by experts
for experts

Transformation Manager

Version 5.2

Tutorial 15 - Logging

Transformation
Manager

Copyright Notice

Table of Contents

Tutorial 15

Logging

This tutorial demonstrates logging within Transformation Manager.

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.
5) You should have completed one of the following tutorials, 5, 7, 8, 9, 11, 13 or 16.

Concepts

Transformation Manager provides you with functionality that lets you log what has happened during the execution of a given transformation project.

Information

We will create two projects in this tutorial. The first project creates a relatively simple transform from the book list element to the BOOK element. We will then add a condition which will be used with the currency values from the source data model and then add the LOGINFO function so that we can log text from the source. Lastly, we will create a $document transform and add the SETLOGGINGCONFIG function specifying the location of our newly created logging configuration file. Once this has been completed we will then execute the transform. The effect will be that for each currency over 500 a record will be logged to the file specified in the logging configuration file.

The second project will take a further step in logging by adding the code to produce two log files. We will enhance the log4j configuration file to add the additional configuration information for an additional log file. We will then update the SETLOGGINGCONFIG function to point to the new file. Then the BOOKS LOGINFO function line will be updated to include a second parameter called BOOK. Then we add a cascade transform for authors and also add another LOGINFO line to log all authors migrated. This will require a second parameter which we will call ‘auth’ for the LOGINFO function. The execution of this transform will produce two log files as specified in the log4j configuration file. Each file will contain information specific to that logger and output destination, often referred to as an appender.

Learning Objectives

Use the SETLOGGINGCONFIG function

You will learn how to use the SETLOGGINGCONFIG function.

Use the LOGINFO function

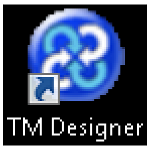
You will learn how to use the LOGINFO function.

Make use of multiple logging files

You will learn how to create and log to more than one logging file.

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**.

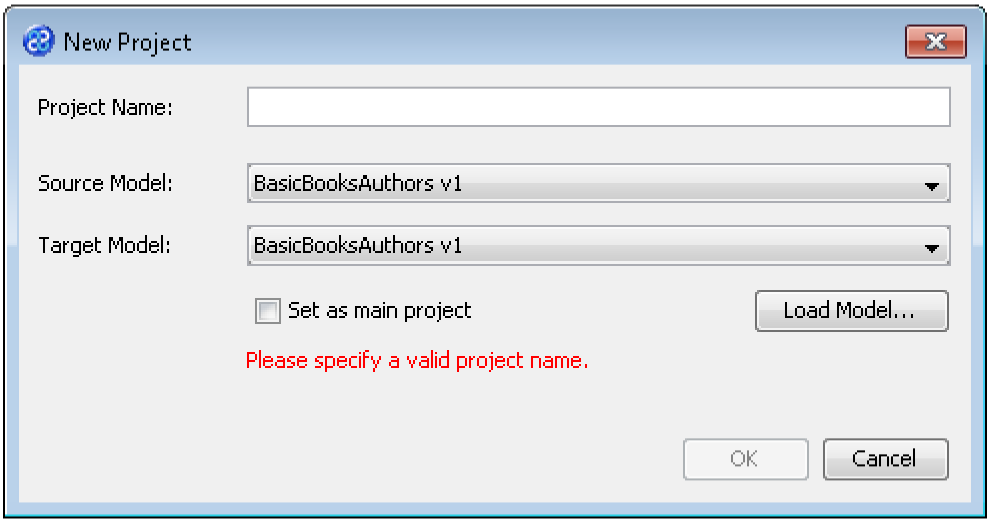
Exercise 2 - Create the LoggingA Project

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.



New Project

Project Name:

Source Model: BasicBooksAuthors v1

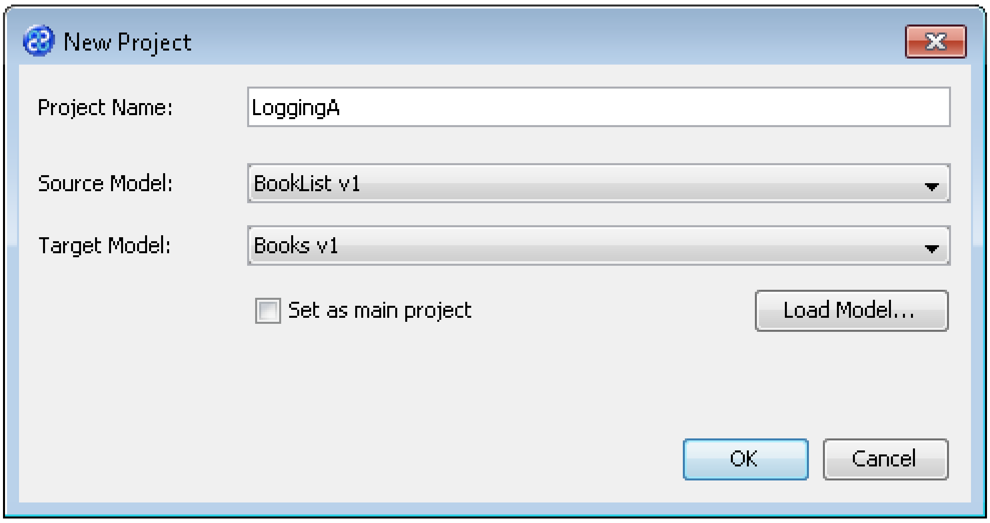
Target Model: BasicBooksAuthors v1

□ Set as main project

Please specify a valid project name.

4) In the **Project Name** field we will provide a name for the project. Let's call our project **LoggingA** 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**.



Project Name: LoggingA

Source Model: BookList v1

Target Model: Books v1

☐ Set as main project

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

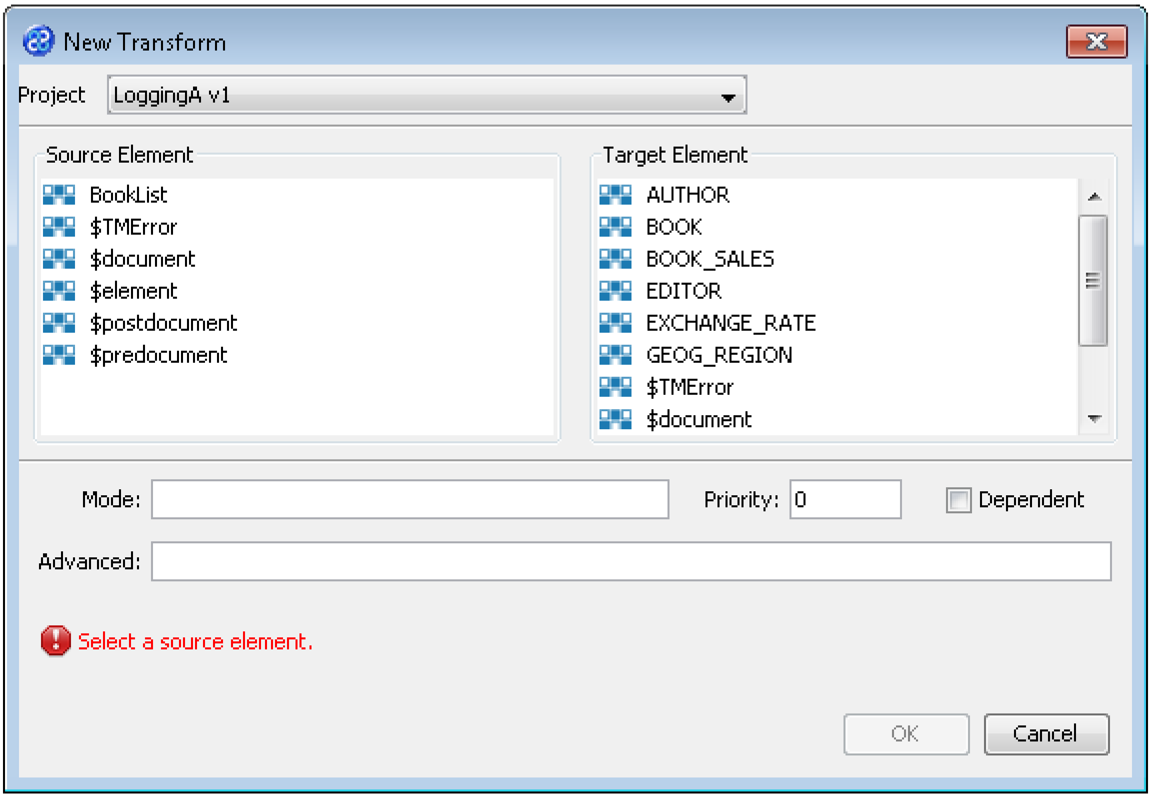
Exercise 3 - Create the Transform for the LoggingA Project

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 **LoggingA v1**.



Project LoggingA v1

Source Element
- BookList
- $TMError
- $document
- $element
- $postdocument
- $predocument

Target Element
- AUTHOR
- BOOK
- BOOK_SALES
- EDITOR
- EXCHANGE_RATE
- GEOG_REGION
- $TMError
- $document

Mode:

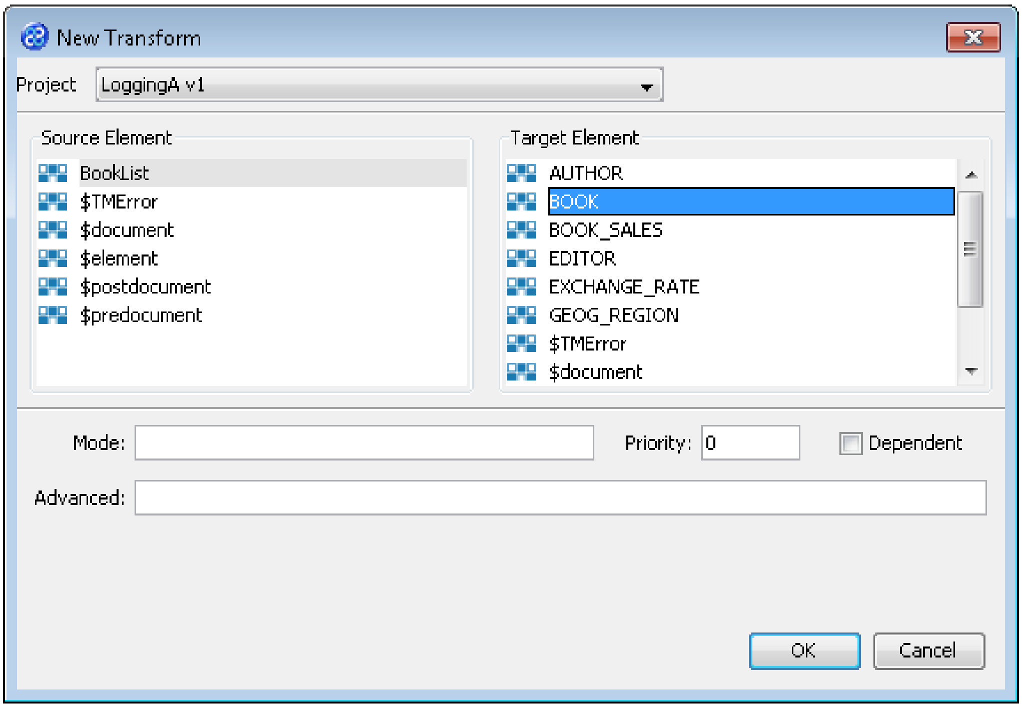
Priority: 0

☐ Dependent

Advanced:

Select a source element.

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.



New Transform

Project: LoggingA v1

Source Element
- BookList
- $TMError
- $document
- $element
- $postdocument
- $predocument

Target Element
- AUTHOR
- BOOK
- BOOK_SALES
- EDITOR
- EXCHANGE_RATE
- GEOG_REGION
- $TMError
- $document

Mode:

Priority: 0

□ Dependent

Advanced:

OK Cancel

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 as shown below.

```
TITLE := title;
<ID> := <id>;
PRICE := price;

IF (currency > '500') THEN
LOGINFO('Currency code greater than 500 for book titled '&title);
ELSE
CURRENCY := currency;
END_IF;
```

8) Click once on the File menu bar option.

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

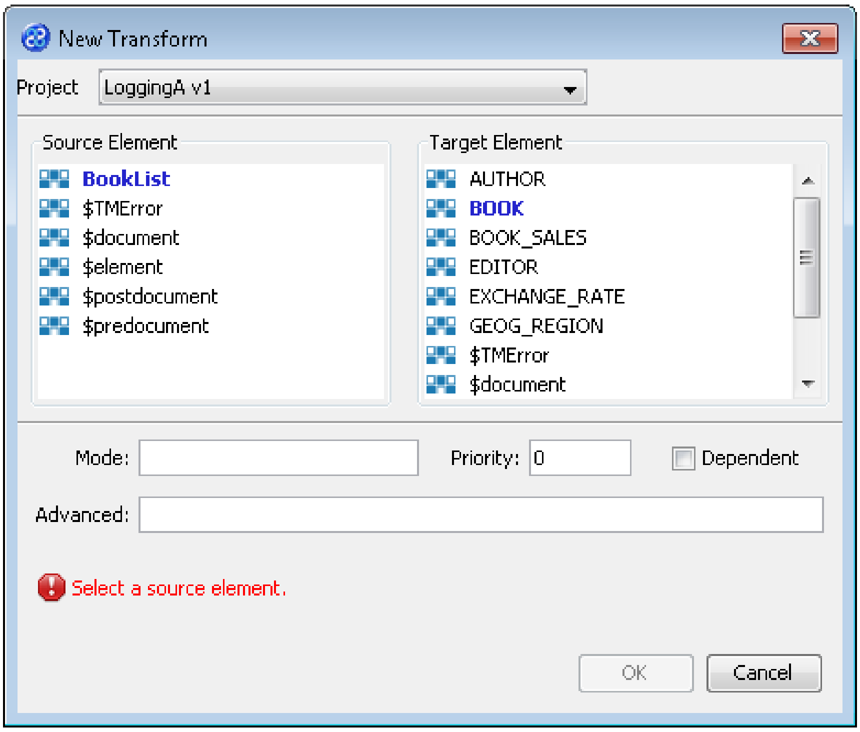
Exercise 4 - Create the $document Transform for the LoggingA Project

In this project we require a $document transform that lets us use a log4j configuration file. The project will use this in order to output our logging file.

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 this case this will be **LoggingA v1**.



New Transform

Project: LoggingA v1

Source Element: **BookList**, $TMError, $document, $element, $postdocument, $predocument

Target Element: AUTHOR, **BOOK**, BOOK_SALES, EDITOR, EXCHANGE_RATE, GEOG_REGION, $TMError, $document

Mode:

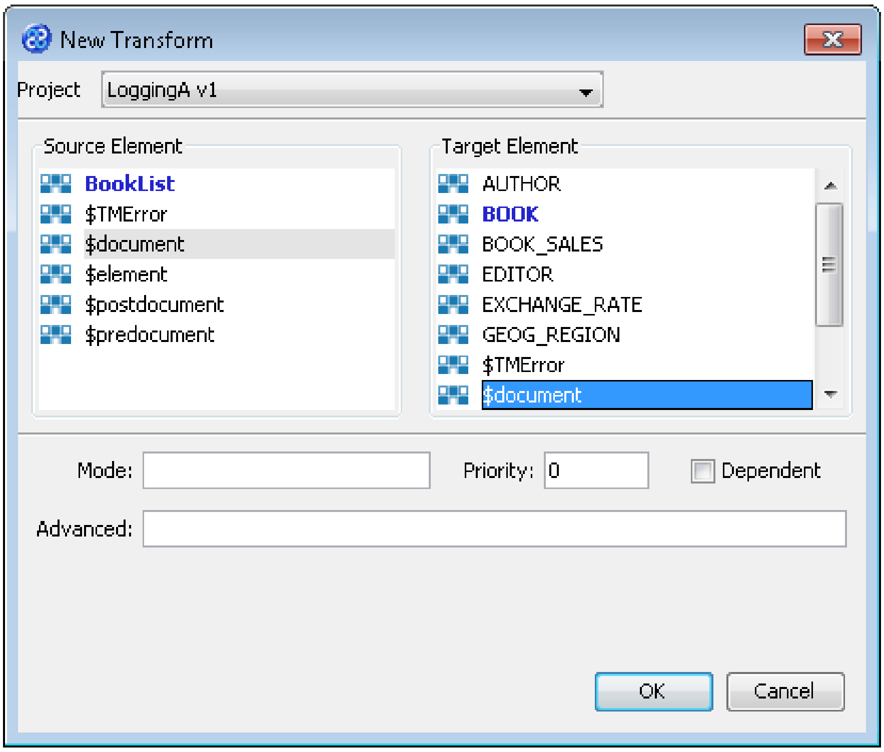
Priority: 0

□ Dependent

Advanced:

Select a source element.

4) With the **New Transform** window open, we will select the elements for the source and target which in this case will be $document for both.



New Transform

Project: LoggingA v1

Source Element: **BookList**, $TMError, $document, $element, $postdocument, $predocument

Target Element: AUTHOR, **BOOK**, BOOK_SALES, EDITOR, EXCHANGE_RATE, GEOG_REGION, $TMError, $document

Mode:

Priority: 0

□ Dependent

Advanced:

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 as shown below.

```
SETLOGGINGCONFIG('[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Loggin
g\\Config\\A\\log4j-LOGGING.xml');
```

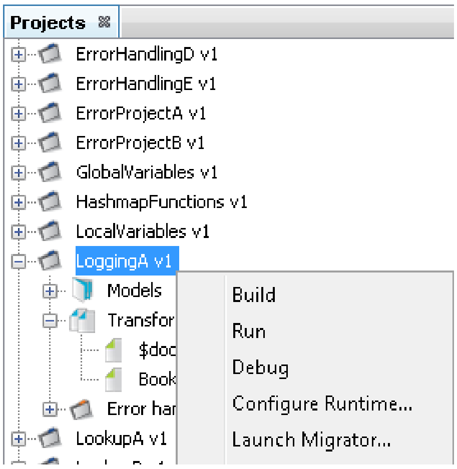
8) Click once on the File menu bar option.

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

Exercise 5 - Build Your LoggingA Project

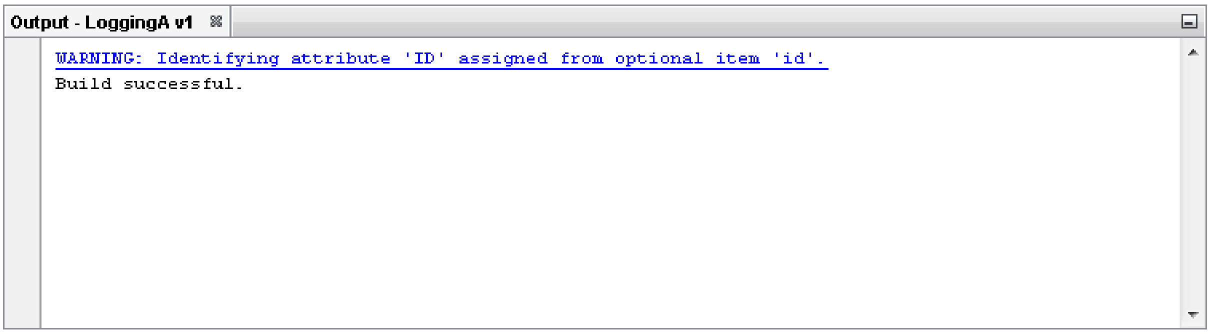
Now we will build our project.

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**.



```
WARNING: Identifying attribute 'ID' assigned from optional item 'id'.
Build successful.
```

Exercise 6 - Edit the log4j Configuration File

We will need to check and edit our log4j configuration file. These can be found in the tutorials directory specifically in the following location.

`[TMHOME]\Tutorials\Source_and_Target\Tutorial_15_Logging\Config\A`

The file we supply has a param line as follows :-

```
<param name="File"
value="C:\\\\[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Log\\A\\B
ooksLog.log" />
```

but for our project we need to edit the `[TMHOME]` section and provide the actual location of your Transformation Manager home directory. Below you can see our supplied configuration file and an example of what it might look like updated for a windows user after [TMHOME] has been updated. Note that the back slash characters are doubled so that log4j can interpret the path correctly i.e. a single \ becomes \\ and a double \\ becomes \\\\.

ETL TemplateFile

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="BOOKS" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File"
value="C:\\\\[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Log\\A\\B
ooksLog.log"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{dd-MMM-yyyy HH:mm:ss} - %p - %m%n"
/>
    </layout>
  </appender>
  <root>
    <level value="INFO" />
    <appender-ref ref="BOOKS" />
  </root>
</log4j:configuration>
```

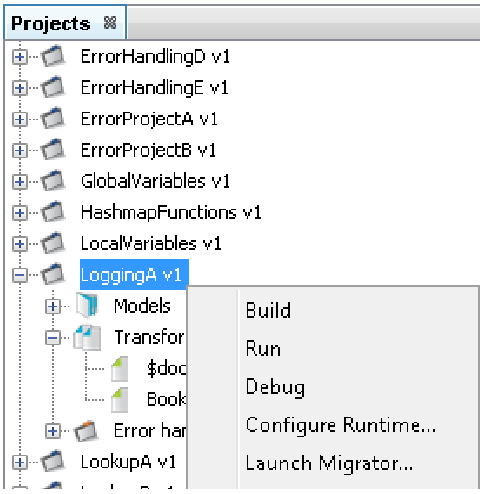
Your Edited File

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="BOOKS" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File"
value="C:\\\\Users\\John\\TM\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Lo
g\\A\\BooksLog.log"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{dd-MMM-yyyy HH:mm:ss} - %p - %m%n"
/>
    </layout>
  </appender>
  <root>
    <level value="INFO" />
    <appender-ref ref="BOOKS" />
  </root>
</log4j:configuration>
```

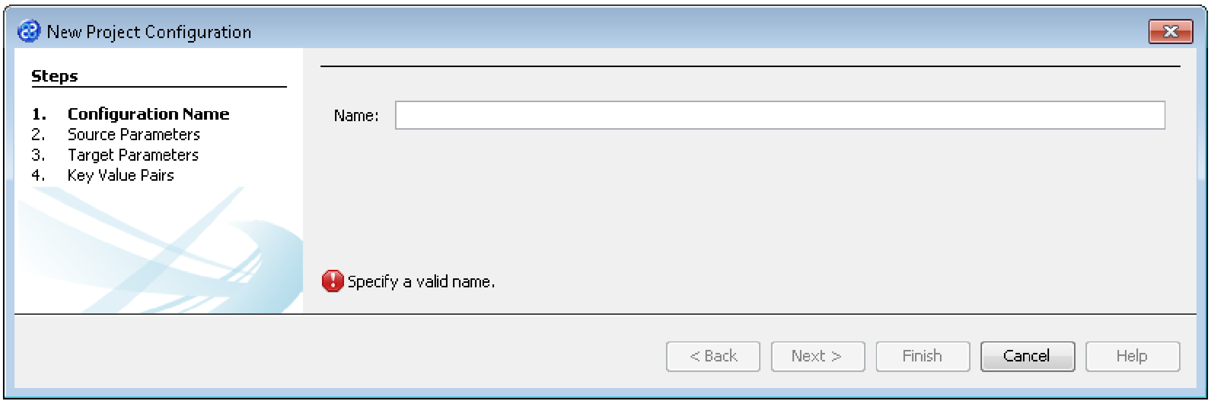
Exercise 7 - Run the Project

Now let's Run our transformation to see what results are produced. We have told the main transform to LOGINFO and provided a $document transform specifying the location of where the log file is.

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.



Steps

1. **Configuration Name**
2. Source Parameters
3. Target Parameters
4. Key Value Pairs

Name:

Specify a valid name.

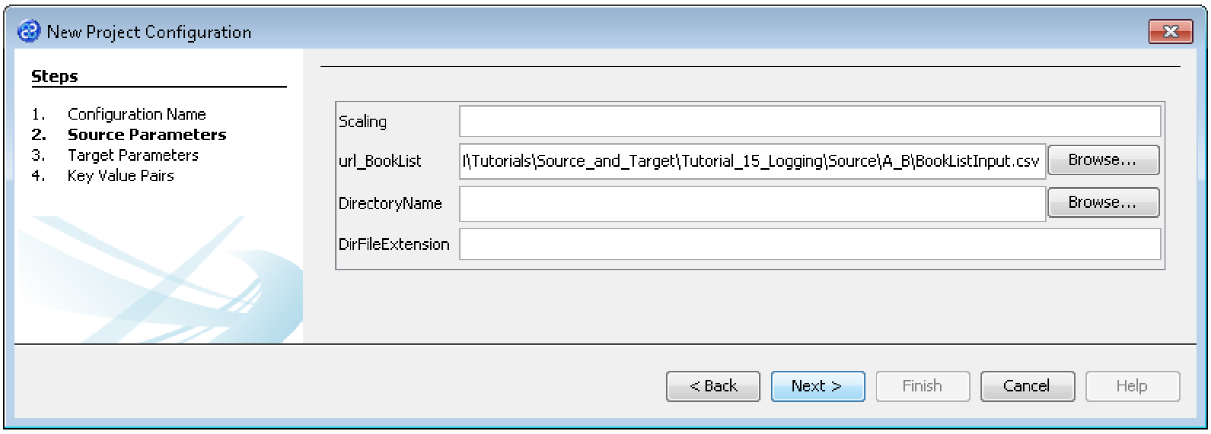
3) Let's give this configuration the name Config1. Type **Config1** into the **Name** field.

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_15_Logging\Source\A_B\BookListInput.csv

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



Steps

1. Configuration Name
2. **Source Parameters**
3. Target Parameters
4. Key Value Pairs

Scaling

url_BookList: l\Tutorials\Source_and_Target\Tutorial_15_Logging\Source\A_B\BookListInput.csv

DirectoryName

DirFileExtension

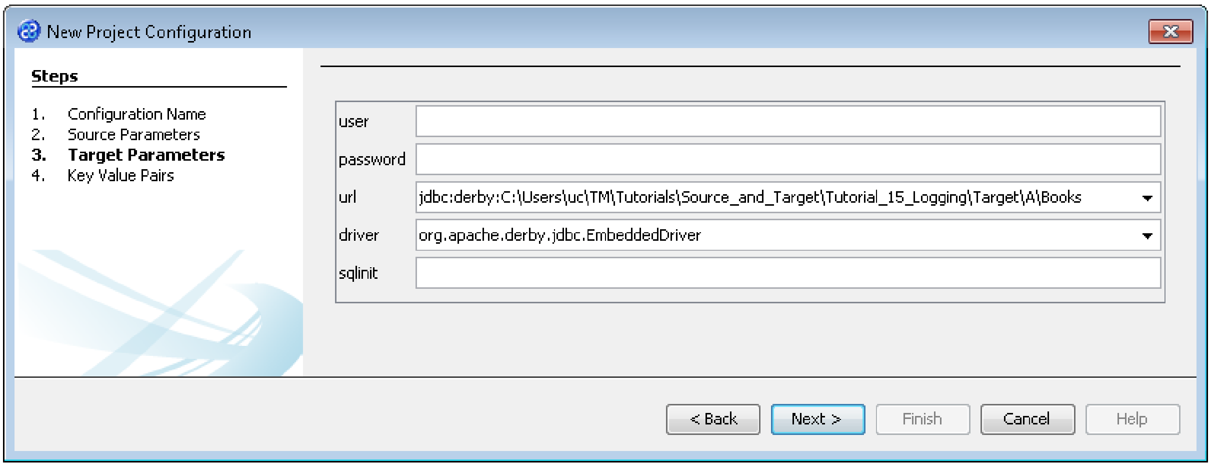
7) Click the Next > button to move to **Step 3. Target Parameters** where we will provide the details for the target data store connection.

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_15_Logging\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.



New Project Configuration

Steps

1. Configuration Name
2. Source Parameters
3. **Target Parameters**
4. Key Value Pairs

user

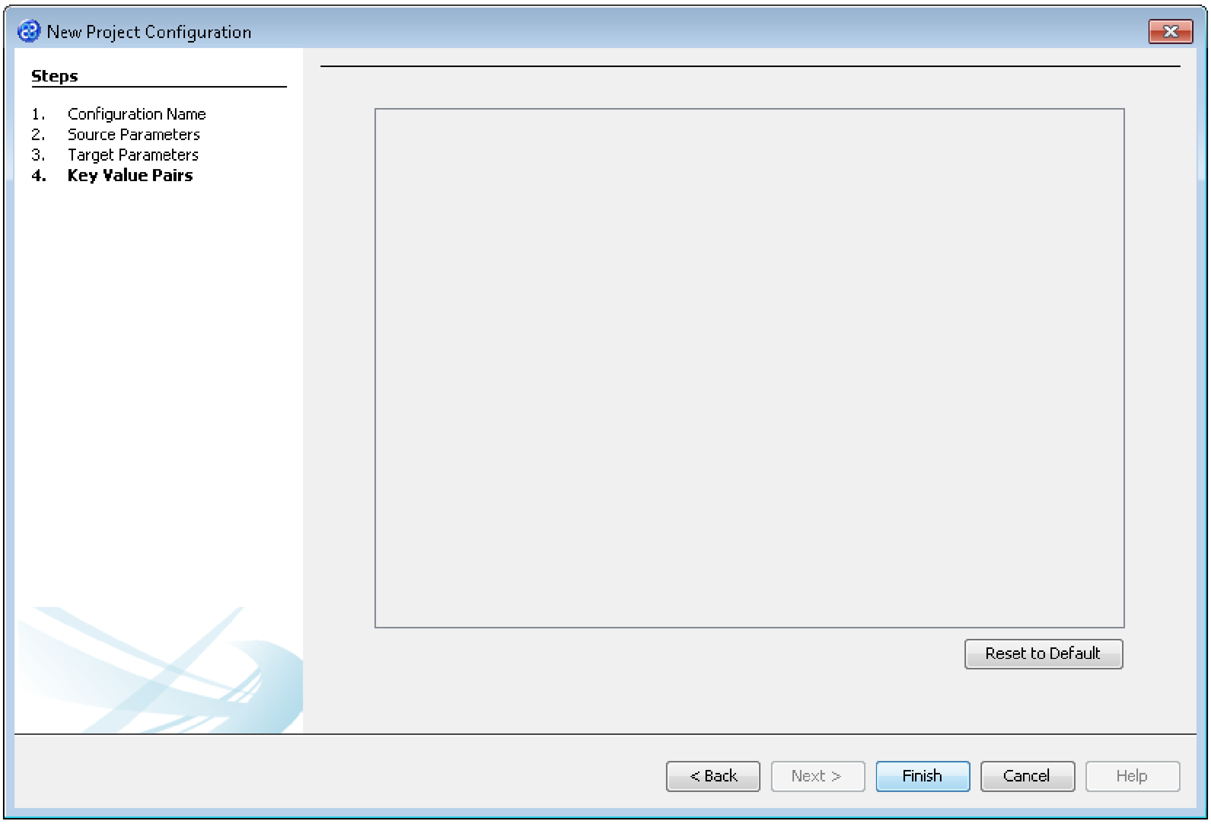
password

url jdbc:derby:C:\Users\uc\TM\Tutorials\Source_and_Target\Tutorial_15_Logging\Target\A\Books

driver org.apache.derby.jdbc.EmbeddedDriver

sqlinit

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.



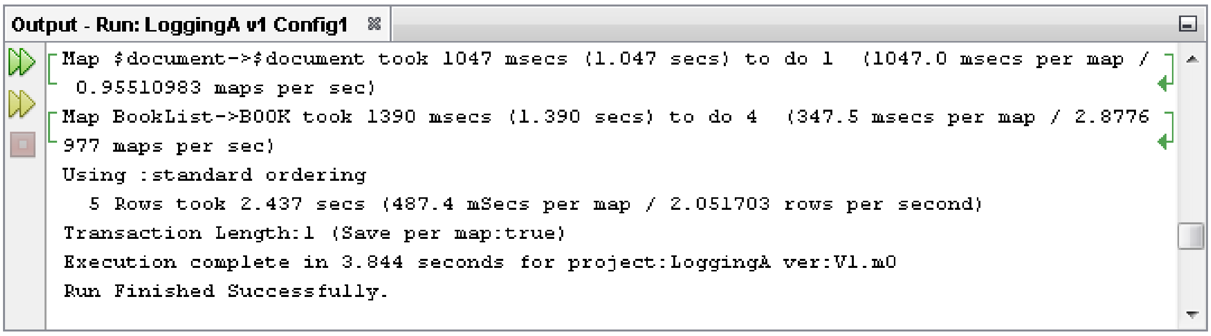
New Project Configuration

Steps

1. Configuration Name
2. Source Parameters
3. Target Parameters
4. **Key Value Pairs**

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.



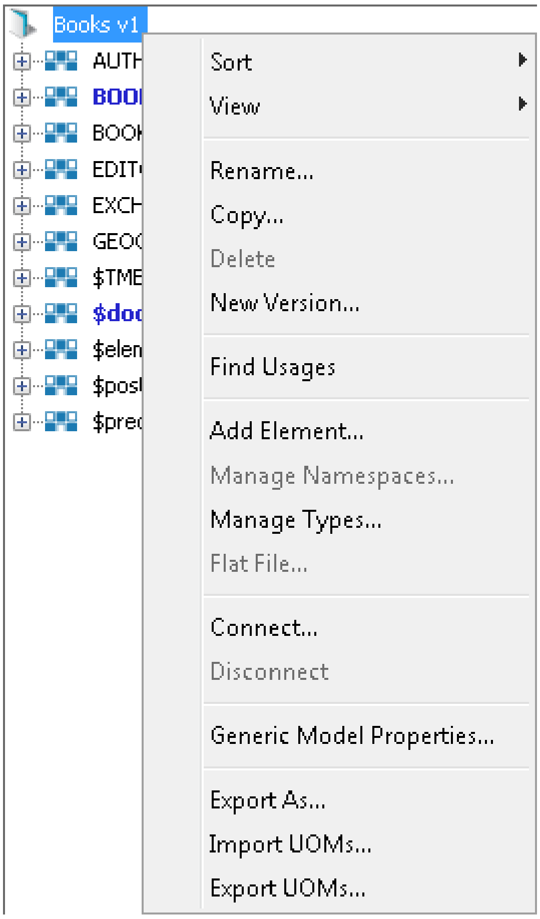
```
Map $document->$document took 1047 msecs (1.047 secs) to do 1  (1047.0 msecs per map /
 0.95510983 maps per sec)
Map BookList->BOOK took 1390 msecs (1.390 secs) to do 4  (347.5 msecs per map / 2.8776
977 maps per sec)
Using :standard ordering
  5 Rows took 2.437 secs (487.4 mSecs per map / 2.051703 rows per second)
Transaction Length:1 (Save per map:true)
Execution complete in 3.844 seconds for project:LoggingA ver:V1.m0
Run Finished Successfully.
```

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

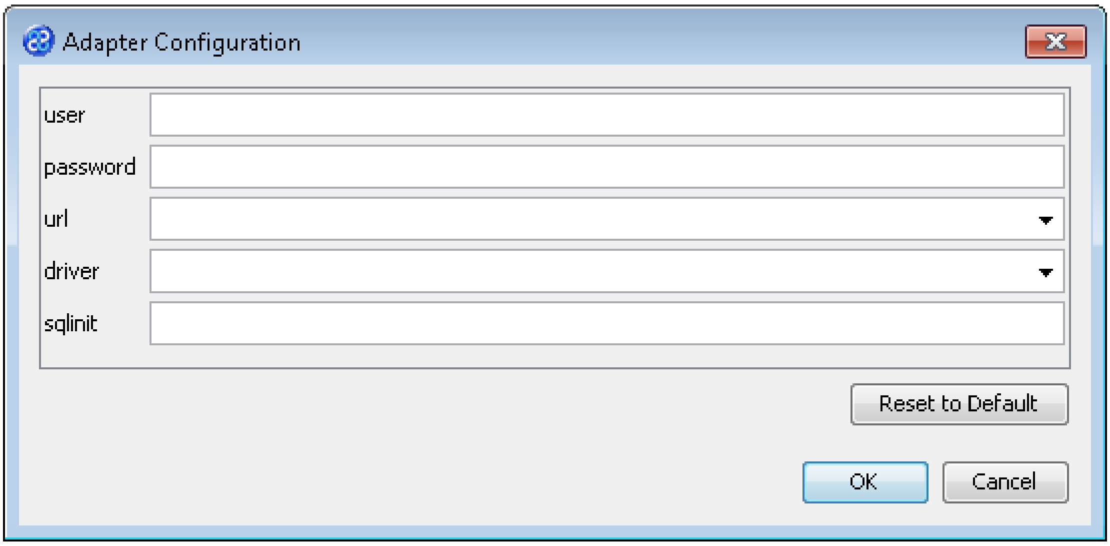
Exercise 8 - View the Target Data

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.

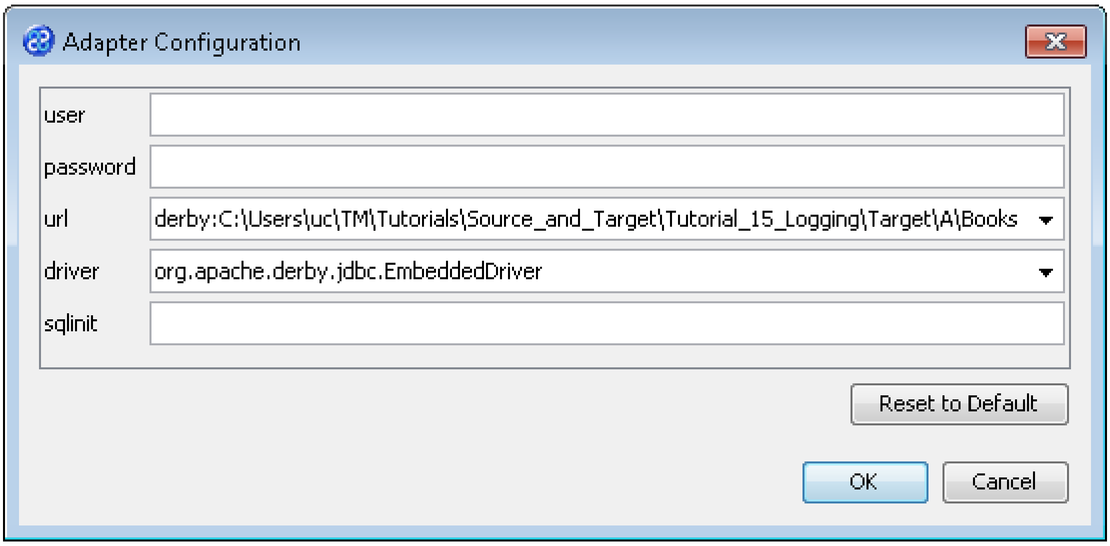


3) Go to the **url** field. You will now need to provide the details of where to go to connect to the source data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> part of the list item, including the angle brackets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

```
[TMHOME]\Tutorials\Source_and_Target\Tutorial_15_Logging\Target\A\Books
```

4) 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.

5) The adapter configuration should look similar to the one below.



user

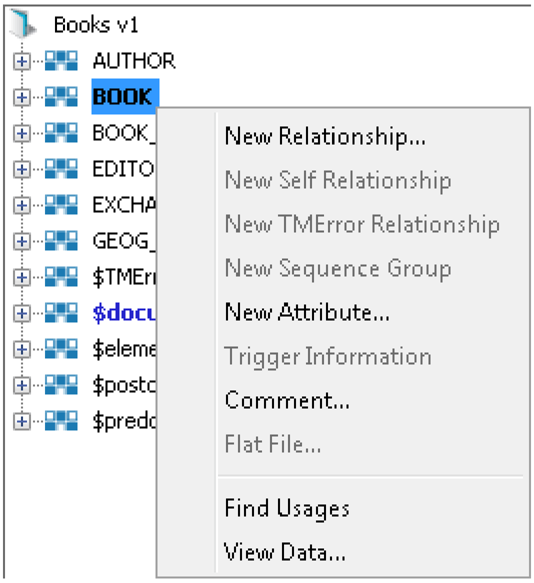
password

url derby:C:\Users\uc\TM\Tutorials\Source_and_Target\Tutorial_15_Logging\Target\A\Books

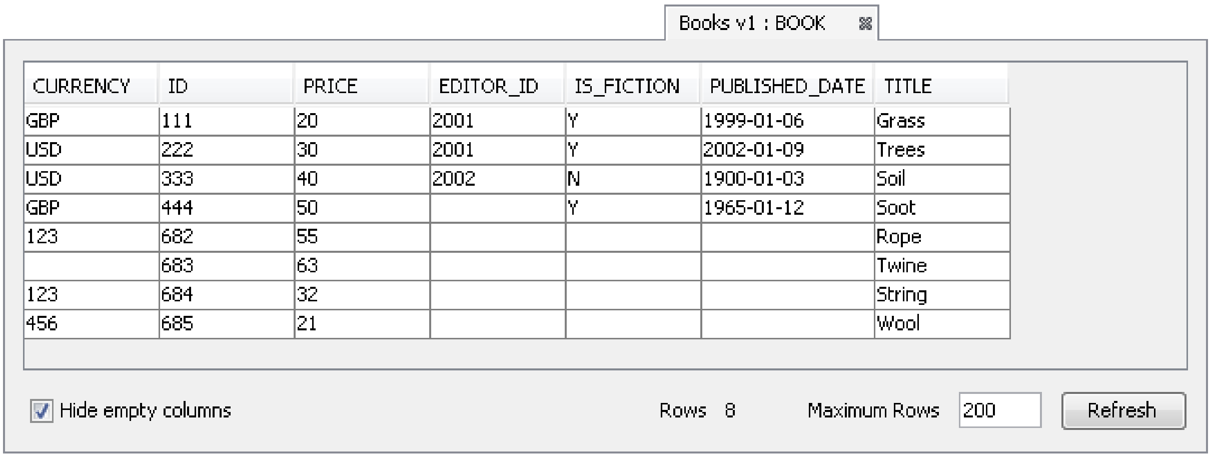
driver org.apache.derby.jdbc.EmbeddedDriver

sqlinit

6) Move your cursor over the **BOOK** 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 BOOK data as shown below.



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
123	682	55				Rope
	683	63				Twine
123	684	32				String
456	685	21				Wool

☑ Hide empty columns Rows 8 Maximum Rows 200

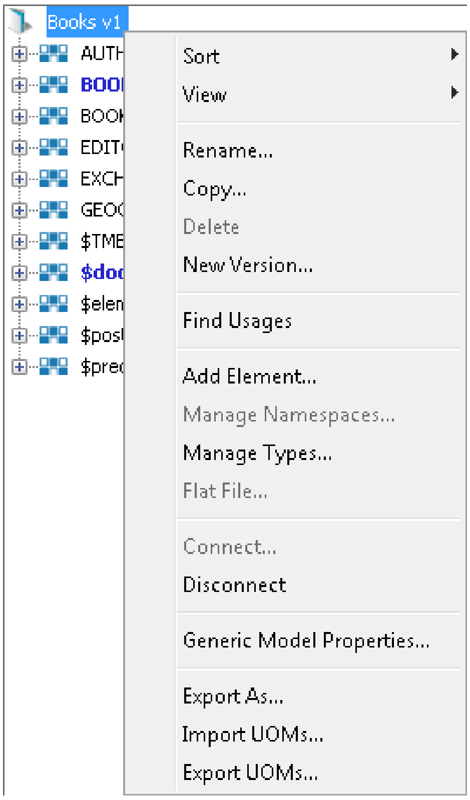
8) We can see above that the book called Twine has a null currency value. This of course occurred because the original currency code was 890 and failed our condition. This error should have been logged in our log file.

9) The record below is taken from our log file BooksLog.log. We can see that the failed record was logged as expected with the event being captured.

```
19-Jun-2012 16:16:05 - INFO - Currency code greater than 500 for book titled Twine
```

10) Let's close the data pane and disconnect from the data store. Click on the ☒ icon to close the data view pane.

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.

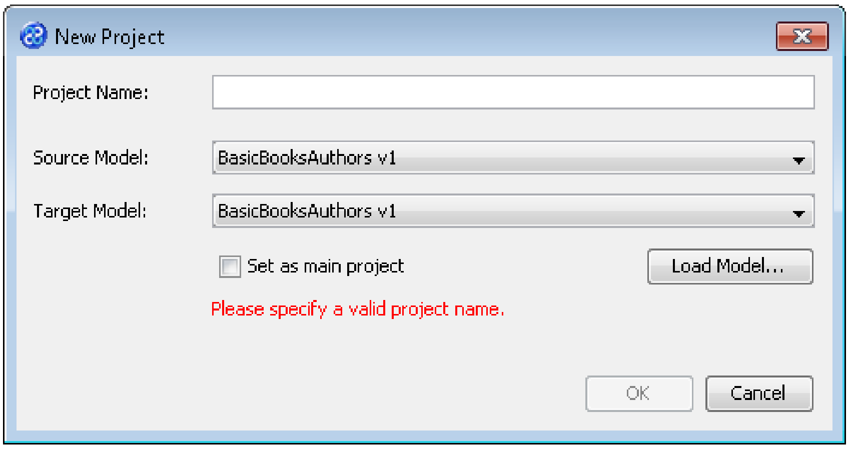
Exercise 9 - Create the LoggingB Project

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.


New Project

Project Name:

Source Model: BasicBooksAuthors v1

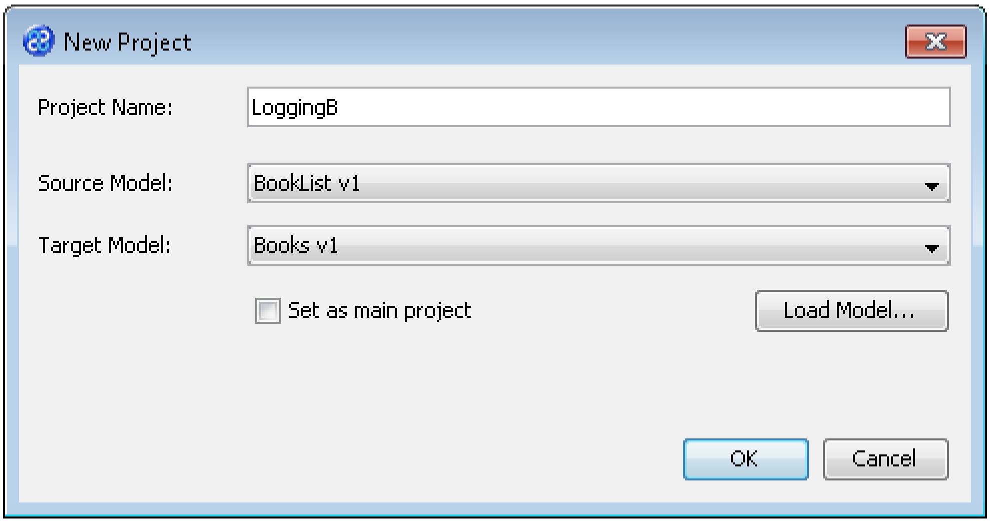
Target Model: BasicBooksAuthors v1

□ Set as main project

Please specify a valid project name.

4) In the **Project Name** field we will provide a name for the project. Let's call our project **LoggingB** 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**.



Project Name: LoggingB

Source Model: BookList v1

Target Model: Books v1

□ Set as main project

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

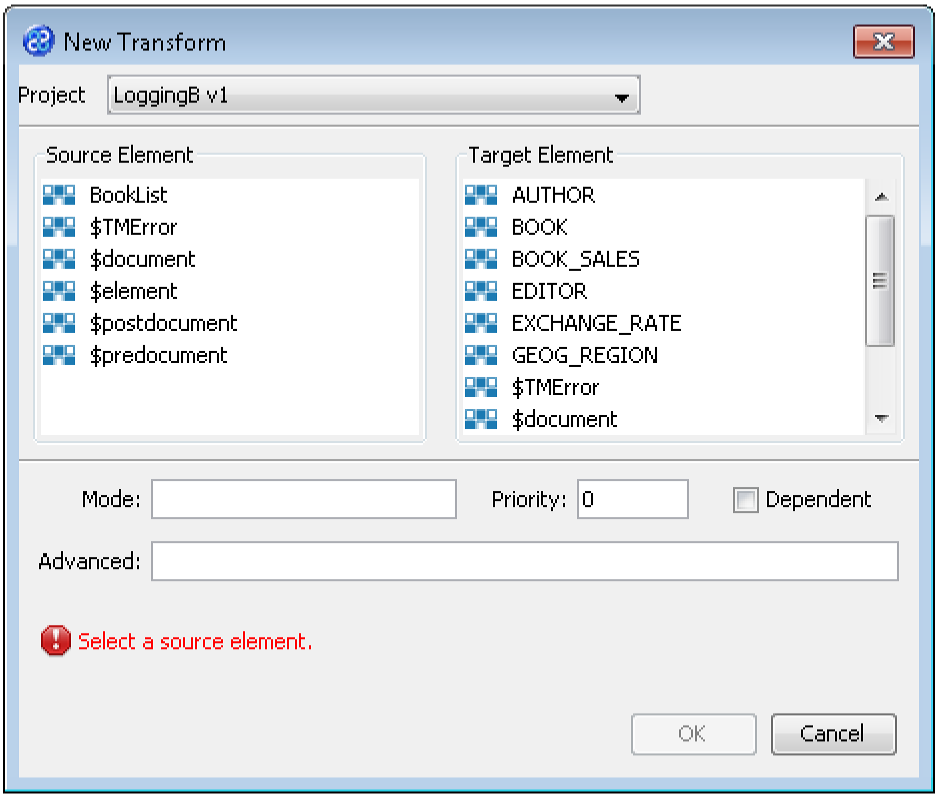
Exercise 10 - Create the Independent Transform for the LoggingB Project

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 **LoggingB v1**.



Project LoggingB v1

Mode:

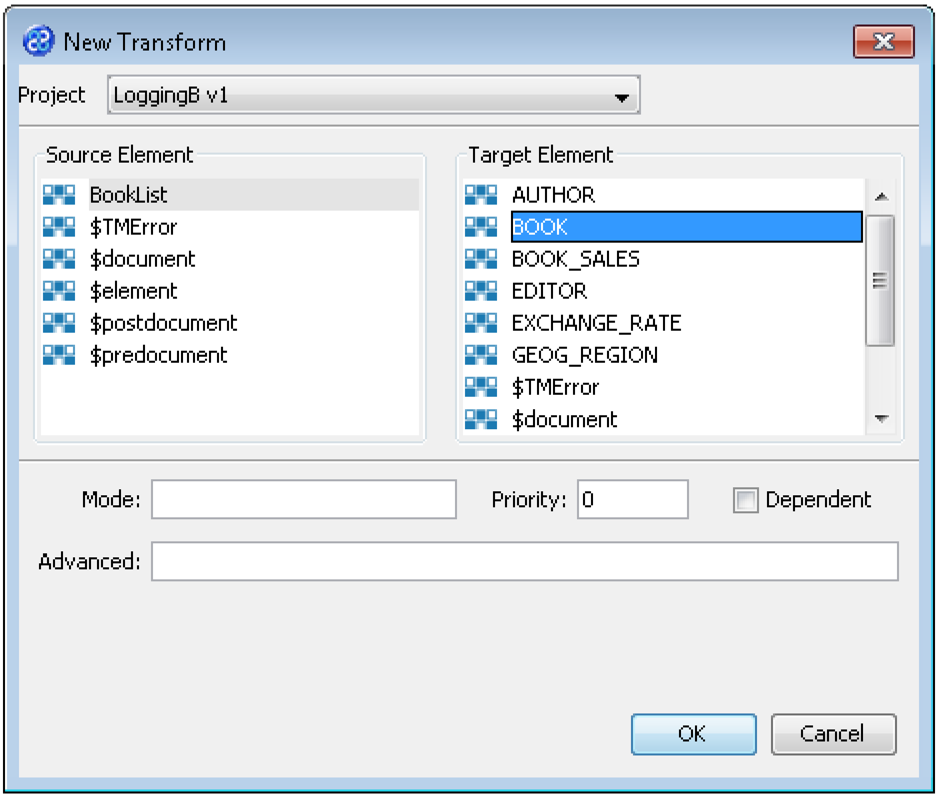
Priority: 0

□ Dependent

Advanced:

Select a source element.

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.



New Transform

Project: LoggingB v1

Source Element
- BookList
- $TMError
- $document
- $element
- $postdocument
- $predocument

Target Element
- AUTHOR
- BOOK
- BOOK_SALES
- EDITOR
- EXCHANGE_RATE
- GEOG_REGION
- $TMError
- $document

Mode:

Priority: 0

□ Dependent

Advanced:

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 as shown below.

```
TITLE := title;
<ID> := <id>;
PRICE := price;

IF (currency > '500') THEN
LOGINFO('Currency code greater than 500 for book titled '&title,'book');
ELSE
CURRENCY := currency;
END_IF;

iAUTHOR := self;
```

8) Click once on the File menu bar option.

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

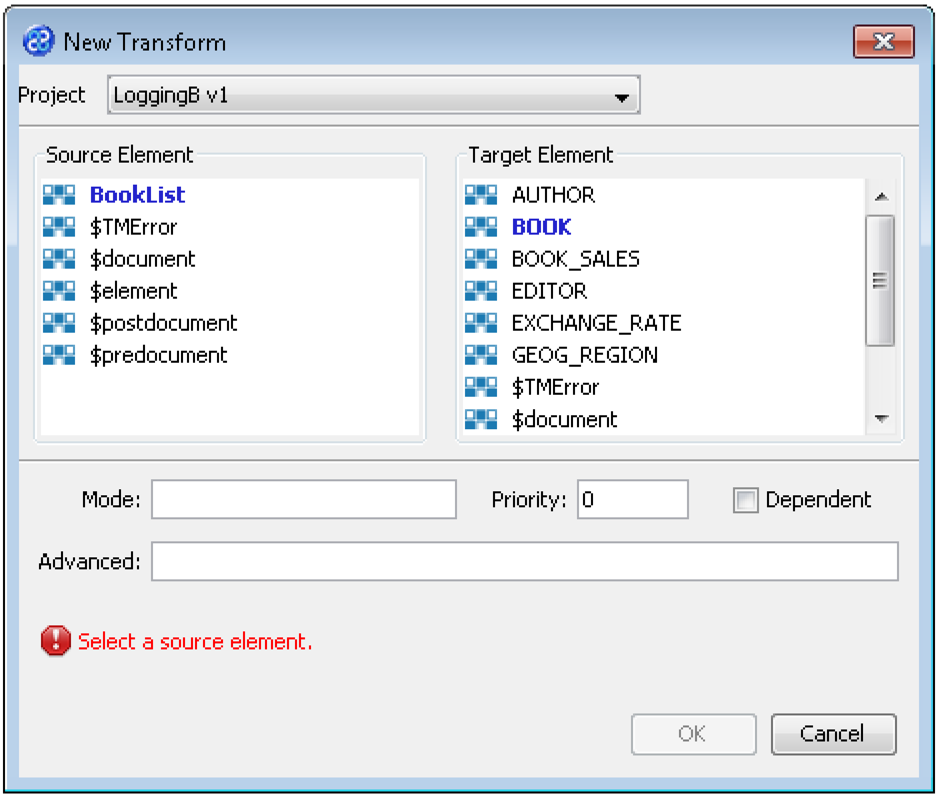
Exercise 11 - Add the Dependent Transform to Update the Author Information

This exercise will 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.

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 **LoggingB v1**.



New Transform

Project: LoggingB v1

Source Element

- BookList
- $TMError
- $document
- $element
- $postdocument
- $predocument

Target Element

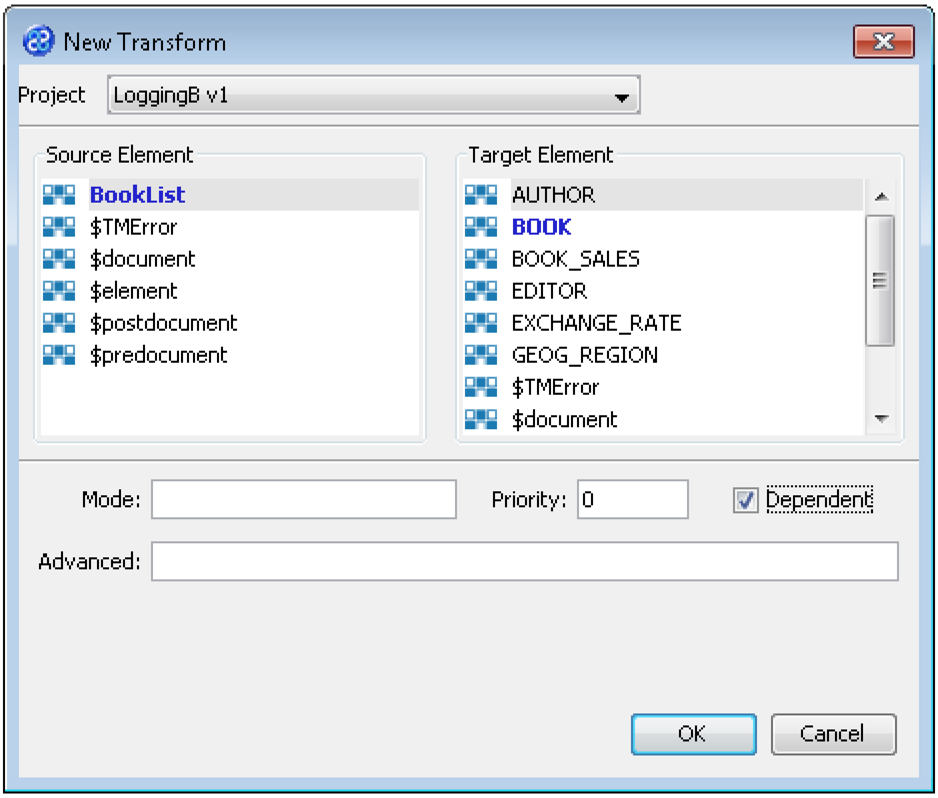
- AUTHOR
- BOOK
- BOOK_SALES
- EDITOR
- EXCHANGE_RATE
- GEOG_REGION
- $TMError
- $document

Mode: Priority: 0 □ Dependent

Advanced:

Select a source element.

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 **AUTHOR**. We must also ensure that the tick box called **Dependent** has a tick. The **New Transform** window will look like the one below.



New Transform

Project: LoggingB v1

Source Element

- BookList
- $TMError
- $document
- $element
- $postdocument
- $predocument

Target Element

- AUTHOR
- BOOK
- BOOK_SALES
- EDITOR
- EXCHANGE_RATE
- GEOG_REGION
- $TMError
- $document

Mode: Priority: 0 ☑ Dependent

Advanced:

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.

```
NAME := <author forename>;
SURNAME := <author surname>;

LOGINFO('Author called '&<author surname>&' moved.','auth');
```

8) Click once on the File menu bar option.

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

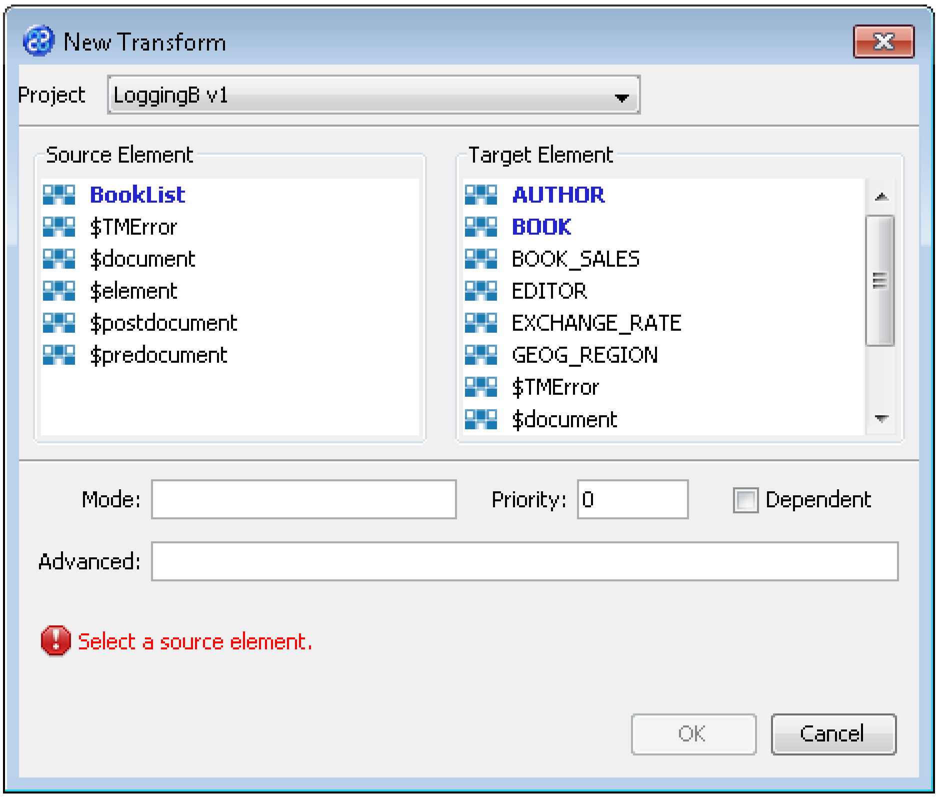
Exercise 12 - Create the $document Transform for the LoggingB Project

In this project we require a $document transform that lets us use a log4j configuration file. The project will use this in order to output our logging file.

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 this case this will be **LoggingB v1**.



New Transform

Project: LoggingB v1

Source Element
- **BookList**
- $TMError
- $document
- $element
- $postdocument
- $predocument

Target Element
- **AUTHOR**
- **BOOK**
- BOOK_SALES
- EDITOR
- EXCHANGE_RATE
- GEOG_REGION
- $TMError
- $document

Mode:

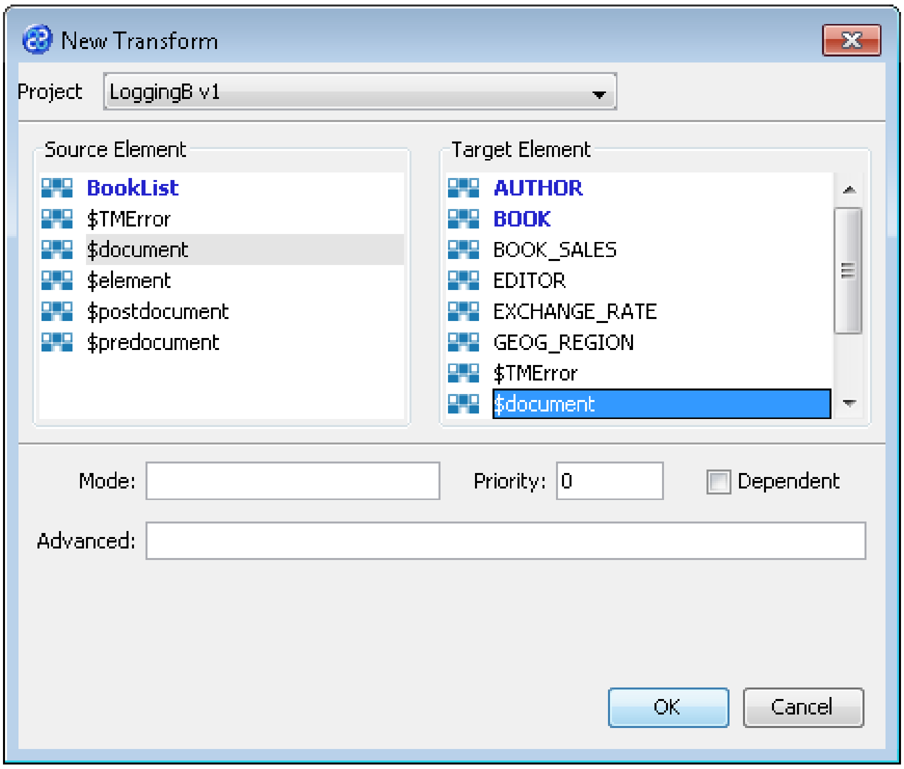
Priority: 0

□ Dependent

Advanced:

Select a source element.

4) With the **New Transform** window open, we will select the elements for the source and target which in this case will be $document for both.



Project: LoggingB v1

Mode:

Priority: 0

□ Dependent

Advanced:

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 as shown below.

```
SETLOGGINGCONFIG('[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Loggin
g\\Config\\B\\log4j-LOGGING-B.xml');
```

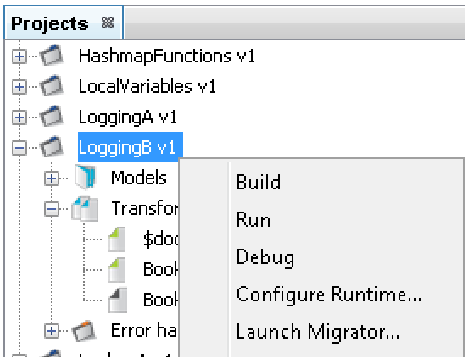
8) Click once on the File menu bar option.

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

Exercise 13 - Build Your LoggingB Project

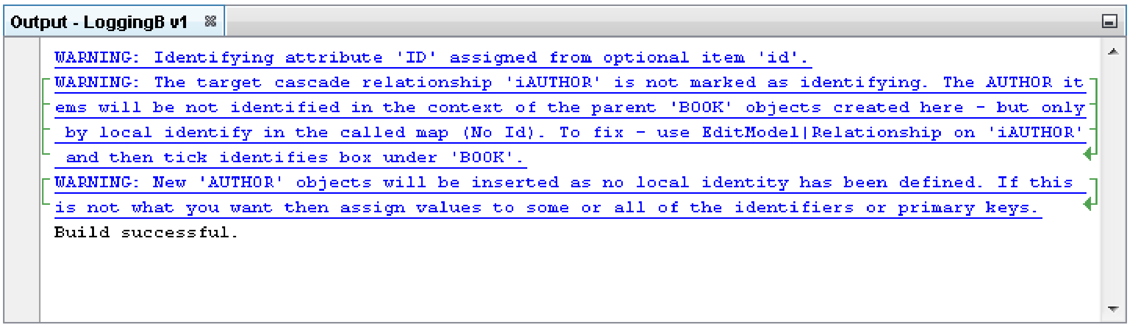
Now we will build our project.

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**.



```
WARNING: Identifying attribute 'ID' assigned from optional item 'id'.
WARNING: The target cascade relationship 'iAUTHOR' is not marked as identifying. The AUTHOR it
ems will be not identified in the context of the parent 'BOOK' objects created here - but only
 by local identify in the called map (No Id). To fix - use EditModel|Relationship on 'iAUTHOR'
 and then tick identifies box under 'BOOK'.
WARNING: New 'AUTHOR' objects will be inserted as no local identity has been defined. If this
is not what you want then assign values to some or all of the identifiers or primary keys.
Build successful.
```

Exercise 14 - Edit the log4j Configuration File

We will need to check and edit our log4j configuration file. The file we supply has two param lines as follows :-

```
<param name="File"
value="C:\\\\[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Log\\B\\A
uthorsLog.log"/>
```

and

```
<param name="File"
value="C:\\\\[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Log\\B\\B
ooksLog.log"/>
```

but for our project we need to edit the `[TMHOME]` sections and provide the actual location of your Transformation Manager home directory. Below you can see our supplied configuration file and an example of what it might look like updated for a windows user after [TMHOME] has been updated.

ETL Template File

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%p - %m%n" />
    </layout>
  </appender>
  <appender name="AUTHORS" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File"
value="C:\\\\[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Log\\B\\A
uthorsLog.log"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{dd-MMM-yyyy HH:mm:ss} - %p - %m%n"
/>
    </layout>
  </appender>
  <appender name="BOOKS" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File"
value="C:\\\\[TMHOME]\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Log\\B\\B
ooksLog.log"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{dd-MMM-yyyy HH:mm:ss} - %p - %m%n"
/>
    </layout>
  </appender>
  <logger name="auth">
    <level value="INFO"/>
    <appender-ref ref="AUTHORS" />
  </logger>
  <logger name="book">
    <level value="INFO"/>
    <appender-ref ref="BOOKS" />
  </logger>
  <root>
    <level value="INFO" />
    <appender-ref ref="CONSOLE" />
  </root>
</log4j:configuration>
```

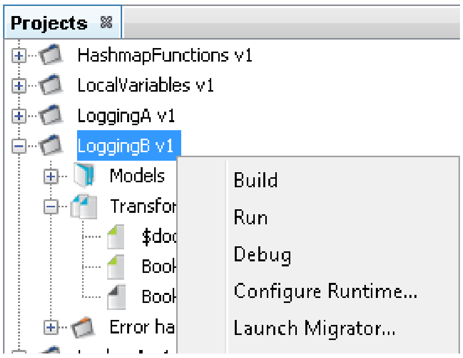
Your Edited File

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="CONSOLE" class="org.apache.log4j.ConsoleAppender">
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%p - %m%n" />
    </layout>
  </appender>
  <appender name="AUTHORS" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File"
value="C:\\\\Users\\John\\TM\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Lo
g\\B\\AuthorsLog.log" />
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{dd-MMM-yyyy HH:mm:ss} - %p - %m%n"
/>
    </layout>
  </appender>
  <appender name="BOOKS" class="org.apache.log4j.DailyRollingFileAppender">
    <param name="File"
value="C:\\\\Users\\John\\TM\\Tutorials\\Source_and_Target\\Tutorial_15_Logging\\Lo
g\\B\\BooksLog.log" />
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{dd-MMM-yyyy HH:mm:ss} - %p - %m%n"
/>
    </layout>
  </appender>
  <logger name="auth">
    <level value="INFO"/>
    <appender-ref ref="AUTHORS" />
  </logger>
  <logger name="book">
    <level value="INFO"/>
    <appender-ref ref="BOOKS" />
  </logger>
  <root>
    <level value="INFO" />
    <appender-ref ref="CONSOLE" />
  </root>
</log4j:configuration>
```

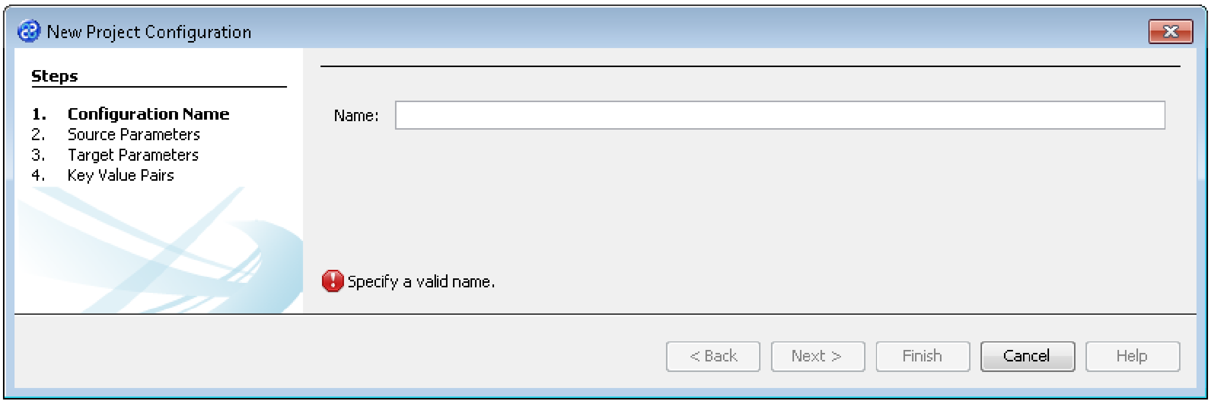
Exercise 15 - Run the Project

Now let's Run our transformation to see what results are produced. We have told the main transform to LOGINFO and provided a $document transform specifying the location of where the log file is.

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.



Specify a valid name.

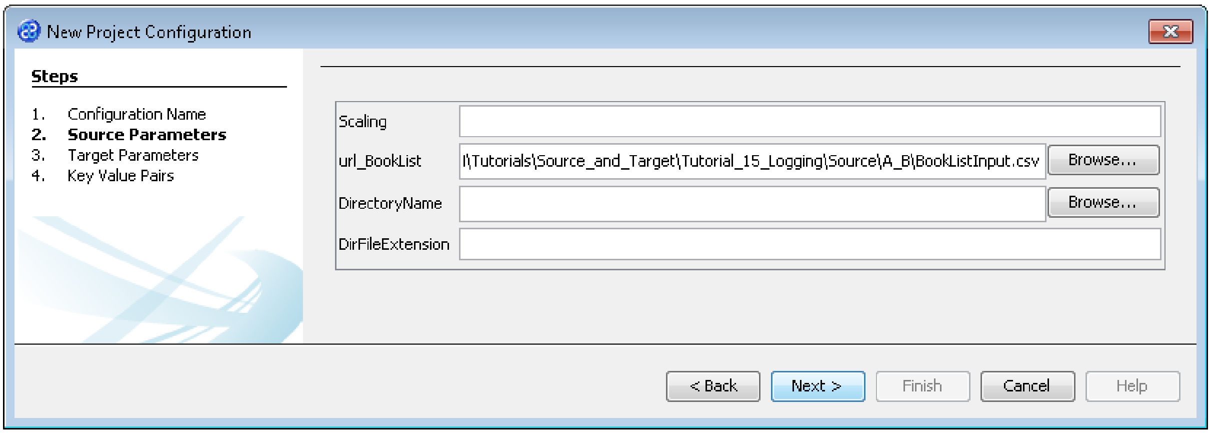
3) Let's give this configuration the name Config1. Type **Config1** into the **Name** field.

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_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_15_Logging\Source\A_B\BookListInput.csv

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



Scaling	
url_BookList	I\Tutorials\Source_and_Target\Tutorial_15_Logging\Source\A_B\BookListInput.csv
DirectoryName	
DirFileExtension	

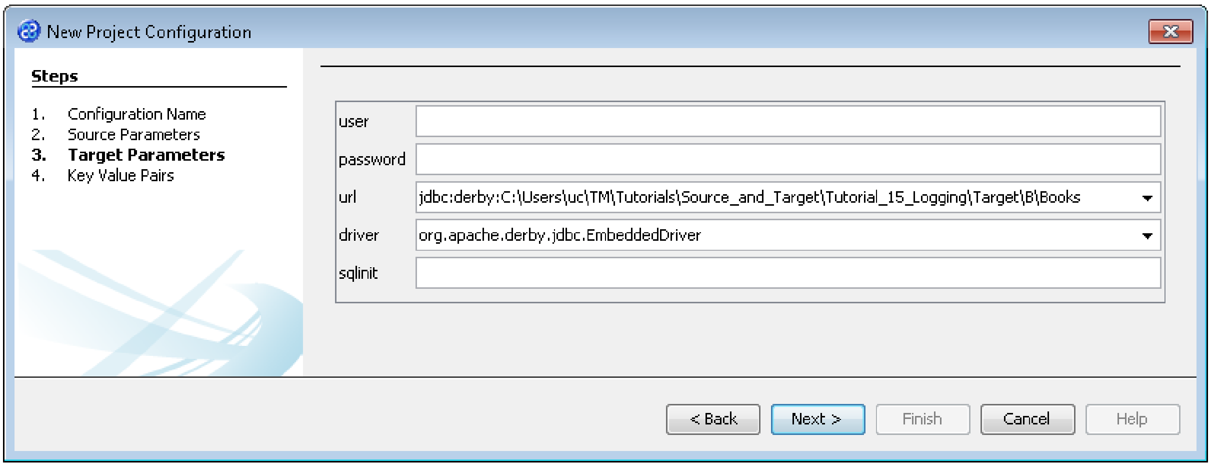
7) Click the button to move to **Step 3. Target Parameters** where we will provide the details for the target data store connection.

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_15_Logging\Target\B\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.



New Project Configuration

Steps

1. Configuration Name
2. Source Parameters
3. **Target Parameters**
4. Key Value Pairs

user

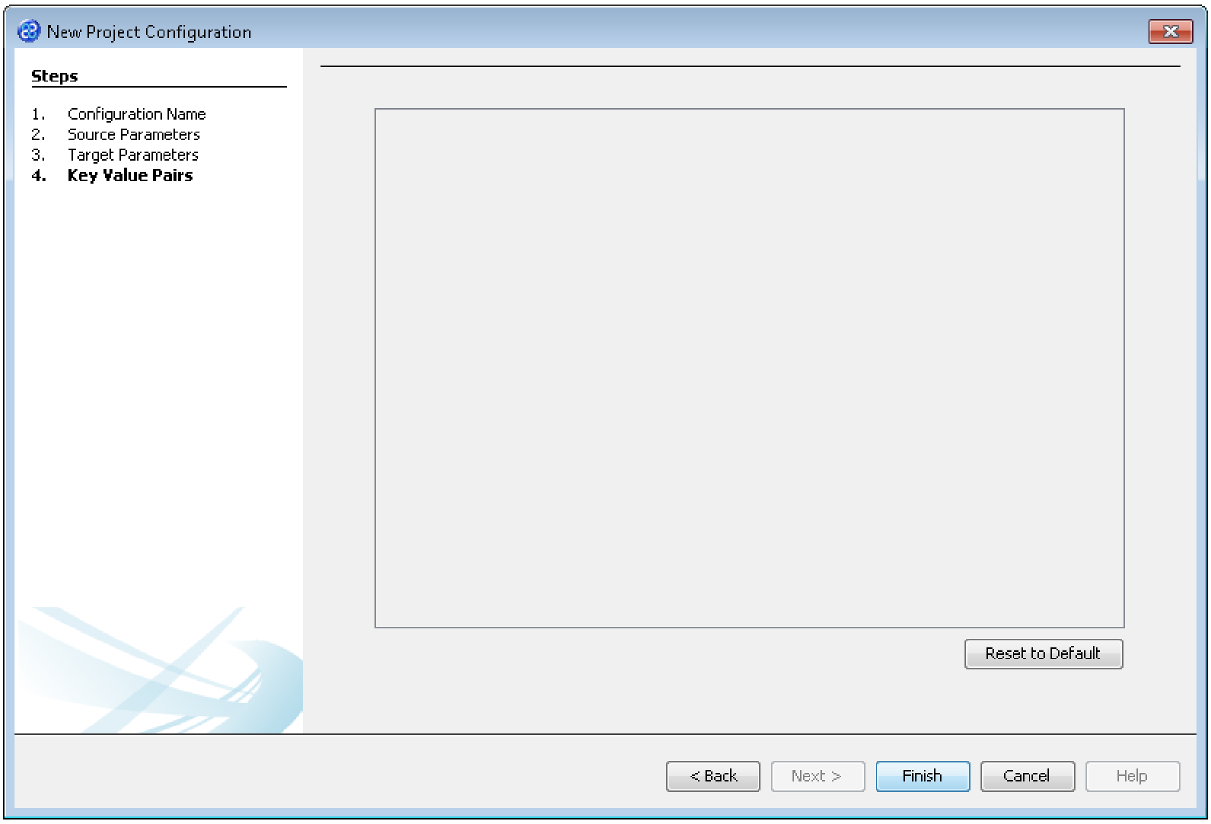
password

url jdbc:derby:C:\Users\uc\TM\Tutorials\Source_and_Target\Tutorial_15_Logging\Target\B\Books

driver org.apache.derby.jdbc.EmbeddedDriver

sqlinit

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.



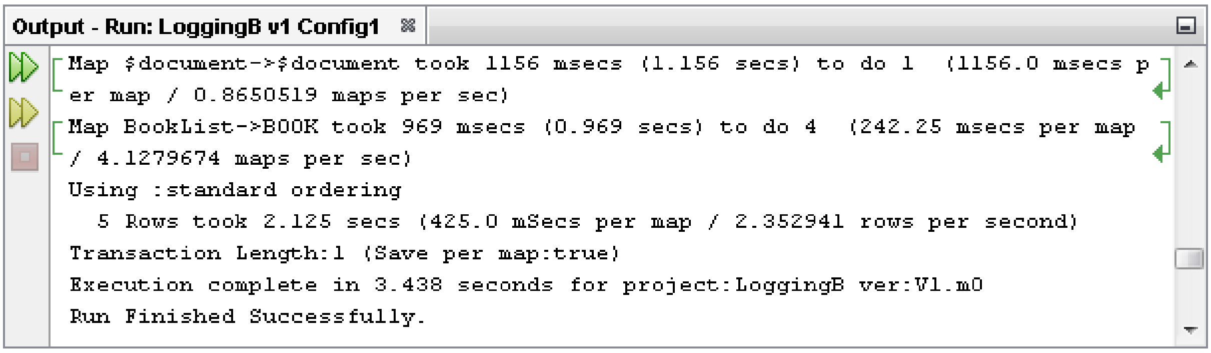
New Project Configuration

Steps

1. Configuration Name
2. Source Parameters
3. Target Parameters
4. **Key Value Pairs**

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.



Output - Run: LoggingB v1 Config1

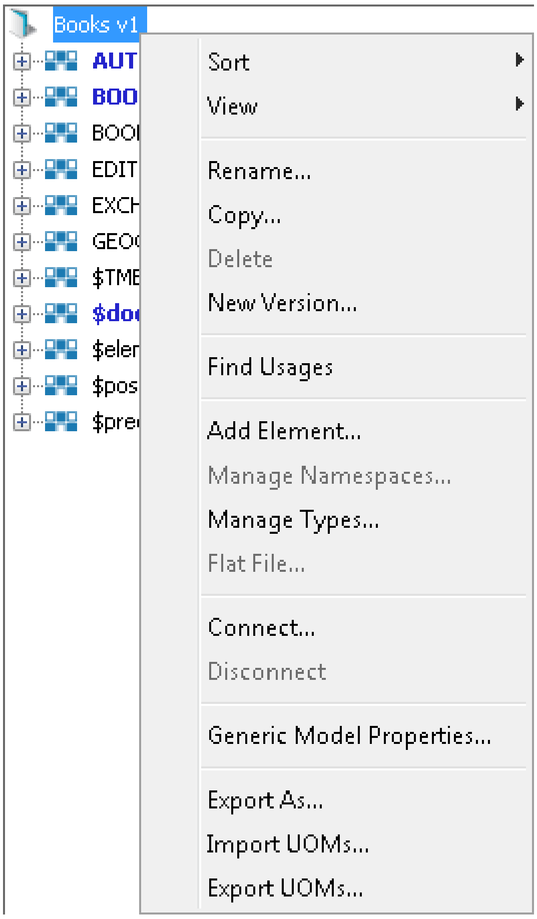
```
Map $document->$document took 1156 msecs (1.156 secs) to do 1  (1156.0 msecs p
er map / 0.8650519 maps per sec)
Map BookList->BOOK took 969 msecs (0.969 secs) to do 4  (242.25 msecs per map
/ 4.1279674 maps per sec)
Using :standard ordering
  5 Rows took 2.125 secs (425.0 mSecs per map / 2.352941 rows per second)
Transaction Length:1 (Save per map:true)
Execution complete in 3.438 seconds for project:LoggingB ver:V1.m0
Run Finished Successfully.
```

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

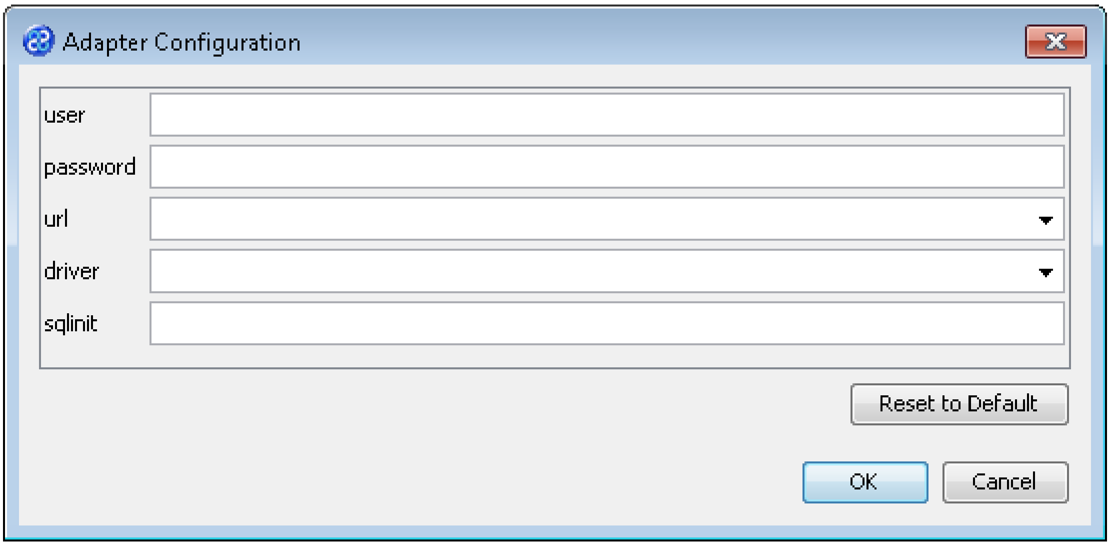
Exercise 16 - View the Target Data

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.

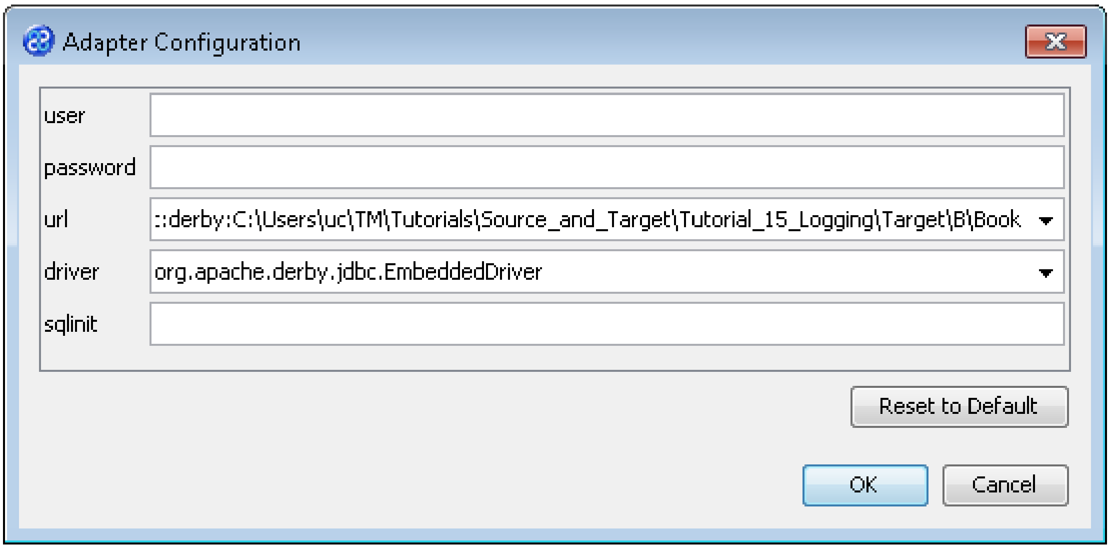


3) Go to the **url** field. You will now need to provide the details of where to go to connect to the source data. View the list of options and select the option called **jdbc:derby:<YOUR_NAME>** from the drop down list. Now replace the <YOUR_NAME> part of the list item, including the angle brackets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

```
[TMHOME]\Tutorials\Source_and_Target\Tutorial_15_Logging\Target\B\Books
```

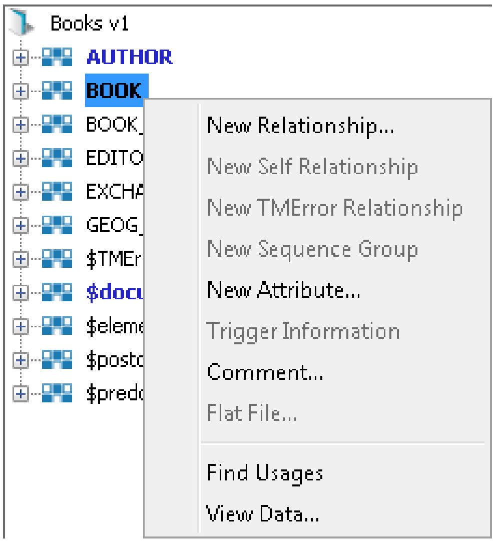
4) Go to the driver field and display the list of options available. Select the option called **org.apache.derby.jdbc.EmbeddedDriver**. The New Project Configuration window will look similar to the image below. **user, password** and **sqlinit** do not require values.

5) The adapter configuration should look similar to the one below.

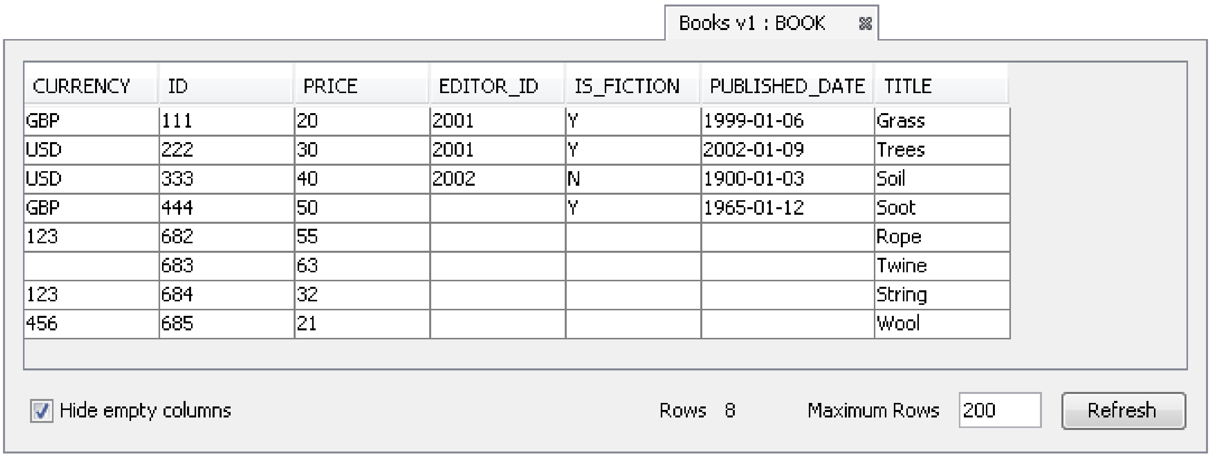


user:
password:
url: ::derby:C:\Users\uc\TM\Tutorials\Source_and_Target\Tutorial_15_Logging\Target\B\Book
driver: org.apache.derby.jdbc.EmbeddedDriver
sqlinit:

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



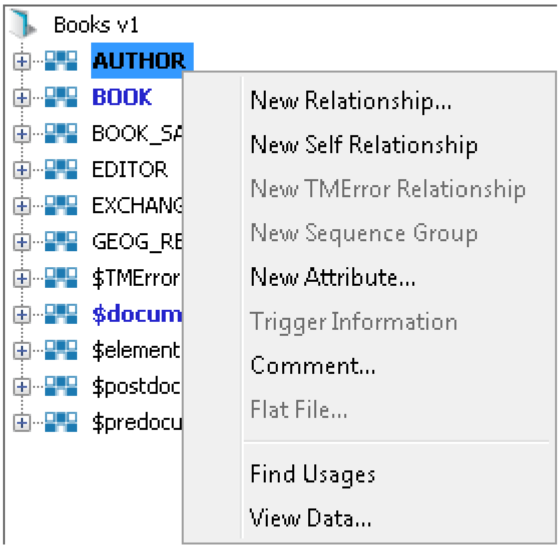
7) Select the View Data... option from the menu. This will open a new pane in the Editor pane displaying the BOOK data as shown below.



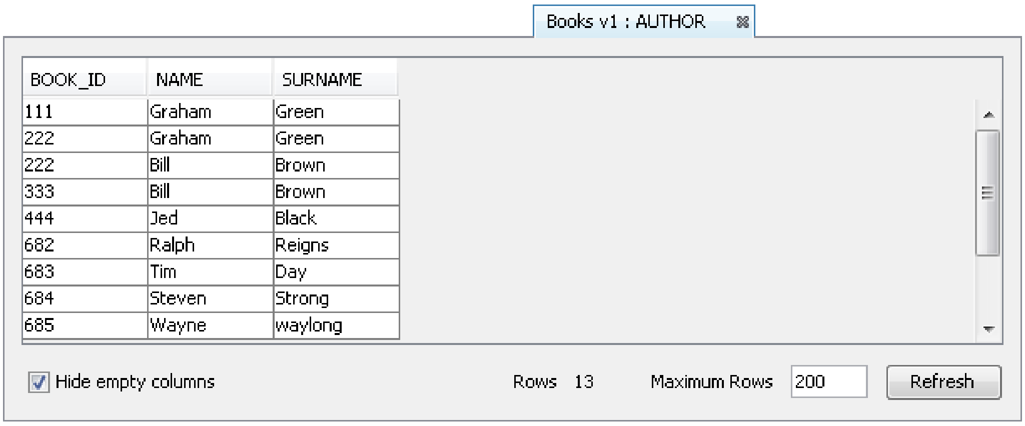
CURRENCY	ID	PRICE	EDITOR_ID	IS_FICTION	PUBLISHED_DATE	TITLE
GBP	111	20	2001	Y	1999-01-06	Grass
USD	222	30	2001	Y	2002-01-09	Trees
USD	333	40	2002	N	1900-01-03	Soil
GBP	444	50		Y	1965-01-12	Soot
123	682	55				Rope
	683	63				Twine
123	684	32				String
456	685	21				Wool

☑ Hide empty columns Rows 8 Maximum Rows 200

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



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



BOOK_ID	NAME	SURNAME
111	Graham	Green
222	Graham	Green
222	Bill	Brown
333	Bill	Brown
444	Jed	Black
682	Ralph	Reigns
683	Tim	Day
684	Steven	Strong
685	Wayne	waylong

☑ Hide empty columns Rows 13 Maximum Rows 200

10) We can see that our transform has executed as expected. Now we should check our two log files.

11) This time our two log files have been populated with the events that have taken place. These are shown below.

BooksLog.log

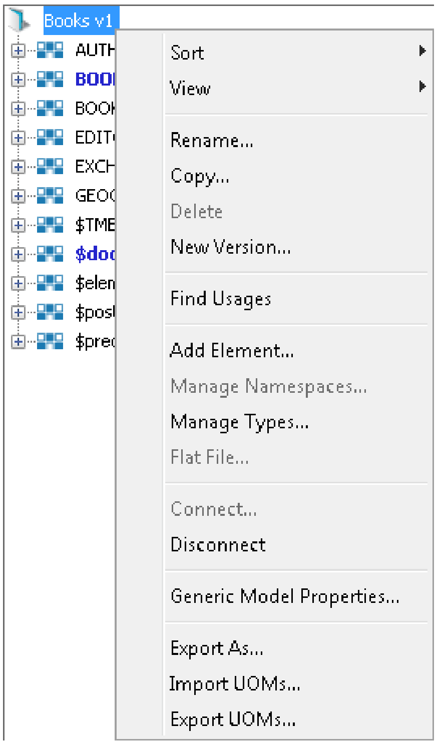
```
19-Jun-2012 16:16:05 - INFO - Currency code greater than 500 for book titled Twine
```

AuthorsLog.log

```
20-Jun-2012 11:43:12 - INFO - Author called Reigns moved.
20-Jun-2012 11:43:12 - INFO - Author called Day moved.
20-Jun-2012 11:43:12 - INFO - Author called Strong moved.
20-Jun-2012 11:43:12 - INFO - Author called waylong moved.
```

12) Let's close the data pane and disconnect from the data store. Click on the ✖ icon to close the data view pane.

13) 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.



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