by experts
for experts

Transformation Manager
Version 5.2

Tutorial 9 - Functions and Hash Maps

Transformation Manager

Copyright Notice

Table of Contents

Tutorial 9

Functions and Hash Maps

This tutorial demonstrates how to use some of the more commonly used functions in Transformation Manager. We will focus on three specific areas, string manipulation, batching data for creation or update and managing hash maps for data arrays. Transformation Manager provides you with a large number of functions for many purposes and these are only a small number of those that are available.

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

Functions are blocks of code that perform a specific task. You will provide a function with some data or parameters and the function will provide you with the answer that it is designed to provide.

Information

Transformation Manager has a large number of built-in functions to help you in all the situations you may come across. There is of course the chance that you will need your own functions or support from us. If you are unsure of what to do then please contact us.

For this tutorial the specific functions used are listed below with links to the relevant help page for the function. During the tutorial you will create three projects. The first project demonstrates the use of string functions, the second uses batch functions and the last shows you the use of hash maps in a transformation.

Below we have listed the functions that we will use in this tutorial. While these are all shown in upper case it is not necessary to write them in upper case as Transformation Manager will recognise both upper and lower case instances of a function.

String Functions

LENGTH

SUBSTRING

PADSTRING

TRIM

INDEXOF

Batching Functions

ALLOWBATCHONCONSTRAINTS

SETBATCHONCONSTRAINTSIZE

Java Functions (Relevant to hash maps.)

MAKEJOBJECT

JMAPADDKEYVALUE

JMAPCONTAINSKEY

JMAPGETBYKEY

Other Functions

PRINTLN

Learning Objectives

String manipulation functions

You will learn how to use string functions such as LENGTH, &, SUBSTRING, PADSTRING, TRIM and INDEXOF. We will also see the use of PRINTLN.

These functions show how strings can be manipulated during migration to ensure the data is suitable for the target system.

Database batching functions

You will learn how to use ALLOWBATCHONCONSTRAINTS and SETBATCHONCONSTRAINTSIZE.

These functions allow data to be written into a database efficiently using batching to group the data into suitable sized batches, prior to connecting to the database to insert or update the target data.

Hashmap functions

You will learn how to use MAKEJOBJECT, JMAPADDKEYVALUE, JMAPCONTAINSKEY and JMAPGETBYKEY.

These functions allow a simple manner in which data can be cached and stored for use in other maps within the project.

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.

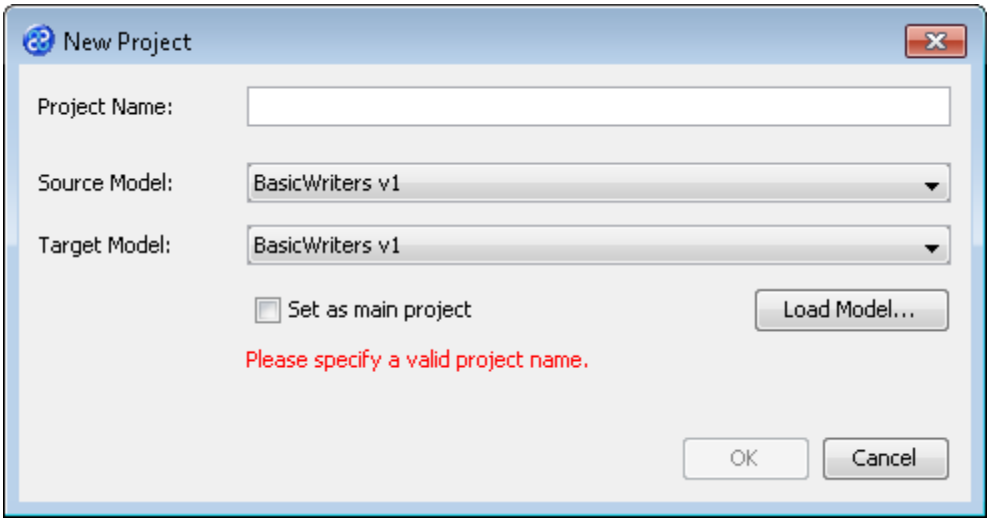
1) Looking at your desktop, find the icon that looks like this, 

.
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 to open the pop-up menu and select Open from the available options.
3) Once Transformation Manager has opened and displays its interface connect to your repository, **MyExamples**.

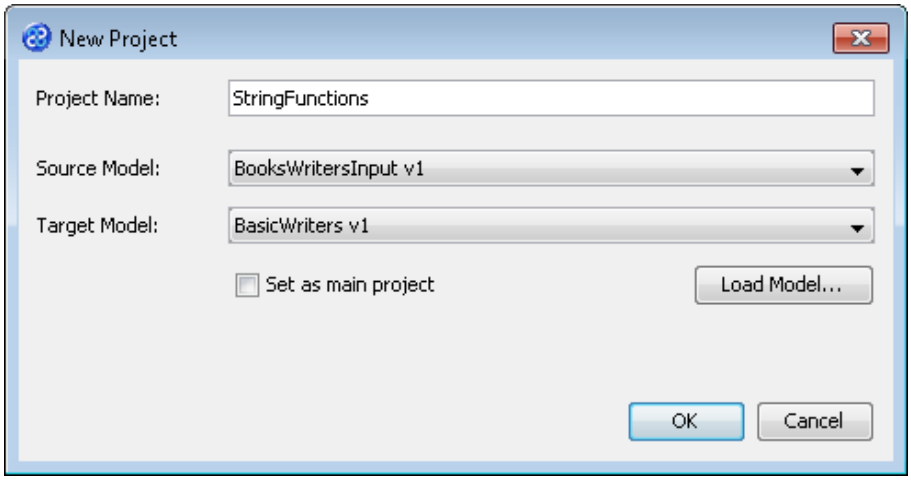
Exercise 2 - Create a New Project

This exercise creates a new project where we will use the flat file data model called **BooksWritersInput v1** as the source data store and a relational database data store called **BasicWriters v1** as the target data model.

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 **StringFunctions** 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 **BooksWritersInput v1** from the list. In the **Target Model** field select **BasicWriters v1**.

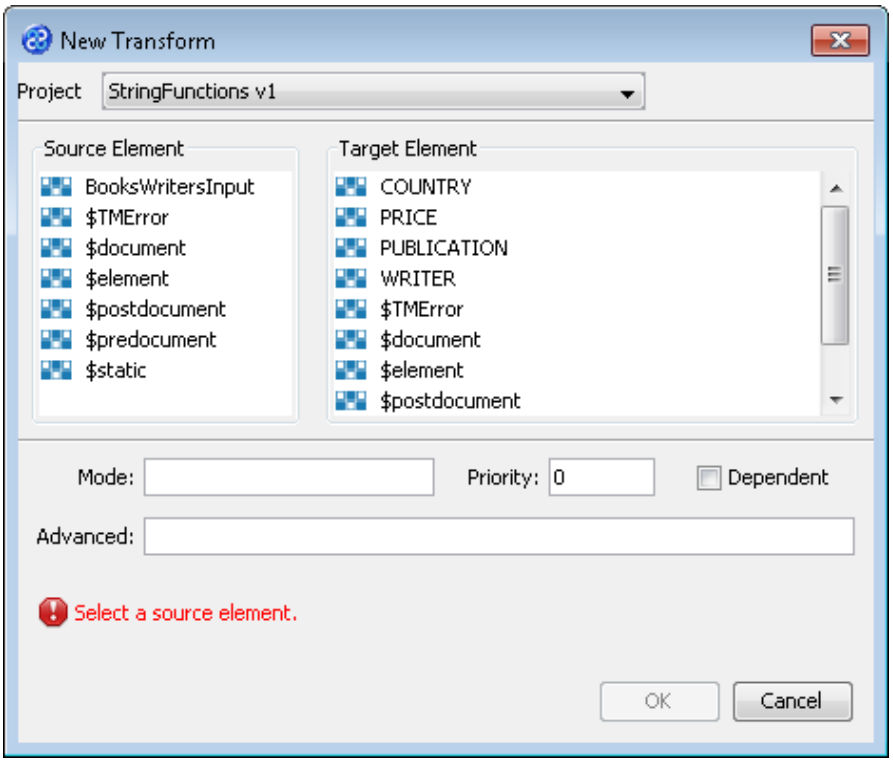


6) Now click once on the OK button. Your project will be created and displayed in the **Projects** pane.

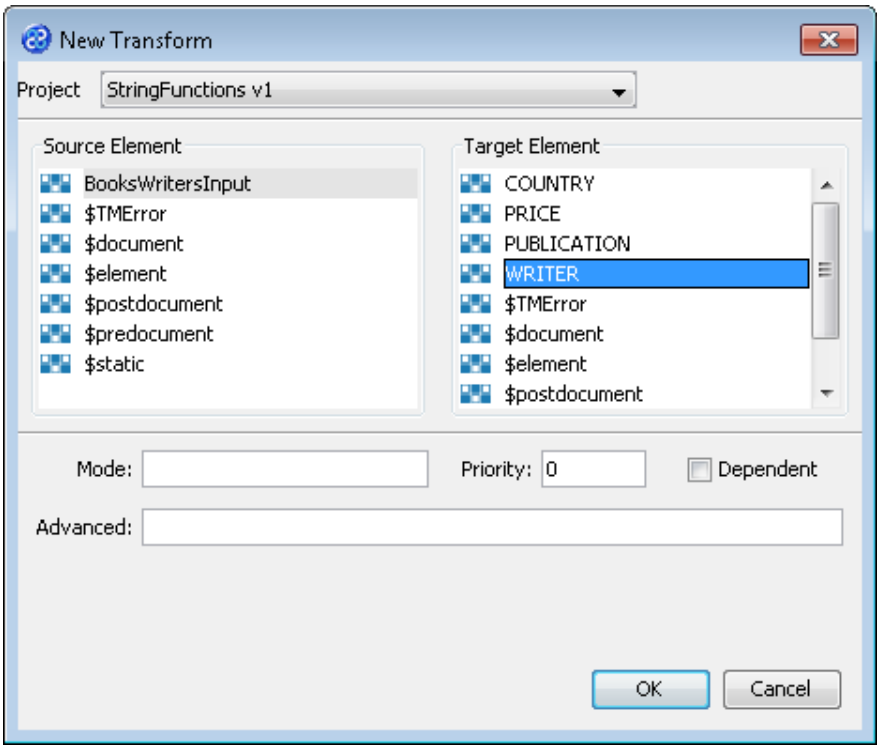
Exercise 3 - Create the String Function Transform

This exercise creates a transform which uses string functions to update data.

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 **StringFunctions 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 **BooksWritersInput** and the target element will be **WRITER**. The **New Transform** window will look like the one below.



5) Click once on the OK 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. Initially we will declare a local variable called myStr. Then we add the basic assignment of <ID> to <id>. Then we have the transform code that sets the FIRST_INITIAL value which uses the length of the surname or if the surname starts with an H. In these cases we take the first character of the <writer forename> and set the FIRST_INITIAL. The lines of code after this use the PRINTLN function to output information about the transformation as it executes using our local variable. The last part of the transformation pads the <writer surname> and assigns it to the NAME attribute.

```
LOCAL
myStr : string;
END_LOCAL;

<ID> := <id>;

IF (LENGTH(<writer surname>) > 6 OR INDEXOF(<writer surname>,'H') > 0) THEN
FIRST_INITIAL := SUBSTRING(<writer forename>,1,2);
END_IF;

myStr := '  test a   ';

PRINTLN('Testing how TRIM works on the fly');
PRINTLN('Here is the test string_' & myStr & '_more text to see the
spaces');
PRINTLN('Here is the test string_' & TRIM(myStr) & '_with no spaces before
or after the text');

NAME := TRIM(myStr) & PADSTRING(<writer surname>,8,true,'_');
```

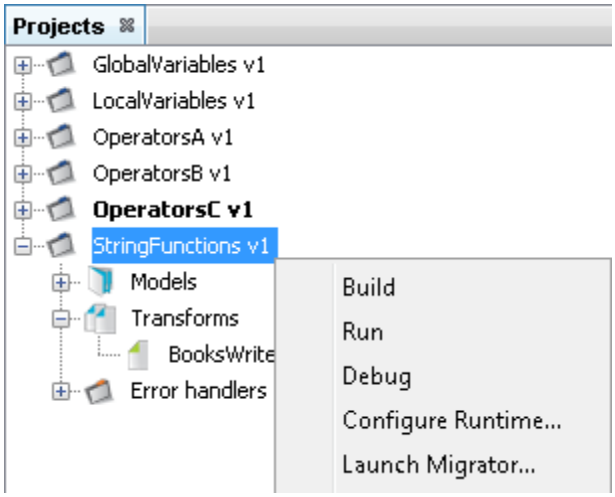
8) We have completed our transform.

9) Click once on the File menu bar option.

10) Click once on the Save menu bar option to save your transform.

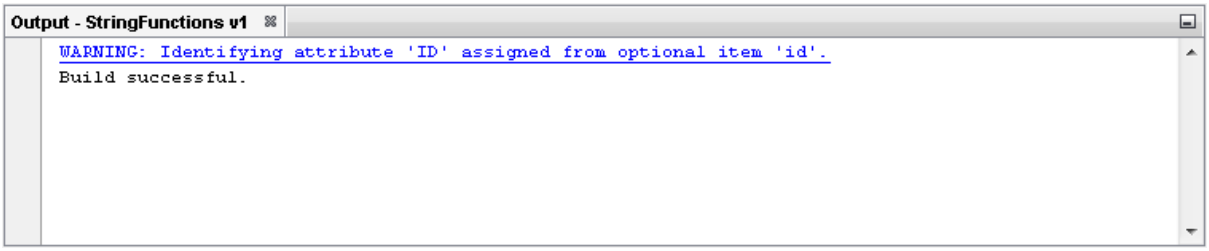
Exercise 4 - Build Your StringFunctions Project

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.

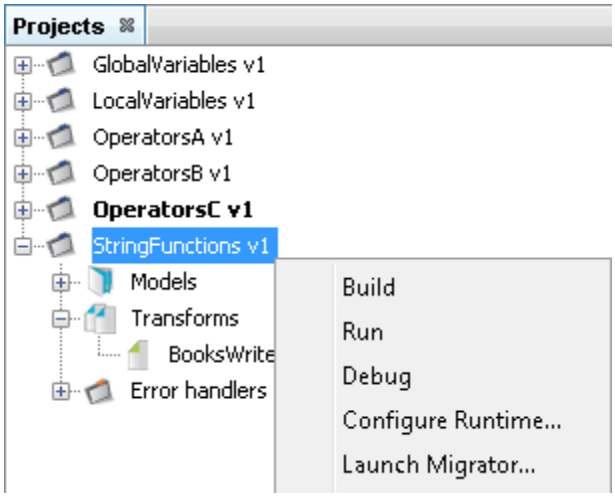
3) The **Output** pane will open and show you if there are any errors in your transform code. In this exercise you should not have any error messages and you should see a message stating **Build successful.**.



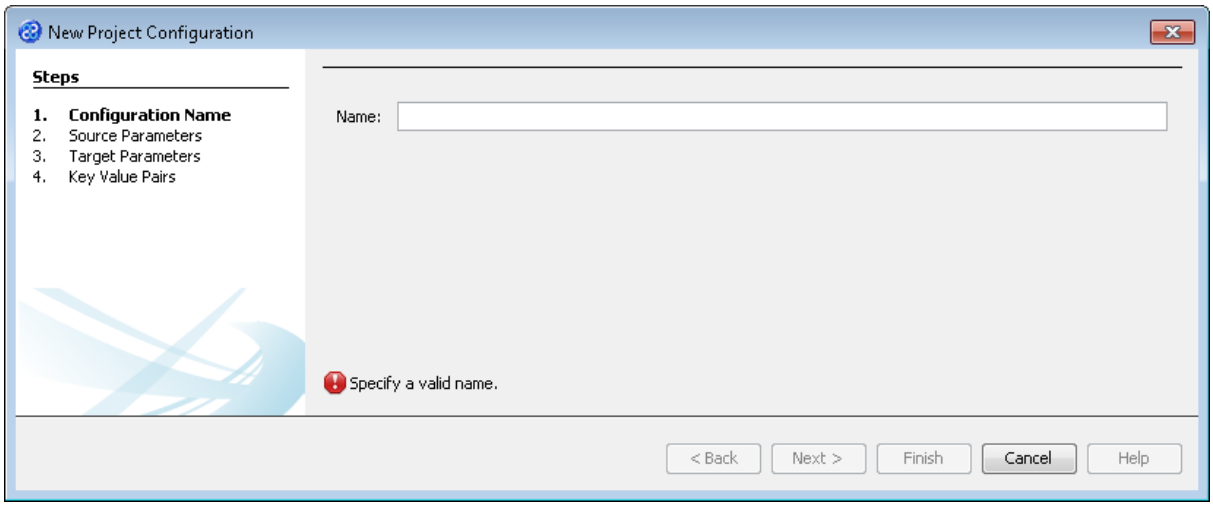
Exercise 5 - Run the Project

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.



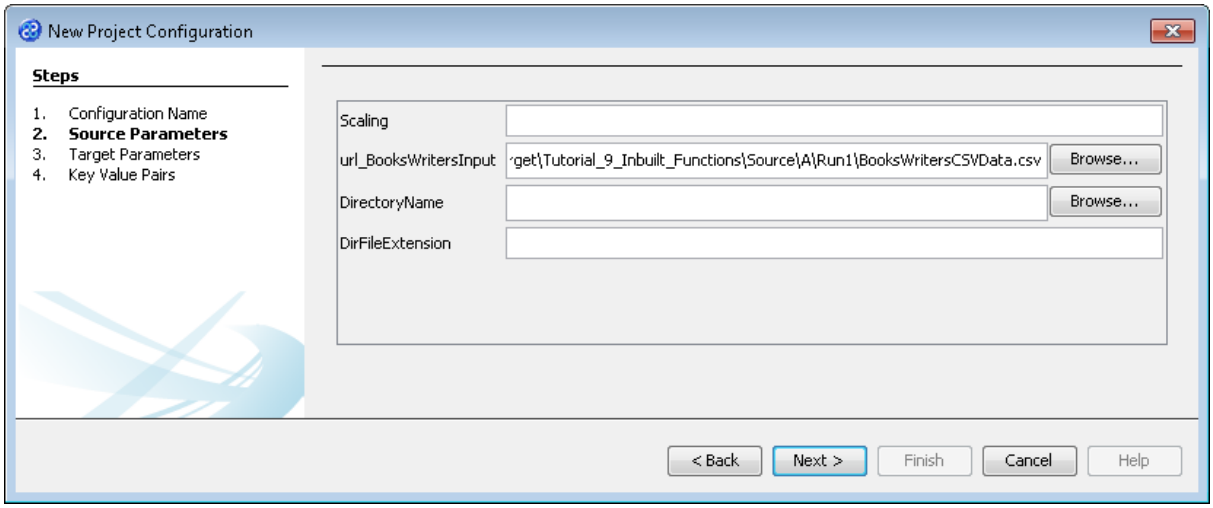
3) Lets give this configuration the name Config1. Type **Config1** into the **Name** field.

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** field of the **Source Parameters** step. In this field we will specify the location of the source data store. You can type the path and file name manually or you can use the button to select the directory and file. This will be in the following directory.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_9_Inbuilt_Functions\Source\A\Run1

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

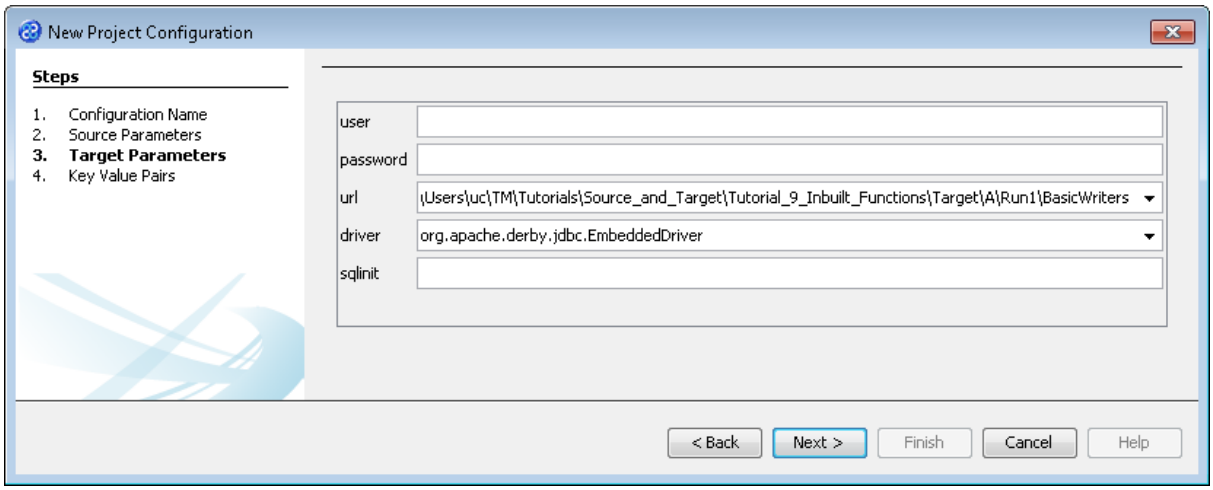


7) Click the Next > button to move to **Step 3. Target Parameters** where we will provide the details for the target data store connection.

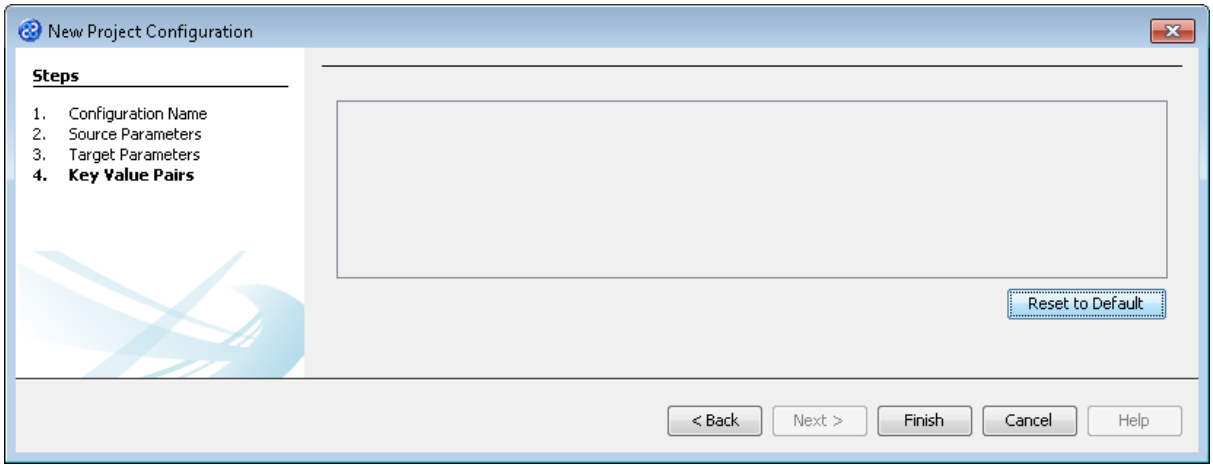
8) Go to the **url** field. You will now need to provide the details of where to go to connect to the source data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> part of the list item, including the angle brackets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

```
[TMHOME]\Tutorials\Source_and_Target\Tutorial_9_Inbuilt_Functions\Target\A\
Run1\BasicWriters
```

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.

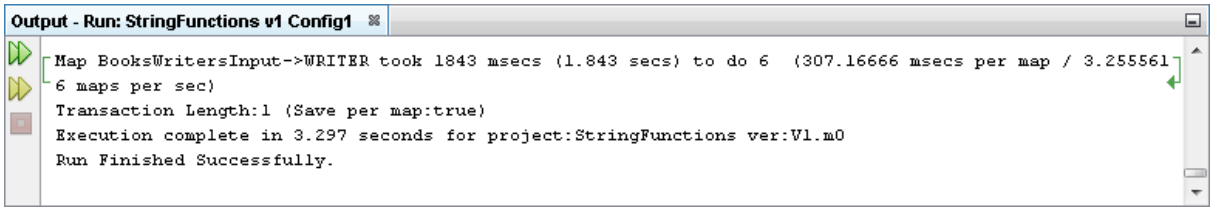


10) Click the Next > button to move to **Step 4. Key Value Pairs**. We do not need to specify any key value pairs in this exercise.



11) 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.

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



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

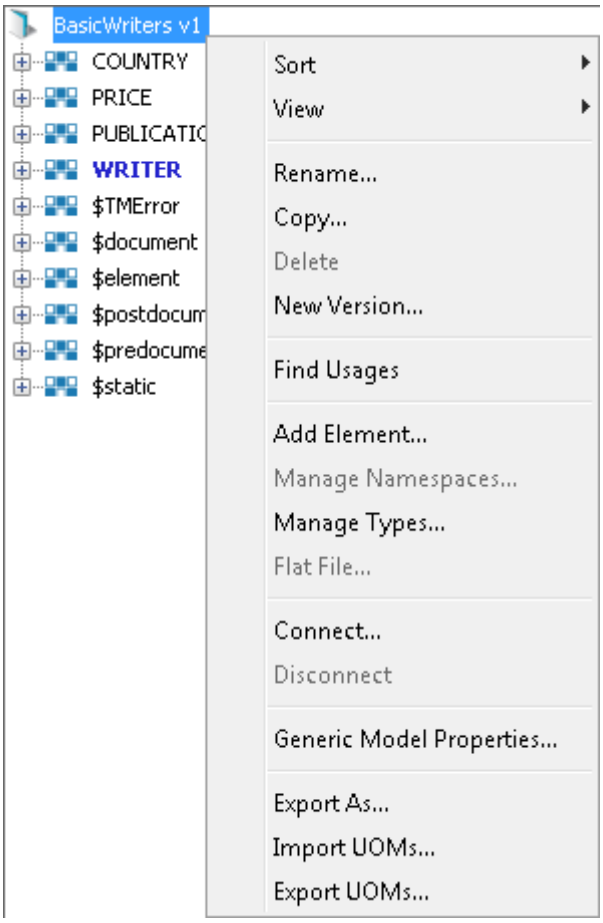
Exercise 6 - View the Target Data

Now let's check our target data store to see if our data has been transformed as expected.

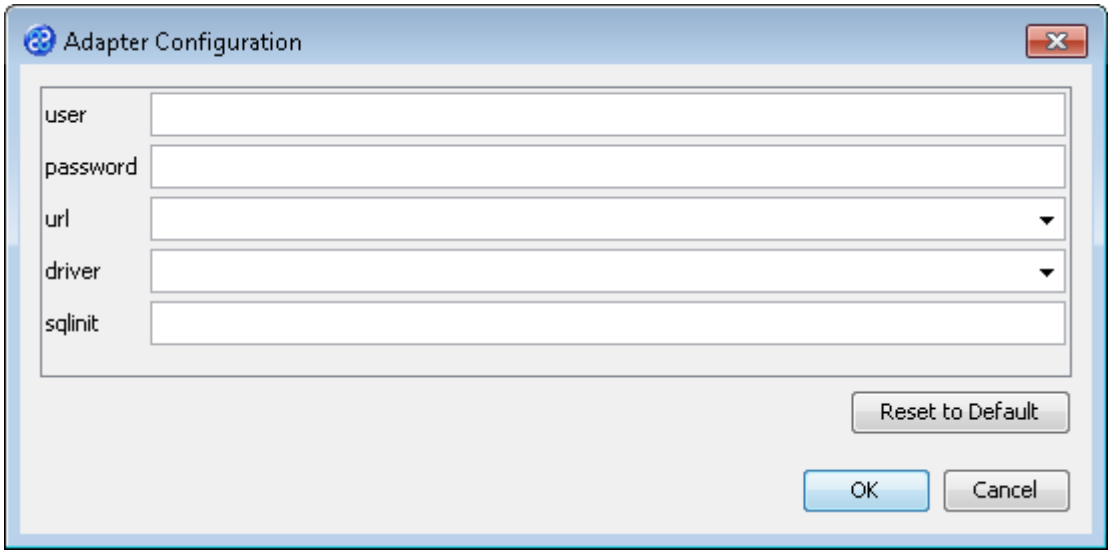
1) First let's see what the source data looked like as shown below. We can see our six authors. The transformation was written to add the text string 'test a' to the start of each surname. We also had to then pad the surname with underscores so that the surname was a minimum of eight characters long. In addition to this we added an IF statement to set the first initial based on the surname being either more than six characters long or having a surname starting with an 'H'.

```
title,price,currency,writer forename,writer surname,id,sold
Bricks,20,USD,Kevin,Hansen,111,6
Steel,30,USD,Kevin,Hansen,222,8
Sand,40,GBP,Fred,Jones,333,4
Timber,80,GBP,Julie,Falcon,444,34
Steel,30,USD,Kevin,Hansen,555,88
Bolts,10,USD,Fred,Holland,666,47
```

2) With the Editor pane still open, move the cursor over the name of the data model in the **Transform Target** pane and display the context menu for the data model.



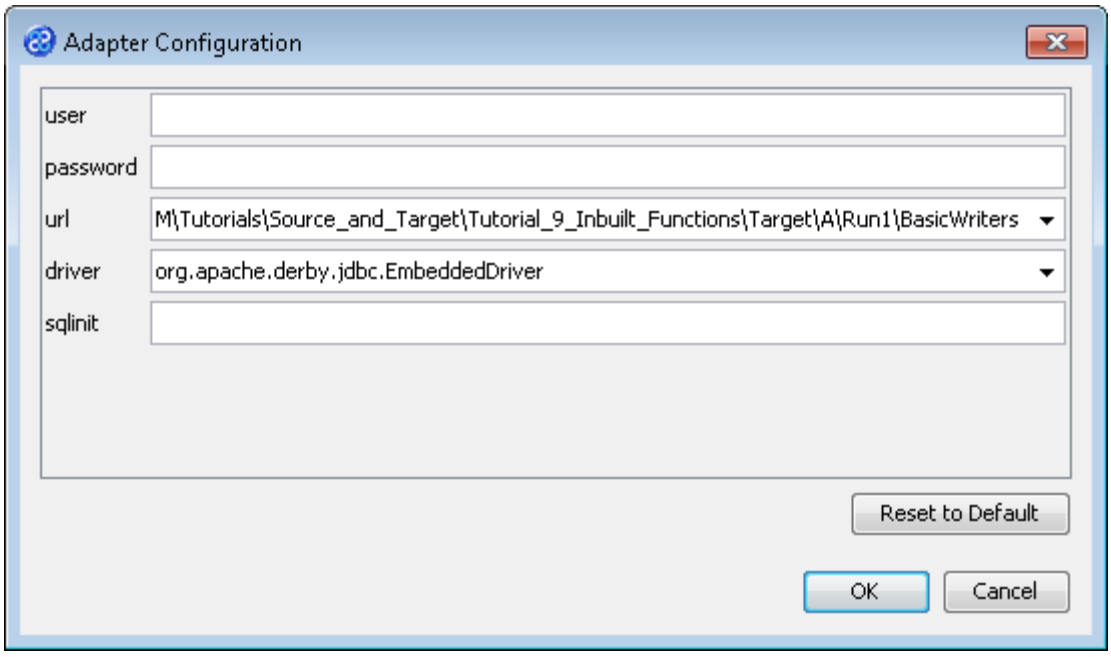
3) Select the Connect... option from the menu. The Adapter Configuration window will open ready for the connection details to be entered for the target data store.

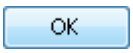


4) Let's provide the relevant connection information as shown below. Go to the **url** field. You will now need to provide the details of where to go to get the target data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> 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_9_Inbuilt_Functions\Target\A\
Run1\BasicWriters
```

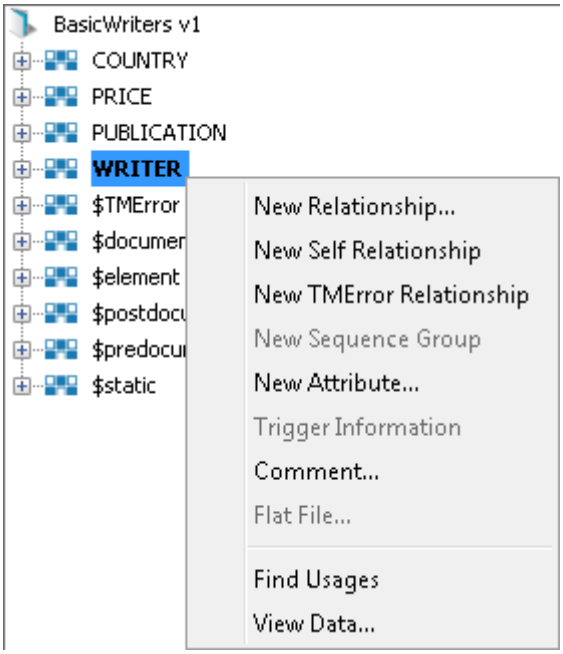
5) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The Adapter Configuration window will look similar to the image below. **user, password** and **sqlinit** are not required.



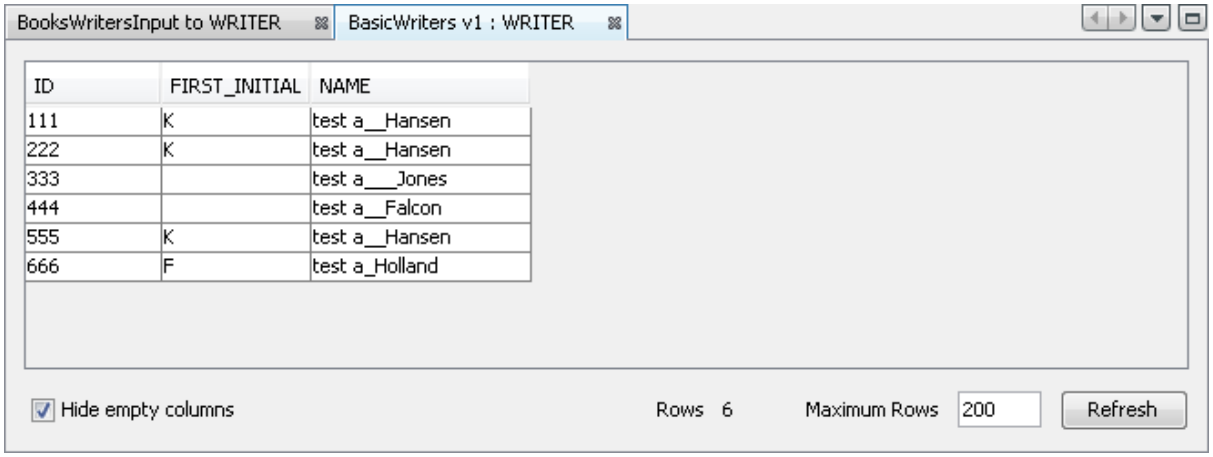
6) Click the 

button to connect to the data store. A message box will appear telling you that the connection is being made.

7) Move your cursor over the **WRITER** element in the data model and display the context menu for the element.



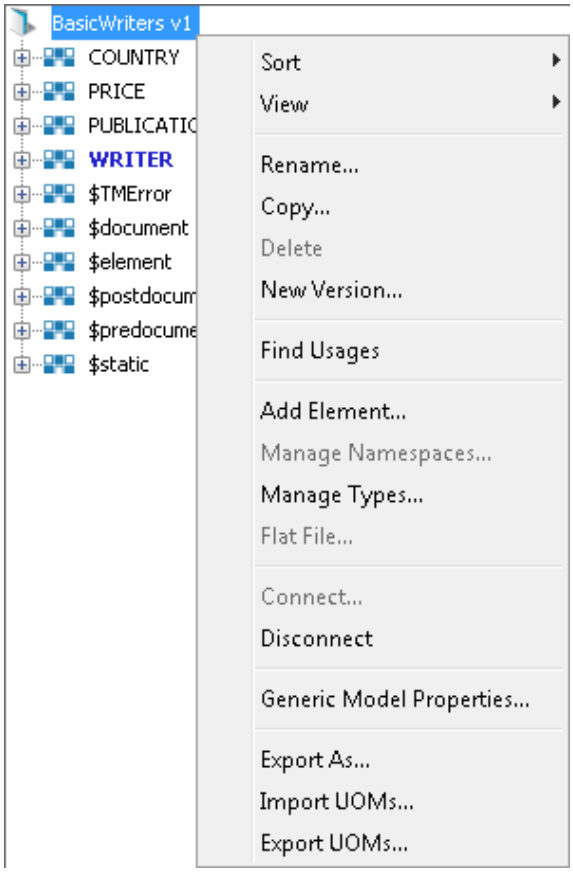
8) Select the View Data... option from the menu. This will open a new pane in the Editor pane displaying the writer data as shown below.



9) Let's look at the **WRITER** element data. The first point is that we can see our `myStr` local variable added `test_a` to the beginning of all the **NAME** values. The **NAME** values have also been padded to 8 characters using underscores as expected. Our IF code has been executed with only those writers with a surname beginning with an H or longer than six characters having their **FIRST_INITIAL** value being set. So, the results are as expected.

10) Let's close the data pane and disconnect from the data store. Click on the ⊠ icon to close the data view pane for **WRITER**.

11) 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.

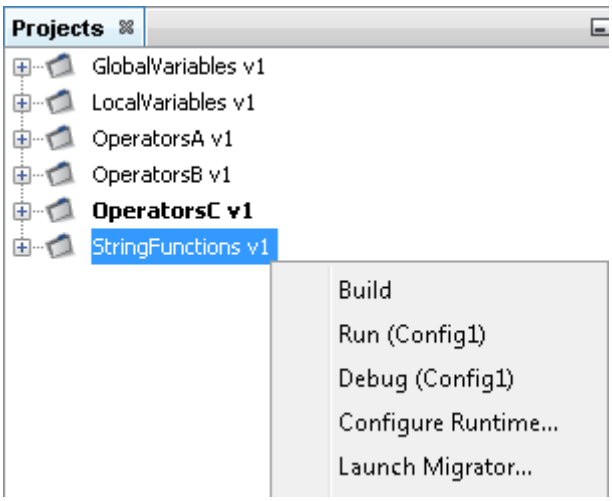


12) Note that the Disconnect option is active now. Click on this to disconnect from the data store.

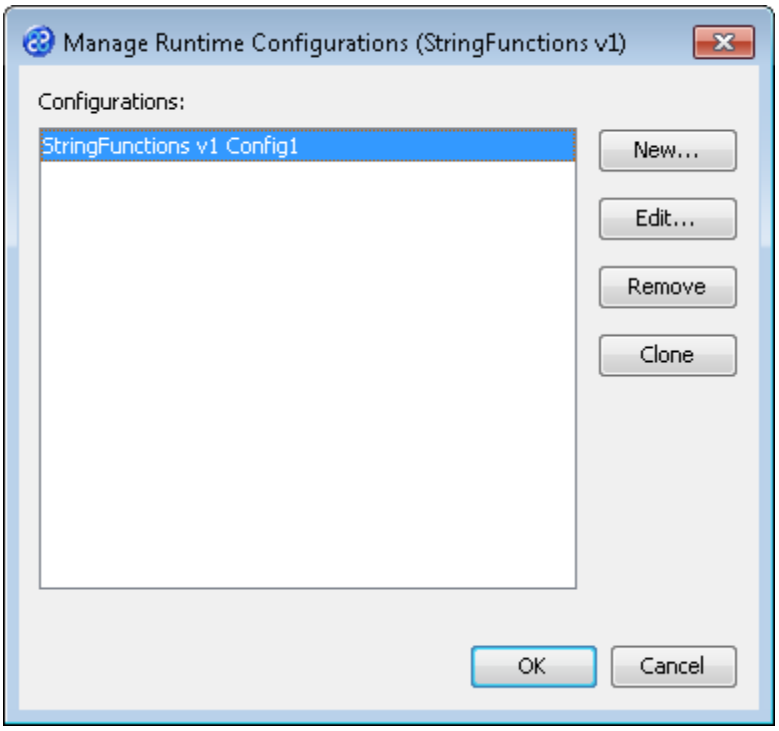
Exercise 7 - Rerun your String Function Transform with a larger data set

In this exercise we will run the same transform code but using different source and target data stores. Our source data store will contain 60000 records. We will re-run our project from the previous exercise. We will be changing the source and target data stores in order to run and see the results of our transformation project.

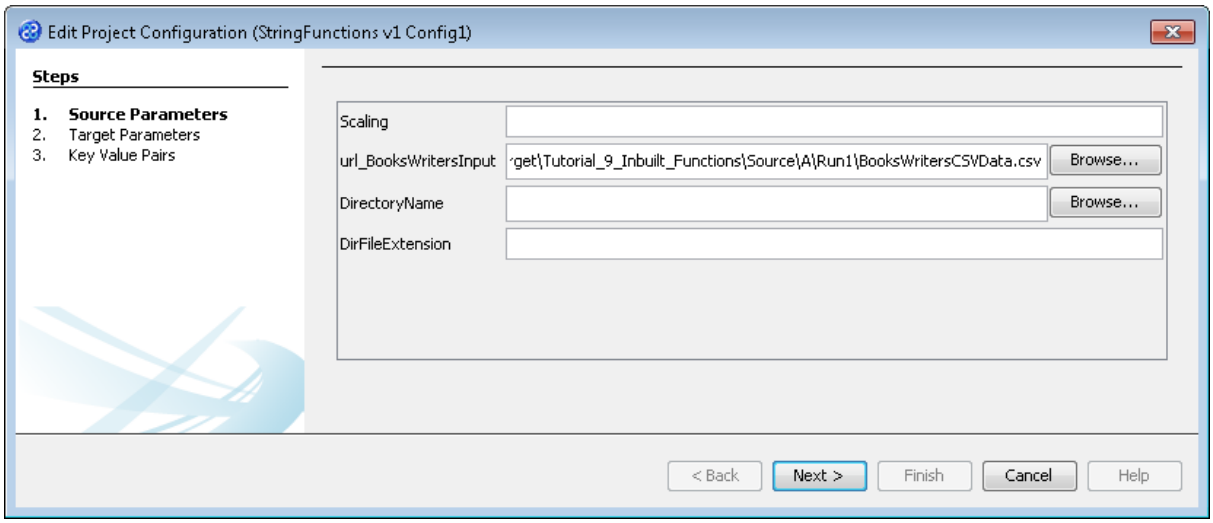
1) Move your mouse pointer so that it is over the title of the project you are working on and use the right mouse or secondary mouse button to display the context menu.



2) Click once on the Configure Runtime... option. This will open the **Manage Runtime Configurations** window as shown below.



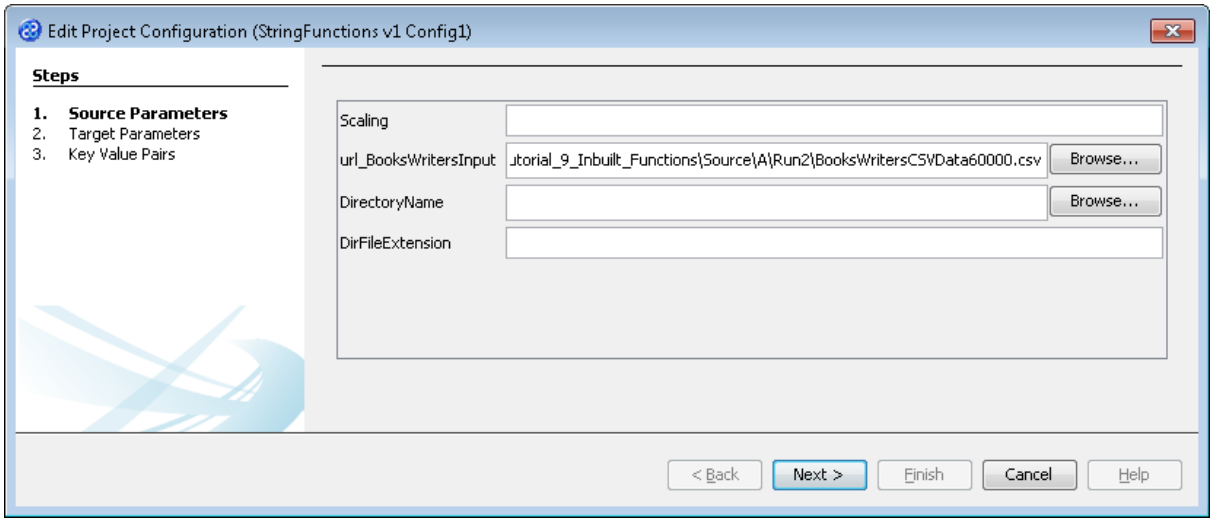
3) Click the button with the **StringFunctions v1 Config 1** item selected. This will open the **Edit Project Configuration** window on **Step 1 Source Parameters**.



4) Go to the **url_BooksWritersInput** field of the **Source Parameters** step. In this field we will update the location of the source data store. You can type the path and file name manually or you can use the button to select the directory and file. This will be in the following directory.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_9_Inbuilt_Functions\Source\A\Run2

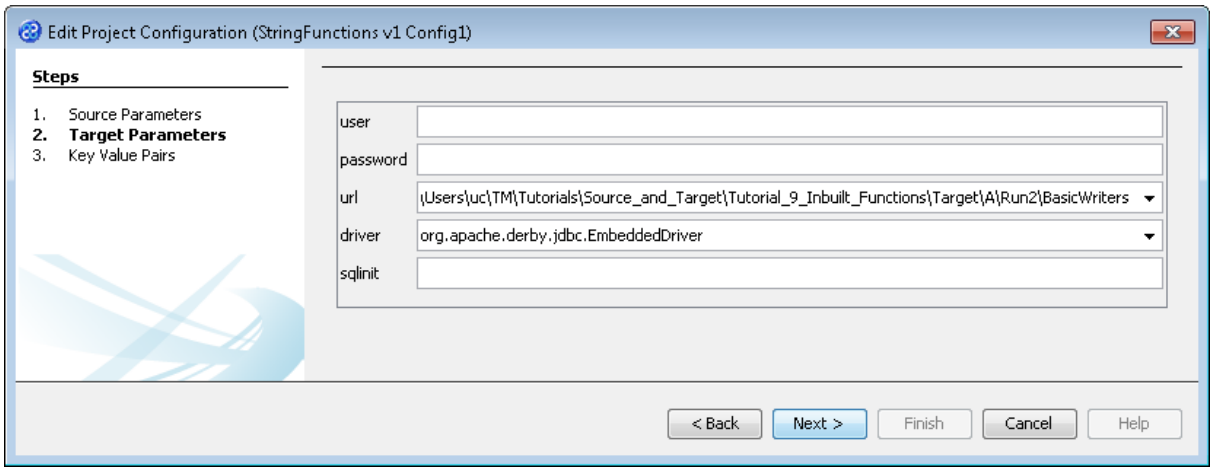
1) The **Source Parameters** step should look similar to the image below.



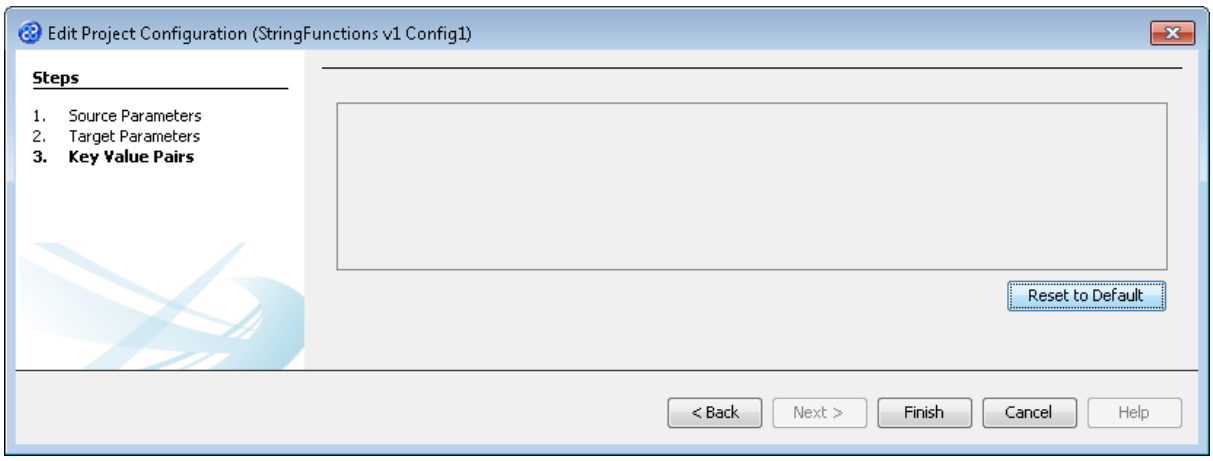
2) Click the Next > button to move to **Step 2. Target Parameters** where we will update the details for the target data store connection.

3) Go to the **url** field. You will now need to update the details of where to go to connect to the source data. Replace the `\Run1\BasicWriters` part of the item with `\Run2\BasicWriters`.

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



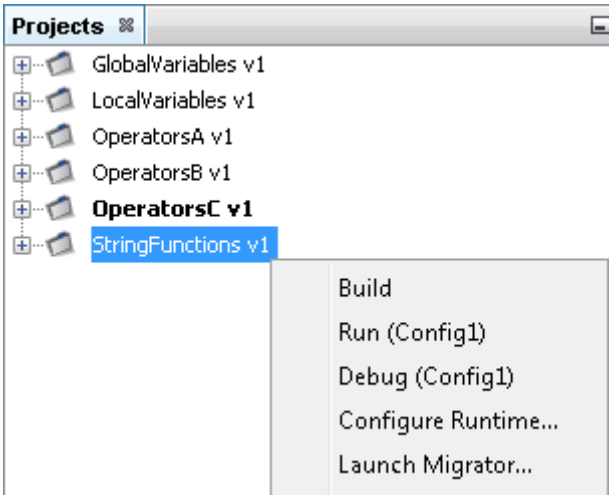
5) Click the Next > button to move to **Step 3. Key Value Pairs**. We do not need to specify any key value pairs in this exercise.



6) Click the Finish button to save the project configuration and return to the **Manage Runtime Configurations** window.

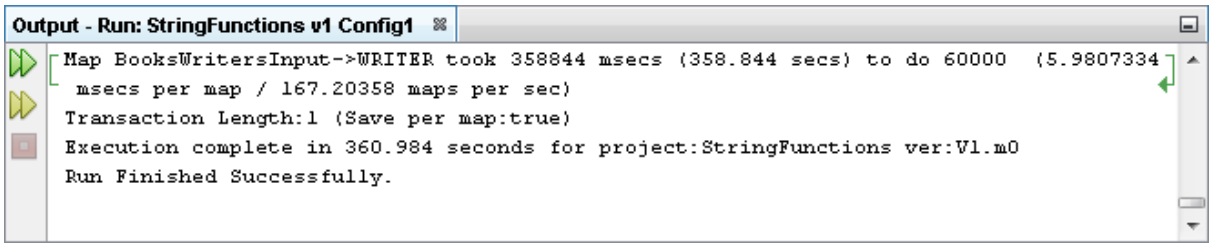
7) Click the OK button to close the **Manage Runtime Configurations** window.

8) 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.



9) Click once on the Run (Config1) option.

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



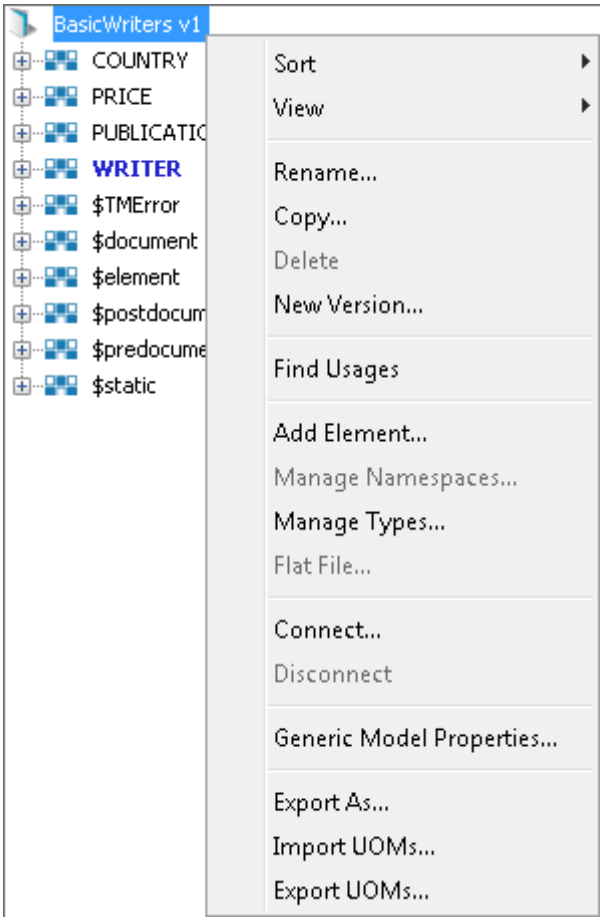
11) The information in the Output pane tells you that 60000 rows were read in the source and 60000 rows have been written to your target. The next project uses batching functions to speed this process up so it is useful to note that this transformation took 360.984 secs seconds to complete.

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

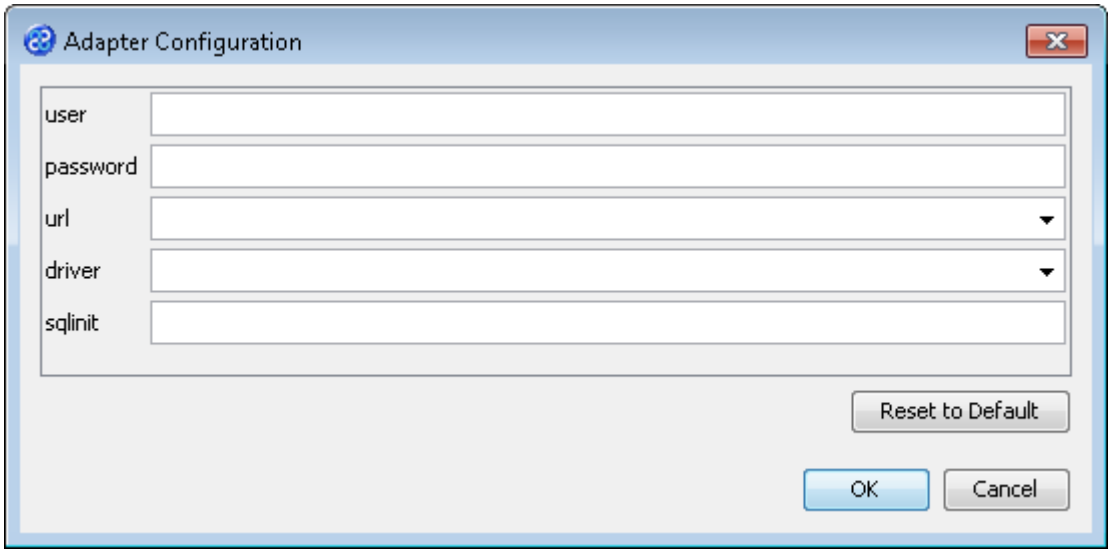
Exercise 8 - View the Target Data

Now let's check our target data store to see if our data has been transformed as expected.

1) With the Editor pane still open, move the cursor over the name of the data model in the **Transform Target** pane and display the context menu for the data model.



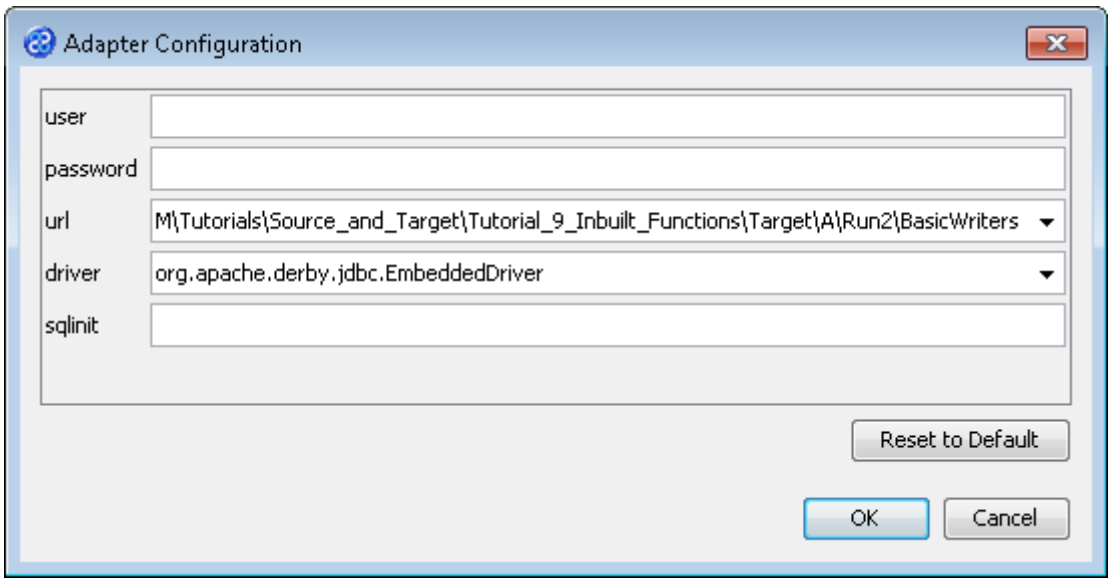
2) Select the Connect... option from the menu. The Adapter Configuration window will open ready for the connection details to be entered for the target data store.

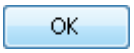


3) Let's provide the relevant connection information as shown below. Go to the **url** field. You will now need to provide the details of where to go to get the target data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> 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_9_Inbuilt_Functions\Target\A\
Run2\BasicWriters
```

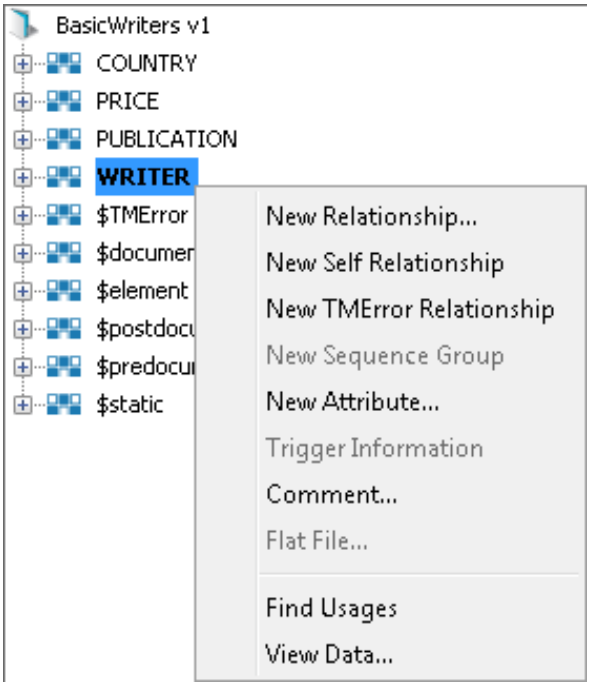
4) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The Adapter Configuration window will look similar to the image below. **user, password** and **sqlinit** are not required.



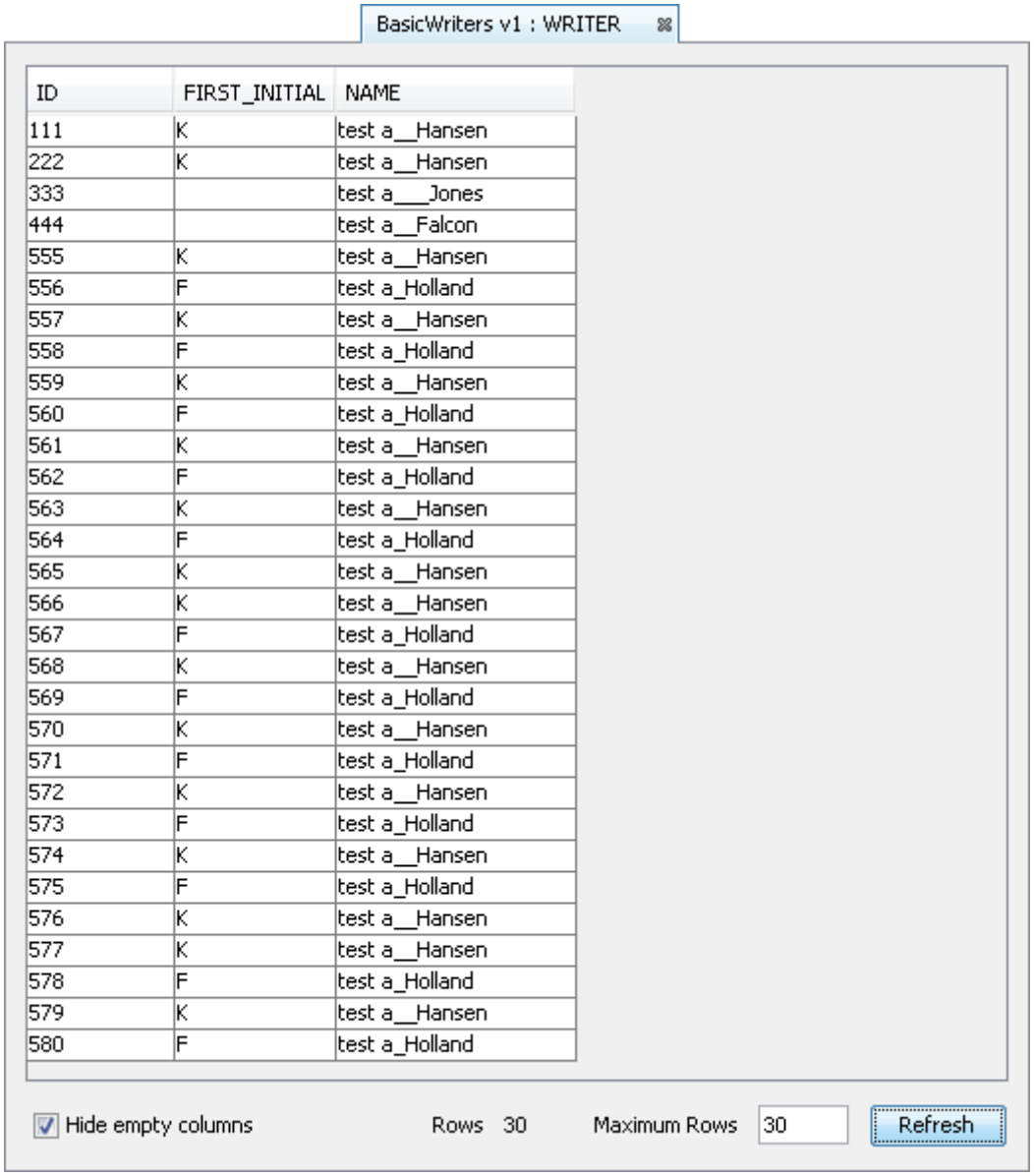
5) Click the 

button to connect to the data store. A message box will appear telling you that the connection is being made.

6) Move your cursor over the **WRITER** element in the data model and display the context menu for the element.

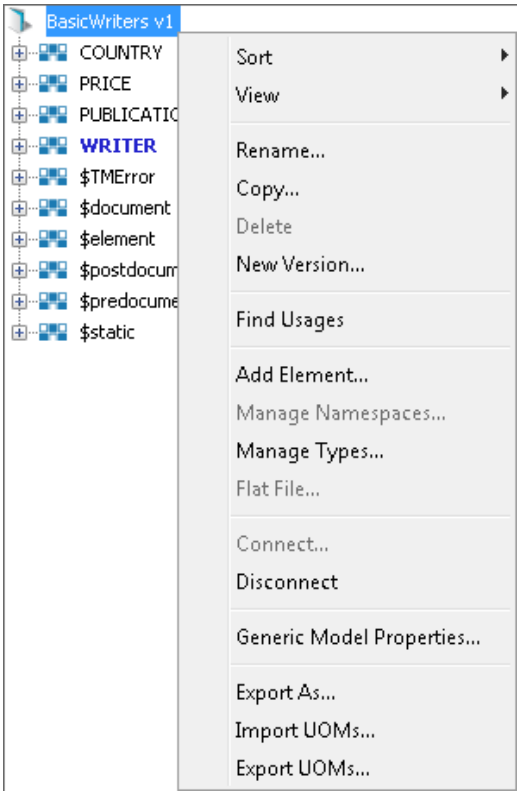


7) Select the View Data... option from the menu. This will open a new pane in the Editor pane displaying the writer data with a sample of 30 records as shown below.



8) Let's close the data pane and disconnect from the data store. Click on the ⊠ icon to close the data view pane for **WRITER**.

9) 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.

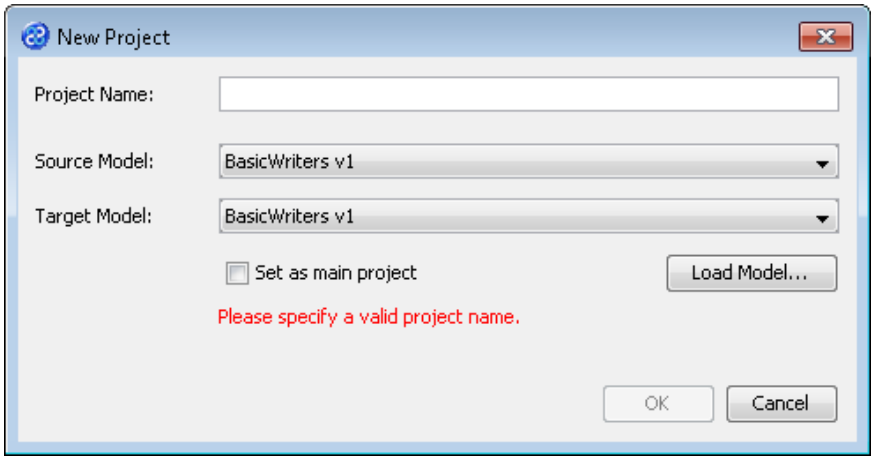


10) Note that the Disconnect option is active now. Click on this to disconnect from the data store.

Exercise 7 - Create a New Project

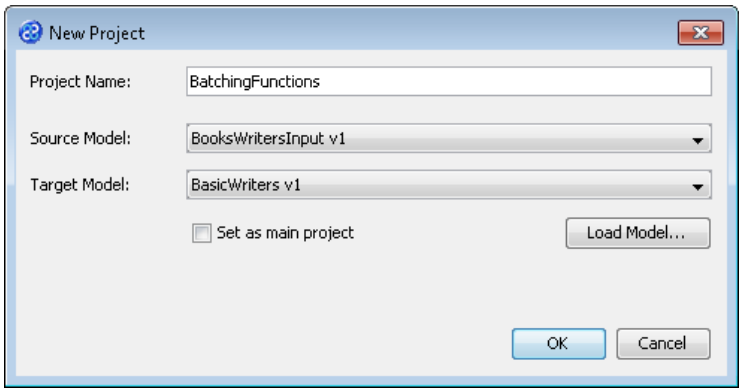
This exercise creates a new project where we will use the flat file data model called **BooksWritersInput v1** as the source data store and a relational database data store called **BasicWriters v1** as the target data model.

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 **BatchingFunctions** 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 **BooksWritersInput v1** from the list. In the **Target Model** field select **BasicWriters v1**.



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

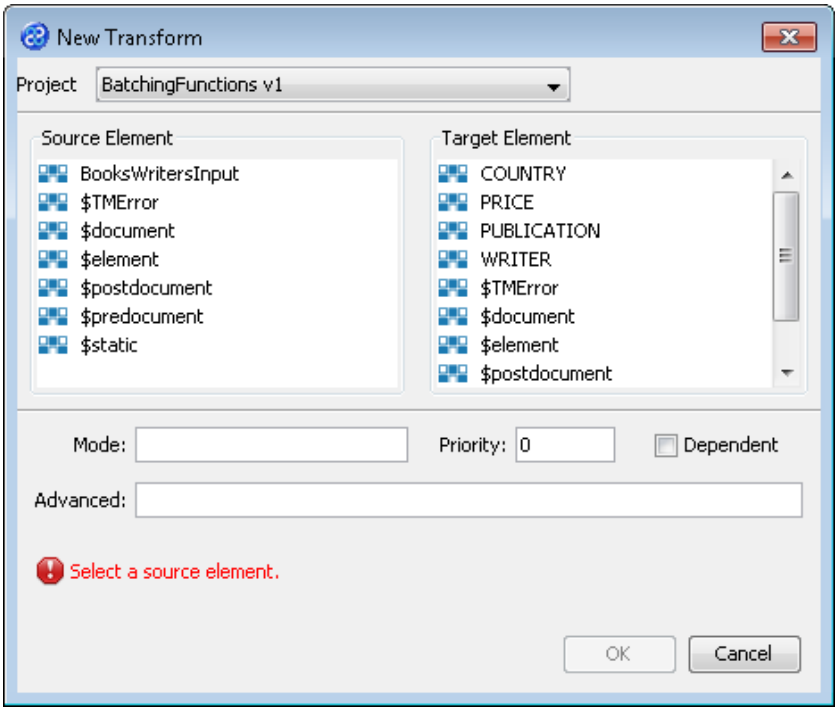
Exercise 10 - Create a Transform Using Batching Functions

This exercise will process the large data file used in exercise 5 but with the use of batching functions the file will be processed more quickly than the previous run. We will place the batching functions in the $document transform. An ordinary independent transform will be used to transform the data from the flat file data store.

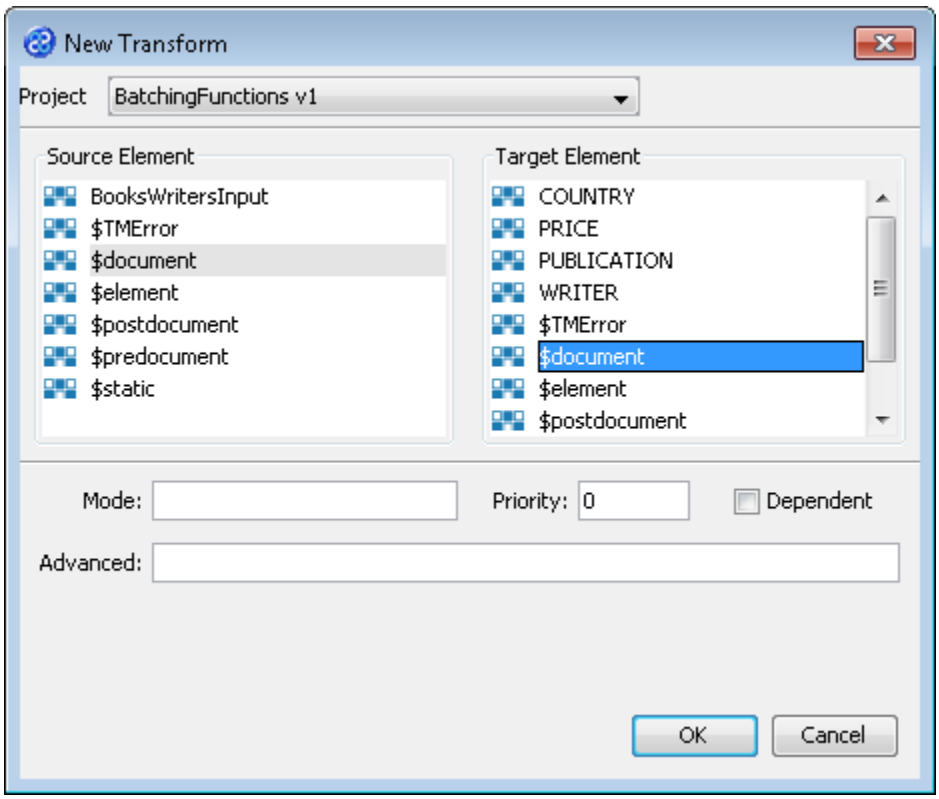
Step 1 - Create the $document Transform

1) Click once on the File menu bar option.

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



3) With the **New Transform** window open, we will select the elements for the source and target. In this case the source and target element will be the same, **$document**. The **New Transform** window will look like the one below.



4) Click once on the OK button to create your new transform.

5) The Editor pane will open ready for you to write your transform code once the **New Transform** is created.

6) Now we will write our code adding, in this case, two batching functions. The first function ensures that a batching constraint is used while the second one sets that constraint to 10,000.

```
ALLOWBATCHONCONSTRAINTS(true);
SETBATCHONCONSTRAINTSIZE(10000);
```

7) We have completed our transform.

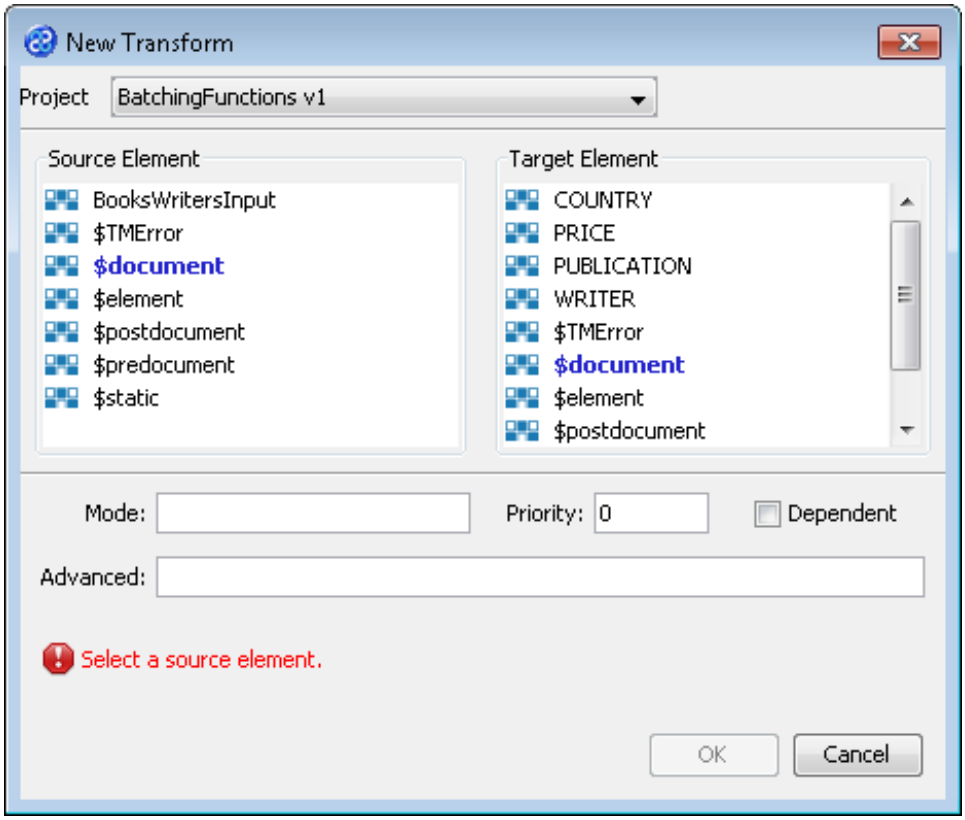
8) Click once on the File menu bar option.

9) Click once on the Save menu bar option to save your transform.

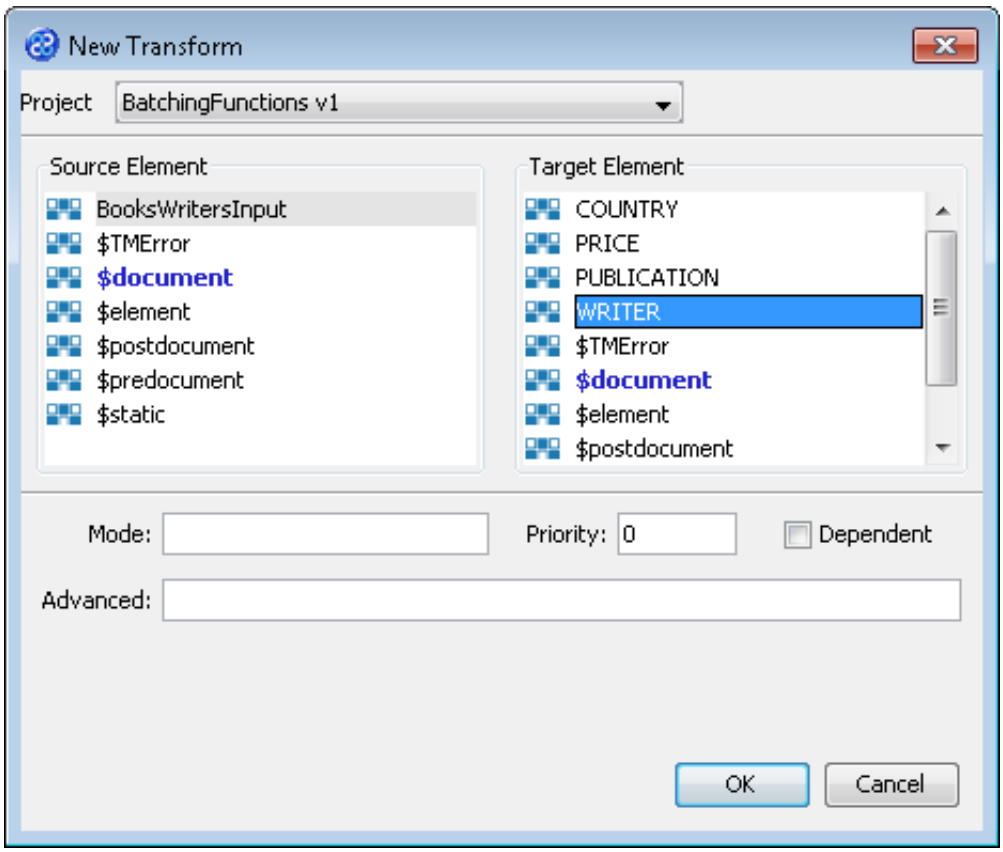
Step 2 - Create the String Function transform

1) Click once on the File menu bar option.

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



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



4) Click once on the 

button to create your new transform.

5) The Editor pane will open ready for you to write your transform code once the **New Transform** is created.

6) Now we will write our transform which will be exactly the same as we used in exercise 3.

```
LOCAL
myStr : string;
END_LOCAL;

<ID> := <id>;

IF (LENGTH(<writer surname>) > 6 OR INDEXOF(<writer surname>,'H') > 0) THEN
FIRST_INITIAL := SUBSTRING(<writer forename>,1,2);
END_IF;

myStr := '  test a   ';

PRINTLN('Testing how TRIM works on the fly');
PRINTLN('Here is the test string_' & myStr & '_more text to see the
spaces');
PRINTLN('Here is the test string_' & TRIM(myStr) & '_with no spaces before
or after the text');

NAME := TRIM(myStr) & PADSTRING(<writer surname>,8,true,'_');
```

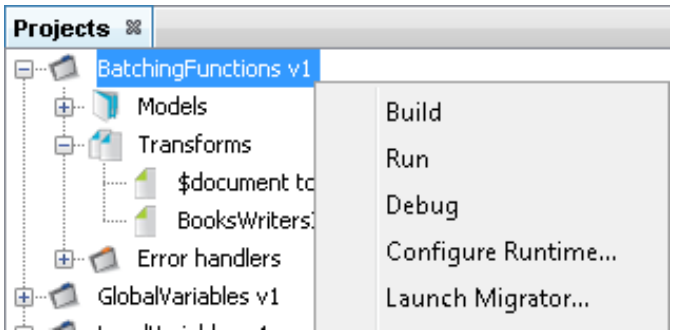
7) We have completed our transform.

8) Click once on the File menu bar option.

9) Click once on the Save menu bar option to save your transform.

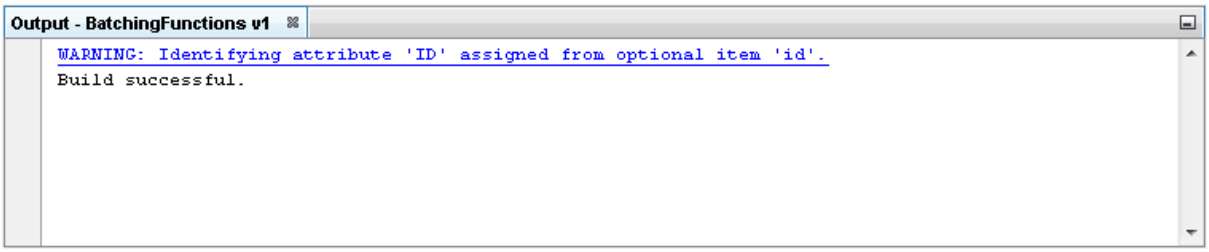
Exercise 11 - Build Your BatchingFunctions Project

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.

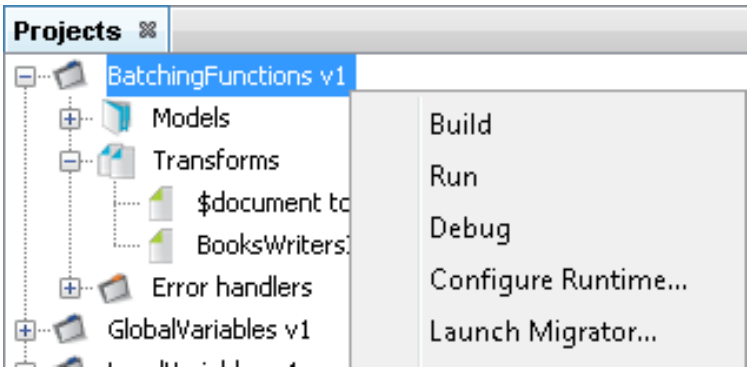
3) The **Output** pane will open and show you if there are any errors in your transform code. In this exercise you should not have any error messages and you should see a message stating **Build successful.**.



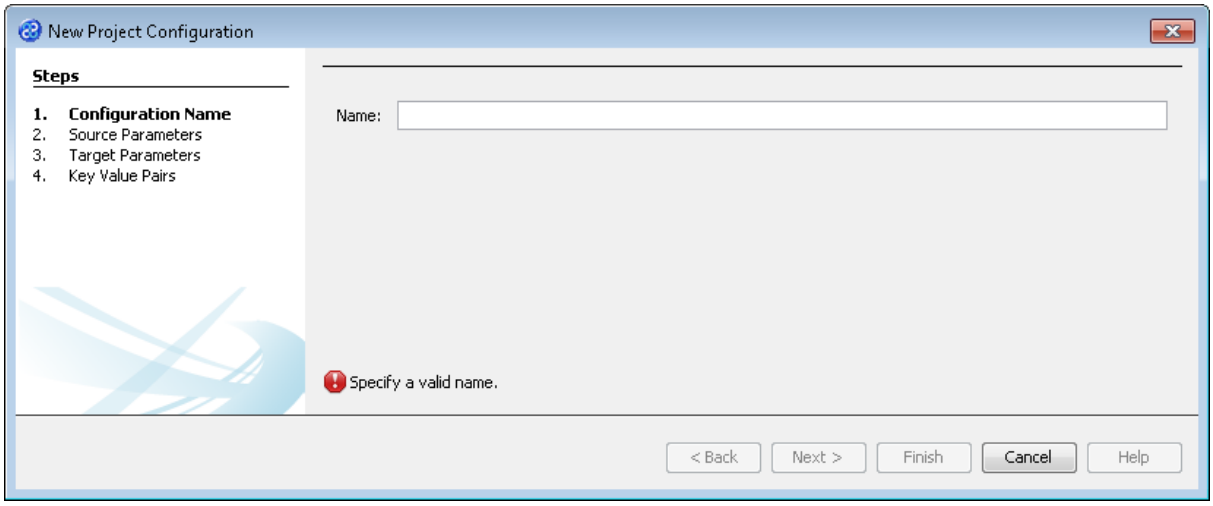
Exercise 12 - Run the Project

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.



3) Lets give this configuration the name Config1. Type **Config1** into the **Name** field.

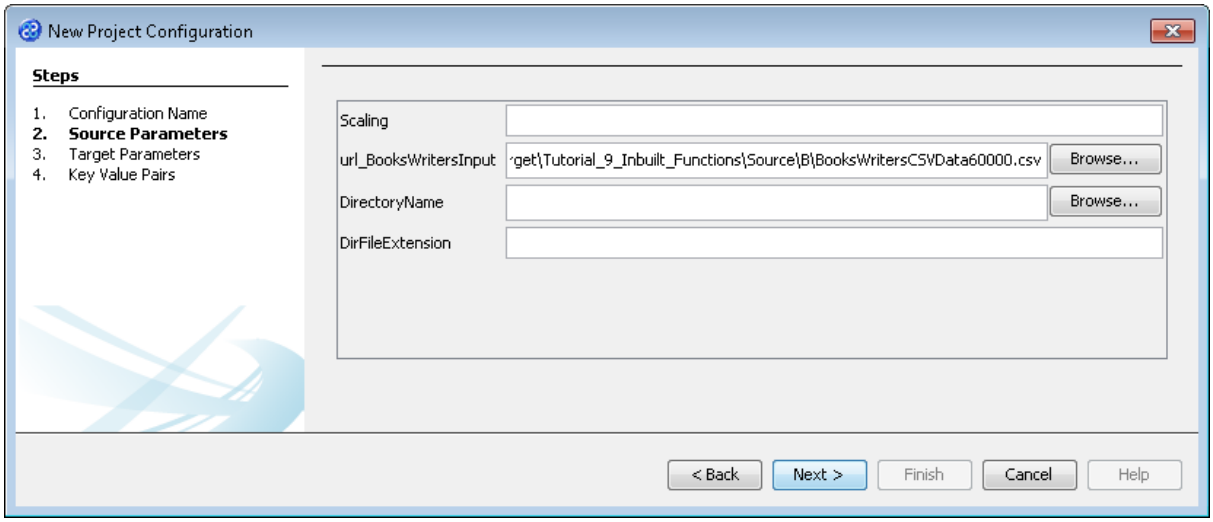
4) Click the 

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_BooksWritersInput** field of the **Source Parameters** step. In this field we will specify the location of the source data store. You can type the path and file name manually or you can use the button to select the directory and file. This will be in the following directory.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_9_Inbuilt_Functions\Source\B

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

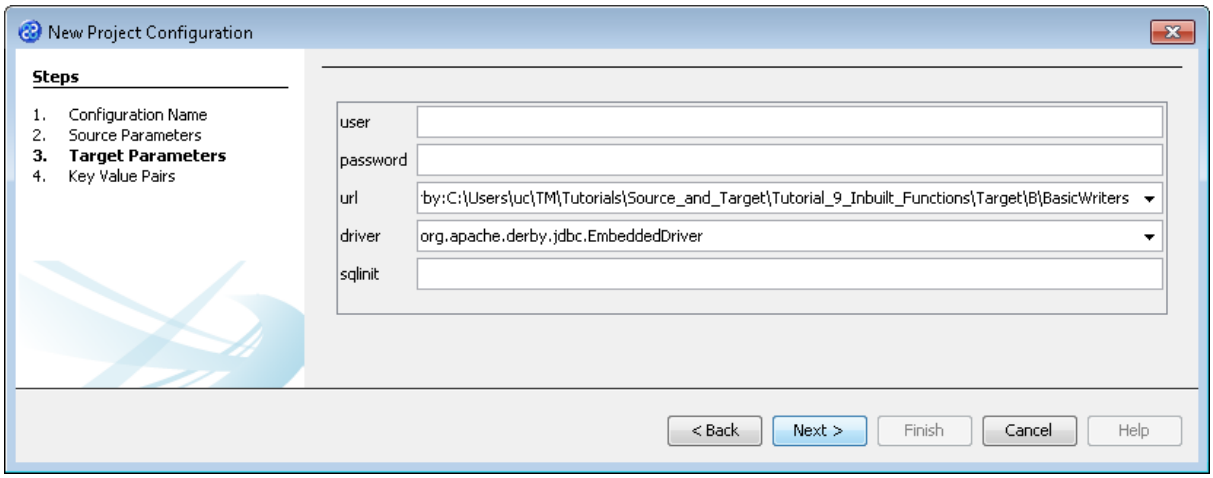


7) Click the Next > button to move to **Step 3. Target Parameters** where we will provide the details for the target data store connection.

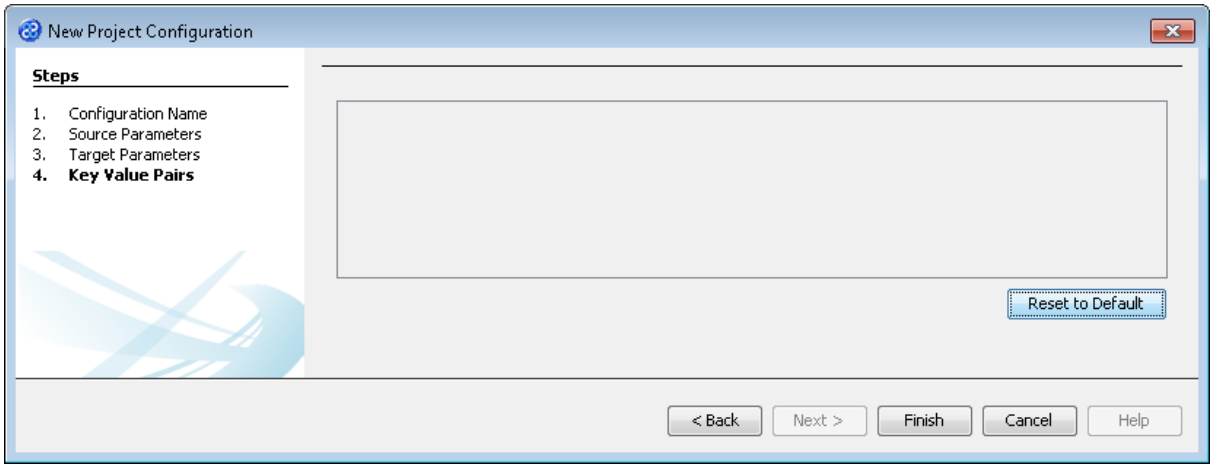
8) Go to the **url** field. You will now need to provide the details of where to go to connect to the source data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> part of the list item, including the angle brackets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

```
[TMHOME]\Tutorials\Source_and_Target\Tutorial_9_Inbuilt_Functions\Target\B\
BasicWriters
```

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.

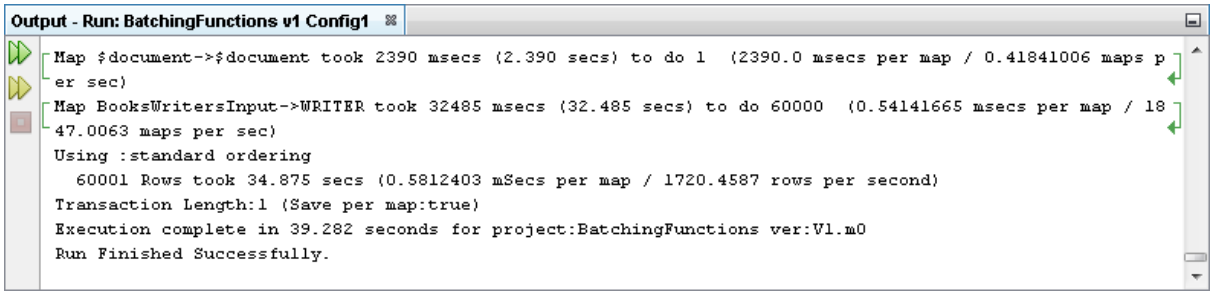


10) Click the Next > button to move to **Step 4. Key Value Pairs**. We do not need to specify any key value pairs in this exercise.



11) 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.

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



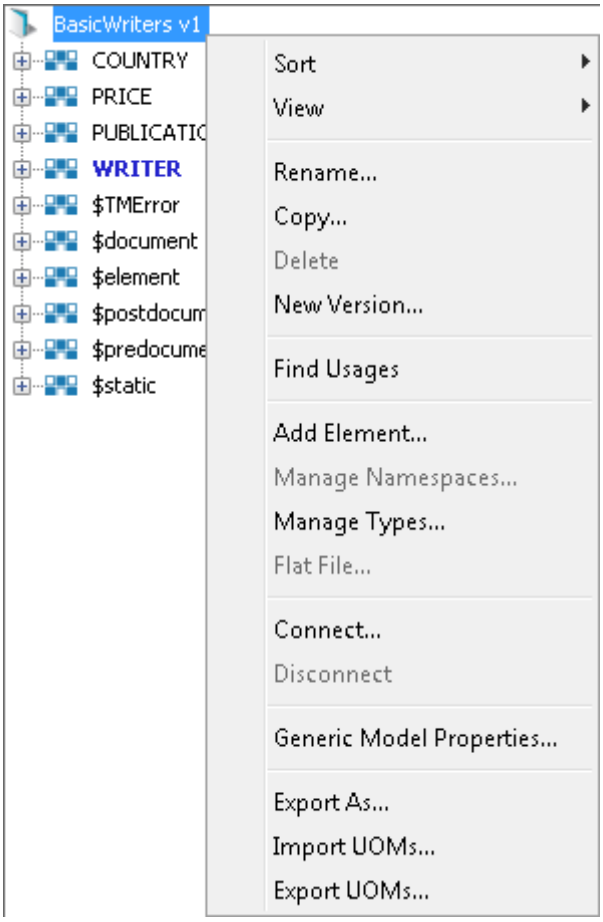
13) The information in the Output pane tells you that 60000 rows were read in the source and 60000 rows have been written to your target. Now we can see the effect of the batching functions used in the $document transform. These have speeded up the processing reducing the time from 360.984 seconds to 39.282 seconds, improving the processing time to just 10% of the original. This may of course vary depending on your hardware, memory, processor speed etc.

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

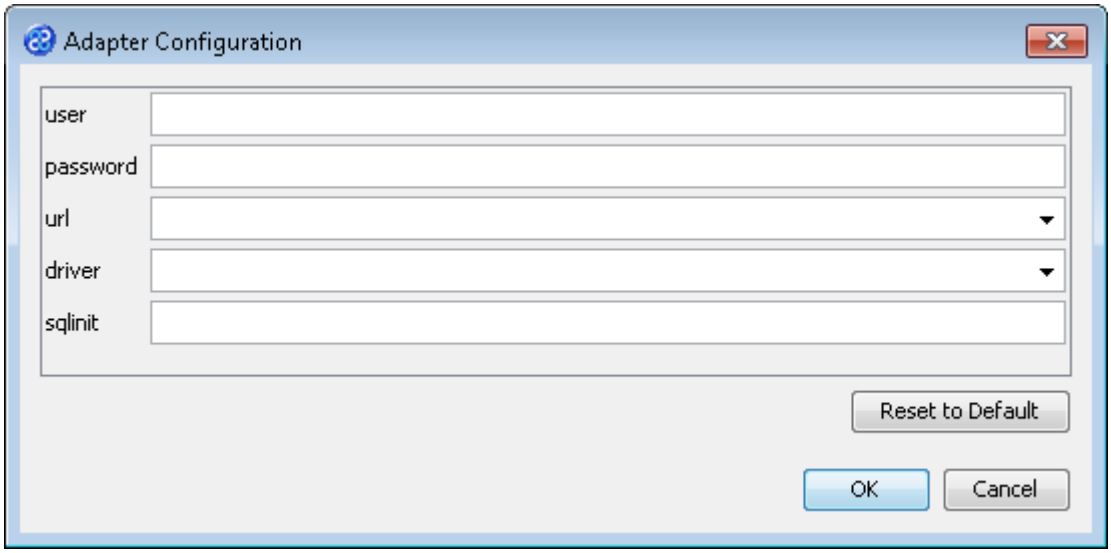
Exercise 13 - View the Target Data

Now let's check our target data store to see if our data has been transformed as expected.

1) With the Editor pane still open, move the cursor over the name of the data model in the **Transform Target** pane and display the context menu for the data model.



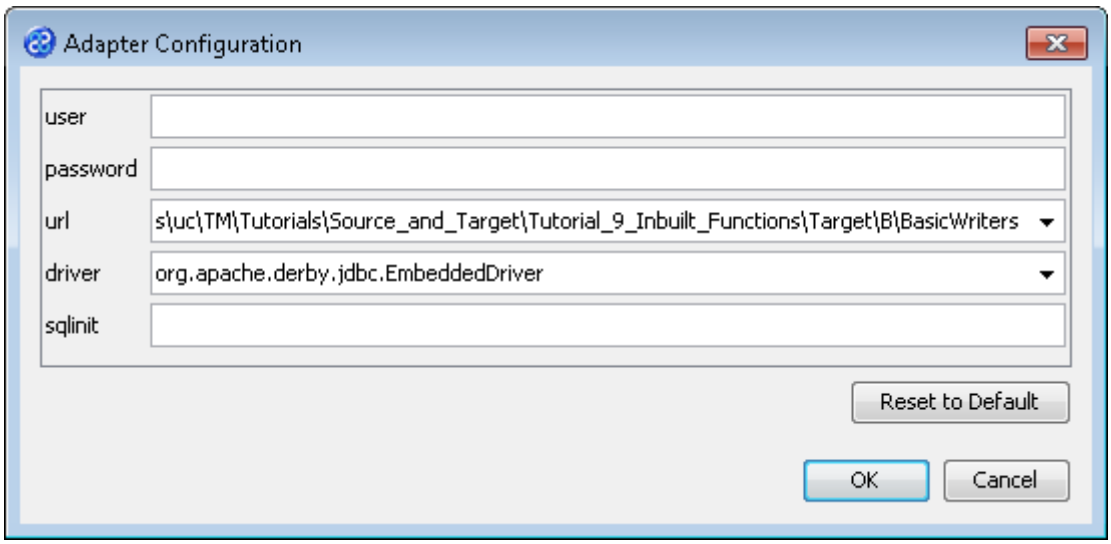
2) Select the Connect... option from the menu. The Adapter Configuration window will open ready for the connection details to be entered for the target data store.

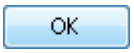


3) Let's provide the relevant connection information as shown below. Go to the **url** field. You will now need to provide the details of where to go to get the target data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> 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_9_Inbuilt_Functions\Target\B\
BasicWriters
```

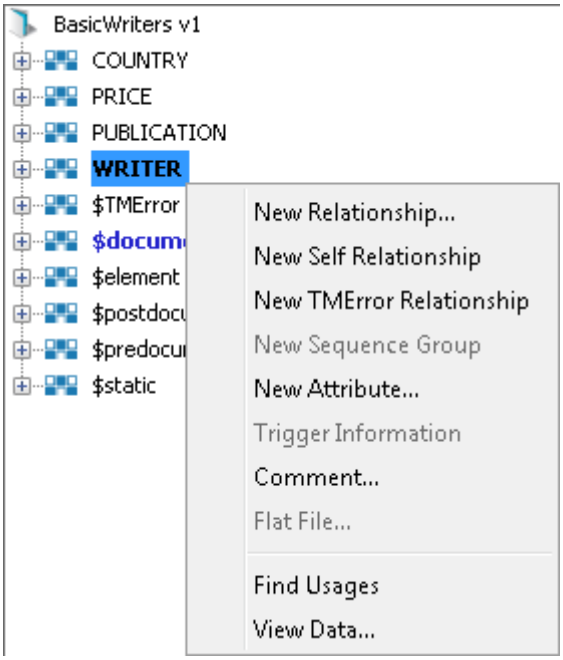
4) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The Adapter Configuration window will look similar to the image below. **user, password** and **sqlinit** are not required.



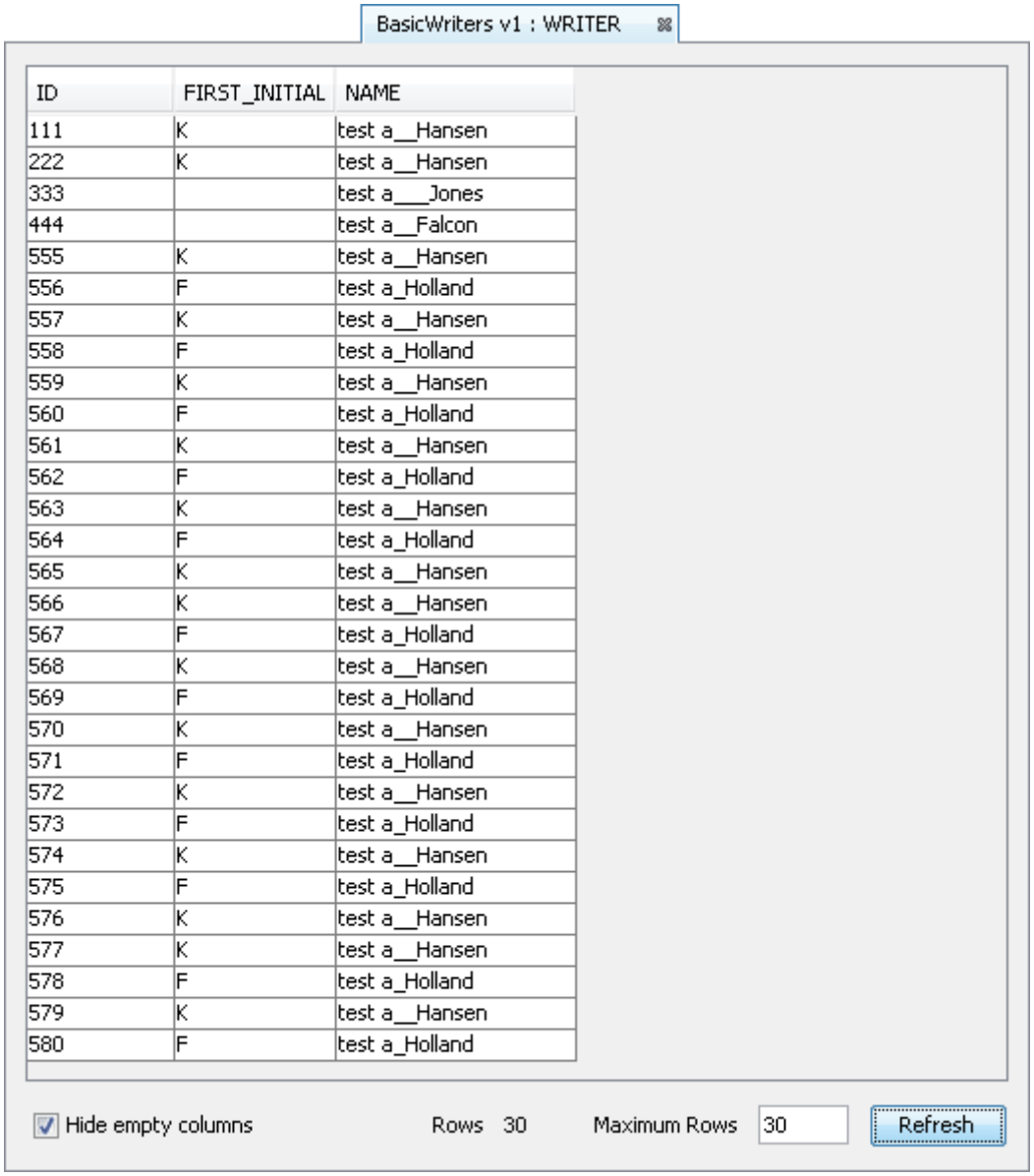
5) Click the 

button to connect to the data store. A message box will appear telling you that the connection is being made.

6) Move your cursor over the **WRITER** element in the data model and display the context menu for the element.

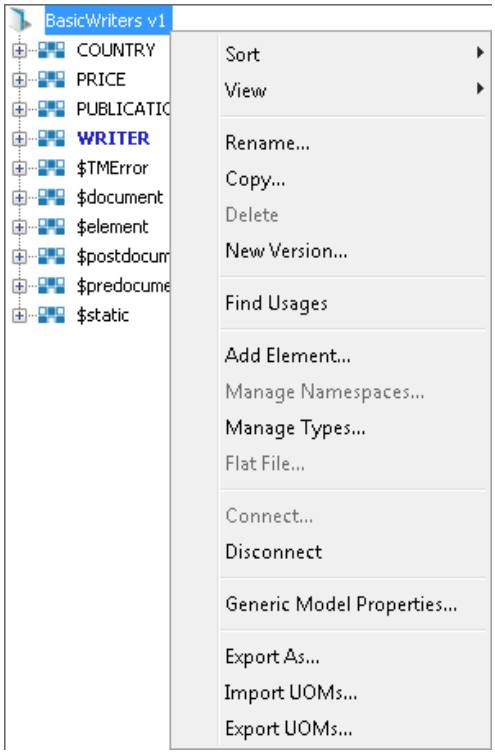


7) Select the View Data... option from the menu. This will open a new pane in the Editor pane displaying the writer data with a sample of 30 records as shown below. These are exactly the same as in the first project but processed much more quickly using the batching functions in the $document transform.



8) Let's close the data pane and disconnect from the data store. Click on the ⊠ icon to close the data view pane for **WRITER**.

9) 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.

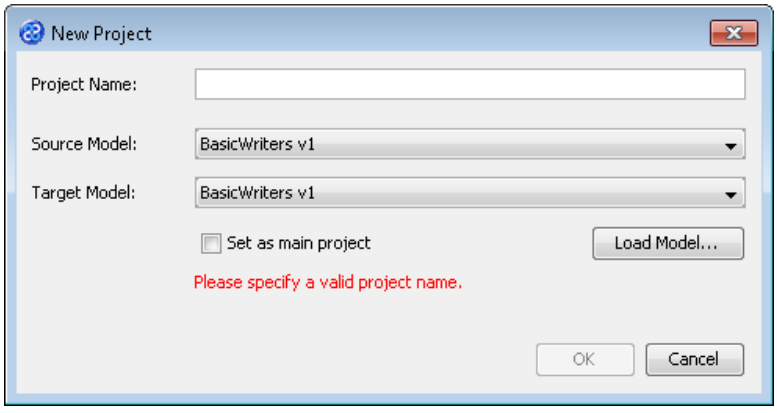


10) Note that the Disconnect option is active now. Click on this to disconnect from the data store.

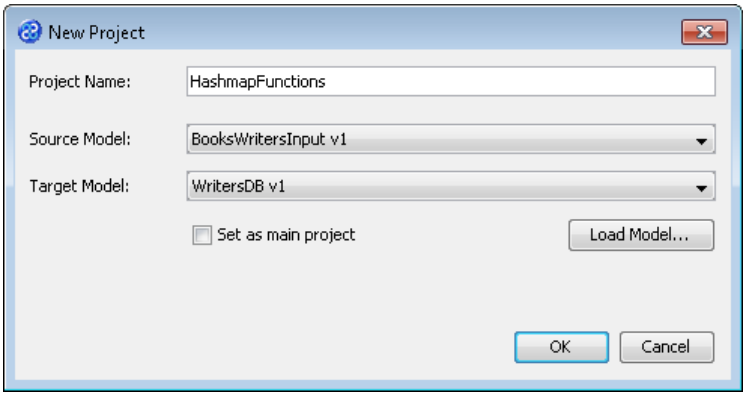
Exercise 14 - Create a New Project

This exercise creates a new project where we will use the flat file data model called **BooksWritersInput v1** as the source data store and a relational database data store called **Writers v1** as the target data model. This is a different target to the one used in the previous two projects.

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 **HashmapFunctions** 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 **BooksWritersInput v1** from the list. In the **Target Model** field select **WritersDB v1**.



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

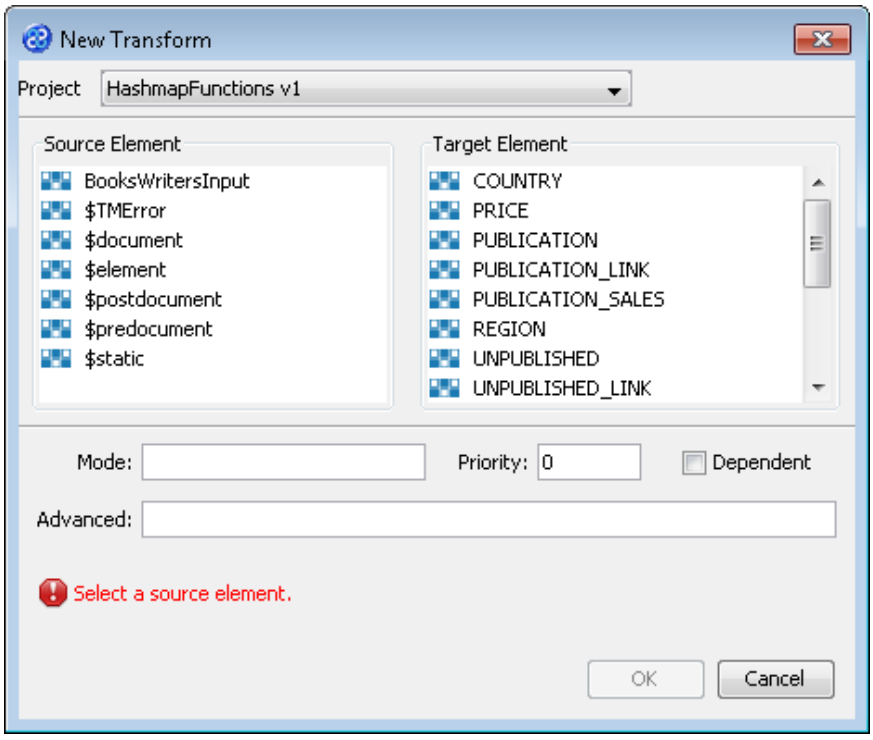
Exercise 15 - Create a Transform Using Java HashMap Functions

This exercise plus exercises 16 and 17 are all steps in the same project and will create three transforms. The first will be a $document transform holding a global variable that tells Transformation Manager that a Java HashMap object will be constructed. The second independent transform uses the source data to populate the HashMap with key and value pairs. The third transform is dependent on the independent transform created in exercise 16 and adds additional data to the HashMap concerning a 20% additional charge to the book price.

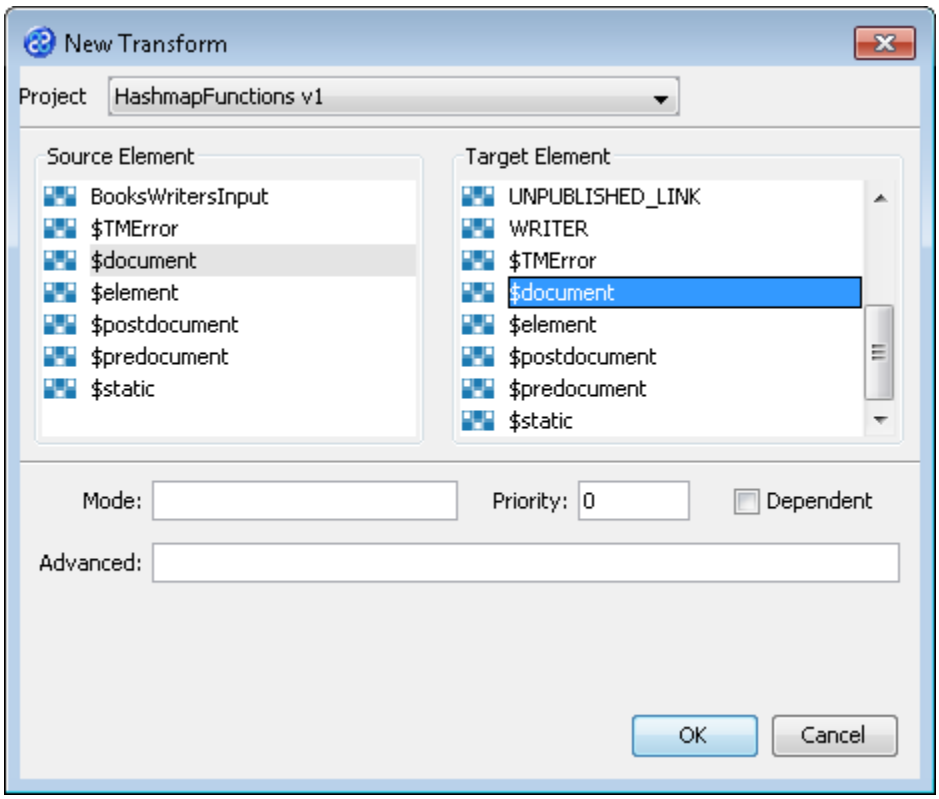
Step 1 - Create a $document Transform for the HashMap Exercise

1) Click once on the File menu bar option.

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



3) With the **New Transform** window open, we will select the elements for the source and target. In this case the source and target element will be the same, **$document**. The **New Transform** window will look like the one below.



4) Click once on the  button to create your new transform.

5) The Editor pane will open ready for you to write your transform code once the **New Transform** is created.

6) Now we will write our code for the global variable that we require.

```
GLOBAL
gblHashmap : jobject;
END_GLOBAL;

gblHashmap:= MAKEJOBJECT('java.util.HashMap');
```

7) We have completed our transform.

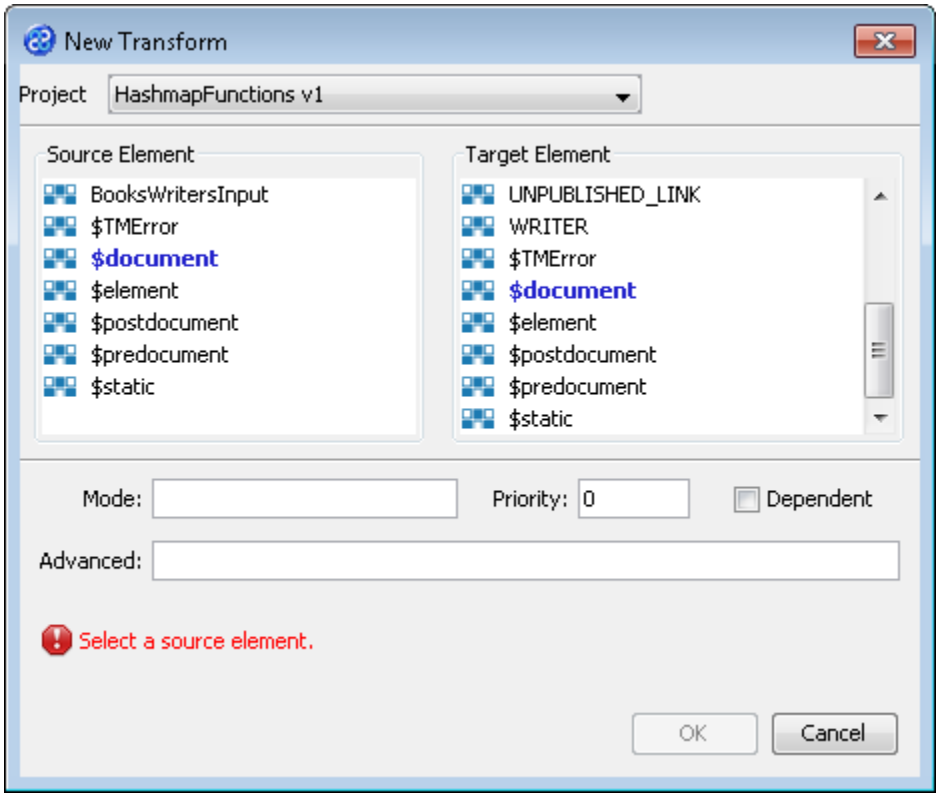
8) Click once on the File menu bar option.

9) Click once on the Save menu bar option to save your transform.

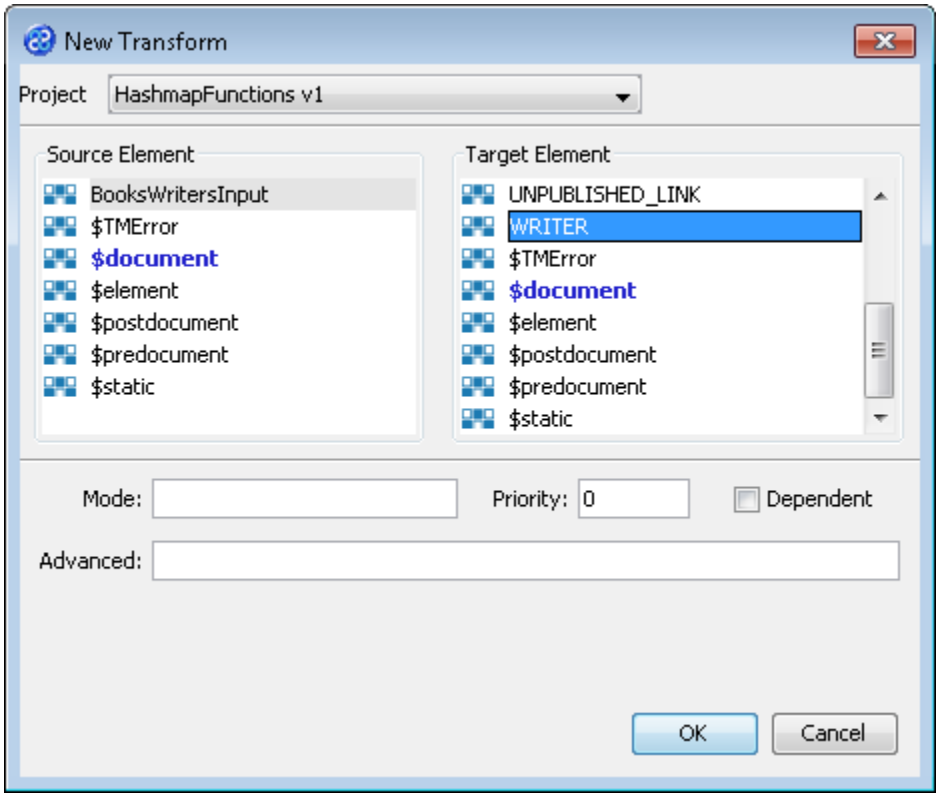
Exercise 16 - Step 2 - Create the Initial HashMap transform

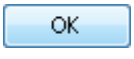
1) Click once on the File menu bar option.

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



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



4) Click once on the 

button to create your new transform.

5) The Editor pane will open ready for you to write your transform code once the **New Transform** is created.

6) Now we will write our transform. The first part of our transform provides an output while the project executes. This will tell us which ID is being processed. After this we have our assignment code providing an ID value with the SURROGATEKEY function and a forename and name value for the target data store. This is followed by the code for our cascade using the self relationship. Lastly, we have the code which checks and retrieves the values from the hashmap if present. This could be carried out within any map in the project as the hashmap is set as a global variable but for this exercise we will check it here.

```
PRINTLN('Processing ID-'&<id>);

<ID> := SURROGATEKEY(<id>+1000);
FIRST_INITIAL := SUBSTRING(<writer forename>,1,2);
*NAME := <writer surname>;

iPUBLICATION    := self;

IF (JMAPCONTAINSKEY(gblHashmap,<id>)) THEN
PRINTLN('Our hashmap contains a record for this key-'&<id>);
TAXABLE_INCOME := JMAPGETBYKEY(gblHashmap,<id>);
ELSE
PRINTLN('Our hashmap does not contain a record for this key at present-
'&<id>);
END_IF;
```

7) We have now completed our transform. You may see an error in your transform,  . This will be because there is no `self` relationship for the **BooksWritersInput** element of the **Transform Source** element. You will need to add this.

   a) Move the mouse pointer over the **BooksWritersInput** element in the **Transform Source** pane and display the context menu with a right mouse click or secondary mouse click.

   b) Click on the New Self Relationship option in the menu. This will add the `self` relationship and the error will be resolved.

8) Click once on the File menu bar option.

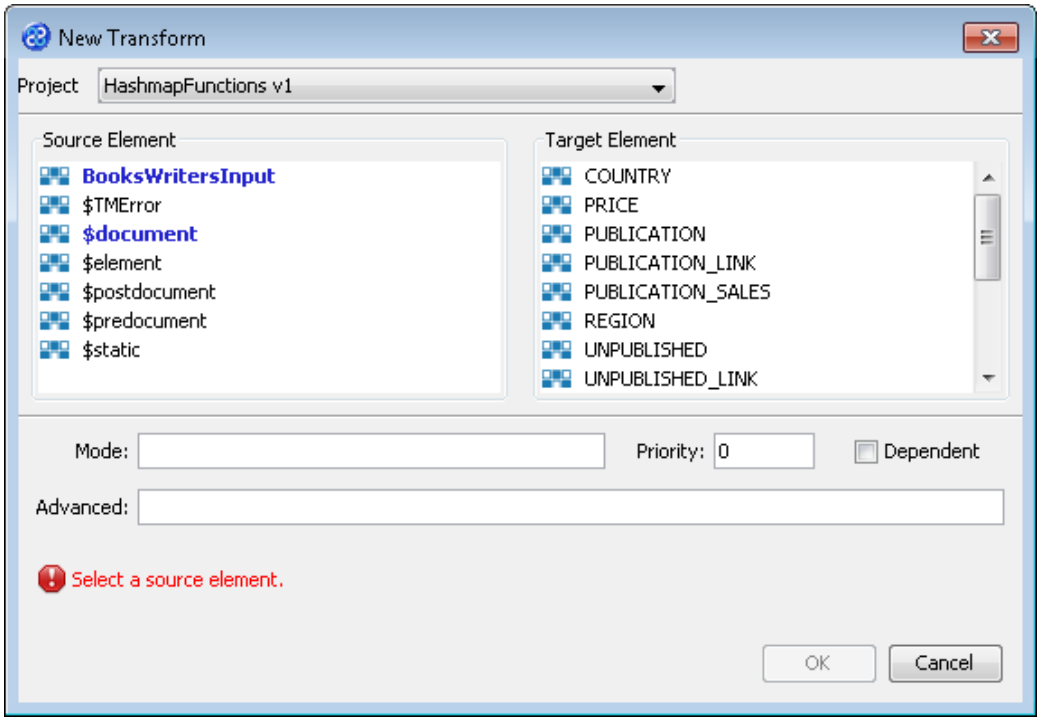
9) Click once on the Save menu bar option to save your transform.

Exercise 17 - Step 3 - Create the dependent transform

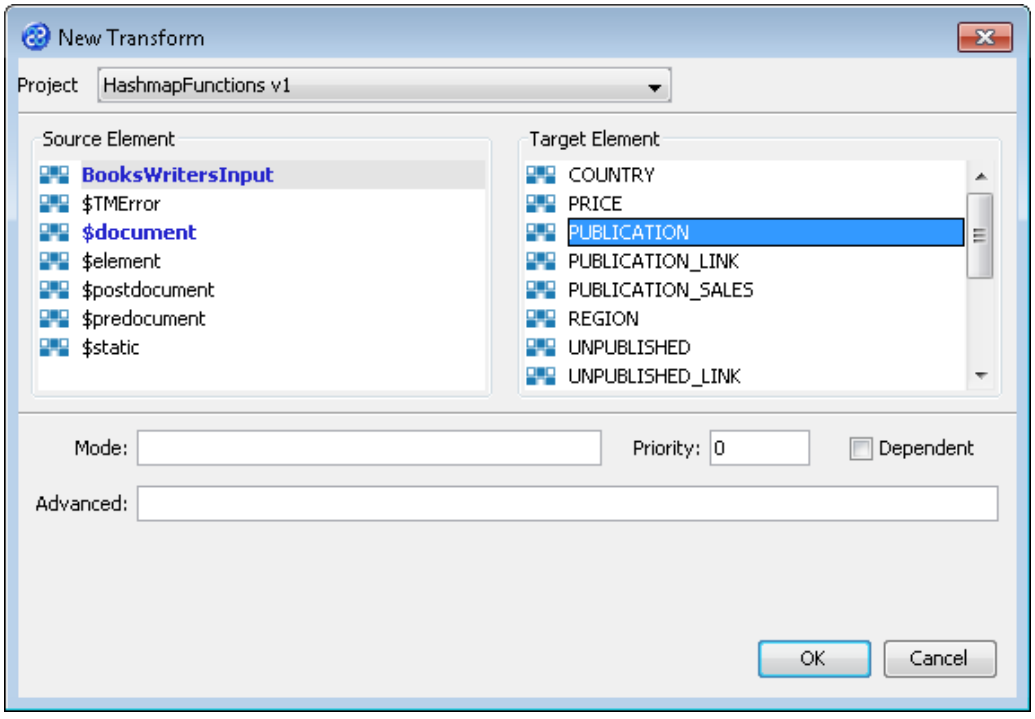
This last step of the HashMap function project is to add our dependent transform.

1) Click once on the File menu bar option.

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



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



4) Move the cursor to the Dependent tick box and place a tick in the box to make the Transform dependent.

5) Click once on the OK 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. The first part of the code creates a local variable called my20Percent which is defined further down in the transform as price multiplied by 0.2. The next step limits the publications we are interested in to those with an ID greater than 400, in this exercise, this means recent publications when we construct our target publication element. Then we derive our values by adding 20% to the book price using our local variable. Finally, we will keep track of the records we add and the modified price for them using our global variable and the JMAPADDKEYVALUE function.

```
LOCAL
my20Percent : integer;
END_LOCAL;

IF (<id> > 400) THEN

CONSTRUCT;
CATNUM:= <id>;
IS_FICTION:= 'N';
NAME:= title;

my20Percent := (price * 0.2) + price;

JMAPADDKEYVALUE(gblHashmap,<id>,my20Percent);

END_IF;
```

8) We have completed our transform.

9) Click once on the File menu bar option.

10) Click once on the Save menu bar option to save your transform.

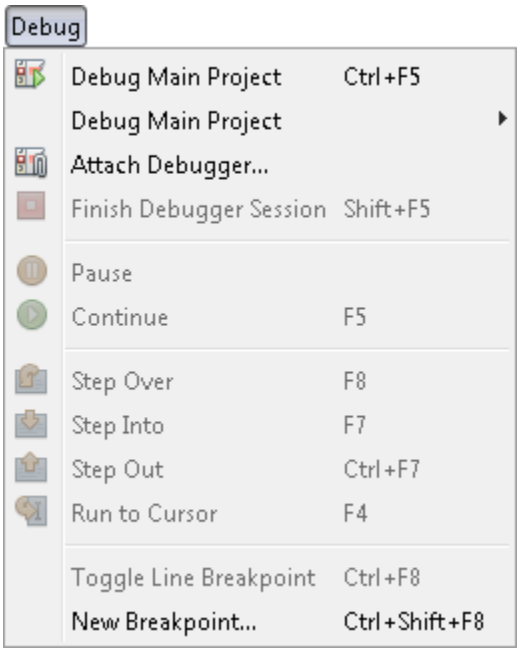
Exercise 18 - Preparing to Debug Your HashmapFunctions Project

In this exercise we are going to use the debugger to check our project is executing correctly.

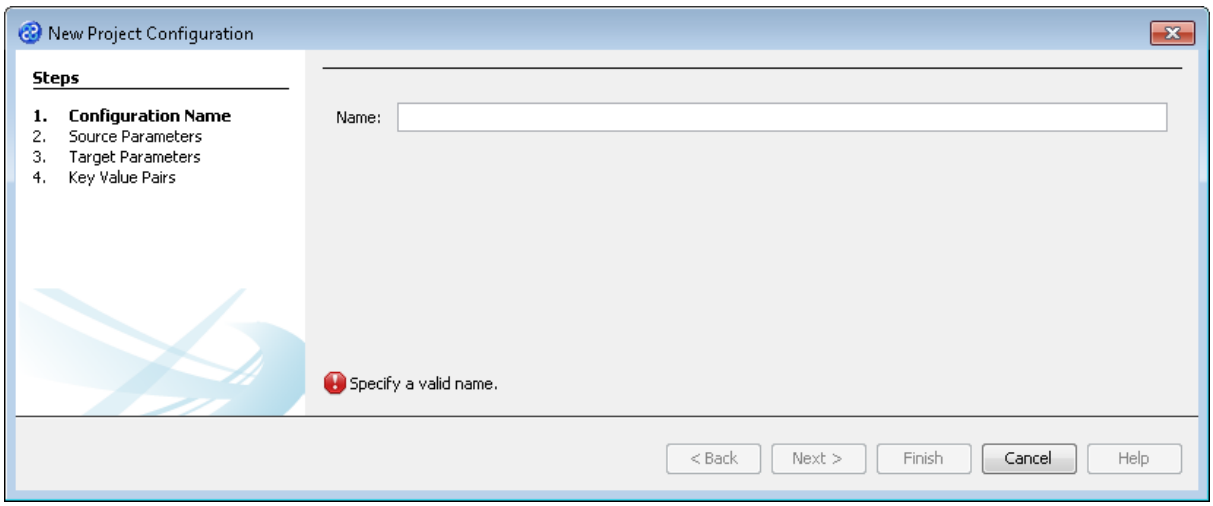
1) Select the **HashmapFunctions v1** project in the **Projects** pane and display the context menu using your secondary or right mouse button.

2) Click the Set as Main Project option to make this project the main project.

3) Click on the Debug menu option.



4) Click on the Debug Main Project option in the sub-menu. The New Project Configuration window will open because we have, as yet, not provided any information on where the data stores are that we will use for the debug session. Now we will provide this information.



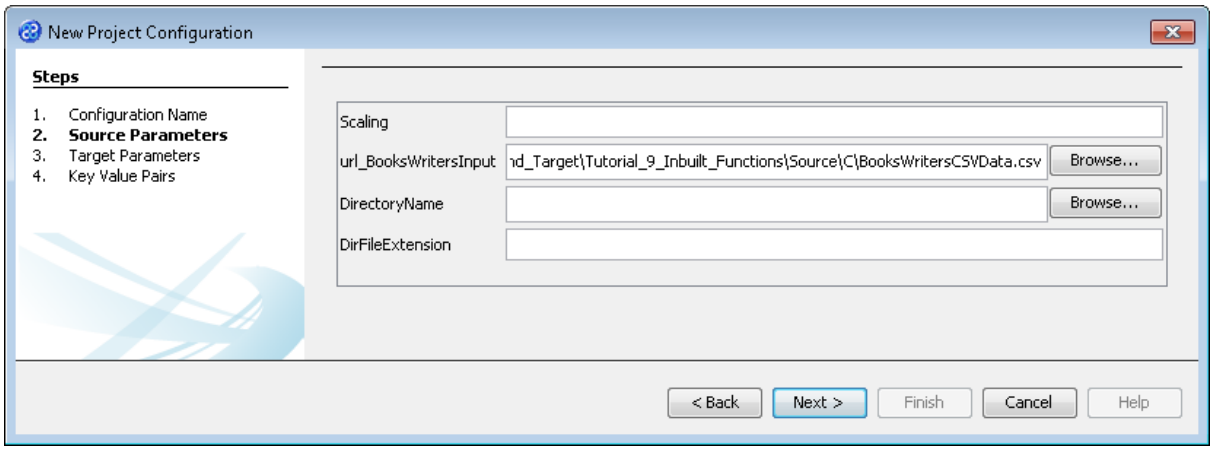
5) Let's give this configuration the name Config1. Type **Config1** into the **Name** field.

6) Click the Next > button to move to **Step 2. Source Parameters** where we will provide the details for the source data store connection.

7) Go to the **url** field of the **Source Parameters** step. In this field we will specify the location of the source data store. You can type the path and file name manually or you can use the button to select the directory and file. This will be in the following directory.

[TMHOME]\Tutorials\Source_and_Target\Tutorial_9_Inbuilt_Functions\Source\C

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



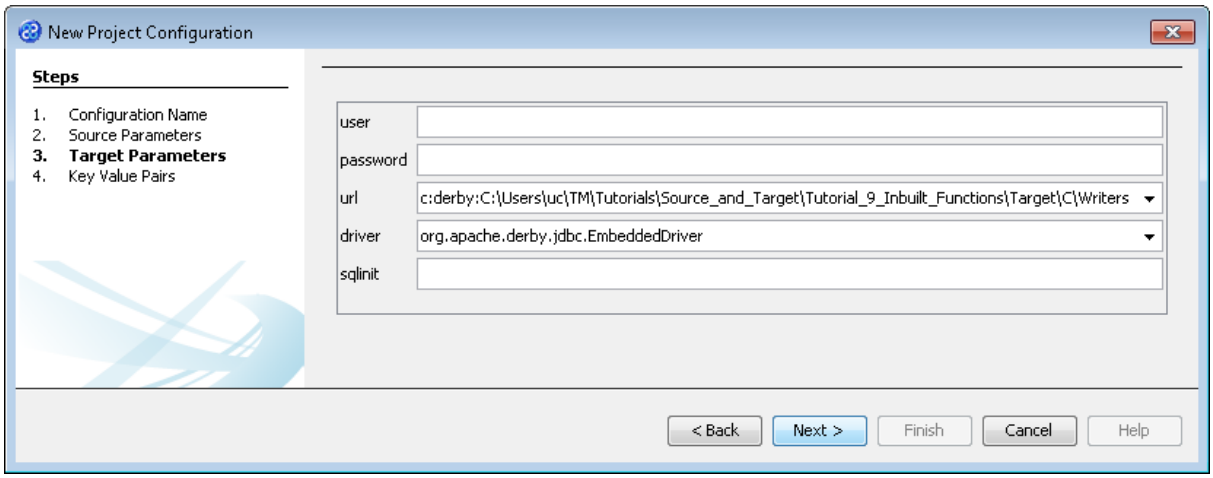
9) Click the Next > button to move to **Step 3. Target Parameters** where we will provide the details for the target data store connection.

10) 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_9_Inbuilt_Functions\Target\C\
Writers
```

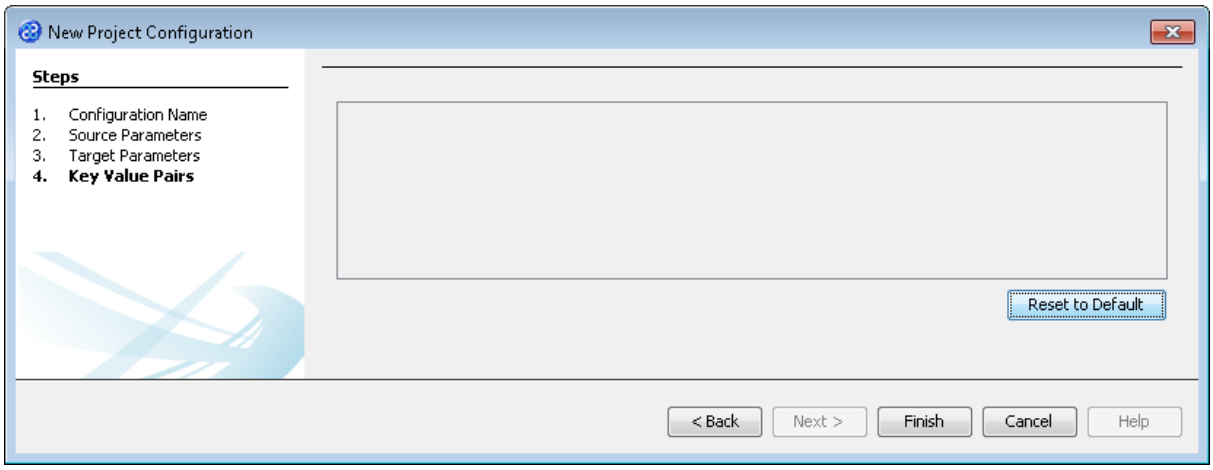
11) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The Adapter Configuration window will look similar to the image below. **user, password** and **sqlinit** do not require values.

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



13) Click the 

button to move to **Step 4. Key Value Pairs**. We do not need to specify any key value pairs in this exercise.



14) Click the 

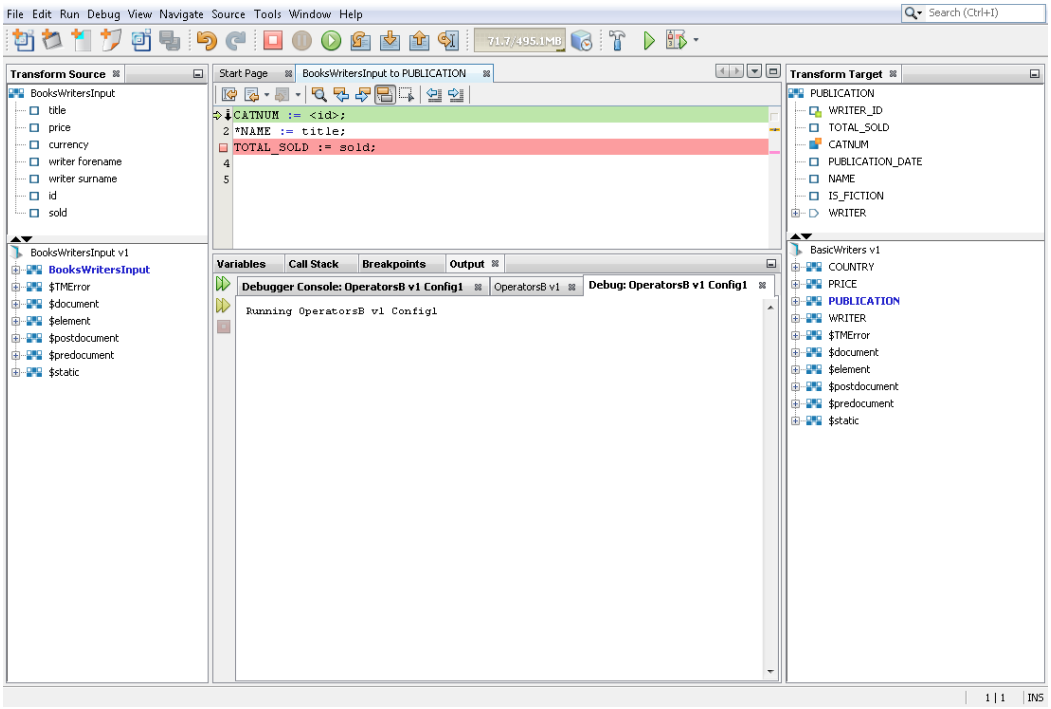
button. This will then cause the project to build making it ready for the debug process.

15) Let's now step through the debugging process for this project.

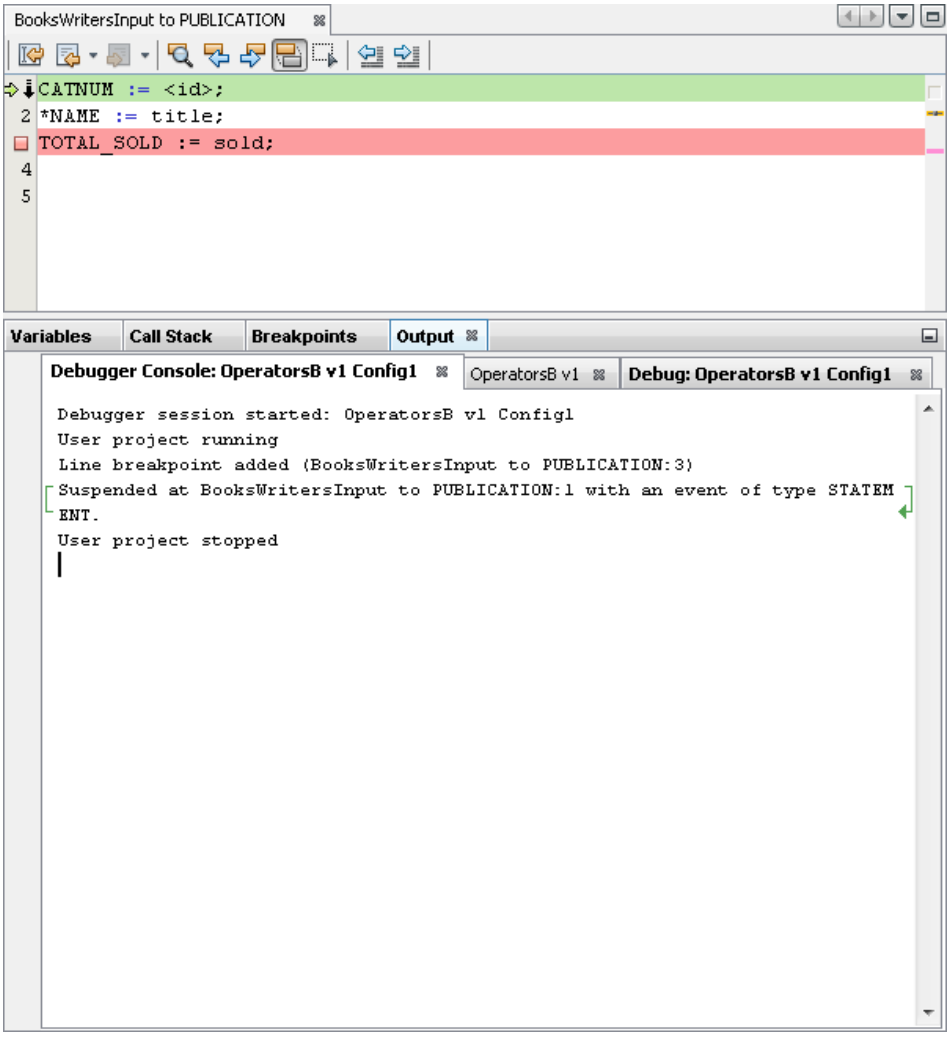
Exercise 19 - Debugging the HashmapFunctions Project

In this project we will use our breakpoint to stop the debug process at a user defined point in the debug process using the **Variables** pane to see how our data is being transformed.

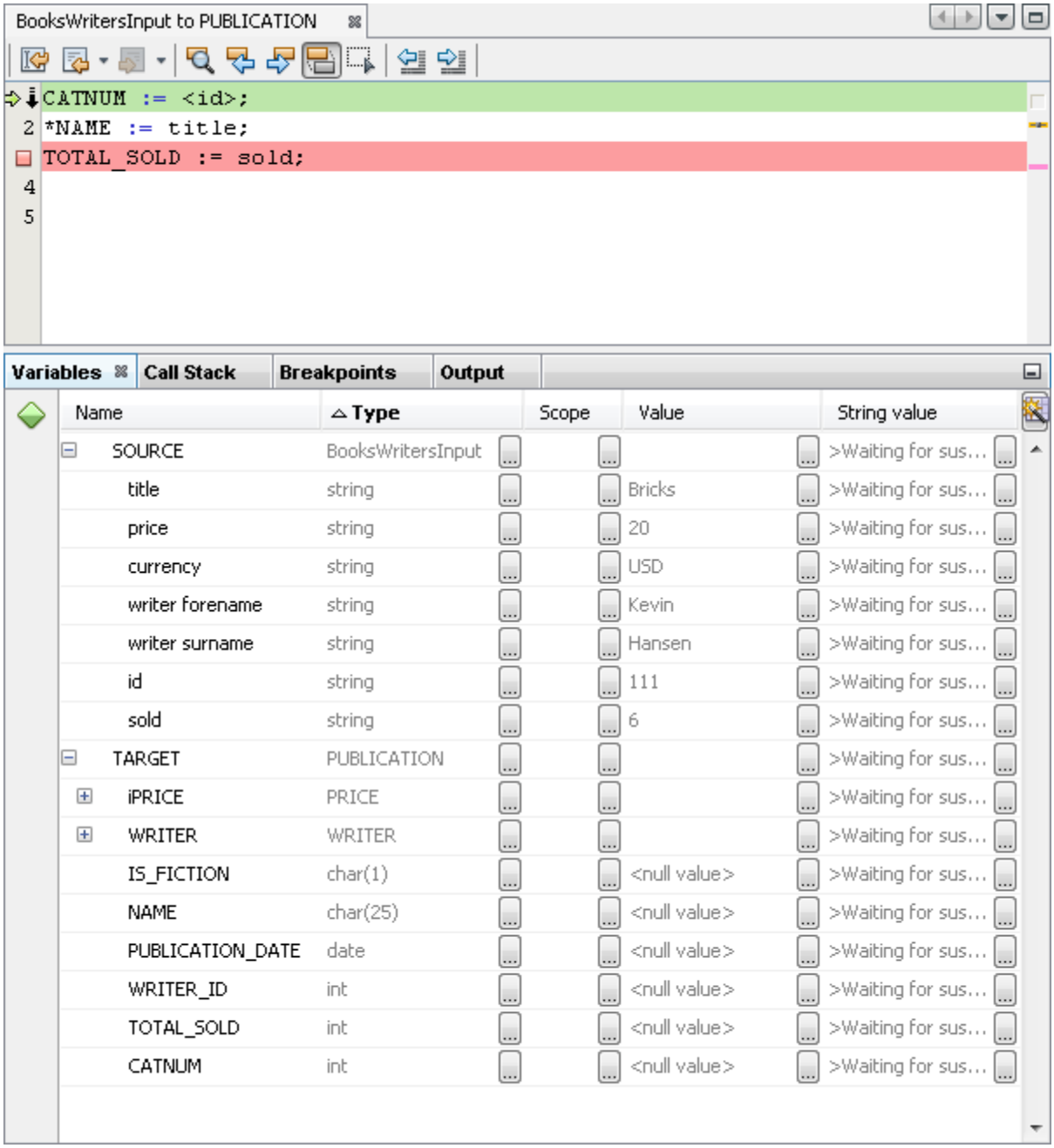
1) Let's look at our starting point. The debug process is suspended at the **CATNUM** line which is highlighted in green and has a small arrow facing right in the left hand margin of the Editor pane. You will also see our breakpoint highlighted in pink.



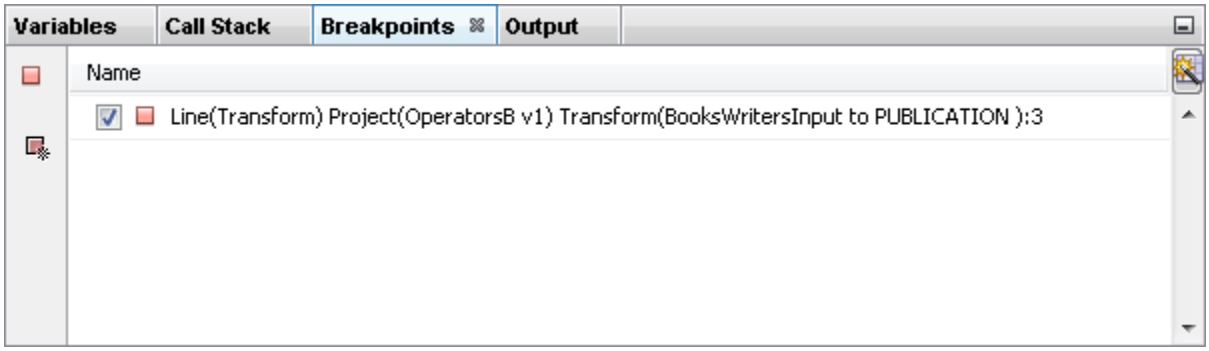
2) Click on the **Debugger Console** pane to display the current state of the debug process. We can see that the first statement has been processed and that it is waiting for the next step to occur.



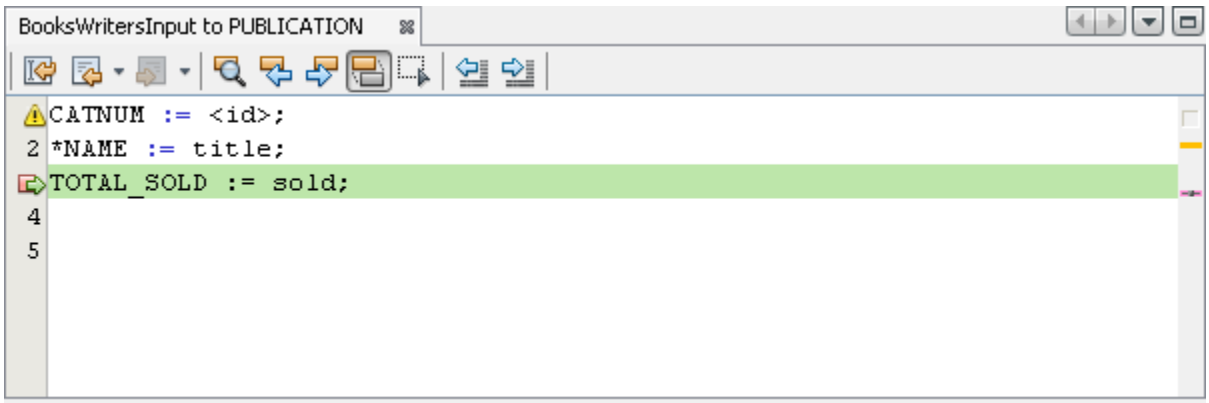
3) Click on the **Variables** pane tab to show its content. You may need to click the ⊞ icon to expand the source and target elements to see the data values. The pane provides access to the state of the data in the transformation process for both the source and the target. We can see that our source and target elements are identified, **BooksWritersInput** and **PUBLICATION**. We can see that the first record has been found in the source but there is nothing for the target element.

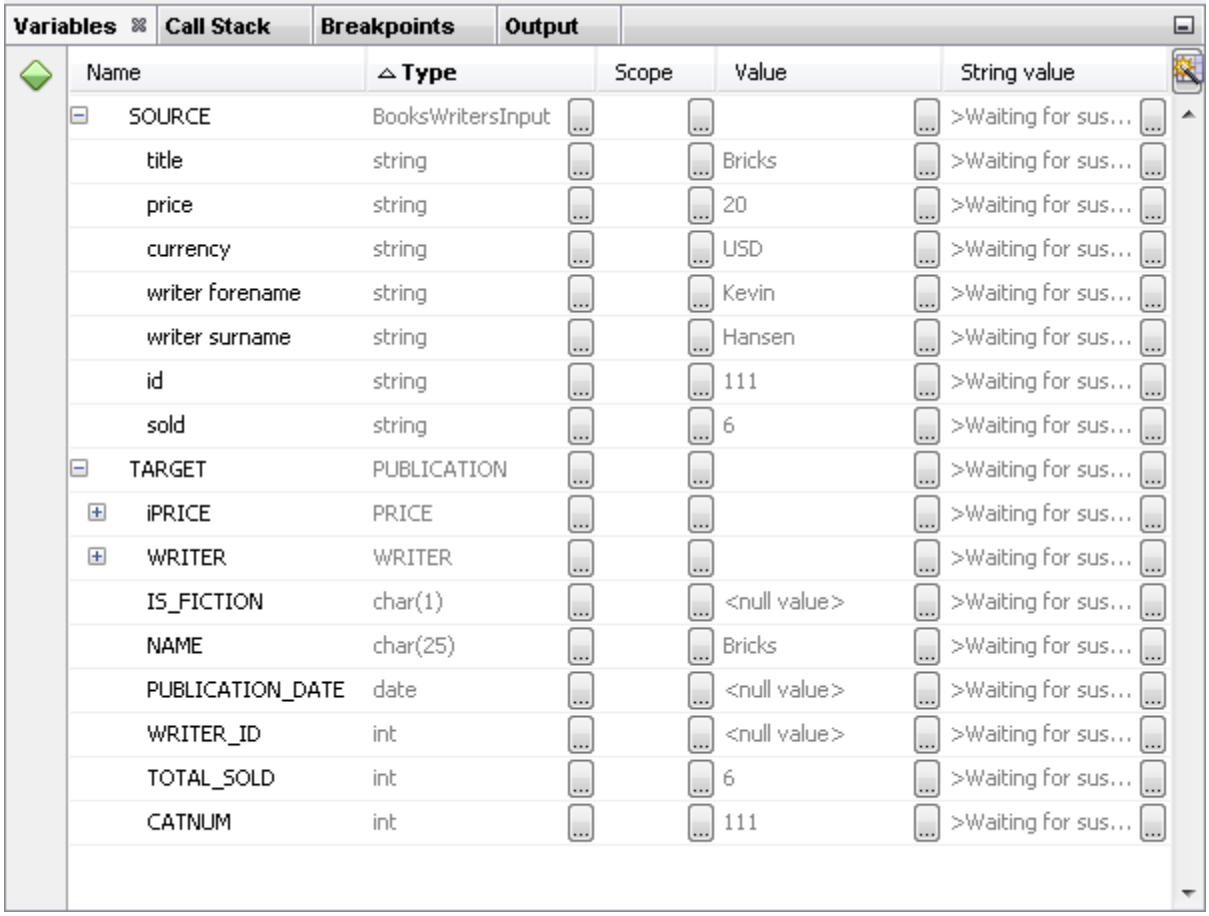


4) Click on the Breakpoints tab to display the Breakpoints pane. This pane provides details of our breakpoint. The tick box lets us turn off the breakpoint if we do not wish to use it for a particular debug process run. The display also tells us that it is a **Line(Transform)** breakpoint, the project it applies to and the transform it is in. In this case it also tells us that it is on line 3 of the transform.

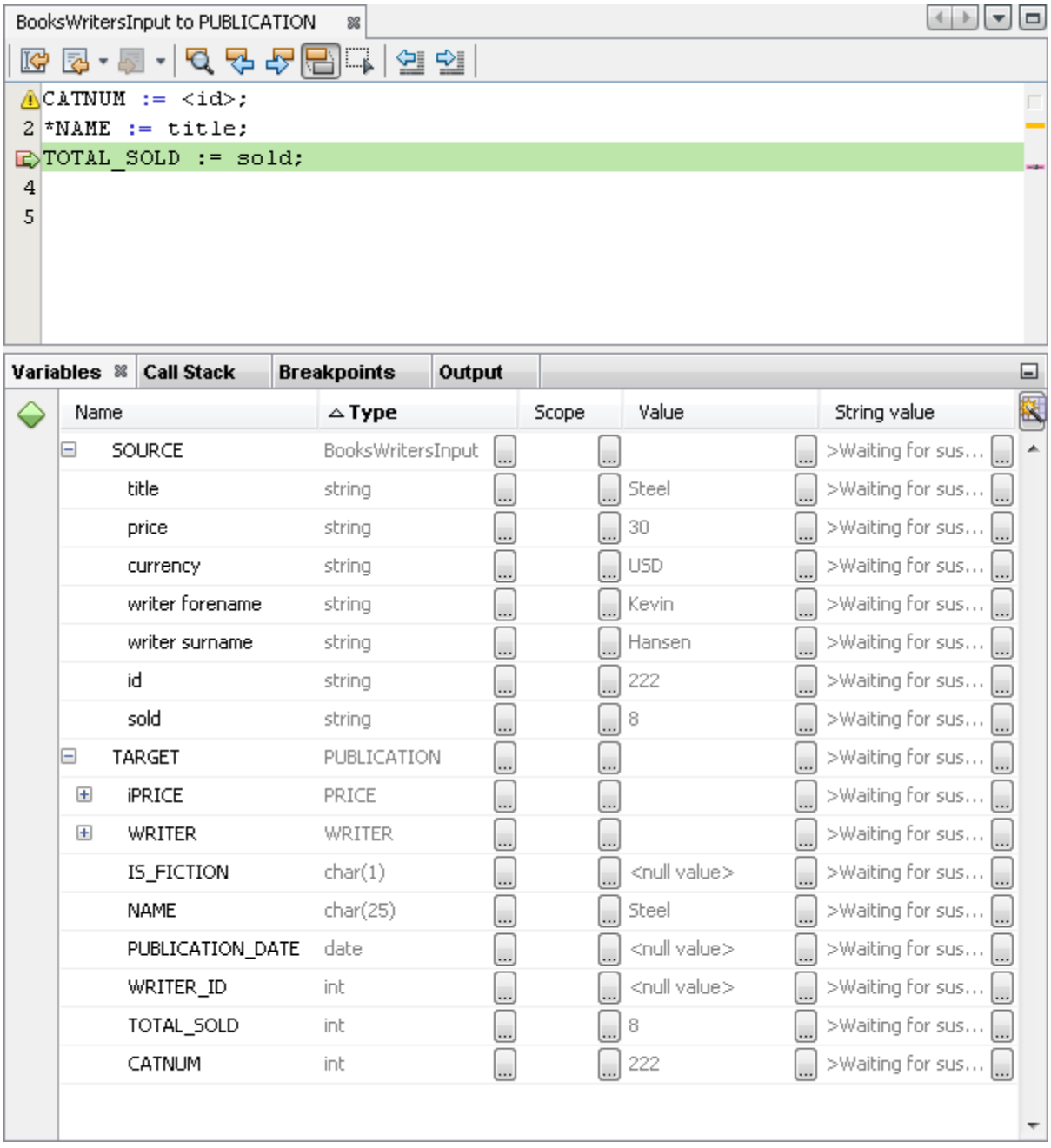


5) Click the  button to now run the process to our breakpoint. Let's check the Variables pane. We can see our first record with id 111 is identified in the source pane. In the target we can see where our three attributes will be placed in the target.

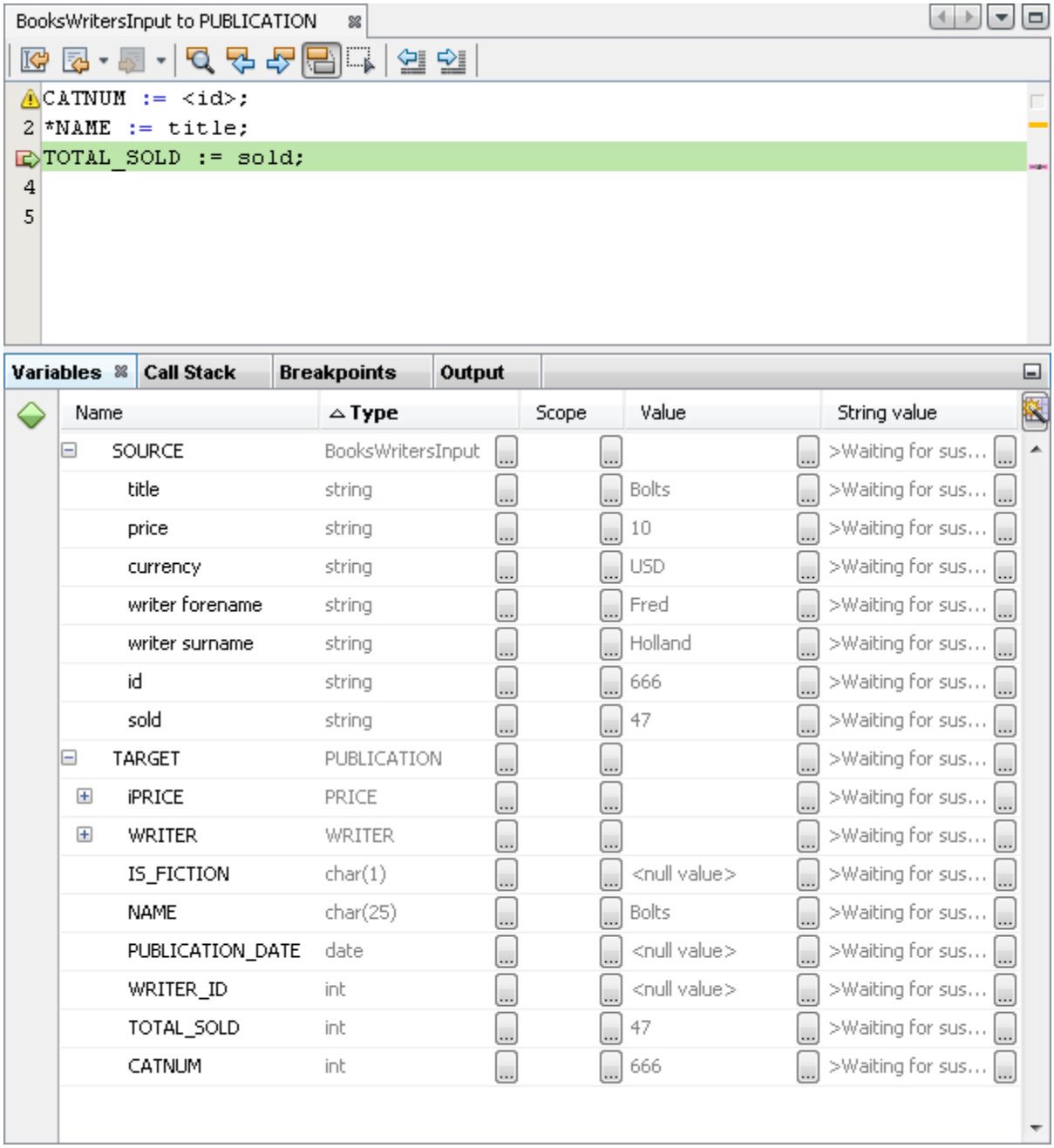




6) Click the  button to run the process to our breakpoint. Now let's check the Variables pane. We can see our second record with id 222 is identified in the source pane. In the target we can see where our three attributes again placed in the target.

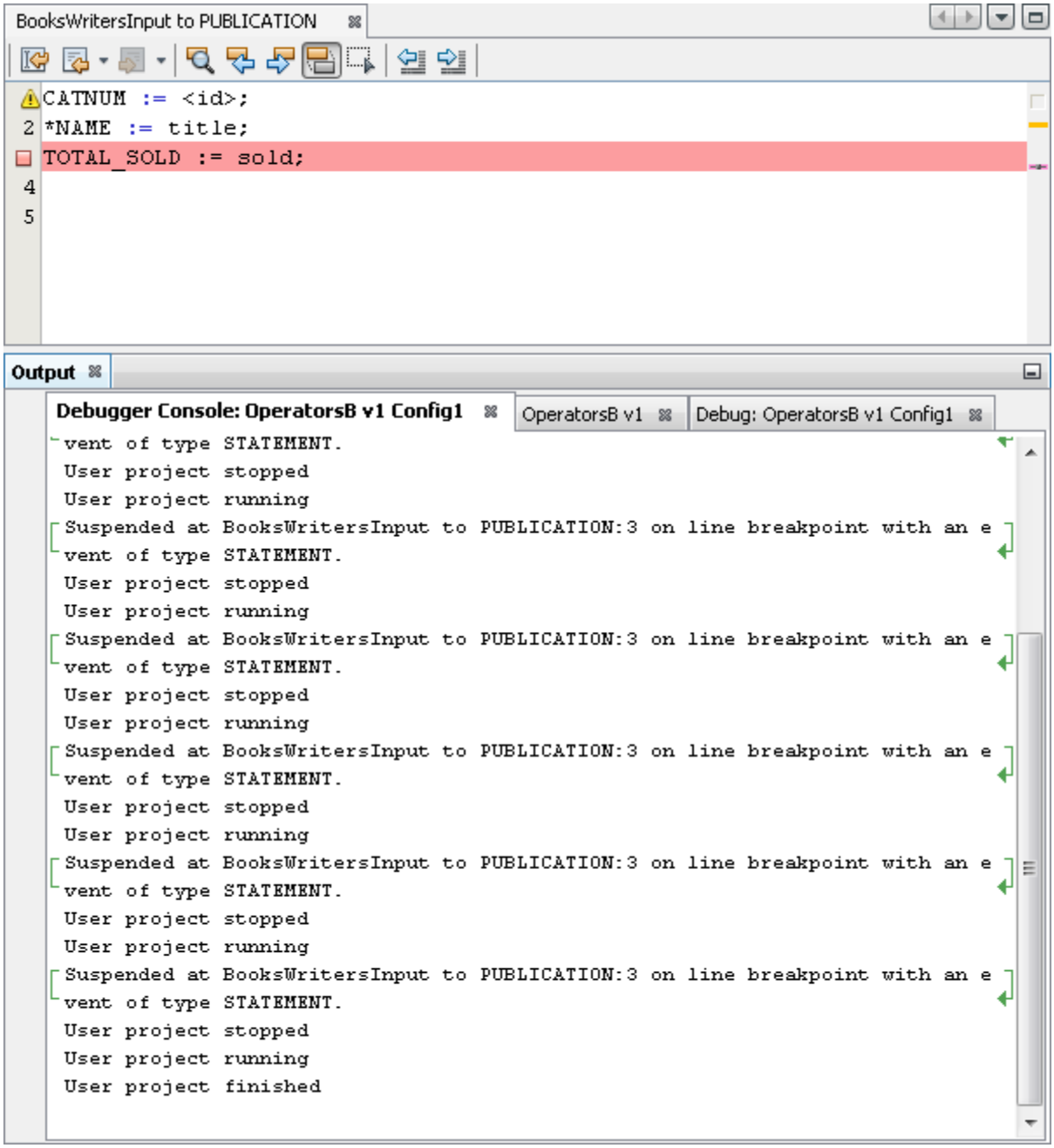


7) Continue clicking the  button and check to see how the data is shown in the Variables pane until the last record with id 666 is displayed in the Variables pane.



8) Click the  button to finish the debug process.

9) Click on the **Debugger Console** pane. We can see that the debug process is complete. Notice the final statement - **User project finished**.

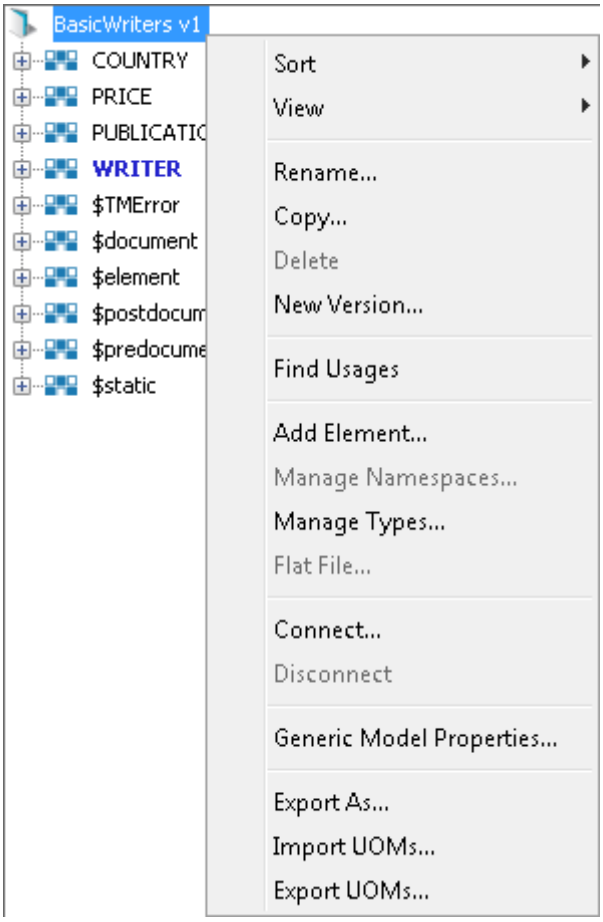


10) Now let's view the data after running the project.

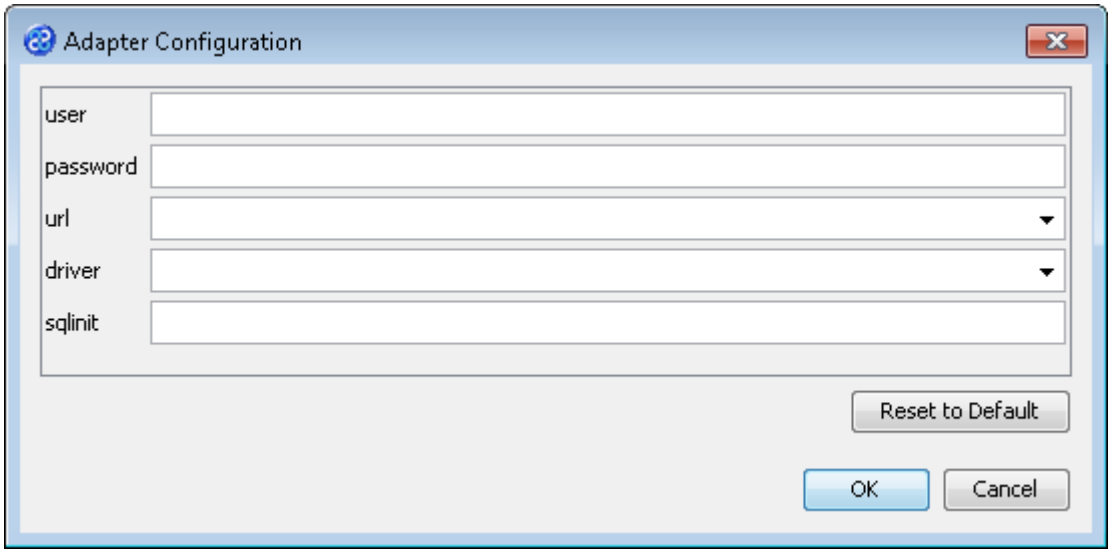
Exercise 20 - View the Target Data

Now lets check our target data store to see if our data has been transformed as expected.

1) With the Editor pane still open, move the cursor over the name of the data model in the **Transform Target** pane and display the context menu for the data model.



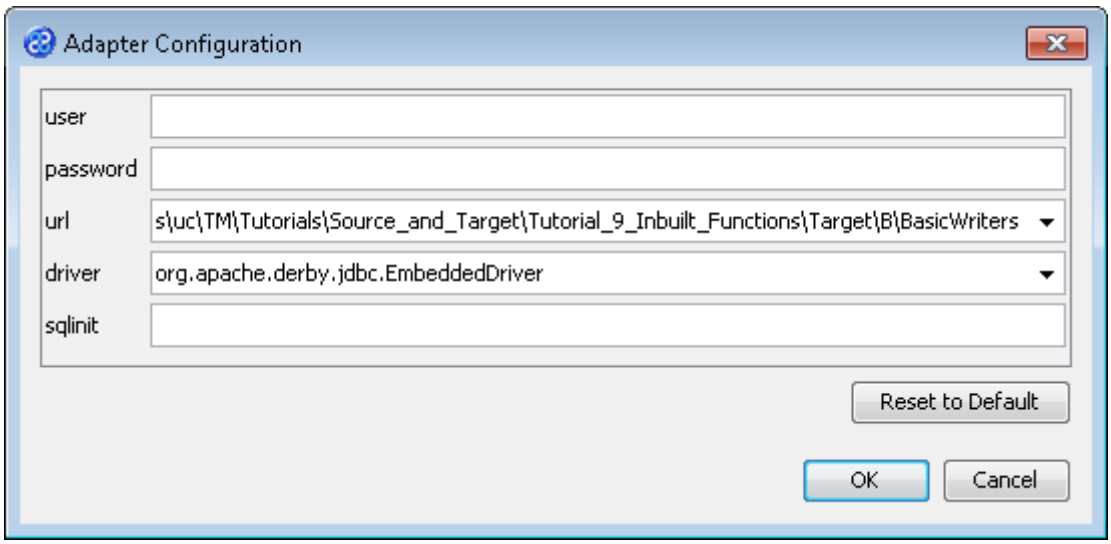
2) Select the Connect... option from the menu. The Adapter Configuration window will open ready for the connection details to be entered for the target data store.

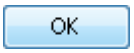


3) Let's provide the relevant connection information as shown below. Go to the **url** field. You will now need to provide the details of where to go to get the target data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> 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_9_Inbuilt_Functions\Target\B\
BasicWriters
```

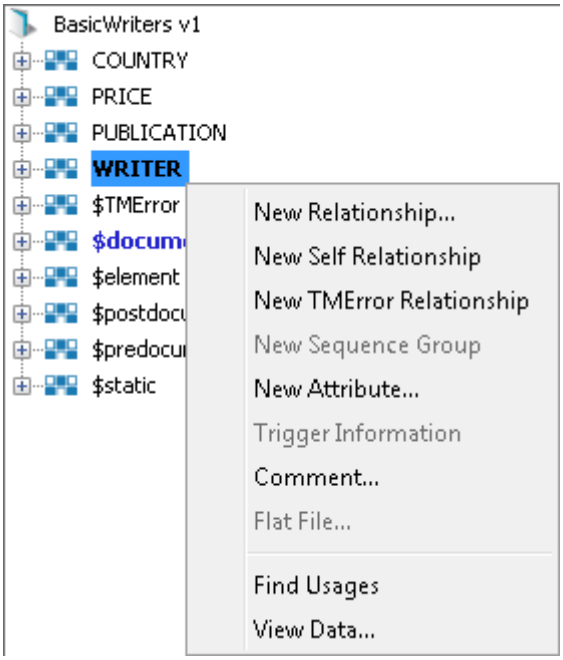
4) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The Adapter Configuration window will look similar to the image below. **user, password** and **sqlinit** are not required.



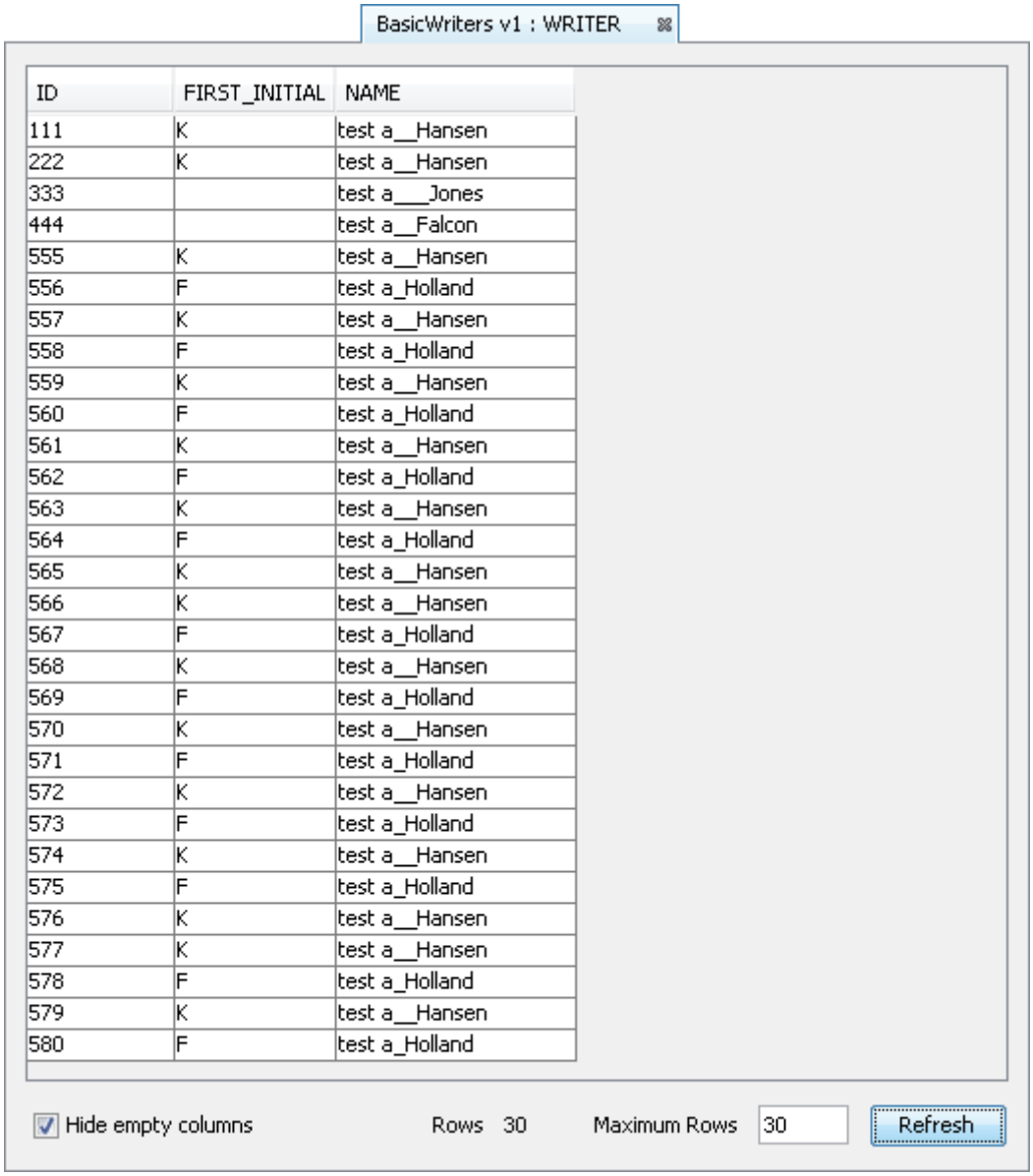
5) Click the 

button to connect to the data store. A message box will appear telling you that the connection is being made.

6) Move your cursor over the **WRITER** element in the data model and display the context menu for the element.

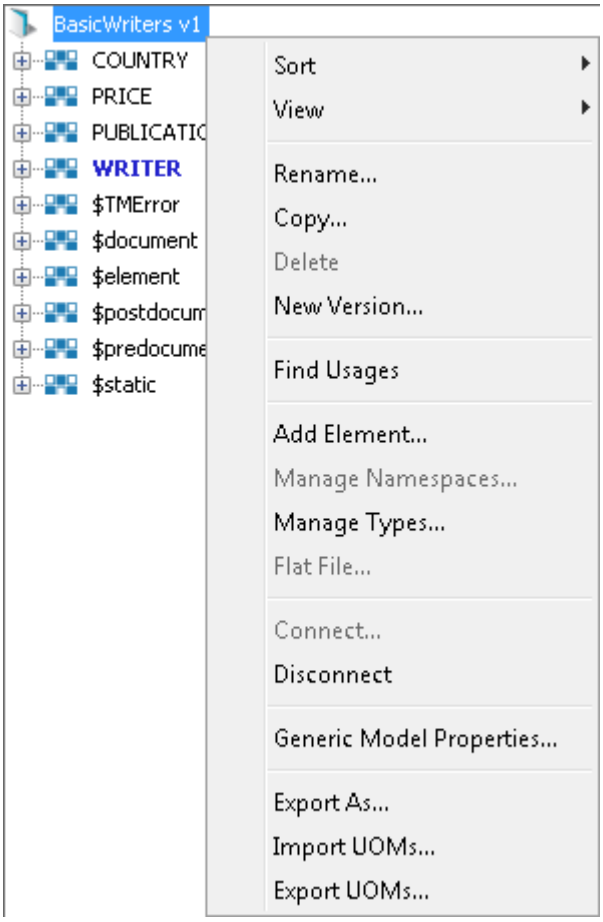


7) Select the View Data... option from the menu. This will open a new pane in the Editor pane displaying the writer data with a sample of 30 records as shown below. These are exactly the same as in the first project but processed much more quickly using the batching functions in the $document transform.



8) Let's close the data pane and disconnect from the data store. Click on the ⊠ icon to close the data view pane for **WRITER**.

9) 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.



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