

OL[®] Connect Walkthrough

Creating an XML data mapping configuration

Version 2022.2

OL Connect Walkthrough
Creating an XML data mapping configuration
Version 2022.2
Last Revision: 4/24/2023
Upland Software, Inc

All trademarks displayed are the property of their respective owners.

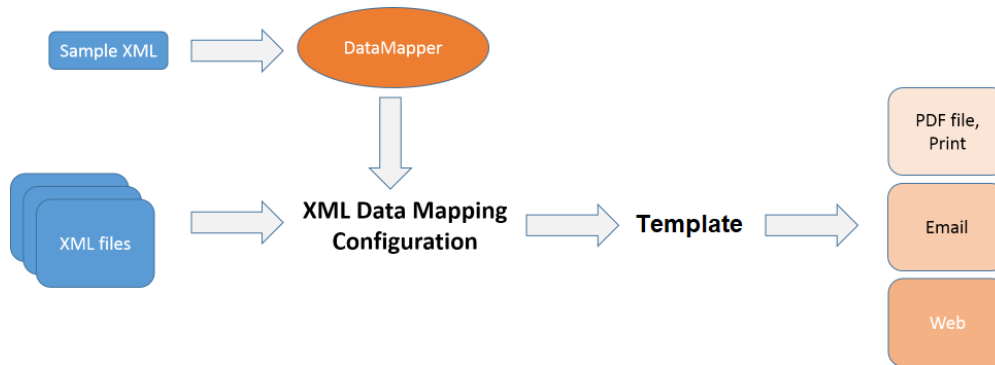
© Upland Software, Inc. 1994-2023. All rights reserved. No part of this documentation may be reproduced, transmitted or distributed outside of Upland Software by any means whatsoever without the express written permission of Upland Software. Upland Software disclaims responsibility for any errors and omissions in this documentation and accepts no responsibility for damages arising from such inconsistencies or their further consequences of any kind. Upland Software reserves the right to alter the information contained in this documentation without notice.

Table of Contents

Table of Contents	4
XML DataMapper walkthrough	5
Creating a DataMapper configuration	6
1. Open the XML file	6
2. Identify records	6
3. Set the default data format	7
4. Save the file	8
Extracting data	9
1. Create an Extract step	9
2. Add extra fields to an Extraction step	9
3. Extract line items using a Repeat step	10
Renaming steps, fields and detail tables	13
1. Rename extraction steps	13
2. Rename fields	13
3. Rename a detail table	14
Changing the data type of fields	16
1. Set a field's data type to Date	16
2. Set a field's type to Integer	16
3. Select a field and set its data type to Currency	17
Fine-tuning using JavaScript	18
1. Post function: capitalize country	18
2. Split a field and keep one part of it	18
3. Add a concatenated field	20
What's next?	21

XML DataMapper walkthrough

Connect's DataMapper lets you extract data from a variety of files. This walkthrough guides you through the process of creating a data mapping configuration for an XML file. This data mapping configuration enables the DataMapper to extract data from all XML files with the same structure. The data mapping configuration can then be used to add variable data to Connect Designer templates.



This tutorial introduces you to a big part of the basic functionality of the DataMapper. You will learn to create a Data Mapping Configuration by opening a sample file and defining records in it. Then you will extract the data by adding extraction steps and detail tables to the configuration. You will also learn how to rename fields and detail tables, and how to fine-tune the data for use in the Designer.

After completing this walkthrough, you'll have a basic knowledge of how to create a Data Mapping Configuration for XML files. However, this walkthrough describes only one way to do things. It shows how to extract data using toolbar buttons, for example, whereas it would also be possible to do that via drag-and-drop, the menu, or the shortcut menu.

So, go on and explore! To discover new ways and enhance your skills, please visit <https://olre-sourcecenter.uplandsoftware.com> and [the PlanetPress Connect Help Center](#) or [the PReS Connect Help Center](#).

Creating a DataMapper configuration

The first step towards a Data Mapping Configuration is to open a source file and help the DataMapper to identify records. In this exercise you will open an XML file and set the record boundaries.

1. Open the XML file

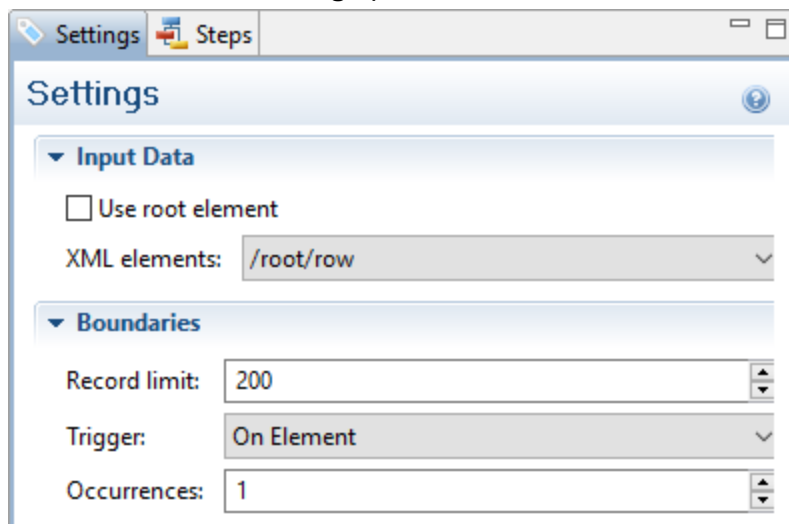
1. Start Connect Designer and select **File > New** from the menu.
2. Expand **Data mapping configuration**, then expand **Files**.
3. Click **XML file**.

There is also a Wizard for XML files. Save trying that for later; it automates a number of the steps that this Walkthrough demonstrates.

4. Click **Next** and select the sample file: **olsg-data.XML**.
5. Click **Finish**.

2. Identify records

Take a look at the **Settings** pane on the left.



The **Input Data** and **Boundaries** settings determine what identifies a record in the source file.

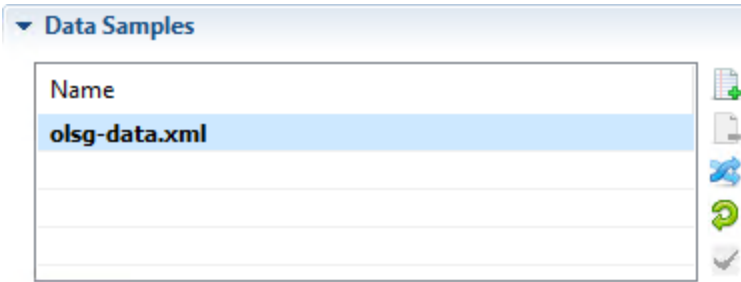
For this file, the correct settings are:

- **XML elements:** /root/row.
- **Trigger:** On element
- **Occurrences:** 1

This means that every time a new <row> element occurs, the DataMapper is triggered to start a new record. Setting **Occurrences** to **2** would make two <row> elements go into one record.

The **Record Limit** limits the number of records that can be browsed in the **Data model** pane and that will be saved as a sample within the Data Mapping Configuration.

Note: The Record Limit does not limit the amount of records that can be extracted from an XML file using this data mapping configuration in OL Connect Workflow.

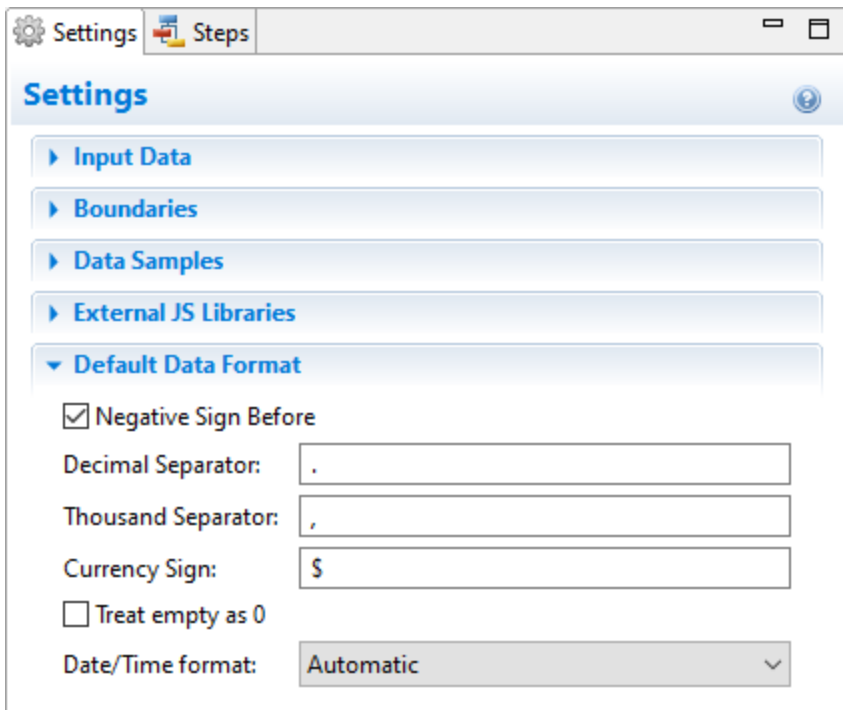


The XML file has been added to the Data Samples. Via the menu **File > Add data** you could add more samples if you'd need to. For this tutorial one sample is sufficient.

3. Set the default data format

By setting the default data format you are telling the DataMapper what format it can expect when it has to read a field as a date or a number from the source file.

1. Below **Data Samples** on the **Settings** pane, there is a heading **Default Data Format**. Click this heading.



Note: The initial settings for the Default Data Format can be set via the menu: **Window > Preferences > Datamapper > Default format**.

2. Both separators look fine. Leave them as they are.
3. In **olsg-data.XML**, amounts of money do not have a currency sign. Remove the currency sign.
4. Next, we can set the default date/time format. The dates in the source data don't look the same. `<Date>` has the format day/month/year (or rather, dd/mm/yyyy), while `<DueDate>` looks like this: yyyy-mm-dd. Leave the setting to **Automatic**, to let the DataMapper try and parse them automatically.

The other options can come in handy, for example when dates in a single source file belong to different time zones, or have different languages. See [the Help Center](#) for an explanation of those options.

4. Save the file

The DataMapper does not automatically save the configuration, so now that the initial settings are done, it's a good idea to save the file.

1. Select **File > Save as**, and give the data mapping configuration a name.
2. Press **Ctrl + S** every now and then to save the file while working on the data mapping configuration.


Tip: In the Save preferences (**Window > Preferences > Save**) you can configure auto-backup and auto-save.

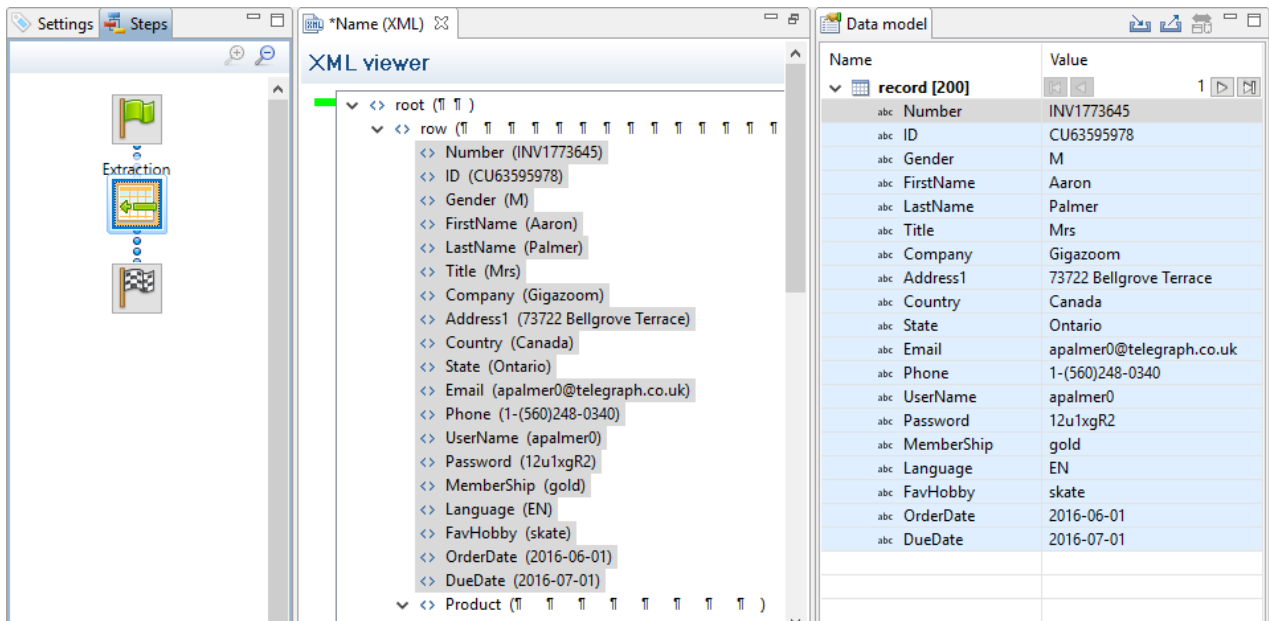
Extracting data

The next step is to extract data. Open the **Steps** pane, next to the **Settings** tab. Here you can see two flags, representing the preprocessor step and the postprocessor step. With these you could, among other things, define variables and modify the source file before extracting data (Preprocessor step) or modify the output file after extracting data (Postprocessor step). But in this case you can immediately start extracting data.

1. Create an Extract step

At the heart of this data mapping configuration are an **Extract** step and a **Repeat** step. First you'll add the **Extract** step.

1. In the first row element, select the elements from **Number** up to and including **DueDate**. (E.g. Hold the **Shift** key, click **Number** and then click **DueDate**.)
2. Click the **Add Extract Step** button . This adds an **Extraction step** to the **Steps** pane, between the Preprocessor step and the Postprocessor step.
3. Take a look at the **Data model** pane at the right. You can now browse the resulting records.




The screenshot shows three panes in a software application. The left pane is the 'Steps' pane, showing a sequence of steps: 'Preprocessor', 'Extraction', and 'Postprocessor'. The middle pane is the 'XML viewer', displaying an XML tree structure for a record. The right pane is the 'Data model' pane, showing a table of data for a record.

Name	Value
record [200]	1
abc Number	INV1773645
abc ID	CU63595978
abc Gender	M
abc FirstName	Aaron
abc LastName	Palmer
abc Title	Mrs
abc Company	Gigazoom
abc Address1	73722 Bellgrove Terrace
abc Country	Canada
abc State	Ontario
abc Email	apalmer0@telegraph.co.uk
abc Phone	1-(560)248-0340
abc UserName	apalmer0
abc Password	12u1xgR2
abc MemberShip	gold
abc Language	EN
abc FavHobby	skate
abc OrderDate	2016-06-01
abc DueDate	2016-07-01

2. Add extra fields to an Extraction step


At the bottom of each row element, below the products, there's some more information that needs to be extracted. You could create a new Extraction step for that, but it's better to add it to the existing Extraction step: the fewer Extraction steps, the faster the data mapping will be.

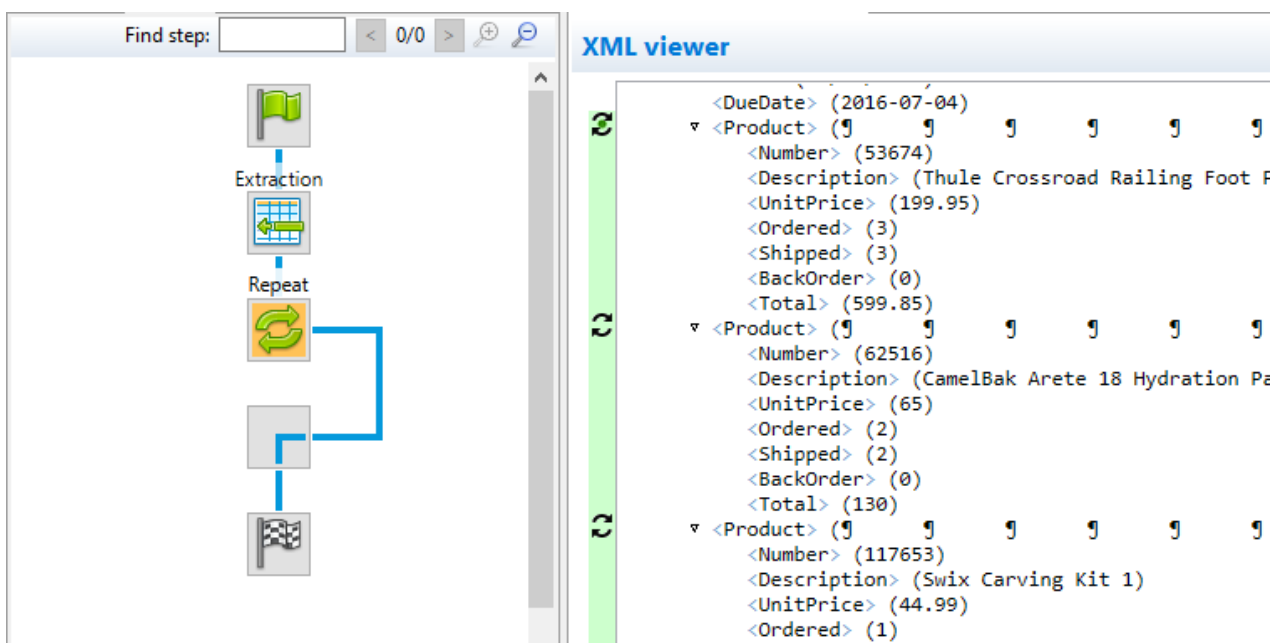
1. Click the **Extraction** step on the **Steps** pane to select it.
2. Click in the **XML viewer** (in the middle) and scroll down past all <product> elements. Select the elements **InvSubTotal**, **InvTaxTotal** and **InvTotal**. (Hold the **Ctrl** key while clicking the fields.)
3. Click the **Add Extract Field** button  to add the fields to the existing Extraction step, instead of creating a new Extraction step.

Tip: To create the fastest possible data mapping configuration, add as few Extraction steps as possible.

3. Extract line items using a Repeat step

When the number of a certain type of element in a record can vary, like the number of products in a row in this XML file, they have to go in a **detail table**.

1. The cursor in the data file should be located at the first of the repeated elements. Click the first **Product** element.
2. Click the **Add Repeat Step** button . In the left margin of the XML viewer you will see a green line. If you don't, there is probably only one product in the current record; browse some more records via the Data model pane. Also check that the path in the **Collection** field on the **Step Properties** pane (below the XML) is `./row/Product`, which means: every Product in every row element. The Collection field determines what the loop will go through. The **Add Repeat Step** icon appears before each product.

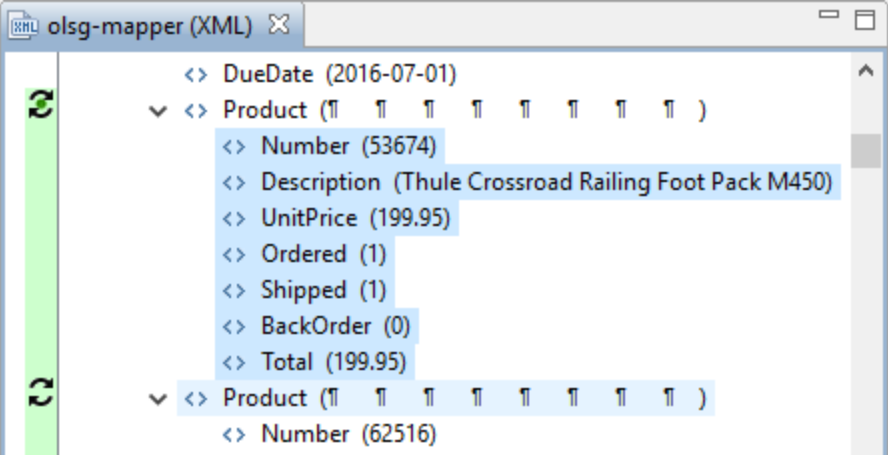



The screenshot displays the software's interface. On the left, the 'Steps' pane shows a sequence of steps: 'Extraction', 'Repeat', and a final step. A blue line connects the 'Repeat' step to the final step. On the right, the 'XML viewer' shows an XML document with three product elements. A green vertical bar highlights the first product element, and a green 'R' icon is placed to its left, indicating the Repeat step is active for that element.

```

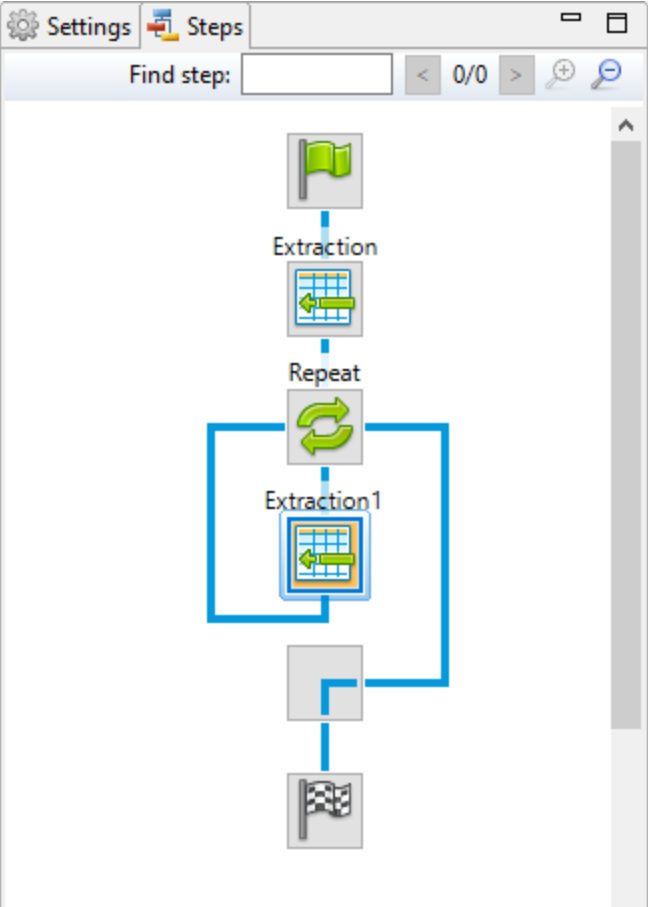
<DueDate> (2016-07-04)
  <Product> ( $ $ $ $ $ )
    <Number> (53674)
    <Description> (Thule Crossroad Railing Foot F
    <UnitPrice> (199.95)
    <Ordered> (3)
    <Shipped> (3)
    <BackOrder> (0)
    <Total> (599.85)
  <Product> ( $ $ $ $ $ )
    <Number> (62516)
    <Description> (CamelBak Arete 18 Hydration Pa
    <UnitPrice> (65)
    <Ordered> (2)
    <Shipped> (2)
    <BackOrder> (0)
    <Total> (130)
  <Product> ( $ $ $ $ $ )
    <Number> (117653)
    <Description> (Swix Carving Kit 1)
    <UnitPrice> (44.99)
    <Ordered> (1)
  
```

3. Select all elements inside the first Product element: click **Number**, and then, while pressing the **Shift** key, click **Total**.



4. Make sure the Repeat step is selected on the Steps pane. Then click the **Add Extract Step** button .

On the **Steps** pane, a new **Extraction** step has appeared within the **Repeat** step.



5. A detail table has been added on the **Data Model** pane. Try browsing the products that belong to one record.

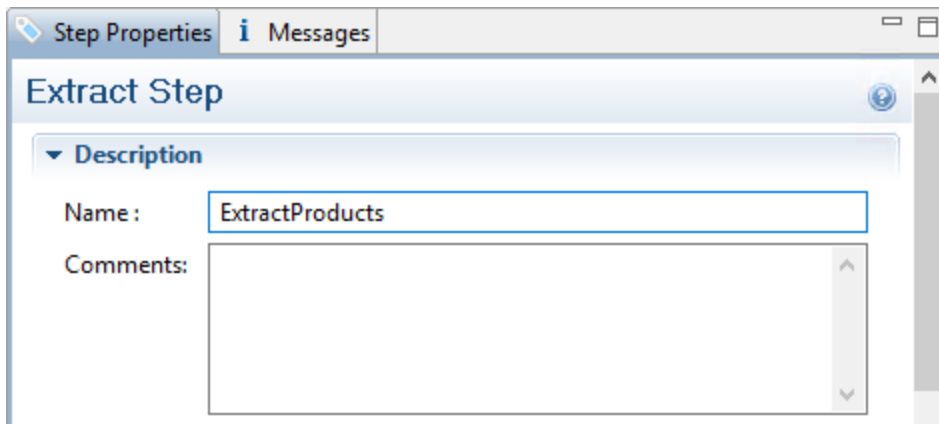
abc	InvTotal	5850.53
▼	detail [21]	⏪ ⏩ 2 ⏪ ⏩
abc	Number	62516
abc	Description	CamelBak Arete 18 Hydrat...
abc	UnitPrice	65
abc	Ordered	1
abc	Shipped	1
abc	BackOrder	0
abc	Total	65

Renaming steps, fields and detail tables

1. Rename extraction steps

With only two extraction steps, this data mapping configuration is fairly simple. Extraction steps in bigger and more complex data mapping configuration need clear names to help you keep track of what each step does. This is how to rename **Extract** steps.


1. On the **Steps** pane, click the **Extraction1** step inside the **Repeat** step.
2. On the **Step Properties** pane (below the XML), click **Description** to expand it, and rename the Repeat step, for example to **ExtractProducts**.

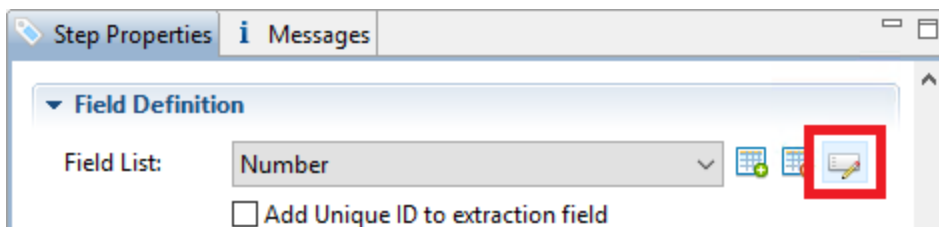


2. Rename fields

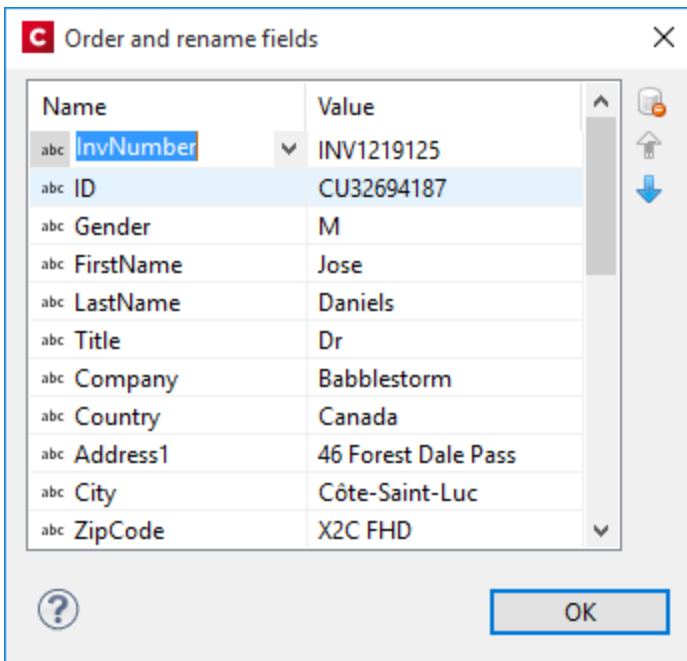
The field names in the **Data model** pane will also be visible in the Designer. Field names like Number and Number2 can be confusing when you are creating a template with variable data. In this exercise you will rename those fields and one of the fields in the detail table.

The first Number field actually contains an invoice number. Rename the field so that the field name makes this clear:

1. On the **Data model** pane, click the **Number** field (or any other field in the same **Extract** step).
2. On the **Step Properties** pane, click the **Order and Rename Fields** button .



3. In the dialog that appears, click the field **Number** and rename it to **InvNumber**.



There is an easier way to do this. We'll use that to rename the Number2 field in the detail table to make clear that that field contains a product number:

1. On the **Data model** pane, **right-click** the **Number** field in the detail table.
2. Select **Rename**.
3. Rename the field **Number** to **ProdNumber**.

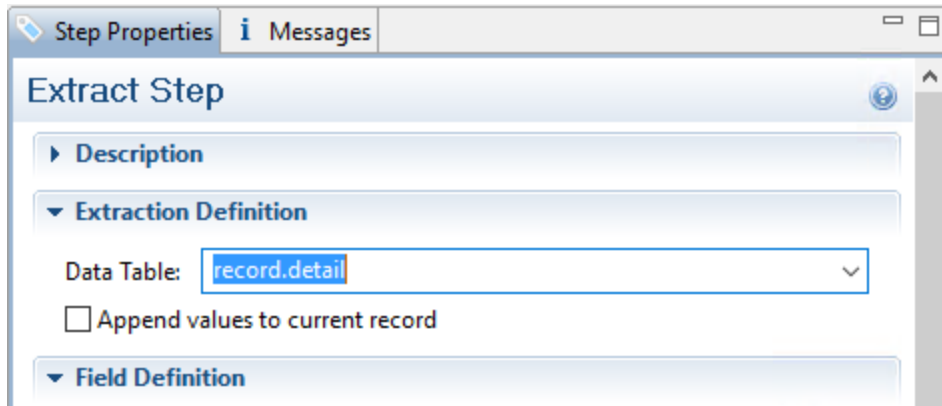
3. Rename a detail table

Renaming detail tables is especially useful when there are more detail tables in one record, or when a detail table contains another detail table. For this detail table, 'products' would be a better name.

This must be done on the Step Properties pane.

1. On the **Data model** pane, click one of the fields in the detail table.

2. On the **Step Properties** pane, click **Extraction Definition**.



3. Rename the detail table from **record.detail** to **record.products**.

Note: Note: A detail table's name should always begin with 'record.'

4. Click somewhere else on the **Step Properties** pane to update the data model. You will see the new name appear.

Changing the data type of fields

In addition to renaming them, fields need to be prepared for use in OL Connect Designer templates by setting their data type.

You have already set a default data format. This has had no effect on the extracted data, because by default, all fields are extracted as Strings (text). But for dates, numbers and currencies, other data types are available. Select a data type for these fields to make it easier to use them in a template.

1. Set a field's data type to Date

Two fields in the Data Model actually contain a date. Set their data type accordingly, so they can be interpreted and processed as such in the DataMapper and in templates.

1. On the **Data Model** pane, click **Date**.
2. On the **Step Properties** pane, under **Field Definition**, set the **Type** to **Date**.
3. Under **Data Format**, make sure **Date/Time Format** is set to **Automatic**.
4. Carefully compare the date in the data source to the date in the Data Model. Is the date parsed correctly?

<Date> has the format day/month/year (dd/mm/yyyy).

According to the ISO 8601 standard the date in the Data Model should be formatted : yyyy-mm-dd . So "25/06/2016" in the data should be represented as "2016-06-25" with the added time and time zone.

Tip: Dates in the Data Model can be displayed in local time or as UTC/ISO. From the menu, select **Window > Preferences > DataMapper > Default format**, and set the Data Model time stamps option.

5. Repeat this procedure for the **DueDate** field.

2. Set a field's type to Integer

Integers are whole numbers. In **olsg-data.XML**, there are a few elements that actually contain an integer. Set the data type of the respective fields in the data model accordingly.

1. On the **Data model** pane, select **Ordered**.

Tip: Hover with the mouse over the Extraction step. Extracted fields are displayed as links that you can click to select the field.

2. On the **Step Properties** pane, under **Field Definition**, set the **Type** to **Integer**.
3. Repeat this procedure for **Shipped** and **Backorder**.

3. Select a field and set its data type to Currency

In **olsg-data.XML**, there is also a number of elements that contain an amount of money. Set the data type of the respective fields in the data model to Currency.

1. On the **Data model** pane, click **InvSubTotal**.
2. On the **Step Properties** pane, under **Field Definition**, set the **Type** to **Currency**.
3. Repeat this procedure for **InvTaxTotal** and **InvTotal**.
4. Also repeat it for the currency fields in the detail table.

Tip: If the extraction fails after you set a field to a different type, check whether its data format is different than expected. You can set its format on the Step Properties pane, under Data Format. Note that the default data format is set for all fields in an Extraction step the moment that step is created.

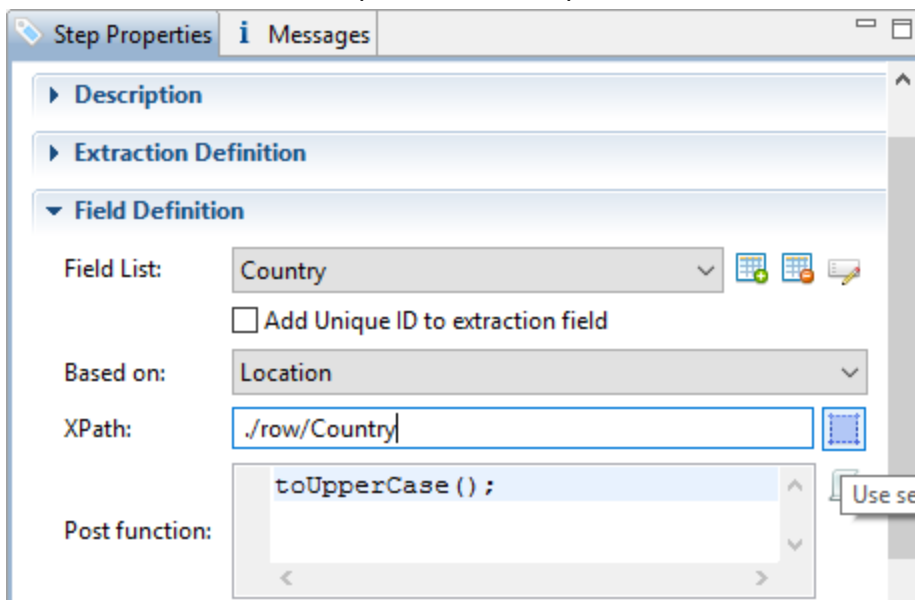
Fine-tuning using JavaScript

This part explains how to fine-tune a record set according to your needs. Understanding JavaScript is an advantage, but for this walkthrough you won't need to write any code yourself.

1. Post function: capitalize country

What if the extracted data is correct, but you'd like the data to be a little different: in capitals for example? Then you could use the "Post function". Code typed in the **Post function** field on the Step Properties pane will be executed after (hence: 'post') the extraction of the selected field, on the extracted data. In order to capitalize the letters of the Country field:

1. On the **Data model** pane, click **Country**.
2. On the **Step Properties** pane, in the **Post function** field, type `toUpperCase () ;`
This is a standard JavaScript function to capitalize the letters of a String (a text).



3. Click somewhere else on the **Step Properties** pane and check the result on the **Data Model** pane.

2. Split a field and keep one part of it

Sometimes you will want to keep only one part of the information that has been extracted to a field, and remove the rest. In `olsg-data.XML`, the Membership level is 'membershiplevel:bronze', 'membershiplevel:silver', or 'membershiplevel:gold'. The word 'membershiplevel' is superfluous. Here's how to remove it from the data field.

1. On the **Data model** pane, click **Membership**.
2. On the **Step Properties** pane, change **Based on** from **Location** to **Javascript**. The **Expression** field now shows this line of code:

```
data.extract('./row/Membership');
```

This is the code that normally extracts data from a specific location, in this case, the data found in the Membership element in a row element.

Replace this line by the following code:

```
var textParts = data.extract('./row/Membership').split(':');  
textParts[1];
```



This code extracts the data and then splits the resulting text in two parts, using a colon (':') as the separator.

The parts of the text are stored in a list variable called `textParts`. The first item in this list is `membershiplevel`, the second item is the level itself.


To get the second item from the list, you need to use `textParts[1]`; because the list is a JavaScript array and JavaScript arrays always start counting at 0.

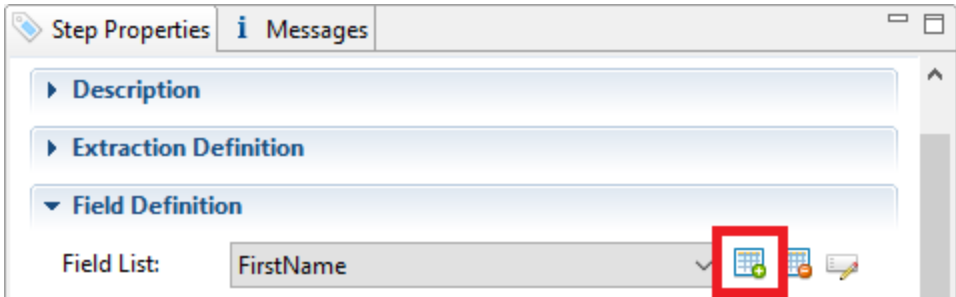
Note: The last line of code is essential: the value of the variable at the end of the code becomes the value of the data field.

3. Take a look at the result on the **Data Model** pane: the Membership field now only contains the membership level itself. The word 'membershiplevel' and the colon have been removed from the data field.

3. Add a concatenated field

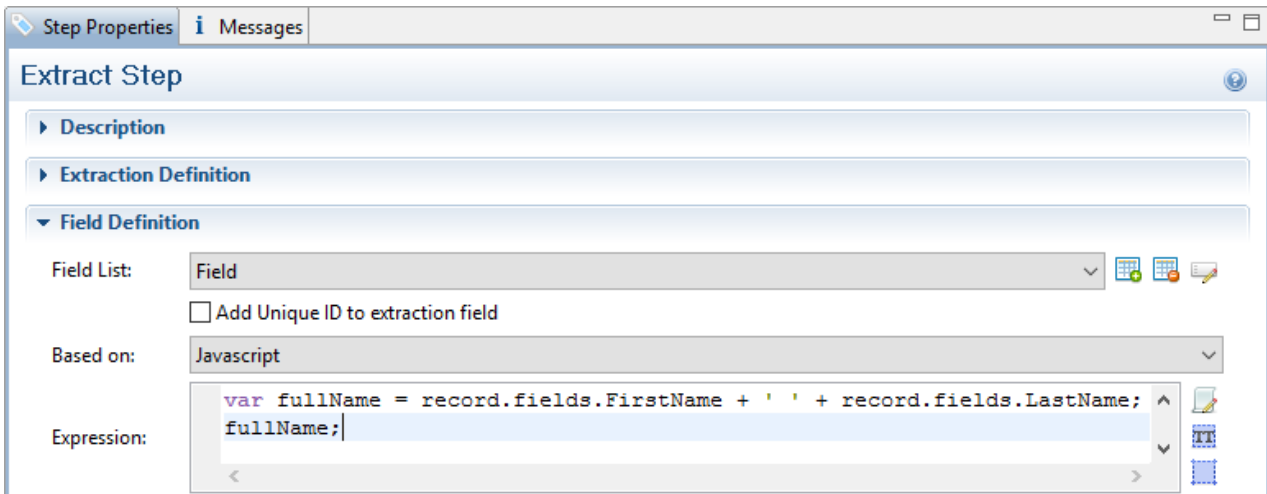
It can be very useful to add a field that isn't filled directly via an extraction. Let's add a field that combines information from two data fields.

1. On the **Steps** pane, click the first **Extraction** step.
2. On the **Step Properties** pane, under **Field Definition**, click the **Add field** button .



3. In **Expression** type this code:

```
var fullName = record.fields.FirstName + ' ' + record.fields.LastName;  
fullName;
```



4. Click another field in the **Step Properties** pane. Now you will see the result on the **Data model** pane.
5. Rename the new field to **FullName**.

What's next?

The Data Mapping Configuration is now ready. It can be used to extract data from any XML file that has the same structure as the sample file, `olsg-data.xml`.

Its Data Model can be used in Connect Designer, to create templates with variable data. To do this, you have to have the data mapping configuration and a Connect Designer template open at the same time. The sample data will be visible in the Data Model pane in Connect Designer.

Alternatively you could export the Data Model from the DataMapper and import it in the DataMapper again when extracting data from other types of files, or in Connect Designer when creating a template. The exported Data Model doesn't contain the data sample so in these cases the sample data will not be visible.

To get an introduction to the Connect Designer, please proceed with the [Designing an invoice walk-through](#) .

