TIBCO EBX® Data Exchange Add-on

Version 3.0.5
May 2022

TIBCO®

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

TIBCO and TIBCO EBX are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (*https://www.tibco.com/patents*) for details.

Table of contents

User Guide

Reference Guide

Administration Area

Import Options

Export Options

Dynamic Data Modeling

User Guide

Release Notes

User Guide

CHAPTER 1

Introduction

This chapter contains the following topics:

1.1 The add-on at a glance

In today's data driven world the ability to seamlessly move and transform data is a must have. Additionally, communication between business teams can greatly benefit from a common representation of data. The TIBCO EBX® Data Exchange Add-on meets all of these requirements and allows you to import, export, transfer, transform and model data.

See also

Overview *[p 15]*

Export overview *[p 29]*

Overview of data transfer *[p 45]*

1.2 How data movement works

All data flow requires mapping between a data source and target. In some cases the add-on can automatically propose a mapping, in others a custom mapping is required. The following bullet points describe these two scenarios further:

- Default add-on functionality allows business users to import, export, and transfer data between sources and targets of similar structure. No additional configuration is needed.
- The add-on also allows users to move data between disparate sources and targets, and to transform data. An administrator will need to configure the add-on if this functionality is required. However, as an added benefit to business users, this happens behind the scenes. Their data flow tasks aren't burdened with increased complexity.

See also

Defining mapping concepts *[p 60]*

Overview *[p 15]*

Export overview *[p 29]*

Overview of data transfer *[p 45]*

1.3 How data modeling works

Simply put, the add-on's data modeling capabilities allow you to generate a TIBCO EBX® compliant data model from an XML, Excel, or DDL file. This process is outlined below:

- Create a data model by pointing the add-on to a compatible source file containing the model's structure.
- Generate an XSD file that contains the data model and is compliant with EBX®.
- Create, or use an existing, data model using the Data Modeler Assistant and import the XSD file.

See the *Dynamic Data Modeling User Guide* for more information.

Getting Started with Data Exchange

CHAPTER 2

Setting default import and export behavior

You can set the default selections for each import and export format. When users perform an import or export, they can:

- change selections if needed.
- load saved preferences.

To set default configuration option values:

1. Navigate to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Additional configuration* > *Default option values* > *Import/Export* and open the record contained in the table.
2. The options are categorized by file type. Select the appropriate tab and after updating values, save to keep changes.

CHAPTER 3

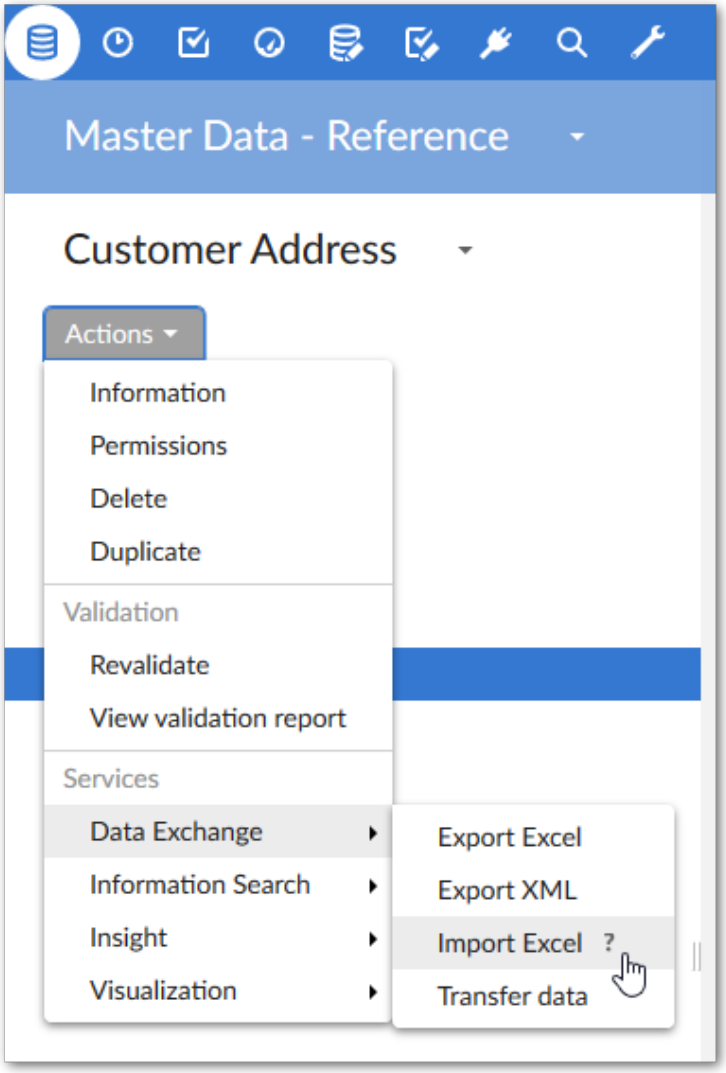
Importing data

This chapter contains the following topics:

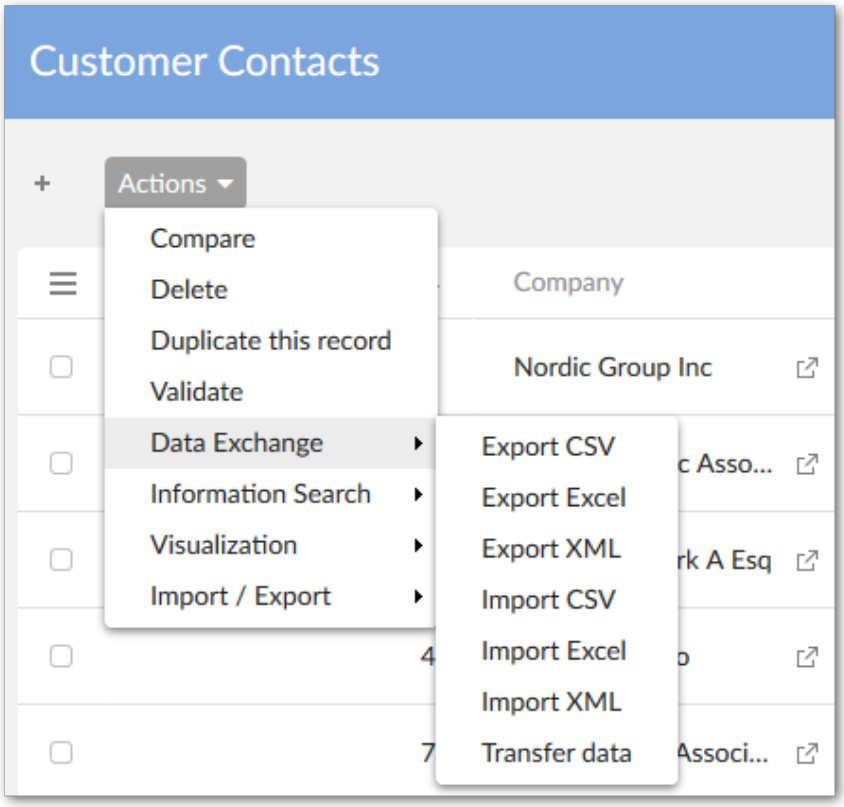
3.1 Overview

When you import data using the add-on, you bring data from a *source* location such as a file, or database, into a *target* EBX® table. You can initiate import from the following two locations:

- A dataset's **Actions** menu. Note that the only available import format here is importing from an Excel file. You will have the option of selecting one or more of the dataset's tables as the import's target.



- A table's **Actions** menu. All import formats are available here and the table from which you run the service becomes the target.



Attention

When importing XML or Excel, you can import multi-valued complex fields. All levels of a complex field can be imported with XML. However, only the first level can be imported from an Excel file.

Common pages

There are several pages common to CSV, Excel and XML import operations. Note that the **Configuration**, **Mapping**, and **Simulation** screens are accessible by default. However, administrators can use permissions to restrict access to these screens for specific profiles. See Setting import and export permissions [p 51] for more information.

The following sections provide brief overviews of each page:

- File selection and Preference page [p 18]
- Configuration page [p 18]
- Mapping page [p 18]
- Preview and simulation page [p 18]

- Results page [p 18]

Attention

At any time, if you have a question about an option on an import page, hover your mouse over its label and click the icon to open the related tooltip. Additionally, you can see the *Reference Guide* which also contains descriptions of each field. For options, or behavior specific to each format, see their corresponding sections below.

File selection and Preference page

When importing from Excel or CSV formats, this page allows you to select the source file and optionally choose a preference to load. Preferences store import configuration settings. If you use a preference, you can still adjust individual configuration settings. For example, you can include or exclude tables after loading a preference. When importing XML, file selection is done from the configuration page (described in the next section).

If an administrator has setup a custom mapping for data transformation, you'll use the preferences to tell the add-on to import using the appropriate configuration.

Configuration page

The configuration page allows you to provide information about the file used as source for this import. For example, when importing from a CSV file, you can specify the types of separators and line breaks used in the file. Additional options allow you to control import behavior such as:

- Specifying the import mode. Options can include update, insert, replace all content.
- Determining whether to force the import by disabling any existing triggers and constraints.
- Ensuring all primary key columns are mapped.
- Specifying that missing values, are not imported into the table.
- Pointing to custom Java classes that transform, or validate imported data.

Mapping page

The add-on has two types of mapping pages; those for tables and those for columns. Only Excel uses the table mapping functionality as you can import to multiple tables and from multiple sheets. The objective of these two pages is to map the data source and target. All columns must either be mapped, or ignored. Additionally, you can view a sample of the data and save preferences for later use.

Preview and simulation page

After all configuration settings are complete, you can simulate the import operation prior to execution. This page provides several options for simulation that help you ensure the desired outcome before importing.

Results page

The results page displays the import operation's results. From this page you can also start another import.

3.2 **Importing from a CSV file**

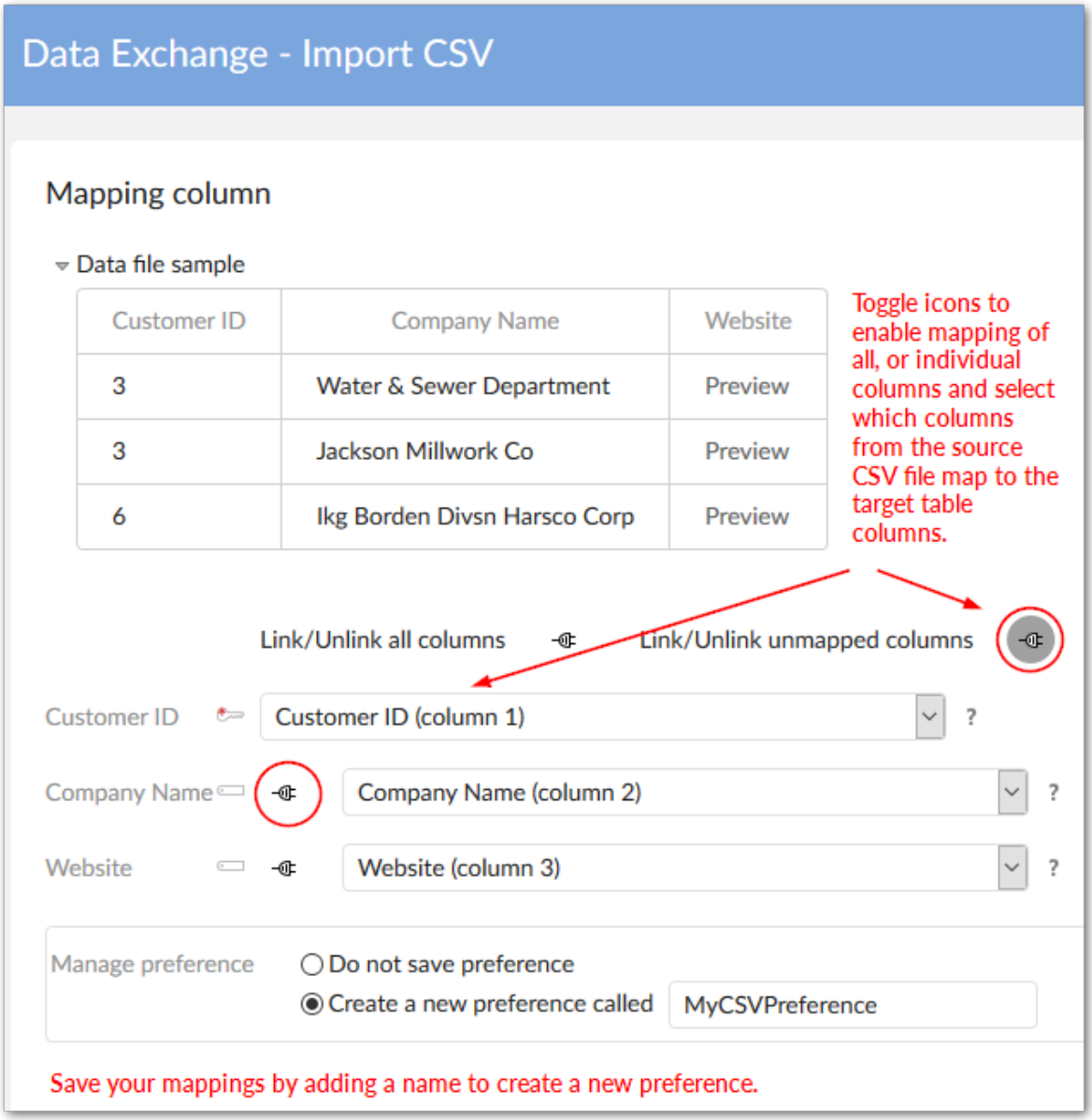
When you import a CSV file, the configuration page allows you to:

- Specify import behavior.
- Supply information about the file you are importing from, such as the separators and line return types used in the file.
- Point to any Java classes used to transform, or validate data.

Attention

If you are importing to a table that has its primary key set to read-only and you want to map to this column, you can run the import in **Update only**, or **Update or insert** mode. In these modes, records will be updated, but an error message will display if an attempt is made to insert a new record.

Remember that on the column mapping page all columns must either be mapped, or ignored. The following image highlights some of these features:



> **Note**
>
> If the source file sample isn't displaying, the wrong separators may be selected in the configuration page.

3.3 **Importing from an Excel file**

Import behavior and options can vary slightly depending on where you initiate Excel import from. However, the first three pages always allow you to select the file to import, load any saved preferences, and adjust configuration settings.

Attention

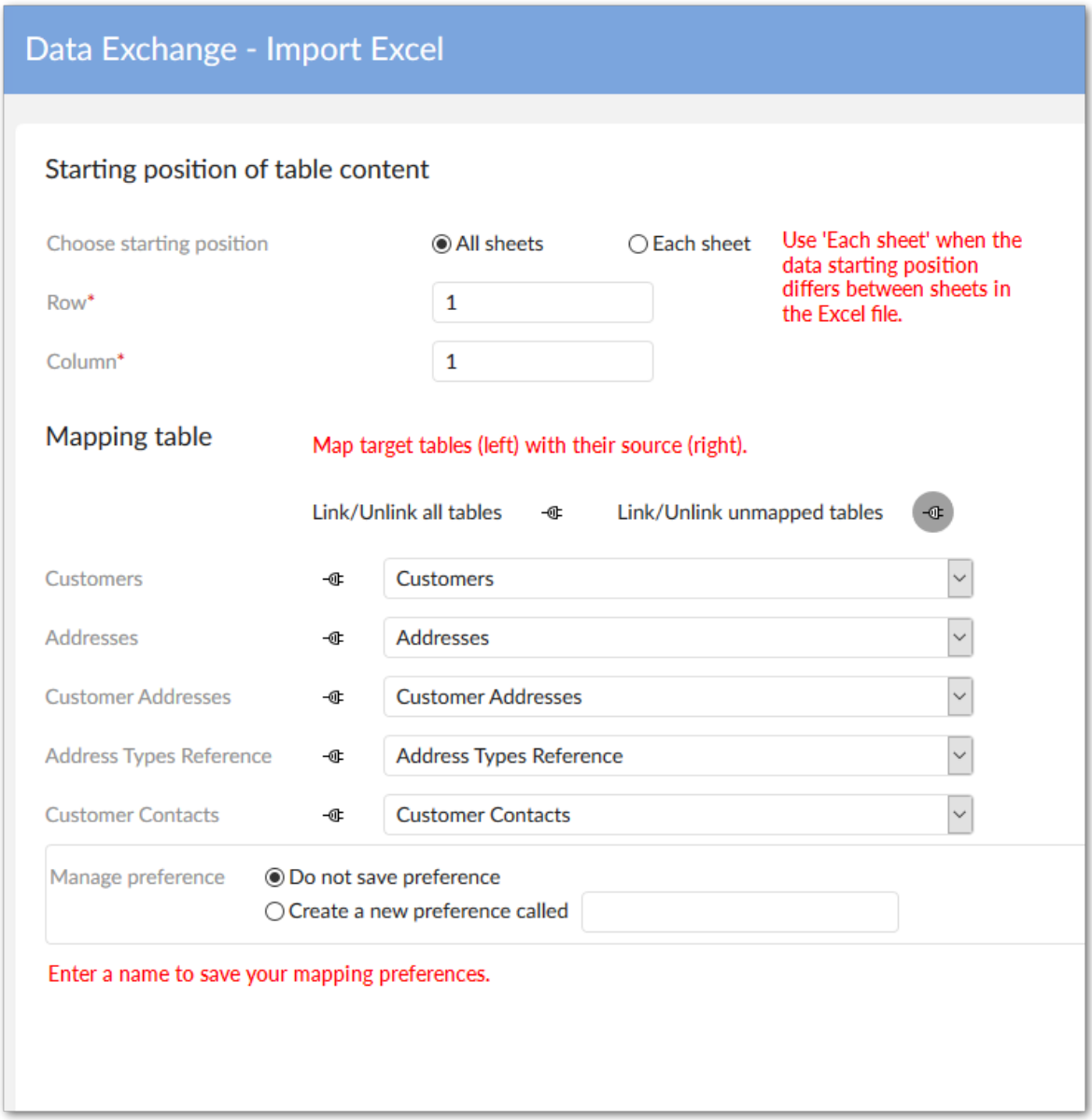
If you are importing to a table that has its primary key set to read-only, and you want to map to this column, you can run the import in **Update only**, or **Update and insert** mode. In these modes, records will be updated, but an error message will display if an attempt is made to insert a new record.

Attention

To avoid issues during import of a large Excel file, please do not activate the **Download file of invalid data** option.

When you run the import service from a dataset's **Actions** menu, the add-on presents you with the additional table mapping screen. This screen allows you to import the same source sheet into

multiple tables, and import each source sheet into different tables. The following image highlights these options:



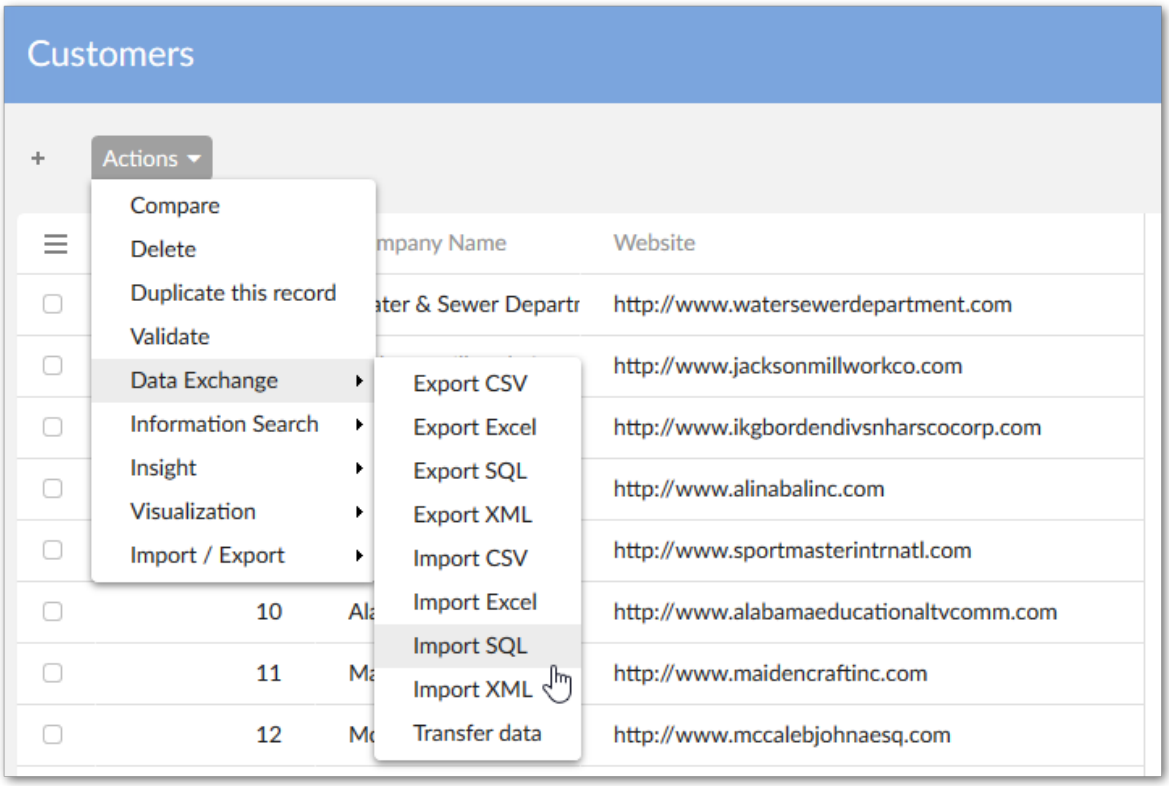
3.4 Overview of importing from an external database

Before importing from an SQL source, an administrator needs to configure the add-on to connect to the database. The service is not available until these steps are completed.

SQL import

The following steps demonstrate how to use the **Import SQL** service

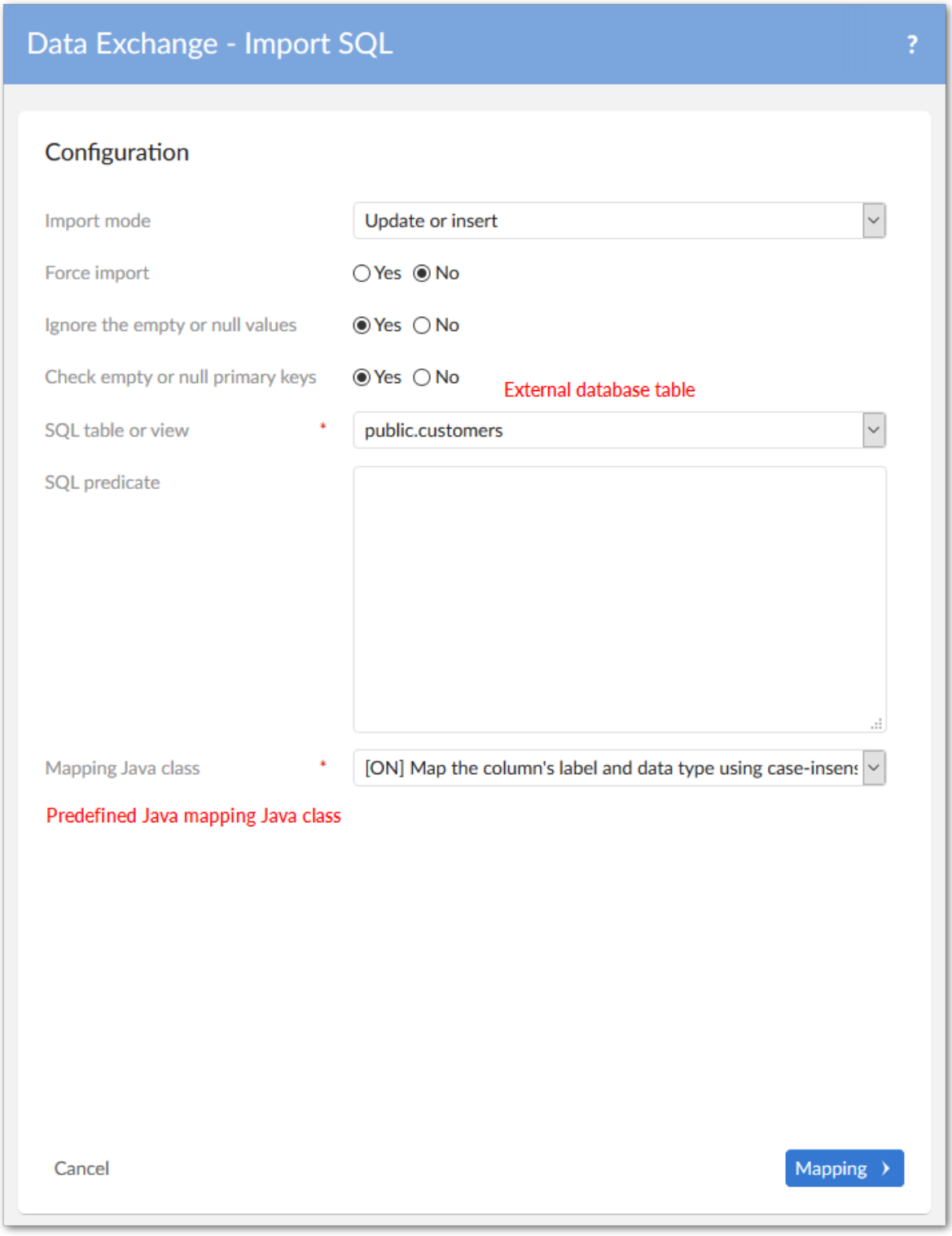
1. From a table's **Actions** menu, select the **Import SQL** service, the **Import SQL** screen displays.



Attention

If you are importing to a table that has its primary key set to Read-only, you can run the import in **Update only**, or **Update and insert** mode. In these modes, records will be updated, but an error message will display if an attempt is made to insert a new record.

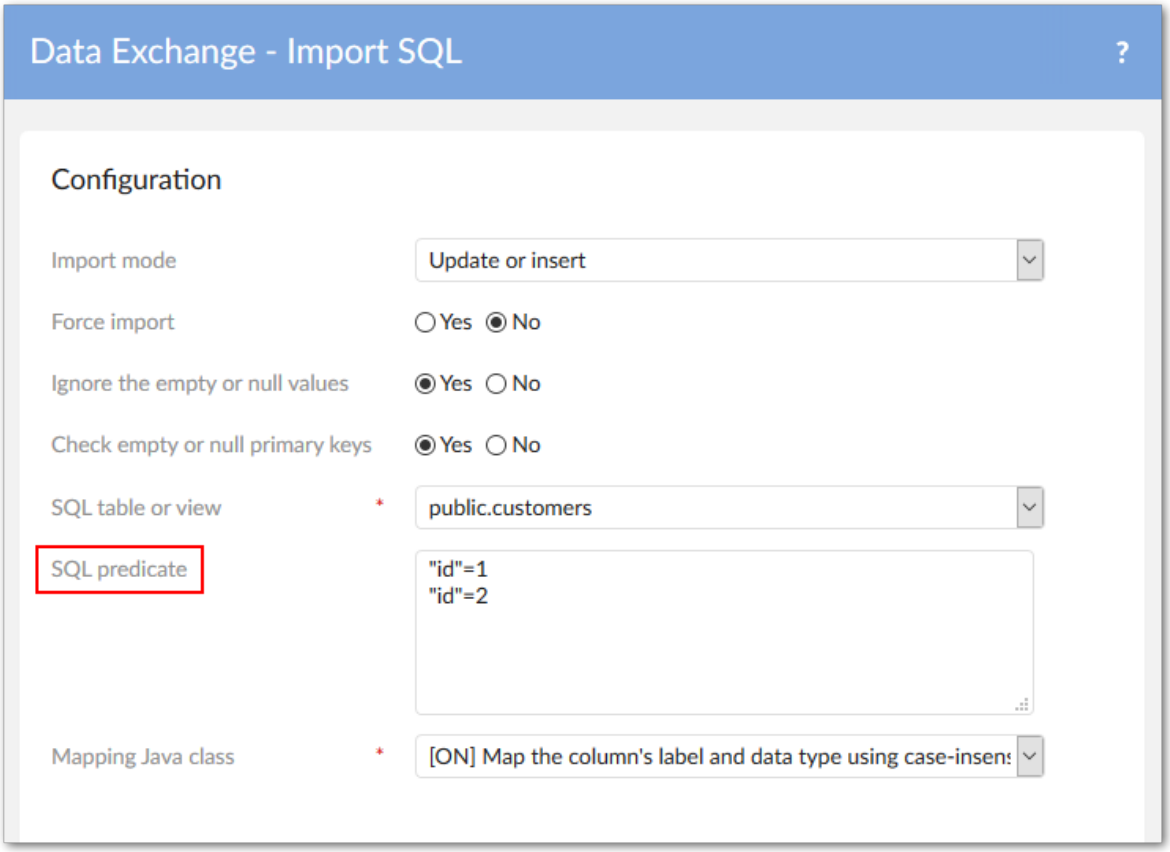
2. On the **Import SQL** screen, you must specify the source table or view in the external database on which the add-on reads data from and choose the Java class used to map data. All the tables in the connected external database display in the **SQL table view** property's drop-down list.



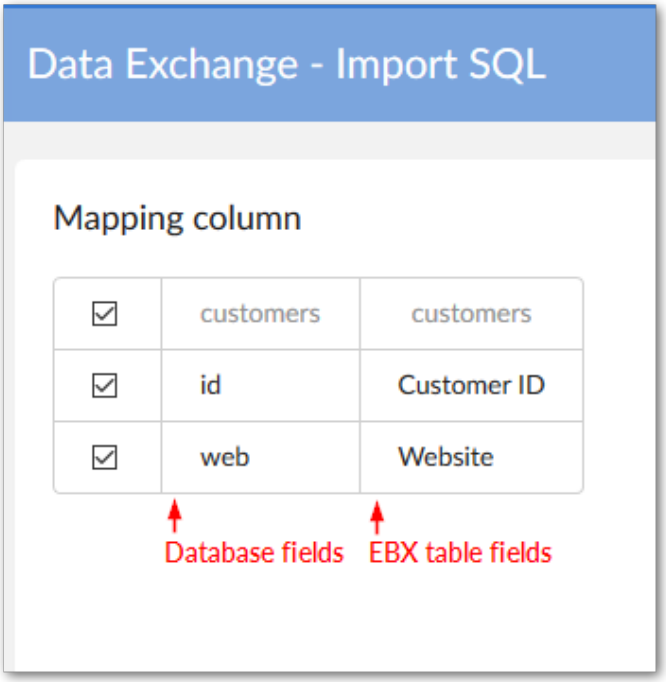
3. If you are connected to a database with a large number of tables or views, it may negatively affect add-on performance when loading. You can narrow down the loading scope by setting the specific

table name or limiting table objects using the relational search character `%` in *Reference data* > *SQL data source* > the *Table name pattern* property.

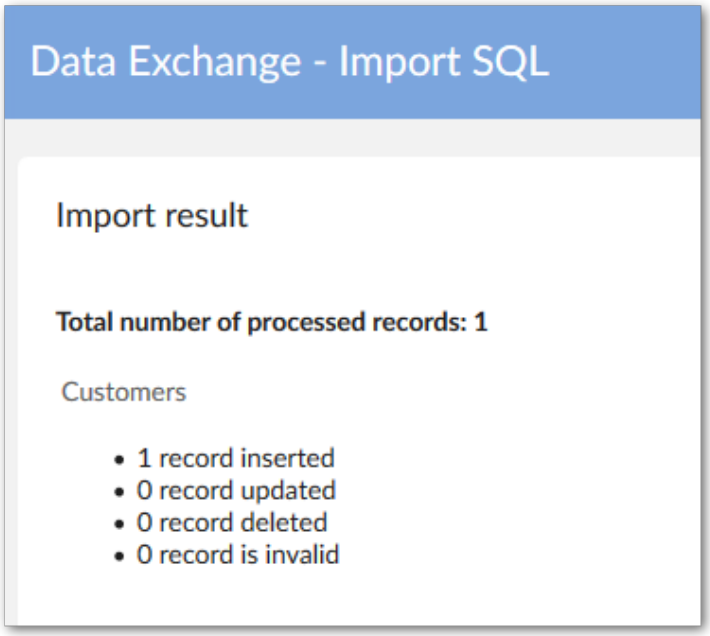
4. As shown below, you also can filter data from the external database when importing by entering the conditions in the **SQL predicate** text box in the same way as you use the *WHERE* SQL clause.



5. Once you have finished the configuration, click the **Mapping** button to enter the **Mapping column** screen. You can select or ignore columns by checking the checkbox before each field name prior to data import.



6. Click **Import** if you've finished mapping. When the import progress completes, a detailed result report displays the number of records proceeded.



Special notation:	
✔	You must have 'Read' permission on the external database to import data to EBX® tables.

3.5 Importing from an XML file

When you import from an XML file, the add-on automatically detects the appropriate data mapping to use. So, all you have to do is select the file and set configuration options.

If you encounter difficulties when importing from XML, it could be that the file either doesn't meet the requirements to use the add-on's default formatting. In this case an administrator needs to customize a mapping configuration.

Attention

If you are importing to a table that has its primary key set to Read-only, and you want to map to this column, you can run the import in **Update and insert** mode. In this mode, records will be updated, but an error message will display if an attempt is made to insert a new record.

CHAPTER 4

Exporting data

This chapter contains the following topics:

4.1 Export overview

Depending on what format you are exporting to, the add-on presents you with slightly different options. In most cases you will see a configuration page, and mapping pages. The sections below cover options relevant to each export format type:

- Exporting to CSV [p 29]
- Exporting to Excel [p 31]
- SQL export [p 35]
- XML export [p 38]

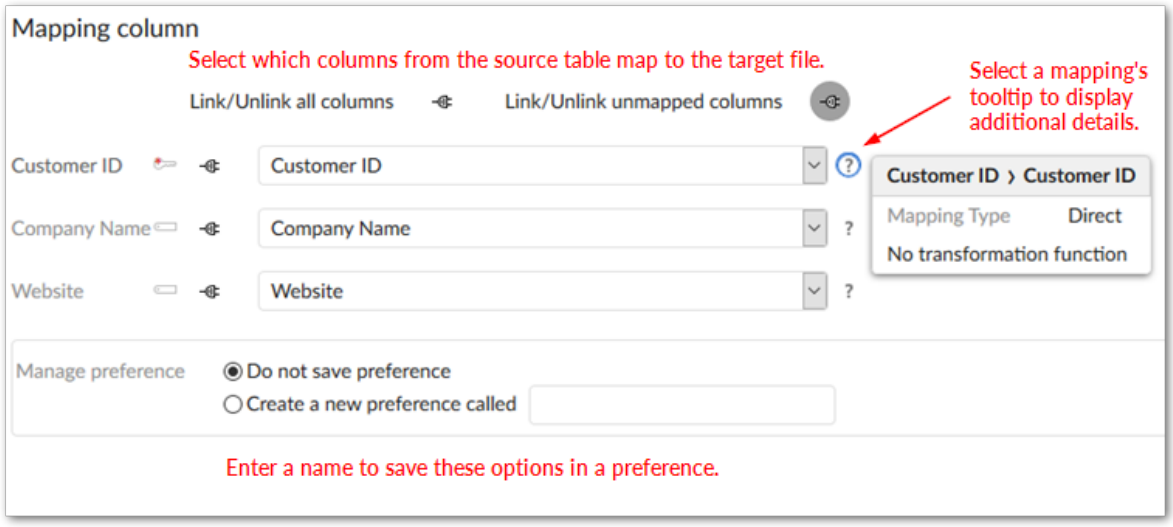
Attention

When exporting XML or Excel, you can export multi-valued complex fields. All levels of a complex field can be exported to XML. However, only the first level can be exported to an Excel file.

4.2 Exporting to CSV

You can run the **Export CSV** service from a table's **Actions** menu. The add-on only exports a single table's contents at a time. You only need to edit configuration settings and create column mappings. The settings in the configuration page determine the exported file's structure. For example, you can specify which separators and date formats to use. Also, you can load preferences that automatically

setup the configuration and mapping. Additionally, you can select a Java class to transform data on export.



Separator behavior

Exported CSV files use new lines (line feed and carriage return) to separate records. When configuring CSV export you can use the following options to define how the exported file separates fields and encapsulates values:

- The **Separator** option defines the character used to separate fields, or columns. The following example shows how the default option of a semicolon separates fields:

```
pk1;Salary
1;50,000
```

- The **Delimiter** option allows you to specify a character to encapsulate field values. As shown below, the exported file adds this character—a pipe character in this case—to the beginning and end of the field value:

```
|pk1|;|Salary|
|1|;|50,000|
```

A conflict would occur when an exported field includes the same character used as a separator. However, the add-on automatically uses double quotes as escape characters to ensure data integrity. The following sample shows a scenario where a comma is specified as the separator character and appears in the Salary field's value:

```
pk1,Salary
1,"50,000"
```

You can use the **Delimiter** option to override default behavior. As shown below, the delimiter marks the start and end of each field eliminating the necessity of escape characters:

```
|pk1|,|Salary|
```

```
|1|,|50,000|
```

> **Note**
>
> The add-on imports correctly from exported files where data values include the defined separator. These files also display correctly in spreadsheet programs. However, if you define a delimiter and view the file in a spreadsheet program, the display might not reflect the actual data structure.

4.3 Exporting to Excel

You can run the **Export Excel** service from a dataset, or table **Actions** menu. Running the service from a dataset allows you to export data from multiple tables to multiple sheets in the Excel file. When you run the service from a table, you can select records to export, or export all table contents. The following sections provide more information on the options available depending on how you execute the export operation:

- Exporting from a table [p 31]
- Exporting from a dataset [p 33]

> **Note**
>
> If invalid foreign keys exist, they will be included in the export and highlighted in the Excel file.

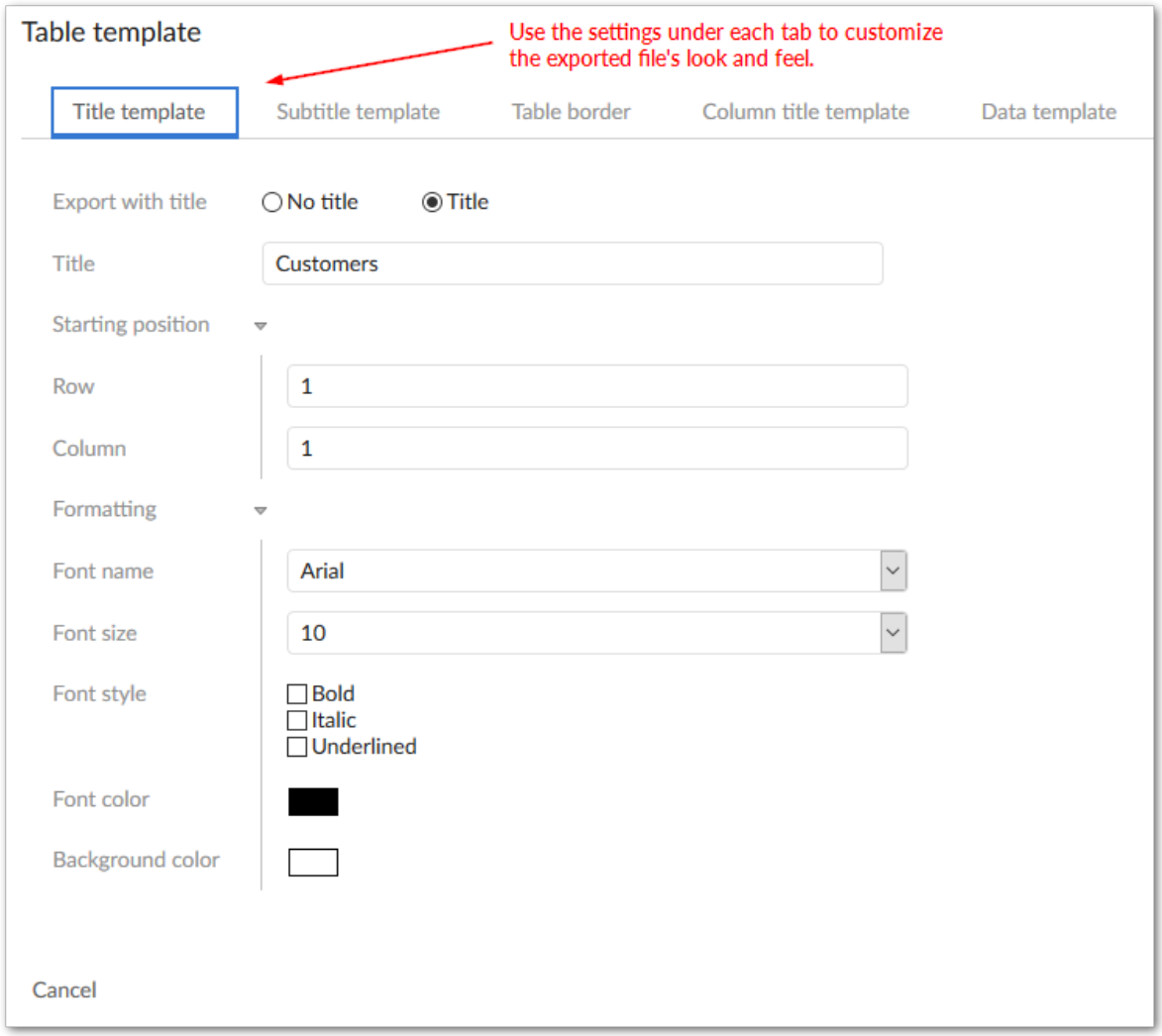
Attention

To avoid issues during export of a large Excel file, please use only the default options and do not activate the **Include validation messages**, **Export related data**, or **Export permalink for the primary or foreign key** options.

Exporting from a table

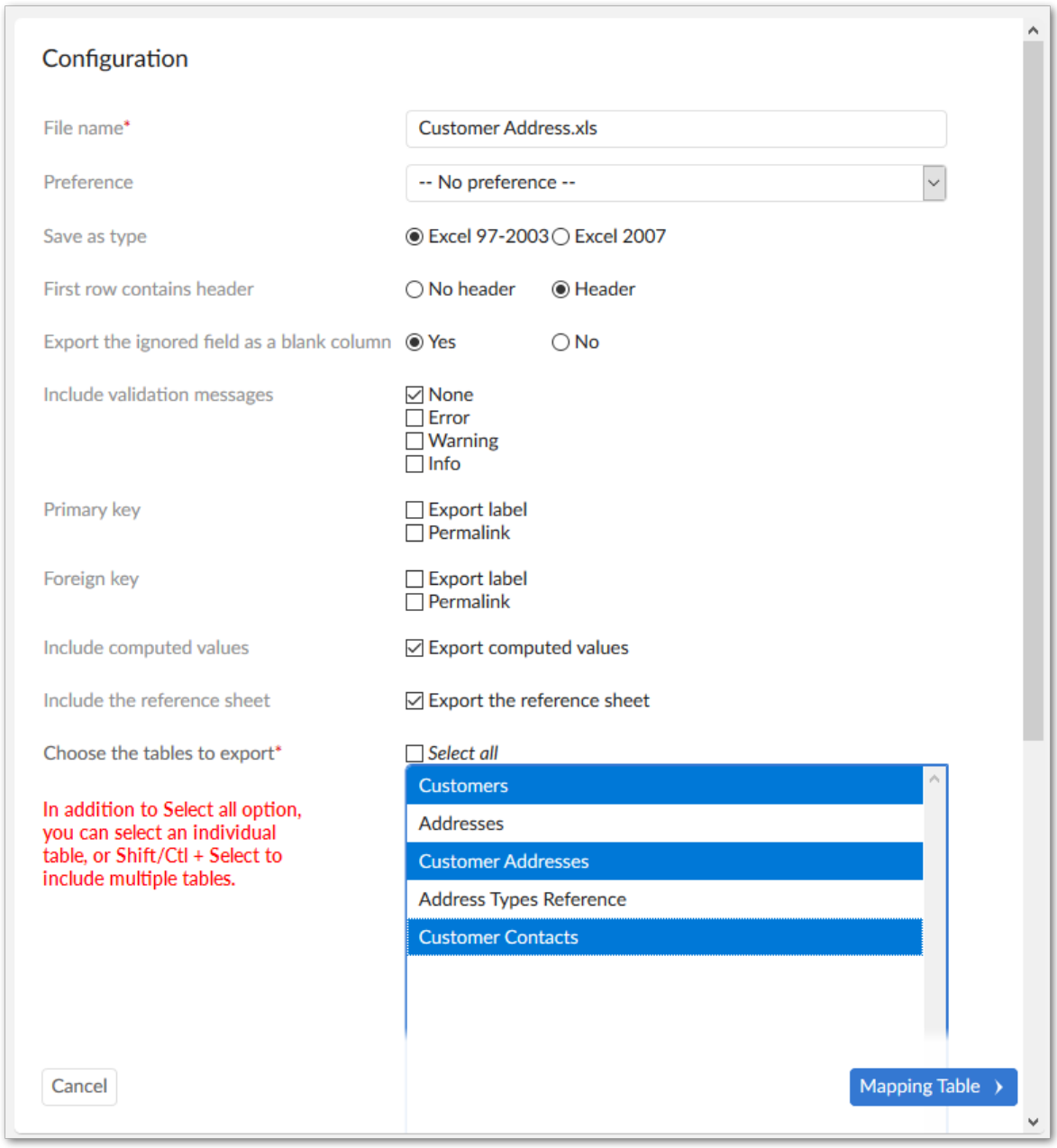
When you export from a table, you can select the records to include in the export, or export all table content. Just select the records before running the service if you only want to export a few. The configuration page allows you to determine the exported file's structure, load preferences, and select a Java class for data transformation. As shown in the following image, you can also create a table template to customize look and feel of the exported file. Once configuration is complete, use the

mapping page to determine how and which source table columns map to the target file columns and choose whether to save a preference.

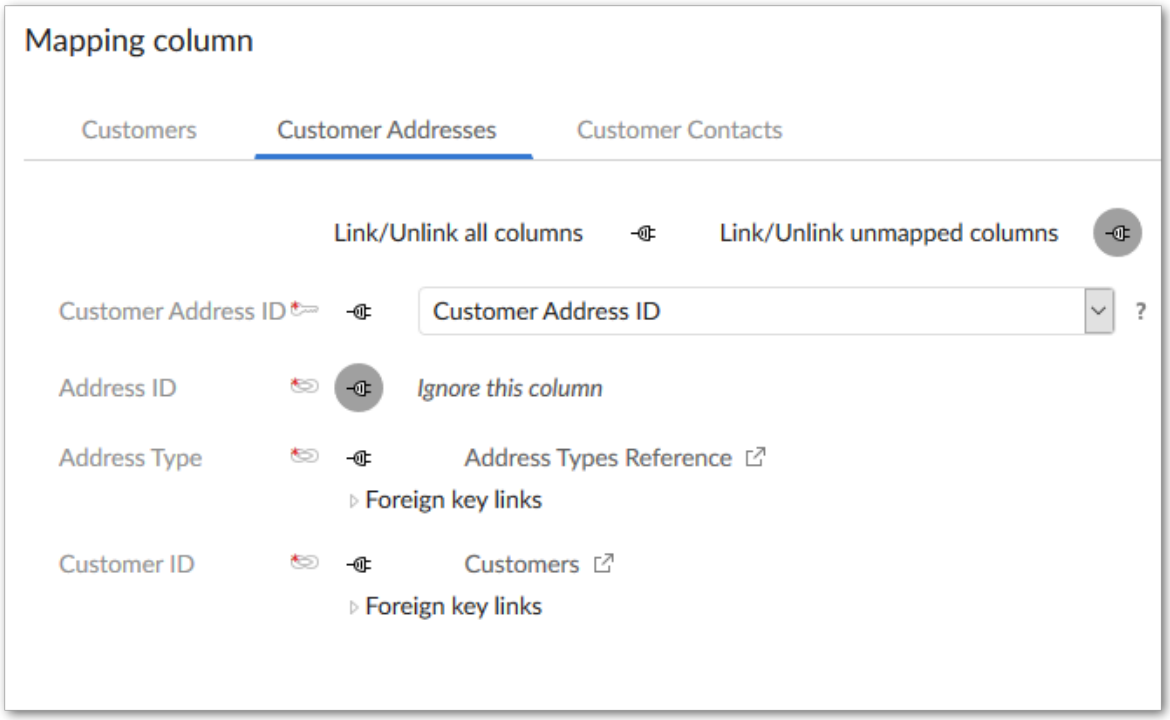


Exporting from a dataset

In addition to the options available when exporting from a table, you can choose one or more tables from the dataset to export. The add-on exports each table into a separate sheet in the Excel file.



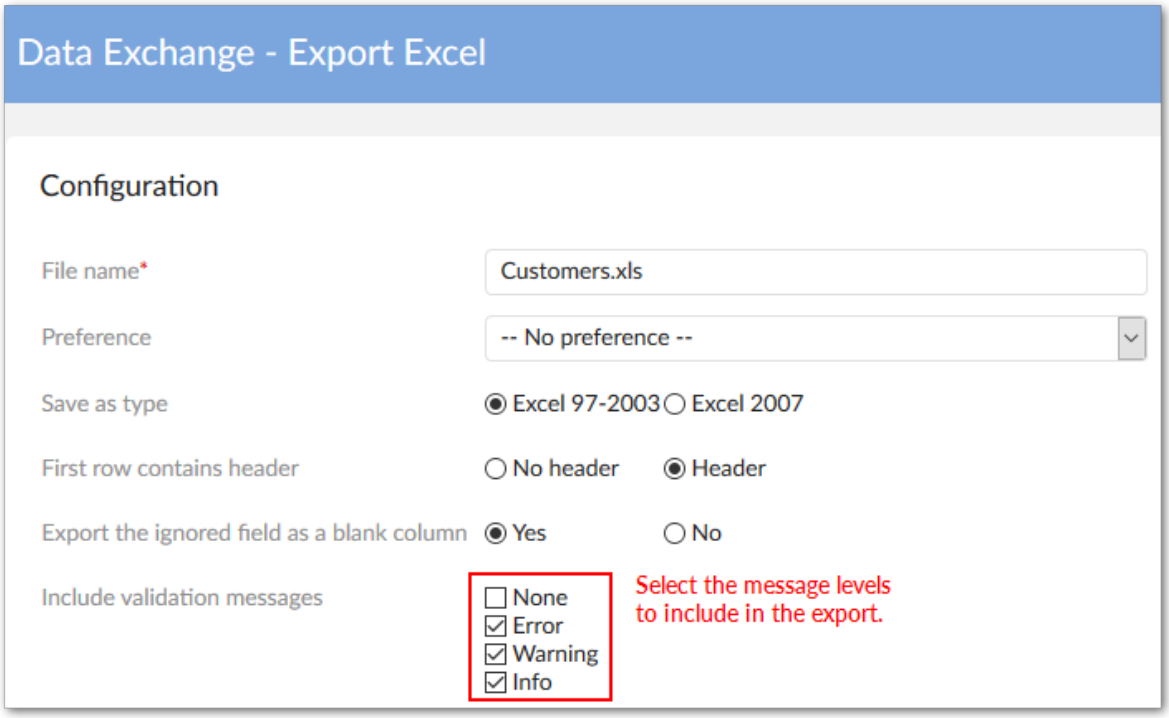
After selecting the tables to export the add-on presents you with two pages that allow you to map which sheet the tables are exported to and the source table columns with the target sheet columns. As shown below the column mapping page uses tabs to indicate which table/sheet you are mapping.



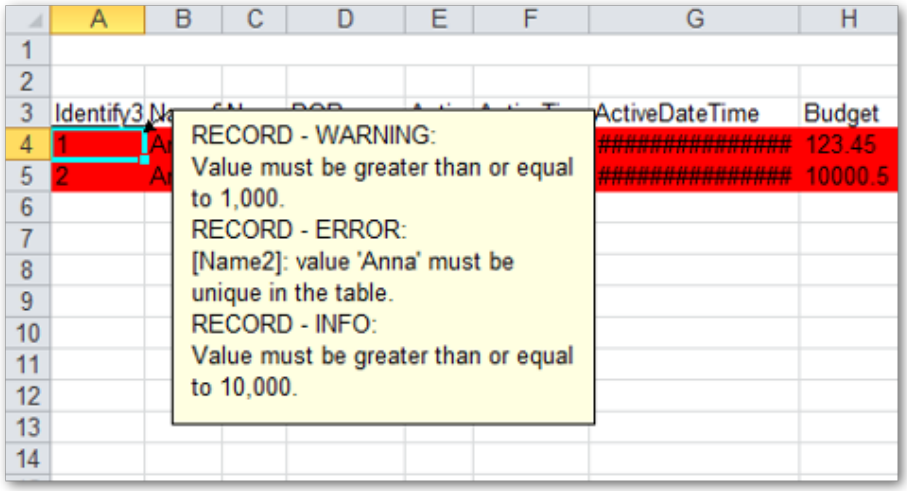
Including messages in exported files

You can include validation messages in exported Excel files. The add-on highlights fields and rows containing messages using: red for error, orange for warning, and blue for information. If a message applies to a table, the add-on creates a new sheet in the exported file and appends "-MSG" to the name.

As shown below, select the level(s) of messages to include in the export. Note that exporting messages forces validation for all tables included in the export.



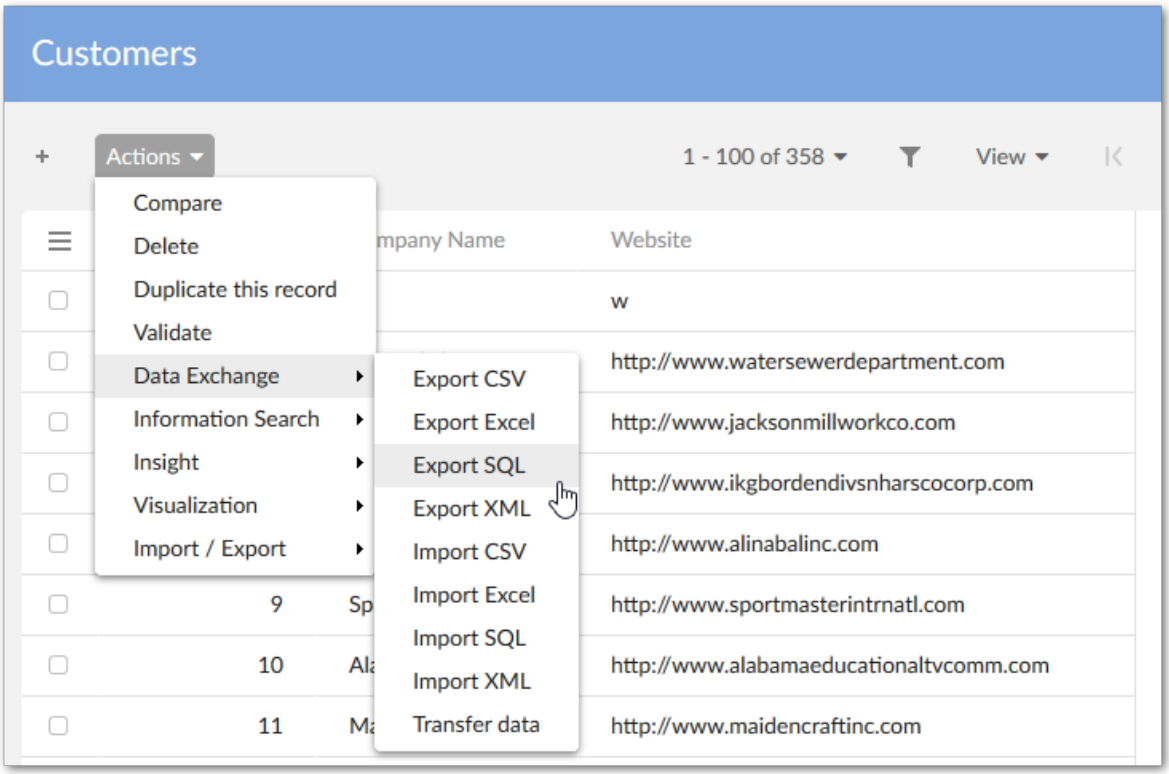
The following image shows an example of how messages are included in an exported Excel file:



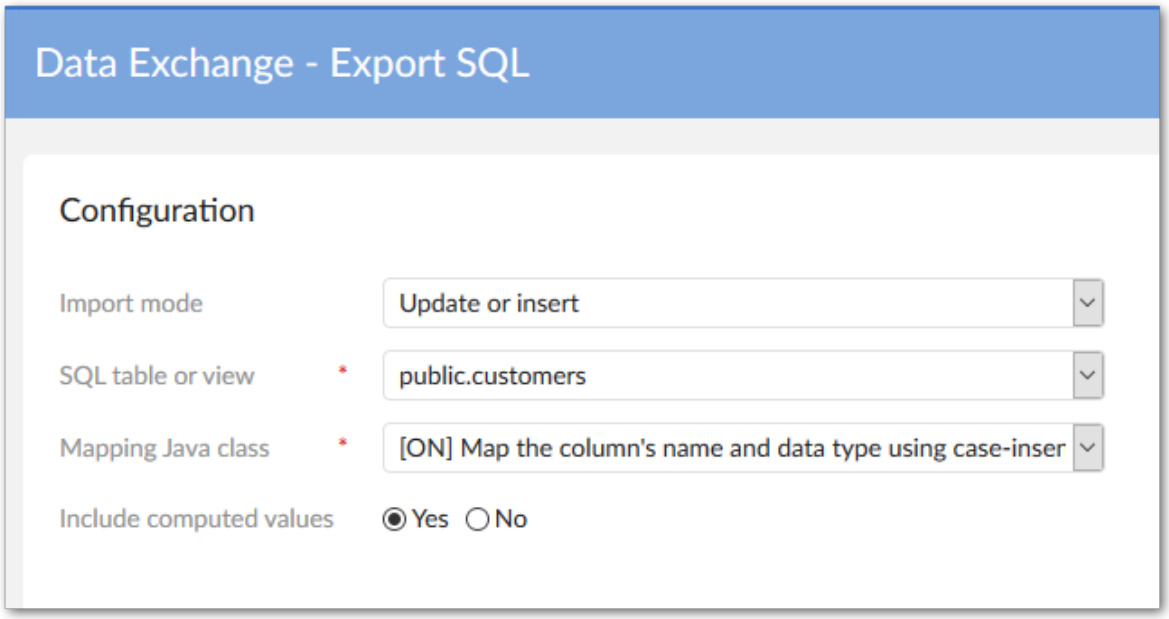
4.4 SQL export

The following steps demonstrate how to use the **Export SQL** service:

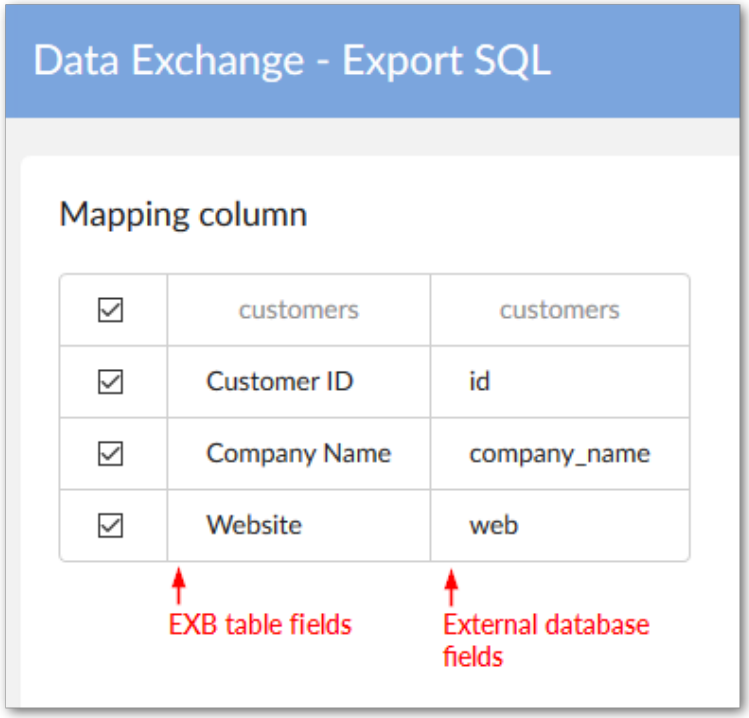
1. From a table's **Actions** menu, select the **Export SQL** service, the **Configuration** screen displays.



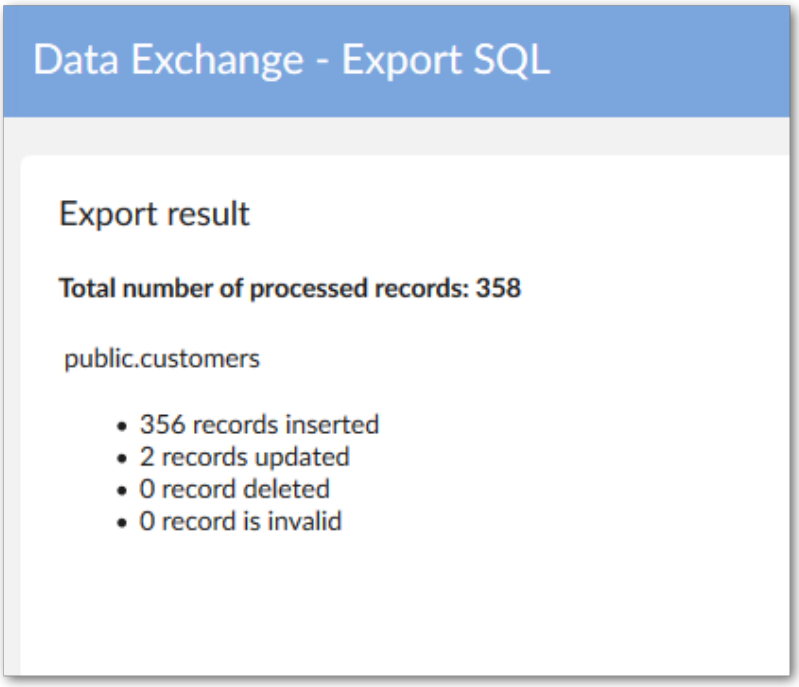
2. From a table's **Actions** menu, select the **Export SQL** service, the configuration screen displays as shown below:



3. Choose an external database table from the **SQL table or view** drop-down list and select the Java class used to map columns. Once you've finished the configuration, click **Mapping** to display the column mapping screen.



The image below shows the result of exporting to the Customers table on the external database.

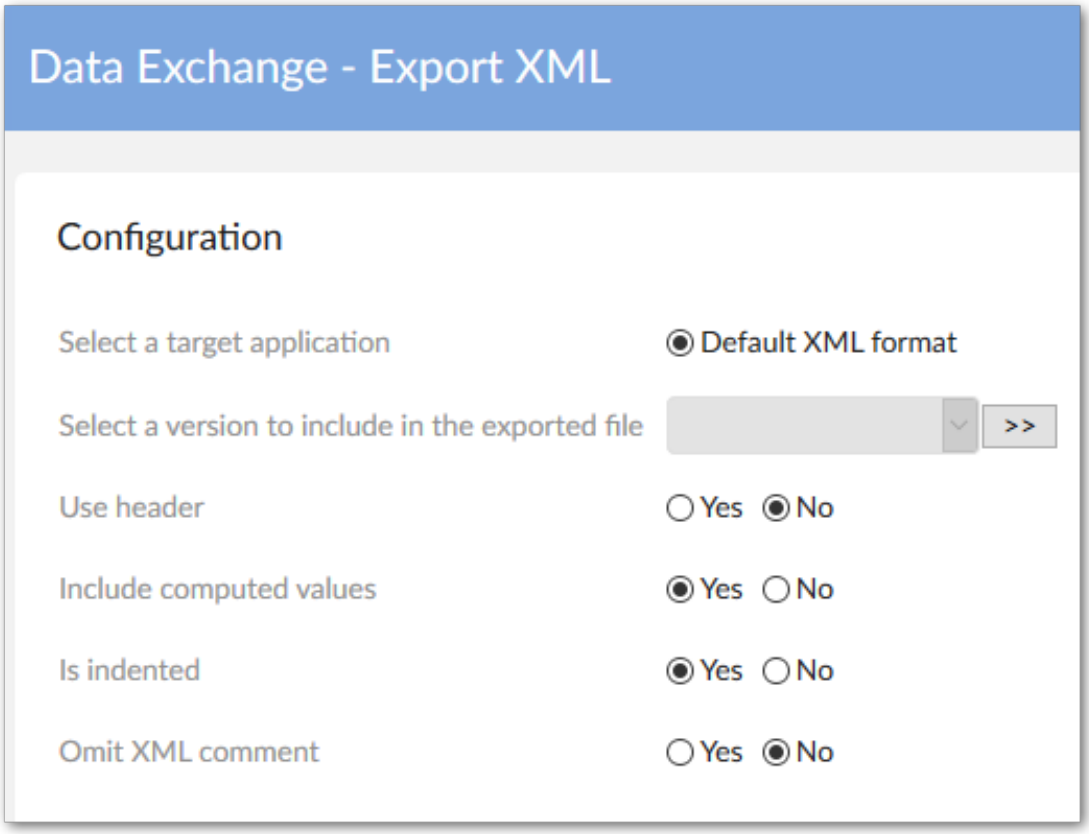


	LIBELLECOURT character(200)	CODEINTERNE integer	LIBELLELONG text	CODENATIONAL character(200)	DATEOFCREATION date	DESCRIPTIONPRODUIT character(200)	LIBELLEREDUIT character(200)
1	Food	1	Fast food	980	2016-01-05		Food - noodle
2	Camera	2		980	2016-01-06		Camera
3	FooD	3		980	2016-01-05		FooD
4	Fod	4		980	2016-01-06		Fod
5	Clothing	5		981	2016-01-07		Clothing
6	Clothe	6		982	2016-01-08		Clothe
7	Cloth	7		982	2016-01-05		Cloth
8	CloThing	8		982	2016-01-08		CloThing
9	Luxury	9		983	2016-01-08		Luxury
10	Luxury	10		983	2016-01-05		Luxury

4.5 **XML export**

If the table is not defined in the data mapping configuration when you attempt to export to XML, **Default XML format** displays as the only available target application. This default format, provided by the add-on, allows you to easily get an XML file and doesn't require a specific data mapping configuration.

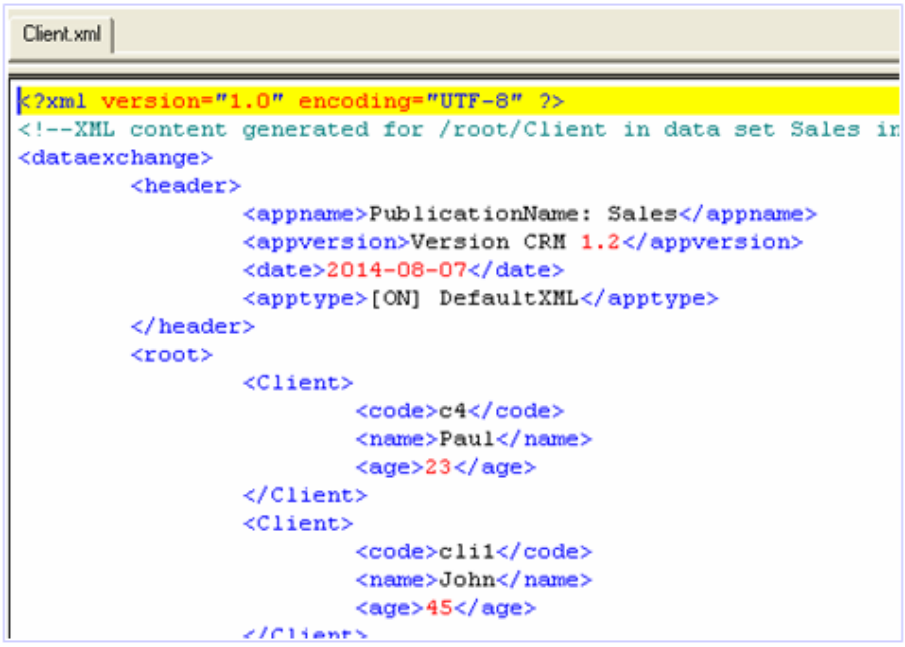
If a field name has to be renamed when exporting to an XML file, or another instance arises where data in the source and target differ, you have to specify a data mapping configuration. In these types of cases, other target applications display as options (see the rest of this user guide).



When the **Use header** option is used, the exported XML file contains a standard header with the following data:

- The name of the application. By default, this is the name of the dataset in which the table is located (a timestamps value is also added to build the name).
- The version that has been selected during the export configuration. If no specific version was specified, this will be blank.
- The export date.

- The type of the XML export, either **Default XML** or **XML** when a user-defined data mapping configuration is used.



4.6 **Exporting related data**

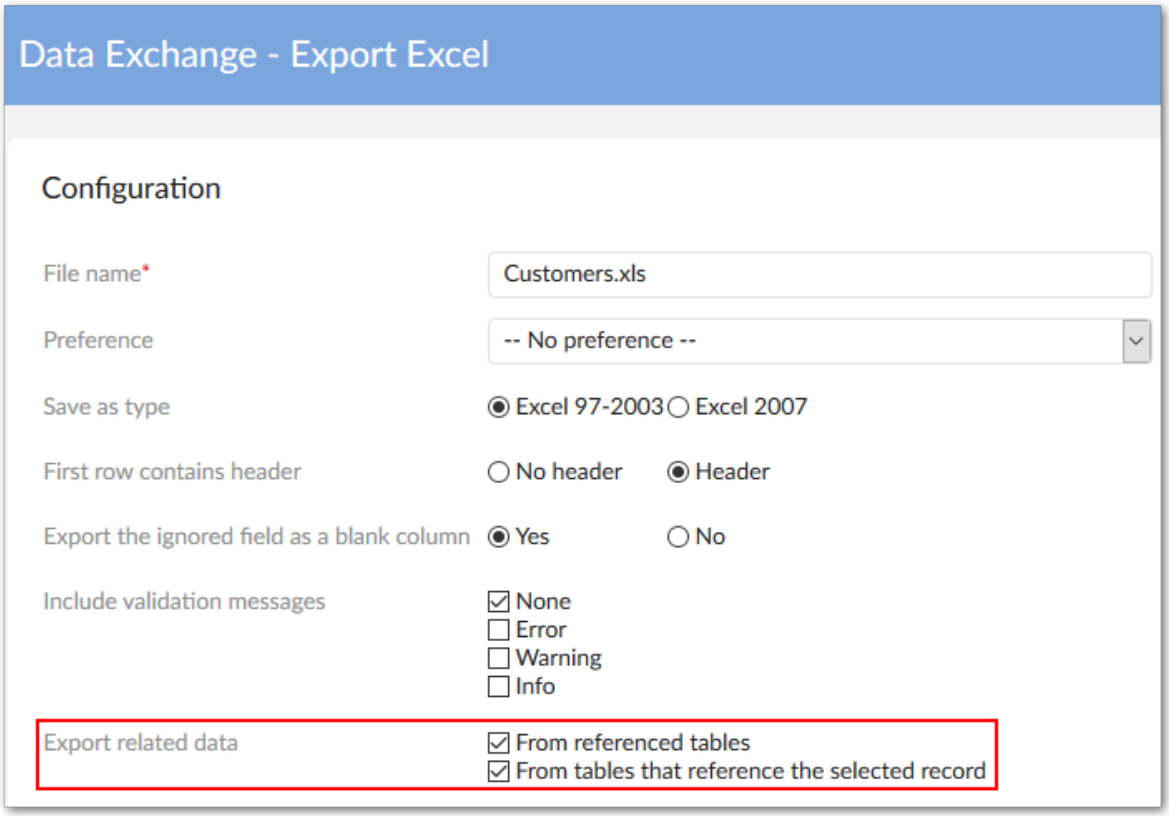
To help you locate data linked to specific records, the add-on allows you to export related data when exporting to:

- CSV: The add-on exports a ZIP file containing individual CSV files. One file contains the export's source data. The remaining files—one for each table—contain the related data.
- Excel: The add-on exports a single Excel file. The first sheet in the file contains the export's source data. Each additional sheet—one for each table—contains the related data.

The option to export related data is only available when exporting an individual record, or from a table. To include related data in a CSV or Excel export, use the following checkboxes for the **Export related data** property on the export's configuration screen:

- **From referenced tables:** to include data referenced by the selected record(s). Note that this may include data from external tables.
- **From tables that reference the selected record:** to include data from tables that hold a relationship to the selected record.

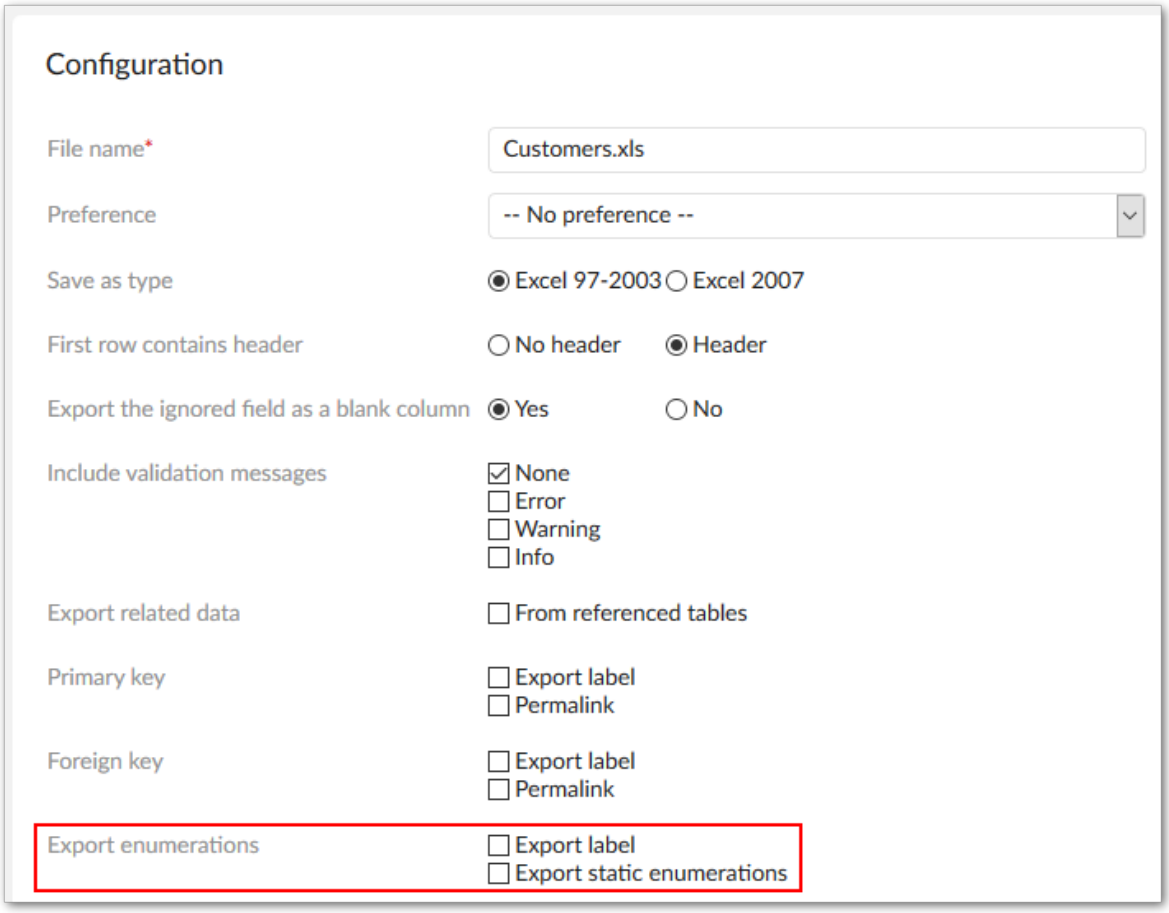
The following images highlights the options when exporting Excel:



> **Note**
>
> The export only includes data on which you have sufficient permissions.

4.7 **Enumeration export options**

The add-on allows you to include enumerations in CSV and Excel exports of tables, or individual records. As highlighted below, the options are available from the main configuration page (the image shows Excel export, CSV only includes the **Export label** option):

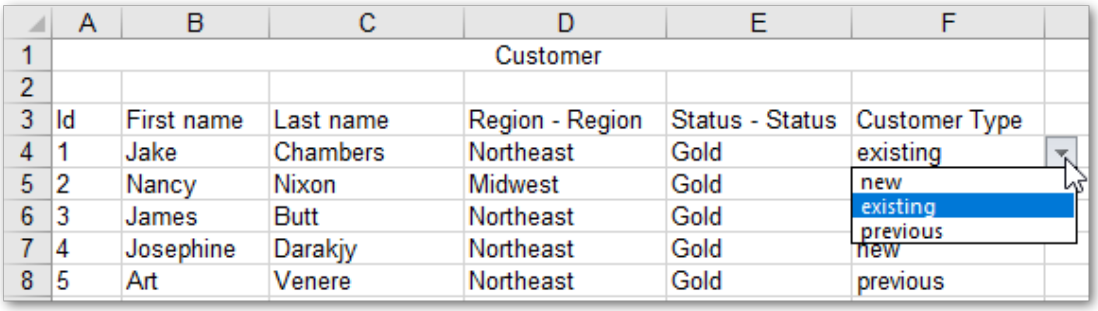


The following describes behavior for each supported export type:

- **Excel** export: On the main configuration page, the add-on presents you with the following options under **Export enumerations**:
  - **Export label**: The export includes an additional column for the enumeration field's label.

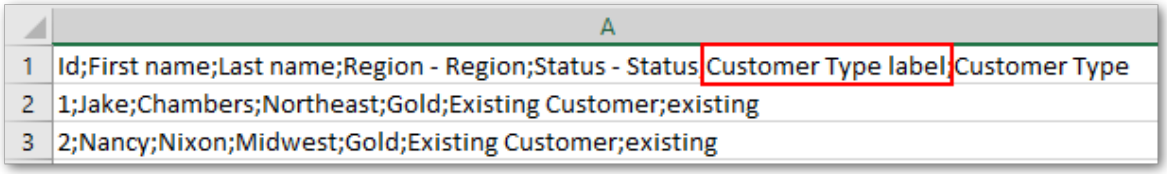
	A	B	C	D	E	F	G
1	Customer						
2							
3	Id	First name	Last name	Region - Region	Status - Status	Customer Type label	Customer Type
4	1	Jake	Chambers	Northeast	Gold	Existing Customer	existing
5	2	Nancy	Nixon	Midwest	Gold	Existing Customer	existing
6	3	James	Butt	Northeast	Gold	New Customer	new
7	4	Josephine	Darakjy	Northeast	Gold	New Customer	new
8	5	Art	Venere	Northeast	Gold	Previous Customer	previous

- **Export static enumerations**: The the enumeration column will contain a drop-down list of enumeration values defined in the data model.



	A	B	C	D	E	F
1			Customer			
2						
3	Id	First name	Last name	Region - Region	Status - Status	Customer Type
4	1	Jake	Chambers	Northeast	Gold	existing
5	2	Nancy	Nixon	Midwest	Gold	new
6	3	James	Butt	Northeast	Gold	existing
7	4	Josephine	Darakjy	Northeast	Gold	previous
8	5	Art	Venere	Northeast	Gold	new
						previous

  - When you select both options, the exported columns both have drop-down lists of their available values.
  - If you leave both options unchecked, the exported column includes only the enumeration value defined in the data model.
- **CSV** export: The exported file will include a dedicated column with the enumeration value's labels.



	A
1	Id;First name;Last name;Region - Region;Status - Status;Customer Type label;Customer Type
2	1;Jake;Chambers;Northeast;Gold;Existing Customer;existing
3	2;Nancy;Nixon;Midwest;Gold;Existing Customer;existing

CHAPTER 5

Transferring data

This chapter contains the following topics:

5.1 Overview of data transfer

The EBX® Data Exchange Add-on allows you to transfer data between EBX® datasets and tables. The following transfer options are available depending on the source and target locations:

- When the source and target locations are based on the same data model, the add-on can automatically handle required mapping. You can even use the transfer data functionality to transfer data to the same table. This can be useful if duplication is required, or you want to modify table data using a transformation.
- When the source and target locations are based on different data models, an administrator must create a custom mapping configuration. See Overview [p 93] for more information. Additionally, administrators must setup any configurations required for data transformation.

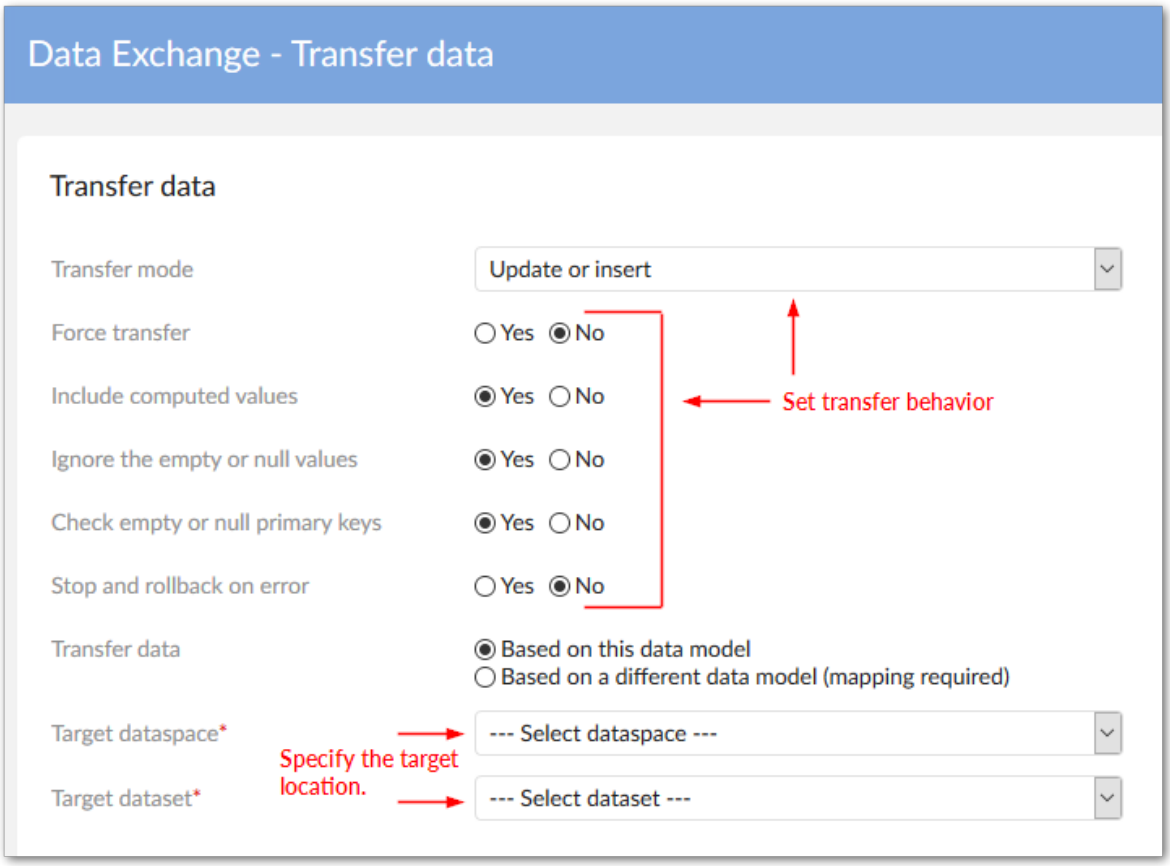
The following actions are available, depending on how you run the service:

- When you run the service from a dataset's **Actions** menu, you'll have the option of transferring data from one, or more tables. Also, you can choose one or more target tables for each source table. All data from the selected tables is transferred.
- When you run the service from a table's **Actions** menu (no records selected), you can choose one or more target tables. All table data is transferred.
- If you select at least one record and run the service from a table's **Actions** menu, you can still choose the target table, however only the selected records will be transferred.

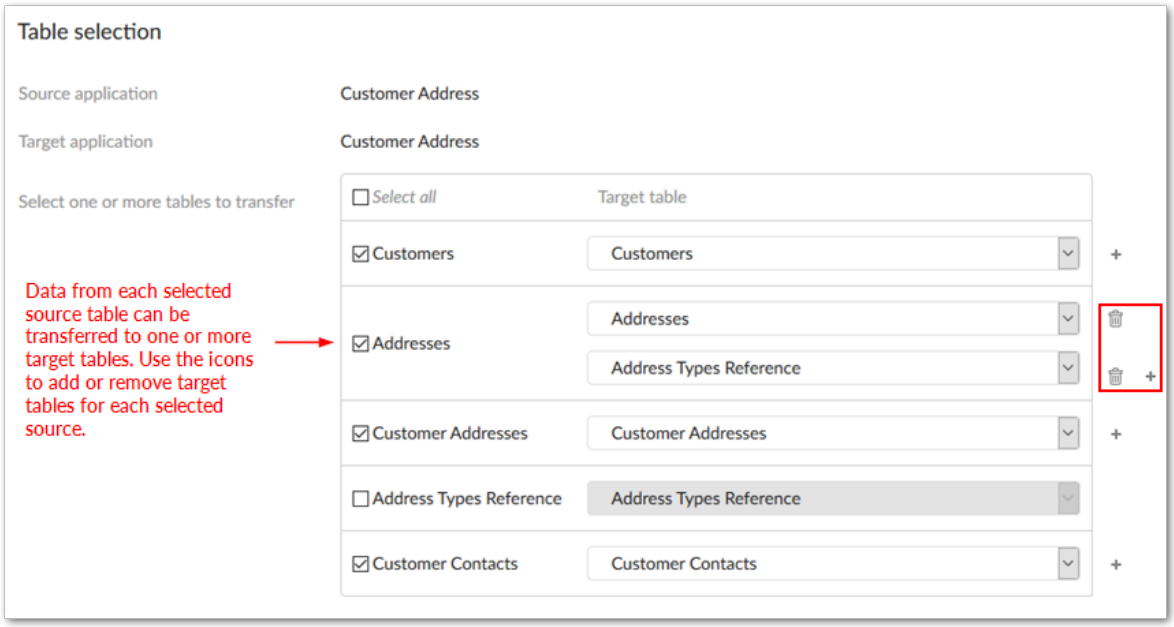
Attention

If you are transferring data to a table that has its primary key set to read-only, you can run the transfer in **Update or insert** mode. In this mode, records will be updated, but an error message will display if an attempt is made to transfer a a record that doesn't exist in the target table.

The image below highlights configuration options.



The next page in the import operation allows you to select a target table, or tables, for the transfer. The following image shows the table selection page with multiple tables.



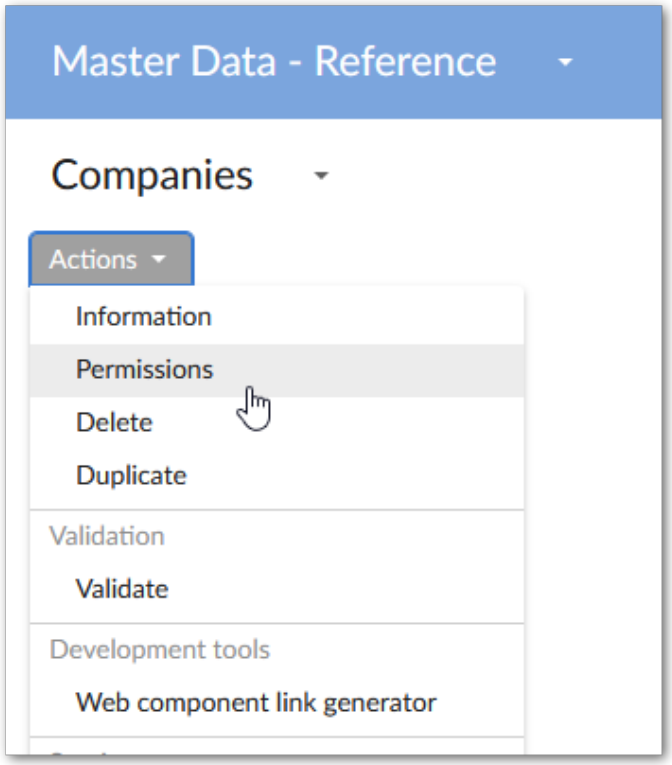
CHAPTER 6

Exporting dataset permission data

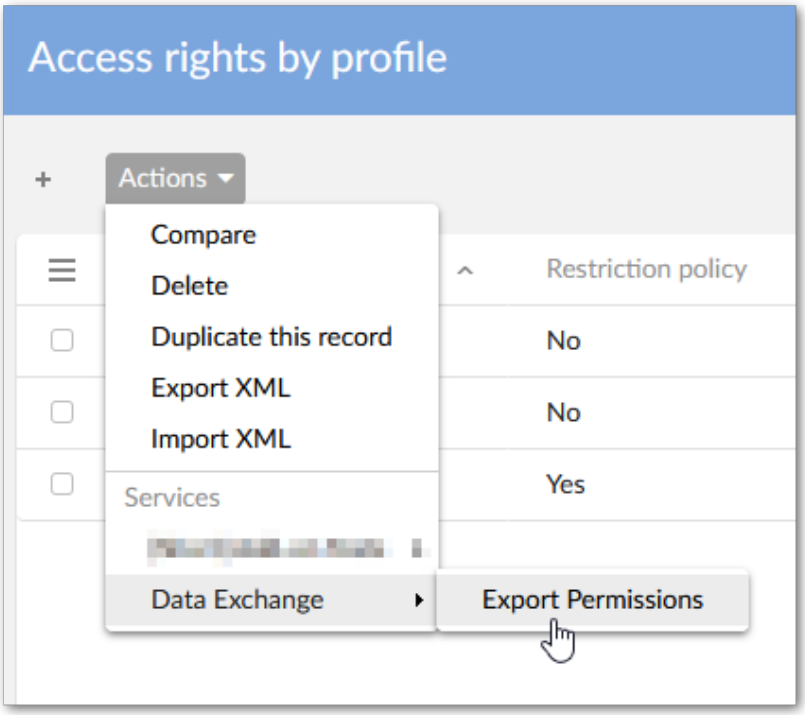
The add-on allows you to export a dataset's permission settings data to an Excel file. Each tab and tab order in the exported file corresponds to the records and order in the **Access rights by profile** table.

To export a dataset's permission settings data:

1. Navigate to the desired dataset.
2. From the dataset's **Actions** menu, select **Permissions**.

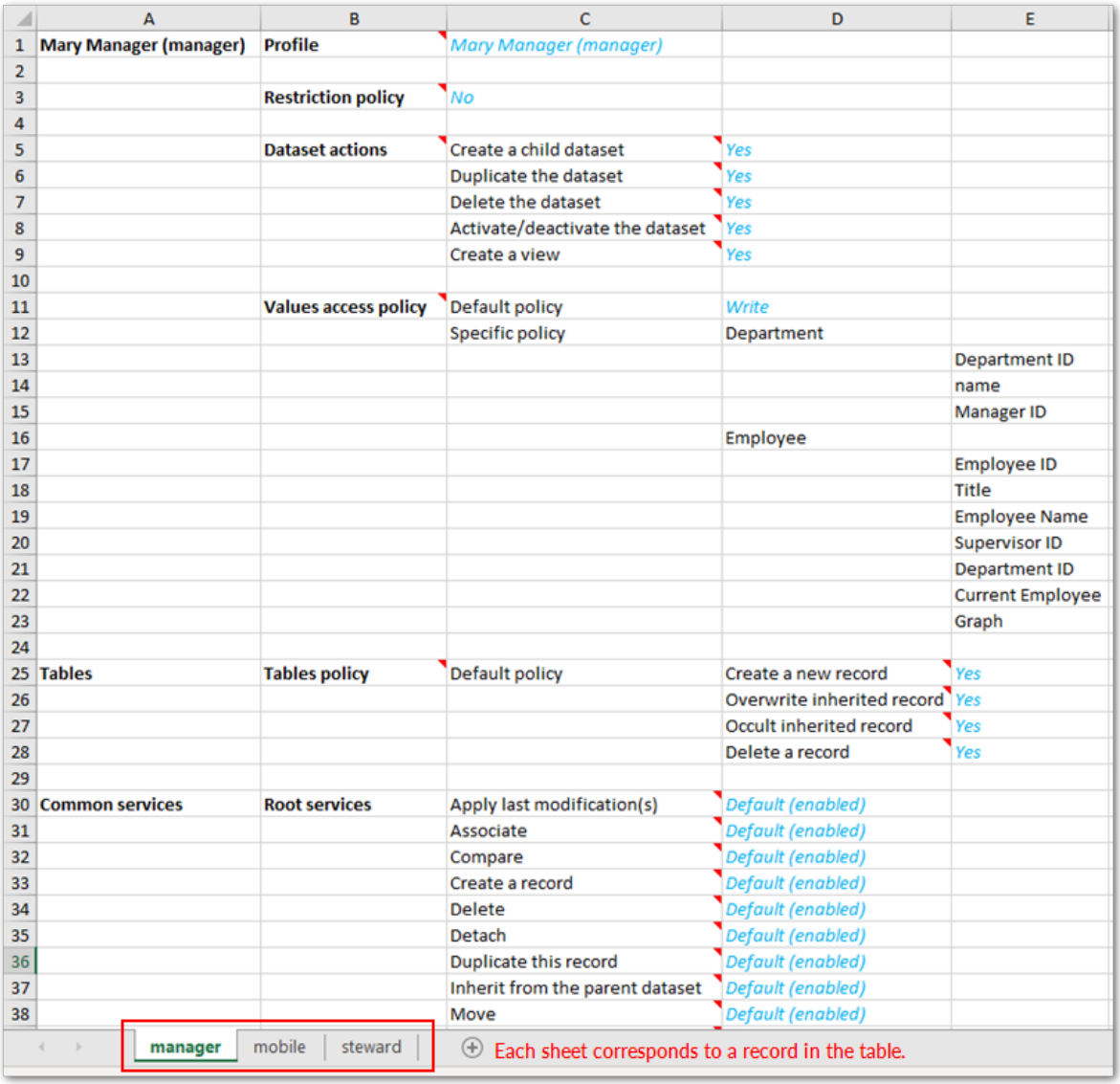


3. From the **Access rights by profile** table's **Actions** menu, select *Data Exchange* > *Export Permissions*.



4. Optionally, update the file name and then select **Export.**

The image below shows an example of an exported file:



	A	B	C	D	E
1	**Mary Manager (manager)**	**Profile**	*Mary Manager (manager)*		
2					
3		**Restriction policy**	*No*		
4					
5		**Dataset actions**	Create a child dataset	*Yes*	
6			Duplicate the dataset	*Yes*	
7			Delete the dataset	*Yes*	
8			Activate/deactivate the dataset	*Yes*	
9			Create a view	*Yes*	
10					
11		**Values access policy**	Default policy	*Write*	
12			Specific policy	Department	
13					Department ID
14					name
15					Manager ID
16				Employee	
17					Employee ID
18					Title
19					Employee Name
20					Supervisor ID
21					Department ID
22					Current Employee
23					Graph
24					
25	**Tables**	**Tables policy**	Default policy	Create a new record	*Yes*
26				Overwrite inherited record	*Yes*
27				Occult inherited record	*Yes*
28				Delete a record	*Yes*
29					
30	**Common services**	**Root services**	Apply last modification(s)	*Default (enabled)*	
31			Associate	*Default (enabled)*	
32			Compare	*Default (enabled)*	
33			Create a record	*Default (enabled)*	
34			Delete	*Default (enabled)*	
35			Detach	*Default (enabled)*	
36			Duplicate this record	*Default (enabled)*	
37			Inherit from the parent dataset	*Default (enabled)*	
38			Move	*Default (enabled)*	

CHAPTER 7

Setting import and export permissions

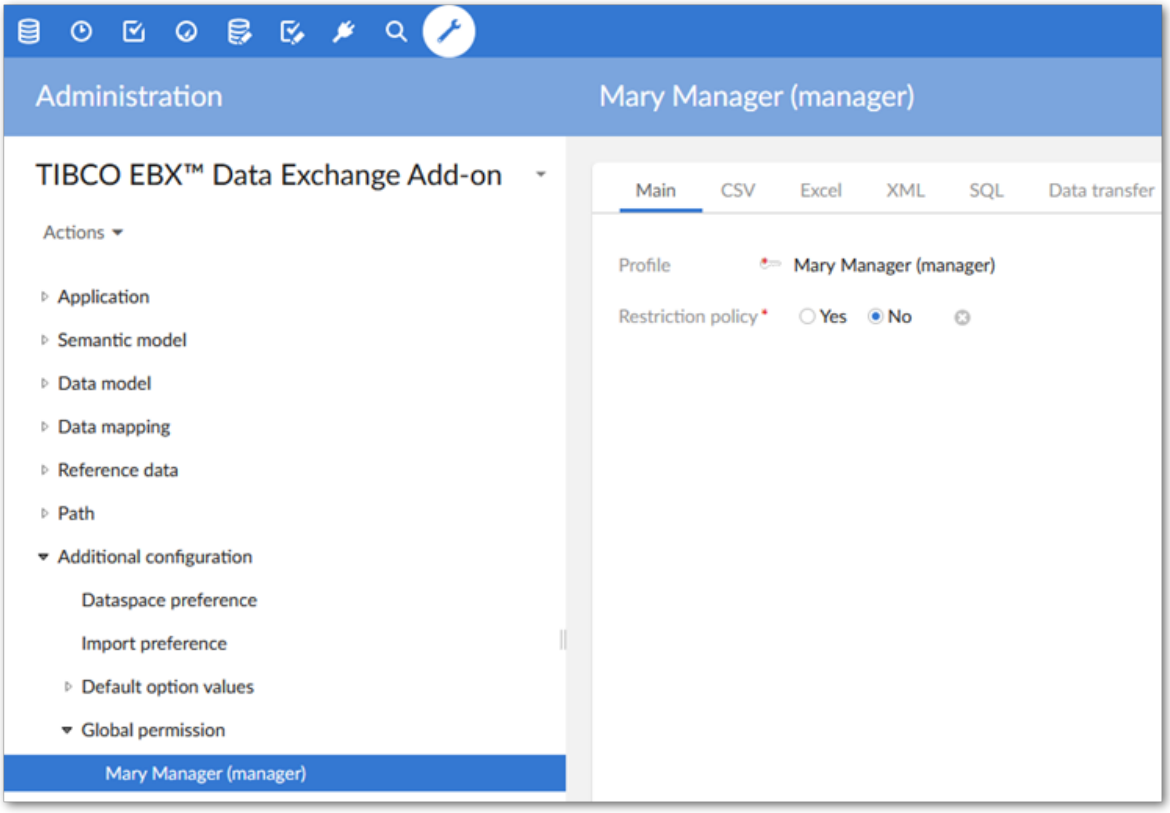
You can apply EBX® permissions to a profile, which is associated with either a user or a role. Each user can have multiple roles, and each role can include multiple users. If more than one set of permissions is associated with the same profile, the **Restrictive policy** setting determines whether the least restrictive or most restrictive policy settings apply. The EBX® Data Exchange Add-on allows you to leverage EBX® behavior related to permissions and apply it to the act of exchanging data. Specifically, administrators can set permissions for users and roles that determine:

- access and interaction with import and export screens. By default all users can access the **Configuration**, **Mapping**, and **Simulation** screens. Administrators can specify that these pages are read-only accessible, or hidden from specific user profiles.
- the default mode or the modes users can choose when importing and transferring data. The modes determine whether an import or transfer operation updates, inserts, or deletes data in the target location.

To set import and export permissions:

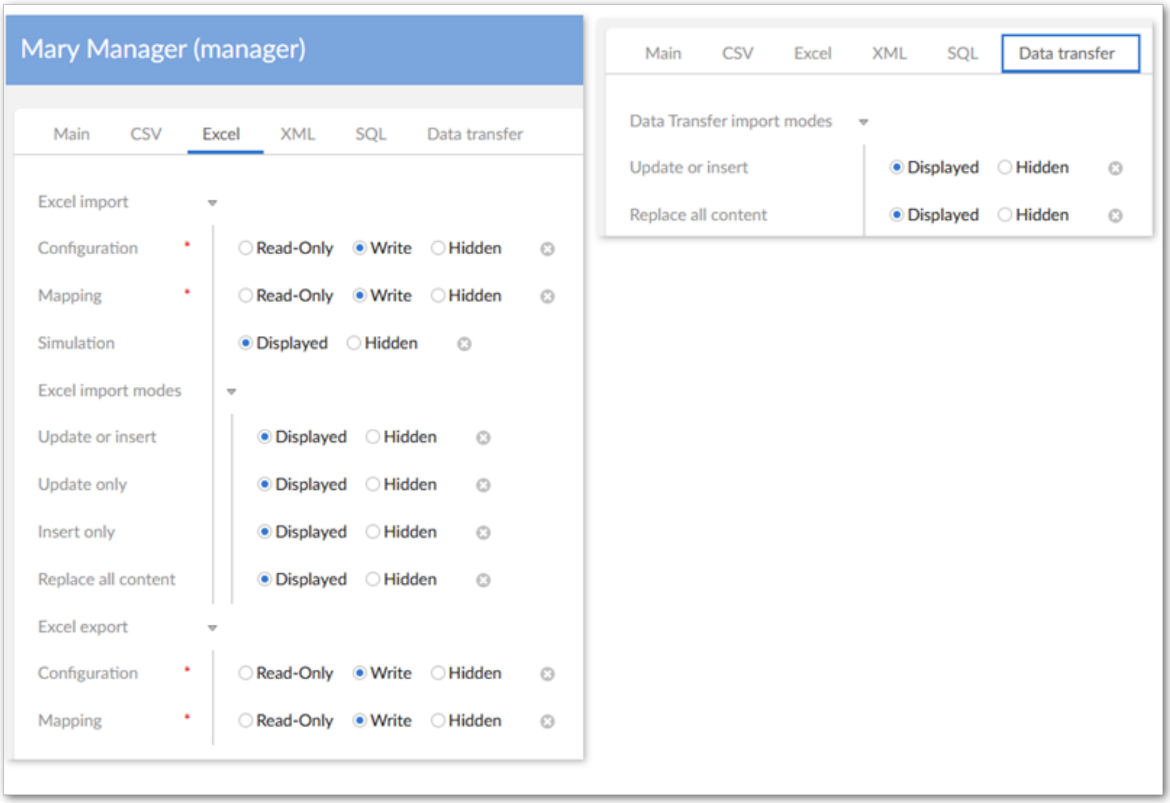
1. Navigate to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Additional configuration* > *Global permission* and create a new record.

2. On the **Main** tab you specify which profile these configuration settings will apply to and whether this will be considered a restrictive policy. If the **Restriction policy** option is enabled, the settings in this configuration are applied wherever they are more restrictive than others.



3. Use the tabs to set permissions for each import, export and transfer format as follows:
   - **Import:** Determines what screens users can access during the import process. Additionally, you can specify what import modes users can apply when importing.
   - **Export:** Determines what screens users can view.

- **Transfer:** Specifies the data transfer modes users can apply when transferring data.



CHAPTER 8

Specifying permissions for add-on preferences

Application preferences store mapping configuration information. You can use an application preference's permissions to specify whether users can view, modify, or delete the preference. To update permissions:

1. Navigate to *Administration* > *Integration* > *EBX® Data Exchange Add-on* > *Application* > *Application interface preference*.
2. Create a new, or open an existing record.
3. Use the **Permissions** property to apply permission settings for this preference to a user profile. Setting **Use the preference** to **No** excludes the preference from the user's list of available preferences.

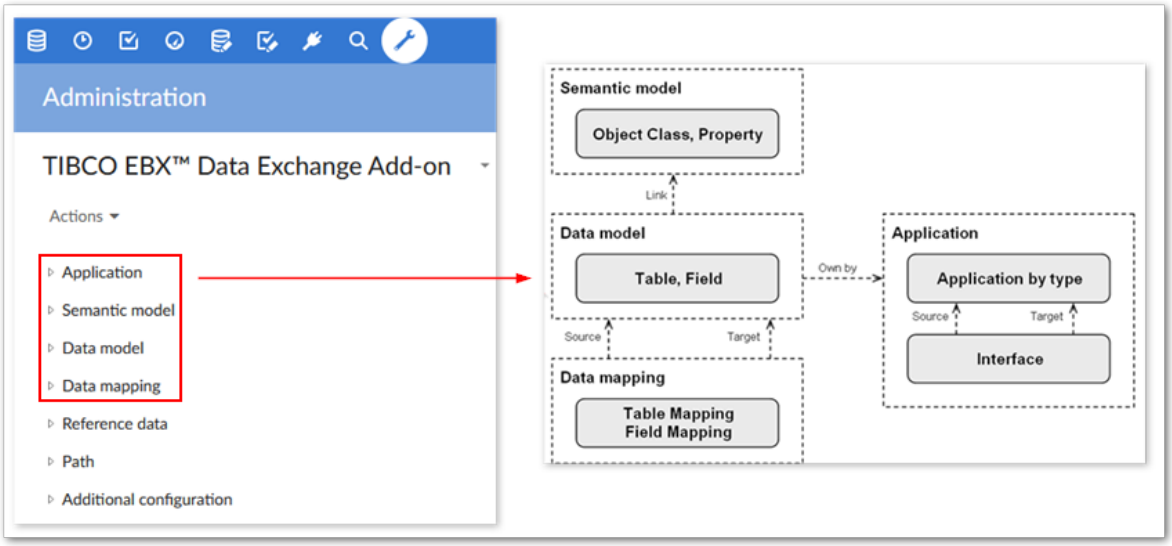
> **Note**
>
> The preference owner will not be listed in the **User profile** drop-down menu.

Mapping concepts

CHAPTER 9

Architectural overview

The EBX® Data Exchange Add-on uses a repository, containing several domains, to collect and manage all information related to data mapping configurations. The following image provides a high-level overview of main domains involved in exchanging data. The table in the Defining mapping concepts [p 60] describes basic functions of the tables in these domains.



CHAPTER 10

Defining mapping concepts

The following table defines key concepts surrounding the main tables used to configure mapping details:

Concept	Definition
Application	In the EBX® Data Exchange Add-on, applications represent the source and target for exchanging data. You store the definition of these applications in the **Application** table.
Application by type	An application type declares the format of the source or target. One application can have multiple types. Some examples are: • EBX: the application corresponds to a data model in EBX®—referenced through a dataset. • Default XML: the application corresponds to a XML data structure fully managed by the add-on with default XML paths. • XML: the application corresponds to a XML data structure issued in the user-defined data mapping configurations. • Default SQL: The application corresponds to an external SQL data source. • CSV: The application corresponds to a CSV formatted file. Note that the file does not strictly have to use the comma character as the value separator. • Excel: The application corresponds to an Excel workbook file (.xsl, .xslx).
Interface	An interface specifies declares the source and target of data exchange. Import, export and transfer of data is not possible between applications without an interface definition. The add-on creates an interface when the end-user executes a default SQL/XML import or export.
Semantic model	The semantic model consists of Object Classes and Properties which store metadata about tables and fields. The add-on can automatically generate mappings between source and target for tables and fields that are linked to the same Object Classes and Properties. Additionally, you can use semantic model metadata to provide data lineage view. The semantic model is described in greater detail in Semantic model [p 63].
Object Class	An Object Class is a container that holds metadata such as a table, a group of fields or a complex data type. An Object Class can be linked to one or more tables.
Property	A Property is a business abstraction of a table's field, group of fields or complex data type. A Property can be linked to one or more Object Classes and fields.
Data model	A data model provides a logical layer to link the EBX® Data Exchange Add-on application with physical tables and fields and optionally the semantic model. For example, EBX type applications link to published EBX® data models. The EBX® Data Exchange Add-on **Data model**'s tables and fields create the link to

Concept	Definition
	the EBX® table and field paths. In contrast, an XML type application defines the corresponding paths in an XML file. See Custom data mapping [p 73] for an example of generating data models.
Table	When referring to an EBX type application, this is the table in the related logical data model. When referring to an XML type application, this is an XML path.
Field	When referring to an EBX type application, this is the field in the related logical data model. When referring to an XML type application, this is an XML path.
Data mapping	The **Data mapping** group is where you create the mapping between source and target tables and fields.
Table mapping	Defines the source and target tables.
Field mapping	Defines the sources and target fields.

CHAPTER 11

Semantic model

This chapter contains the following topics:

11.1 Semantic model

Data mapping configurations can be enriched by creating links from Tables and Fields to corresponding business concepts also known as *Object Class* and *Property* items. A single Object Class can be linked to many tables, and a Property can be linked to many fields. An Object Class is a container of data, predominantly a Table, but can include a group of fields or a complex data type. A property is a business abstraction of a table's field. This vocabulary comes from the ISO11179 standard also used in the *TIBCO EBX® Information Governance Add-on*.

The use of the semantic model is not mandatory to create data mapping configurations or to enable import, export and data transfer. Once the semantic model is configured, it facilitates the management of the data mapping configurations as follows:

- Business data lineage.
- Automatic data mapping.
- Integration with the governance process.

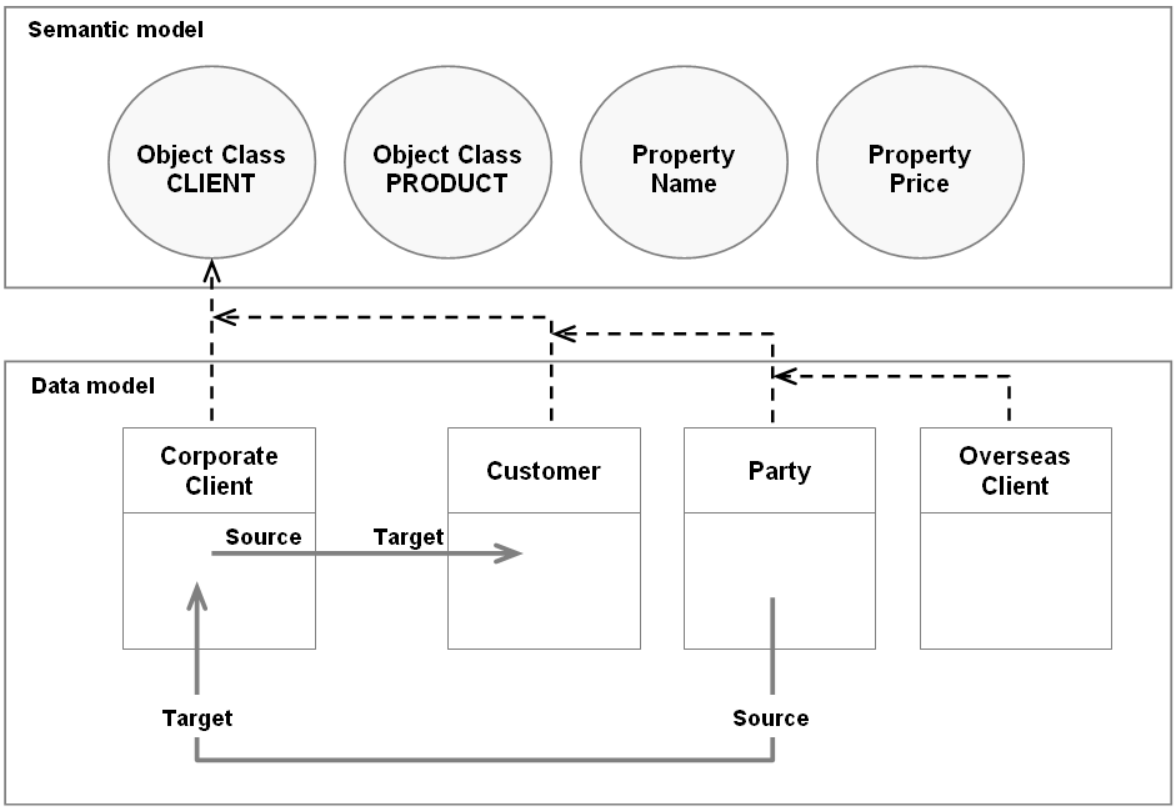
Business data lineage

Data lineage shows a global view of how a data is transformed and conveyed between applications playing the roles of producers and consumers. When data lineage applies solely to the logical data model level, it is not easy to enforce a full understanding of the transformation. The following shows how two tables can have indirect data flow processes that cannot be figured out using just the logical level:

- There are four tables: A, B, C and D. There is a data mapping from A to B and another mapping from C to D. Data lineage only recognizes two possible ways for data to flow A->B and C->D. But from a business point of view, table D and table A have similar significance. Even though a logical data mapping between A and D doesn't exist, the business data lineage must represent the global linking meaning between the four tables. It could be considered that there is a missing data mapping configuration between A and D.

In the following example, the **CLIENT** Object Class is linked to the **Corporate Client**, **Customer**, **Party** and **Overseas Client** tables. From this Object Class, it is easy to get a full data lineage applied

to the business concept of **Client**, including the information that the table **Overseas Client** is not involved to feed the **Customer** table.



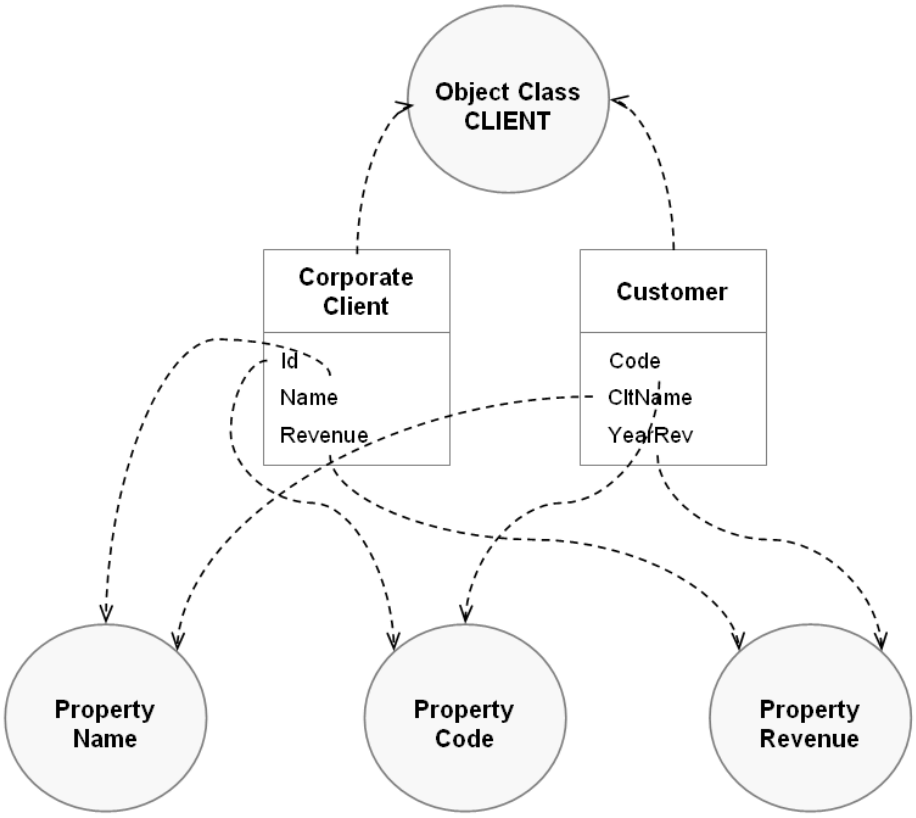
Special notation:	
✕	In the current version of the add-on, it is possible to link the Tables and Fields with the Object Class and Properties items, but the UI to display the data lineage is not yet available.

Automatic data mapping

The semantic model is also used to create automatic data mappings between tables and fields sharing the same Object Class and Property items or a part of them. As illustrated below, the two tables **Corporate Client** and **Customer** share the same **CLIENT** Object Class, meaning that they can be mapped with each other. The **Id** and **Code** fields are linked to the same **Name** Property, meaning that they can also be mapped with each other.

Based on this link between the data model Tables and Fields and the semantic model Object Classes and Properties, the add-on can automatically generate the data mapping configuration between source

and target tables, such as, in the example below, the **Corporate Client** table and the target **Customer** table.



> **Note**
>
> Transferring between tables based on different models [p 93] provides an example of a custom configuration that uses the semantic model to auto-generate mappings.

Integration with the governance process

In the **EBX® Information Governance Add-on** field, the *EBX Information Governance Add-on* allows you to manage all metadata of any data asset, such as: data models, workflow, rules, dataspaces, applications, etc. This add-on uses Object Class and Property items as concepts to arrange the metadata and govern their definitions.

It is also possible to declare the parties and their roles involved in each Object Class and Property. For instance, if the **Sales** application is referenced as the **Consumer** of the **Client** Object Class, it should be forbidden to declare this application as a source for an export process in the EBX® Data Exchange Add-on.

CHAPTER 12

Mapping tasks

This chapter contains the following topics:

12.1 Overview of tasks

Generally speaking, there are two types of mapping responsibilities:

- Customizing mappings automatically generated by the add-on. A majority of the time business users will be responsible for performing this task during the import, export, or transfer process.
- Using add-on functionality to create a user-defined mapping. This task must be performed by an administrator.

Business user tasks

When data structures are similar the add-on automatically proposes a mapping solution during transfer, or import/export. An example of a similar data structure is when importing from an Excel file and all column names in the file match those of the table to which they are imported. If the structure is not similar, or users want to change the generated mapping, they can do so on the mapping pages presented during the operation.

Attention

A pro-tip for users who often have to change the auto-generated mapping is to create preferences to save their changes for later re-use.

See also

Overview *[p 15]*

Export overview *[p 29]*

Overview of data transfer *[p 45]*

Administrative tasks

Administrators must create custom mapping solutions in the following scenarios:

- To enable users to transfer data (within EBX®) between a source and target based on different EBX® data model publications.

- To enable import, or export from an XML file that has a data structure that is not compliant with the *Default XML* type.
- To enable data transformation during transfer, and CSV, Excel, and XML based operations.

See also

Custom data mapping *[p 73]*

Generating constant values *[p 85]*

Transferring between tables based on different models *[p 93]*

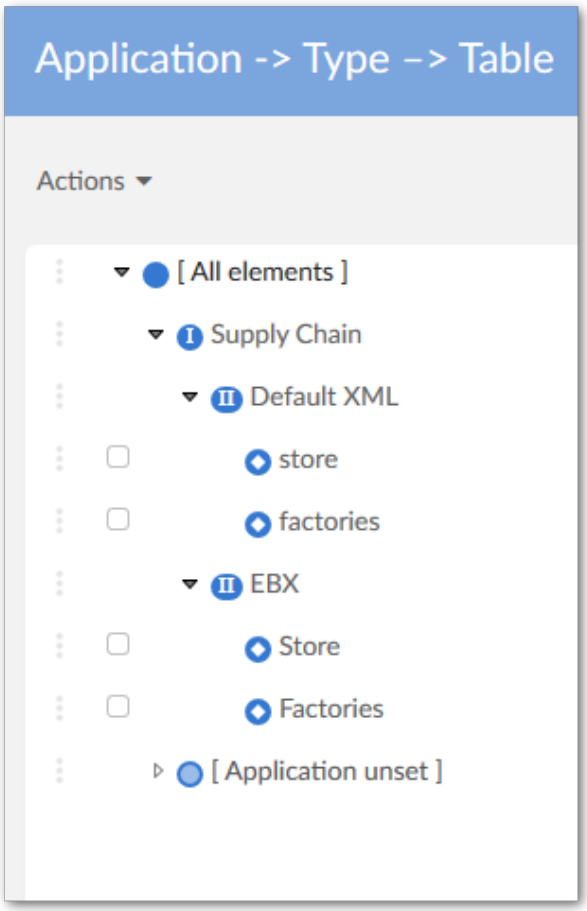
Advanced Data Exchange Options

CHAPTER 13

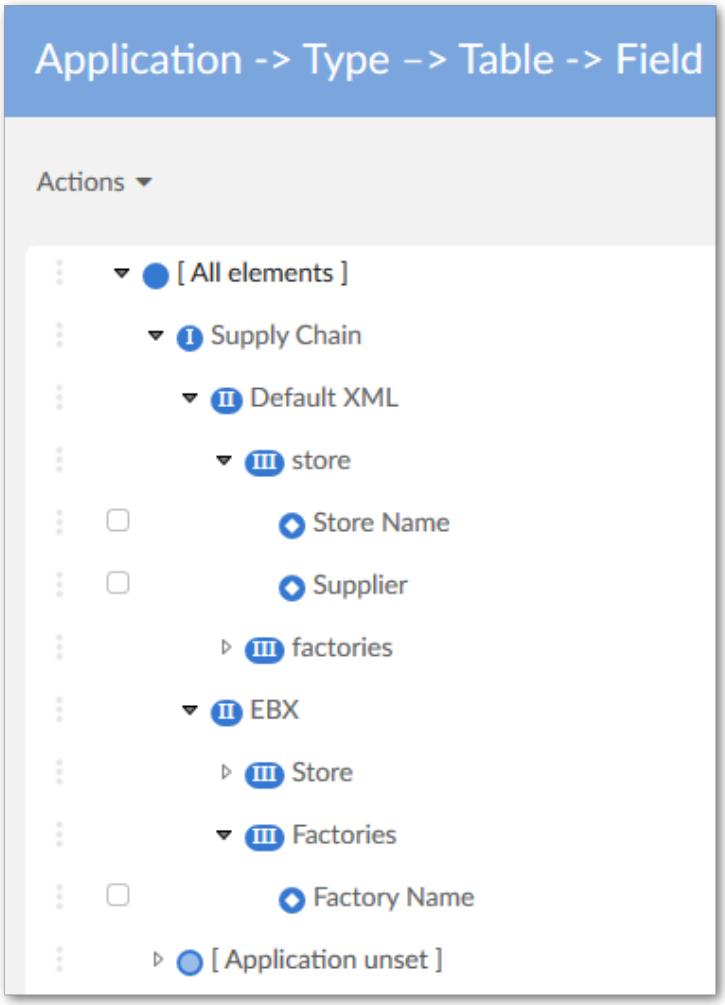
XML default mapping overview

When the XML export execution process is based on the default XML configuration, the add-on generates the related data mapping configuration. It contains the correct XML paths used for each of the exported table's fields. Even though this default XML configuration cannot be modified, understanding its architecture can be of benefit to you. Indeed, if you need to configure a user-defined data mapping, the same architecture applies (see the rest of the user guide).

The following images give an overview of how this configuration is saved in the add-on's configuration. You can access the configuration from *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on*.



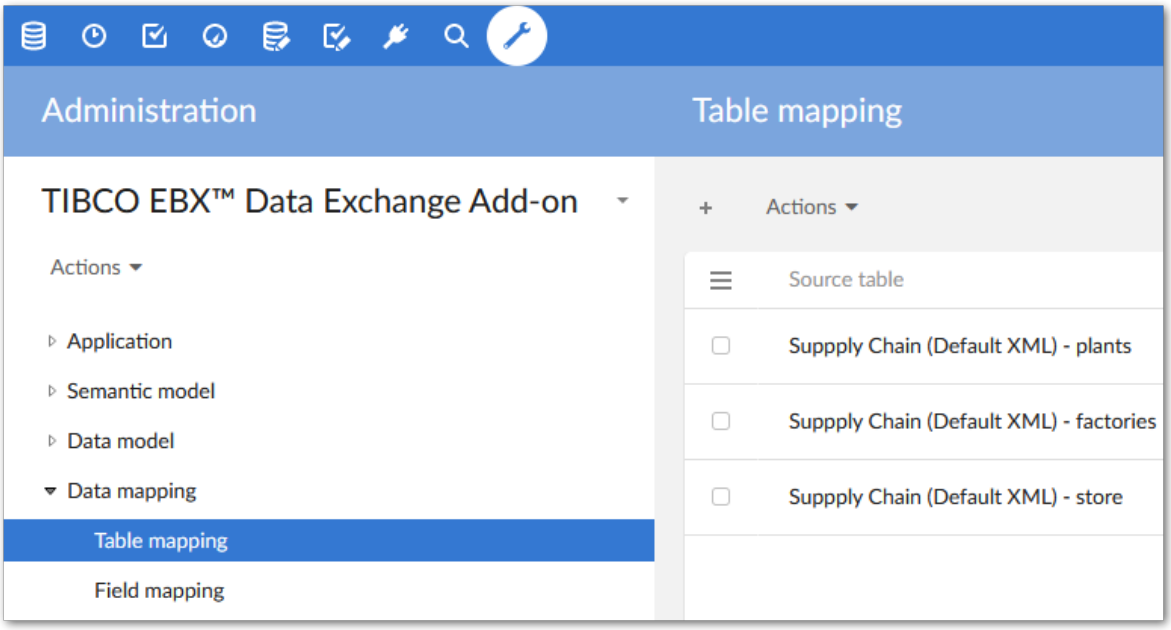
The **Supply Chain** (name of the dataspace where the table is located) application is created with the two application types, **Default XML** and **EBX**. For the **Default XML** type, only the exported tables are declared (**store** and **factories**). For the **EBX** type, all tables and field groups located in the dataspace are declared.



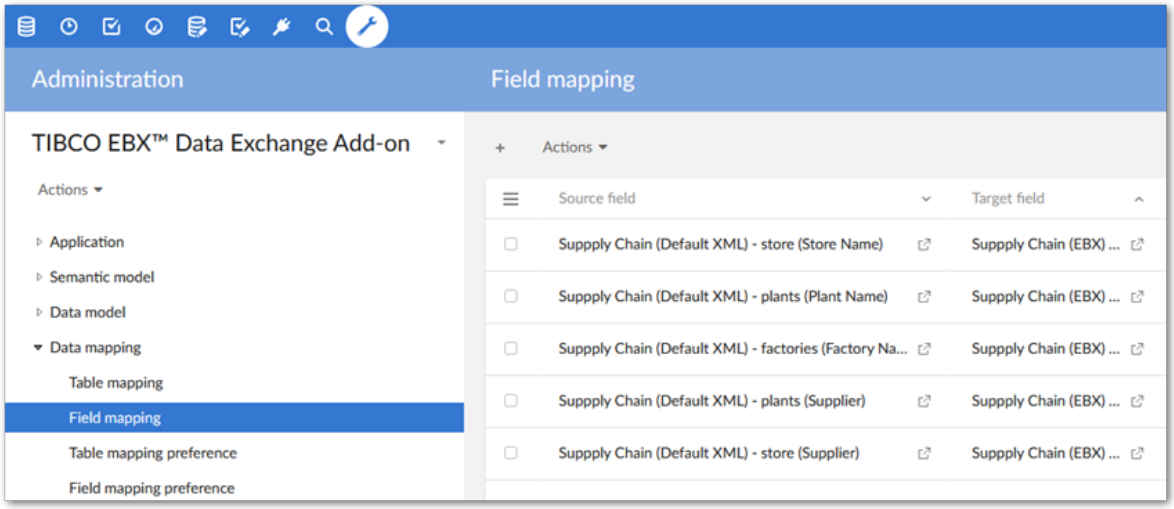
The declaration of the XML nodes for the **store**, **factories**, and **plants** tables is done with the default XML naming convention (direct reuse of the naming from the table in EBX®).

The declaration of the fields for every table in EBX® is also performed.

The **Table mapping** is automatically provided. The source table has been declared as EBX for the export process. And the source table has been declared as **Default XML** for the import process.



Then the add-on automatically creates the declaration for the **Field mapping**, thus providing the link between the source and the target file.



CHAPTER 14

Custom data mapping

This chapter contains the following topics:

14.1 User-defined XML mapping for export

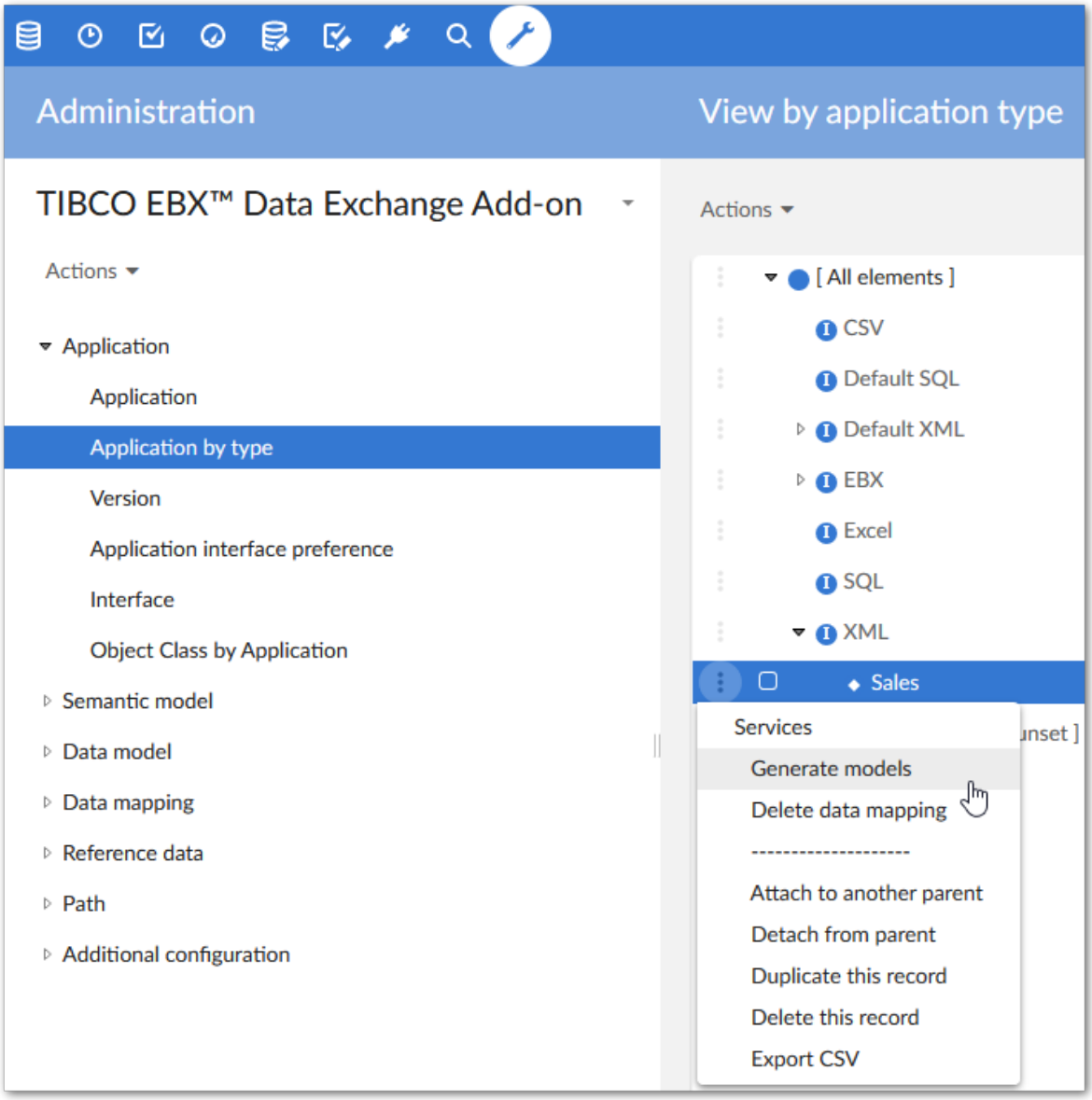
Certain use cases require you to create custom data mappings. The options may vary, but the process is similar for mapping between different source and target formats. This section provides an example using XML and you can apply the same principals to create custom mappings for other formats.

The following section describes a user-defined data mapping.

XML export

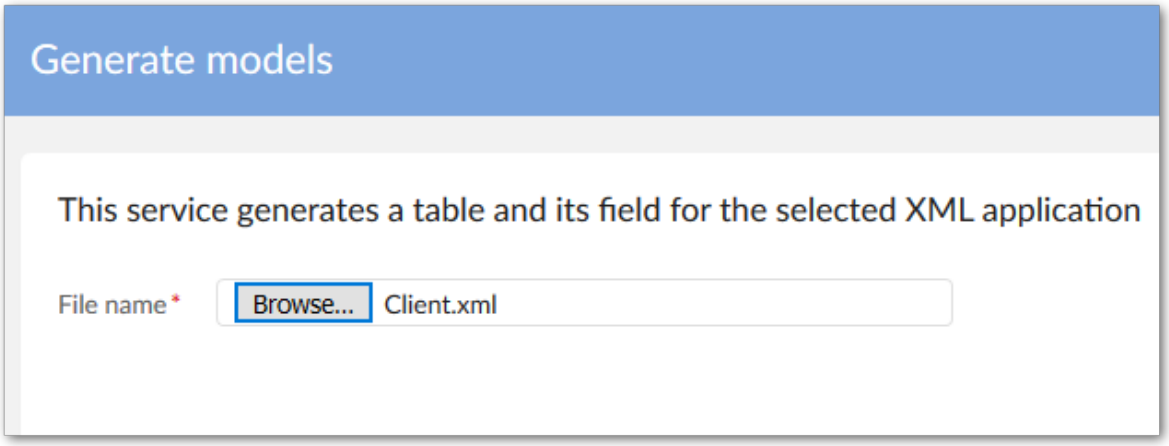
To export an XML file with a data structure that is not compliant with the Default XML type, you can manually configure a user-defined data mapping.

To facilitate the configuration, the add-on can automatically analyze your XML file to generate the XML configuration as illustrated below. In most cases the XML file you want to get already exists, and you can reuse it as a template to automatically feed the configuration.

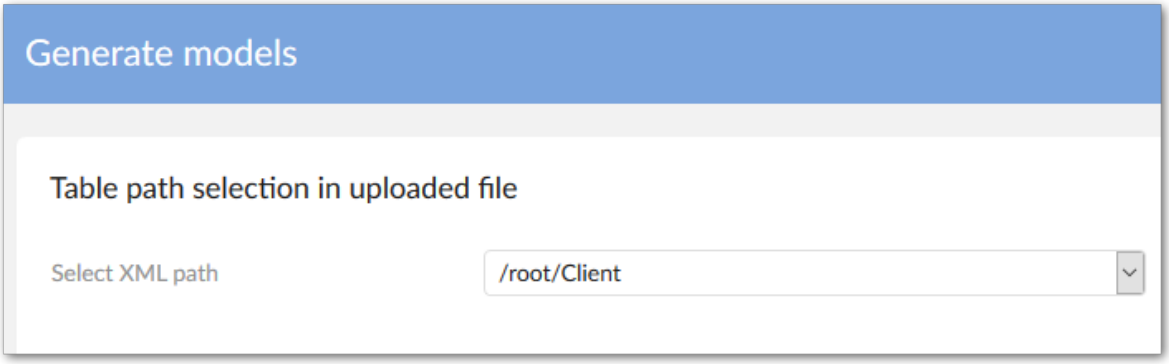


First, a new XML type application has been created manually. In the following image, this is the Sales application. Since the type is XML (not Default XML), the add-on's **Generate models** service is available and allows you to get the XML configuration automatically.

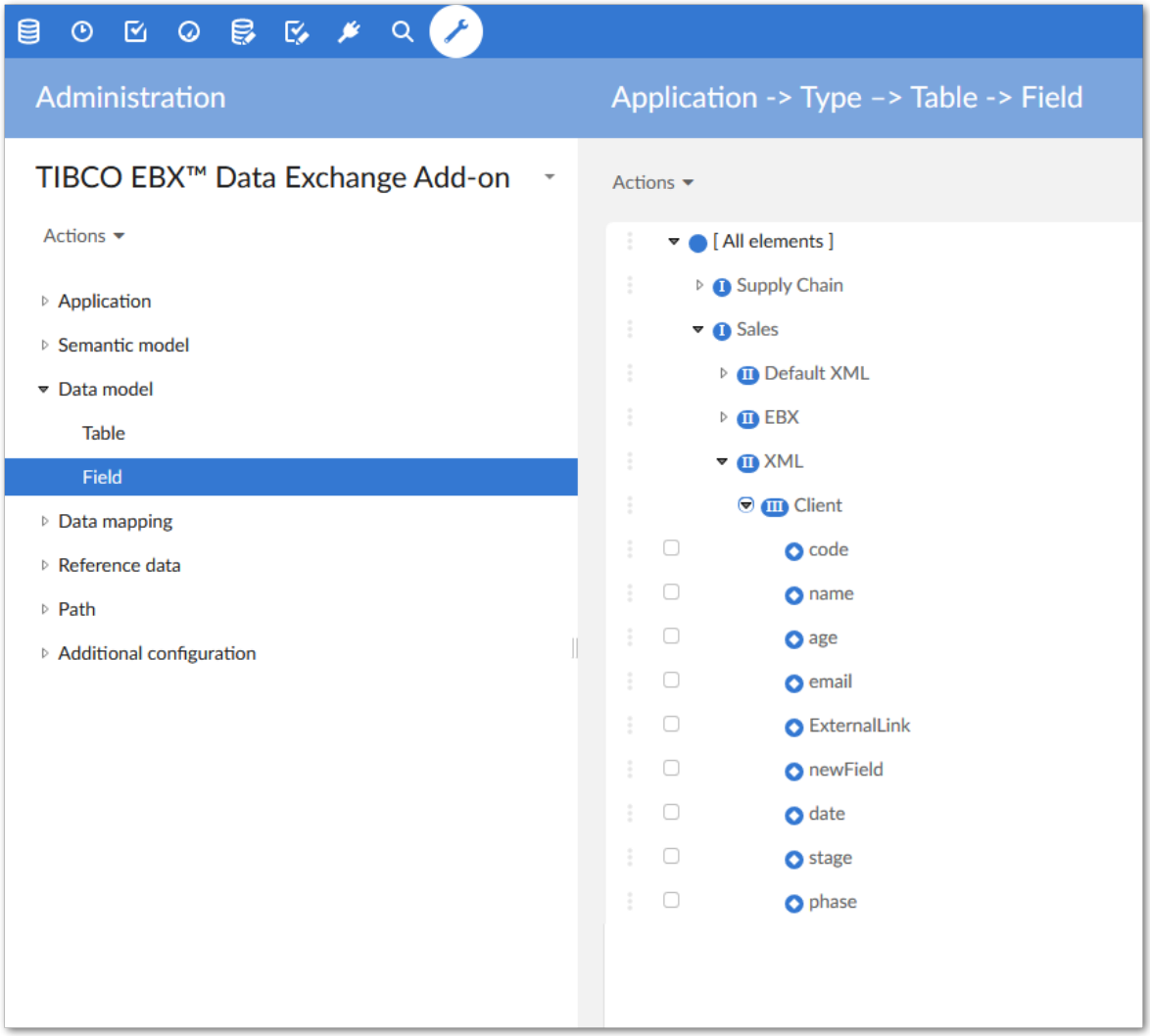
The add-on allows you to enter the XML file that will be used to create the XML configuration.



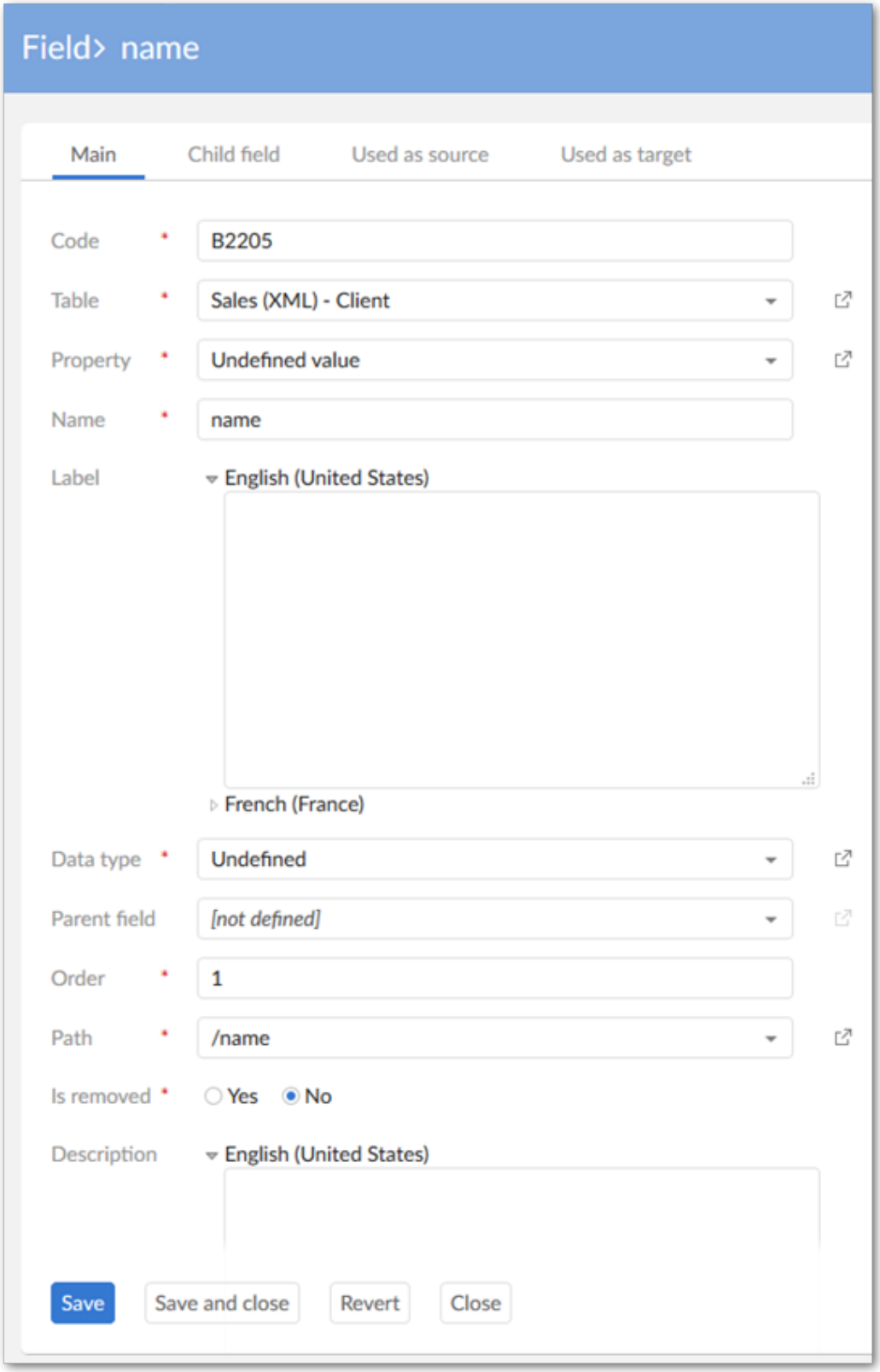
Then you select a node path in the XML file from which the XML configuration must be applied.



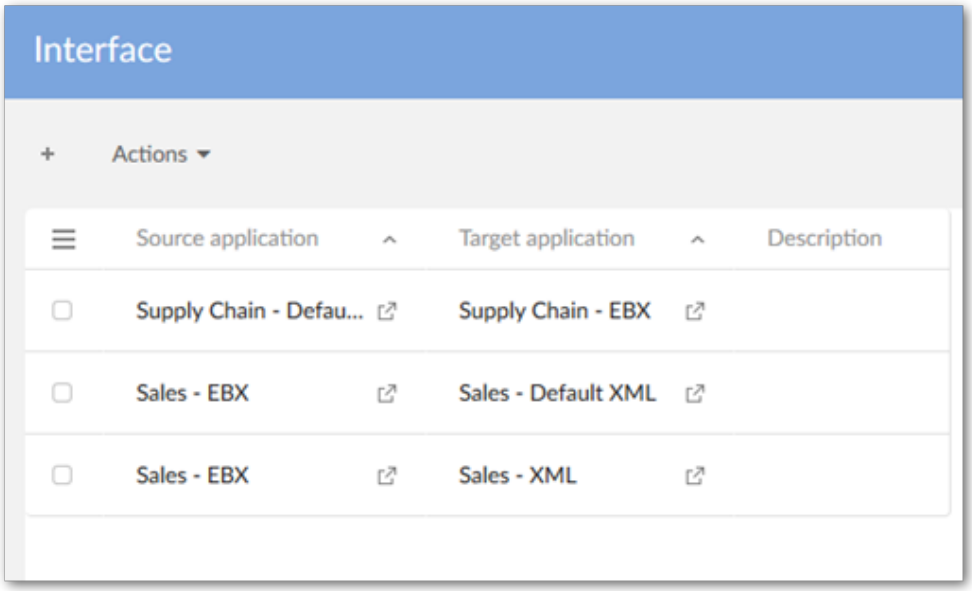
The result of the configuration is displayed below. The Sales XML type application is now declared with the fields corresponding to the XML tags existing in the file.



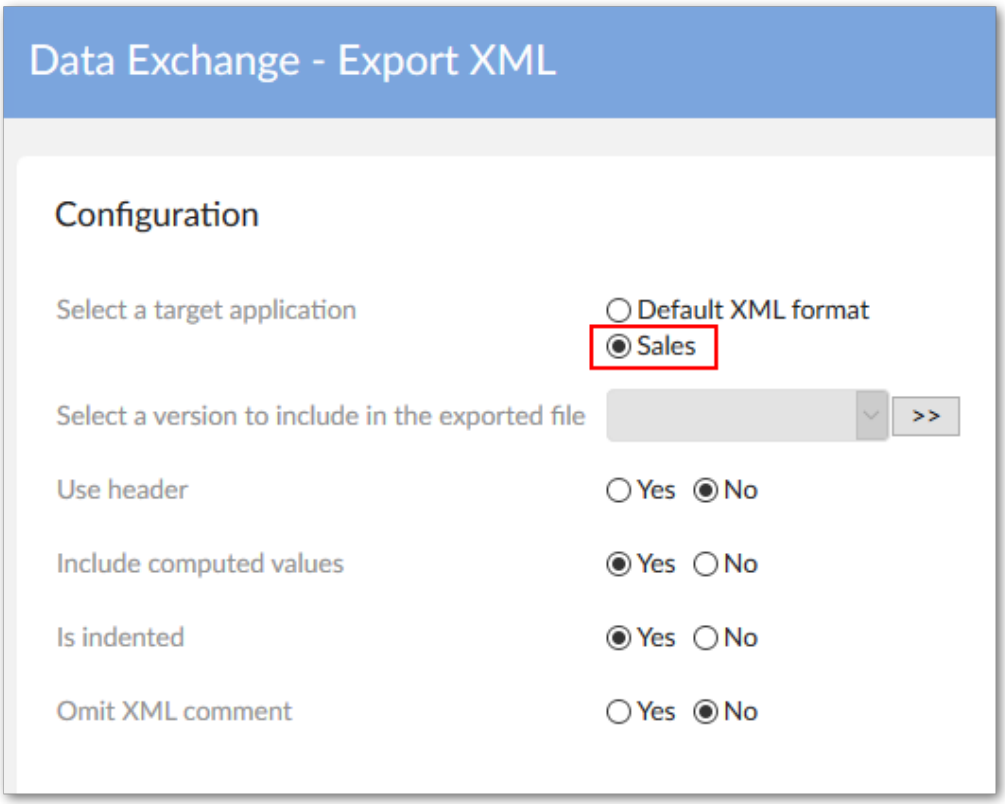
The configuration of a field with its XML path can be adapted manually as illustrated below. You can change the name, the path, etc. Conversely, in the case of a configuration based on the Default XML, the names and paths cannot be modified and are under add-on control.



To make the export process available from the EBX type application (**Sales** dataspace) to the Sales application, the **Interface** between both must be declared as follows:

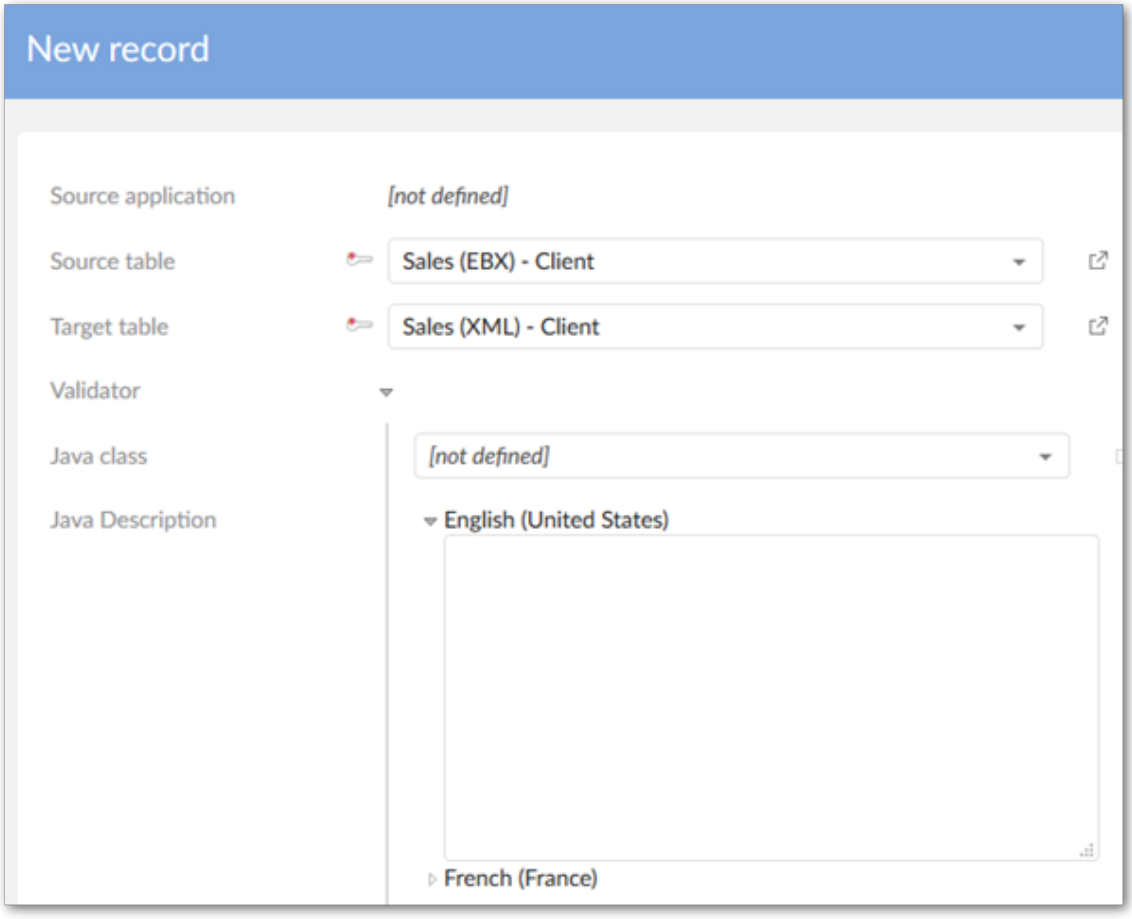


Now, the export UI displays the Sales target application as an option:



The options are already described in the previous section. However, when executing the export process based on the above configuration an error is raised.

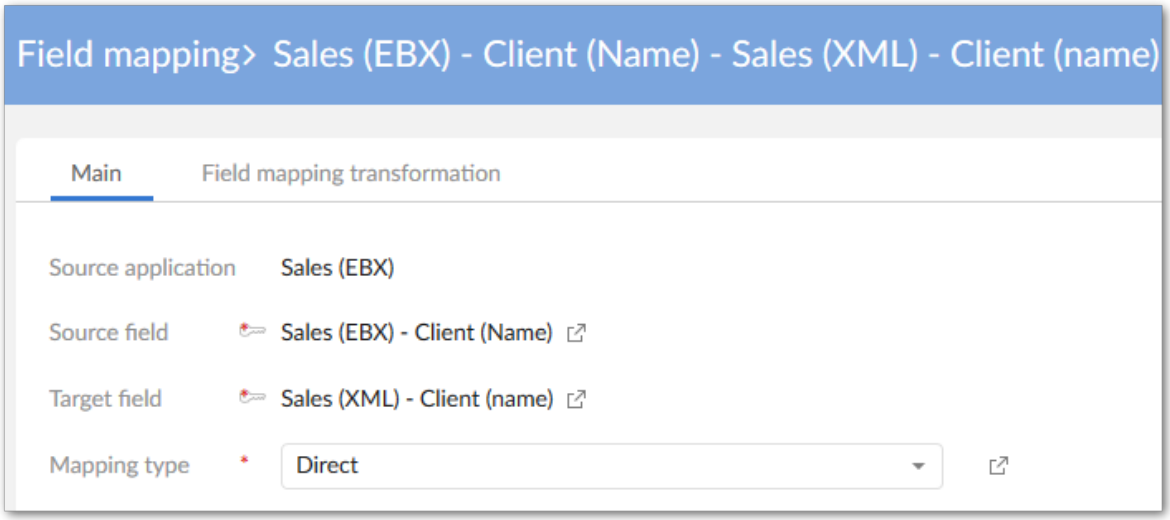
Indeed, even though the XML field configuration is done, the **Table mapping** and **Field mapping** are not yet configured. The table mapping is declared as follows by creating a new record in the **Table mapping** table:



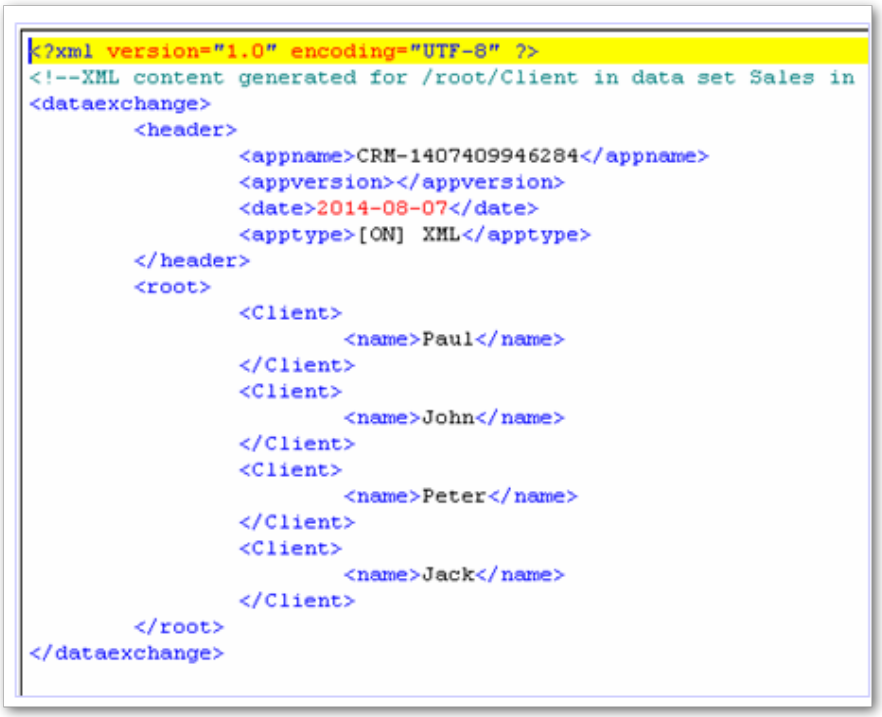
A new execution of the export process entails a useless result because the fields are not exported until the **Field mapping** is configured:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--XML content generated for /root/Client in data set Sale
<dataexchange>
        <header>
                <appname>CRM-1407409946284</appname>
                <appversion></appversion>
                <date>2014-08-07</date>
                <apptype>[ON] XML</apptype>
        </header>
        <root>
                <Client>
                </Client>
                <Client>
                </Client>
                <Client>
                </Client>
                <Client>
                </Client>
        </root>
</dataexchange>
```

For every field, the mapping from the source to the target is declared as follows:

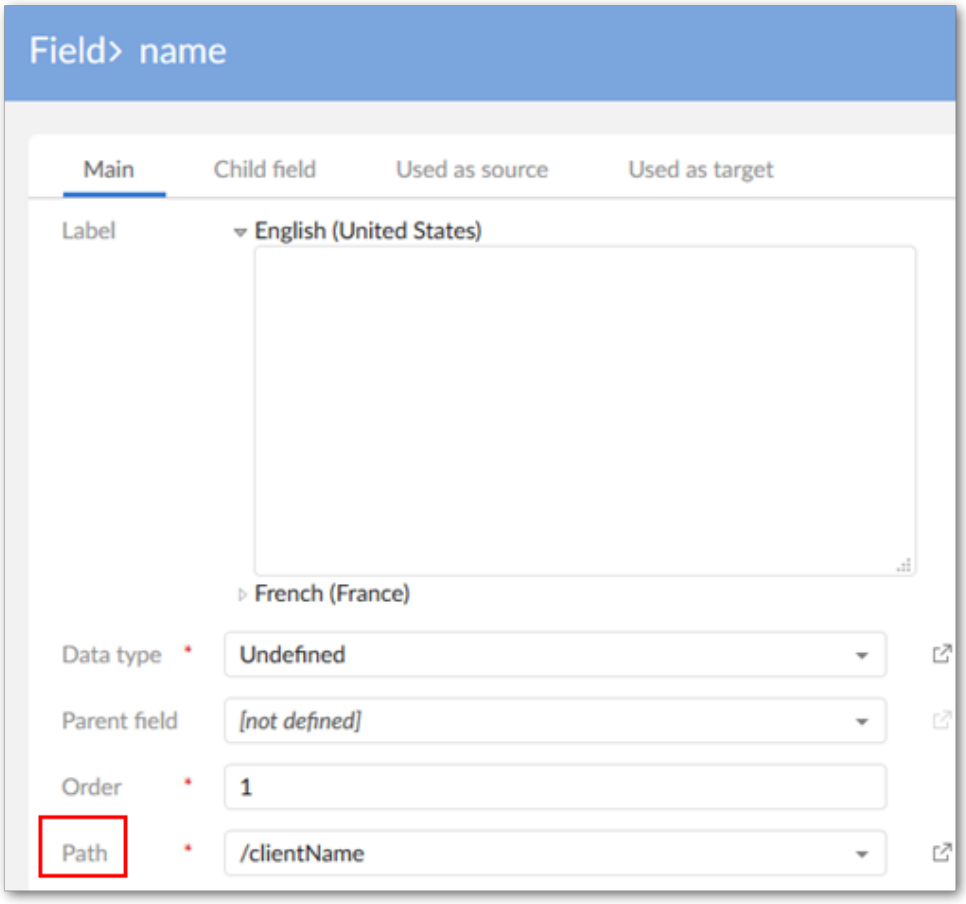


Now, the export process integrates the fields that have been configured (in this example only the name is declared in the **Field mapping** table).



```
<?xml version="1.0" encoding="UTF-8" ?>
<!--XML content generated for /root/Client in data set Sales in
<dataexchange>
        <header>
                <appname>CRM-1407409946284</appname>
                <appversion></appversion>
                <date>2014-08-07</date>
                <apptype>[ON] XML</apptype>
        </header>
        <root>
                <Client>
                        <name>Paul</name>
                </Client>
                <Client>
                        <name>John</name>
                </Client>
                <Client>
                        <name>Peter</name>
                </Client>
                <Client>
                        <name>Jack</name>
                </Client>
        </root>
</dataexchange>
```

To change the path of a field, a direct modification in the path configuration can be done as illustrated below. The initial path **`/Client`** has been changed into **`/clientName`**.



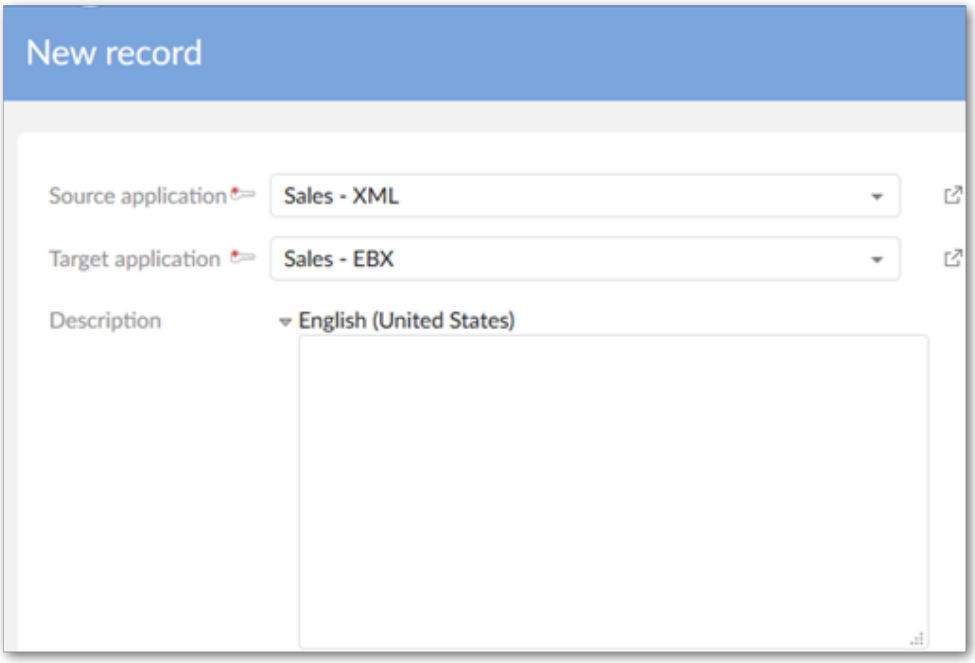
Now, an export process execution generates the following result (new path **`clientName`**):

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--XML content generated for /root/Client in data set Sales in
<dataexchange>
        <header>
                <appname>CRM-1407409946284</appname>
                <appversion></appversion>
                <date>2014-08-07</date>
                <apptype>[ON] XML</apptype>
        </header>
        <root>
                <Client>
                        <clientName>Paul</clientName>
                </Client>
                <Client>
                        <clientName>John</clientName>
                </Client>
                <Client>
                        <clientName>Peter</clientName>
                </Client>
                <Client>
                        <clientName>Jack</clientName>
                </Client>
        </root>
</dataexchange>
```

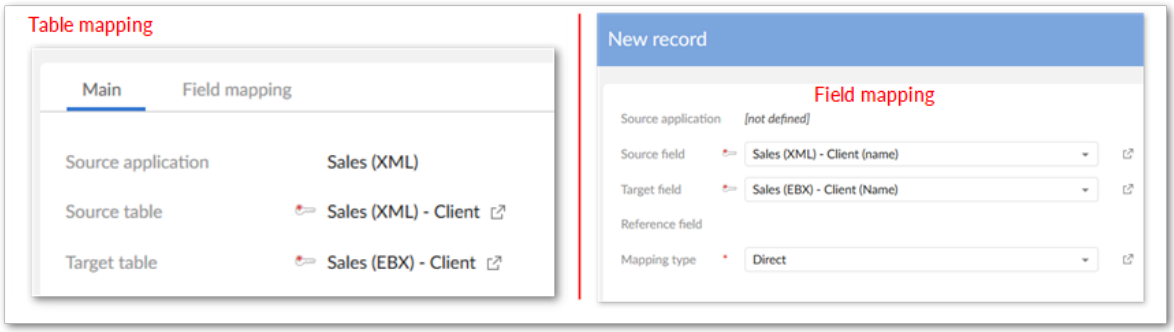
High-level overview of XML import with custom mapping

To import an XML file with a data structure that is not compliant with the Default XML configuration, you can manually configure a user-defined data mapping using the same process described in the previous XML export section. You can use the XML import file as a template to automatically generate the XML configuration.

Then you create an **Interface** from the source XML type application to the target EBX type application **Sales**.



The **Table mapping** and **Field mapping** must also be configured:



Special notation:	
✕	The under-terminal node is child of complex and multiple occurrences node. It will be ONLY imported/ exported if mapping of its parent node is declared in the configuration

CHAPTER 15

Generating constant values

This chapter contains the following topics:

15.1 Overview

The add-on allows you to populate a column's fields with a single pre-defined value during import, export, or transfer. You specify the value in the add-on mapping configuration. The example shown below demonstrates the feature with an import.

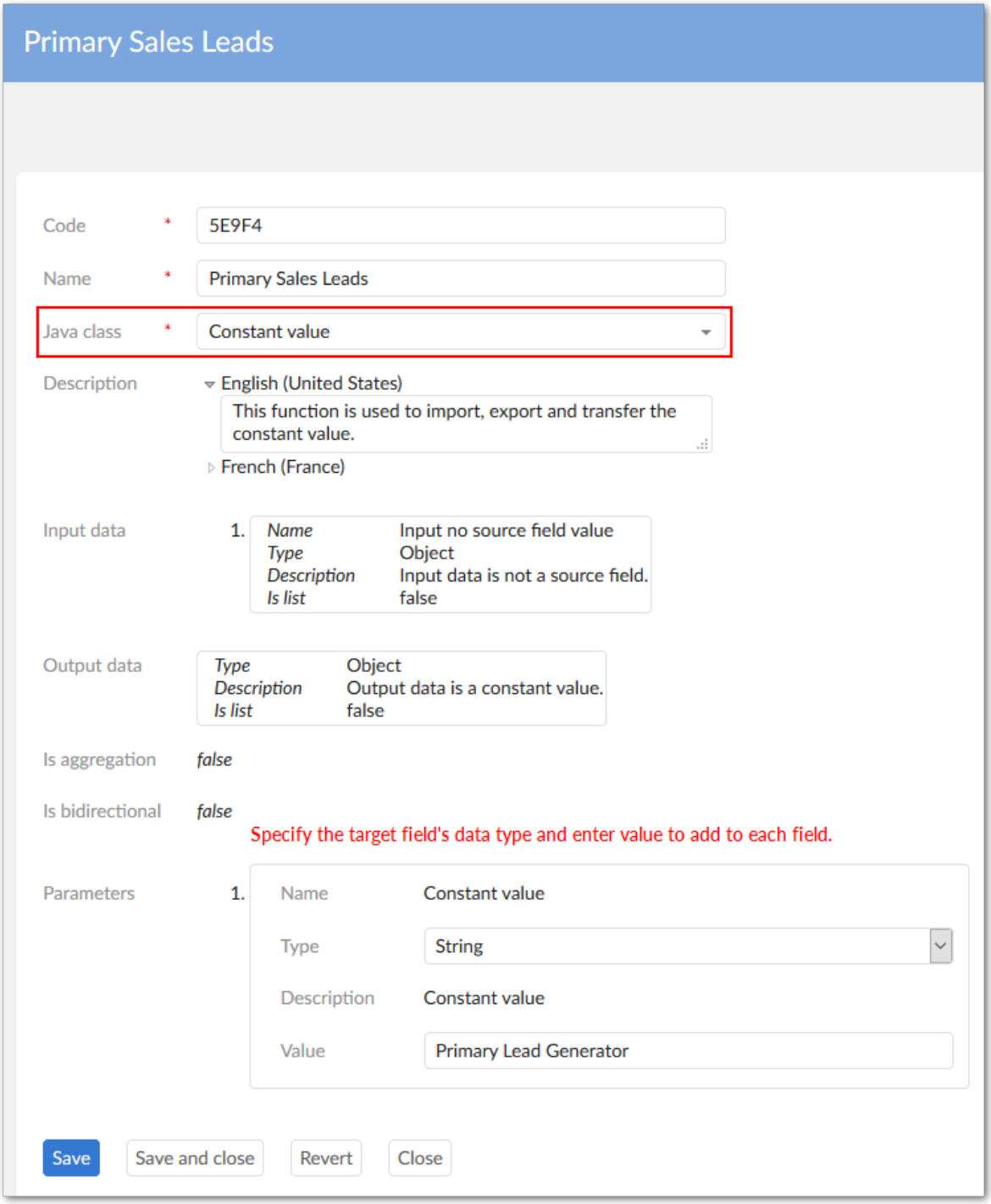
> **Note**
>
> The add-on automatically creates a new column containing the constant value that isn't mapped with a column in the source.

Generating constant values

This example demonstrates how to generate a constant value while importing data. However, you can use similar steps for export and transfer. For example, a company needs to track where sales leads come from. Lead generators provide Excel spreadsheets that do not indicate their source. The target EBX® table has a column specifying whether the lead came from an internal department, or external vendor. Internal leads should be labeled 'primary' and external leads 'secondary'. To add the primary or secondary label, identify the lead generator, create a transformation function, map source and target fields, and specify the constant value.

To create a mapping that uses a transformation function to generate a constant value:

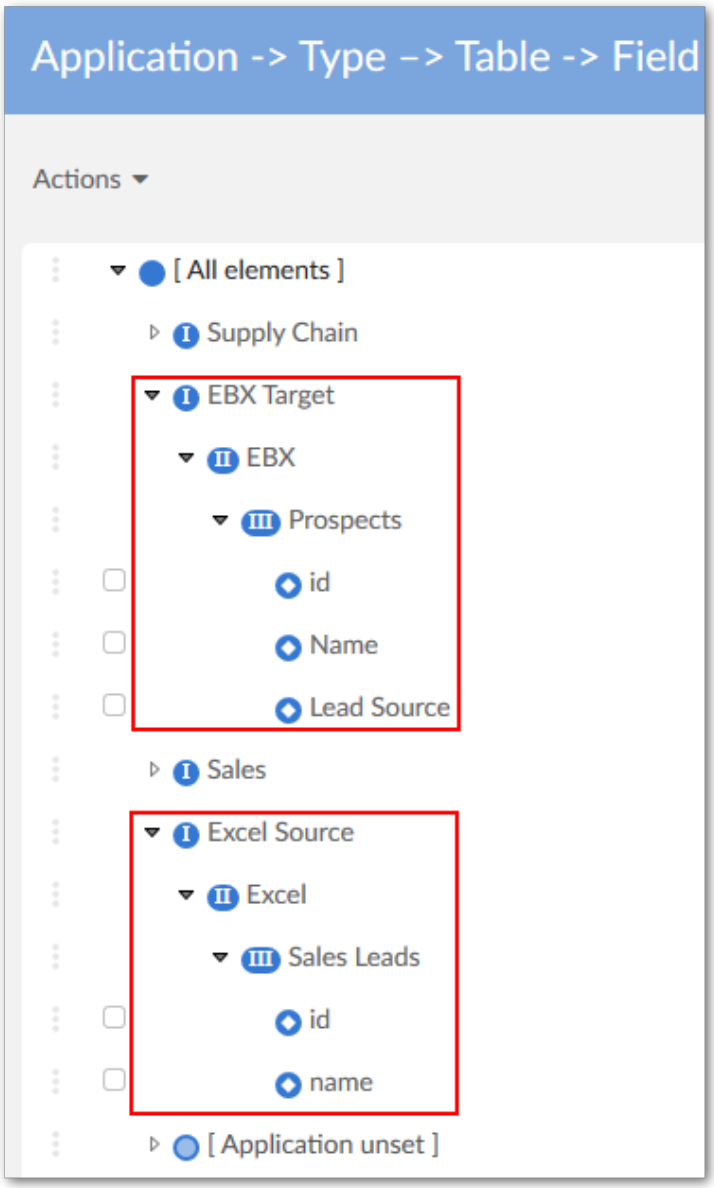
1. Navigate to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Reference data* > *Transformation function* and create a new transformation function that uses the *Constant Value* Java class. See the image below for the sample values used in this example.



2. Navigate to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Application* and create source and target applications in the **Application** table and use the **Application by type** view to associate them with the appropriate types. In this example we are importing from a source spreadsheet, so we created one Excel type application (source) and one EBX type application

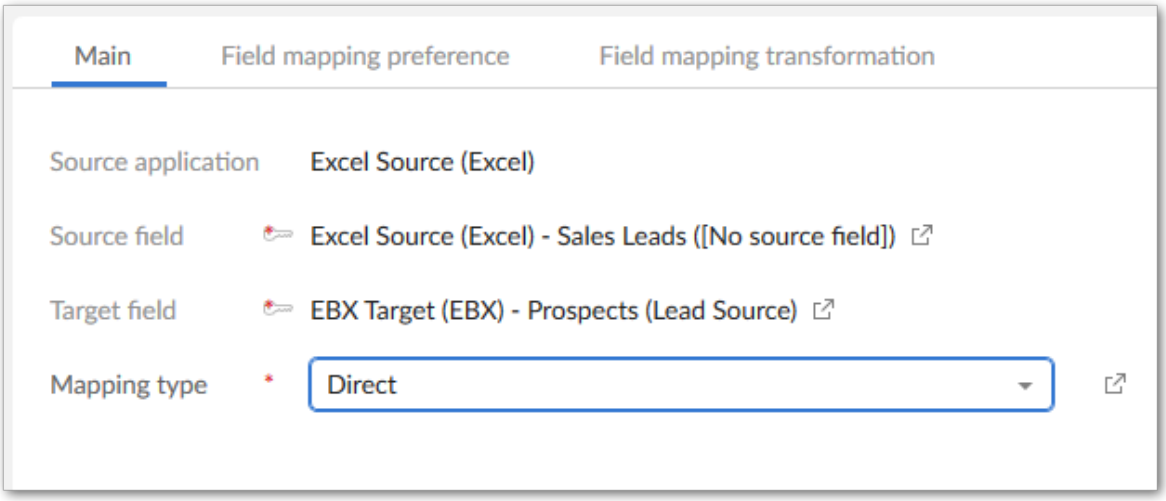
(target). For the EBX type, you must provide the path to the published data model that the target dataset is based on.

3. In the **Application by type** table, run the **Generate models** service for both application types and choose **Data model** for both. You'll need to point to the file and provide a little information when generating the models for the Excel file. You can view your results to compare with the expected outcome. The image below shows the generated tables and fields used in this example. Notice that the source does not include the field that designates where the leads come from.

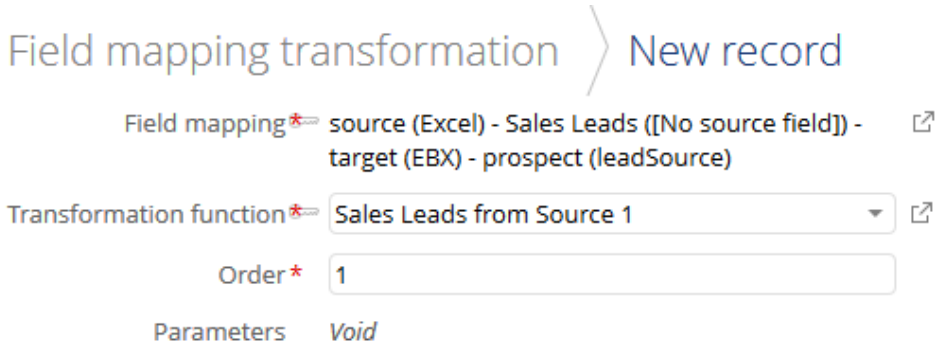


4. Use the **Interface** and **Application interface preference** tables to create an interface between the source and target applications and an application interface preference, respectively. If you have any questions about options, mouse over the label and open the tooltip for more information.
5. Use the corresponding tables in the **Data mapping** domain to create the table mapping, field mappings, and preferences for both. When creating the field mapping for the field using the transformation function:

1. The **Source field** must use the source's [No source field] option. The following image highlights the field used in this example.



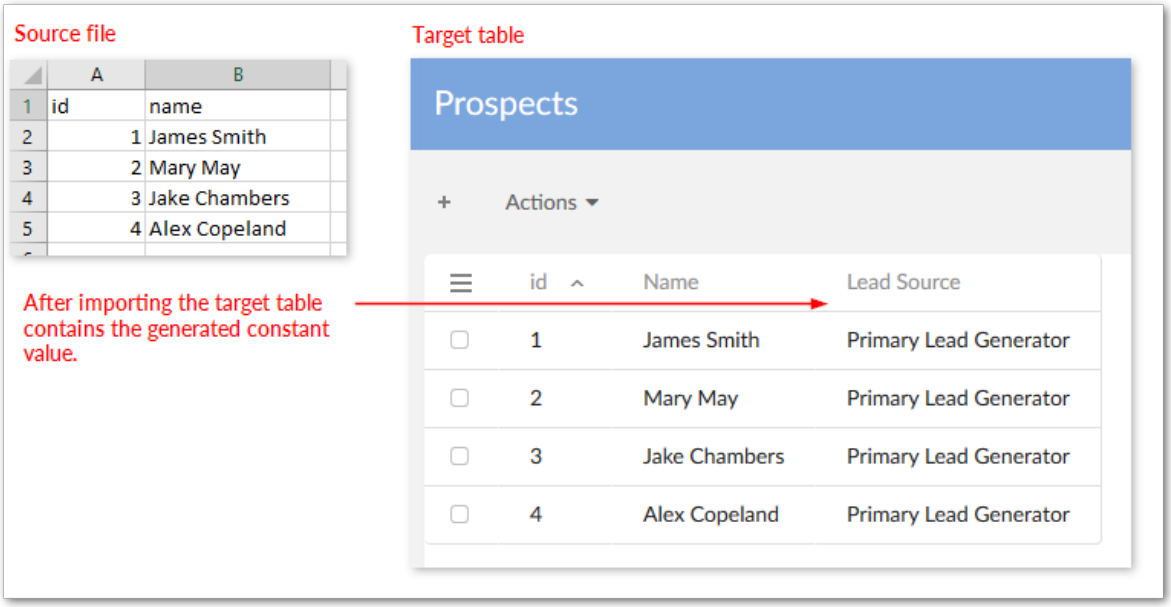
2. Indicate the appropriate target field.
3. Save, but do not close.
4. Select the **Field mapping transformation** tab and create a new transformation using the previously defined transformation function. Once you've selected the function from the list, save to refresh the page and enter any required values.



5. Save and close out to finish the configuration.

Attention

After completing the configuration, users need to use the **Preferences** option to load the correct preferences for this transformation mapping. The image below shows the results from our example.



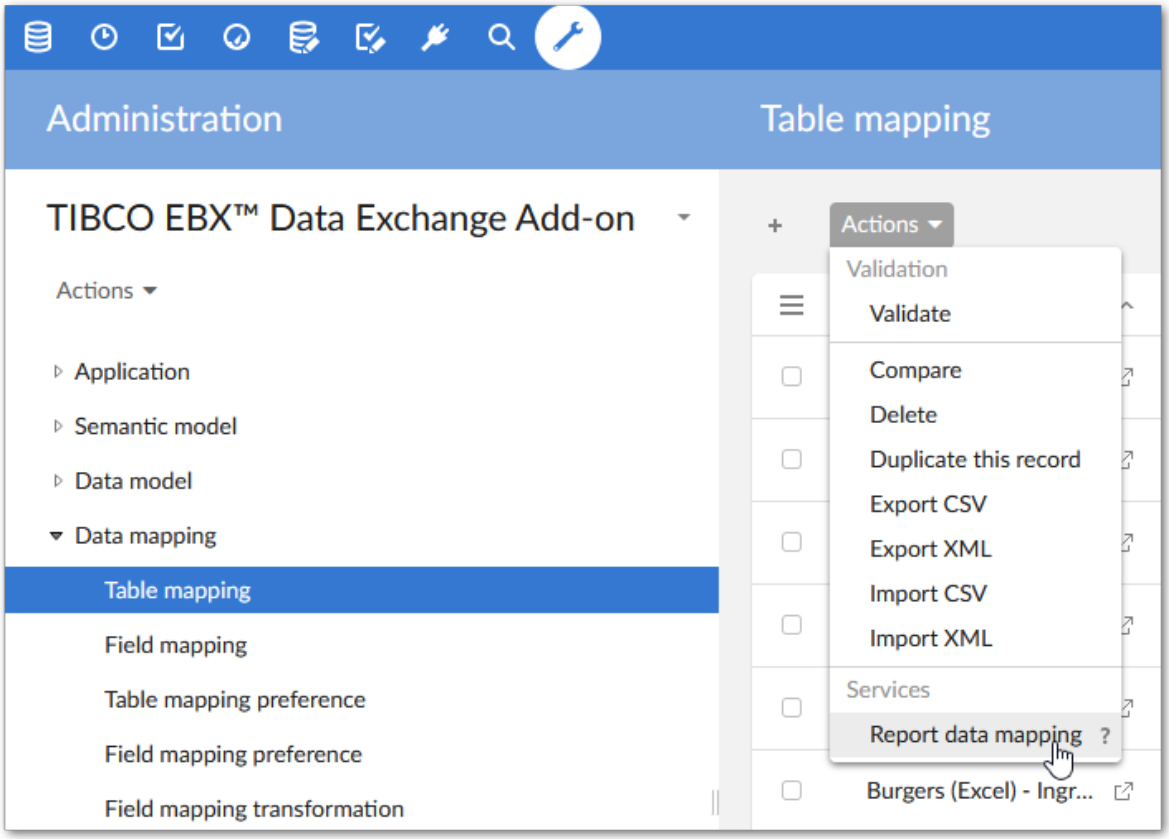
CHAPTER 16

Generating mapping reports

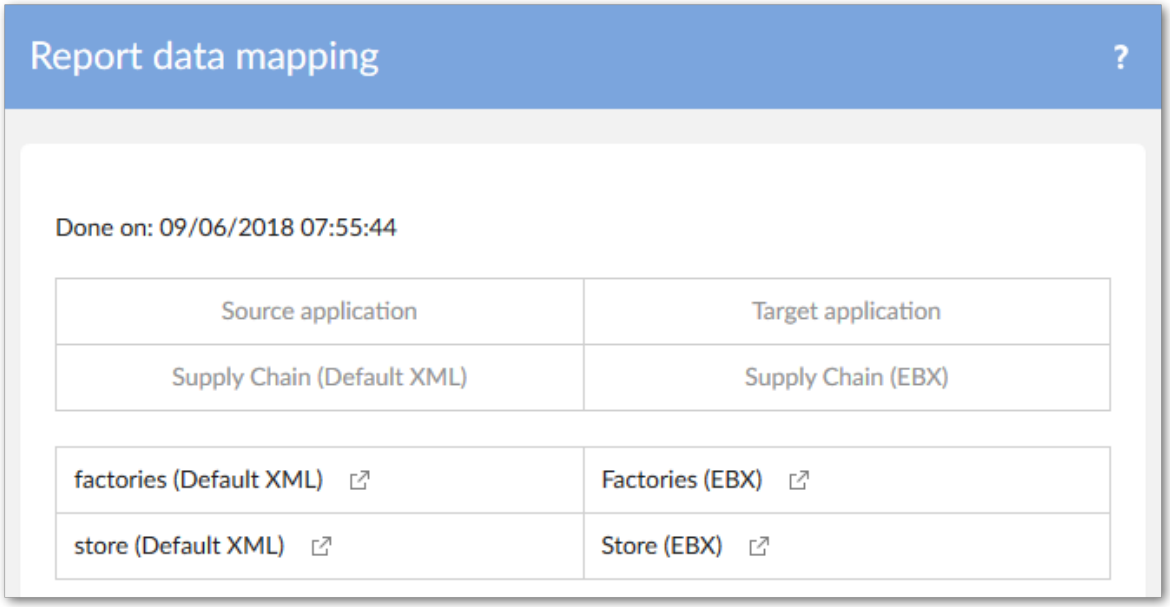
This chapter contains the following topics:

16.1 Mapping report service

You can obtain an easy-to-read report of all your data mapping configurations between source and target applications. The **Report data mapping** service on the **Table mapping** and **Field mapping** tables allows you to activate this report.



The following image shows an example of a mapping report:



CHAPTER 17

Transferring between tables based on different models

This chapter contains the following topics:

17.1 Overview

When the source and target locations are based on different data models, you must create a custom mapping configuration. This process is illustrated below and described in detail in the following section.

> **Note**
>
> All steps below must be completed by an administrator.

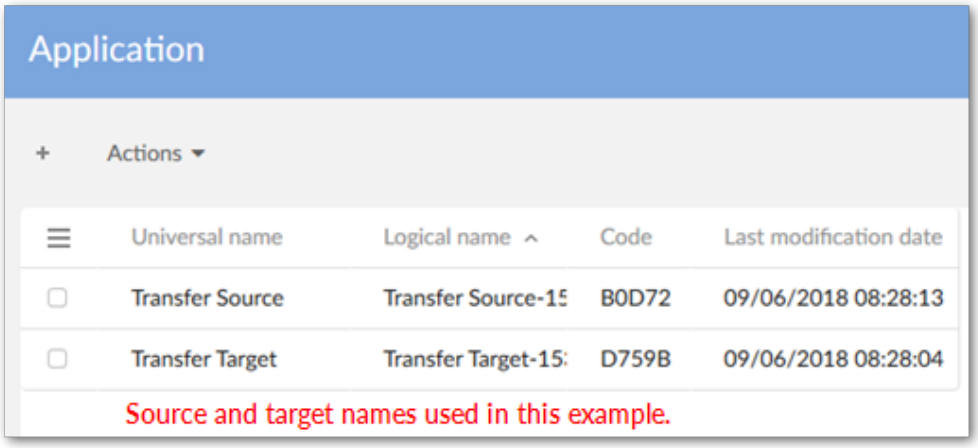
17.2 Configuring a custom mapping for data transfer

For simplicities sake, the steps below demonstrate the configuration process using a blank canvas and a very simple model. In other words, everything will be shown as if starting from scratch. No mapping components were previously created—either user-defined, or automatically by the add-on.

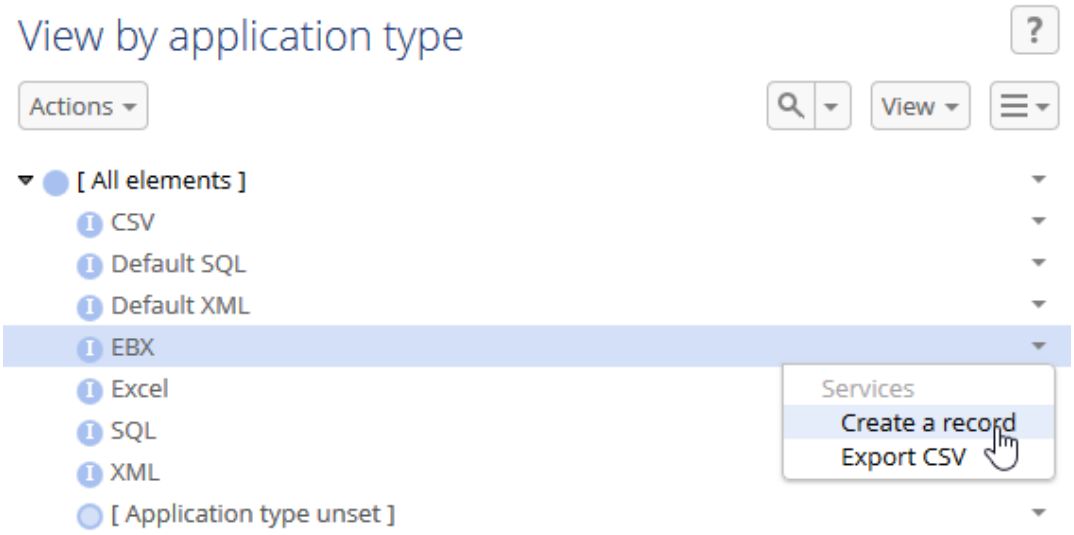
> **Attention**
>
> This example uses the add-on's **Auto data mapping** feature and thus includes the additional steps involving the semantic layer. Depending on your requirements, you can automatically generate the source and target models, and manually map the tables and fields. This option will be presented in the appropriate step below.

To create a custom mapping configuration for transfer:

1. Create source and target applications by navigating to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Application* > *Application* and creating two new records.

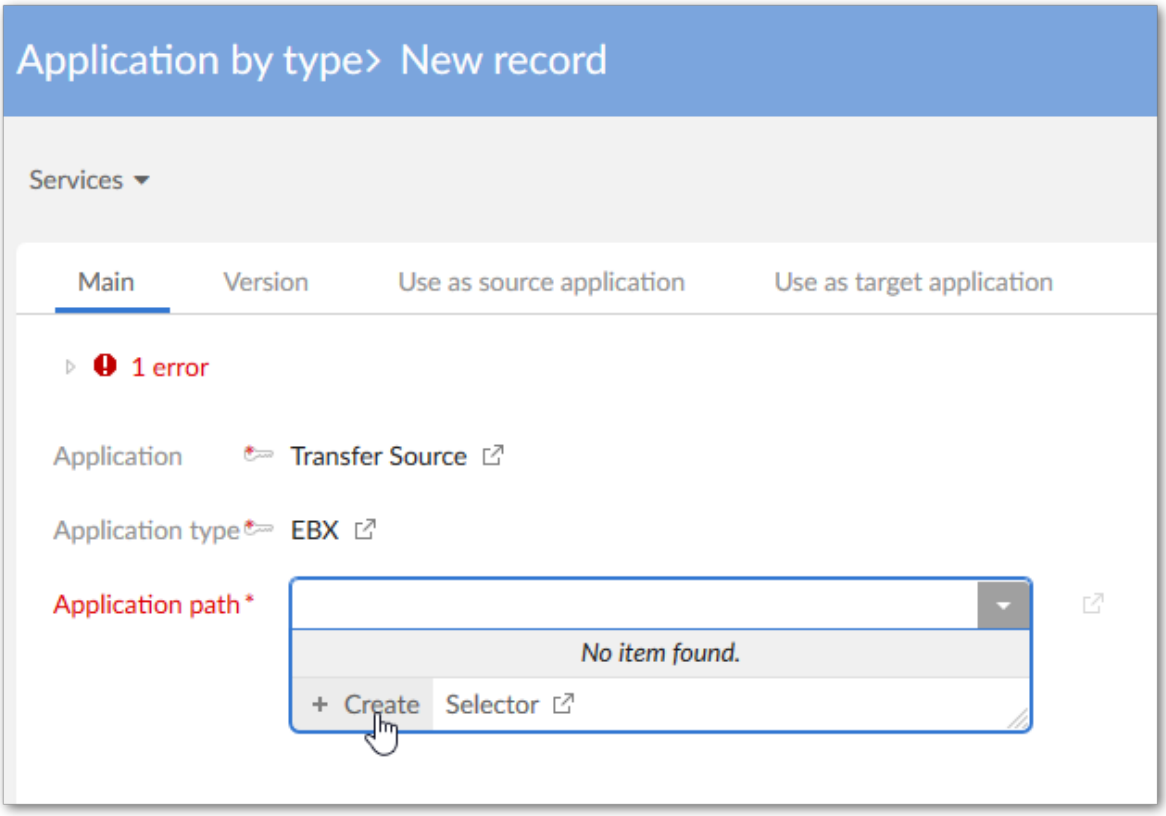


2. Use the **Application by type** table to assign these applications to the *EBX* application type. For each application:
    1. Create a new record under the **EBX** node.



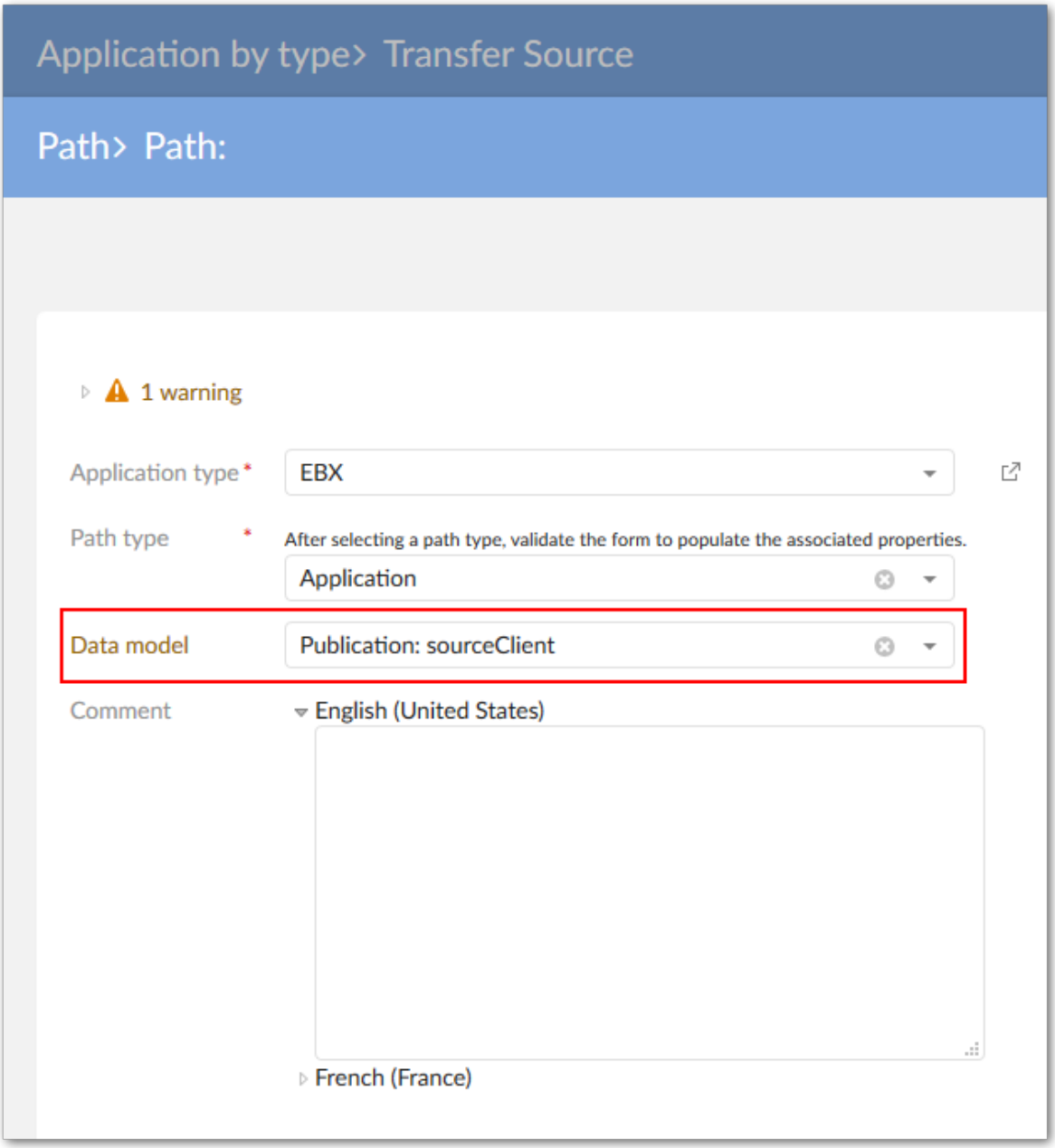
    2. Select the application from the **Application type** drop-down menu and save to display additional fields.

3. From the **Application path** drop-down menu, select **Create**.

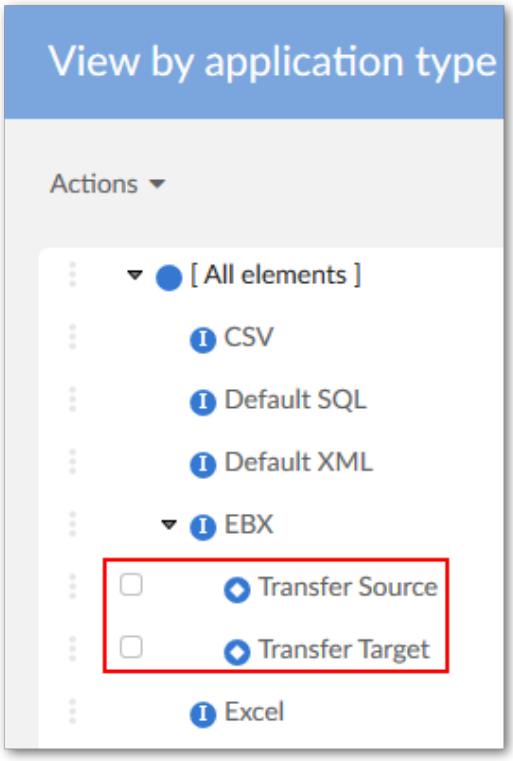


4. Leave all options as is, and click **Save**, but do not close.

5. Use the **Data model** drop-down list to select the data model you want to use as the *Source* in the transfer operation.

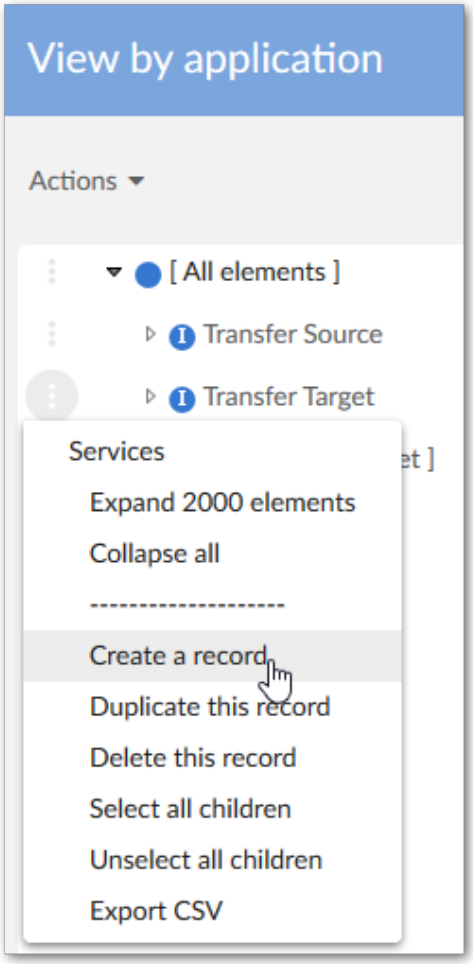


6. Save and close until you return to the **Application by type** table. You can repeat the above steps to add the target application. Our example configuration is shown below:



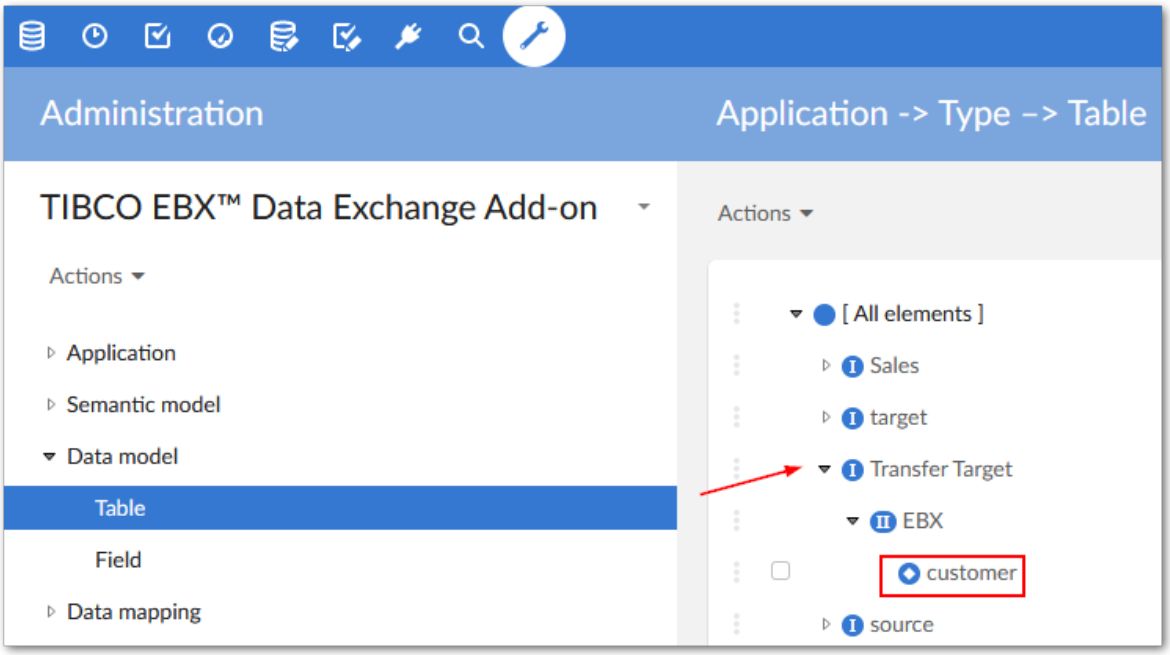
3. In the **Application by type** table, run the **Generate models** service on the *source* and *target* applications.
   - On the **Generation** screen select the **Semantic model and data model** option. This option follows along with the remaining steps presented here. If you want to skip auto-data mapping, choose **Data model**. You will then need to create an interface that defines the applications as a source and target, and use the **Data mapping** domain to manually map tables and fields.
   - Click **Generate**.
4. This step is very important for successful transfer. You must link the *Target's* generated table and fields with the *Source's* generated Object Class and Properties. To accomplish this:
   1. Navigate to the **Object Class by Application** table.

2. From the *target* application's **Services** menu select **Create a record**.

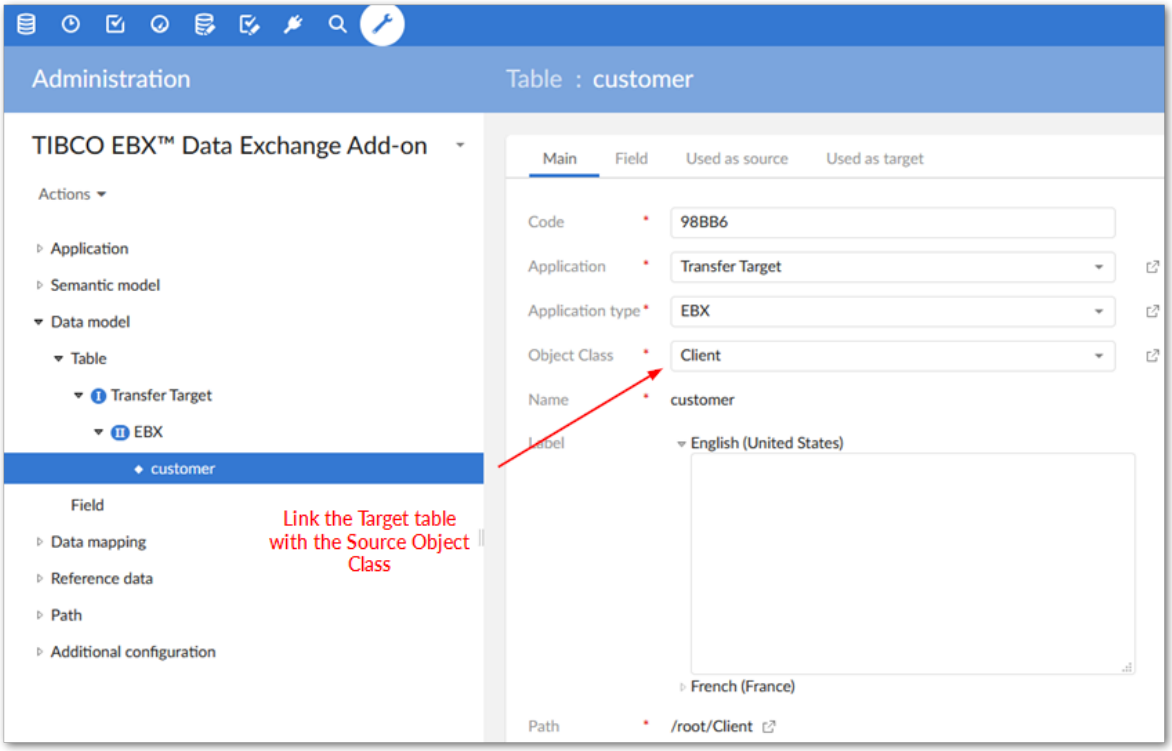


3. In the **Object class** field select the *source* table (Client in this case). Save and close.

4. Navigate to, and open the *target* table in the **Data Model** domain (see image below for sample location):



5. In the **Object Class** field select the *Source* Object Class that you created in an earlier step.



6. Save, but do not close the record. You now do this same process for each field you want to map by selecting the **Field** tab, opening each field, changing its **Property** drop-down value to the corresponding field from the *source*.

7. Navigate to the **Application by type** table and from the *source* application's **Services** menu, select **Auto data mapping**.
8. Select the **Target** application and click **Mapping**. From the results page that displays you can view the auto-generated mappings.

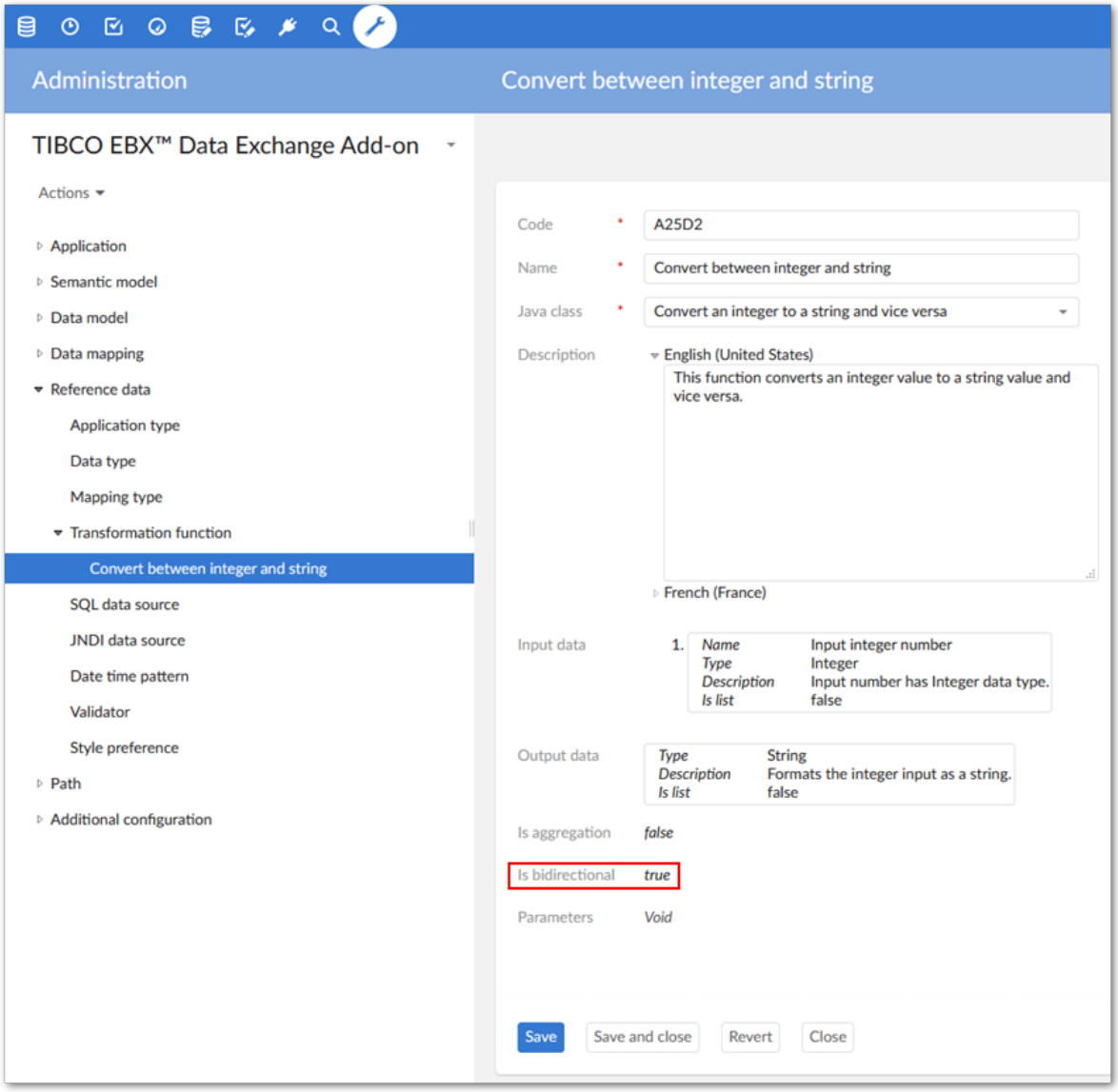
CHAPTER **18**

Bi-directional mapping

When the data transfer relies on a user-defined data mapping, the add-on can ensure bidirectional execution. This means that when a mapping from table A to table B exists, you can also transfer from B to A. This execution depends on the transformation functions used for transferring the data. Indeed, the **Is bidirectional** property must be activated.

The bidirectional mode is activated automatically when the data types of the source and target fields are the same and the **Direct** mapping type is used.

Fields are ignored during the reverse transfer when they are not associated with the **Is bidirectional** option.



CHAPTER 19

Splitting and aggregating fields

This chapter contains the following topics:

19.1 Split and aggregation functions

During CSV, Excel, and XML export/import and data transfer, you can use transformation functions to split and aggregate data. The following sections provide basic examples:

- Splitting field values during import
- Aggregating fields during export

Two split approaches exist:

- A direct split for which every single part of the source string field is mapped to a target field.
- A specific split for which you can configure how the different parts of the source field are mapped to the target fields.

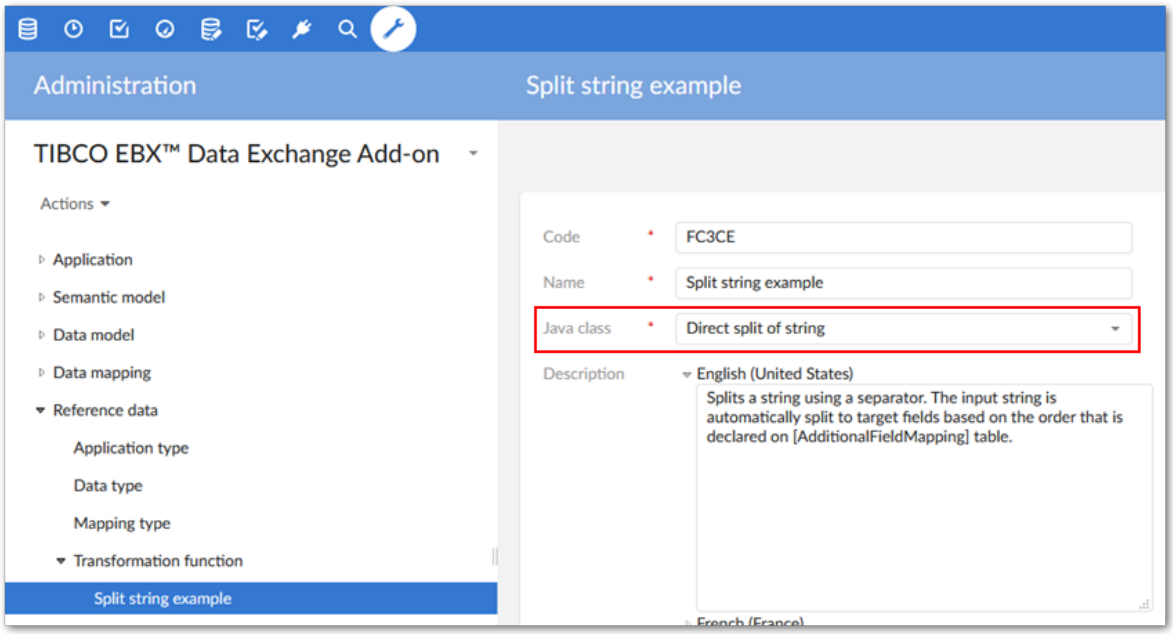
The next sections highlight some examples of split and aggregation using Excel. Similar principals can be applied to other formats.

19.2 Splitting field values during import

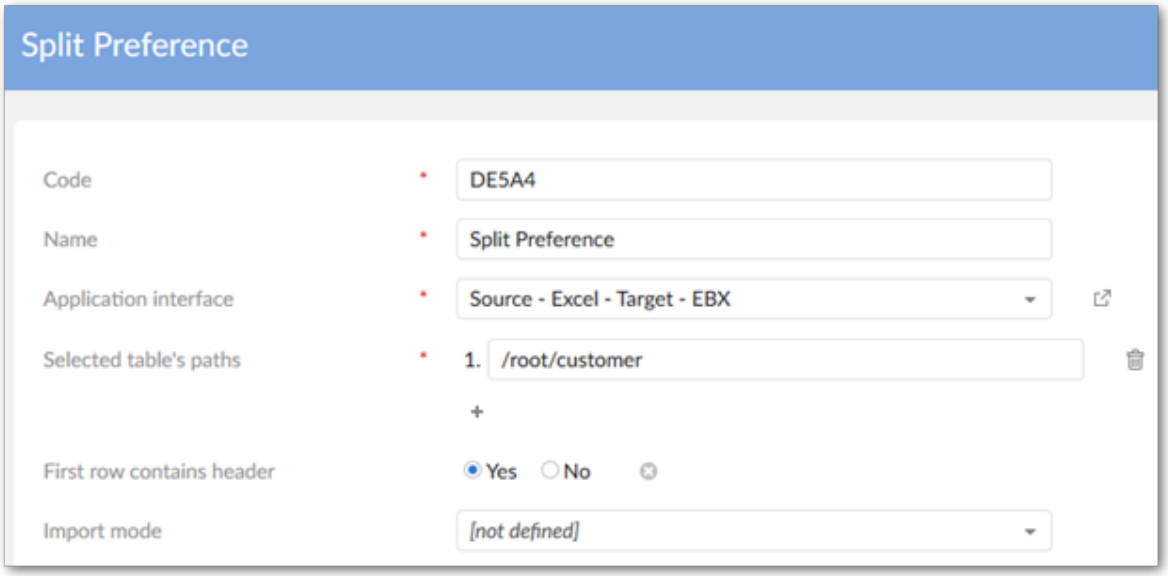
The following examples demonstrate splitting field values on import. This example uses the built-in **`Direct split of string`** Java class. This class allows you to define one separator as a parameter. You can use the **`Split of string`** Java class when more parameters are required.

To automatically split a string value during Excel import:

1. Create a transformation function that implements the built-in **`Direct split of string`** Java class. For this example, we will leave the separator parameter at its default value of **`ebx:anywhitespace`**. If required you can alter this parameter to your business needs.

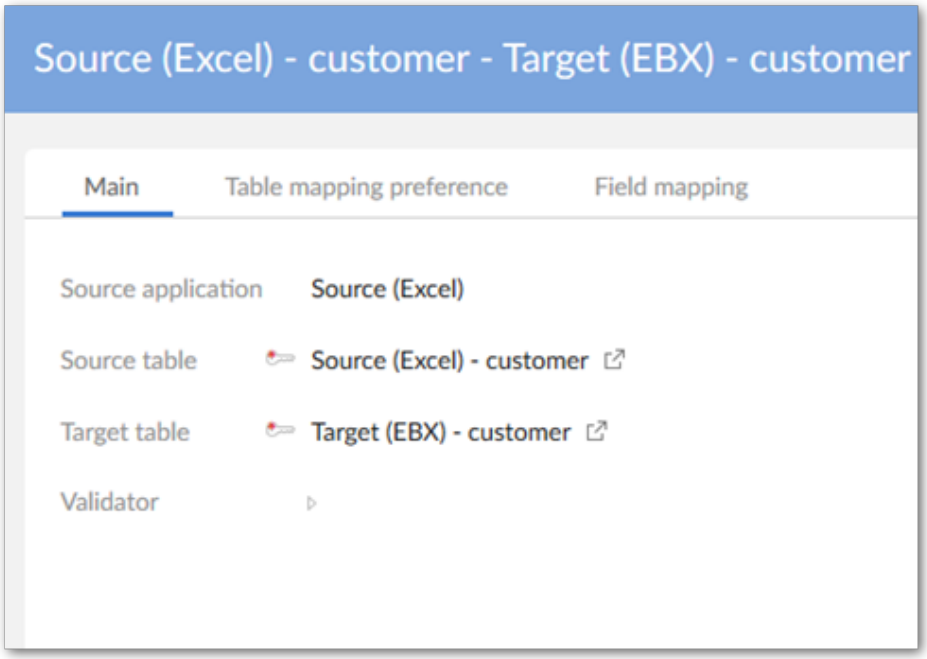


2. Under *Application* > *Application by type* create source and target applications. In this example, the target application is an EBX® table and the source application is an external Excel file.
3. In the **Application by type** table, run the **Generate models** service for both applications.
4. Create an interface and application interface preference. The interface defines the Excel file as the source and the EBX® table as the target. The preference stores settings for access during the import operation.



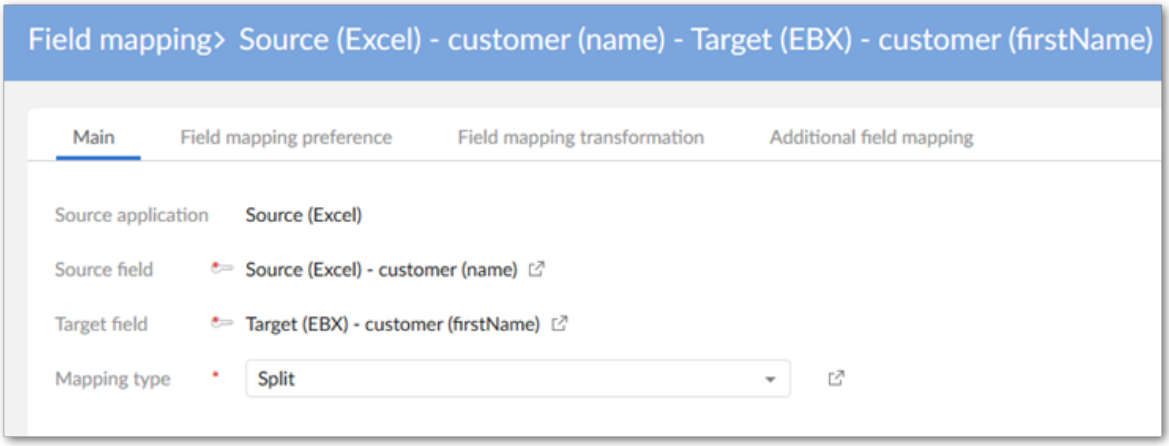
5. Navigate to *Data mapping* > *Table mapping* and create the following table mapping and preference:

- Table mapping example:



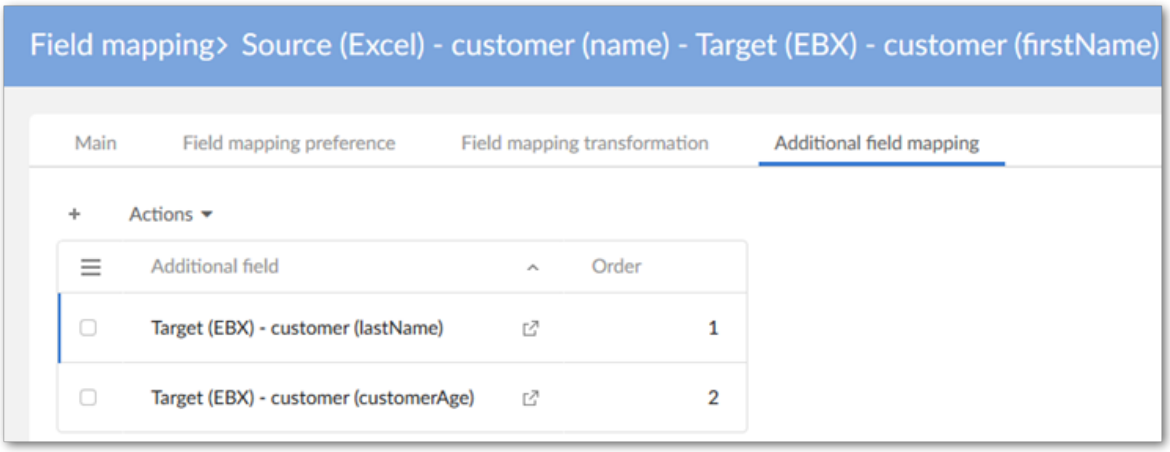
- Under the **Table mapping preference** tab, associate this table mapping with the preference we created earlier. This is done using the **Table mapping** and **Application interface preference** properties.

6. Navigate to *Data mapping* > *Field mapping* and create the following field mapping and preference:
   - Map the name field from the source with the firstName field in the target. Set the **Mapping type** property to **Split**.

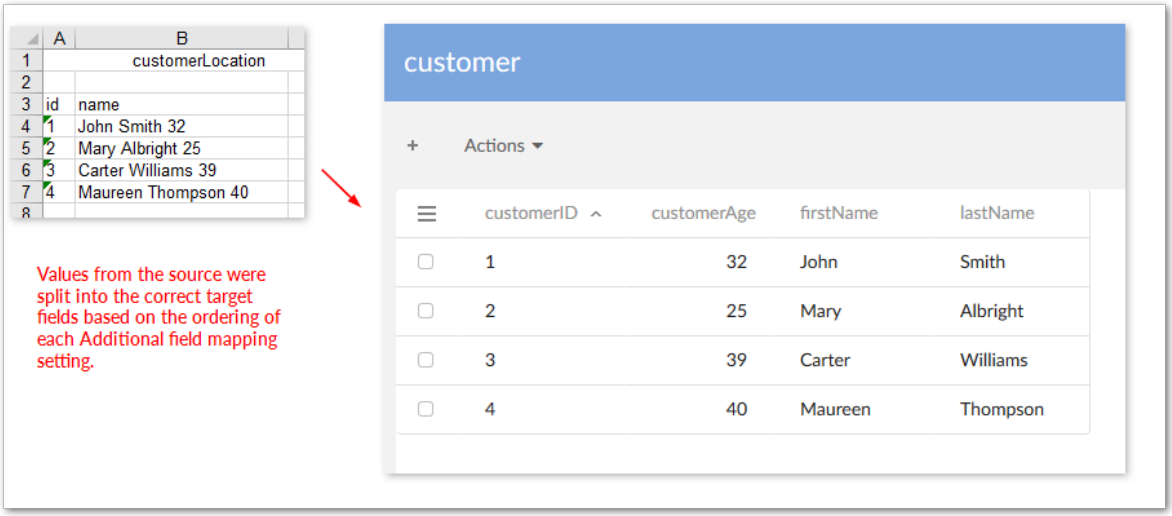


   - Use the **Field mapping preference** tab to create a new preference and associate with the table preference defined earlier.
   - Create a new record in the **Field mapping transformation** tab and add the transformation function we previously created.
   - Use the **Additional field mapping** tab to associate other fields with this mapped field. In this case the other fields are where the split values will be populated. In our target example, the

name field includes [0] first name, [1] last name, and [2] age. Since our main field mapping includes the firstName field in the target, we need to map the other two parts of the string to the correct fields. Additionally, we need to specify the order in which the substrings occur in the source. This is done using the **Order** property.



The following image shows the result after importing:

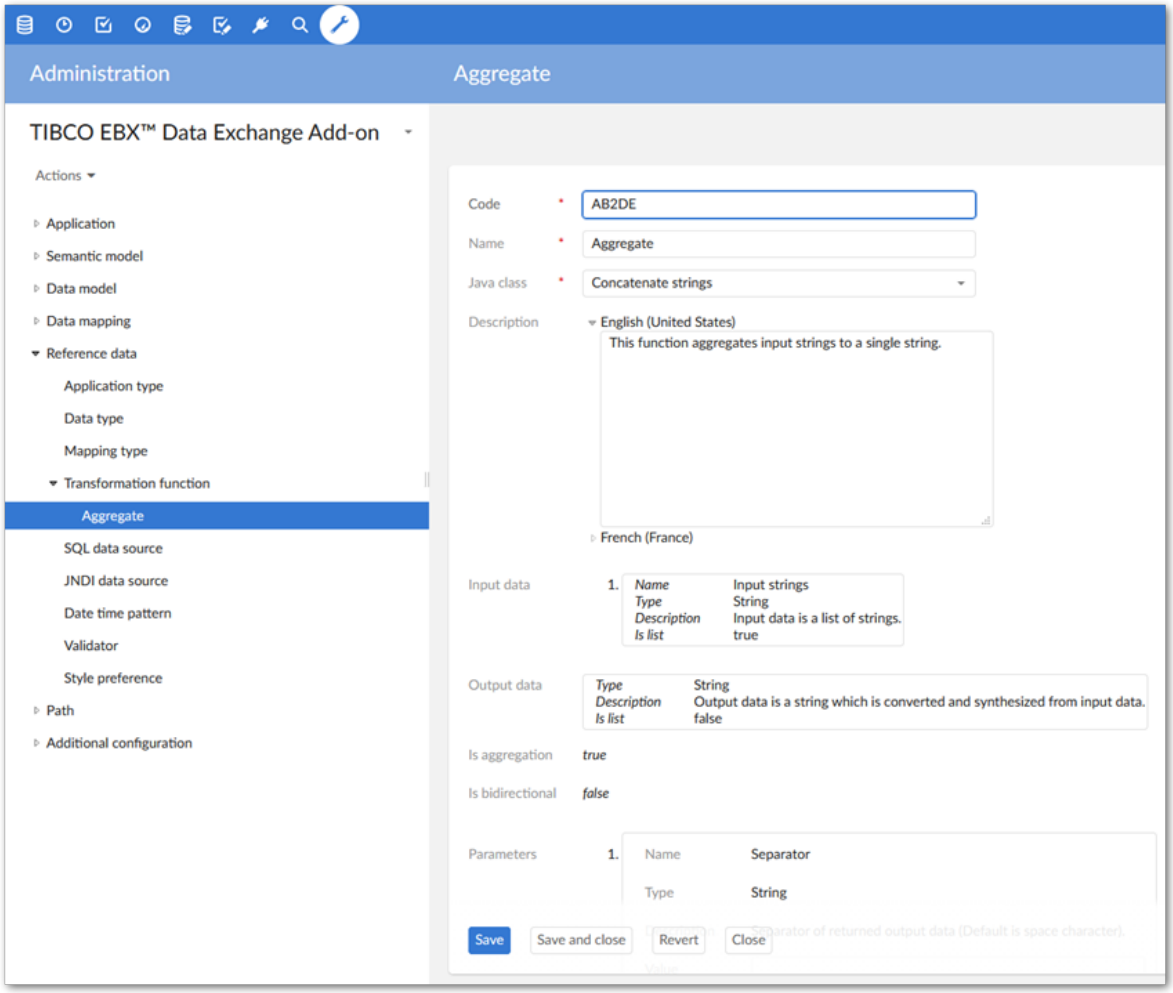


19.3 **Aggregating fields during export**

The following example shows how to configure the add-on to aggregate fields during export:

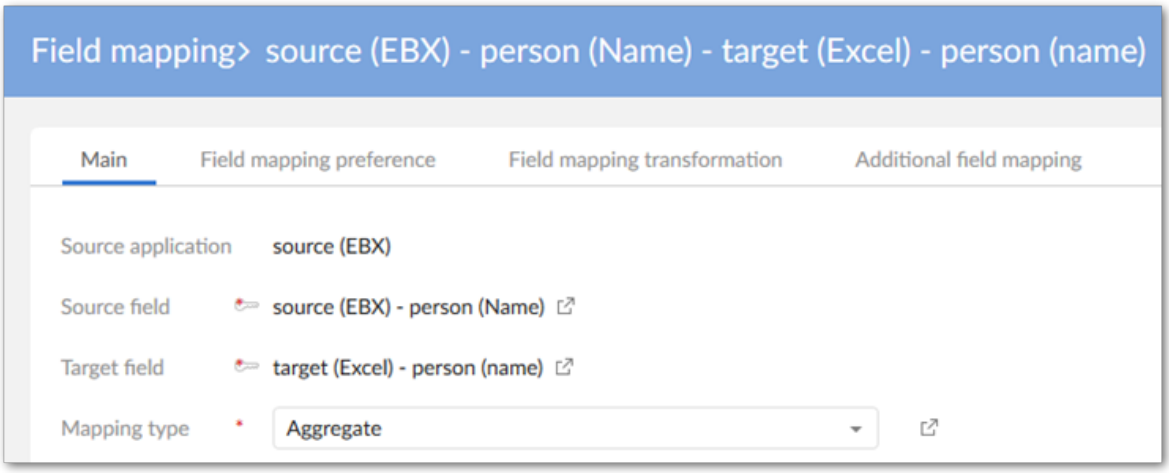
1. This example assumes the following conditions before configuring aggregation specific settings:
   - Source and target applications exist and are mapped to their corresponding types.
   - Data models have been generated from the applications. You can create models from scratch, but using the add-ons automatic generation features makes things easier.
   - An interface between the applications and application interface preference has been created.

2. Create the transformation function used to aggregate the fields. In this case we use the built in **Concatenate strings** Java class with the default separator parameter.

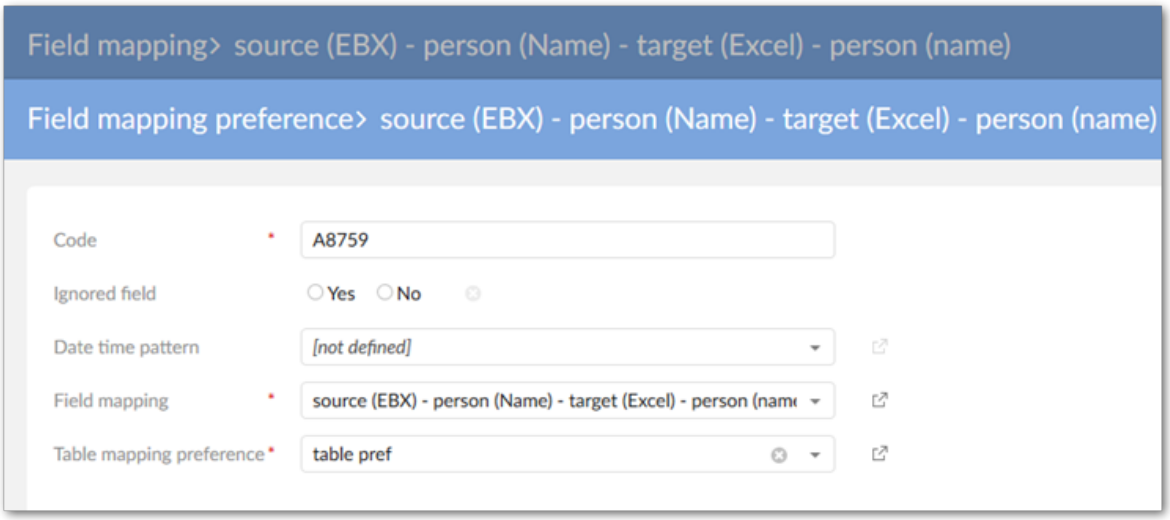


3. Create a table mapping that maps the source and target.
4. Create a table mapping preference that includes the above mapping.
5. Create the following field mappings and preferences:
   - Map the main source and target fields. In the exported file, this mapping provides the first part of the string added to the field.

- Set the **Mapping type** property to **Aggregate**.

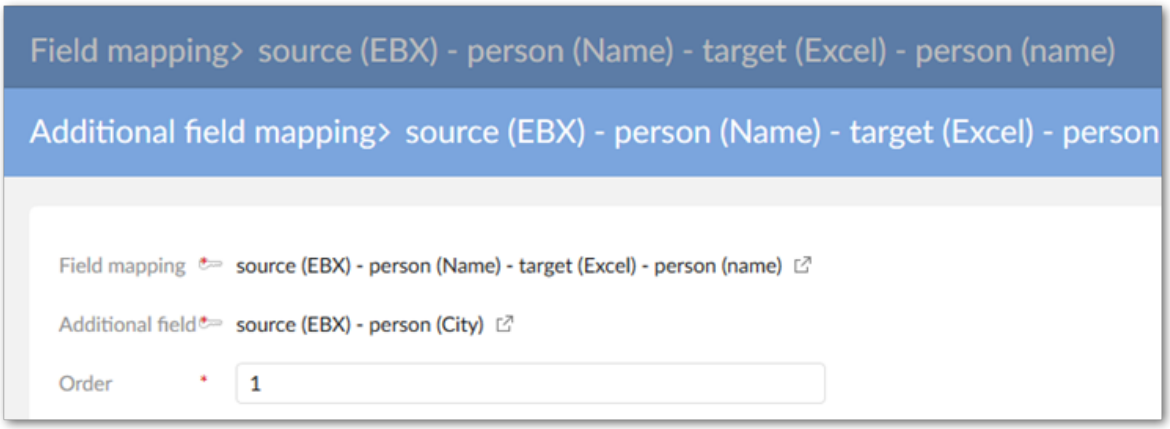


- Create a field mapping preference that is associated with the previously created table mapping preference.



- In the **Field mapping transformation** tab, add the transformation function created in the earlier step.

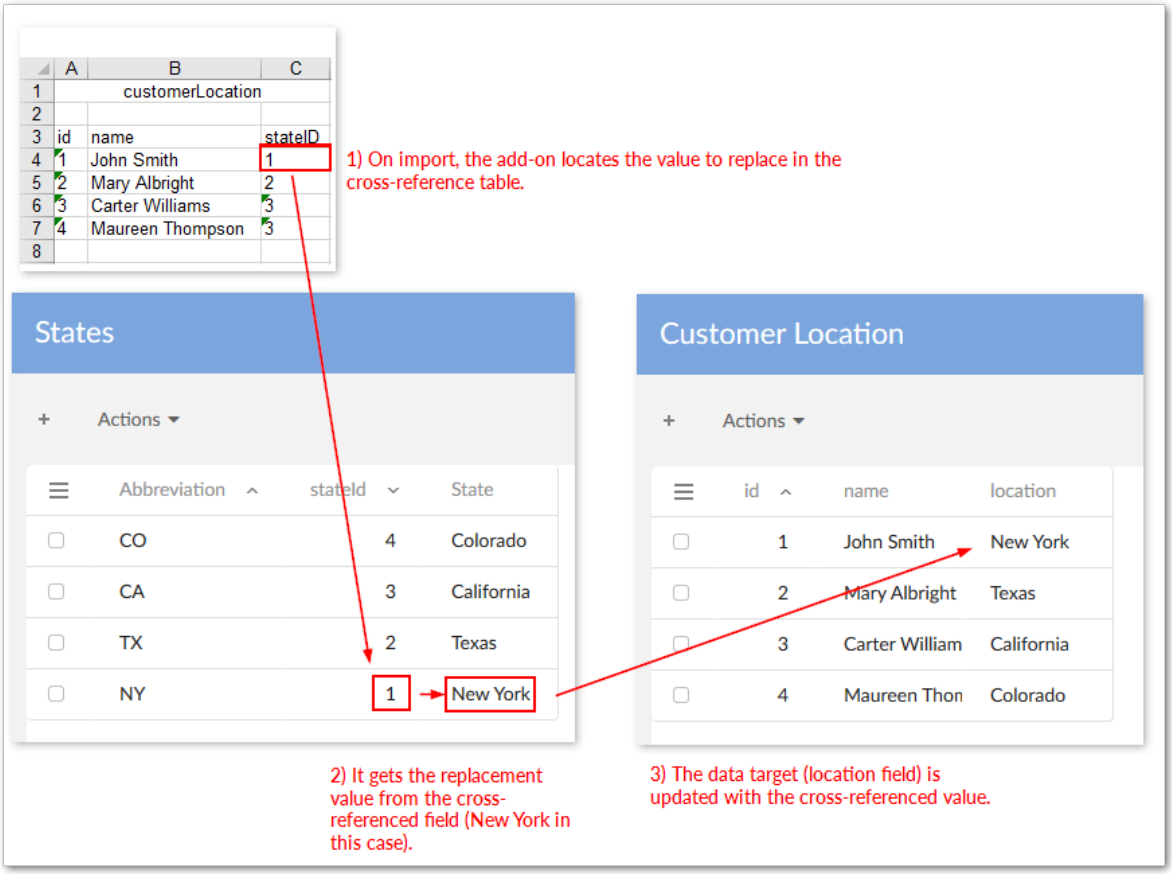
- In the **Additional field mapping** tab, create the mapping for the second field. In the exported file, this source field value will be appended to primary field value.



CHAPTER 20

Converting values using a cross reference table

You can use *cross-reference* transformation functionality to change data values when moving data. For example, a data source field indicating a customer's location might use a numerical identifier. Whereas, the data's target requires the full state name. Using the pairing of id and name in another table, the function can cross-reference the state ids with their full names, and send the replacement value to the target. This is illustrated in the image below:



Cross-reference functionality works with all data types. However, it requires the following when source and target data types differ:

- The source field and the field used to lookup the desired value (located in the cross-referenced table) must be the same data type.
- The field containing the replacement value (located in the cross-referenced table) and the target field must be the same data type.

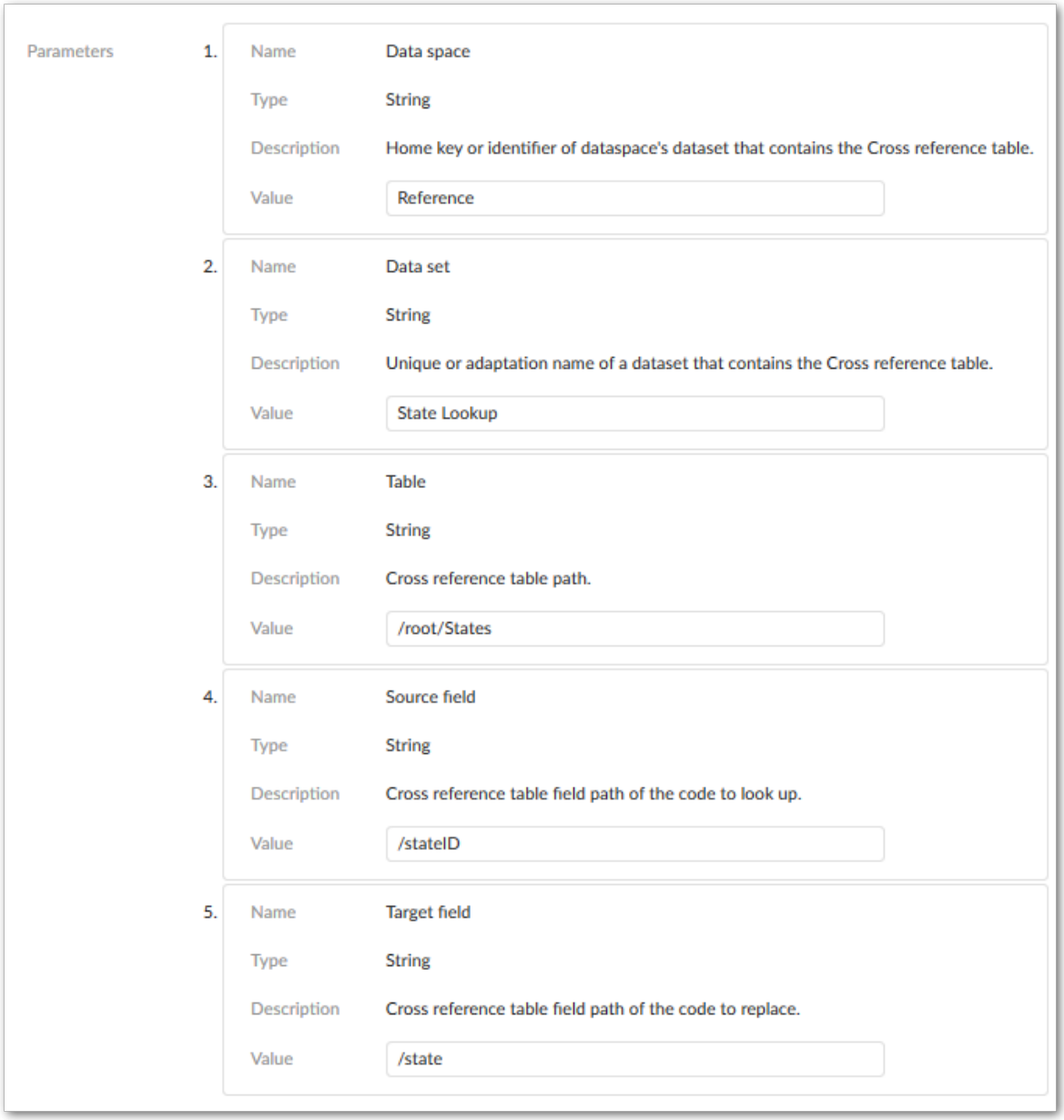
Attention

The following example assumes the correct data mappings between data source, cross-referenced fields, and data target exist. Also, users must be able to select this functionality using a preference. So, application interface, table, and field preferences need to be created before using the transformation during import or export.

To change data values using a cross-reference transformation:

1. Start by navigating to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Reference data* > *Transformation function*.
2. Create a new a new record and input the following:
   - The name for this configuration.
   - Select **Cross reference** from the **Java class** drop-down menu and save, but do not close the record.

- Enter the parameter values to point to the correct fields to cross reference. The following image shows the paths used for the example described at the outset of this section:



3. Save and close the record.
4. Locate the field mapping to which you want to apply this transformation and under the **Field mapping transformation** tab, create a new record.
5. Use the **Transformation function** drop-down menu to select the function you just created. If needed, you can also edit the parameters from this screen. Remember that the transformation function will only be applied if (during import, export. transfer) users select the preference that includes the correct field mapping.

CHAPTER 21

Connecting to an SQL data source

This chapter contains the following topics:

21.1 Overview

Administrators can create a connection to an SQL data source by:

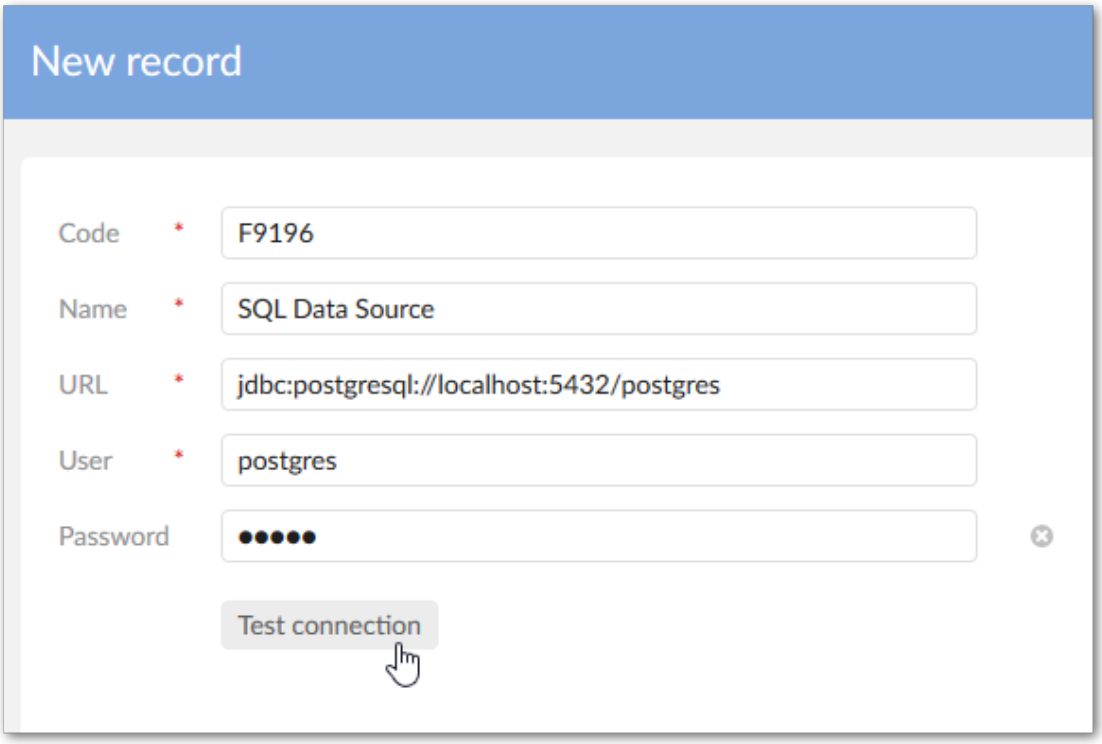
- Configuring all settings via the add-on.
- Defining the connection information in the application server. Then, using the connection information, associate the database or view to a data model.

21.2 Configuring data source via the add-on

To define a JNDI database connection and associate it with a data model:

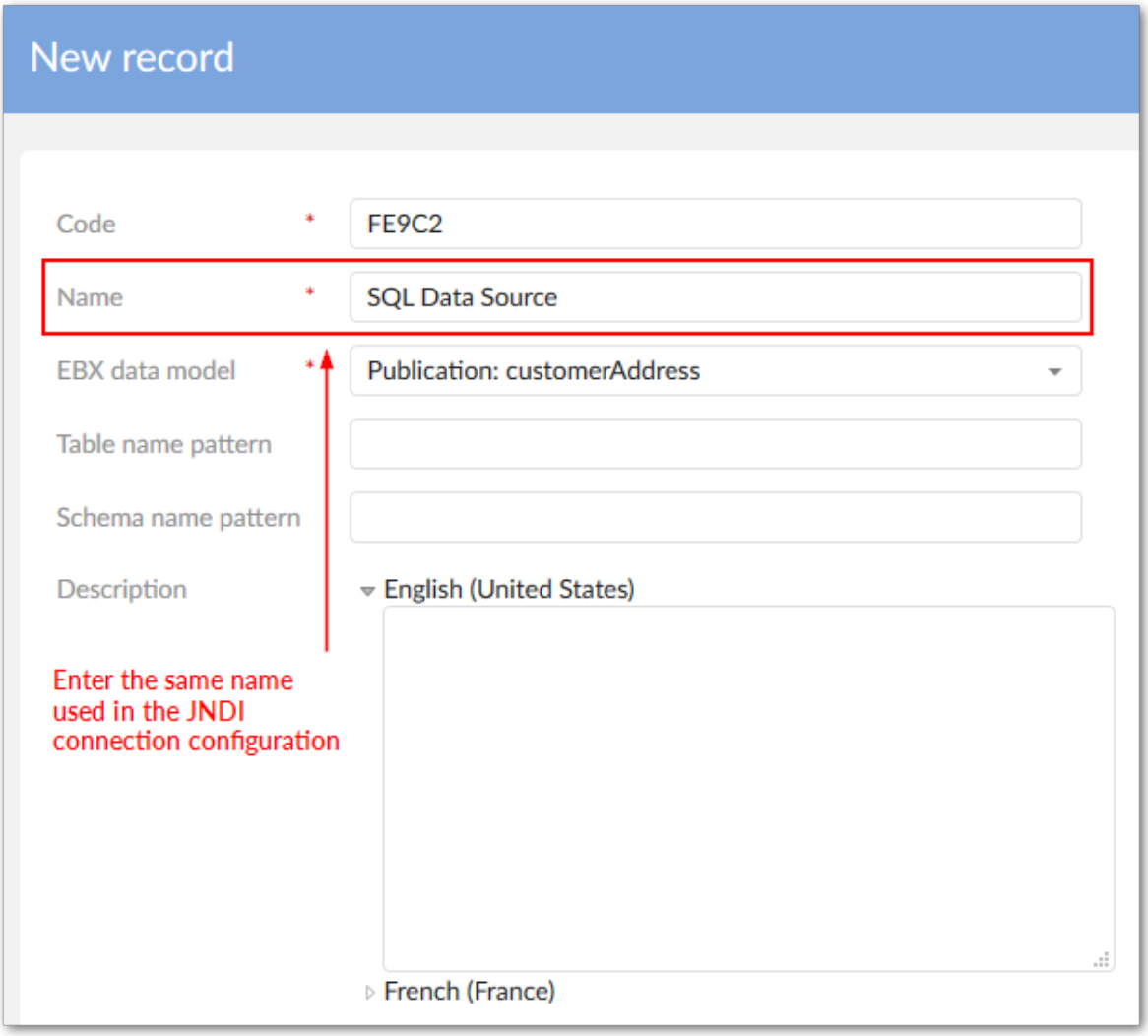
1. Navigate to *Administration > Integration > TIBCO EBX® Data Exchange Add-on > Reference data > JNDI data source* and create a new record.
2. Define the following connection parameters:
   - **Name**: The connection name you specify will be used in a later configuration step to associate this connection information with a data model.
   - **URL**: Add the appropriate URL for this database connection.
   - **User/Password**: The permissions obtained from the login credentials will be granted to each EBX® user accessing this data source.

3. Click **Test connection** to check configuration settings and save and close if successful.



4. Navigate to the **SQL data source** table and create a new record.
5. Associate the previously created connection to a data model using the following properties:
   - **Name**: You must enter the same name used for the JNDI connection.
   - **EBX® data model**: Select the data model publication to link to this connection.

- If desired, use the optional properties to filter schemas and tables and add a description.



6. After saving and closing, users can access the SQL import and export services.

21.3 Configuring a data source in the application server

You can setup an SQL data source connection in your application server and use the add-on to associate the data source with an EBX® data model publication. The connection configuration requirements differ from environment to environment. Please, consult your application server's documentation. The high-level steps are outlined below:

- Configure the JNDI data source in your application server.
- Expose any required resources to the **ebx**, **ebx-manager**, and **ebx-addon-adix** web applications.
- Use the add-on to create a configuration that links the data source to an EBX® data model publication.

The example below demonstrates how configuration might be completed using Tomcat 8.5:

1. In the `server.xml` file, use a **Resource** to declare the JNDI data source. Note that the JDBC driver for this resource must deployed in Tomcat's **lib** folder. Additionally, you may need to refer to Tomcat's documentation for the correct **Resource** parameter values.

```
<GlobalNamingResources>
     <Resource name="jdbc/postgres" auth="Container"
          type="javax.sql.DataSource" factory="org.apache.commons.dbcp.BasicDataSourceFactory"
 driverClassName="org.postgresql.Driver"
          url="jdbc:postgresql://123.1.2.3:5432/postgres"
          username="postgres" password="postgres" maxActive="20" maxIdle="10" maxWait="-1"/>
    </GlobalNamingResources>
```

2. In the `server.xml` file, add a **ResourceLink** to the **ebx**, **ebx-manager** and **ebx-addon-adix** contexts that links to the JNDI resource created in the first step.

```
<Host name="localhost" appBase="webapps" workDir="work"
    unpackWARs="false" autoDeploy="false">
<Context path="/ebx" docBase="ebx.war">
 <ResourceLink name="jdbc/postgres" type="javax.sql.DataSource" global="jdbc/postgres"/>
</Context>
<Context path="/ebx-manager" docBase="ebx-manager.war">
 <ResourceLink name="jdbc/postgres" type="javax.sql.DataSource" global="jdbc/postgres"/>
</Context>
<Context path="/ebx-addon-adix" docBase="ebx-addon-adix.war">
 <ResourceLink name="jdbc/postgres" type="javax.sql.DataSource" global="jdbc/postgres"/>
</Context>
</Host>
```

3. In each webapp's `web.xml` file, declare the JNDI resource.

```
<resource-ref>
  <res-ref-name>jdbc/postgres</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

4. Navigate to the *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* > *Reference data* > *SQL data source* table and create a new record.
5. Associate the data source configuration with a published data model using the following properties:
   - **Name**: You must enter the same name used by the **Resource**, in this case `jdbc/postgres`.
   - **EBX® data model**: Select the data model publication to link with this connection.
6. When you save the record the add-on validates the connection information. Upon successful save, users can access the SQL import and export services.

CHAPTER 22

Migration overview

The add-on allows you to export and import an XML file containing configuration and preference setting data. You can use this feature to backup add-on settings, or when migrating from one environment to another.

Depending on the location from which you export, you can include all data or only data related to specific **Application interface preference** entries.

During import, you can choose:

- The import scope which determines the type of data to import. For example, you could choose to only import only global options which would include data from the **Reference data** and **Additional configuration** groups.
- To import only preferences and related data owned by a specific profile. This would help ensure that existing data for other profiles is not overwritten when migrating from one environment to another.

The following sections cover these topics in more detail:

- Exporting configuration settings [p 121]
- Importing configuration settings [p 123]

CHAPTER 23

Exporting configuration settings

To export all configuration settings:

1. Navigate to *Administration > Integration > TIBCO EBX® Data Exchange Add-on* and from the dataset's **Actions** menu select **Export XML**.
2. After supplying a file name and optionally specifying whether the file is indented and XML comments are omitted, click **Export**.

To export only data related to specific **Application interface preferences** records:

1. Navigate to *Administration > Integration > TIBCO EBX® Data Exchange Add-on > Application > Application interface preference*.
2. Select one or more records to export and from the table's **Actions** menu, select **Export preferences**. Note that if you do not select any records, the export includes all records in the table.
3. Supply a file name and select **Export**.

CHAPTER 24

Importing configuration settings

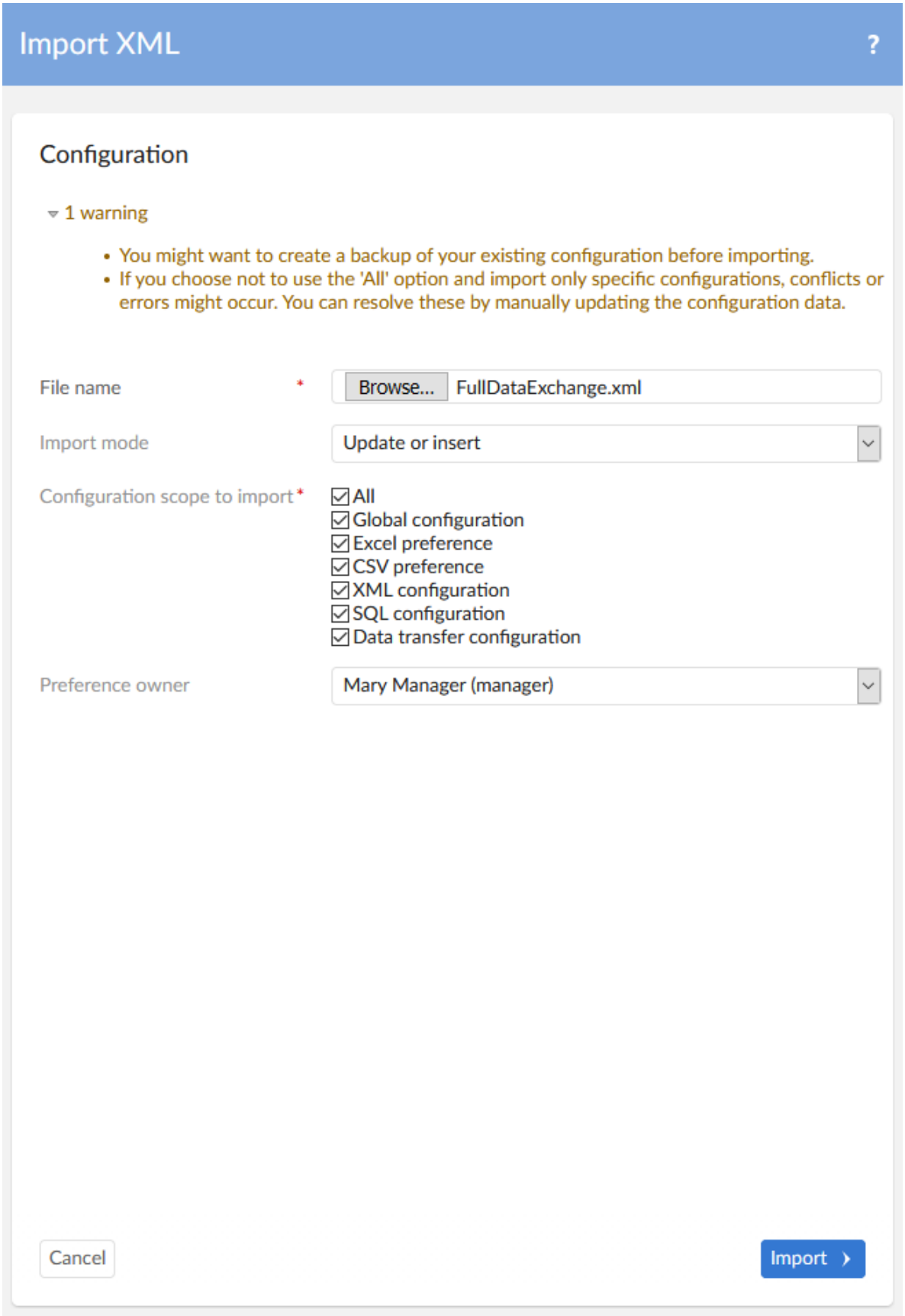
To import configuration and preference settings:

Attention

Before importing, you can create a backup of you existing configuration by performing the steps in Exporting configuration settings [p 121].

1. Navigate to *Administration* > *Integration* > *TIBCO EBX® Data Exchange Add-on* and from the dataset's **Actions** menu select **Import XML**.
2. On the import configuration page, specify the following:
   - The file to import that contains the configuration and preference data.
   - Optionally, choose the **Replace all** import mode if you want to completely overwrite existing data.
   - Use the checkboxes to determine the scope of the import. See the table below for information on how add-on configuration data corresponds to each option.

- If you only want to include preferences owned by a specific profile, use the **Preference owner** menu to select this profile.


Import XML
?
Configuration
1 warning
You might want to create a backup of your existing configuration before importing.
If you choose not to use the 'All' option and import only specific configurations, conflicts or errors might occur. You can resolve these by manually updating the configuration data.
File name *
Browse...
FullDataExchange.xml
Import mode
Update or insert
Configuration scope to import *
All
Global configuration
Excel preference
CSV preference
XML configuration
SQL configuration
Data transfer configuration
Preference owner
Mary Manager (manager)
Cancel
Import

3. After selecting **Import**, the add-on displays a results page where you can select a tab corresponding to each imported table to see the results.

The following table shows how the scope options correspond to which data is imported:

Option	Data imported
All	Everything in the add-on configuration dataset.
Global configuration	All data under the **Reference data** and **Additional configuration** groups.
Excel preference	All Excel preference settings in the **Application**, **Semantic model**, **Data model**, **Data mapping**, and **Path** groups. Additionally, the following tables from the **Reference data** group are included: • **Transformation function** • **Date time pattern** • **Validator** • **Style preference**
CSV preference	The data specific to CSV preferences from the same locations as **Excel preference** with the exception of the **Style preference** table.
XML configuration	The data specific to XML configuration settings from the same locations as **Excel preference** with the exception of the **Style preference** table.
SQL configuration	All SQL configuration related settings in the **Application**, **Semantic model**, **Data model**, **Data mapping**, and **Path** groups. Additionally, the following tables from the **Reference data** group are included: • **Transformation function** • **SQL data source** • **JNDI data source** • **Validator**
Data transfer configuration	The data specific to data transfer configuration settings from the same locations as **Excel preference** with the exception of the **Style preference** table.

CHAPTER 25

Procedural overview

You can extend the add-on's existing transformation catalog with your own custom Java implementations. Once implemented and deployed, an EBX® administrator can use the transformation when configuring the add-on. The following table highlights the process:

1) Development	Write Java classes required to implement a transformation. See Implementing a custom transformation [p 129] for code samples and information.
2) Register and deploy	Add your transformation to the add-on's existing catalog and deploy required JAR files. See Deploying and Adding to the add-on's catalog [p 133] for instructions. If you have not yet deployed a custom module for EBX®, see the EBX® user documentation for instructions.
3) Configuration	An administrator must create a configuration to add the transformation to a field mapping. See Using a custom transformation [p 135].

CHAPTER 26

Implementing a custom transformation

This chapter contains the following topics:

26.1 Overview

This section shows how to implement a basic transformation function for the EBX® Data Exchange Add-on. This type of implementation requires the following:

- A definition class to specify options for add-on transformation configuration requirements.
- The class that contains the transformation logic. For example, how to convert data types or transform values.

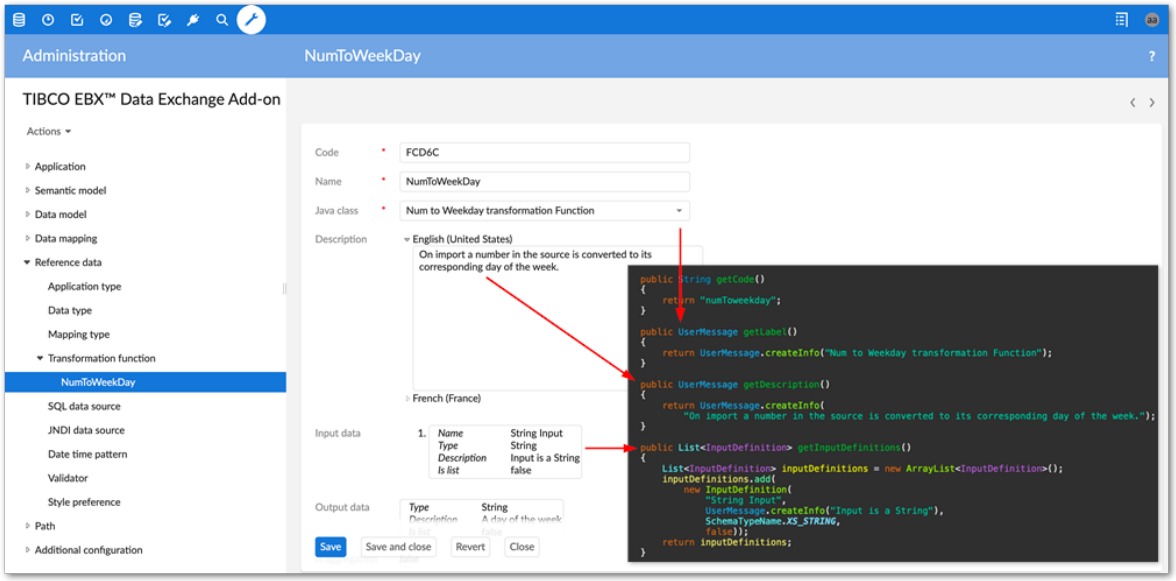
26.2 Transformation definition class

The following code sample shows a definition class for an add-on transformation function:

```
public class NumToWeekdayTransformationDefinition implements TransformationDefinition
{

 public String getCode()
 {
  return "numToweekday";
 }

 public UserMessage getLabel()
 {
  return UserMessage.createInfo("Num to Weekday transformation Function");
 }

 public UserMessage getDescription()
 {
  return UserMessage.createInfo(
   "On import a number in the source is converted to its corresponding day of the week.");
 }

 public List<InputDefinition> getInputDefinitions()
 {
  List<InputDefinition> inputDefinitions = new ArrayList<InputDefinition>();
  inputDefinitions.add(
   new InputDefinition(
    "String Input",
    UserMessage.createInfo("Input is a String"),
```

```
    SchemaTypeName.XS_STRING,
    false));
  return inputDefinitions;
 }

 public OutputDefinition getOutputDefinition()
 {
  return new OutputDefinition(
   UserMessage.createInfo("A day of the week"),
   SchemaTypeName.XS_STRING,
   false);
 }

 public List<ParameterDefinition> getParameterDefinitions()
 {
  return new ArrayList<ParameterDefinition>();
 }

 //You can use this transformation definition to automatically detect the file type. Based on the type, it can
 return a different transformation.
 public Transformation getTransformation(ServiceType serviceType)
 {
  switch (serviceType)
  {
  case SPREADSHEET_IMPORT:
   return new ConvertNumToWeekday();
  case CSV_IMPORT:
   //Add your own transformation for CSV or other formats.
  default:
   return null;
  }
 }

 public boolean isBidirectional()
 {
  return false;
 }

 public boolean isAggregation()
 {
  return false;
 }

}
```

Note that as shown in the image, the methods shown above return values used by the add-on to define a transformation configuration.



26.3 **Transformation implementation**

The following code sample shows an implementation of a transformation that takes a numeric value in the source and outputs the value's corresponding day of the week in the target:

```
public class ConvertNumToWeekday implements Transformation<ImportTransformationExecutionContext>
{
 private Locale locale;
 public void setup(TransformationConfigurationContext configurationContext)
  throws DataExchangeException
 {
  if (configurationContext == null)
  {
   throw new DataExchangeException(UserMessage.createError("Context is not initialized."));
  }
  this.locale = configurationContext.getSession().getLocale();
 }

 //This method gets the input data to transform and defines the transformation logic.
 public Object execute(ImportTransformationExecutionContext executionContext)
  throws DataExchangeException
 {
  if (executionContext == null)
  {
   throw new DataExchangeException(UserMessage.createError("Context is not initialized."));
  }

  //Obtain the value to import from the source application.
  Object inputValue = executionContext.getInputValue();
  if (inputValue == null)
  {
   return null;
  }

  //Performs a check on the target location.
  SchemaNode schemaNode = null;
  if (EBXField.class.isInstance(executionContext.getTargetField()))
  {
   EBXField ebxField = (EBXField) executionContext.getTargetField();
   schemaNode = ebxField.getSchemaNode();

   if (schemaNode.isComplex())
   {
    throw new DataExchangeException(
     UserMessage.createError(
      schemaNode.getLabel(this.locale)
       + " is a complex type node. The transformation function 'Convert an integer to a string and vice versa'
only supports simple type node."));
   }
  }

  //Sets how the data is transformed. In this case it is from one value to another. You could also specify that
data types be transformed, values concatenated, etc.
  try
  {
   switch (schemaNode.formatToXsString(inputValue))
   {

   case "1":
    return "Monday";
   case "2":
    return "Tuesday";
   case "3":
    return "Wednesday";
   case "4":
    return "Thursday";
   case "5":
    return "Friday";
   case "6":
    return "Saturday";
   case "7":
    return "Sunday";
   }

   throw new DataExchangeException(UserMessage.createError("Invalid input data."));
  }
  catch (ClassCastException ex)
  {
   throw new DataExchangeException(ex);
  }
```

```
  catch (ConversionException ex)
  {
   throw new DataExchangeException(ex);
  }
  catch (Exception ex)
  {
   throw new DataExchangeException(ex);
  }
 }
}
```

See Deploying and Adding to the add-on's catalog [p 133] for instructions on the next steps.

CHAPTER 27

Deploying and Adding to the add-on's catalog

The add-on ships with a catalog of several predefined transformations. To add your custom transformation to the add-on's catalog:

- Add the following code to the `handleRepositoryStartup()` method in your module's registration servlet.

```
TransformationCatalog.add(new NumToWeekdayTransformationDefinition());
```

- You must also add the transformation definition and implementation classes to a JAR file and include it in the same location on your server as `ebx.jar`.

> **Note**
>
> Until you complete the steps above, an administrator cannot use the custom transformation when creating a configuration. You can use the EBX® documentation for developers to get instructions on deploying a custom module.

See Using a custom transformation [p 135] for instructions on adding a custom transformation implementation to a mapping configuration.

CHAPTER 28

Using a custom transformation

This chapter contains the following topics:

28.1 Overview

This section gives administrators an example of how to configure the add-on to import and transform data from a spreadsheet. In this example, we also demonstrate the add-on's ability to automatically generate the components required for configuration by creating and saving an import preference. In the final step we add a the transformation to the auto-generated mapping configuration. Once these steps are complete we can re-import the file to apply the transformation.

Attention

The process demonstrated in this section might not be the best solution depending on your data structure. For example, if your target table uses an auto-generated primary key and the source has no primary key or it is not mapped, re-importing might result in duplicate records. To avoid this you can follow the general step by step instructions to configure manually. See Generating constant values [p 85] for more information.

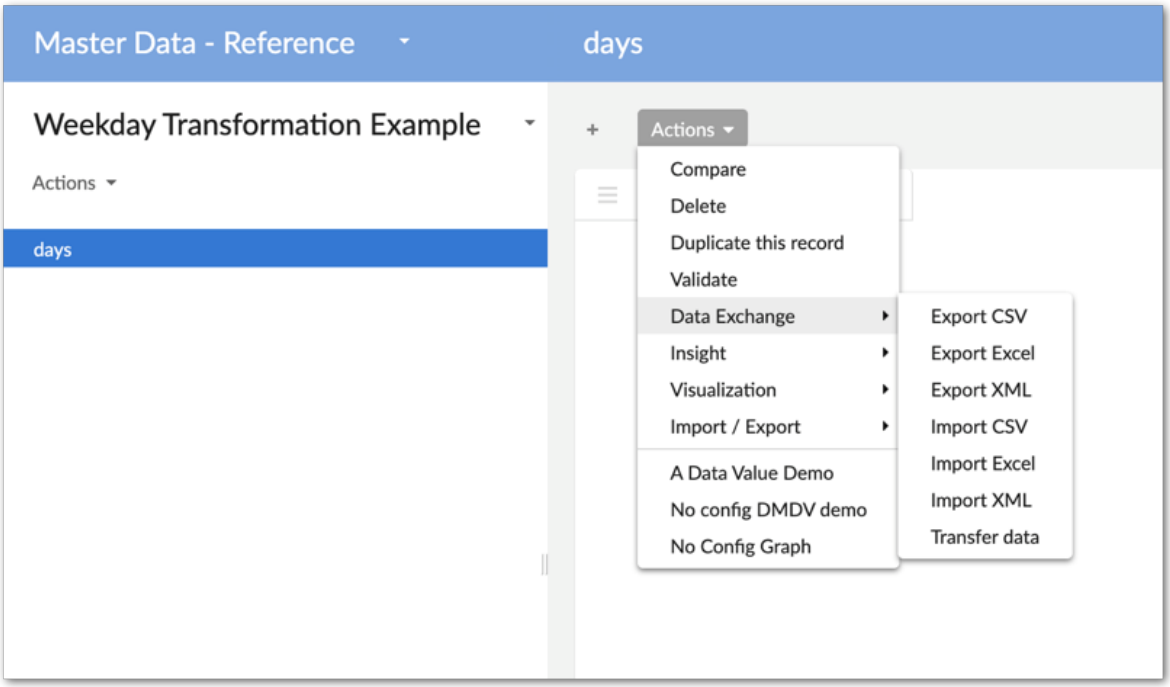
28.2 Creating the import preference and adding a transformation

When you import data from a spreadsheet, the add-on automatically creates required applications, interfaces, models, and mappings behind the scene. If you create a preference when importing, the add-on saves all of these settings. You can access the settings and add a transformation to a saved field mapping to alter how data is processed.

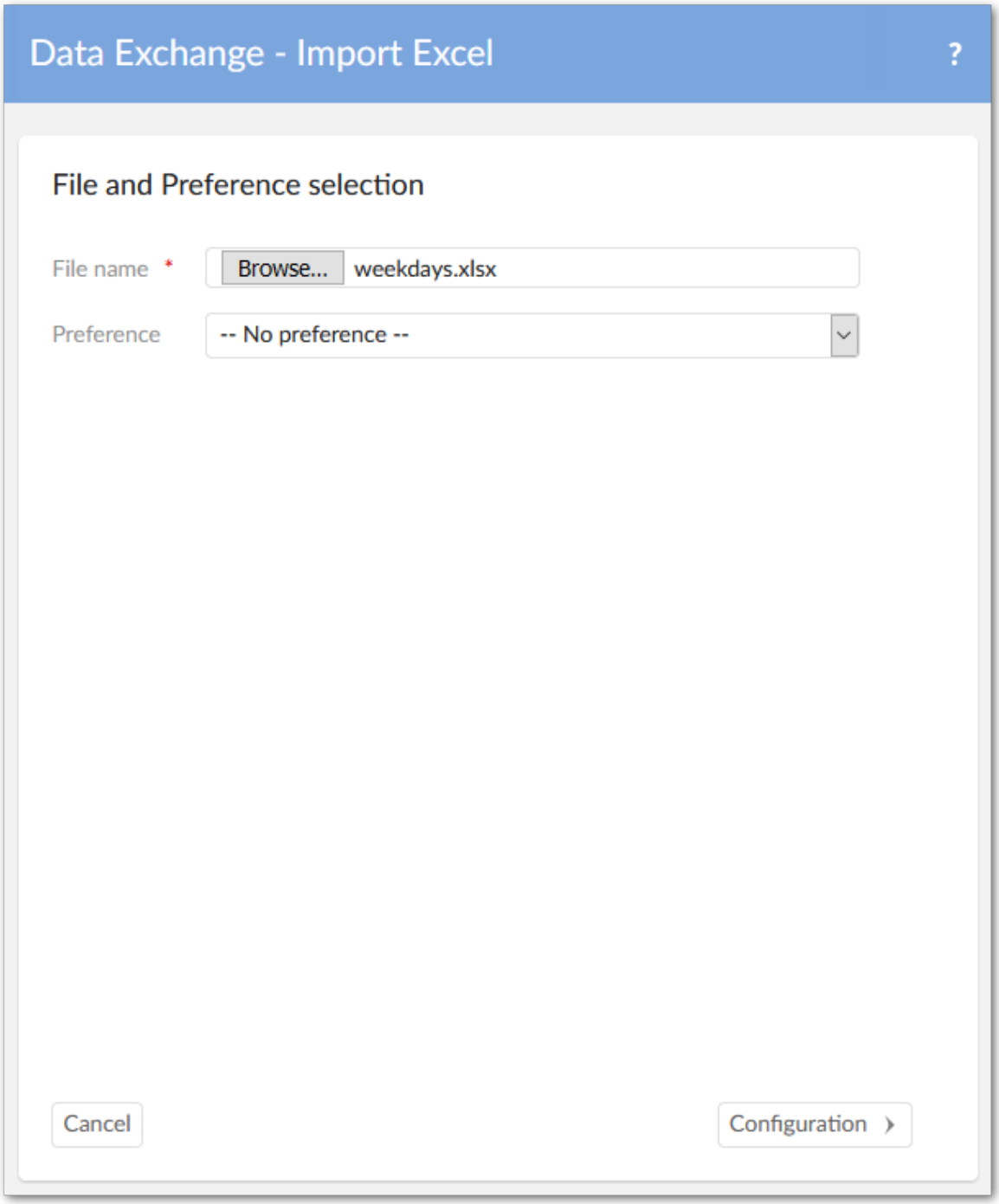
To create an import preference and add a transformation:

1. Open the table in EBX® where you want to import data.

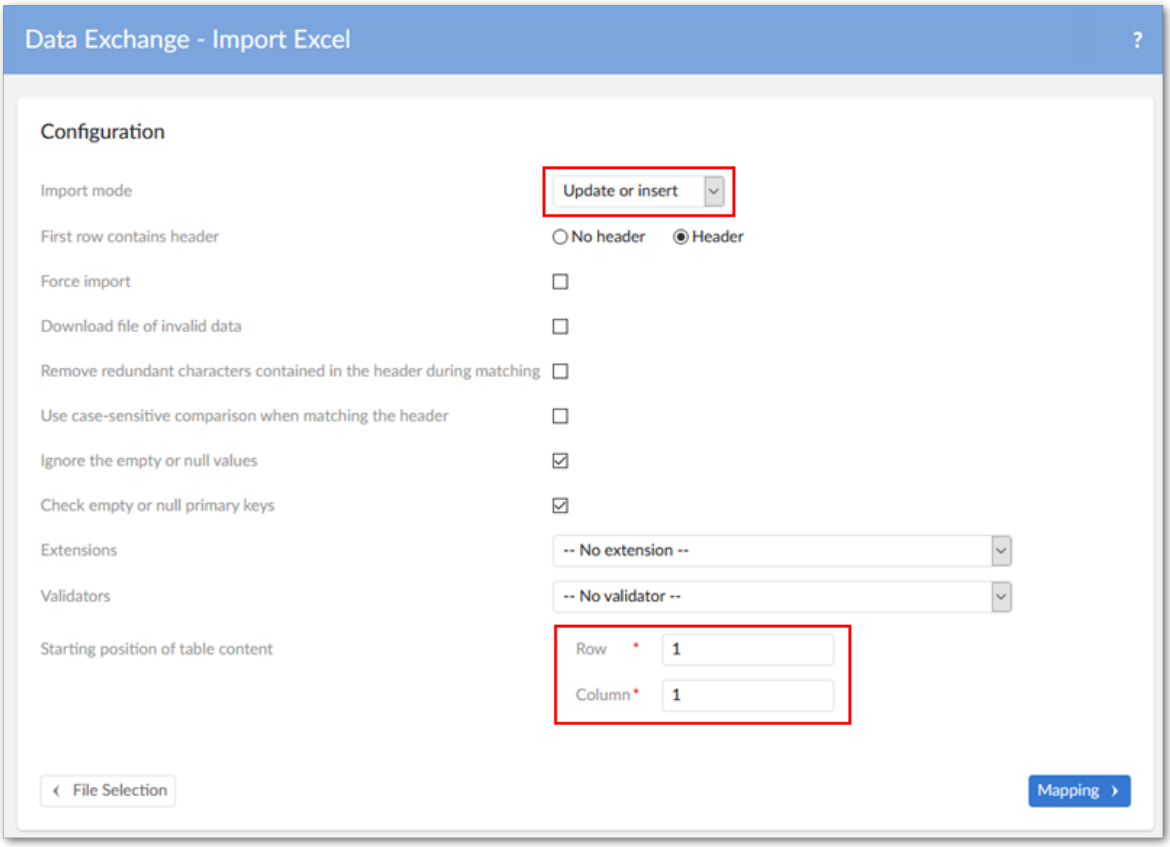
2. From the **Actions** menu, select **Data Exchange** and the import option. In this example we are importing from a spreadsheet so we use **Import Excel**.



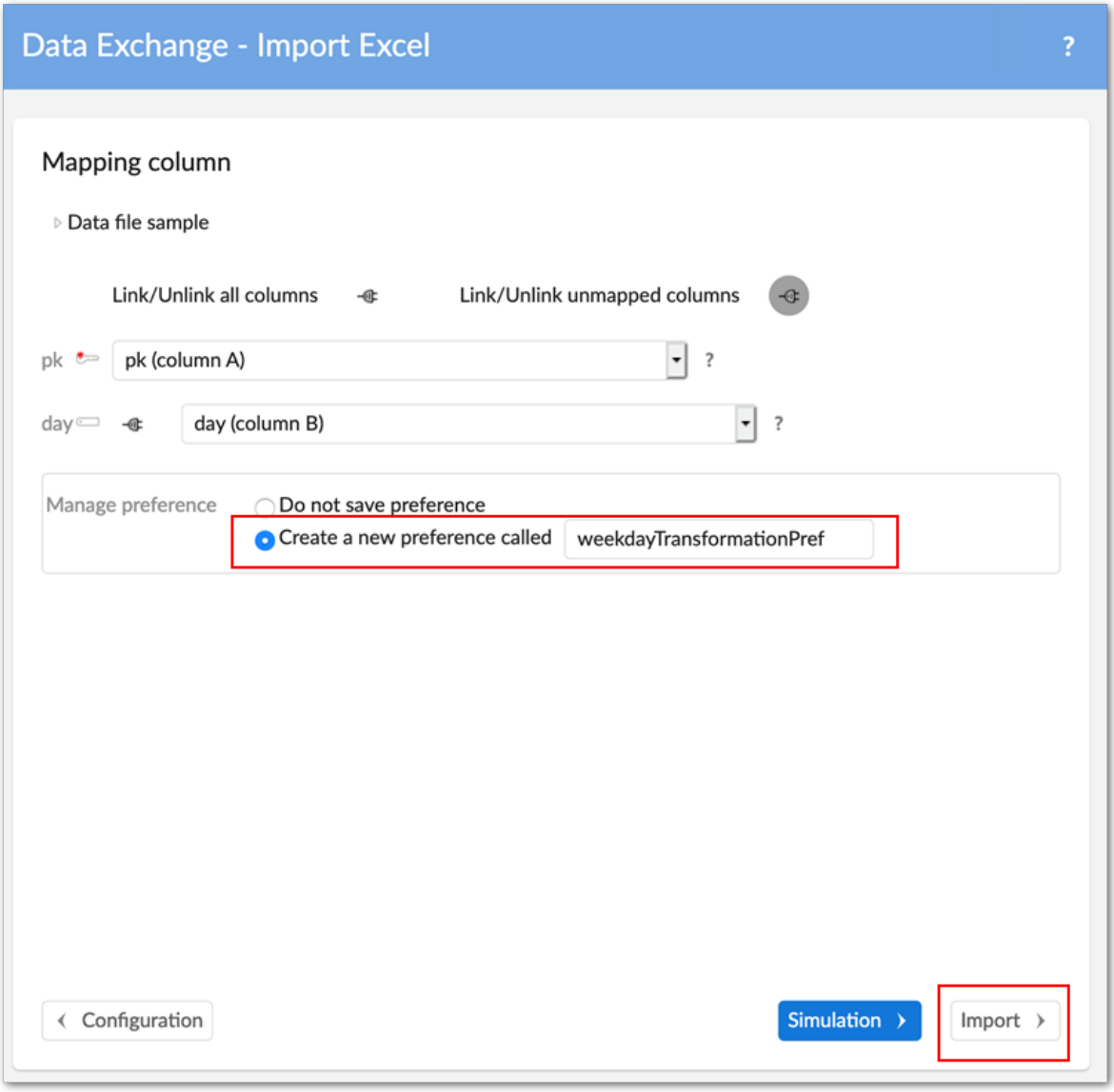
3. Select **Browse** to choose the file to import and select **Configuration** at the bottom of the screen.



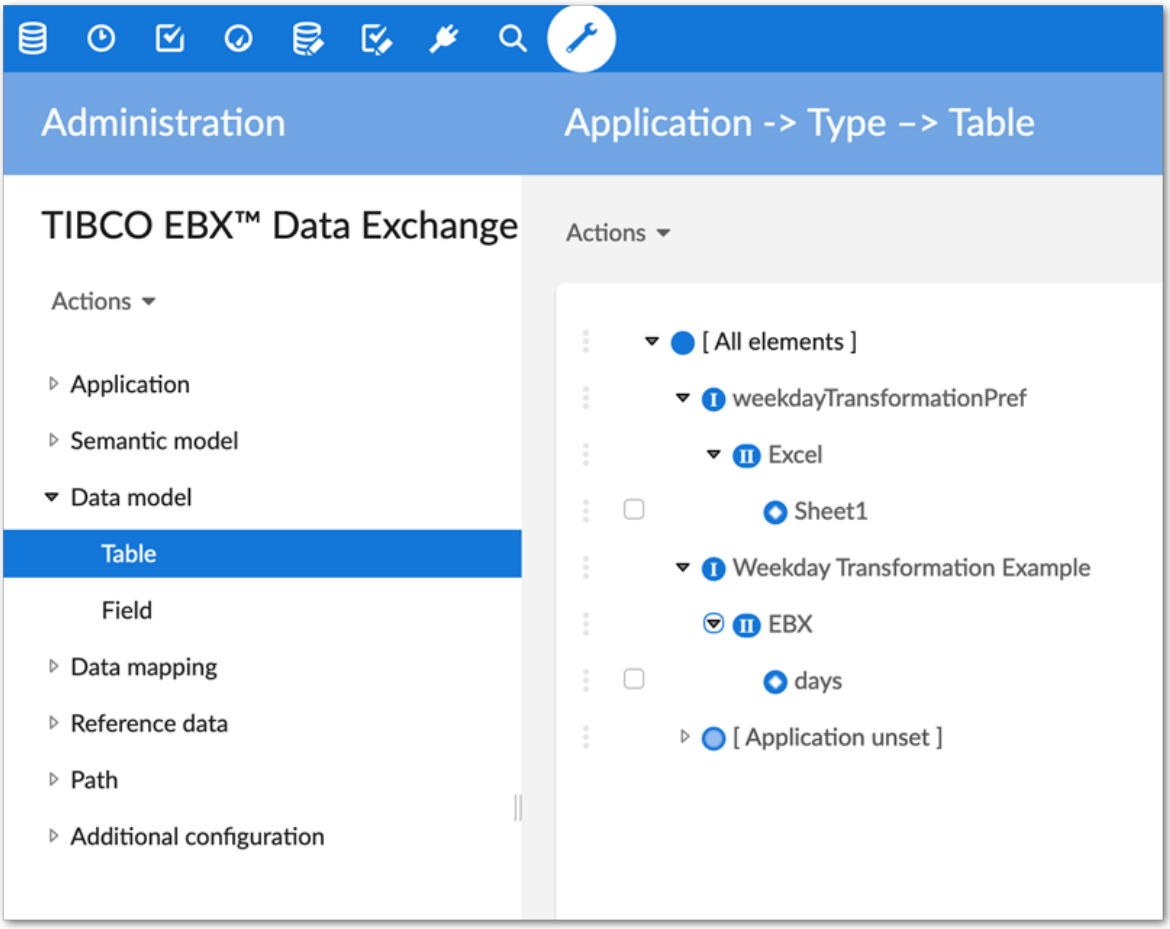
4. Double-check the default options here to ensure the mode and starting position of table data is set correctly and select **Mapping** at the bottom of the screen.



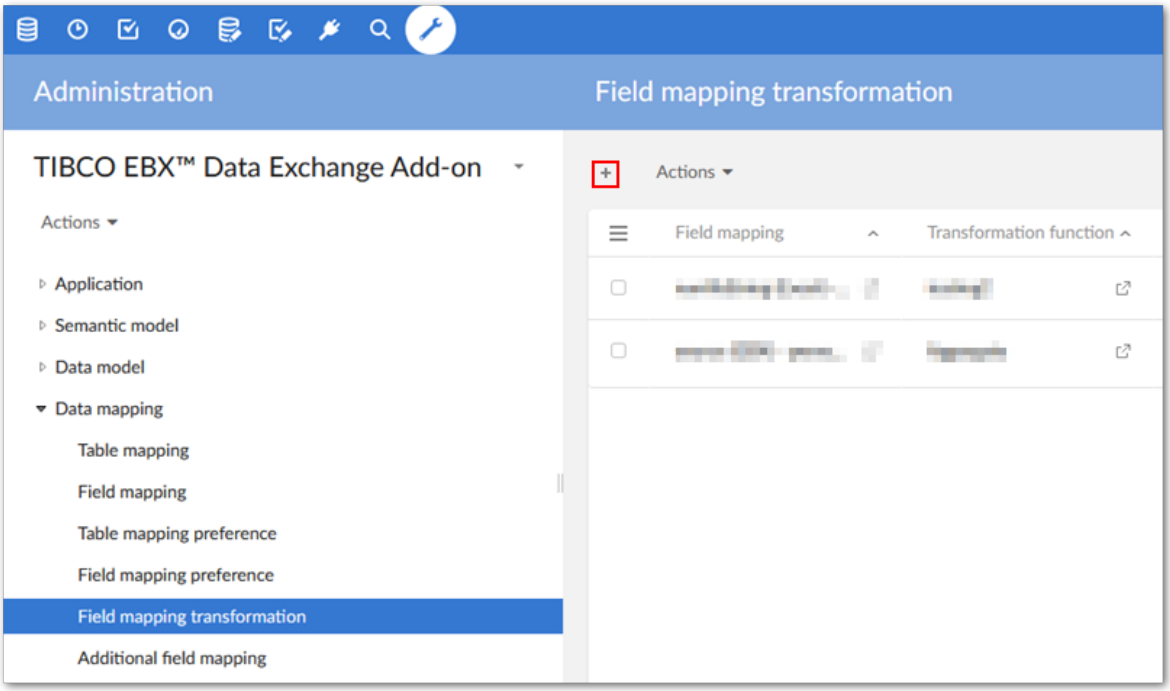
5. In the **Manage preference** box, select **Create** a new preference called and provide a name. After selecting **Import** at the bottom of the page you can close the result page and open the **Administration** area from the main menu.



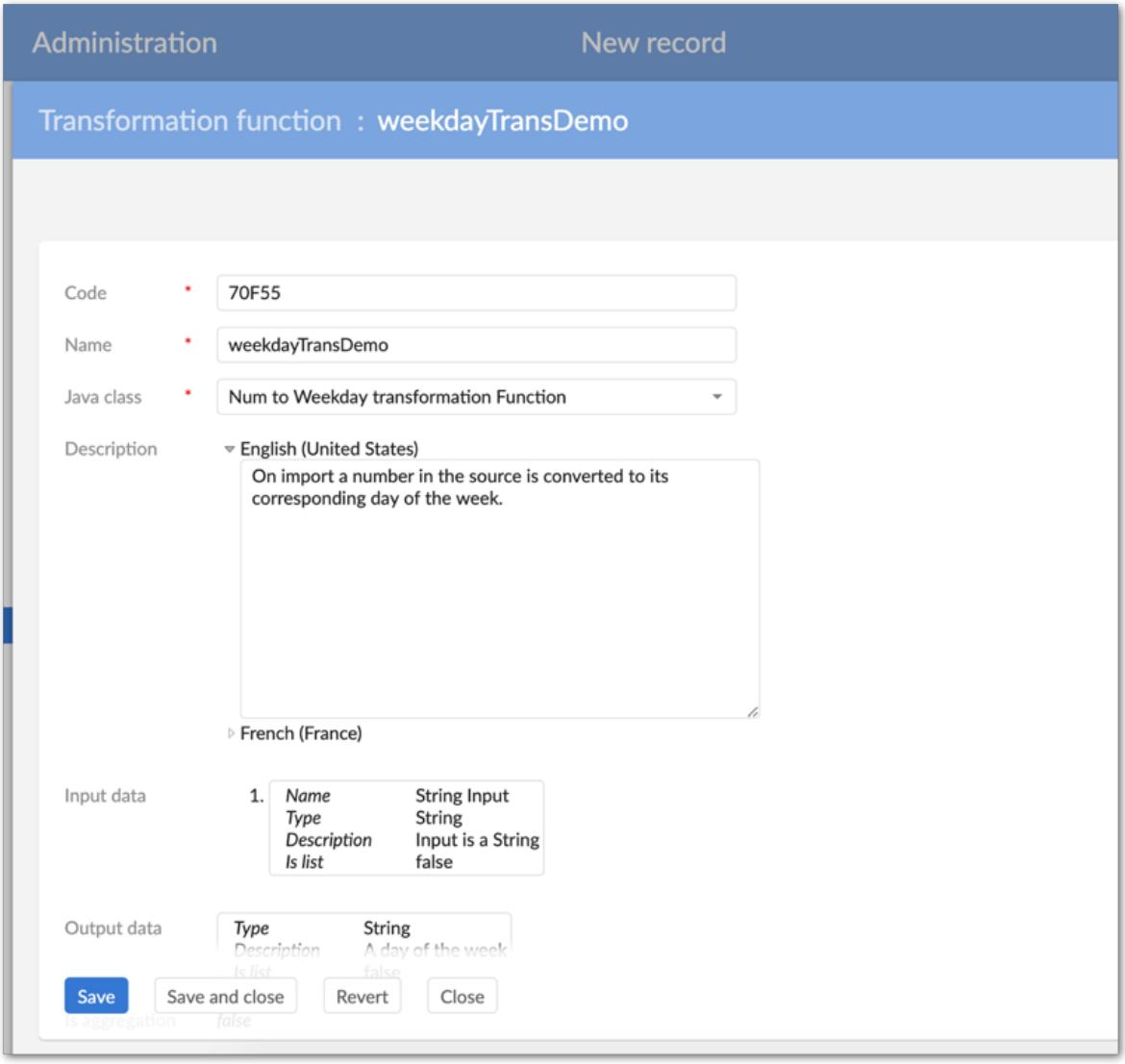
In the **Administration** area, browse the contents of the configuration area to see how configuration settings were generated.



6. We can now add our custom transformation built using the add-on's API by navigating to *Data mapping* > *Field mapping transformation* and creating a new record.

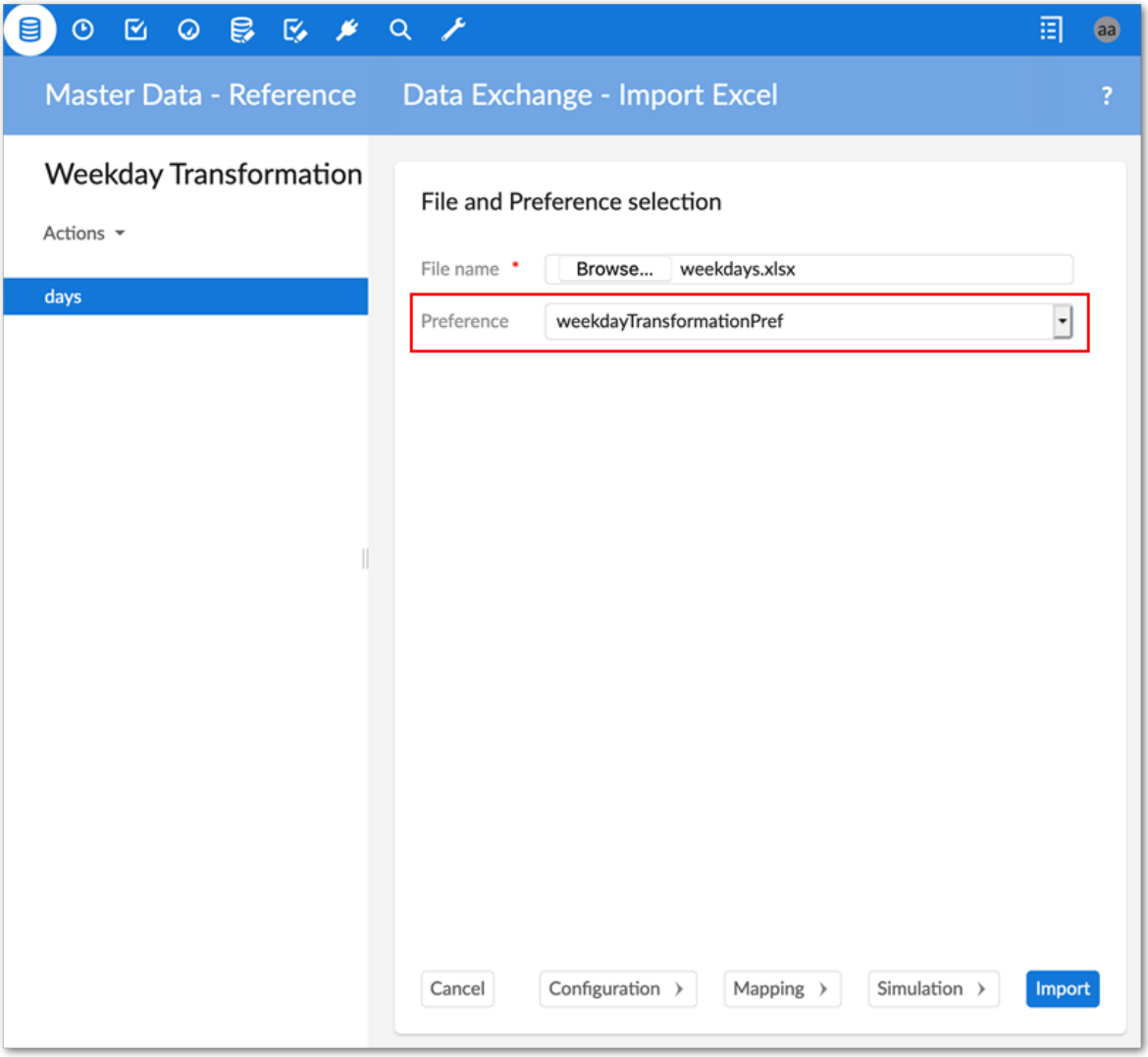


7. In the **Field mapping** drop-down menu, choose the mapping for the data to transform and use the **Transformation function** drop-down menu to select the custom transformation. After saving the record populates with the settings specified in the Java class.

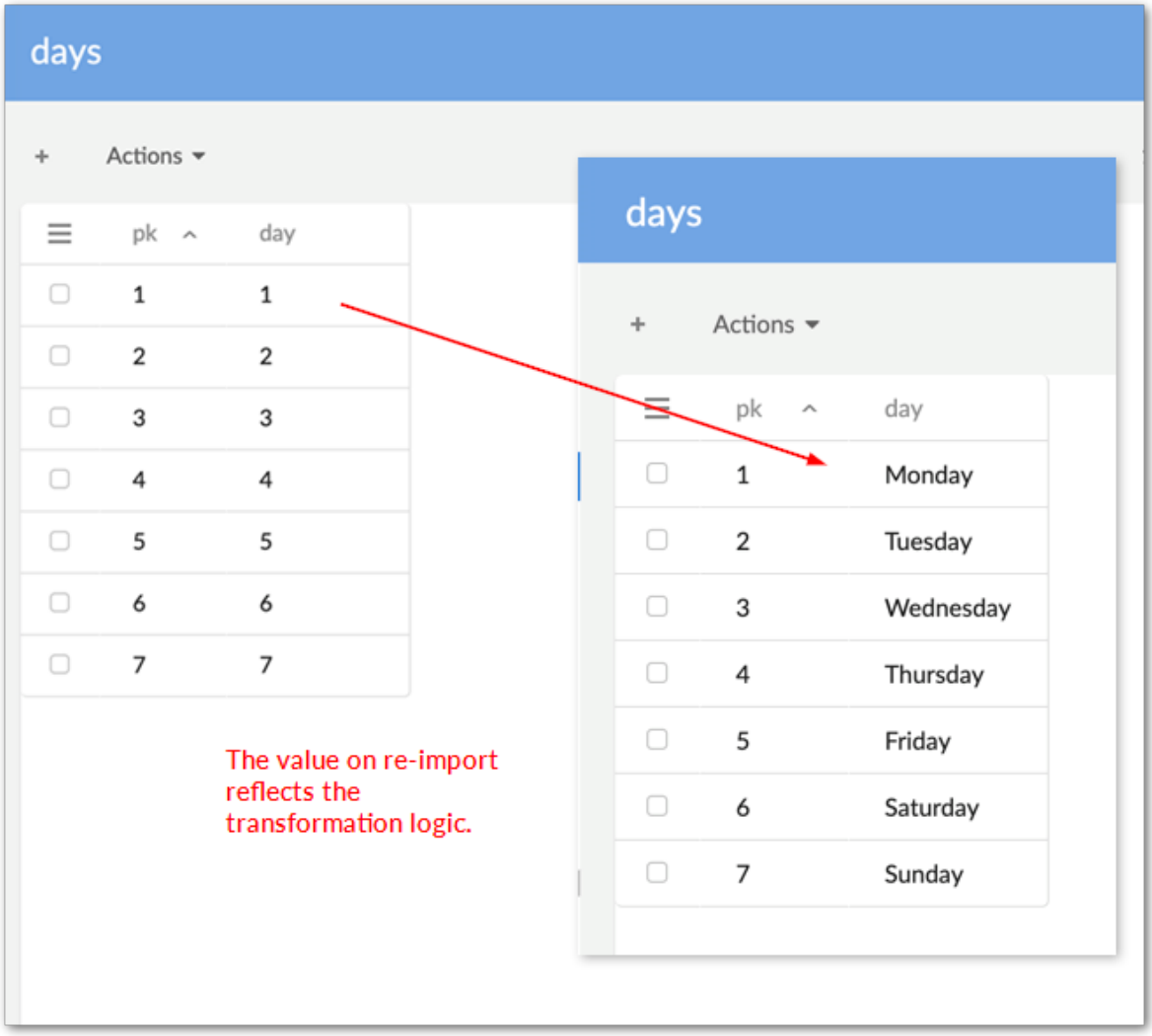


8. After saving and closing the record, navigate to the import target table from step 1 and re-import, making sure to select the preference you just created on the import's file selection page.

Note that depending on your data structure, you might need to update the import mode, or other options. After making changes, the **Mapping** screen allows you to update the preference, create a new one, or ignore changes.



Notice that upon re-import the add-on applied the transformation function to the data. Anytime this preference is used the function will apply.



Reference Guide

Administration Area

CHAPTER 29

Application domain

This chapter contains the following topics:

29.1 Function

The **Application** domain contains the declaration of every application that is involved in the data flow (import, export, transfer) managed by the TIBCO EBX® Data Exchange Add-on.

Application

An application is used as a source and/or target in the execution of a data flow process (import, export or data transfer).

Property	Definition
Universal name	Any naming convention is valid.
Logical name	The logical name is automatically provided by the add-on to ensure a unique identification.
Code	Any naming convention is valid.
Last modification date	Date of the last modification applied to the application.

Application by type

An application can be associated with one or many of the following types: Default SQL, SQL, CSV, Excel, Default XML, XML and EBX.

Property	Definition
Application	The reference to an application.
Application type	The type of the application.
Application path	When this is an EBX type application, the path gives the dataspace and dataset.

Version

An application can be associated to a version. This version is used as documentation included in the exported data file when the targeted format can integrate it (this is the case in XML as app info tag). It has no impact on the data mapping configuration management.

Property	Definition
Code	Any naming convention is valid.
Version	Any naming convention is valid.
Application by type	The reference to an application with its type.

Application interface preference

The preferences here store option values of an Application interface. These option values can be retrieved during import and export. The preferences are available from the **Load application mapping preferences** dropdown list (Import) and **Application mapping preferences** dropdown list (Export).

Property	Definition
Code	Code of an application interface configuration.
Name	Specific name of the application interface configuration object.
Application interface	Description of a mapping between two applications.
Selected table's paths	The selected tables that will be exported or imported.
First row contains header	Imported or exported file with header or no header.
File encoding	Specifies the character encoding to use. The default is UTF-8.
Import mode	Modes of import such as update and insert, update only, insert only and replace all content.
Separator	Describing separator option to use for importing/ exporting CSV file such as commas, semicolon, space, tab and other (in case of inputting a character that is customized by user).
Separator character	Describing a character separator in case of choosing "other" option when selecting a separator.
List separator	Specifies the separator to use for values lists.
List separator character	Specifies the separator to use for values lists in case of using "Other" option.
Line return	Type of line return used in this preference.
Delimiter character	The character that is used to mark the beginning and end of a cell in the CSV file.
Delimiter for string only	The delimiter is only used for string data types.
Download file of invalid data	Export all records with errors in their data rows to a downloadable file.
Save as type	Specific type of the exported Excel file includes: xls or xlsx.
Primary key label	Add an information column with the label of the primary key in exported file.
Primary key permalink	Add an information column with the hyperlink for the primary key in the exported file.
Foreign key label	Add an information column with the label of the foreign key in the exported file.

Property	Definition
Foreign key permalink	Add an information column with the hyperlink of the foreign key in the exported file.
Export mapping mode	Export all columns of table based on their order or on the mapping defined by the selected import preference.
Force import	Specifies that all triggers and constraints are disabled when this option is activated. Otherwise, all triggers and constraints are enabled.
Number format	The character used to denote a decimal.
Ignore the empty or null values	By default, the existing record is not updated with null and empty cell values from the imported file.
Check empty or null primary keys	The add-on verifies that all primary keys are mapped and validates the data before importing. If you disable this option, the add-on does not perform this verification.
Validate data before transformation	Determines whether data is validated before the transformation.
Transformer Java class	Stores the path of the old transformer class (from Adix).
Validator Java class	Stores the path of the old validator class (from Adix).
Remove redundant characters	Replace all line-breaks and continuous space characters in the Excel header with a single space character during matching.
Use case-sensitive comparison	Differentiate between lower-case and upper-case when matching the imported file's column header with the field's label in EBX®.
Include computed values	Specifies whether computed values must be included in the export.
Export static enumerations	The exported file will contain enumerations matching the one in the data model. Warning: it leads to increase the size of the file.
Include reference sheet	You can add a reference sheet called 'Reference table mapping' at the end of the Excel file. This sheet contains metadata such as the sheet name, the table label and the table path used to detect the mapping between sheets and tables upon import.
Date format	Specify the format of date values in the CSV exported file.
Date/time format	Specify the format of date/time values in the CSV exported file.
Force precision as displayed	Specifies the accuracy of numbers when importing Excel. If set to 'Yes': The displayed value of numbers in each cell will be read and imported. If set to 'No': The accuracy value of numbers in each cell will be read and imported. Default value: 'No'. This feature is only applied to the Number format cells (Number is used for general display of numbers). For the other format cells, the accuracy value in each cell will be read and imported.
Export the ignored field as a blank column	If the option is activated, the ignored field will be exported as a blank column. Otherwise, the ignored field will not be exported. This option is only available on Excel export. Default value: 'Yes'.

Property	Definition
Info message export	The exported file will contain any information level validation messages.
Warning message export	The exported file will contain any warning level validation messages.
Error message export	The exported file will contain any error level validation messages.
From referenced tables	Include details of records referenced in the current export (might include records from external tables).
From tables that reference the selected record	Include data from external tables that reference the selected record.
Owner	Specifies the owner user created preference.
Restriction policy	If ‘Restriction policy’ value is ‘Yes’, the minimum permissions of these restricted rules are applied. If ‘Restriction policy’ value is ‘No’, the maximum permissions of all matching rules are applied.
Permissions group	
User profile	The profile impacted by the permission settings.
Use the preference	Determines whether the specified profile can use the preference. If disabled, the preference will be hidden from the profile.
Modify the preference	Determines whether the specified profile can modify the preference.
Delete the preference	Determines whether the specified profile can delete the preference.

Interface

An interface allows data to flow between two applications.

Property	Definition
Source application	Reference to the application with its type playing the role of the source.
Target application	Reference to the application with its type playing the role of the target.
Description	Description of the interface.

Object class by Application

An application can hold one to many Object Class items. You can use the Object Class to create the relationship between a Table in the Data model and its related item in the Semantic model. Several tables can be linked to the same Object Class.

Property	Definition
Application	Reference to an application.
Object class	Reference to an Object Class.
Description	Description of the relation between the application and the Object Class.

CHAPTER 30

Semantic model domain

This chapter contains the following topics:

30.1 Function

The **Semantic model** provides a business data architecture based on Object Class and Property items. An Object Class can be linked to one or many tables. A Property can be linked to one or many fields.

Object class

An Object Class is a business concept that can be linked to one or many tables declared in the data models.

Property	Definition
Code	Any naming convention is valid.
Name	Any naming convention is valid.
Is removed	If set to True: The Object Class is no longer valid. It is logically removed. By using the purge service, the data will be physically deleted. If set to False: The Object Class is valid.
Description	Description of the Object Class.

Property by Object Class

A property is held by one or many Object Class items.

Property	Definition
Object Class	Reference to an Object Class.
Property	Reference to a Property held by the referenced Object Class.
Description	Description of the association between the Object Class and the Property.

Property

A Property is a business concept that can be linked to one to many fields declared in the data models.

Property	Definition
Code	Any naming convention is valid.
Name	Any naming convention is valid.
Is removed	If set to True: The Property is no longer valid. It is logically removed. By using the purge service, the data will be physically deleted. If set to False: The Property is valid.
Description	Description of the Property.

CHAPTER 31

Data model domain

This chapter contains the following topics:

31.1 Function

The **Data model** provides a logical data architecture based on Table and Field items. The data mapping configuration is also based on these items.

Table

A table is a container of data that depends on the application type. For EBX type applications, it is directly a table in EBX®. For XML type applications, it is a node in the XML file.

Property	Definition
Code	Any naming convention is valid.
Application	Reference to the application that owns the table.
Application type	Type of the application.
Object Class	The table can be linked to an Object Class. This is the way to make an association between the logical data architecture and the business architecture.
Name	Logical name of the table.
Label	Label of the table.
Path	The path of the table.
Is removed	If set to True: The Table is no longer valid. It is logically removed. By using the purge service, data will be physically deleted If set to False: The Table is valid.
Description	Description of the table.
Index table	Determines the table's index position. During Excel export, this determines to which sheet number this table's data will be exported.

Field

A field is held by one Table only.

Property	Definition
Code	Any naming convention is valid.
Table	Reference to the table that owns the field.
Property	The field can be linked to a Property. This is the way to make an association between the logical data architecture and the business architecture.
Name	Logical name of the field.
Label	Label of the field.
Data type	Reference to a Data type for the field.
Parent field	Foreign key to a parent field in case the current field is a child node of another complex field.
Order	Order of the field in table. The first position is '0'.
Path	The path of the field.
Is removed	If set to True: The Field is no longer valid. It is logically removed. By using the purge service, data will be physically deleted. If set to False: the Field is valid.
Description	Description of the field.

CHAPTER 32

Data mapping domain

This chapter contains the following topics:

32.1 Function

The **Data mapping** domain contains the data mapping configuration for tables and fields.

Table mapping

The data mapping configuration between the tables.

Property	Definition
Source table	Reference to the table used as the source for the data flow.
Target table	Reference to the table used as the target for the data flow.
Source application	Used to sort the records based on the application name and type. This data is computed automatically from the source table value.
Java class	This property specifies the Java class used to validate data on import, or transfer. This class ensures that only records that meet specific conditions are imported, or transferred. When used in a bidirectional data transfer this class applies when data moves from the table specified in the 'Source table' property to the table in the 'Target table' property.
Java class description	Description of the 'Java class'.
Inverse java class	This property specifies an optional Java class you can use to validate data in a bidirectional transfer. The class you specify here validates data when it moves in the inverse direction of the table mapping. So, it applies when data moves from the specified 'Target table' to the 'Source table'.
Inverse java class description	Description of the 'Inverse Java class'.

Field mapping

The data mapping configuration between the fields.

Property	Definition
Source field	Reference to the field used as the source for the data flow.
Target field	Reference to the field used as the target for the data flow.
Source application	Used to sort the records based on the application's name and type. This data is computed automatically from the source field value.
Reference field	Contains the path to any referenced field.
Mapping type	The mapping can be performed in a 'direct' way (the target field is equal to the source field) or based on split, or aggregation policies. In the current version of the add-on, only the direct mapping type is available.

Table mapping preference

This table stores any data mapping configuration preferences between tables.

Property	Definition
Code	Any naming convention is valid.
Name	Any naming convention is valid.
Ignored table	The table will be ignored during import/export execution if value is set to True.
First row of content	The first row of table content in imported file.
First column of content	The first column of table content in imported file.
Export with title	Exported Excel file will contain the title if value is set to True.
Title of table	Any naming convention is valid.
Style for title	Reference to a style object on which title style is defined.
Export with subtitle	Exported Excel file will contain the subtitle title if value is set to True.
Subtitle of table	Any naming convention is valid.
Style for subtitle	Reference to a style on which the subtitle style is defined.
Export with border	Cell of export excel file has borders if value is set to True.
Style for column title	Reference to a style object on which column title style is defined.
Style for data content	Reference to a style object on which data content style is defined.
Table mapping	Reference to a table mapping.
Application interface preference	Reference of an application interface preference.

Field mapping preference

This table stores any data mapping configuration preferences between fields.

Property	Definition
Code	Any naming convention is valid.
Ignored field	Field is ignored during import/export execution if value is set to True.
Date time pattern	Date time pattern that is used for importing/ exporting CSV.
Field mapping	Reference to a field mapping.
Table mapping preference	Reference of a table mapping preference.

Field mapping transformation

The data transformation applied during field data mapping.

Property	Definition
Field mapping	Reference to a Field mapping declaration.
Transformation function	Reference to the transformation function to be applied.
Order	Execution order of the transformation function in case several functions must be applied. When a field mapping contains multiple transformation functions, the function with the lowest order will be applied.
Parameters	Lists each input parameter's name and value. Values can be edited.

Additional field mapping

The **Additional field mapping** table allows you to configure fields that are involved in a one-to-many fields transformation function, such as aggregation and split.

Property	Definition
Field mapping	Reference to a 'Field mapping' declaration.
Additional field	Field involved in the mapping.
Order	Order of source field when executing aggregation of fields or order of target field when executing a split of fields.

Additional field mapping transformation

The **Additional field mapping transformation** table is used when you have to declare a transformation function on a field that is used in an **Additional field mapping** record.

Property	Definition
Additional field mapping	Reference to an 'Additional field' mapping declaration.
Transformation function	Reference to the transformation function to be applied.
Order	Execution order of the transformation function in case several functions must be applied.

CHAPTER 33

Reference data domain

This chapter contains the following topics:

33.1 Function

The **Reference data** domain stores values used by other configuration options.

Application type

The current version of the add-on manages the following types: Default SQL, SQL, CSV, Excel, Default XML, XML and EBX.

Property	Definition
Code	Code of the application type.
Name	Name of the application type.

Predefined Application type

The add-on provides these predefined Application types.

Application type	Description
EBX	An EBX type application is referenced through a dataset. Its tables and fields are described through an EBX® path.
Default XML	A 'Default XML' type application relies on default XML paths automatically created by the add-on when an Export XML process executes.
XML	An XML type application relies on user-defined XML path declarations.
Default SQL	A 'Default SQL' type application relies on default SQL paths automatically created by the add-on when an Export SQL process executes.
SQL	A defined configuration is required. The **Generate models** service can be used to look up SQL channels and generate tables and fields for the configuration. Note, that SQL functionality is currently limited to the Default SQL application type.
CSV	A CSV application relies on defined paths that correspond to an external CSV file.
Excel	An Excel application relies on defined paths that correspond to an external Excel file.

Data type

This table references all the possible data type of a field.

Property	Definition
Code	Code of the data type.
Name	Name of the data type.

Mapping type

The mapping types that the add-on can use.

Property	Definition
Code	Code of the mapping type.
Name	Name of the mapping type.
Description	Description of the mapping type.

Predefined Mapping type

The add-on provides this predefined Mapping type.

Mapping type	Description
Direct	The mapping between the source and target fields is executed in a direct way: the target value is equal to the source value. There is no aggregation and/or split of fields values.
Split	The source field is split into two or more target fields.
Aggregate	Two or more source fields are aggregated to one target field.

Transformation function

During data mapping, the data value can be transformed before moving to the target application. A portfolio of predefined functions is delivered with the add-on and you can enrich it with bespoke functions. An API allows you to develop these new functions.

Property	Definition
Code	Code of the transformation function.
Name	Name of the transformation function.
Java class	Java class of the transformation function.
Description	Description of the transformation function.
Input data	List of input data used by the transformation function.
Output data	List of output data provided by the transformation function.
Is aggregation	If set to 'Yes': the transformation function is used for aggregating fields and will be available when the mapping type = 'Aggregate'.
Is bidirectional	If set to 'Yes': the transformation function can be executed in bidirectional mode. This means that the configuration from T1 to T2 tables is runnable for T2 to T1.
Parameters	List of parameters that allow you to configure and adapt the behavior of the transformation function.

Predefined Transformation function

The add-on provides these predefined Mapping types.

Transformation function	Description
No export	The source field value is not exported.
No import	The source field value is not imported.
No transfer	The source field value is not transferred.

SQL data source

The **SQL data source** table allows you to associate an EBX® data model with an external SQL database table or view. This allows users to import and export between the two components.

Property	Definition
Code	Code of the SQL data source.
Name	Name of the SQL data source. If you want to use a connection configured in the **JNDI data source** table, enter the value from that table's **Name** property.
Data model	Stores the information schema location of a data model in EBX®. Each data model is configured to a JNDI Data Source of EBX®.
Table name pattern	A string that specifies a table naming pattern. When importing or exporting SQL, the add-on will only allow users to select from tables or views whose names match this pattern.
Schema name pattern	A string that specifies a schema naming pattern. When importing or exporting SQL, the add-on will only allow users to select from tables or views whose names match this pattern.
Description	Description of the SQL data source.

JNDI data source

The **JNDI data source** Specifies the configuration used when establishing a connection to an external database. Creating the connection via this table alleviates the need to create the configuration in your application server.

Property	Definition
Code	Code of the JNDI data source.
Name	Name of the JNDI data source. This name must be unique as you need to use it in the **SQL data source** table to associate this connection information with a data model.
URL	A valid database URL in the form: jdbc:subprotocol:subname.
User	The database user on whose behalf the connection is being made. These permissions will be granted to each EBX® user that accesses this data source.

Date time pattern

Stores the date or date/time pattern string for the import and export processes.

Property	Definition
Pattern	Stores a string that specifies the date or date/time pattern.
Mode	Stores the mode of date or date/time or time pattern string. Available values are: Default CSV pattern, Default Excel pattern, Default time pattern and Common pattern (use for customer date or datetime).
Type	Stores the type of date or date/time or time pattern string. Available values are: Date pattern, Time pattern and Date time pattern.

Validator

Stores references to Java classes used to validate imported, or transferred data.

Property	Definition
Code	A unique identifier for this validator.
Java class	This property specifies the Java class used to validate data on import, or transfer.
Label and description	An optional label and description for this validator.

Style preference

This table contains all style configurations such as font type and color.

Property	Definition
Code	Any naming convention is valid.
Row position	Row position of element on which the style is applied.
Column position	Column position of element on which the style is applied.
Font name	Font name of the style.
Font size	Font size of the style.
Bold style	Font weight of the style.
Italic style	Font has italic style if value is set to True.
Underlined style	Text will be underlined if value is set to True.
Font color	Font color of the style.
Background color	Background color of the style.

CHAPTER 34

Path domain

This chapter contains the following topics:

34.1 Function

The path domain contains the declaration of all paths used in the data mapping configurations. The **Path** table collects all the path declarations used in the data mapping configuration.

Property	Definition
Application type	A reference to the application type (EBX, Default SQL, XML and XML).
Path type	Either Table, Field or Application.
Path	The path value.
SQL data source	Records of SQL data source table which define the data model and the corresponding external database information.
Comment	Description of the path.

CHAPTER 35

Additional configuration domain

This chapter contains the following topics:

35.1 Function

The **Additional configuration** domain contains some additional configuration options for dataspace and import preferences and permissions.

Import preference

These options allow you to set some additional import preferences.

Property	Definition
Force precision as displayed	Specifies the accuracy of numbers when importing Excel. If set to 'Yes': The displayed value of numbers in each cell will be read and imported. If set to 'No': The accuracy value of numbers in each cell will be read and imported. Default value: 'No'. This feature is only applied to the Number format cells (Number is used for general display of numbers). For the other format cells, the accuracy value in each cell will be read and imported.
Display precision setting in UI	Specifies whether to display the 'Force precision as displayed' option in user interface. If 'No': The 'Force precision as displayed' option is not displayed on the Excel import configuration screens. If 'Yes': The option is displayed on the Excel import configuration screens and user can change its value. Default value: 'No'.
Allow transient record context management	A transient record context holds temporary values used when creating a new, or updating an existing record. If table triggers are activated and you enable this option, the `handleNewContext()` method will be called when creating or updating records during import or transfer. Please refer to the EBX® API documentation for more information about the `handleNewContext()` method.

Default option values for Import and Export

The **Import/Export** table allows you to define default settings for each import and export format. For descriptions hover your mouse over each property and click the '?' icon to open the associated tooltip.

Global permissions

The **Global permissions** table allows you to define which screens display for profiles during import and export. It also allows you to specify the import modes accessible by specific profiles for import

services. Each configuration record for permissions contains tabs that allow you to specify the user or role these settings apply to and set permissions for each supported data exchange format.

Property	Definition
Profile	Defines the user/role on which the permission applies.
Restriction policy	Defines whether the permission is restrictive or not. If more than one set of permissions is associated with the same profile, the Restrictive policy setting determines whether the least restrictive or most restrictive policy settings apply.
Configuration	Defines the permission on the Configuration screen.
Mapping	Defines the permission on the Mapping table & Mapping column screen.
Simulation	Defines the permission to access the Simulation screen
Update or insert	Grant permissions for a specific profile to access a set of import modes:
Update only	Grant permissions for a specific profile to access a set of import modes:
Insert only	Grant permissions for a specific profile to access a set of import modes:
Replace all content	Grant permissions for a specific profile to access a set of import modes:

Import Options

CHAPTER 36

CSV import options

This chapter contains the following topics:

36.1 Overview

This section describes the fields and available options when importing from an CSV file. You initiate import from a table's *Actions > Data Exchange > Import CSV* menu. CSV import presents you with several pages. Follow the links in the descriptions below for detailed information on each page:

- File selection—From this page you can browse to select the file to import.
- Preferences—Use the drop-down menu on this page to specify an existing set of preferences to use for this import.
- Configuration page—The options here determine basic configuration information.
- Column mapping—This page allows you to map the columns from tables to those in the Excel file.
- Simulation page—From this page you can run a simulation and view results prior to performing the actual import operation.
- Results page—This page shows you the import results.

Preferences page

Preferences allow you to automatically populate fields, and map between a source and target using saved information. This page contains the options shown in the following table:

List	Definition
Preference	Choose a preference to load configuration and mapping information.

Configuration page

This page allows you to modify behavior of the import. Part of the modification can include specifying Java classes to transform, or validate data. This page contains the options shown in the following table:

Property/Field/List	Definition
Import mode	• **Update or insert**: If a record with the same primary key already exists in the target table, it is updated. Otherwise, a new record is inserted. • **Insert only**: Only record creations are allowed. If a record exists in the target table with the same primary key as in the source, an error is returned. **Note** Use of this mode is not recommended when importing a large volume of data. When working with larger amounts of data, you can instead use the **Update or insert**, or **Update only** modes. • **Update only**: Only modifications of existing records are allowed. If a record with the same primary key does not exist in the target table, an error is returned. • **Replace all content**: All existing data in the table will be deleted before importing the data from the file.
First row contains header	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
File encoding	Specifies the character encoding to use. The default is UTF-8.
Separator	Specifies the separator to use. The default is ';'.
List separator	Specifies the separator to use for values lists.
Line return	Specifies the line return used in the file.
Delimiter	The character used to mark the beginning and end of a cell in the CSV file. By default, CSV files do not use a delimiter. If you select 'Only string', the delimiter is only used for String data types.
Decimal symbol	The character used to denote a decimal.
Force import	Specifies that all triggers and constraints are disabled when this option is activated. Otherwise, all triggers and constraints are enabled.
Download file of invalid data	Export all records with errors in their data rows to a downloadable file.
Use case-sensitive comparison when matching the header	Differentiate between lower-case and upper-case when matching the imported file's column header with the field's label in EBX®.
Ignore the empty or null values	By default, the existing record is not updated with null and empty cell values from the imported file.
Check empty or null primary keys	The add-on verifies that all primary keys are mapped and validates the data before importing. If you disable this option, the add-on does not perform this verification.

Property/Field/List	Definition
Extensions	Specifies the Java class that is implemented to transform data during import/export processing.
Validators	Specifies the Java class that is implemented to validate data during import.

Column mapping page

This page allows you to map source columns in the CSV file with EBX® table columns. Use the drop-down list next to each table column to specify which source gets mapped to which target. You can use the tooltip icon next to each mapping to display additional transformation information. Additionally, at the top of the screen, you can expand **Data file sample** to see a preview of how the current mapping would import.

Simulation page

This page allows you to use the current configuration and mapping settings to simulate import. The simulation results can include different levels (error, warning, and information) of technical, validation, and business messages. Use the result information to make any needed changes before performing the actual import.

The table below describes the page's main components.

Options	Definition
Validate imported records only	When selected, the validation report only runs on the records being imported. If unselected, the report also includes all records in the target table.
Skip validation report	When checked, the simulation result does not include the detailed validation report.
Simulation stopped	Select conditions that determine when to stop the simulation.
Simulation result	Displays simulation results on a table-by-table basis. Each tab corresponds to a simulation for a table.
Import option	Choose whether to stop the import process when there are validation errors.

CHAPTER 37

Excel import options

This chapter contains the following topics:

37.1 Overview

This section describes the fields and available options when importing from an Excel file. Note that options vary slightly depending on whether you started the import from a dataset, or table **Actions** menu. Excel import presents you with several pages. Follow the links in the descriptions below for detailed information on each page:

- File selection—From this page you can browse to select the file to import.
- Preferences—Use the drop-down menu on this page to specify an existing set of preferences to use for this import.
- Configuration page—The options here determine basic configuration information.
- Table mapping page—This page allows you to map the tables in the dataset to those in the Excel file. You will only see this page when initiating import from a dataset's **Actions** menu.
- Column mapping—This page allows you to map the columns from tables to those in the Excel file.
- Simulation page—From this page you can run a simulation and view results prior to performing the actual import operation.
- Results page—This page shows you the import results.

Preferences page

Preferences allow you to automatically populate fields, and map between a source and target using saved information. This page contains the option shown in the following table:

List	Definition
Preference	Choose a preference to load configuration and mapping information.

Configuration page

This page allows you to modify behavior of the import. Part of the modification can include specifying Java classes to transform, or validate data. This page contains the options shown in the following table:

Attention

To avoid issues during import of a large Excel file, please do not activate the **Download file of invalid data** option.

Property/Field/List	Definition
Import mode	• **Update or insert**: If a record with the same primary key already exists in the target table, it is updated. Otherwise, a new record is inserted. • **Insert only**: Only record creations are allowed. If a record exists in the target table with the same primary key as in the source, an error is returned. **Note** Use of this mode is not recommended when importing a large volume of data. When working with larger amounts of data, you can instead use the **Update or insert**, or **Update only** modes. • **Update only**: Only modifications of existing records are allowed. If a record with the same primary key does not exist in the target table, an error is returned. • **Replace all content**: All existing data in the table will be deleted before importing the data from the file.
First row contains header	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
Force precision as displayed	Specifies the accuracy of numbers when importing Excel. If set to 'Yes': The displayed value of numbers in each cell will be read and imported. If set to 'No': The accuracy value of numbers in each cell will be read and imported. Default value: 'No'. This feature is only applied to the Number format cells (Number is used for general display of numbers). For other format cells, the accuracy values in each cell will be read and imported.
Force import	Specifies that all triggers and constraints are disabled when this option is activated. Otherwise, all triggers and constraints are enabled.
Download file of invalid data	Export all records with errors in their data rows to a downloadable file.
Remove redundant characters contained in the header during matching	Replace all line-breaks and continuous space characters in the Excel header with a single space character during matching.
Use case-sensitive comparison when matching the header	Differentiate between lower-case and upper-case when matching the imported file's column header with the field's label in EBX®.
Ignore the empty or null values	By default, the existing record is not updated with null and empty cell values from the imported file.

Property/Field/List	Definition
Check empty or null primary keys	The add-on verifies that all primary keys are mapped and validates the data before importing. If you disable this option, the add-on does not perform this verification.
Extensions	Specifies the Java class that is implemented to transform data during import/export processing.
Validators	Specifies the Java class that is implemented to validate data during import.
Starting position of table content (only visible on the configuration page when importing into a single table)	Choose the row and column number in the Excel file containing data to import.

Table Mapping page

This page displays when importing into a dataset and allows you to map the target tables with the source sheets in the Excel file. Additionally, you can specify where the data begins in the file using the **Choose starting position** options. If all sheets are the same, set a single starting position. Otherwise, you'll need to specify a position for each sheet.

The source sheets and target tables display on the right and left, respectively. The icon between the to lists allows you to use, or ignore a table during import. All used tables will require a column mapping.

Below the mapping configuration information, the **Save preference** group allows you to save the mappings as a reusable preference. Just tic the **Create new preference called** radio button and enter a name in the field.

Column mapping page

This page allows you to map source columns in the Excel file with EBX® table columns. When starting this import from:

- A table: Use the drop-down list next to each table column to specify which source gets mapped to which target. Additionally, at the top of the screen, you can expand **Data file sample** to see a preview of how the current mapping would import.
- A dataset: In addition to the options described in the previous bullet, a list of tabs displays that allows you to select and edit mappings for each table.

Simulation page

This page allows you to use the current configuration and mapping settings to simulate import. The simulation results can include different levels (error, warning, and information) of technical, validation, and business messages. Use the result information to make any needed changes before performing the actual import.

The table below describes the page's main components:

Options	Definition
Validate imported records only	When selected, the validation report only runs on the records being imported. If unselected, the report also includes all records in the target table.
Skip validation report	When checked, the simulation result does not include the detailed validation report.
Simulation stopped	Select conditions that determine when to stop the simulation.
Simulation result	Displays simulation results on a table-by-table basis. Each tab corresponds to a simulation for a table.
Import option	Choose whether to stop the import process when there are validation errors.
Import type	Either specify that the add-on simulates importing by importing all sheets at once, or sequentially.

CHAPTER 38

SQL import options

This chapter contains the following topics:

38.1 Overview

The sections below describe the pages and options you will be presented with when importing from an external database:

- Configuration page—The options here determine basic configuration information including selection of the table, or view to import from.
- Column mapping—This page allows you to map the columns from tables to those in the database.
- Results page—This page shows you the import results.

Configuration page

This screen allows you to specify the source table or view in the external database on which the add-on reads data from and choose the Java class used to map data. All the tables in the connected external database display in the SQL table view property's drop-down list. The options include:

Options	Definition
Import mode	• **Update or insert**: If a record with the same primary key already exists in the target table, it is updated. Otherwise, a new record is inserted. • **Replace all content**: All existing data in the table will be deleted before importing the data.
Force import	When activated, this option disables all triggers and constraints. Otherwise, all triggers and constraints are enabled.
Ignore the empty or null values	By default, the existing record is not updated with null and empty cell values from the imported data.
Check empty or null primary keys	The add-on verifies that all primary keys are mapped and validates the data before importing. If you disable this option, the add-on does not perform this verification.
SQL table or view	Allows you to choose from external database tables or views based on the data source names which are declared in the configuration.
SQL predicate	Defines the SQL predicate as conditions to filter out data of SQL table or view to import into an EBX® table.
Mapping Java class	Displays all pre-defined SQL java classes used to map columns.

Column mapping page

This page allows you to map source columns in the database table with EBX® table columns. You can select or ignore columns by checking the checkbox before each field name prior to data import.

Results page

This page provides an overview of results from an import. From this page you can exit, or start a new import.

CHAPTER 39

XML import options

This chapter contains the following topics:

39.1 Overview

When importing from an XML file you'll see a configuration screen that presents the following options:

Options	Definition
File name	Name of the file to import.
Import mode	• **Update or insert**: If a record with the same primary key already exists in the target table, it is updated. Otherwise, a new record is inserted. • **Replace all content**: All existing data in the table will be deleted before importing the data.
Use header	You can add an EBX® Data Exchange Add-on header that contains metadata such as Name, Type, Application Version and export time. The add-on reuses this 'header' XML tag at import time.
Force import	Specifies that all triggers and constraints are disabled when this option is activated. Otherwise, all triggers and constraints are enabled.
Ignore the empty or null values	By default, the existing record is not updated with null and empty cell values from the imported data.
Check empty or null primary keys	The add-on verifies that all primary keys are mapped and validates the data before importing. If you disable this option, the add-on does not perform this verification.

Export Options

CHAPTER 40

CSV export options

This chapter contains the following topics:

40.1 Overview

This section describes the fields and available options when exporting to a CSV file.

After running the export service, you are presented with configuration and mapping pages. See the sections below for more information.

Configuration page

This page allows you to modify behavior of the export. Part of the modification can include specifying Java classes to transform, or validate data. This page contains the options shown in the following table:

Property/Field/List	Definition
File name	Name of the exported file.
Preference	All table columns will be exported based on configuration and mapping defined by the selected export preference.
First row contains header	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
Disable write access lock	Specifies the write access lock on the data space while exporting. If 'No': The dataspace is locked and the export is executed inside the read only procedure. If 'Yes': The export is executed outside a procedure and users have write access on the data space while exporting. Default value: 'No'.
File encoding	Specifies the character encoding to use. The default is UTF-8.
Separator	Specifies the separator character used to separate fields. The default is ';'.
List separator	Specifies the separator to use for values lists.
Delimiter	Specifies the character used to enclose a field value. The exported file will include this delimiter character before and after each field value. By default, CSV files do not use a delimiter. The add-on imports correctly from exported files where data values include the defined separator. These files also display correctly in spreadsheet programs. However, if you define a delimiter and view the file in a spreadsheet program, the display might not reflect the actual data structure. If you select 'Only string', the delimiter is only used for String data types.
Date format	Specifies the format of date values in the exported file.
Date/time format	Specifies the format of date/time values in the exported file.
Use language	If this option is activated, the language and corresponding number format policy are applied.
Language	Specify the language and the corresponding number format policy used in the exported file.
Export related data	Specifies whether the export includes related data. The add-on exports a ZIP file containing the selected record(s) and all related data. Each file in the archive contains data from a related table. When you select: • **From referenced tables**: The export includes data referenced by a foreign key from the exported record(s). Referenced data may include data from tables outside of the current data model. • **From tables that reference the selected record**: The export includes data from tables with foreign keys to the exported record. This option is only available when exporting a single record.
Primary key	The exported file will contain an information column with the label of the primary key.

Property/Field/List	Definition
Foreign key	The exported file will contain an information column with the label of the foreign key.
Export enumerations	The exported file will contain a dedicated column for the labels of the static enumerations.
Include computed values	Specifies whether computed values must be included in the export.
Extensions	Specifies the Java class that is implemented to transform data during import/export processing.

Column mapping page

This page allows you to map source table columns in the exported CSV file. Use the drop-down list next to each table column to specify which source gets mapped to which target. Below the mapping configuration information, the **Save preference** group allows you to save the mappings as a reusable preference. Just tic the **Create new preference called** radio button and enter a name in the field.

CHAPTER 41

Excel export options

This chapter contains the following topics:

41.1 Overview

This section describes the fields and available options when exporting to an Excel file. Note that options vary slightly depending on whether you started the export from a dataset, or table **Actions** menu. Exporting from a dataset allows you to export one, or more tables—each table added to a sheet. Whereas, exporting from a table only creates a single sheet in the exported file.

Attention

To avoid issues during export of a large Excel file, please use only the default options and do not activate the **Include validation messages**, **Export related data**, or **Export permalink for the primary or foreign key** options.

After running the export service, you are presented with several pages. Follow the links in the descriptions below for detailed information on each page:

- Configuration page—The options here determine basic configuration information.
- Table mapping page—This page allows you to map the tables in the dataset to those in the Excel file. You will only see this page when initiating import from a dataset's **Actions** menu.
- Column mapping—This page allows you to map the columns from tables to those in the Excel file.
- Results page—This page shows you the export results.

Dataset export Configuration page

This page allows you to modify behavior of the export. Part of the modification can include specifying Java classes to transform, or validate data. This page contains the options shown in the following table:

Property/Field/List	Definition
File name	Name of the exported file.
Preference	All table columns will be exported based on configuration and mapping defined by the selected export preference.
Save as type	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
First row contains header	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
Disable write access lock	Specifies the write access lock on the dataspace while exporting. If 'No': The dataspace is locked and the export is executed inside the read only procedure. If 'Yes': The export is executed outside a procedure and users have write access on the data space while exporting. Default value: 'No'. Note that this property is only visible when enabled through dataspace preferences by an administrator.
Export the ignored field as a blank column	Specifies whether the ignored field will be exported as a blank column. This option is only available on Excel export.
Include validation messages	Specifies options to include different severity levels of validation messages in the export.
Primary key	When set to 'Export label', the exported file will contain an information column with the label of the primary key. When set to 'Permalink', the exported file will contain a permalink for the primary key.
Foreign key	When set to 'Export label', the exported file will contain an information column with the label of the foreign key. When set to 'Permalink', the exported file will contain a permalink for the foreign key.
Include computed values	Specifies whether computed values must be included in the export.
Include the reference sheet	You can add a reference sheet called 'Reference table mapping' at the end of the Excel file. This sheet contains metadata such as the sheet name, the table label and the table path used to detect the mapping between sheets and tables upon import.
Choose the tables to export	The selected tables will be exported into multiple sheets in the Excel file.
Extensions	Specifies the Java class that is implemented to transform data during import/export processing.

Table export Configuration page

This page allows you to modify behavior of the export. Part of the modification can include specifying Java classes to transform, or validate data. This page contains the options shown in the following table:

Property/Field/List	Definition
File name	Name of the exported file.
Preference	All table columns will be exported based on configuration and mapping defined by the selected export preference.
Save as type	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
First row contains header	Use this option to specify whether the first row of every column in the file being imported is a label (Header), or contains data to import (No header).
Disable write access lock	Specifies the write access lock on the data space while exporting. If 'No': The dataspace is locked and the export is executed inside the read only procedure. If 'Yes': The export is executed outside a procedure and users have write access on the data space while exporting. Default value: 'No'.
Export the ignored field as a blank column	Specifies whether the ignored field will be exported as a blank column. This option is only available on Excel export.
Include validation messages	Specify options to include different severity levels of validation messages in the export. None: The exported file will not include any validation messages. Info: The exported file will include any information messages. Warning: The exported file will include any warning messages. Error:The exported file will include any error messages.
Export related data	Specifies whether the export includes related data. The add-on exports a single Excel file. The first sheet in the file contains the source data (where the service was run from). Each additional sheet contains related data from individual tables. When you select: • **From referenced tables**: The export includes data referenced by a foreign key from the exported record(s). Referenced data may include data from tables outside of the current data model. • **From tables that reference the selected record**: The export includes data from tables with foreign keys to the exported record. This option is only available when exporting a single record.
Primary key	When set to 'Export label', the exported file will contain an information column with the label of the primary key. When set to 'Permalink', the exported file will contain a permalink for the primary key.
Foreign key	When set to 'Export label', the exported file will contain an information column with the label of the foreign key. When set to 'Permalink', the exported file will contain a permalink for the foreign key.
Export enumerations	This option allows you to determine behavior when exporting enumerations. The **Export label** option inserts an additional column in the exported file. This column contains the enumeration value labels. The **Export static enumerations** option exports the normal column, but modifies it to include a drop-down list of enumeration values defined in the data model. If you select both options, both columns will contain a drop-down list of available values.
Include computed values	Specifies whether computed values must be included in the export.
Extensions	Specifies the Java class that is implemented to transform data during import/export processing.

Property/Field/List	Definition
Table template tabs	
Title template	These settings determine the title's starting position and text style in the exported Excel file.
Subtitle template	These settings determine whether to export with a subtitile and its starting position and text style in the exported Excel file. By default, no subtitle gets exported.
Table border	This setting determines whether to use a border for data exported to the Excel file. By default, a border is not included.
Column title template	These settings determine the header's starting position and text style in the exported Excel file.
Data template	These settings determine the starting position and text style of data exported to the Excel file.

Table Mapping page

This page displays when exporting from a dataset and allows you to map the source tables with the target sheets in the Excel file.

The source tables and target sheets display on the left and right, respectively. The icon between the lists allows you to use, or ignore a table during export. All used tables will require a column mapping.

Below the mapping configuration information, the **Save preference** group allows you to save the mappings as a reusable preference. Just tic the **Create new preference called** radio button and enter a name in the field.

Column mapping page

This page allows you to map source table columns with columns in the exported Excel file. When starting this export from:

- A table: Use the drop-down list next to each table column to specify which source gets mapped to which target. Below the mapping configuration information, the **Save preference** group allows you to save the mappings as a reusable preference. Just tic the **Create new preference called** radio button and enter a name in the field.
- A dataset: In addition to the options described in the previous bullet, a list of tabs displays that allows you to select and edit mappings for each table.

CHAPTER 42

SQL export options

This chapter contains the following topics:

42.1 Overview

The sections below describe the pages and options you will be presented with when importing from an external database:

- Configuration page—The options here determine basic configuration information including selection of the table, or view to import from.
- Column mapping—This page allows you to map the columns from tables to those in the database.
- Results page—This page shows you the import results.

Simulation page

This screen allows you to select an external database location to export to and select a Java class used to map columns. The options include:

Options	Definition
Import mode	• **Update or insert**: If a record with the same primary key already exists in the target table, it is updated. Otherwise, a new record is inserted. • **Replace all content**: All existing data in the table will be deleted before importing the data.
SQL table or view	Allows you to choose from external database tables or views based on the data source names which are declared in the configuration.
Mapping Java class	Displays all pre-defined SQL java classes used to map columns.

Results page

This page provides an overview of results from an import. From this page you can exit, or start a new import.

CHAPTER 43

XML export options

This chapter contains the following topics:

43.1 Overview

The sections below describe the pages and options you will be presented with when importing from an external database:

- Configuration page—The options here determine basic configuration information including selection of the table, or view to import from.
- Table selection page—This page only displays when running the export service from a dataset's **Actions** menu and allows you to select a table to export.
- Results page—This page shows you the import results.

Configuration page

This screen allows you to select an external database location to export to and select a Java class used to map columns. The options include:

Options	Definition
Select a target application	With 'Default XML format' selected the export uses default configuration options. When additional target applications have been configured, they can be selected here.
Select a version to include in the exported file	For information only. If you use the 'use header' option, this information is inserted in the exported file as the 'appversion' XML tag within the header.
Use header	You can add a EBX® Data Exchange Add-on header that contains metadata such as Name, Type, Application Version and export time. EBX® Data Exchange Add-on reuses this 'header' XML tag at import time.
Include computed values	Specifies whether computed values must be included in the export.
Is indented	Specifies whether the file should be indented to improve readability.
Omit XML comment	Specifies whether the generated XML comment that describes the location of data and the export date should be omitted from the file.
Disable write access lock	Specifies the write access lock on the dataspace while exporting. If 'No': The dataspace is locked and the export is executed inside the read only procedure. If 'Yes': The export is executed outside a procedure and users have write access on the dataspace while exporting. Default value: 'No'.

CHAPTER 44

Transfer options

Property	Definition
Transfer mode	Update or insert: If a record with the same primary key already exists in the target table, it is updated. Otherwise, a new record is inserted. Replace all content: All existing data in the table will be deleted before importing the data.
Force transfer	Specifies that all triggers and constraints are disabled when this option is activated. Otherwise, all triggers and constraints are enabled.
Include computed values	Specifies whether computed values must be included in the transfer.
Ignore the empty or null values	By default, the existing record is not updated with null and empty values from the source table.
Check empty or null primary keys	The add-on verifies that all primary keys are mapped and validates the data before transferring. If you disable this option, the add-on does not perform this verification.
Stop and rollback on error	Enabling this option causes the add-on to stop the data transfer in progress when errors occur and rollback all data. If disabled, the add-on skips transfers where errors occur and executes remaining transfers. Take as an example a transfer from table A to tables B, C, and D. If errors occur only during transfer to B, the add-on rolls back data for B, but completes transfer to C and D.
Transfer data	Based on this data model: Specify the datasets that have the same data model as the current dataset. Based on a different data model (mapping required): Specify the datasets that don't have the same data model as the current dataset.
Target dataspace	List of relevant target dataspaces is obtained by the system after selecting the 'Transfer data' option.
Target dataset	List of relevant target datasets are obtained by the system after selecting the 'Transfer data' option.

CHAPTER 45

Add-on Services

This chapter contains the following topics:

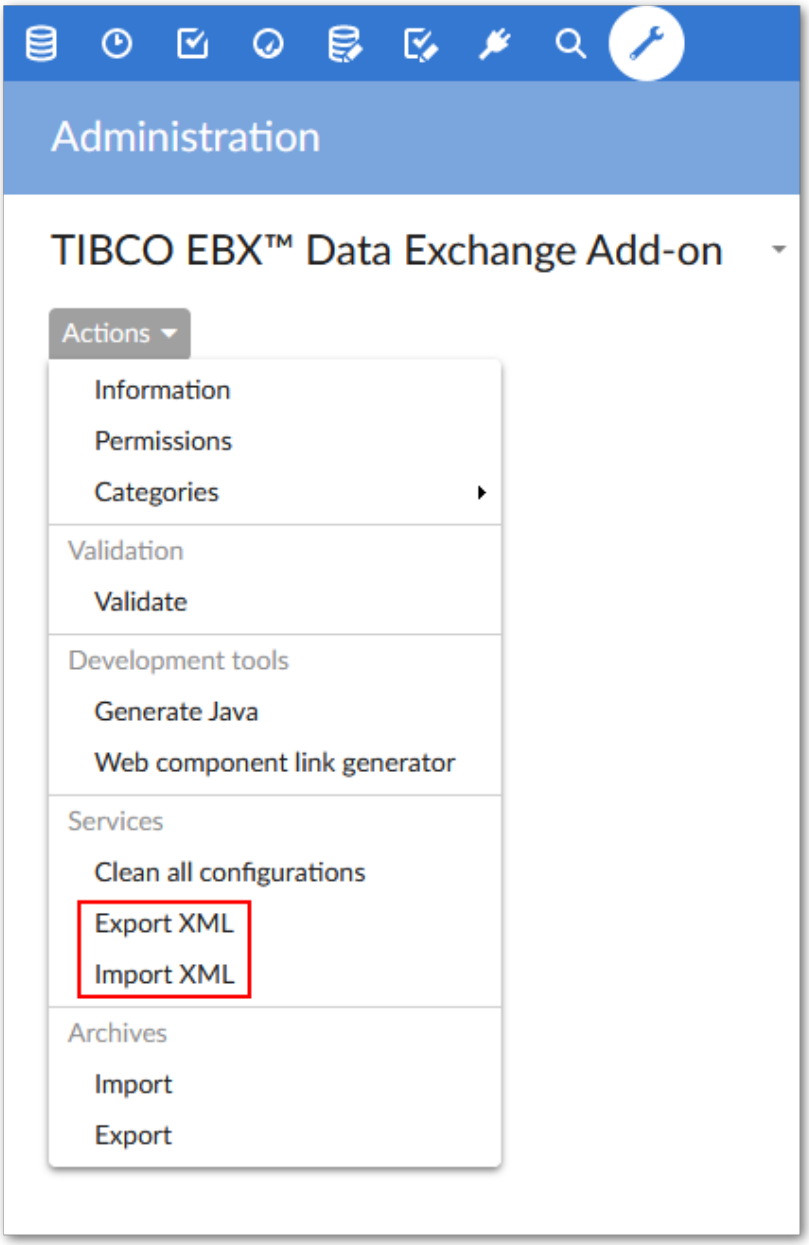
45.1 Overview

This section describes EBX® Data Exchange Add-on services.

45.2 Importing and Exporting EBX® Data Exchange Add-on configuration

You can import and export all configuration options contained in a EBX® Data Exchange Add-on dataset from and to an XML file. As shown below, these services are accessed from the EBX® Data

Exchange Add-on dataset **Actions** menu. For more detailed instructions on migrating configuration settings, see Migration overview [p 119].

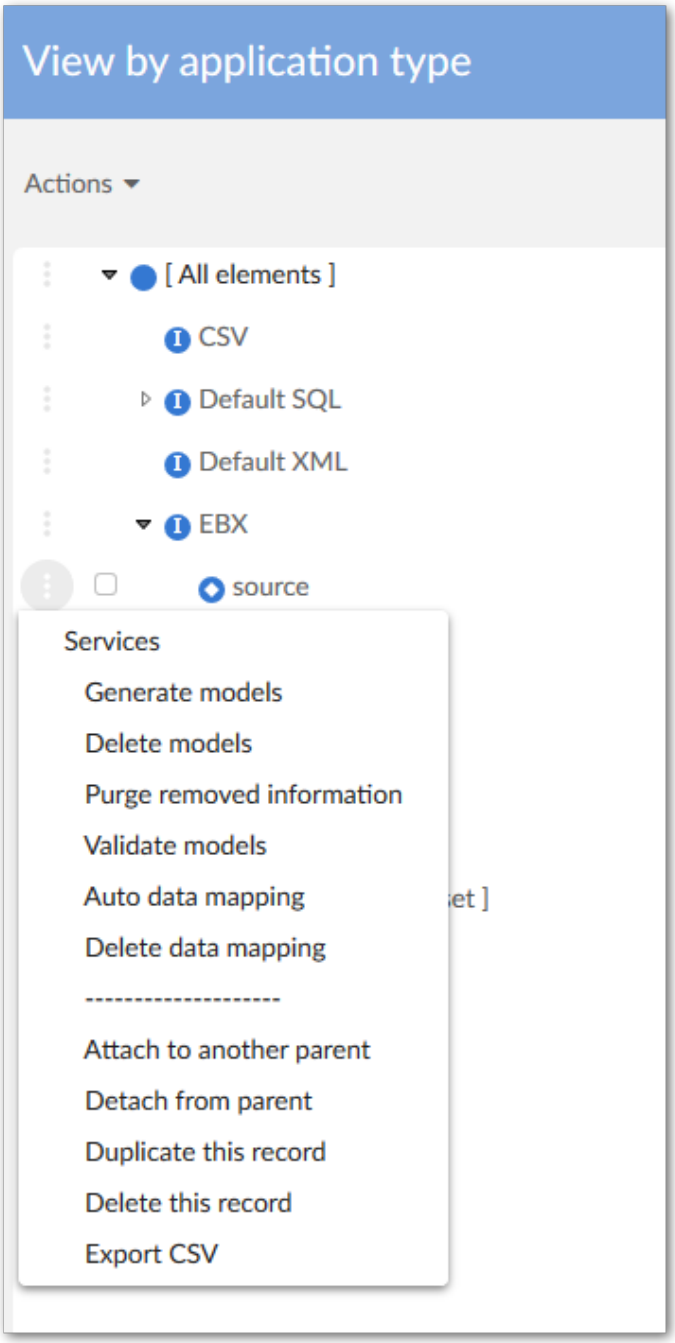


The following points highlight features of the services.:

- **Export XML**: All table records (including hidden tables) will be exported to an XML file. You can specify whether the file is indented to improve readability and whether to omit XML comments that describe data location and export date. After you set these options and click **Export**, the add-on prompts you for a save location.
- **Import XML**: All triggers and constraints are disabled during the import process. You can validate after import. If an error occurs during import, the process stops and an error message is displayed. On the import screen you can select the import mode. **Update or insert**—If a record in a target table has the same primary key as an imported record, it is updated. Otherwise, new records are inserted. **Replace all content**—All existing table data is deleted prior to import.

45.3 **Application type services**

In order to manage user-defined data mapping configurations, a portfolio of services is available depending on the application type. These services are located on the **Application by type** table in the **TIBCO EBX® Data Exchange Add-on** dataspace under the EBX® **Administration** tab.



The creation of user-defined data mapping configurations requires IT skills. Based on these configurations, end-users can export, import and transfer the data easily, all the technical aspects of data mapping configuration are hidden from them.

The table below gives a short description of the services available to manage the user-defined data mapping configurations.

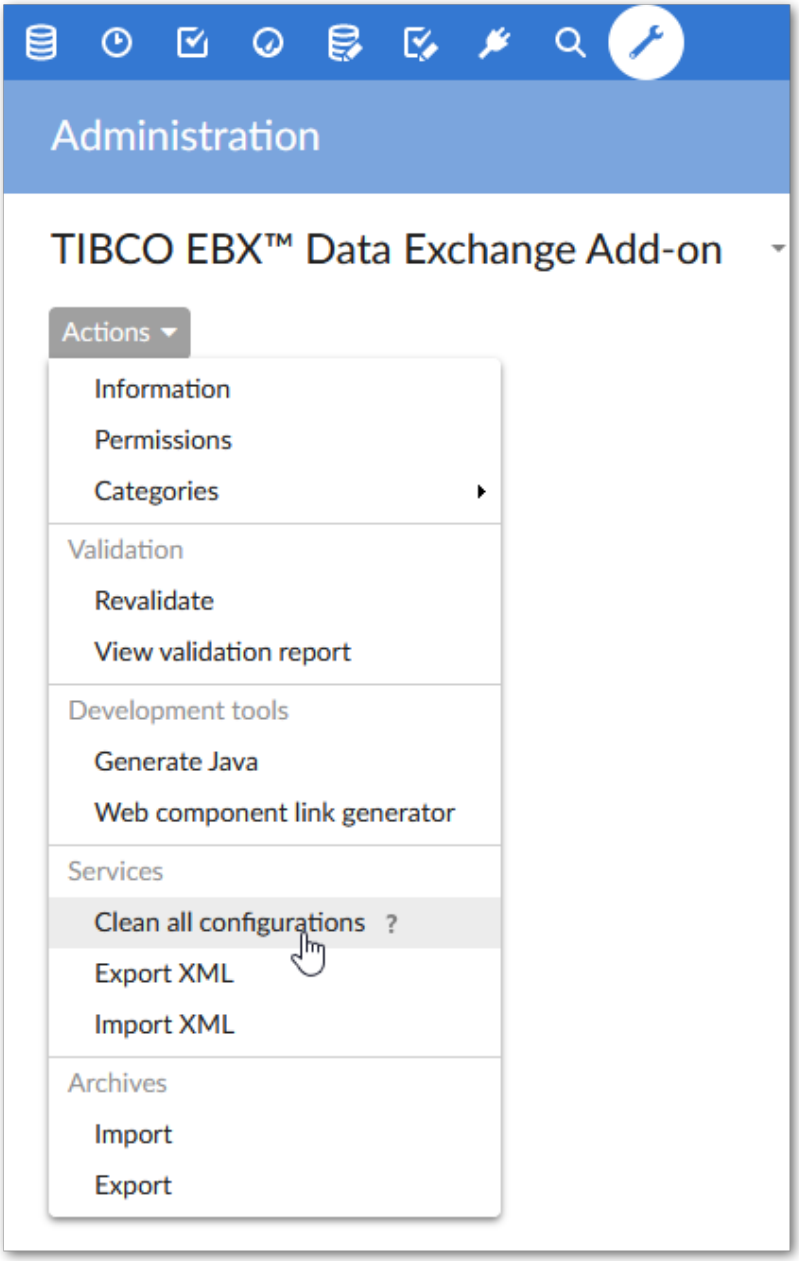
Services applied to an Application of type	Default XML	XML	CSV/Excel	EBX
Delete data mapping	Physical deletion of all Tables, Fields and related data mapping (Tables: Version, Interface Application, Table mapping, Field mapping, Table mapping preference, Field mapping preference, Field mapping transformation and Path) for the Application. For an EBX type application, the Object class, Property and related tables (Object Class by Application, Property by Object Class) are also removed.			
Generate models	N/A	Tables and Fields are declared as XML Tags. The generation is based on a sample XML file that must be provided as input data.	Tables and Fields are declared as column names. The generation is based on a sample CSV/Excel file that must be provided as input data.	Tables and Fields are declared as an EBX® path. Object classes and Properties can also be generated to get the semantic data model. The generation is based on a dataspace - dataset corresponding to the Application.
Delete models	N/A	N/A		Logical deletion of Object class and Property type items-if not used by another application. Logical deletion of all application Tables and Fields. The logical deletion is registered by using the property 'Is removed'
Purge removed information	N/A	N/A		Physical deletion of the items that are tagged 'Is removed' = 'Yes' by the 'Delete models' service. All data mappings that refer to the deleted items are also physically removed.
Validate models	N/A	N/A		Checks if the configuration is still updated with the EBX®' data model. All unaligned Tables and Fields are then modified by changing the 'Is removed' property into the value 'Yes'

Services applied to an Application of type	Default XML	XML	CSV/Excel	EBX
Auto data mapping	N/A	N/A		Automatically configures the data mapping between two EBX® applications sharing the same Object class and property items.

Table 7: Services on 'Application by type' table

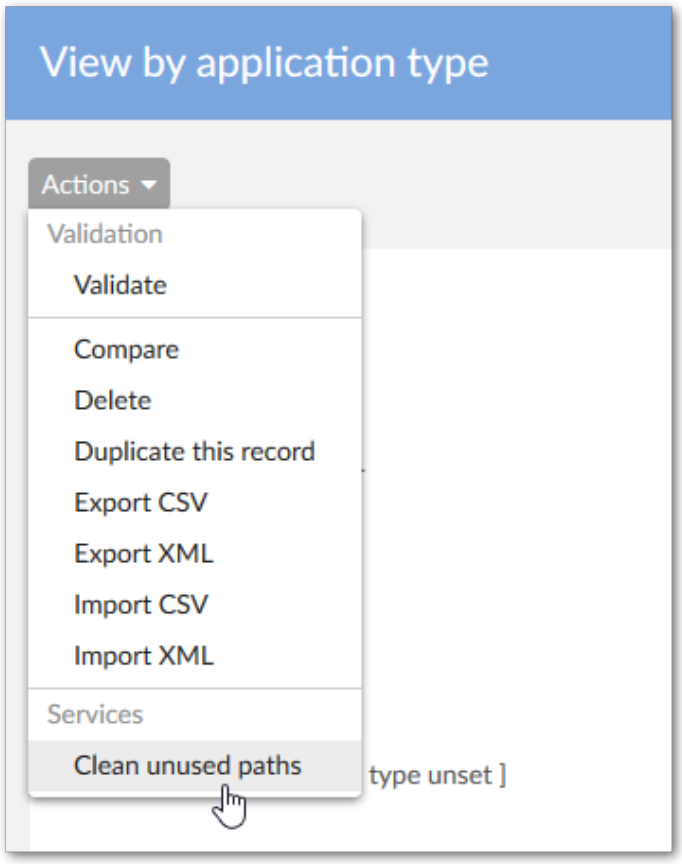
45.4 **Service: Clean all configuration**

This service is located at the **TIBCO EBX® Data Exchange Add-on** dataset level and allows you to remove all data mapping configurations and get an empty EBX® Data Exchange Add-on repository.



45.5 **Service: Clean unused paths**

This service is located at the **Path** table level and allows you to remove all unused paths that have been declared in the data mapping repository.



Dynamic Data Modeling

User Guide

CHAPTER **46**

Overview

Dynamic Data Modeling is part of the TIBCO EBX® Data Exchange Add-on and allows you to generate TIBCO EBX® compliant data models from definitions in XML, Excel and DDL formats.

Special notation key:	
✔	Important recommendation for the use of the feature
✘	This feature is not yet available in the current release

CHAPTER 47

User operations

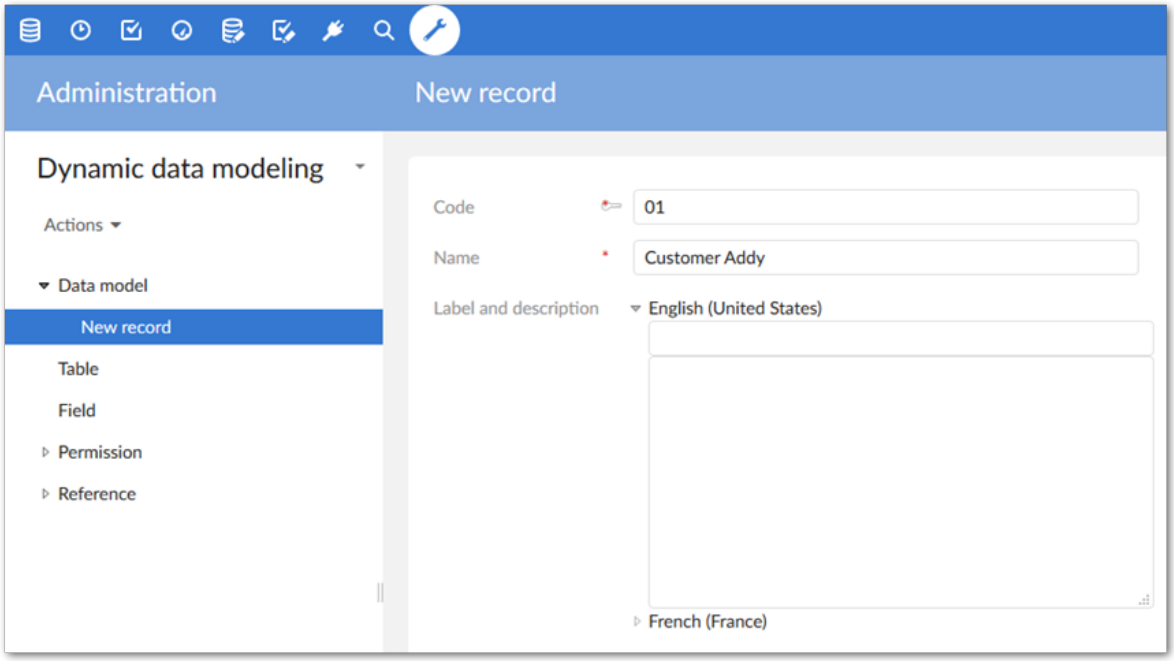
The **Dynamic data modeling** feature allows you to generate and export an XSD file containing a data model by importing its structure from a supported file type (XML, Excel and DDL). This section describes the operations used to generate this file. All user operations are available from the **Dynamic data modeling** dataset under the **Administration** tab.

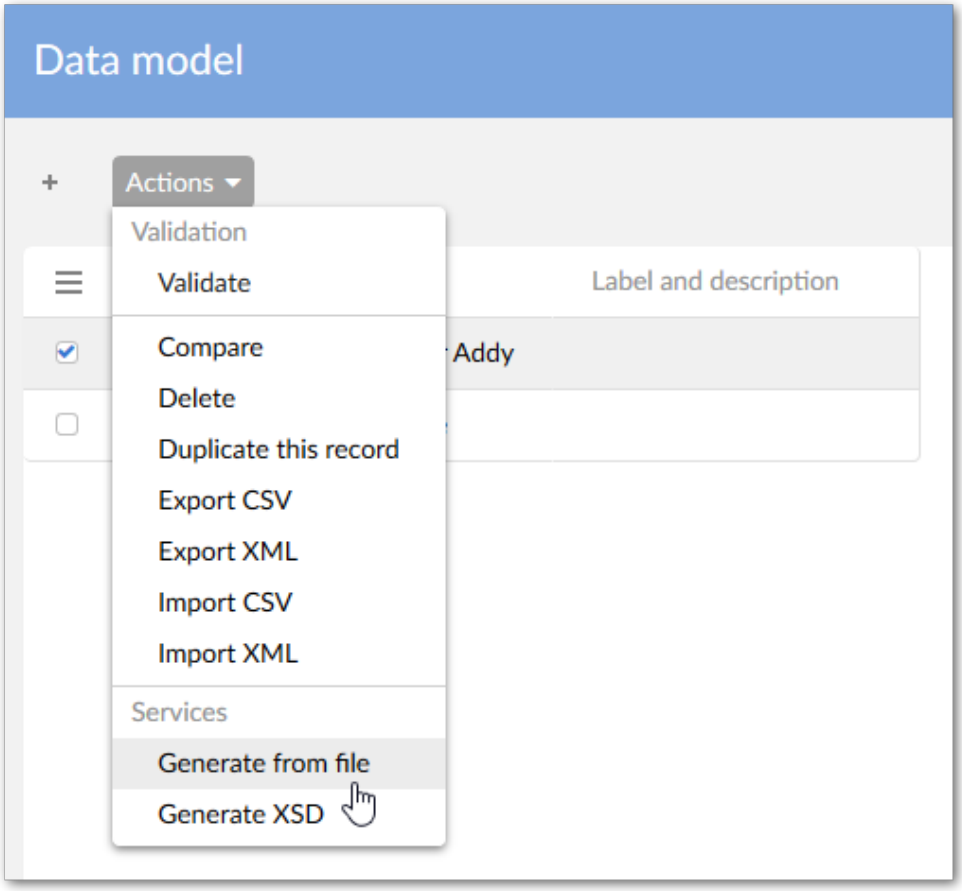
You can refer to the second part of this user guide for a full description of all available options.

This chapter contains the following topics:

47.1 Using the Generate from file service to import

Before importing a file that defines a data model's structure, you must create a record in the **Data model** table that allows you to provide a name and description for a data model configuration. From this record, you can specify a file to import using the **Generate from file** service, as shown below.





The data format of the imported file is automatically identified by the service. The following formats are supported and discussed in further detail in the following sections: Excel, XML and DDL (SQL).

> **Note**
>
> If you execute the **Generate from file** service on an existing **Data model** record, all your previous descriptions are erased.

Generating a data model from Excel

The first four cells in the first row of the spreadsheet allow the service to automatically detect the worksheet tab's layout. Data contained in an Excel file can be arranged in one of the following two ways:

- Either a declaration of the data model structure.
- Or a snapshot of actual data values—from which the data model structure is interpreted.

All generated table and field names are normalized by replacing special characters using the underscore ('_') character.

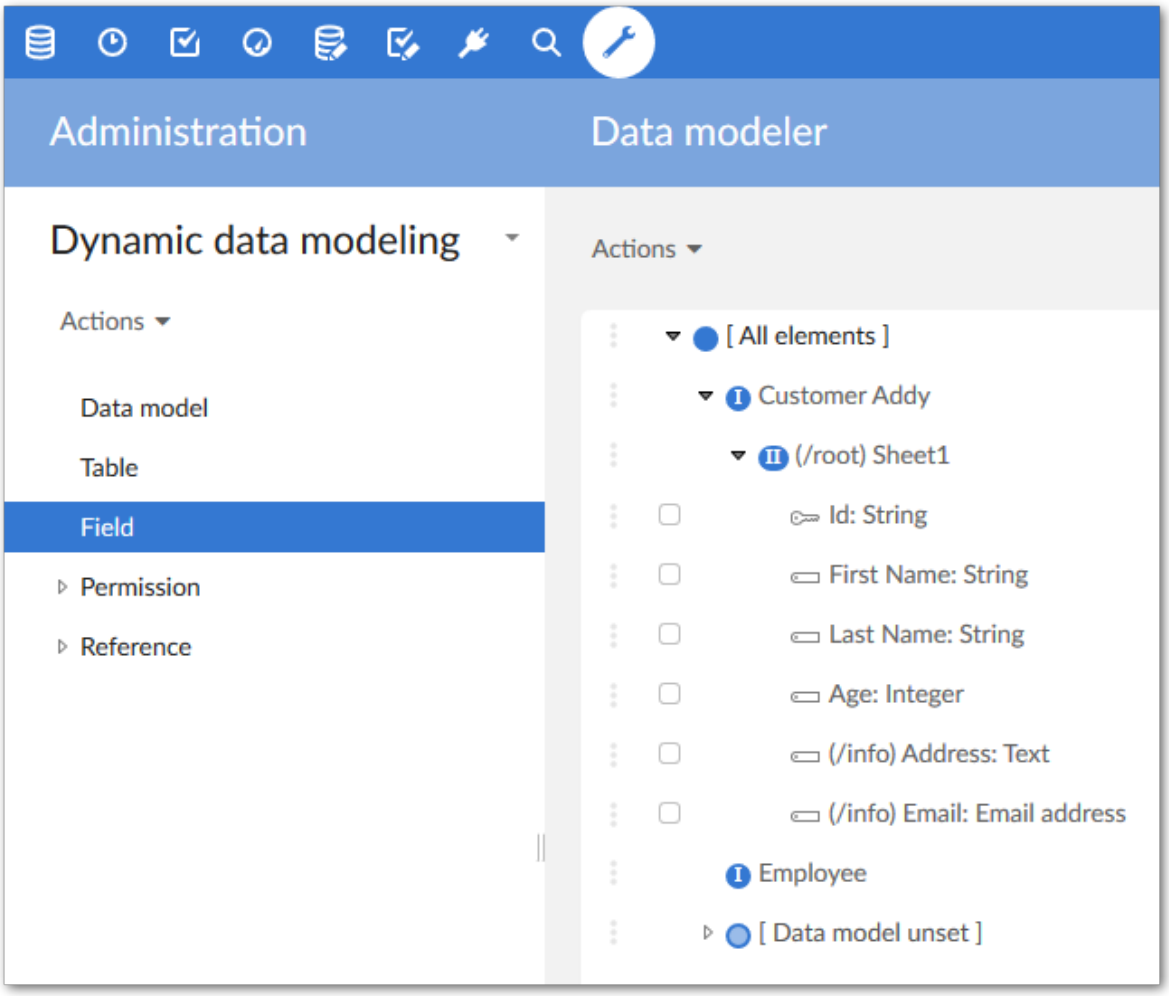
Using an Excel layout containing a data model's structure

An Excel input file is considered as a data model structure layout when the first four cells of the first row contain exactly properties: **Data type**, **Field name**, **Is primary key**, and **Group path**. You can change the order of these properties in the input file. Once the tab's layout is determined, **Data modeler** execute generating data model based on this structure. Each worksheet tab in the file corresponds to

a table. Thus, the tab's name provides the table name and each row defines a data model field with properties as declared in four columns. The list of possible **Datatype** values is defined in the **Data type mapping** table. If you leave **Data type** blank, it defaults to `String`. The default value of Group path is `/root`.

	A	B	C	D
1	Data type	Field name	Is primary key	Group path
2	String	Id	Yes	
3		First Name	No	
4		Last Name	No	
5	Integer	Age	No	
6	Text	Address	No	/info
7	Email address	Email	No	/info

The figure below shows the result of the generation based on the previous spreadsheet example:



Using an Excel layout containing data values

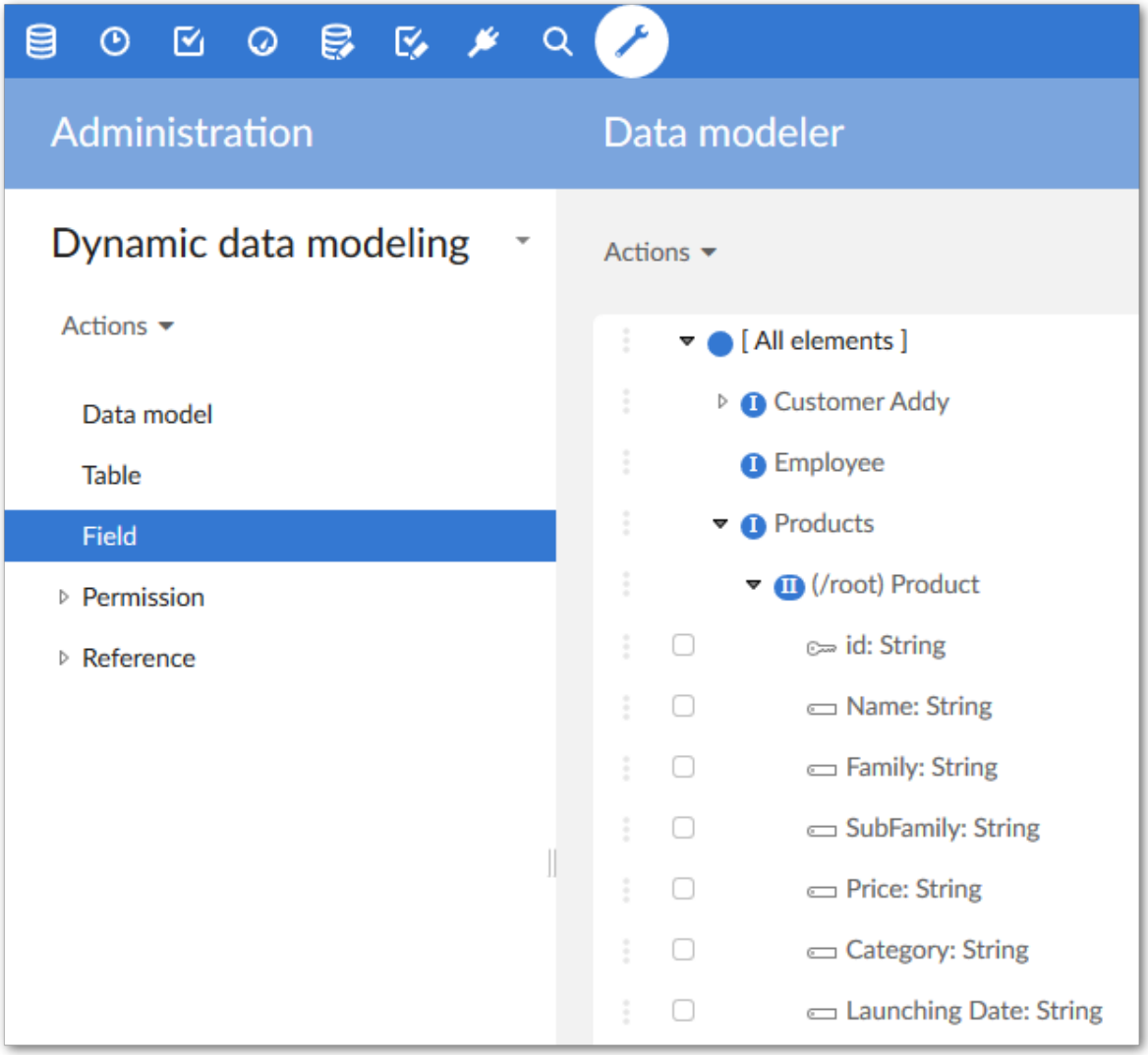
When a spreadsheet contains a set of actual data values—not just a data model outline—the add-on can ascertain data model structure. Each worksheet tab of the spreadsheet corresponds to one table

to generate. The name of the worksheet tab becomes the table name in the data model. The first row in the spreadsheet gives the names of every field (one field per column). Systematically, the first cell value is considered the primary key. You cannot have duplicate names in the list of the fields.

All generated fields default to the String data type due to the inability to detect types during generation. Once imported, you can change the data types manually before generating the XSD file by editing the values in the field's **Type** property.

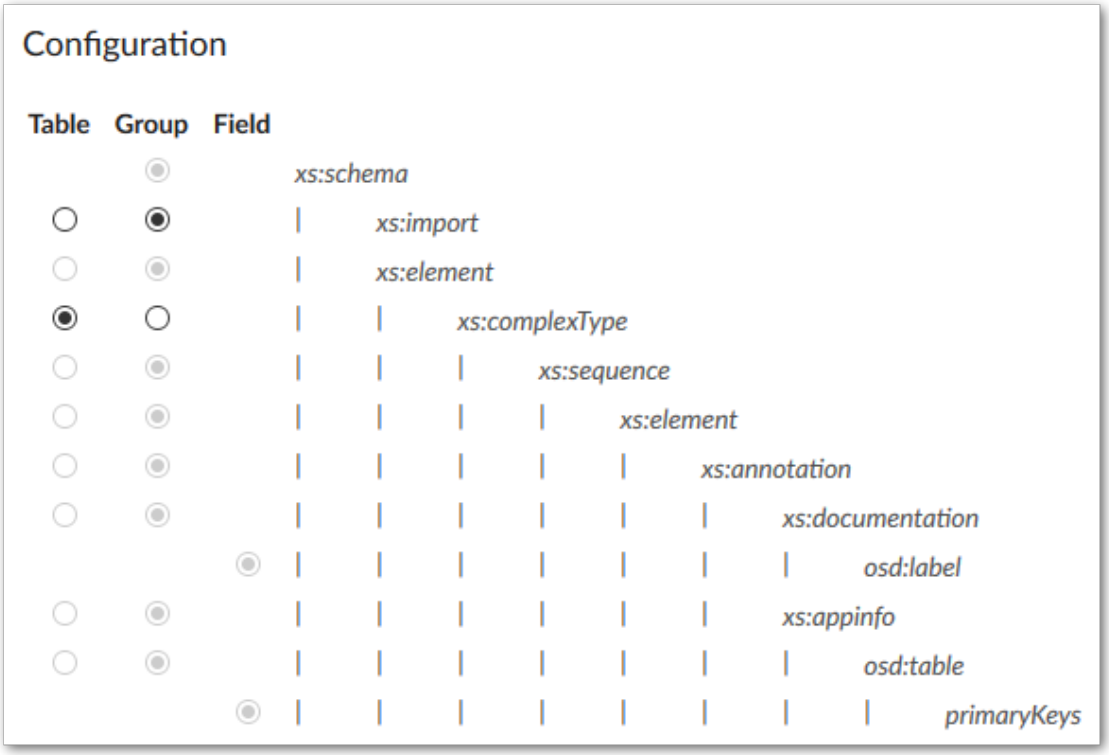
Id	Name	Family	SubFamily	Price	Category	Launching Date
1	Product 1	Shoes	Sports	12.4	2	2007-12-13
2	Product 2	Shoes	Sports	12.4	2	2007-12-13
3	Product 3	Shoes	Sports	12.4	2	2007-12-13

The figure below shows the result of the generation based on the previous spreadsheet example:



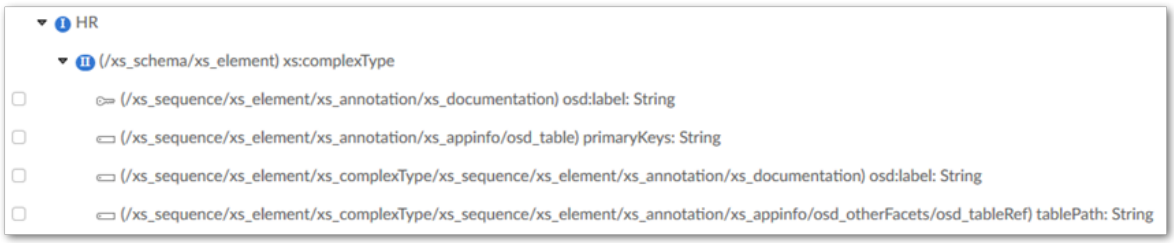
Generating a data model from XML format

You can import an XML file to generate the data model. The **Generate from file** service allows you to decide which elements of the XML file correspond to the tables, groups and fields you want to obtain.



All generated fields default to the **String** data type due to the inability to detect types during generation. Once imported, you can change the data types manually before generating the XSD file by editing the values in the **Field** table's **Type** field.

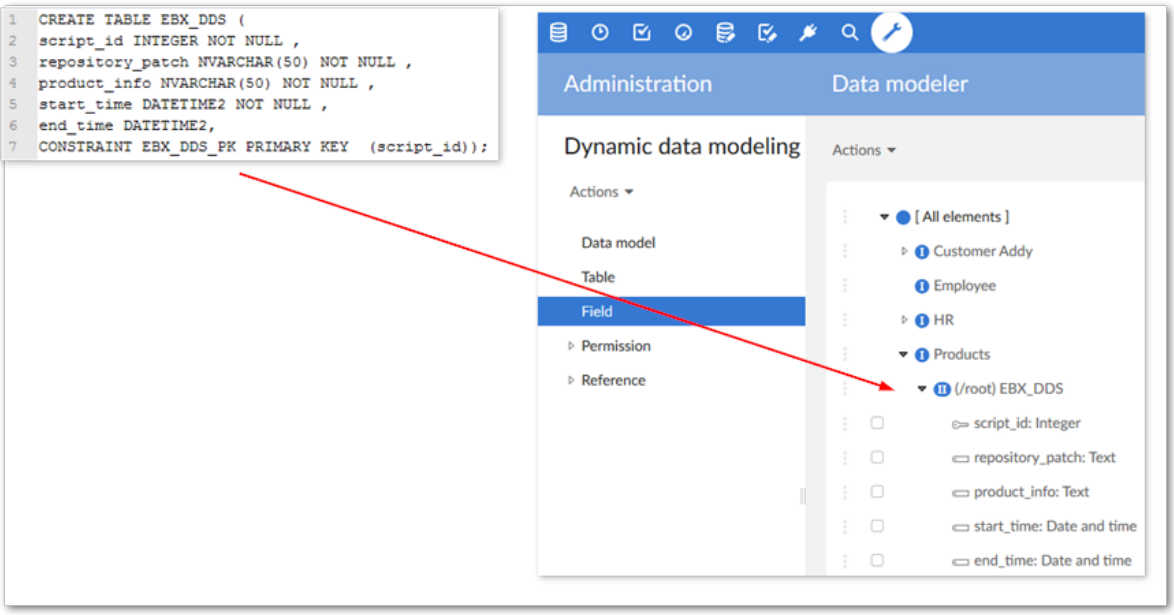
The figure below shows the result of the generation based on the previous XML example:



All fields at the same level as the selected table element are ignored in the generated data model. Since the **Data modeler** does not support to generate a field outside of a table.

Generating a data model from DDL format

You can generate the data model from a DDL description (SQL statements). The following example uses a DDL file that declares a table named **EBX_DDS**.

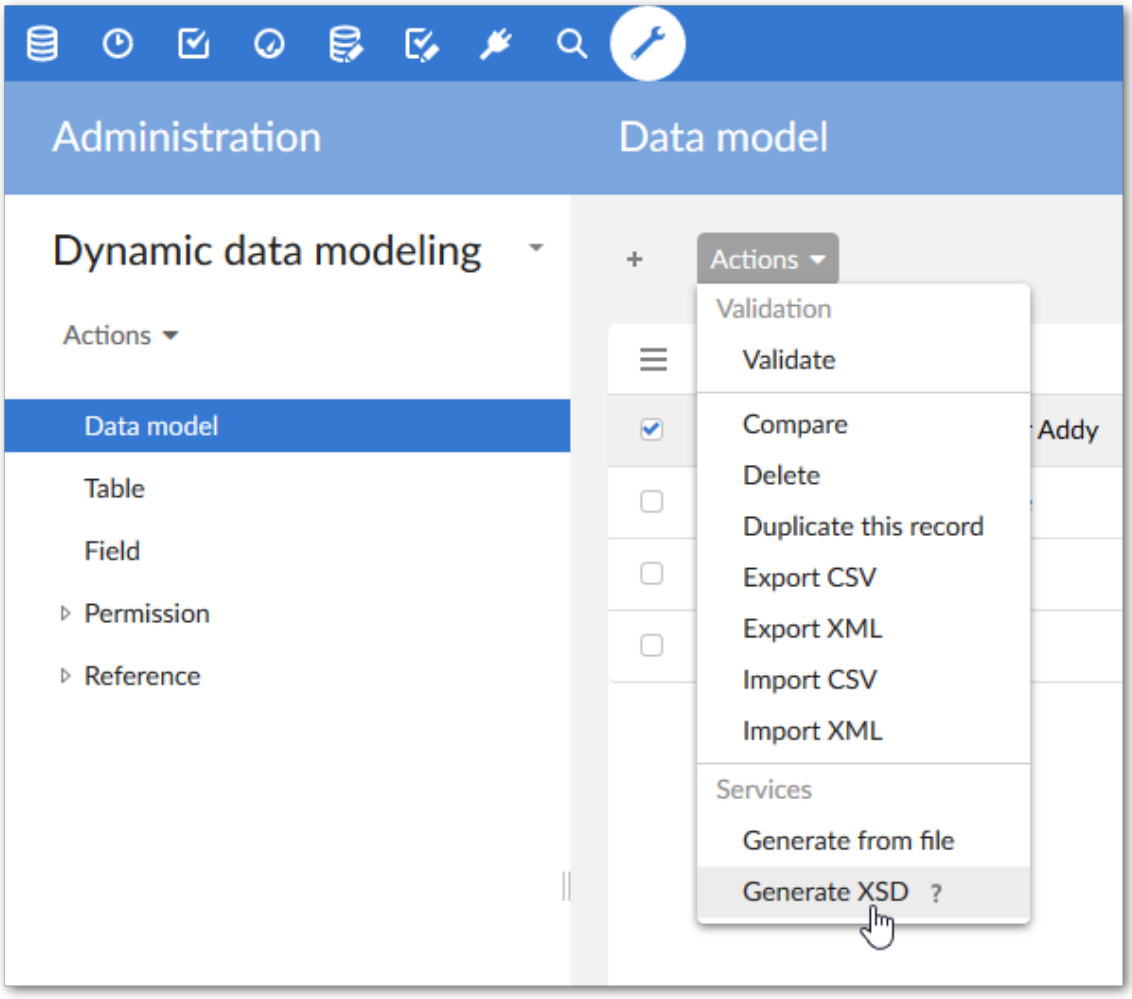


Please refer to appendixes A and B to get the list of supported SQL statements and Databases.

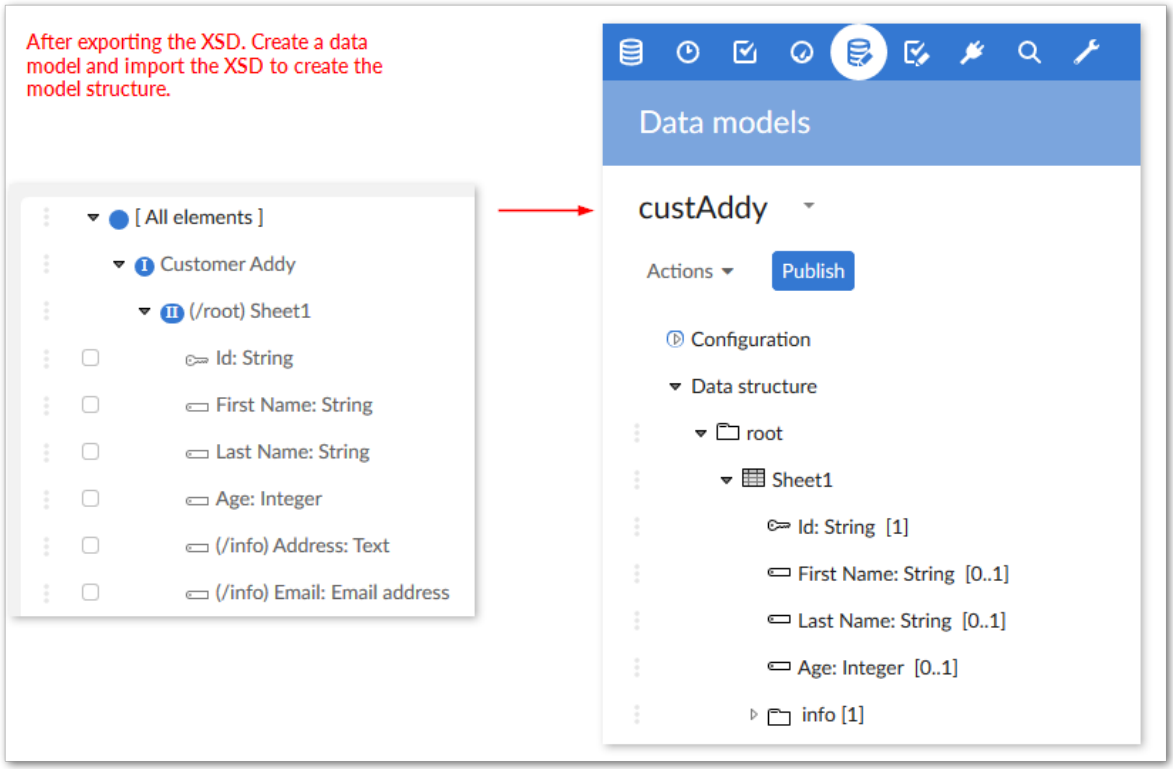
47.2 Creating an XSD file with the 'Generate XSD' service

After importing a data model's structure, you can execute the **Generate XSD** service to obtain the XSD file compliant with EBX®.

This service is located on the **Data model** table as highlighted below.



Here is an example of an XSD file generated from a data model. This XSD file is imported into EBX® to obtain the actual data model.



You can author the data model in the **DMA (Data modeler Assistant)** to adapt it depending on your needs. In case you re-generate the corresponding XSD all these adaptations will be lost.

CHAPTER 48

Configuring Dynamic Data Modeling

This chapter contains the following topics:

48.1 Global view

The **Dynamic data modeling** dataset is located under the EBX® **Administration** tab.

The **Data model** table contains the data models that have been declared. A data model is described through the **Table** and **Field** tables.

The **Permission** domain contains the permissions on use of the **Dynamic data modeling** per user-profile.

The **Reference** domain contains the list of permitted data types used as input data and their corresponding data type in EBX®.

48.2 Data model

The **Data model** table contains the declaration of the data models that are available for generation.

Property	Definition
Code	Any naming convention can be used (without whitespaces).
Name	Name of the current data model.
Label & description	The label and description are used as metadata for the data model description in EBX®.

Table 1: Data model properties

48.3 **Table**

This table contains the table declarations per data model.

Property	Definition
Data model	Name of the data model for which the table is declared.
Name	The name of the table is automatically retrieved from the imported file during generation (Excel, XML and DDL).
Path	Path of the table in the data model.
XSD name	Normalized name from the original table name. This name is used to generate the XSD file and guarantee its compliance with the XSD standard.
Label and description	Label and description used as metadata for the table description.
XSD path	Path of the table used to generate the XML file. You can adapt this value if needed.
Full path	For information only. The 'Full path' shows you the actual and complete path+name of the table that is used to generate the XSD file.

Table 2: Table properties

48.4 **Field**

This table contains the field declarations for data model tables.

Property	Definition
Table	Name of the table for which the field is declared.
Name	The name of the field is automatically retrieved from the input file during generation (Excel, XML and DDL).
Path	Path of the group that contains the field.
XSD name	Normalized name from the original field name. This name is used to generate the XSD file and guarantee its compliance with the XSD standard.
Type	This field's data type. You can modify the data type to adapt it to your needs before generating the XSD file.
Is primary key	If the field is a primary key you can set this property to 'Yes'. In the generated XSD file the field is then considered as a primary key for EBX®.
Label and description	Text used as metadata for the field description.
XSD path	Path of the group that contains the field also used during XSD file generation.
Full path	For information only. The 'Full path' shows you the actual and complete path+name of the field that is used to generate the XSD file.

Table 3: Field properties

48.5 **Permission - User profile**

By default only the Admin user-profile can access to the **Dynamic data modeling** feature. If you want to grant permission to other user-profiles, you have to declare them in this table.

Property	Definition
User profile	User-profile on which the permission is applied.
Is data modeler available	Determines whether the 'Dynamic Data Modeling' service is available for the current user.

Table 4: User profile properties

48.6 **Reference - Data type mapping**

The **Data type mapping** table provides the list of mapping data types between XML, Excel and DLL files and the corresponding data types for the generated EBX® field.

Property	Definition
Schema type name	Corresponding data type of generated field in EBX®.
Value in excel	Data type of field in excel file.
SQL Data type	Data type of field in DDL file.

Table 5: Data type mapping properties

All source and the corresponding data types of generated EBX® fields are listed in the following table:

EBX® data type	Excel data type	SQL data type
AnyURI	URI	N/A
Boolean	Boolean	Boolean
Date	Date	Date
Date time	Date and time	- SMALLDATETIME - DATETIME - TIMESTAMP - TIMESTAMP WITH TIMEZONE - TIMESTAMP WITHOUT TIMEZONE - TIMESTAMP WITH LOCAL TIMEZONE
Decimal	Decimal	- DECIMAL - NUMERIC - MONEY - BIGINT - NUMBER(PRECISION) - BINARY_FLOAT - BINARY_DOUBLE - LONG - BIGSERIAL - DOUBLE PRECISION - REAL
Email	Email address	N/A
Int	Integer	- INT - SAMLLINT - INTEGER - SMALLSERIAL - SERIAL - NUMBER
Locale	Locale	N/A
Name	XML name	N/A
Password	Password	N/A
String	String	- CHAR - VARCHAR - VARCHAR2 - ROWID

EBX® data type	Excel data type	SQL data type
		- BIT - BIT VARYING - BYTEA - CIDR - CHARACTER - CHARACTER VARYING - INET - TSQUERY - TSVECTOR - UUID
Text	Text	- TEXT - NCHAR - NCHAR2 - NVARCHAR - NVARCHAR2 - NTEXT - XML - UROWID - JSON
Time	Time	- TIME - TIME WITH TIMEZONE - TIME WITHOUT TIMEZONE

Table 6: Data type mapping

CHAPTER 49

Appendix

This chapter contains the following topics:

49.1 SQL Statement

The **Data Modeler** supports SQL statements as listed in the following table:

Statement	Description
CREATE TABLE	Creates a new table.
CONSTRAINT...PRIMARY KEY()	Defines fields as Primary key of a table.
ALTER TABLE... ADD PRIMARYKEY()	Defines fields as Primary key of a table.
ALTER TABLE... ADD CONSTRAINT...PRIMARY KEY()	Defines fields as Primary key of a table.
CREATE TABLE...AS SELECT... FROM...	Creates a new table with fields as selected from one table.

Table 7: Supported SQL statements

49.2 **Database list**

The **Data model** supports exported DLL files from the databases below:

No	Name
1	SQLServer
2	Oracle
3	PostgreSQL

Table 8: Supported Databases

Release Notes

CHAPTER 50

Version 3.0.5

Released: March 2022

This chapter contains the following topics:

50.1 New features

This release contains no new features.

50.2 Changes in Functionality

This release contains no functionality changes.

50.3 Changes to third-party libraries

The jQuery library was upgraded to version 3.6.0.

50.4 Closed Issues

[ADIX-3552] [API] `DataExchangeResult.SpreadsheetImport.getErrorFile()` returns null.

[ADIX-3560] There is an inconsistency in handling multi-valued groups under single value group when exporting Excel.

50.5 Known Issues

This release contains the following known issues:

- Association object and selection node are not supported.
- Data lineage.
- Graphical view of the data mapping configurations.

- **Split of string** transformation function is not supported for a multi-valued string field.
- Data transfer between complex terminal nodes is possible only if they use the same Java Bean.
- You can not export data from multiple complex nodes in EBX® that when there is at least one complex node with multiple occurrences.
- You can not import data from multiple complex nodes in an XML file when there is at least one complex node with multiple occurrences.
- **Split of data** works for terminal nodes only.

SQL export:

- You can only export data from EBX® to the selected columns in the same tables of a view which is joined by multiple tables.
- The add-on doesn't support exporting SQL with the **Replace all content** mode to a view which is joined by multiple tables.
- Checking mandatory fields of a view in SQL database is not supported.
- The export still executes normally and shows an error message from database when no privilege granted.
- Exporting data from multi-valued fields and multi-valued groups is not supported.
- You can not export data from EBX® to a view in PostgreSQL database.
- Exporting data from EBX® to an auto-increment column in SQL Server is not supported because the **IDENTITY_INSERT** property is always set to `OFF`.

SQL import: Checking privileges for a view in PostgreSQL is not supported.

Dynamic data modeling: Foreign key statements in the DDL file are not managed.

CHAPTER 51

All release notes

This chapter contains the following topics:

51.1 Version 3.0.5

Released: March 2022

New features

This release contains no new features.

Changes in Functionality

This release contains no functionality changes.

Changes to third-party libraries

The jQuery library was upgraded to version 3.6.0.

Closed Issues

[ADIX-3552] [API] `DataExchangeResult.SpreadsheetImport.getErrorFile()` returns null.

[ADIX-3560] There is an inconsistency in handling multi-valued groups under single value group when exporting Excel.

Known Issues

This release contains the following known issues:

- Association object and selection node are not supported.
- Data lineage.
- Graphical view of the data mapping configurations.
- **Split of string** transformation function is not supported for a multi-valued string field.
- Data transfer between complex terminal nodes is possible only if they use the same Java Bean.
- You can not export data from multiple complex nodes in EBX® that when there is at least one complex node with multiple occurrences.
- You can not import data from multiple complex nodes in an XML file when there is at least one complex node with multiple occurrences.

- **Split of data** works for terminal nodes only.

SQL export:

- You can only export data from EBX® to the selected columns in the same tables of a view which is joined by multiple tables.
- The add-on doesn't support exporting SQL with the **Replace all content** mode to a view which is joined by multiple tables.
- Checking mandatory fields of a view in SQL database is not supported.
- The export still executes normally and shows an error message from database when no privilege granted.
- Exporting data from multi-valued fields and multi-valued groups is not supported.
- You can not export data from EBX® to a view in PostgreSQL database.
- Exporting data from EBX® to an auto-increment column in SQL Server is not supported because the **IDENTITY_INSERT** property is always set to `OFF`.

SQL import: Checking privileges for a view in PostgreSQL is not supported.

Dynamic data modeling: Foreign key statements in the DDL file are not managed.

51.2 Version 3.0.4

Released: December 2021

New features

This release contains no new features.

Changes in Functionality

This release contains no functionality changes.

Changes to third-party libraries

This release contains no third-party library changes.

Closed Issues

[ADIX-3527] Validate SQL data source inputs.

[ADIX-3528] Validate column header inputs.

Known Issues

This release contains the following known issues:

- Association object and selection node are not supported.
- Data lineage.
- Graphical view of the data mapping configurations.
- **Split of string** transformation function is not supported for a multi-valued string field.
- Data transfer between complex terminal nodes is possible only if they use the same Java Bean.

- You can not export data from multiple complex nodes in EBX® that when there is at least one complex node with multiple occurrences.
- You can not import data from multiple complex nodes in an XML file when there is at least one complex node with multiple occurrences.
- **Split of data** works for terminal nodes only.

SQL export:

- You can only export data from EBX® to the selected columns in the same tables of a view which is joined by multiple tables.
- The add-on doesn't support exporting SQL with the **Replace all content** mode to a view which is joined by multiple tables.
- Checking mandatory fields of a view in SQL database is not supported.
- The export still executes normally and shows an error message from database when no privilege granted.
- Exporting data from multi-valued fields and multi-valued groups is not supported.
- You can not export data from EBX® to a view in PostgreSQL database.
- Exporting data from EBX® to an auto-increment column in SQL Server is not supported because the **IDENTITY_INSERT** property is always set to `OFF`.

SQL import: Checking privileges for a view in PostgreSQL is not supported.

Dynamic data modeling: Foreign key statements in the DDL file are not managed.

51.3 Version 3.0.3

Released: September 2021

New features

Fields are now exported even when their visibility in the Data Model Assistant (DMA) is set to hidden.

Changes in Functionality

If a table contains records and you import an empty file using the **Replace all content** mode, all records in the table are deleted. Note that this behavior applies to all import formats, but does not apply to data transfer.

Changes to third-party libraries

The Apache Commons Compress library was upgraded to version 1.21.

Closed Issues

[ADIX-3461] A `NullPointerException` is thrown when printing a foreign key field mapping.

Known Issues

This release contains the following known issues:

- Association object and selection node are not supported.

- Data lineage.
- Graphical view of the data mapping configurations.
- **Split of string** transformation function is not supported for a multi-valued string field.
- Data transfer between complex terminal nodes is possible only if they use the same Java Bean.
- You can not export data from multiple complex nodes in EBX® that when there is at least one complex node with multiple occurrences.
- You can not import data from multiple complex nodes in an XML file when there is at least one complex node with multiple occurrences.
- **Split of data** works for terminal nodes only.

SQL export:

- You can only export data from EBX® to the selected columns in the same tables of a view which is joined by multiple tables.
- The add-on doesn't support exporting SQL with the **Replace all content** mode to a view which is joined by multiple tables.
- Checking mandatory fields of a view in SQL database is not supported.
- The export still executes normally and shows an error message from database when no privilege granted.
- Exporting data from multi-valued fields and multi-valued groups is not supported.
- You can not export data from EBX® to a view in PostgreSQL database.
- Exporting data from EBX® to an auto-increment column in SQL Server is not supported because the **IDENTITY_INSERT** property is always set to `OFF`.

SQL import:

- Checking privileges for a view in PostgreSQL is not supported.

Dynamic data modeling:

- Foreign key statements in the DDL file are not managed.

51.4 **Version 3.0.2**

Released: July 2021

New features

This release includes an updated trademark.

Changes in Functionality

This release includes no updated functionality.

Closed Issues

This release includes no closed issues.

Known Issues

This release contains the following known issues:

- Association object and selection node are not supported.
- Data lineage.
- Graphical view of the data mapping configurations.
- **Split of string** transformation function is not supported for a multi-valued string field.
- Data transfer between complex terminal nodes is possible only if they use the same Java Bean.
- You can not export data from multiple complex nodes in EBX® that when there is at least one complex node with multiple occurrences.
- You can not import data from multiple complex nodes in an XML file when there is at least one complex node with multiple occurrences.
- **Split of data** works for terminal nodes only.

SQL export:

- You can only export data from EBX® to the selected columns in the same tables of a view which is joined by multiple tables.
- The add-on doesn't support exporting SQL with the **Replace all content** mode to a view which is joined by multiple tables.
- Checking mandatory fields of a view in SQL database is not supported.
- The export still executes normally and shows an error message from database when no privilege granted.
- Exporting data from multi-valued fields and multi-valued groups is not supported.
- You can not export data from EBX® to a view in PostgreSQL database.
- Exporting data from EBX® to an auto-increment column in SQL Server is not supported because the **IDENTITY_INSERT** property is always set to `OFF`.

SQL import:

- Checking privileges for a view in PostgreSQL is not supported.

Dynamic data modeling:

- Foreign key statements in the DDL file are not managed.

51.5 **Version 3.0.1**

Released: June 2021

Updates

A warning is now provided when discrepancies exist between preference settings and an import's source file.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-3420]** An error prevents the **Generate from file** service from being executed when generating a model using the **Excel 2007** option.
- **[ADIX-3421]** Users cannot select an export preference after creating an existing preference during a Excel/CSV file export operation.
- **[ADIX-3422]** The defined column name of foreign key fields is not displayed after exporting a CSV file.
- **[ADIX-3424]** The string to decimal number conversion transformation does not work.
- **[ADIX-3425]** [Transfer] The result screen displays incorrect information when transferring data from many tables to one table in the same data model.
- **[ADIX-3426]** Formula values cannot be imported.
- **[ADIX-3427]** Java classes of drivers always get loaded without checking first.
- **[ADIX-3428]** The **Data file sample** displays incorrectly when importing a CSV file into a table.
- **[ADIX-3429]** When using the split transformation function to export CSV, a redundant separator character is included in the exported file.
- **[ADIX-3430]** On a record that contains separator characters, enumeration labels are not enclosed by delimiters (double quotes).
- **[ADIX-3431]** The data of a multi-valued field which contains a separator character is not enclosed by double quotes after being exported to CSV.
- **[ADIX-3432]** Importing Excel ignores boolean values in the XLS file.
- **[ADIX-3433]** The **Concatenate strings**, **Aggregate of integer numbers**, and **Aggregate of decimal numbers** transformation functions do not work on multiple value fields when executing Export Excel or CSV.
- **[ADIX-3435]** When exporting Excel, data is exported under the wrong column on a table that has a list group with a Foreign key field.
- **[ADIX-3436]** A redundant button displays in the **Data file sample** screen when importing a CSV file.
- **[ADIX-3437]** An incorrect message displays when exporting Excel (2003, 2007) with no header.
- **[ADIX-3438]** The Data file sample displays incorrectly when importing a CSV file.
- **[ADIX-3441]** The exported Excel and CSV files always display the old name field instead of the new one.

51.6 **Version 3.0.0**

Released: March 2021

New and updated behavior

This release includes the following updates:

- This version of the add-on has been updated to ensure compatibility with the TIBCO EBX® 6.0.0 GA release.
- Data export and transfer operations were adapted to ensure consistent results. As an example, in previous versions, data might get modified by another user before the operation completed. This

would result in the exported or transferred data containing the modifications. This is no longer the case. The data will be in the same state as when the operation was initiated.

- When importing, an error message now displays to prevent the incorrect configuration of line return or separator characters.
- When importing or exporting using the Excel format, the add-on now defaults to the Excel 2007 option.
- The readability was improved for validation error messages that display during data transfer. The messages now display the label of the record they refer to, which can help you locate and fix the issue.
- You can now define multiple SQL connections for the same data model.
- The cross reference feature was optimized.
- Depending on your deployment environment's resources, you might have issues using the **Insert only** mode when importing a large volume of data. If this issue occurs, use any of the other import modes. Additionally, the results screen will not display the number of updated and unchanged records when using this import mode.
- The **Disable write access lock** option was removed from the add-on. The corresponding APIs were deprecated and calling the any of the following will result in an `UnsupportedOperationException` being thrown:
  - `ExportConfigurationSpec` interface.
  - `FileExportConfigurationSpec`, `SQLExportConfigurationSpec` and `DataExchangeExportSpec` classes.
- Updates to import modes:
  - **Update or insert**: the results screen will not display the number of updated and unchanged records when using this import mode.
  - Use of the **Insert only** mode is not recommended when importing a large volume of data.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-3110]** An Excel/CSV import operation cannot be executed if the complex foreign key field contains special characters.
- **[ADIX-3117]** Transformation function parameters are lost after duplicating it.
- **[ADIX-3121]** [All browsers] The **No** mode of the **Disable write access log** field is not aligned with others on the **Configuration** screen when exporting XML.
- **[ADIX-3123]** An incorrect import result displays when importing an Excel file with the import type as **Import sheets sequentially** after running a Simulation.
- **[ADIX-3128]** An error occurs at the second time when exporting CSV/Excel on workflow and using an invalid preference.
- **[ADIX-3130]** Newly mapped fields are not exported after exporting an Excel file using a preference created from the add-on configuration settings.
- **[ADIX-3131]** Users can set export Foreign/Primary key with permalink and without Export label on the add-on configuration settings.

- **[ADIX-3134]** The **Undefined** value is displayed on the **User profile** field and an exception occurs in the log file if the **Owner** is not defined on the **Application interface preference** record.
- **[ADIX-3135]** The table name is incorrect when generating a model using a CSV file.
- **[ADIX-3136]** An incorrect error occurs when importing CSV using the preference and navigating to the **Mapping column** screen.
- **[ADIX-3140]** Table mappings display incorrectly after removing a table path from the preference when importing Excel at the dataset level.
- **[ADIX-3142]** Services created in the EBX displays incorrectly in the exported Excel file after exporting permissions.
- **[ADIX-3150]** An Excel file cannot be imported at the dataset level after clicking on the **Table mapping**/**Column mapping** link.
- **[ADIX-3152]** The **Disable write access lock** field displays twice in the **Configuration** screen after defining global permissions in an Excel/CSV export operation.
- **[ADIX-3161]** A JavaScript error occurs after pressing Import/Export SQL with no field mapping.
- **[ADIX-3164]** The error message "Service not available for the content" displays when exporting an XML file at the dataset level after accessing a table of this dataset.
- **[ADIX-3173]** A redundant button is displayed on the **File and Preference selection** screen when running the Import Excel/CSV service after setting permission.
- **[ADIX-3214]** A redundant text is displayed on the **Configuration** screen when running the Export XML service after setting permissions.
- **[ADIX-3219]** An exception occurs during CSV import when the previous CSV import operation was interrupted.
- **[ADIX-3230]** The foreign key field is still mapped in the **Mapping column** screen when exporting/importing CSV/Excel using the created preference from the add-on configuration.
- **[ADIX-3242]** A Javascript error occurs when clicking on the **Preview** button of data file sample in Excel import operation at the dataset level.
- **[ADIX-3247]** Parameter of the **Constant value** transformation function is changed after updating a preference in Excel/CSV export operation.
- **[ADIX-3252]** The data sample is displayed incorrectly when importing a CSV file into a table if the data contains more than one separator character.
- **[ADIX-3256]** The **Disable write access lock** field can be modified in the **Configuration** screen although the global permission is "read-only" in an XML export operation.
- **[ADIX-3257]** The target field and target table in the warning message are wrong after exporting/importing Excel/CSV using a transformation function that contains an invalid parameter.
- **[ADIX-3273]** A deadlock occurs when the Excel/CSV export includes related data.
- **[ADIX-3293]** All selected records are not exported successfully when exporting Excel or CSV and the **Export related data** mode is enabled.
- **[ADIX-3320]** An unexpected error occurs after deleting the **Mode** field in the **Date time pattern** table.
- **[ADIX-3352]** A CSV file cannot be imported into a table after stopping the simulation process.

51.7 Version 2.7.7

Released: February 2021

Bug fixes

[ADIX-3370] The import results are not correct when using the **Import CSV** service.

51.8 Version 2.7.6

Released: January 2021

Product updates

The add-on no longer supports the Document Type Definition declaration in XML files. If this causes an exception, the add-on will inform you that this declaration is not allowed.

Bug fixes

[ADIX-3334] An unexpected exception occurs when selecting the **Import XML** service.

51.9 Version 2.7.5

Release Date: October 20, 2020

Bug fixes

This release includes the following bug fixes:

- **[ADIX-2938]** A deadlock occurs when an Excel or CSV export includes related data.
- **[ADIX-3006]** A blocking screen displays during a CSV import operation if the separator and delimiter are different from the imported file.
- **[ADIX-3195]** An unexpected exception occurs while exporting a subset of columns to CSV or Excel.

51.10 Version 2.7.4

Release Date: September 18, 2020

Updates

This release contains the following updates:

- Support has been updated for the JDK11 library.
- The JQuery library was updated to version 3.5.0.
- The Apache POI library was updated to version 4.1.2.

Bug fixes

[ADIX-3308] An invalid Excel file is displayed incorrectly after being imported using the **Download file of invalid data** mode.

51.11 Version 2.7.3

Release Date: June 23, 2020

New and updated behavior

This release includes the following updates:

- Data in hidden fields in the DaqaMetaData group can now be transferred.
- The steps in the API documentation that describe how to create an SQL field mapper for import and export were updated.
- Support has been updated for the JDK8 and JDK11 libraries. Additionally, the log4j library version 1.2.17 was removed.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-3216]** An error message is not shown when trying to export more records than Excel can handle.
- **[ADIX-3221]** Excel import and export fails to export a table with one million records.
- **[ADIX-3224]** The progress value is not displayed on the progress bar when exporting CSV or Excel.

51.12 Version 2.7.2

Release Date: April 20, 2020

New and updated behavior

This release includes the following updates:

- The add-on now includes the progress of import operations in its log.
- Performance has been improved when importing from a CSV file.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-188]** A validation operation on the add-on's **Field** table does not catch errors.
- **[ADIX-219]** Import from a spreadsheet fails when a sheet's name includes a space at the end.
- **[ADIX-225]** When importing CSV or Excel, the current preference cannot be removed after navigating to the previous step from the **Simulation** screen.
- **[ADIX-1428]** A field with default view defined as hidden should not be exported.
- **[ADIX-2119]** The **Import** button does not perform the default submit action.
- **[ADIX-2849]** When importing or exporting Excel or CSV, a transformation function is not created when creating a new preference based on an existing preference.
- **[ADIX-2960]** Add-on services do not return users to the correct location.

- **[ADIX-2980]** Data transfer operations do not complete.
- **[ADIX-2997]** A transformation function is removed after updating a preference.
- **[ADIX-2998]** Newly added table columns are not available to map in existing preferences.
- **[ADIX-3002]** An exception occurs when importing Excel at the dataset level and updating or creating a preference.
- **[ADIX-3005]** The preference is not updated after modifying the mapping column under a multi-valued group during an Excel import operation.
- **[ADIX-3050]** When importing or exporting using a preference, selecting the **Remove current preference** label does not select its checkbox.
- **[ADIX-3126]** The add-on's performance was slow when importing a CSV file.
- **[ADIX-3129]** A field cannot be mapped if it is both a primary key and foreign key and its permission is set to Read-only.
- **[ADIX-3141]** Groups that contain an association do not display when mapping.
- **[ADIX-3159]** A `NullPointerException` occurs while exporting a dataset's permissions.

51.13 Version 2.7.1

Release Date: December 10, 2019

Updated behavior

When using the **Validate imported records only** option, the add-on only validates records included in the import. This update can improve performance as previously the entire table was validated and results were filtered to only show those related to imported records.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-3014]** The validation message cannot be seen in the comment of an Excel 2007 file after exporting using the **Include validation messages** mode.
- **[ADIX-3071]** Data is exported to the wrong column when using the option to export foreign key labels.

51.14 Version 2.7.0

Release Date: November 8, 2019

New features and updates

The new features and updates for this release are described below:

Updates and enhancements

This release contains the following updates and enhancements:

- For Excel and CSV, the **Update and insert** import mode has been renamed to **Update or insert**.
- Administrators can now use permission settings to determine access rights to import, export, and transfer screens. Additionally, they can specify the type of import modes accessible to users. For example, they could set permissions so that a profile can only access the **Update or insert** option and not the **Replace all** option.
- The add-on now offers fine-grained control over how configuration and preference data can be imported and exported. When importing, administrators can now choose the scope of configuration data to import, as well as specify that only preference data from a specific profile gets imported.
- The new **Export preference** service allows export of configuration data related to selected preferences. The service is accessible from the **Application interface preference** table.
- If a user has made adjustments to import settings and not saved a preference, a prompt now displays to inform them that they can save settings as a preference.
- Dataset permissions data can now be exported to an Excel file.

API Updates

API updates included in this release provide the:

- methods to get a `PrimaryKey` set that identifies records in the **Application interface preference** table. See `ApplicationInterfaceConfigurationFactory` in the Java API documentation for more information.
- methods to export the records in the **Application interface preference** table and the related data from other tables to an XML file. See `ExportDataExchangeConfigurationSpec` in the Java API documentation for more information.
- methods to import add-on configuration data from an XML file based on the specified preference owner and import scope. See `ImportDataExchangeConfigurationSpec` and `ImportScope` in the Java API documentation for more information.
- context to get the `com.orchestranetworks.addon.dex.DataExchangeSpec` from the configuration declared in the add-on's dataset. See `DataExchangeHelper`, `DataExchangeHelperContext`, `CSVExportDataExchangeHelperContext`, `CSVImportDataExchangeHelperContext`, `SpreadsheetExportDataExchangeHelperContext` and `SpreadsheetImportDataExchangeHelperContext` in the Java API documentation for more information.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-2745]** Associations under a group still get handled while exporting Excel.
- **[ADIX-2752]** Remove error messages regarding new records cannot be imported when importing data with the "Update only"/"Insert only" mode.
- **[ADIX-2754]** Missing error message on import result screen.
- **[ADIX-2901]** Nested groups are exported improperly under XML format.
- **[ADIX-2955]** When re-importing from CSV using a preference where the separator is a single quote character and the delimiter a double quote, the separator is not imported correctly.
- **[ADIX-3017]** An unexpected exception occurs while exporting data under Excel format.

51.15 Version 2.6.4

Release Date: September 3, 2019

Updates and enhancements

This release contains the following updates and enhancements:

- During the simulation phase of Excel and CSV import you can now:
  - Use the **Validate imported records only** option to specify whether the validation report runs only on imported records, or also includes all records in the target table.
  - Skip the detailed validation report.
- For Excel import and export, multi-valued complex fields are now supported.
- During CSV export, when an exported field includes the same character used as a separator, the add-on now automatically uses double quotes as escape characters to ensure data integrity.
- The option to specify whether the `handleNewContext()` method executes on import or transfer is now available. The new **Allow transient record context management** property can be accessed from the *Administration > Integration > TIBCO EBX® Data Exchange Add-on > Additional configuration > Import preference* table.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-367]** Preference does not get an updated starting position when importing Excel.
- **[ADIX-2896]** Columns can be missing or exported in an incorrect order when exporting from a custom table view.
- **[ADIX-2898]** The simulation progress bar should display the status when getting the validation report.
- **[ADIX-2972]** When re-importing from CSV using a preference where the separator is a single quote character and the delimiter a double quote, the separator is not imported correctly.

51.16 Version 2.6.3

Release Date: June 20, 2019

Updates and enhancements

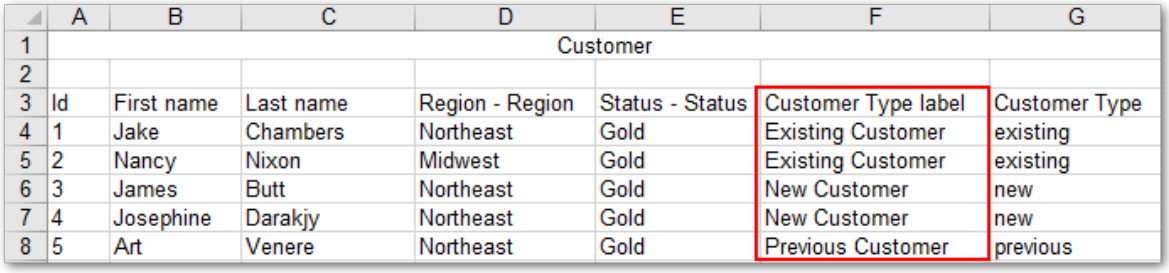
This release contains the following updates and enhancements:

- When exporting Excel using preferences, you now have the option of adding or removing tables in the main configuration screen.
- For Excel export, the **Preference** field is now located at the top of the main configuration screen.
- Export of fields that are enumerations has been enhanced. See the section below for more details.
- An API to return a `CommonApplication` using the application's Universal Name is now available.
- After selecting a preference, the **Export** and **Mapping** buttons are now displayed. This allows an export to be performed without having to navigate through each screen.

Enumeration export enhancements

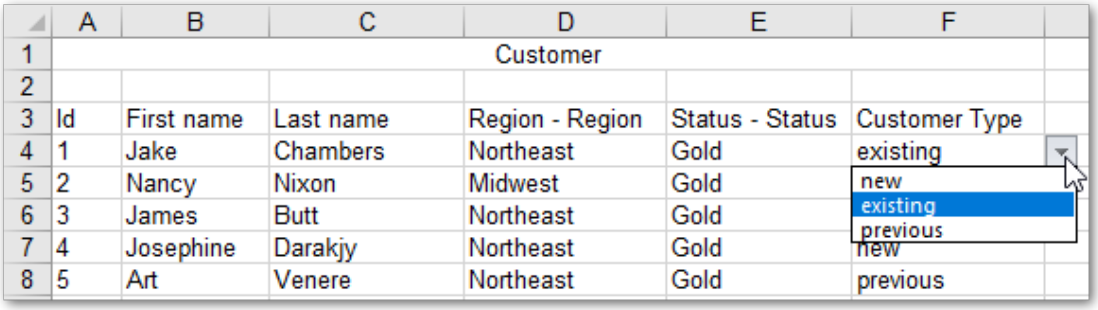
The add-on allows you to include enumerations in CSV and Excel exports of tables, or individual records. The following describes behavior for each supported export type:

- **Excel** export: On the main configuration page, the add-on presents you with the following options under **Export enumerations**:
  - **Export label**: The export includes an additional column for the enumeration field's label.



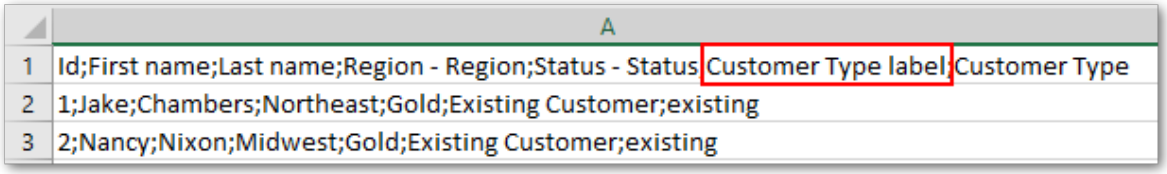
	A	B	C	D	E	F	G
1	Customer						
2							
3	Id	First name	Last name	Region - Region	Status - Status	Customer Type label	Customer Type
4	1	Jake	Chambers	Northeast	Gold	Existing Customer	existing
5	2	Nancy	Nixon	Midwest	Gold	Existing Customer	existing
6	3	James	Butt	Northeast	Gold	New Customer	new
7	4	Josephine	Darakjy	Northeast	Gold	New Customer	new
8	5	Art	Venere	Northeast	Gold	Previous Customer	previous

  - **Export static enumerations**: The enumeration column will contain a drop-down list of enumeration values defined in the data model.



	A	B	C	D	E	F
1	Customer					
2						
3	Id	First name	Last name	Region - Region	Status - Status	Customer Type
4	1	Jake	Chambers	Northeast	Gold	existing
5	2	Nancy	Nixon	Midwest	Gold	new
6	3	James	Butt	Northeast	Gold	existing
7	4	Josephine	Darakjy	Northeast	Gold	previous
8	5	Art	Venere	Northeast	Gold	previous

  - When you select both options, the exported columns both have drop-down lists of their available values.
  - If you leave both options unchecked, the exported column includes only the enumeration value defined in the data model.
- **CSV** export: The exported file will include a dedicated column with the enumeration value's labels.



	A
1	Id;First name;Last name;Region - Region;Status - Status;Customer Type label;Customer Type
2	1;Jake;Chambers;Northeast;Gold;Existing Customer;existing
3	2;Nancy;Nixon;Midwest;Gold;Existing Customer;existing

Bug fixes

This release includes the following bug fixes:

- **[ADIX-2677]** When exporting CSV, static enumerations on integer fields are not properly exported.
- **[ADIX-2680]** An unexpected exception occurs when importing Excel.

- **[ADIX-2682]** A `NullPointerException` occurs when starting EBX®.

51.17 Release Notes 2.6.2

Release Date: January 18, 2019

Updates and enhancements

This release contains the following updates and enhancements:

- As part of an effort to improve overall user experience, the file and preference selection pages have been combined for Excel and CSV import.
- Dynamic Data Modeling services can now be invoked via the API.

Bug fixes

This release includes the following bug fixes:

- **[ADIX-573]** A temporary exported file is created when there are no records to export via the API.
- **[ADIX-2663]** An error occurs when using a constant transformation to transfer data on a primary key field with no source field.

51.18 Release Notes 2.6.1

Release Date: December 14, 2018

Updates and enhancements

This release contains the following updates and enhancements:

- The add-on can now automatically handle line break characters when importing and exporting.
- The add-on no longer displays warning messages pertaining to write permission for hidden or read-only fields, associations, and selection nodes.
- The option to export ignored fields as blank columns now defaults to **No**.

Bug fixes

This release includes the following bug fixes:

- **[36460]** Service labels are duplicated in workflows.
- **[36461]** Transformation functions do not get applied on foreign key fields when using a preference.
- **[36462]** Errors are logged at the debug level.
- **[36463]** When importing Excel at the dataset level, nothing happens when selecting the **Import** button.
- **[36720]** When importing or exporting Excel/CSV, an exception is raised in the log when loading the **Extensions** drop-down list.

51.19 Release Notes 2.6.0

Release Date: October 26, 2018

Featured updates

The EBX® Data Exchange Add-on has undergone significant updates to ensure compatibility with the EBX® 5.9.0 GA release.

Bug fixes

This release includes the following bug fixes:

- **[33698]** An incorrect result is returned when importing a string that contains only numbers and starts with zero.
- **[35697]** An unexpected error occurs when creating a new preference during an Excel import operation.
- **[36030]** A javascript error occurs when exporting an empty XML file.
- **[36041]** The add-on crashes when exporting a large number of records in the Excel format.
- **[36051]** The number of unmatched fields is wrong after running the **Auto data mapping** service.
- **[36085]** An unexpected exception occurs when transferring data using a Validator.

51.20 Release Notes 2.5.1

Release Date: July 17, 2018

Bug fixes

This release includes the following bug fixes:

- **[33926]** An unexpected exception occurs when running the simulation.
- **[33951]** The template for tables is not applied when exporting Excel and ignoring all columns.
- **[33952]** Technical error is not displayed when running simulation of importing CSV into table which has a read-only primary key.
- **[33953]** Validation error at table level is not displayed in the Simulation result screen when importing Excel/CSV file.
- **[33955]** An unexpected error occurs when creating a new preference during a CSV/Excel import/export operation.
- **[33957]** An unexpected error occurs when exporting CSV file after importing an archive file in EBX® Data Exchange Add-on dataset.

51.21 Release Notes 2.5.0

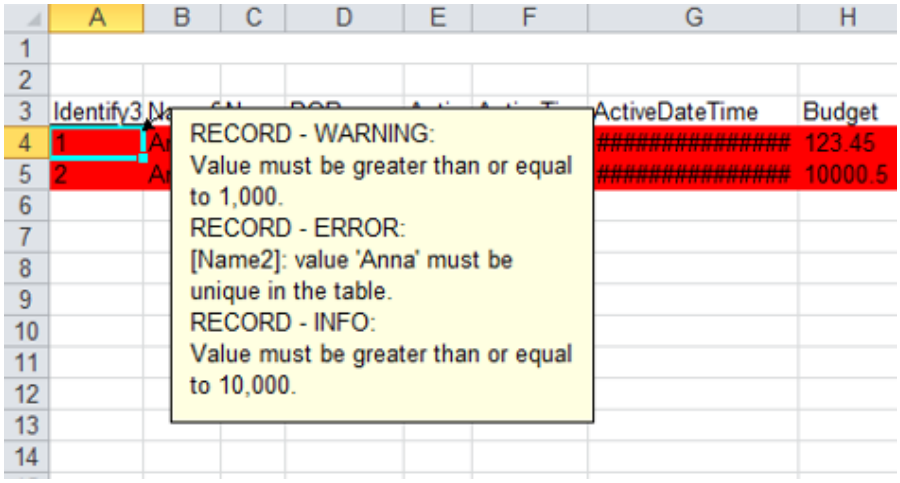
Release Date: July 4, 2018

New features and enhancements

The new features and enhancements in this release are highlighted in the following sections:

Inclusion of validation messages in Excel export

You can now include validation messages when exporting to Excel. In the exported file, the add-on highlights fields and rows containing messages. If a message applies to a table, the add-on creates a new sheet in the exported file and appends "-MSG" to the name. See Including messages in exported files [p 34] for more details. The image below shows an example of how messages display in an exported file:



Simplified SQL data source connections

Administrators can now create a connection to an SQL data source without having to manually define the connection information in the application server. Connection settings can be configured using the new **JNDI data source** table located under *Administration* > *Integration* > *EBX® Data Exchange Add-on* > *Reference data*. For further instructions, see Connecting to an SQL data source [p 115].

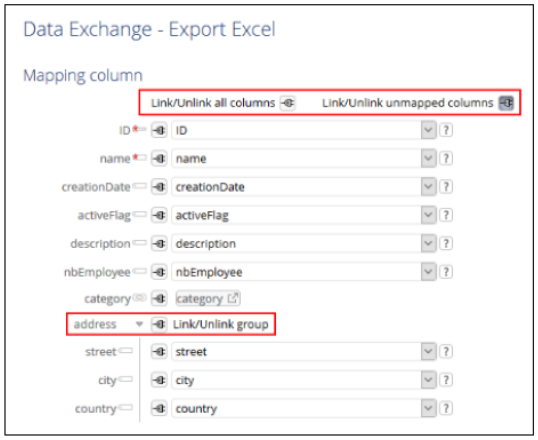
Enhanced data transfer options

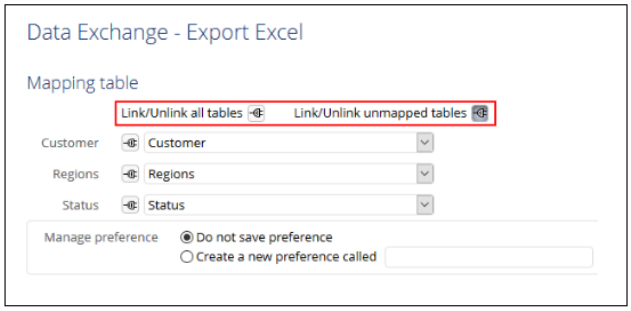
You can now select multiple target tables when transferring data. Additionally, the new **Stop and rollback on error** option allows you to specify behavior when errors occur during transfer. When you enable this option and errors are encountered during transfer, the add-on cancels all transfers in progress and rolls back any transferred data. When the option is disabled and errors are encountered, the add-on only cancels transfers for the tables where errors occurred and completes any remaining transfers.

Updated table and column mapping screen features

As highlighted in the following image:

- All tables, columns, unmapped tables, and unmapped columns can be linked and unlinked with a single click.
- Fields that are part of a complex type are now grouped. Each group can also be linked and unlinked without having to select individual fields.





Improved transformation function features

The following enhancements have been made to transformation functions:

- Transformation function parameter values now display in the **Field mapping transformation** table.
- You can now use fields of all data types with a cross reference transformation function. If the source and target data types are different:
  - The *source* field and the field used to lookup the desired value (located in the cross-referenced table) must be the same data type.
  - The field containing the replacement value (located in the cross-referenced table) and the *target* field must be the same data type.

New functionality to export related data

To help you locate data linked to specific records, the add-on now allows you to include related data when exporting to:

- CSV: The add-on exports a ZIP file containing individual CSV files. One file contains the export's source data. The remaining files—one for each table—contain the related data.
- Excel: The add-on exports a single Excel file. The first sheet in the file contains the export's source data. Each additional sheet—one for each table—contains the related data.

The option to export related data is only available when exporting from the table level, or individual records. For more information, see Exporting related data [p 40].

Addition of default options and preference permissions settings

Administrators can now set the default selections and preference permissions for each import and export operations. When users perform an import/export, they can still change selections if needed, or load available preferences.

Additional improvements

This release contains the following additional improvements:

- Business rules can now be checked when simulating CSV or Excel import.
- When opening an existing, or creating a new **Table mapping** record, the **Field mapping** tab displays to provide quick access to field mappings.
- The UI has been updated when importing and exporting Excel at the dataset level.

Java API

Additions to the API included in this release allow you to:

- Generate table and application mappings for import and export of Excel and CSV formats. See `CSVTableGeneration`,`SpreadsheetTableGeneration`, `CSVExportApplicationMappingHelper`, `CSVImportApplicationMappingHelper`, `SpreadsheetExportApplicationMappingHelper` and `SpreadsheetImportApplicationMappingHelper` in the Java API documentation for more information.
- Get the table declared in the EBX® Data Exchange Add-on configuration. See `TableHelper`, `TableHelperSpec` and `TableHelperFactory` in the Java API documentation for more information.
- Export related data when exporting to Excel or CSV formats. See `FileExportConfigurationSpec` in the Java API documentation for more information.
- Define the JNDI configuration used for SQL import and export without requiring application server configuration. See `JNDIDataSource`, `SQLExportConfigurationSpec` and `SQLImportConfigurationSpec` in the Java API documentation for more information.
- Export validation messages when exporting to Excel. See `SpreadsheetExportConfigurationSpec` in the Java API documentation for more information.
- Stop a data transfer in progress when errors occur and rollback all data. See `TransferConfigurationSpec` in the Java API documentation for more information.
- Export tables with a prepared list of each table's records, or table filter. See `DataExchangeExportTableSpec`, `DataExchangeExportTableSpecBuilder` and `DataExchangeService` in the Java API documentation for more information.

Bug fixes

This release includes the following bug fixes:

- **[23544]** A record cannot be imported if the length of a field value violates min/max length with a specific error management policy enabled.
- **[30567]** The parameter value of a transformation function cannot be deleted.
- **[30993]** The date and date-time formats in the configuration are not applied when using a preference to export CSV.

- **[31716]** Incorrect behavior occurs when the correct cross reference transformation function is not used to import CSV.
- **[31775]** An incorrect result screen displays when importing an Excel file containing a constraint violation at the dataset level.
- **[32020]** An incorrect warning displays when importing from a CSV file using an incorrect cross reference.
- **[32027]** [IE11] Data transfer cannot be executed due to an incorrect validation warning.
- **[32074]** [IE11] A java script error displays when exporting a CSV/Excel file and selecting an item in the 'Mapping column' screen.
- **[32088]** The transformation function defined for a recursive foreign key is not shown when exporting Excel.
- **[32262]** The tooltip for the mapping of a foreign key is not shown when exporting/importing CSV/Excel on a table or data set.
- **[32360]** A warning is not displayed when using an aggregate mapping type without any transformation function during CSV/Excel export.
- **[32368]** The invalid request error is shown during a CSV/Excel export operation if the target field name of a 'No source field' is changed.
- **[32377]** An unexpected error occurs when exporting a CSV file and updating a preference.
- **[32421]** An unexpected error occurs when re-creating a deleted preference during a CSV export operation.
- **[32479]** The error message includes the wrong content when importing from CSV using an extension and 'Validate data before transforming data' is un-checked.
- **[32581]** Cannot import XLSX file without the "Shared String Tables" package part.
- **[32698]** The wrong error message displays when generating a CSV file if the same character is set in the 'Delimiter' and 'Separator' fields.
- **[32707]** An unexpected error occurs when a preference containing undefined fields is used to export Excel files.
- **[32709]** A spelling mistake can be found in the description of the 'Refresh hierarchical views' service.
- **[32716]** An incorrect error displays when running the 'Generate models' service on the Excel application type.
- **[32718]** The tooltip on the 'Generate models' service contains insufficient information.
- **[32742]** The wrong message displays when deleting a record staring with [ON] in the 'Transformation function' table.
- **[32753]** The wrong warning is raised when users run bi-directional transfer using the constant value transformation function.
- **[32786]** The display of multi-value group is not consistent when users export Excel at the table and data set levels.
- **[32840]** After stopping the 'Import Excel' service, the error message's display is inconsistent between the dataset and table levels.

- **[32894]** If a dataset contains some tables with the same label, users should be able to choose a table to export to XML from this dataset.
- **[32899]** The 'EBX® Data Exchange Add-on export XML' screen is not refreshed after navigating to another dataset.
- **[33072]** Some field values in a transformation function are incorrect.
- **[33153]** An error occurs when running the 'Report data mapping' service after selecting a record on the 'Table mapping' table.
- **[33519]** An unexpected error occurs when exporting default XML file using 'No source field'.
- **[33525]** The order of data is wrong when importing XML from multiple source fields to one target field.
- **[33568]** Data cannot be imported into a time/date time data type field when using 'No source field' during an Excel/CSV import/export.
- **[33588]** [IE11] The table name is wrong when importing CSV and creating a new preference.
- **[33726]** The validator declared in the EBX® Data Exchange Add-on dataset does not get applied when importing Excel/CSV using a preference.
- **[33747]** An unexpected exception occurs when launching Excel export/import in a multi-process environment.
- **[33748]** An Excel/CSV import operation cannot be executed if the composite foreign key field contains special characters.
- **[33792]** The EBX® Data Exchange Add-on's 'Import XML' service does not work properly.

51.22 **Release Note 2.4.9**

Release Date: June 8, 2018

Bug fixes

This release includes the following bug fixes:

- **[33297]** An unexpected exception occurs when displaying the mapping column screen.

51.23 **Release Note 2.4.8**

Release Date: May 23, 2018

Bug fixes

This release includes the following bug fixes:

- **[31779]** Users cannot import a CSV/Excel/XML file that contains a null string field after setting length constraint for this field.
- **[32669]** If any of a record's fields that have a specified minimum length are empty or null in the source file, the record will not be imported.
- **[32799]** The add-on does not retain the date format selected in the mapping configuration when importing a CSV file.

51.24 Release Note 2.4.7

Release Date: May 2, 2018

Latest updates

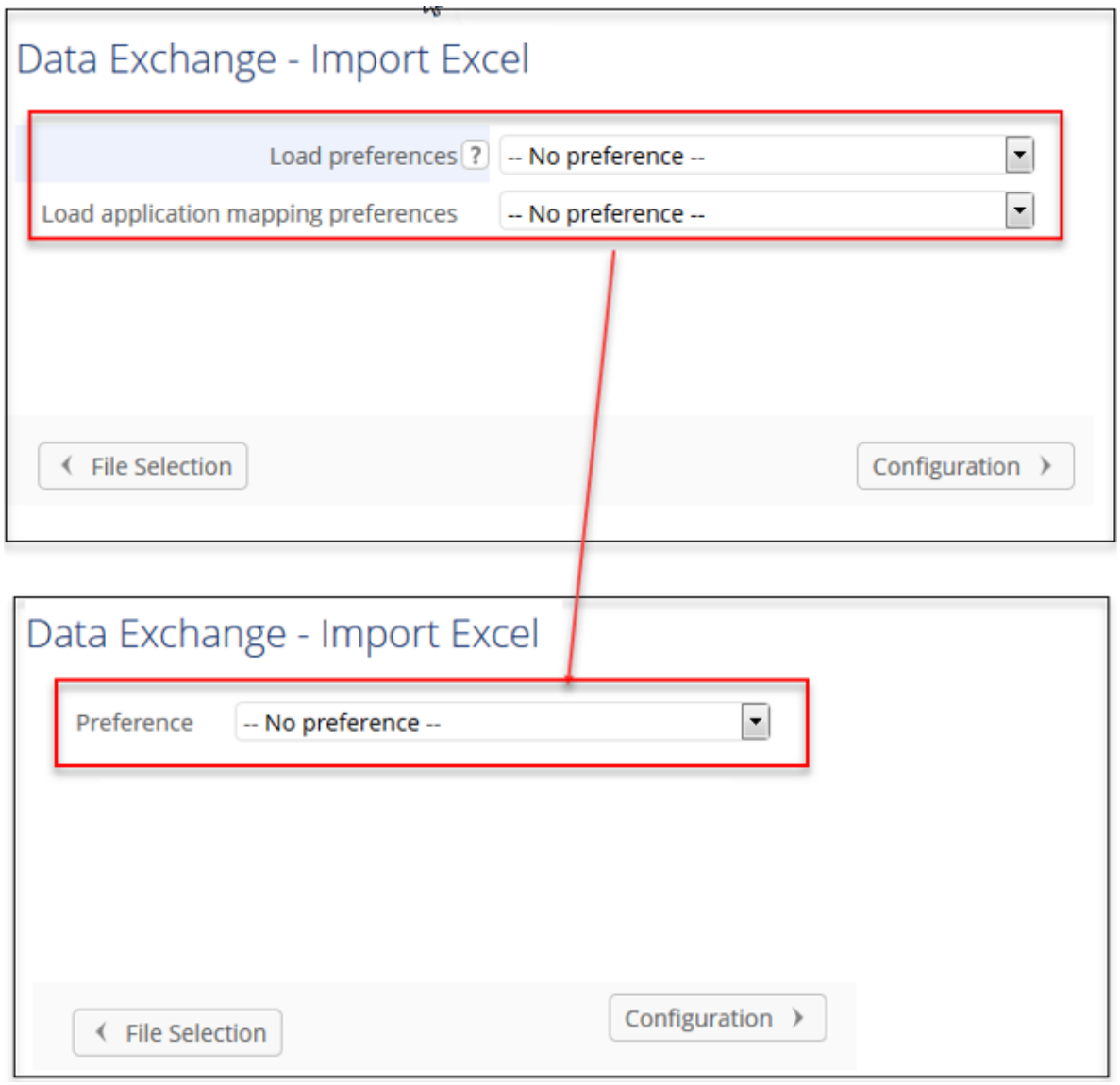
The EBX® Data Exchange Add-on import now relies on the commit threshold value set in ebx.properties. If left undefined, the threshold value defaults to 100.

51.25 Release Note 2.4.6

Release Date: April 20, 2018

Latest updates

This release of the EBX® Data Exchange Add-on simplifies preference loading by consolidating functionality. You now use only the **Preference** drop-down menu to specify preferences for Excel and CSV import and export. The new behavior also applies to workflows.



Java API

It is now possible to get the procedure context of an ongoing transaction when importing. See `com.orchestranetworks.addon.dex.transformation.TransformationExecutionContext` in the Java API documentation for more information.

Backwards compatibility

The `getImportPreference` and `setImportPreference` methods are no longer used. Export mapping based on an import mapping preference is no longer supported. The mapping defined in the export preference is used instead. See `AdixExportSpec` in the Java API documentation for more information.

Bug fixes

This release includes the following bug fixes:

- **[31789]** When importing CSV, the add-on always performs data validation prior to transformation.

- **[32182]** Incorrect table labels display on the mapping column screen.

51.26 Release Note 2.4.5

Release Date: April 6, 2018

Bug fixes

This release includes the following bug fixes:

- **[32019]** Performance has been improved when exporting data that includes foreign keys to Excel or CSV files.

51.27 Release Note 2.4.4

Release Date: March 28, 2018

Bug fixes

This release includes the following bug fixes:

- **[31874]** To improve performance during SQL import, the Java class should be loaded less frequently when converting the data type from SQL.

51.28 Release Note 2.4.3

Release Date: March 16, 2018

Bug fixes

This release includes the following bug fixes:

- **[31283]** In Excel and CSV import, the mapping of computed foreign key fields has been disabled.
- **[31356]** An unexpected exception occurs when exporting Excel.
- **[31664]** NullPointerException is thrown when using API to transfer data.

51.29 Release Note 2.4.2

Release Date: January 31, 2018

New features

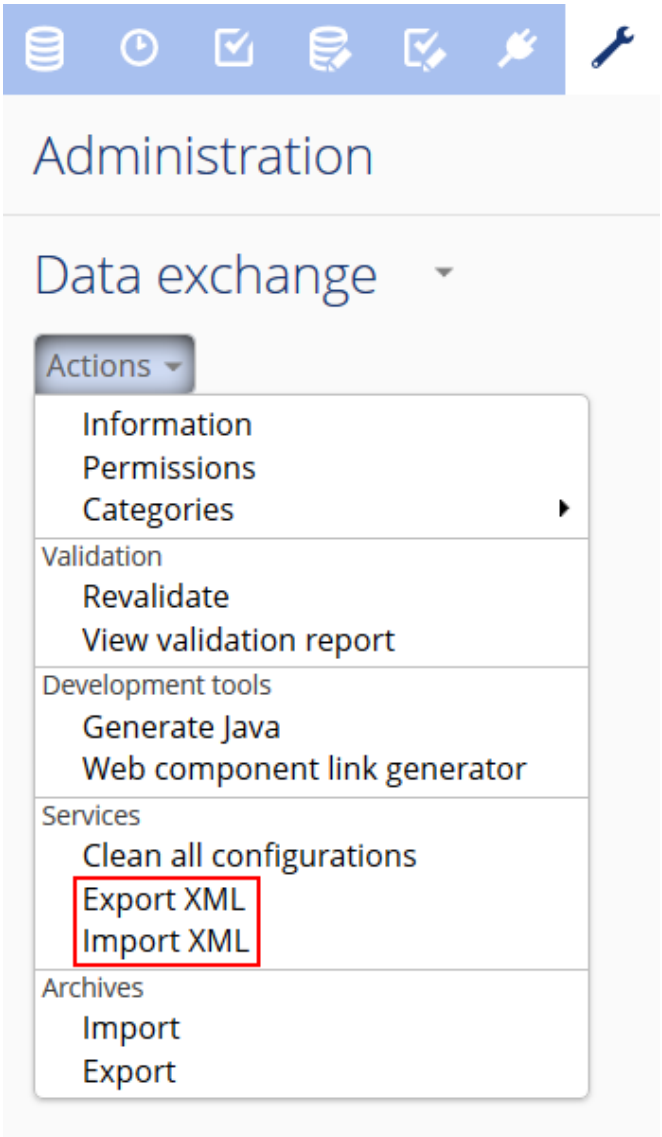
This release contains the following new features and bug fixes:

EBX® Data Exchange Add-on configuration Import/Export

You can now import and export entire EBX® Data Exchange Add-on configurations using the following services:

- **Export XML**: Exports all table records from the EBX® Data Exchange Add-on dataset to an XML file.
- **Import XML**: Imports data from an XML file to the EBX® Data Exchange Add-on dataset.

As shown below, the services are available to administrators from the EBX® Data Exchange Add-on dataset **Actions** menu:



For more information, see Importing and Exporting EBX® Data Exchange Add-on configuration [p 201].

Importing when a table's PK is read-only

The following list outlines improved behavior when importing to a table that has a read-only primary key:

- When importing CSV/Excel: If you map the primary key field, you can use the **Update only** and **Update and insert** modes. When using these modes records are updated normally, but an error is displayed if the import attempts to add a new record to the target table.
- When importing from SQL/XML and transferring data: You can use the **Update or insert** mode to update records on import/transfer. Note that you will not be able to import/transfer new records to the target table.

Java API

It is now possible to import and export EBX® Data Exchange Add-on configurations from/to an XML file. See `DataExchangeConfigurationService` in the Java API documentation for more information.

Bug fixes

- **[27717]** Unclear warning messages display when transferring data from a dataset containing tables that have the same label.
- **[30249]** CSV/ Excel import using a preference causes the 'Password' and 'Computed value' fields to not display in a user-friendly manner.
- **[30476]** Exporting to CSV is not possible when an ignored column mapping contains an invalid date/date-time/time pattern.
- **[30530]** Some semi-colons are redundant in a file containing incorrect data when importing CSV.
- **[30550]** When working with Excel, a table that has been ignored does not display in the 'Mapping column' screen after being remapped.
- **[30555]** An unexpected exception occurs when creating a new field mapping with the 'Ignored field' set to 'Yes' in the configuration.
- **[30562]** CSV/ Excel import using a preference causes the primary key field mapping in the Mapping column screen to not display in a user-friendly manner.
- **[30627]** Import of Excel using a preference without table mappings and field mappings results in an incorrect screen and a javascript error.
- **[30698]** After setting a composite foreign key to ignored, its mapping is still shown during an Excel/CSV import operation.
- **[30726]** Incorrect behavior occurs when importing Excel with an invalid primary key value.
- **[30751]** The column configuration is not applied when generating the model of an Excel file.
- **[30752]** Incorrect behavior occurs when exporting several tables to Excel with the 'Load import mapping' option activated.
- **[30754]** An incorrect error message is raised when exporting many tables to Excel using the 'Load import mapping' preference.

51.30 **Release Note 2.4.0**

Release Date: December 15, 2017

Overview of features and enhancements

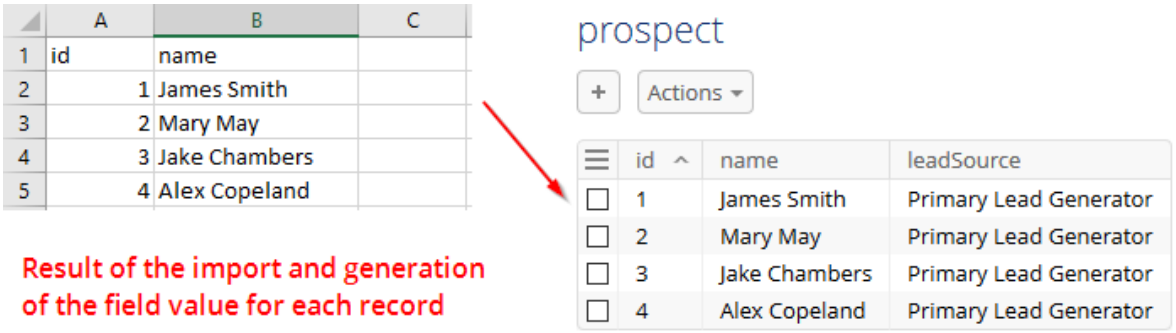
The following list contains features and enhancements for EBX® Data Exchange Add-on version 2.4.0. You can use the links provided for more detailed descriptions of the larger features.

- Generating constant values [p 264]
- Generating models [p 264]
- Data transfer enhancements [p 265]
- When exporting SQL, or transferring data, you can now use the **Include computed values** option to specify whether to include the source's computed values.
- Preferences and permissions have been consolidated into the EBX® Data Exchange Add-on administration area.
- A **Snapshot** input parameter is now included for export and transfer data services in workflows and perspectives.
- You can now define table mappings for Excel export and field mappings for Excel and CSV export.
- During Excel export you can now choose whether ignored fields are included in the export. If included, the add-on exports them as blank columns in the spreadsheet.
- Bug fixes [p 267]

Generating constant values

A new transformation function allows you to populate a column's fields with a pre-defined value on import, export, or transfer. This allows you to set a specific value to a target field even when there is no mapped source field.

In the image below, the add-on automatically inserted a pre-defined value in the table's *leadSource* column during import. This column did not exist in the source spreadsheet. See the *User Guide > Generating constant values* [p 85] for more information and step by step instructions that demonstrate a use case.

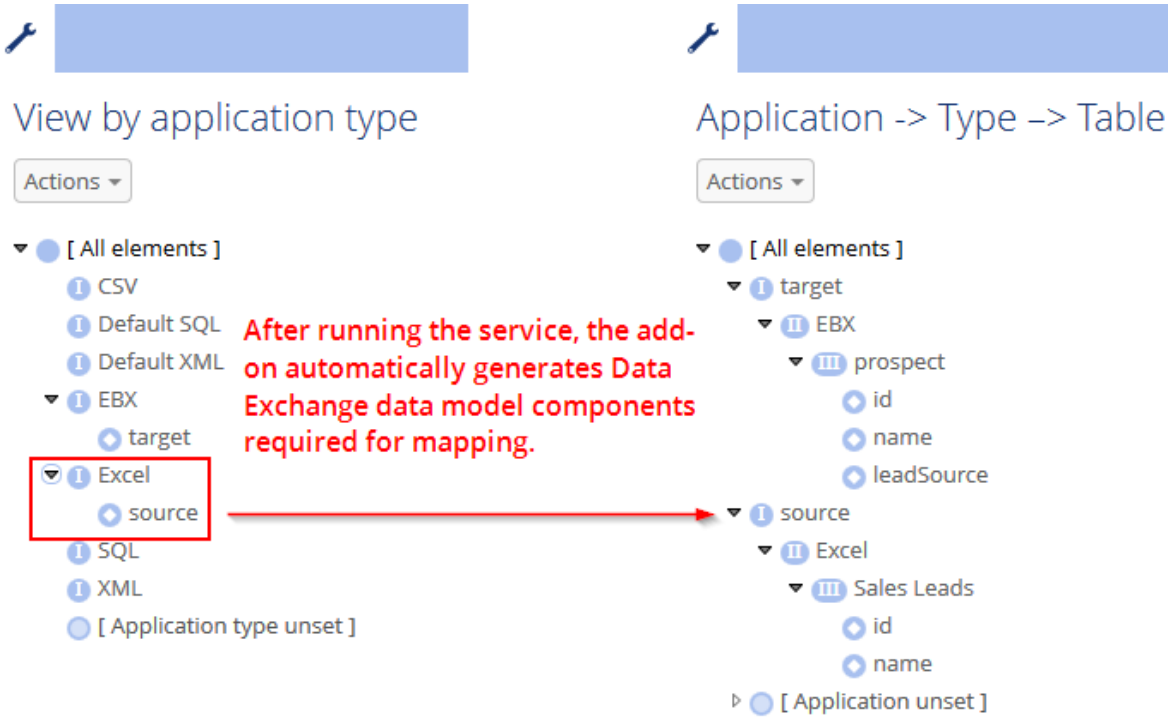


Generating models

In EBX® Data Exchange Add-on, models are required for user-defined and auto-generated mappings. Previously, administrators were required to manually create models for Excel and CSV type applications. This process has been greatly simplified and the add-on can now automatically generate models from a supplied file.

Suppose an administrator needed to create a configuration that uses a transformation function during Excel import. The following provides a high-level outline of the simplified process:

- Create applications and their types corresponding to the EBX® table and Excel file.
- For each application type, run the **Generate models** service. When run for Excel, just provide the file location and basic configuration information.



- Map the tables, fields, and supply the transformation information.

Data transfer enhancements

This release includes the following new features and enhancements to data transfer functionality:

- Transfer data to/from any table [p 265]
- Filtering records [p 266]
- API Updates [p 266]

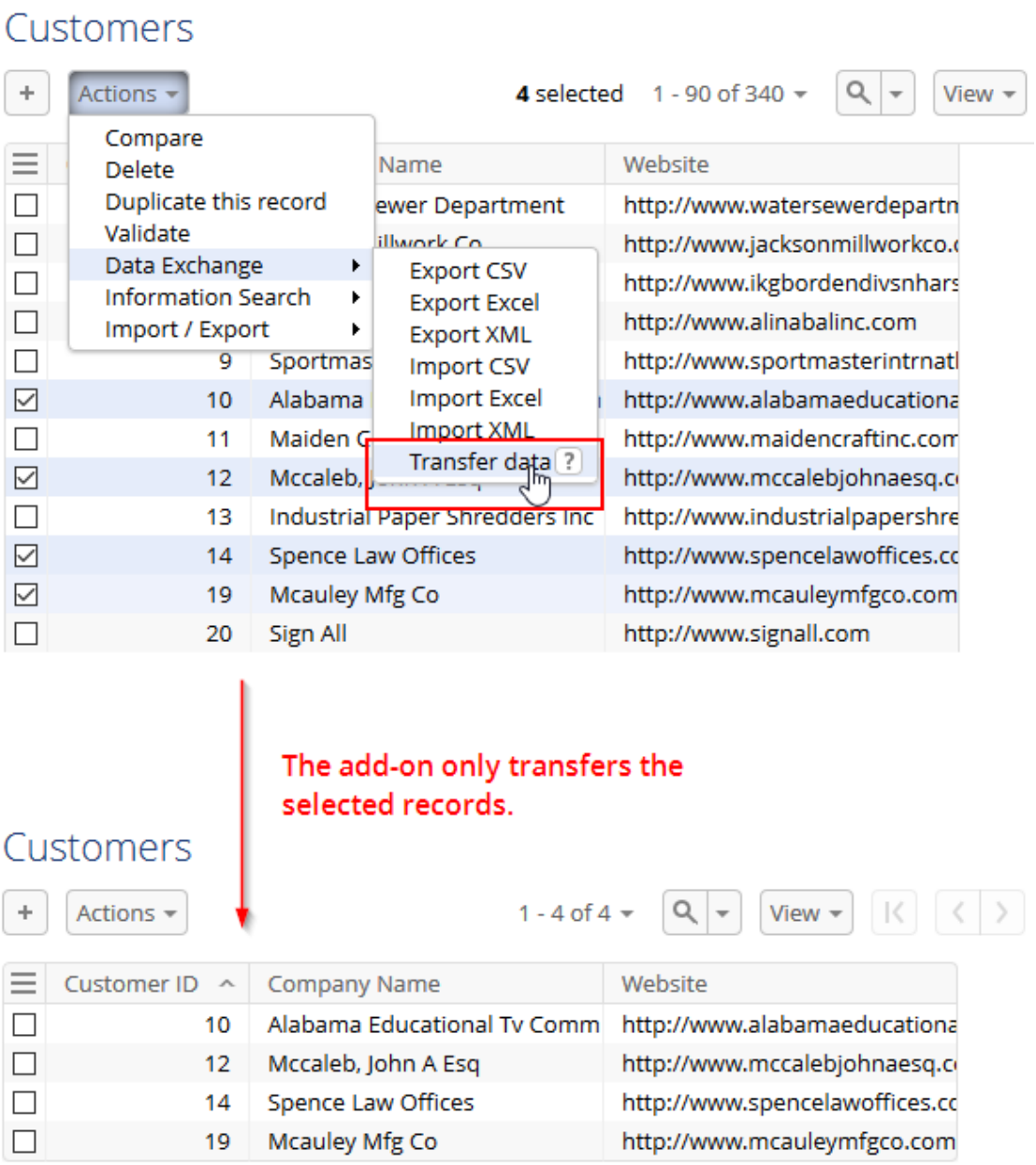
Transfer data to/from any table

A previous limitation prevented you from transferring data between tables within the same dataset. This limitation has been removed and you can transfer data between all tables in the repository (of course, mapping configuration requirements and permission constraints still apply). The following describes behavior when transferring data within the same table:

- The default behavior is that data will be replaced. For example, a business need may require an update of values in a table field to all uppercase letters. In this instance, data transfer could be used in conjunction with a transformation function to update the values.
- When the table has an auto-incremented primary key, you can choose to duplicate the data rather than replace it.

Filtering records

When transferring data between tables, you may only want to transfer a subset of records. Data transfer now supplies this functionality. As shown below, you can select the records you want to transfer in the tabular and hierarchical views:



API Updates

When transferring data using the API, you can now retrieve information such as source and target record primary keys. Additionally, the API now allows you to filter records during transfer. See the API documentation for examples.

Java API

- It is now possible to specify the characters used to signify the end of a CSV line. See `CSVImportConfigurationSpec` and `LineReturnCharacters` in the Java API documentation for more information.
- It is now possible to export ignored fields as blank columns. See `SpreadsheetExportConfigurationSpec` in the Java API documentation for more information.
- It is now possible to transfer computed values from source to target tables. See `TransferConfigurationSpec` in the Java API documentation for more information.
- It is now possible to specify the configuration used for an EBX® reference field. See `EBXLinkField` in the Java API documentation for more information.
- It is now possible to define possible field attributes. See `FieldAttribute` in the Java API documentation for more information.
- It is now possible to create a field with the NO_SOURCE_FIELD field attribute. See `CSVField`, `EBXField`, `SpreadsheetField`, `SQLField` and `XMLField` in the Java API documentation for more information.
- It is now possible to return a list of the targeted deleted records' primary keys before data transfer, and a map between source and target record primary keys for created or updated records in the transfer result. See `TransferResult` in the Java API documentation for more information.
- Provides the necessary context for integrating EBX® Data Exchange Add-on services. See `DataExchangeServiceContext` and `DataExchangeServiceContextFactory` in the Java API documentation for more information.

Bug fixes

- **[23598]** A CSV formatted file cannot be imported when it uses the line feed character for line breaks.
- **[27417]** There is a spelling mistake in the field label on the Transfer data configuration screen.
- **[27436]** An unexpected error occurs when exporting XML using an invalid parameter in the 'Split of string' transformation function.
- **[27555]** When importing CSV using a transformer the result is not as expected.
- **[27638]** During CSV and Excel import or export data is incorrectly imported/exported in certain scenarios.
- **[27657]** There is an unclear warning message when transferring data with some hidden/read-only fields.
- **[27675]** Incorrect behavior occurs when transferring data from one field under a terminal node.
- **[27702]** An exception occurs in the log file when running import/export SQL using a workflow.
- **[28902]** Import Excel fails when the file extension is in uppercase.
- **[29258]** The correct result is not achieved when exporting Excel with a specific access rule.
- **[29338]** Users are redirected to the wrong screen when importing Excel with multiple tables using the 'Import sheets sequentially' mode.
- **[29463]** A CSV file that contains only a primary key cannot be imported into a table with an auto-increment primary key.

- **[29622]** Excel and CSV files cannot be imported after stopping the import simulation.
- **[29640]** An incorrect result screen displays when users who do not have delete permissions at the dataset level import multiple tables from Excel.

51.31 Release Note 2.3.2

Release Date: November 10, 2017

Bug fixes

- **[27655]** Data cannot be imported from an Excel file at the dataset level when splitting data between tables.
- **[29311]** An unexpected exception occurs when importing Excel from one to many tables using the 'Import sheets sequentially' mode.

51.32 Release Note 2.3.1

Release Date: September 20, 2017

Bug fixes

- **[28338]** Fields under non-terminal nodes are not mapped in the Import SQL mapping column configuration screen.

51.33 Release Note 2.3.0

Release Date: August 3, 2017

New features

- The add-on's architecture has been updated and includes a common, unified API. Additionally, this version has been optimized for improved performance.

Java API

- It is now possible to use the common API for all import, export and transfer services. See `com.orchestranetworks.addon.dex` in the Java API for more information.

Bug fixes

- **[24772]** In EBX® Data Exchange Add-on configuration the path of EBX® application is specified by the data model.
- **[25618]** The tab name of the 'Object Class' table under the 'EBX® Data Exchange Add-on' configuration is incorrectly displayed.
- **[25977]** The defined modes on 'EBX® Data Exchange Add-on export XML' screen are not refreshed after canceling Export XML from the dataset level.
- **[25984]** There are inconsistent behaviors when exporting XML between table and dataset levels on a child dataset which has occulted records.

- **[25985]** A incorrect warning message is displayed when transferring data containing a computed field.
- **[25987]** A warning message is duplicated when transferring data containing a hidden primary key.
- **[26036]** Spelling mistakes are found in the error message when transferring the data without a mapping configuration.
- **[26288]** The number of processed records is incorrect when running the 'Validate models' service after generating the model.
- **[26347]** An incorrect error message is raised when exporting XML using the 'by default' mode at the dataset level.
- **[26361]** After changing the mapping type of field mappings, then transferring data, an error occurs.
- **[26562]** Exporting Excel with 'Load import mapping' leads to an incorrect result.
- **[26643]** There is an incorrect behavior in exporting CSV with the 'Primary key Export label' mode when the PK field is located at the end of the table.
- **[26683]** An unexpected exception occurs when using 'Field mapping transformation' of a complex terminal node.
- **[26685]** There is a spelling mistake in the warning message in transferring data using the 'No transfer' function.
- **[26751]** There is an unclear error message on the configuration screen in importing CSV when the list separator is null or empty.
- **[26767]** A CSV file can be imported if there is a unmapped field on the 'Mapping column' screen.
- **[26837]** The incorrect source field is loaded when using the 'Used as target' tab in the EBX® Data Exchange Add-on dataset.
- **[26865]** Transferring data with the Split mapping type and some hidden or read-only additional fields leads to incorrect behavior.
- **[26866]** No warning message is displayed in transferring data when using an Aggregate mapping type and a hidden additional field.
- **[26869]** An incorrect warning message is displayed when exporting data in XML format with hidden fields.
- **[26888]** There is a spelling mistake in the warning message when transferring data between terminal nodes without using Java Bean class.
- **[26940]** An incorrect error message is displayed in transferring data when using a wrong transformation function.
- **[27014]** An error occurs if you cancel the import process when importing Excel sheets by sequence.
- **[27284]** When launched from a workflow, the 'Import SQL' service fails if using a JBoss server.

51.34 **Release Note 2.2.7**

Release Date: July 6, 2017

Bug fixes

- **[26847]** With JBoss 6.x and above, an exception occurs when looking up the datasource in JNDI context.

51.35 **Release Note 2.2.6**

Release Date: May 19, 2017

New features

- It is now possible to export and transfer data on a snapshot.

51.36 **Release Note 2.2.5**

Release Date: April 18, 2017

New features

- It is now possible to import the number values as displayed in the cell while importing Excel.

Bug fixes

- **[24773]** A white page is shown during Import XML.
- **[24775]** Auto increment primary key should be ignored when 'Check empty or null primary key' is activated in the XML import with default mode.

51.37 **Release Note 2.2.4**

Release Date: March 31, 2017

Bug fixes

- **[25031]** When importing CSV or Excel, it is not possible to map a group's fields if group permissions are not Read-Write.

51.38 **Release Note 2.2.3**

Release Date: February 23, 2017

Bug fixes

- **[24431]** The method getOccurrentContext always returns ValueContext with values retrieved from a file.

51.39 **Release Note 2.2.2**

Release Date: January 23, 2017

Bug fixes

- **[21814]** Users without the permission to create records cannot use the EBX® Data Exchange Add-on import services.

51.40 **Release Note 2.2.1**

Release Date: December 16, 2016

New features

- It is now possible to lock or unlock the dataspace while exporting.
- You can now transfer data from many source tables to many target tables.

Java API

- It is now possible to disable the write access lock on the dataspace while executing SQL/XML export. See `DataExchangeExportSpec` in the Java API for more information.
- It is now possible to transfer from many source tables to many target tables via API. See `DataExchangeTransferSpec` and `TableMapping` in the Java API for more information.

Bug fixes

- **[23078]** Record is deselected on UI after cancelling an export.

51.41 **Release Note 2.2.0**

Release Date: November 18, 2016

New features

- Data can be imported from or exported to an external SQL database via a JNDI data source using default mapping.
- Creation and modification privileges can be defined on a profile's preferences.

Bug fixes

- **[21865]** An incorrect number of total records to be exported is displayed in the progress bar.
- **[22220]** All fields of a table are exported when executing 'XML export' on a tabular view.
- **[22838]** No error message displays when creating an Excel or CSV application type.

51.42 **Release Note 2.1.9**

Release Date: October 12, 2016

New features

- The new 'Include the reference sheet' option is available during the Excel export at the dataset level to determine whether the reference sheet 'Reference table mapping' will be added to the end

of an Excel file. This sheet contains information such as the sheet name, table label and table path that was used to detect the mapping between sheets and tables at import time.

- The new 'Include computed values' option is available to determine whether the computed values must be included during an Excel or CSV export.

Bug fixes

- **[22158]** Data import cannot be done when the primary key field of the imported file contains special characters.

51.43 **Release Note 2.1.8**

Release Date: August 4, 2016

New features

- The new 'Use language' option is available to determine whether the language and corresponding number format policy are applied during CSV export.

Java API

- It is possible to set the specific locale when executing Excel or CSV export via the API using the filter data function. See `ExportDataAccessSpec` in the Java API for more information.

Bug fixes

- **[21108]** Import CSV works improperly when importing a line which only contains separators.
- **[21189]** Cannot import Excel or CSV when there is a hidden table which links two additional tables.
- **[21205]** Error message displays when user cancels or closes the Excel import process at the dataset level.
- **[21222]** Cannot import an Excel file without style.

51.44 **Release Note 2.1.7**

Release Date: July 8, 2016

Bug fixes

- **[20965]** When the 'Check empty or null primary keys' option is disabled, primary and foreign key field mappings cannot be ignored.
- **[20982]** A foreign key field cannot be exported if its source record does not exist.
- **[21003]** Export works improperly when the dataset's default policy is set to 'Hidden'.

51.45 **Release Note 2.1.6**

Release Date: June 10, 2016

New features

- **Import Excel, CSV, XML file and transfer data**

  The primary key mapping and its data validation are handled by the new 'Check empty or null primary keys' property.

Java API

- The new methods `getRepository`, `getSession`, `getSourceTable` and `getReferencedTable` for transforming data before exporting Excel or CSV are added to the `TransformContextForExport` interface.
- The new methods `getRepository`, `getSession`, `getReferencedTable` and `getValueInSpreadsheet` for transforming data before importing Excel or CSV are added to the `TransformContextForImport` interface.
- The method `getTargetTable` has been removed from the `TransformContextForExport` interface.
- The Javadoc of the method `getTargetTable` is updated on the `TransformContextForImport` interface.

Bug fixes

- **[20270]** The number formats do not comply with EBX® formatting policies when exporting CSV.
- **[20587]** Improved French error messages when using EBX® Data Exchange Add-on to import CSV/Excel.
- **[20689]** The value of a hidden field is transferred between two datasets that share the same model.
- **[20691]** A hidden additional field is still aggregated when transferring data.
- **[20762]** DDM works improperly when the column name in the SQL script includes the word 'go'.

51.46 **Release Note 2.1.5**

Release Date: May 19, 2016

New features

- **Import Excel, CSV, XML file and transfer data**

  Null and empty values in the imported file are handled by the new 'Ignore the empty or null values' property.

- **Documentation improvement**

  Add JavaDoc for TransformerDefinition constructor.

  Add sample for Transformer.

Bug fixes

- **[19755]** DDM does not work properly when EBX® has more than two locales.
- **[20121]** The 'Import CSV' service does not work properly when the imported value contains separator character.

51.47 Release Note 2.1.4

Release Date: March 18, 2016

New features

- Ability to filter data when executing the Excel or CSV export via the API.

Bug fixes

- **[19516]** The source file is not released after being imported using 'import CSV' via the API.

51.48 Release Note 2.1.3

Release Date: February 26, 2016

New features

- Preferences can be defined as parameters in the user task declaration when importing or exporting Excel/CSV.

Bug fixes

- **[19157]** When it is declared in the 'Extension' table, the Transformation Java class does not load on the import configuration screen.
- **[19239]** [All browsers] Some labels on the export and import Excel/CSV configuration screens overlap when the language is set to French.

51.49 Release Note 2.1.2 fix 002

Release Date: January 18, 2016

Bug fixes

- **[19077]** The pre-loading and pre-validation on three dataspaces of 'EBX® Data Exchange Add-on' must be disabled upon start-up time.

51.50 Release Note 2.1.2 fix 001

Release Date: January 11, 2016

Bug fixes

- **[18993]** Disable the pre-loading and pre-validation on EBX® Data Exchange Add-on dataset at the start-up time of EBX®.

51.51 Release Note 2.1.2

Release Date: December 11, 2015

New features

- **Excel/CSV import and export**

  The CSV import and export feature has been improved and been brought into alignment with EBX® behavior. It can now use a list separator to separate values in a list.

  The CSV import feature now works with simple field lists, simple foreign key lists and complex foreign key lists.

  The Excel import feature can now work with complex foreign key lists.

- **Validator**

  A Java class can be used to specify criteria and validate data being imported, or transferred.

Bug fixes

- **[17936]** The wrong source field is filtered when using the 'Used as target' tab.
- **[17946]** The 'Split of string' transformation function failed when source data contains special characters.

51.52 **Release Note 2.1.1**

Release Date: October 14, 2015

New features

User interface improvements

- **Export Excel format**

  A new option allows you to select and export all tables in the current dataset.

  Table template options now display in tabs on the 'Export Excel' configuration screen, unless you are exporting a dataset.

- **Import Excel and CSV formats**

  The 'Import Excel' configuration screen contains a new option that allows you to remove redundant characters in the imported file's header.

  The 'Import Excel' and 'Import CSV' configuration screens contain a new option to use case-sensitive comparison in matching the imported file's header with the field label.

  The 'Ignore' button is now in front of the column mapping drop-down lists.

  The 'Starting position of table content' group's borders on the mapping page were removed to match the existing format. Additionally, its location was changed to the top of the screen.

  The 'Simulation' result screen is fixed to a specific height.

Bug fixes

- **[17633]** String value is trimmed if there are leading or trailing spaces or line breaks when exporting Excel 2007.

51.53 Release Note 2.1.0

Release Date: August 24, 2015

New features

- **Transformation functions**

  Transformation functions are used to adapt data values during exchange. They are now defined in a portfolio. This portfolio is fed from a publication of functions done by an API. At configuration time, you can customize the functions without any bespoke development. The Add-on provides you with pre-defined transformation functions (aggregate, split, concatenate, etc.) which you can enrich using the API.

  New transformation functions such as 'split', 'aggregation' and 'conversion' are now available to drive the export, import and transfer processes.

  A new transformation function called 'Cross-reference' is available to replace a source code field value with another value during export, import or transfer.

- **Data transfer**

  When a data mapping can be reused in both directions for data transfer ('source to target' and 'target to source') you no longer need to duplicate the configuration.

  A new option allows you to deactivate all triggers and constraints on tables during a data transfer.

- **XML format**

  The 'EBX® Data Exchange Add-on' header included by the add-on in exported XML files is now optional. You can export and import XML data without using this header.

  A new option allows you to deactivate all triggers and constraints on tables during XML import.

- **Excel – CSV formats**

  A new UI option is available that determines whether the 'Validate data before transforming data' option is active for Excel and CSV data formats.

- **Data mapping configuration**

  Export, import and transfer of multi-occurs complex data type is now possible.

  A new service called 'Clean all configurations' allows you to remove all the data mapping configurations to get an empty repository.

  A new service called 'Clean unused paths' allows you to clean up all deprecated paths in the 'Path' table.

  A new service called 'Report data mapping' is now available to get a report on the data mapping between source and target applications. This service is located on the 'Table mapping' and 'Field mapping' tables.

  New 'Data Hierarchy' views are now available on the 'Table' and 'Field' tables to get a clearer view of the data model structure by 'Application → Table → Field'. A new service called 'Refresh hierarchical views' allows you to initialize these views in case you already have an existing data mapping configuration (located on the 'Table' table).

- **Dynamic Data Modeling**

  You can now generate an EBX® Data Model from XML and DDL files.

There is a new option to automatically generate values for table and field labels when importing from an Excel file.

Java API

- API for data transfer and export XML is enriched to enable the use of a data filter. In this version, the filters are available at the API level only. In the next version, it will be possible to configure filters at the administration level.

  See `DataExchangeExportSpec` and `DataExchangeTransferSpec` in the Java API for more information.

- The method `addErrorMessage` and `setErrorMessages` have been removed from the class `DataExchangeExportResult` .

Bug fixes

- **[17386]** No error message displays when exporting a CSV or Excel file with an existing export preference.

51.54 **Release Note 2.0.11**

Release Date: July 9, 2015

Bug fixes

- **[17023]** When importing an xlsx file, if a function is presented on a cell of type String, the import fails due to an `exception` .

51.55 **Release Note 2.0.10**

Release Date: June 29, 2015

Bug fixes

- **[16854]** A `NullPointerException` may occur with some Excel files containing more than 32,767 lines.

51.56 **Release Note 2.0.9**

Release Date: May 6, 2015

Memory usage improvements

- The Excel import and export for Excel 2007 have been improved in order to be able to manage hundreds of thousands of records.

51.57 **Release Note 2.0.8**

Release Date: April 27, 2015

New updates

- It is now possible to import file with the extension is in upper-case such as XLSX.

51.58 **Release Note 2.0.7**

Release Date: March 17, 2015

New updates

- New UI option to force the import of data by making disable all triggers and constraints.
- Improve error messages in importing data.
- It is now possible to import records by ignoring mandatory foreign keys.
- It is now possible to ignore mandatory columns in the data mapping configuration.
- When a field is not mapped, the import data process is not stopped and raised a warning.

Bug Fixes

- **[14491]** Error messages are not written to error file in case of violating triggers or constraints.
- **[14864]** Import excel on dataset level has failed when a table sheet in excel file has no data to import.
- **[14918]** In case the option 'Download invalid data" is activated, the download error file does not contain the header in case of errors.

51.59 **Release Note 2.0.6**

Release Date: February 12, 2015

New updates

- Add the ability to choose the first line/column of the file to import when importing an Excel file in multiple tables.

51.60 **Release Note 2.0.5**

Release Date: January 26, 2015

Bug Fixes

- **[13150]** Support for ignoring the mapping of an auto increment primary key field.

51.61 **Release Note 2.0.4**

Release Date: December 8, 2014

Bug Fixes

- **[13642]** File is corrupted when exporting in Excel 2007 format.

51.62 Release Note 2.0.3

Release Date: December 4, 2014

Bug Fixes

- **[13580]** Trailing zero are added when importing a number into a string field.
- **[13484]** When exporting a String column, the column in Excel is not typed as "text" but is instead "general".

51.63 Release Note 2.0.2

Release Date: November 19, 2014

Bug Fixes

- **[13470]** Identical labels on different columns may lead to incorrect preferences loading.

51.64 Release Note 2.0.1

Release Date: October 10, 2014

New updates

- Support for exporting enumeration in the data model as an enumeration also in Excel file.

Bug Fixes

- **[12897]** "An unexpected exception occurred" is shown on result screen when executing import data from an Excel file.

  User has an Excel file that contains data of a dataset then executes Import Excel service to import data from this file to another dataset, there is a "An unexpected exception occurred" message shown on result screen.
- **[12947]** Scientific formatted cells are not well integrated at import if the target column has a String type.

  If the value of a cell is a big number (for example 9121323656454) with a scientific format, when it is mapped into a String column, the import result will be the scientific formatted value (9,E +12) instead of the "real" value.

51.65 Release Note 2.0.0

Release Date: September 12, 2014

New updates

- **User's operations for XML import-export and data transfer**

  Export and Import in XML based on a 'by default' data mapping.

  Export and Import in XML based on a user-defined data mapping.

Data transfer between EBX® tables relying on the same data model.

Data transfer between EBX® tables relying on different data models and based on a user-defined data mapping.

- **Extended operations to manage user-defined data mapping configuration**

  These operations are located in the EBX® Data Exchange Add-on's configuration dataset. They are used by IT specialists to create and maintain user-defined data mapping configurations between source and target applications. These configurations allow end-users to import, export and transfer data.

  Creation of the application portfolio for XML and EBX types.

  From an EBX type application, automatic generation of the Tables, Fields, Object Classes and Properties declaration.

  From an XML type application, automatic generation of the Tables and Fields declaration with XML paths (based on an XML sample that is provided as input parameter of the operation).

  Automatic detection of any misaligned items between a data mapping configuration and its underling EBX® application.

  Automatic data mapping configuration between two EBX type applications sharing the same Object Class and Property items.

51.66 **Release Note 1.3.0**

Release Date: December 16, 2013

New updates

- Support for transforming data before the processing an export to file or a data import.
- Display a progress bar when importing data from a file into the repository or exporting data to file.
- Support for importing a non-mandatory complex foreign key with empty data.

51.67 **Release Note 1.2.1**

Release Date: October 3, 2013

New updates

- Support for defining and exporting a template to an Excel file.

51.68 **Release Note 1.2.0**

Release Date: August 5, 2013

New updates

- Beta version of **API** `index`[API] .
- The option 'Only string' has been added for specifying a delimiter that is only applied for string data types when importing or exporting CSV files.
- Support for exporting a hyperlink of data type `anyURI` to an Excel file.

Known limitations

This version contains the following known limitations:

- Not support for exporting a record to a CSV/Excel file in a hierarchy view.

51.69 **Release Note 1.1.2**

Release Date: June 11, 2013

New updates

- Support for exporting to Excel 2007 files (.xlsx).
- Support for exporting the labels of the imported file columns when using "import preferences".

Bug Fixes

- **[6938]** No header included in exports of empty tables.

  Even though the header option is selected, the add-on exports an empty spreadsheet tab for empty tables. The headers should still be exported for empty tables so that import options that take the header into account are not affected just because the table has no rows of data.
- **[6939]** Delimiters should not be required for CSV exports.

  When exporting tables to CSV, the delimiter field is required. The default character is a quotation mark. This should not be required, as it is not necessarily desirable to have a delimiter surrounding each field. By default, there should be no delimiter, as that is how built-in CSV exports work.
- **[6940]** Fields of type `dateTime` cannot be formatted.

  Dates can be formatted for CSV export, but fields of type `dateTime` cannot. Both types should have customizable formats.
- **[6941]** CSV exports add extra separator characters at the end of lines.

  When performing a CSV export of a table, each row ends with a separator character. This is not consistent with what the built-in CSV export does, where there are only separators between fields. For example, `field1;field2;` is currently exported instead of `field1;field2` .

51.70 **Release Note 1.1.1**

Release Date: April 11, 2013

New updates

- Support for inputting non-predefined date formats when importing from and exporting to CSV files.
- Support for exporting to an Excel file using the mapping information from import preferences.
- Support for saving export configurations for reuse in subsequent exports.

51.71 **Release Note 1.1.0**

Release Date: February 25, 2013

In this version, the import and export features have been updated to be more user-friendly. When exporting the labels of primary or foreign keys to Excel, it is now possible to also define a hyperlink to the underlying record. The export of multiple tables to an Excel file and the import of multiple sheets from an Excel file are now supported.

New updates

- Export of the labels of primary or foreign keys to Excel and CSV files.
- Export of permalinks for primary or foreign keys to Excel files.
- Support for export of multiple tables to an Excel file.
- Support for the import of multiple sheets from an Excel file.
- Support for generating a CSV/ Excel file that contains records with errors during import of CSV/ Excel files. The user can download this file at the end of the import.
- When exporting, the labelling of foreign keys has been enhanced.
- Fix for the former limitation:

  When exporting an empty table or importing a file with a simulation error and not selecting the option 'Import only valid records', the waiting animation is not displayed properly.

51.72 **Release Note 1.0.0**

Release Date: October 31, 2012

Known limitations

This version contains the following known limitations:

- When two columns have the same name and have same type of validation error, only the first invalid column is included in the validation report message.
- When exporting an empty table or importing a file with a simulation error and not selecting the option 'Import only valid records', the waiting animation is not displayed properly.
- The user interface has the same known browser limitations as EBX®.

  See the section **Administration Guide >Installation & configuration** > **Supported environments** > **Browsing environments** in the main EBX® documentation for more information.