EBX5

Data Exchange Documentation

Version 2.0.9

EBX5 includes:

- software developped by the Apache Software Foundation - http://www.apache.org/
- software developped by the JDOM Project - http://www.jdom.org/
- software developped by the JSON Project - http://www.json.org/
- the Gagawa HTML Generator - http://code.google.com/p/gagawa/
- the Jericho HTML Parser - http://jericho.htmlparser.net/docs/index.html
- the H2 Database Engine - http://www.h2database.com/
- Font Awesome by Dave Gandy - http://fontawesome.io/

Table of contents

Data Exchange Documentation

Introduction

User-guide

Data Exchange Documentation

Introduction

CHAPTER 1

EBX5 Add-on for Data Exchange

This chapter contains the following topics:

1.1 Data Exchange at a glance

EBX5's Data Exchange add-on allows you to export and import data. You can initiate import and export in the following ways:

- A *'by default'* mode allows the end-users–with limited IT knowledge–to easily export and import data when source and target data structures are similar. Because this mode uses a direct mapping between the source and the target, a specific data mapping configuration isn't needed.
- When the source and target data structures are different, the extended mode allows IT staff to manage user-defined data mapping configurations. This mode is used to change the name of the field from the source to the target, to transform the data during the data flow processes, to aggregate or split data values, etc.

Data mapping configuration is based on basic concepts of Tables and Fields. When a company wants to strengthen business representation of data mapping configurations, Tables and Fields can be connected to business concepts also known as Object Classes and Properties, respectively. The addition of this business level improves the data lineage audit and encourages actual information governance.

For further information beyond the online help, refer to the User guide (PDF) and the release notes [p 77].

1.2 Functional scope

EBX5's Data Exchange Add-On relies on a repository that collects all data mapping configurations. This repository automatically updates when a user executes a mapping operation that invokes the default data mapping.

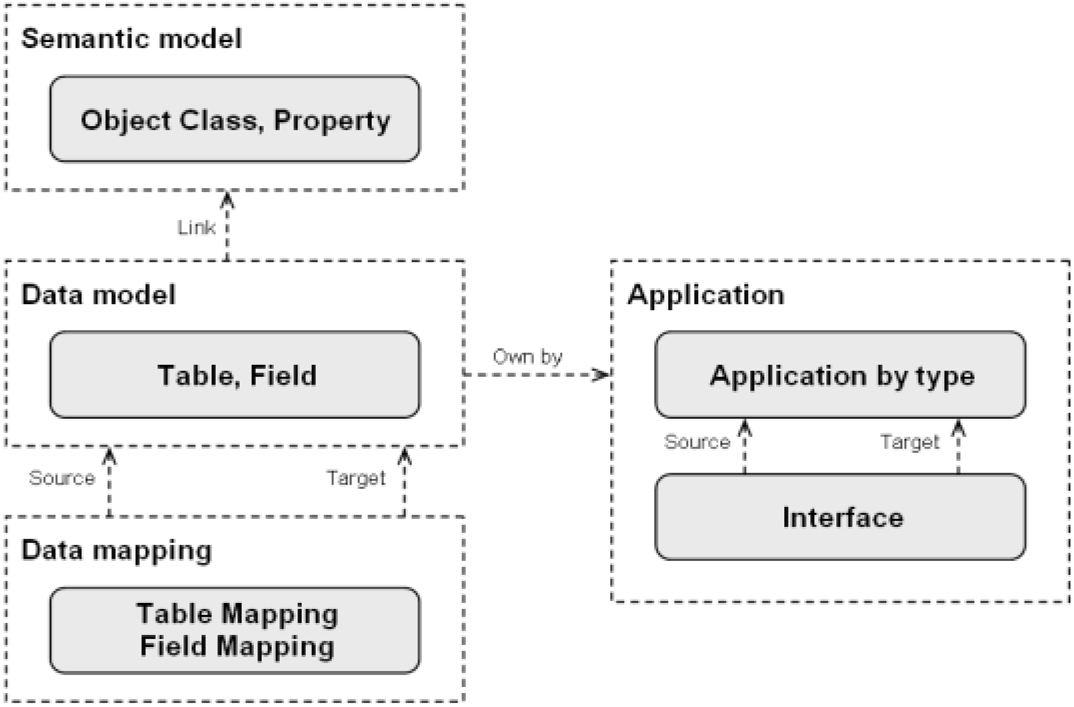
Semantic model

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. This vocabulary comes from the ISO11179 standard also used in the EBX5's 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.

Key definitions



1.3 User operations

At the table level, Data Exchange services allow you to export and import data in the CSV, Excel and XML formats.

At the data set level–in addition to being able to import and export in Excel and XML–Data Exchange services allow you to transfer data. The transfer copies data directly between EBX tables.

XML with the default data mapping

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.

The add-on automatically identifies which data mapping to apply in order to import the data into the table. When the *'Default XML'* is declared in the header of the imported XML file, then the default data mapping is applied. A UI result message provides an overview of the import process, as illustrated below.

XML with a user-defined data mapping

To export or import 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. 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. The add-on can automatically analyze your XML file to generate the XML configuration.

Data transfer

Data transfer is used to copy data from one EBX table to another. If the tables share the same data model, the add-on automatically manages the transfer.However, transferring data between tables that don't share the same data model requires a bespoke data mapping configuration.

Excel and CSV data flows

The Excel and CSV export and import processes are based on the previous version of Data Exchange, called *'Add-on for Microsoft® Excel'*. You can refer directly to the online help for this part of the add-on.

In the current version of Data Exchange Add-on, the user-defined configuration is not used for the Excel and CSV data flows.

1.4 **Advanced operations**

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 *'Data exchange'* data space under the EXB5 *'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.

1.5 **Configuring Data Exchange**

The Data Exchange repository for data mapping configurations is located in the *'Data exchange'* data set under the EBX5 *'Administration'* tab.The *'Application'* domain contains the portfolio of applications with the interface configuration.An interface is used to configure the source and the target applications involved in a data flow.

The *'Semantic model'* domain contains the Object Class and Property items used to link the Tables and Fields to the semantic layer. The Semantic model is not mandatory to configure data mapping. However, it allows you to improve data lineage and integration with Information governance.The *'Data model domain'* contains the configuration for Tables and Fields.The *'Data mapping'* domain contains the actual data mapping configuration applied at the table level, and then the field level.

1.6 **Known limitations**

- Multi-occurs complex data type, association object, selection node, user defined attribute (UDA).
- Split and aggregation of fields.
- Transformation of fields.
- Data lineage.
- Graphical view of the data mapping configurations.

1.7 **See also**

- Importing data from a CSV or Excel file [p 11]
- Exporting data to a CSV or Excel file [p 15]
- **Java API** `index`[API]
- Release notes [p 77]
- Data Exchange - User guide (PDF)

Disclaimer

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. Product names, logos, brands, and other trademarks featured or referred to are the property of their respective trademark holders. These trademark holders are not affiliated with Orchestra Networks, our products, or our website. They do not sponsor or endorse our product or any of our solutions.

CHAPTER **2**

Importing data from a CSV or Excel file

The process for importing data into EBX5 tables from a CSV or Excel file involves the following steps:

- Uploading the file
- Choosing the import configuration
- Mapping the spreadsheet to fields
- Simulating, then executing the import

To initiate the import of a file, navigate to the table into which the data will be imported in its data set in the **Data** area. Next, select the import option for the type of your file from the **Services** menu in the workspace.

This chapter contains the following topics:

2.1 Configuring the import

Several settings describing the format of the file must be configured before the import is performed.

Options specific to CSV files

Import mode	The options are as follows: • **Update and 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. • **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	The imported file can contain a header row or no header row. By default, the option of importing the file with a header row is used.
File encoding	The character encoding of the file being imported
Separator	The character used to separate cells in the spreadsheet file. By default, CSV files use **;** semicolons.
Delimiter	The character used to mark the beginning and the end of the content of a cell in the spreadsheet file. By default, CSV files use **"** double-quotes.
Only string	If selected, the delimiter is only used for string data types. Otherwise, it is applied to all data types.
Decimal symbol	The character to used to mark a decimal
Force the import	All triggers and constrains will be deactivated if this option is choosen. Otherwise, all triggers and constrains are activated.
Download file of invalid data	The wrong records will be saved a file. When users click the download link, the temporary file will be popup to save.
Extensions	List of transformer classes that will be used to transform data before importing into the repository.

Options specific to Excel files

Import mode	See the CSV import configuration option.
First row contains header	See the CSV import configuration option.
Force the import	See the CSV import configuration option.
Download file of invalid data	See the CSV import configuration option.
Extensions	See the CSV import configuration option.
Starting position of table content	Define row and column of starting cell of table content (in case of importing an Excel file into a table).

2.2 Mapping table form

Mapping table will be executed in the importing from a Excel file with multiple sheets. For each sheet into uploaded Excel file, you must select a suitable table in the dataset to map and you can also define row and column index of starting cell data of each sheet in case 'Choose starting position' option is activated for 'Each sheet'.

You can apply the row and column index of the starting cell for all sheets if 'Choose starting position' option is activated for 'All sheets'.

From this page, you can also save a preference with the current configuration and mapping. In order to apply the same mapping during a future import, you can select this saved preference on the file upload page from the 'Load preferences' pull-down menu.

2.3 Data structure mapping form

You must define how the data in the source spreadsheet maps to the data structure of the table into which you are importing the data.

For each field in the table into which you are importing, you must select a source column from the uploaded spreadsheet. If the spreadsheet being imported includes a header row, the add-on for Microsoft® Excel import automatically makes default selections if it finds matches between the names of the fields in the table and the headers of the spreadsheet.

Importing a CSV file or Excel file with single sheet, from this page, you can also save a preference with the current configuration and mapping. In order to apply the same mapping during a future import, you can select this saved preference on the file upload page from the 'Load preferences' pull-down menu.

2.4 Simulating and completing the import

Before actually executing the import, you can run a simulation of the process to detect whether any problems would occur during the import. This verification detects inconsistencies between cell values in the spreadsheet and the data types defined in the data model for the destination table.

You may choose to proceed with the import even if validation errors exist, by selecting the checkbox 'Import only valid records'. If validation errors exist and you proceed without selecting 'Import only valid records', the import will fail and no records will be imported.

2.5 **Supported data for import**

CSV imports

Supported data	Unsupported data
• Group of fields • Simple foreign keys • Complex foreign keys	• Lists of fields • Lists of groups • Lists of simple foreign keys • Complex foreign keys

Excel imports

Supported data	Unsupported data
• Group of fields • Simple foreign keys • Lists of simple foreign keys • Complex foreign keys	• Lists of groups • Lists of complex foreign keys

See also

Exporting data to a CSV or Excel file *[p 15]*

EBX5 Add-on for Microsoft® Excel main page *[p 7]*

CHAPTER **3**

Exporting data to a CSV or Excel file

To export data from a table in EBX5 to a CSV or Excel file, navigate to the table in its data set in the **Data** area. If you want to export the table data filtered by a view, ensure that the view is currently applied to the table. Next, select the export option for a file type from the **Services** menu in the workspace.

> **Note**
>
> The add-on for Microsoft® Excel supports exporting data from simple tabular views only. It is not possible to export data from multi-table views or hierarchical table views.

This chapter contains the following topics:

3.1 Configuring the export

In the export form, you must configure some options for the resulting spreadsheet file.

Options specific to CSV files

First row contains header	The exported file can contain a header row or no header row. By default, exporting file with a header row option is chosen
File encoding	The character encoding of the file to be exported
Separator	The character that will be used to separate cells in the spreadsheet file. By default, CSV files use **;** semicolons.
Delimiter	The character that will be used to mark the beginning and the end of the content of a cell in the spreadsheet file. By default, CSV files use**"** double-quotes.
Only string	If selected, the delimiter is only used for string data types. Otherwise, it is applied to all data types.
Date format	The date formatting will be used to parse the date values in the exported file. By default, CSV files use the format "MM-dd-yyyy". Users can enter a custom date format by choosing the option "Other" from the list and inputting the date format in the textbox. Date format should be declared using the characters [yMwWDdE ._,:|/-].
Date/time format	The date/time formatting will be used to parse the date/time values in the exported file. By default, CSV files use the format "MM-dd-yyyy'T'HH:mm:ss". Users can enter a custom date/time format by choosing the option "Other" from the list and inputting the desired date/time format in the textbox. Date time format should be declared using the characters [ymMwWdDEHhsa ._,:|/-].
Language	The language will be used in the exported file. By default, CSV files use the default language of EBX session.
Primary key	The exported file will add more an information column of the primary key. The header of the column is the label of the table.
Foreign key	The exported file will add more an information column of the foreign key. The header of the column is its name.
Load export preference	List of export preferences. All columns of the table will be exported based on configuration of the selected export preference.

Extensions	List of transformer classes that will be used to transform data before exporting to file
Save preference	Save the export configuration to reuse later.

Options specific to Excel files

Options specific to export a table to Excel file

Save as type	With "Excel 97-2003" option, the exported file will be saved as .xls file extension. With "Excel 2007" option, the exported file will be saved as .xlsx file extension.
First row contains header	The exported file can contain a header row or no header row. By default, exporting file with a header row option is chosen
Primary key	With "export label" option, the exported file will add more an information column of the primary key. The header of the column is the label of the table. With "permalink" option, the exported file will add more an information column of the primary key under hyperlink which redirects to a specific record.
Foreign key	With "export label" option, the exported file will add more an information column of the foreign key. The header of the column is its name. With the "permalink" option, the exported file will add more an information column of the foreign key under hyperlink which redirect to a specific record. With "permalink" option, the exported file will add more an information column of the foreign key under hyperlink which redirects to a specific record.
Choose the mapping	The exported file will be exported based on the selected mapping With "Default export" option, all columns of the table will be exported based on their orders. When "Default export" option is checked, "Load import mapping" option will be forbidden and conversely. With "Load import mapping" option, all columns of the table will be exported based on mapping of the selected import preference. When "Load import mapping" option is checked, "Default export" option, all specify options for primary key and foreign key (export label, permalink) will be forbidden.
Load import mapping	List of import preferences whose mapping will be use in exporting.

Load export preference	List of export preferences. All columns of the table will be exported based on configuration of the selected export preference.
Extensions	List of transformer classes that will be used to transform data before exporting to file
Table template	Define a template for title, subtitle, header and content of a exported table. You can customize formatting exported data including name, size, style, color of font, background color or adding borders to cells.
Table border	Adding border to cells. By default, no border option is chosen.
Title template	Define title, its starting position and customize its formatting including name, size, style, color of font and background color.
Subitle template	Define subtitle, its starting position and customize its formatting including name, size, style, color of font and background color.
Header template	Define starting position of table header and customize its formatting including name, size, style, color of font and background color.
Content template	Define starting position of table content and customize its formatting including name, size, style, color of font and background color.
Save preference	Save the export configuration to reuse later.

Options specific to export the multipletables to Excel file

Save as type	See the table export configuration option.
First row contains header	See the table export configuration option.
Primary key	See the table export configuration option.
Foreign key	See the table export configuration option.
Choose the tables to export	The selected tables will be exported into multiple sheets in the excel file. The label of a selected table is the name of a sheet in the excel file.
Choose the mapping	See the table export configuration option. However, when "Load import mapping" option is checked, "Choose the tables to export" option will be also forbidden.
Load import mapping	See the table export configuration option.
Load export preference	See the table export configuration option.
Extensions	See the table export configuration option.
Save preference	See the table export configuration option.

3.2 Running the export

Click the **Export** button for the form to finalize the export. The export function will pop-up a browser download dialog for the spreadsheet file containing the exported data. You can save the file to your local file system.

3.3 Supported data for export

Supported data	Unsupported data
• Group of fields • Simple foreign keys • Lists of simple foreign keys • Complex foreign keys • Lists of complex foreign keys	• Lists of groups

3.4 Known limitations

- The export of hyperlinks for a list of foreign keys is not supported.
- The export for a list of complex foreign keys is not supported when using import mapping.

See also

Importing data from a CSV or Excel file *[p 11]*

EBX5 Add-on for Microsoft® Excel main page *[p 7]*

User-guide

CHAPTER **4**

Overview

EBX5's Data Exchange add-on allows you to export and import data. You can initiate import and export in the following ways:

- A 'by default' mode allows the end-users-with limited IT knowledge-to easily export and import data when source and target data structures are similar. Because this mode uses a direct mapping between the source and the target, a specific data mapping configuration isn't needed.
- When the source and target data structures are different, the extended mode allows IT staff to manage user-defined data mapping configurations. This mode is used to change the name of the field from the source to the target, to transform the data during the data flow processes, to aggregate or split data values, etc.

Data mapping configuration is based on basic concepts of Tables and Fields. When a company wants to strengthen business representation of data mapping configurations, Tables and Fields can be connected to business concepts also known as Object Classes and Properties, respectively. The addition of this business level improves the data lineage audit and encourages actual information governance.

Special notation key

Special notation key:	
✓	Important recommendation for the use of a feature.
✗	This feature is no yet available in the current release.

CHAPTER **5**

Overview of updates

This chapter contains the following topics:

5.1 GA 2.0.0

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 EBX5 tables relying on the same data model.
- Data transfer between EBX5 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 Data Exchange's configuration data set. 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 EBX5 types.
- From an EBX5 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 EBX5 application.
- Automatic data mapping configuration between two EBX5 type applications sharing the same Object Class and Property items.

CHAPTER **6**

1. Functional scope

This chapter contains the following topics:

6.1 **1.1. Basic use case**

EBX5's Data Exchange Add-On relies on a repository that collects all data mapping configurations. This repository automatically updates when a user executes a mapping operation that invokes the default data mapping. For instance, during export execution for the 'Client' table in the 'Sales' data set, the Data Exchange repository is enriched with new source and target 'Sales' applications of EBX5 and 'Default XML' types, respectively. The data mapping configuration between the source fields (EBX5 paths) and the target fields (XML nodes) are created automatically. The next time this export process is executed, the repository updates automatically if needed (changes in the data structure).

This data can be reused to create user-defined data mapping configurations. Indeed, the repository can be enriched manually if default mappings do not meet business requirements. For example, when the names of the metadata from the source are different from the target, a user-defined data mapping configuration must be declared.

6.2 **1.2. 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 EBX5's 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:

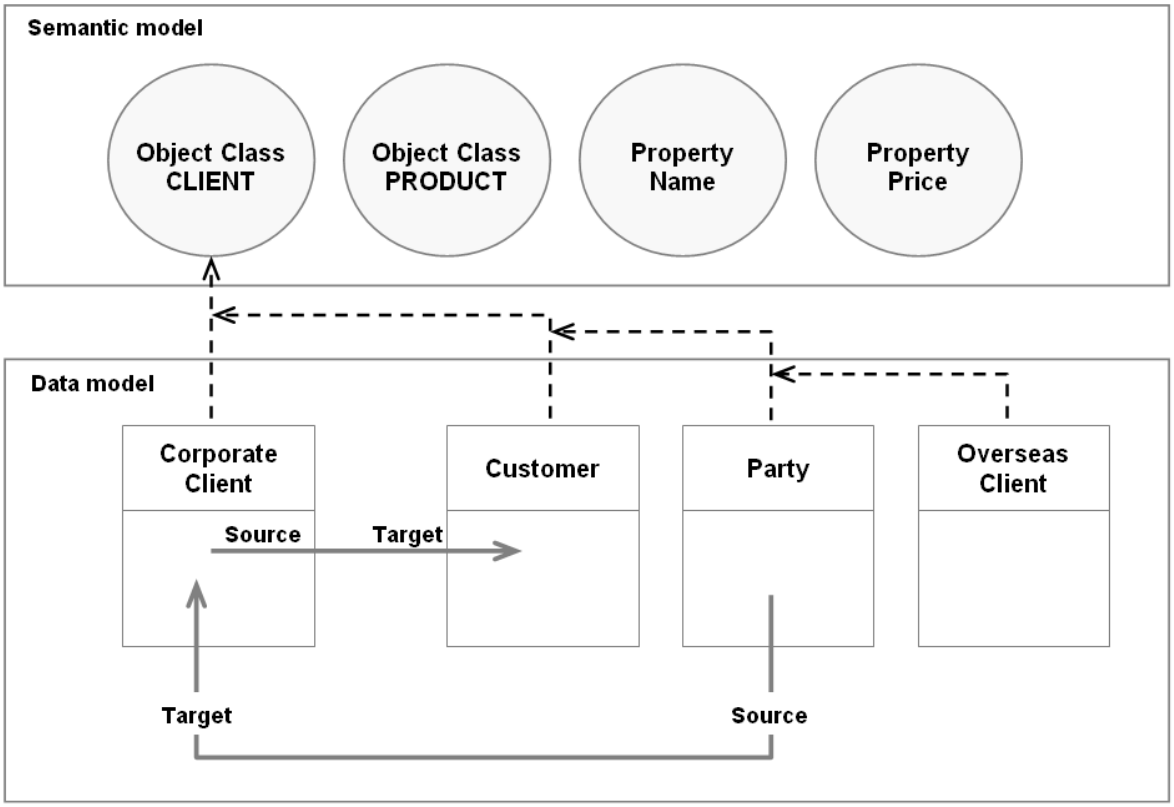
- Business data lineage.
- Automatic data mapping.
- Integration with the governance process.

1.2.1. 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.

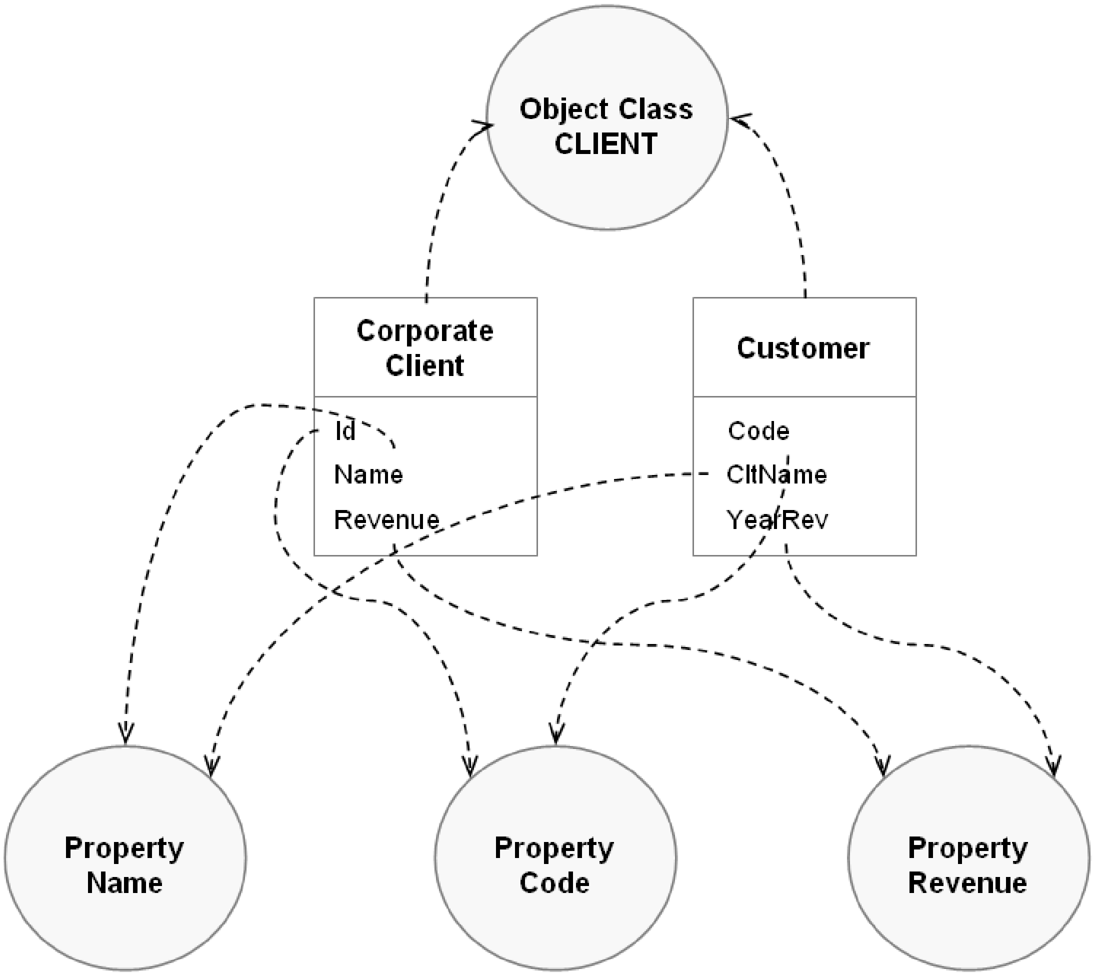


Special notation:	
✖	In the current version of the add-on, it is possible to link the Tables and Fields with the Object Class and Properties items, but the UI to display the data lineage is not yet available.

1.2.2. 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.



Special notation:	
✓	You can refer to the appendix 'Automatic data mapping for transferring data' to get a use case relying on a semantic data model.

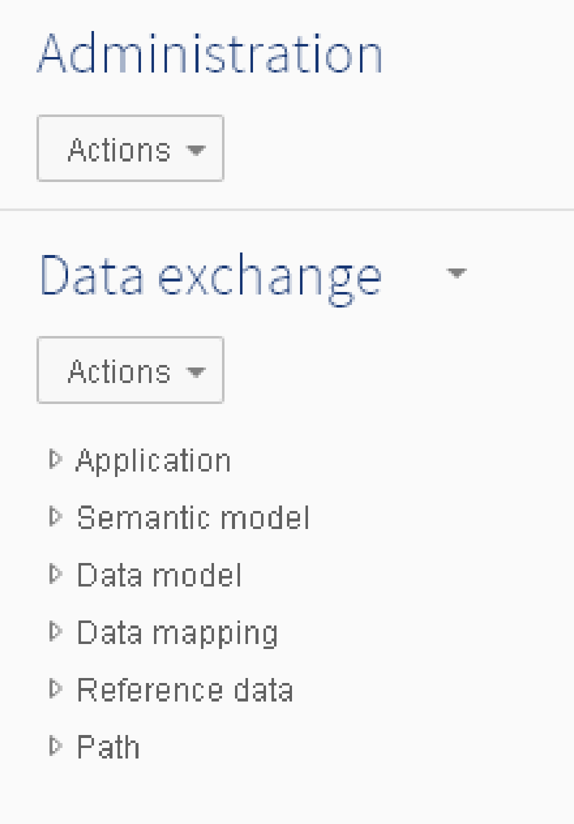
1.2.3. Integration with the governance process

In the Information governance field, EBX5's Information Governance add-on allows you to manage all metadata of any data asset, such as: data models, workflow, rules, data spaces, 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 a export process in Data Exchange.

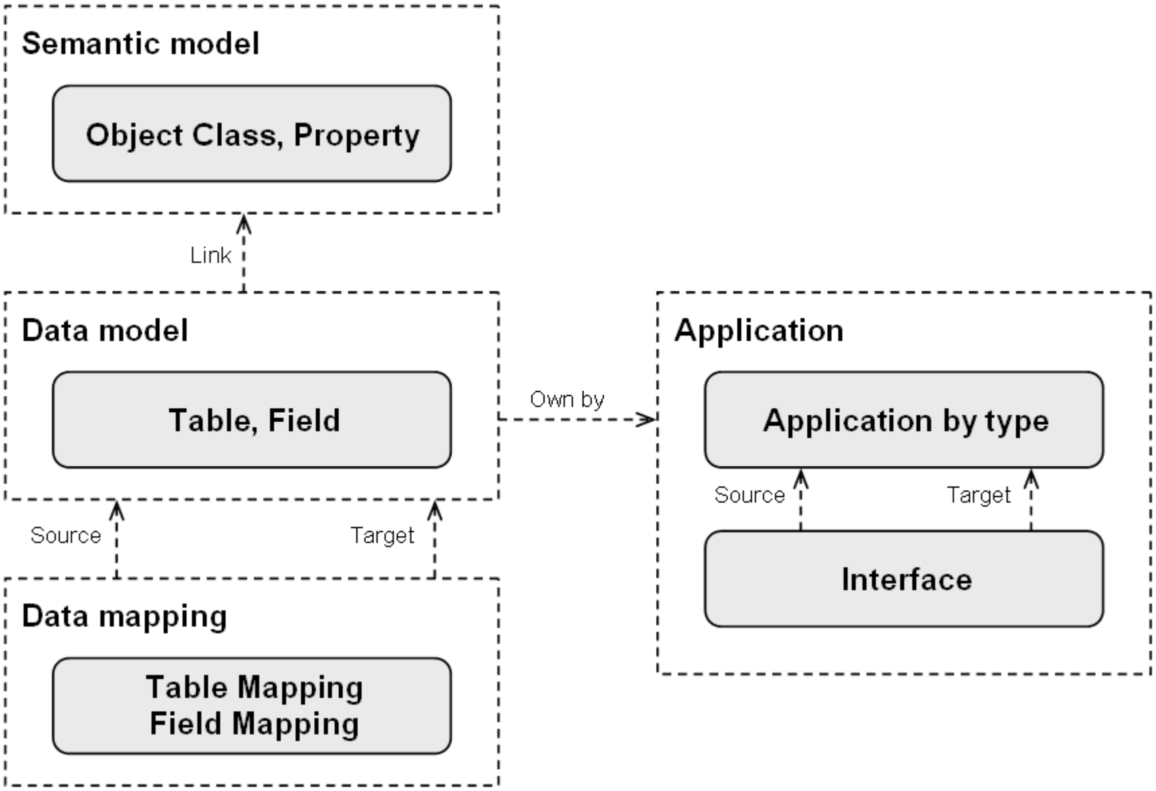
Special notation:	
✖	The integration between the add-ons Information Governance and Data Exchange will be available in the further version.

6.3 **1.3. Key definitions**



The table on the next page defines Data Exchange repository key concepts.

This repository is located under the EBX5 'Administration' tab in the 'Data exchange' data space. It contains the following domains: Application, Semantic model, Data model and Data mapping.



Concept	Definition
Semantic model	Using the semantic model is not mandatory to create data mapping configurations. However, it provides an improved data lineage view. Also, automatic data mapping between tables and fields-sharing the same Object Classes and properties-uses the semantic model.
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 to many 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 to many Object Classes and Fields.
Data model	A data model contains the Tables and Fields that comprise an application. If it is an EBX5 type application (referenced as a data set), it corresponds to the logical data model. When it is an XML type application, it corresponds to the XML paths.
Table	When referring to an EBX5 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 EBX5 type application, this is the field in the related logical data model. When referring to an XML type application, this is an XML path.
Application	The application level declares the portfolio of source and target applications involved in the import, export and transfer of data.
Application by type	An application is declared for one to many format types: • EBX5: the application corresponds to a data model in EBX5—referenced through a data set.

Concept	Definition
	• 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.
Interface	An interface is comprised of a source and target application and enables data flow between the two. 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 XML import or export.
Data mapping	This level contains the actual mapping configuration between tables and fields.
Table mapping	Defines the source and target tables.
Field mapping	Defines the sources and target fields.

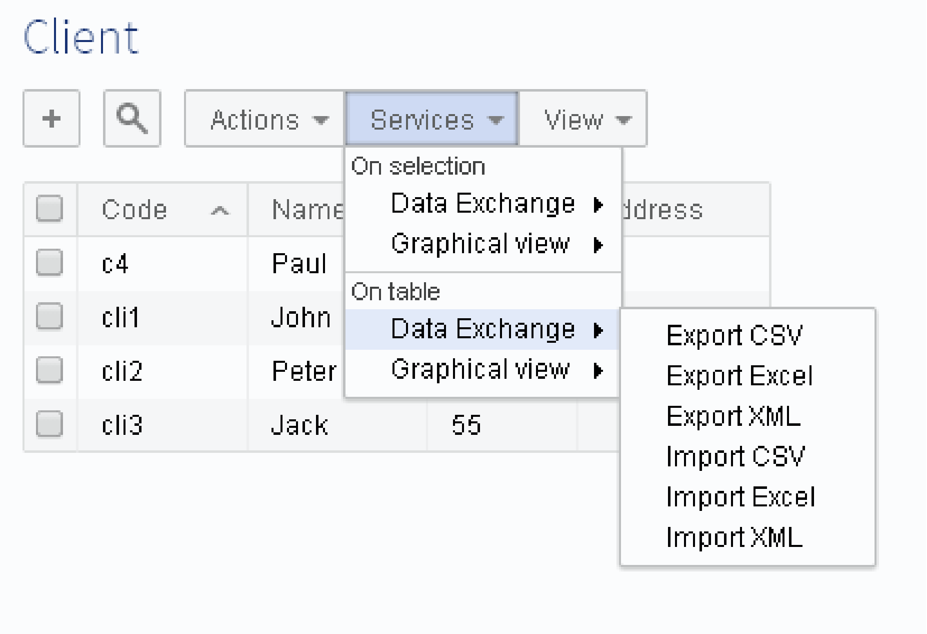
CHAPTER **7**

2. User operations

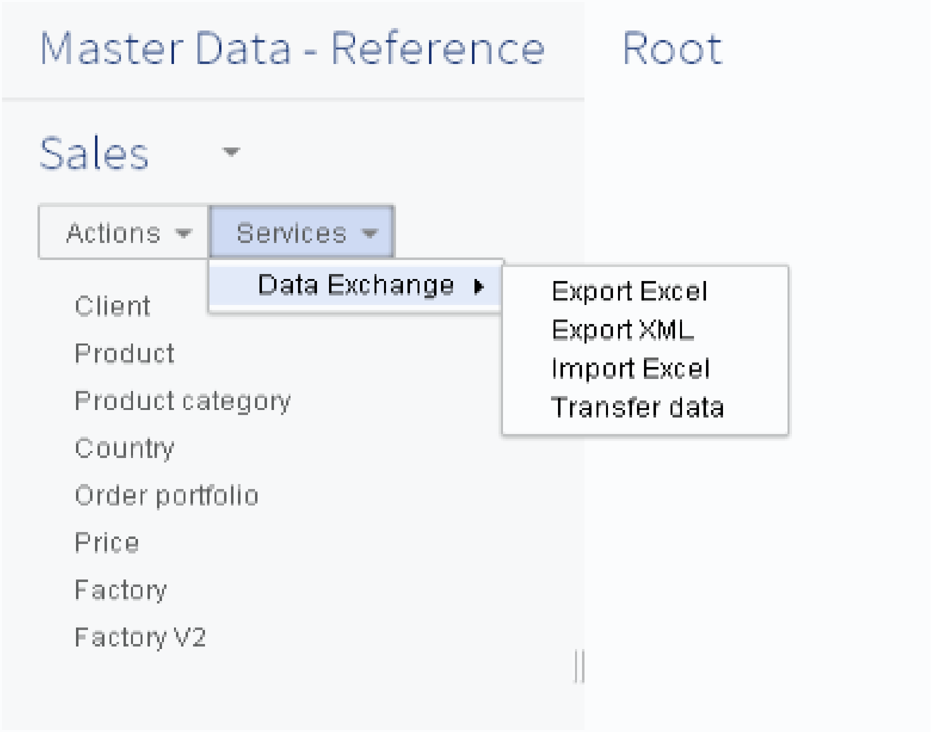
This chapter contains the following topics:

7.1 **2.1. Data Exchange services**

At the table level, Data Exchange services allow you to export and import data in the CSV, Excel and XML formats.



At the data set level-in addition to being able to import and export in Excel and XML-Data Exchange services allow you to transfer data. The transfer copies data directly between EBX5 tables.



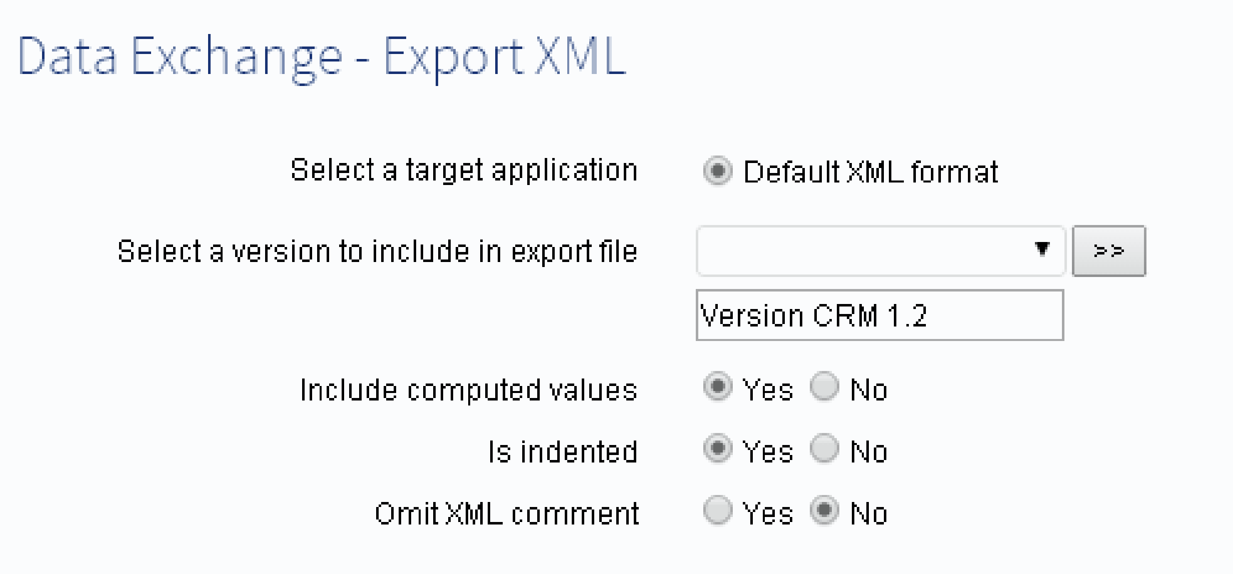
7.2 **2.2. XML with the default data mapping**

The following sections describe options when you use the 'Data Exchange' 'Export XML' and 'Import XML' services.

2.2.1. 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).



Data Exchange - Export XML

Select a target application ● Default XML format

Select a version to include in export file ▼ >>

Version CRM 1.2

Include computed values ● Yes ○ No

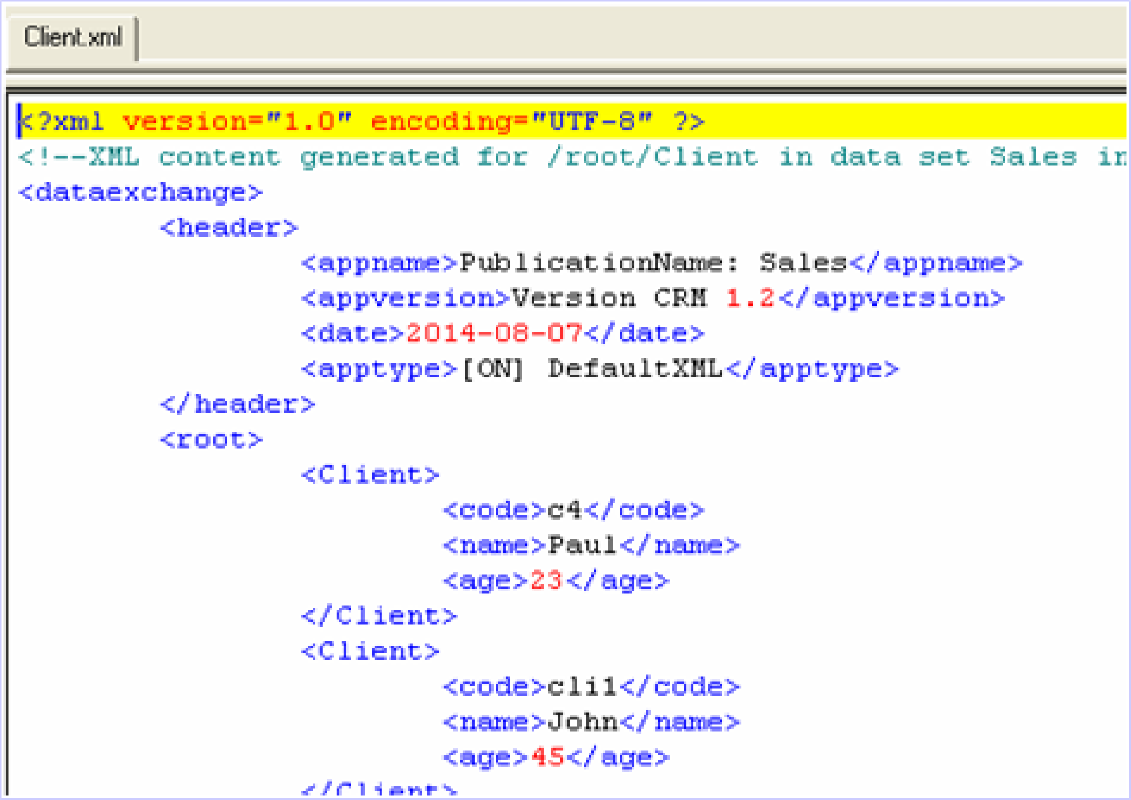
Is indented ● Yes ○ No

Omit XML comment ○ Yes ● No

Option	Description
Select a version to include into the exported file	The version is for information only. It is inserted in the exported file as the 'appversion' XML tag.
Include computed value	Specifies whether or not computed values must be included in the export.
Is indented	Specifies whether or not the file should be indented to improve readability.
Omit XML comment	Specifies whether or not the generated XML comment that describes the data location and export date should be omitted from the file.

The exported XML file contains a standard header with the following data:

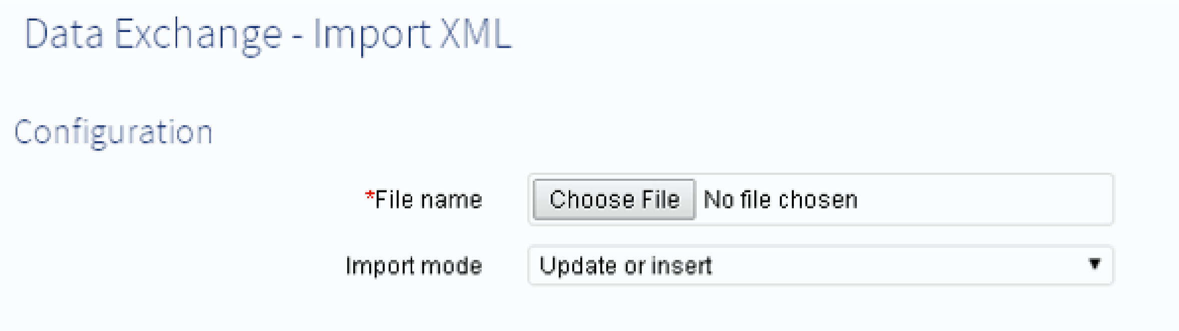
- The name of the application. By default, this is the name of the data set in which the table is located (a timestamps value is also added to build the name).
- The version that has been selected during the export configuration. If no specific version was specified, this will be blank.
- The export date.
- The type of the XML export, either 'Default XML' or 'XML' when a user-defined data mapping configuration is used.



```
<?xml version="1.0" encoding="UTF-8" ?>
<!--XML content generated for /root/Client in data set Sales in
<dataexchange>
        <header>
                <appname>PublicationName: Sales</appname>
                <appversion>Version CRM 1.2</appversion>
                <date>2014-08-07</date>
                <apptype>[ON] DefaultXML</apptype>
        </header>
        <root>
                <Client>
                        <code>c4</code>
                        <name>Paul</name>
                        <age>23</age>
                </Client>
                <Client>
                        <code>cli1</code>
                        <name>John</name>
                        <age>45</age>
                </Client>
```

2.2.2. XML import

Two import modes are available: 'Update or insert' or 'Replace all content'.

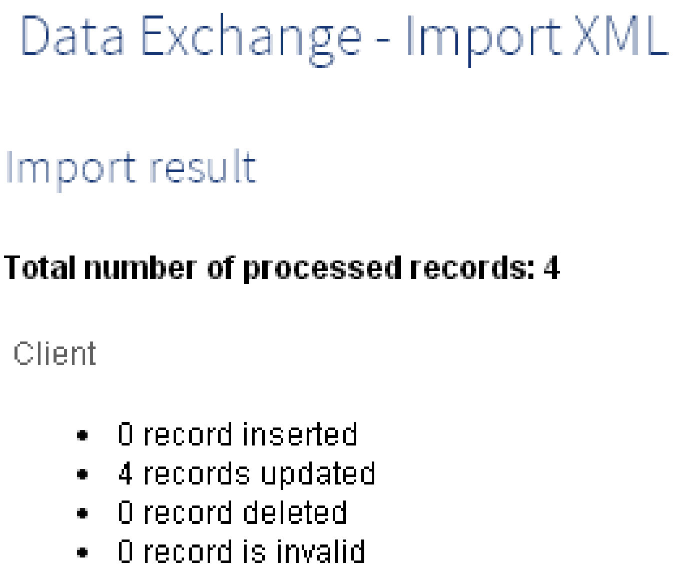

Data Exchange - Import XML

Configuration

*File name: Choose File No file chosen

Import mode: Update or insert ▼

The add-on automatically identifies which data mapping to apply in order to import the data into the table. When the 'Default XML' is declared in the header of the imported XML file, then the default data mapping is applied. A UI result message provides an overview of the import process, as illustrated below.



Data Exchange - Import XML

Import result

Total number of processed records: 4

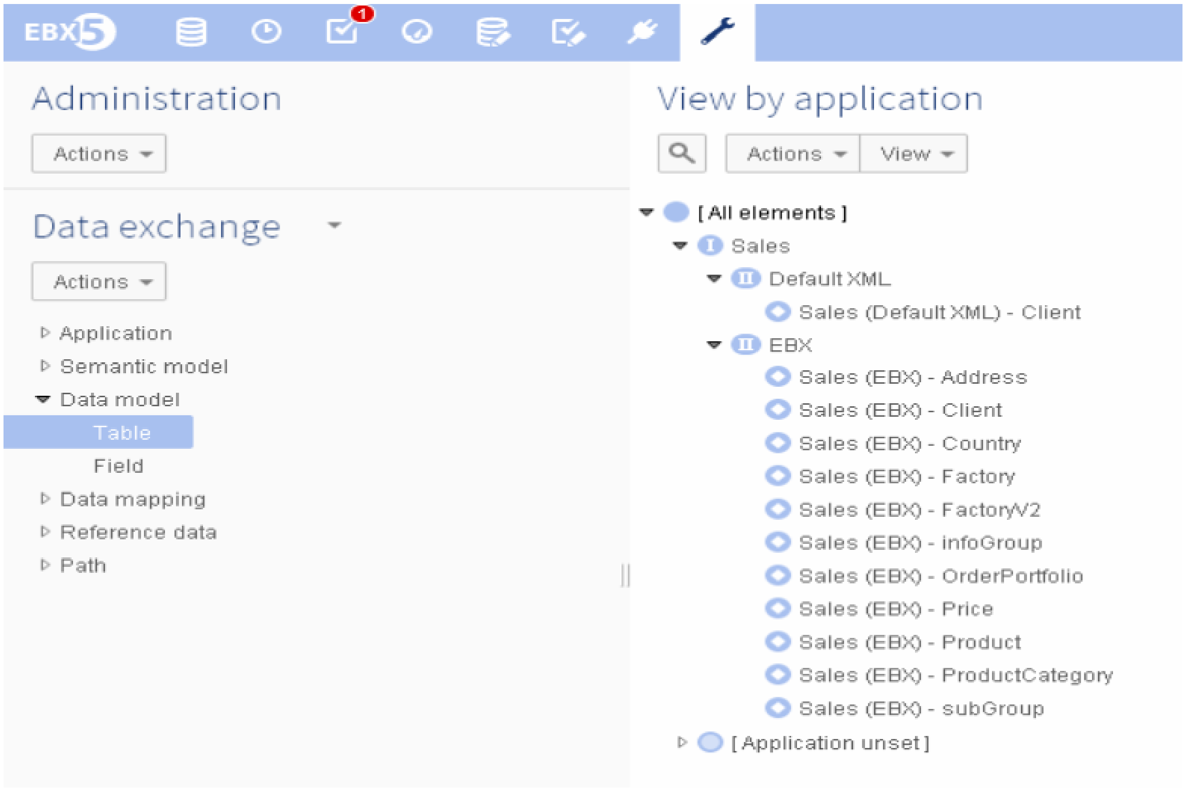
Client

- 0 record inserted
- 4 records updated
- 0 record deleted
- 0 record is invalid

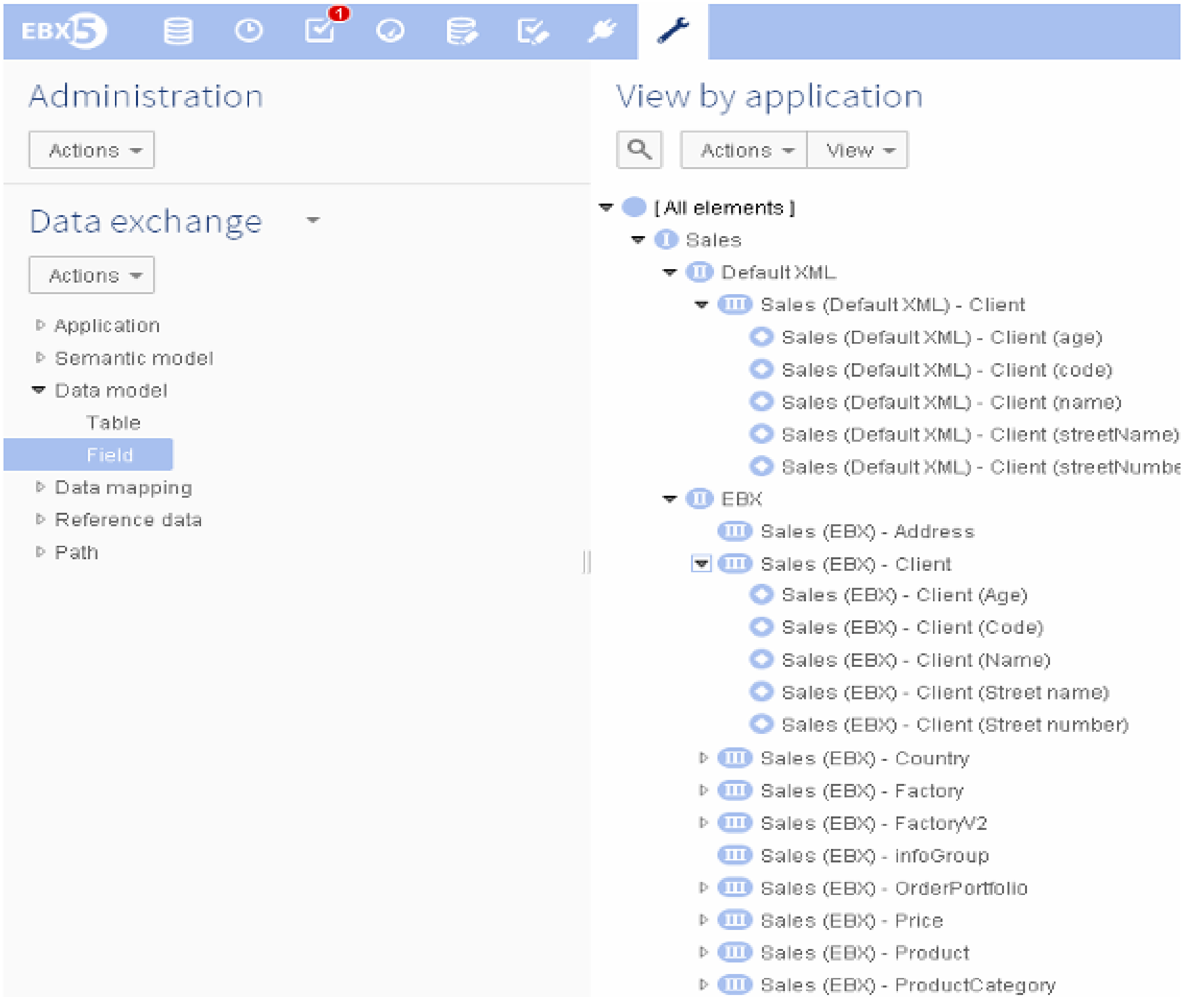
2.2.3. The default XML data mapping configuration

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 Data Exchange repository. You can access to this repository from the 'Administration' tab's 'Data Exchange' data space.



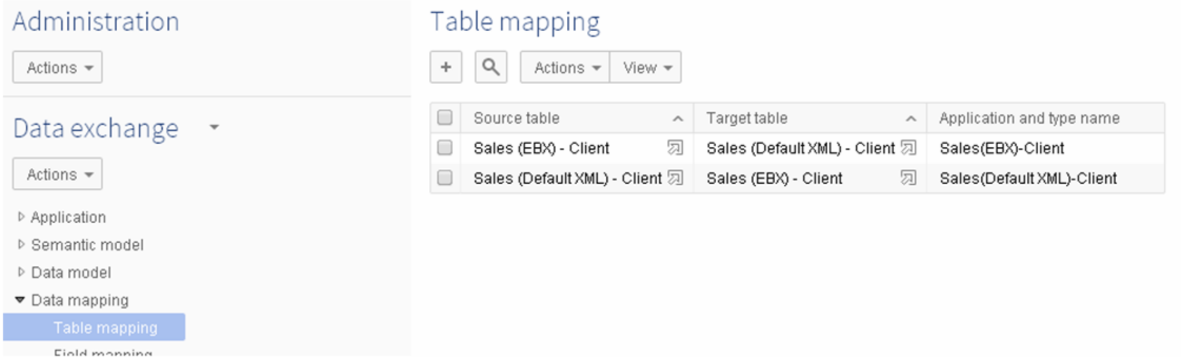
The 'Sales' (name of the data space where the table is located) application is created with the two application types, 'Default XML' and 'EBX5'. For the 'Default XML' type, only the exported table is declared ('Client'). For the 'EBX5' type, all tables and field groups located in the data space are declared.


EBX5
1
Administration
Actions
Data exchange
Actions
Application
Semantic model
Data model
Table
Field
Data mapping
Reference data
Path
View by application
Actions
View
[All elements]
Sales
Default XML
Sales (Default XML) - Client
Sales (Default XML) - Client (age)
Sales (Default XML) - Client (code)
Sales (Default XML) - Client (name)
Sales (Default XML) - Client (streetName)
Sales (Default XML) - Client (streetNumbe
EBX
Sales (EBX) - Address
Sales (EBX) - Client
Sales (EBX) - Client (Age)
Sales (EBX) - Client (Code)
Sales (EBX) - Client (Name)
Sales (EBX) - Client (Street name)
Sales (EBX) - Client (Street number)
Sales (EBX) - Country
Sales (EBX) - Factory
Sales (EBX) - FactoryV2
Sales (EBX) - infoGroup
Sales (EBX) - OrderPortfolio
Sales (EBX) - Price
Sales (EBX) - Product
Sales (EBX) - ProductCategory

The declaration of the XML nodes for the 'Client' table is done with the default XML naming convention (direct reuse of the naming from the table in EBX5).

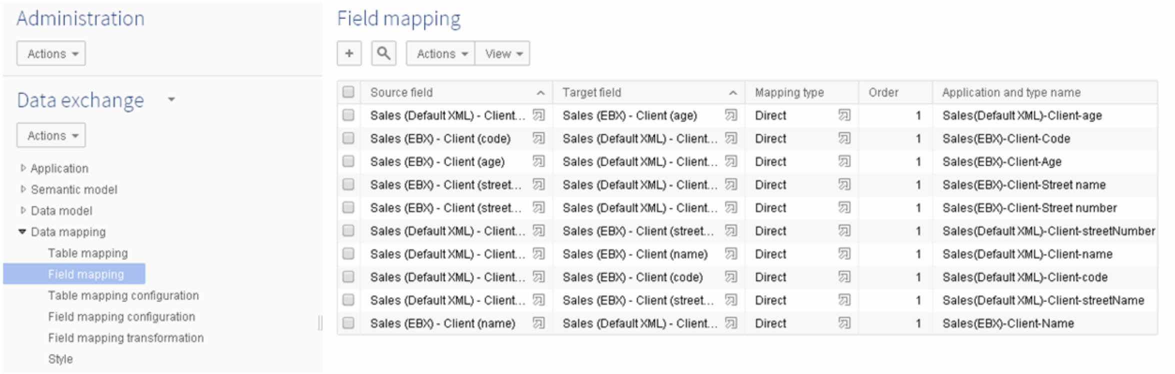
The declaration of the fields for every table in EBX5 is also performed.

The 'Table mapping' is automatically provided. The source table has been declared as EBX5 for the export process. And the source table has been declared as 'Default XML' for the import process.



□	Source table	Target table	Application and type name
□	Sales (EBX) - Client	Sales (Default XML) - Client	Sales(EBX)-Client
□	Sales (Default XML) - Client	Sales (EBX) - Client	Sales(Default XML)-Client

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



	Source field	Target field	Mapping type	Order	Application and type name
□	Sales (Default XML) - Client...	Sales (EBX) - Client (age)	Direct	1	Sales(Default XML)-Client-age
□	Sales (EBX) - Client (code)	Sales (Default XML) - Client...	Direct	1	Sales(EBX)-Client-Code
□	Sales (EBX) - Client (age)	Sales (Default XML) - Client...	Direct	1	Sales(EBX)-Client-Age
□	Sales (EBX) - Client (street...	Sales (Default XML) - Client...	Direct	1	Sales(EBX)-Client-Street name
□	Sales (EBX) - Client (street...	Sales (Default XML) - Client...	Direct	1	Sales(EBX)-Client-Street number
□	Sales (Default XML) - Client...	Sales (EBX) - Client (street...	Direct	1	Sales(Default XML)-Client-streetNumber
□	Sales (Default XML) - Client...	Sales (EBX) - Client (name)	Direct	1	Sales(Default XML)-Client-name
□	Sales (Default XML) - Client...	Sales (EBX) - Client (code)	Direct	1	Sales(Default XML)-Client-code
□	Sales (Default XML) - Client...	Sales (EBX) - Client (street...	Direct	1	Sales(Default XML)-Client-streetName
□	Sales (EBX) - Client (name)	Sales (Default XML) - Client...	Direct	1	Sales(EBX)-Client-Name

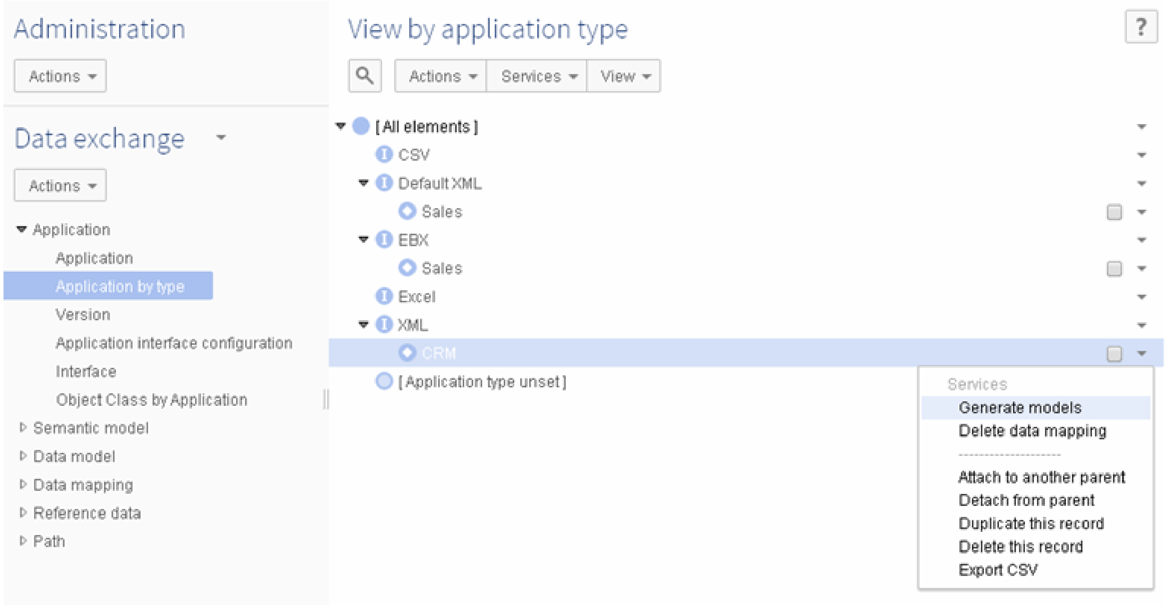
7.3 2.3. XML with a user-defined data mapping

The following section describes a user-defined data mapping.

2.3.1. XML export

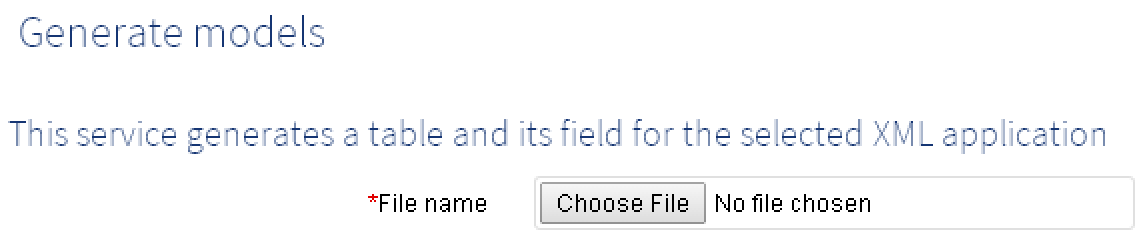
To export an XML file with a data structure that is not compliant with the 'Default XML' type, you can manually configure a user-defined data mapping.

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 'CRM' application. Since the type is XML (not 'Default XML'), the Data Exchange '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.

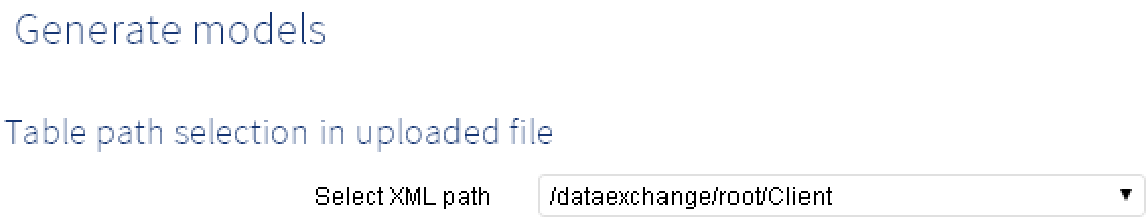


Generate models

This service generates a table and its field for the selected XML application

*File name: Choose File No file chosen

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

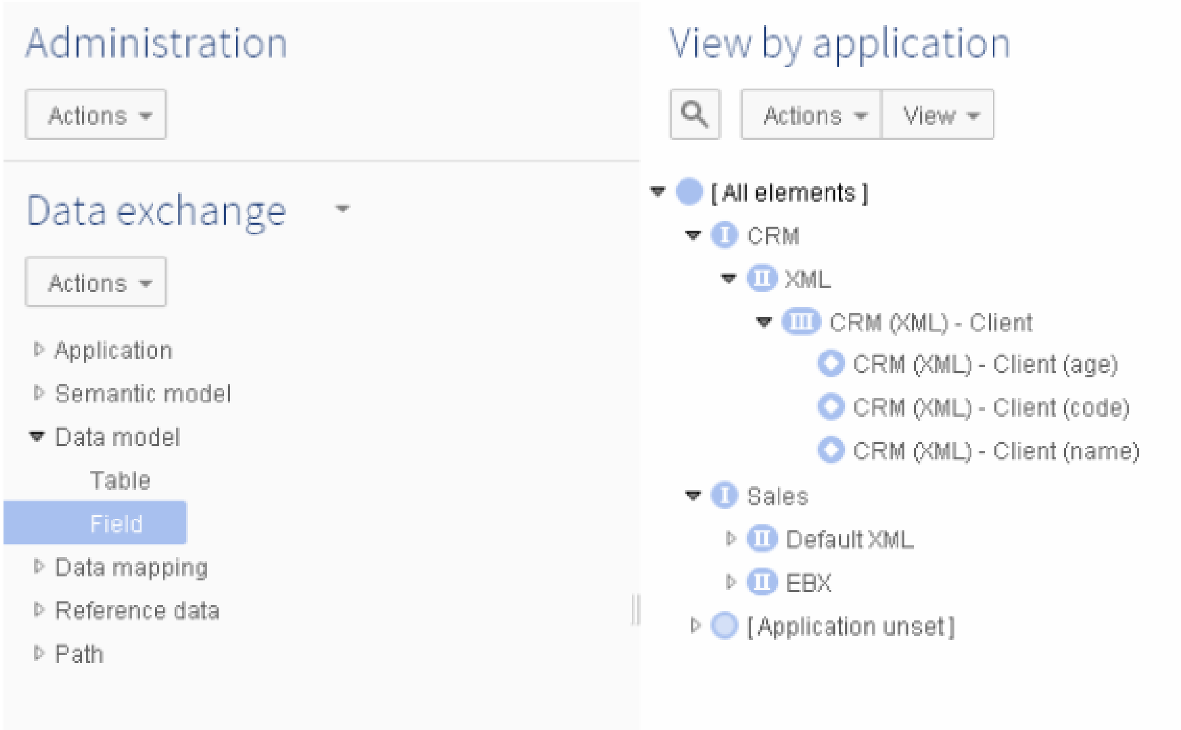


Generate models

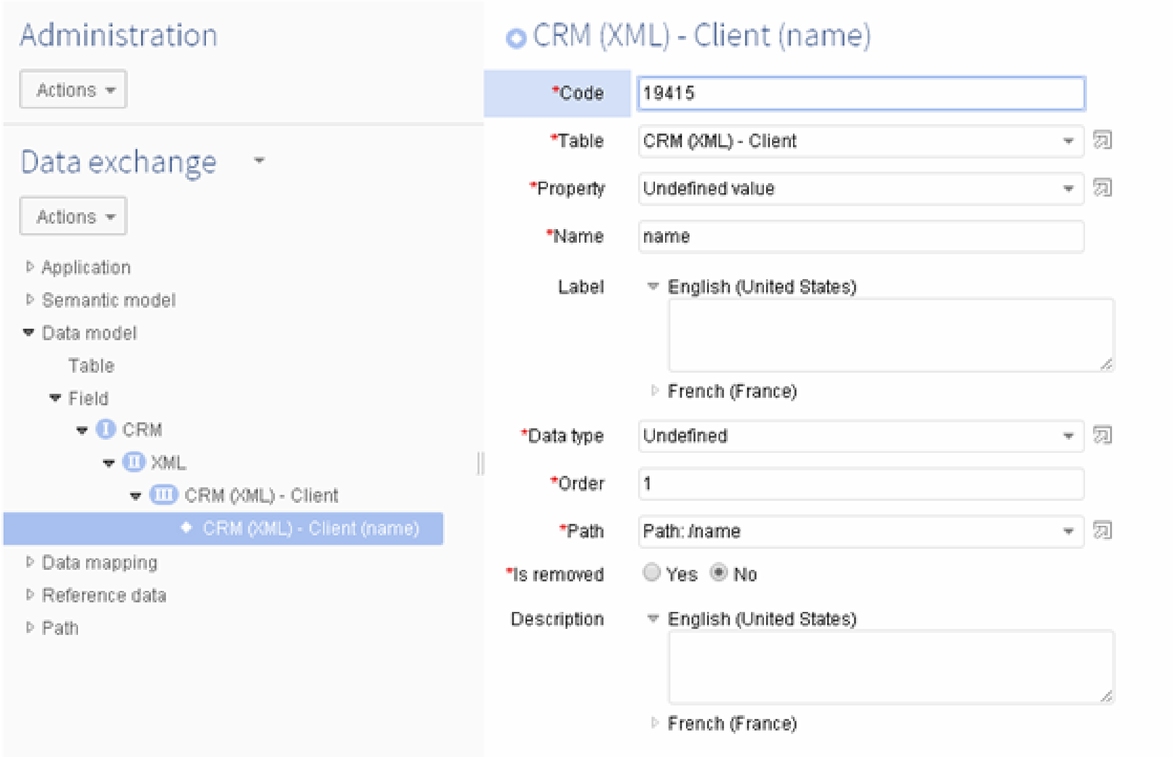
Table path selection in uploaded file

Select XML path: /dataexchange/root/Client

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



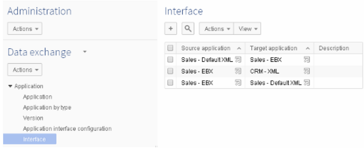
The configuration of a field with its XML path can be adapted manually as illustrated below. You can change the name, the path, etc. Conversely, in the case of a configuration based on the 'Default XML', the names and paths cannot be modified and are under add-on control.



CRM (XML) - Client (name)

*Code	19415
*Table	CRM (XML) - Client
*Property	Undefined value
*Name	name
Label	English (United States) French (France)
*Data type	Undefined
*Order	1
*Path	Path: /name
*Is removed	○ Yes ● No
Description	English (United States) French (France)

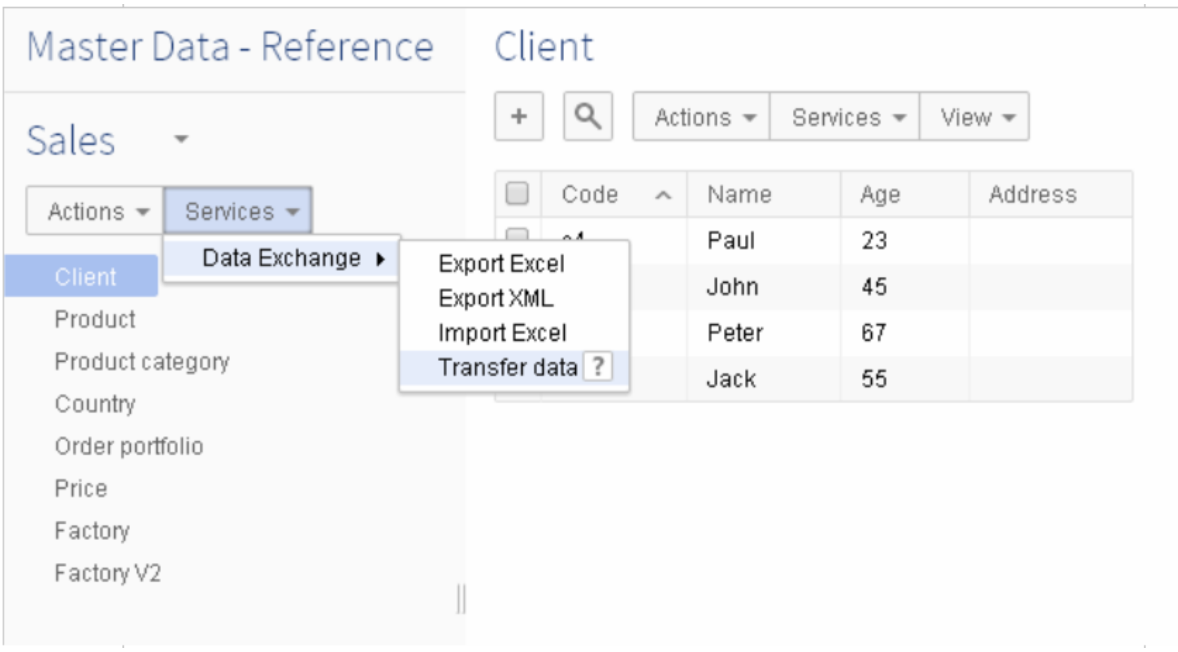
To make the export process available from the EBX5 type application ('Sales' data space) to the CRM application, the 'Interface' between both must be declared as follows:



Interface

□	Source application	Target application	Description
□	Sales - Default XML	Sales - EBX	
□	Sales - EBX	CRM - XML	
□	Sales - EBX	Sales - Default XML	

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

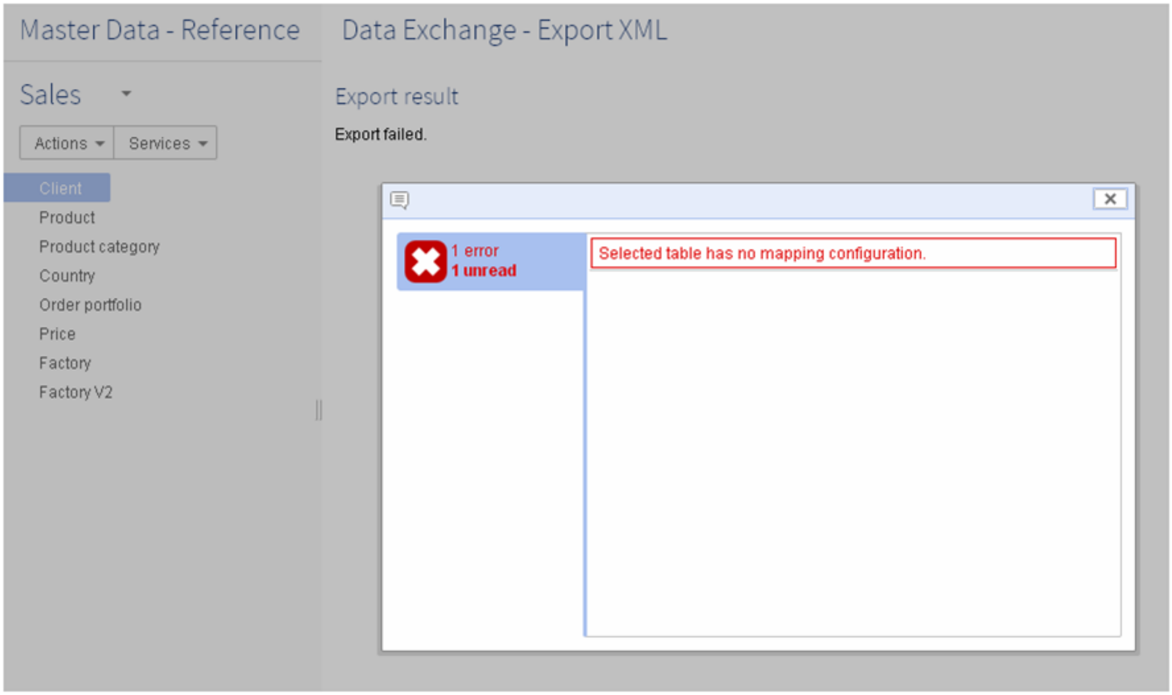


Client

□	Code	Name	Age	Address
□	01	Paul	23	
		John	45	
		Peter	67	
		Jack	55	

The options are already described in the previous section.

When executing the export process based on this configuration, the add-on raises this error:



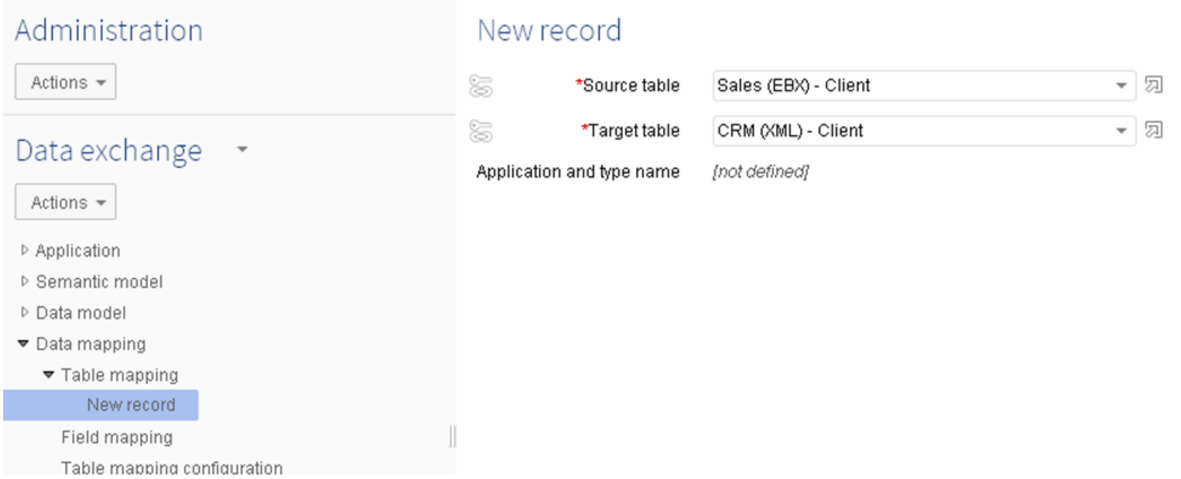
Export result

Export failed.

1 error
1 unread

Selected table has no mapping configuration.

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:

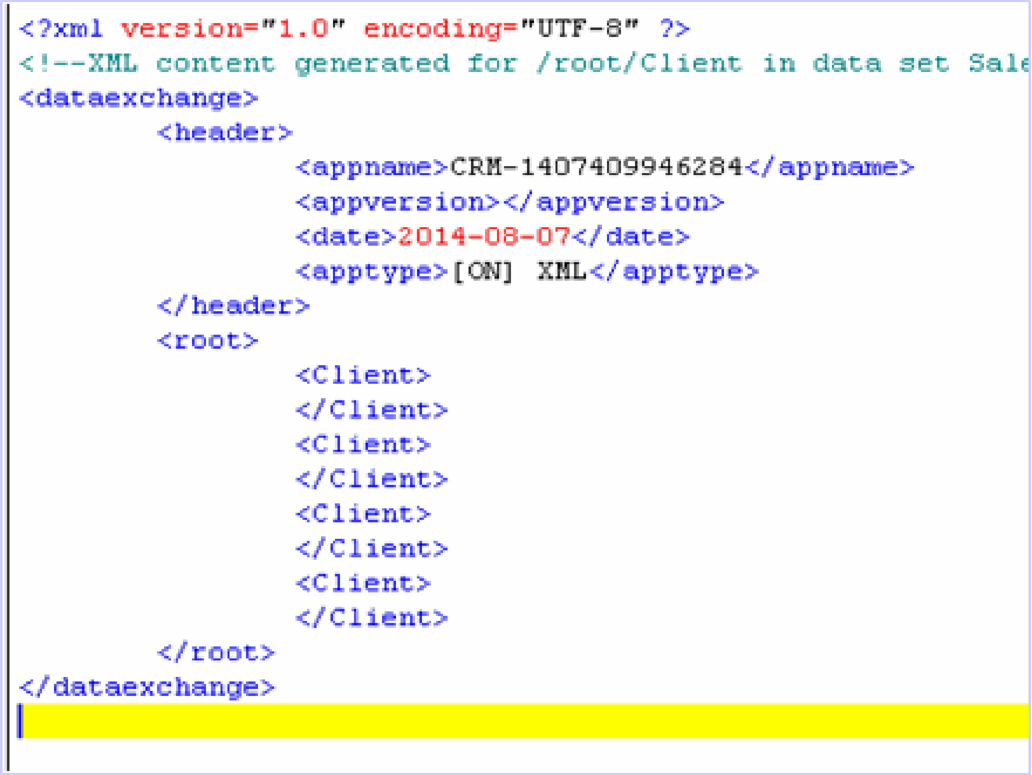


*Source table: Sales (EBX) - Client

*Target table: CRM (XML) - Client

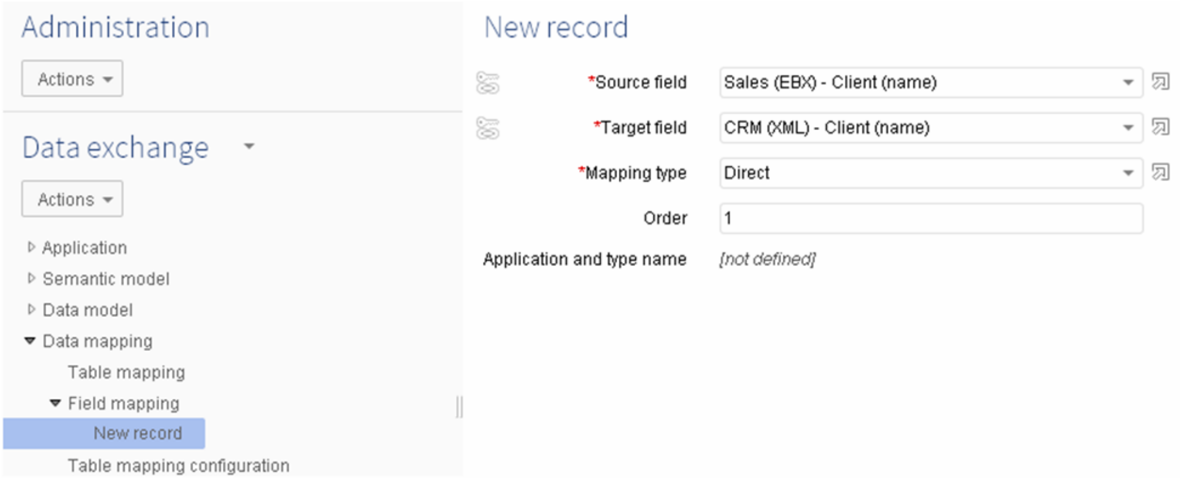
Application and type name: [not defined]

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>
```

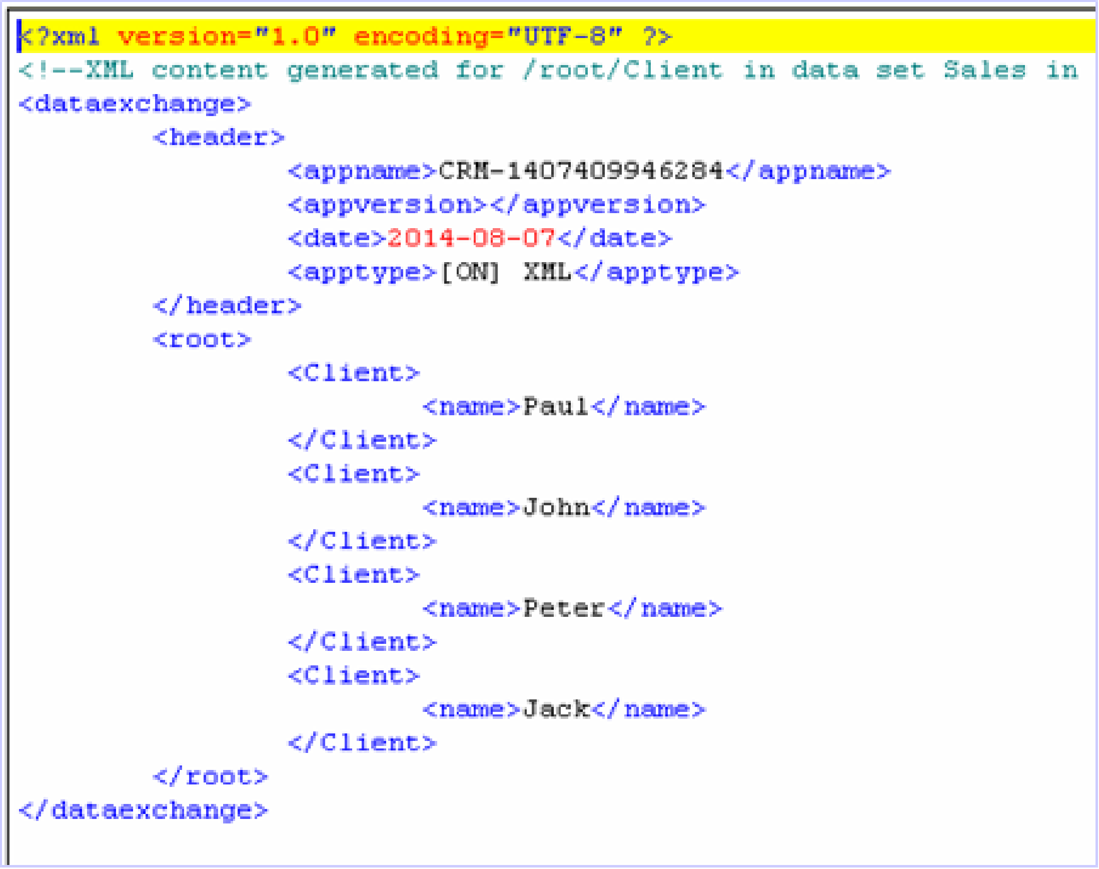
For every field, the mapping from the source to the target is declared as follows:



New record

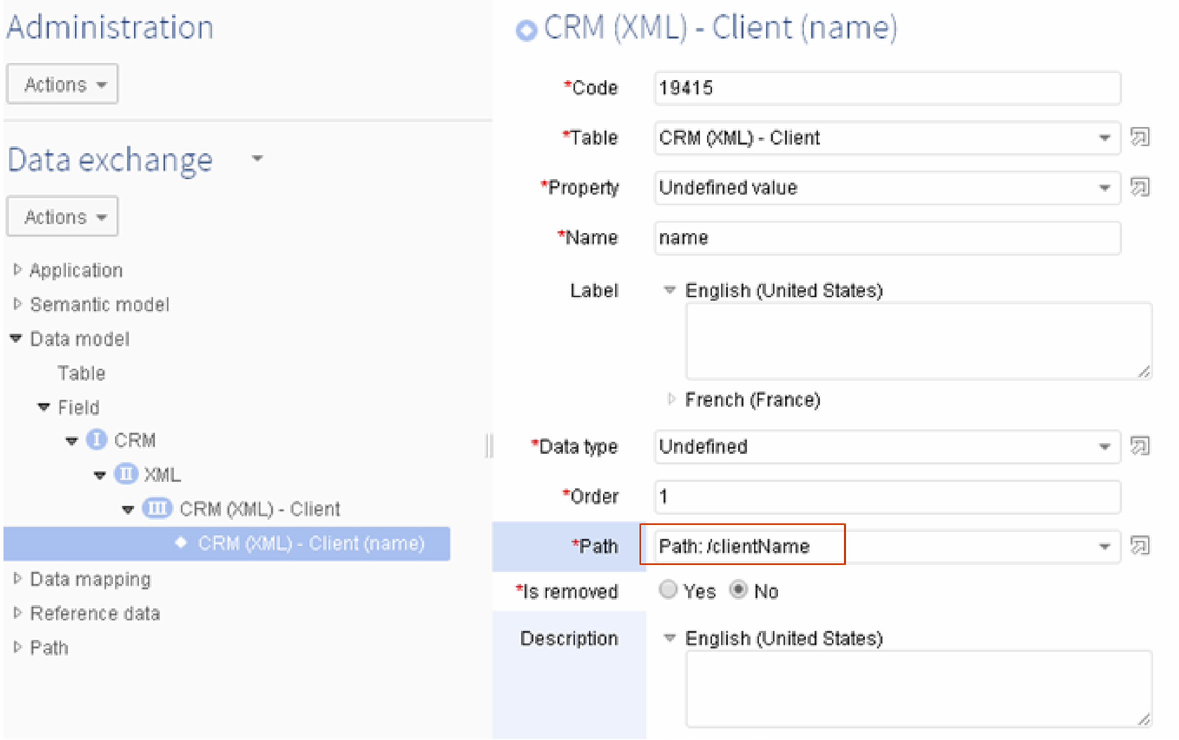
*Source field	Sales (EBX) - Client (name)
*Target field	CRM (XML) - Client (name)
*Mapping type	Direct
Order	1
Application and type name	[not defined]

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>
```

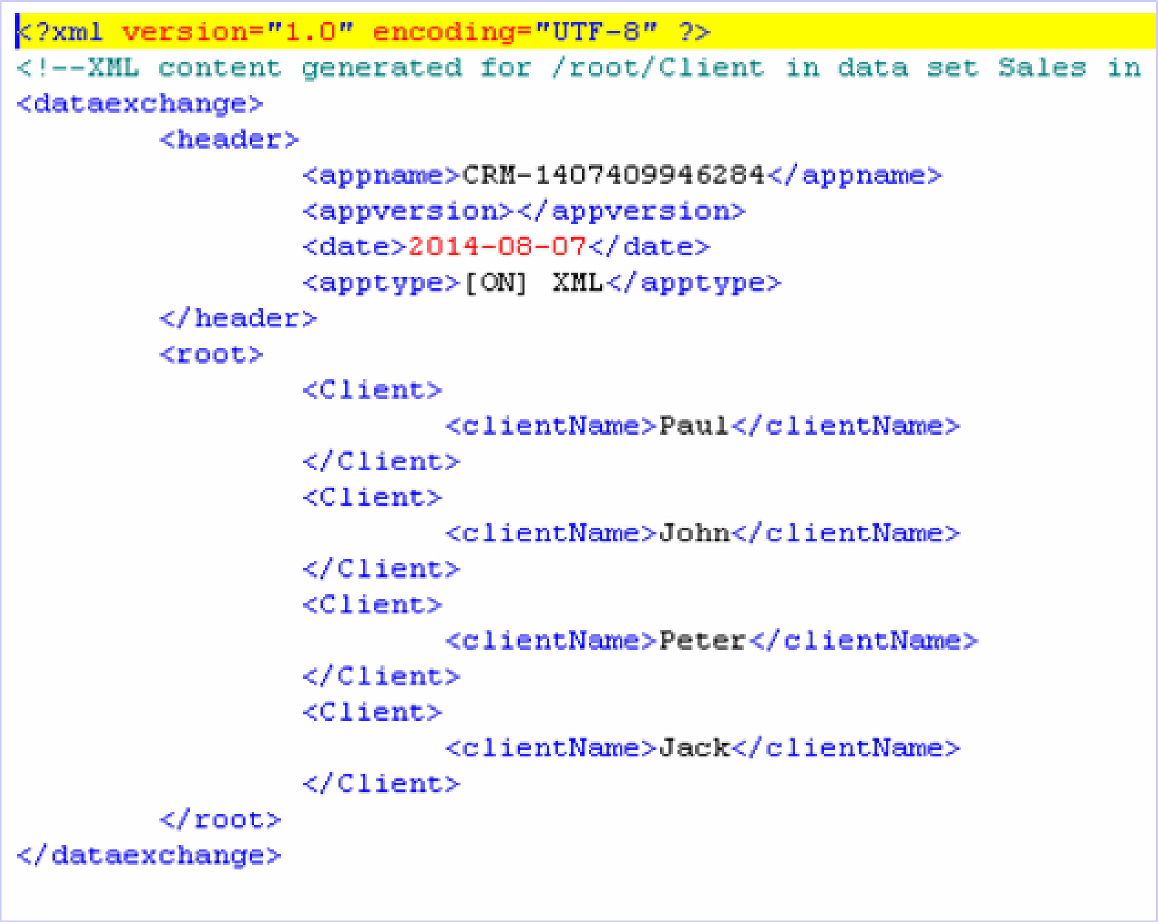
To change the path of a field, a direct modification in the path configuration can be done as illustrated below. The initial path '/Client' has been changed into '/clientName'.



CRM (XML) - Client (name)

*Code	19415
*Table	CRM (XML) - Client
*Property	Undefined value
*Name	name
Label	English (United States) French (France)
*Data type	Undefined
*Order	1
*Path	Path: /clientName
*Is removed	○ Yes ● No
Description	English (United States)

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

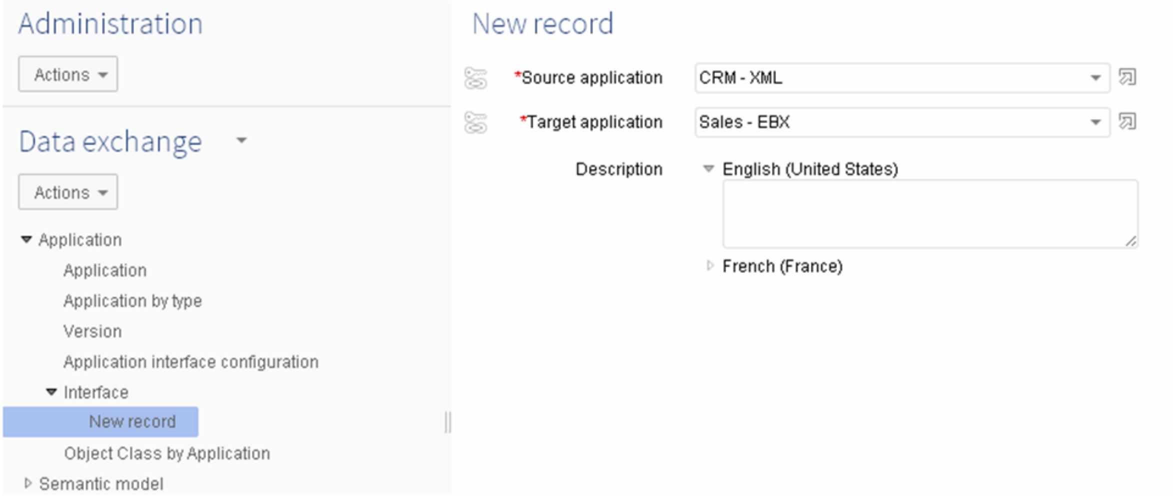


```
<?xml version="1.0" encoding="UTF-8" ?>
<!--XML content generated for /root/Client in data set Sales in
<dataexchange>
        <header>
                <appname>CRM-1407409946284</appname>
                <appversion></appversion>
                <date>2014-08-07</date>
                <apptype>[ON]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

2.3.2. XML import

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.

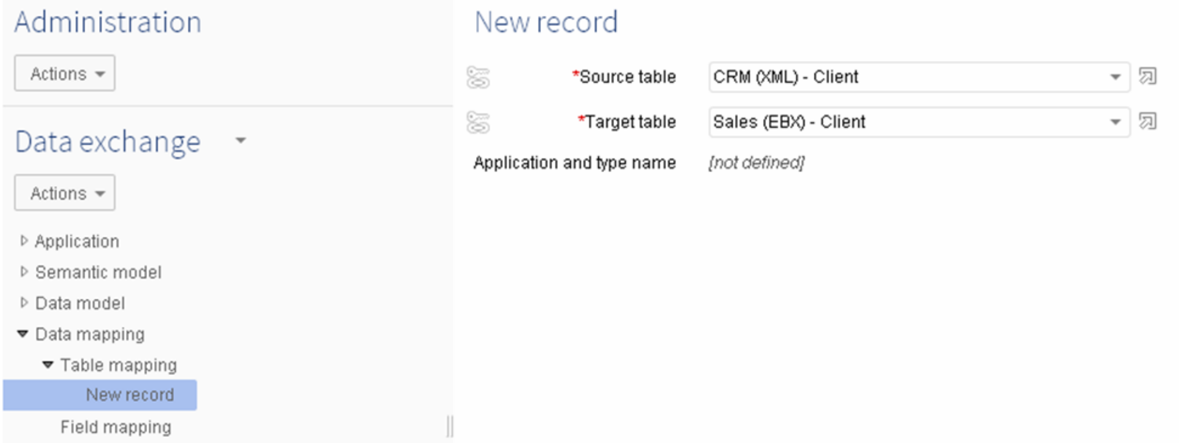
Then you create an 'Interface' from the source XML type application ('CRM') to the target EBX5 type application ('Sales').



New record

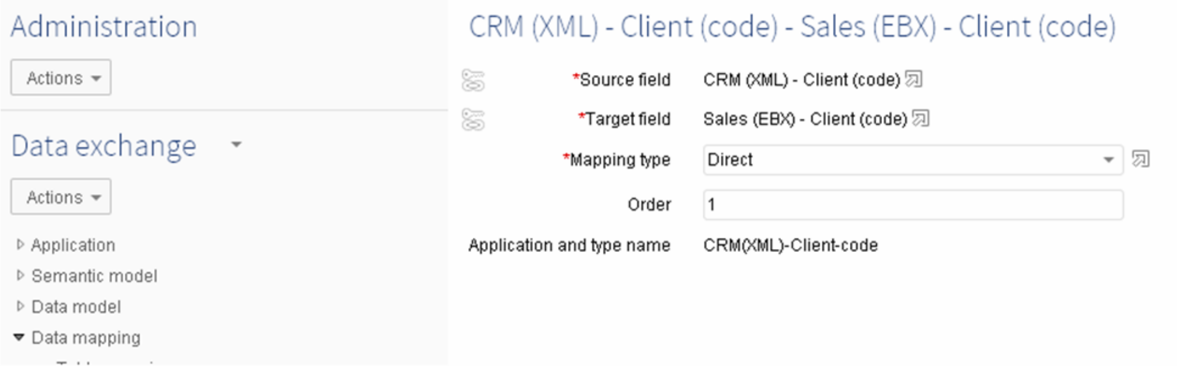
*Source application	CRM - XML
*Target application	Sales - EBX
Description	English (United States)
	French (France)

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



New record

*Source table	CRM (XML) - Client
*Target table	Sales (EBX) - Client
Application and type name	[not defined]



CRM (XML) - Client (code) - Sales (EBX) - Client (code)

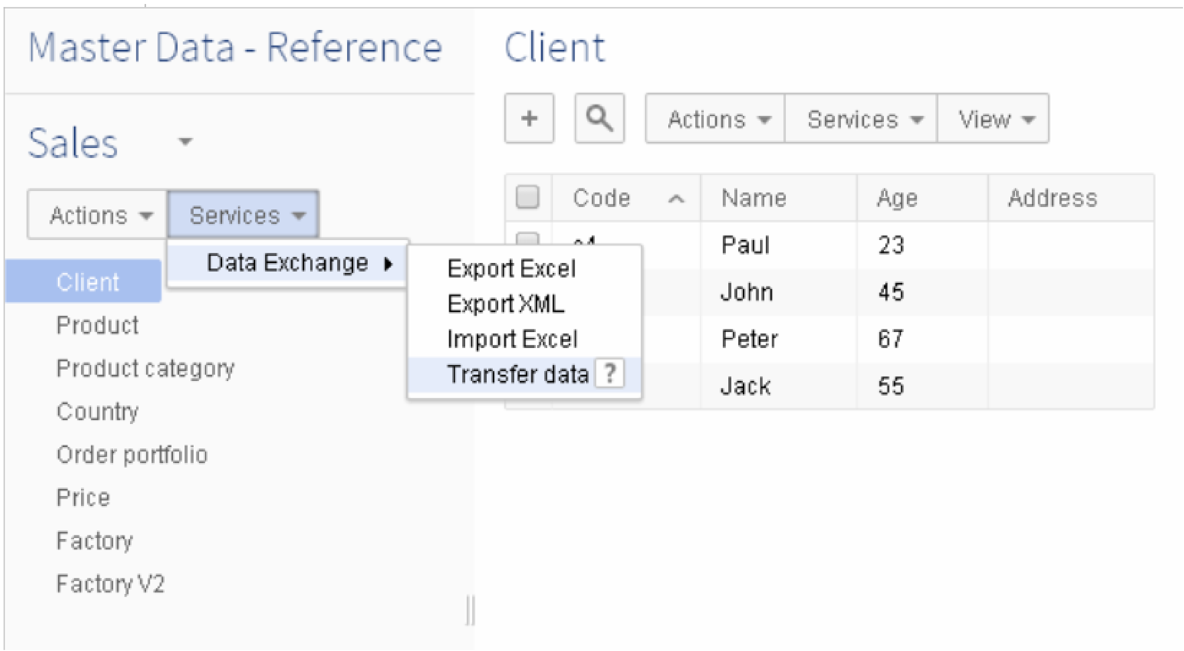
*Source field	CRM (XML) - Client (code)
*Target field	Sales (EBX) - Client (code)
*Mapping type	Direct
Order	1
Application and type name	CRM(XML)-Client-code

7.4 **2.4. Data transfer**

Data transfer is used to copy data from one EBX5 table to another. If the tables share the same data model, the add-on automatically manages the transfer. However, transferring data between tables that don't share the same data model requires a bespoke data mapping configuration.

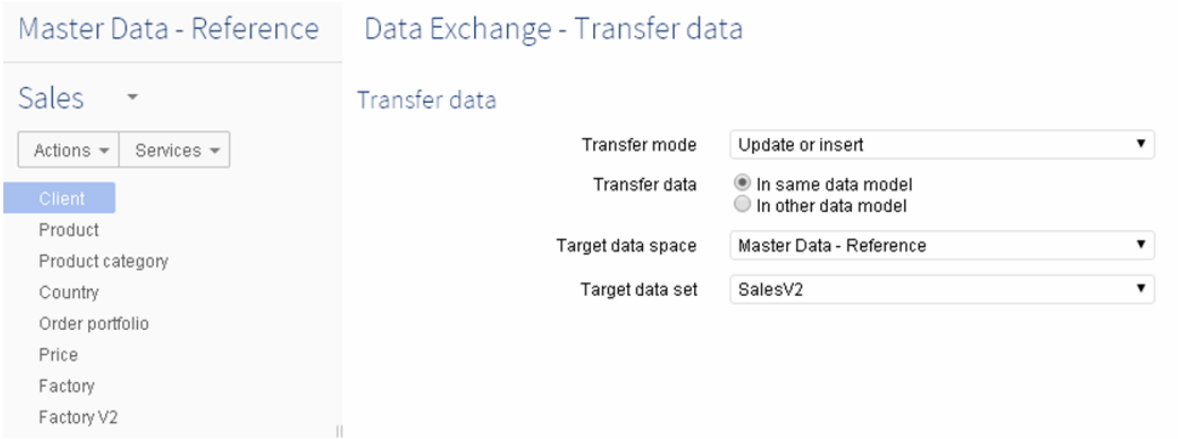
2.4.1. Data transfer within tables sharing the same data model

The data transfer is executed at the data set level. This example uses the 'Sales' data set:



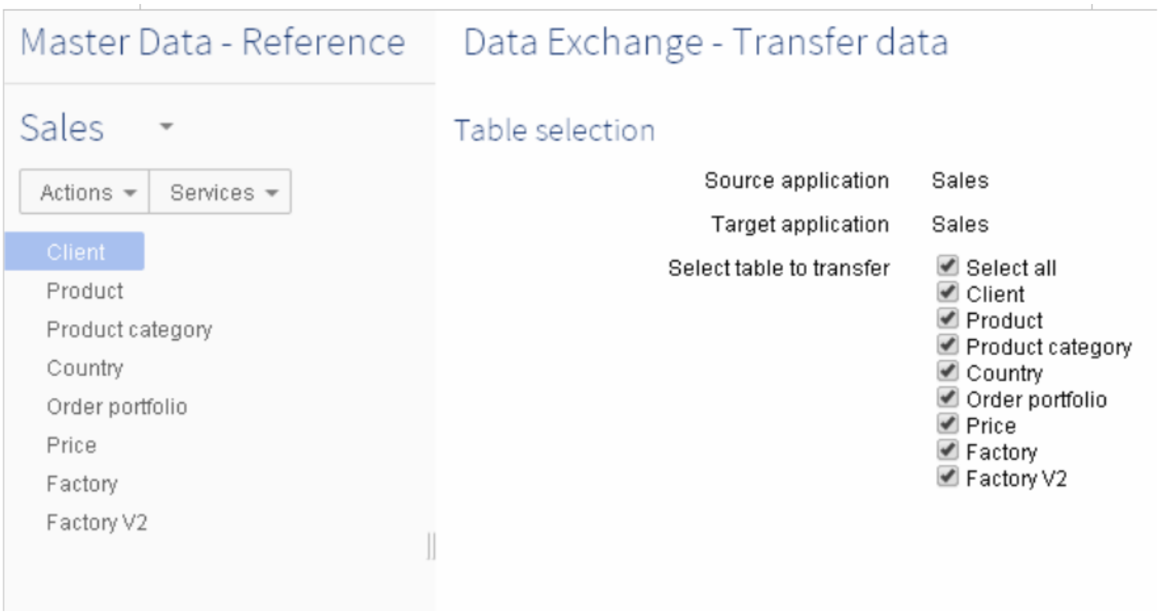
☐	Code	Name	Age	Address
☐		Paul	23	
		John	45	
		Peter	67	
		Jack	55	

The 'In same data model' option allows you to get all the data spaces and data sets that are based on the same data model as the 'Sales' data set. By selecting this option, data transfer is fully managed by the add-on without requiring any user-defined data mapping configuration.



Transfer mode	Update or insert
Transfer data	● In same data model ○ In other data model
Target data space	Master Data - Reference
Target data set	SalesV2

The data transfer can be done for one or several tables in the data set. You can select which tables to transfer as illustrated below:

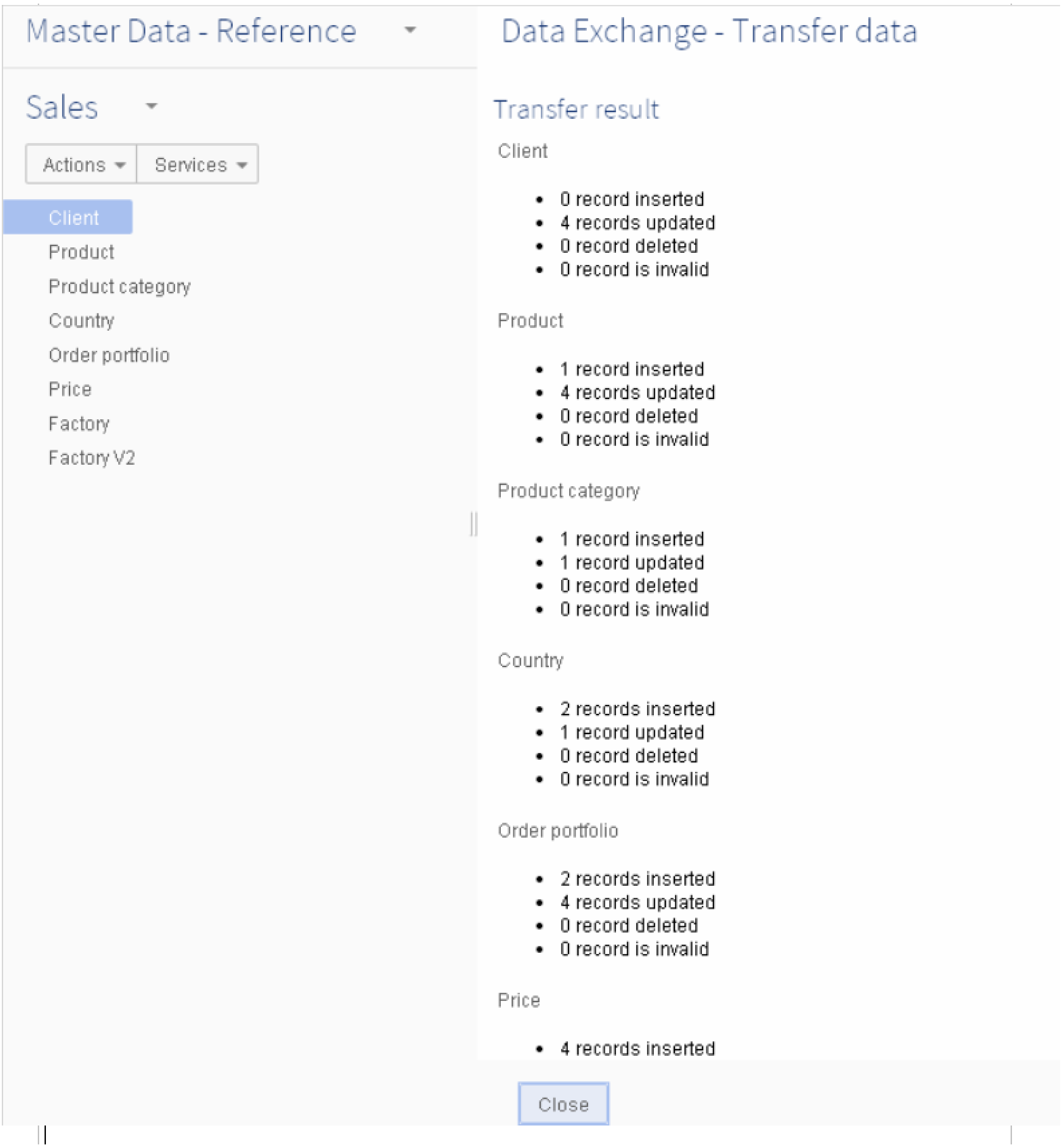


Data Exchange - Transfer data

Table selection

Source application	Sales
Target application	Sales
Select table to transfer	☑ Select all ☑ Client ☑ Product ☑ Product category ☑ Country ☑ Order portfolio ☑ Price ☑ Factory ☑ Factory V2

This overview displays the data transfer result:



Transfer result

Client

- 0 record inserted
- 4 records updated
- 0 record deleted
- 0 record is invalid

Product

- 1 record inserted
- 4 records updated
- 0 record deleted
- 0 record is invalid

Product category

- 1 record inserted
- 1 record updated
- 0 record deleted
- 0 record is invalid

Country

- 2 records inserted
- 1 record updated
- 0 record deleted
- 0 record is invalid

Order portfolio

- 2 records inserted
- 4 records updated
- 0 record deleted
- 0 record is invalid

Price

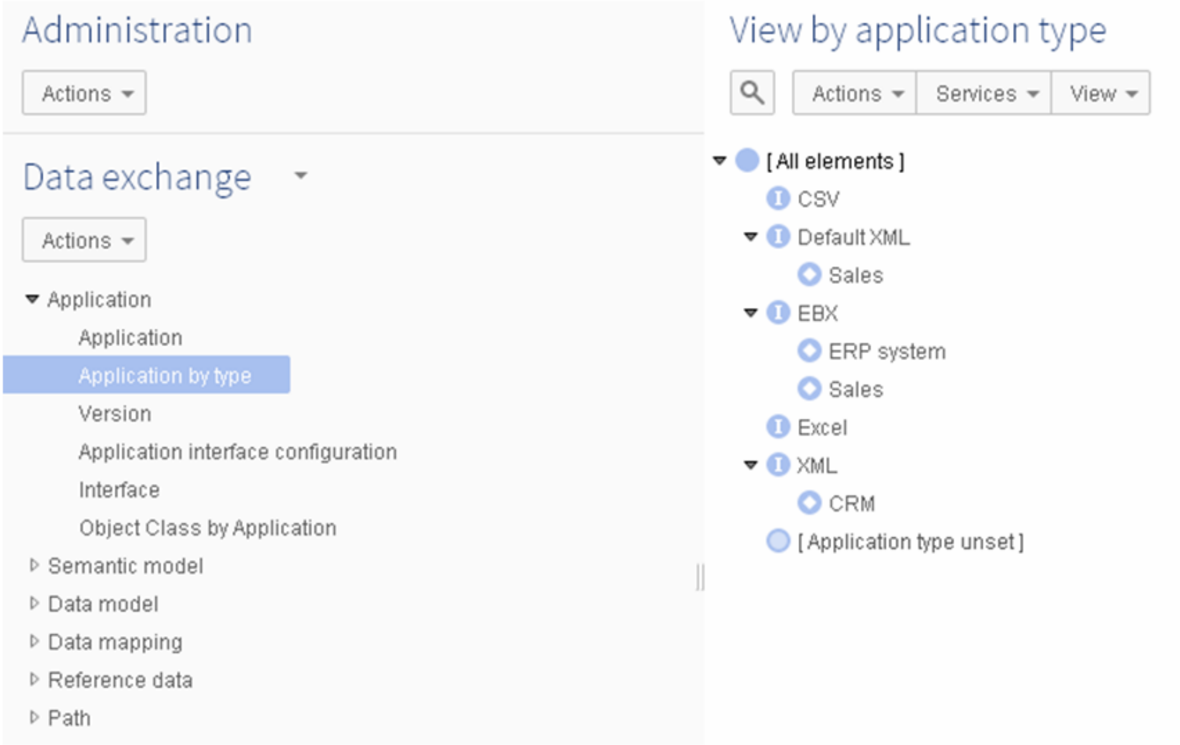
- 4 records inserted

During the transfer process, the add-on does not generate any data mapping configuration in the repository.

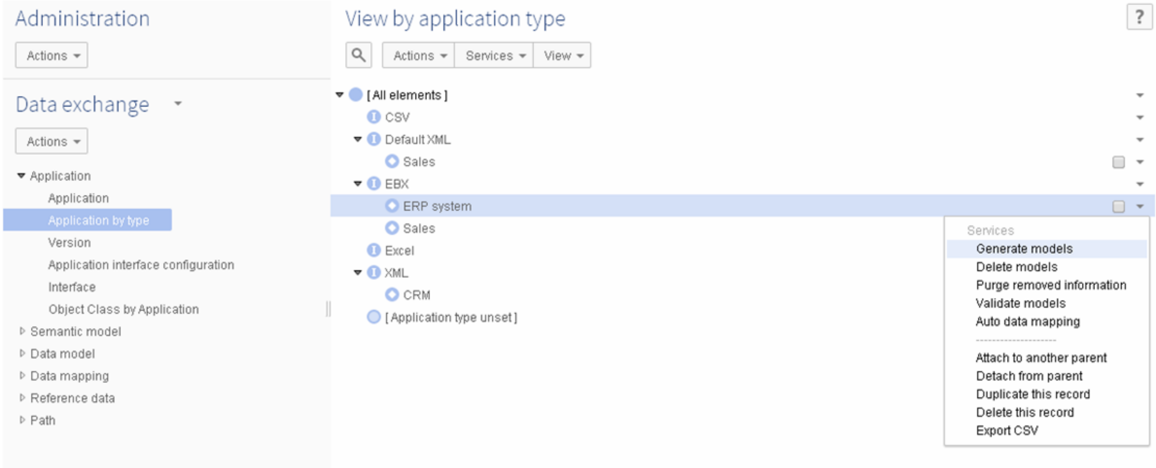
2.4.2. Data transfer within tables based on different data models

When the data transfer is applied between two different data models, a user-defined configuration in Data Exchange must be done, as described below.

First of all, the EBX5 type applications must be declared. The 'Sales' application has already been configured automatically during the previous XML data flow operations. Now, the new 'ERP system' EBX5 type application is declared.

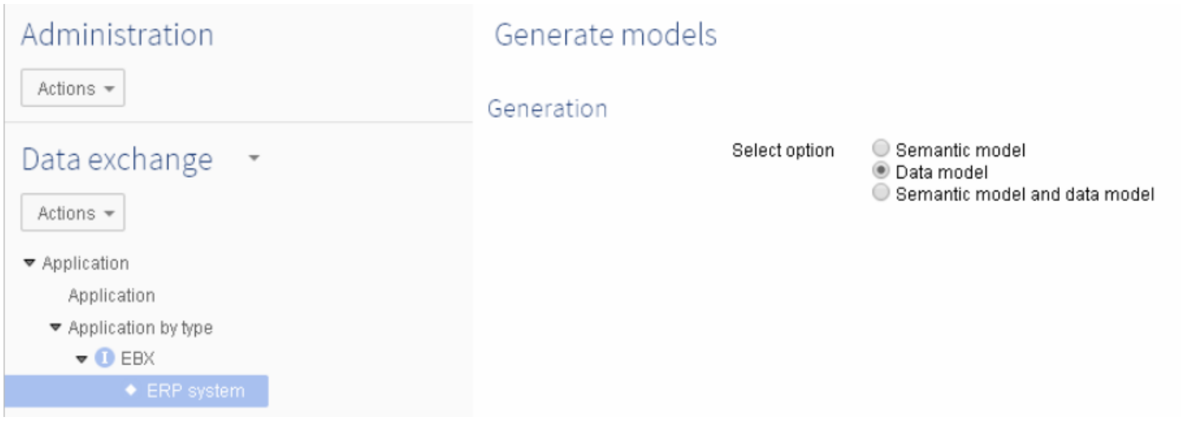


To automatically configure this new application's tables and fields, the 'Generate models' service is used:



The 'Generate models' service allows you to automatically retrieve the data model (tables and fields) configuration and the semantic model (Object class and Property).

In the current version of the add-on, the semantic data model makes the auto-data mapping between two different EBX5 data models possible.

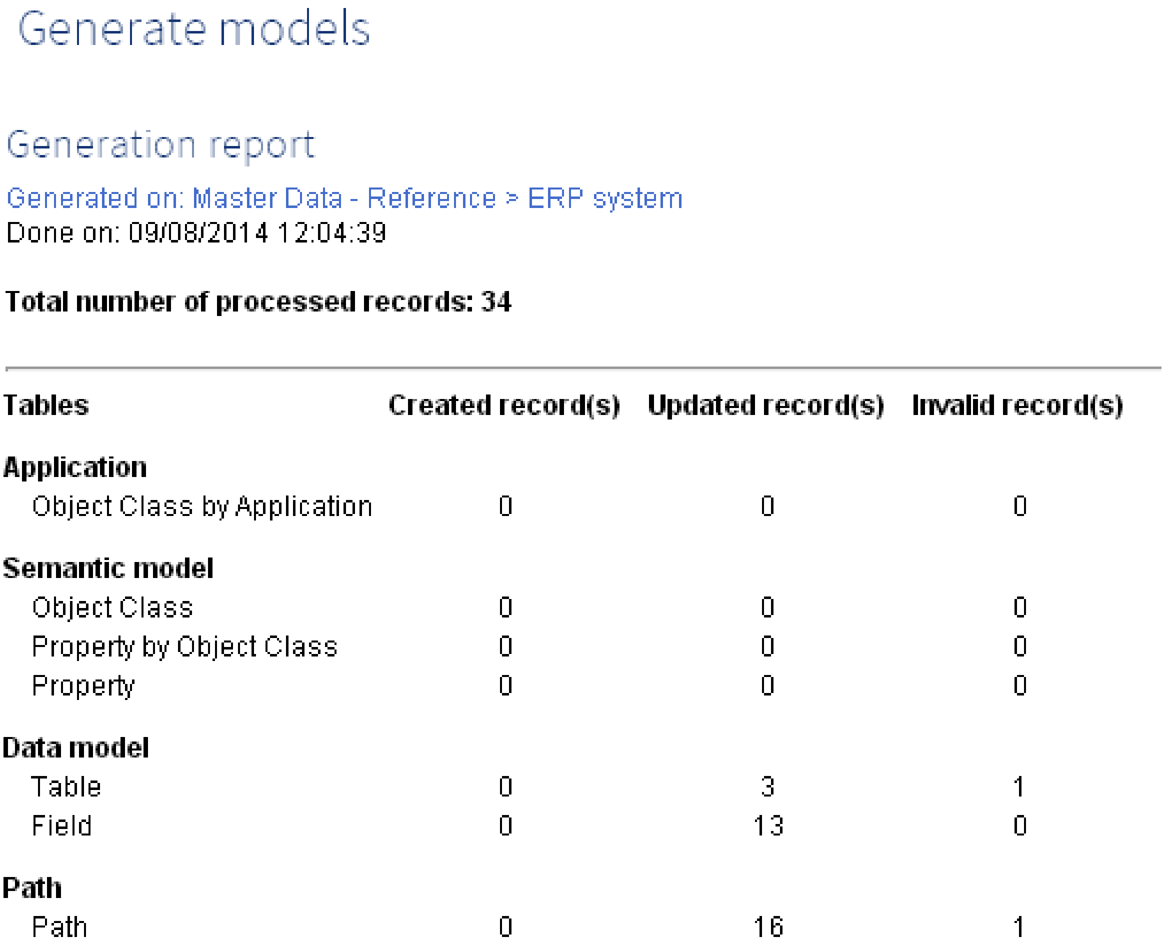


Generate models

Generation

Select option
○ Semantic model
● Data model
○ Semantic model and data model

An overview of the service result displays, as illustrated below:



Generate models

Generation report

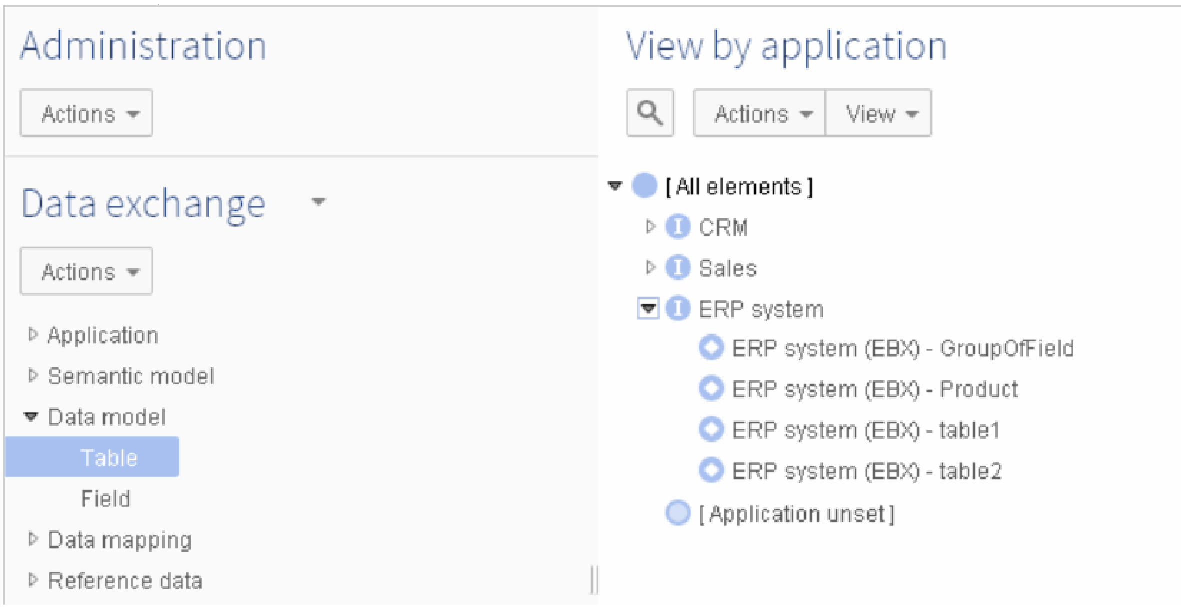
Generated on: Master Data - Reference > ERP system

Done on: 09/08/2014 12:04:39

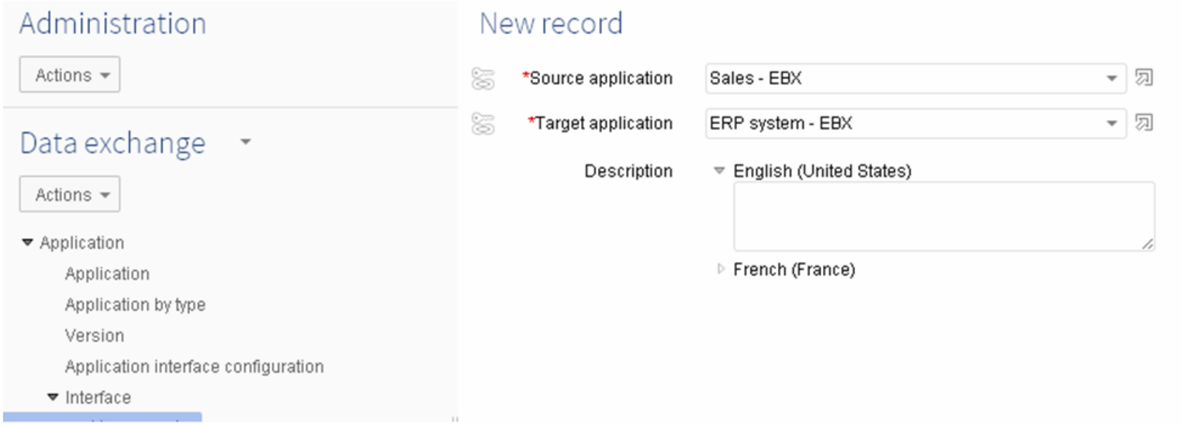
Total number of processed records: 34

Tables	Created record(s)	Updated record(s)	Invalid record(s)
Application			
Object Class by Application	0	0	0
Semantic model			
Object Class	0	0	0
Property by Object Class	0	0	0
Property	0	0	0
Data model			
Table	0	3	1
Field	0	13	0
Path			
Path	0	16	1

Then, the tables are declared automatically for the 'ERP system' EBX5 type application:



It is now possible to configure the data transfer between the 'Sales' source application and the 'ERP system' target application. To declare this transfer process, an 'Interface' is created as illustrated below:

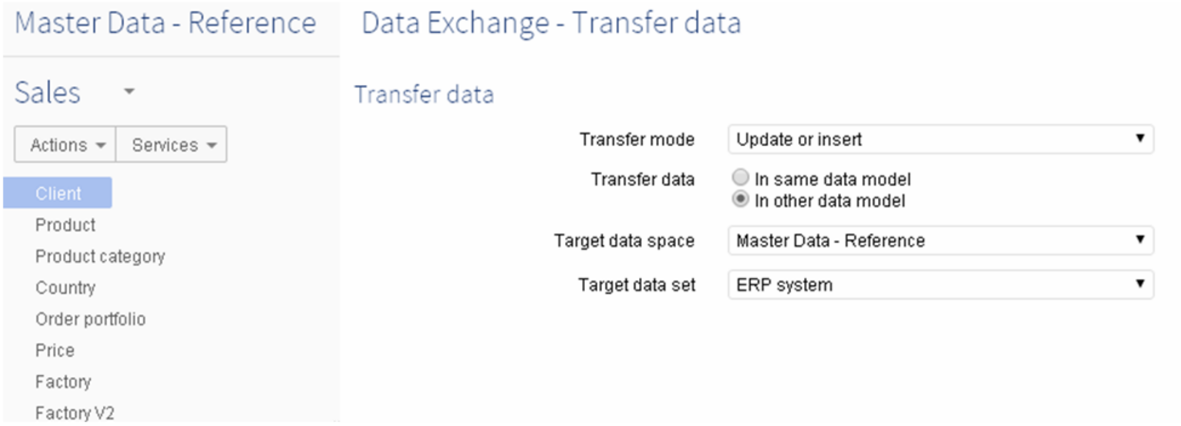


New record

*Source application	Sales - EBX
*Target application	ERP system - EBX
Description	English (United States) French (France)

The table mapping and field mapping use the same procedure already described for the XML data flow for configuration.

Based on this configuration, data transfer can be performed between the two applications relying on different data models as illustrated below.



Data Exchange - Transfer data

Transfer data

Transfer mode	Update or insert
Transfer data	○ In same data model ● In other data model
Target data space	Master Data - Reference
Target data set	ERP system

7.5 **2.5. Excel and CSV data flows**

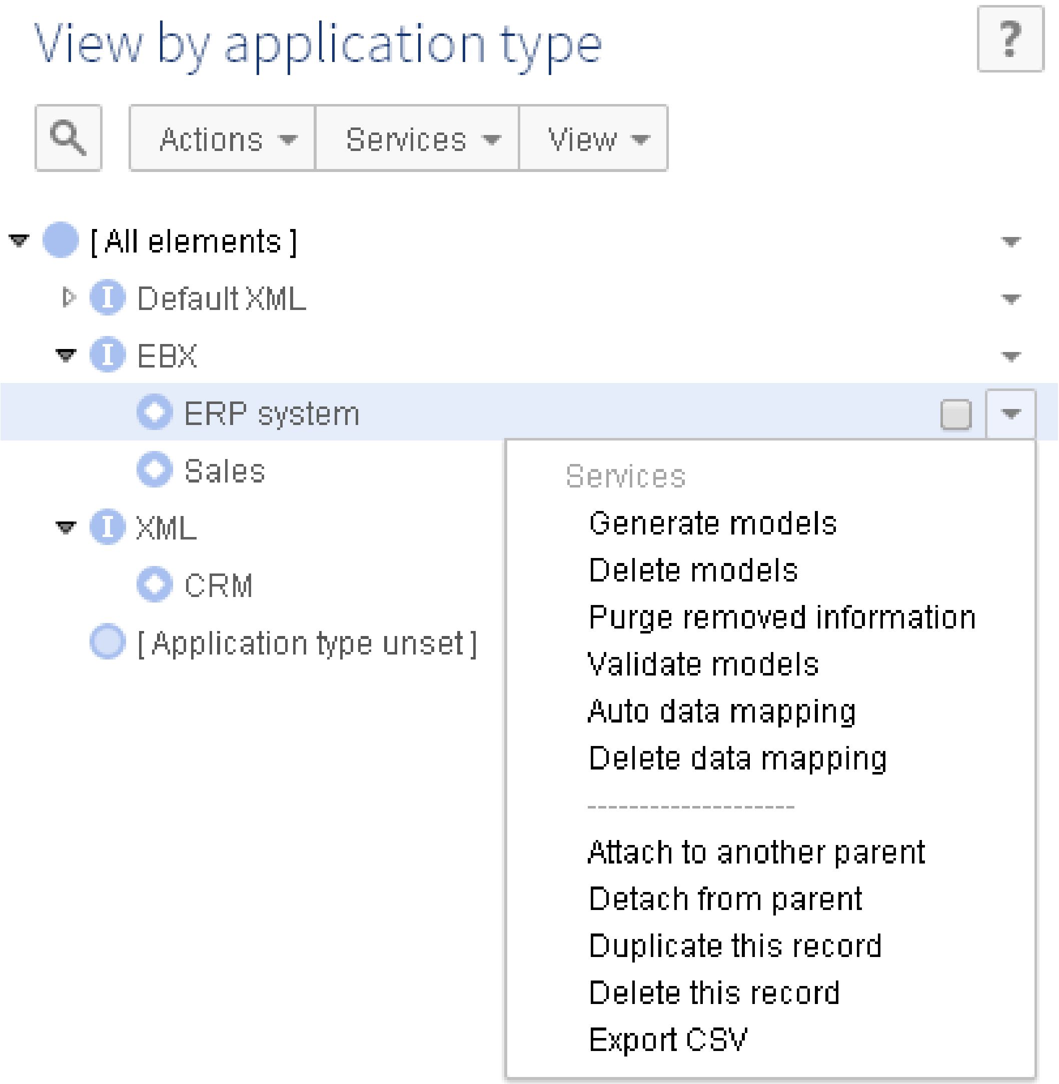
The Excel and CSV export and import processes are based on the previous version of Data Exchange, called 'Add-on for Microsoft® Excel'. You can refer directly to the online help for this part of the add-on.

In the current version of Data Exchange Add-on, the user-defined configuration is not used for the Excel and CSV data flows.

CHAPTER **8**

3. Advanced operations

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 'Data exchange' data space under the EXB5 '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.

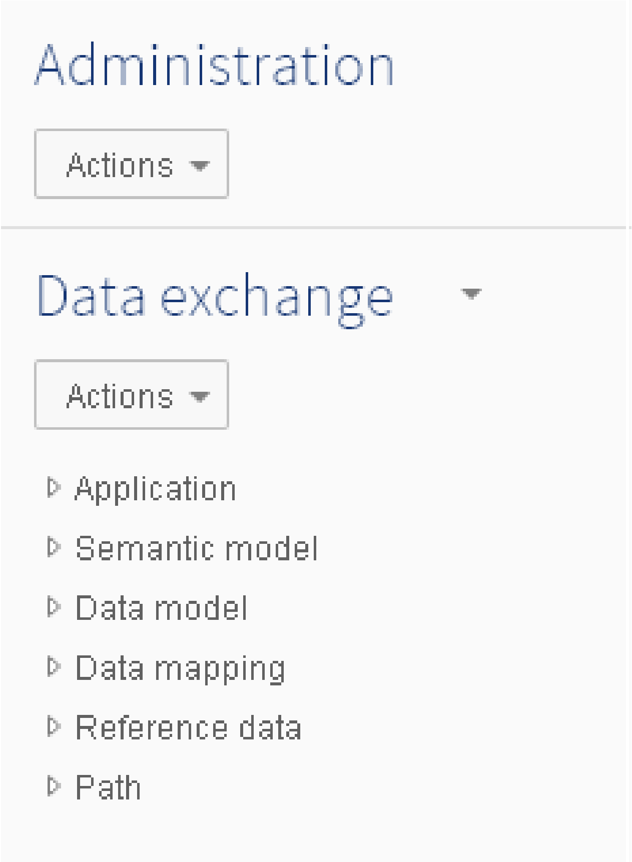
Services applied to an Application of type ->	Default XML	XML	EBX5
Delete data mapping	Physical deletion of all Tables, Fields and related data mapping (Tables: Version, Interface Application, Table mapping, Field mapping, Field mapping transformation and Path) for the Application. For an EBX5 type application, 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 an EBX5 path. Object classes and Properties can also be generated to get the semantic data model. The generation is based on a data space - data set 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 EBX5' data model. All unaligned Tables and Fields are then modified by changing the 'Is removed' property into the value 'Yes'
Auto data mapping	N/A	N/A	Automatically configures the data mapping between two EBX5 applications sharing the same Object class and property items.

CHAPTER 9

4. Configuring Data Exchange

This chapter contains the following topics:

9.1 **4.1. Global view**



The Data Exchange repository for data mapping configurations is located in the 'Data exchange' data set under the EBX5 'Administration' tab. The 'Application' domain contains the portfolio of applications with the interface configuration. An interface is used to configure the source and the target applications involved in a data flow.

The 'Semantic model' domain contains the Object Class and Property items used to link the Tables and Fields to the semantic layer. The Semantic model is not mandatory to configure data mapping. However, it allows you to improve data lineage and integration with Information governance. The 'Data model domain' contains the configuration for Tables and Fields. The 'Data mapping' domain contains the actual data mapping configuration applied at the table level, and then the field level.

9.2 **4.2. Application**

The Application domain contains the declaration of every application that is involved in the data flow (import, export, transfer) managed by Data Exchange.

4.2.1. 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.

4.2.2 Application by type

An application can be associated with one or many of the following types: CSV, Excel, Default XML, XML and EBX5.

Property	Definition
Application	The reference to an application.
Application type	The type of the application.
Application path	When this is an EBX5 type application, the path gives the data space and data set.

4.2.3. 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 appinfo 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.

4.2.4. 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.

4.2.5. 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.

9.3 4.3. Semantic model

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.

4.3.1. 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.

4.3.2. 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.

4.3.3. 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.

9.4 **4.4. Data model**

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

4.4.1. Table

A table is a container of data that depends on the application type. For EBX5 type applications, it is directly a table in EBX5. 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.

4.4.2. 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.
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.

9.5 4.5. Data mapping

This domain contains the data mapping configuration for tables and fields.

4.5.1. 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.

4.5.2. 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.
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.
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.

4.5.3. 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.

9.6 **4.6. Reference data**

4.6.1. Application type

The current version of the add-on manages the following types: XML, Default XML and EBX5.

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

4.6.2. Predefined Application type

The add-on provides these predefined Application types.

Application type	Description
EBX5	An EBX5 type application is referenced through a data set. Its tables and fields are described through an EBX5 path.
Default XML	A 'Default XML' type application relies on default XML paths automatically created by Data Exchange when an export XML process executes.
XML	An XML type application relies on user-defined XML path declarations.

4.6.3. 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.

4.6.4. Mapping type

In the current version of the add-on, only the 'Direct' mapping type is available. In future versions, other types will be provided to manage aggregation and split of fields.

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

4.6.5. 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.

Special notation:	
✖	Aggregation and split of fields values.

4.6.6. Transformation function

During data mapping, the data value can be transformed before moving to the target application. In the current version of the add-on, the 'No import' (and 'No export', 'No transfer') transformation function is delivered.

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.

4.6.7. 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 target field value is not imported.
No transfer	The field value is not transferred.

9.7 **4.7. Path**

The path domain contains the declaration of all paths used in the data mapping configurations.

4.7.1. Path

The Path table collects all the path declarations used in the data mapping configuration.

Property	Definition
Application type	A reference to the application type (EBX5, Default XML, XML).
Path type	Either Table, Field or Application.
Path	The path value.
Comment	Description of the path.

CHAPTER 10

5. Known limitations

- Multi-occurs complex data type are not supported (will be fixed in further release).
- Association object and selection node are not supported.
- User defined attribute (UDA) are not supported for the export and import processes.
- Split and aggregation of fields (will be added in further release).
- Transformation of fields (will be added in further release).
- Data lineage (will be added in further release).
- Graphical view of the data mapping configurations (will be added in further release).

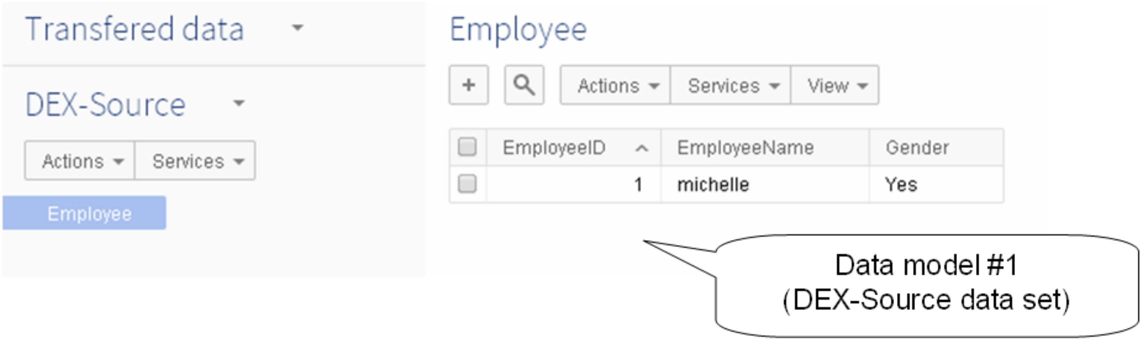
Appendix

CHAPTER **11**

Appendix 1 - Automatic data mapping for transferring data

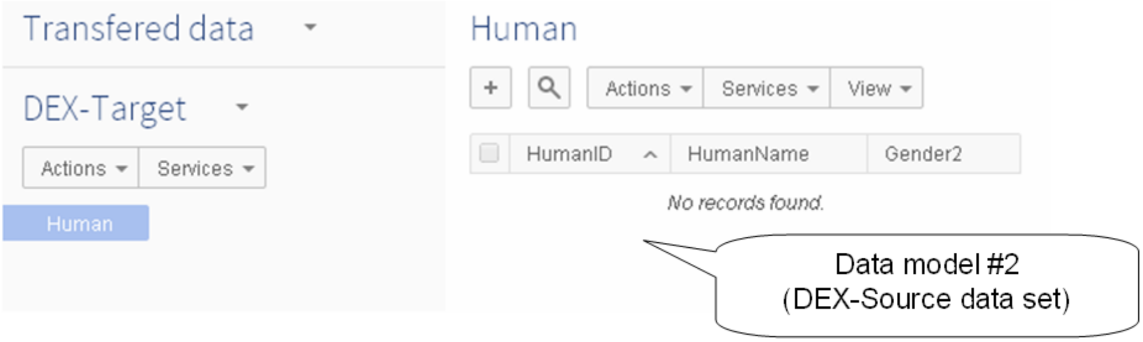
This appendix presents a use case about the automatic data mapping feature. The user wants to transfer data between two tables in two different applications relying on different data models (DEX-Source and DEX-Target).

Since the source and target applications are not based on the same data model, a configuration must be declared in Data exchange.



□	EmployeeID	EmployeeName	Gender
□	1	michelle	Yes

Data model #1
(DEX-Source data set)



□	HumanID	HumanName	Gender2

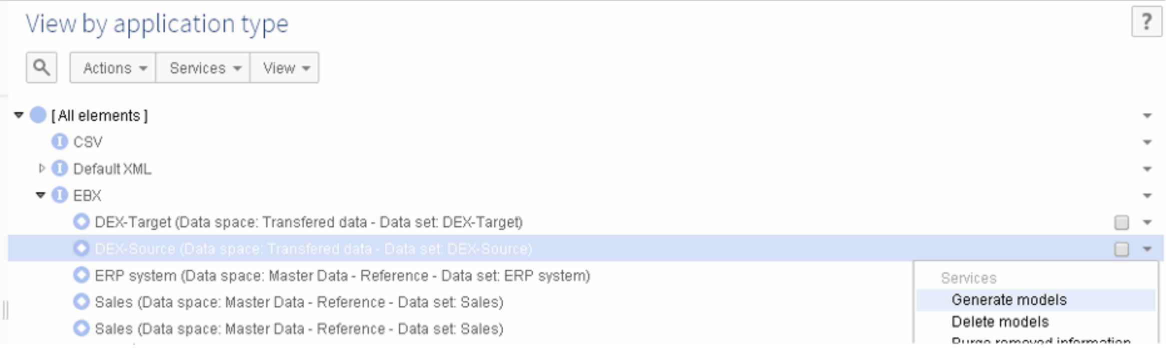
No records found.

Data model #2
(DEX-Source data set)

To facilitate the data mapping configuration, one of this application is used as the pivot (DEX-Source) to generate the semantic and data model.

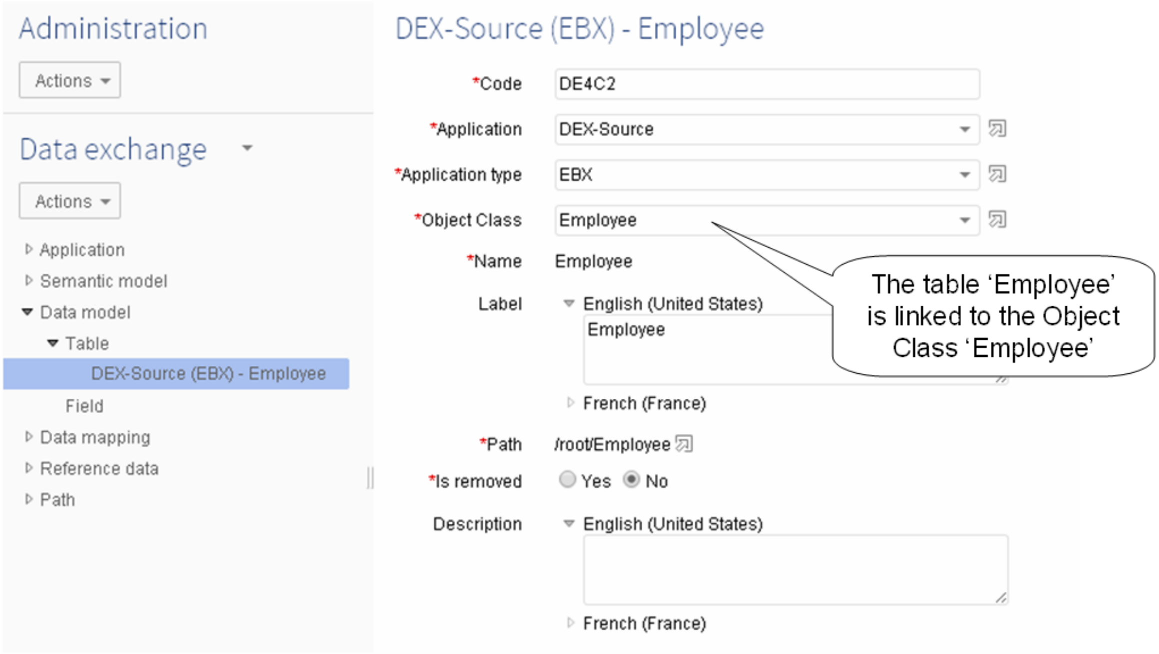
The generated Tables and Fields are linked to Object Class and Properties respectively of the source application.

Execution of the service 'Generate models' on the application 'DEX Source'.


View by application type

- [All elements]
  - CSV
  - Default XML
  - EBX
    - DEX-Target (Data space: Transfered data - Data set: DEX-Target)
    - DEX-Source (Data space: Transfered data - Data set: DEX-Source)
    - ERP system (Data space: Master Data - Reference - Data set: ERP system)
    - Sales (Data space: Master Data - Reference - Data set: Sales)
    - Sales (Data space: Master Data - Reference - Data set: Sales)

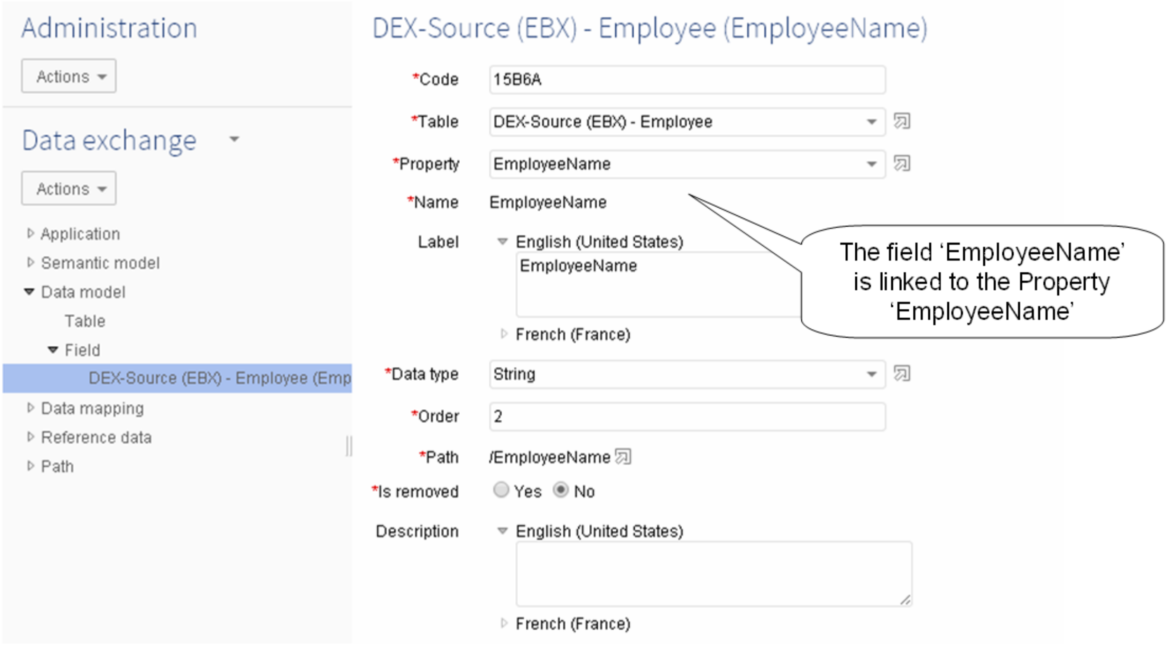
The Table 'Employee' is then linked to the Object Class 'Employee'.


DEX-Source (EBX) - Employee

*Code	DE4C2
*Application	DEX-Source
*Application type	EBX
*Object Class	Employee
*Name	Employee
Label	English (United States) Employee French (France)
*Path	/root/Employee
*Is removed	○ Yes ● No
Description	English (United States) French (France)

The table 'Employee' is linked to the Object Class 'Employee'

Each field of the table 'Employee' is linked to its related Property. Below the example of the field 'EmployeeName'.



DEX-Source (EBX) - Employee (EmployeeName)

*Code: 15B6A

*Table: DEX-Source (EBX) - Employee

*Property: EmployeeName

*Name: EmployeeName

Label: English (United States): EmployeeName

French (France)

*Data type: String

*Order: 2

*Path: /EmployeeName

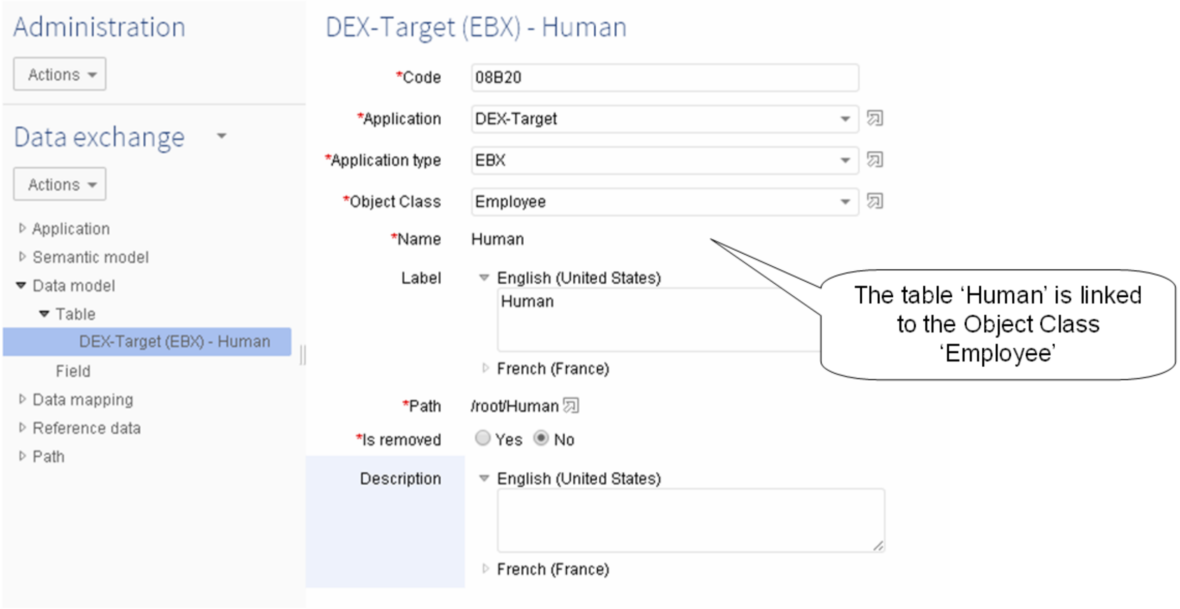
*Is removed: ○ Yes ● No

Description: English (United States)

French (France)

The field 'EmployeeName' is linked to the Property 'EmployeeName'

The table and fields of the target application are manually linked to the same Object Class and Properties of the source application.



DEX-Target (EBX) - Human

*Code: 08B20

*Application: DEX-Target

*Application type: EBX

*Object Class: Employee

*Name: Human

Label: English (United States): Human

French (France)

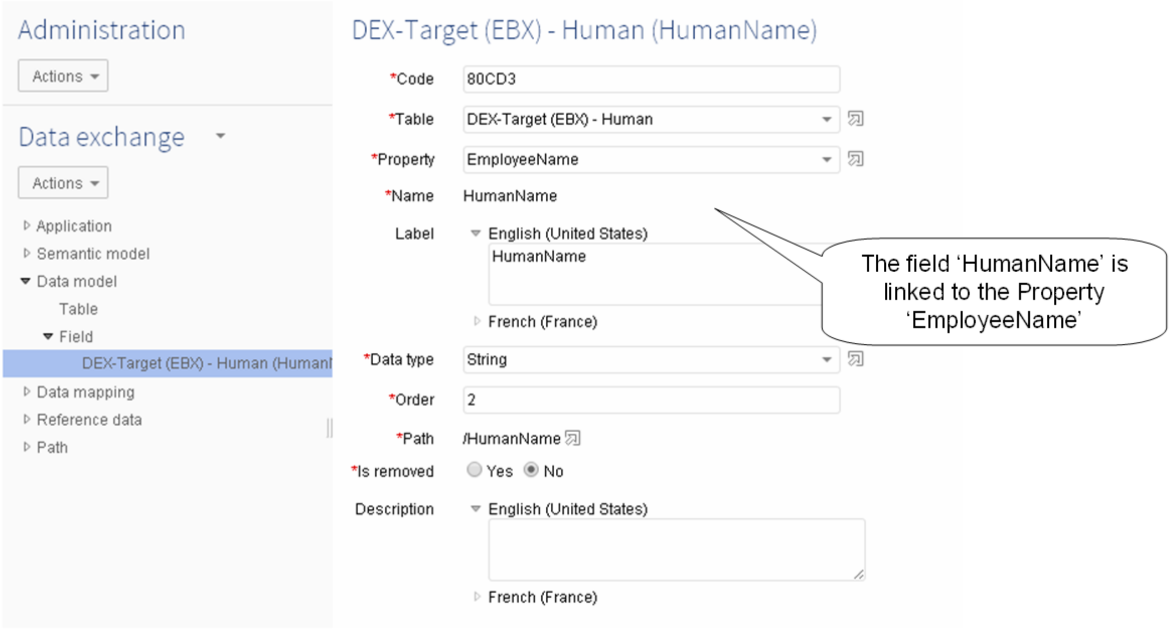
*Path: /root/Human

*Is removed: ○ Yes ● No

Description: English (United States)

French (France)

The table 'Human' is linked to the Object Class 'Employee'

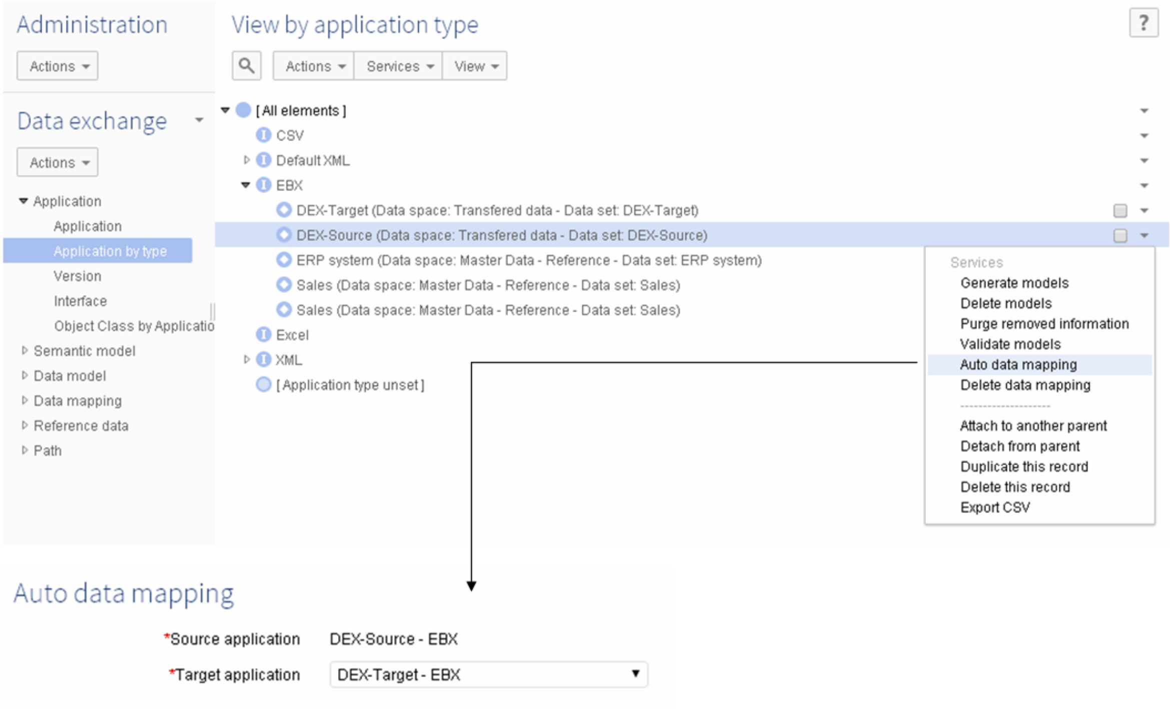


DEX-Target (EBX) - Human (HumanName)

*Code	80CD3
*Table	DEX-Target (EBX) - Human
*Property	EmployeeName
*Name	HumanName
Label	English (United States) HumanName French (France)
*Data type	String
*Order	2
*Path	/HumanName
*Is removed	○ Yes ● No
Description	English (United States) French (France)

The field 'HumanName' is linked to the Property 'EmployeeName'

Once the target application has been configured to share the same Object Class and Properties of the source application, the automatic generation of the data mapping is executed and the data transfer becomes available.

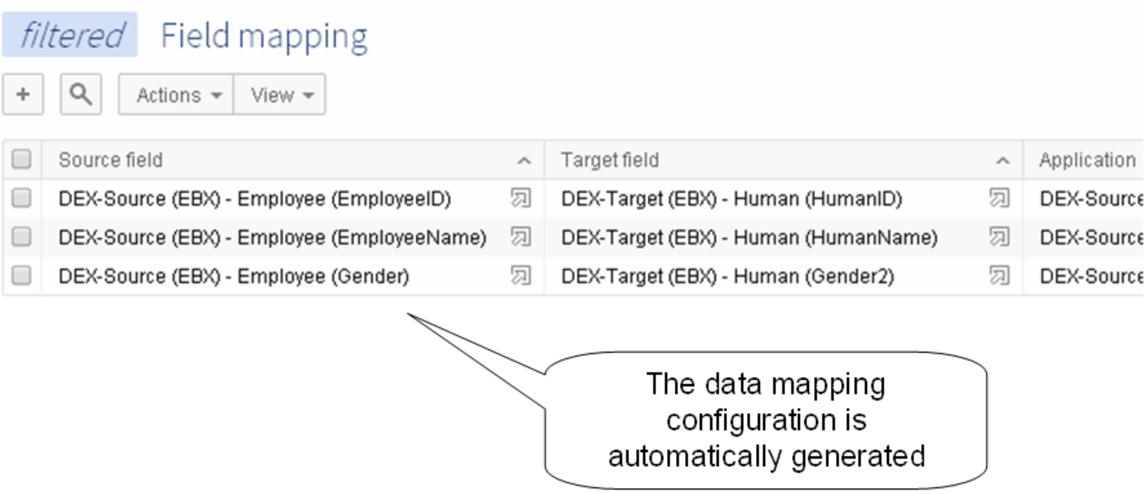


- [All elements]
  - CSV
  - Default XML
  - EBX
    - DEX-Target (Data space: Transfered data - Data set: DEX-Target)
    - DEX-Source (Data space: Transfered data - Data set: DEX-Source)
    - ERP system (Data space: Master Data - Reference - Data set: ERP system)
    - Sales (Data space: Master Data - Reference - Data set: Sales)
    - Sales (Data space: Master Data - Reference - Data set: Sales)
  - Excel
  - XML
  - [Application type unset]

Auto data mapping

*Source application	DEX-Source - EBX
*Target application	DEX-Target - EBX

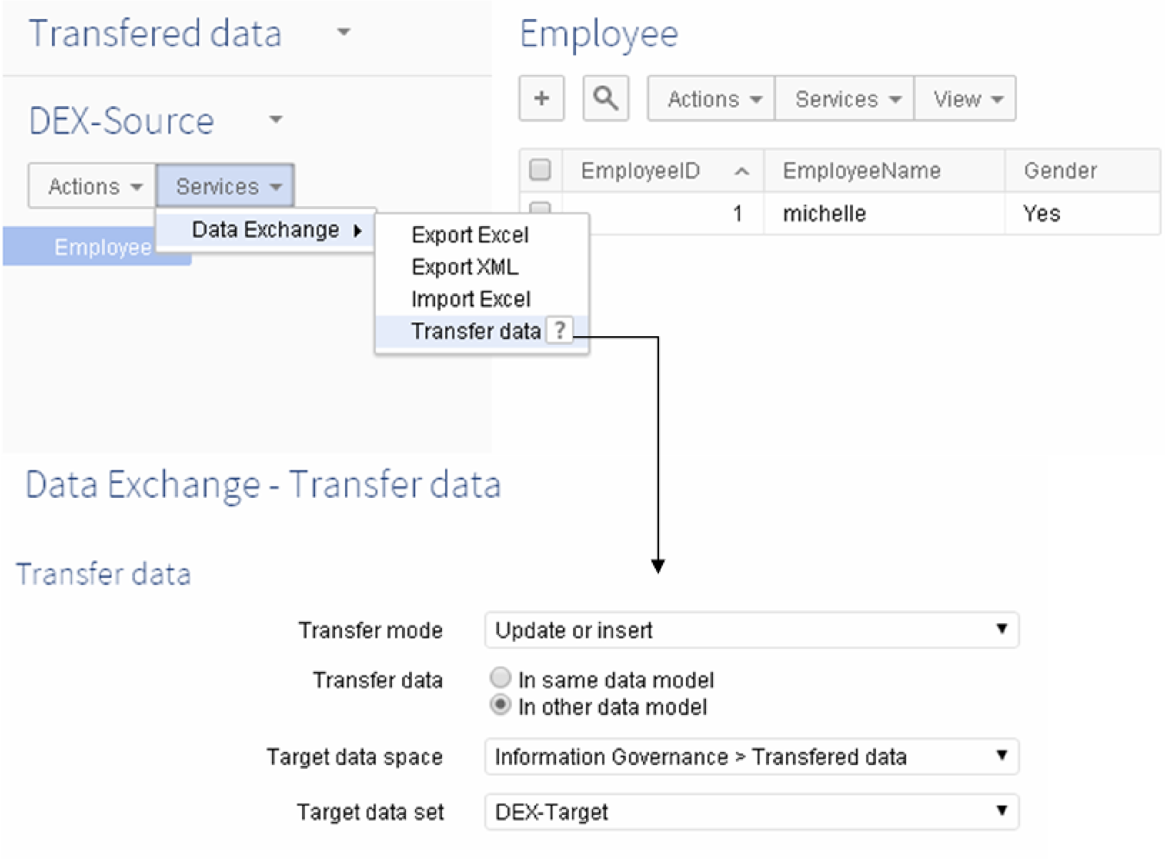
The result of the 'Auto data mapping' is illustrated below. It is used to make possible the data transfer from the source application 'DEX-Source' to the target 'DEX-Target'.



□	Source field	Target field	Application
□	DEX-Source (EBX) - Employee (EmployeeID)	DEX-Target (EBX) - Human (HumanID)	DEX-Source
□	DEX-Source (EBX) - Employee (EmployeeName)	DEX-Target (EBX) - Human (HumanName)	DEX-Source
□	DEX-Source (EBX) - Employee (Gender)	DEX-Target (EBX) - Human (Gender2)	DEX-Source

The data mapping configuration is automatically generated

On the data set of the source application, the 'Transfer data' service allows you to transfer the data from the DEX-Source to the DEX-Target directly.



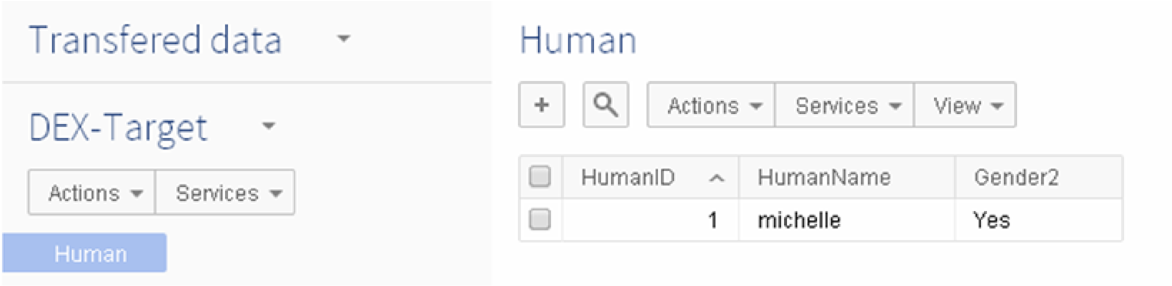
□	EmployeeID	EmployeeName	Gender
□	1	michelle	Yes

Data Exchange - Transfer data

Transfer data

Transfer mode	Update or insert
Transfer data	○ In same data model ● In other data model
Target data space	Information Governance > Transfered data
Target data set	DEX-Target

Then the target table 'Human' is fed with the source table 'Employee'.



□	HumanID	HumanName	Gender2
□	1	michelle	Yes

CHAPTER 12

Data Exchange Release Notes

This chapter contains the following topics:

12.1 **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.

12.2 **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.

12.3 **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.

12.4 **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.

12.5 **2.0.5**

Release Date: January 26, 2015

Bug Fixes

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

12.6 **2.0.4**

Release Date: December 8, 2014

Bug Fixes

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

12.7 **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".

12.8 **2.0.2**

Release Date: November 19, 2014

Bug Fixes

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

12.9 **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 data set then executes Import Excel service to import data from this file to another data set, 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.

12.10 **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 Data Exchange's configuration data set. 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.

12.11 **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.

12.12 **1.2.1**

Release Date: October 3, 2013

New updates

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

12.13 **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.

12.14 **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` .

12.15 **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.

12.16 **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.

12.17 **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 EBX5.

  See the section **Administration Guide >Installation & configuration > Supported environments > Browsing environments** in the main EBX5 documentation for more information.