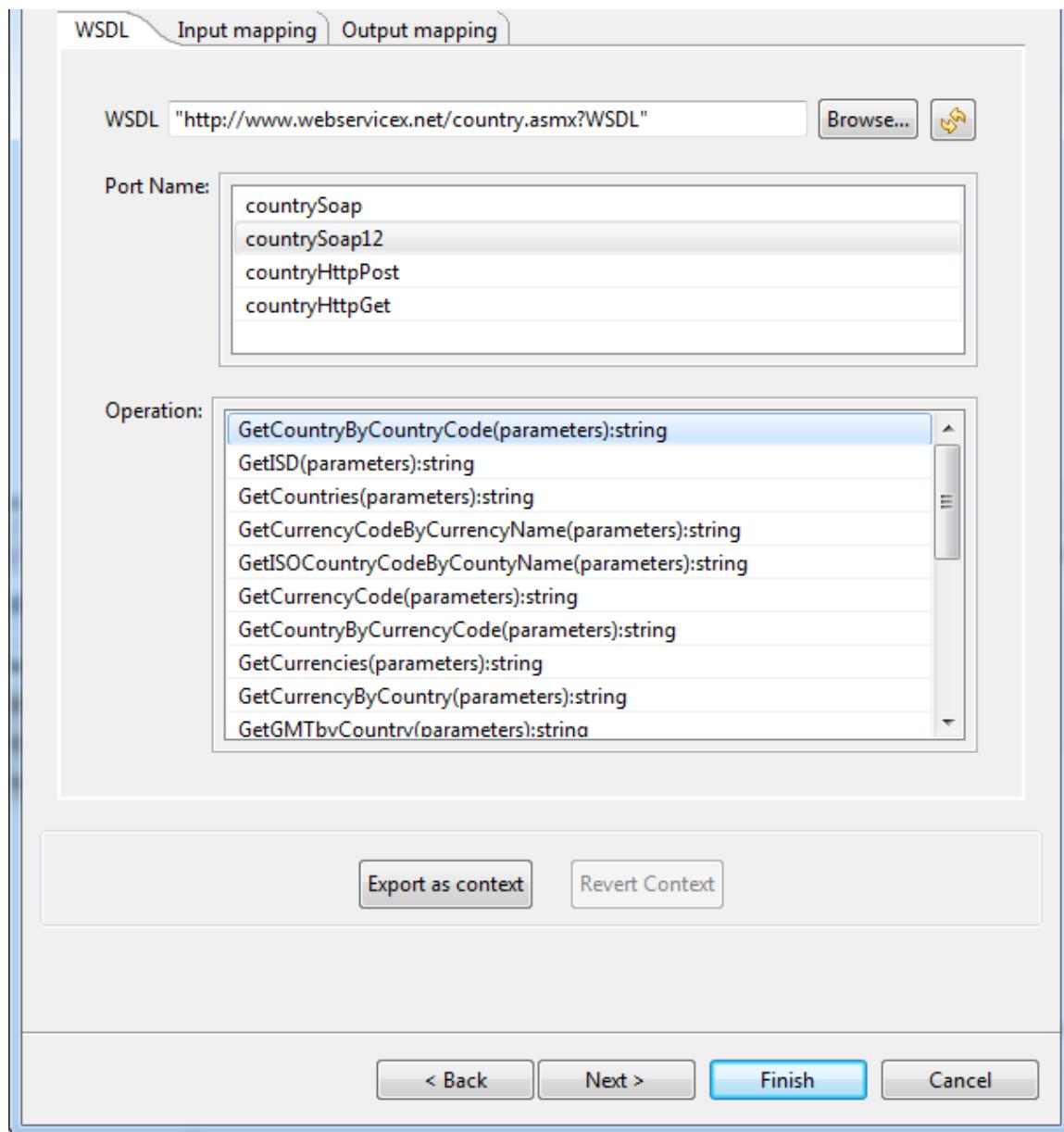
2. Cliquez sur **Next** pour définir plus précisément les paramètres du service web.

### Définir l'opération et le nom du port

1. Dans le champ **WSDL**, saisissez l'URI du fichier Web Service WSDL. Vous pouvez également cliquer sur le bouton **Browse...** afin de parcourir votre répertoire si votre WSDL est stocké localement.



2. Cliquez sur le bouton **Refresh** afin de récupérer la liste des noms de port et des opérations disponibles.



3. Dans la zone **Port Name:**, sélectionnez le nom du port à utiliser, *countrySoap12* dans cet exemple.
4. Dans la zone **Operation:**, sélectionnez l'opération à effectuer.

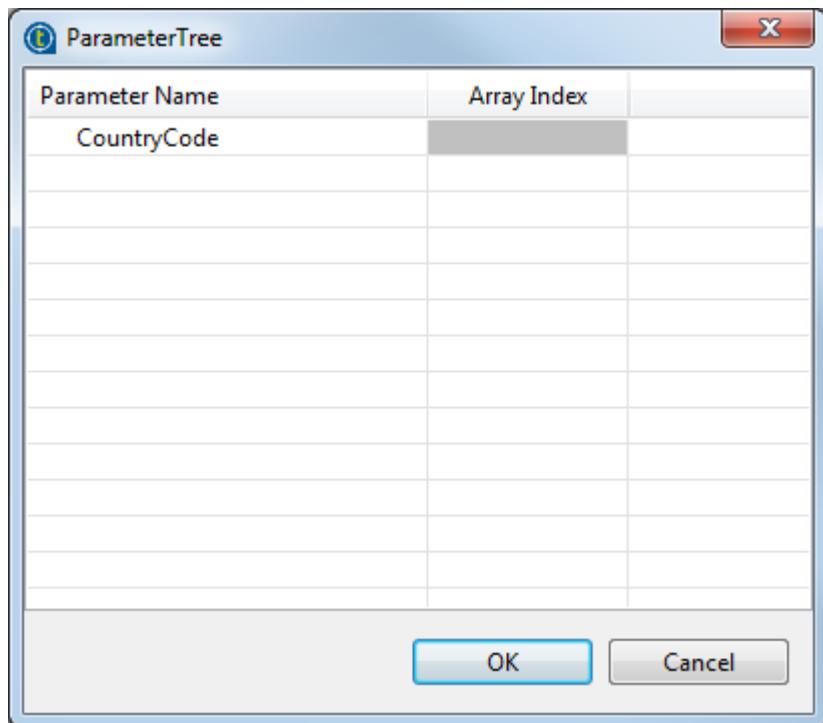
Dans cet exemple, *GetCountryByCountryCode(parameters):string* pour retrouver le nom du pays correspondant au code de pays donné.

Définissez les schémas d'entrée et de sortie ainsi que les mappings entre les schémas et les paramètres dans les onglets **Input mapping** et **Output mapping**.

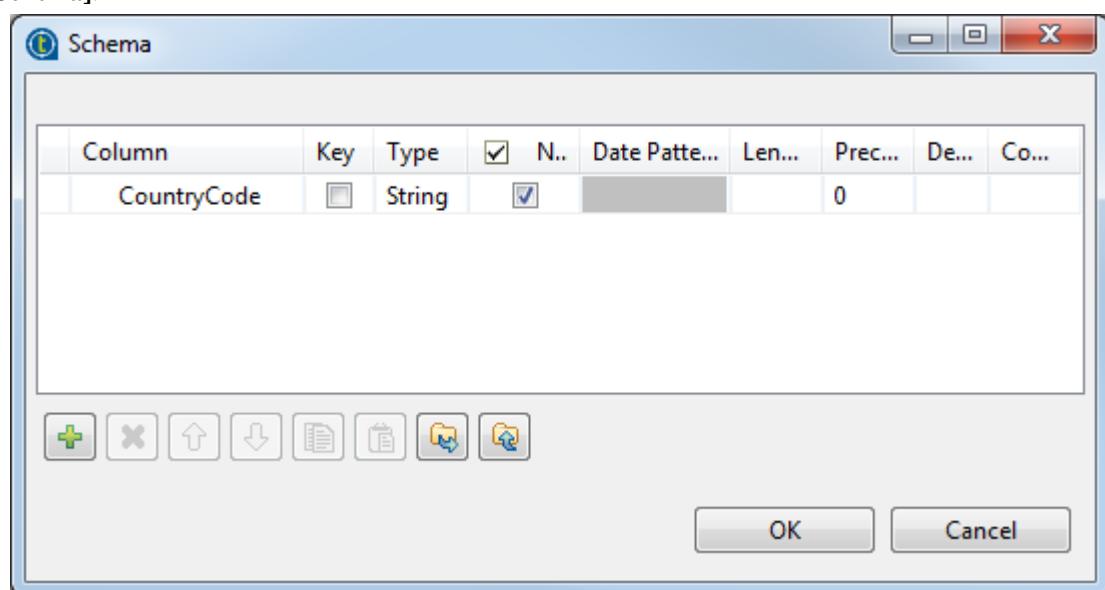
### Définir les schémas d'entrée et les mappings

Pour définir le schéma d'entrée et ses mappings, procédez comme suit :

1. Cliquez sur l'onglet **Input mapping** afin de définir le schéma d'entrée et les paramètres nécessaires à l'exécution de l'opération.
2. Dans le tableau de droite, sélectionnez la ligne **parameters** et cliquez sur le bouton **[+]** afin d'ouvrir la boîte de dialogue **[ParameterTree]**.

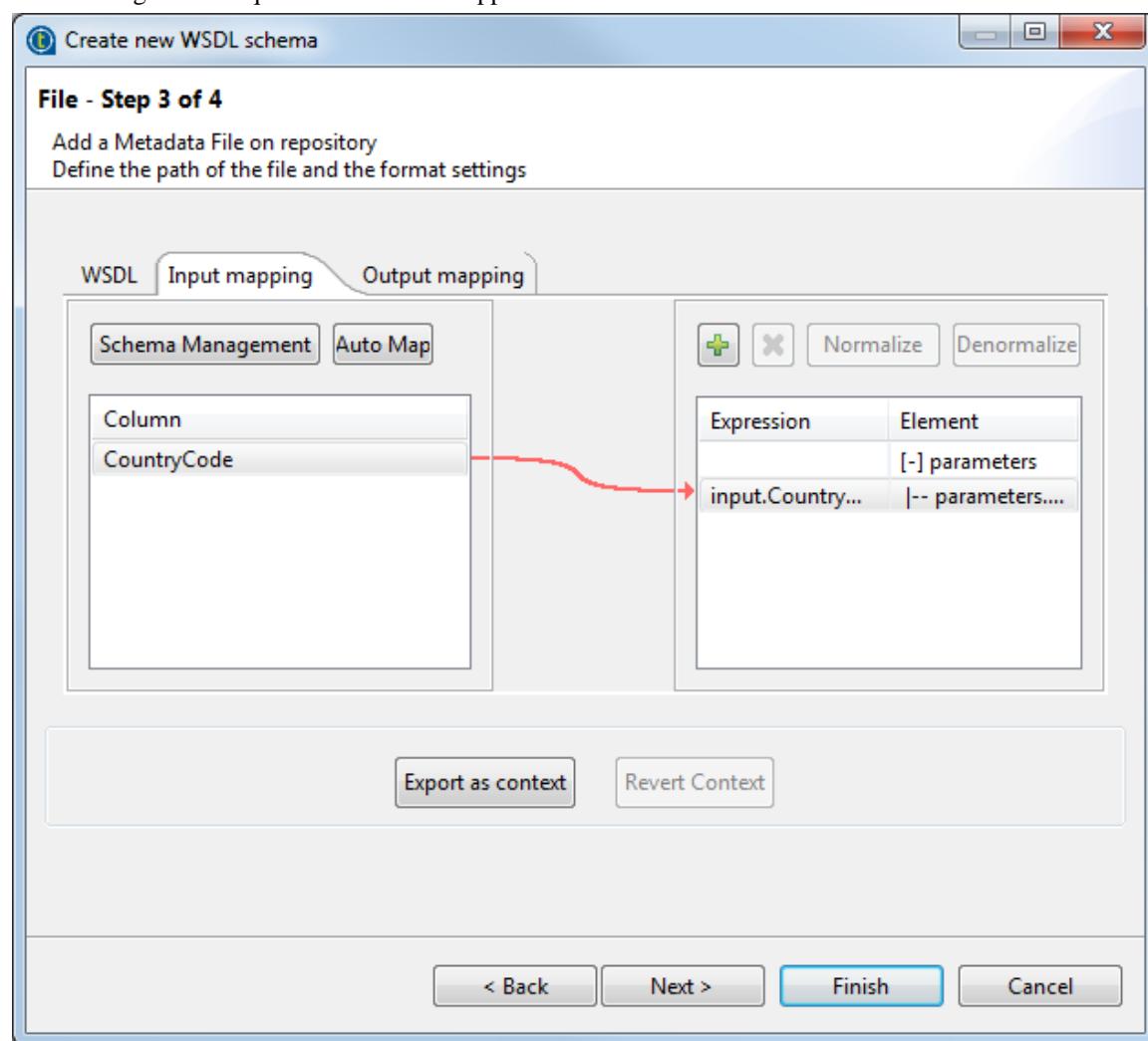


3. Sélectionnez le paramètre que vous souhaitez utiliser et cliquez sur **OK** afin de fermer la boîte de dialogue.  
Une nouvelle ligne s'affiche avec le paramètre que vous avez ajouté, *CountryCode* dans cet exemple.
4. Dans le tableau de gauche, cliquez sur le bouton **Schema Management** afin d'ouvrir la boîte de dialogue **[Schema]**.



5. Définissez le schéma d'entrée.  
Dans cet exemple, le schéma n'a qu'une colonne : *CountryCode*.
6. Cliquez **OK** afin de valider votre ajout et fermer la boîte de dialogue.
7. Créez les mappings entre les colonnes du schéma et les paramètres.  
Dans cet exemple, cliquez sur la colonne *Country*, dans la table de gauche, puis glissez-la dans la ligne *parameters.CountryCode*, dans la table de droite.

Un lien rouge montre que la colonne est mappée.



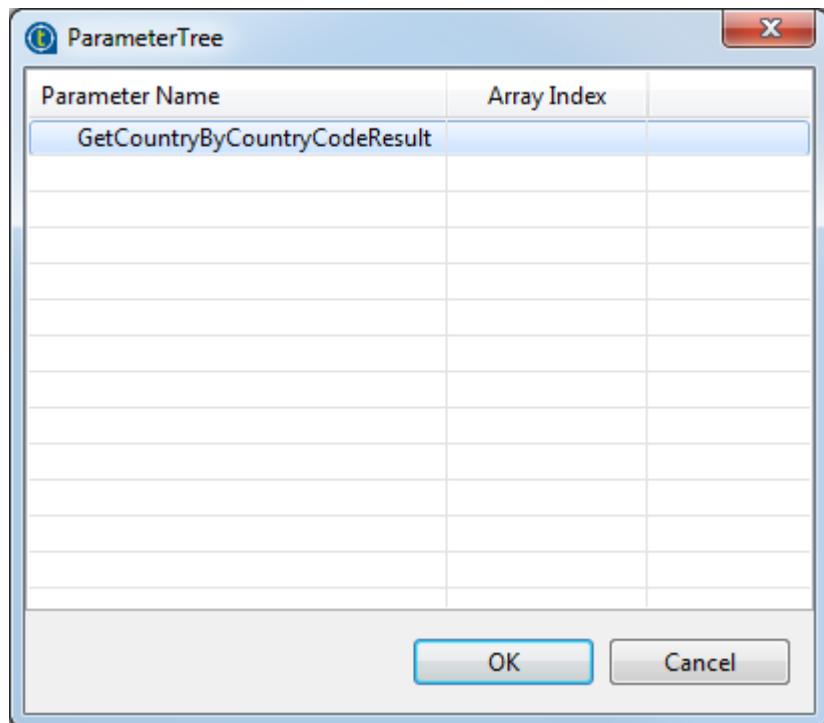
S'il est disponible, utilisez le bouton **Auto map** situé en haut de l'onglet, il permet d'effectuer cette opération automatiquement.

## Définir les schémas de sortie et les mappings

Pour définir le schéma de sortie et ses mappings, procédez comme suit :

1. Cliquez sur l'onglet **Output mapping** afin de définir le schéma de sortie et ses paramètres.
2. Dans le tableau de gauche, sélectionnez la ligne **parameters** et cliquez sur le bouton **[+]**, afin d'ajouter un paramètre.

La boîte de dialogue **[ParameterTree]** s'ouvre.

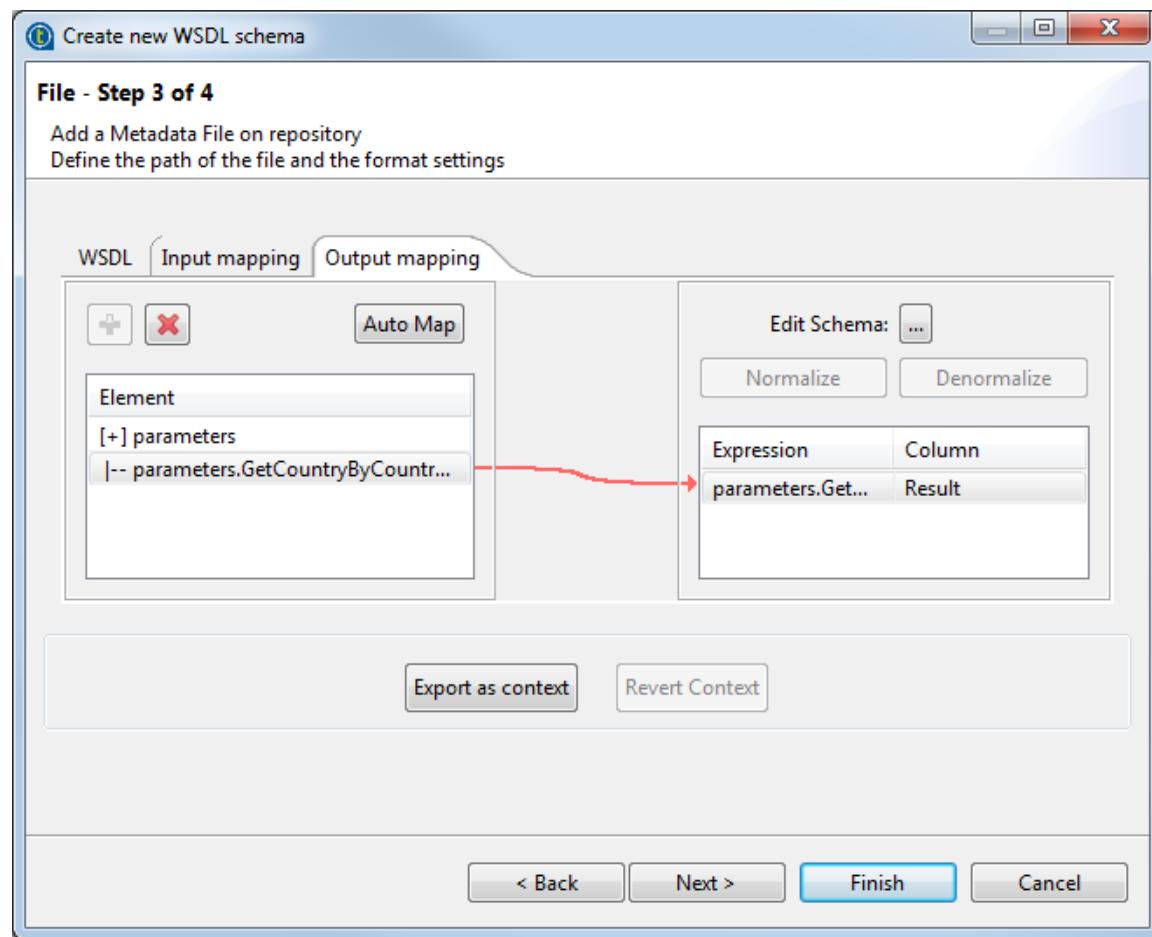


3. Sélectionnez le paramètre et cliquez sur **OK** pour fermer la boîte de dialogue.

Une nouvelle ligne s'affiche avec le paramètre que vous avez ajouté, *GetCountryByCountryCodeResult* dans cet exemple.

4. Dans le tableau de droite, cliquez sur le bouton [...] afin d'ouvrir la boîte de dialogue [**Schema**].
5. Définissez le schéma de sortie.  
Dans cet exemple, le schéma n'a qu'une colonne : *Result*.
6. Cliquez sur **OK** afin de valider votre ajout et fermer la boîte de dialogue.
7. Créez les mappings entre le schéma de sortie et les paramètres.

Dans cet exemple, cliquez sur la ligne *parameters.GetCountryByCountryCodeResult*, dans le tableau de gauche, puis déposez-la dans la colonne *Result*, à droite.



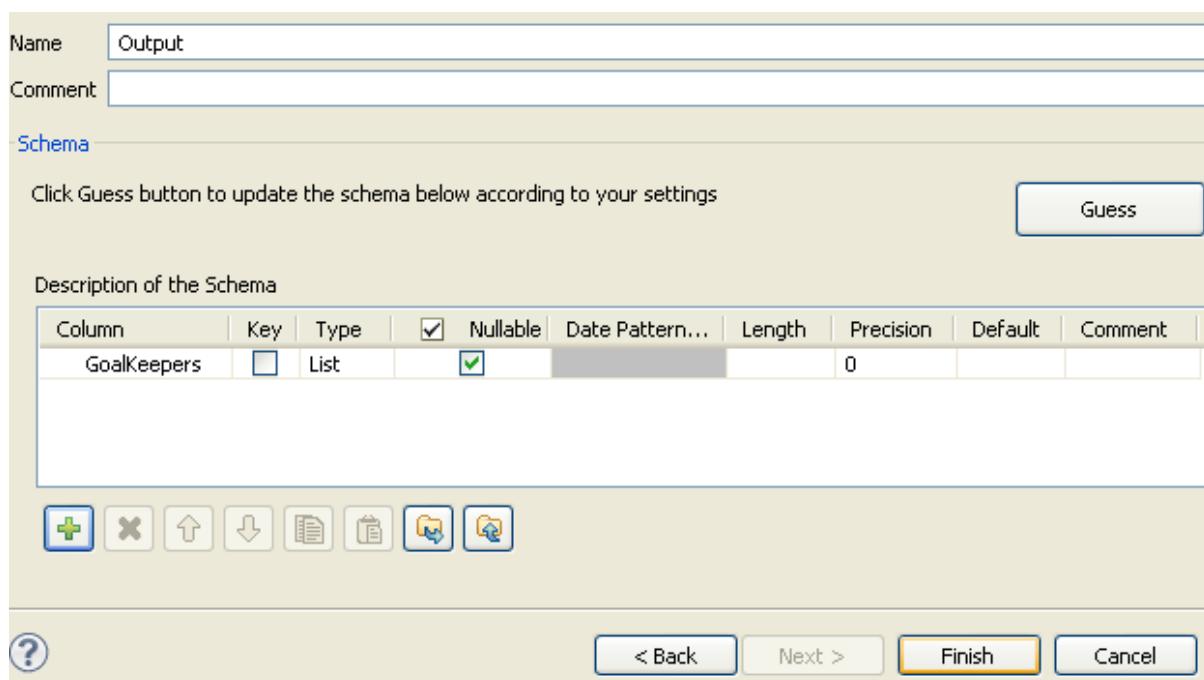
8. Cliquez sur **Next** pour passer à l'étape suivante.



Selon le type de sortie choisi, vous pouvez choisir de normaliser ou de dénormaliser les résultats, en cliquant respectivement sur les bouton **Normalize** ou **Denormalize**.

## Finaliser le schéma

Dans cette étape, l'assistant affiche le schéma de sortie généré.



Vous pouvez personnaliser le nom de la métadonnée dans le champ **Name** (par défaut *Output*), ajouter un commentaire dans le champ **Comment** et apporter des modifications à l'aide de la barre d'outils, par exemple :

1. Ajoutez ou supprimez une colonne à l'aide des boutons et .
2. Modifiez l'ordre des colonnes à l'aide des boutons et .
3. Cliquez sur **Finish** pour terminer la création du schéma avancé.

Le nouveau schéma apparaît dans le **Repository**, sous le nœud Web Service correspondant. Vous pouvez à présent le déposer dans l'espace de modélisation graphique de votre Job en tant que composant **tWebService**.

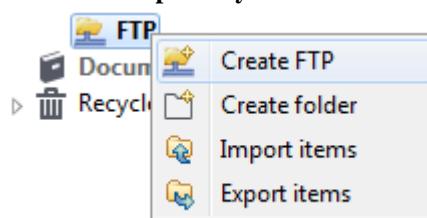
## 10.20. Centraliser une connexion FTP

Si vous vous connectez régulièrement à un serveur FTP vous pouvez centraliser les informations de connexion à cette base de données sous le nœud **Metadata** de la vue **Repository**.

### Configurer les propriétés générales

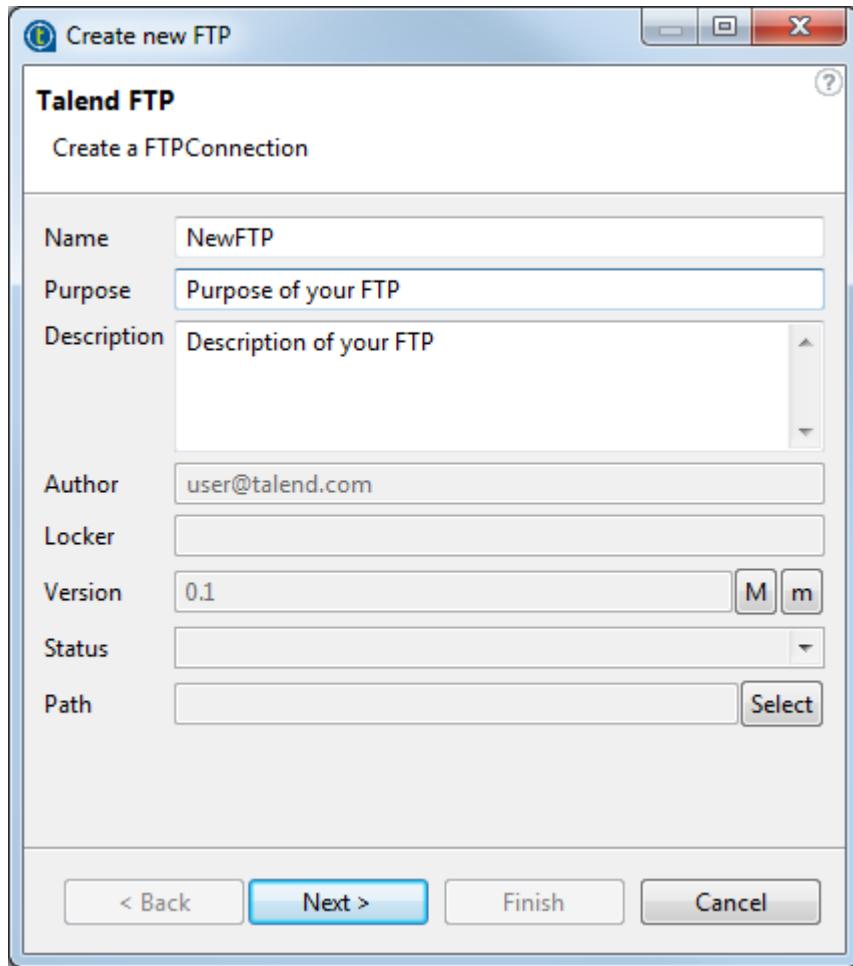
Pour créer une connexion à un serveur FTP, suivez les étapes ci-dessous.

1. Développez le nœud **Metadata** dans la vue **Repository**.



2. Cliquez-droit sur **FTP** puis sélectionnez **Create FTP** dans le menu contextuel.

Un assistant de connexion s'ouvre.



- Renseignez les propriétés générales du schéma, notamment les champs **Name** et **Description**.



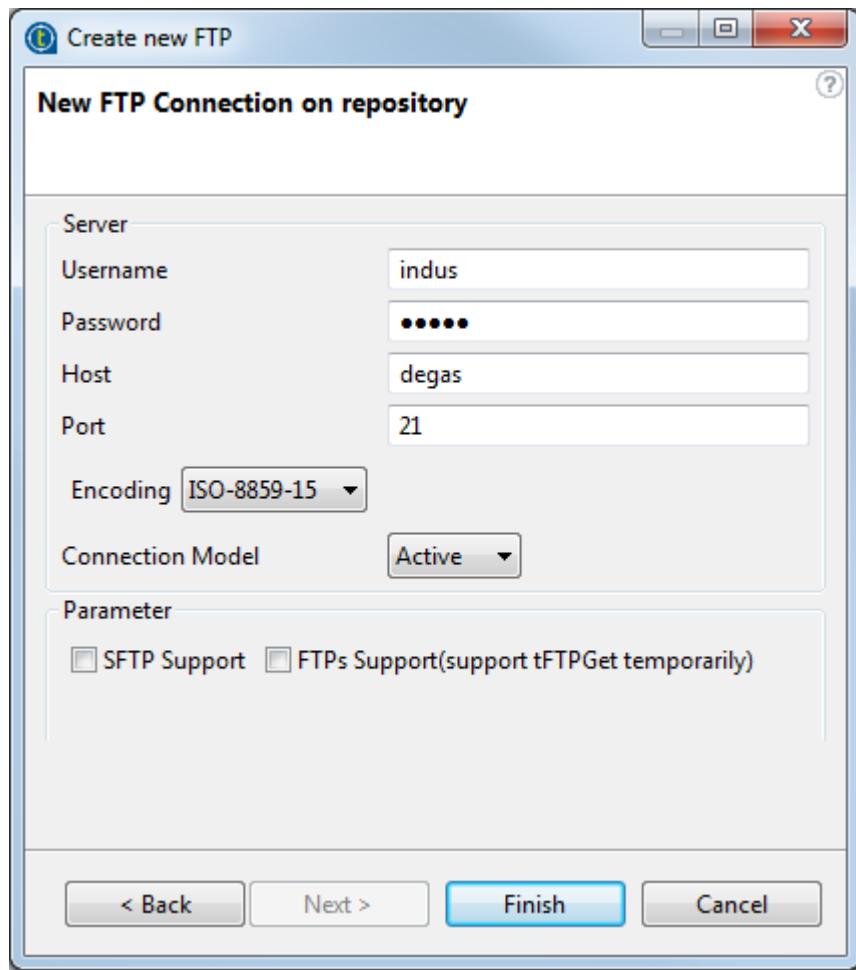
Le champ **Status** est un champ personnalisé que vous pouvez définir dans les Préférences (**Window > Preferences**). Pour plus d'informations concernant la définition des préférences, consultez [Configuration des préférences du Studio Talend](#).

- Cliquez sur **Next** une fois terminé. L'étape suivante vous permet de renseigner les données de connexion au serveur FTP.

## Se connecter à un serveur FTP

Dans cette étape, renseignez les informations et les paramètres de connexion à votre serveur FTP.

- Dans les champs **Username** et **Password**, saisissez respectivement votre identifiant de connexion au serveur et votre mot de passe.



2. Dans le champ **Host**, saisissez le nom de l'hôte de votre serveur FTP.
3. Dans le champ **Port**, saisissez le numéro du port correspondant.
4. Dans la liste **Encoding**, sélectionnez l'encodage utilisé.
5. Dans la liste **Connection Model**, sélectionnez le mode de connexion que vous souhaitez utiliser.
  - Sélectionnez **Passive** si vous souhaitez que le serveur FTP détermine le port de connexion pour le transfert des données.
  - Sélectionnez **Active** si vous souhaitez déterminer vous-même ce port.
6. Dans la zone **Parameter**, sélectionnez le type d'utilisation du serveur FTP. Pour une utilisation standard de votre serveur FTP, vous n'avez pas besoin de sélectionner une option.
  - Cochez la case **SFTP Support** pour utiliser votre serveur FTP via le protocole de sécurité SSH.  
Une liste **Authentication method** apparaît alors. Sélectionnez **Public key** ou **Password** selon ce que vous utilisez.
  - Cochez la case **FTPs Support** pour utiliser votre serveur via le protocole de sécurité SSL.
  - Cochez la case **Use Socks Proxy** si vous souhaitez utiliser un proxy Socks, puis renseignez les informations du proxy (respectivement le nom de l'hôte, le numéro du port, l'identifiant et le mot de passe de l'utilisateur).
7. Cliquez sur **Finish** pour fermer l'assistant.

Toutes les connexions ainsi créées apparaissent sous le nœud des connexions aux serveurs FTP, dans la vue **Repository**.

Vous pouvez glisser-déposer la métadonnée du **Repository** dans l'espace de modélisation graphique. Une boîte de dialogue s'ouvre alors et vous pouvez choisir quel composant utiliser dans votre Job.

Pour plus d'informations concernant l'action de déposer des métadonnées dans l'espace de modélisation graphique, consultez [Comment utiliser les métadonnées centralisées dans un Job](#).

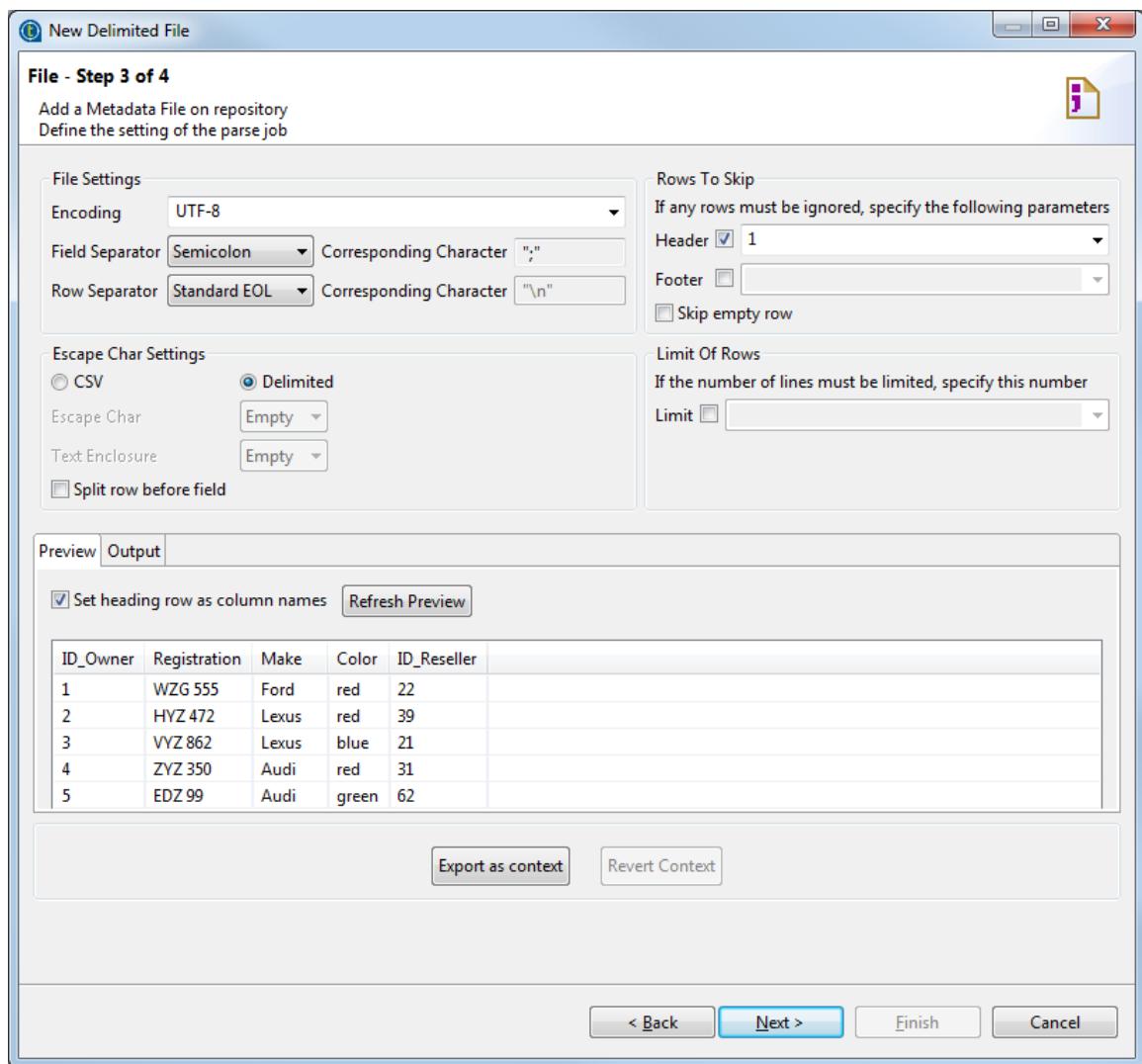
## 10.21. Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion

Pour chaque métadonnée (fichier, base de données, ou Talend MDM par exemple), vous pouvez exporter les informations de connexion en tant que nouveau groupe de contextes dans le **Repository**, afin de le réutiliser dans d'autres connexions ou dans différents Jobs, ou encore pour réutiliser des variables d'un groupe de contextes afin de configurer votre métadonnée.

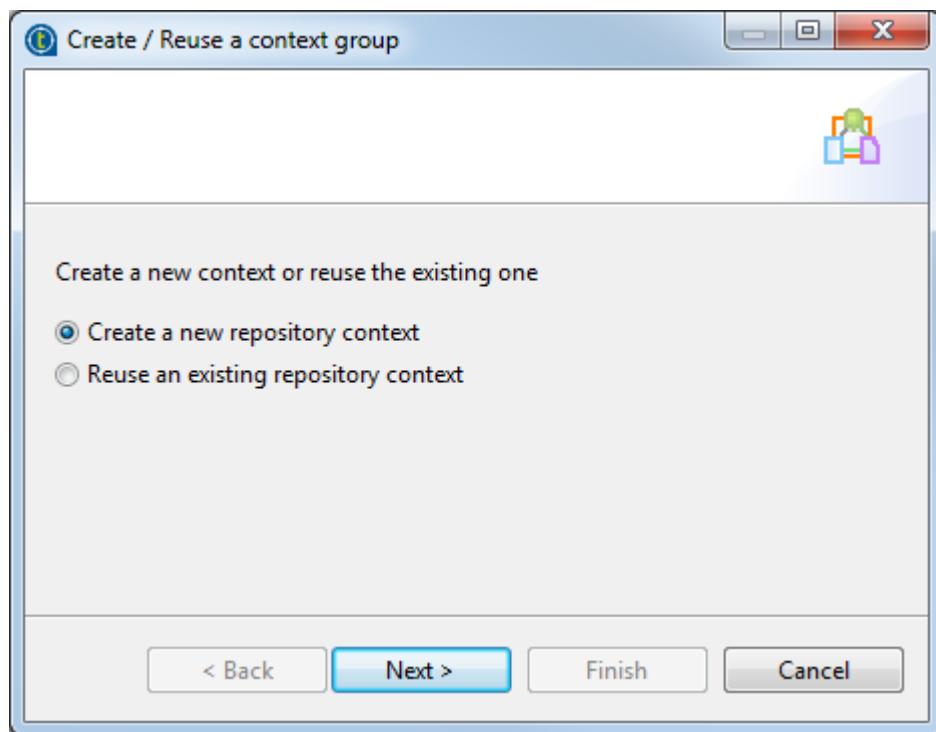
### 10.21.1. Exporter des informations de connexion en tant que variables de contexte

Pour exporter des informations de connexion en tant que variables de contexte dans un nouveau groupe de contextes dans le **Repository**, effectuez les étapes suivantes :

1. Après avoir créé ou modifié une métadonnée dans l'assistant, cliquez sur **Export as context**.

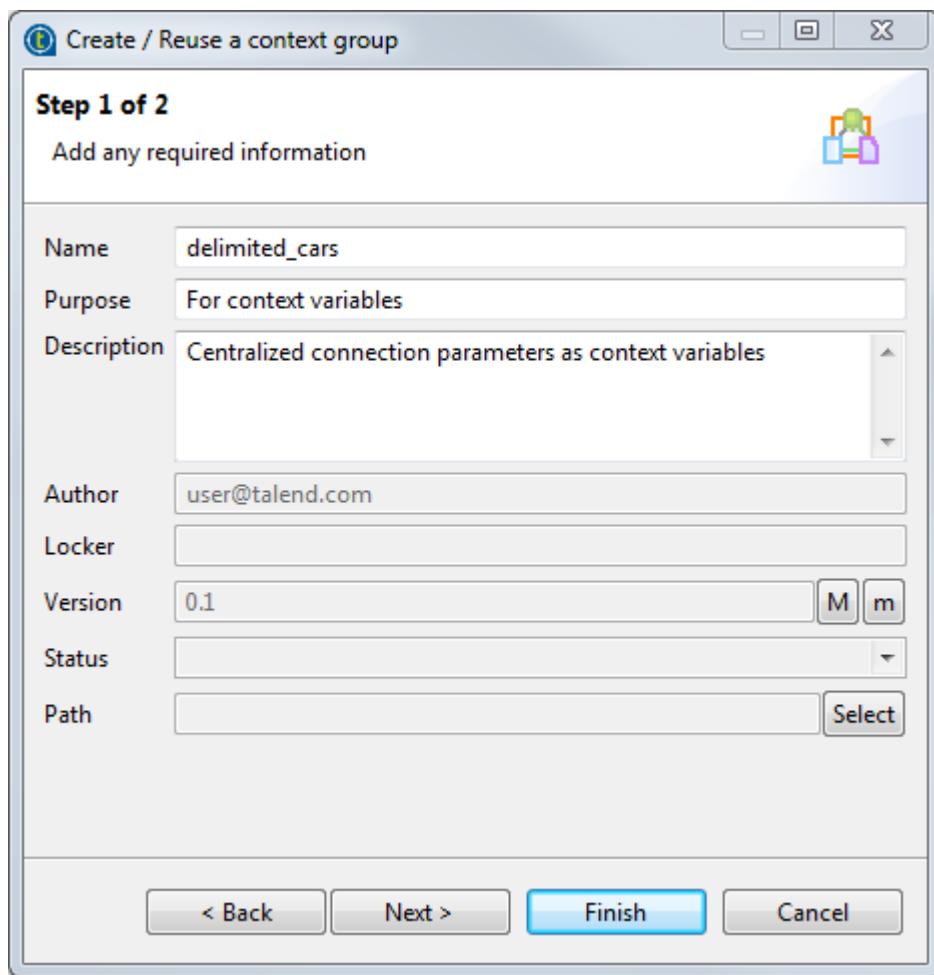


2. Dans l'assistant [Create / Reuse a context group] qui s'ouvre, sélectionnez **Create a new repository context** et cliquez sur **Next**.



3. Saisissez un nom pour le groupe de contextes à créer et ajoutez les informations générales, comme une description, si nécessaire.

Le nom de la métadonnée est proposé par l'assistant comme nom du groupe de contextes et les informations que vous fournissez dans le champ **Description** apparaissent sous la forme d'une infobulle lorsque vous placez votre souris sur le groupe de contexte dans le **Repository**.



4. Cliquez sur **Next** pour créer et voir le groupe de contextes ou cliquez sur **Finish** afin de terminer la création du contexte et retourner directement à l'assistant de connexion.

Dans cet exemple, cliquez sur **Next**.

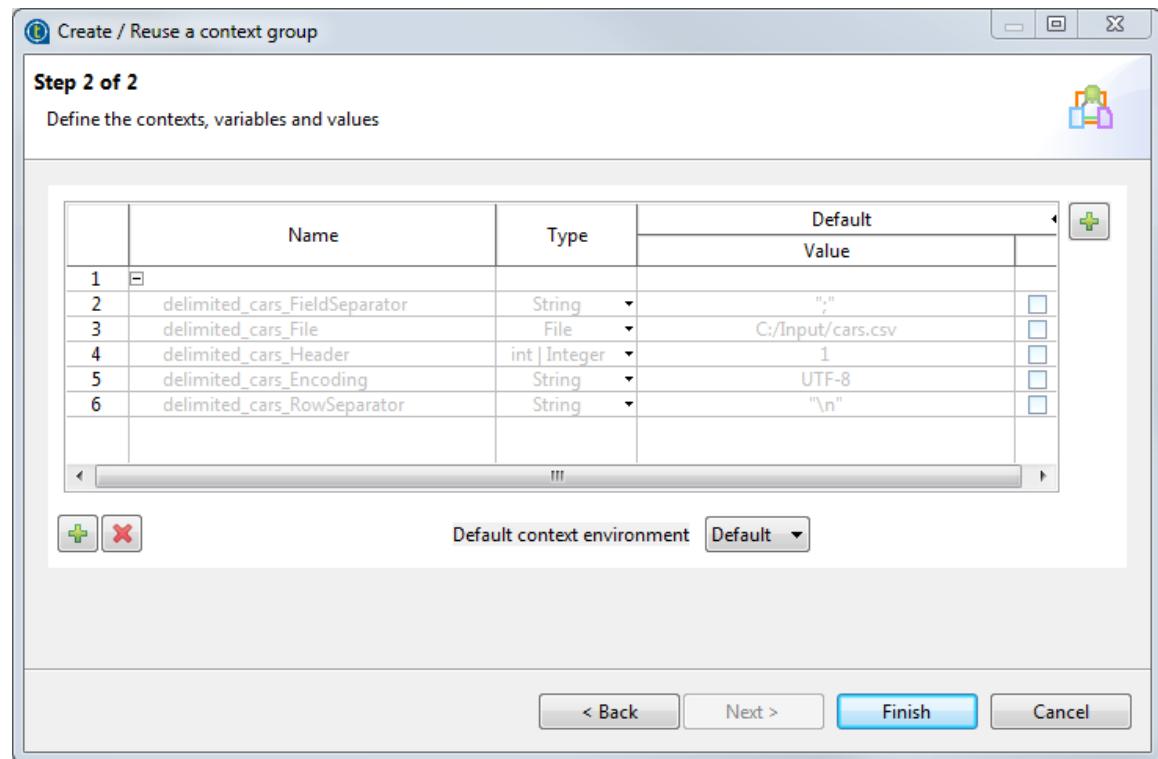
5. Vérifiez les résultats de génération du groupe de contextes.

Pour modifier les variables de contexte, allez dans le nœud **Contexts** de la vue **Repository**, cliquez-droit sur le nouveau groupe de contextes et sélectionnez **Edit context group** pour ouvrir l'assistant [**Create / Edit a context group**] après fermeture de l'assistant de connexion.

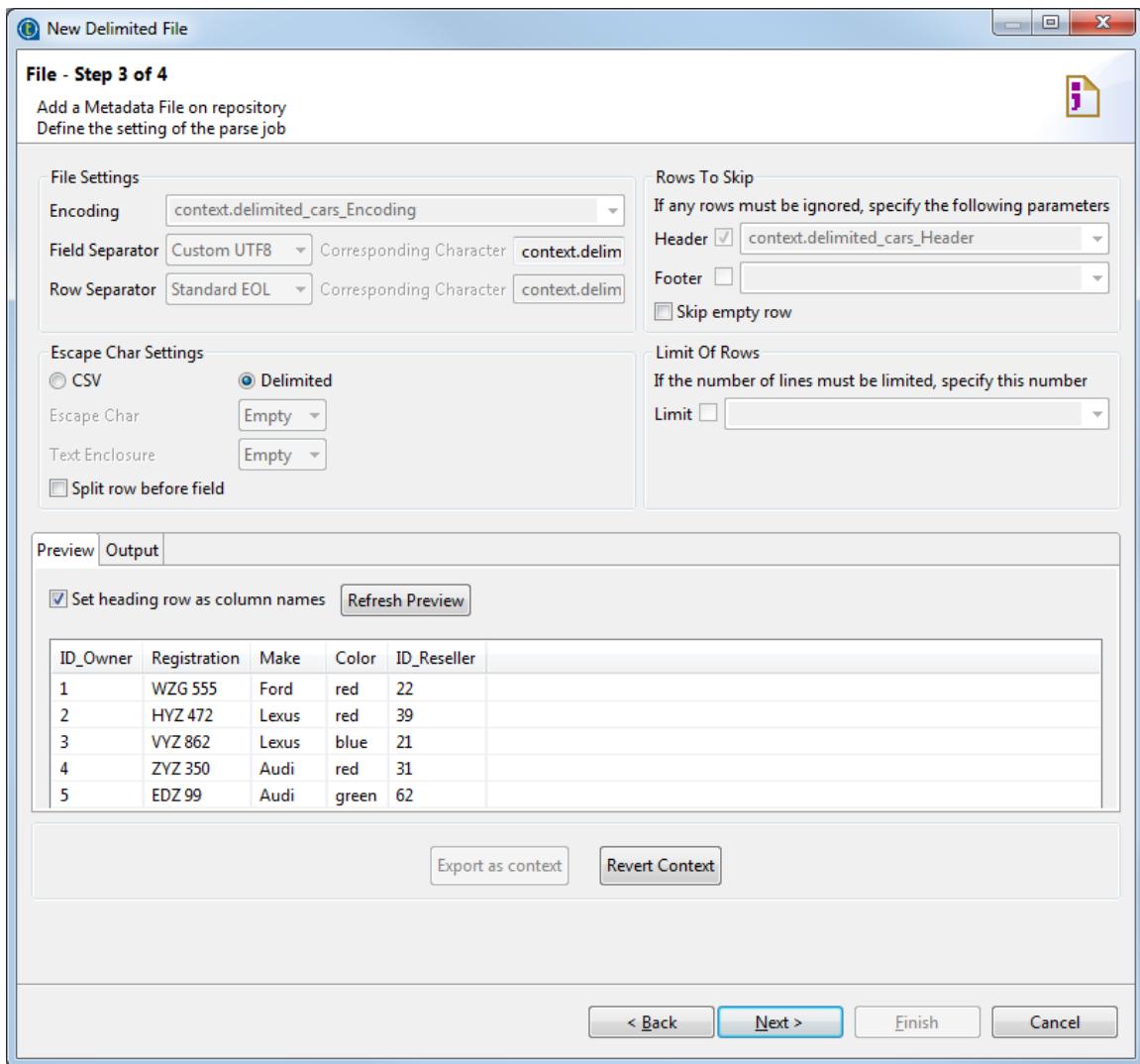
Pour modifier le contexte par défaut ou ajouter de nouveaux contextes, cliquez sur le bouton **[+]** dans le coin supérieur droit de l'assistant.

Pour ajouter une nouvelle variable de contexte, cliquez sur le bouton **[+]** au bas de l'assistant.

Pour plus d'informations concernant la gestion des contextes et des variables, consultez [\*Utiliser les contextes et les variables\*](#).



6. Cliquez sur **Finish** afin de terminer la création du contexte et retourner à l'assistant de connexion.



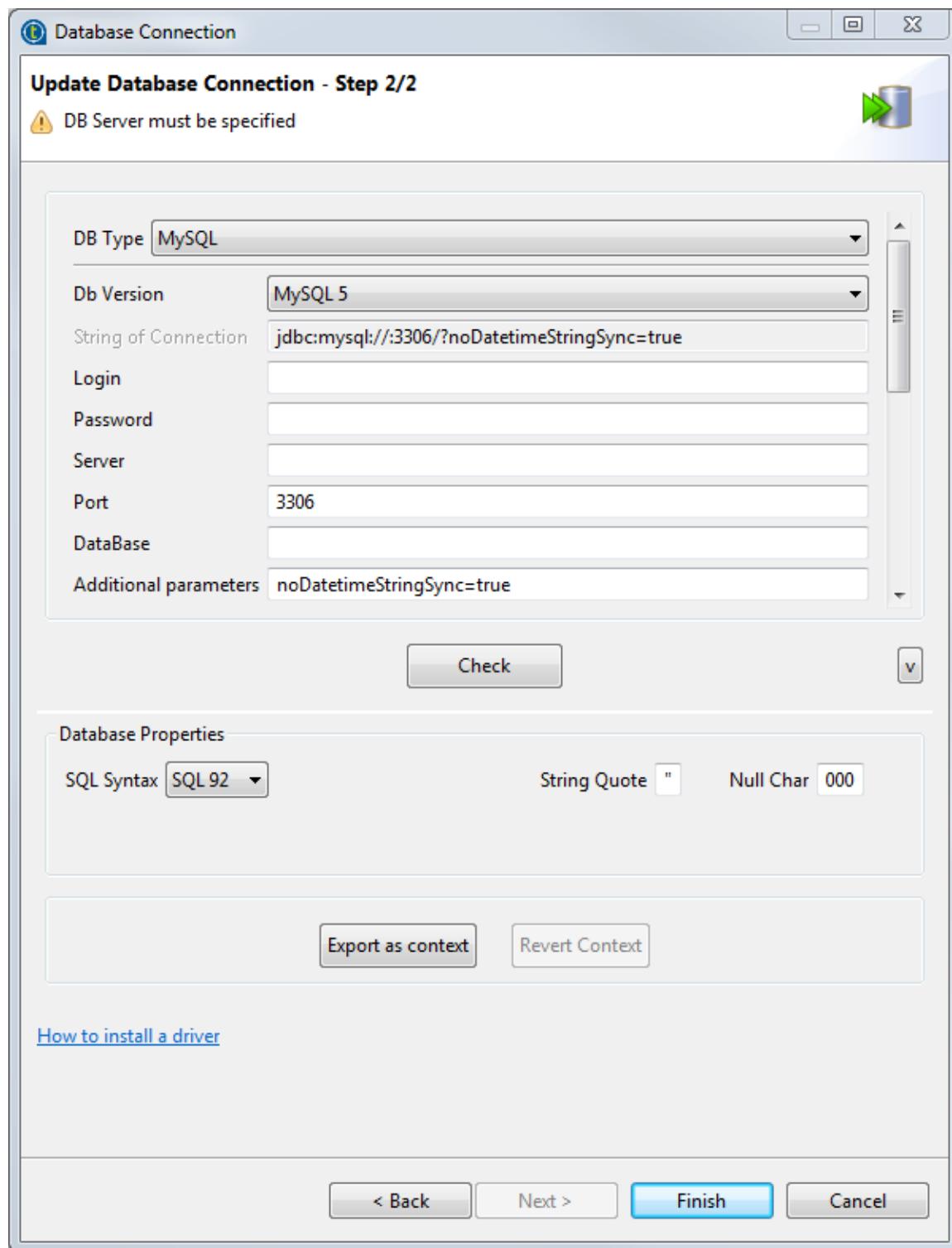
Les champs des informations de connexion de l'assistant sont paramétrés avec les variables de contexte.

Pour supprimer les paramètres définis, cliquez sur le bouton **Revert Context**.

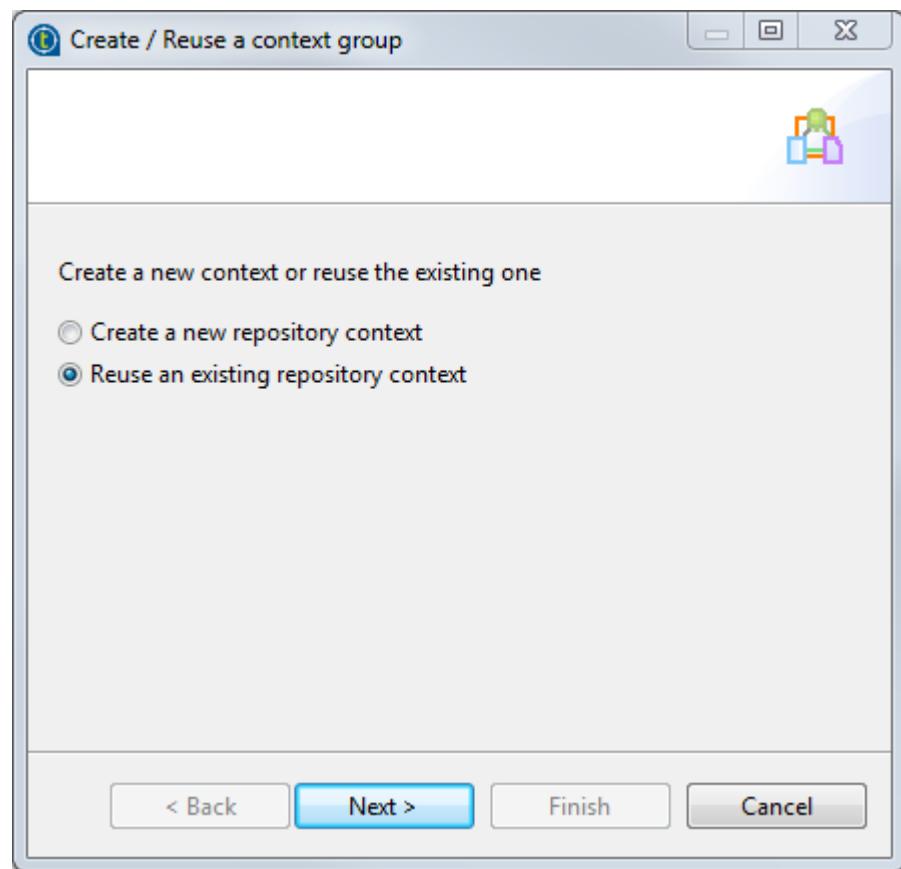
## 10.21.2. Utiliser les variables d'un groupe de contextes existant afin de configurer une connexion

Pour utiliser les variables d'un groupe de contextes existant stocké dans le référentiel afin de configurer une connexion, suivez les étapes ci-dessous :

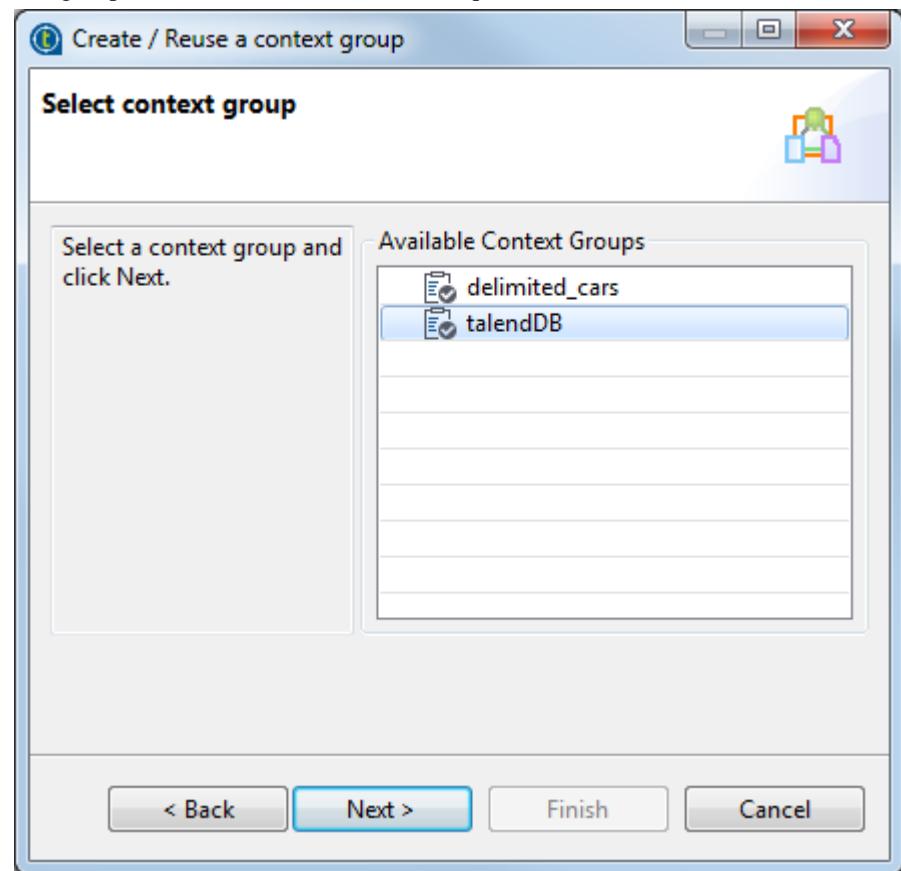
- Après avoir créé ou modifié une métadonnée dans l'assistant, cliquez sur **Export as context**.



2. Dans l'assistant [Create / Reuse a context group] qui s'ouvre, sélectionnez **Reuse an existing repository context** et cliquez sur Next.

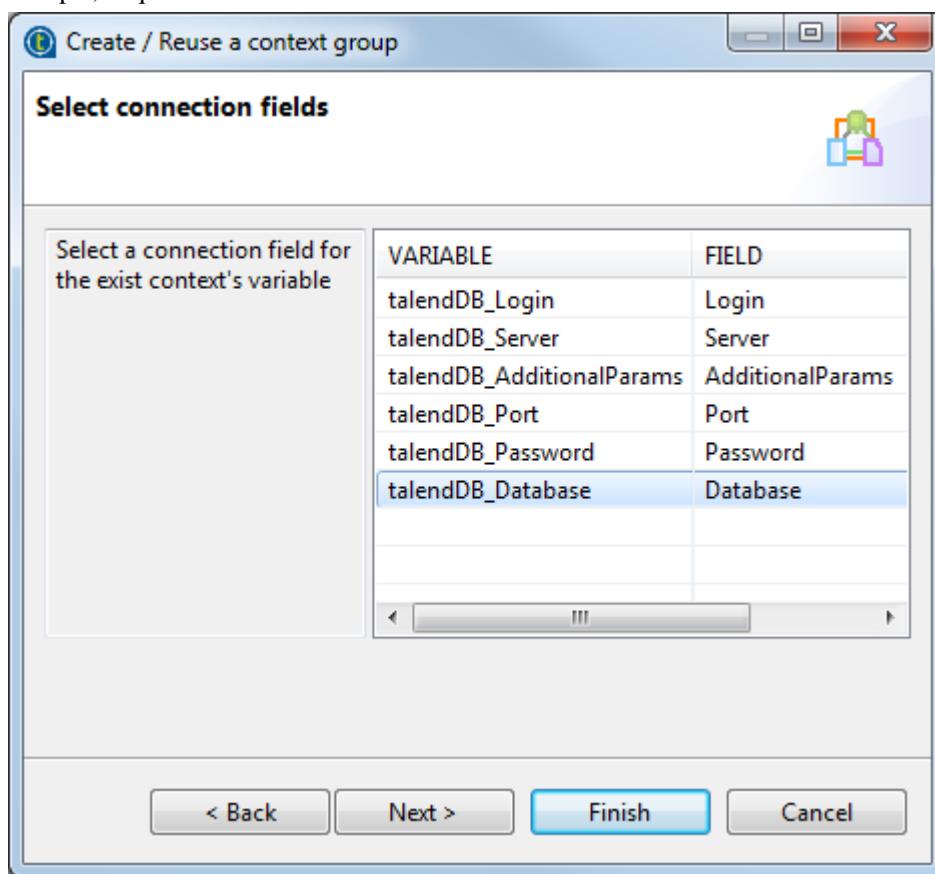


3. Sélectionnez un groupe de contextes dans la liste et cliquez sur **Next**.



4. Pour chaque variable, sélectionnez le champ correspondant des informations de connexion puis cliquez sur **Next** pour visualiser et modifier les variables de contexte, ou cliquez sur **Finish** pour afficher directement les résultats de configuration de la connexion.

Dans cet exemple, cliquez sur **Next**.

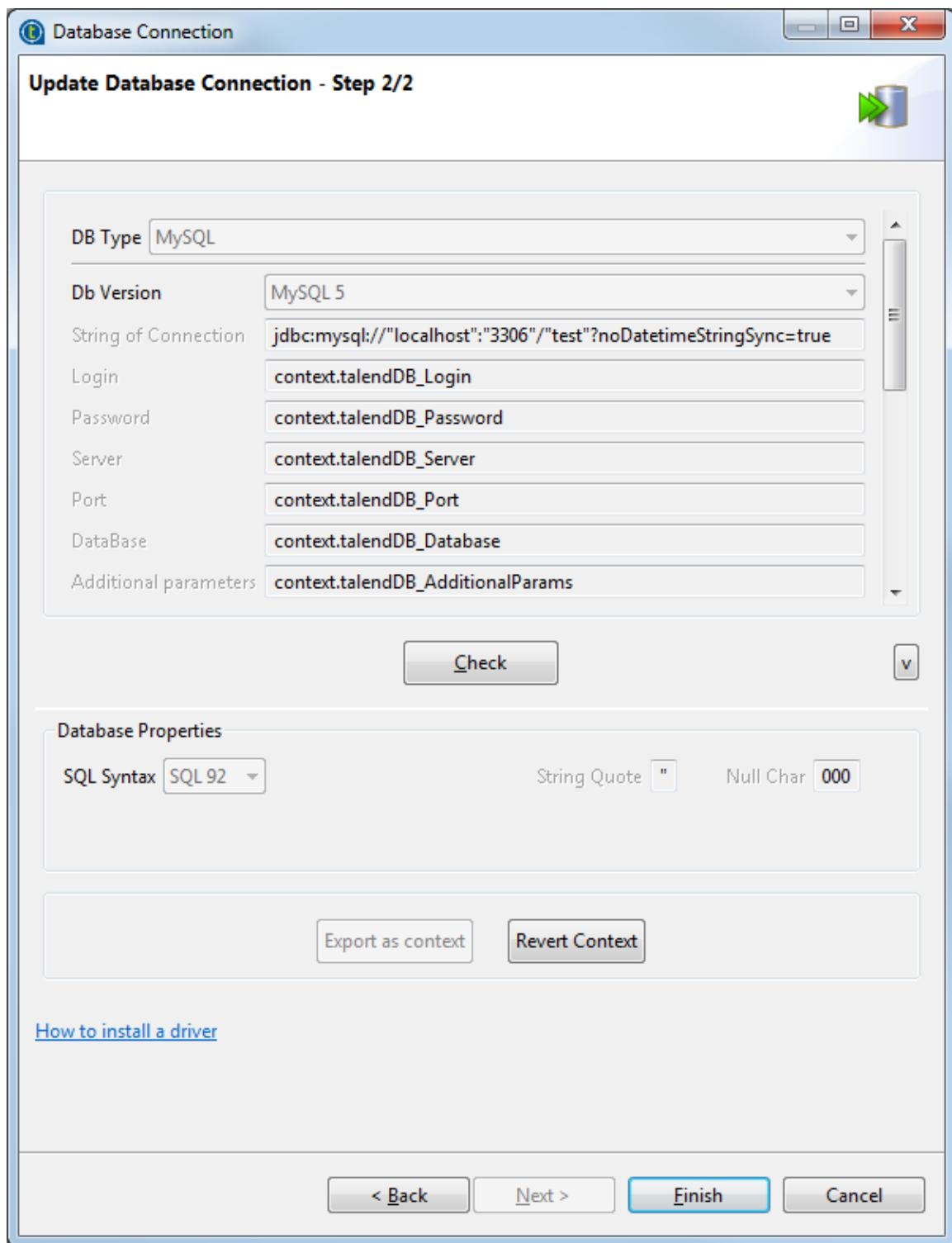


5. Modifiez les contextes et/ou les variables de contexte si nécessaire. Si vous avez effectué des modifications, votre groupe de contextes centralisé est automatiquement mis à jour.

Pour plus d'informations concernant la gestion des contextes et des variables, consultez [Utiliser les contextes et les variables](#).

Name	Type	Test		Prod	
		Value		Value	Prompt
talendDB_Login	String	"root"	<input type="checkbox"/>	"192.168.30.110"	<input checked="" type="checkbox"/> talendDB_Login?
talendDB_Server	String	"localhost"	<input type="checkbox"/>		<input checked="" type="checkbox"/> talendDB_Server?
talendDB_AdditionalParams	String	noDatetimeStringSync=true	<input type="checkbox"/>	noDatetimeStringSync=true	<input type="checkbox"/> talendDB_AdditionalParam?
talendDB_Port	String	"3306"	<input type="checkbox"/>	"3308"	<input type="checkbox"/> talendDB_Port?
talendDB_Password	Password	*****	<input type="checkbox"/>	*****	<input type="checkbox"/> talendDB_Password?
talendDB_Database	String	"test"	<input type="checkbox"/>	"prod_db"	<input type="checkbox"/> talendDB_Database?

6. Cliquez sur **Finish** pour valider la réutilisation du contexte et retourner à l'assistant de connexion.



Les paramètres de la connexion dans l'assistant sont configurés avec les variables de contexte.

Pour supprimer les paramètres définis, cliquez sur le bouton **Revert Context**.

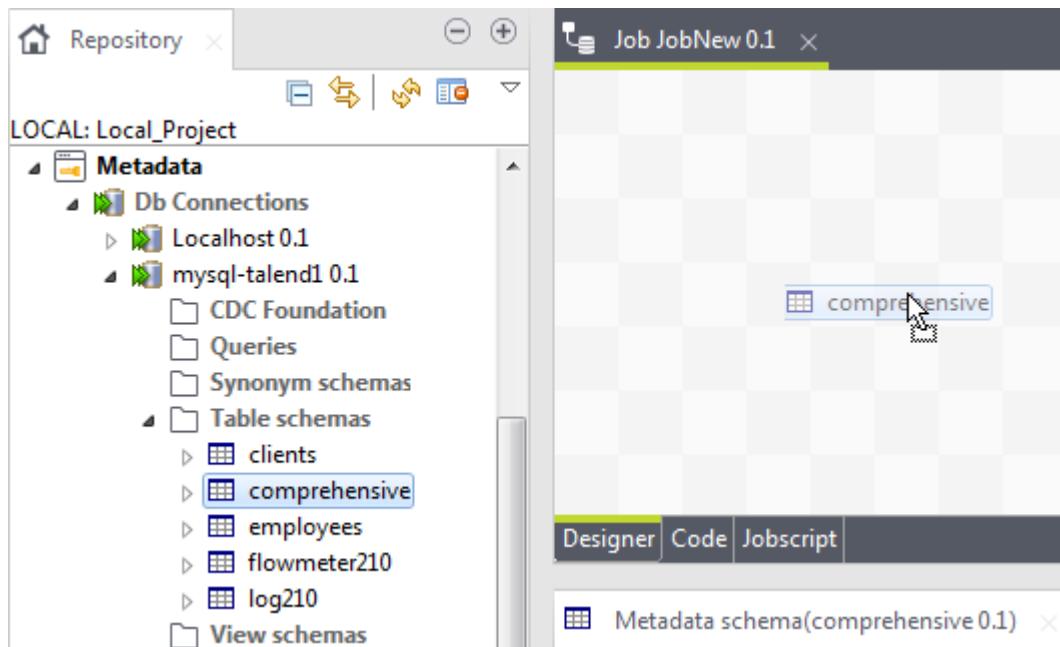
## 10.22. Comment utiliser les métadonnées centralisées dans un Job

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

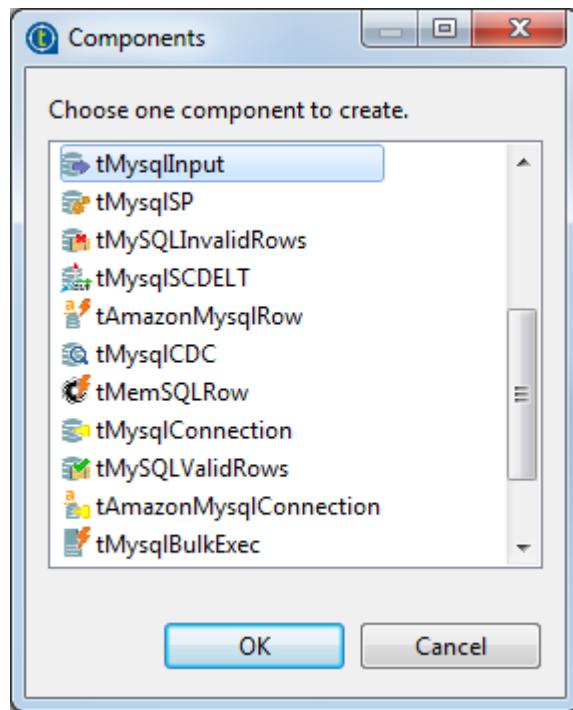
Différents assistants vous permettront de centraliser les connexions et les métadonnées des schémas, dans la vue Repository. Pour plus d'informations concernant les assistants de gestion des métadonnées, consultez [Gestion des métadonnées dans l'intégration de données](#).

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. Dans la perspective **Integration**, développez le noeud **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** enfoncee pour déposer l'élément sélectionné dans l'éditeur graphique et ainsi l'ajouter à votre Job Design.
- **Output** : Maintenez la touche **Ctrl** enfoncee pour déposer le composant dans l'éditeur et ainsi l'ajouter à votre Job Design.

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



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





## Chapitre 11. Gestion des métadonnées pour Talend Big Data

Dans la perspective **Integration** du studio, le dossier **Metadata** stocke les informations réutilisables concernant les fichiers, les bases de données et/ou les systèmes dont vous avez besoin pour créer vos Jobs.

Ce chapitre contient des procédures vous aidant à utiliser les assistants appropriés pour créer et gérer différents éléments de métadonnées pouvant être utilisés dans vos Job Designs Big Data.

- Pour plus de détails concernant les connexions à une base de données NoSQL, consultez [\*Gérer les métadonnées NoSQL\*](#).
- Pour plus de détails concernant les connexions Hadoop, consultez [\*Gérer les métadonnées Hadoop\*](#).

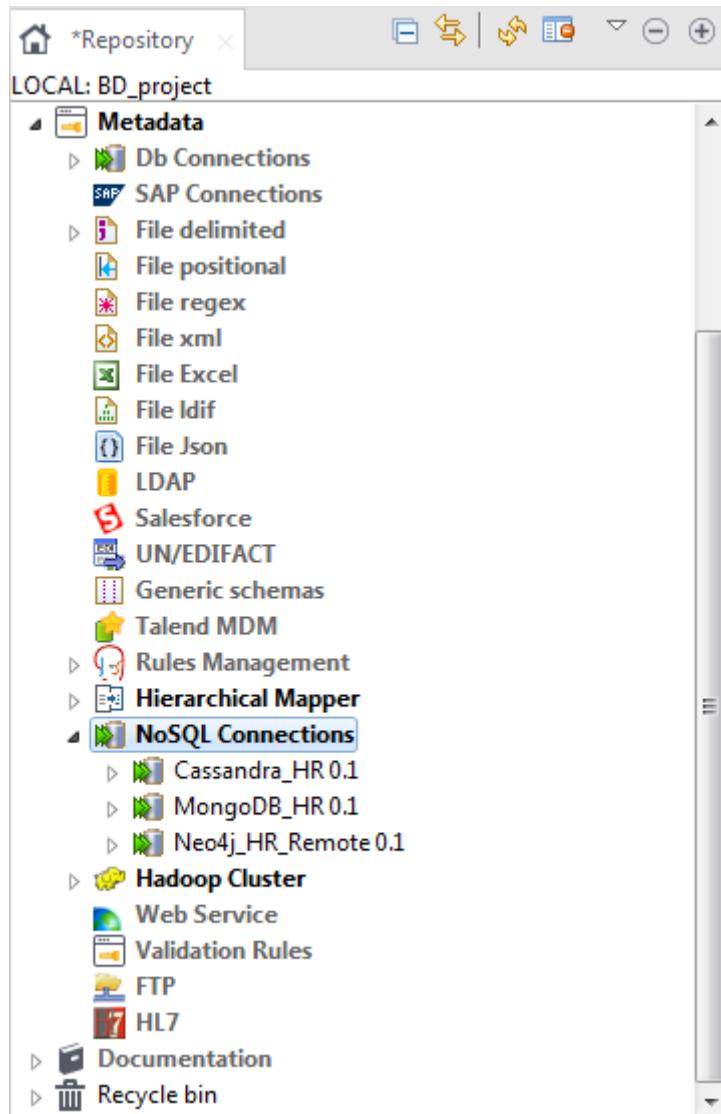
Pour plus d'informations concernant les autres types d'assistants de métadonnées, consultez [\*Gestion des métadonnées dans l'intégration de données\*](#).

Avant de commencer un processus de gestion de données, vous devez vous être familiarisé avec l'interface graphique de votre studio. Pour plus d'informations, consultez l'annexe traitant les éléments de l'interface graphique.

## 11.1. Gérer les métadonnées NoSQL

Dans la vue **Repository**, le nœud **NoSQL Connections** dans le dossier **Metadata** regroupe les métadonnées des connexions aux bases de données NoSQL telles que Cassandra, MongoDB et Neo4j. Cela vous permet de centraliser les propriétés de connexion que vous avez définies et de les réutiliser dans vos Job designs qui utilisent les composants des bases de données NoSQL (les composants Cassandra, MongoDB et Neo4j).

Cliquez sur le nœud **Metadata** dans la vue **Repository** pour développer le dossier. Chaque nœud regroupe les connexions et schémas que vous avez paramétrés. Parmi ces nœuds de connexions, se trouve le nœud **NoSQL Connections**.



Les sections suivantes expliquent en détail comment utiliser le nœud **NoSQLConnections** pour paramétriser :

- une connexion Cassandra,
- une connexion MongoDB et
- une connexion Neo4j.

## 11.1.1. Centraliser les métadonnées de Cassandra

Si vous avez souvent besoin de manipuler des données d'une base de données Cassandra, alors il peut être intéressant pour vous de centraliser la connexion à une base de données Cassandra et les détails de schémas dans un dossier **Metadata**, dans la vue **Repository**.

La procédure d'installation des métadonnées de Cassandra est constituée de deux tâches majeures différentes mais liées :

1. Créer une connexion à une base de données Cassandra.
2. Récupérer les schémas Cassandra qui vous intéressent.

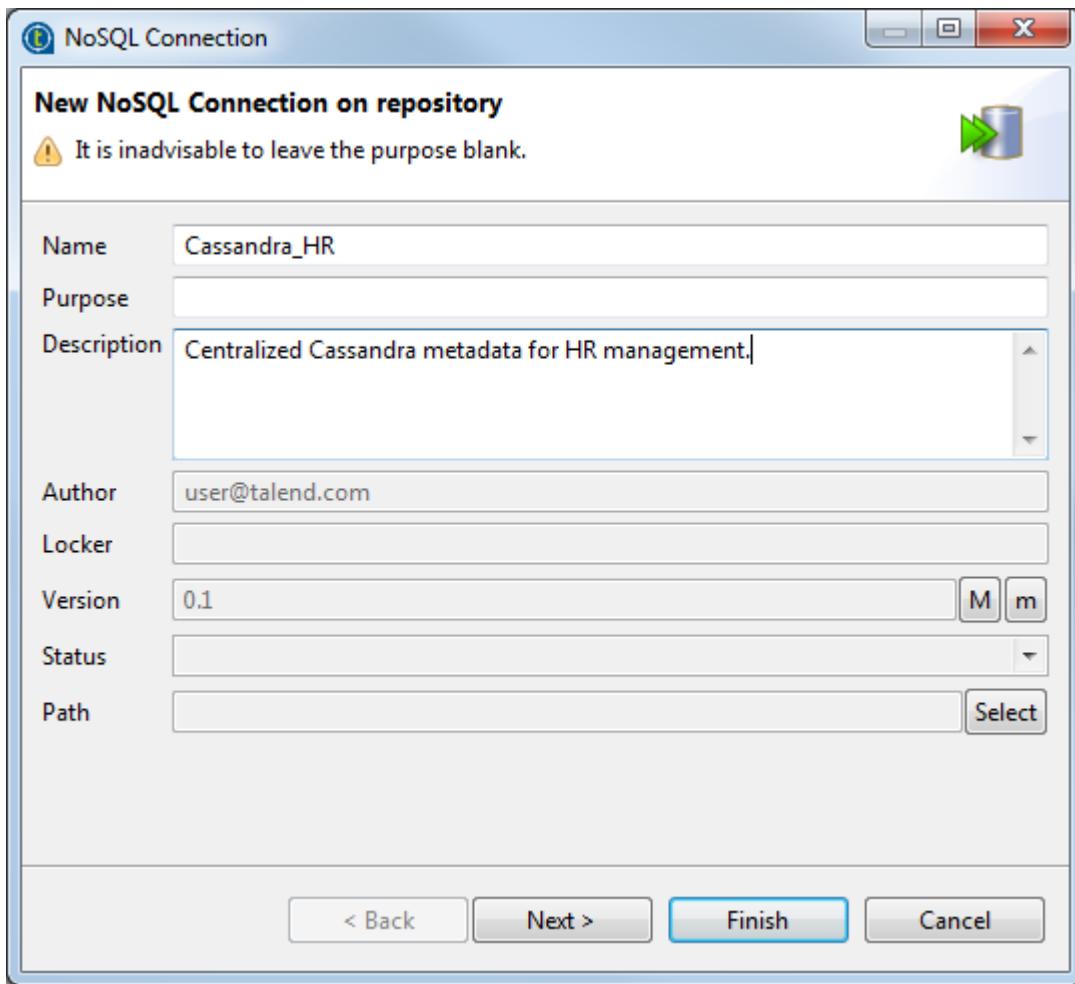
### Prérequis :

- Tous les modules externes requis manquant dans le *Studio Talend* en raison des restrictions de licence doivent être installés. Pour plus d'informations, consultez le *Guide d'installation et de migration Talend*.

### 11.1.1.1. Créez une connexion à une base de données Cassandra

1. Dans la vue **Repository**, développez le nœud **Metadata**, cliquez-droit sur **NoSQL Connection** et sélectionnez **Create Connection** dans le menu contextuel. L'assistant de connexion s'ouvre.
2. Dans l'assistant de connexion, renseignez les propriétés générales de la connexion que vous avez besoin de créer, telles que le nom (**Name**), le but (**Purpose**) et la description (**Description**).

Les informations que vous renseignez dans le champ **Description** apparaissent dans une info-bulle lorsque vous bougez le pointeur de votre souris au-dessus de la connexion.



Lorsque vous avez terminé, cliquez sur **Next** pour passer à l'étape suivante.

3. Sélectionnez **Cassandra** dans la liste **DB Type** ainsi que la version de la base de données Cassandra à laquelle vous vous connectez dans la liste **DB Version**, puis spécifiez les détails suivants :
  - Dans la liste **API type**, sélectionnez **Datastax** pour utiliser CQL 3 (Cassandra Query Language) avec Cassandra ou sélectionnez **Hector** pour utiliser CQL 2.

Notez que l'API **Hector** est dépréciée depuis la version 2.0 de Cassandra, mais est toujours disponible pour une utilisation dans le Studio, à des fins de flexibilité de version du langage de requête à utiliser avec Cassandra 2.0.0.

- Saisissez le nom d'hôte ou l'adresse IP du serveur Cassandra dans le champ **Server**.
- Saisissez le numéro de port du serveur Cassandra dans le champ **Port**.

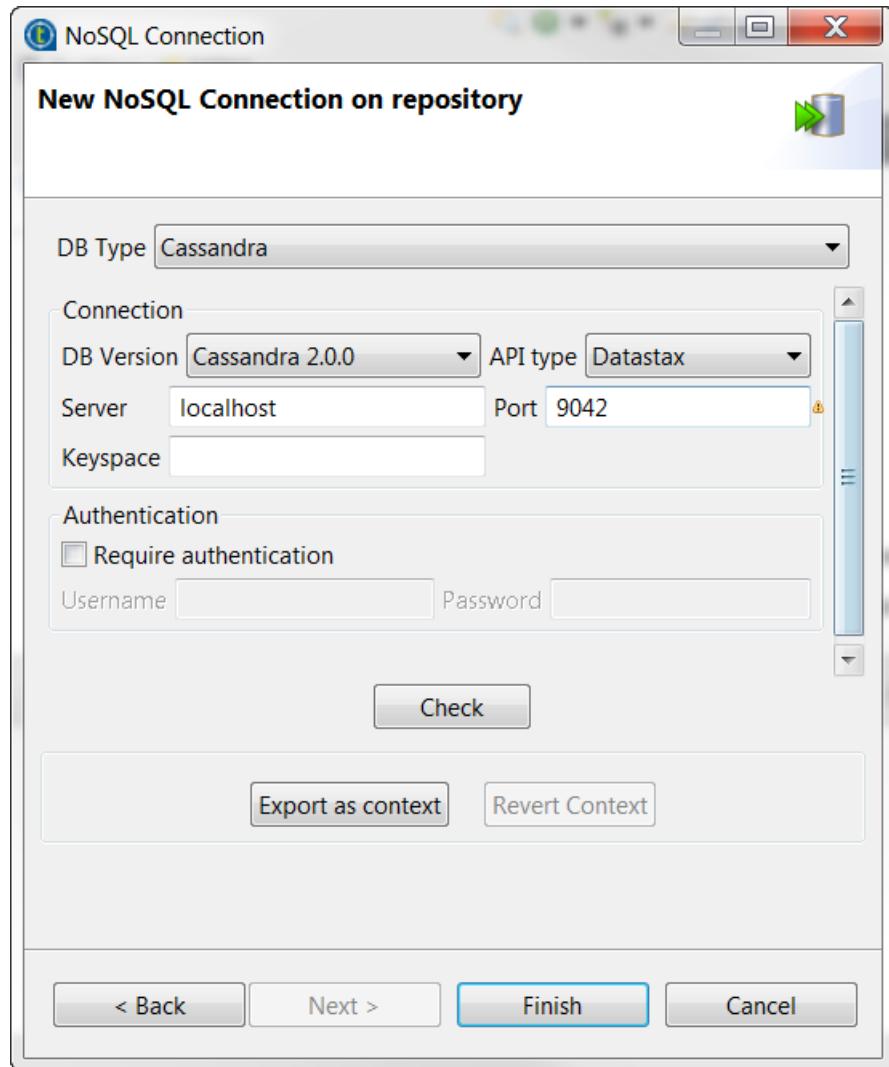


L'assistant peut se connecter à votre base de données Cassandra sans que vous ayez à spécifier un port. Le port que vous fournissez ici est uniquement utilisé dans le composant Cassandra que vous déposez dans l'espace de modélisation graphique à partir de cette connexion centralisée.

- Si vous souhaitez restreindre votre connexion Cassandra à un seul keyspace en particulier, saisissez le keyspace dans le champ **Keyspace**.

Si vous laissez ce champ vide, l'assistant listera les familles de colonnes de tous les keyspaces existants de la base de données connectée lors de la récupération des schémas.

- Si votre serveur Cassandra requiert des informations d'authentification pour accéder à la base de données, cochez la case **Require authentication** et renseignez vos nom d'utilisateur et mot de passe dans les champs correspondants.



4. Cliquez sur le bouton **Check** afin de vous assurer que la connexion fonctionne.

5. Cliquez sur **Finish** pour valider les paramètres.

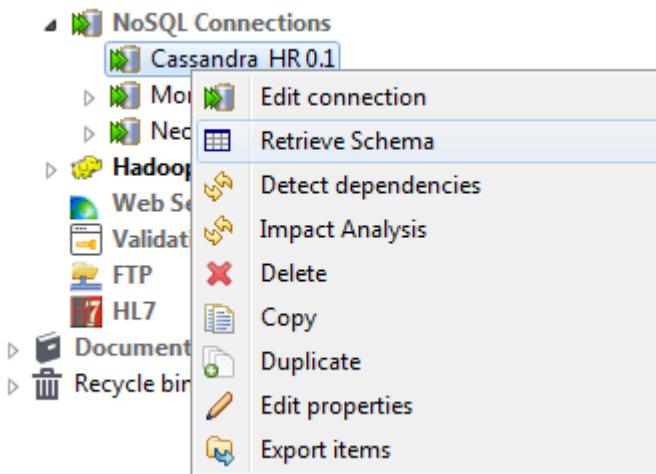
La connexion à la base de données Cassandra nouvellement créée apparaît sous le nœud **NoSQL Connection** dans la vue **Repository**. Vous pouvez maintenant la déposer dans votre espace de modélisation graphique en tant que composant Cassandra, mais vous devez tout de même définir les informations de schéma là où cela est nécessaire.

Vous devez ensuite récupérer un ou plusieurs schéma(s) qui vous intéresse(nt) pour votre connexion.

### 11.1.1.2. Récupérer les schémas

Dans cette étape, vous allez récupérer les schémas qui vous intéressent de la base de données connectée Cassandra.

- Dans la vue Repository, cliquez-droit sur la connexion nouvellement créée et sélectionnez **Retrieve Schema** dans le menu contextuel.



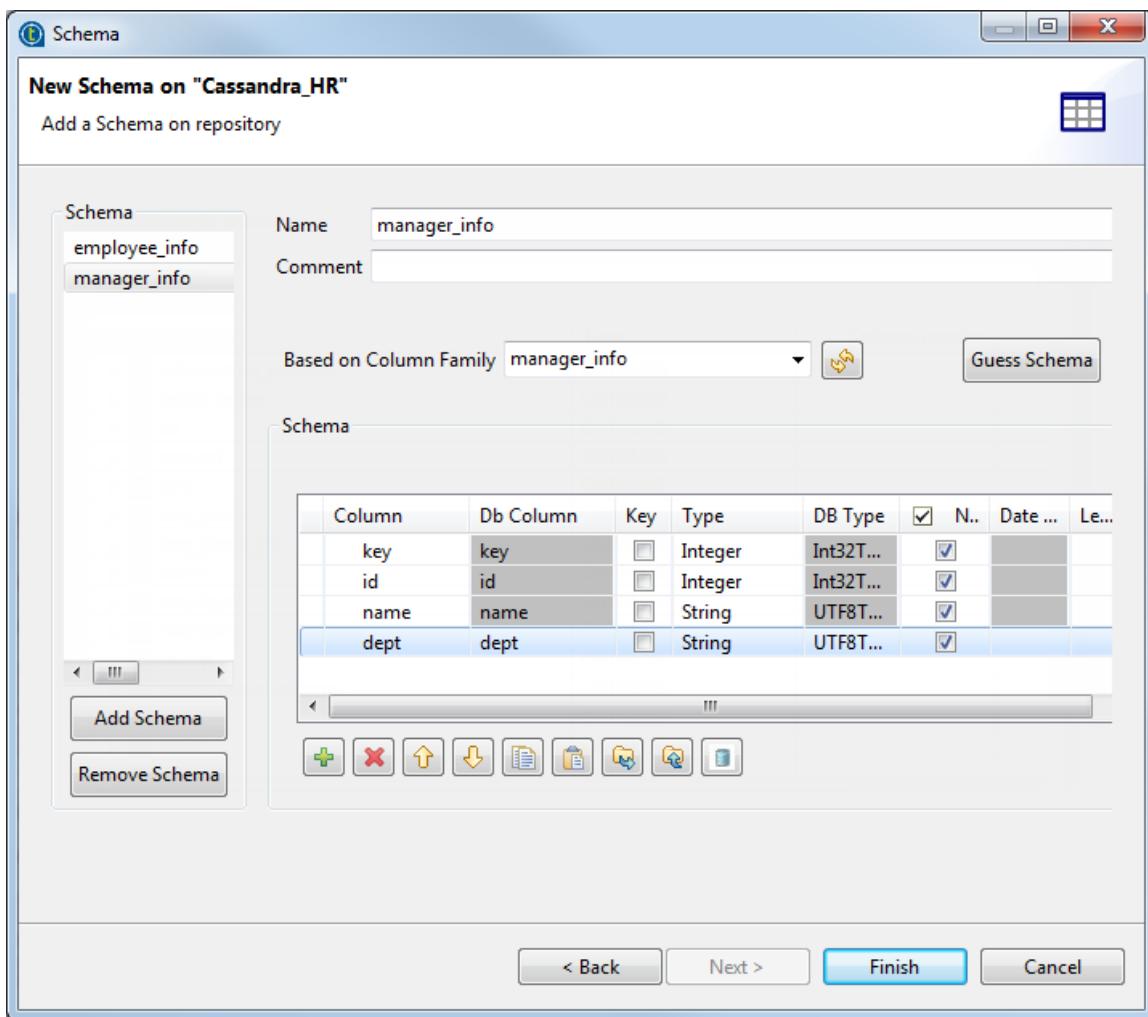
L'assistant ouvre une nouvelle vue qui liste toutes les familles de colonnes disponibles du keyspace spécifié, ou alors tous les keyspace disponibles si vous n'en avez spécifié aucun lors de l'étape précédente.

- Développez le keyspace, ou alors les keyspace(s) qui vous intéresse(nt) si vous n'avez pas spécifié de keyspace lors de l'étape précédente, comme c'est le cas dans cet exemple, puis sélectionnez la ou les famille(s) de colonnes qui vous intéresse(nt).

Name	Type	Column Number	Creation Status
<input checked="" type="checkbox"/> employee	Keyspace		
<input checked="" type="checkbox"/> employee_info	Column Family	5	Success
<input type="checkbox"/> system_traces	Keyspace		
<input checked="" type="checkbox"/> manager	Keyspace		
<input checked="" type="checkbox"/> manager_info	Column Family	4	Success
<input type="checkbox"/> system	Keyspace		

- Cliquez sur **Next** pour passer à l'étape suivante de l'assistant dans laquelle vous pouvez éditer le ou les schéma(s) générés.

Par défaut, un schéma généré prend le nom de la famille de colonnes sur laquelle il est basé.



Selectionnez un schéma dans le panneau **Schema** pour afficher ses détails sur le côté droit et modifiez le schéma si nécessaire. Vous pouvez renommer n'importe quel schéma et personnaliser la structure d'un schéma selon vos besoins dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, de supprimer ou de remplacer des colonnes dans votre schéma, ou encore de remplacer le schéma avec le schéma défini dans un fichier XML.

Pour baser un schéma sur une autre famille de colonnes, sélectionnez le nom du schéma dans le panneau **Schema** et sélectionnez une nouvelle famille de colonnes dans la liste **Based on Column Family**. Cliquez ensuite le bouton **Guess schema** pour écraser le schéma avec celui de la famille de colonnes sélectionnée. Vous pouvez cliquer sur le bouton de rafraîchissement pour rafraîchir la liste des familles de colonnes.

Pour ajouter un nouveau schéma, cliquez sur le bouton **Add Schema** dans le panneau **Schema** pour créer un schéma vide que vous devez définir.

Pour supprimer un schéma, sélectionnez le nom du schéma dans le panneau **Schema** et cliquez sur le bouton **Remove Schema**.

Pour écraser les modifications que vous avez effectuées sur le schéma sélectionné par son schéma par défaut, cliquez sur **Guess schema**. Notez que toutes modifications que vous avez apportées au schéma seront perdues si vous cliquez sur ce bouton.

4. Cliquez sur **Finish** pour terminer la création du schéma. Les schéma créés apparaissent sous votre connexion Cassandra dans la vue **Repository**. Vous pouvez désormais déposer la connexion ou n'importe quel noeud de schéma sous cette connexion dans votre espace de modélisation graphique en tant que composant Cassandra. Toutes les informations sur les métadonnées seront automatiquement renseignées.

Si vous avez besoin de continuer à éditer un schéma, cliquez-droit sur ce schéma et sélectionnez **Edit Schema** dans le menu contextuel pour ouvrir de nouveau cet assistant et effectuer vos modifications.



*Si vous modifiez les schémas, assurez-vous que le type de données dans la colonne **Type** est correctement défini.*

## 11.1.2. Centraliser les métadonnées de MongoDB

Si vous avez souvent besoin de manipuler des données d'une base de données MongoDB, alors il peut être intéressant pour vous de centraliser la connexion à une base de données MongoDB et les détails de schémas dans le dossier **Metadata**, dans la vue **Repository**.

La procédure d'installation des métadonnées de MongoDB est constituée de deux tâches majeures mais liées :

1. Créer une connexion à une base de données MongoDB.
2. Récupérer les schémas MongoDB qui vous intéressent.

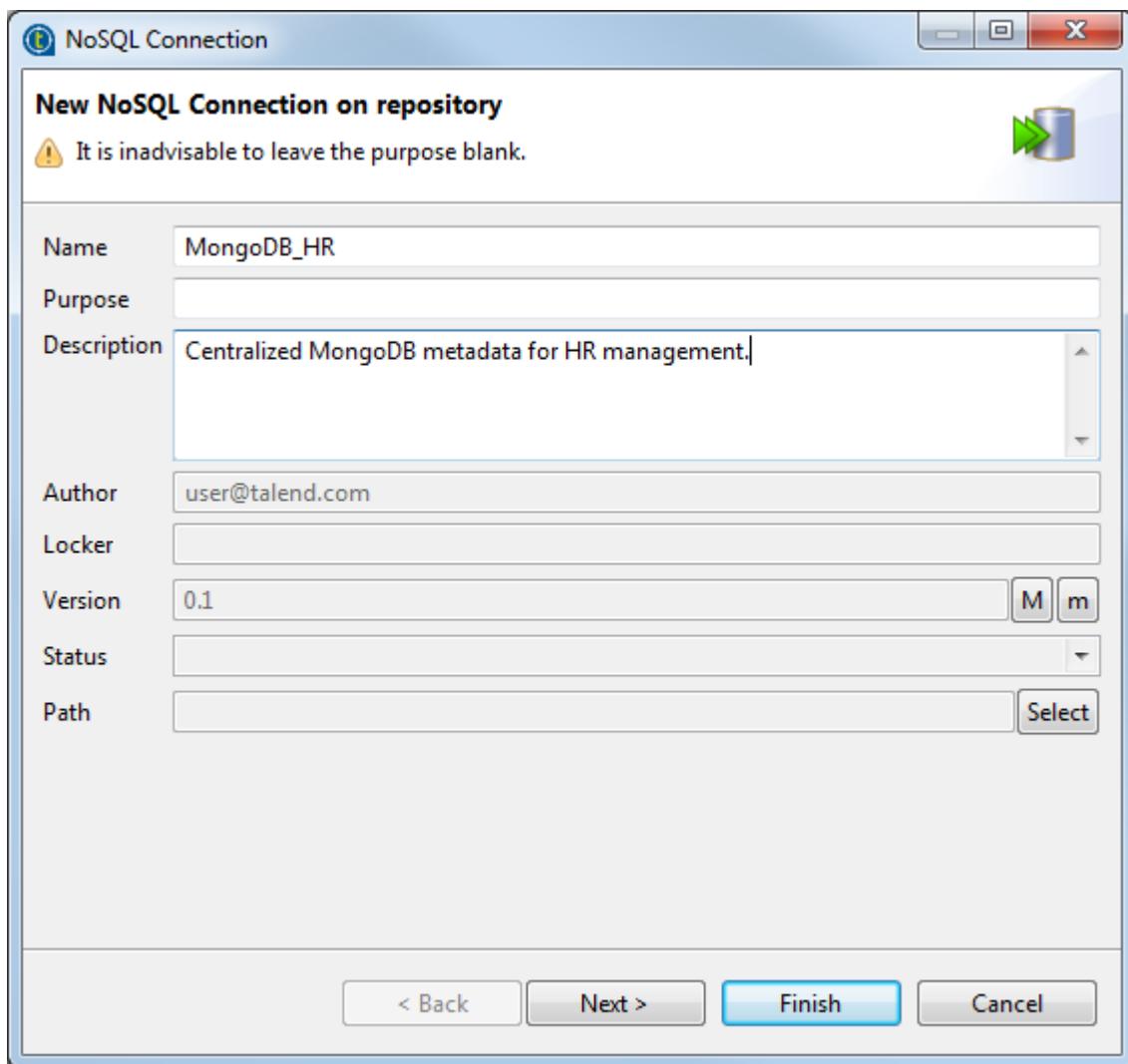
**Prérequis :**

- Tous les modules externes requis manquant dans le *Studio Talend* en raison des restrictions de licence doivent être installés. Pour plus d'informations, consultez le *Guide d'installation et de migration Talend*.

### 11.1.2.1. Crée une connexion à une base de données MongoDB

1. Dans la vue **Repository**, développez le nœud **Metadata**, cliquez-droit sur **NoSQL Connection** et sélectionnez **Create Connection** dans le menu contextuel. L'assistant de connexion s'ouvre.
2. Dans l'assistant de connexion, renseignez les propriétés générales de la connexion que vous avez besoin de créer, telles que le nom (**Name**), le but (**Purpose**) et la description (**Description**).

Les informations que vous renseignez dans le champ **Description** apparaissent dans une info-bulle lorsque vous bougez le pointeur de votre souris au-dessus de la connexion.



Lorsque vous avez terminé, cliquez sur **Next** pour passer à l'étape suivante.

3. Sélectionnez **MongoDB** dans la liste **DB Type** ainsi que la version de la base de données MongoDB à laquelle vous vous connectez dans la liste **DB Version**, puis spécifiez les détails suivants :

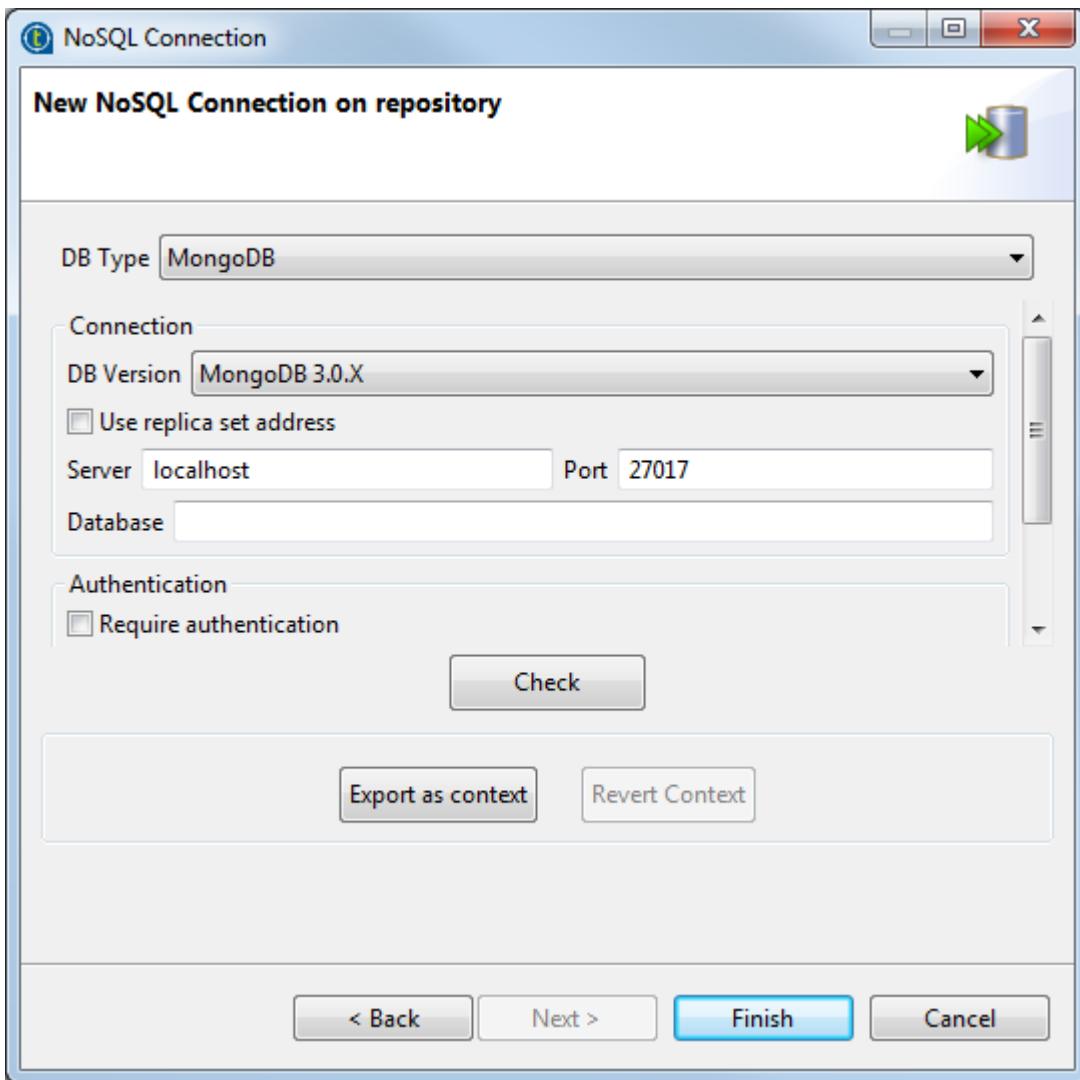
- Saisissez le nom d'hôte ou l'adresse IP ainsi que le numéro de port du serveur MongoDB dans les champs correspondants.

Si la base de données à laquelle vous vous connectez est répliquée sur différents serveurs d'un replica set, cochez la case **Use replica set address** et spécifiez les noms d'hôtes ou les adresses IP ainsi que les ports respectifs dans la table **Replica set address**. Cela peut améliorer la fiabilité et les performances de la manipulation de données.

- Si vous souhaitez restreindre votre connexion MongoDB à une seule base de données en particulier, saisissez le nom de cette base de données dans le champ **Database**.

Si vous laissez ce champ vide, l'assistant listera les collections de toutes les bases de données existantes sur le serveur connecté lors de la récupération des schémas.

- Si votre serveur MongoDB requiert des informations d'authentification pour accéder à la base de données, cochez la case **Require authentication** et renseignez vos nom d'utilisateur et mot de passe dans les champs correspondants.



4. Cliquez sur le bouton **Check** afin de vous assurer que la connexion fonctionne.
5. Cliquez sur **Finish** pour valider les paramètres.

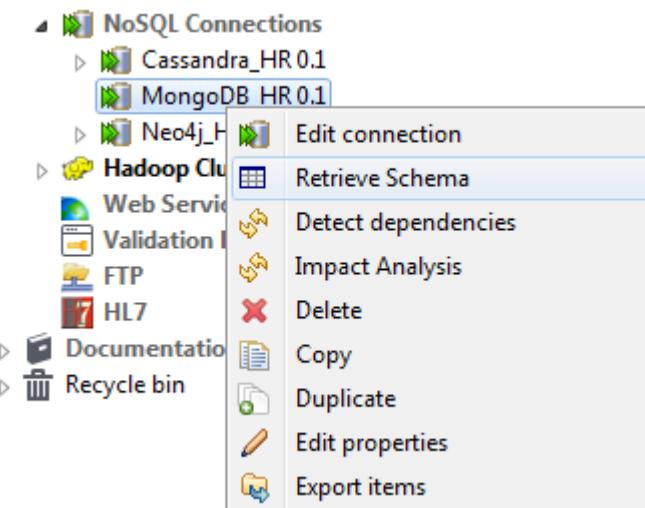
La connexion à la base de données MongoDB nouvellement créée apparaît sous le nœud **NoSQL Connection** dans la vue **Repository**. Vous pouvez maintenant la déposer dans votre espace de modélisation graphique en tant que composant MongoDB, mais vous devez tout de même définir les informations de schéma là où cela est nécessaire.

Vous devez ensuite récupérer un ou plusieurs schéma(s) qui vous intéresse(nt) pour votre connexion.

### 11.1.2.2. Récupérer les schémas

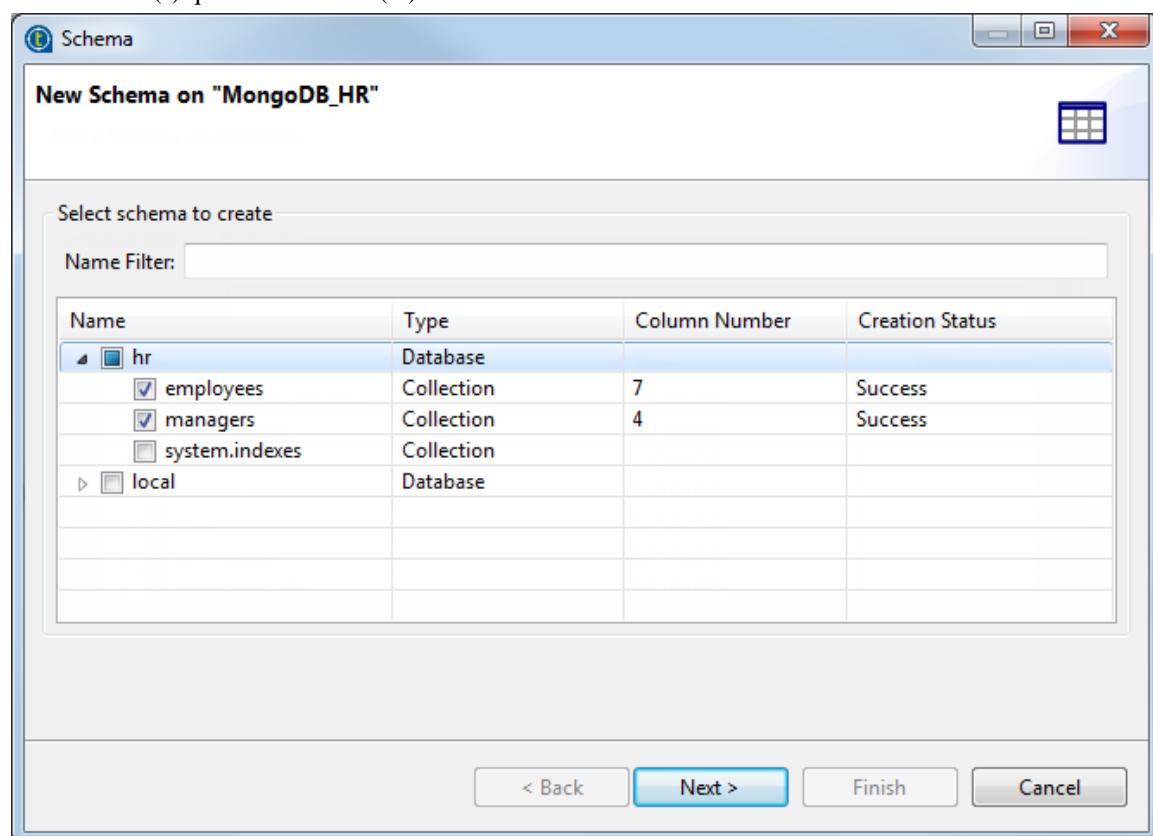
Dans cette étape, vous allez récupérer les schémas qui vous intéressent de la base de données connectée MongoDB.

1. Dans la vue **Repository**, cliquez-droit sur la connexion nouvellement créée et sélectionnez **Retrieve Schema** dans le menu contextuel.



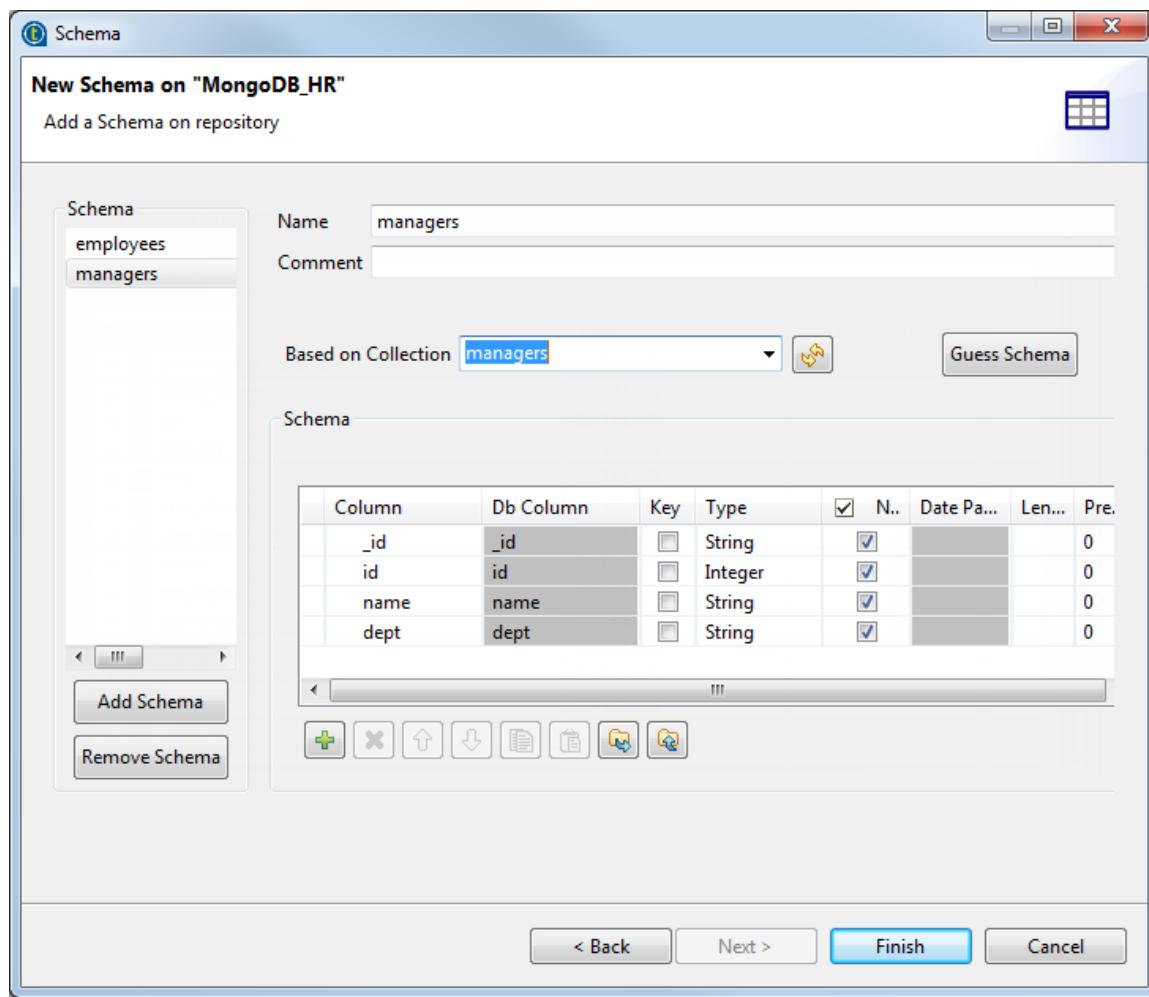
L'assistant ouvre une nouvelle vue qui liste toutes les collections disponibles de la base de données spécifiée, ou toutes les bases de données disponibles si vous n'en avez spécifié aucune lors de l'étape précédente.

2. Développez la base de données, ou alors les bases de données qui vous intéressent si vous n'avez pas spécifié de base de données lors de l'étape précédente, comme c'est le cas dans cet exemple, puis sélectionnez la ou les collection(s) qui vous intéresse(nt).



3. Cliquez sur **Next** pour passer à l'étape suivante de l'assistant dans laquelle vous pouvez éditer le ou les schéma(s) générés.

Par défaut, chaque schéma généré prend le nom de la colonne sur laquelle il est basé.



Selectionnez un schéma dans le panneau **Schema** pour afficher ses détails sur le côté droit et modifiez le schéma si nécessaire. Vous pouvez renommer n'importe quel schéma et personnaliser la structure d'un schéma selon vos besoins dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, de supprimer ou de déplacer des colonnes dans votre schéma, ou encore de remplacer le schéma avec le schéma défini dans un fichier XML.

Pour baser un schéma sur une autre collection, sélectionnez le nom du schéma dans le panneau **Schema** et sélectionnez une nouvelle collection dans la liste **Based on Collection**. Cliquez ensuite sur le bouton **Guess Schema** pour écraser le schéma avec celui de la collection sélectionnée. Vous pouvez cliquer sur le bouton de rafraîchissement pour rafraîchir la liste des collections.

Pour ajouter un nouveau schéma, cliquez sur le bouton **Add Schema** dans le panneau **Schema** pour créer un schéma vide que vous devez définir.

Pour supprimer un schéma, sélectionnez le nom du schéma dans le panneau **Schema** et cliquez sur le bouton **Remove Schema**.

Pour écraser les modifications que vous avez effectuées sur le schéma sélectionné par son schéma par défaut, cliquez sur le bouton **Guess schema**. Notez que toutes les modifications que vous avez apportées au schéma seront perdues si vous cliquez sur ce bouton.

4. Cliquez sur **Finish** pour terminer la création du schéma. Les schémas créés apparaissent sous votre connexion MongoDB dans la vue **Repository**. Vous pouvez désormais déposer la connexion ou n'importe quel noeud sous cette connexion dans votre espace graphique de modélisation en tant que composant MongoDB. Toutes les informations sur les métadonnées seront automatiquement renseignées.

Si vous avez besoin de continuer à éditer un schéma, cliquez-droit sur ce schéma et sélectionnez **Edit Schema** dans le menu contextuel pour ouvrir de nouveau cet assistant et effectuer vos modifications.



*Si vous modifiez les schémas, assurez-vous que le type de données dans la colonne **Type** est correctement défini.*

## 11.1.3. Centraliser les métadonnées de Neo4j

Si vous avez souvent besoin de manipuler des données d'une base de données Neo4j, alors il peut être intéressant pour vous de centraliser la connexion à une base de données Neo4j et les détails de schémas dans le dossier **Metadata**, dans la vue **Repository**.

La procédure d'installation des métadonnées de Neo4j est constituée de deux tâches majeures différentes mais liées :

1. Créer une connexion à une base de données Neo4j.
2. Récupérer les schémas Neo4j qui vous intéressent.

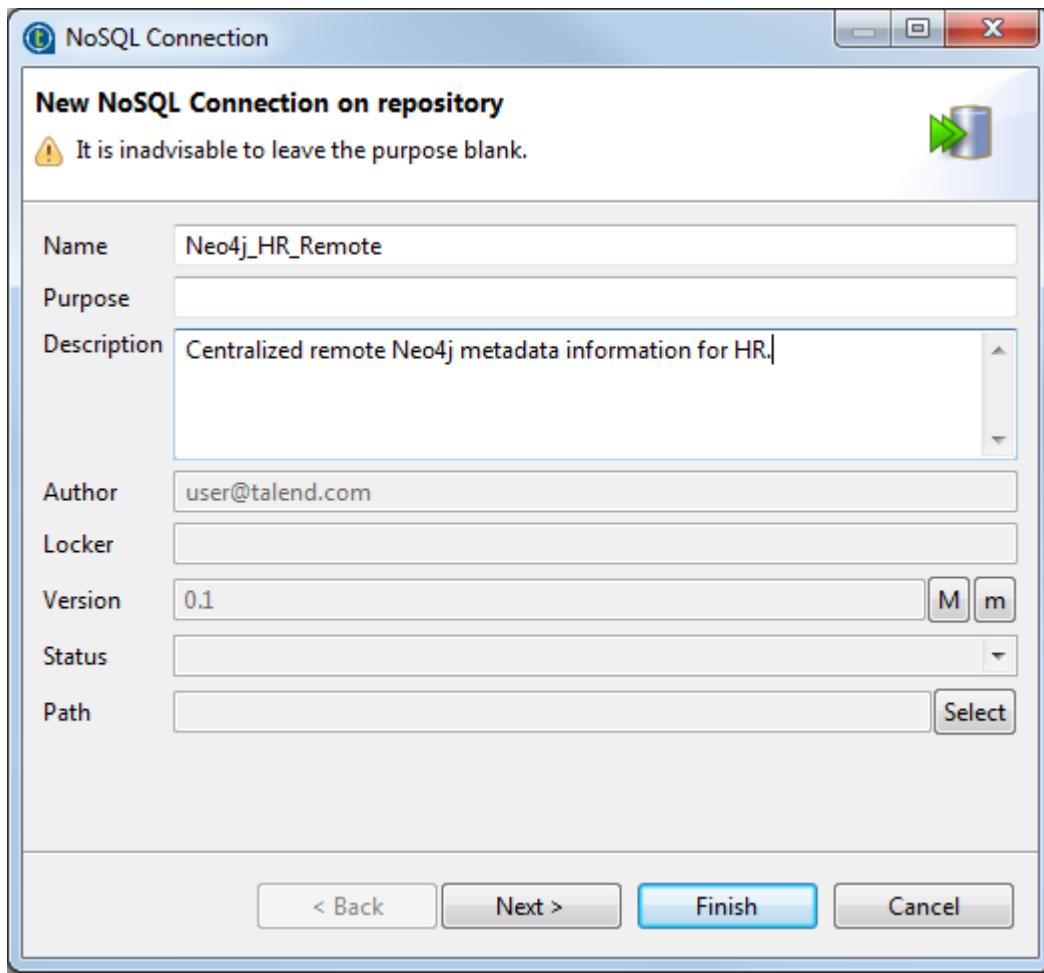
### Prérequis :

- Tous les modules externes requis manquant dans le *Studio Talend* en raison des restrictions de licence doivent être installés. Pour plus d'informations, consultez le *Guide d'installation et de migration Talend*.
- Vous devez connaître les requêtes Cypher utilisées pour lire les données dans Neo4j.
- Le serveur Neo4j doit être fonctionnel si vous avez besoin de vous connecter à la base de données Neo4j en mode distant.

### 11.1.3.1. Créeer une connexion à une base de données Neo4j

1. Dans la vue **Repository**, développez le noeud **Metadata**, cliquez-droit sur **NoSQL Connection** et sélectionnez **Create Connection** dans le menu contextuel. L'assistant de connexion s'ouvre.
2. Dans l'assistant de connexion, renseignez les propriétés générales de la connexion que vous avez besoin de créer, telles que le nom (**Name**), le but (**Purpose**) et la description (**Description**).

Les informations que vous renseignez dans le champ **Description** apparaissent dans une info-bulle lorsque vous bougez le pointeur de votre souris au-dessus de la connexion.

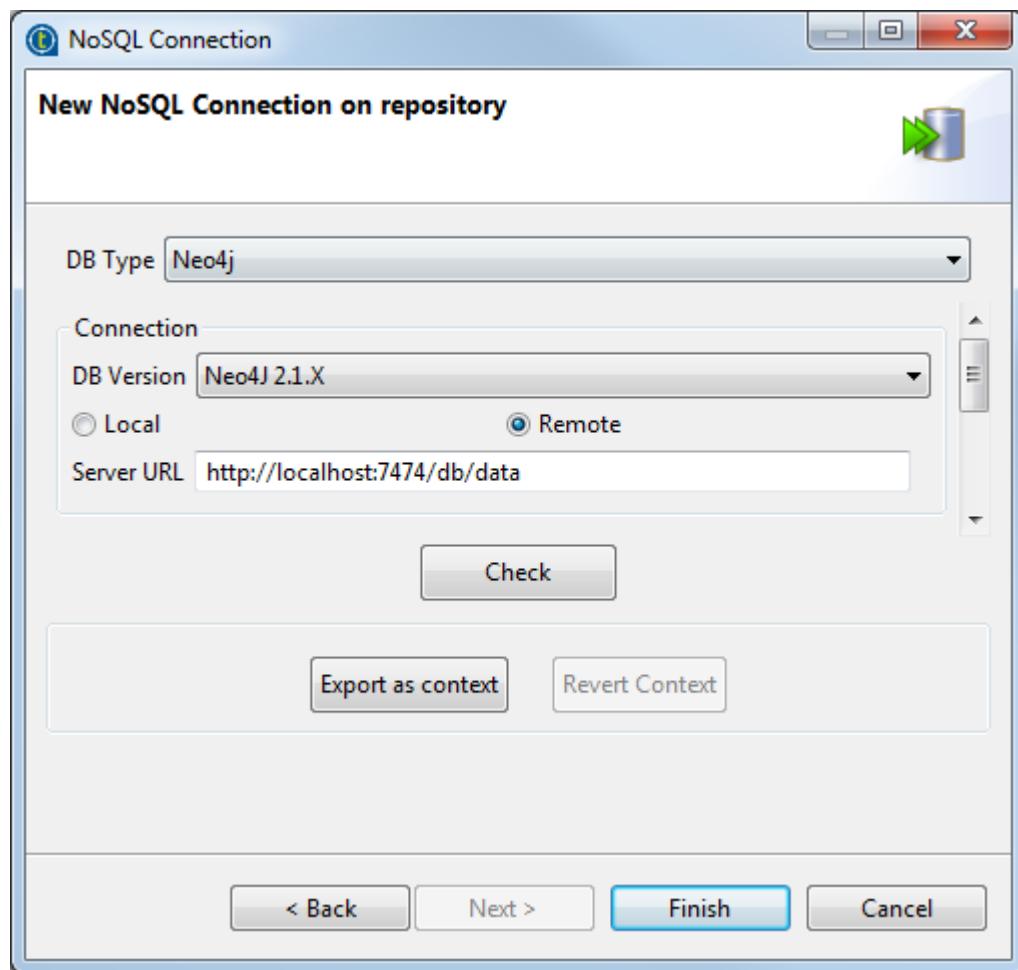


Lorsque vous avez terminé, cliquez sur **Next** pour passer à l'étape suivante.

3. Sélectionnez **Neo4j** dans la liste **DB Type** et spécifiez les détails de connexion :

- Pour vous connecter à une base de données Neo4j en mode local (également connu sous le nom de mode embarqué), sélectionnez l'option **Local** et spécifiez le répertoire contenant vos fichiers de données Neo4j.
- Pour vous connecter à une base de données Neo4j en mode distant (également connu sous le nom de mode REST), sélectionnez l'option **Remote** et saisissez l'URL du serveur Neo4j.

Dans cet exemple, la base de données Neo4j est accessible en mode distant et l'URL du serveur Neo4j est l'URL que l'assistant propose par défaut.



4. Cliquez sur le bouton **Check** afin de vous assurer que la connexion fonctionne.
5. Cliquez sur **Finish** pour valider les paramètres.

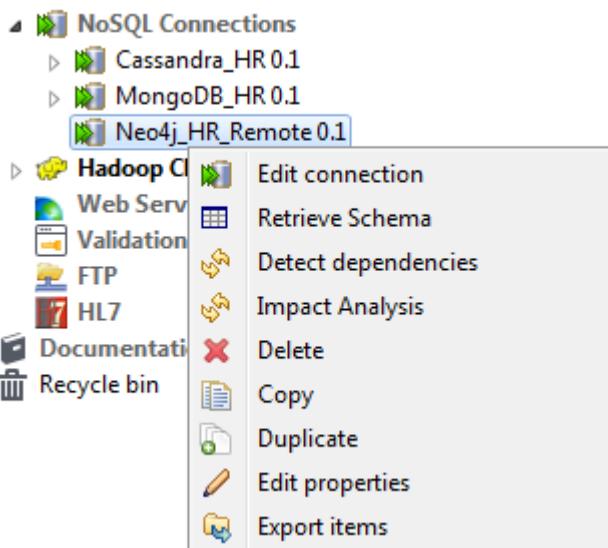
La connexion à la base de données Neo4j nouvellement créée apparaît alors sous le nœud **NoSQL Connection** dans la vue **Repository**. Vous pouvez maintenant la déposer dans votre espace de modélisation graphique en tant que composant Neo4j, mais vous devez tout de même définir les informations de schéma là où cela est nécessaire.

Vous devez ensuite récupérer un ou plusieurs schéma(s) qui vous intéresse(nt) pour votre connexion.

### 11.1.3.2. Récupérer un schéma

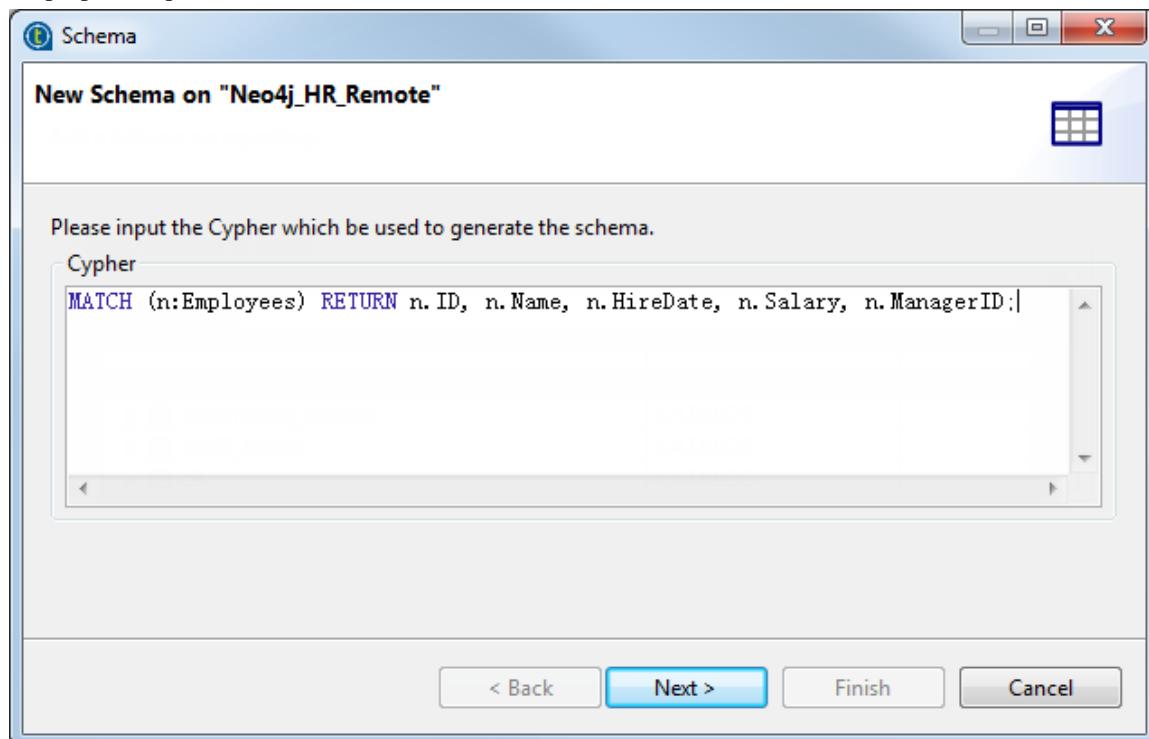
Dans cette étape, vous allez récupérer le schéma qui vous intéresse de la base de données connectée Neo4j.

1. Dans la vue **Repository**, cliquez-droit sur la connexion nouvellement créée et sélectionnez **Retrieve Schema** dans le menu contextuel.



L'assistant ouvre une nouvelle vue pour la génération de schéma effectuée d'après une requête Cypher.

- Dans le champ Cypher, saisissez votre requête Cypher de sorte qu'elle corresponde aux nœuds et récupérez les propriétés qui vous intéressent.



*Si votre requête Cypher comprend des chaînes de caractères, entourez-les par des guillemets simples au lieu de guillemets doubles, sous peine de causer des erreurs dans les composants Neo4j déposés à partir de vos métadonnées centralisées.*

Dans cet exemple, la requête suivante est utilisée de sorte qu'elle corresponde aux nœuds ayant pour nom *Employees* et récupère leurs propriétés *ID*, *Name*, *HireDate*, *Salary* et *ManagerID* comme colonnes de schéma :

```
MATCH (n:Employees) RETURN n.ID, n.Name, n.HireDate, n.Salary, n.ManagerID;
```

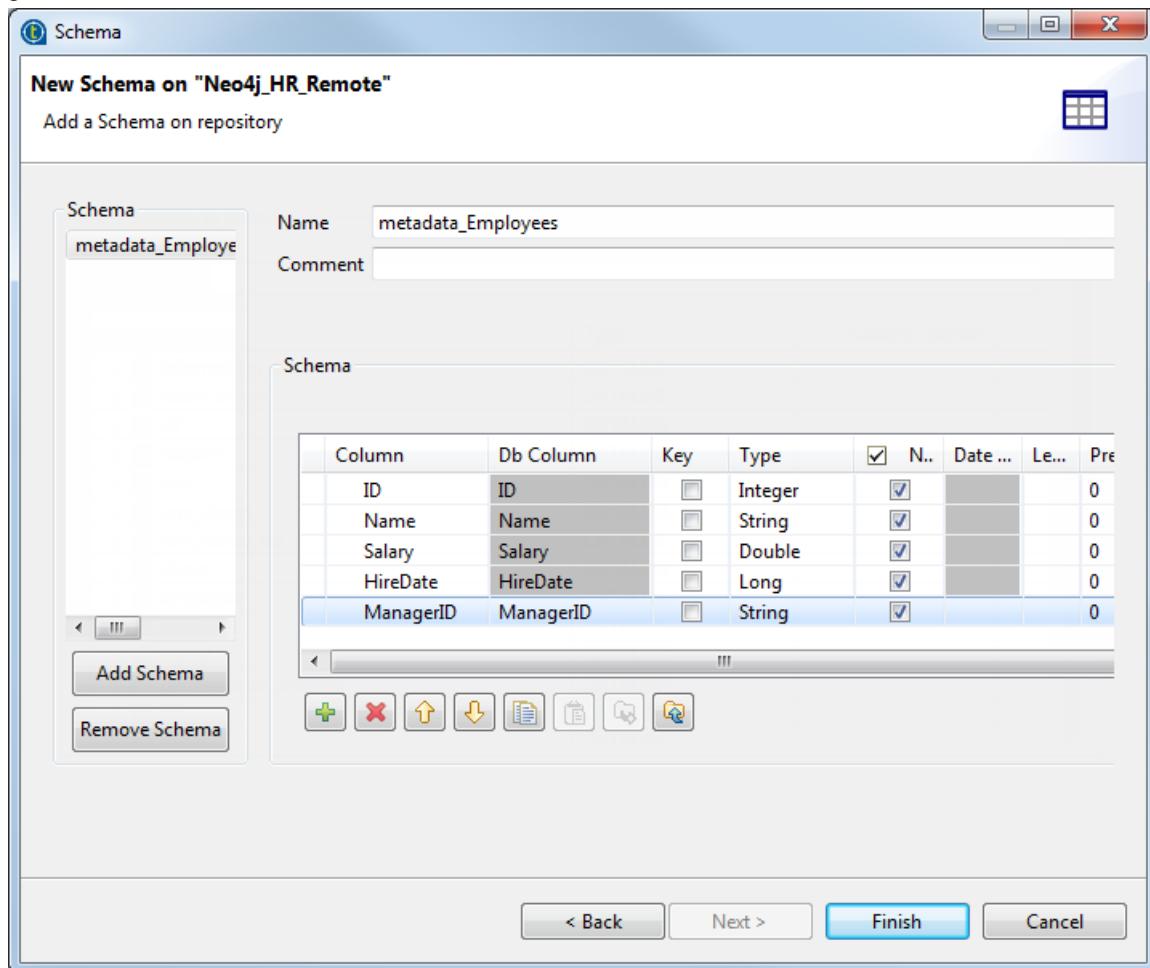
Si vous souhaitez récupérer toutes les propriétés des nœuds s'appelant *Employees* dans cet exemple, vous pouvez saisir la requête suivante :

```
MATCH (n:Employees) RETURN n;
```

ou :

```
MATCH (n:Employees) RETURN *;
```

- Cliquez sur **Next** pour passer à l'étape suivante de l'assistant dans laquelle vous pouvez éditer le schéma généré.



Modifiez le schéma si nécessaire. Vous pouvez renommer le schéma et personnaliser la structure du schéma selon vos besoins dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, de supprimer ou de déplacer des colonnes dans votre schéma, ou encore de remplacer le schéma avec le schéma défini dans un fichier XML.

Pour ajouter un nouveau schéma, cliquez sur le bouton **Add Schema** dans le panneau **Schema** pour créer un schéma vide que vous devez définir.

Pour supprimer un schéma, sélectionnez le nom du schéma dans le panneau **Schema** et cliquez sur le bouton **Remove Schema**.

- Cliquez sur **Finish** pour terminer la création du schéma. Le schéma créé apparaît sous votre connexion Neo4j dans la vue **Repository**. Vous pouvez désormais déposer la connexion ou n'importe quel nœud de schéma sous cette connexion dans votre espace de modélisation graphique en tant que composant Neo4j. Toutes les informations sur les métadonnées seront automatiquement renseignées.

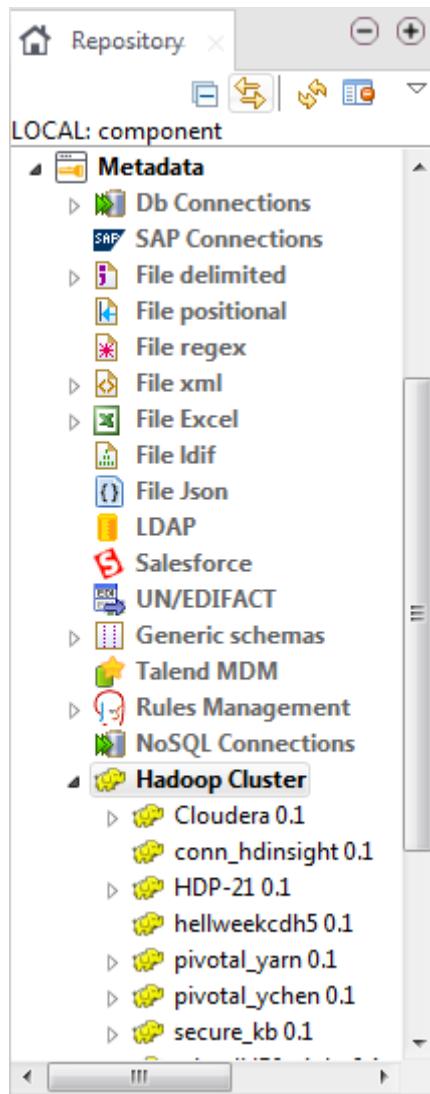
Si vous avez besoin de continuer à éditer un schéma, cliquez-droit sur ce schéma et sélectionnez **Edit Schema** dans le menu contextuel pour ouvrir de nouveau cet assistant et effectuer vos modifications.

 Si vous modifiez les schémas, assurez-vous que le type de données dans la colonne **Type** est correctement défini.

## 11.2. Gérer les métadonnées Hadoop

Dans la vue **Repository**, le noeud **Hadoop cluster** du dossier **Metadata** regroupe les métadonnées des connexions aux éléments Hadoop comme HDFS, Hive ou HBase. Cela vous permet de centraliser les propriétés de connexion définies pour une distribution Hadoop donnée et de réutiliser ces propriétés afin de créer des connexions séparées pour chaque élément de Hadoop.

Cliquez sur le noeud **Metadata** dans la vue **Repository** pour développer le dossier. Chaque noeud regroupe les connexions et schémas que vous avez paramétrés. Parmi ces noeuds de connexion, se trouve le noeud **Hadoop cluster**.



Les sections suivantes expliquent en détail comment utiliser le noeud **Hadoop cluster** afin de configurer :

- une connexion HBase,
- une connexion HCatalog,
- un schéma de fichier HDFS,

- une connexion Hive,
- une connexion Oozie.

Si vous souhaitez créer une connexion à la base de données analytique de Cloudera, Impala, utilisez le nœud **DB Connection** sous le nœud **Metadata** du **Repository**. Sa configuration est similaire à celle d'un composant Hive mais moins compliquée que celle-là.

Pour plus d'informations concernant le nœud **DB connection**, consultez *Gestion des métadonnées dans l'intégration de données*.

## 11.2.1. Centraliser une connexion Hadoop

Configurer une connexion à une distribution Hadoop donnée dans le **Repository** vous permet d'éviter d'avoir à configurer cette connexion à chaque fois que vous souhaitez utiliser la même distribution Hadoop.

Vous devez tout d'abord définir une connexion Hadoop avant de pouvoir créer, à partir du nœud **Hadoop cluster**, les connexions de chaque élément de Hadoop comme HDFS, Hive ou Oozie.

### Prérequis :

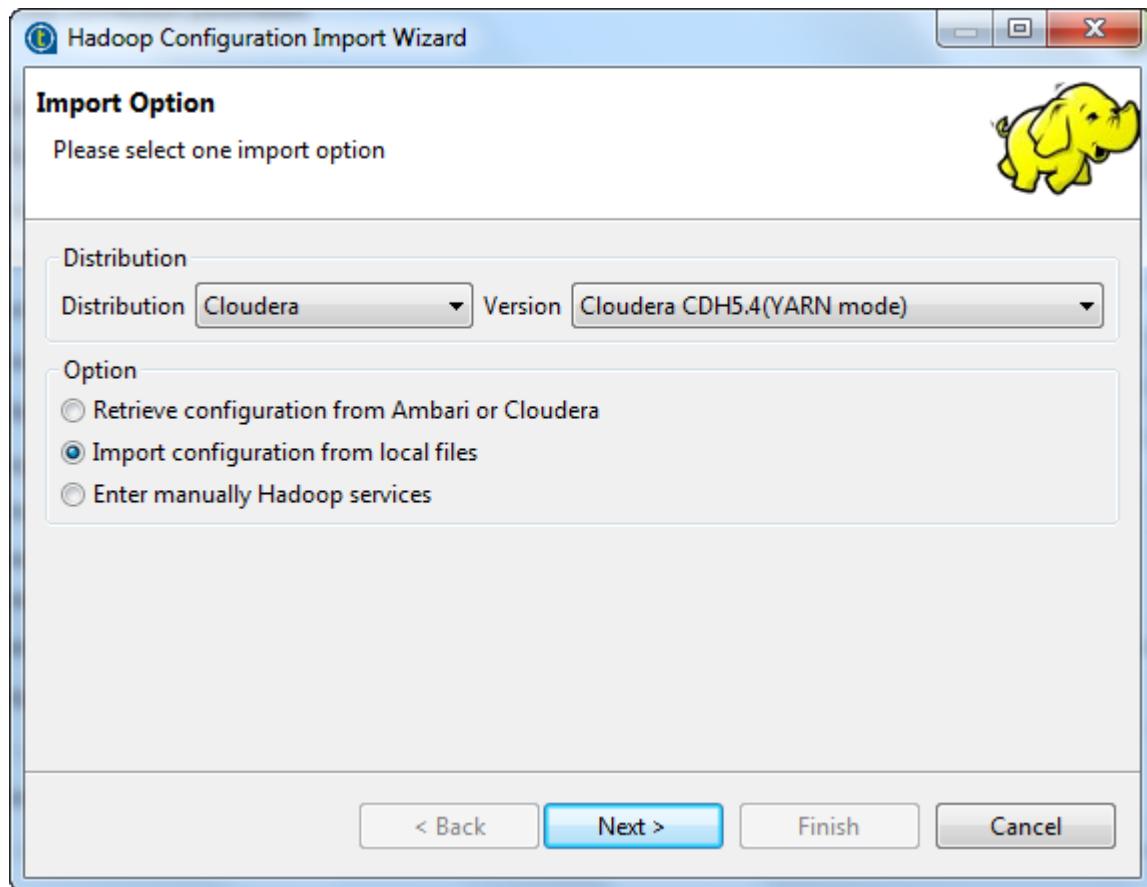
- Vous devez avoir vérifié que la machine cliente sur laquelle est installé le *Studio Talend* peut reconnaître les noms d'hôtes des nœuds du cluster Hadoop à utiliser. Dans cet objectif, ajoutez les mappings des entrées adresse IP/nom d'hôte pour les services de ce cluster Hadoop dans le fichier *hosts* de la machine cliente.

Par exemple, si le nom d'hôte du serveur du NameNode Hadoop est *talend-cdh550.weave.local* et son adresse IP est *192.168.x.x*, l'entrée du mapping est la suivante *192.168.x.x talend-cdh550.weave.local*.

- Le cluster Hadoop à utiliser doit avoir été correctement configuré et être en cours de fonctionnement.
- Si vous devez vous connecter à MapR à partir du studio, assurez-vous d'avoir installé le client MapR sur la même machine que le studio et d'avoir ajouté la bibliothèque client de MapR dans la variable PATH de cette machine. D'après la documentation de MapR, la ou les librairies du client MapR correspondant à chaque OS peuvent être trouvées *MAPR\_INSTALL\hadoop\hadoop-VERSION\lib\native*. Par exemple, pour Windows, la bibliothèque est *lib\MapRClient.dll* dans le fichier Jar du client MapR. Pour plus d'informations, consultez la page suivante sur le site de MapR : <http://www.mapr.com/blog/basic-notes-on-configuring-eclipse-as-a-hadoop-development-environment-for-mapr> (en anglais).

Afin de créer une connexion Hadoop dans le **Repository**, procédez comme suit :

1. Dans la vue **Repository** de votre studio, développez le nœud **Metadata** puis cliquez-droit sur **Hadoop cluster**.
2. Sélectionnez **Create Hadoop cluster** dans le menu contextuel afin d'ouvrir l'assistant [**Hadoop cluster connection**].
3. Renseignez les informations génériques du schéma, comme le nom, dans le champ **Name** et la **Description** puis cliquez sur **Next** pour ouvrir la fenêtre [**Hadoop Configuration Import Wizard**] vous permettant de sélectionner le mode manuel ou automatique pour configurer la connexion.



### 11.2.1.1. Configurer automatiquement la connexion à Hadoop

Ce mode automatique peut être uniquement appliqué aux distributions Hadoop officiellement supportées par le Studio, c'est-à-dire les distributions que vous pouvez trouver dans l'assistant **[Hadoop Configuration Import Wizard]**.

1. Dans la zone **Distribution**, sélectionnez la distribution Hadoop à utiliser et sa version.
2. Sélectionnez la manière dont vous souhaitez paramétrier la configuration dans cet assistant d'import.
  - **Retrieve configuration from Ambari or Cloudera** : si vous utilisez un cluster Hortonworks Data Platform ou Cloudera CDH et que votre cluster contient sa propre plateforme de gestion : Hortonworks Ambari pour Hortonworks Data Platform et Cloudera manager pour Cloudera CDH, cochez cette case pour importer directement la configuration.

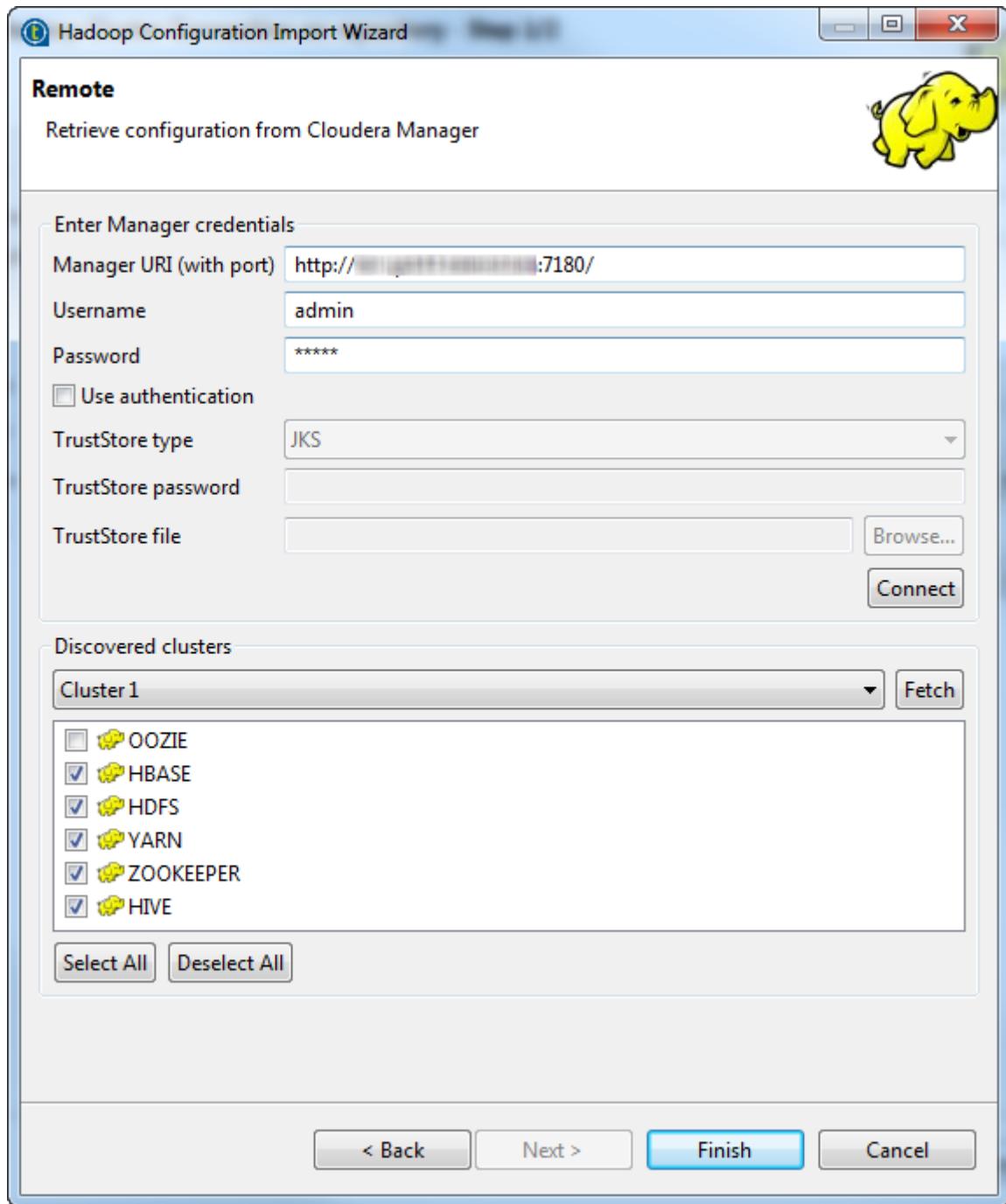
Pour plus d'informations, consultez [Récupérer la configuration depuis Ambari ou Cloudera](#).

- **Import configuration from local files** : lorsque vous avez obtenu ou lorsque vous pouvez obtenir les fichiers de configuration (principalement les fichiers `*-site.xml`), par exemple, de l'administrateur de votre cluster Hadoop ou les télécharger directement du service de gestion de cluster basé Web, utilisez cette option pour importer les propriétés directement depuis ces fichiers.

Pour plus d'informations, consultez [Importer la configuration depuis des fichiers locaux](#).

## Récupérer la configuration depuis Ambari ou Cloudera

Si vous pouvez accéder au service basé Web de gestion de votre cluster, c'est-à-dire Ambari pour Hortonworks ou Cloudera manager pour Cloudera, sélectionnez cette option **Retrieve configuration from Ambari or Cloudera** pour importer les informations de configuration directement depuis ce service de gestion.



Cette image montre un exemple de cet assistant pour récupérer la configuration.

Dans cet assistant, procédez comme suit :

1. Dans la zone relatives aux informations d'authentification, saisissez votre identifiant et mot de passe pour accéder au service basé Web de gestion de votre cluster. Dans cet exemple, connectez-vous à Cloudera manager.

2. Si le système de certificat a été configuré pour le service de gestion auquel vous devez vous connecter, cochez la case **Use authentication** pour activer les champs relatifs et renseignez-les à l'aide de votre fichier TrustStore.

Si vous n'avez pas accès à ce fichier TrustStore, contactez l'administrateur du cluster.

Hortonworks et Cloudera fournissent les informations relatives à la sécurité autour du service de gestion dans leur documentation. Vous pouvez trouver plus d'informations relatives à la documentation sur leur site Web :

- Cloudera :

[http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm\\_sg\\_config\\_tls\\_security.html](http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_sg_config_tls_security.html).

- Hortonworks :

[http://docs.hortonworks.com/HDPDocuments/Ambari-1.6.1.0/bk\\_ambari\\_security/content/ambari-security-overview.html](http://docs.hortonworks.com/HDPDocuments/Ambari-1.6.1.0/bk_ambari_security/content/ambari-security-overview.html).

3. Si votre machine locale a des permissions d'accès à votre Ambari ou Cloudera Manager, cliquez sur le bouton **Connect** pour créer la connexion à Ambari ou Cloudera Manager à partir du Studio.

Le nom du cluster géré par ce service de gestion de cluster s'affiche dans la liste **Discovered clusters**.

4. Cliquez sur le bouton **Fetch** afin de récupérer et lister les configurations des services de ce cluster dans cet assistant.
5. Sélectionnez les services pour lesquels vous souhaitez importer les informations de configuration.
6. Cliquez sur **Finish**.

Les informations de configuration sont automatiquement renseignées dans l'étape suivante de l'assistant **[Hadoop cluster connection]**.

7. Dans cet assistant **[Hadoop cluster connection]**, vérifiez que la case **Use custom Hadoop configurations** est cochée, afin d'assurer que la configuration complète importée est prise en compte. Si vous décochez cette case, le Studio utilise sa configuration Hadoop par défaut (sous forme d'un fichier .jar) au lieu de prendre en compte les paramètres supplémentaires explicitement configurés par vos soins dans l'assistant.

Pour cette raison, il est important de cocher cette case et de vous assurer que votre configuration personnalisée écrase celle par défaut.

8. Cliquez sur le bouton **Check services** afin de vérifier que le Studio peut se connecter aux services NameNode et ResourceManager spécifié dans cet assistant.

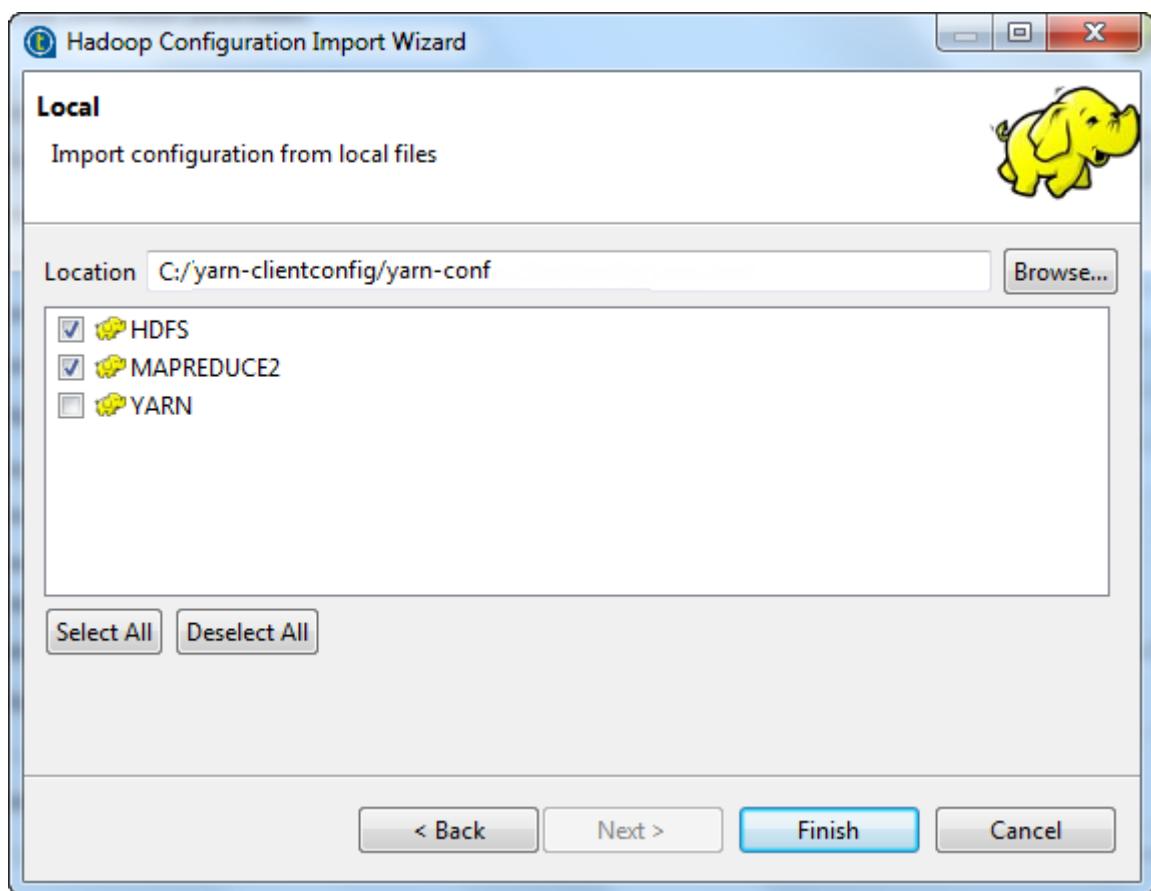
Une boîte de dialogue s'ouvre pour indiquer le processus de vérification et le statut de connexion. Si elle affiche un échec de la connexion, vous devez vérifier et mettre à jour les informations de connexion définies dans l'assistant de connexion.

9. Cliquez sur **Finish** pour valider les modifications.

Si vous avez besoin de plus d'informations concernant les champs auto-complétés dans cet assistant **[Hadoop cluster connection]**, consultez [Configurer manuellement la connexion](#)

## Importer la configuration depuis des fichiers locaux

Une fois que vous avez sélectionné **Import configuration from local files** dans l'assistant d'import, l'assistant suivant s'ouvre et vous permet de sélectionner les fichiers de configuration Hadoop (principalement des fichiers `*-site.xml`) à utiliser depuis la machine locale.

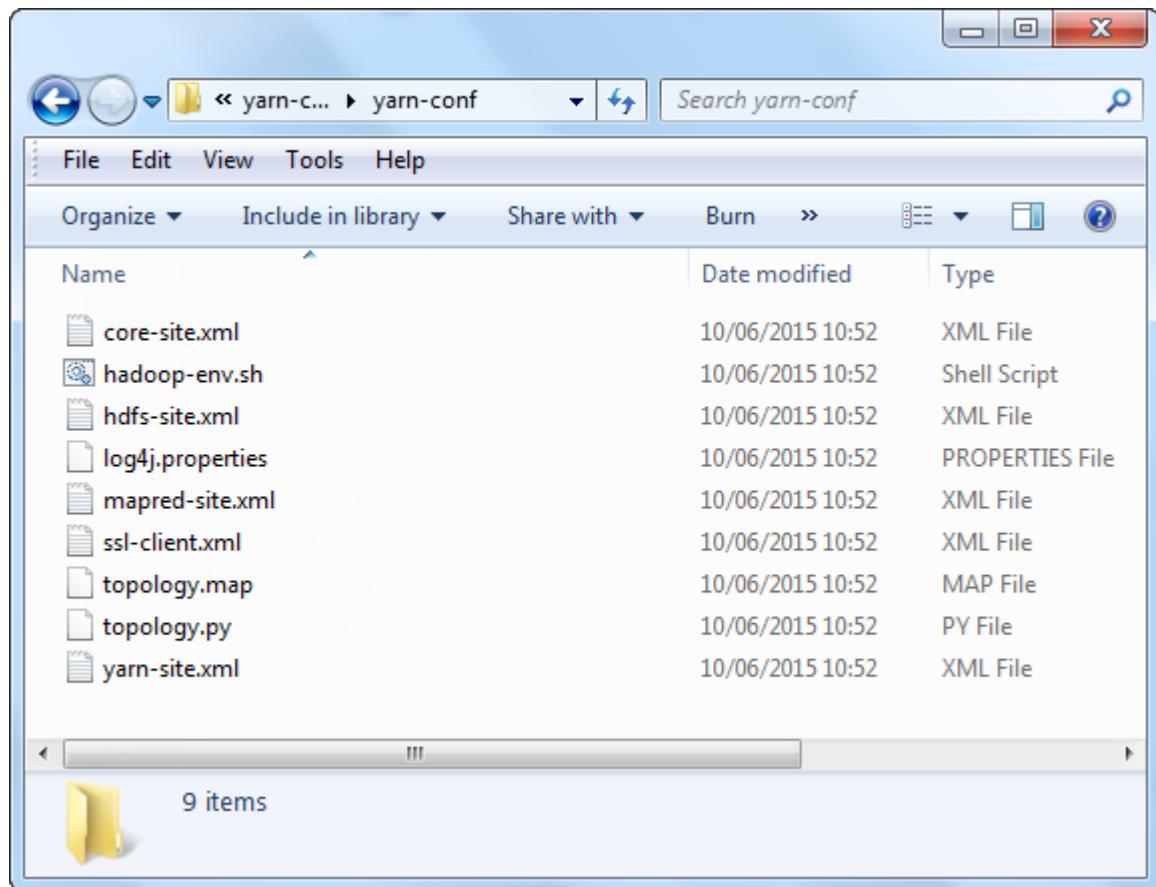


Dans cet assistant, procédez comme suit :

1. Cliquez sur **Browse...** pour accéder au dossier dans lequel les fichiers de configuration locaux à utiliser sont stockés. Cliquez sur **OK** pour lister les configurations dans cet assistant.

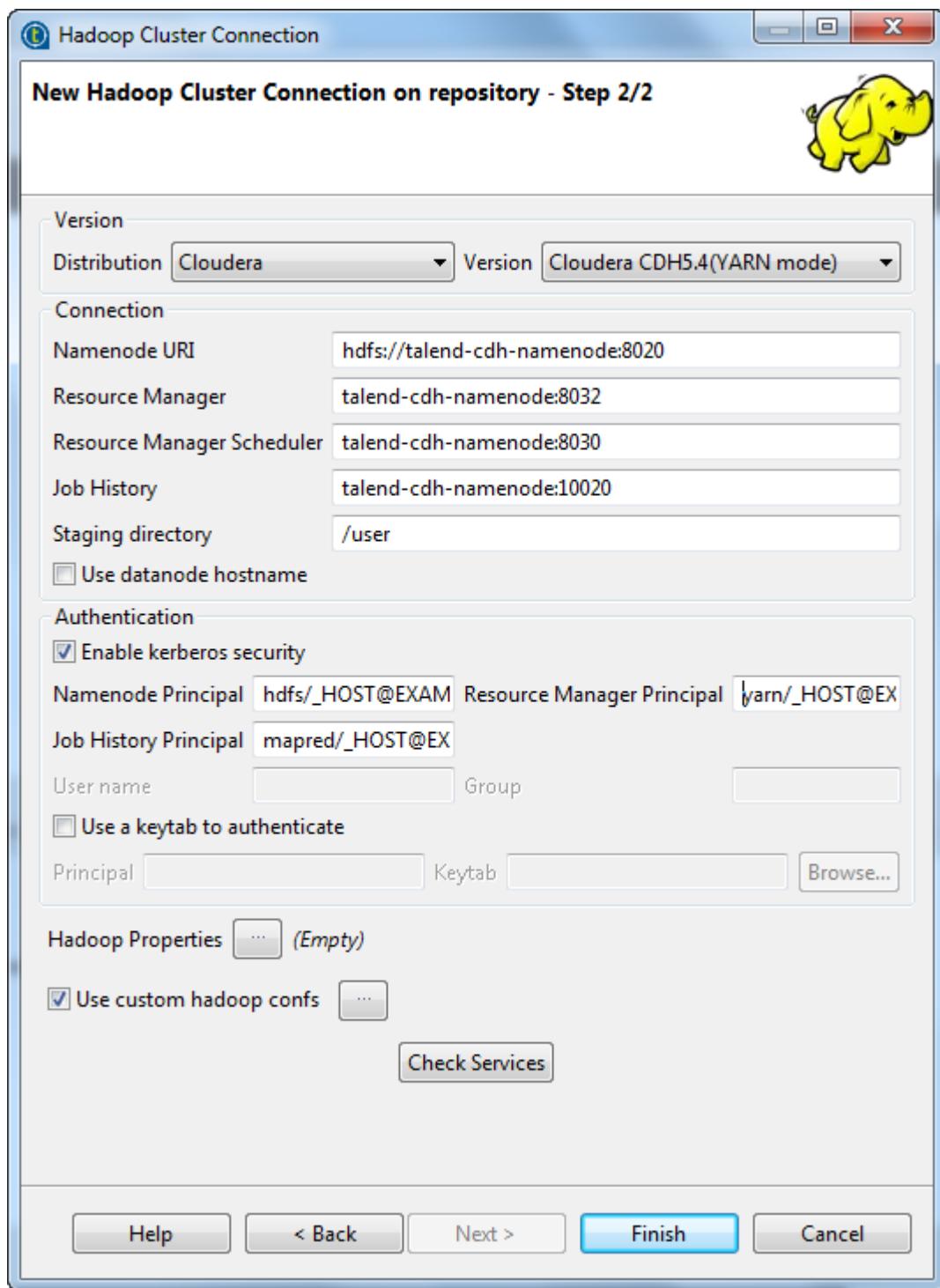
Il est recommandé de stocker ces fichiers de configuration avec un chemin d'accès rapide sur la machine locale.

L'image suivante montre certains fichiers utilisés pour la configuration de HDFS, MapReduce et Yarn dans Cloudera. Ces fichiers d'exemple sont téléchargés et automatiquement générés par Cloudera manager.



2. Dans la liste de configuration, sélectionnez les configurations à importer, par exemple celles pour *HDFS* et *MAPREDUCE2*, puis cliquez sur **Finish**.

Les informations de configuration sont automatiquement renseignées dans l'étape suivante de l'assistant **[Hadoop cluster connection]**.



3. Dans cet assistant [**Hadoop cluster connection**], vérifiez que la case **Use custom Hadoop configurations** est cochée, afin d'assurer que la configuration complète importée est prise en compte. Si vous décochez cette case, le Studio utilise sa configuration Hadoop par défaut (sous forme d'un fichier .jar) au lieu de prendre en compte les paramètres supplémentaires explicitement configurés par vos soins dans l'assistant.

Pour cette raison, il est important de cocher cette case et de vous assurer que votre configuration personnalisée écrase celle par défaut.

4. Cliquez sur le bouton **Check services** pour vérifier que le Studio peut se connecter aux services NameNode et ResourceManager spécifiés dans cet assistant.

Une boîte de dialogue s'ouvre pour indiquer le processus de vérification et le statut de connexion. Si elle indique un échec de connexion, vous devez vérifier et mettre à jour les informations de connexion définie dans l'assistant.

5. Cliquez sur **Finish** afin de valider les modifications.

Si vous souhaitez plus de détails concernant les champs auto-complétés dans cet assistant [**Hadoop cluster connection**], consultez [Configurer manuellement la connexion](#)

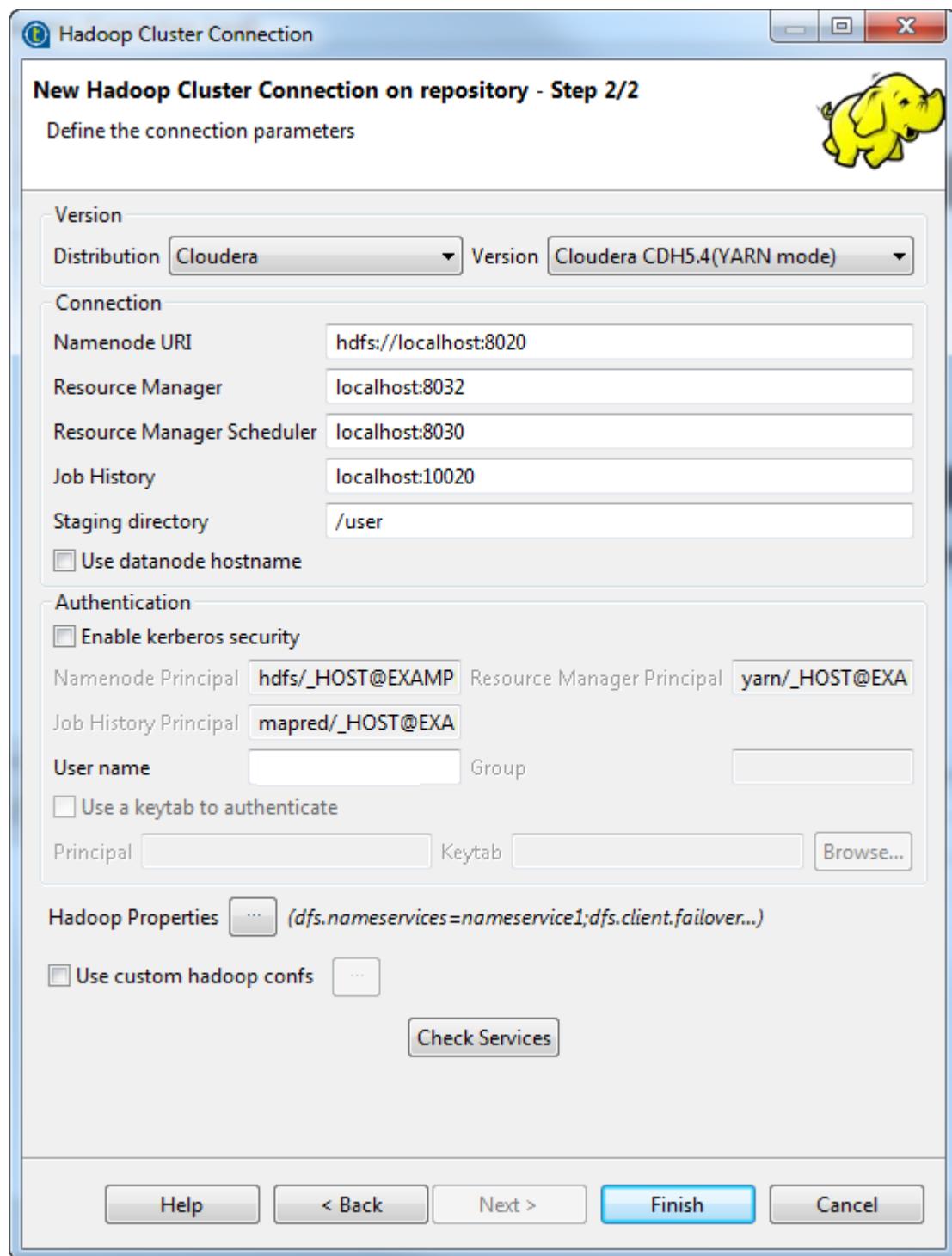
### 11.2.1.2. Configurer manuellement la connexion

Même si l'import d'une configuration Hadoop donnée est efficace, il est possible que vous deviez configurer manuellement la connexion, dans certaines circonstances, par exemple si vous n'avez pas les configurations à importer sous la main.

Ce mode vous permet de vous connecter à une distribution Hadoop personnalisée. Pour plus d'informations, consultez [Connexion à une distribution Hadoop personnalisée](#).

1. Renseignez les champs disponibles selon la version sélectionnée.

Notez que, parmi ces champs, les champs **NameNode URI** et **Resource Manager**) ont été automatiquement renseignés avec la syntaxe par défaut et le numéro de port correspondants à la distribution sélectionnée. Vous devez mettre à jour uniquement la partie dont vous avez besoin, selon la configuration du cluster Hadoop à utiliser. Pour plus d'informations concernant les différents champs à utiliser, consultez la liste suivante.



Les champs peuvent être :

- **Namenode URI :**

Saisissez l'URI pointant vers la machine utilisée comme NameNode de la distribution Hadoop à utiliser.

Le NameNode est le nœud maître d'un système Hadoop. Par exemple, si vous avez choisi une machine nommée *machine1* comme NameNode d'une distribution Apache Hadoop, l'emplacement à saisir est *hdfs://machine1:portnumber*.

Si vous utilisez une distribution MapR, vous pouvez simplement laisser *maprfs:///* dans le champ. Le client MapR va gérer les informations à la volée lors de la création de la connexion. Le client MapR doit être

correctement installé. Pour plus d'informations concernant la configuration d'un client MapR, consultez la page suivante de la documentation MapR : <http://doc.mapr.com/display/MapR/Setting+Up+the+Client> (en anglais).

- **Resource Manager :**

Saisissez l'URI pointant vers la machine utilisée comme service du gestionnaire de ressources (Resource Manager) de la distribution Hadoop à utiliser.

Notez que, dans certaines anciennes versions des distributions Hadoop, vous devez configurer l'emplacement du service du JobTracker au lieu du service du gestionnaire de ressources.

Vous devez configurer les adresses des services relatifs, comme l'adresse du **Resourcemanager scheduler**. Lorsque vous utilisez cette connexion dans un composant Big Data, comme le **tHiveConnection**, vous pouvez allouer de la mémoire aux calculs **Map** et **Reduce** et à l'**ApplicationMaster** de YARN dans la vue **Advanced settings**. Pour plus d'informations concernant le Resource Manager, son ordonnanceur et son ApplicationMaster, consultez la documentation de YARN pour votre distribution, par exemple à l'adresse suivante :

<http://hortonworks.com/blog/apache-hadoop-yarn-concepts-and-applications/>.

- **Job history :**

Saisissez l'emplacement du serveur de JobHistory du cluster Hadoop à utiliser. Cela permet de stocker les méreiques du Job courant sur le serveur de JobHistory.

- **Staging directory :**

Saisissez le répertoire défini dans votre cluster Hadoop pour les fichiers temporaires créées par les programmes en cours d'exécution. Généralement, ce répertoire se trouve sous la propriété *yarn.app.mapreduce.am.staging-dir* dans les fichiers de configuration comme *yarn-site.xml* ou *mapred-site.xml* de votre distribution.

- **Use datanode hostname :**

Cochez cette case pour permettre au Job d'accéder aux noeuds de données (datanodes) via leurs noms d'hôtes. Cela permet de configurer la propriété *dfs.client.use.datanode.hostname* à *true*. Si cette connexion doit être utilisée par un Job se connectant à un système de fichiers S3N, cochez cette case.

- **Enable Kerberos security :**

Si vous accédez à une distribution Hadoop s'exécutant avec la sécurité Kerberos, cochez cette case, puis saisissez les noms des Principaux de Kerberos pour le NameNode dans le champ activé.

Ces Principaux se trouvent dans les fichiers de configuration de votre distribution. Par exemple, dans une distribution CDH4, le Principal du Resource Manager est configuré dans le fichier *yarn-site.xml* et le Principal de Job History dans le fichier *mapred-site.xml*.

Si vous devez utiliser un fichier Keytab pour vous connecter, cochez la case **Use a keytab to authenticate**. Un fichier Keytab contient les paires des Principaux et clés cryptées Kerberos. Vous devez saisir le Principal à utiliser dans le champ **Principal**. Dans le champ **Keytab**, parcourez votre système jusqu'au fichier Keytab à utiliser.

L'utilisateur exécutant un Job utilisant un fichier Keytab n'est pas nécessairement celui désigné par un Principal mais doit avoir le droit de lire le fichier Keytab utilisé. Par exemple, lr nom d'utilisateur que vous utilisez pour exécuter un Job est *user1* et le Principal à utiliser est *guest*. Ici, assurez-vous que *user1* a le droit de lire le fichier Keytab à utiliser.

- Si vous vous connectez à un cluster MapR V4.0.1 ou supérieure et que le système de sécurité de votre cluster par ticket MapR est activé, vous devez cocher la case **Force MapR Ticket Authentication** et configurer les paramètres suivants :

1. Dans le champ **Password**, spécifiez le mot de passe utilisé pour l'authentification de l'utilisateur.

Un ticket de sécurité MapR est généré pour cet utilisateur par MapR et stocké dans la machine où est exécuté le Job que vous configurez.

2. Dans le champ **Cluster name**, saisissez le nom du cluster MapR auquel vous souhaitez que l'utilisateur se connecte.

Ce nom de cluster se trouve dans le fichier *mapr-clusters.conf* situé dans */opt/mapr/conf* du cluster.

3. Dans le champ **Ticket duration**, saisissez la période de temps (en secondes) durant laquelle le ticket est valide.

4. Laissez la case **Launch authentication mechanism when the Job starts** cochée box afin d'assurer que le Job utilisant cette connexion prend en compte la configuration actuelle de la sécurité lorsqu'il commence à s'exécuter.

Si la configuration de la sécurité par défaut de votre cluster MapR a été modifiée, vous devez configurer la connexion pour qu'elle prenne en compte cette configuration personnalisée de la sécurité.

MapR spécifie sa configuration de sécurité dans le fichier *mapr.login.conf* situé dans le dossier */opt/mapr/conf* du cluster. Pour plus d'informations concernant ce fichier de configuration et le service Java utilisé derrière, consultez [mapr.login.conf](#) et [JAAS](#) (liens en anglais).

Procédez comme suit pour la configuration :

1. Vérifiez ce qui a été modifié concernant le fichier *mapr.login.conf*.

Vous devriez pouvoir obtenir les informations relatives à l'administrateur ou au développeur de votre cluster MapR.

2. Si l'emplacement de vos fichiers de configuration MapR a été modifié pour pointer vers un autre emplacement dans le cluster, c'est-à-dire si le répertoire Home MapR a été modifié, cochez la case **Set the MapR Home directory** et saisissez le nouveau répertoire Home. Sinon, laissez cette case décochée pour utiliser le répertoire Home par défaut.

3. Si le module de connexion à utiliser a été modifié dans le fichier *mapr.login.conf*, cochez la case **Specify the Hadoop login configuration** et saisissez le module à appeler depuis le fichier *mapr.login.conf*. Sinon, laissez décochée cette case pour utiliser le module de connexion par défaut.

Par exemple, saisissez *kerberos* pour appeler le module *hadoop\_kerberos* ou *hybrid* pour appeler le module *hadoop\_hybrid*.

- **User name :**

Saisissez le nom d'authentification de l'utilisateur de la distribution Hadoop à utiliser.

Si vous laissez ce champ vide, le Studio utilise votre identifiant à la machine client sur laquelle vous travailler, pour accéder à la distribution Hadoop. Par exemple, si vous utilisez le Studio sur une machine Windows et que votre identifiant est *Company*, l'identifiant utilisé lors de l'exécution est *Company*.

- **Group:**

Saisissez le nom du groupe auquel l'utilisateur authentifié appartient.

Notez que ce champ est activé selon la distribution que vous utilisez.

- **Hadoop properties :**

Si vous devez utiliser une configuration personnalisée pour la distribution Hadoop à utiliser, cliquez sur le bouton [...] pour ouvrir la table des propriétés et ajouter la ou les propriété(s) à personnaliser. Lors de l'exécution, ces modifications écrasent les propriétés par défaut correspondantes utilisées par le Studio pour son moteur Hadoop.

Notez que les propriétés définies dans cette table sont héritées et réutilisées par les connexions filles que vous pouvez créer à partir de cette connexion Hadoop.

Pour plus d'informations concernant les propriétés Hadoop, consultez la documentation Apache Hadoop à l'adresse suivante <http://hadoop.apache.org/docs/current/> (en anglais), ou la documentation de la distribution Hadoop que vous utilisez. Par exemple, la page suivante liste certaines des propriétés Hadoop par défaut : <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml> (en anglais).

Pour plus d'informations concernant cette table des propriétés, consultez *Configuration des propriétés réutilisables de Hadoop*.

- Lorsque la distribution à utiliser est **Microsoft HD Insight**, vous devez paramétrier la configuration de **WebHCat configuration**, **HDInsight configuration** et **Window Azure Storage** au lieu des paramètres mentionnés ci-dessus. En plus des informations d'authentification à fournir dans ces zones de configuration, vous devez configurer les paramètres suivants :
  - Dans le champ **Job result folder**, saisissez l'emplacement dans lequel stocker le résultat d'exécution d'un Job **Talend** dans Azure Storage.
  - Dans le champ **Deployment Blob**, saisissez l'emplacement dans lequel stocker un Job **Talend** et ses bibliothèques dépendantes dans ce compte Azure Storage.  
Une vidéo de démonstration concernant la configuration est disponible à l'adresse suivante : <https://www.youtube.com/watch?v=A3QTT6VsNoM> (en anglais).
- Si vous utilisez Cloudera V5.5 ou une version supérieure, vous pouvez cocher la case **Use Cloudera Navigator** pour permettre au Cloudera Navigator de votre distribution de suivre le lignage de vos Jobs jusqu'au niveau des composants, notamment les modifications des schémas entre les composants.

Vous devez cliquer sur le bouton [...] pour ouvrir la fenêtre [**Cloudera Navigator Wizard**] et définir les paramètres suivants :

1. **Username** et **Password** : informations d'authentification utilisées pour vous connecter à votre Cloudera Navigator.
2. **URL** : saisissez l'emplacement du Cloudera Navigator auquel se connecter.
3. **Metadata URL** : saisissez l'emplacement de Navigator Metadata.
4. **Client URL** : laissez la valeur par défaut.
5. **Autocommit** : cochez cette case pour permettre à Cloudera Navigator de générer le lignage du Job courant à la fin de son exécution.

Comme cette option force le Cloudera Navigator à générer des lignages de toutes ses entités disponibles, tels que les fichiers et répertoires HDFS, les requêtes Hive ou les scripts Pig, il n'est pas recommandé de l'utiliser dans un environnement de production, car elle ralentit le Job.

6. **Die on error** : cochez cette case pour arrêter l'exécution de ce Job lorsque la connexion à Cloudera Navigator échoue.

Sinon, laissez cette case décochée pour que votre Job continue à s'exécuter.

7. **Disable SSL** : cochez cette case pour permettre à votre Job de se connecter à Cloudera Navigator sans processus de validation SSL.

Cette fonctionnalité est conçue pour simplifier les tests de vos Jobs mais il n'est pas recommandé de l'utiliser dans un cluster en production.

Une fois la configuration effectuée, cliquez sur **Finish** pour valider les paramètres.

2. Pour chaque distribution officiellement supportée par **Talend**, une configuration Hadoop par défaut (sous forme d'un fichier .jar) est automatiquement chargée par le Studio afin de compléter les paramètres explicitement définis par vos soins dans l'assistant.

Si vous souhaitez utiliser votre configuration personnalisée afin de remplacer celle par défaut, cochez la case **Use custom Hadoop confs** puis cliquez sur le bouton [...] pour ouvrir l'assistant d'import, pour importer la configuration depuis Ambari, Cloudera Manager ou depuis des fichiers locaux.

Notez que cet import écrase uniquement la configuration Hadoop par défaut utilisée par le Studio mais n'écrase pas les paramètres définis dans l'assistant [**Hadoop cluster connection**].

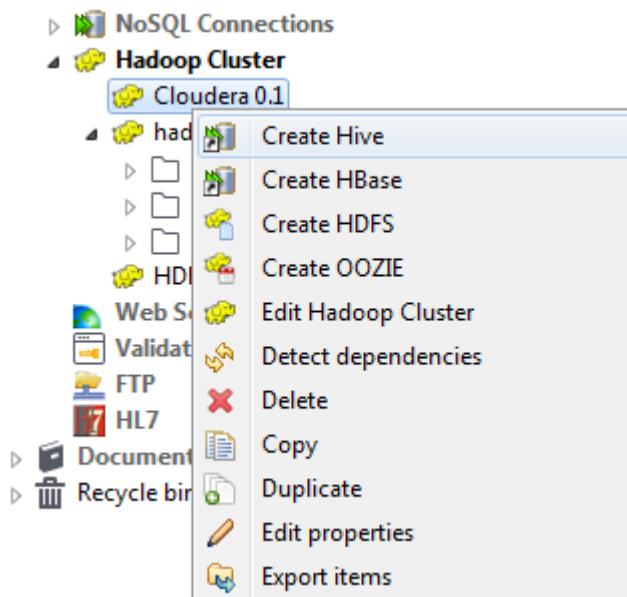
Pour plus d'informations concernant cet import, consultez [Récupérer la configuration depuis Ambari ou Cloudera](#) et [Importer la configuration depuis des fichiers locaux](#).

3. Cliquez sur le bouton **Check services** afin de vérifier que le Studio peut se connecter au NameNode et au JobTracker ou ResourceManager spécifiés dans cet assistant.

Une boîte de dialogue s'ouvre et indique le processus de vérification ainsi que le statut de connexion. Si la connexion échoue, vous devez vérifier et mettre à jour les informations de connexion définies dans l'assistant de connexion.

4. Cliquez sur **Finish** afin de valider vos modifications et fermer l'assistant.

La nouvelle connexion à Hadoop s'affiche dans le dossier **Hadoop cluster** de la vue **Repository**. Cette connexion ne contient pas de sous-dossier tant que vous ne créez pas d'élément dans cette distribution.



## 11.2.2. Connexion à une distribution Hadoop personnalisée

Lorsque vous sélectionnez l'option **Custom** dans la liste déroulante **Distribution** mentionnée ci-dessus, vous pouvez vous connectez à une distribution Hadoop différente des autres distributions Hadoop fournies dans la liste **Distribution** dans le Studio.

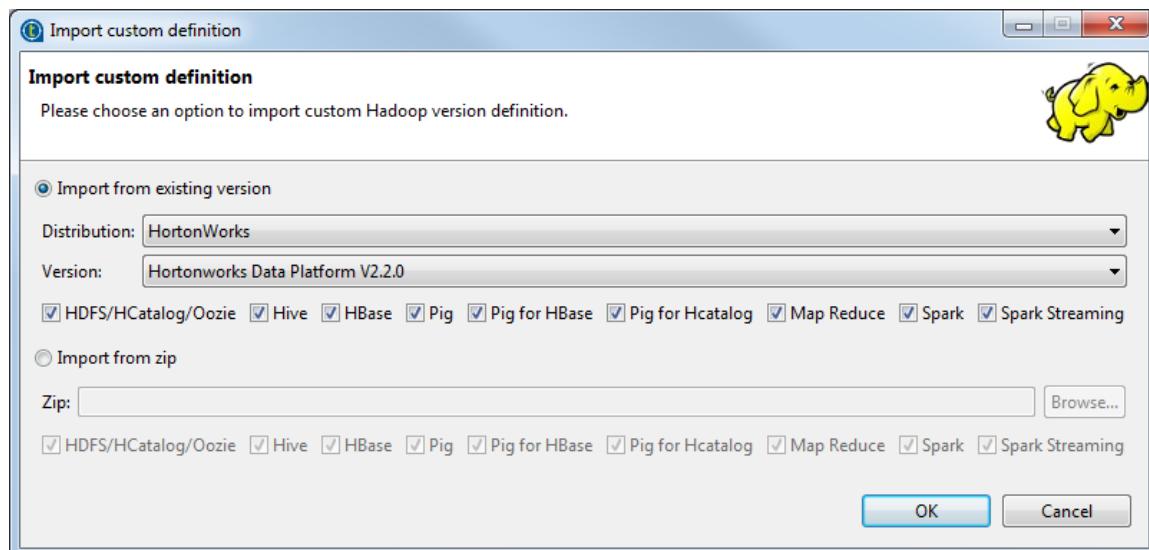
Une fois l'option **Custom** sélectionnée, cliquez sur le bouton  pour afficher la boîte de dialogue **[Import custom definition]**, puis procédez comme suit :

Notez que certaines versions personnalisées ne sont pas officiellement supportées par **Talend**. **Talend** et sa Communauté fournissent l'opportunité de vous connecter à des versions personnalisées depuis le Studio mais ne peuvent garantir que la configuration de la version choisie sera simple, car de nombreuses versions et distributions d'Hadoop différentes sont disponibles. Il est recommandé de configurer une telle connexion uniquement si vos connaissances relatives à Hadoop sont suffisantes pour réparer les problèmes par vous-même.

1. Selon votre cas, sélectionnez **Import from existing version** ou **Import from zip** afin de configurer la distribution Hadoop personnalisée à laquelle vous souhaitez vous connecter.
  - Si vous avez le fichier .zip de configuration de la distribution Hadoop à laquelle vous souhaitez vous connecter, sélectionnez **Import from zip**. Dans **Talend Exchange**, des membres de la Communauté **Talend** ont partagé des fichiers zip de configuration prêts à utiliser, que vous pouvez télécharger depuis cette liste **Hadoop configuration** et utiliser directement dans votre connexion. Cependant, avec l'évolution en continu des différents projets relatifs à Hadoop, il est possible que vous ne trouviez pas dans la liste le zip de configuration correspondant à votre distribution. Il est alors recommandé d'utiliser l'option **Import from existing version**, afin de se baser sur une distribution existante pour ajouter les .jars requis par votre distribution.

Les fichiers .zip sont des fichiers de configuration et ne peuvent être installés directement depuis **Talend Exchange**.

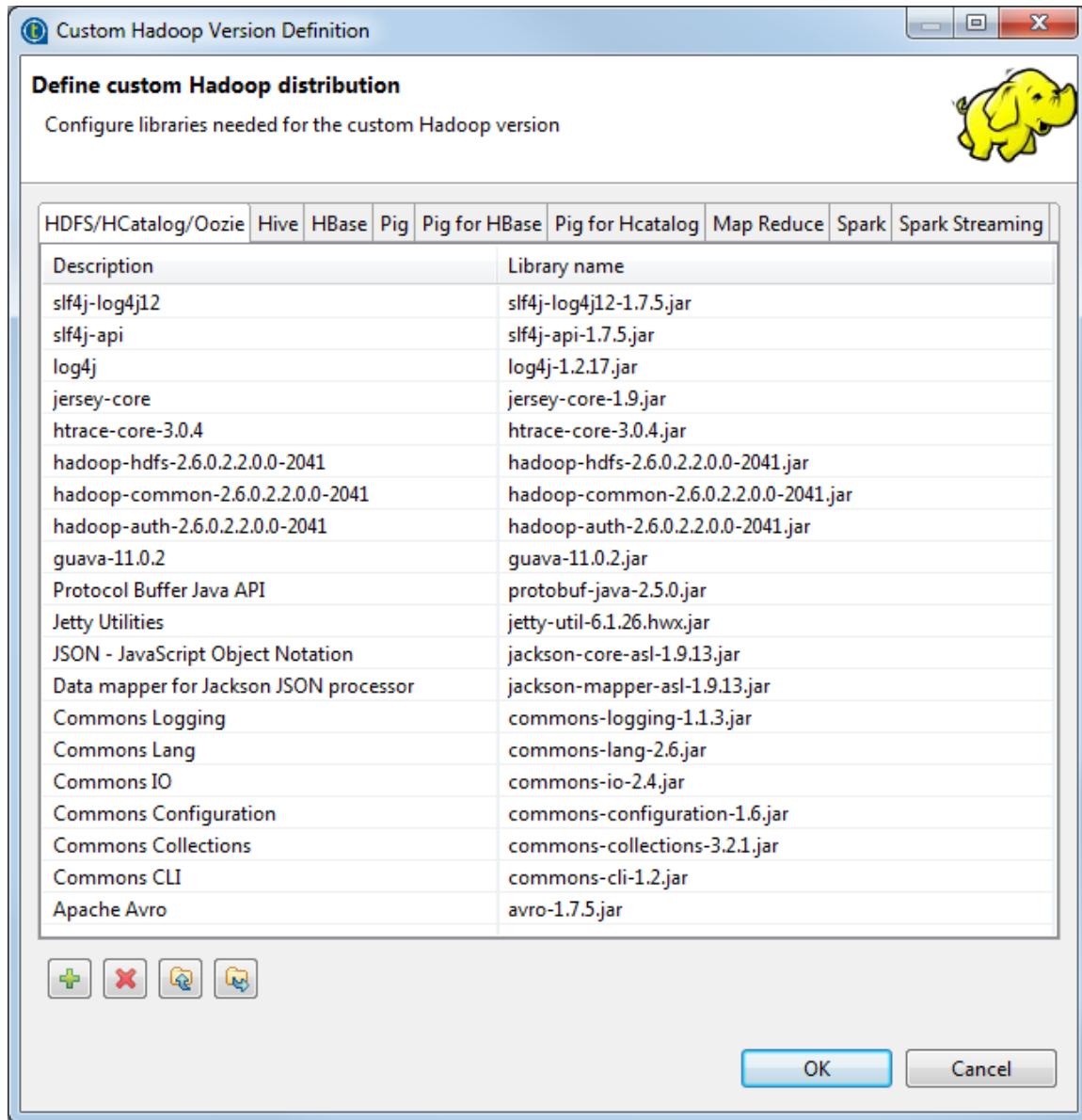
- Sinon, sélectionnez **Import from existing version** pour importer une distribution de base officiellement supportée afin de la personnaliser à l'aide de l'assistant. Adopter cette approche nécessite des connaissances relatives à la configuration de la distribution Hadoop à utiliser.



Notez que les cases de cet assistant vous permettent de sélectionner le(s) élément(s) Hadoop que vous souhaitez importer. Selon le contexte dans lequel vous créez la connection, toutes les cases ne sont pas disponibles. Par exemple, si vous créez cette connexion pour Oozie, seule la case **Oozie** est disponible.

2. Que vous ayez sélectionné **Import from existing version** ou **Import from zip**, vérifiez que chaque case à côté de l'élément Hadoop que vous souhaitez importer est cochée.
3. Cliquez sur **OK** et, dans la fenêtre d'avertissement, cliquez sur **Yes** pour accepter d'écraser toute configuration personnalisée des fichiers Jar précédemment implémentée.

La boîte de dialogue [**Custom Hadoop version definition**] devient active.



Cette boîte de dialogue liste les éléments Hadoop et les fichiers Jar que vous importez.

4. Si vous avez coché l'option **Import from zip**, cliquez sur **OK** afin de valider la configuration importée.
- Si avez sélectionné l'option **Import from existing version**, afin d'importer une distribution de base, vous devez également importer des fichiers jar pour personnaliser cette distribution. Dans l'onglet de l'élément Hadoop que vous souhaitez personnaliser, par exemple l'onglet **HDFS/HCatalog/Oozie**, cliquez sur le bouton **[+]** pour ouvrir la boîte de dialogue [**Select libraries**].
5. Sélectionnez l'option **External libraries** pour ouvrir sa vue.
  6. Cliquez sur le bouton **Browse...** et parcourez votre système jusqu'au fichier Jar que vous souhaitez importer.
  7. Cliquez sur **OK** pour valider les modifications et fermer la boîte de dialogue [**Select libraries**].

Le fichier Jar sélectionné apparaît dans la liste de l'onglet de l'élément Hadoop configuré.

Notez que si vous souhaitez partager la configuration personnalisée Hadoop avec un autre *Studio Talend*, vous pouvez exporter cette connexion personnalisée de la boîte de dialogue **[Custom Hadoop version definition]**

en cliquant sur le bouton 

8. Dans la boîte de dialogue **[Custom Hadoop version definition]**, cliquez sur **OK** afin de valider la configuration personnalisée et retourner à la vue de configuration dans laquelle vous avez sélectionné l'option **Custom**.

Une fois la configuration de la distribution Hadoop personnalisée terminée vous pouvez, depuis la vue de configuration de la connexion Hadoop, continuer de saisir d'autres paramètres requis pour la connexion.

Si la distribution Hadoop personnalisée à laquelle vous souhaitez vous connecter contient YARN, cochez la case **Use YARN** à côté de la liste **Distribution**.

Une vidéo est disponible à l'adresse suivante, pour expliquer, en prenant l'exemple de HDFS, comment configurer une connexion vers un cluster Hadoop personnalisé, également appelé distribution Hadoop non supportée : [How to add an unsupported Hadoop distribution to the Studio](#) (en anglais).

## 11.2.3. Centraliser des métadonnées HBase

Si vous utilisez fréquemment une table d'une base de données HBase, vous pouvez centraliser les informations de connexion à la base de données HBase ainsi que les détails du schéma de la table, dans le dossier **Metadata** de la vue **Repository**.

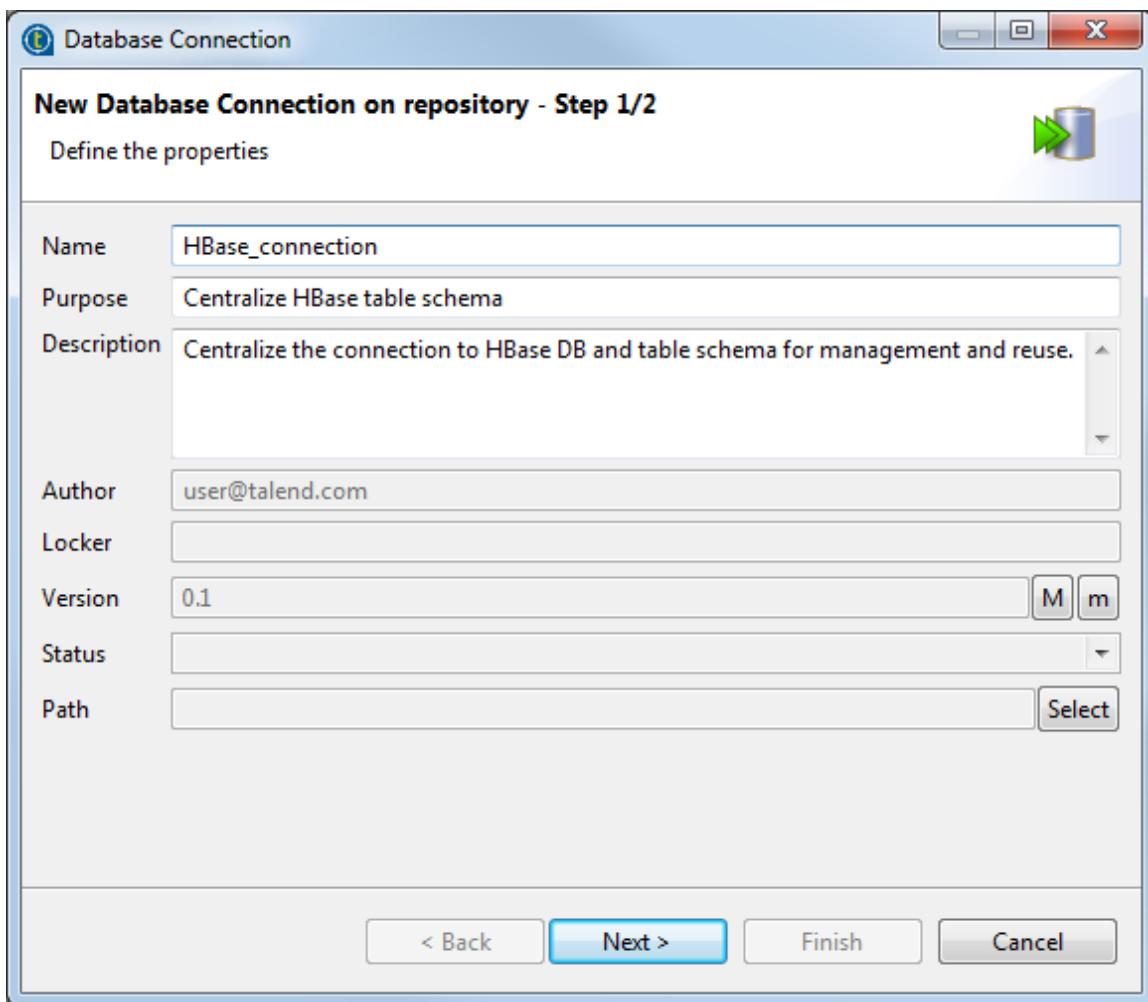
Bien que vous puissiez toujours le faire depuis le nœud **DB connection**, l'utilisation du nœud **Hadoop cluster** est une alternative qui permet une meilleure utilisation des propriétés de connexion centralisées pour une distribution Hadoop donnée.

### Prérequis :

- Lancez la distribution Hadoop que vous souhaitez utiliser et assurez-vous que vous disposez des permissions d'accès nécessaires pour cette distribution et sa base HBase.
- Créez la connexion vers cette distribution Hadoop depuis le nœud **Hadoop cluster**. Pour plus d'informations, consultez [Centraliser une connexion Hadoop](#).

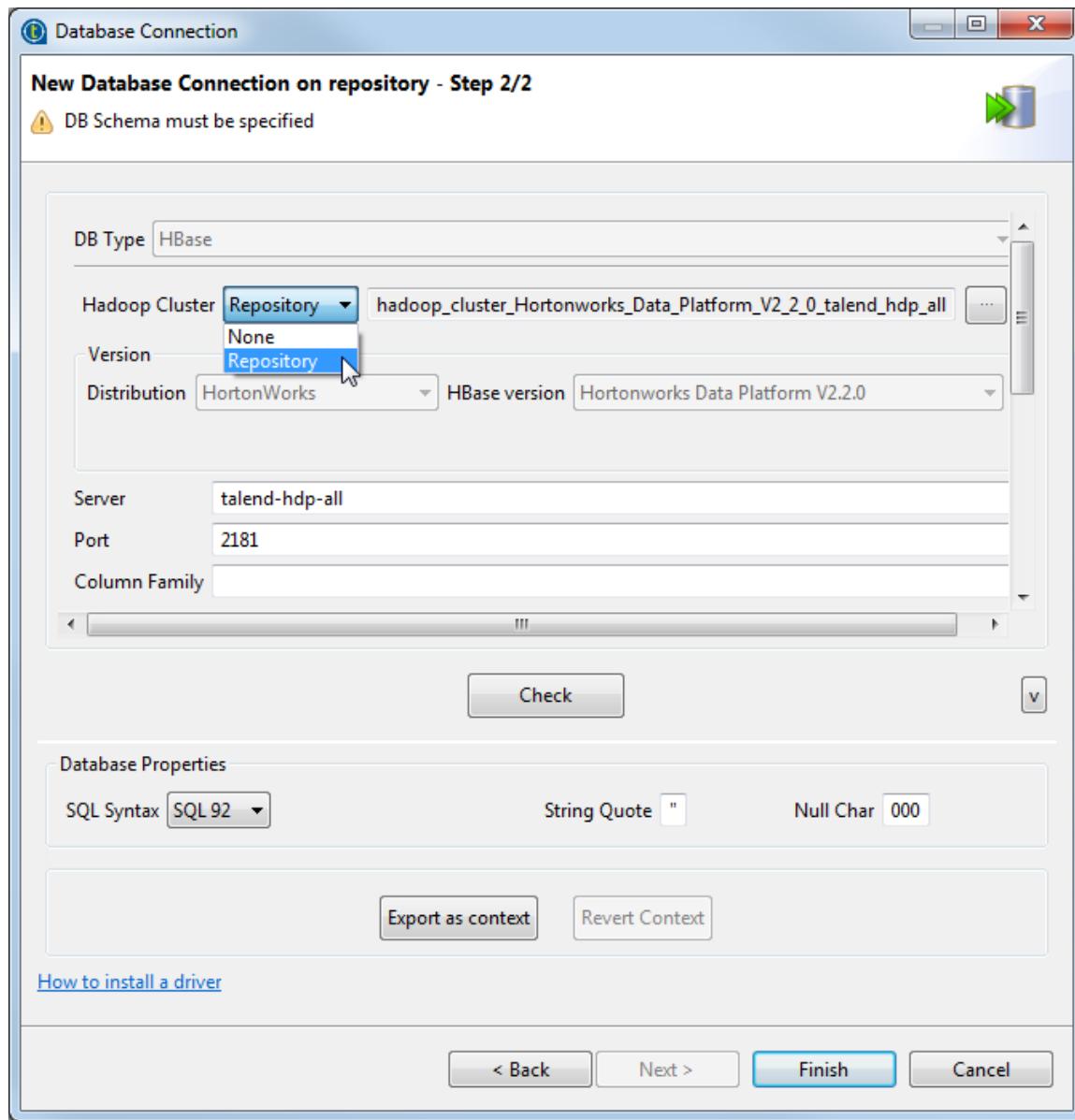
### 11.2.3.1. Créer une connexion à HBase

1. Développez le nœud **Hadoop cluster** situé sous le nœud **Metadata** dans la vue **Repository**. Cliquez-droit sur la connexion Hadoop à utiliser et sélectionnez **Create HBase** dans le menu contextuel.
2. Dans l'assistant de connexion qui s'ouvre, renseignez les propriétés génériques de la connexion que vous devez créer, notamment le nom (champ **Name**), l'objectif (champ **Purpose**) et la **Description**. Le champ **Status** est un champ que vous pouvez personnaliser dans la fenêtre **[File] > [Edit project properties]**.



3. Cliquez sur **Next** pour passer à l'étape suivante et renseigner les informations de connexion à la base de données HBase. Parmi ces informations, **DB Type**, **Hadoop cluster**, **Distribution**, **HBase version** et **Server** sont automatiquement pré-remplies avec les propriétés héritées de la connexion Hadoop que vous avez sélectionnée dans les étapes précédentes.

Notez que si vous choisissez **None** dans la liste **Hadoop cluster**, vous basculez dans un mode manuel dans lequel les données héritées sont abandonnées. Vous devez donc configurer chaque propriété vous-même. La connexion créée apparaît sous le nœud **Db connection** uniquement.



4. Dans le champ **Port**, saisissez le numéro du port de la base de données HBase à laquelle vous connecter.



Afin que le nom d'hôte du serveur Hadoop soit reconnu par les ordinateurs client et hôte, vous devez établir une entrée de mapping de l'adresse IP et du nom d'hôte pour ce nom d'hôte dans le fichier *hosts* des ordinateurs client et hôte. Par exemple, si le nom d'hôte du serveur Hadoop est *talend-all-hdp* et que son adresse IP est *192.168.x.x*, l'entrée de mapping est *192.168.x.x talend-all-hdp*. Pour un système Windows, vous devez ajouter l'entrée dans le fichier *C:\WINDOWS\system32\drivers\etc\hosts* (en supposant que Windows est installé sur le lecteur C). Pour un système Linux, vous devez modifier l'entrée du fichier */etc/hosts*.

5. Dans le champ **Column family**, saisissez, si vous souhaitez filtrer les colonnes, la famille de colonnes puis cliquez sur **Check** afin de vérifier votre connexion.
6. Si vous accédez à une distribution Hadoop s'exécutant avec la sécurité de Kerberos, cochez cette case puis, saisissez le Principal name pour le NameNode dans le champ qui s'affiche. Cela vous permet d'utiliser votre identifiant pour vous connecter avec les informations de connexion stockées dans Kerberos.

Si vous devez utiliser un fichier Keytab pour vous identifier, cochez la case **Use a keytab to authenticate**. Un fichier Keytab contient des paires de Principals et de clés de cryptage Kerberos. Vous devez saisir le Principal à utiliser dans le champ **Principal**. Dans le champ **Keytab**, parcourez votre système jusqu'au fichier Keytab à utiliser.

L'utilisateur exécutant un Job utilisant un Keytab n'est pas nécessairement celui désigné par un Principal mais doit avoir le droit de lire le fichier Keytab utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter un Job est *user1* et le Principal à utiliser est *guest*. Dans cette situation, assurez-vous que *user1* a les droits en lecture sur le fichier Keytab à utiliser.

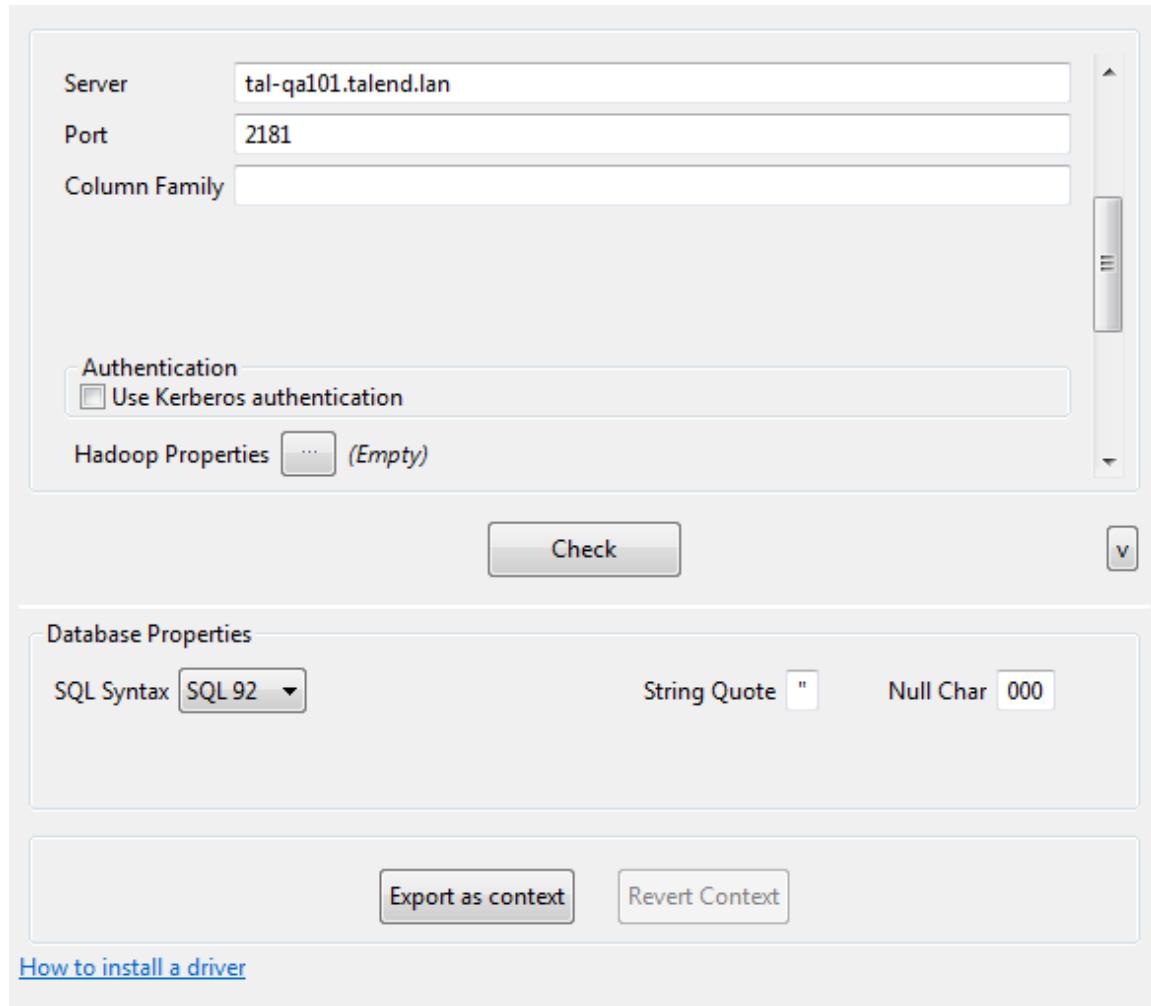
7. Si vous devez utiliser une configuration personnalisée pour la distribution d'Hadoop ou HBase à utiliser, cliquez sur le bouton [...] à côté de **Hadoop properties** afin d'ouvrir la table des propriétés et ajouter la ou les propriété(s) à personnaliser. Lors de l'exécution, les propriétés personnalisées écrasent celles par défaut utilisées par le studio pour son moteur Hadoop.

Une table **Parent Hadoop properties** s'affiche au-dessus de la table des propriétés que vous modifiez. La table mère est en lecture seule et affiche les propriétés Hadoop ayant été définies dans l'assistant de la connexion Hadoop mère, sur laquelle se base la connexion à HBase.

Pour plus d'informations concernant les propriétés de Hadoop, consultez la documentation de Apache Hadoop à l'adresse suivante : <http://hadoop.apache.org/docs/current/> (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. Par exemple, la page suivante liste certaines propriétés Hadoop par défaut <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml>.

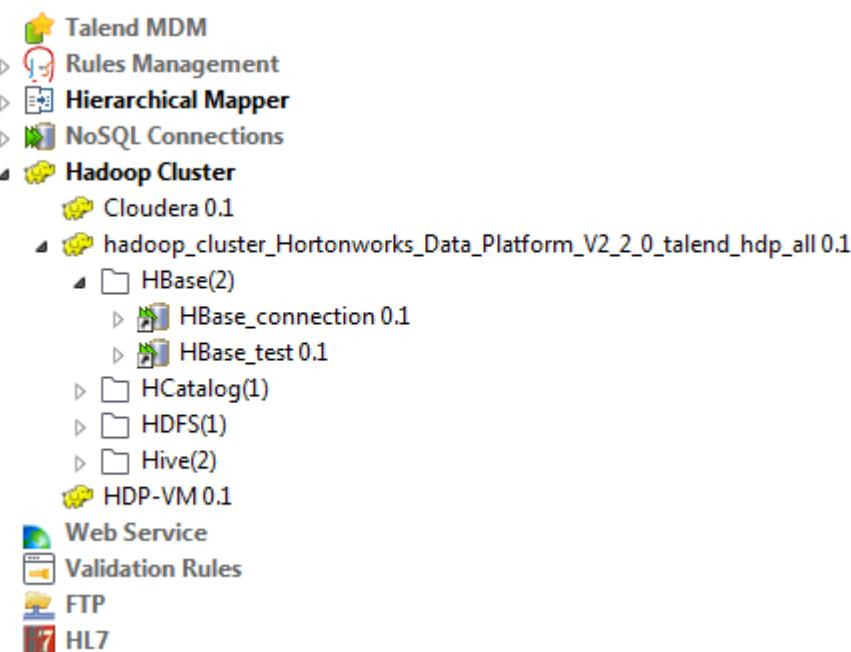
Pour plus d'informations concernant les propriétés de HBase, consultez la documentation de Apache HBase. Par exemple, la page suivante liste certaines propriétés de configuration de HBase par défaut [http://hbase.apache.org/book.html#\\_configuration\\_files](http://hbase.apache.org/book.html#_configuration_files).

Pour tirer parti de cette table des propriétés, consultez [Configuration des propriétés réutilisables de Hadoop](#).



8. Cliquez sur **Finish** pour valider les modifications.

La nouvelle connexion HBase apparaît dans la vue **Repository**, sous le nœud **Hadoop cluster**. De plus, comme la connexion HBase est une connexion de base de données, cette nouvelle connexion apparaît également sous le nœud **Db connections**.



La vue **Repository** dépend de l'édition du studio que vous utilisez.

Si vous devez utiliser un contexte environnemental pour définir les paramètres de cette connexion, cliquez sur le bouton **Export as context** pour ouvrir l'assistant correspondant et choisir parmi les options suivantes :

- **Create a new repository context** : créer le contexte environnemental depuis la connexion Hadoop courante, c'est-à-dire que les paramètres à configurer dans l'assistant sont pris comme variables de contexte avec les valeurs données à ces paramètres.
- **Reuse an existing repository context** : utiliser les variables d'un contexte environnemental afin de configurer la connexion courante.

Si vous devez annuler l'implémentation du contexte, cliquez sur **Revert context**. Les valeurs des variables de contexte utilisées sont directement mises dans l'assistant.

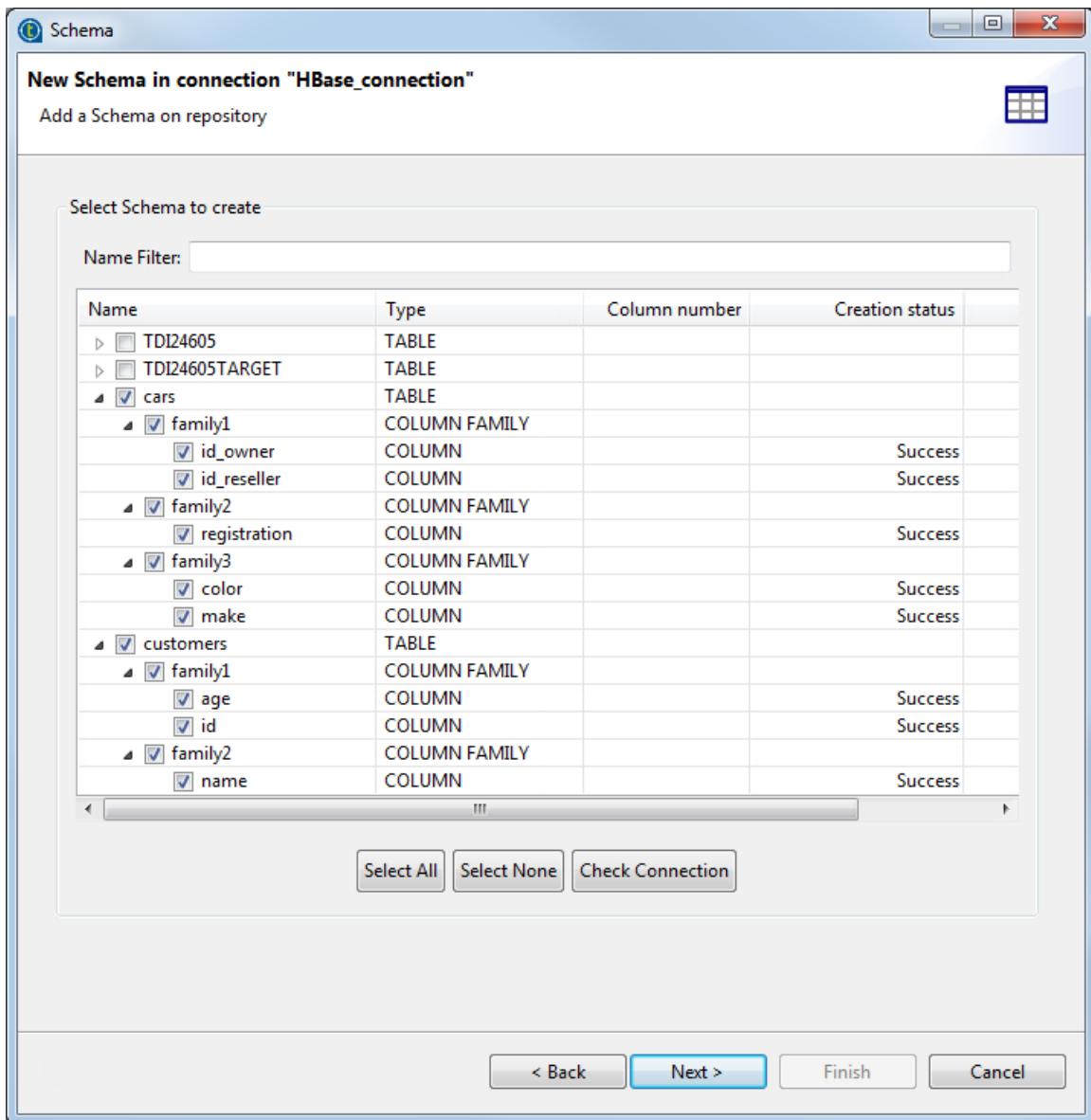
Pour un exemple étape par étape concernant l'utilisation de la fonctionnalité **Export as context**, consultez [Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion](#).

### 11.2.3.2. Récupérer le schéma d'une table

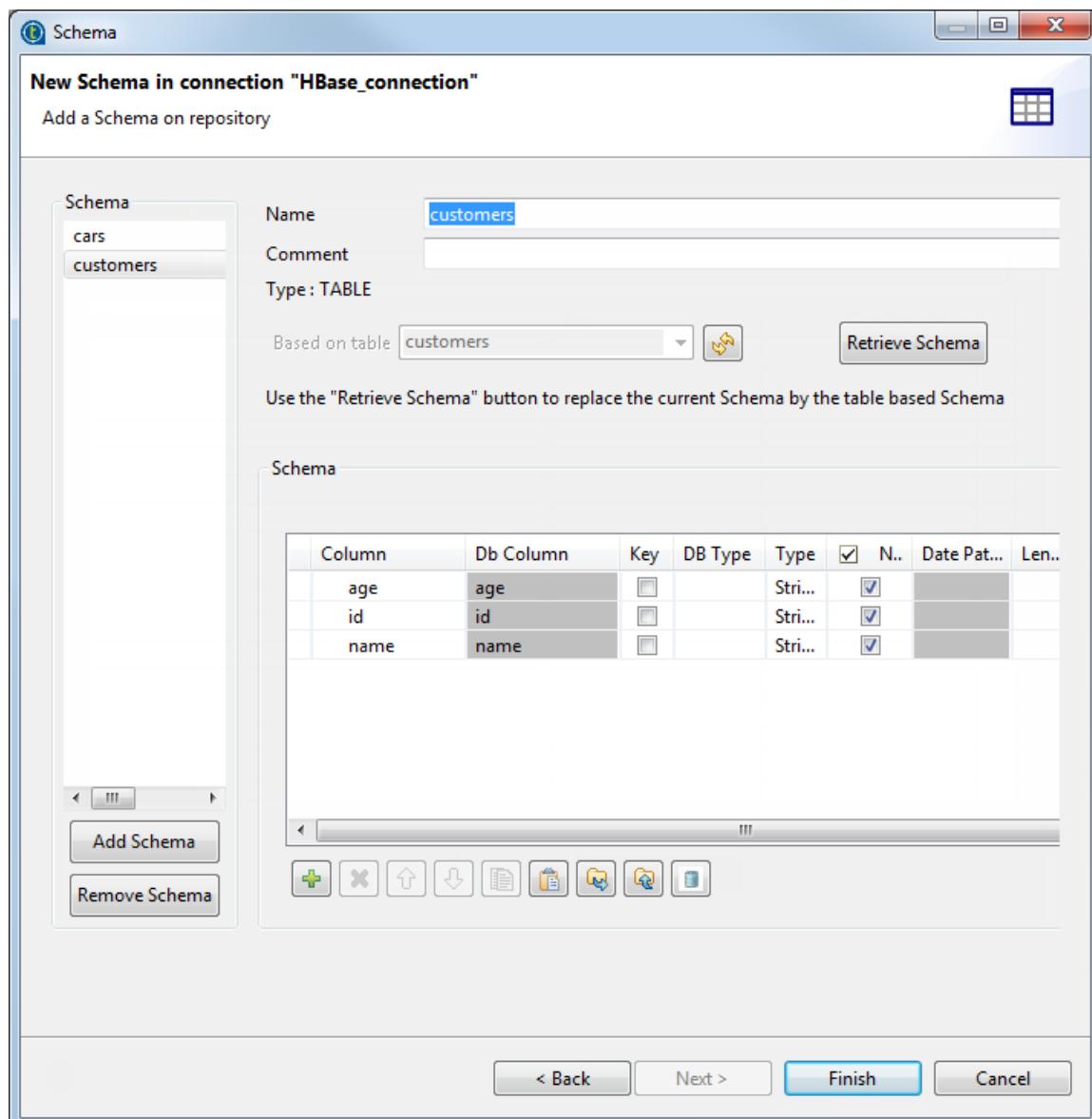
Dans cette étape, vous allez récupérer le schéma de la table de la base de données HBase connectée.

1. Dans la vue **Repository**, cliquez-droit sur la nouvelle connexion et sélectionnez **Retrieve schema** dans le menu contextuel puis cliquez sur **Next** dans l'assistant qui s'ouvre afin de voir et filtrer les différentes tables de la base de données HBase.

Vous pouvez définir le nombre de colonnes à afficher pour chaque famille de colonnes dans le champ **Limit**.



2. Développez le nœud de la table ainsi que la famille de la colonne et sélectionnez les colonnes qui vous intéressent. Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'assistant, listant les schémas de la table sélectionnée. Vous pouvez sélectionner n'importe quel schéma pour afficher ses détails dans la zone **Schema**, du côté droit de l'assistant.



*Si la table de la base de données source contient une valeur par défaut étant une fonction ou une expression et non une chaîne de caractères, assurez-vous de supprimer les guillemets simples entourant la valeur par défaut du schéma final, s'il y en a, afin d'éviter des résultats inattendus lors de la création de tables de bases de données à l'aide de ce schéma.*

*Pour plus d'informations, consultez Vérification des valeurs par défaut dans un schéma récupéré à l'adresse <https://help.talend.com>.*

- Modifiez le schéma si nécessaire. Vous pouvez renommer le schéma et personnaliser sa structure selon vos besoins, dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.

Pour écraser les modifications effectuées sur le schéma sélectionné avec le schéma par défaut, cliquez sur **Retrieve schema**. Notez que toutes les modifications apportées au schéma seront perdues si vous cliquez sur ce bouton.

- Cliquez sur **Finish** pour terminer la création du schéma de la base de données. Tous les schémas récupérés sont affichés sous le noeud **HBase connection** correspondant dans la vue **Repository**.

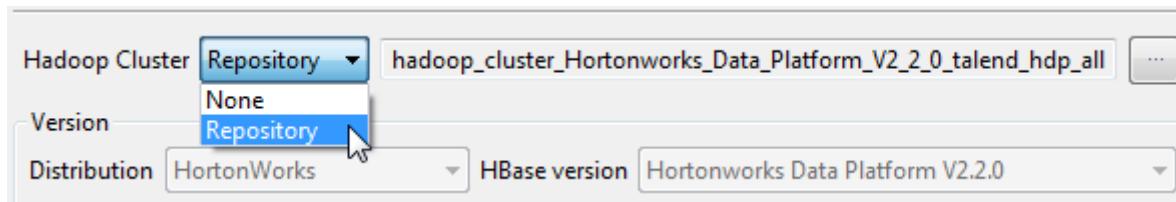
Si vous devez éditer ultérieurement le schéma, cliquez-droit sur le schéma et sélectionnez **Edit Schema** dans le menu contextuel pour ouvrir à nouveau l'assistant et effectuer vos modifications.



*Si vous modifiez le schéma, assurez-vous que le Type des données est correctement défini.*

Comme expliqué précédemment, vous pouvez utiliser le noeud **Hadoop cluster** ou également créer une connexion HBase et récupérer les schémas du noeud **Db connection**. Dans tous les cas, vous devez toujours configurer les propriétés spécifiques de connexion à HBase. A cette étape :

- si vous sélectionnez l'option **Repository** dans la liste **Hadoop cluster** pour réutiliser les détails d'une connexion Hadoop établie, la connexion HBase créée sera classée à la fois sous le noeud **Hadoop cluster** et le noeud **Db connection** ;
- sinon, si vous sélectionnez l'option **None** dans la liste **Hadoop cluster** afin de saisir vous-même les propriétés de la connexion Hadoop, la connexion HBase créée apparaît uniquement sous le noeud **Db connection**.



## 11.2.4. Centraliser les métadonnées MapR-DB

Si vous utilisez régulièrement une table d'une base de données MapR-DB, vous pouvez centraliser les informations de connexion à MapR-DB et les informations du schéma de la table dans le dossier **Metadata** de la vue **Repository**.

Même si vous pouvez le faire depuis le noeud **DB connection**, utiliser le noeud **Hadoop cluster** est recommandé, afin d'utiliser au mieux les propriétés de la connexion centralisée pour une distribution MapR donnée.

### Prérequis :

- Lancez la distribution MapR à utiliser et assurez-vous que vous avez les bonnes autorisations d'accès à cette distribution et à sa base de données MapR-DB.
- Créez la connexion à cette distribution MapR à partir du noeud du cluster Hadoop. Pour plus d'informations, consultez [Centraliser une connexion Hadoop](#).

### 11.2.4.1. Créer une connexion à MapR-DB

1. Développez le noeud **Hadoop cluster** sous **Metadata**, dans la vue **Repository**, cliquez-droit sur la connexion MapR à utiliser et sélectionnez **Create MapRDB** dans le menu contextuel.
2. Dans l'assistant de connexion qui s'ouvre, renseignez les propriétés génériques de la connexion à créer, telles que le nom (**Name**), l'objectif (**Purpose**) et la **Description**. Le champ **Status** est un champ personnalisé que vous pouvez configurer dans **File > Edit project properties**.
3. Cliquez sur **Next** pour passer à l'étape suivante, qui nécessite de renseigner les informations de connexion à MapR-DB. Parmi ces informations, les champs (**DB Type**), **Hadoop cluster**, **Distribution**, **MapR-DB version** et **Server** sont automatiquement prérenseignés par les propriétés héritées de la connexion MapR sélectionnée dans les étapes précédentes.

Notez que, si vous sélectionnez **None** dans la liste **Hadoop cluster**, vous passez en mode manuel. Dans ce mode, les propriétés héritées ne sont pas prises en compte. Vous devez configurer manuellement les propriétés. La connexion créée s'affiche sous le noeud **Db connection** uniquement.

4. Dans le champ **Port**, saisissez le numéro du port de la base de données MapR-DB à laquelle vous connecter. Le numéro par défaut est 5181, qui est le port des nœuds exécutant les services Zookeeper.



Afin de rendre reconnaissable le nom de l'hôte du serveur MapR par les machines client et hôte, vous devez établir une entrée de mapping adresse IP/nom de l'hôte pour ce nom d'hôte, dans les fichiers *hosts* relatifs des ordinateurs client et hôte. Par exemple, le nom d'hôte du serveur MapR est *myMapR* et son adresse IP est *192.168.x.x*. L'entrée de mapping est la suivante *192.168.x.x myMapR*. Pour les systèmes Windows, vous devez ajouter l'entrée au fichier *C:\WINDOWS\system32\drivers\etc\hosts* (si Windows est installé sur votre disque C). Pour les systèmes sous Linux, vous devez ajouter l'entrée au fichier */etc/hosts*.

5. Dans le champ **Column family**, saisissez le nom de la famille de colonnes, si vous souhaitez filtrer les colonnes, puis cliquez sur le bouton **Check** afin de vérifier votre connexion
6. Si la base de données à utiliser s'exécute avec la sécurité Kerberos, cochez la case **User Kerberos authentication**, puis saisissez les noms des Principaux dans les champs qui s'affichent. Vous devriez trouver l'information dans le fichier *hbase-site.xml* du cluster MapR à utiliser.

Si vous devez utiliser un fichier keytab pour vous authentifier, cochez la case **Use a keytab to authenticate**. Un fichier keytab contient des paires de Principaux Kerberos et de clés cryptées. Vous devez saisir le Principal à utiliser dans le champ **Principal** et, dans le champ **Keytab**, parcourrez votre système jusqu'au fichier keytab à utiliser.

Notez que l'utilisateur exécutant un Job contenant un keytab n'est pas nécessairement celui désigné par un Principal, mais qu'il doit posséder les autorisations en lecture sur le fichier utilisé. Par exemple, le nom d'utilisateur que vous utilisez pour exécuter un Job est *user1* et le Principal à utiliser est *guest*. Dans cette situation, assurez-vous que l'utilisateur *user1* a le droit de lire le fichier keytab à utiliser.

7. Si le cluster MapR à utiliser est sécurisé par un mécanisme d'authentification par ticket MapR, cochez la case **Force MapR Ticket authentication**, afin de paramétrier la sécurité associée.

1. Cochez la case **Force MapR ticket authentication** pour afficher les paramètres à configurer.
2. Dans le champ **Username**, saisissez l'utilisateur à authentifier et, dans le champ **Password**, spécifiez le mot de passe utilisé par cet utilisateur.

Un ticket de sécurité MapR est généré pour cet utilisateur par MapR et est stocké dans la machine où s'exécute le Job que vous configurez.

3. Si le champ **Group** est disponible dans cet onglet, vous devez saisir le nom du groupe auquel l'utilisateur à authentifier appartient.
4. Dans le champ **Cluster name**, saisissez le nom du cluster MapR auquel vous souhaitez cette utilisateur.

Ce nom de cluster se trouve dans le fichier *mapr-clusters.conf* situé dans le répertoire */opt/mapr/conf* du cluster.

5. Dans le champ **Ticket duration**, saisissez la durée (en secondes) durant laquelle un ticket est valide.
8. Si vous devez utiliser une configuration personnalisée pour la distribution MapR-DB à utiliser, cliquez sur le bouton [...] à côté de **Hadoop properties**, pour ouvrir la table des propriétés et ajouter la ou les propriété(s) à personnaliser. Lors de l'exécution, ces modifications vont écraser les propriétés par défaut correspondantes utilisées par le Studio pour son moteur Hadoop.

Notez qu'une table **Parent Hadoop properties** s'affiche au-dessus de la table que vous modifiez. Cette table parent est en lecture seule et liste les propriétés MapR définies dans l'assistant de la connexion MapR parent sur laquelle se base la connexion MapR-DB courante.

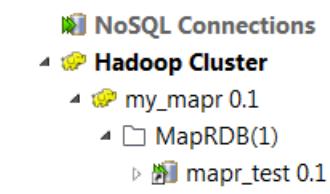
Pour plus d'informations concernant les propriétés de MapR, consultez la documentation de MapR ou une documentation plus générale d'Apache Hadoop.

Comme HBase et MapR-DB sont étroitement liées, pour plus d'informations concernant les propriétés de MapR-DB, consultez la documentation Apache pour HBase. Par exemple, la page suivante décrit quelques propriétés de configuration HBase : [http://hbase.apache.org/book.html#\\_configuration\\_files](http://hbase.apache.org/book.html#_configuration_files) (en anglais).

Pour plus d'informations concernant cette table de propriétés, consultez *Configuration des propriétés réutilisables de Hadoop*.

- Cliquez sur **Finish** afin de valider les modifications.

La nouvelle connexion MapR s'affiche sous le nœud Hadoop de la vue **Repository**. De plus, comme une connexion à MapR-DB est une connexion à une base de données, celle-ci s'affiche également sous le nœud **Db connections**.



Si vous devez utiliser un contexte environnemental pour définir les paramètres de cette connexion, cliquez sur le bouton **Export as context** pour ouvrir l'assistant correspondant et choisir parmi les options suivantes :

- Create a new repository context** : créer le contexte environnemental depuis la connexion Hadoop courante, c'est-à-dire que les paramètres à configurer dans l'assistant sont pris comme variables de contexte avec les valeurs données à ces paramètres.
- Reuse an existing repository context** : utiliser les variables d'un contexte environnemental afin de configurer la connexion courante.

Si vous devez annuler l'implémentation du contexte, cliquez sur **Revert context**. Les valeurs des variables de contexte utilisées sont directement mises dans l'assistant.

Pour un exemple étape par étape concernant l'utilisation de la fonctionnalité **Export as context**, consultez *Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion*.

#### 11.2.4.2. Récupérer le schéma d'une table

Le moyen de récupérer un schéma d'une table MapR-DB est le même que celui pour récupérer un schéma d'une table HBase. Pour plus d'informations concernant la récupération d'un schéma HBase, consultez *Récupérer le schéma d'une table*.

#### 11.2.5. Centraliser des métadonnées HCatalog

Si vous utilisez régulièrement une table d'une base de données de HCatalog, une couche de gestion des tables et du stockage pour Hadoop, vous pouvez centraliser les informations de connexion à HCatalog, ainsi que les détails du schéma de la table, dans le dossier **Metadata** de la vue **Repository**.

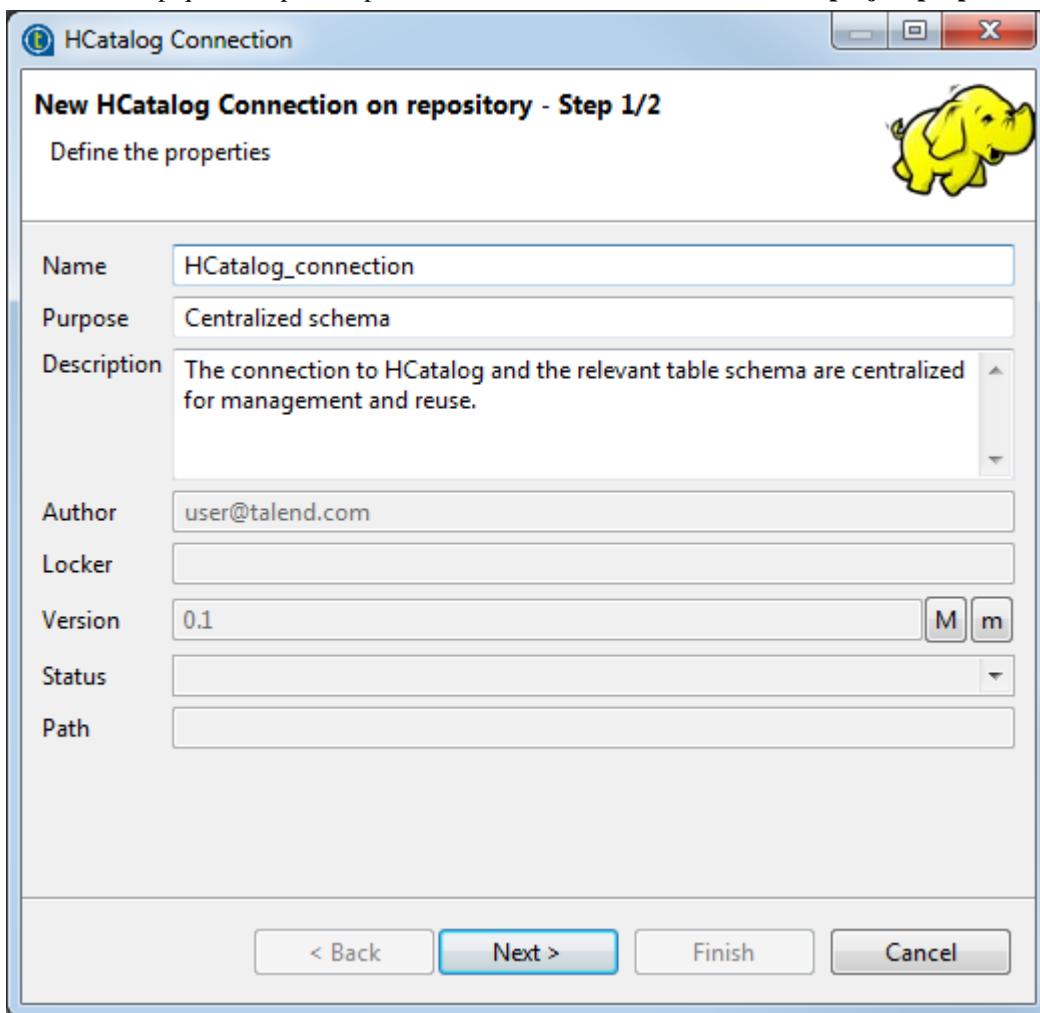
**Prérequis :**

- Lancez la distribution HortonWorks Hadoop que vous souhaitez utiliser et assurez-vous que vous disposez des permissions d'accès nécessaires pour cette distribution et la base de données HCatalog.

- Créez la connexion vers cette distribution Hadoop depuis le nœud **Hadoop cluster**. Pour plus d'informations, consultez [Centraliser une connexion Hadoop](#).

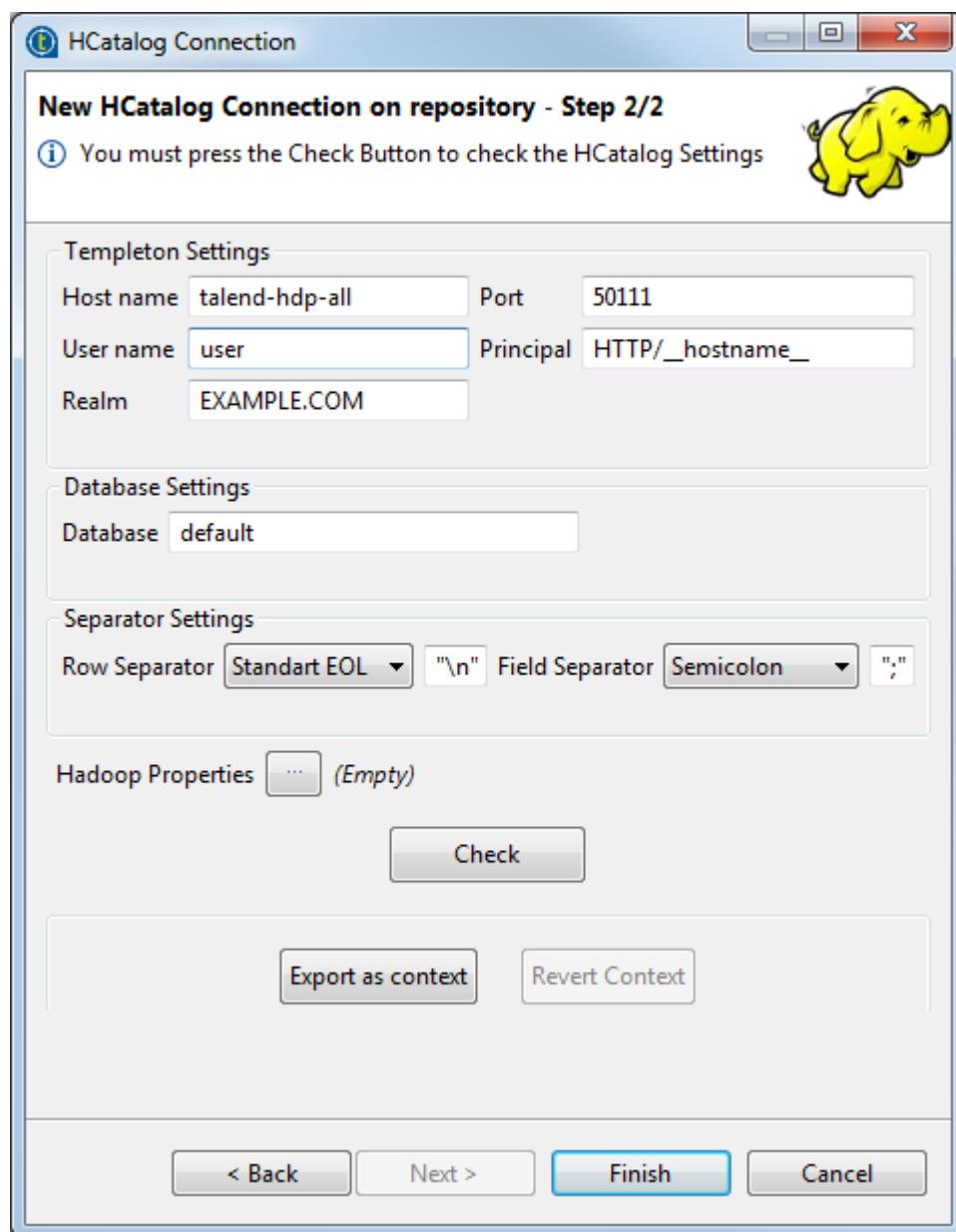
### 11.2.5.1. Créer une connexion à HCatalog

1. Développez le nœud **Hadoop cluster** situé sous le nœud **Metadata** dans la vue **Repository**. Cliquez-droit sur la connexion Hadoop à utiliser et sélectionnez **Create HCatalog** dans le menu contextuel.
2. Dans l'assistant de connexion qui s'ouvre, renseignez les propriétés génériques de la connexion que vous devez créer, notamment le nom (champ **Name**), l'objectif (champ **Purpose**) et la **Description**. Le champ **Status** est un champ que vous pouvez personnaliser dans la fenêtre [File] > [Edit project properties].



3. Cliquez sur **Next** pour passer à l'étape suivante et renseigner les informations de connexion à la base de données Hive. Parmi ces informations, le champ **Host name** est automatiquement renseigné avec la valeur héritée de la connexion Hadoop que vous avez sélectionnée dans les étapes précédentes. Le champ **Port** de Templeton et le champ **Database** utilisent des valeurs par défaut.

Cette base de données est une base de données Hive et Templeton est utilisé en tant qu'API Web de type REST par HCatalog afin d'émettre des commandes. Pour plus d'informations sur Templeton, consultez la documentation d'Apache sur la page [http://people.apache.org/~thejas/templeton\\_doc\\_latest/index.html](http://people.apache.org/~thejas/templeton_doc_latest/index.html) (en anglais).



Les champs **Principal** et **Realm** ne sont affichés que si la connexion Hadoop utilisée autorise la sécurité Kerberos. Ce sont les propriétés requises par Kerberos pour authentifier le client HCatalog et le serveur HCatalog entre eux.



Afin que le nom d'hôte du serveur Hadoop soit reconnu par les ordinateurs client et hôte, vous devez établir une entrée de mapping de l'adresse IP et du nom d'hôte pour ce nom d'hôte dans le fichier *hosts* des ordinateurs client et hôte. Par exemple, si le nom d'hôte du serveur Hadoop est *talend-all-hdp* et que son adresse IP est *192.168.x.x*, l'entrée de mapping est *192.168.x.x talend-all-hdp*. Pour un système Windows, vous devez ajouter l'entrée dans le fichier *C:\WINDOWS\system32\drivers\etc\hosts* (si Windows est installé sur le lecteur C). Pour un système Linux, vous devez modifier l'entrée du fichier */etc/hosts*.

4. Si nécessaire, remplacez ces valeurs par défaut par celles du port et de la base de données utilisées votre HCatalog.
5. Si besoin, renseignez les propriétés **Principal** et **Realm**.
6. Si vous devez utiliser une configuration personnalisée pour la distribution d'Hadoop ou HCatalog à utiliser, cliquez sur le bouton [...] à côté de **Hadoop properties** pour ouvrir la table des propriétés et ajouter la ou les propriété(s) à personnaliser. Lors de l'exécution, les propriétés personnalisées écrasent celles par défaut utilisées par le studio pour son moteur Hadoop.

Une table **Parent Hadoop properties** s'affiche au-dessus de la table des propriétés que vous modifiez. La table mère est en lecture seule et affiche les propriétés Hadoop ayant été définies dans l'assistant de la connexion Hadoop mère, sur laquelle se base la connexion à HCatalog.

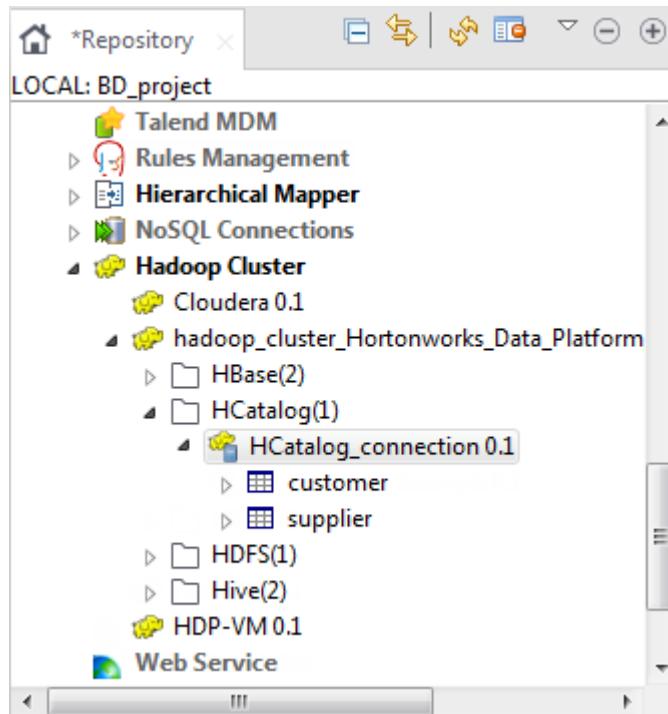
Pour plus d'informations concernant les propriétés de Hadoop, consultez la documentation de Apache Hadoop à l'adresse suivante : <http://hadoop.apache.org/docs/current/> (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. Par exemple, cette page liste certaines propriétés Hadoop par défaut <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml> (en anglais).

Pour plus d'informations concernant les propriétés de HCatalog, consultez la documentation Apache HCatalog. Par exemple, la page suivante décrit certaines propriétés de configuration HCatalog : <https://cwiki.apache.org/confluence/display/Hive/HCatalog+Configuration+Properties> (en anglais).

Pour tirer parti de cette table des propriétés, consultez *Configuration des propriétés réutilisables de Hadoop*.

7. Cliquez sur le bouton **Check** afin de tester la connexion que vous venez de définir. Un message s'affiche indiquant si la connexion est établie.
8. Cliquez sur **Finish** pour valider vos modifications.

La nouvelle connexion HCatalog créée est disponible sous le nœud **Hadoop cluster** dans la vue **Repository**.



Cette vue **Repository** peut varier suivant l'édition du studio que vous utilisez.

Si vous devez utiliser un contexte environnemental pour définir les paramètres de cette connexion, cliquez sur le bouton **Export as context** pour ouvrir l'assistant correspondant et choisir parmi les options suivantes :

- **Create a new repository context** : créer le contexte environnemental depuis la connexion Hadoop courante, c'est-à-dire que les paramètres à configurer dans l'assistant sont pris comme variables de contexte avec les valeurs données à ces paramètres.
- **Reuse an existing repository context** : utiliser les variables d'un contexte environnemental afin de configurer la connexion courante.

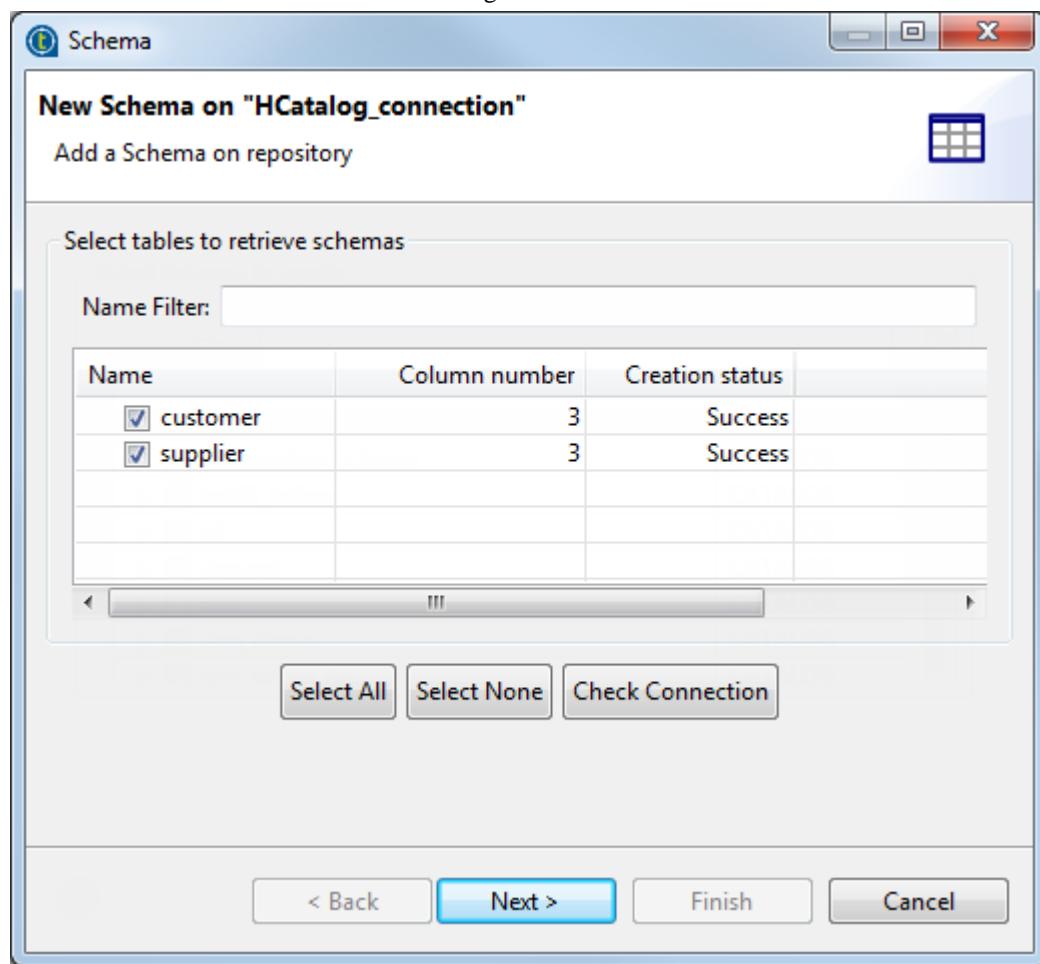
Si vous devez annuler l'implémentation du contexte, cliquez sur **Revert context**. Les valeurs des variables de contexte utilisées sont directement mises dans l'assistant.

Pour un exemple étape par étape concernant l'utilisation de la fonctionnalité **Export as context**, consultez [Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion](#).

9. Cliquez-droit sur la nouvelle connexion et sélectionnez **Retrieve schema** dans la liste déroulante, afin de charger le schéma de la table souhaitée.

### 11.2.5.2. Récupérer le schéma d'une table

1. Lorsque vous cliquez sur **Retrieve Schema**, un nouvel assistant s'ouvre dans lequel vous pouvez filtrer et afficher les différentes tables de HCatalog.

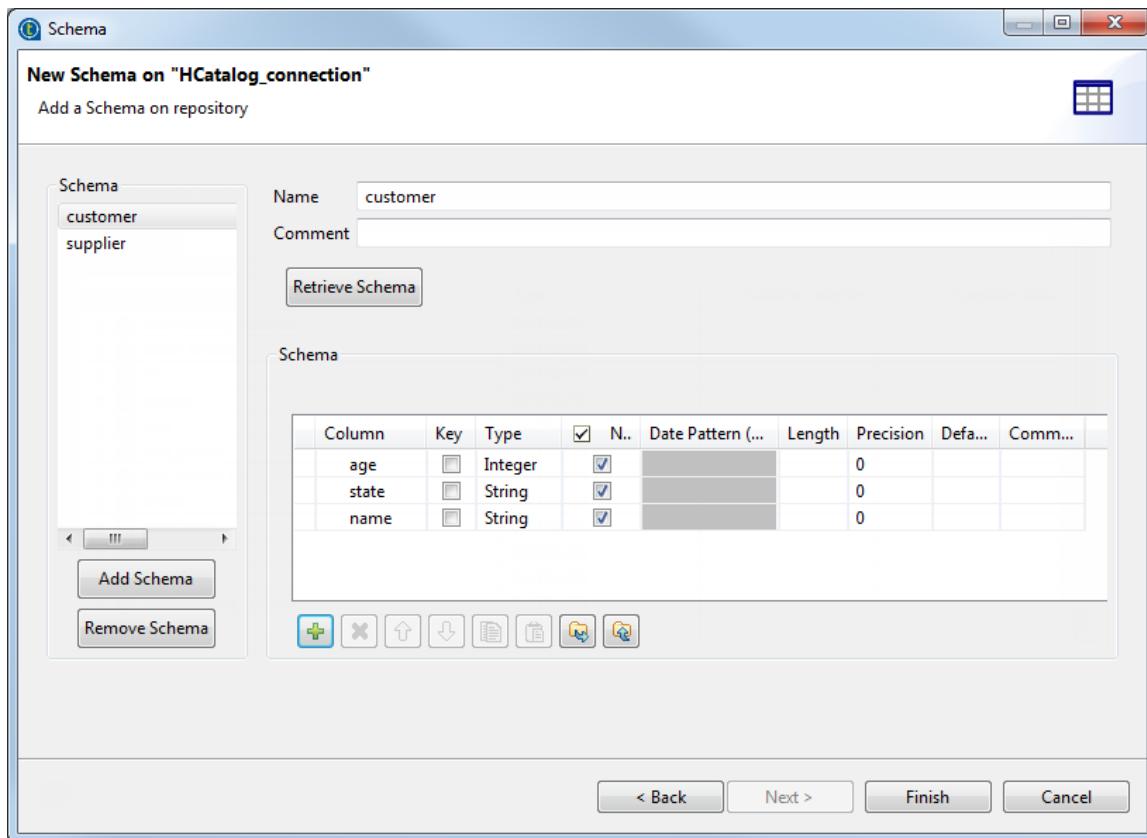


2. Dans le champ **Name filter**, saisissez le nom de la (les) table(s) que vous recherchez.

Vous pouvez également trouver et sélectionner directement la (les) table(s) de laquelle (desquelles) vous souhaitez récupérer le schéma.

Chaque fois qu'un schéma est récupéré pour un fichier sélectionné, le statut **Creation status** du fichier devient *Success*.

3. Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'assistant, listant tous les schémas sélectionnés. Vous pouvez en sélectionner un pour afficher ses détails dans la zone **Schema**.



4. Modifiez le schéma sélectionné, si nécessaire. Vous pouvez changer son nom ou personnaliser sa structure, dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, de supprimer ou de déplacer des colonnes dans le schéma.

Pour écraser les modifications effectuées sur le schéma sélectionné et retrouver le schéma par défaut, cliquez sur **Retrieve schema**. Notez que cette opération écrase toutes les modifications personnalisées.

5. Cliquez sur **Finish** pour terminer la création du schéma HCatalog. Tous les schémas récupérés s'affichent sous le nœud des connexions HCatalog correspondant, dans la vue **Repository**.

Si vous devez modifier un schéma, cliquez-droit sur ce schéma, sous le nœud des connexions HCatalog dans le **Repository** et, dans le menu contextuel, sélectionnez **Edit Schema** pour ouvrir à nouveau l'assistant et effectuer les modifications.



Si vous modifiez les schémas, assurez-vous que le type de données est bien défini dans la colonne **Type**.

## 11.2.6. Centraliser des métadonnées HDFS

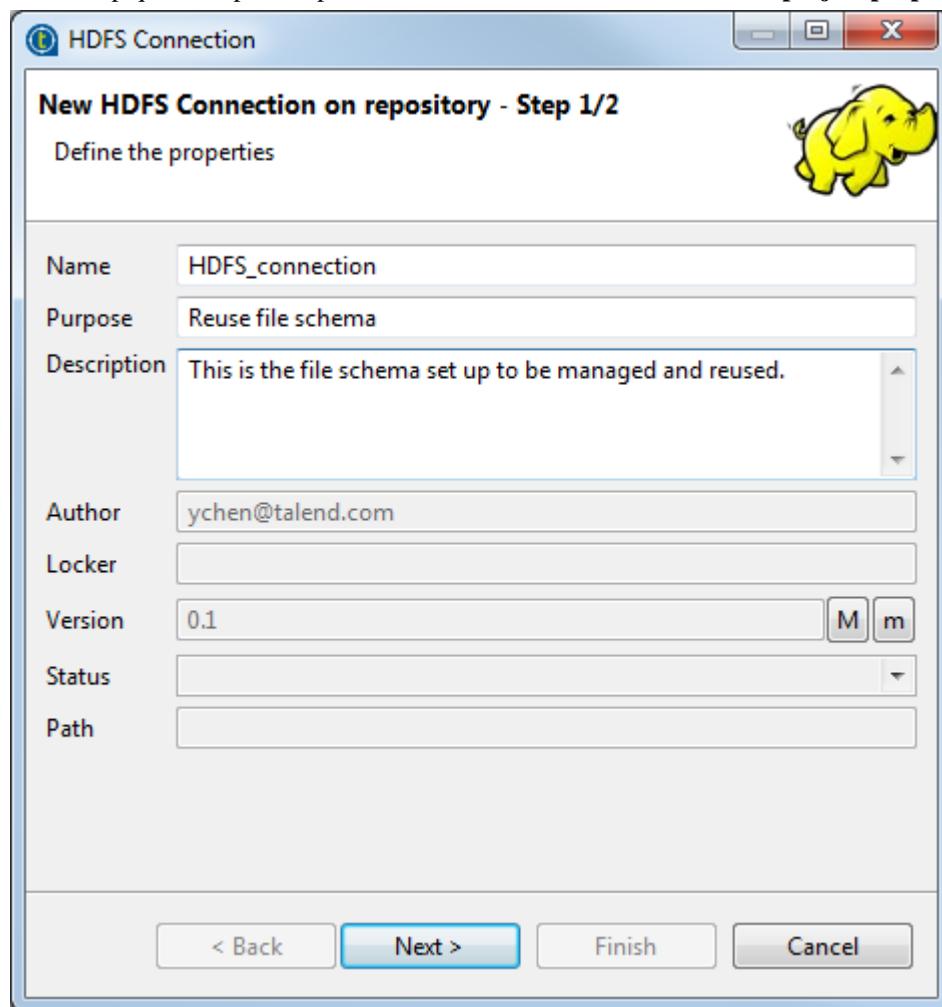
Si vous utilisez souvent un schéma de fichier provenant de HDFS, le système de fichiers distribué Hadoop, vous pouvez centraliser les informations de connexion à HDFS ainsi que les détails du schéma dans le dossier **Metadata** dans la vue **Repository**.

### Prérequis :

- Lancez la distribution Hadoop que vous souhaitez utiliser et assurez-vous que vous disposez des permissions d'accès nécessaires pour cette distribution et son système de fichiers HDFS.
- Créez la connexion vers cette distribution Hadoop depuis le nœud **Hadoop cluster**. Pour plus d'informations, consultez [Centraliser une connexion Hadoop](#).

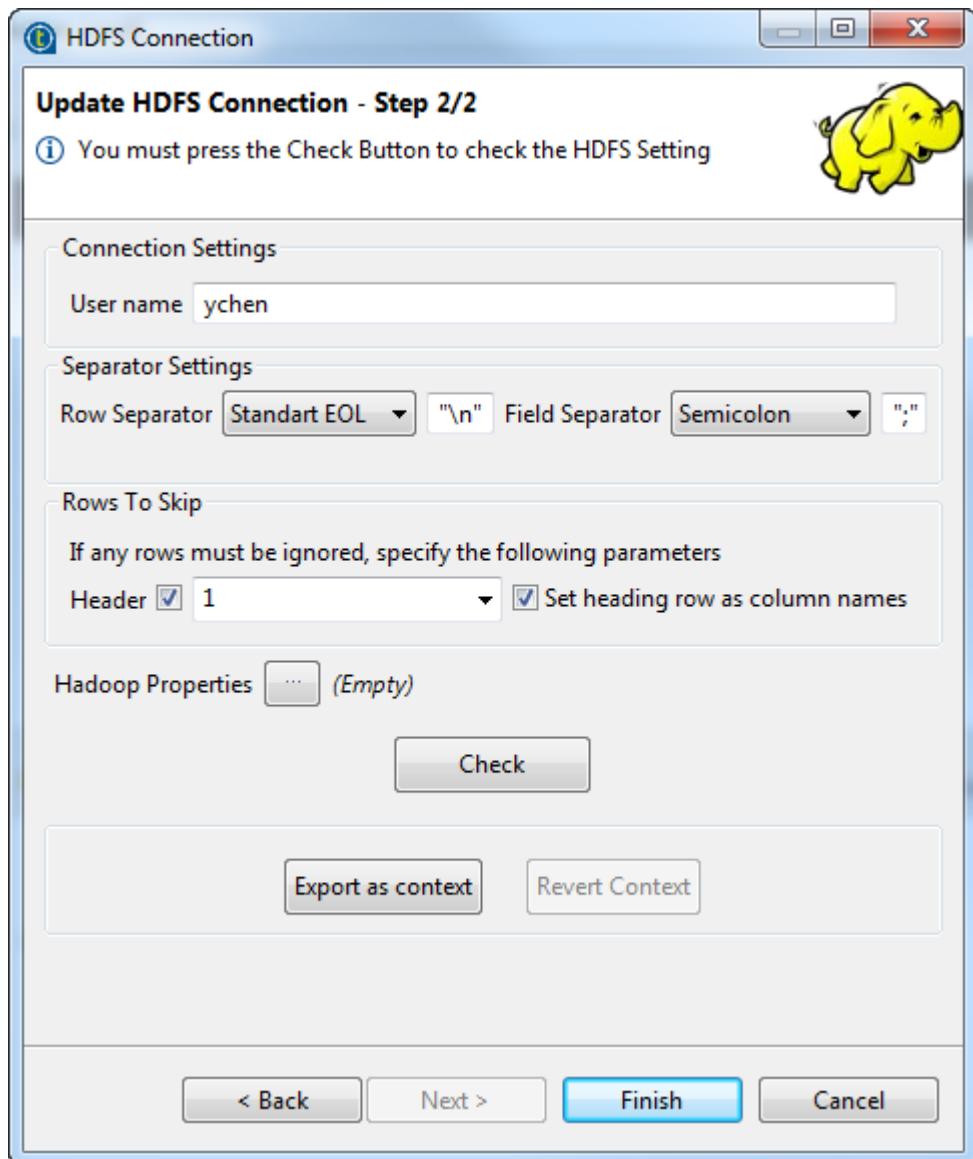
### 11.2.6.1. Créer une connexion à HDFS

1. Développez le nœud **Hadoop cluster** sous le nœud **Metadata** dans la vue **Repository** puis cliquez-droit sur la connexion Hadoop utilisée et sélectionnez **Create HDFS** dans le menu contextuel.
2. Dans l'assistant de connexion qui s'ouvre, renseignez les propriétés génériques de la connexion que vous devez créer, notamment le nom (champ **Name**), l'objectif (champ **Purpose**) et la **Description**. Le champ **Status** est un champ que vous pouvez personnaliser dans la fenêtre [File] > [Edit project properties].



3. Cliquez sur **Next** une fois terminé. L'étape suivante vous permet de renseigner les données de connexion à HDFS. Le champ **User name** est automatiquement renseigné avec une valeur héritée de la connexion Hadoop que vous avez sélectionnée dans les étapes précédentes.

Les champs **Row separator** et **Field separator** utilisent les valeurs par défaut.



Si la connexion que vous utilisez autorise la sécurité Kerberos, le champ **User name** est automatiquement désactivé.

4. Si les données auxquelles accéder dans HDFS comprennent un message d'en-tête que vous souhaitez ignorer, cochez la case **Header** et saisissez le nombre de lignes d'en-tête à ignorer.
5. Si vous devez définir des noms de colonnes pour les données auxquelles accéder, cochez la case **Set heading row as column names**. Cela permet au studio de sélectionner la dernière ligne ignorée et de l'utiliser pour nommer les colonnes dans les données.

Par exemple, cochez cette case et saisissez *1* dans le champ **Header**. Lorsque vous récupérez le schéma des données à utiliser, la première ligne est ignorée en tant que corps des données mais est utilisée pour nommer les colonnes des données.

6. Si vous devez utiliser une configuration HDFS personnalisée pour la distribution d'Hadoop à utiliser, cliquez sur le bouton [...] à côté de **Hadoop properties** pour ouvrir la table des propriétés correspondantes et ajouter la ou les propriété(s) à personnaliser. Lors de l'exécution, les propriétés personnalisées écrasent celles par défaut utilisées par le studio pour son moteur Hadoop.

Une table **Parent Hadoop properties** s'affiche au-dessus de la table des propriétés que vous modifiez. La table mère est en lecture seule et affiche les propriétés Hadoop ayant été définies dans l'assistant de la connexion Hadoop mère, sur laquelle se base la connexion à HDFS.

Pour plus d'informations concernant les propriétés relatives à HDFS de Hadoop, consultez la documentation de Apache Hadoop à l'adresse suivante : <http://hadoop.apache.org/docs/current/> (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. Par exemple, la page suivante liste certaines propriétés HDFS par défaut <http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml>.

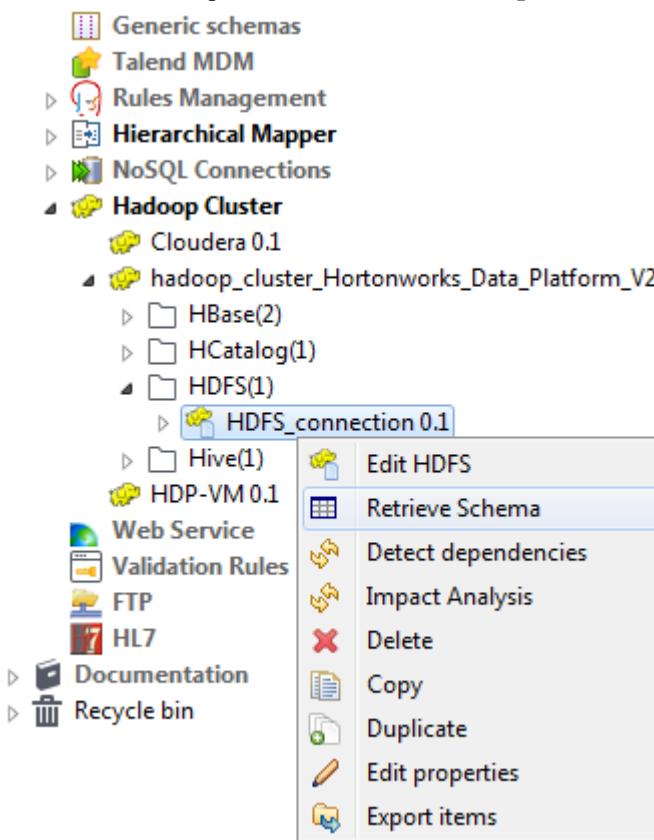
Pour tirer parti de cette table des propriétés, consultez [Configuration des propriétés réutilisables de Hadoop](#).

7. Si nécessaire, modifiez les séparateurs par défaut et cliquez sur le bouton **Check** afin de vérifier votre connexion.

Un message apparaît pour indiquer si la connexion est établie.

8. Cliquez sur **Finish** pour valider les modifications.

La nouvelle connexion HDFS créée est disponible sous le nœud **Hadoop cluster** dans la vue **Repository**.



La vue **Repository** varie en fonction de l'édition du studio que vous utilisez.

Si vous devez utiliser un contexte environnemental pour définir les paramètres de cette connexion, cliquez sur le bouton **Export as context** pour ouvrir l'assistant correspondant et choisir parmi les options suivantes :

- **Create a new repository context** : créer le contexte environnemental depuis la connexion Hadoop courante, c'est-à-dire que les paramètres à configurer dans l'assistant sont pris comme variables de contexte avec les valeurs données à ces paramètres.
- **Reuse an existing repository context** : utiliser les variables d'un contexte environnemental afin de configurer la connexion courante.

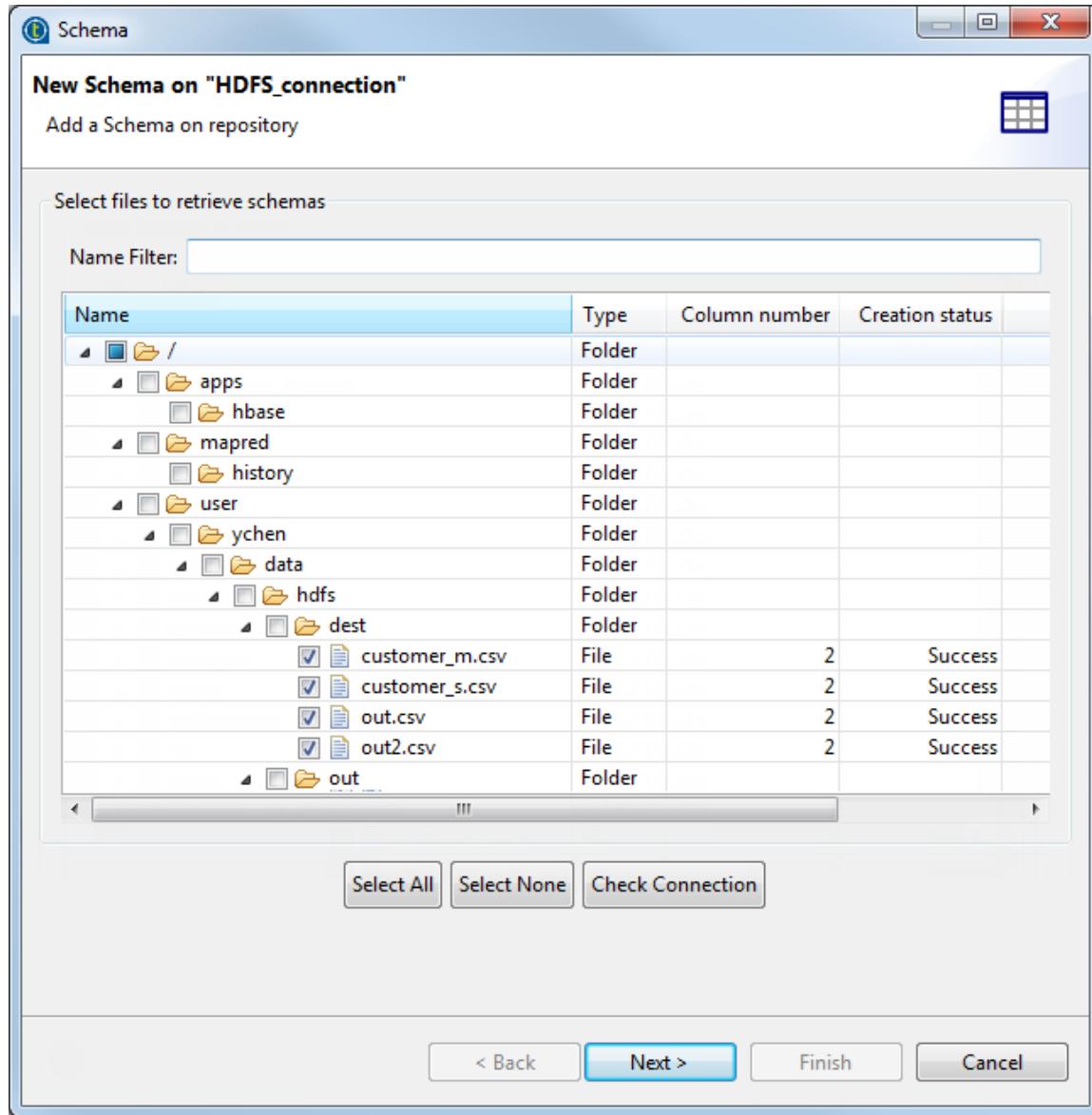
Si vous devez annuler l'implémentation du contexte, cliquez sur **Revert context**. Les valeurs des variables de contexte utilisées sont directement mises dans l'assistant.

Pour un exemple étape par étape concernant l'utilisation de la fonctionnalité **Export as context**, consultez [Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion](#).

9. Cliquez-droit sur la nouvelle connexion et sélectionnez **Retrieve schema** dans la liste déroulante afin de charger le schéma de fichier pour établir la connexion.

### 11.2.6.2. Retrouver un schéma de fichier

1. Lorsque vous cliquez sur **Retrieve Schema**, un nouvel assistant s'ouvre, dans lequel vous pouvez filtrer et afficher différents objets (un fichier Avro, par exemple) dans HDFS.

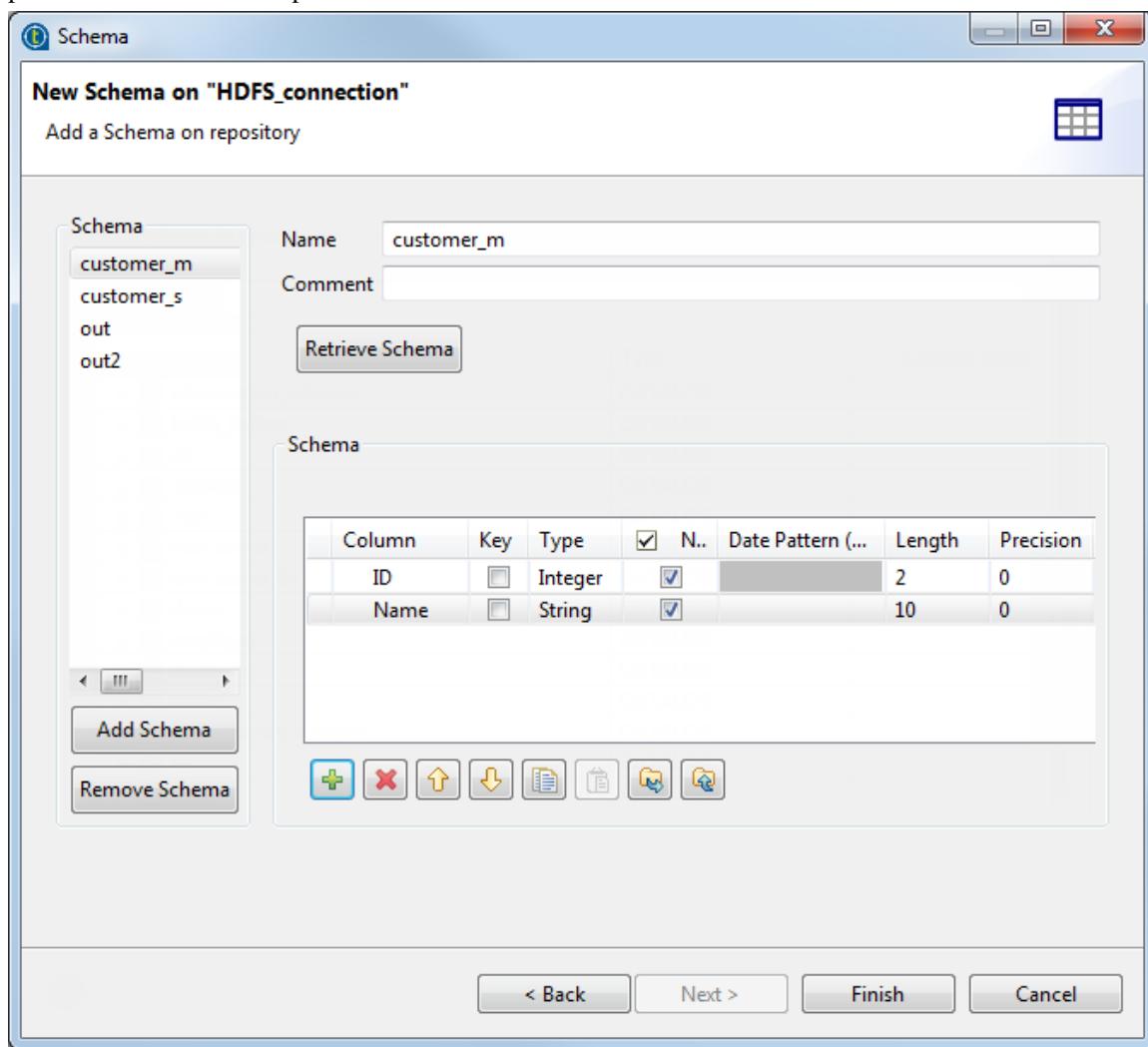


2. Dans le champ **Name filter**, vous pouvez saisir le nom du (des) fichier(s) que vous recherchez, afin d'effectuer un filtre.

Si non, vous pouvez développer les dossiers listés dans l'assistant, en cochant la case devant leur nom. Sélectionnez le(s) fichier(s) dont vous souhaitez récupérer le(s) schéma(s).

Chaque fois qu'un schéma est récupéré pour un fichier sélectionné, le statut **Creation status** du fichier devient *Success*.

3. Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'assistant qui liste le(s) schéma(s) sélectionné(s). Vous pouvez en sélectionner un pour afficher ses informations dans la zone **Schema**.



4. Modifiez, si nécessaire, le schéma sélectionné. Vous pouvez modifier son nom ou personnaliser la structure du schéma dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, de supprimer ou de déplacer des colonnes dans le schéma.

Pour écraser les modifications effectuées sur le schéma sélectionné et retrouver le schéma par défaut, cliquez sur **Retrieve schema**. Notez que cette opération écrase toutes les modifications personnalisées.

5. Cliquez sur **Finish** pour finaliser la création du schéma HDFS. Tous les schémas récupérés sont listés sous le nœud correspondant des connexions HDFS dans la vue **Repository**.

Si vous devez modifier un schéma, cliquez-droit sur ce schéma, sous le nœud des connexions HDFS dans la vue **Repository** et, dans le menu contextuel, sélectionnez **Edit Schema** pour ouvrir à nouveau l'assistant et effectuer les modifications.



Si vous modifiez les schémas, assurez-vous que le type de données est bien défini dans la colonne **Type**.

## 11.2.7. Centraliser des métadonnées Hive

Si vous avez souvent besoin d'utiliser une table de base de données provenant de Hive, vous pouvez centraliser les informations de connexion à la base de données Hive et les détails du schéma de la table dans le dossier **Metadata** du **Repository**.

Bien que vous puissiez toujours le faire à partir du mode **DB connection**, l'utilisation du nœud **Hadoop cluster** est une alternative qui permet une meilleure utilisation des propriétés de connexion centralisées pour une distribution Hadoop donnée.

### Prérequis :

- Lancez la distribution Hadoop que vous souhaitez utiliser et assurez-vous que vous disposez des permissions d'accès nécessaires pour cette distribution et sa base de données Hive.
- Créez la connexion vers cette distribution Hadoop depuis le nœud **Hadoop cluster**. Pour plus d'informations, consultez [Centraliser une connexion Hadoop](#).

Si vous souhaitez utiliser une distribution MapR et une version Hive MapR 2.0.0 ou MapR 2.1.2, procédez comme suit avant de configurer la connexion à Hive :

1. Ajoutez le chemin du client MapR vers le fichier *.ini* du *Studio Talend*, par exemple *Talend-Studio-win-x86\_64.ini*. Ce chemin peut varier selon votre système d'exploitation : pour Windows, le chemin est -Djava.library.path=maprclientpath\lib\native\Windows\_7-amd64-64).
2. Pour MapR 2.0.0, installez le module *maprfs-0.1.jar*.

Pour MapR 2.1.2, installez les modules *maprfs-0.20.2-2.1.2.jar* et *maprfs-jni-0.20.2-2.1.2.jar*.

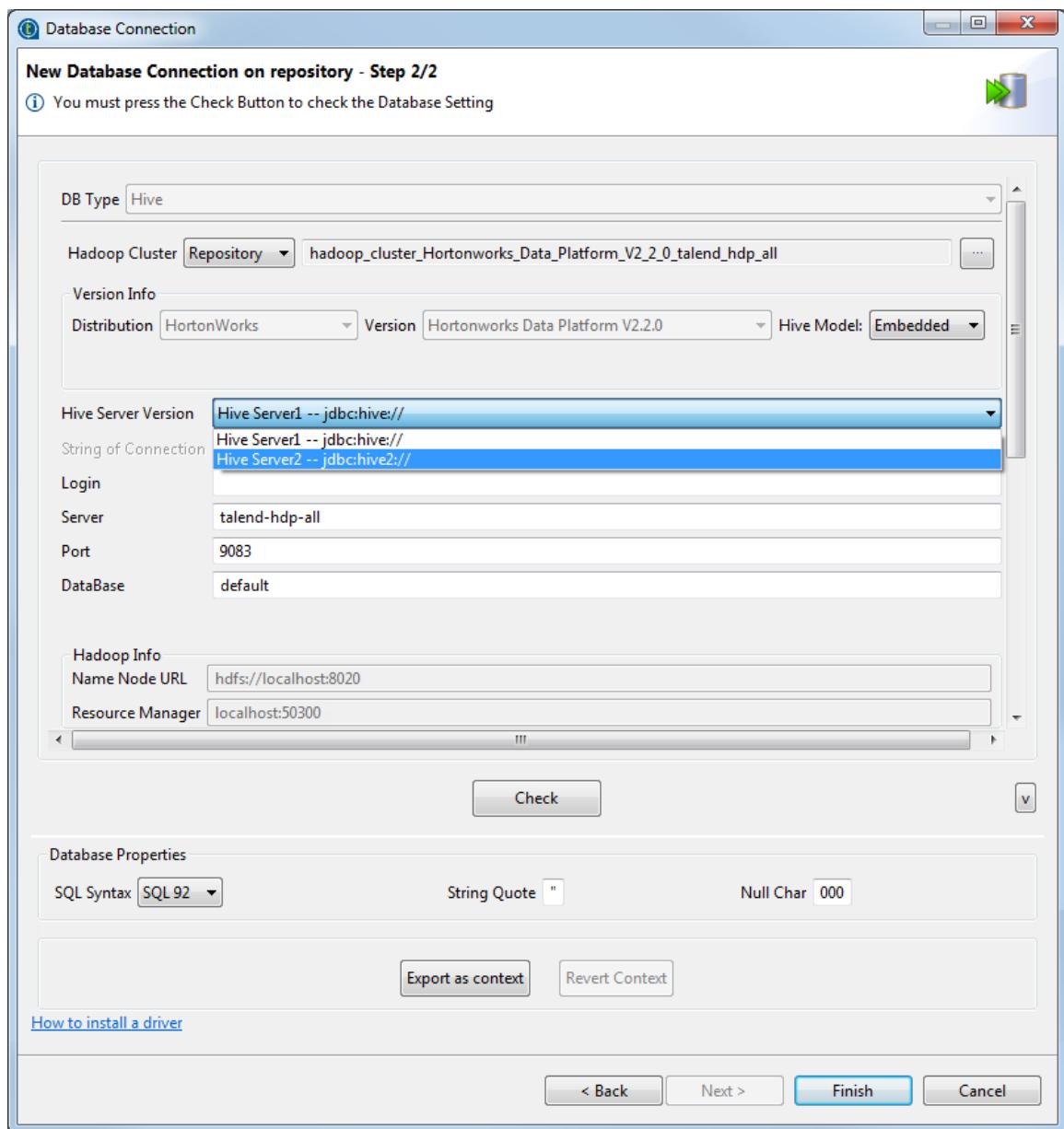
3. Redémarrez votre studio pour valider vos changements.

Pour plus d'informations sur l'installation de modules, consultez la description sur l'identification et l'installation de modules externes dans le *Guide d'installation et de migration Talend*.

### 11.2.7.1. Créer une connexion à Hive

1. Développez le nœud **Hadoop cluster** situé sous le nœud **Metadata** dans la vue **Repository**. Cliquez-droit sur la connexion Hadoop à utiliser et sélectionnez **Create Hive** dans le menu contextuel.
2. Dans l'assistant de connexion qui s'ouvre, renseignez les propriétés génériques de la connexion que vous devez créer, notamment le nom (champ **Name**), l'objectif (champ **Purpose**) et la **Description**. Le champ **Status** est un champ que vous pouvez personnaliser dans la fenêtre **[File] > [Edit project properties]**.
3. Cliquez sur **Next** pour passer à l'étape suivante et renseigner les informations de connexion à la base de données Hive. Parmi ces informations, les champs **DB Type**, **Hadoop cluster**, **Distribution**, **Version**, **Server**, **NameNode URL** et **JobTracker URL** sont automatiquement renseignés avec les propriétés héritées de la connexion Hadoop que vous avez sélectionnée dans les étapes précédentes.

Notez que si vous choisissez **None** dans la liste **Hadoop cluster**, vous basculez dans un mode manuel dans lequel les données héritées sont abandonnées. Vous devez donc configurer chaque propriété vous-même. La connexion créée apparaît sous le nœud **Db connection** uniquement.



Les propriétés à définir peuvent varier suivant la distribution Hadoop utilisée.

4. Dans la zone **Version info**, sélectionnez le modèle de la base de données de Hive auquel vous souhaitez vous connecter. Pour certaines distributions Hadoop, les deux modèles **Standard** et **Embedded** sont disponibles à l'utilisation. Pour d'autres, vous pouvez accéder uniquement au modèle **Standard** ou **Embedded**.

Selon la distribution sélectionnée, vous pouvez sélectionner **Hive Server2** dans la liste **Hive Server version**, supportant mieux les connexions simultanées de différents clients que **Hive Server1**. Pour plus d'informations concernant Hive Server2, consultez la page <https://cwiki.apache.org/confluence/display/Hive/Setting+up+HiveServer2> (en anglais).

5. Les champs affichés varient selon le modèle sélectionné.

Lorsque vous laissez le champ **Database** vide, la sélection du modèle **Embedded** permet au Studio de se connecter automatiquement à toutes les bases de données existantes dans Hive. A l'inverse, la sélection du modèle **Standalone** permet de se connecter uniquement à la base de données *default* de Hive.

String of Connection	<code>jdbc:hive://</code>
Login	root
Password	*****
Server	talend-hdp-all
Port	8094
DataBase	

A part la propriété **Server** qui est déjà pré-remplie avec une valeur héritée, vous devez renseigner les propriétés restantes requises par cette connexion Hive.

- Si vous accédez à une distribution de Hadoop fonctionnant avec la sécurité Kerberos, cochez la case **Use Kerberos authentication**. Renseignez les champs selon les configurations du serveur Hive :

- Saisissez le nom du Principal Kerberos dans le champ **Hive principal**,
- Saisissez l'URL de la base de données du Metastore dans le champ **Metastore URL**,
- Cliquez sur le bouton [...] à côté du champ **Driver jar** et parcourez votre système jusqu'au fichier Jar du pilote de la base de données du Metastore,
- Cliquez sur le bouton [...] à côté du champ **Driver class** et sélectionnez la classe, puis
- Saisissez votre identifiant et votre mot de passe dans les champs **Username** et **Password** respectivement.

Si vous devez utiliser un fichier Keytab pour vous connecter, cochez la case **Use a keytab to authenticate**, saisissez le Principal à utiliser, dans le champ **Principal**, puis, dans le champ **Keytab**, parcourez votre système jusqu'au fichier Keytab à utiliser.

Un fichier Keytab contient des paires de Principaux et de clés cryptées Kerberos. Notez que l'utilisateur exécutant un Job contenant un Keytab n'est pas nécessairement celui désigné par le Principal mais doit posséder les droits de lire le fichier Keytab utilisé. Par exemple, l'identifiant utilisé pour exécuter un Job est *user1* et le Principal est *guest*. Dans ce cas, assurez-vous que l'utilisateur *user1* possède le droit de lire le fichier Keytab à utiliser.

<b>Authentication</b>	
<input checked="" type="checkbox"/> Use Kerberos authentication	
Hive principal	hive/_HOST@EXAMPLE.COM
Metastore URL	<code>jdbc:mysql://condor-secured-server/hive?createDatabaseIfNotExist=true</code>
Driver jar	E:\\DBLib\\mysql-connector-java-5.1.0-bin.jar
Driver class	com.mysql.jdbc.Driver
Username	hive
Password	****
<input checked="" type="checkbox"/> Use a keytab to authenticate	
principal	guest
Keytab	D:/test/kerberos/guest.keytab
<input type="button" value="Browse..."/>	

- Si vous utilisez l'une des distributions suivantes, sélectionnez **Tez** comme framework de vos Jobs dans la liste **Execution engine**.

- Hortonworks : V2.1 et V2.2.
- MapR : V4.0.1.

- **Custom** : cette option vous permet de vous connecter à une distribution supportant Tez mais non officiellement supportée par *Talend*.

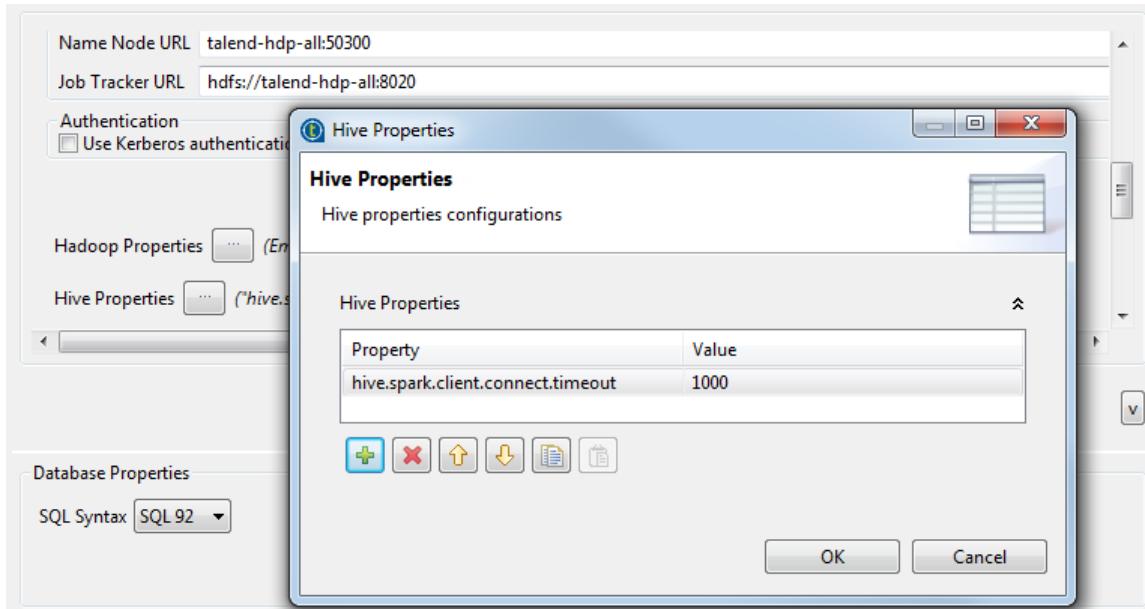
Lorsque vous réutilisez cette connexion dans un composant Hive, vous devez configurer l'accès aux bibliothèques de Tez via la vue **Advanced settings** du composant.

8. Si vous devez utiliser une configuration personnalisée pour la distribution d'Hadoop ou de Hive à utiliser, cliquez sur le bouton [...] à côté du champ **Hadoop properties** ou **Hive Properties**, pour ouvrir la table des propriétés correspondante et ajouter une ou des propriété(s) à personnaliser. Lors de l'exécution, les propriétés personnalisées écrasent celles par défaut utilisées par le studio pour son moteur Hadoop.

Pour plus d'informations concernant les propriétés de Hadoop, consultez la documentation de Apache Hadoop à l'adresse suivante : <http://hadoop.apache.org/docs/current/> (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. Par exemple, la page suivante liste certaines des propriétés par défaut de Hadoop : <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/core-default.xml>.

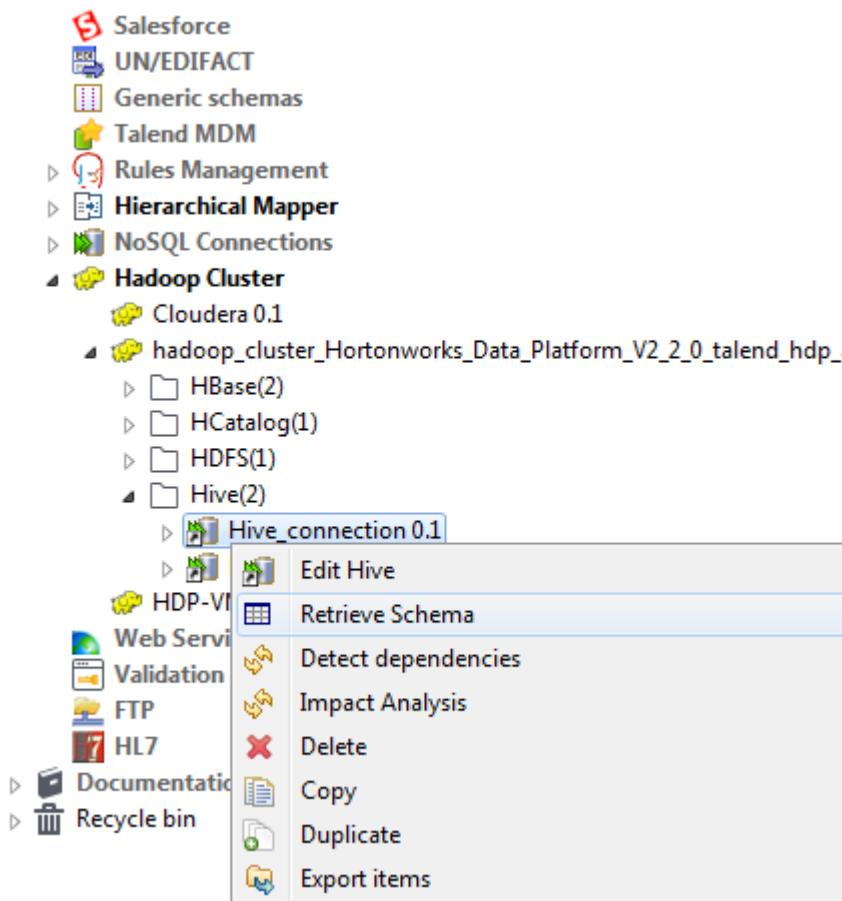
Pour plus d'informations concernant les propriétés de Hive, consultez la documentation de Apache Hive. Par exemple, la page suivante décrit certaines propriétés de configuration de Hive : <https://cwiki.apache.org/confluence/display/Hive/Configuration+Properties>.

Pour tirer parti de cette table des propriétés, consultez [Configuration des propriétés réutilisables de Hadoop](#).



9. Cliquez sur le bouton **Check** afin de vérifier si votre connexion est établie.
10. Si besoin, définissez les propriétés de la base de données, dans les champs respectifs de la zone **Database Properties**.
11. Cliquez sur **Finish** pour valider vos changements et fermer l'assistant.

La nouvelle connexion spécifiée à la base de données Hive s'affiche sous le nœud **DB Connections** de la vue **Repository**. Cette connexion contient quatre sous-dossiers parmi lesquels **Table schema** peut regrouper tous les schémas relatifs à cette connexion.



Si vous devez utiliser un contexte environnemental pour définir les paramètres de cette connexion, cliquez sur le bouton **Export as context** pour ouvrir l'assistant correspondant et choisir parmi les options suivantes :

- **Create a new repository context** : créer le contexte environnemental depuis la connexion Hadoop courante, c'est-à-dire que les paramètres à configurer dans l'assistant sont pris comme variables de contexte avec les valeurs données à ces paramètres.
- **Reuse an existing repository context** : utiliser les variables d'un contexte environnemental afin de configurer la connexion courante.

Si vous devez annuler l'implémentation du contexte, cliquez sur **Revert context**. Les valeurs des variables de contexte utilisées sont directement mises dans l'assistant.

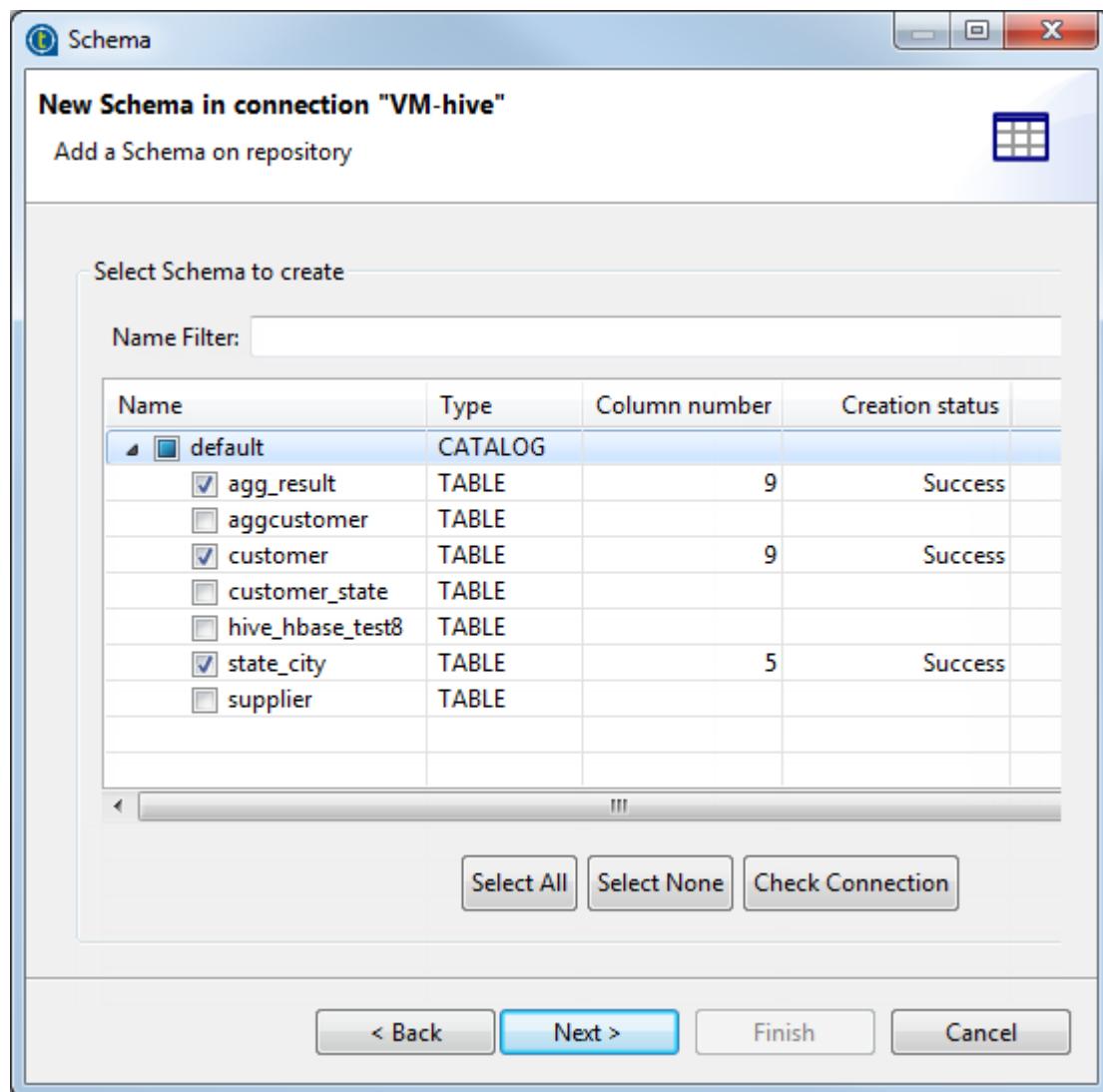
Pour un exemple étape par étape concernant l'utilisation de la fonctionnalité **Export as context**, consultez [Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion](#).

12. Cliquez-droit sur la connexion Hive que vous avez créée et sélectionnez **Retrieve Schema** pour extraire tous les schémas de la base de données Hive définie.

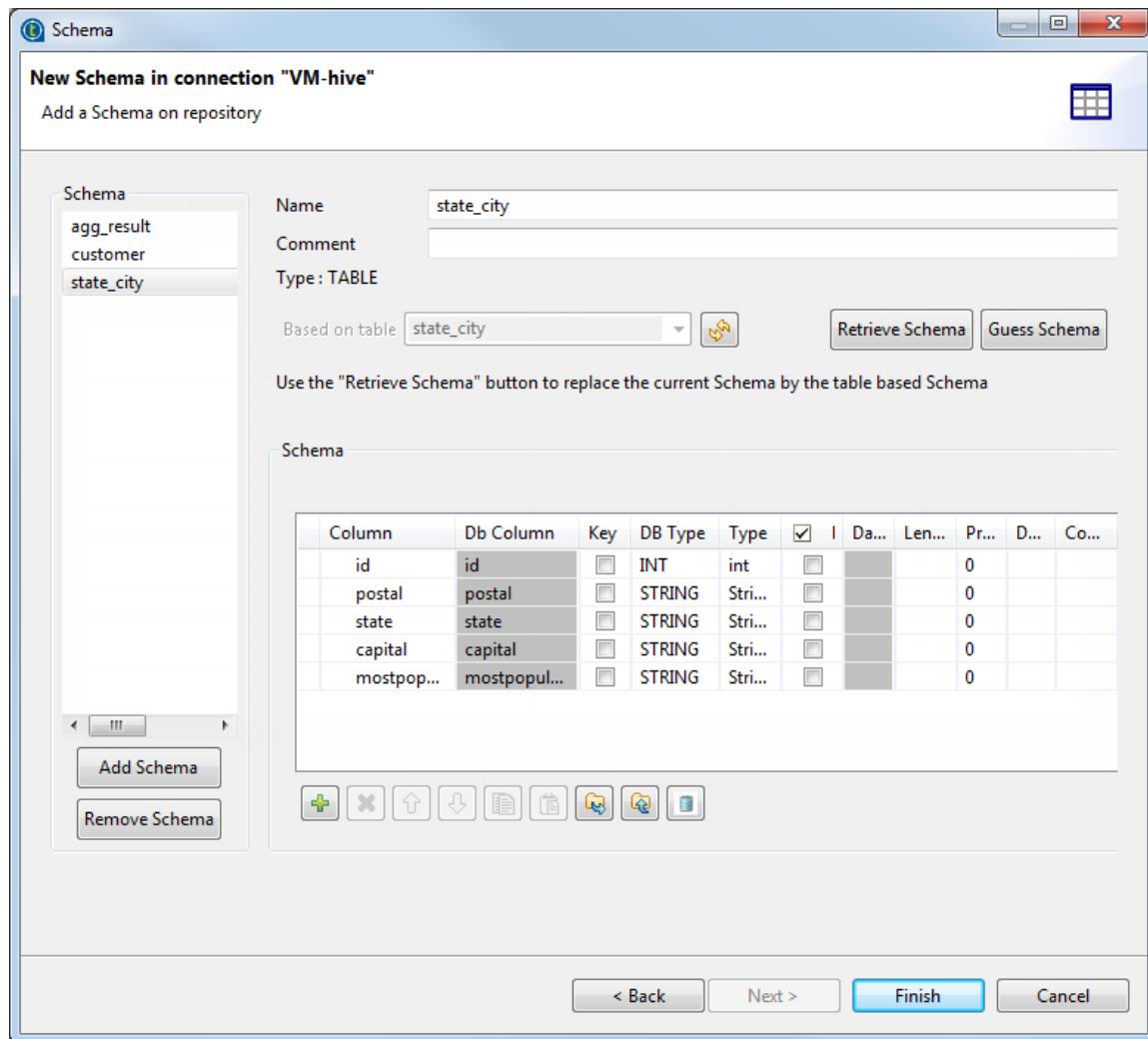
### 11.2.7.2. Récupérer le schéma d'une table Hive

Dans cette étape, vous allez récupérer le schéma de la table de la base de données Hive connectée.

1. Dans la vue **Repository**, cliquez-droit sur la connexion Hive utilisée et sélectionnez **Retrieve schema** dans le menu contextuel. Cliquez ensuite sur **Next** dans l'assistant qui s'ouvre afin de voir et filtrer les différentes tables de cette base de données Hive.



2. Développez le nœud des tables que vous souhaitez utiliser et sélectionnez les colonnes que vous souhaitez récupérer. Cliquez sur **Next** pour ouvrir une nouvelle vue dans l'assistant, listant le(s) schéma(s) de la table sélectionnée. Vous pouvez sélectionner n'importe quel schéma pour afficher ses détails dans la zone **Schema**, du côté droit de l'assistant.



*Si la table de la base de données source contient une valeur par défaut étant une fonction ou une expression et non une chaîne de caractères, assurez-vous de supprimer les guillemets simples entourant la valeur par défaut du schéma final, s'il y en a, afin d'éviter des résultats inattendus lors de la création de tables de bases de données à l'aide de ce schéma.*

*Pour plus d'informations, consultez Vérification des valeurs par défaut dans un schéma récupéré à l'adresse <https://help.talend.com>.*

3. Modifiez le schéma, si nécessaire. Vous pouvez renommer le schéma et personnaliser sa structure selon vos besoins, dans la zone **Schema**.

La barre d'outils vous permet d'ajouter, supprimer ou déplacer des colonnes dans votre schéma.

Pour écraser les modifications effectuées sur le schéma sélectionné avec le schéma par défaut, cliquez sur **Retrieve schema**. Notez que toutes les modifications apportées au schéma seront perdues si vous cliquez sur ce bouton.

4. Cliquez sur **Finish** pour terminer la récupération du schéma de la table Hive. Tous les schémas récupérés sont affichés sous le nœud **Hive** correspondant dans la vue **Repository**.

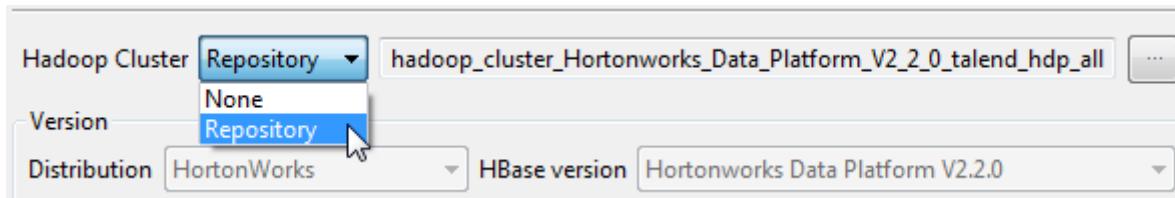
Si vous devez éditer ultérieurement le schéma, cliquez-droit sur le schéma et sélectionnez **Edit Schema** dans le menu contextuel pour ouvrir à nouveau l'assistant et effectuer vos modifications.



*Si vous modifiez le schéma, assurez-vous que le Type des données est correctement défini.*

Comme expliqué précédemment, en plus d'utiliser le nœud **Hadoop cluster**, vous pouvez également partir du nœud **Db connection** afin de créer une connexion Hive et récupérer des schémas. Dans tous les cas, vous devez configurer les propriétés spécifiques de connexion à Hive. A cette étape :

- si vous sélectionnez l'option **Repository** dans la liste **Hadoop cluster** pour réutiliser les détails d'une connexion Hadoop établie, la connexion Hive créée sera classée à la fois sous le nœud **Hadoop cluster** et le nœud **Db connection** ;
- si vous sélectionnez l'option **None** dans la liste **Hadoop cluster** afin de renseigner vous-même les propriétés de la connexion Hadoop, la connexion Hive créée apparaît uniquement sous le nœud **Db connection**.



## 11.2.8. Centraliser une connexion Oozie

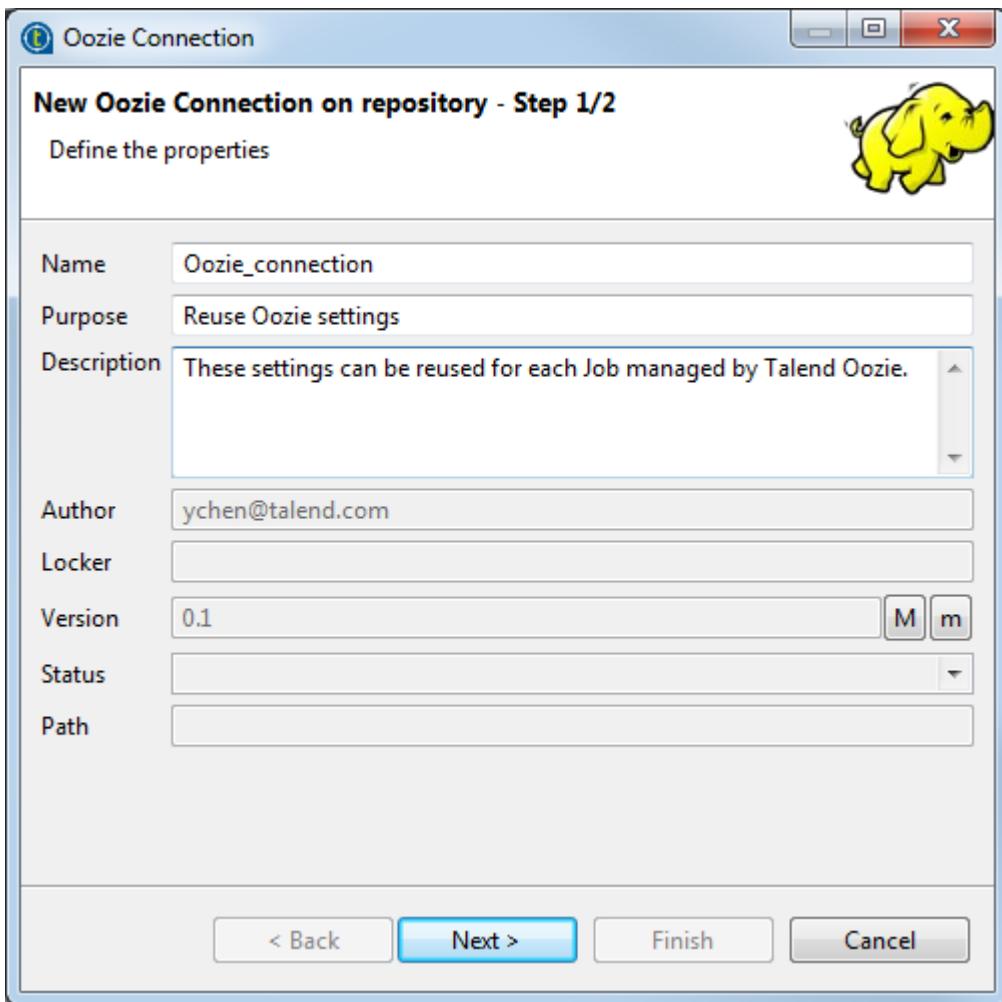
Si vous avez souvent besoin d'utiliser **Oozie scheduler** (l'ordonnanceur Oozie) afin de lancer et de moniter des Jobs dans Hadoop, vous pouvez également centraliser les paramètres d'Oozie dans le dossier **Metadata** du **Repository**.

### Prérequis :

- Lancez la distribution Hadoop que vous souhaitez utiliser et assurez-vous que vous disposez des permissions d'accès nécessaires pour cette distribution et sa connexion Oozie.
- Créez la connexion vers cette distribution Hadoop depuis le nœud **Hadoop cluster**. Pour plus d'informations, consultez [Centraliser une connexion Hadoop](#).

Le composant **Oozie scheduler** est utilisé pour planifier les exécutions d'un Job, déployer ou exécuter des Jobs dans HDFS et moniter les exécutions. Afin de créer une connexion Oozie, procédez comme suit :

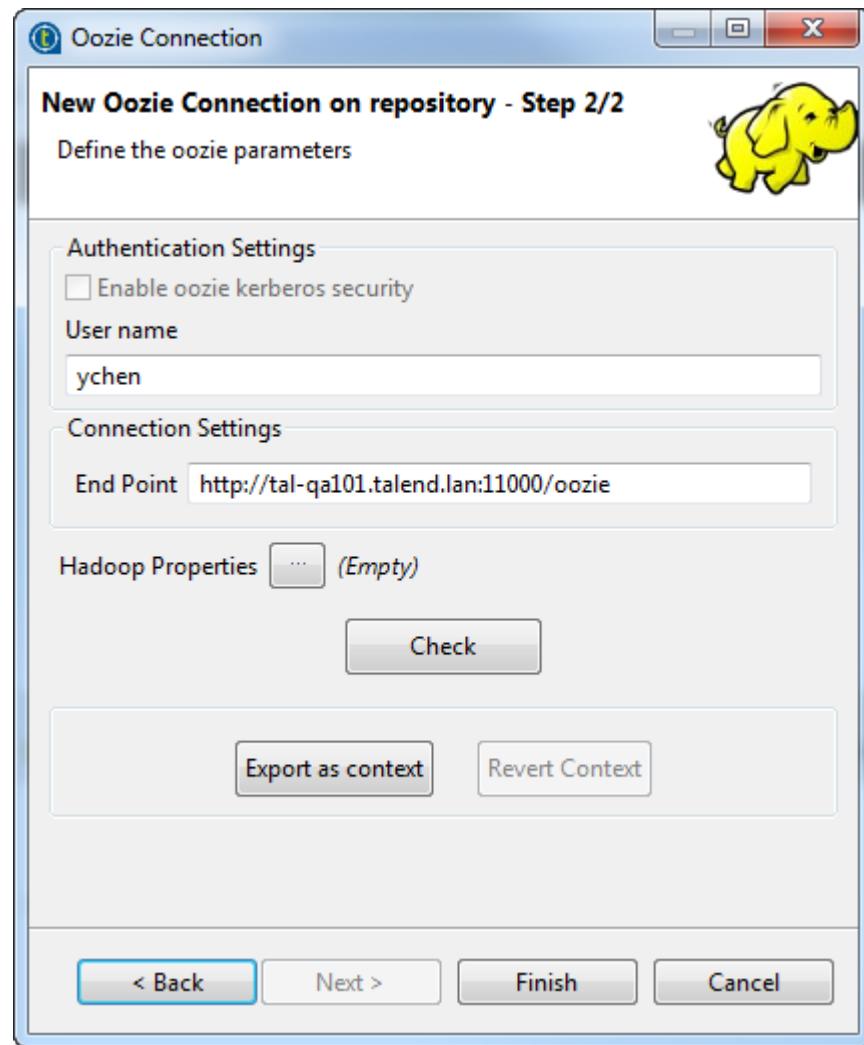
1. Développez le nœud **Hadoop cluster** situé sous le nœud **Metadata** dans la vue **Repository**. Cliquez-droit sur la connexion Hadoop à utiliser et sélectionnez **Create Oozie** dans le menu contextuel.
2. Dans l'assistant de connexion qui s'ouvre, renseignez les propriétés génériques de la connexion que vous devez créer, notamment le nom (champ **Name**), l'objectif (champ **Purpose**) et la **Description**. Le champ **Status** est un champ que vous pouvez personnaliser dans la fenêtre **[File] > [Edit project properties]**.



- Une fois les champs remplis, cliquez sur **Next** pour passer à l'étape suivante, puis renseignez les informations de connexion à Oozie. Dans le champ **End Point**, l'URL de l'application Web est automatiquement construite avec le nom d'hôte du NameNode de la connexion Hadoop que vous utilisez et le port par défaut d'Oozie. Cette application Web vous permet également de consulter le statut des exécutions de Job planifiées dans la console **Oozie Web Console** dans votre navigateur Web.

Si la distribution Hadoop que vous sélectionnez autorise la sécurité Kerberos, le champ **User name** est désactivé.

Vous pouvez modifier l'URL d'Oozie si nécessaire.



4. Si vous devez utiliser une configuration personnalisée pour votre distribution d'Hadoop, cliquez sur le bouton [...] à côté de **Hadoop properties** pour ouvrir la table des propriétés et ajouter la ou les propriété(s) à personnaliser. Lors de l'exécution, les propriétés personnalisées écrasent celles par défaut utilisées par le studio pour son moteur Hadoop.

Une table **Parent Hadoop properties** s'affiche au-dessus de la table des propriétés que vous modifiez. La table mère est en lecture seule et affiche les propriétés Hadoop ayant été définies dans l'assistant de la connexion Hadoop mère, sur laquelle se base la connexion à Oozie.

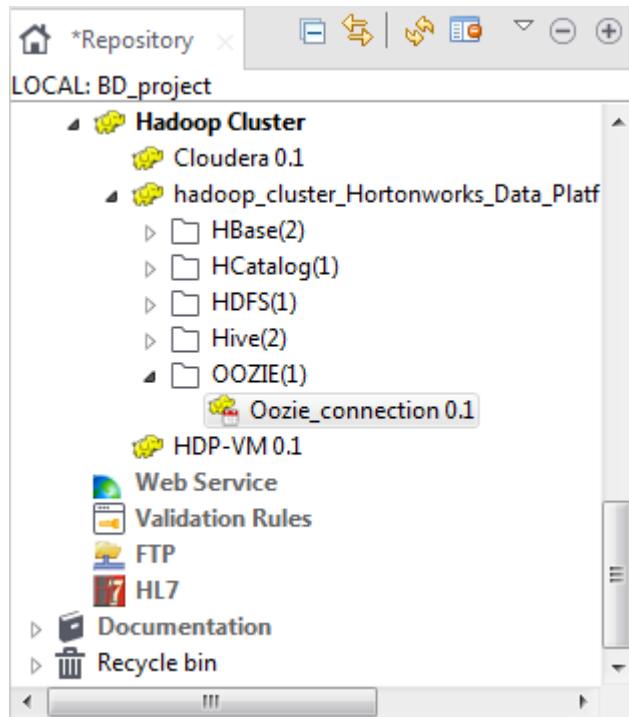
Pour plus d'informations concernant les propriétés Oozie de Hadoop, consultez la documentation de Apache Hadoop relative à Oozie à l'adresse suivante : <http://oozie.apache.org> (en anglais) ou la documentation de la distribution Hadoop que vous utilisez. Par exemple, la page suivante liste certaines propriétés Hadoop : [https://oozie.apache.org/docs/4.1.0/AG\\_HadoopConfiguration.html](https://oozie.apache.org/docs/4.1.0/AG_HadoopConfiguration.html) (en anglais).

5. Dans le champ **User name**, saisissez votre nom d'utilisateur Oozie ou laissez ce champ vide pour utiliser un accès anonyme. Dans ce cas, le nom de la machine client est utilisé.
6. Cliquez sur **Check** afin de vérifier si votre connexion est établie.

Un message apparaît indiquant que la connexion a bien été effectuée.

7. Cliquez sur **Finish** afin de valider ces changements.

La connexion Oozie que vous avez créée apparaît sous le noeud **Hadoop cluster** dans le **Repository**.



Cette vue **Repository** peut varier suivant l'édition du studio que vous utilisez.

Lorsque vous configurez l'ordonnanceur Oozie pour un Job dans la vue **Oozie scheduler**, vous pouvez réutiliser les paramètres centralisés d'Oozie.

Pour plus d'informations sur la manière d'utiliser **Oozie scheduler** pour un Job, consultez [Exécution d'un Job via Oozie](#).

Si vous devez utiliser un contexte environnemental pour définir les paramètres de cette connexion, cliquez sur le bouton **Export as context** pour ouvrir l'assistant correspondant et choisir parmi les options suivantes :

- **Create a new repository context** : créer le contexte environnemental depuis la connexion Hadoop courante, c'est-à-dire que les paramètres à configurer dans l'assistant sont pris comme variables de contexte avec les valeurs données à ces paramètres.
- **Reuse an existing repository context** : utiliser les variables d'un contexte environnemental afin de configurer la connexion courante.

Si vous devez annuler l'implémentation du contexte, cliquez sur **Revert context**. Les valeurs des variables de contexte utilisées sont directement mises dans l'assistant.

Pour un exemple étape par étape concernant l'utilisation de la fonctionnalité **Export as context**, consultez [Exporter une métadonnée en tant que contexte et réutiliser ses paramètres pour configurer une connexion](#).

## 11.2.9. Configuration des propriétés réutilisables de Hadoop

Lorsque vous configurez une connexion Hadoop, vous pouvez définir un ensemble de propriétés Hadoop communes qui seront réutilisées par les connexions filles à chaque élément Hadoop individuel, comme Hive, HDFS ou HBase.

Par exemple, dans le cluster Hadoop à utiliser, vous avez configuré la fonctionnalité HDFS High Availability (Haute disponibilité) dans le fichier *hdfs-site.xml* du cluster même. Vous devez paramétriser les propriétés correspondantes dans l'assistant de connexion, afin d'activer la fonctionnalité de haute disponibilité dans le Studio. Notez que ces propriétés peuvent également être configurées dans un composant Hadoop spécifique. Le processus pour cela est expliqué dans l'article concernant l'activation de la Haute Disponibilité HDFS sur Talend Help Center (<https://help.talend.com>). Dans cette section, seule la méthode avec l'assistant est présentée.

### Prérequis :

- Lancez la distribution Hadoop à utiliser et assurez-vous d'avoir les droits d'accès à cette distribution et son Oozie.
- Les propriétés de la haute disponibilité à configurer dans le Studio sont définies dans le fichier *hdfs-site.xml* du cluster à utiliser.

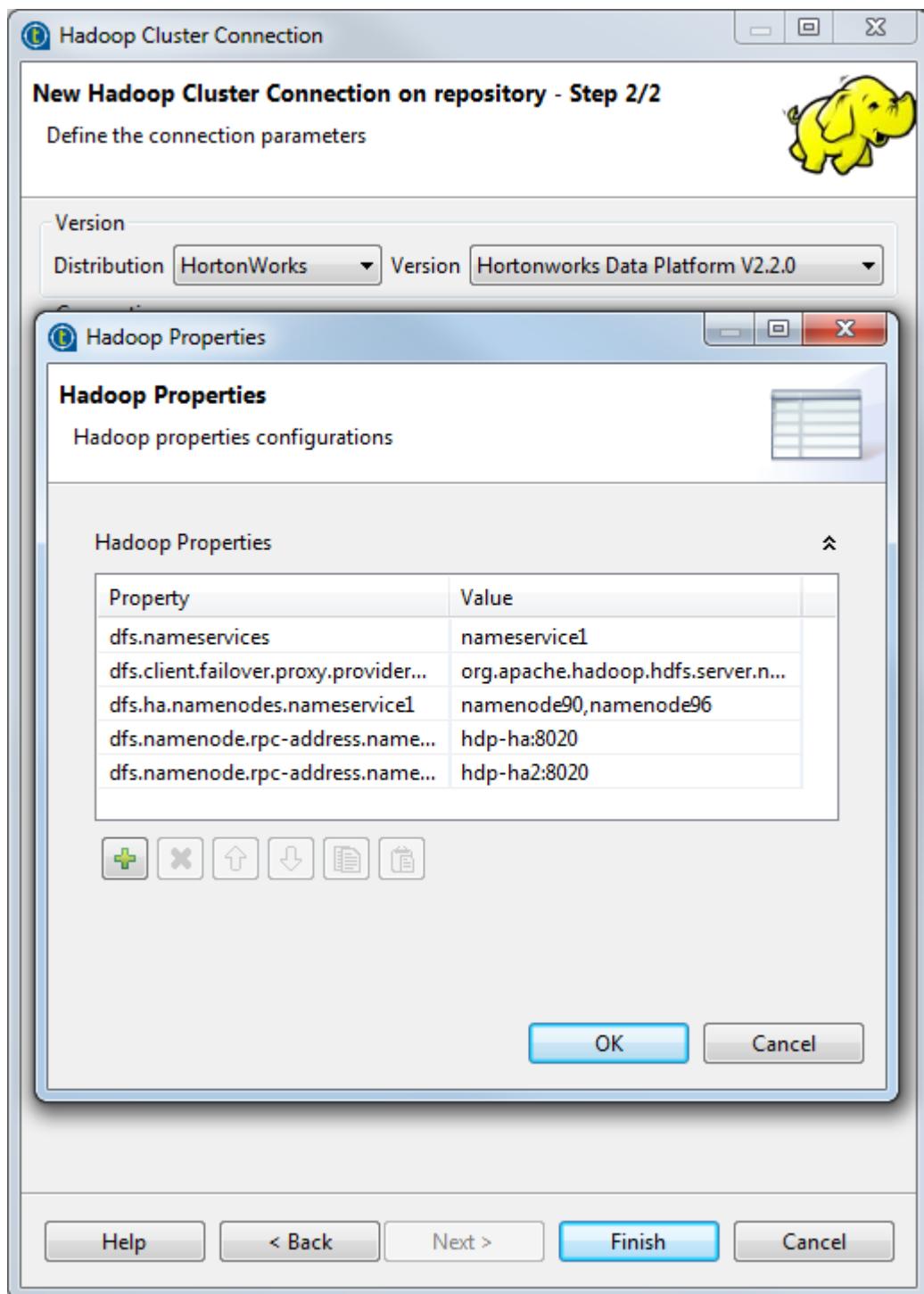
Dans cet exemple, les propriétés de la haute disponibilité se présentent comme suit :

```
<property>
 <name>dfs.nameservices</name>
 <value>nameservice1</value>
</property>
<property>
 <name>dfs.client.failover.proxy.provider.nameservice1</name>
 <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvider</value>
</property>
<property>
 <name>dfs.ha.namenodes.nameservice1</name>
 <value>namenode90,namenode96</value>
</property>
<property>
 <name>dfs.namenode.rpc-address.nameservice1.namenode90</name>
 <value>hdp-ha:8020</value>
</property>
<property>
 <name>dfs.namenode.rpc-address.nameservice1.namenode96</name>
 <value>hdp-ha2:8020</value>
</property>
```

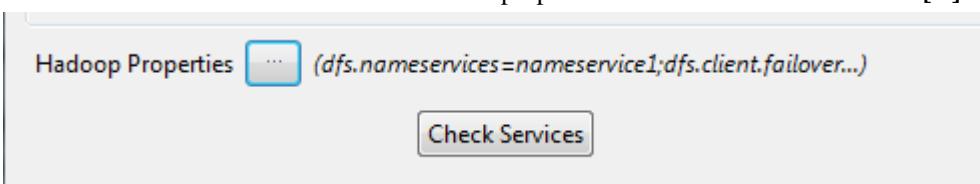
Les valeurs des propriétés servent uniquement à la démonstration.

Pour configurer ces propriétés dans la connexion Hadoop, ouvrez l'assistant [**Hadoop Cluster Connection**] depuis le nœud **Hadoop cluster** du **Repository**. Pour plus d'informations concernant l'accès à cet assistant, consultez [Centraliser une connexion Hadoop](#).

1. Si ce n'est pas encore fait, configurez la connexion au cluster Hadoop à utiliser, comme expliqué dans les sections précédentes.
2. Cliquez sur le [...] à côté de **Hadoop properties**, pour ouvrir la table **Hadoop properties**.



3. Ajoutez les propriétés de haute disponibilité mentionnées ci-dessus dans la table.
4. Cliquez sur **OK** afin de valider les modifications. Ces propriétés sont listées à côté du bouton [...].

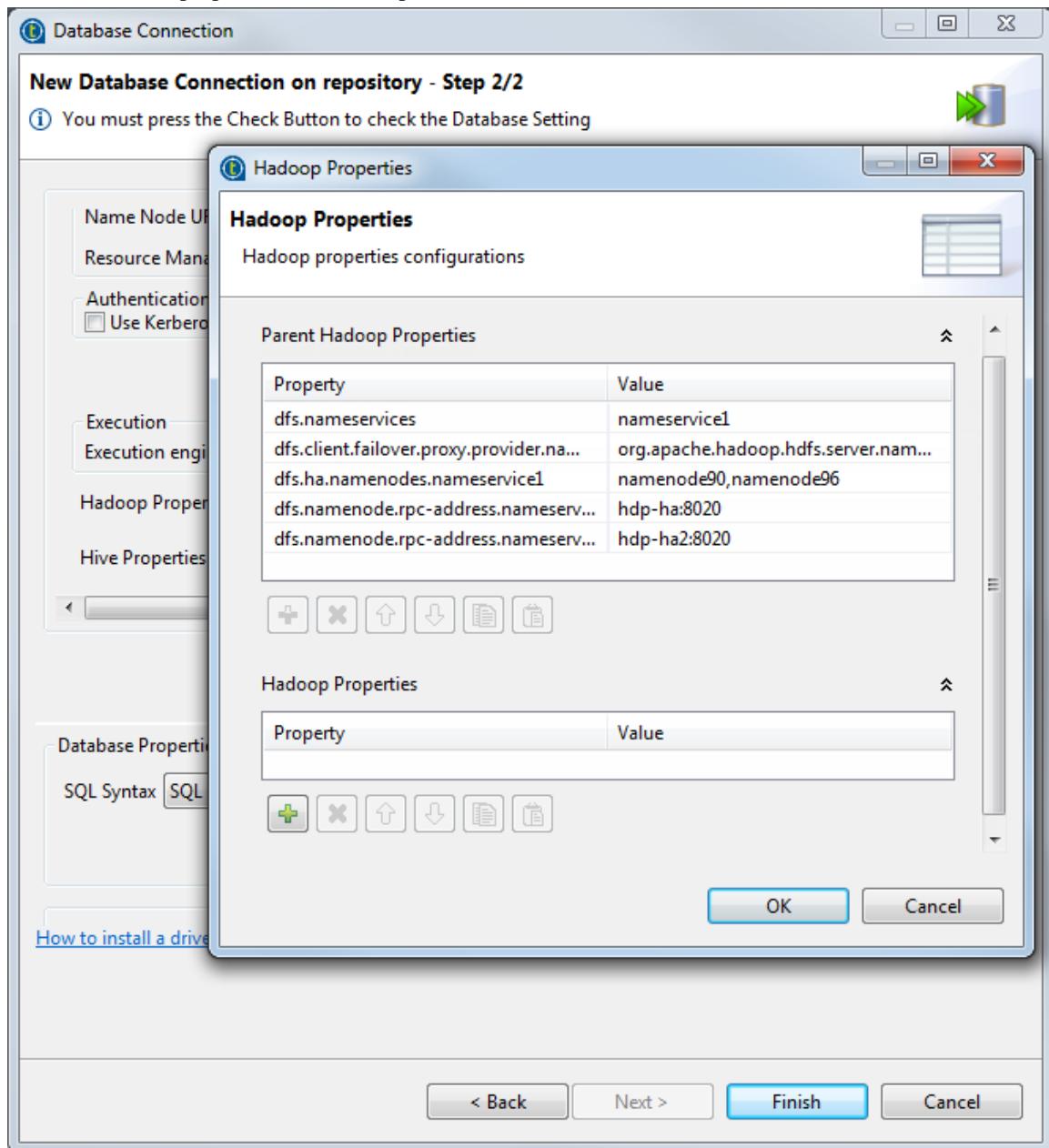


5. Cliquez sur le bouton **Check services** afin de vérifier la connexion.

Une boîte de dialogue s'ouvre, affichant la vérification et le statut de la connexion. S'il est indiqué que la connexion a échoué, vous devez vérifier et mettre à jour les informations de connexion définies dans l'assistant.

- Cliquez sur **Finish** afin de valider la connexion.

Lorsque vous créez une connexion fille, par exemple à Hive, à partir de cette connexion Hadoop, la connexion fille hérite de ces propriétés de haute disponibilité, en lecture seule.



Ces propriétés peuvent donc être réutilisées automatiquement par toute connexion fille de la connexion Hadoop.

L'image ci-dessus présente les propriétés héritées dans l'assistant de connexion Hive. Pour plus d'informations concernant l'accès à l'assistant de connexion Hive présenté dans cette section, consultez [Centraliser des métadonnées Hive](#).





## Chapitre 12. 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 [\*Routines système\*](#).

## 12.1. 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 [Accès aux routines système](#).

Pour plus d'informations concernant la création des routines utilisateur, consultez [Créer des routines personnalisées](#).



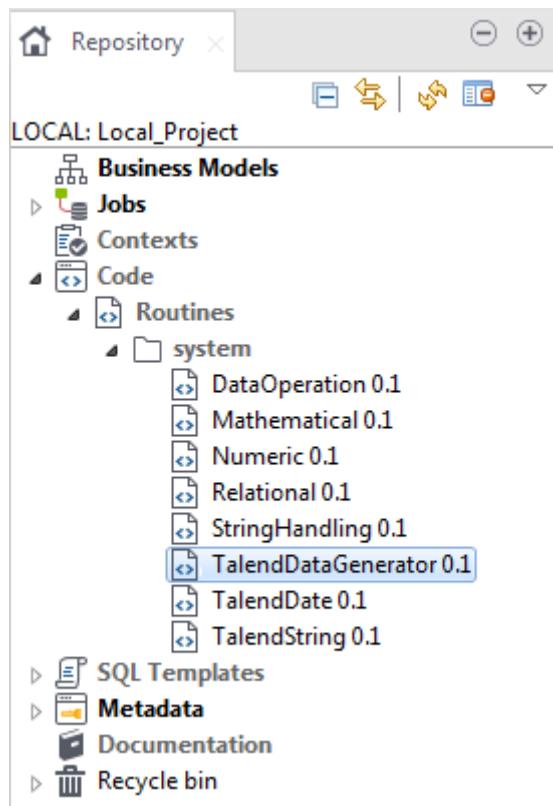
Vous pouvez également définir les dépendances des routines dans des Jobs. Pour ce faire, cliquez-droit sur un Job dans la vue **Repository** et sélectionnez **Set up routine dependencies**. Dans la boîte de dialogue qui s'ouvre, toutes les routines sont définies par défaut. Vous pouvez utiliser la barre d'outils afin de supprimer les routines, si nécessaire.

## 12.2. 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.

1. Appuyez sur **Ctrl + O** dans la routine ouverte.

Une boîte de dialogue affiche une liste des différentes routines de la catégorie.

2. Cliquez sur la routine souhaitée.

La vue bascule vers la section comprenant le texte descriptif de la routine et le code correspondant.



La syntaxe d'appel des routines est sensible à la casse.

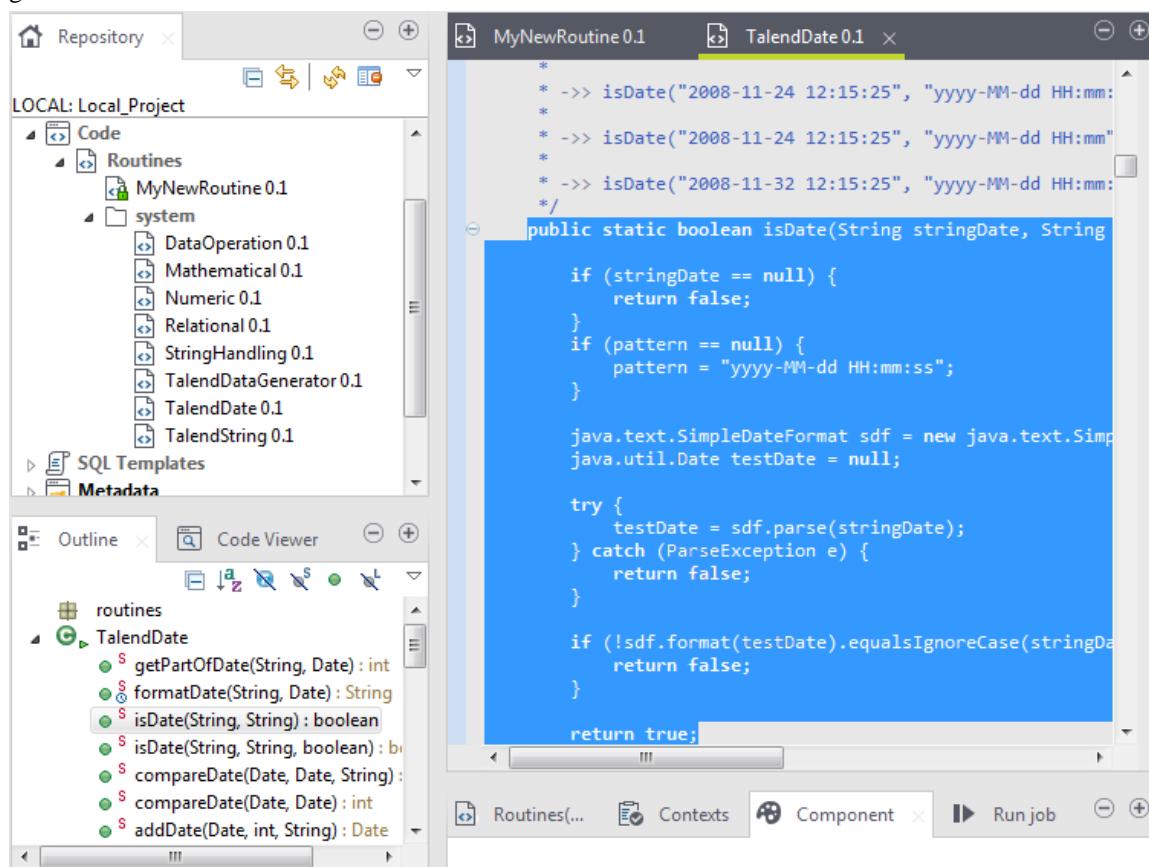
## 12.3. Personnalisation des routines système

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 :

1. Créez d'abord une routine utilisateur en suivant les étapes décrites dans la [Créer des routines personnalisées](#). Cette routine s'ouvre dans l'espace de travail et contient l'exemple de base de routine.
2. Puis dans **Code > Routines > system**, sélectionnez une classe de routines système qui contient la ou les routine(s) que vous souhaitez personnaliser.
3. Double-cliquez sur la classe contenant la routine pour l'ouvrir dans l'espace de travail.

4. Retrouvez la routine/fonction dont vous souhaitez copier tout ou partie, à l'aide du panneau **Outline** sur la gauche du studio.



5. Dans l'espace de travail, sélectionnez tout ou partie du code, et copiez-le via **Ctrl+C**  
 6. Cliquez sur l'onglet correspondant à votre routine utilisateur et collez le code du presse-papier (**Ctrl+V**).  
 7. Adaptez ensuite le code à vos besoins, et appuyez sur **Ctrl+S** afin de l'enregistrer.

Il est recommandé d'utiliser la partie commentaire (en bleu) pour détailler les paramètres d'entrée et de sortie de votre routine et ainsi en faciliter la maintenance et la réutilisation.

## 12.4. Gestion des routines personnalisées

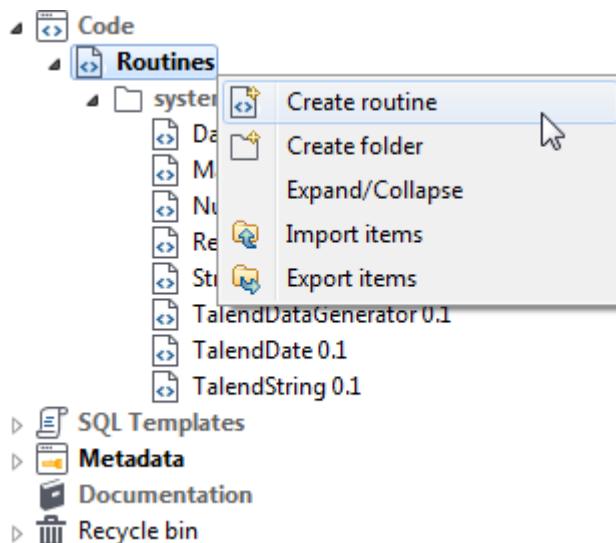
Le *Studio Talend* vous offre la possibilité de créer des routines personnalisées, de les modifier, ou de modifier des routines existantes, afin de répondre à vos besoins spécifiques.

### 12.4.1. 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 :

1. Dans la vue **Repository**, cliquez sur **Code** pour accéder aux **Routines**,



2. Puis cliquez-droit sur **Routines** et sélectionnez **Create Routine**.
3. L'assistant **[New routine]** s'ouvre. Saisissez les informations nécessaires à la création de la routine, telles que son nom, sa description...
4. Cliquez sur **Finish** pour valider la création et accéder à l'éditeur.

```

/*
 * 4. {param} 's format is: {param} <type>[(<default value or closed list>)
 *
 * <type> 's value should be one of: string, int, list, double, object, boolean
 * Function's parameter name. the {param} is optional. so if you the Function
 * added. you can have many parameters for the Function.
 *
 * 5. {example} gives a example for the Function. it is optional.
 */
public class MyNewRoutine {
 /**
 * helloExample: not return value, only print "hello" + message.
 *
 * {talendTypes} String
 *
 * {Category} User Defined
 *
 * {param} string("world") input: The string need to be printed.
 *
 * {example} helloExemple("world") # hello world !.
 */
 public static void helloExample(String message) {
 if (message == null) {
 message = "World"; //NON-NLS-1$
 }
 System.out.println("Hello " + message + " !");
 }
}

```

La routine utilisateur nouvellement créée apparaît directement sous le nœud **Routines** dans la vue **Repository**. L'espace de travail s'ouvre sur un modèle de routine contenant par défaut un exemple simple, composé d'un commentaire en bleu, suivi du code correspondant.



Il est recommandé de documenter votre routine utilisateur à l'aide d'un commentaire détaillé. Ce commentaire inclut généralement les paramètres d'entrée et de sortie attendus pour l'utilisation de votre routine, ainsi que le résultat retourné par la routine et un exemple d'illustration. Ces informations sont généralement utiles pour la maintenance des routines et le travail collaboratif.

L'exemple de code suivant est fourni par défaut :

```
public static void helloExample(String message) {
 if (message == null) {
 message = "World"; //NON-NLS-1$
 }
 System.out.println("Hello " + message + " !");
```

5. Adaptez ou remplacez ce modèle par votre propre code et sauvegardez-le à l'aide de **Ctrl+S**, ou refermez le fichier pour l'enregistrer automatiquement.



Vous pouvez également copier tout ou partie d'une classe ou routine système vers une routine utilisateur à l'aide des raccourcis du presse-papier : **Ctrl+C** et **Ctrl+V** afin de les adapter à vos besoins. Pour plus d'informations concernant la personnalisation des routines, consultez [Personnalisation des routines système](#).

## 12.4.2. Modifier 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 :

1. Cliquez-droit sur la routine que vous souhaitez éditer, puis sélectionnez **Edit Routine**.
2. La routine s'ouvre dans l'espace de travail, et vous pouvez la modifier.
3. Une fois adaptée à vos besoins, appuyez sur **Ctrl+S** afin de l'enregistrer.

Si vous souhaitez réutiliser une routine système pour vos besoins spécifiques, consultez [Personnalisation des routines système](#).

## 12.4.3. Modifier les bibliothèques des routines utilisateur

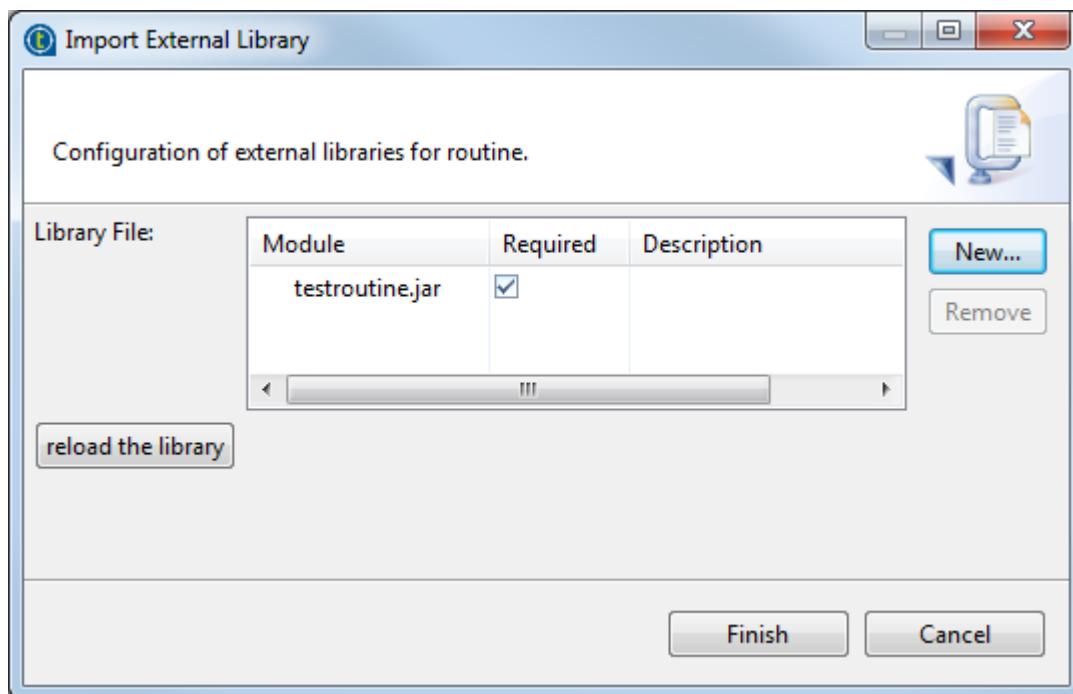
Vous pouvez éditer la bibliothèque de chaque routine utilisateur en important des fichiers .jar externes pour la routine sélectionnée. Ces fichiers externes seront affichés, comme les modules, dans la vue **Modules** de votre Studio. Pour plus d'informations concernant la vue **Modules**, consultez le *Guide d'installation et de migration Talend*.

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 :

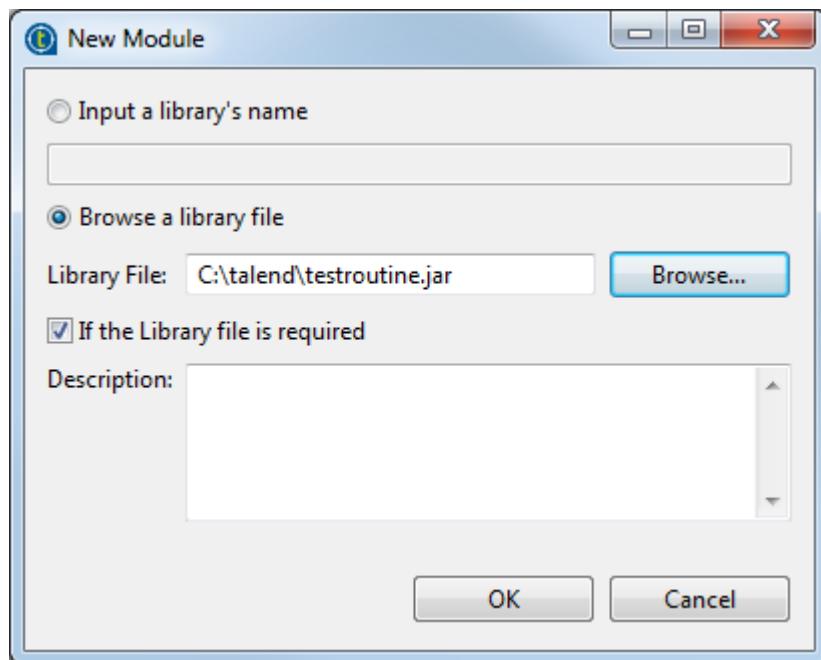
1. Si la bibliothèque qui doit être importée n'est pas disponible sur votre machine, téléchargez-la ou installez-la en utilisant la vue **Modules**. Vous pouvez également la télécharger ou la stocker dans un dossier local.
2. Dans la vue **Repository**, développez les noeuds **Code > Routines**.
3. Cliquez-droit sur la routine utilisateur dont vous souhaitez éditer la bibliothèque, puis sélectionnez **Edit Routine Library**.

La boîte de dialogue **[Import External Library]** s'ouvre.



4. Cliquez sur **New** pour ouvrir la boîte de dialogue **[New Module]** dans laquelle vous pourrez importer la bibliothèque externe.

 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**.



5. Spécifiez le fichier de la bibliothèque à importer :

- Si vous avez installé la bibliothèque en utilisant la vue **Modules**, saisissez le nom complet du fichier dans le champ **Input a library's name**.
- Si vous avez stocké le fichier de la bibliothèque dans un dossier local, sélectionnez l'option **Browse a library file** et cliquez sur **Browse** pour définir le chemin d'accès au fichier dans le champ correspondant.

6. Au besoin, saisissez une description dans le champ **Description**.

7. Cliquez sur **OK** afin de confirmer vos modifications.

Le fichier bibliothèque importé est affiché dans la liste **Library File**, dans la boîte de dialogue [**Import External Library**].

8. 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. Vous aurez peut-être besoin de redémarrer le Studio pour que la bibliothèque externe soit activée.

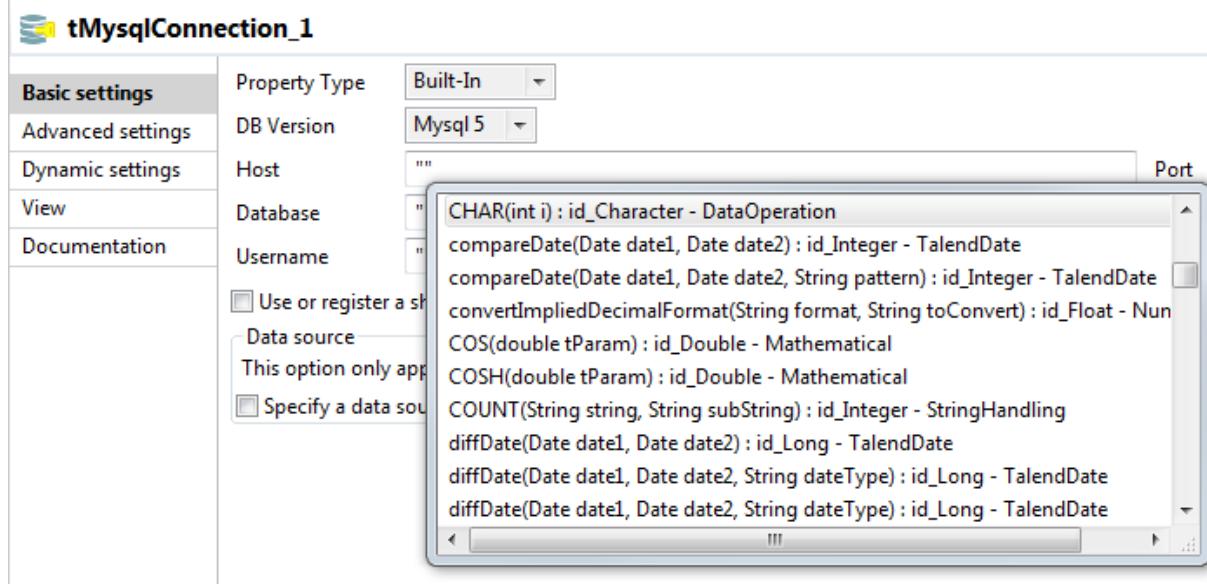
Pour plus d'informations concernant la vue **Modules**, consultez le *Guide d'installation et de migration Talend*.

## 12.5. 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 [Créer un Job](#).

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 :

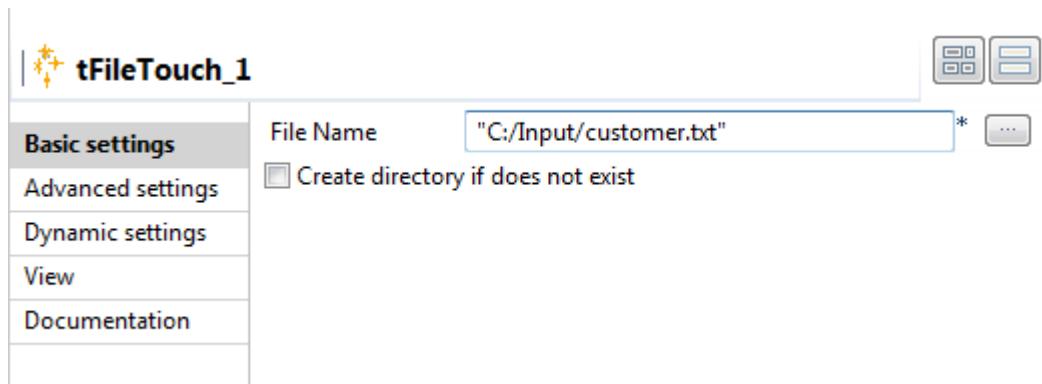
```
<ClassName>. <RoutineName>
```

## 12.6. Cas d'utilisation : créer un fichier à la date du jour

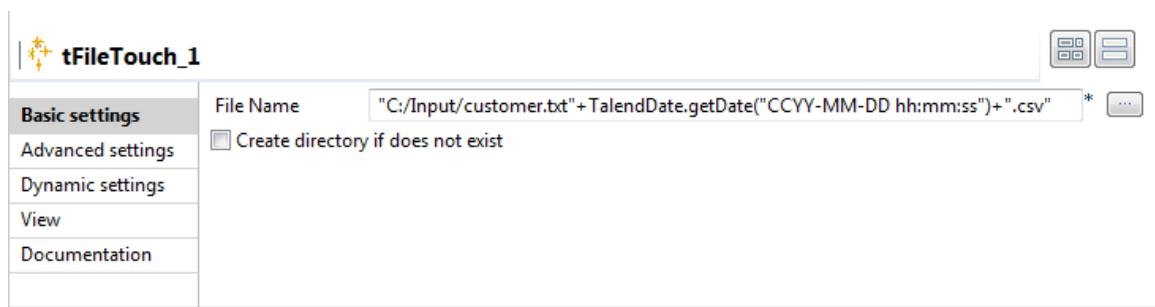
Le scénario suivant illustre l'utilisation d'une routine. Le Job se compose d'un composant unique qui appelle une routine système.



1. Dans la **Palette**, cliquez sur le dossier **File > Management**, puis glissez un **tFileTouch** dans l'éditeur graphique. Ce composant permet de créer un fichier vide.
2. Double-cliquez sur le composant afin d'afficher sa vue **Basic settings** dans l'onglet **Component**.
3. Dans le champ **FileName**, saisissez le chemin d'accès à votre fichier, ou cliquez sur le bouton [...] afin de parcourir votre répertoire.



4. Fermez les guillemets avant l'extension de votre fichier, tel que : *"D:/Input/customer".txt*.
5. Ajoutez le signe plus (+) entre les guillemets fermants et l'extension du fichier.
6. Puis appuyez sur **Ctrl+Espace** pour ouvrir la liste exhaustive des routines. Dans la liste d'auto-complétion qui s'affiche, sélectionnez *TalendDate.getDate* pour utiliser la routine de **Talend** qui permet d'obtenir la date courante.
7. Modifiez le format de date fourni par défaut si besoin.
8. Saisissez + après la variable *getDate* pour terminer l'appel de routine, puis entourez de guillemets l'extension du fichier.



*Si vous êtes sous Windows, les ":" entre les heures et les minutes, et entre les minutes et les secondes doivent être retirés.*

9. 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 13. 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.

## 13.1. 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 le *Studio Talend*, vous ne pouvez ni passer ni rejeter une ligne de données, comme vous pouvez le faire avec l'ETL. Pour plus d'informations concernant le rejet de lignes, consultez [Types de connexions](#).

Les modèles SQL sont conçus pour faciliter l'utilisation du mode ELT.

## 13.2. 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. Le *Studio Talend* fournit de nombreux modèles SQL pour simplifier les tâches les plus communes. Il comprend également un éditeur SQL vous permettant de personnaliser ou de créer vos propres modèles SQL, afin de répondre à vos besoins.

Ces modèles SQL sont utilisés avec les composants de la famille **Talend** ELT, tels que le **tSQLTemplate**, le **tSQLTemplateFilterColumns**, le **tSQLTemplateCommit**, le **tSQLTemplateFilterRows**, le **tSQLTemplateRollback**, le **tSQLTemplateAggregate** et le **tSQLTemplateMerge**. Ces composants exécutent les instructions SQL sélectionnées. 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.

Le *Studio Talend* fournit les types suivants de modèles SQL, sous le nœud **SQL templates** de la vue **Repository** :

- Modèles SQL **system** : ils sont classés selon le type de base de données.
- Modèles SQL personnalisés : ce sont les modèles que vous avez créés ou adaptés des modèles existants.

Des informations plus précises concernant les modèles SQL sont présentées dans les sections suivantes.



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.

## 13.3. Gérer les SQL Templates Talend

Le *Studio Talend* vous permet, via le dossier **SQL Templates** de la vue **Repository**, d'utiliser des modèles SQL système ou personnalisés dans les Jobs que vous créez dans le Studio, à l'aide des composants de la famille ELT.

Les sections suivantes vous expliquent comment gérer ces deux types de modèles SQL.

### 13.3.1. Les types de SQL Templates système

Cette section fournit des informations détaillées concernant les différents types de modèles SQL prédéfinis.

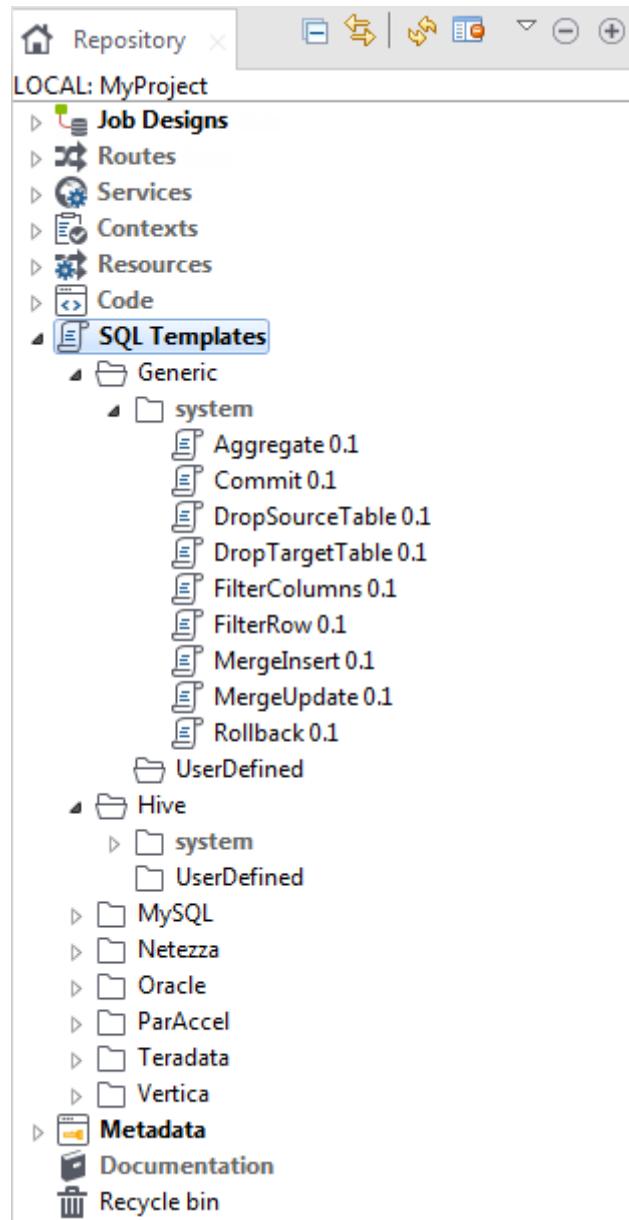
Les instructions de chaque groupe de modèles varient d'une base de données à l'autre, selon les opérations à effectuer. Elles sont également groupées dans chaque dossier, selon leur type.

Le tableau ci-dessous présente ces types ainsi que les informations liées à chacun.

Nom	Fonction	Composants associés	Paramètres requis dans le composant
<i>Aggregate</i>	Réalise l'agrégation (sum, average, count, etc.) d'un ensemble de données.	<b>tSQLTemplateAggregate</b>	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.	<b>tSQLTemplate</b> <b>tSQLTemplateAggregate</b> <b>tSQLTemplateCommit</b> <b>tSQLTemplateFilterColumns</b> <b>tSQLTemplateFilterRows</b> <b>tSQLTemplateMerge</b> <b>tSQLTemplateRollback</b>	Null
<i>Rollback</i>	Envoie une instruction Rollback au SGBDR.	<b>tSQLTemplate</b> <b>tSQLTemplateAggregate</b> <b>tSQLTemplateCommit</b> <b>tSQLTemplateFilterColumns</b> <b>tSQLTemplateFilterRows</b> <b>tSQLTemplateMerge</b> <b>tSQLTemplateRollback</b>	Null
<i>DropSourceTable</i>	Supprime une source table.	<b>tSQLTemplate</b> <b>tSQLTemplateAggregate</b> <b>tSQLTemplateFilterColumns</b> <b>tSQLTemplateFilterRows</b>	Nom de la table (lors de l'utilisation du <b>tSQLTemplate</b> ) Nom de la table source
<i>DropTargetTable</i>	Supprime une table cible.	<b>tSQLTemplateAggregate</b> <b>tSQLTemplateFilterColumns</b> <b>tSQLTemplateFilterRows</b>	Nom de la table cible
<i>FilterColumns</i>	Sélectionne et extrait un ensemble de données dans des colonnes données du SGBDR.	<b>tSQLTemplateAggregate</b> <b>tSQLTemplateFilterColumns</b> <b>tSQLTemplateFilterRows</b>	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.	<b>tSQLTemplateFilterRows</b>	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.	<b>tSQLTemplateMerge</b> <b>tSQLTemplateCommit</b>	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.	<b>tSQLTemplateMerge</b> <b>tSQLTemplateCommit</b>	Nom de la table cible (et schéma) Nom de la table source (et schéma) Conditions

## 13.3.2. 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 prédéfinies, et un sous-dossier **UserDefined** dans lequel vous pouvez stocker les instructions SQL que vous avez créées ou personnalisées.

Chaque dossier système contient différents types de modèles SQL, chaque modèle étant créé pour accomplir une tâche dédiée.

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 le *Studio Talend*.

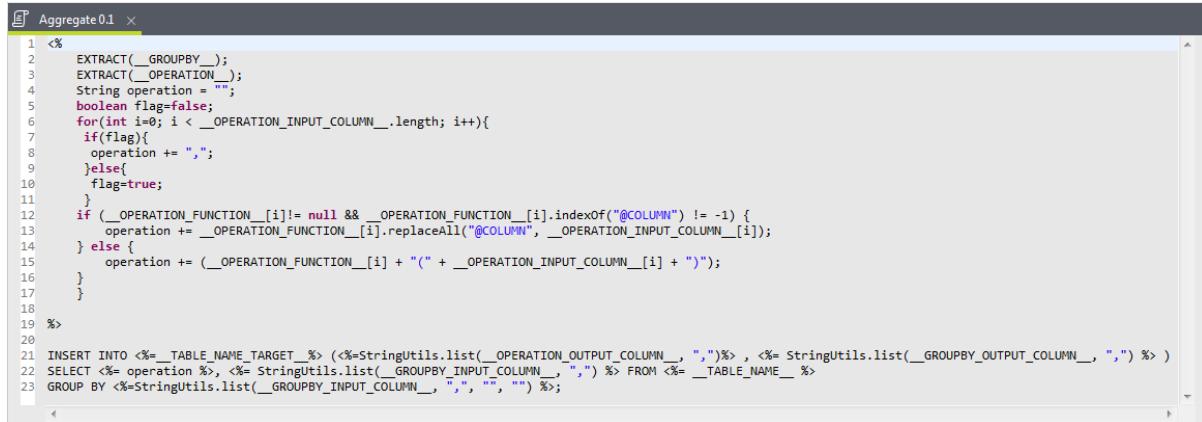


Les dossiers **system** et leur contenu sont en lecture seule.

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.



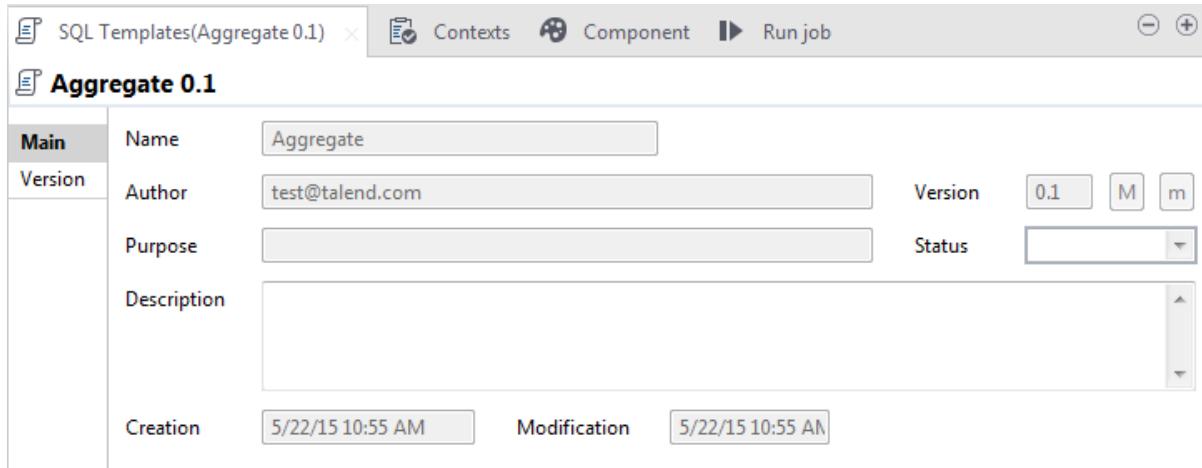
```

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 if (_OPERATION_FUNCTION_[i]!= null && _OPERATION_FUNCTION_[i].indexOf("@COLUMN") != -1) {
13 operation += _OPERATION_FUNCTION_[i].replaceAll("@COLUMN", _OPERATION_INPUT_COLUMN_[i]);
14 } else {
15 operation += (_OPERATION_FUNCTION_[i] + "(" + _OPERATION_INPUT_COLUMN_[i] + ")");
16 }
17 }
18 %
19
20
21 INSERT INTO <%=__TABLE_NAME_TARGET__> (<%=StringUtils.list(_OPERATION_OUTPUT_COLUMN_, ",")%> , <%= StringUtils.list(_GROUPBY_OUTPUT_COLUMN_, ",")%>)
22 SELECT <%= operation %>, <%= StringUtils.list(_GROUPBY_INPUT_COLUMN_, ",")%> FROM <%= __TABLE_NAME__ %>
23 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 précédente.

A chaque fois que vous cliquez sur un modèle SQL ou que vous ouvrez un modèle SQL, la vue des propriétés correspondante à ce modèle s'affiche en bas du Studio. Par exemple, vous cliquez le modèle *Aggregate* ou vous l'ouvrez, ses propriétés seront présentées comme suit :



Pour plus d'informations concernant les types de modèles SQL, consultez [Les types de SQL Templates système](#).

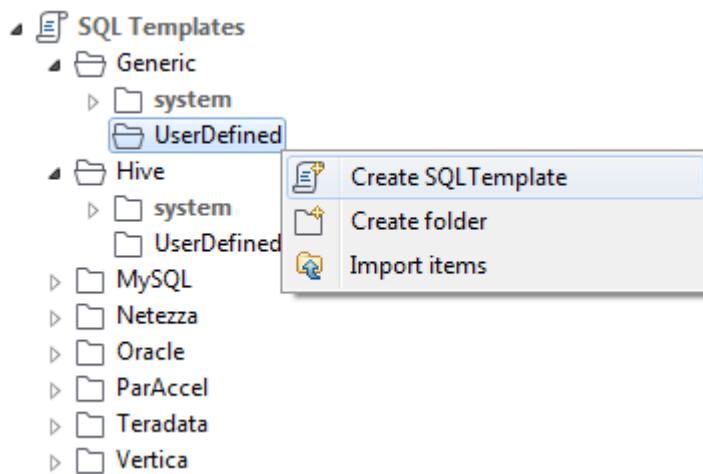
### 13.3.3. Créer des SQL Templates personnalisés

Comme la transformation que vous devez effectuer en mode ELT peut dépasser les possibilités des modèles SQL, le *Studio Talend* vous permet de développer vos propres modèles SQL, à condition de respecter quelques règles d'écriture. Ces modèles SQL sont stockés dans les dossiers **User-Defined**, regroupés selon la base de données dans laquelle ils seront utilisés.

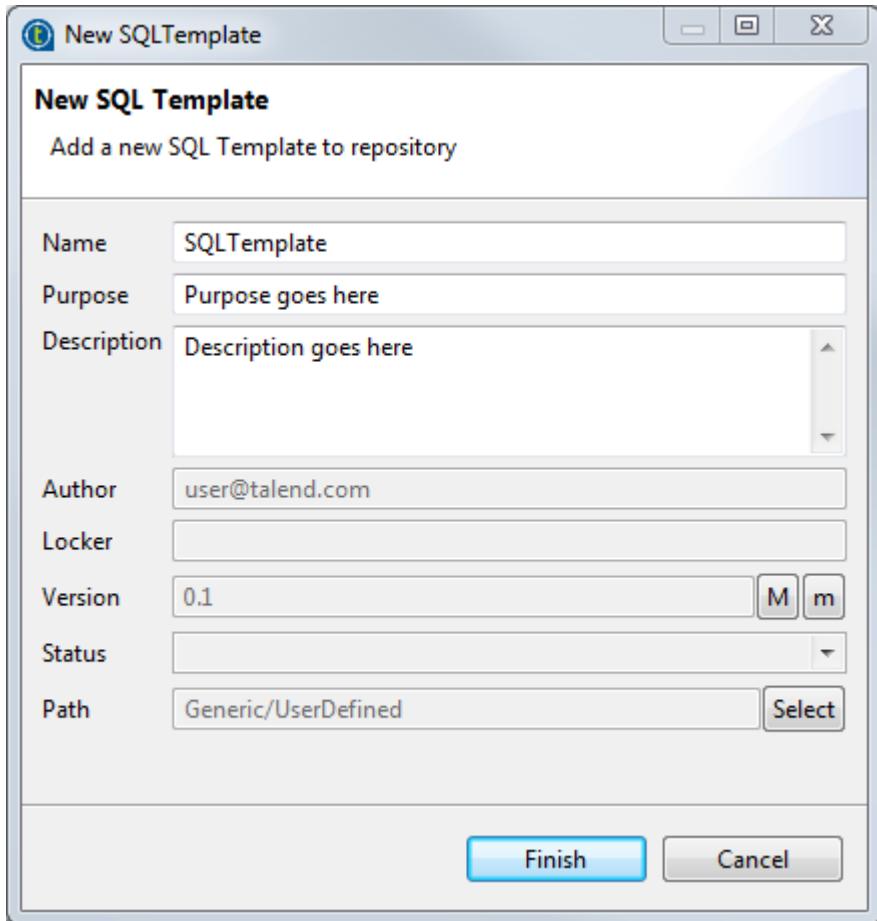
Pour plus d'informations concernant les règles d'écriture des modèles SQL, consultez [Règles d'écriture des SQL Templates](#).

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 SQLTemplate**] pour ouvrir l'assistant [**New SQLTemplate**].



- 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 scénario du composant **tMysqlTableList** à l'adresse <https://help.talend.com>.

### 13.3.4. Exemple d'utilisation des SQL Templates système

Puisque beaucoup d'instructions SQL sont communes, standardisées, le *Studio Talend* vous permet de bénéficier de plusieurs modèles SQL système.

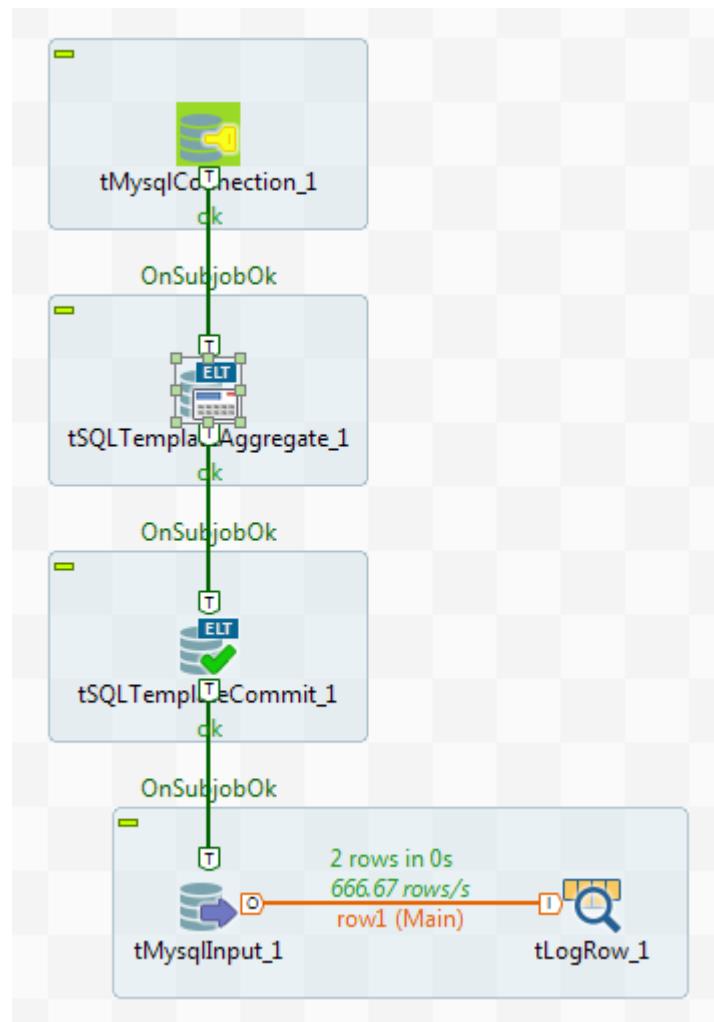
Cette section vous présente un exemple d'utilisation des modèles SQL système, à travers les étapes de l'utilisation de modèles SQL système Mysql dans un Job qui :

- ouvre une connexion à une base de données MySQL,
- collecte des données regroupées par valeurs spécifiques à partir d'une table de la base de données et écrit des données agrégées dans une table cible de la base de données,
- supprime la table source d'où proviennent les données agrégées,
- lit la table cible et liste les résultats d'exécution du Job.

Pour vous connecter à la base de données et agréger les colonnes de la table :

#### Configurer une connexion à une base de données MySQL

1. A partir de la **Palette**, glissez les composants suivants dans l'espace de modélisation graphique : **tMysqlConnection**, **tSQLTemplateAggregate**, **tSQLTemplateCommit**, **tMysqlInput**, et **tLogRow**.
2. Reliez le **tMysqlConnection** au **tSQLTemplateAggregate** à l'aide d'un lien **Trigger > OnComponentOk** afin de lier le **tMysqlConnection** au **tSQLTemplateAggregate**.
3. Répétez l'opération afin de connecter le **tSQLTemplateAggregate** au **tSQLTemplateCommit** et ce dernier au **tMysqlInput**.
4. Connectez le **tMysqlInput** au **tLogRow** à l'aide d'un lien **Main > Row**.



5. Dans l'espace de modélisation graphique, double-cliquez sur le composant **tMySQLConnection** pour ouvrir sa vue **Basic settings**.
  6. Dans la vue **Basic settings**, configurez manuellement les paramètres de connexion.

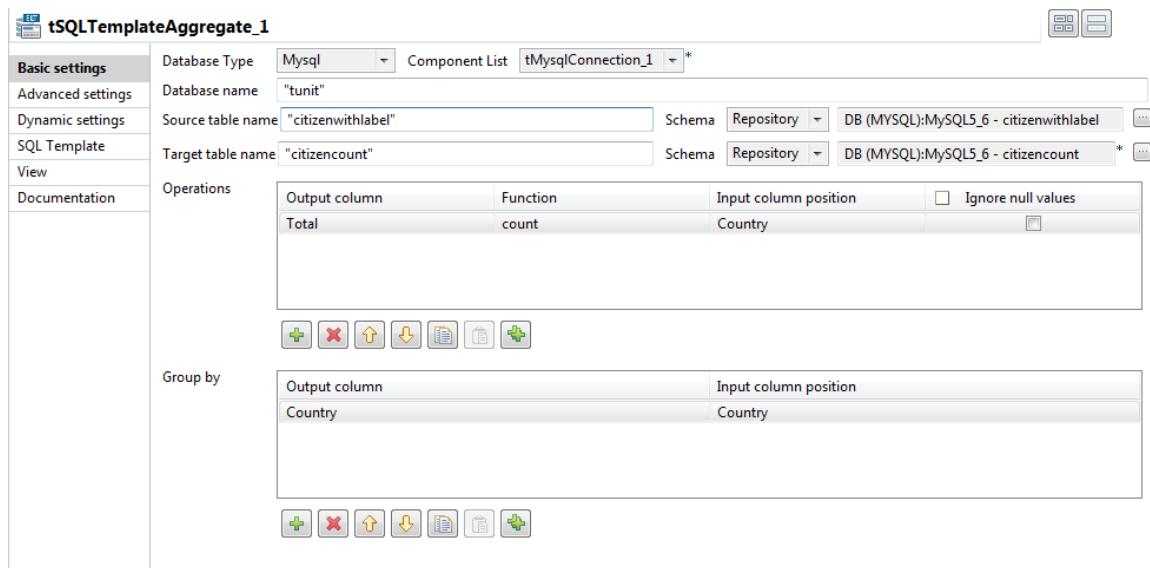
**tMysqlConnection\_1**

<b>Basic settings</b>	Property Type	Repository	DB (MYSQL):MySQL5_6	
Advanced settings	DB Version	Mysql 5		
Dynamic settings	Host	"localhost"	Port	"3306"
View	Database	"tunit"	Additional JDBC Parameters "noDatetimeStringSync=true"	
Documentation	Username	"root"	Password	*****
<input type="checkbox"/> Use or register a shared DB Connection <b>Data source</b> This option only applies when deploying and running in the Talend Runtime <input type="checkbox"/> Specify a data source alias				

7. Dans l'espace de modélisation graphique, double-cliquez sur le composant **tSQLTemplateCommit** pour ouvrir sa vue **Basic settings**.
  8. Dans la liste **Database Type**, sélectionnez la base de données que vous utilisez, et dans la liste **Component List**, sélectionnez le composant de connexion adéquat, si plus d'une connexion est ouverte.

**Regroupier les données, écrire des données agrégées et supprimer la table**

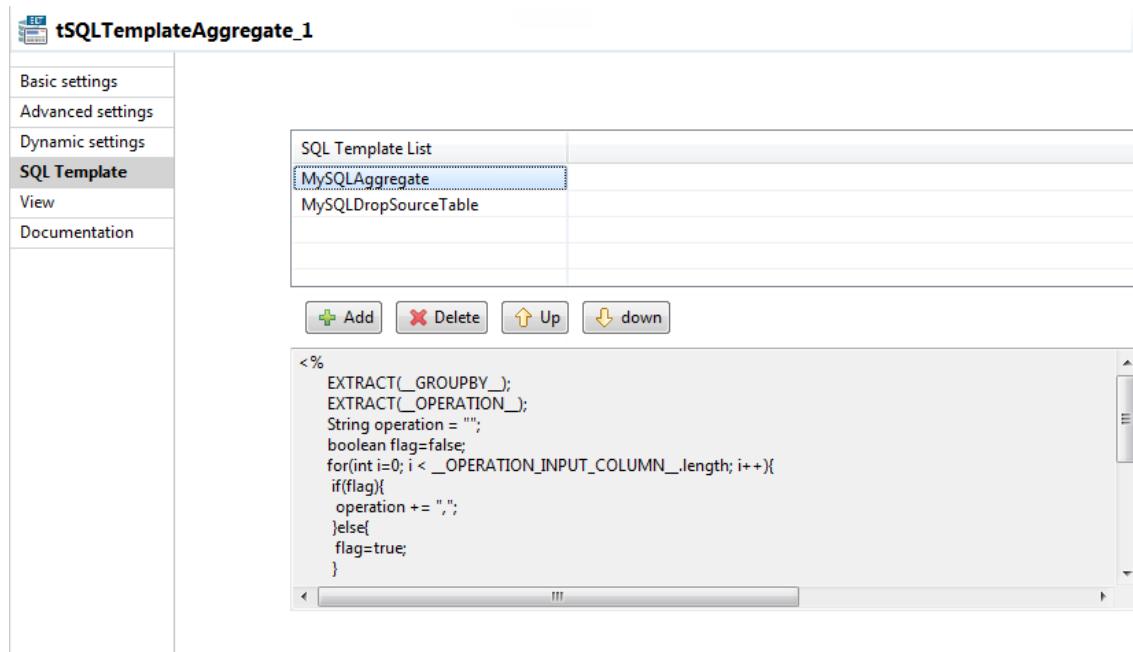
1. Dans l'espace de modélisation graphique, double-cliquez sur le **tSQLTemplateAggregate** pour ouvrir sa vue **Basic settings**.



2. Dans la liste **Database Type**, sélectionnez la base de données correspondante et dans la liste **Component List**, sélectionnez le composant de connexion adéquat, si plus d'une connexion est ouverte dans le Job.
3. Saisissez le nom de la base de données, de la table source et de la table cible dans les champs correspondants et définissez la structure des données, dans la table source et dans la table cible.

Le schéma de la table source comprend trois colonnes : *First\_Name*, *Last\_Name* et *Country*. Le schéma de la table cible comprend deux colonnes : *country* et *total*. Dans cet exemple, regroupez les citoyens par nationalité, et comptez le nombre de personnes dans chaque pays. Pour ce faire, définissez les paramètres **Operations** et **Groupby** selon vos besoins.

4. Dans le tableau **Operations**, cliquez sur le bouton **[+]** afin d'ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** et sélectionnez dans la liste déroulante la colonne de sortie qui contiendra les données comptées.
5. Cliquez sur la ligne **Function**, puis sélectionnez dans la liste l'opération à effectuer.
6. Dans le tableau **Group by**, cliquez sur le bouton **[+]** afin d'ajouter une ou plusieurs lignes, puis cliquez sur la ligne **Output column** et sélectionnez dans la liste la colonne de sortie qui contiendra les données agrégées.
7. Cliquez sur l'onglet **SQL Template** pour ouvrir la vue correspondante.



8. Cliquez deux fois sur le bouton [+] sous le tableau **SQL Template List** afin d'ajouter deux modèles SQL.
9. Cliquez sur la ligne du premier modèle SQL et sélectionnez le modèle **MySQLAggregate** dans la liste déroulante. Ce modèle génère le code pour agréger des données, selon la configuration des **Basic settings**.
10. Répétez l'opération et sélectionnez le modèle **MySQLDropSourceTable** pour la ligne du deuxième modèle. Ce modèle génère le code pour supprimer la table source d'où proviennent les données agrégées.



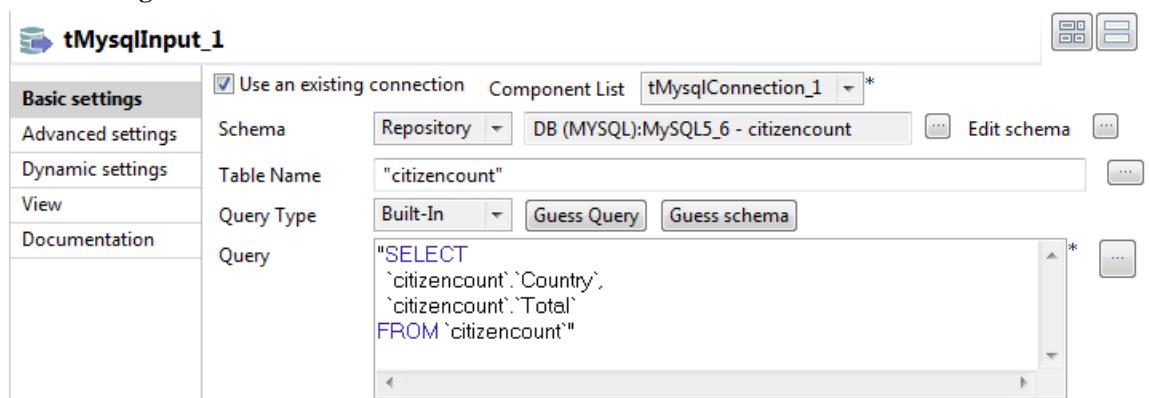
Afin d'ajouter de nouveaux modèles SQL à un composant ELT chargé de les exécuter, vous pouvez simplement glisser le(s) modèle(s) de votre choix à ce composant soit dans l'espace de travail soit dans le tableau **SQL Template List** de ce composant.



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** pour la première ligne de modèle et **MySQLAggregate** pour la seconde, la table source sera supprimée définitivement et l'agrégation ne pourra être effectuée.

## Lire la base de données cible et lister le résultat de l'exécution du Job

1. Dans l'espace de modélisation graphique, Double-cliquez sur le composant **tMysqlInput** pour ouvrir sa vue **Basic settings**.



2. Cochez la case **Use an existing connection** pour utiliser la connexion à la base de données définie dans le composant **tMySQLConnection**.

3. Afin de définir le schéma, sélectionnez **Repository** et cliquez sur le bouton [...] pour sélectionner la table de la base de données dont le schéma est utilisé. Dans cet exemple, la table cible contenant les données agrégées est sélectionnée.
4. Dans le champ **Table Name**, saisissez le nom de la table sur laquelle effectuer votre requête. Dans cet exemple, la table est celle qui contient les données agrégées.
5. Dans la zone **Query**, saisissez l'instruction de la requête pour sélectionner les colonnes à afficher.
6. Sauvegardez votre Job et appuyez sur **F6** pour l'exécuter.

La table source est supprimée.

```
Starting job ELTYudong at 02:43 24/05/2010.

[statistics] connecting to socket on port 3918
[statistics] connected
+-----+
| tLogRow_1 |
+-----+
| country | total |
+-----+
| Canada | 2030 |
| China | 2012 |
| France | 2009 |
| Japan | 1925 |
| USA | 2024 |
+-----+

[statistics] disconnected
Job ELTYudong ended at 02:43 24/05/2010. [exit code=0]
```

Une table composée de deux colonnes, *citizenCount*, est créée dans la base de données. Elle regroupe les citoyens par nationalité et affiche le total de personnes dans chaque pays.





## Annexe A. Personnalisation du Studio Talend et de ses préférences

Ce chapitre fournit des informations relatives à la personnalisation du *Studio Talend* et à la configuration de ses préférences, vous permettant de faire fonctionner votre *Studio Talend* comme vous le souhaitez.

Dans les sections suivantes, vous trouverez des informations concernant :

- *Personnalisation des paramètres du projet*
- *Personnaliser l'espace de travail*
- *Filtrage des entrées listées dans la vue Repository*
- *Configuration des préférences du Studio Talend*

## A.1. Personnalisation des paramètres du projet

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

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.

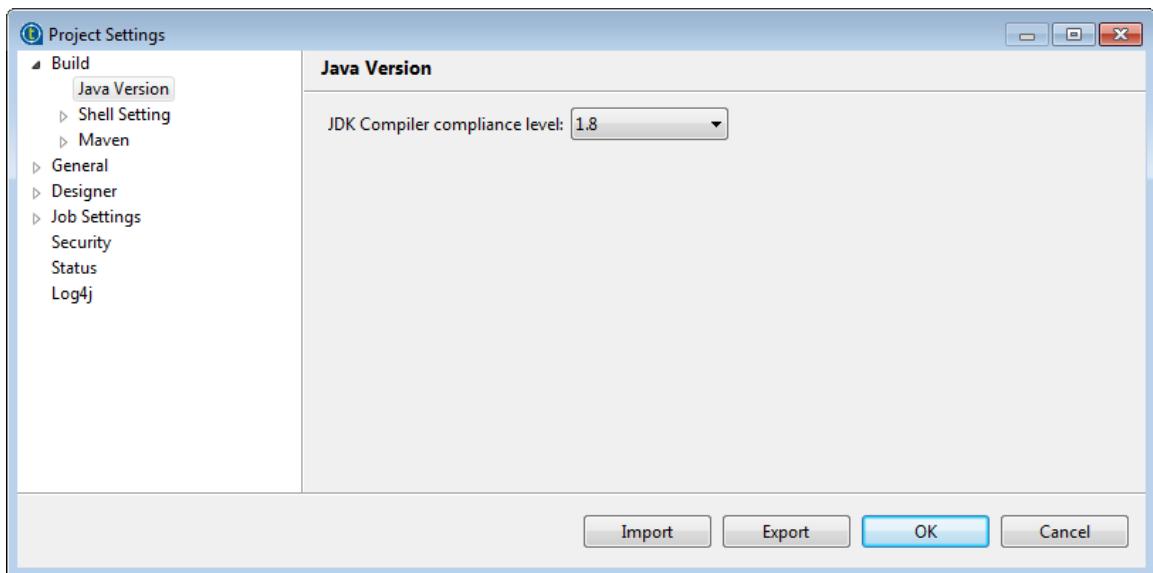
- Pour exporter les paramètres, cliquez sur le bouton **Export**. L'export générera un fichier XML contenant tous les paramètres de votre projet.
- Pour importer les paramètres, cliquez sur le bouton **Import** et sélectionnez le fichier XML contenant les paramètres d'un autre projet à appliquer au projet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

### A.1.1. Configuration du niveau de conformité du compilateur

Le niveau de conformité du compilateur (**Compiler compliance level**) correspond à la version de Java utilisée pour la génération de code des Jobs.

Pour plus d'informations sur la compatibilité des niveaux de conformité du compilateur, consultez le *Guide d'installation et de migration Talend*.

1. Cliquez sur  dans la barre d'outils de la fenêtre du Studio ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.
2. Dans l'arborescence, développez le nœud **Build** et cliquez sur **Java Version**.
3. Dans la liste **JDK Compiler compliance level**, sélectionnez le niveau de conformité du compilateur que vous souhaitez utiliser puis cliquez sur **OK**.



## A.1.2. Personnaliser les modèles de scripts de build Maven

Votre *Studio Talend* fournit les modèles suivants par défaut pour générer des scripts de build.

- des modèles de script Maven pour un export en tant que Job standalone
- un modèle de script Maven pour l'export d'un bundle OSGI de Jobs

En vous basant sur les modèles de build globaux, par défaut, vous pouvez créer des scripts au niveau des dossiers. Les scripts de build générés, basés sur ces modèles, sont exécutés lors de la construction de Jobs.

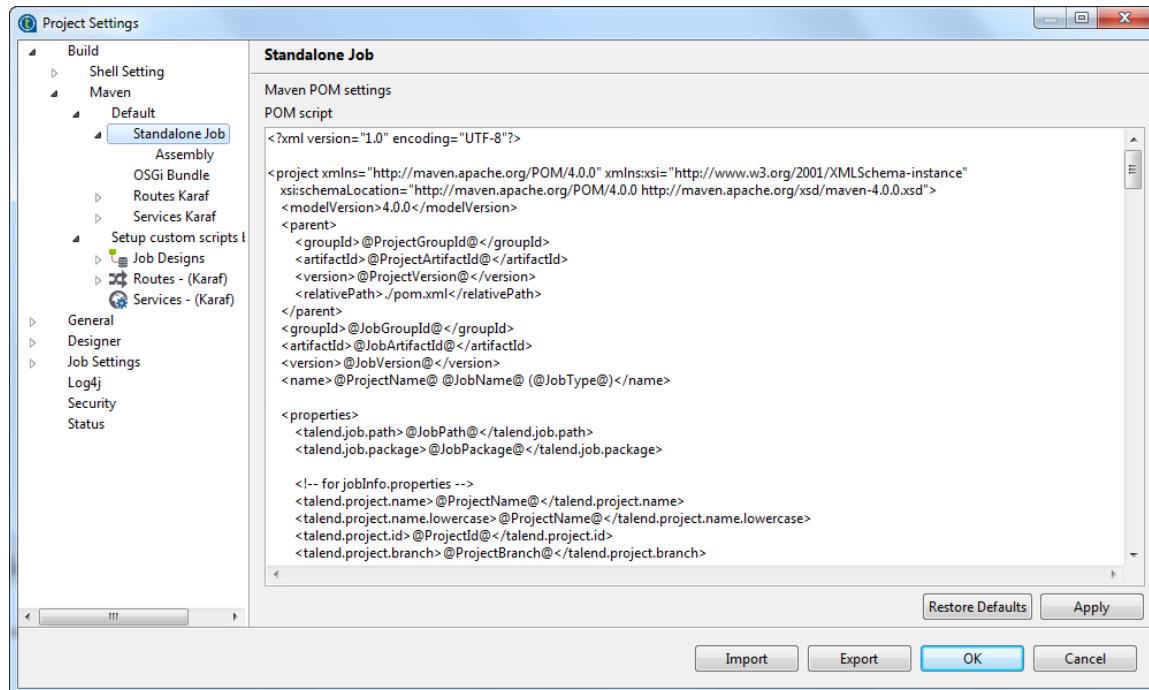
Cette section fournit des informations concernant la personnalisation des modèles de scripts de build. Pour plus d'informations concernant la construction d'un Job, consultez *Construction de Jobs*.

### A.1.2.1. Personnaliser les modèles de scripts de build globaux

Dans la boîte de dialogue [**Project Settings**], vous pouvez voir et personnaliser les modèles de scripts de build globaux, sous le nœud **Build > Maven > Default**. Ces modèles de scripts s'appliquent à tous les Jobs dans le dossier racine et tous les sous-dossiers, exceptés ceux ayant leurs propres modèles de scripts de build configurés.

L'exemple suivant vous montre comment personnaliser le modèle de script POM global pour les Jobs standalone :

1. Dans la barre de menu, cliquez sur **File > Edit Project properties** pour ouvrir la boîte de dialogue [**Project Settings**].
2. Développez les nœuds **Talend > Maven > Default** puis cliquez sur le nœud **Standalone Job** pour ouvrir la vue correspondante affichant le contenu du modèle de script POM.



Selon le Studio que vous utilisez, les éléments des paramètres de projet de votre Studio peuvent différer de ce qui vous est présenté ici.

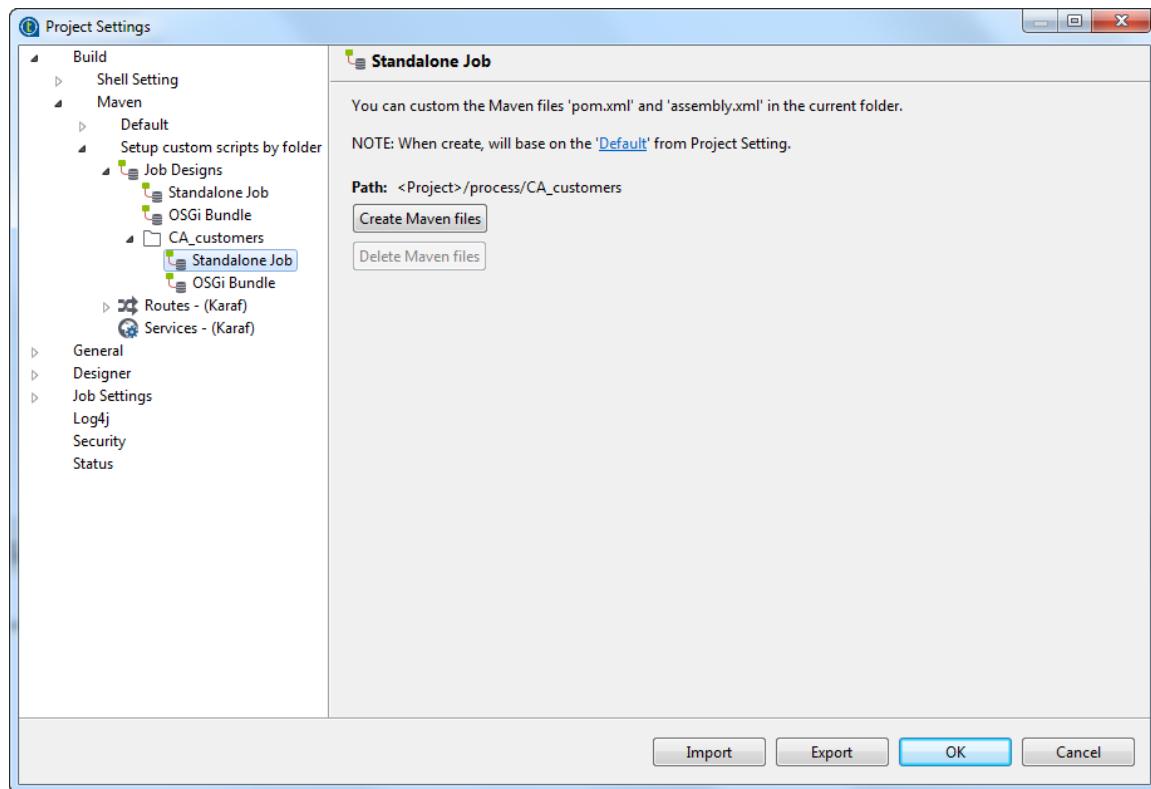
- Modifiez le code du script dans le panneau de texte puis cliquez sur **OK** pour terminer la personnalisation.

## A.1.2.2. Personnaliser les modèles de scripts de build de niveau dossier

En vous basant sur les modèles de scripts de build globaux, vous pouvez ajouter et personnaliser les modèles de scripts pour les Jobs dossier par dossier, sous le nœud **Build > Maven > Setup custom scripts by folder**. Les modèles de scripts de build ajoutés dans un dossier s'appliquent à tous les Jobs de ce dossier et ses sous-dossiers, sauf ceux possédant leurs propres modèles de scripts de build configurés.

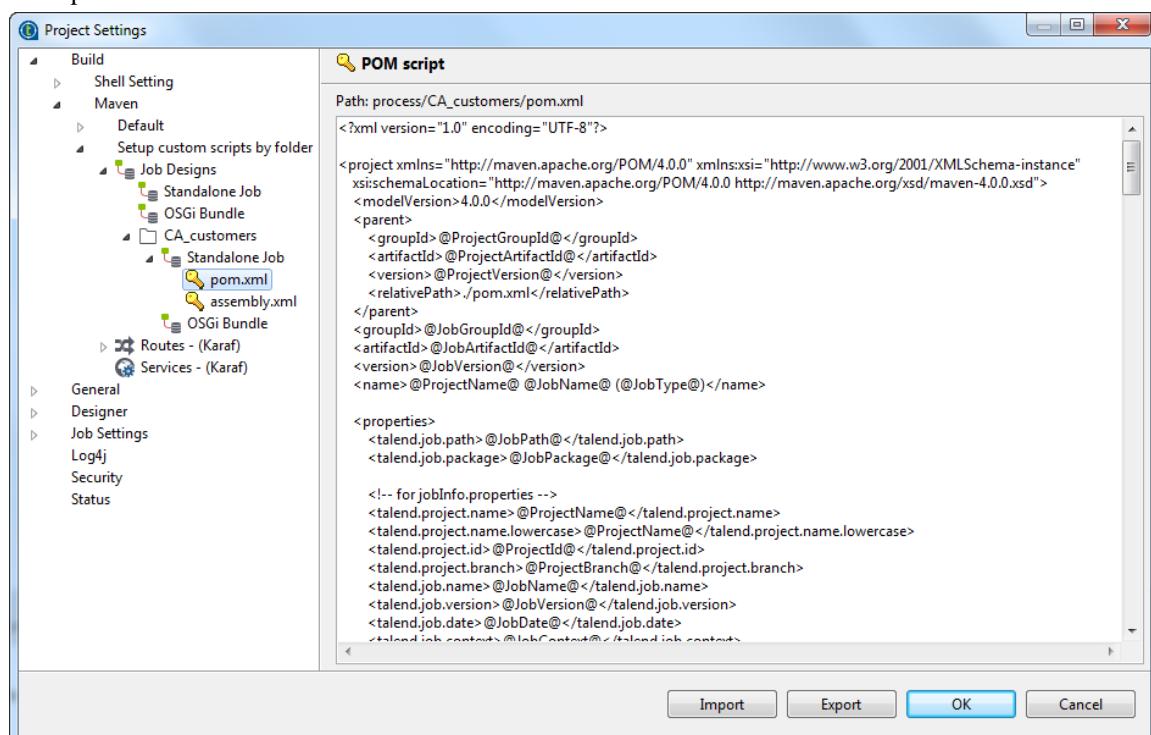
L'exemple suivant explique comment ajouter et personnaliser le modèle de script POM pour la construction de Jobs standalone depuis les Jobs du dossier *CA\_customers* :

- Dans le menu, cliquez sur **File > Edit Project properties** pour ouvrir la boîte de dialogue **[Project Settings]**.
- Développez les nœuds **Build > Maven > Setup custom scripts by folder > Job Designs > CA\_customers** puis cliquez sur le nœud **Standalone Job** pour ouvrir la vue correspondante, de laquelle vous pouvez ajouter des modèles de scripts ou supprimer tous les modèles existants.



Selon le Studio que vous utilisez, les éléments des paramètres de projet de votre Studio peuvent différer de ce qui vous est présenté ici.

3. Cliquez sur le bouton **Create Maven files** afin de créer des modèles de scripts basés sur les modèles globaux pour les Jobs standalone.
4. Sélectionnez le modèle de script à personnaliser, *pom.xml* dans cet exemple, pour afficher le code du script dans la vue du code. Modifiez le code du script dans le panneau de texte et cliquez sur **OK** afin de terminer votre personnalisation.



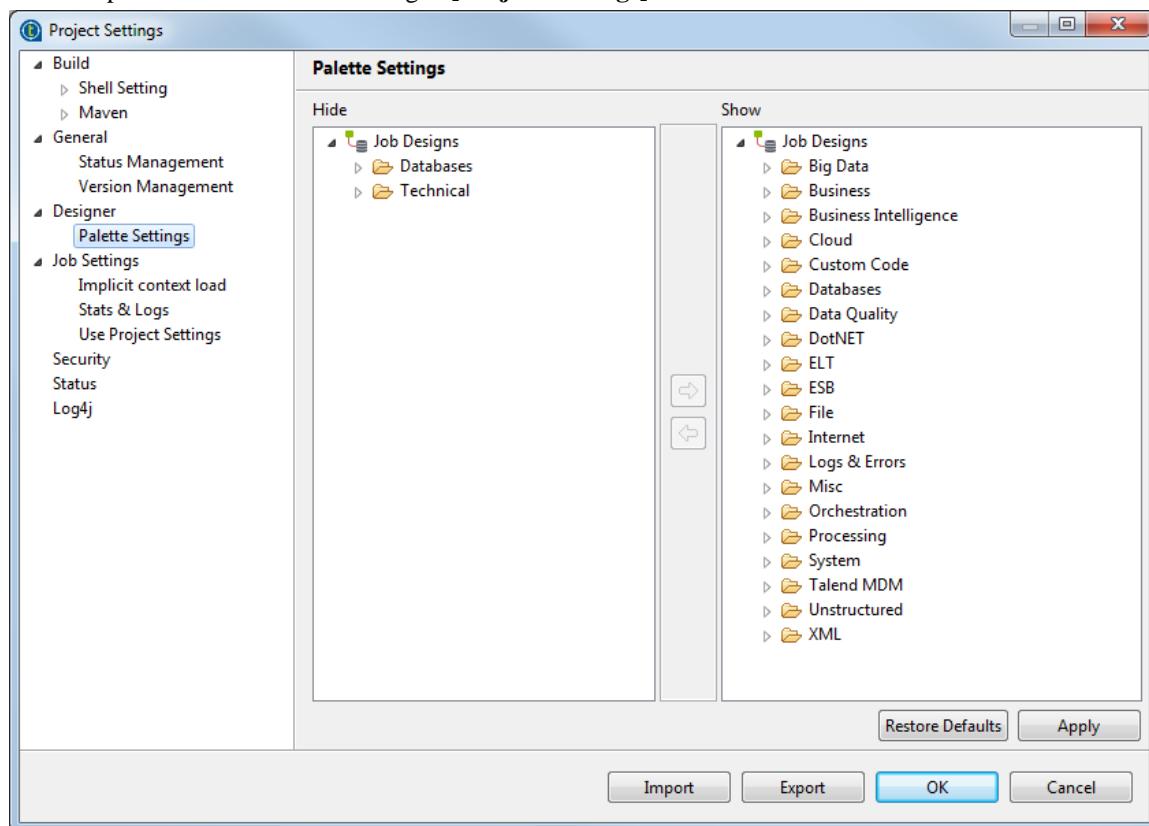
Une fois les modèles de scripts de build créés pour un dossier, vous pouvez également aller dans le répertoire où sont stockés les fichiers XML, <studio\_installation\_directory>\workspace\<project\_name>\process\CA\_customers dans cet exemple et modifier directement le fichier XML du modèle à personnaliser. Vos modifications affectent tous les Jobs du dossier et des sous-dossiers, sauf ceux possédant leurs propres scripts configurés.

### A.1.3. 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.



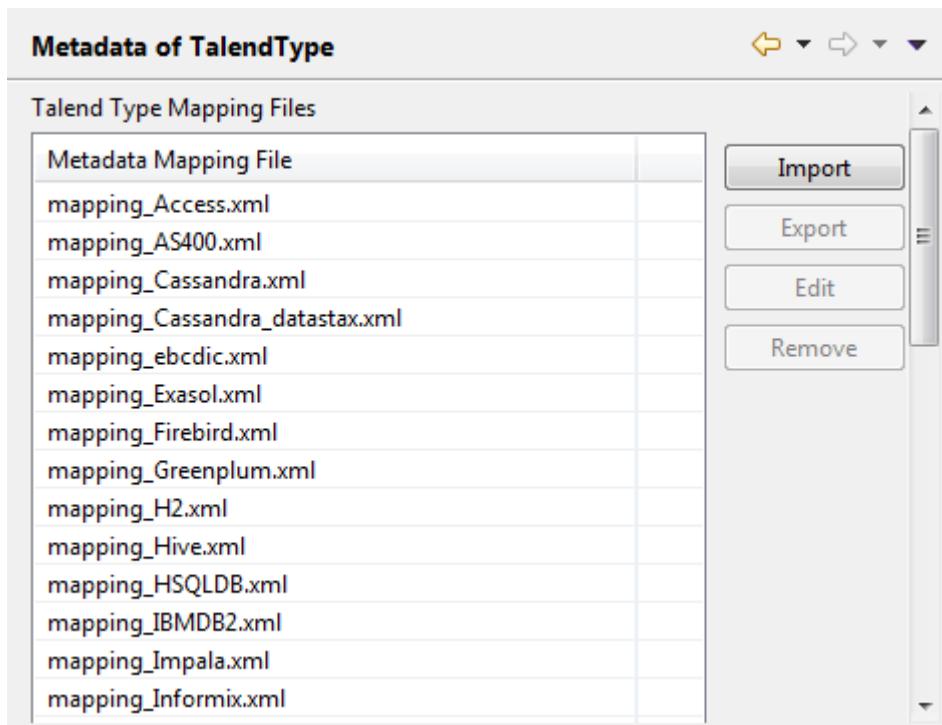
Pour revenir aux paramètres par défaut, cliquez sur le bouton **Restore Defaults**.

Pour plus d'informations concernant la **Palette**, consultez [Modifier la disposition et les paramètres de la Palette](#).

## A.1.4. Mapping de type

Vous pouvez définir les paramètres de conversion de type dans le *Studio Talend*, de Java vers les bases de données et des bases de données vers le langage Java.

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 **Metadata of Talend Type** pour ouvrir la vue correspondante.



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

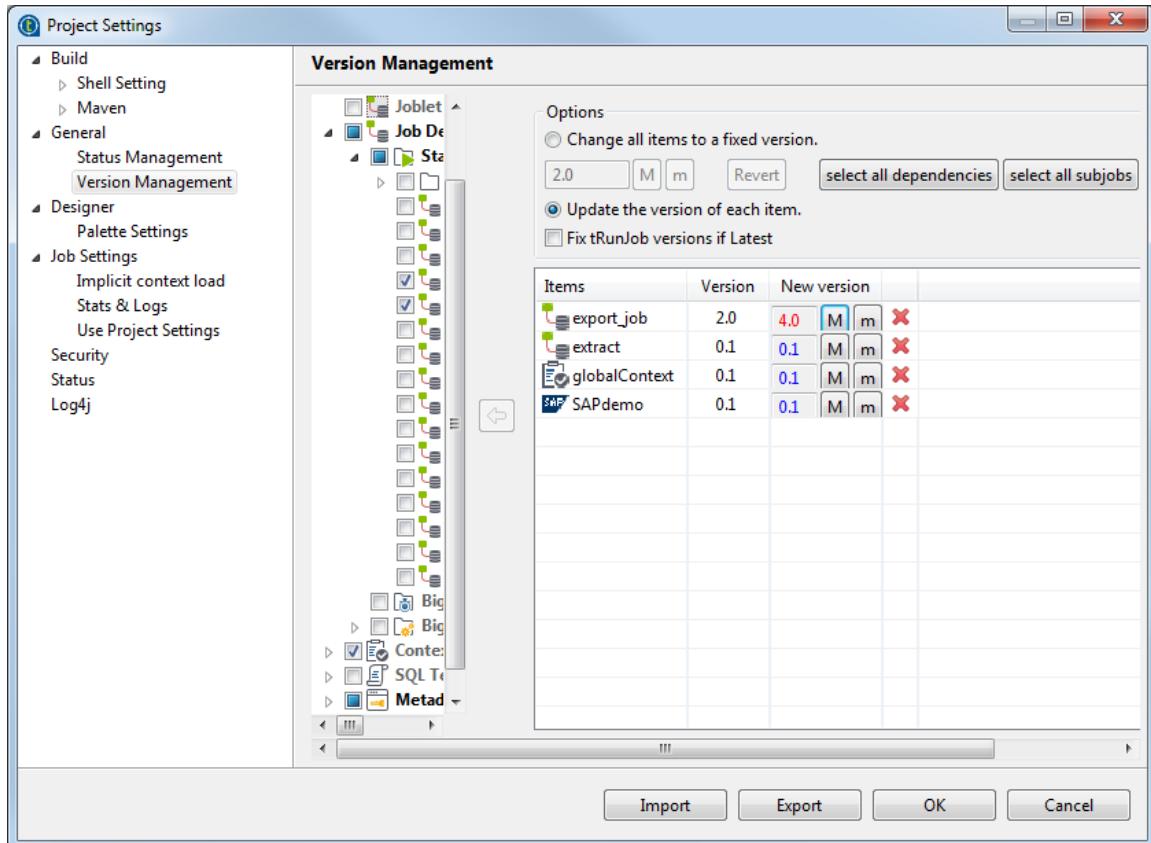
- Vous pouvez importer, exporter ou supprimer chaque fichier de conversion en cliquant respectivement sur les boutons **Import**, **Export** et **Remove**.
- Vous pouvez ouvrir et modifier chaque fichier afin d'affiner les conversions en fonction de votre utilisation en cliquant sur le bouton **Edit**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

## A.1.5. 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 (des) Job(s) fils. Pour plus d'informations concernant le **tRunJob**, consultez la documentation du composant **tRunJob** à l'adresse <https://help.talend.com>.

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

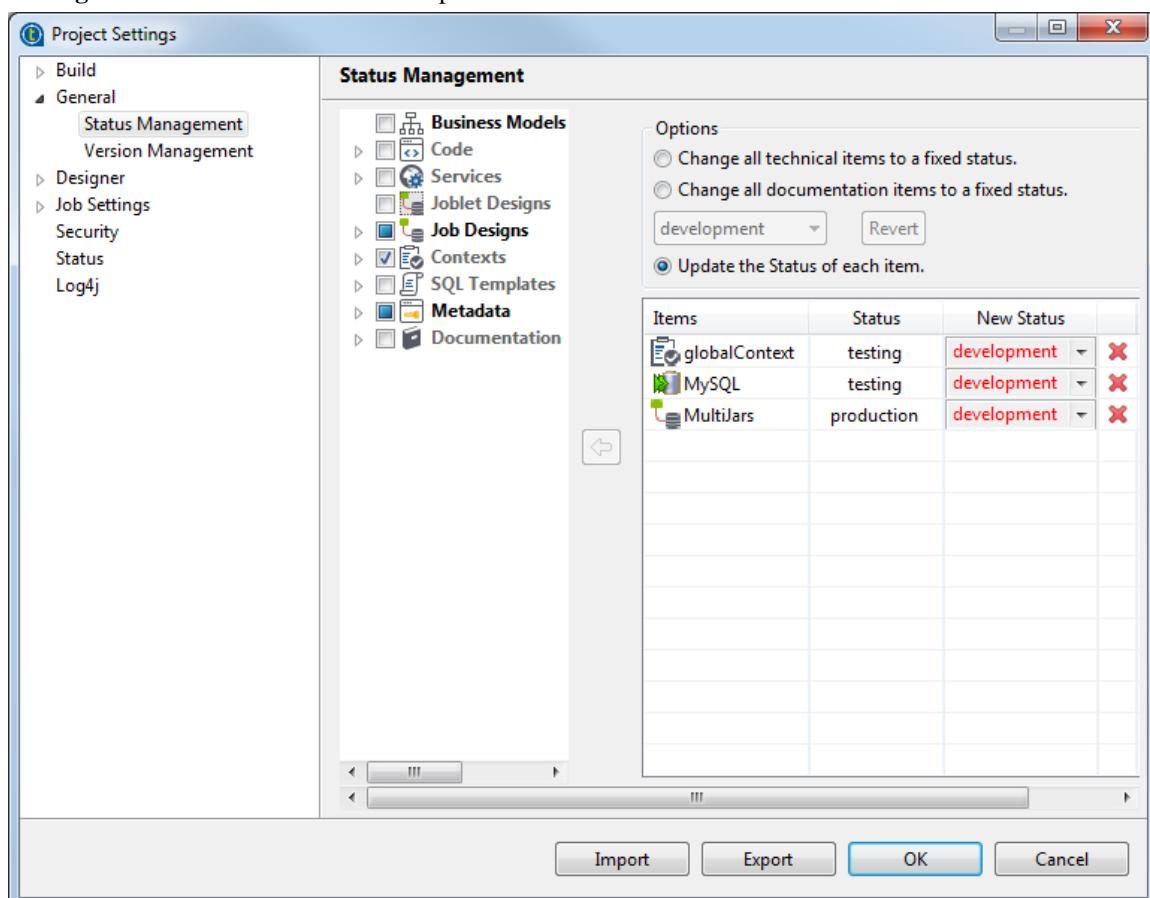


Pour plus d'informations concernant la gestion des versions, consultez [Gérer les versions](#).

## A.1.6. 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.

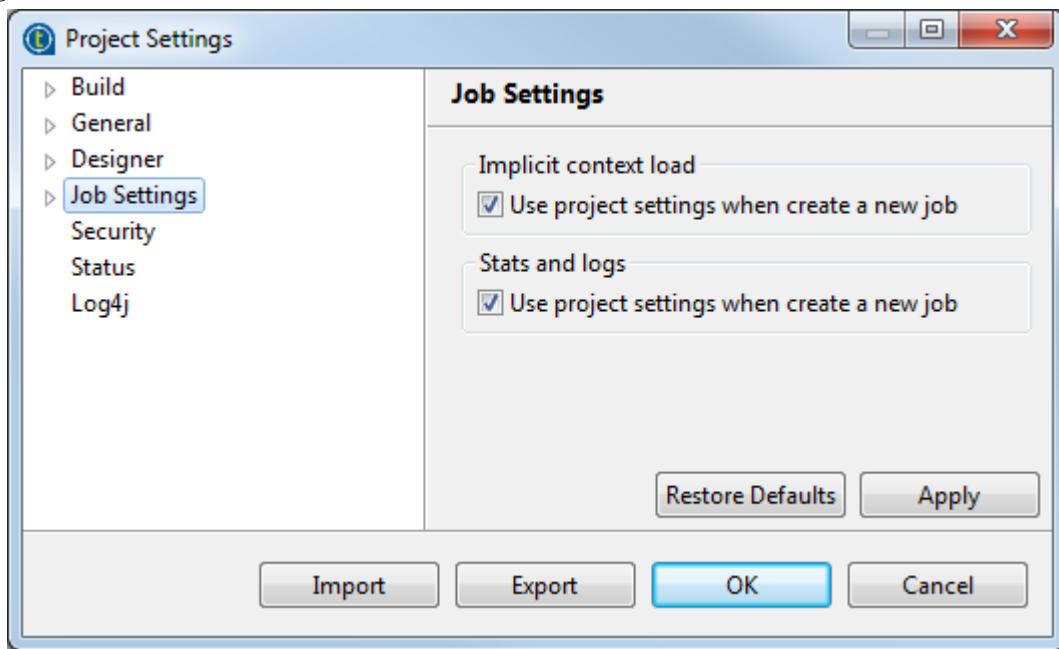
 Pour plus d'informations concernant les statuts des Jobs, consultez [Personnaliser les paramètres du Job](#).

## A.1.7. 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 nœud **Job Settings** pour ouvrir la vue correspondante.
3. Cochez la case **Use project settings when create a new job** des zones **Implicit Context Load** et **Stats and Logs**.



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

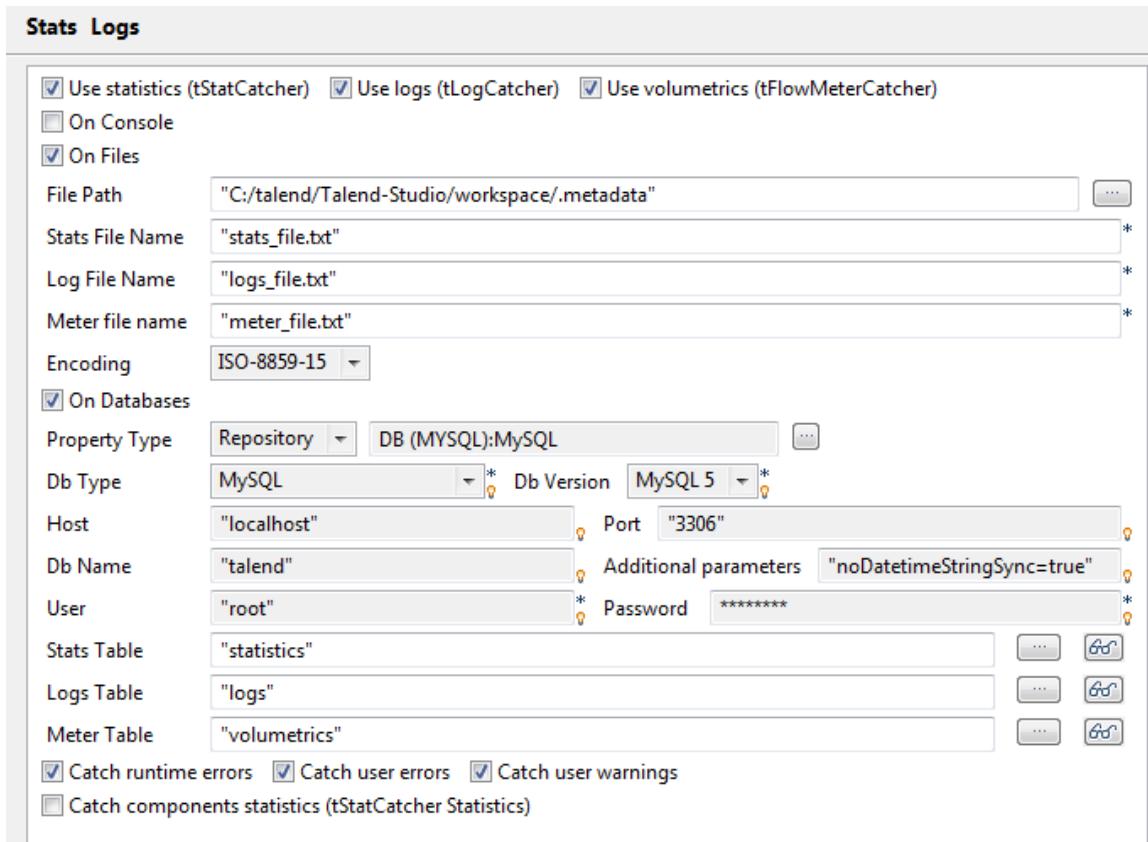
## A.1.8. Configurer les informations de logs

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 nœud **Job Settings** et cliquez sur **Stats & Logs** pour afficher la vue correspondante.



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



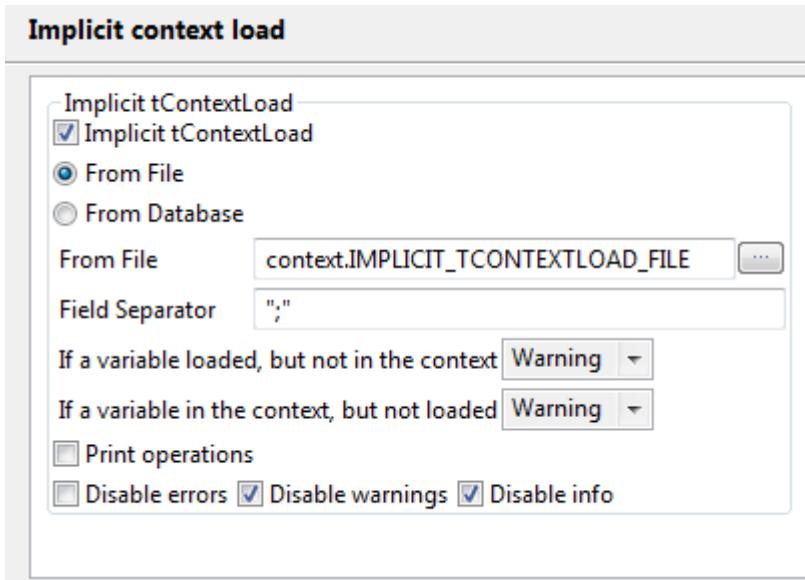
Si vous avez sauvé les informations de connexion dans une variable de contexte, vous pouvez y accéder via le raccourci **Ctrl+Espace**.

## A.1.9. 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**.



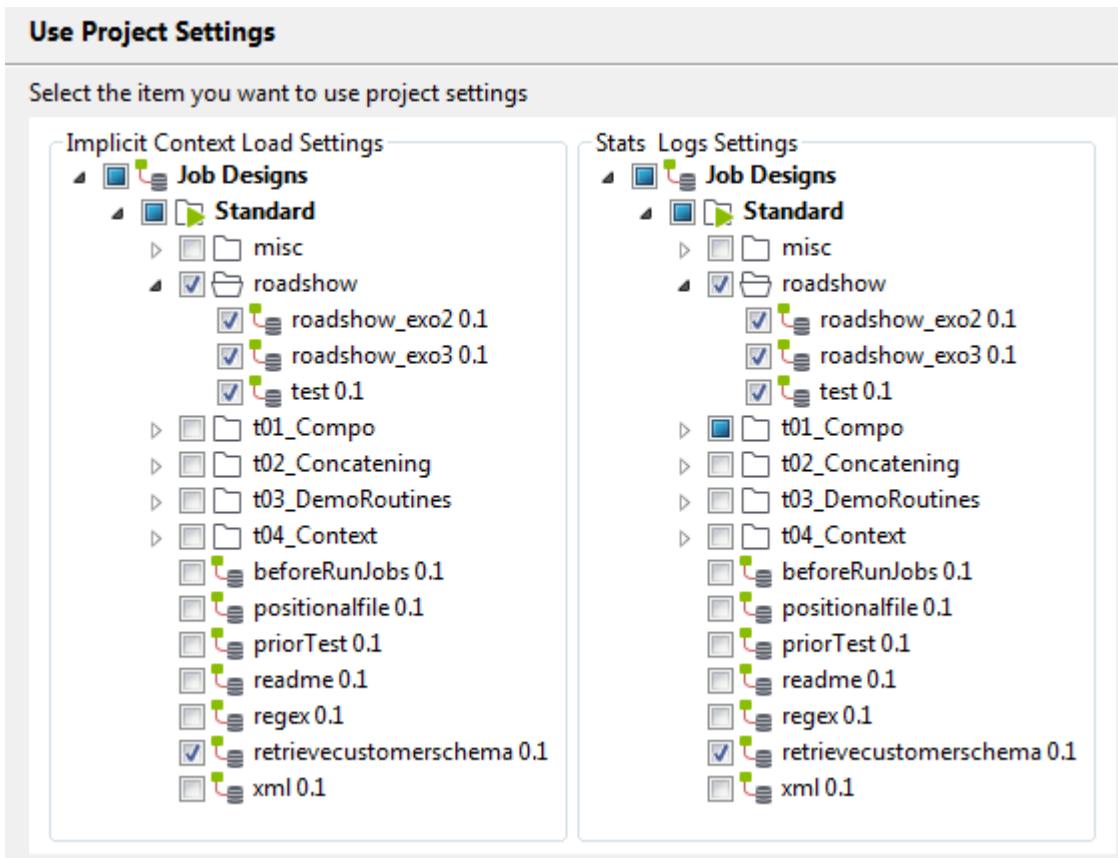
3. Sélectionnez le type de fichier dans lequel les données de contextes seront conservées, soit sous forme de fichier en sélectionnant le champ **From File**, soit sous forme de base de données en sélectionnant **From Database**.
4. Pour un fichier, renseignez le chemin d'accès et le séparateur de champ 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.

## A.1.10. 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.

## A.1.11. Définir les paramètres des Statuts

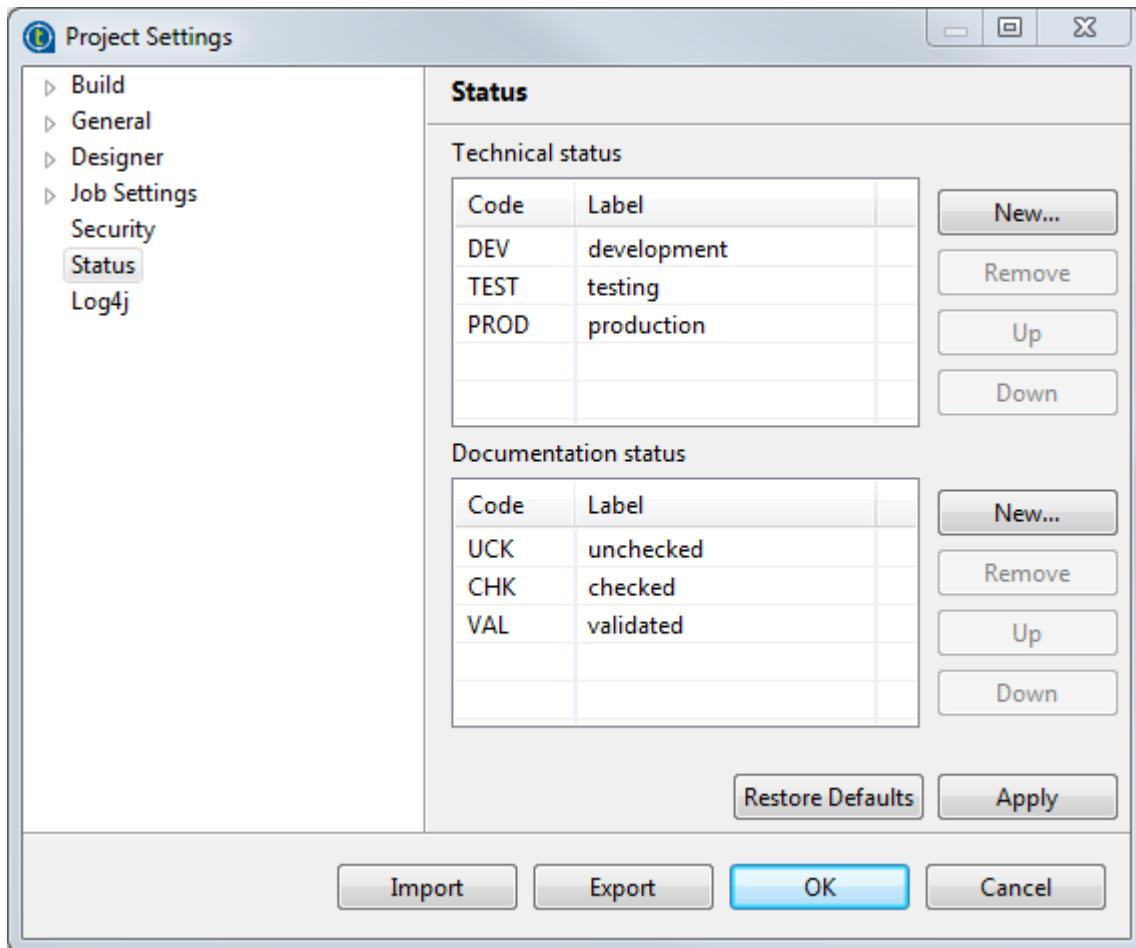
Dans les paramètres du projet, vous pouvez définir des niveaux de statuts.

Pour cela :

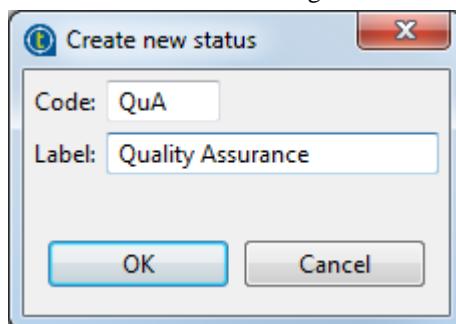
1. Dans la barre d'outils du Studio, cliquez sur ou cliquez sur **File > Edit Project Properties** dans la barre de menu pour ouvrir la boîte de dialogue **[Project Settings]**.

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

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



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



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

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

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

- Une fois le statut renseigné, cliquez sur **OK** sauvegarder.

Désormais la liste **Status** vous permet d'appliquer vos paramètres de classification personnalisés aux Jobs et aux Business Models 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.

## A.1.12. 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 :

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



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

Property Type	<b>Repository</b>	DB (MYSQL):demoMysql
DB Version	Mysql 5	
<input checked="" type="checkbox"/> Use an existing connection		
Host	"localhost"	Port "3306"
Username	"root"	Password *****

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

## A.2. Personnaliser l'espace de travail

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



Tous les panneaux, onglets et vues décrites dans cette documentation sont spécifiques au *Studio Talend*. Certaines vues listées dans la boîte de dialogue **[Show view]** sont spécifiques à Eclipse et ne concernent pas cette documentation. Pour plus d'informations sur ces vues, veuillez consulter la documentation Eclipse sur <http://www.eclipse.org/documentation/>

## A.2.1. Modifier la disposition et les paramètres de la Palette

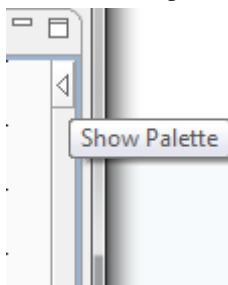
La **Palette** contient tous les composants techniques, les formes ainsi que les branches de base nécessaires à la création de Jobs et de Business Models dans l'espace de modélisation graphique. Ces composants, ces formes ainsi que ces branches sont regroupés par familles et sous-familles.

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

### A.2.1.1. 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.

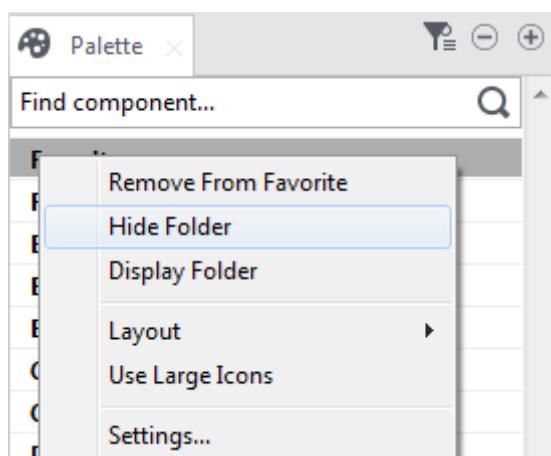


Si vous souhaitez que la **Palette** soit visible en permanence, cliquez sur la flèche de gauche dans le coin supérieur droit de l'éditeur graphique.

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

### A.2.1.2. Afficher/masquer les familles de composants

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





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

### A.2.1.3. 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.

### A.2.1.4. Filtrer la Palette

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

Pour plus d'informations sur comment filtrer la **Palette**, consultez [Personnaliser la Palette](#).

Pour plus d'informations concernant l'ajout de composants à la **Palette**, à partir de **Talend Exchange** ou développés par vous-même, consultez [Télécharger/charger des composants de la communauté](#) et/ou [Définir le dossier de composants utilisateur \(Talend > Components\)](#).

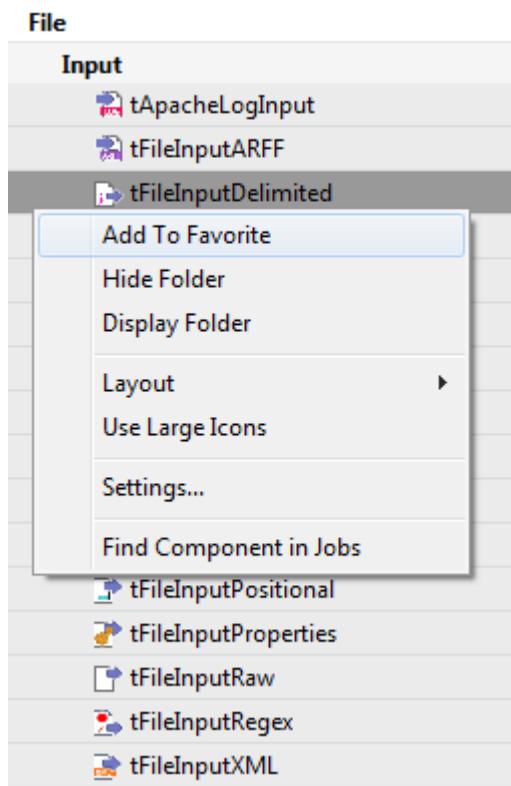
### A.2.1.5. Configurer la Palette favorite

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

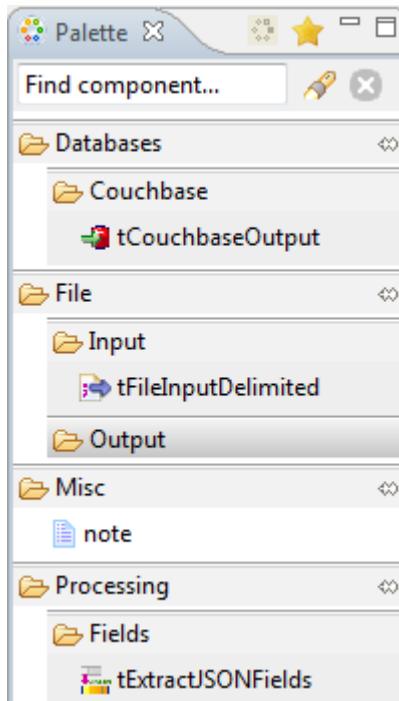
Vous pouvez ajouter et retirer des composants à votre **Palette** favorite, afin d'accéder plus rapidement aux composants que vous utilisez le plus souvent.

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**.

## A.2.1.6. Modifier la disposition des composants dans la Palette

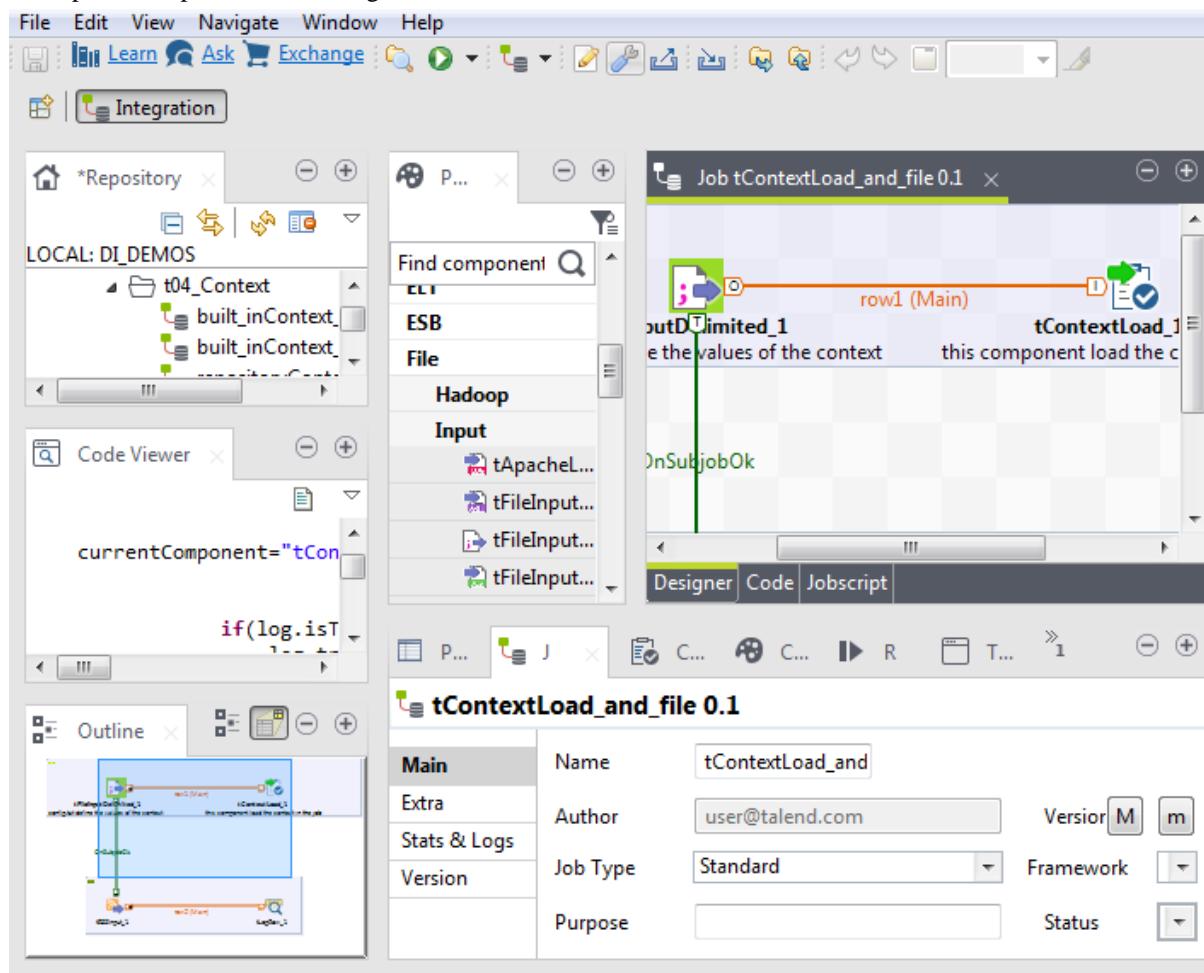
Vous pouvez modifier la disposition de la liste des composants dans la **Palette** pour les afficher en colonnes ou en listes, avec seulement une icône ou une icône avec une courte description.

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.

## A.2.2. Réorganiser les onglets d'un Job

Vous pouvez déplacer tous les onglets selon vos besoins.



Cliquez sur la bordure ou sur l'onglet, maintenez le bouton de la souris enfoncé pendant que vous déplacez la fenêtre vers l'emplacement cible, puis relâchez.

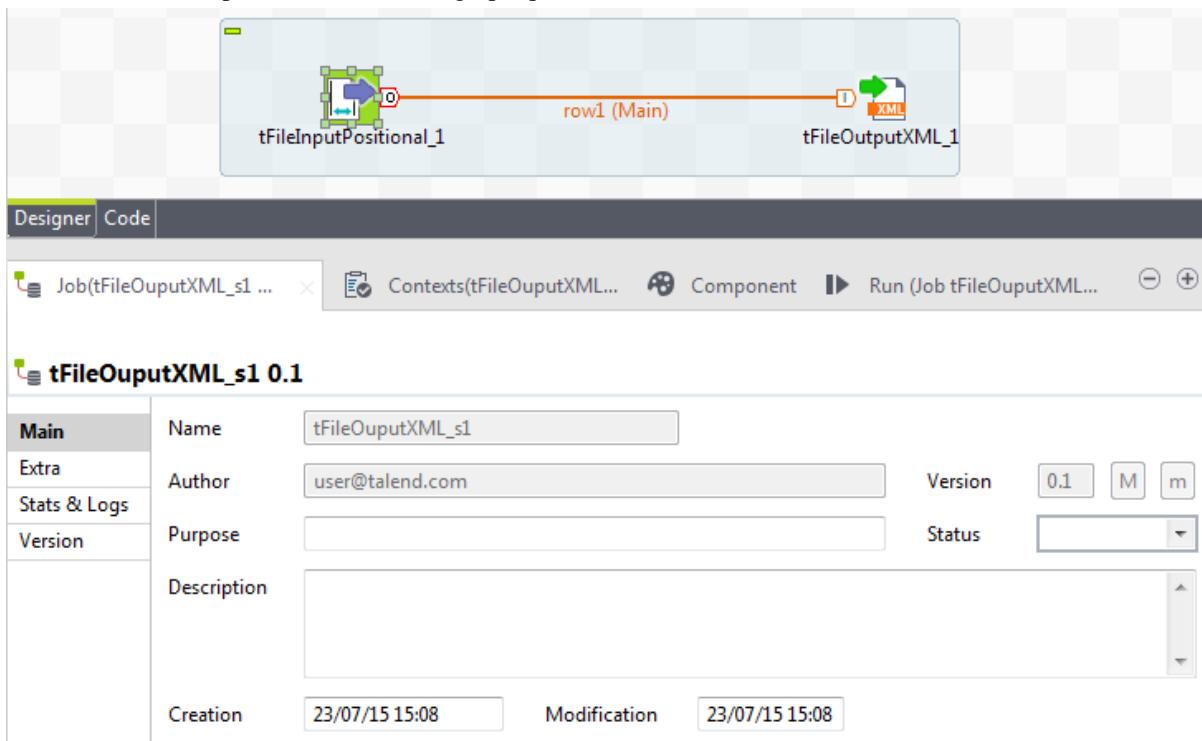
Cliquez sur l'icône minimiser/maximiser (□/□) pour réduire ou agrandir le panneau correspondant. Pour plus d'informations sur comment afficher ou masquer un panneau ou une vue, consultez [Afficher les onglets/vues de configuration des Jobs](#).

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.

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 perspective.

## A.2.3. Afficher les onglets/vues de configuration des Jobs

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



Les onglets **Component**, **Run Job** et **Contextet Oozie scheduler** 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 le *Studio Talend* pour la première fois, l'onglet **Problems** ne s'affichera qu'au moment où vous créez 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 noeud correspondant et sélectionnez l'élément que vous souhaitez afficher.

## A.3. Filtrage des entrées listées dans la vue Repository

Le *Studio Talend* offre la possibilité de choisir les noeuds, les Jobs ou éléments que vous souhaitez lister dans la vue **Repository**.

Vous pouvez filtrer la vue **Repository** par le nom du Job, le statut du Job ou l'utilisateur ayant créé le Job ou les éléments, simplement en cochant ou décochant la case située à côté du noeud ou de l'élément que vous souhaitez afficher ou cacher dans la vue. Vous pouvez aussi définir simultanément plusieurs filtres.

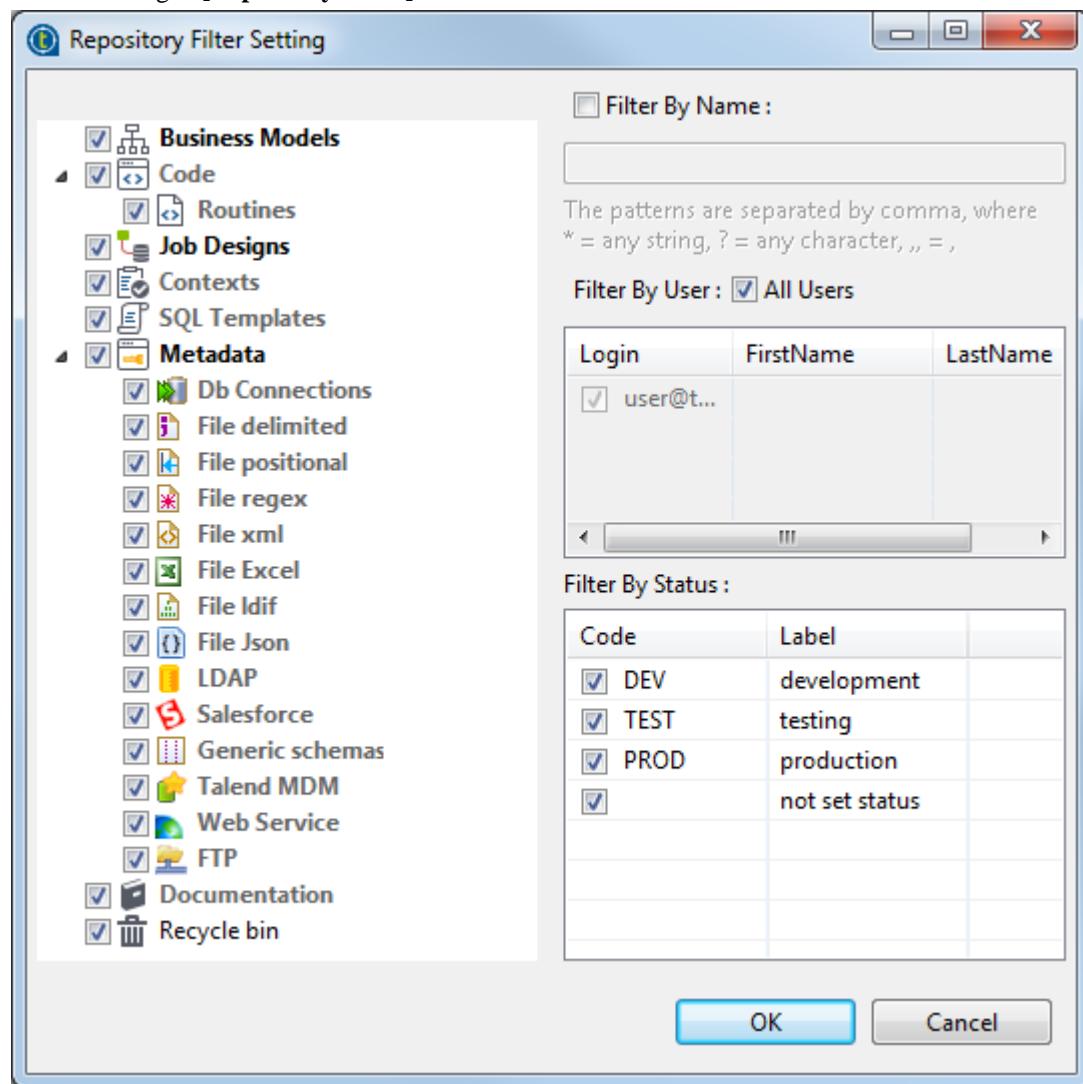
### A.3.1. Filtrer avec le nom du Job

Pour filtrer les Jobs listés dans la vue **Repository** par le nom du Job, procédez comme suit :

1.

Dans le Studio, cliquez-droit sur l'icône  dans la vue **Repository** en haut à droite et sélectionnez **Filter settings** à partir du menu contextuel.

La boîte de dialogue [**Repository Filter**] s'ouvre.



2. Cochez la case **Filter By Name**.

Le champ correspondant devient accessible.

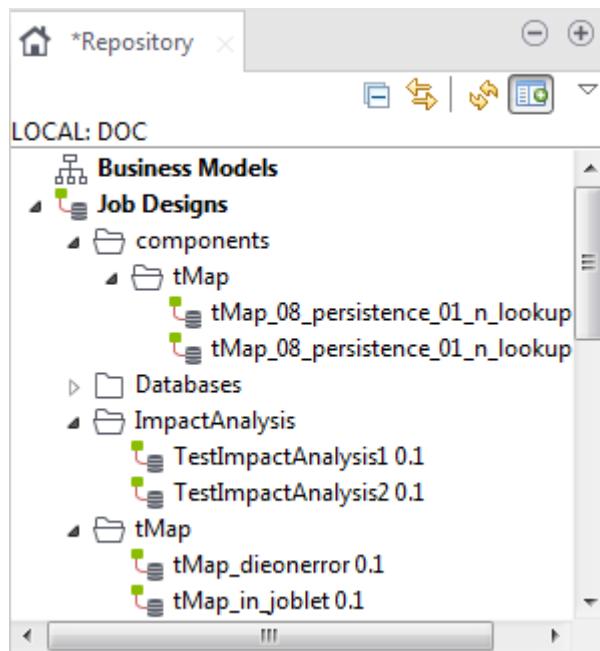
<input checked="" type="checkbox"/>	Filter By Name :
<input type="text" value="tMap*, test*"/>	
The patterns are separated by comma, where * = any string, ? = any character, , = ,	

3. Suivez les règles affichées sous le champ lorsque vous entrez les éléments que vous souhaitez utiliser pour filtrer les Jobs.

L'objectif de cet exemple est de lister tous les Jobs de l'arborescence commençant par *tMap* ou *test*.

- Dans la case **[Repository Filter]**, cliquez sur **OK** pour valider vos modifications, ensuite fermer la boîte de dialogue.

Seuls les Jobs correspondants au filtre que vous avez défini apparaîtront dans l'arborescence, ceux commençant par *tMap* et *test* dans cet exemple.



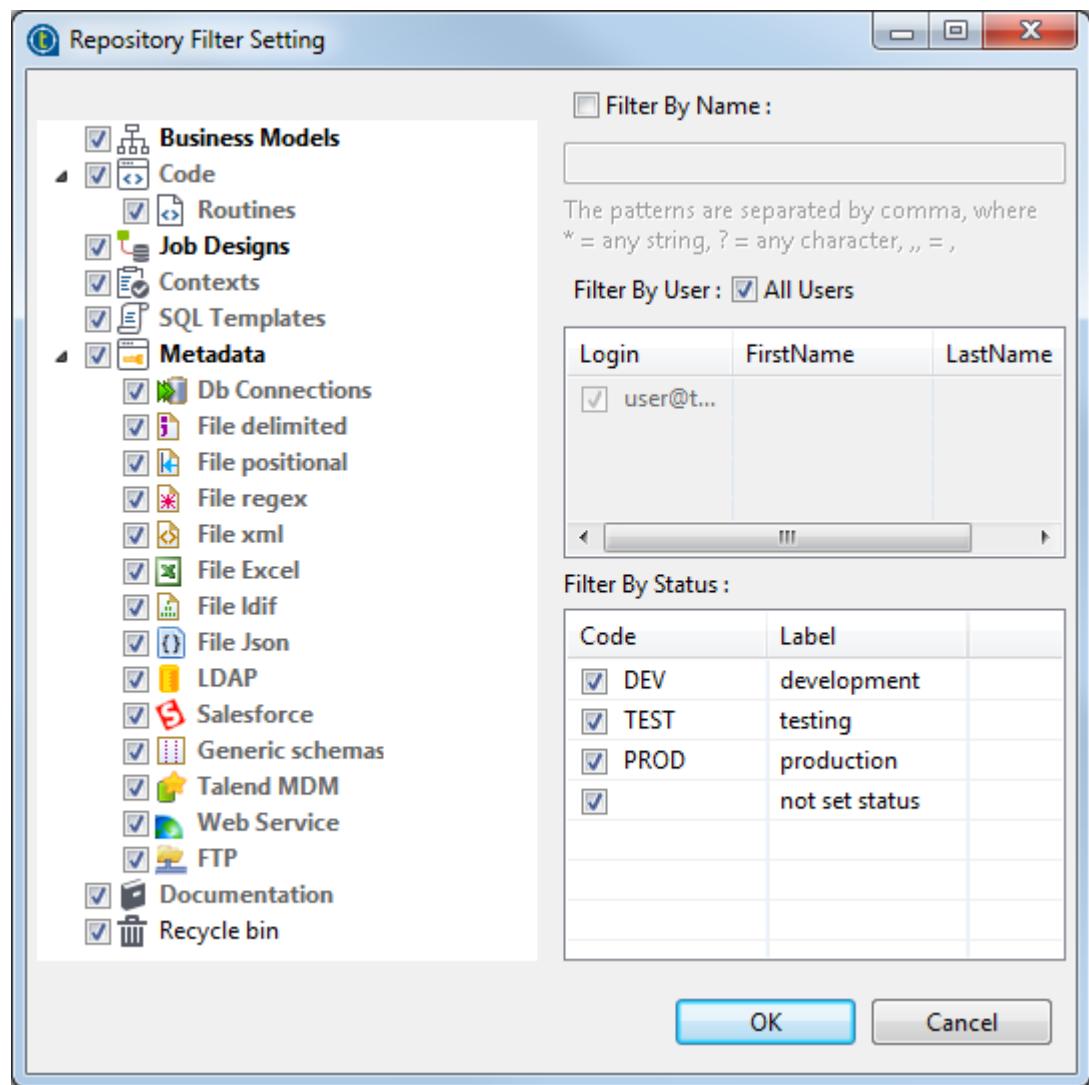
Vous pouvez retourner à l'arborescence par défaut, listant tous les noeuds, Jobs et éléments, simplement en cliquant sur l'icône . Alors le signe plus vert se change en signe moins rouge ().

### A.3.2. Filtrer avec l'utilisateur

Pour filtrer des entrées dans la vue **Repository** avec l'utilisateur qui a créé les Jobs ou les éléments, procédez comme suit :

- Dans la perspective **Integration** du Studio, cliquez-droit sur l'icône dans la vue **Repository** en haut à droite et sélectionnez **Filter settings** à partir du menu contextuel.

La boîte de dialogue **[Repository Filter]** s'ouvre.



2. DÉCOchez la case **All Users**.

Les champs correspondants dans la table ci-dessous deviennent accessibles.

Filter By User : <input type="checkbox"/> All Users		
Login	FirstName	LastName
<input checked="" type="checkbox"/> user@talend.com	User	User
<input checked="" type="checkbox"/> dev1@talend.com	dev1	Talend
<input checked="" type="checkbox"/> dev2@talend.com	dev2	Talend

La table liste les informations authentification de tous les utilisateurs qui se sont connectés au *Studio Talend* et qui ont créé un Job ou un élément.

3. DÉCOchez la case à côté du nom de l'utilisateur si vous souhaitez cacher dans la vue **Repository** tous les Jobs ou éléments créés par l'utilisateur.
4. Cliquez sur **OK** pour valider vos modifications, ensuite fermez la boîte de dialogue.

Tous les Jobs ou éléments créés par l'utilisateur sélectionné disparaîtront de l'arborescence.

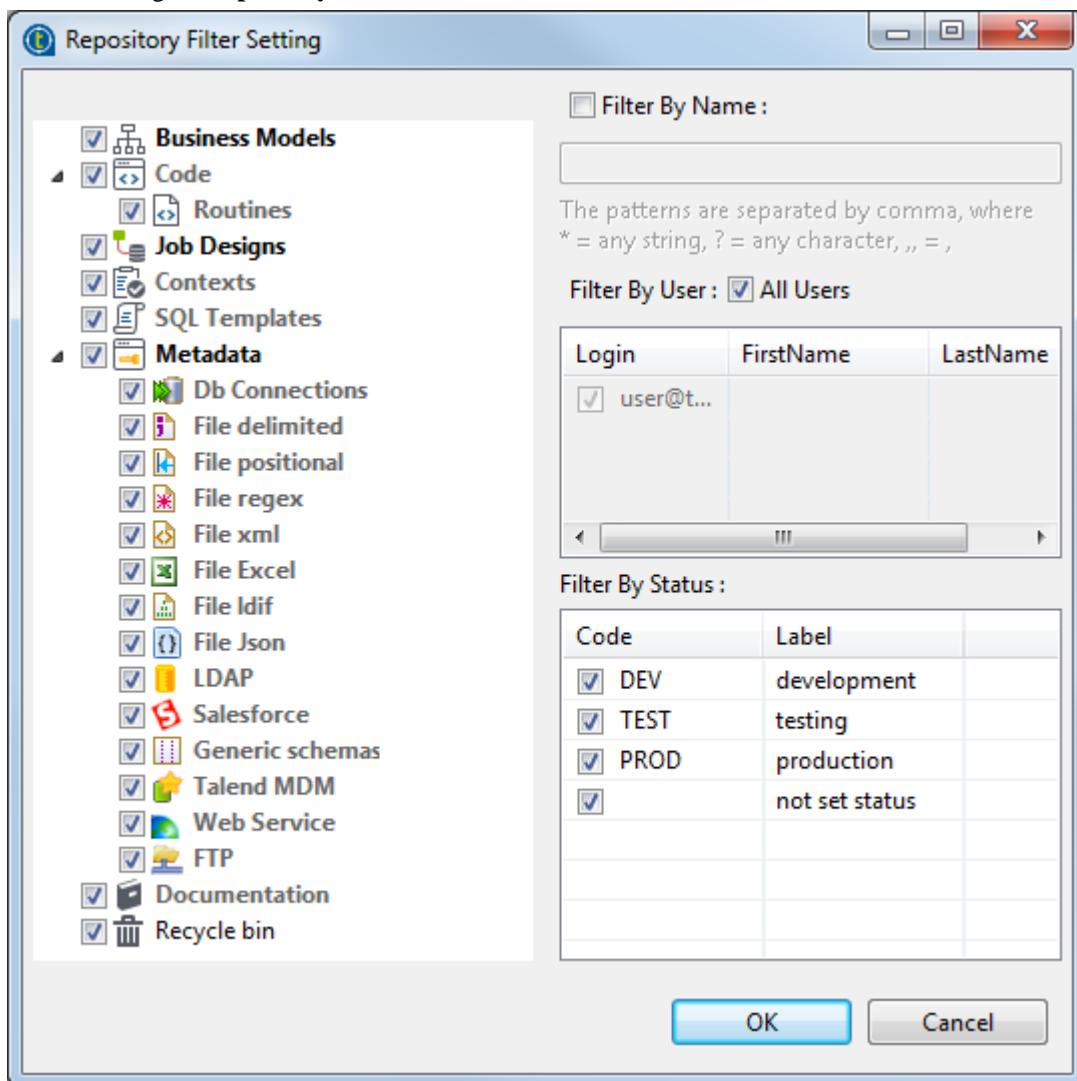
 Vous pouvez retourner à l'arborescence par défaut, listant tous les nœuds, Jobs et éléments, simplement en cliquant sur l'icône . Alors le signe plus vert se change en signe moins rouge ().

### A.3.3. Filtrer avec le statut du Job

Pour filtrer les Jobs dans la vue **Repository** par le statut du Job, procédez comme suit :

1. Dans le Studio, cliquez-droit sur l'icône  dans la vue **Repository** en haut à droite et sélectionnez **Filter settings** à partir du menu contextuel.

La boîte de dialogue **[Repository Filter]** s'ouvre.



2. Dans la zone **Filter By Status**, décochez la case située à côté de statut si vous souhaitez cacher tous les Jobs ayant le statut sélectionné.
3. Cliquez sur **OK** pour valider vos modifications, ensuite fermer la boîte de dialogue.

Tous les Jobs ayant le statut sélectionné disparaîtront de l'arborescence.

 Vous pouvez retourner à l'arborescence par défaut, listant tous les nœuds, Jobs et éléments, simplement en cliquant sur l'icône . Alors le signe plus vert se change en signe moins rouge (.

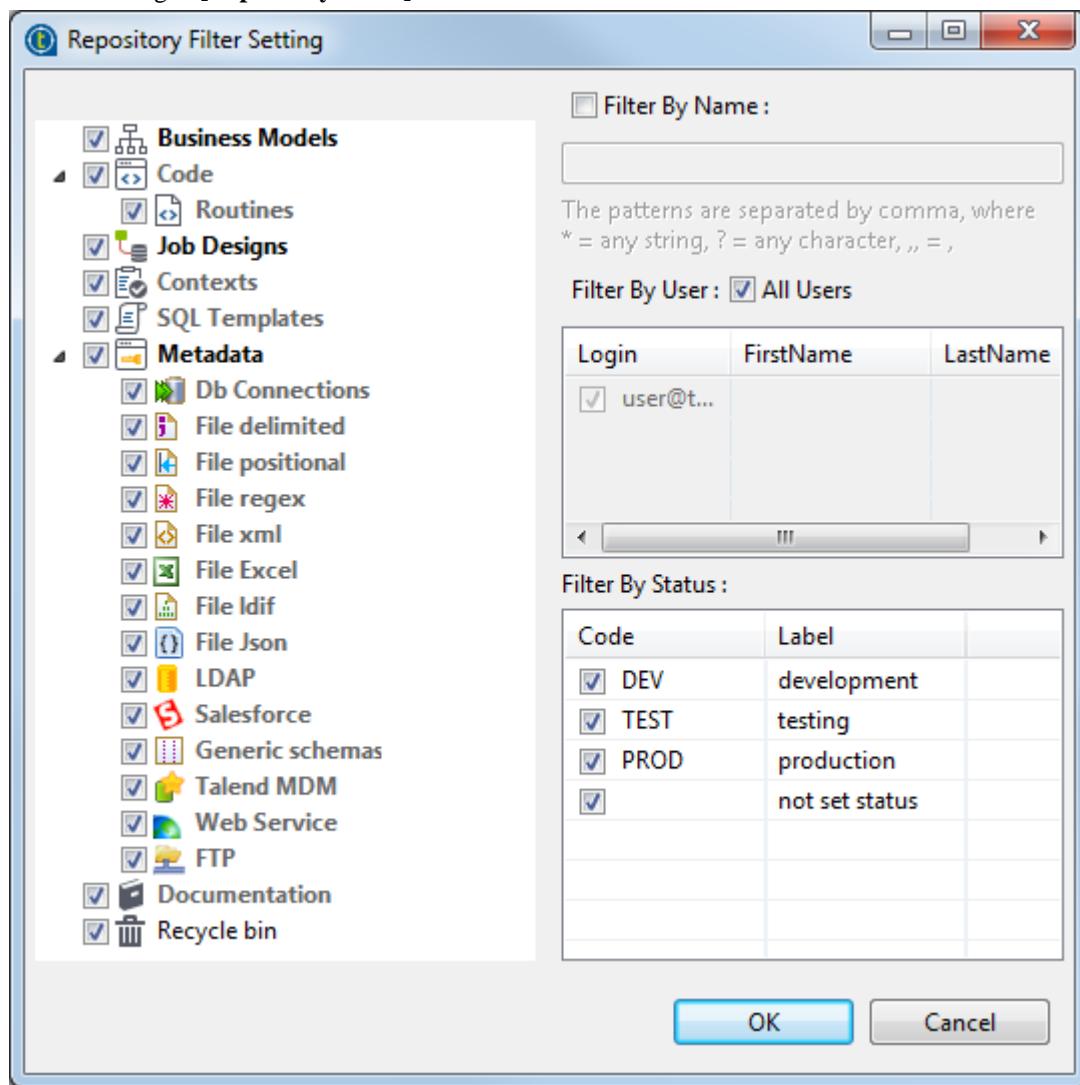
## A.3.4. Sélectionner les nœuds du Repository à afficher

Pour filtrer les nœuds du **Repository**, procédez comme suit :

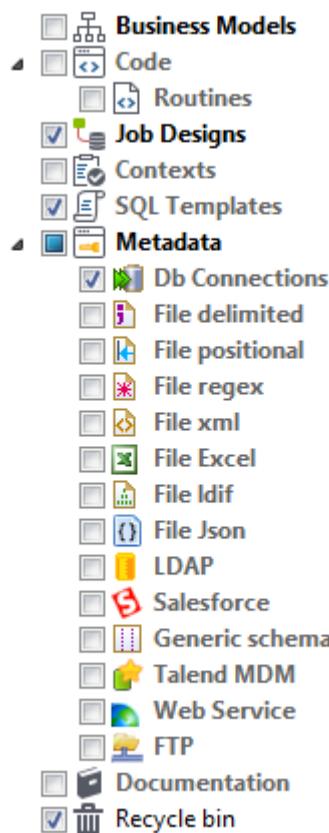
1.

Dans le Studio, cliquez-droit sur l'icône  dans la vue **Repository** en haut à droite et sélectionnez **Filter settings** à partir du menu contextuel.

La boîte de dialogue [**Repository Filter**] s'ouvre.



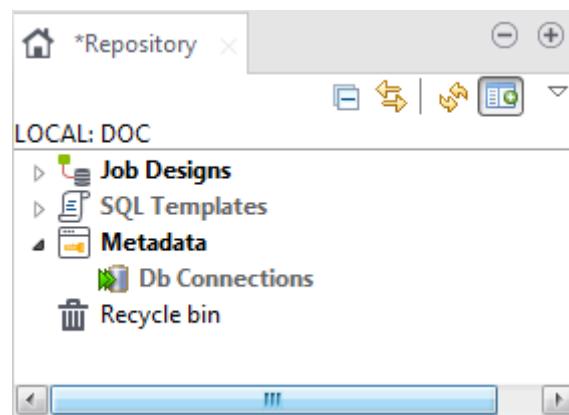
2. Cochez la case située à côté des nœuds que vous souhaitez afficher dans la vue **Repository**.



Par exemple, vous souhaitez montrer dans l'arborescence tous les Jobs listés sous le nœud **Job Designs**, et trois des dossiers listés sous le nœud **SQL Templates** et l'un des éléments des métadonnées listé sous le nœud **Metadata**.

3. Cliquez sur **OK** pour valider vos modifications et fermer la boîte de dialogue.

Seuls les nœuds/dossiers dont vous avez coché la case apparaissent dans l'arborescence.



Si vous ne souhaitez pas afficher tous les Jobs listés sous le nœud **Job Designs**, vous pouvez filtrer les Jobs en cochant la case **Filter By Name**. Pour plus d'informations sur le filtrage des Jobs, consultez [Filtrer avec le nom du Job](#).

## A.4. Configuration des préférences du Studio Talend

Vous pouvez définir plusieurs propriétés d'affichage pour toutes les perspectives du *Studio Talend* pour les adapter à vos besoins et à vos préférences.

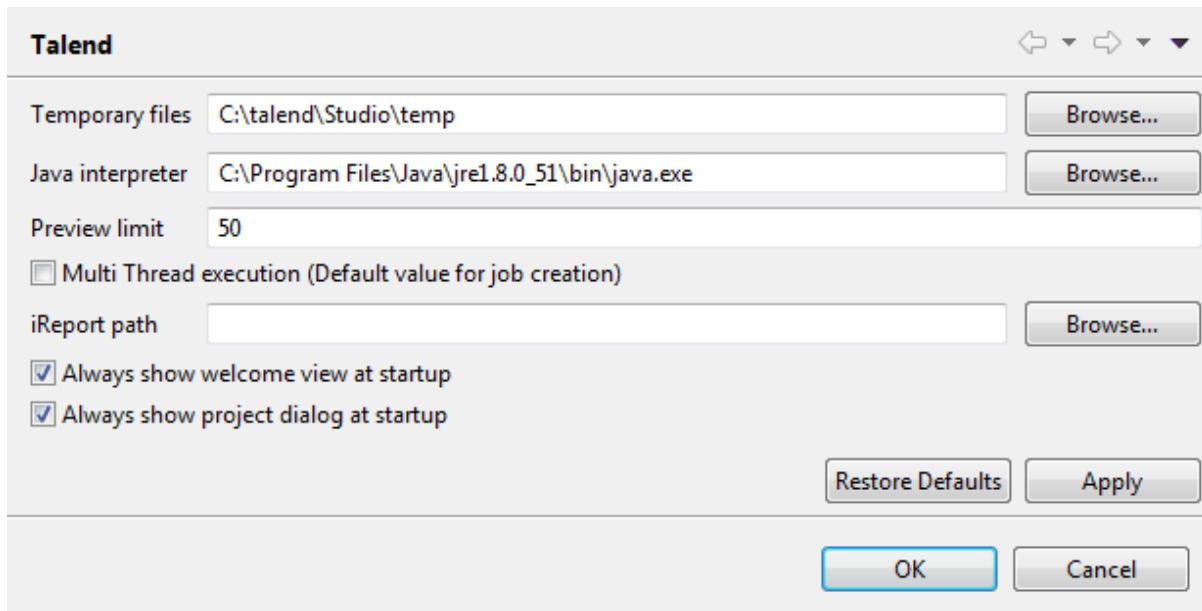
Un grand nombre des configurations que vous avez définies peuvent être enregistrées dans **Preferences** et seront donc utilisées par défaut pour tous les nouveaux Jobs que vous créerez.

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

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

### A.4.1. Interpréteur Java (Talend)

Le chemin de l'interpréteur Java est défini par défaut selon l'emplacement du fichier Java de votre ordinateur (par exemple C:\Program Files\Java\jre1.8.0\_51\bin\java.exe).



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

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.

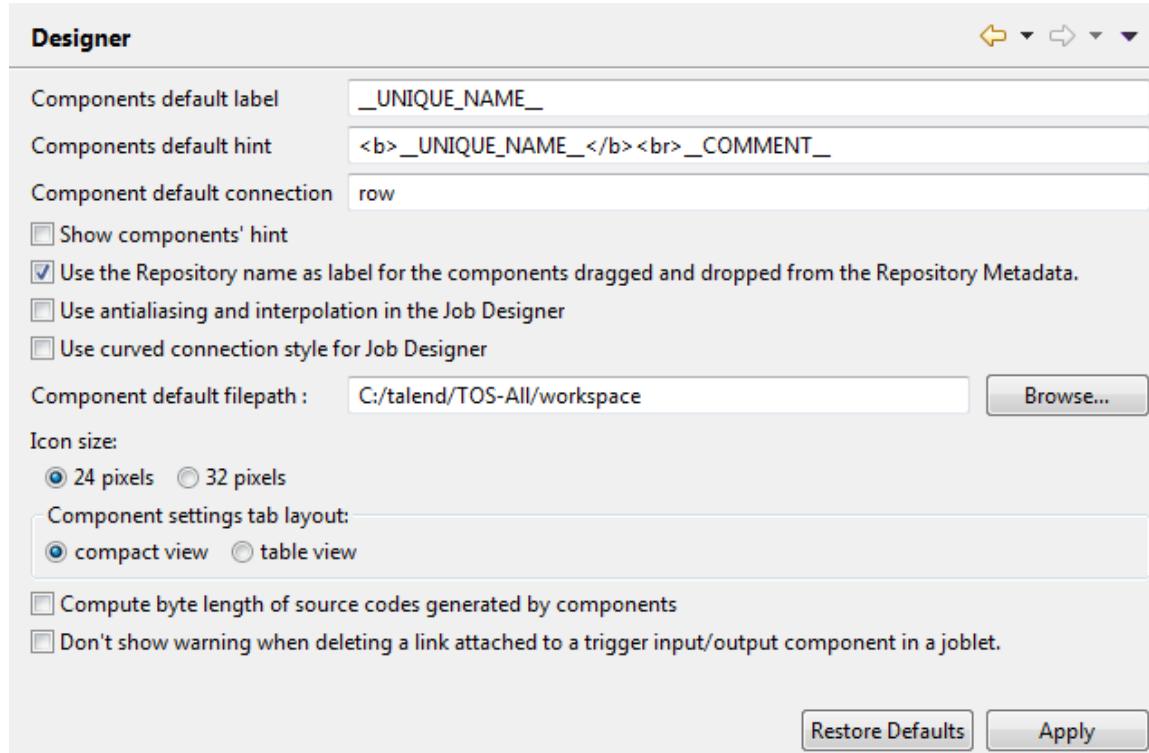
Dans la même vue, vous pouvez également modifier le nombre de lignes de données affichées dans l'aperçu et le chemin d'accès vers les fichiers temporaires.

### A.4.2. Préférences du Designer (Talend > Appearance)

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 du *Studio Talend* en fonction de votre utilisation.

### A.4.3. Définir le dossier de composants utilisateur (Talend > Components)

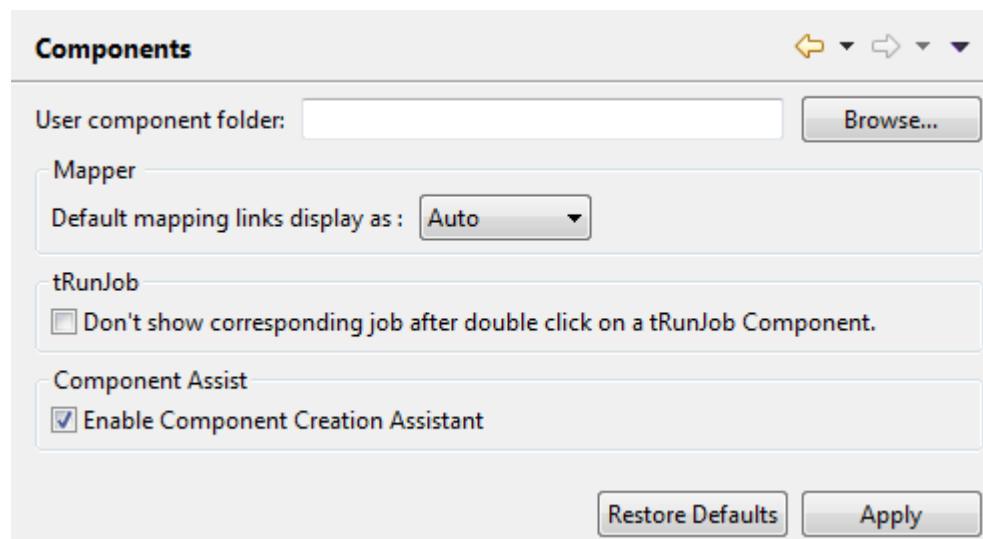
Vous pouvez créer ou développer vos propres composants et les utiliser dans la perspective **Integration** du *Studio Talend*.

Pour plus d'informations concernant la création et le développement de composants utilisateur, consultez *Création d'un composant personnalisé* à l'adresse <https://help.talend.com>.

La procédure suivante s'applique uniquement aux composants externes. Pour les préférences relatives aux autres composants, consultez [Définir les préférences de composants spécifiques \(Talend > Components\)](#).

Le dossier des composants contient les composants que vous créez et/ou ceux que vous téléchargez depuis TalendForge. Pour le définir, procédez comme suit :

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



2. Renseignez le champ **User component folder** pour indiquer le chemin d'accès au dossier contenant les composants à ajouter à la **Palette** du Studio.

Afin qu'ils soient importés dans la **Palette** du Studio, les composants utilisateur doivent être dans des dossiers séparés à la racine du dossier juste défini.

3. Cliquez sur **Apply** puis sur **OK** afin de valider vos préférences et fermer la boîte de dialogue.

Le Studio redémarre et les composants externes sont ajoutés à la **Palette**.

La configuration est stockée dans les métadonnées du workspace. Si le workspace du *Studio Talend* change, vous devez définir cette configuration à nouveau.

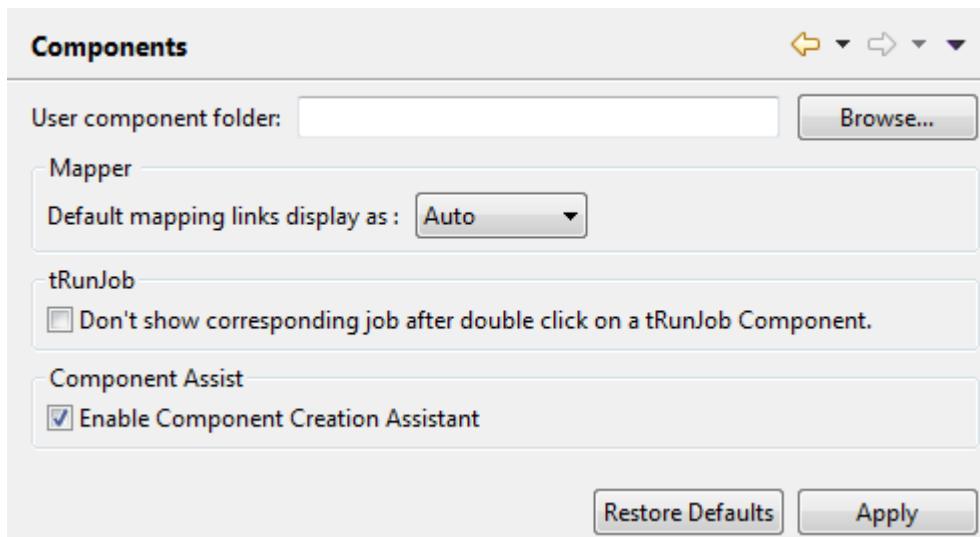
## A.4.4. Définir les préférences de composants spécifiques (Talend > Components)

Vous pouvez modifier certaines préférences de composants spécifiques telles que l'affichage par défaut des links de mapping.

La procédure suivante s'applique aux composants externes et aux composants inclus dans le Studio. Pour les préférences spécifiques aux composants utilisateur, consultez [Définir le dossier de composants utilisateur \(Talend > Components\)](#).

Pour modifier ces paramètres de composants spécifiques, procédez comme suit :

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



2. 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**.
3. 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**.

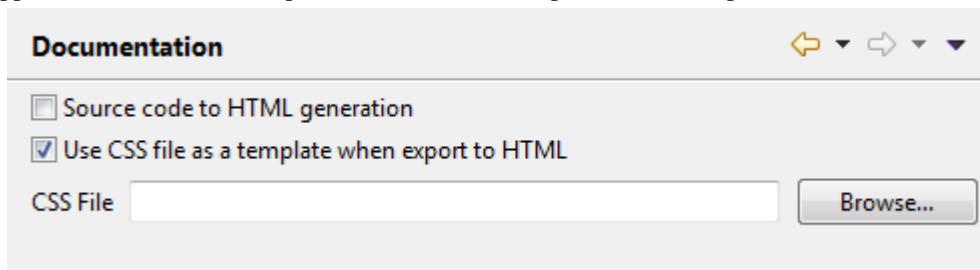
4. Cochez la case **Enable Component Creation Assistant** afin de pouvoir ajouter un composant en saisissant son nom directement dans l'espace de modélisation graphique. Pour plus d'information, consultez [Ajouter des composants au Job](#).
5. Cliquez sur **Apply** puis sur **OK** afin de valider les préférences définies et fermer la boîte de dialogue.

La configuration est stockée dans les métadonnées du workspace. Si le workspace du *Studio Talend* change, vous devez définir cette configuration à nouveau.

## A.4.5. Préférences de la documentation (Talend > 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 nœud **Talend** et cliquez sur **Documentation** pour afficher les préférences de la documentation.



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

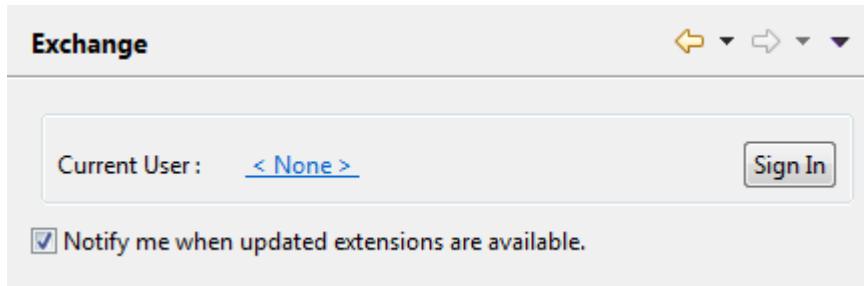
- Cochez la case **Source code to HTML generation** pour intégrer le code source dans la documentation HTML que vous générerez.
- Cochez la case **Use CSS File as a template when export to HTML** afin d'activer le champ **CSS File**, si vous devez utiliser un fichier CSS pour personnaliser les fichiers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

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

## A.4.6. Préférence Exchange (Talend > Exchange)

Vous pouvez configurer les préférences relatives à votre connexion à **Talend Exchange**, site faisant partie de la Communauté **Talend**, dans le *Studio Talend*. 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 :

- 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**.

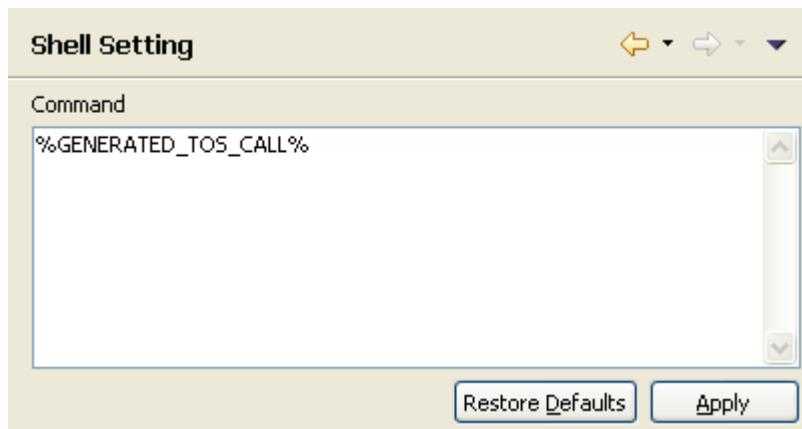
- Par défaut, lorsque vous êtes connecté à la Communauté **Talend**, lorsqu'une mise à jour d'un produit installé est disponible, une boîte de dialogue apparaît pour vous le signaler. Si vous vérifiez fréquemment les mises à jour disponibles et que vous ne souhaitez pas que cette boîte de dialogue réapparaisse,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

Pour plus d'informations concernant la connexion à la Communauté **Talend**, consultez le Guide de prise en main de votre Studio. Pour plus d'informations concernant l'utilisation des composants de la communauté dans le Studio, consultez [Télécharger/charger des composants de la communauté](#).

## A.4.7. Ajouter du code par défaut (Talend > Import/Export)

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 nœuds **Talend** et **Import/Export** et cliquez sur **Shell Settings** pour afficher le champ permettant d'ajouter du code par défaut.

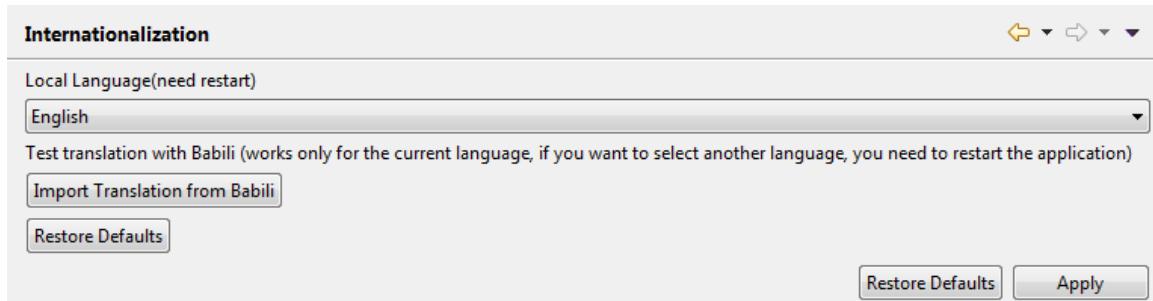


3. Dans le champ **Command**, ajoutez votre/vos ligne(s) de code avant ou après le code %GENERATED\_TOS\_CALL% pour qu'elles apparaissent respectivement au début ou à la fin du code de votre Job.

## A.4.8. Préférences de langue (Talend > Internationalization)

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

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.

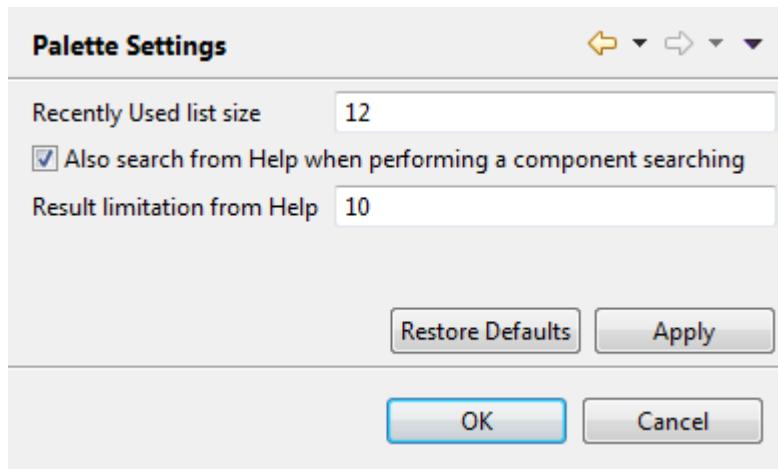


3. Dans le champ **Local Language**, sélectionnez la langue que vous souhaitez utiliser pour l'interface du *Studio Talend*.
4. Cliquez sur **Apply**, puis sur **OK** pour valider vos changements et fermer la boîte de dialogue **[Preferences]**.
5. Redémarrez le *Studio Talend* pour que ce changement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

## A.4.9. Préférences de la Palette (Talend > Palette Settings)

Dans la vue **Palette Settings**, vous pouvez configurer les préférences liées à la recherche de composants dans la **Palette** et même dans la liste qui s'affiche dans l'espace de modélisation graphique lors de l'ajout d'un composant sans utilisation de la **Palette**.

1. Dans le menu, cliquez sur **Window > Preferences** pour afficher la fenêtre [**Preferences**].
2. Développez le nœud **Talend** et cliquez sur **Palette Settings** pour afficher la vue **Palette Settings**.

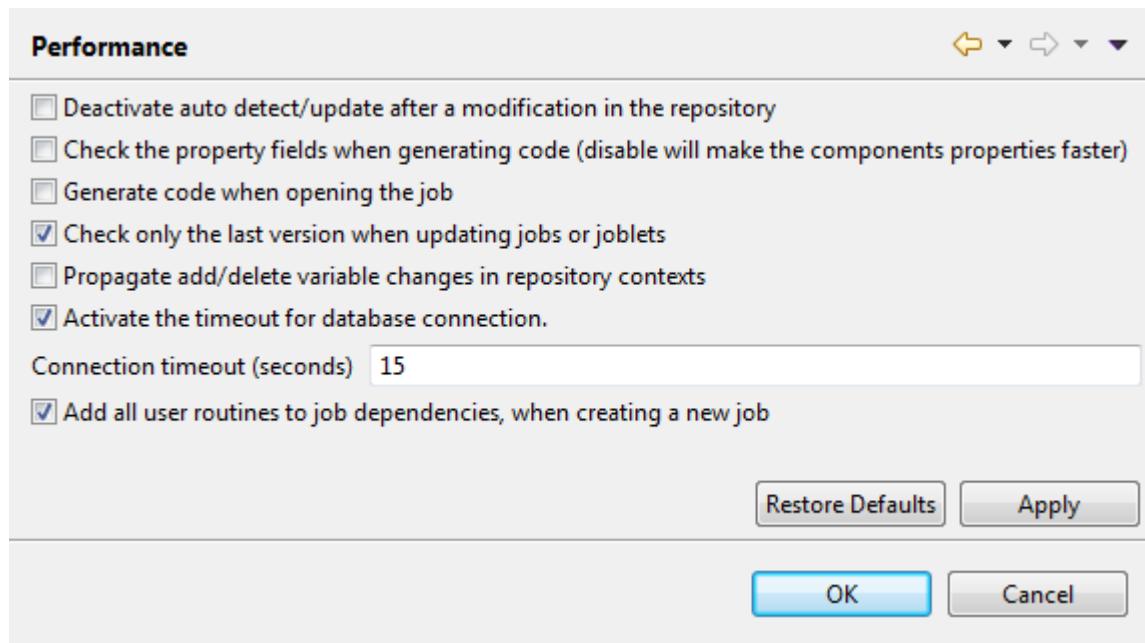


3. Afin de limiter le nombre de composants pouvant être affichés dans la liste **Recently Used**, saisissez le nombre maximal que vous souhaitez afficher dans le champ **Recently used list size**.
4. Pour activer la possibilité de rechercher un composant en saisissant une phrase décrivant sa fonctionnalité ou sa finalité dans le champ de recherche de la **Palette** ou dans le champ de texte s'affichant dans l'espace de modélisation graphique lorsque vous commencez à saisir, cochez la case **Also search from Help when performing a component searching**. Lorsque cette case est cochée, vous pouvez voir votre composant dans la **Palette** ou dans la liste s'affichant dans l'espace de modélisation graphique, tant qu'il est disponible dans l'aide (touche **F1**) lorsque vous saisissez une phrase descriptive incluse dans cette aide.
5. Pour limiter le nombre de résultats de recherche ou en afficher plus, lorsque vous saisissez une phrase dans le champ de recherche, saisissez le nombre de résultats que vous souhaitez voir retournés, dans le champ **Result limitation from Help**.

## A.4.10. Préférences de Performance (Talend > Performance)

Dans les préférences, vous pouvez définir des options de performance en fonction de votre utilisation du *Studio Talend*. Pour ce faire, procédez comme suit :

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 du *Studio Talend* en fonction de votre utilisation :

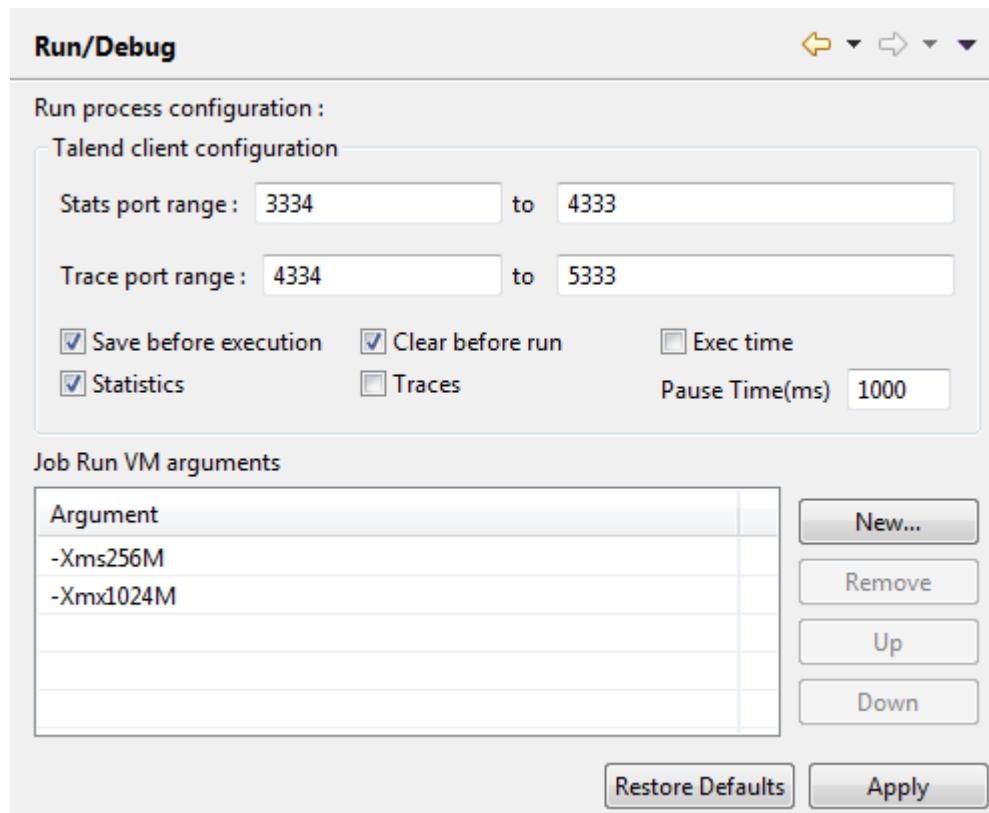
- Cochez la case **Deactivate auto detect/update after a modification in the repository** pour désactiver la détection et la mise à jour automatique du Repository après modification.
- Cochez la case **Check the property fields when generating code** pour activer la vérification des champs de propriétés 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 .
- 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.

## A.4.11. Préférences d'exécution et de débogage (Talend > Run/Debug)

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

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.

<b>Stats port range</b>	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.
<b>Trace port range</b>	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.
<b>Save before run</b>	Cochez cette case pour automatiquement enregistrer votre Job avant de l'exécuter.
<b>Clear before run</b>	Cochez cette case pour nettoyer les résultats d'une exécution précédente avant d'exécuter de nouveau le Job.
<b>Exec time</b>	Cochez cette case pour afficher la durée d'exécution du Job.
<b>Statistics</b>	Cochez cette case le suivi de flux au cours de l'exécution du Job.
<b>Traces</b>	Cochez cette case pour afficher le suivi du traitement des données au cours de l'exécution du Job.
<b>Pause time</b>	Indiquez le temps de pause souhaitée entre chaque ligne de données du tableau Traces.

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

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

Pour plus d'informations concernant les paramètres d'exécution avancés d'un Job spécifique, consultez [Configurer les paramètres d'exécution avancés](#).

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

## A.4.12. Afficher des caractères spéciaux dans les colonnes de schémas (Talend > Specific settings)

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

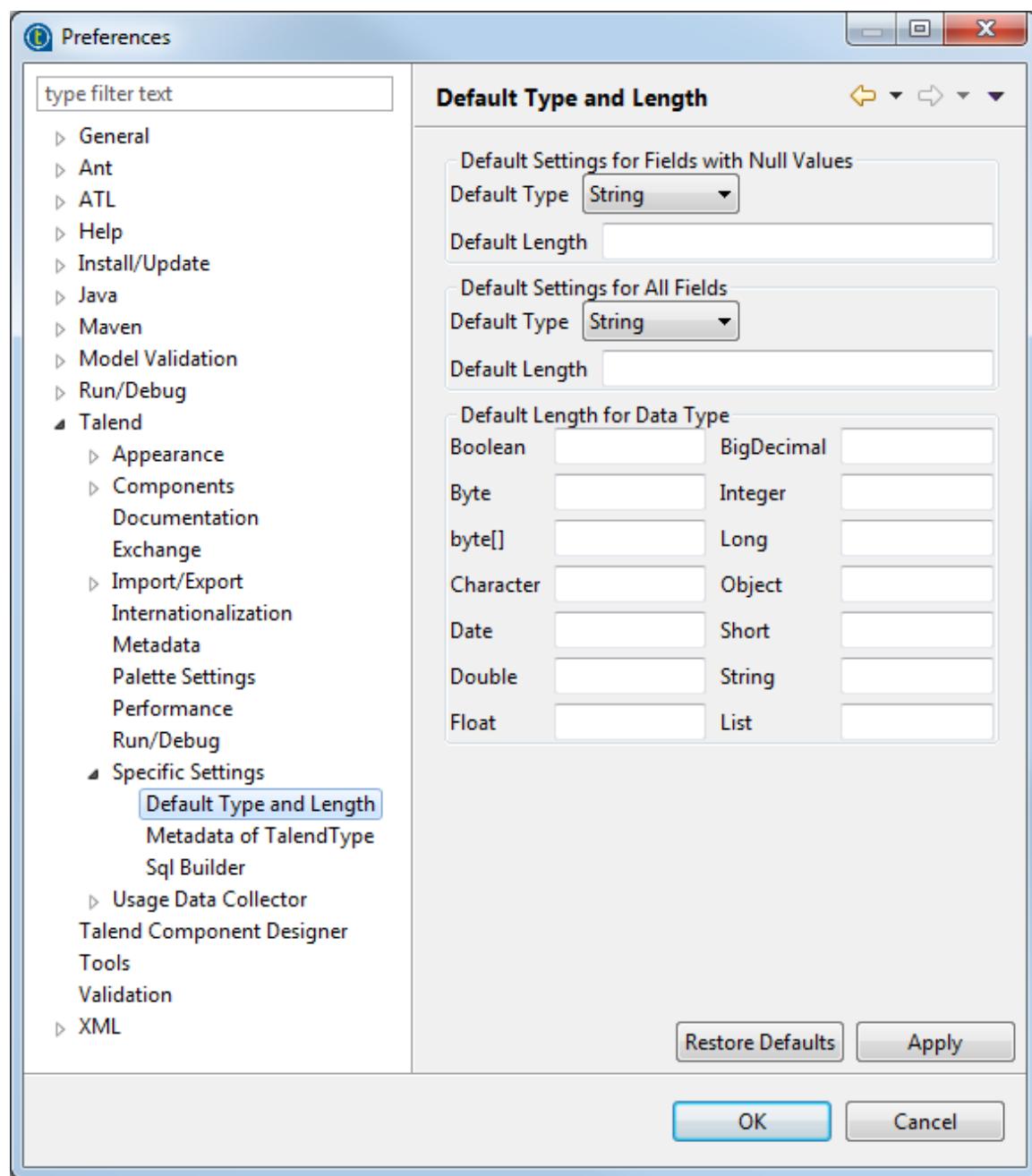
1. Dans la barre de menu, cliquez sur **Window > Preferences** afin d'ouvrir la boîte de dialogue [**Preferences**].
2. Dans l'arborescence de la boîte de dialogue, développez le nœud **Talend**.
3. Cliquez sur le nœud **Specific settings** pour afficher la vue correspondante dans la partie droite de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



## A.4.13. Préférences des schémas (Talend > Specific Settings)

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.



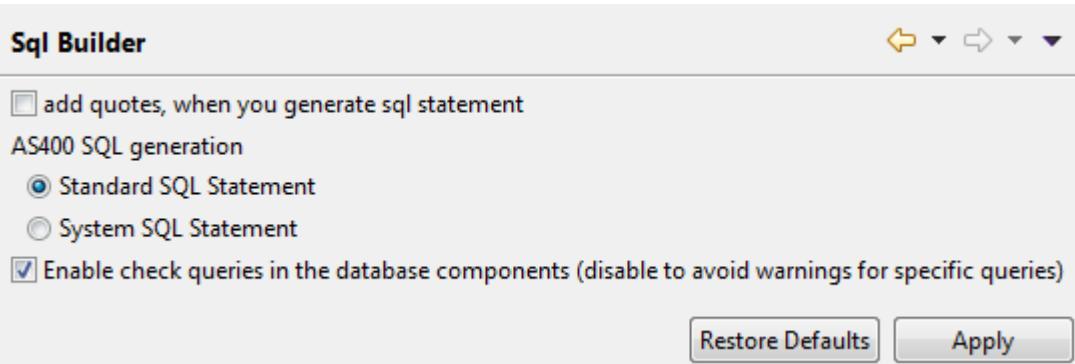
3. Paramétrez ces préférences en fonction de vos besoins :

- 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.

## A.4.14. Préférences du SQL Builder (Talend > Specific Settings)

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 AS/400.
  - DÉCOchez la case **Enable check queries in the database components (disable to avoid warnings for specific queries)** pour désactiver la vérification des requêtes dans les composants base de données.

## A.4.15. Préférences du collecteur de données d'utilisation (Talend > Usage Data Collector)

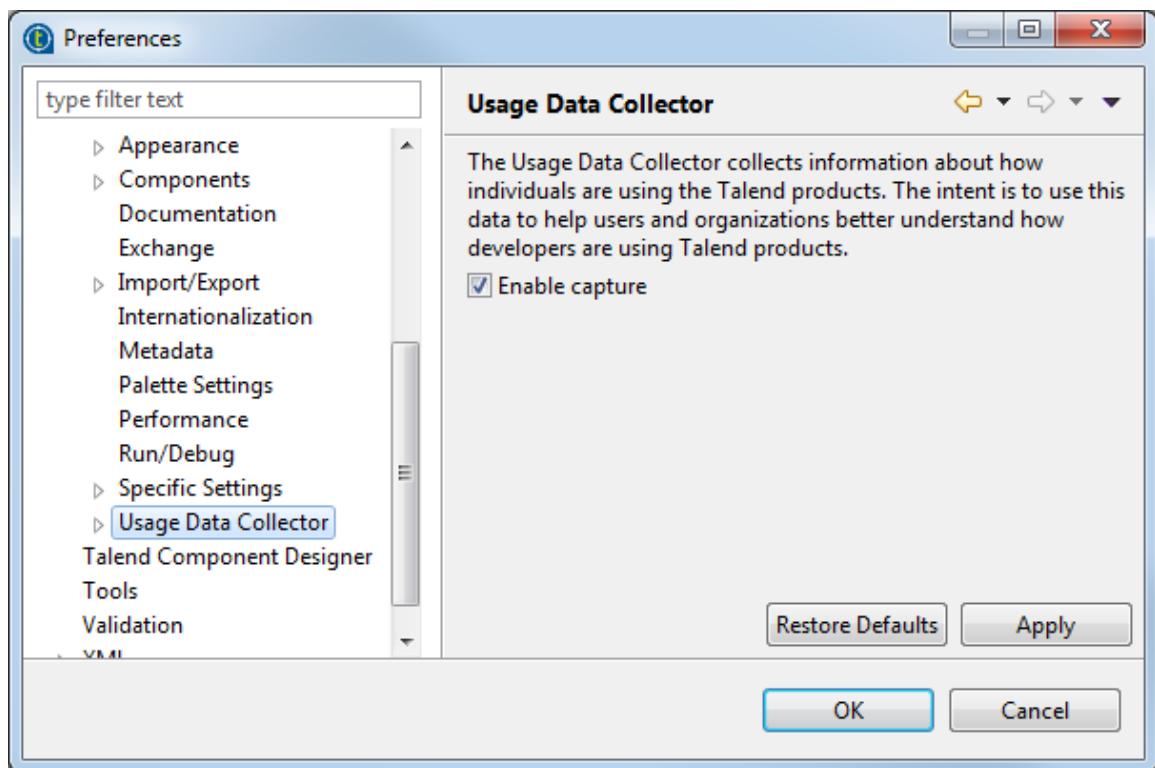
En autorisant le *Studio Talend* à collecter vos statistiques d'utilisation du Studio, vous permettez aux utilisateurs de mieux comprendre les produits **Talend** et vous permettez à **Talend** de savoir comment les utilisateurs utilisent les produits. Cela rend possible l'amélioration de la qualité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

Par défaut, le Studio 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	
Key	Value
tokenStudio	
stopUsageCollection	0
typeStudio	tos_di
uniqueId	cSrRusnn4QmFzwMvoHoAxHP3MR71UtrF3...
version	6.1.0.20150714_1935-SNAPSHOT
properties	
tos.count.businessModels	0
tos.count.componentsPerJob	0.2
tos.count.contextVariablesPerJob	0.0
tos.count.debugRuns	0
tos.count.generatedJobDocs	0
tos.count.jobExports	5
tos.count.jobs	18
tos.count.jobsPerProject	9.0
tos.count.localProjects	2
tos.count.metadata	0
tos.count.runs	5
tos.count.userComponents	0
tos.job.frequentComponents	
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	<input type="text" value="10"/> Days
Last Upload	<input type="text"/>
<input type="button" value="Restore Defaults"/> <input type="button" value="Apply"/>	

- 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*.



## Annexe B. 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 [\*Gestion des routines\*](#).

## B.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>convertImplied DecimalFormat</i>	Retourne un décimal à l'aide d'un modèle décimal implicite.	Numeric.convertImpliedDecimalFormat("Format Cible", valeur à convertir)

Les trois routines *sequence*, *resetSequence* et *removeSequence* sont très proches.

- La routine *sequence* est utilisée pour créer un identifiant de séquence, nommé *s1* par défaut, dans le Job. Cet identifiant de séquence est global dans le Job.
- La routine *resetSequence* peut être utilisée pour initialiser la valeur de l'identifiant de séquence créé par la routine *sequence*.
- La routine *removeSequence* est utilisée pour supprimer l'identifiant de séquence d'une liste de variables globales dans le Job.

### B.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
```

### B.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]
```

## B.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 **Relational**, dans le dossier **system**. La classe **Relational** contient plusieurs routines notamment :

Routine	Description	Syntaxe
<i>ISNULL</i>	Vérifie si la variable donnée est de valeur nulle.  Retourne <i>true</i> si la valeur est NULL et <i>false</i> si la valeur ne l'est pas.	Relational.ISNULL(variable)
<i>NOT</i>	Retourne le complément de la valeur logique d'une expression.	Relational.NOT(expression)
<i>isNull</i>	Vérifie que variable fournie est une valeur null.  Retourne <i>1</i> si la valeur est NULL et <i>0</i> si la valeur n'est pas NULL.	Relational.isNull(variable)

Vous pouvez tester une routine Relational, comme la routine **ISNULL**, à l'aide d'un composant **tJava** par exemple :

```
String str = null;
System.out.println(Relational.ISNULL(str));
```

Dans cet exemple, le résultat de la vérification s'affiche dans la vue **Run** :

```
Starting job test_routine at 14:14 04/02/2010.

[statistics] connecting to socket on port 3375
[statistics] connected
true
[statistics] disconnected
Job test_routine ended at 14:14 04/02/2010. [exit code=0]
```

## B.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
<i>ALPHA</i>	Vérifie si l'expression est triée par ordre alphabétique. Retourne le booléen <i>true</i> si l'ordre alphabétique est vérifié, et <i>false</i> , dans le cas inverse.	StringHandling.ALPHA("chaîne à vérifier")
<i>IS_ALPHA</i>	Vérifie si l'expression ne contient que des caractères alphabétiques. Retourne le booléen <i>true</i> si c'est le cas, et <i>false</i> dans le cas inverse.	StringHandling.IS_ALPHA("chaîne à vérifier")

Routine	Description	Syntaxe
<i>CHANGE</i>	Remplace un élément d'une chaîne de caractères par l'élément de remplacement défini et retourne la nouvelle chaîne.	<code>StringHandling.CHANGE("chaîne à vérifier", "chaîne à remplacer", "chaîne de remplacement")</code>
<i>COUNT</i>	Retourne le nombre d'occurrences d'une sous-chaîne dans une chaîne de caractères.	<code>StringHandling.COUNT("chaîne à vérifier", "chaîne à compter")</code>
<i>DOWNCASE</i>	Convertit toutes les majuscules d'une expression en minuscules et retourne la nouvelle chaîne.	<code>StringHandling.DOWNCASE("chaîne à convertir")</code>
<i>UPCASE</i>	Convertit toutes les minuscules d'une expression en majuscules et retourne la nouvelle chaîne.	<code>StringHandling.UPCASE("chaîne à convertir")</code>
<i>DQUOTE</i>	Entoure une expression de guillemets doubles.	<code>StringHandling.DQUOTE("chaîne à traiter")</code>
<i>EREPLACE</i>	Remplace toutes les sous-chaînes de caractères correspondant à l'expression régulière donnée, dans l'ancienne chaîne de caractères par le remplacement donné et retourne une nouvelle chaîne de caractères.	<code>StringHandling.EREPLACE(oldStr, regex, replacement)</code>
<i>INDEX</i>	Retourne la position, dans une chaîne de caractères, du premier caractère de la sous-chaîne recherchée.  Si la sous-chaîne recherchée n'existe pas dans la chaîne, -1 est retourné.	<code>StringHandling.INDEX("chaîne à vérifier", "sous-chaîne recherchée")</code>
<i>LEFT</i>	Retourne une sous-chaîne correspondant aux n premiers caractères d'une chaîne de caractères.	<code>StringHandling.LEFT("chaîne à vérifier", nombre de caractères)</code>
<i>RIGHT</i>	Retourne une sous-chaîne correspondant aux n derniers caractères d'une chaîne de caractères.	<code>StringHandling.RIGHT("chaîne à vérifier", nombre de caractères)</code>
<i>LEN</i>	Retourne la longueur d'une chaîne de caractères.	<code>StringHandling.LEN("chaîne à vérifier")</code>
<i>SPACE</i>	Retourne une chaîne faite du nombre de caractères vides indiqué.	<code>StringHandling.SPACE(nombre d'espaces à créer)</code>
<i>SQUOTE</i>	Entoure une expression de guillemets simples.	<code>StringHandling.SQUOTE("chaîne à traiter")</code>
<i>STR</i>	Retourne un caractère répété le nombre de fois indiqué.	<code>StringHandling.STR('caractère à générer', nombre de répétition)</code>
<i>TRIM</i>	Supprime les espaces et les tabulations en début et fin d'une chaîne de caractères et retourne la nouvelle chaîne.	<code>StringHandling.TRIM("chaîne à traiter")</code>
<i>BTRIM</i>	Supprime tous les espaces et les tabulations après le dernier caractère non vide d'une chaîne de caractères et retourne la nouvelle chaîne.	<code>StringHandling.BTRIM("chaîne à traiter")</code>
<i>FTRIM</i>	Supprime tous les espaces et les tabulations jusqu'au premier caractère non vide d'une chaîne de caractères et retourne la nouvelle chaîne.	<code>StringHandling.FTRIM("chaîne à traiter")</code>
<i>SUBSTR</i>	Retourne une partie d'une chaîne de caractères. Compte tous les caractères, y compris les blancs, en commençant au début de la chaîne de caractères.	<p><code>StringHandling.SUBSTR(string, start, length)</code></p> <ul style="list-style-type: none"> <li>• <b>string</b> : chaîne de caractères à rechercher.</li> <li>• <b>start</b> : position dans la chaîne de caractères à partir de laquelle commencer à compter.</li> <li>• <b>length</b> : nombre de caractères à retourner.</li> </ul>
<i>LTRIM</i>	Supprime les blancs ou les caractères au début d'une chaîne de caractères.	<p><code>StringHandling.LTRIM(string[, trim_set])</code></p> <ul style="list-style-type: none"> <li>• <b>string</b> : chaîne de caractères à modifier.</li> <li>• <b>trim_set</b> : caractères à supprimer du début de la chaîne. <i>LTRIM</i> va comparer <i>trim_set</i> à la chaîne, caractère par caractère, en commençant par le côté gauche de la chaîne et supprimer les caractères jusqu'à ne plus trouver de caractère correspondant dans <i>trim_set</i>. Si ce paramètre n'est pas spécifié, <i>LTRIM</i> va supprimer tous les caractères blancs du début de la chaîne.</li> </ul>

Routine	Description	Syntaxe
<i>RTRIM</i>	Supprime les blancs ou les caractères à la fin d'une chaîne de caractères.	<pre>StringHandling.RTRIM(string[, trim_set])</pre> <ul style="list-style-type: none"> <li>• <b>string</b> : chaîne de caractères à modifier.</li> <li>• <b>trim_set</b> : caractères à supprimer de la fin de la chaîne. <i>RTRIM</i> va comparer <i>trim_set</i> à la chaîne, caractère par caractère, en commençant par le côté droit de la chaîne et supprimer les caractères jusqu'à ne plus trouver de caractère correspondant dans <i>trim_set</i>. Si ce paramètre n'est pas spécifié, <i>RTRIM</i> va supprimer tous les caractères blancs de la fin de la chaîne.</li> </ul>
<i>LPAD</i>	Convertit une chaîne de caractères en une chaîne d'une longueur spécifiée en ajoutant des blancs ou des caractères au début de la chaîne de caractères.	<pre>StringHandling.LPAD(first_string, length[, second_string])</pre> <ul style="list-style-type: none"> <li>• <b>first_string</b> : chaîne de caractères à modifier.</li> <li>• <b>length</b> : longueur souhaitée pour la chaîne après remplissage.</li> <li>• <b>second_string</b> : caractères à ajouter à gauche de <i>first_string</i>.</li> </ul>
<i>RPAD</i>	Convertit une chaîne de caractères en une chaîne d'une longueur spécifiée, en ajoutant des blancs ou des caractères à la fin de la chaîne.	<pre>StringHandling.RPAD(first_string, length[, second_string])</pre> <ul style="list-style-type: none"> <li>• <b>first_string</b> : chaîne de caractères à modifier.</li> <li>• <b>length</b> : longueur souhaitée pour la chaîne après remplissage.</li> <li>• <b>second_string</b> : caractères à ajouter à droite de <i>first_string</i>.</li> </ul>
<i>INSTR</i>	<p>Retourne la position d'un jeu de caractères dans une chaîne de caractères, en comptant de gauche à droite et à partir de <i>l</i>.</p> <p>Notez que cette routine retourne <i>0</i> si la recherche n'obtient pas de résultat et <i>NULL</i> si la valeur de la recherche est <i>NULL</i>.</p>	<pre>StringHandling.INSTR(string, search_value, start, occurrence)</pre> <ul style="list-style-type: none"> <li>• <b>string</b> : chaîne de caractères à rechercher.</li> <li>• <b>search_value</b> : jeu de caractères à rechercher.</li> <li>• <b>start</b> : position dans la chaîne de caractères à partir de laquelle commencer la recherche. La valeur par défaut est <i>1</i>, ce qui signifie rechercher à partir du premier caractère de la chaîne.</li> <li>• <b>occurrence</b> : occurrences à rechercher.</li> </ul> <p>Par exemple, <code>StringHandling.INSTR("Talend Technology", "e", 3, 2)</code> démarre la recherche au troisième caractère <i>l</i> et retourne <i>7</i>, la position du second caractère <i>e</i>.</p>
<i>TO_CHAR</i>	Convertit des valeurs numériques en chaînes de caractères textuelles.	<pre>StringHandling.TO_CHAR(numeric_value)</pre>

### B.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.IS_ALPHA("ab33cd"));
```

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
```

## B.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 :

La vérification retourne un booléen.

```
Starting job routine1 at 11:45 23/02/2010.
[statistics] connecting to socket on port 3892
[statistics] connected
false
[statistics] disconnected
Job routine1 ended at 11:45 23/02/2010. [exit code=0]
```

## B.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!
```

## B.3.4. Exemple de vérification de chaîne

Vous pouvez tester simplement la routine **INDEX**, à l'aide d'un composant **tJava** par exemple, pour vérifier si la chaîne contient ou pas un caractère ou une chaîne de caractères :

```
System.out.println(StringHandling.INDEX("hello world!", "hello"));
System.out.println(StringHandling.INDEX("hello world!", "world"));
System.out.println(StringHandling.INDEX("hello world", "!"));
System.out.println(StringHandling.INDEX("hello world", "?"));
```

La routine retourne un entier qui correspond à la position du premier caractère de la chaîne recherchée, ou retourne -1 si la chaîne recherchée n'a pu être trouvée :

```
Starting job routine1 at 15:47 24/02/2010.
[statistics] connecting to socket on port 4027
[statistics] connected
0
6
11
-1
[statistics] disconnected
Job routine1 ended at 15:47 24/02/2010. [exit code=0]
```

## B.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

## B.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]
```

## B.4. Routines de type TalendDataGenerator

Les routines de génération de données factices sont des fonctions qui permettent de générer des ensembles de données de test. Elles se basent sur les listes (factices) de noms, prénoms, adresses, villes et Etats fournies par **Talend**. Ces routines sont généralement utilisées au moment du développement des Jobs, à l'aide d'un **tRowGenerator** par exemple, pour éviter d'utiliser des données de production ou de l'entreprise.

Pour accéder à ces routines, double-cliquez sur la classe **TalendDataGenerator** du dossier **system** :

Routine	Description	Syntaxe
<code>getFirstName</code>	retourne un prénom pris aléatoirement dans une liste factice de prénoms américains courants.	<code>TalendDataGenerator.getFirstName()</code>
<code>getLastName</code>	retourne un nom de famille pris aléatoirement dans une liste factice de noms de famille américains courants.	<code>TalendDataGenerator.getLastName()</code>
<code>getUsStreet</code>	retourne une adresse prise aléatoirement dans une liste factice de rues américaines courantes.	<code>TalendDataGenerator.getUsStreet()</code>
<code>getUsCity</code>	retourne une ville prise aléatoirement dans une liste de villes connues des Etats-Unis.	<code>TalendDataGenerator.getUsCity()</code>
<code>getUsState</code>	retourne le nom d'un Etat pris aléatoirement dans une liste des Etats américains.	<code>TalendDataGenerator.getUsState()</code>
<code>getUsStateId</code>	retourne le code pris aléatoirement dans une liste d'abréviations correspondant aux Etats américains.	<code>TalendDataGenerator.getUsStateId()</code>



Aucun paramètre d'entrée n'est attendu car la liste des données factices est fournie par **Talend**.

Vous pouvez personnaliser les données factices en modifiant les routines de génération de données. Pour plus d'informations concernant la personnalisation des routines, consultez [Personnalisation des routines système](#).

## B.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.printIn(TalendDataGenerator.getFirstname());
System.out.printIn(TalendDataGenerator.getLastname());
System.out.printIn(TalendDataGenerator.getUsCity());
System.out.printIn(TalendDataGenerator.getUsState());
System.out.printIn(TalendDataGenerator.getUsStateId());
System.out.printIn(TalendDataGeneraor.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]
```

## B.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 antérieure et -1 si la deuxième est antérieure.	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")
<i>diffDateFloor</i>	Retourne la différence entre deux dates, en nombre d'années, mois, jours, heures, minutes, secondes ou millisecondes selon le paramètre de comparaison spécifié.	TalendDate.diffDateFloor(Date1(), Date2(), "format de données à comparer - ex MM")
<i>formatDate</i>	Retourne une expression de type date formatée selon le modèle date spécifié.	TalendDate.formatDate("format de date - ex : yyyy-MM-dd HH:mm:ss ", Date() à formater)

Routine	Description	Syntaxe
<i>formatDateLocale</i>	Transforme une date en une chaîne de caractères de type date/heure selon le modèle et la locale spécifiés. Retourne la nouvelle date formatée.	TalendDate.formatDateLocale("format cible", java.util.Date date, "code de la langue ou du pays")
<i>getCurrentDate</i>	Retourne la date courante. Aucun paramètre d'entrée n'est attendu.	TalendDate.getCurrentDate()
<i>getDate</i>	Retourne les date et heure courantes dans le format spécifié (optionnel). Cette chaîne peut contenir des chaînes de caractères fixes, ou des variables liées à la date. Par défaut, le format de la chaîne de caractères attendu est DD/MM/CCYY.	TalendDate.getDate("Format de chaîne de caractère - ex : CCYY-MM-DD")
<i>getFirstDayOfMonth</i>	Change le jour d'une date en premier jour du mois courant et retourne la nouvelle date.	TalendDate.getFirstDayMonth(Date)
<i>getLastDayOfMonth</i>	Change le jour d'une date en dernier jour du mois courant et retourne la nouvelle date.	TalendDate.getLastDayMonth(Date)
<i>getPartOfDay</i>	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.	TalendDate.getPartOfDay("Chaîne de caractères indiquant la partie de la date à récupérer", "Chaîne de caractères au format date à parser")
<i>getRandomDate</i>	Retourne une date aléatoire, au format ISO.	TalendDate.getRandomDate("Chaîne de caractère de type Date, de ", String maxDate)
<i>isDate</i>	Vérifie si l'expression est de type Date et correspond au modèle spécifié. Retourne le booléen true si c'est le cas, et false dans le cas inverse.	TalendDate.isDate(Date() à vérifier, "format de date cible - ex : yyyy-MM-dd HH:mm:ss")
<i>parseDate</i>	Transforme une chaîne de caractères en Date. Retourne une date formatée en standard.	TalendDate.parseDate("format date de la chaîne à parser", "Chaîne de caractères au format date à parser")
<i>parseDateInUTC</i>	modifie une chaîne de caractères en une Date UTC. Retourne une date au format UTC.  Dans un Job Big Data s'exécutant dans un environnement de production, comme plusieurs machines dans différents fuseaux horaires sont utilisées, vous devez toujours utiliser cette routine afin de garantir une sortie correcte du parsing de l'heure et de la date.	TalendDate.parseDateInUTC("format date de la chaîne à parser", "chaîne de caractères au format date à parser", "booléen indiquant si le parsing est lenient, c'est-à-dire accepte la correspondance heuristique avec le format")
<i>parseDateLocale</i>	Parse une chaîne de caractères correspondant à un modèle spécifié, et en extrait une date. Retourne une date formatée selon la locale spécifiée.	TalendDate.parseDateLocale("format date de la chaîne à parser", "Chaîne de caractères au format date à parser", "code de la langue ou du pays")
<i> setDate</i>	Modifie une partie de la date en entrée par un entier spécifié et basé sur le format spécifié.	TalendDate.setDate(Date, entier n, "format de la donnée à changer - ex :yyyy")
<i>TO_CHAR</i>	Convertit une date en une chaîne de caractères.	TalendDate.TO_CHAR(date[,format])  • <b>date</b> : valeur de la date à convertir en une chaînes de caractères.  • <b>format</b> : chaîne de caractères définissant le format de la valeur à retourner.
<i>TO_DATE</i>	Convertit une chaîne de caractères en un type Date/Time.	TalendDate.TO_DATE(string[, format])  • <b>string</b> : chaîne de caractères à convertir en type Date/Time.  • <b>format</b> : chaîne de caractères de format correspondant à la partie de l'argument <i>string</i> . Si elle n'est pas spécifiée, la valeur <i>string</i> doit être au format de date MM/dd/yyyy HH:mm:ss.SSS.

Routine	Description	Syntaxe
<i>ADD_TO_DATE</i>	Ajoute une somme spécifiée à une partie d'une valeur datetime et retourne une date au même format que la date passée à la fonction.	<p>Par exemple, <code>TalendDate.TO_DATE("04/24/2017 13:55:42.123")</code> retourne <i>Mon Apr 24 13:55:42 CST 2017</i>.</p> <p><code>TalendDate.ADD_TO_DATE(date, format, amount)</code></p> <ul style="list-style-type: none"> <li>• <b>date</b> : valeur de la date à modifier.</li> <li>• <b>format</b> : chaîne de caractères de format spécifiant la partie de la valeur de date à modifier. <ul style="list-style-type: none"> <li>• Chaînes de caractères de format valide pour les années : <i>Y, YY, YYY</i> et <i>YYYY</i>.</li> <li>• Chaînes de caractères de format valide pour les mois : <i>MONTH, MM</i> et <i>MON</i>.</li> <li>• Chaînes de caractères de format valide pour les jours : <i>D, DD, DDD, DAY</i> et <i>DY</i>.</li> <li>• Chaînes de caractères de format valide pour les heures : <i>HH, HH12</i> et <i>HH24</i>.</li> <li>• Chaînes de caractères de format valide pour les minutes : <i>MI</i>.</li> <li>• Chaînes de caractères de format valide pour les secondes : <i>SS</i>.</li> <li>• Chaînes de caractères de format valide pour les millisecondes : <i>MS</i>.</li> </ul> </li> <li>• <b>amount</b> : la valeur de l'entier spécifiant la somme des années, mois, jours, heures, etc via laquelle vous souhaitez modifier la valeur de la date.</li> </ul> <p>Par exemple,</p> <p>si <code>TalendDate.getCurrentDate()</code> retourne <i>Mon Apr 24 14:26:03 CST 2017</i>,</p> <p><code>TalendDate.ADD_TO_DATE(TalendDate.getCurrentDate(), "YY", 1)</code> va retourner <i>Tue Apr 24 14:26:03 CST 2018</i>.</p>

## B.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.format("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]
```

## B.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 :

Un booléen s'affiche

```
System.out.println(TalendDate.isDate("2010-02-09 00:00:00", "yyyy-MM-dd
HH:mm:ss"));
```

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]
```

### B.5.3. Exemple de comparaison de Dates

Vous pouvez tester simplement la routine **compareDate**, à l'aide d'un composant **tJava** afin de comparer deux dates pour, par exemple, vérifier si la date du jour est identique, antérieure ou ultérieure à une date spécifiée, selon le format défini :

```
System.out.println(TalendDate.compareDate(new Date(),
TalendDate.parseDate("yyyy-MM-dd", "2010/11/24", "yyyy-MM-dd"));
```

Dans cet exemple, 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 antérieure à la deuxième date :

```
Starting job routine1 at 18:09 25/02/2010.

-1
Job routine1 ended at 18:09 25/02/2010. [exit code=0]
```

### B.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]
```

### B.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]
```

## B.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.getMonth(D));
System.out.println(TalendDate.getYear(D));
System.out.println(TalendDate.getDayOfYear(D));
System.out.println(TalendDate.getDayOfWeek(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.

## B.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]
```

## B.6. Routines de type TalendString

Les routines de traitement de chaînes de caractères **Talend** permettent d'effectuer diverses opérations sur des expressions alphanumériques.

Pour accéder à ces routines, double-cliquez sur la classe **TalendString** du dossier **system**. La classe **TalendString** contient notamment les routines suivantes :

Routine	Description	Syntaxe
<i>replaceSpecialCharForXML</i>	retourne une chaîne de caractères où les caractères spéciaux (ex : <, >, &...) ont été remplacés par des caractères XML équivalents.	TalendString.replaceSpecialCharForXML ("chaîne de caractères contenant les caractères spéciaux - ex: Thelma & Louise")
<i>checkCDATAForXML</i>	identifie les chaînes de caractères commençant par <! [CDATA[ et se terminant par ]]> en tant que XML et les retourne sans modification. Transforme les chaînes identifiées comme non-XML sous une forme compatible XML et les retourne ainsi formatées.	TalendString.checkCDATAForXML("chaîne de caractères à parser")
<i>talendTrim</i>	parse la chaîne de caractères en entrée et en retire le/les caractères de remplissage en début ou en fin de chaîne selon la valeur d'alignement spécifiée: -1 pour les caractères de remplissage de fin de chaîne, 1 pour ceux de début de chaîne et 0 pour les deux.  Puis retourne la chaîne nettoyée.	TalendString.talendTrim("chaîne de caractères à parser", "caractère de remplissage à retirer", position du caractère)
<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)

### B.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]
```

### B.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.printIn(TalendString.talendTrim("**talend open studio****",
```

```
'*' , -1));
System.out.printIn(TalendString.talendTrim("**talend open studio****",
'*', 1));
System.out.printIn(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]
```

### B.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.printIn(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]
```

## B.7. Routines TalendStringUtil

La classe **TalendStringUtil** contient une seule routine **DECODE** vous permettant de rechercher une valeur dans un port. Pour accéder à la routine, double-cliquez sur **TalendStringUtil** dans le dossier **system**.

Routine	Description	Syntaxe
<b>DECODE</b>	Recherche un port pour une valeur que vous spécifiez. Si la fonction trouve la valeur, elle retourne une valeur de résultat, que vous définissez. Vous pouvez construire un nombre illimité de recherches dans une fonction DECODE	<pre>TalendStringUtil.DECODE(value, defaultValue, search1, result1[, search2, result2]...)</pre> <ul style="list-style-type: none"> <li>• <b>value</b> : valeur à rechercher.</li> <li>• <b>defaultValue</b> : valeur à retourner si la recherche ne trouve pas de valeur correspondante. La valeur par défaut peut être configurée à <i>null</i>.</li> <li>• <b>search</b> : valeur à rechercher. La valeur de la recherche doit être du même type de données que l'argument <i>value</i>.</li> <li>• <b>result</b> : valeur à retourner si la recherche trouve une valeur correspondante.</li> </ul>

Voici un exemple d'utilisation de la routine **DECODE** avec un composant **tJava**. Vous devez ajouter un composant **tJava** à un nouveau Job, puis saisir le code suivant, qui recherchera la valeur pour *10*, dans le champ **Code** de l'onglet **Basic settings** du composant **tJava**.

```
TalendStringUtil<Integer, String> example = new TalendStringUtil<Integer, String>();
System.out.println(example.DECODE(10, "error", 5, "five", 10, "ten", 15, "fifteen", 20,
"twenty"));
```

Vous devez créer un nouvel objet de type **TalendStringUtil**. Il est recommandé d'utiliser le type générique pour contraindre les données d'entrée, puis d'utiliser l'objet pour appeler la routine **DECODE**.

Appuyez sur **F6** pour exécuter le Job. Cela retourne *ten*, qui est le résultat de la valeur *10*.





## 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 du *Studio Talend* 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 [Conception de Jobs](#).

Pour plus d'informations concernant la syntaxe <%...%>, consultez [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 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 [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.

