



# Talend Open Studio for MDM

Release Notes

## 5.2.1

## **Copyleft**

This documentation is provided under the terms of the Creative Commons Public License (CCPL).

For more information about what you can and cannot do with this documentation in accordance with the CCPL, please read: <http://creativecommons.org/licenses/by-nc-sa/2.0/>

## **Notices**

All brands, product names, company names, trademarks and service marks are the properties of their respective owners.

---

## Table of Contents

System Requirements .....	1
MDM: New Features .....	2
1. Studio .....	2
2. Web User Interface .....	2
MDM: Bug Fixes / Change Log .....	3
1. Bug Fixes .....	3
MDM: Known Issues and Known Limitations .....	4
1. Foreign keys when using multiple MDM versions and a SQL database .....	4
2. With SQL storage, data model and data container names must be the same .....	4
3. Automatic database indexing .....	4
4. Journal does not apply polymorphism .....	4
5. Foreign Key integrity when importing or migrating data .....	5
6. Full-text search with a join condition in a View .....	5
7. Full-text matches in multi-level records .....	5
8. Timeout settings when migrating large volumes of data with the dbmigration tool .....	5
9. Migration issues with recursive data models .....	5
10. Inheritance issues with a SQL database .....	6
11. XQuery with a SQL database .....	6
Documentation .....	7
1. New documents .....	7
2. Revised documents .....	7
3. Known issues .....	7

---

# System Requirements

Talend Open Studio for MDM users should refer to the <http://www.talend.com/docs/community/prerequisites.html> and [http://talendforge.org/wiki/doku.php?id=mdmce:installation\\_guide](http://talendforge.org/wiki/doku.php?id=mdmce:installation_guide) for more information on Installation and System Requirements.

# MDM: New Features

## 1. Studio

- *Talend MDM* comes with the option to use a SQL database to store MDM data records, and ships with an embedded H2 database for this purpose.
- Numerous usability improvements have been made in the Studio, including the addition of a Foreign Key icon and the ability to jump between entities linked by a Foreign Key; the possibility to switch easily to XML when browsing the data container and the addition of code complete and error highlighting features; and shortcuts for creating elements of a specific type.
- Data internationalization: A multilingual custom type has been added in the data model which administrators can use to render certain elements localizable. Web users are then able to provide a localized version of the corresponding fields in a data record.
- The tMDMDelete component has been enhanced to enable the delete of records based on conditions, or all records in a specific entity.
- Views and Processes are grouped in predefined categories, users no longer need to know the naming convention to create or edit Views thanks to improvements in the new and edit dialog boxes, and Processes have more functional names that are more closely related to the Web User Interface.

## 2. Web User Interface

- Data internationalization: A multilingual custom type has been added in the data model which administrators can use to render certain elements localizable. Web users are then able to provide a localized version of the corresponding fields in a data record.

# MDM: Bug Fixes / Change Log

## 1. Bug Fixes

In addition to the above new features a number of minor improvements within the entire product and significant bug fixes have been done.

Please see the corresponding Change Log on our bug tracking system for more details on the individual issues:

- Talend MDM 5.2.1:

<https://jira.talendforge.org/secure/IssueNavigator.jspa?requestId=13951>

# MDM: Known Issues and Known Limitations

We encourage you to consult the JIRA bug tracking tool for a full list of open issues and to see which issues will be fixed in the upcoming release:

<http://jira.talendforge.org/secure/IssueNavigator.jspa?requestId=13949>

## 1. Foreign keys when using multiple MDM versions and a SQL database

When you are using a SQL database to store your data records, each revision in the data model has its own SQL storage.

If you have a model in which one entity references another (for example, *Product* references *Product Family*) and these entities are stored in different versions (for instance, if *ProductFamily* is in *HEAD* and *Product* in *VI*), you must disable the foreign key integrity option. Otherwise, when you try to create a new *Product* in version *VI* that references a *ProductFamily* in *HEAD*, this will cause a constraint validation error and the *Product* record will not be saved.

## 2. With SQL storage, data model and data container names must be the same

When you are using a SQL database to store your data records, the data model and the data container must have the same name.

## 3. Automatic database indexing

The SQL storage automatically sets indexes in the database based on the "searchable" elements in the *Browse\_items* Views. For automatic indexing to work, the View needs to be deployed first and the data model second.

If you deployed the View after the data model, redeploy the data model to add the indexes.

Note that the SQL storage also restores the indexes when the MDM Server restarts.

## 4. Journal does not apply polymorphism

When you use Entity inheritance in your data model (for instance, *Employee* extends *Person* and *Manager* extends *Employee*), the Journal only shows records of the selected type instead of applying polymorphism.

For example, if you select *Employee*, it does not show managers even though managers are also employees.

## 5. Foreign Key integrity when importing or migrating data

In certain complex cases, Foreign Key integrity violations may occur when importing data or performing a migration from another server.

To avoid this issue, temporarily disable the Foreign Key integrity constraint that causes the error in the data model.

## 6. Full-text search with a join condition in a View

Full-text searches with a join condition are not supported in Views.

## 7. Full-text matches in multi-level records

The full-text index may find matches in complex-types that are several levels deep in a hierarchical MDM record. This will cause errors when clicking the link to open the record.

## 8. Timeout settings when migrating large volumes of data with the dbmigration tool

When migrating a database, the dbmigration tool needs to retrieve the IDs of all the records in the database. This operation may take a long time and as a result it could exceed the default transaction timeout, which is set to 300.

To change this setting, edit the *TransactionTimeout* value in the jboss-service.xml file, which is located in `<jboss_dir>/server/default/conf/jboss-service.xml`.

For example, a value of 1800 is recommended for a database with one million instances.

## 9. Migration issues with recursive data models

When migrating a database, you may encounter issues with recursive data models.

While the migration handles the creation of tables to populate the RDBMS, it does not specify the order for the content of each table. As a result, you may encounter integrity constraint issues if the migration tool tries to insert a child record for which the parent record does not yet exist.

### Workarounds:

- Set the batch size in the dbmigration tool that is equal to the number of records in the XML database, which creates one transaction per record.

- Disable Foreign Key integrity for the recursive relation before migrating the database and re-enable it once migration is complete.
- Migrate your data to the Staging Area and then validate the Staging Area data.

## 10. Inheritance issues with a SQL database

When you are using a SQL database to store your data records, IDs must be unique within the whole inheritance tree.

For example, if *Employee* inherits from *Person*, *Person* can have ID=1 and *Employee* can have ID=2 but *Person* and *Employee* cannot both have ID=1 because an instance of *Employee* is also an instance of *Person*, and this would create a conflicting ID.

## 11. XQuery with a SQL database

XQuery is not supported if you are using a SQL database rather than an XML database to store your master data records.

This has an impact on the following features:

- **Stored Procedures**

SQL queries are supported instead, and are converted to XML. Note that the data container must be selected - *All* is not supported.

- **Custom FK filters**

No workaround is available at this time.

# Documentation

## 1. New documents

The following new documentation is available for this release.

- *Talend Big Data Studio User Guide*: this guide addresses only big data related features and capabilities offered in the big data studios.

## 2. Revised documents

In addition to updates to the content across the documentation set, the following specific documentation changes have been made.

- *Talend ESB Getting Started User Guide* has been renamed *Talend ESB Getting Started Guide*, and *Talend ESB Service Factory User Guide* has been renamed *Talend ESB Service Developer Guide*.
- Updates have been made to several *Talend ESB* guides to reflect changes in certain Apache products.
  - *Talend ESB Development Guide*: Apache CXF 2.7
  - *Talend ESB Mediation Developer Guide*: Apache Camel 2.10
  - *Talend ESB Service Developer Guide*: Apache CXF 2.7
  - *Talend ESB STS User Guide*: Apache CXF 2.7 - STS

## 3. Known issues

The documentation of certain technical fixes that arrived too late to be included in the current release will be dealt with in the next release.

<https://jira.talendforge.org/secure/IssueNavigator.jspa?requestId=13952>