



Transformation Manager

Version 5.2

Tutorial 16 - User Defined Functions



Copyright Notice

All information contained in this document is the property of ETL Solutions Limited. The information contained in this document is subject to change without notice and does not constitute a commitment on the part of ETL Solutions Limited. No part of this document may be reproduced in any manner, including storage in a retrieval system, transmission via electronic means or other reproduction medium or method (electronic, mechanical, photocopying, recording or otherwise) without the prior written permission of ETL Solutions Limited.

© 2013 ETL Solutions Limited. All rights reserved.

All trademarks mentioned herein belong to their respective owners.

Table of Contents

Tutorial 16	1
User Defined Functions.....	1
Prerequisites	1
Concepts	1
Information.....	1
Learning Objectives	1
Provide access to User Defined Functions.....	1
Exercise 1 - Start TM Designer and open the repository	2
Exercise 2 - Create the UserDefinedFunctions Project.....	2
Exercise 3 - Create the Transform for the UserDefinedFunctions Project	3
Exercise 4 - Define the External Functions to be Used With Your Project.....	5
Exercise 5 - Build Your UserDefinedFunctions Project	7
Exercise 6 - Run the Project.....	8
Exercise 7 - View the Target Data	10

Tutorial 16

User Defined Functions

This tutorial demonstrates how to use your own user defined functions within Transformation Manager. We have supplied a sample file containing user defined functions which we will use in a simple transformation project but the process of accessing these external functions will be the same for your own projects. The file we supply is called ExchangeRates.java, there is also an accompanying file called ExchangeRates.class. These can both be found in the following directory.

```
[TMHOME]\tutorials\tutorial_16_User_Defined_Functions\Java\exchange  
directory
```

Prerequisites

Before starting this tutorial we recommend that you have completed the following tasks.

- 1) Transformation Manager has been installed.
- 2) An appropriate license has been installed.
- 3) The tutorial resources including data models, samples and source and target data stores have been downloaded and extracted to your Transformation Manager home directory.
- 4) You have completed tutorials 1, 2, 3, 4 and 6.

Concepts

Transformation Manager lets you use your own defined functions. These can be created inside Transformation Manager or implemented from external java files. This tutorial uses functions that are completely external to Transformation Manager using the External Defined Functions functionality.

Information

The project we will create assumes that the target wishes to have all prices in GBP. Therefore GBP gets transformed without manipulation, USD gets converted if the USD code comes in and all other codes are classed as being euros. Evidently the fact that other codes convert to euro could have been found during source analysis.

Learning Objectives

This tutorial shows you how to use externally defined functions in Transformation Manager.

Provide access to User Defined Functions

You will learn how to add external user defined functions to the functions you can use within Transformation Manager.

Exercise 1 - Start TM Designer and open the repository

This exercise will start Transformation Manager and open the tutorials repository ready to create a new project.

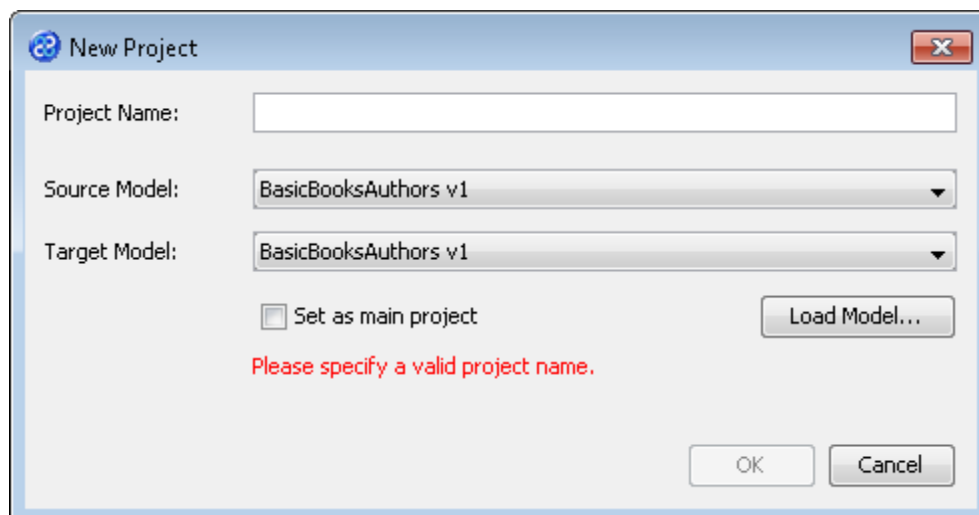


- 1) Looking at your desktop, find the icon that looks like this,
- 2) Then you have two options.
 - a) Using your mouse double-click the icon using the primary mouse button.
 - b) Using your mouse press the secondary mouse button, commonly the right-mouse button, while the cursor is over the icon. This will open the pop-up menu and then select **Open** from the available options.
- 3) Once Transformation Manager has opened and displays its interface connect to your repository, **MyExamples**.

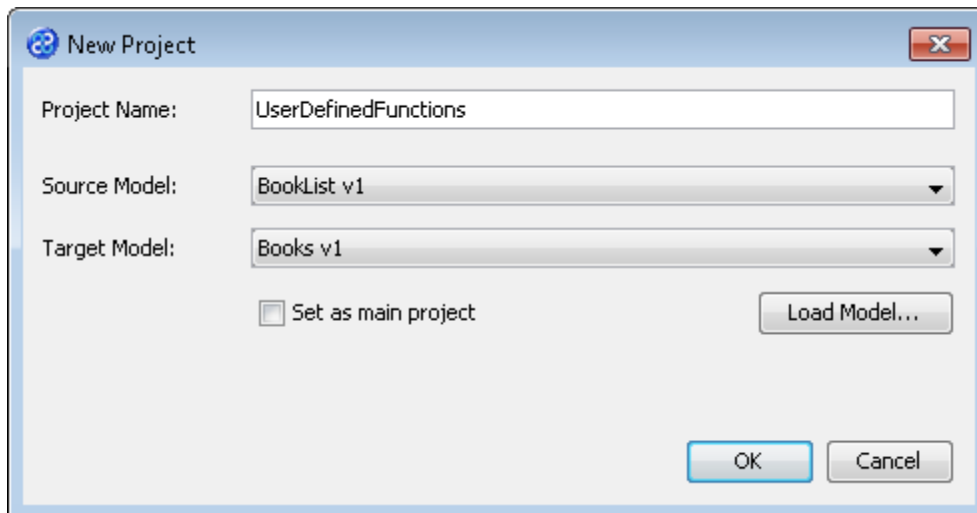
Exercise 2 - Create the UserDefinedFunctions Project

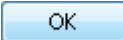
This exercise creates a new project where we will use a Derby relational database data model called **Books v1** and a flat file data model called **BookList v1**.

- 1) Using your mouse click on the **File** option from the menu bar of TM Designer.
- 2) From the menu, click once on the **New Project...** option.
- 3) The **New Project** window will open.



- 4) In the **Project Name** field we will provide a name for the project. Let's call our project **UserDefinedFunctions** by typing the name into the field.
- 5) Now we must select the target and source models for our project. In the **Source Model** field select **BookList v1** from the list. In the **Target Model** field select **Books v1**.

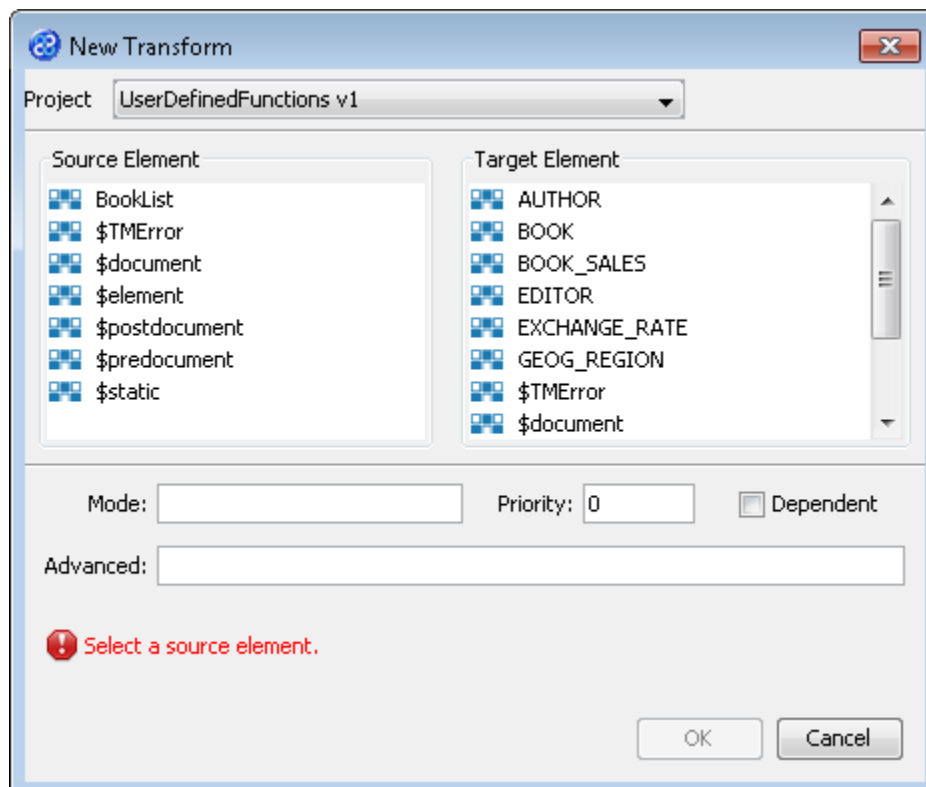


- 6) Now click once on the  button. Your project will be created and be displayed in the **Projects** pane.

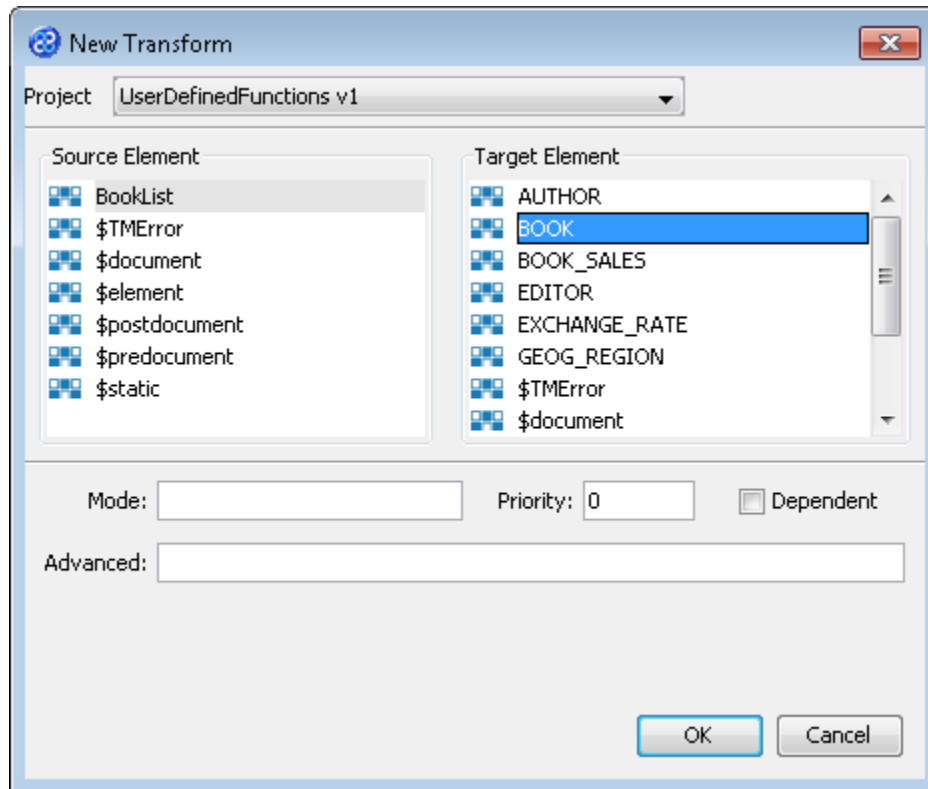
Exercise 3 - Create the Transform for the UserDefinedFunctions Project

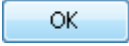

This exercise creates our transform.

- 1) Click once on the **File** menu bar option.
- 2) Click once on the **New Transform...** option from the sub-menu.
- 3) This will open the **New Transform** window. Ensure that you have the correct project selected in the **Project** drop down list, in our case this will be **UserDefinedFunctions v1**.



- 4) With the **New Transform** window open, we will select the elements for the source and target. In this case the source element will be **BookList** and the target element will be **BOOK**. The **New Transform** window will look like the one below.



- 5) Click once on the  button to create your new transform.
- 6) The Editor pane will open ready for you to write your transform code once the **New Transform** is created.
- 7) Now we will write our transform as shown below. You will see an error icon appear,  and this is because we have not yet defined our external function, `ExchangeRates:getUSDConversionRate()`. This will be our next exercise.

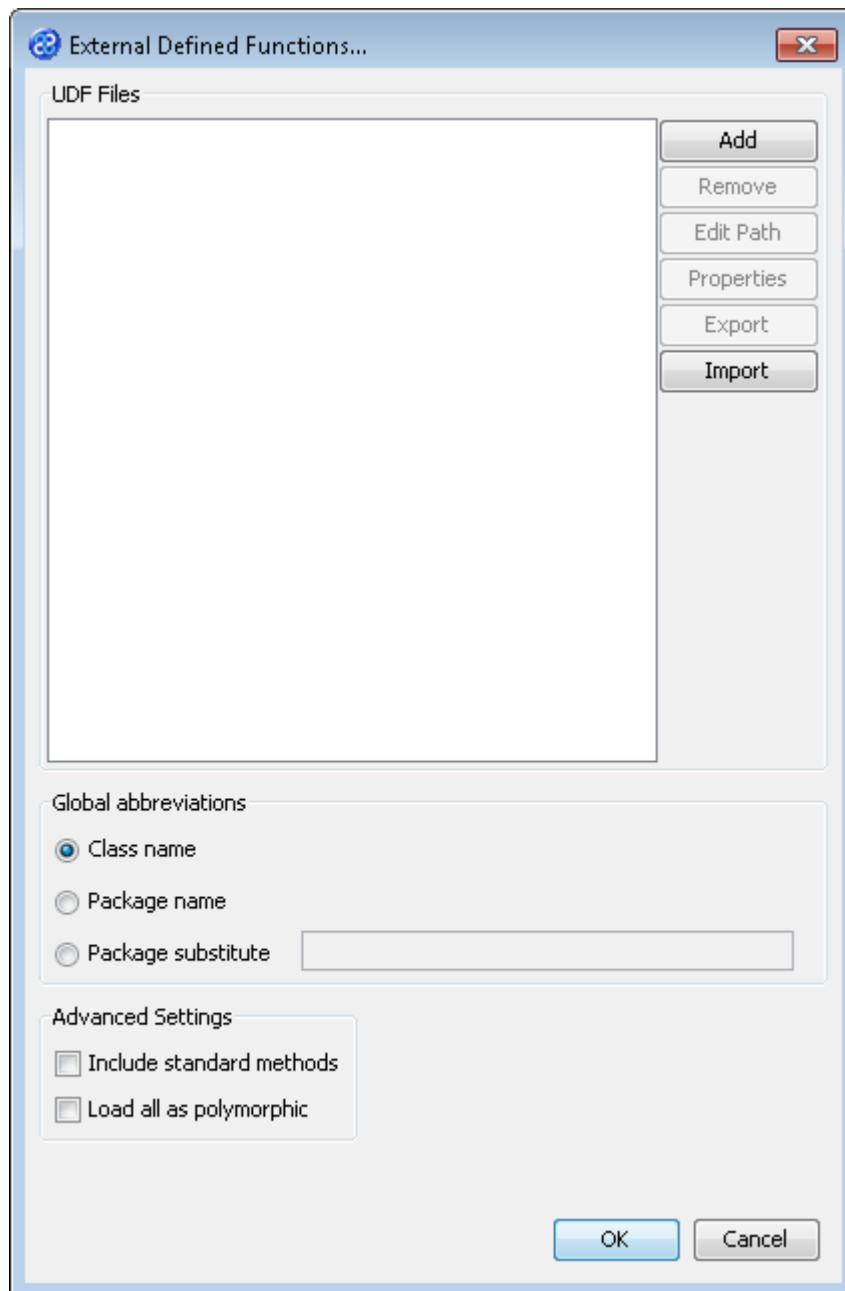
```
TITLE := title;
<ID> := <id>;
--If we want all currencies being input to be in the form of
--GBP then we could use a case statement for the codes
--and convert USD or EUR (assumed to be unknown codes) to GBP
CASE currency OF
123 : BEGIN
PRICE := price;
END;
456 : BEGIN
PRICE := price / ExchangeRates:getUSDConversionRate();
END;
OTHERWISE : BEGIN
PRICE := price / ExchangeRates:getEURConversionRate();
END;
END_CASE;
```

- 8) Click once on the **File** menu bar option.
- 9) Click once on the **Save** menu bar option to save your transform.

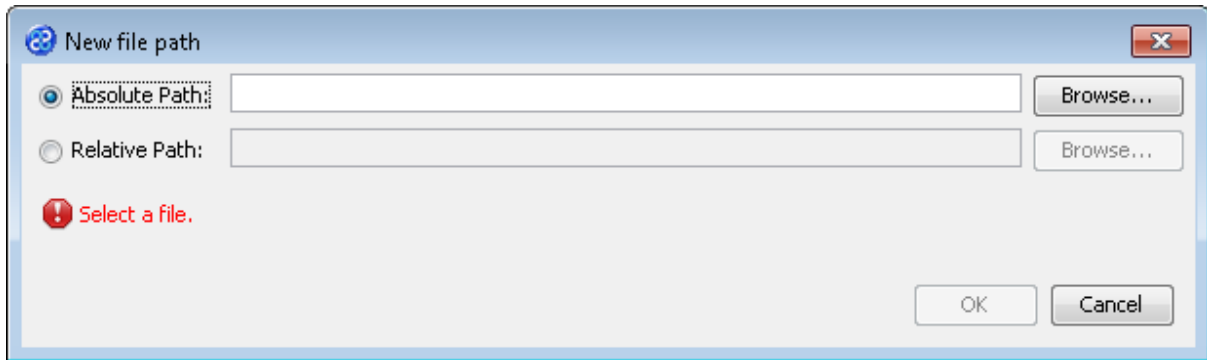
Exercise 4 - Define the External Functions to be Used With Your Project

In this exercise we will tell Transformation Manager where to find the externally defined function we need for our transformation project.

- 1) Click once on the **File** menu bar option.
- 2) Click once on the **External Defined Functions...** option from the sub-menu.
- 3) This will open the **External Defined Functions...** window as shown below. Ensure that the **Global abbreviations** radio button option is set to **Class name**.



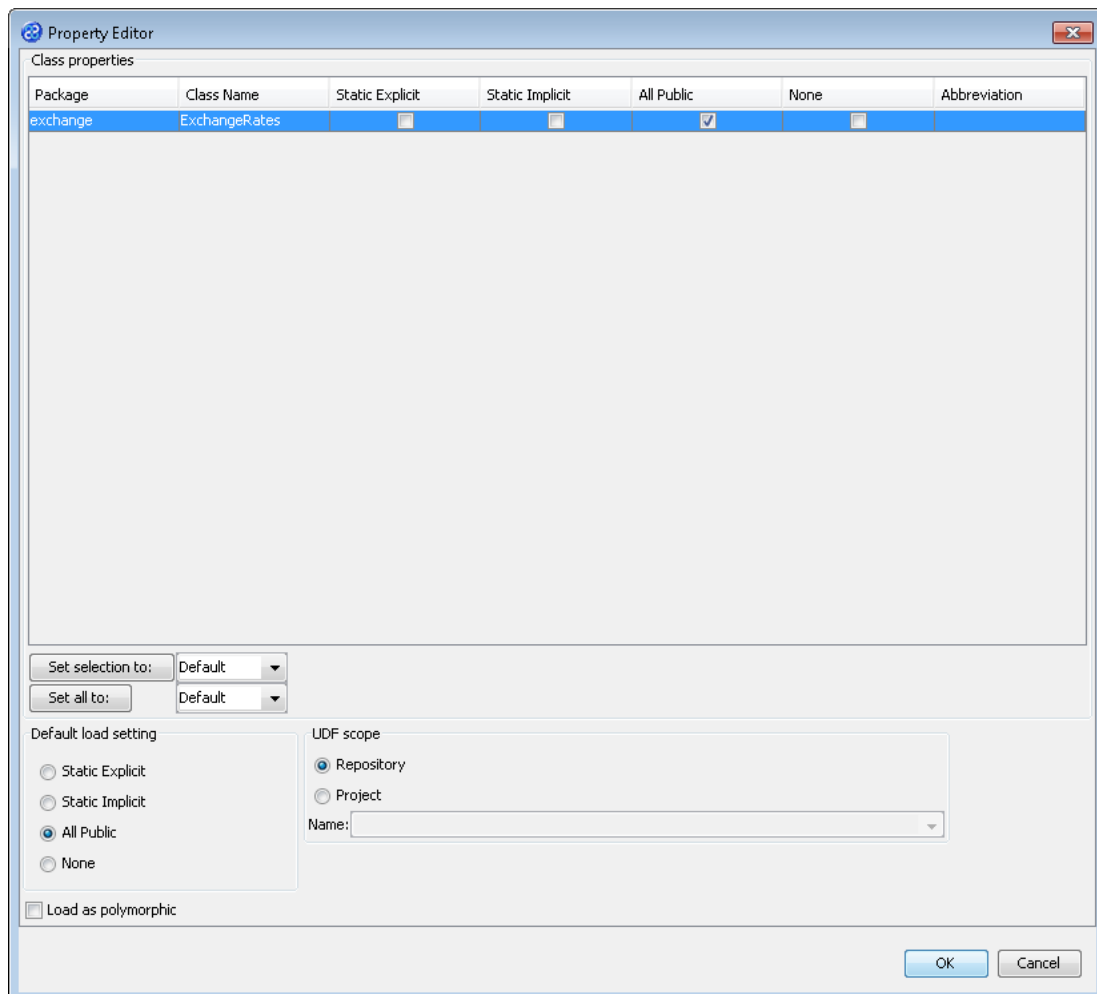
- 4) Click once on the **Add** button. This will open the New file path window, shown below, where we will tell Transformation Manager where to find the external functions.

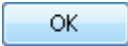
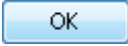


- 5) We will use the Absolute Path option for this exercise so check that is selected. Click once on the button to open the Select File window and navigate to the ExchangeRates.class file in the following directory.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_16_User_Defined_Functions\Java\exchange\

- 6) Click once on the **Select File** button to set the path for the Absolute Path field.
- 7) Click once on the **OK** button. This will open the **Property Editor** window as shown below. In this exercise we can accept the default setting for the properties. In this case the **All Public** tick box with the **UDF scope** set to **Repository** means that all public functions in the ExchangeRates class will be available for use in all your transforms in this repository.

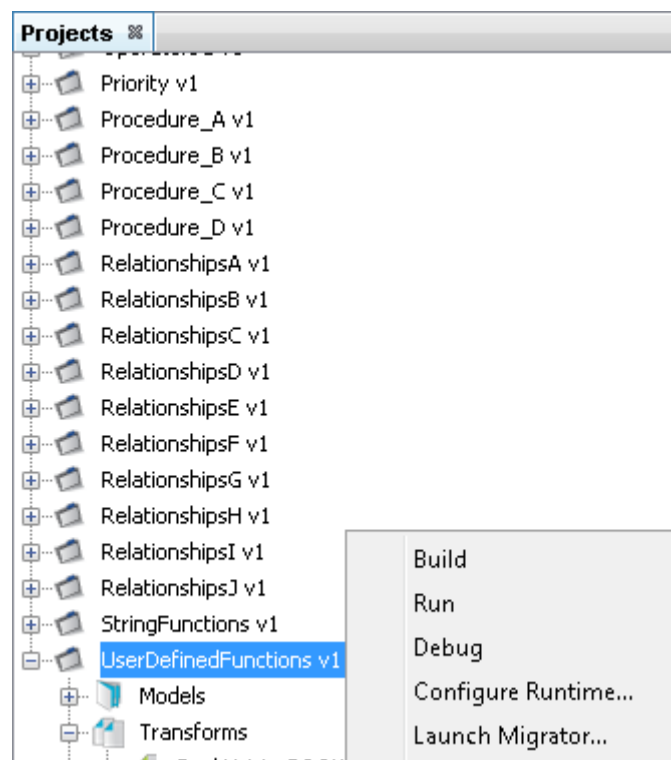


- 8) Click once on the  button to close the **Property Editor** window and return to the **External Defined Functions...** window.
- 9) Click once on the  button to complete the definition of our external functions. Note that the error icon will disappear from the transformation code since the functions have been defined.

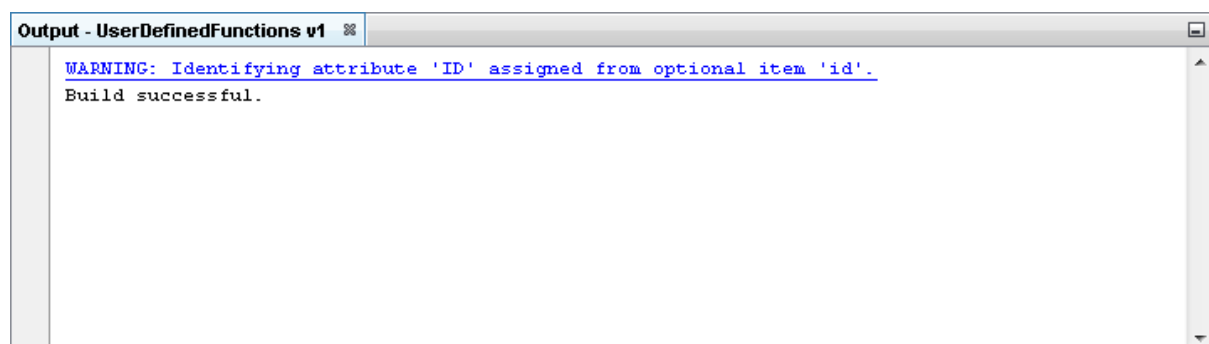
Exercise 5 - Build Your UserDefinedFunctions Project

Now we will build our project.

- 1) Move your mouse pointer so that it is over the title of the project you are working on and use the right mouse or secondary mouse button to display the context menu.



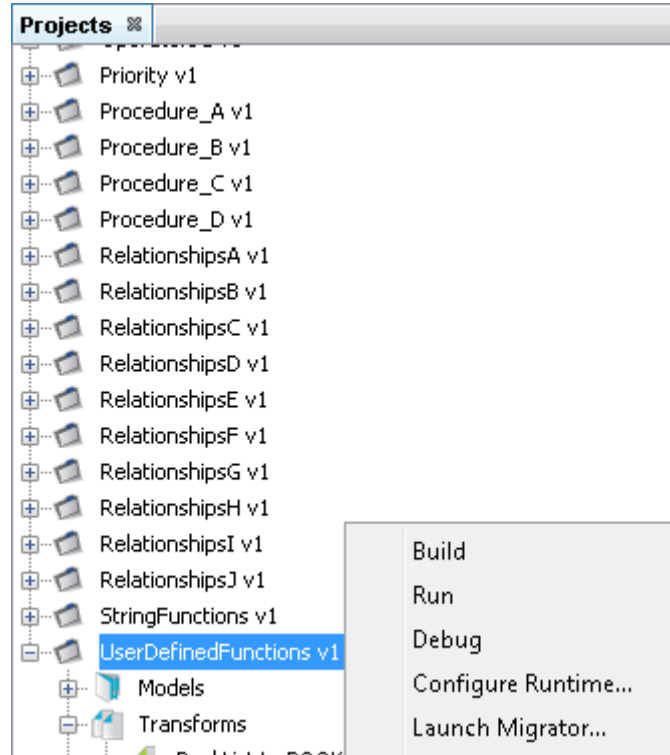
- 2) Click once on the **Build** option from the menu.
- 3) The **Output** pane will open and show you if there are any errors in your transform code. In this exercise you should not have any error messages and you should see a message stating **Build successful**.



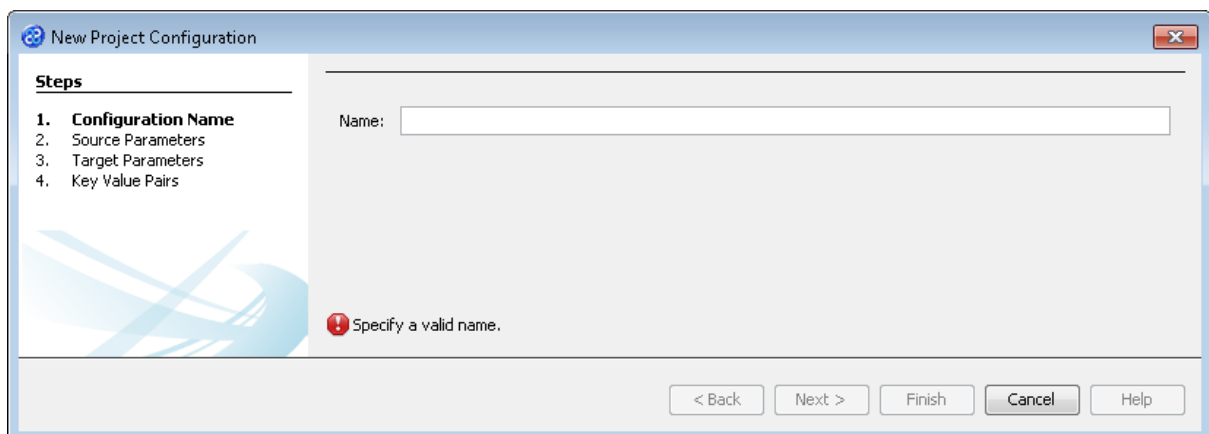
Exercise 6 - Run the Project

Now let's Run our transformation to see what results are produced.

- 1) Move your mouse pointer so that it is over the title of the project you are working on and use the right mouse or secondary mouse button to display the context menu.



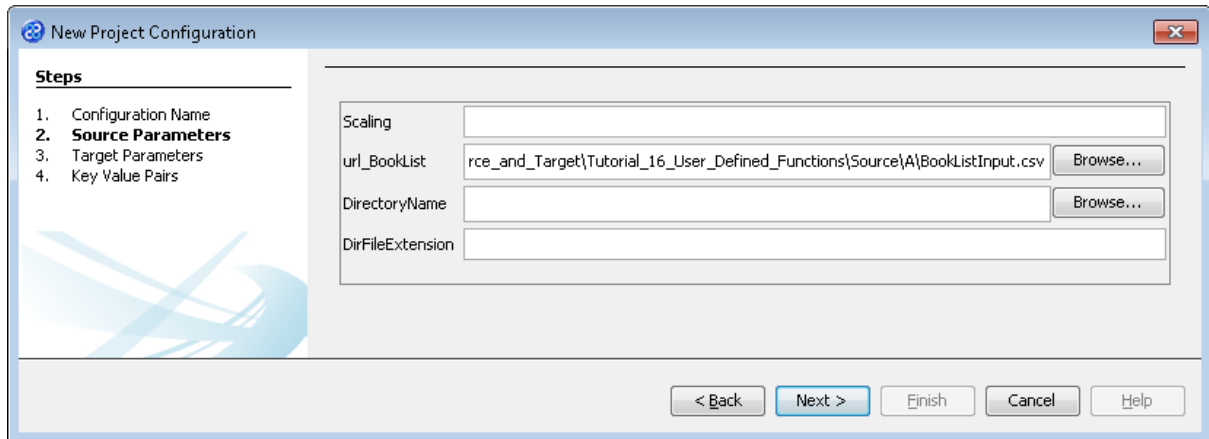
- 2) Click the **Run** option from the menu. TM Designer will now open the New Project Configuration window where you will set the connection information for the project. Notice the additional step for the error handler.



- 3) Let's give this configuration the name Config1. Type **Config1** into the **Name** field.
- 4) Click the **Next >** button to move to **Step 2. Source Parameters** where we will provide the details for the source data store connection.
- 5) Go to the **url_BookList** field of the **Source** pane. In this field we will specify where to get our data store file. Enter the following into the field or you can use the button to select the file.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_16_User_Defined_Functions\Source\A\BookListInput.csv

6) The **Source Parameters** step should look similar to the image below.



The screenshot shows the 'New Project Configuration' dialog box. On the left, a 'Steps' list shows: 1. Configuration Name, 2. **Source Parameters**, 3. Target Parameters, 4. Key Value Pairs. The main area contains the following fields: 'Scaling' (empty), 'url_BookList' (filled with 'rce_and_Target\Tutorial_16_User_Defined_Functions\Source\A\BookListInput.csv'), 'DirectoryName' (empty), and 'DirFileExtension' (empty). There are 'Browse...' buttons next to the 'url_BookList' and 'DirectoryName' fields. At the bottom, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted in blue.

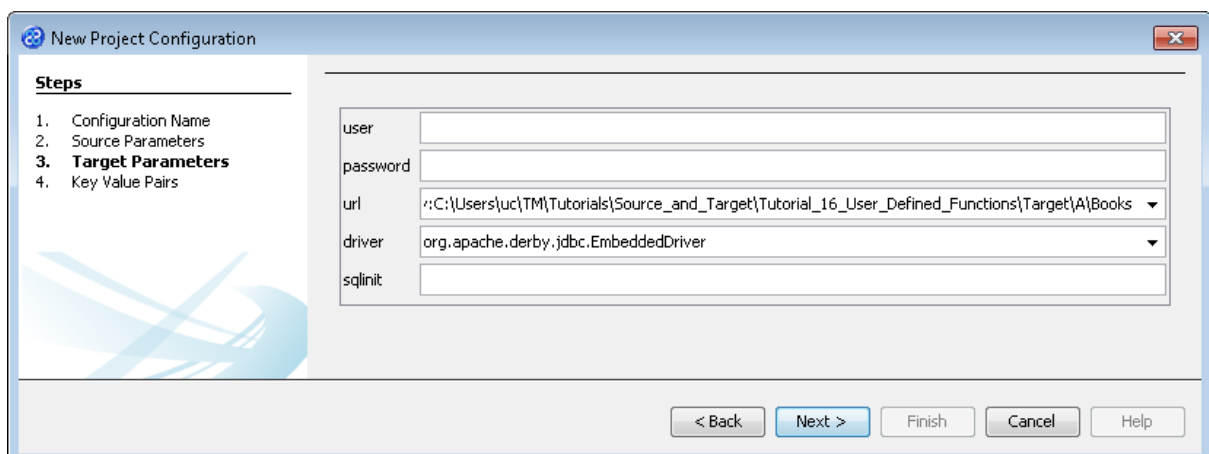
7) Click the **Next >** button to move to **Step 3. Target Parameters** where we will provide the details for the target data store connection.

8) Go to the **url** field. You will now need to provide the details of where to go to connect to the source data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> part of the list item, including the angle brackets with the directory where the Derby database is stored. This will be in the following location.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_16_User_Defined_Functions\Target\A\Books

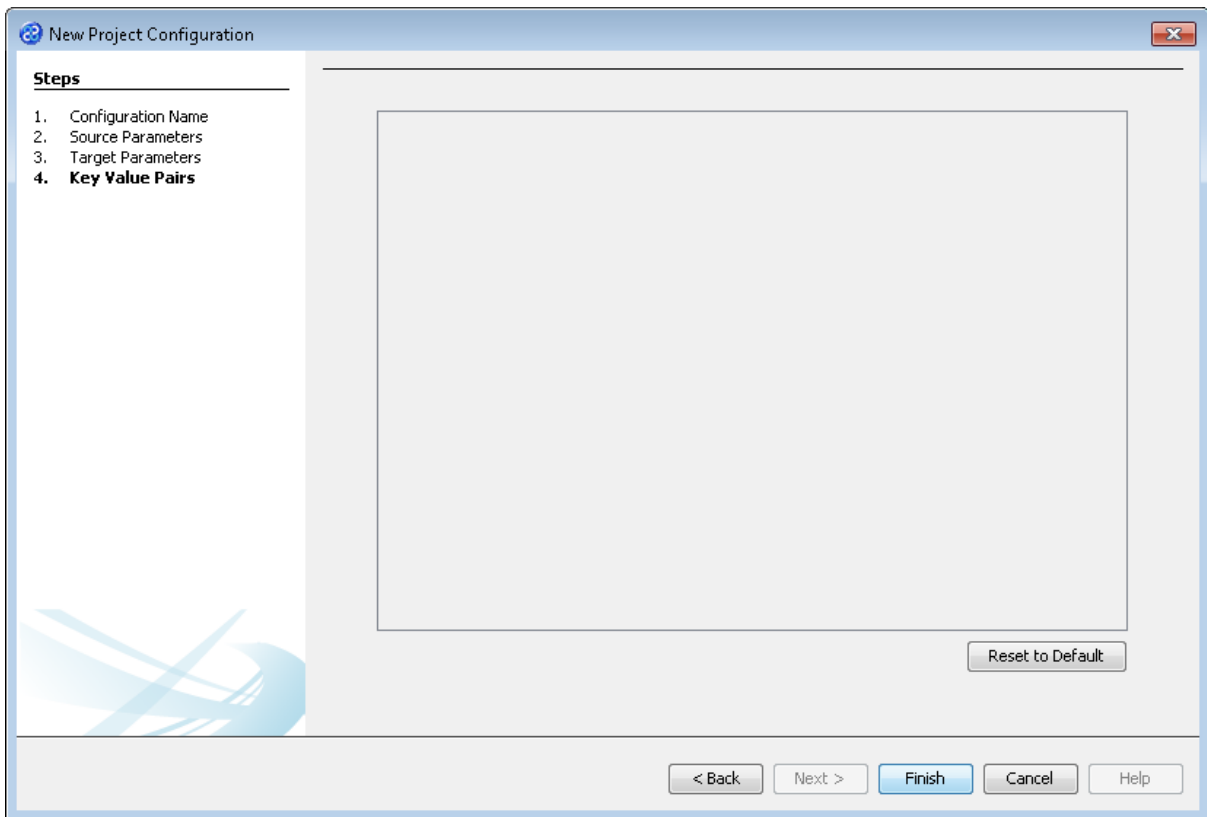
9) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The New Project Configuration window will look similar to the image below. **user**, **password** and **sqlinit** do not require values.

10) The **Target Parameters** step should look similar to the image below.



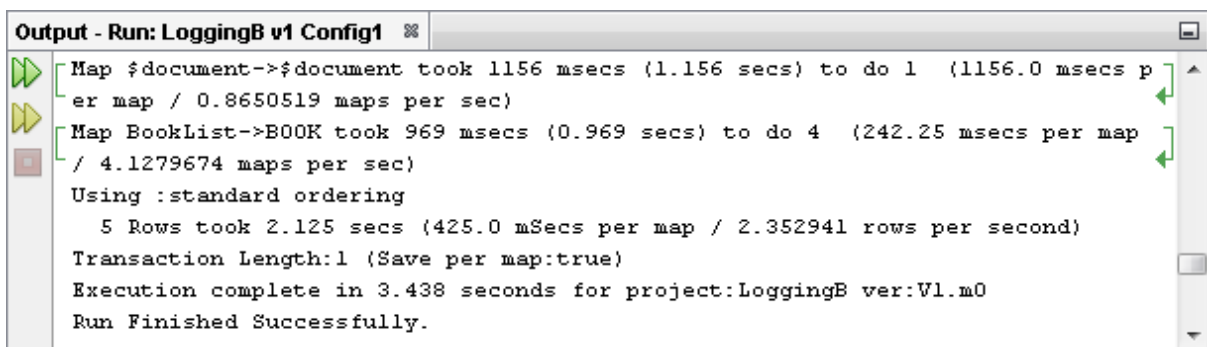
The screenshot shows the 'New Project Configuration' dialog box. On the left, a 'Steps' list shows: 1. Configuration Name, 2. Source Parameters, 3. **Target Parameters**, 4. Key Value Pairs. The main area contains the following fields: 'user' (empty), 'password' (empty), 'url' (filled with 'jdbc:derby::C:\Users\uc\TM\Tutorials\Source_and_Target\Tutorial_16_User_Defined_Functions\Target\A\Books'), 'driver' (filled with 'org.apache.derby.jdbc.EmbeddedDriver'), and 'sqlinit' (empty). At the bottom, there are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The 'Next >' button is highlighted in blue.

11) Click the **Next >** button to move to **Step 4. Key Value Pairs**. We do not need to specify any key value pairs in this exercise.



12) Click the **Finish** button to save the project configuration and to run the project within TM Designer. This performs the same execution as the **Launch Migrator...** option from the project context menu.

13) TM Designer will open the Output - Run pane with the results of running the project as shown below.

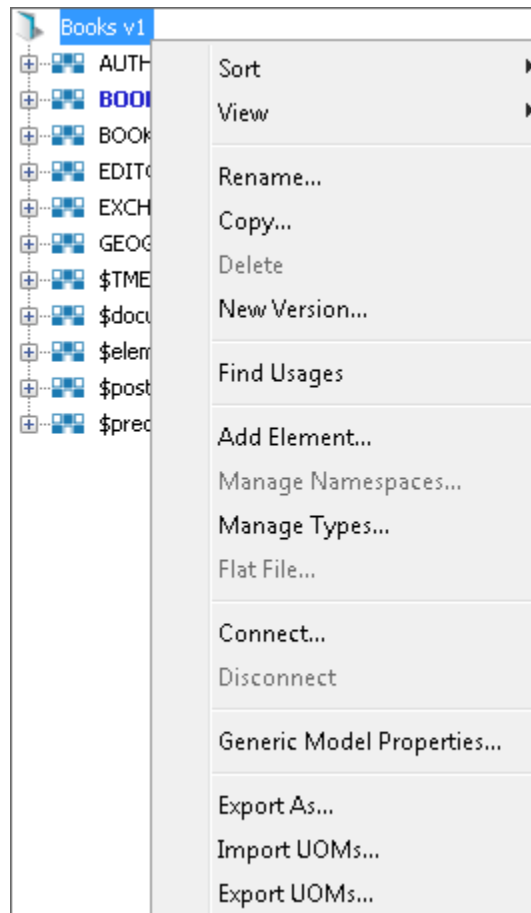


14) Let's now check the data using the **View Data...** functionality.

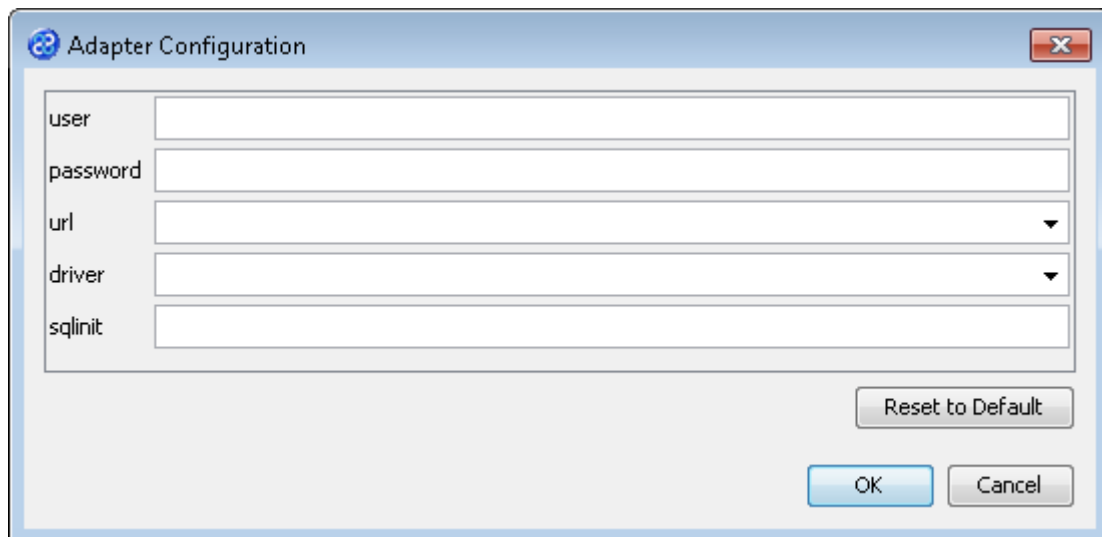
Exercise 7 - View the Target Data

Now let's check our error handler and target data stores to see if our data has been transformed as expected.

- 1) Move the cursor over the name of the data model in the **Transform Target** pane and display the context menu for the data model.



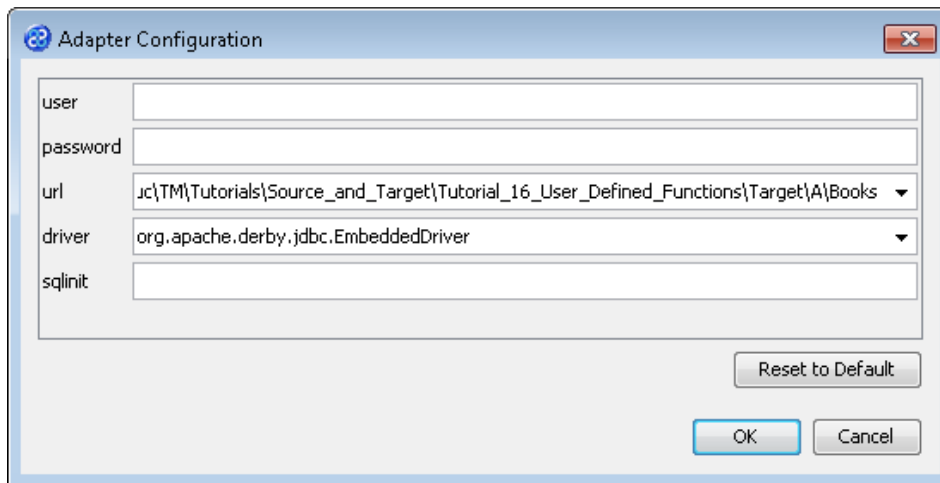
- 2) Select the **Connect...** option from the menu. The Adapter Configuration window will open ready for the connection details to be entered for the error handler data store.



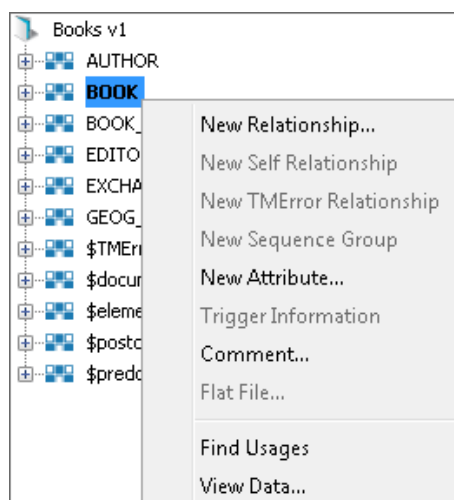
- 3) Go to the **url** field. You will now need to provide the details of where to go to connect to the source data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> part of the list item, including the angle brackets with the directory where the Derby database is stored. This will be in the following location.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_16_User_Defined_Functions\Target\A\Books

- Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The New Project Configuration window will look similar to the image below. **user**, **password** and **sqlinit** do not require values.
- The adapter configuration should look similar to the one below.



- Move your cursor over the **BOOK** element in the data model and display the context menu for the element.




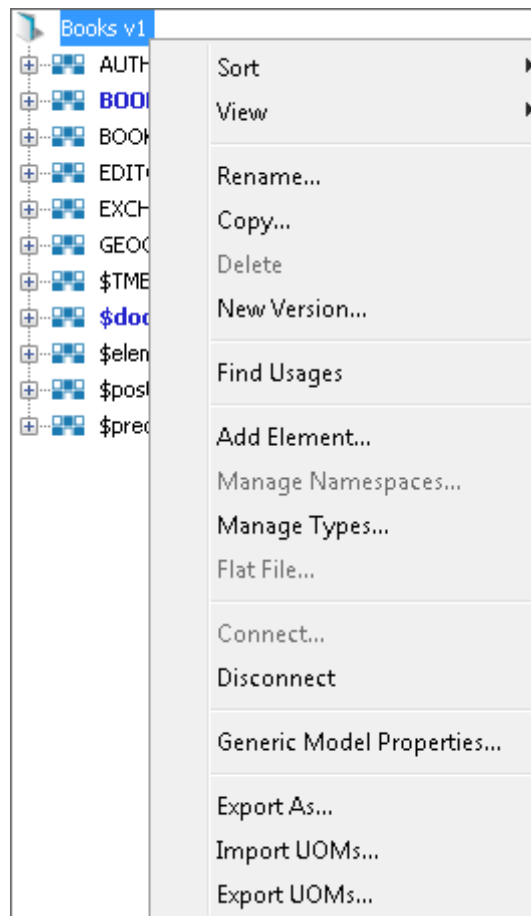
- Select the **View Data...** option from the menu. This will open a new pane in the Editor pane displaying the **BOOK** data as shown below.

Books v1 : BOOK

CURRENCY	ID	PRICE	EDITOR_ID	IS_FICTION	PUBLISHED_DATE	TITLE
GBP	111	20	2001	Y	1999-01-06	Grass
USD	222	30	2001	Y	2002-01-09	Trees
USD	333	40	2002	N	1900-01-03	Soil
GBP	444	50		Y	1965-01-12	Soot
	682	55				Rope
	683	51				Twine
	684	32				String
	685	12				Wool

Hide empty columns Rows 8 Maximum Rows 200 Refresh

- 8) Now let's look at the transformed data in the data pane above. The records we are concerned with are Rope, Twine, String and Wool or the last four records in the image. We know that the books Rope and String are priced in GBP so the data should have been transformed without change and this is the case. The book Wool is priced in USD and this has been transformed from a price of 21 to 12. In this instance the transformation has used the externally defined function to calculate the price. The function used a conversion rate of 1.8148. The book called Twine has a currency value of 890, neither GBP nor USD so used the Euro conversion rate of 1.2287 leading to a price of 51. Note that the prices have been rounded.
- 9) Let's close the data pane and disconnect from the data store. Click on the  icon to close the data view pane.
- 10) Now move the cursor over the name of the data model in the **Transform Target** pane again and display the context menu for the data model.



- 11) Note that the **Disconnect** option is active now. Click on this to disconnect from the data store.