

Transfer Data from Text File to Database

Creation Time: 2013-03-28

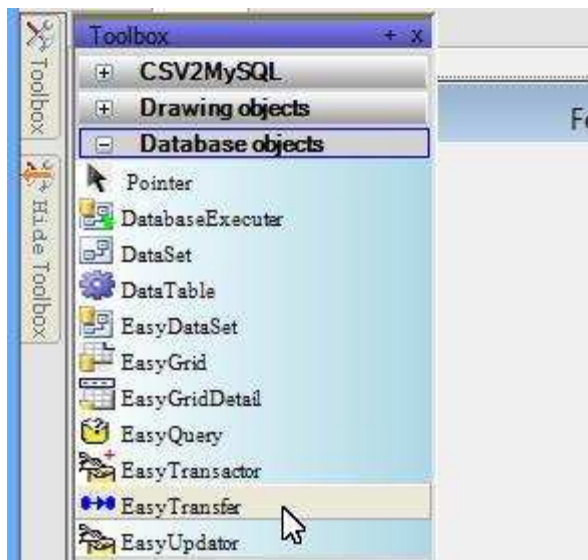
Contents

Introduction	2
Data Source	2
Data Destination	3
Use EasyTransfer in a Windows Form.....	3
Set End Point Type	4
Set Data Source.....	4
Set Destination Type	4
Set text file path of data source.....	4
Include Header	5
Set Delimiter	6
Set Data Destination	6
Set Destination Type	6
Set Database Connection.....	7
Select Table Name	7
Select Fields.....	8
Start data transfer.....	8
Show errors.....	10
Test.....	13
Use EasyTransfer in a Web Page.....	14
Sample Web Page	14
Set data source	15
Set data destination	16
Upload File	17
Execute data transfer	19
Check data transfer errors	21
Create web client method	21

Execute the method.....	26
Test.....	29
Feedback	32

Introduction

EasyTransfer is a component for transferring data from data source to data destination:

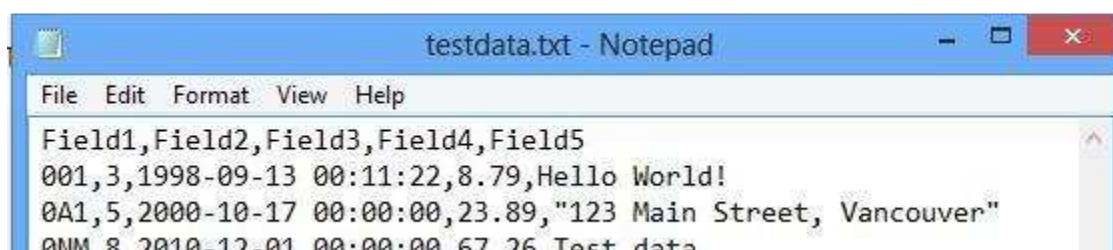


The data source can be database or text file; the data destination can also be database or text file.

This sample shows a process of transferring data from a CSV (comma separated values) text file to a MySQL database.

Data Source

Sample source data are in a text file; each line is for one record; field values are separated by commas; a text value should be double-quoted if it contains commas so that the commas will not be taken as field separators.



In this sample data file, the first line lists the field names.

Data Destination

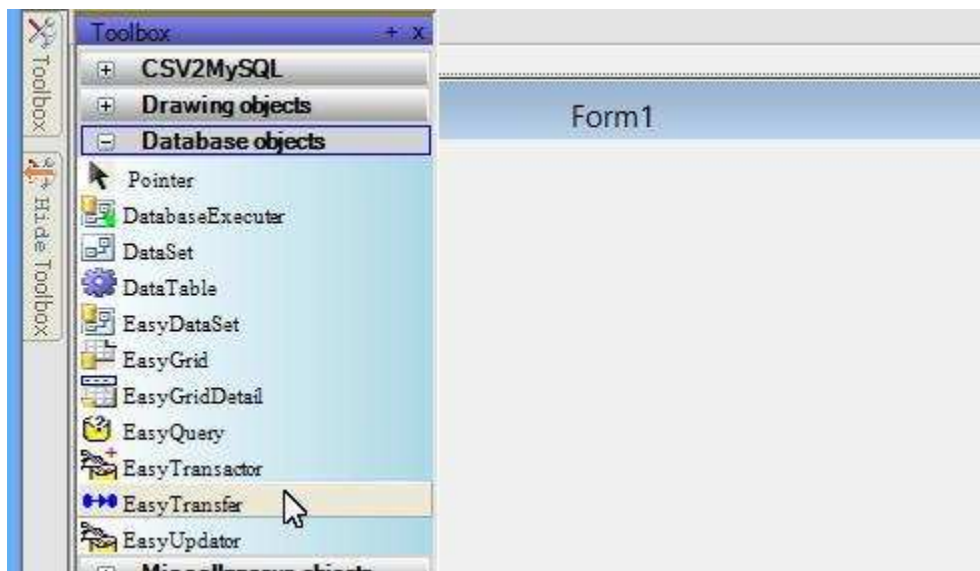
For this sample, we want to import the text file into a table in a MySQL database. The table is defined in the following script:

```
CREATE TABLE testtable (  
    TestID INT NOT NULL AUTO_INCREMENT,  
    Field1 CHAR(60) NOT NULL,  
    Field2 INT NULL,  
    Field3 DATETIME NULL,  
    Field4 DECIMAL(10,2) NULL,  
    Field5 TEXT NULL,  
    PRIMARY KEY (TestID)  
);
```

Note that except the first field every field in the table corresponds to a field in the text file. The first field is an auto-number.

Use EasyTransfer in a Windows Form

Drop an EasyTransfer to a form:

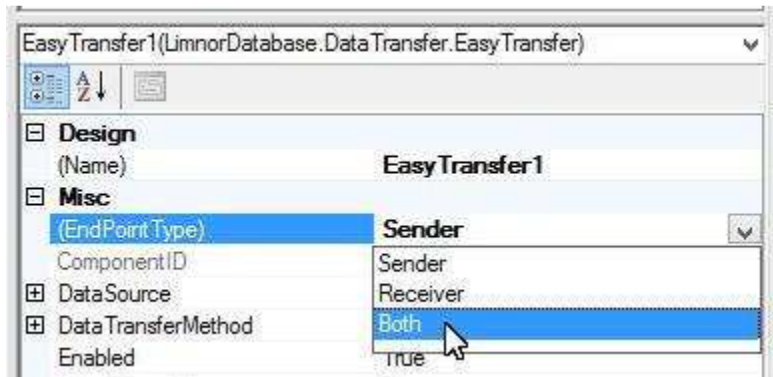


AN instance of EasyTransfer, EasyTransfer1, is created:



Set End Point Type

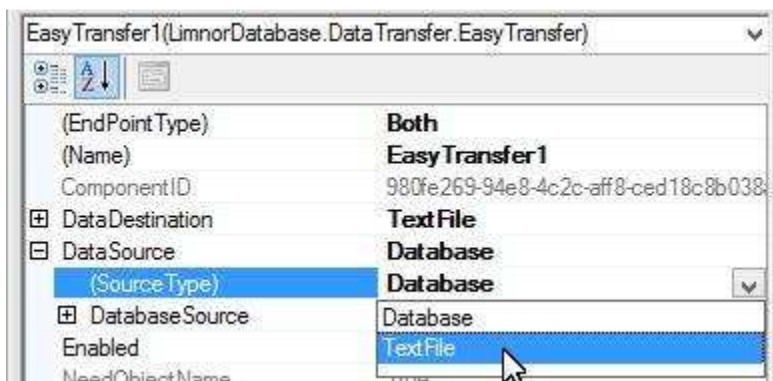
An EasyTransfer can be used as a data source, a data destination, or both. For this sample, we use EasyTransfer1 both as a data source and a data destination:



Set Data Source

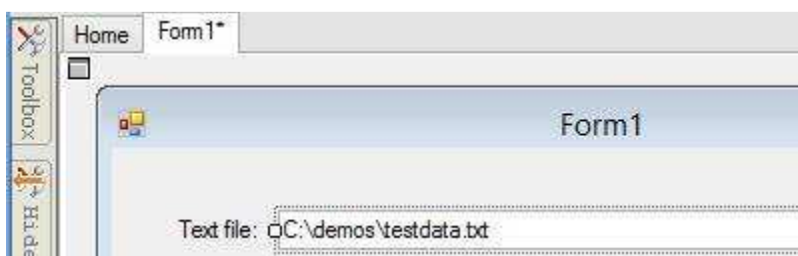
Set Destination Type

For this sample, we want to import data from a text file, so, choose TextFile as the data source type:

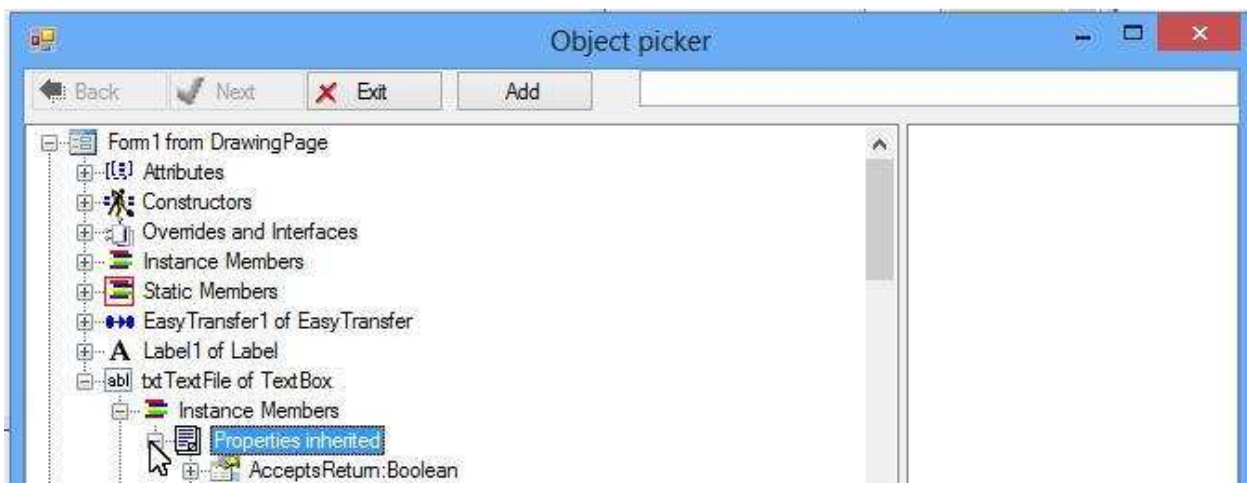
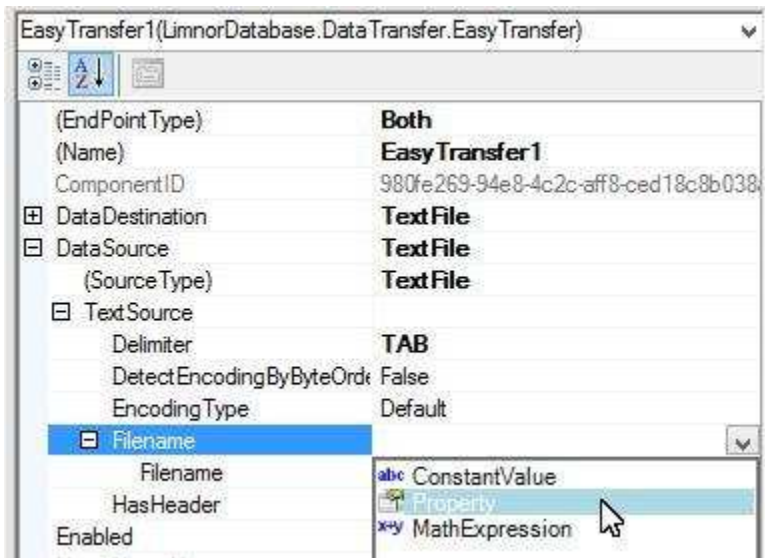


Set text file path of data source

Suppose we use a text box for entering text file path:

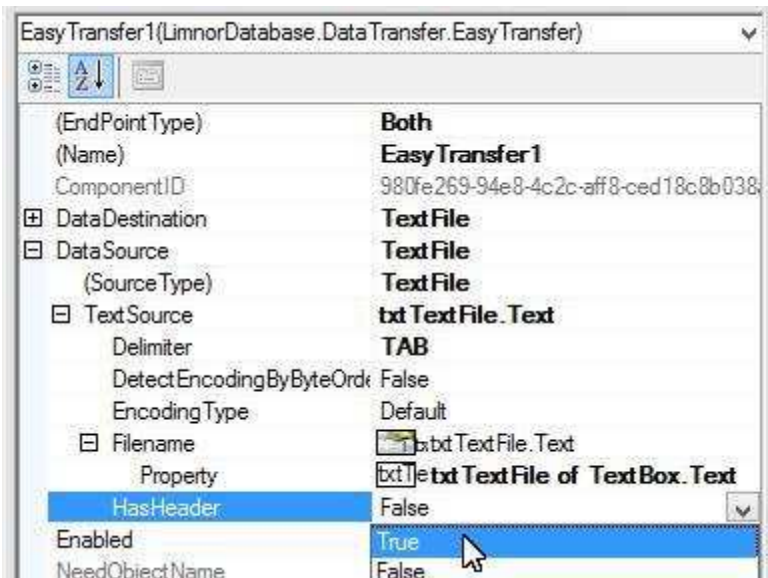


We may set the file path of the data source to the Text property of the text box:



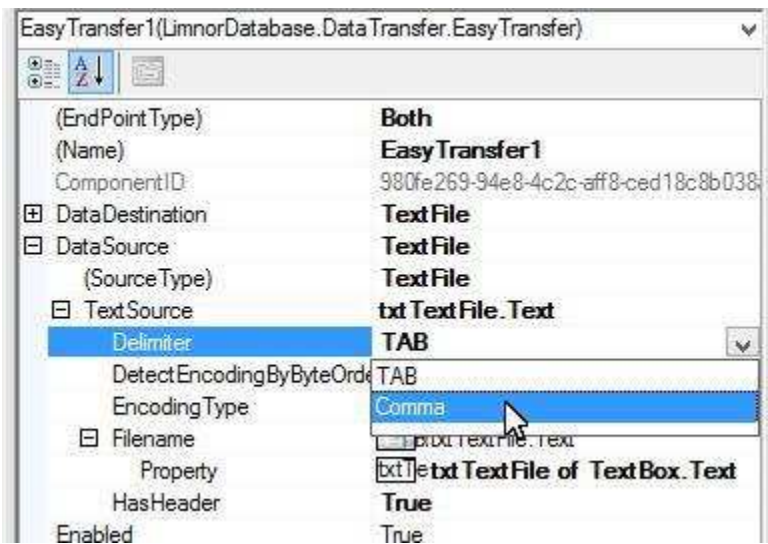
Include Header

Because the first line of our text file is the field names, we need to set HasHead property to True:



Set Delimiter

Because our text file uses comma to separate fields, we need to set Delimiter to Comma:

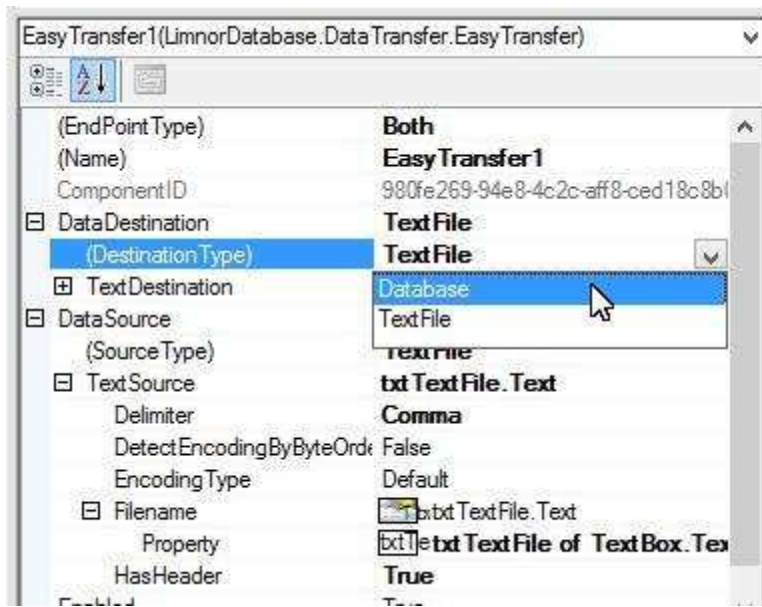


We thus finish setting up data source.

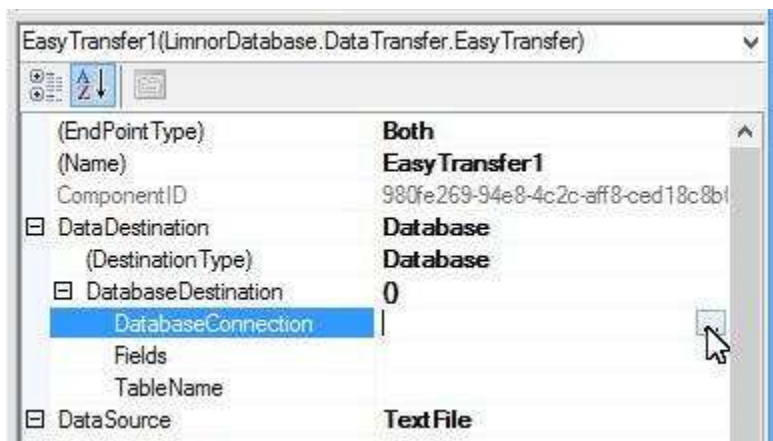
Set Data Destination

Set Destination Type

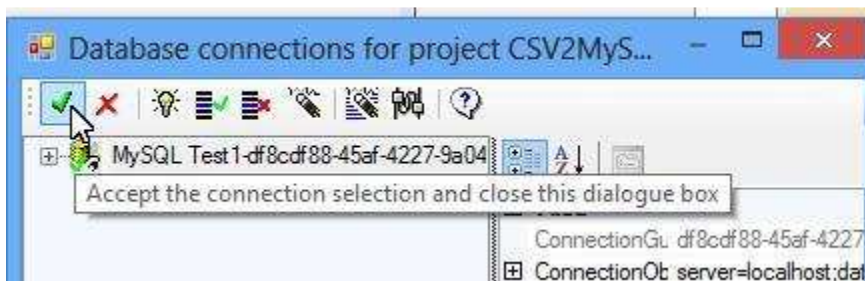
Because we want to import data into a database table, we need to set DestinationType to Database:



Set Database Connection



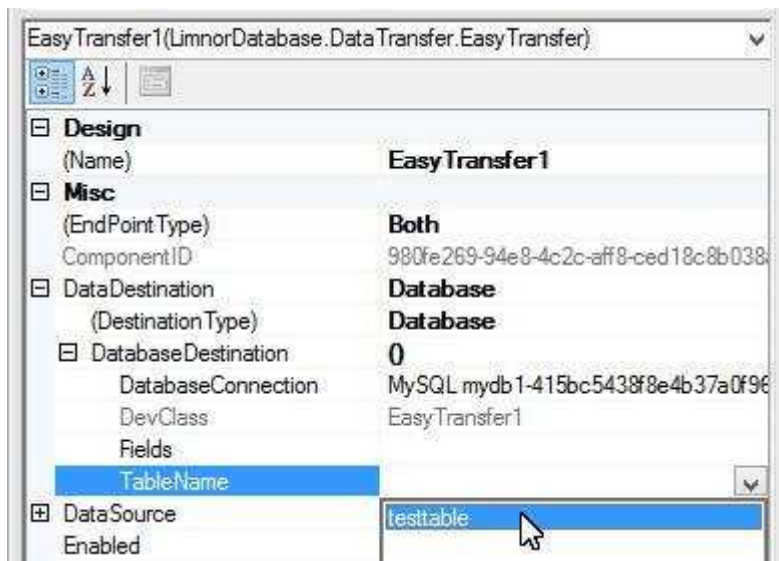
Choose the database connection containing the table we want to import data to:



For details on how to create new database connection or select an existing database connection, see <http://www.limnor.com/support/Limnor%20Studio%20-%20User%20Guide%20-%20Part%20VI.pdf>

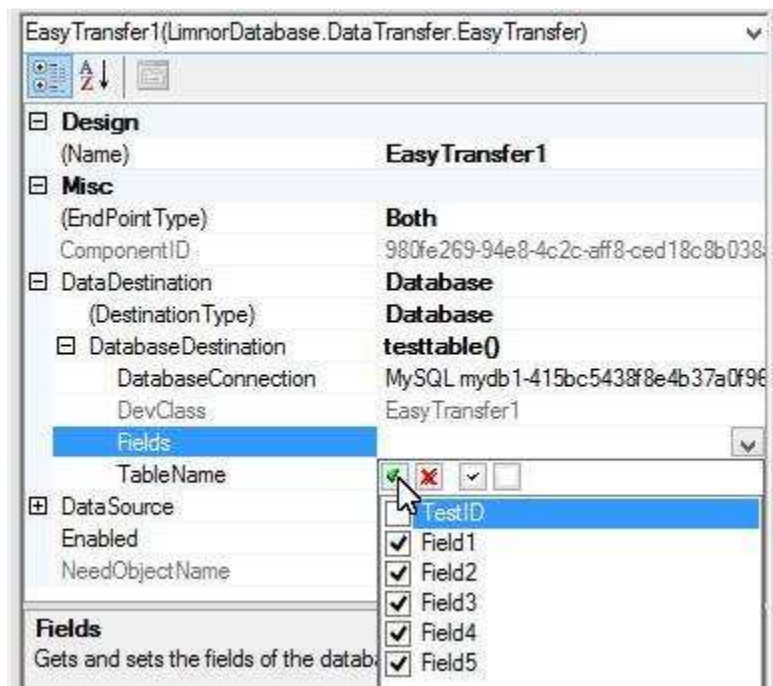
Select Table Name

Select the table for receiving data:



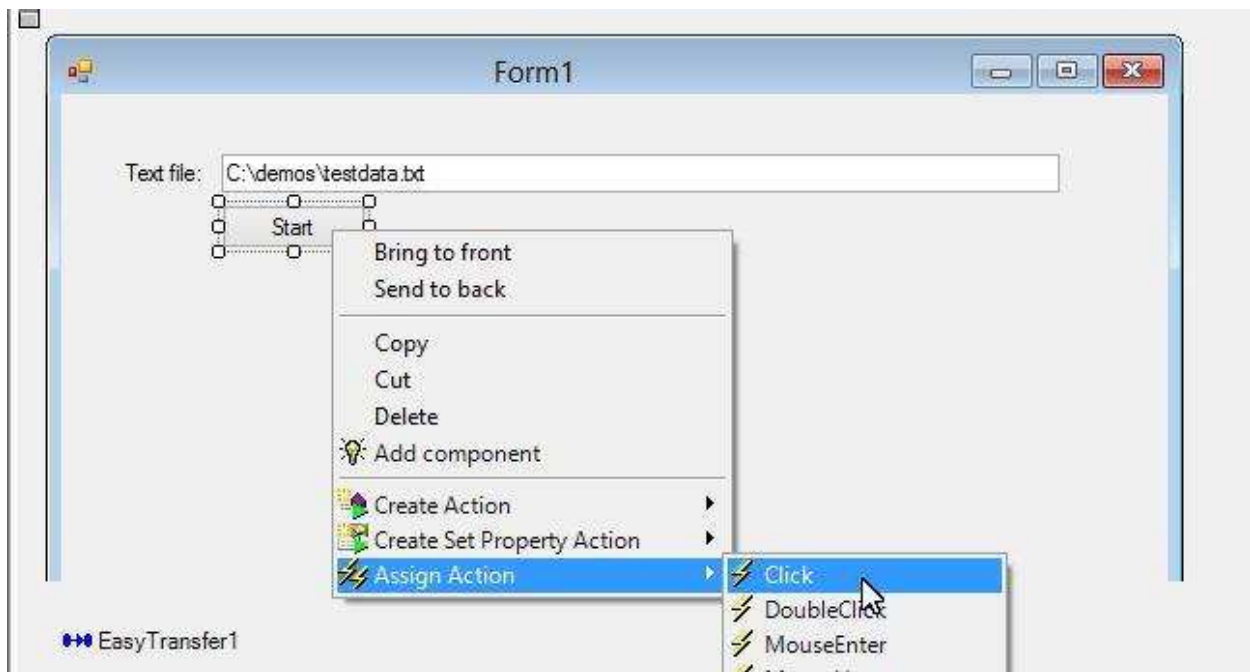
Select Fields

Select which fields should receive data. In this sample, we select all fields except the first field. The field is an auto-number; its value is not included in the text file.

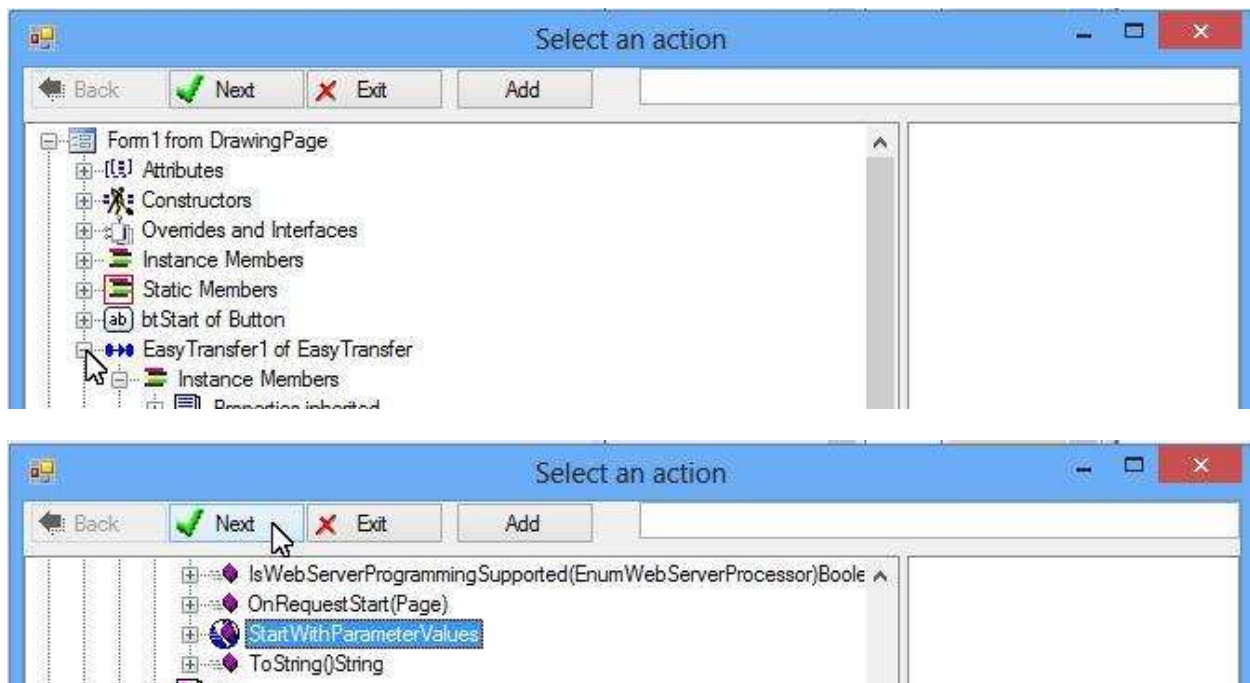


Start data transfer

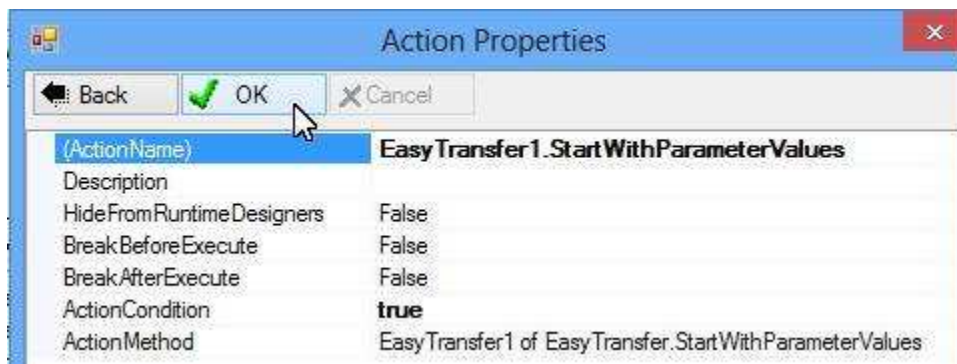
We use a button to trigger data transfer. Right-click the button; choose "Assign Action"; choose "Click" event:



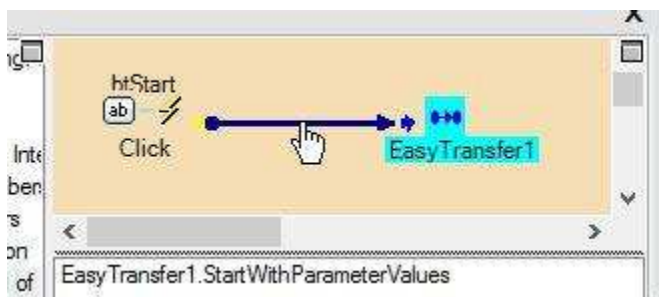
Choose Start method of EasyTransfer1:



Click OK:

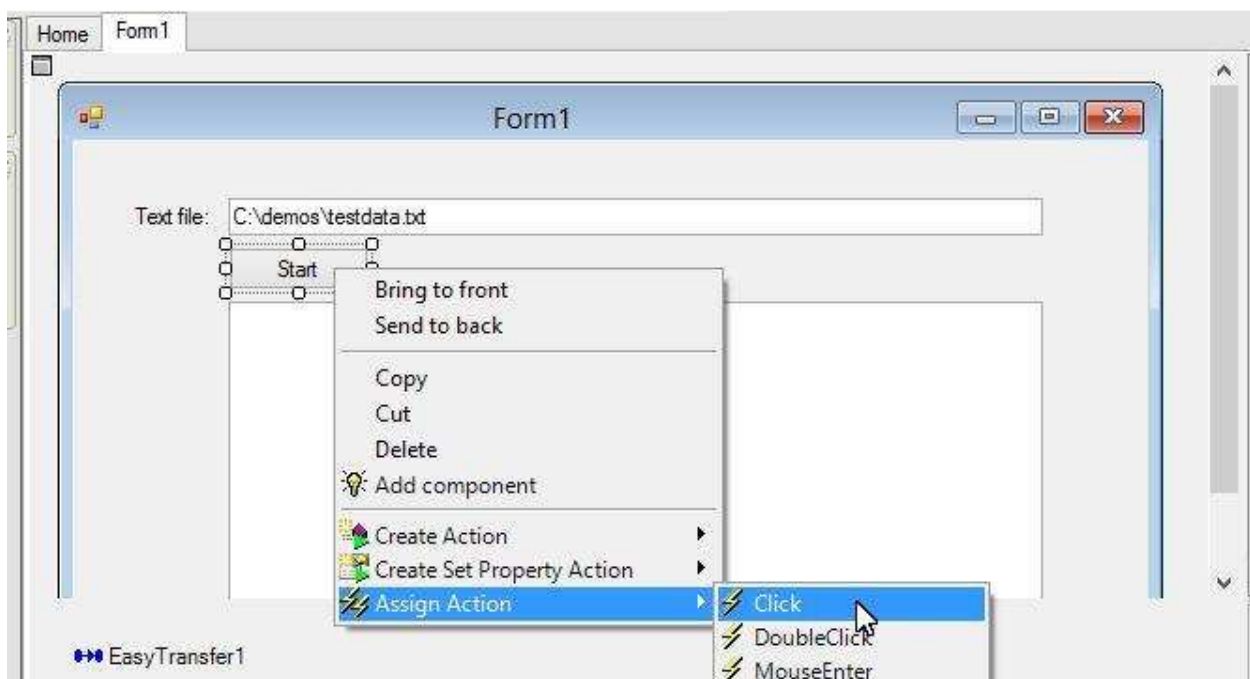


The action is created and assigned to the button:

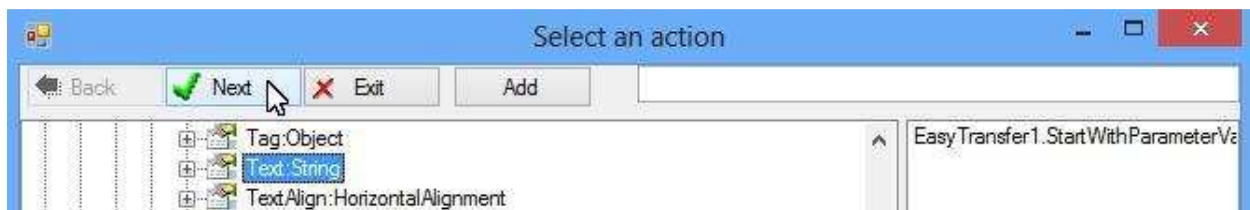
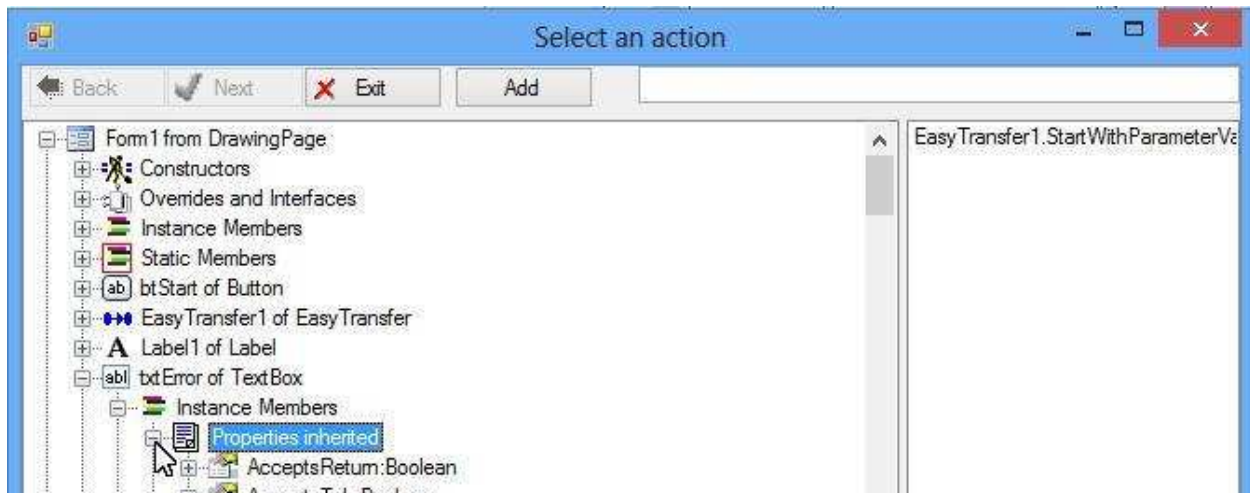


Show errors

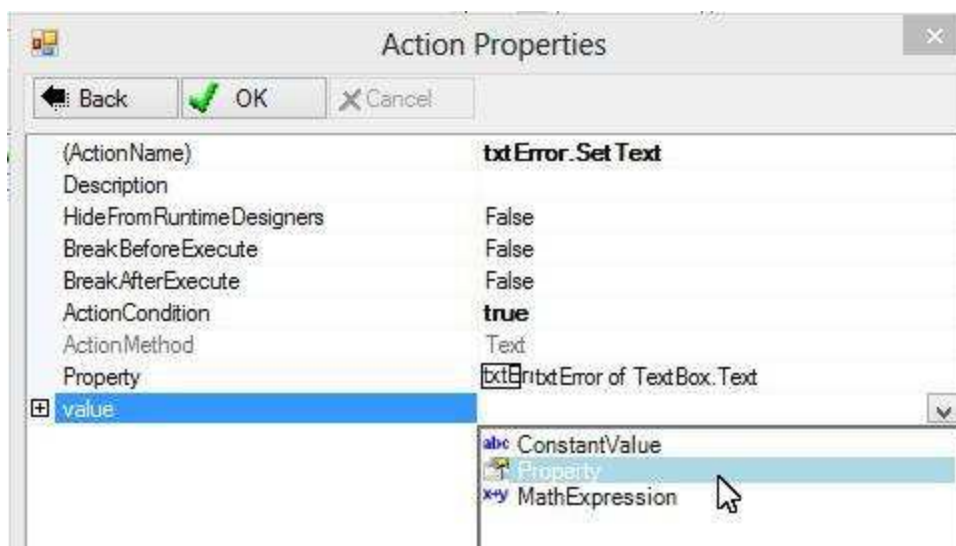
The data transfer may generate errors. We may use a text box to display errors. We may do it after data transfer. Right-click the button; choose "Assign Action"; choose "Click" event:

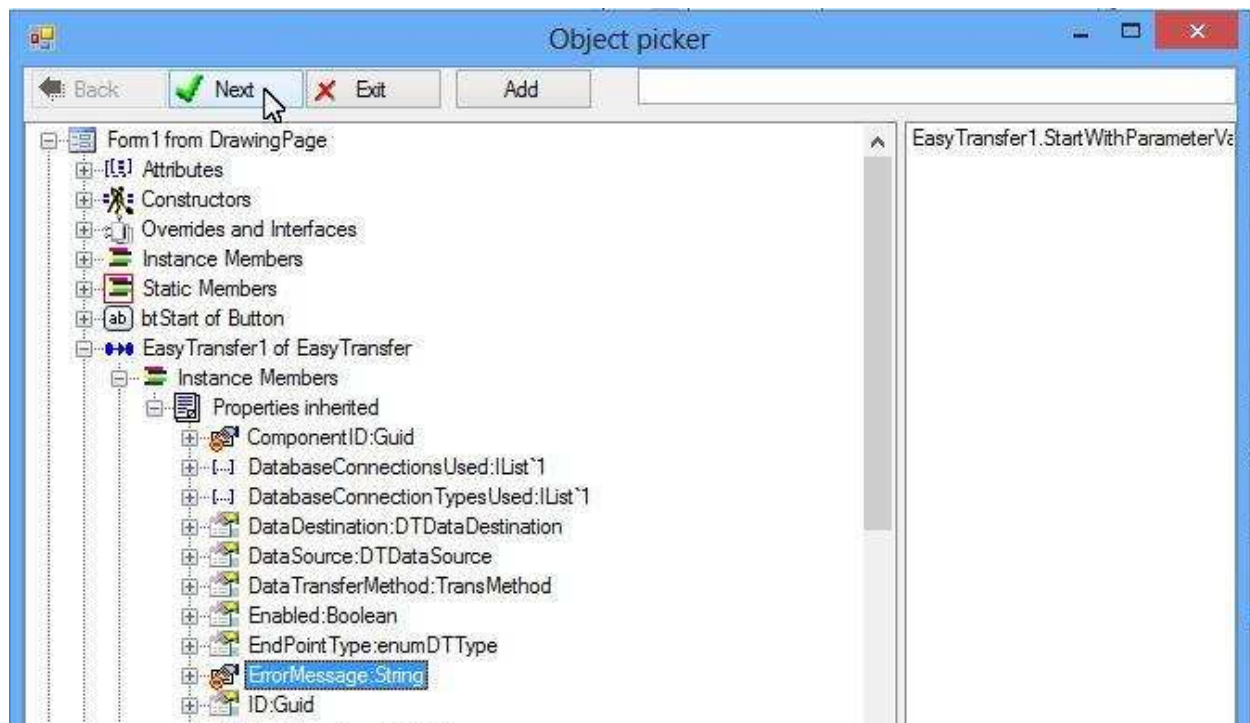


Choose Text property of the text box:

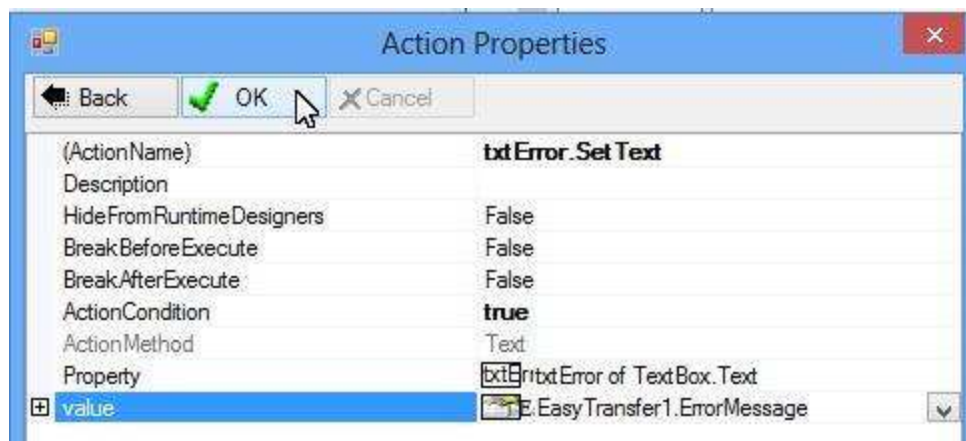


A “Set Property” action is created. We set the “value” to the ErrorMessage property of EasyTransfer1:

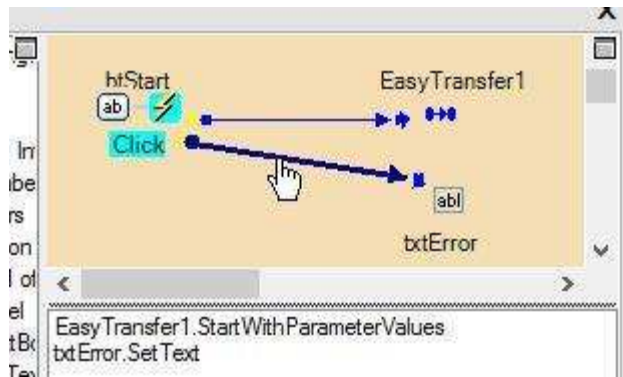




Click OK:



The action is created and assigned to the button:



Test

That is all for our programming. We may test it now. For our test, the table initially is empty:

```

C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql.exe

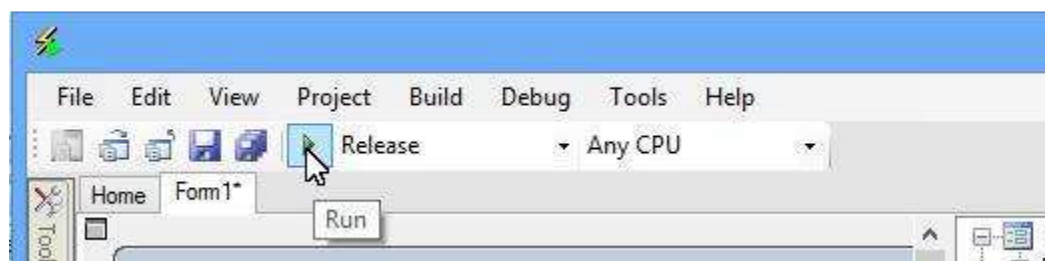
mysql> show columns from testtable;
+----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+----+-----+-----+-----+-----+-----+
| TestID | int(11)       | NO   | PRI | NULL    | auto_increment |
| Field1 | char(60)      | NO   |     | NULL    |                |
| Field2 | int(11)       | YES  |     | NULL    |                |
| Field3 | datetime     | YES  |     | NULL    |                |
| Field4 | decimal(10,2) | YES  |     | NULL    |                |
| Field5 | text          | YES  |     | NULL    |                |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.03 sec)

mysql> select * from testtable;
Empty set (0.00 sec)

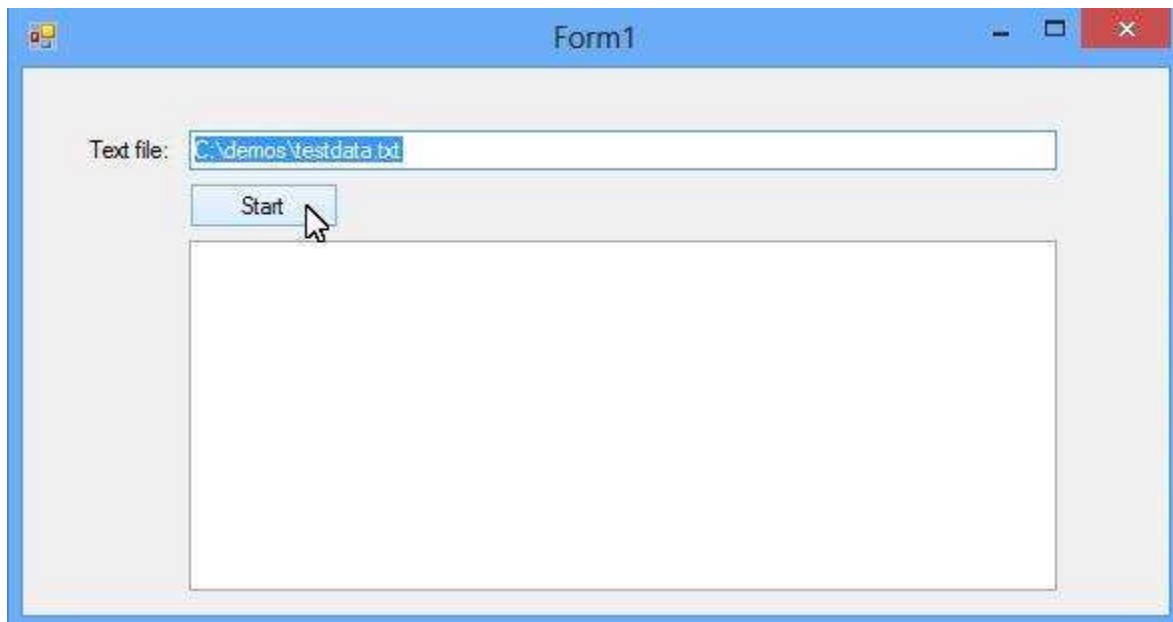
mysql>

```

Start the application from within the IDE:



Click Start button:



If we do not see error message in the text box then the data transfer went through successfully. We may check the table:

```
mysql> show columns from testtable;
+----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra          |
+----+-----+-----+-----+-----+-----+
| TestID | int(11)   | NO   | PRI | NULL    | auto_increment |
| Field1 | char(60)  | NO   |     | NULL    |                |
| Field2 | int(11)   | YES  |     | NULL    |                |
| Field3 | datetime | YES  |     | NULL    |                |
| Field4 | decimal(10,2) | YES |     | NULL    |                |
| Field5 | text      | YES  |     | NULL    |                |
+----+-----+-----+-----+-----+-----+
6 rows in set (0.03 sec)

mysql> select * from testtable;
Empty set (0.00 sec)

mysql> select * from testtable;
+----+-----+-----+-----+-----+-----+
| TestID | Field1 | Field2 | Field3 | Field4 | Field5 |
+----+-----+-----+-----+-----+-----+
| 1 | 001 | 3 | 1998-09-13 00:11:22 | 8.79 | Hello World! |
| 2 | 001 | 5 | 2000-10-17 00:00:00 | 23.89 | 123 Main Street, Vancouver |
| 3 | 0NM | 8 | 2010-12-01 00:00:00 | 67.26 | Test data |
+----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

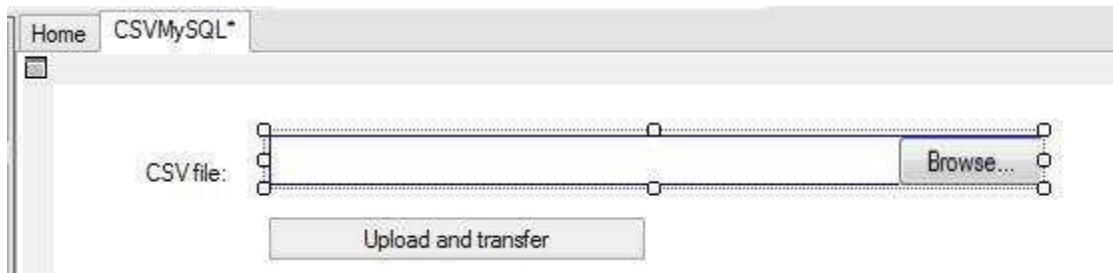
We can see that the data in the text file are transferred in to the table in the MySQL database.

Use EasyTransfer in a Web Page

EasyTransfer can also be used in web pages. We show such use in a PHP project. We use the same data source text file and data destination MySQL table as we used in the previous Windows Form sample.

Sample Web Page

We use an HtmlFileUpload component for uploading a text file as data source.

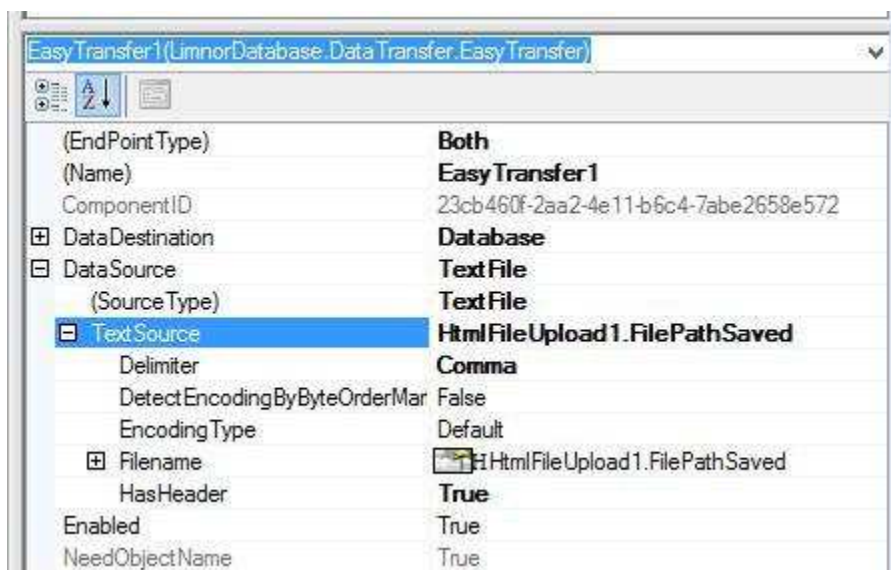


Use an HtmlTextArea to show process errors:

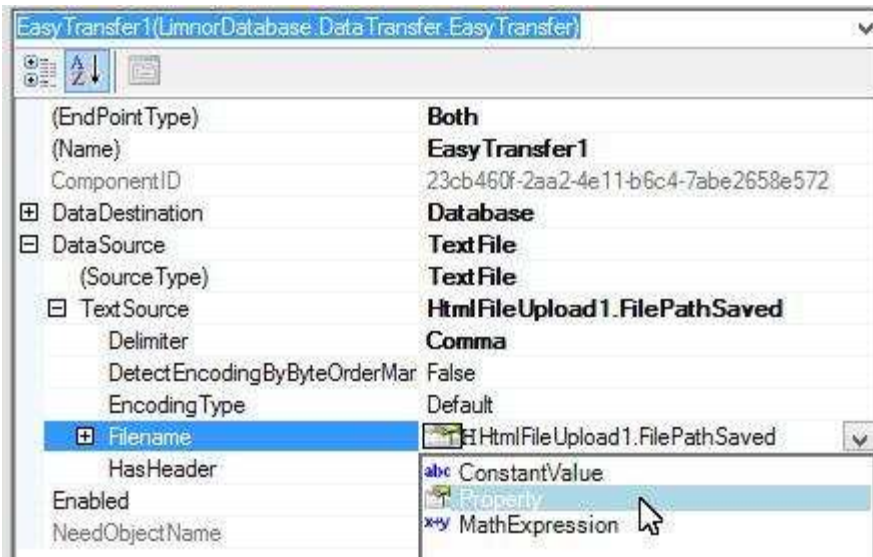


Set data source

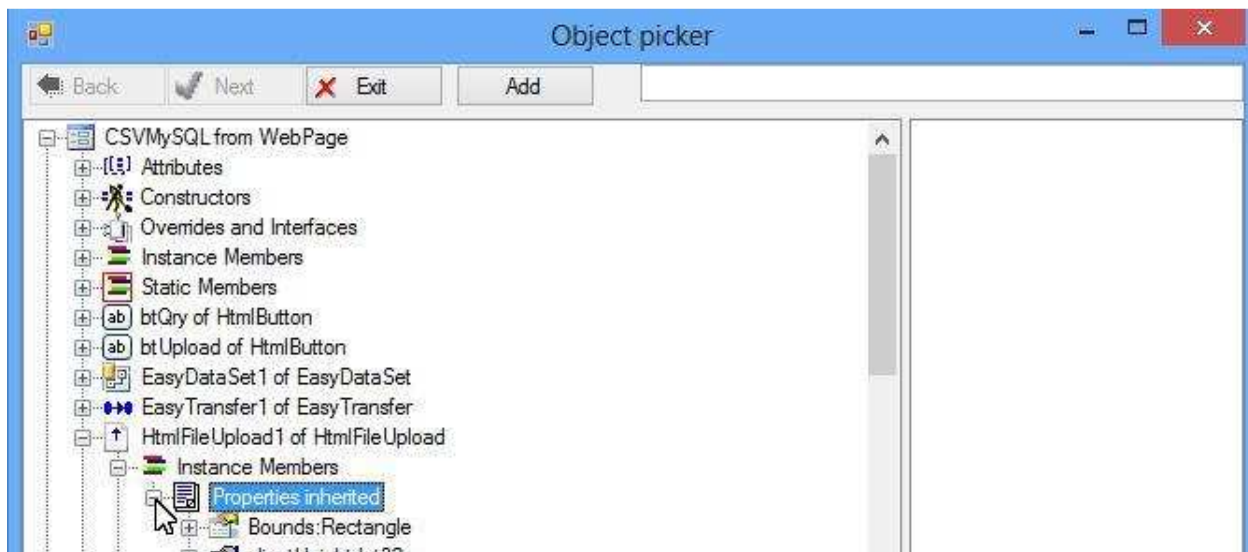
Set End Point Type to Both; set Delimiter to Comma; set HasHeader to True:



Note that for the Filename we use the FilePathSaved property of the File Upload component. It is set by choose "Property" on the Filename:

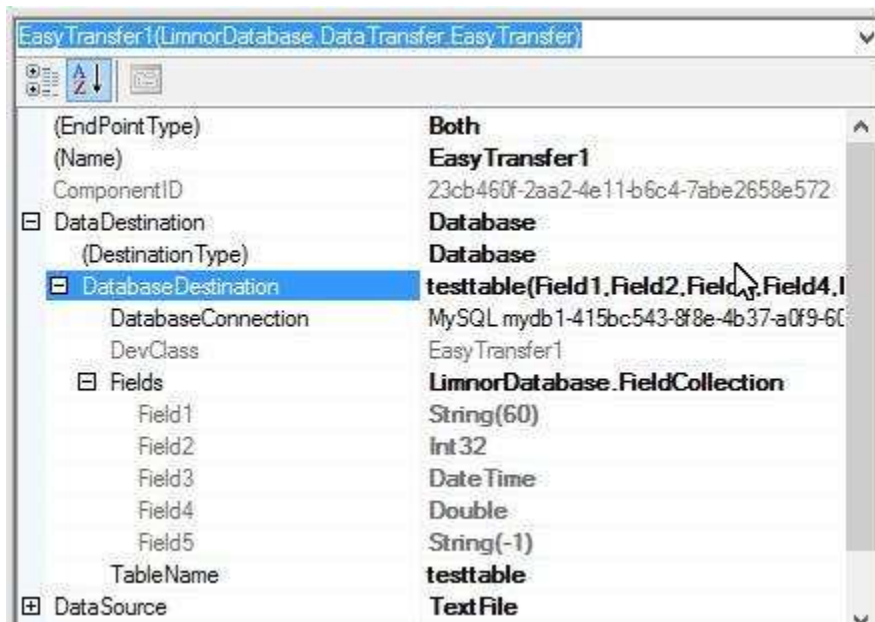


Select FilePathSaved property of the HtmlFileUpload:



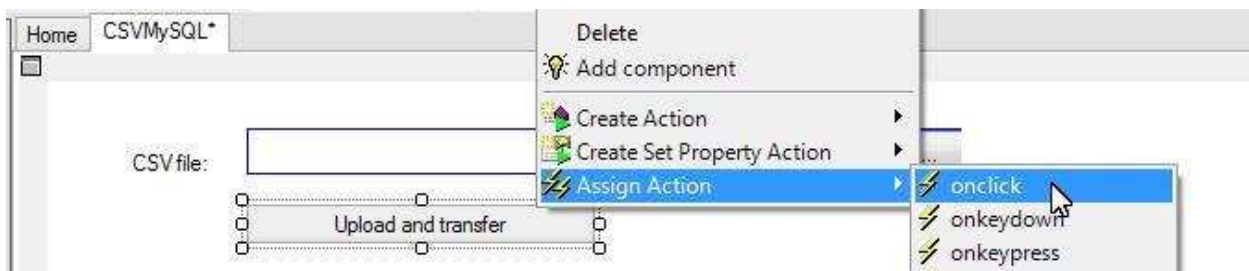
Set data destination

Setting data destination is the same as we did for Windows Form sample:

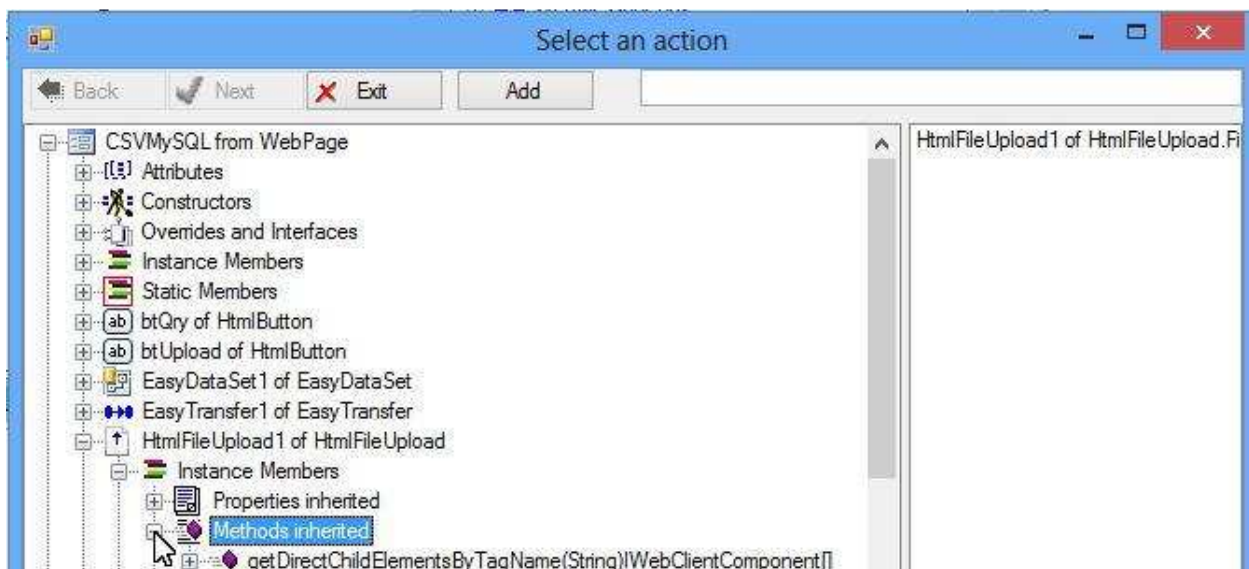


Upload File

We use a button to trigger file uploading:

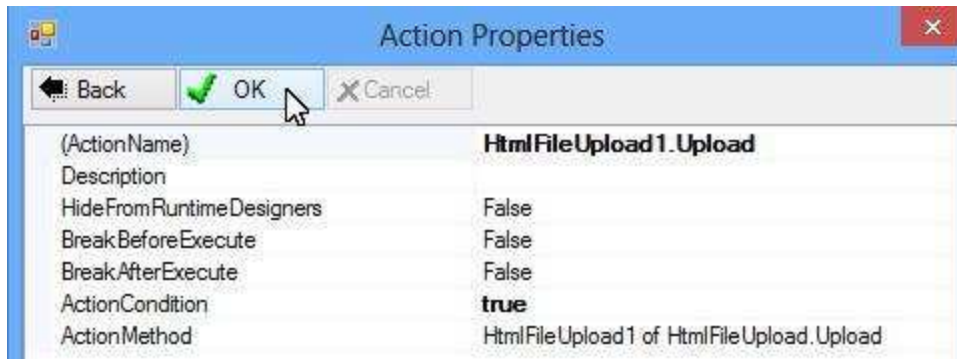


Choose Upload method:

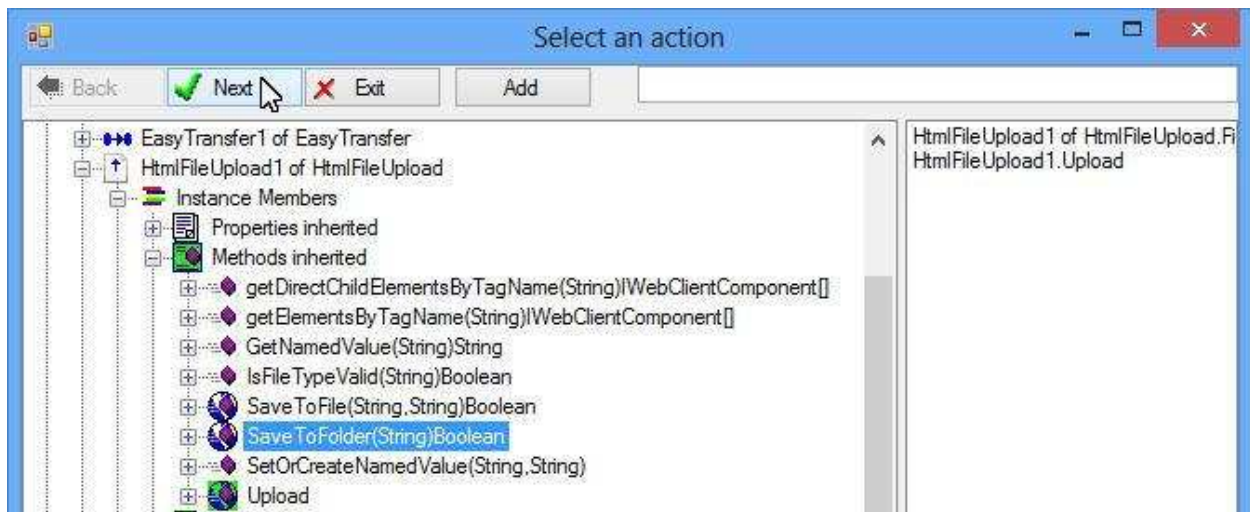
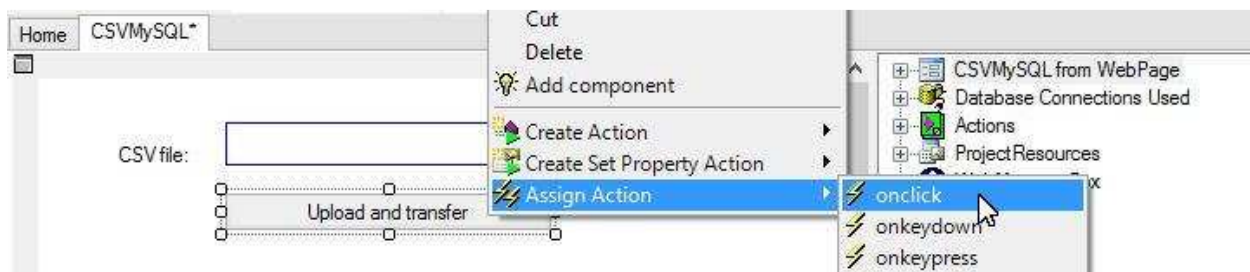




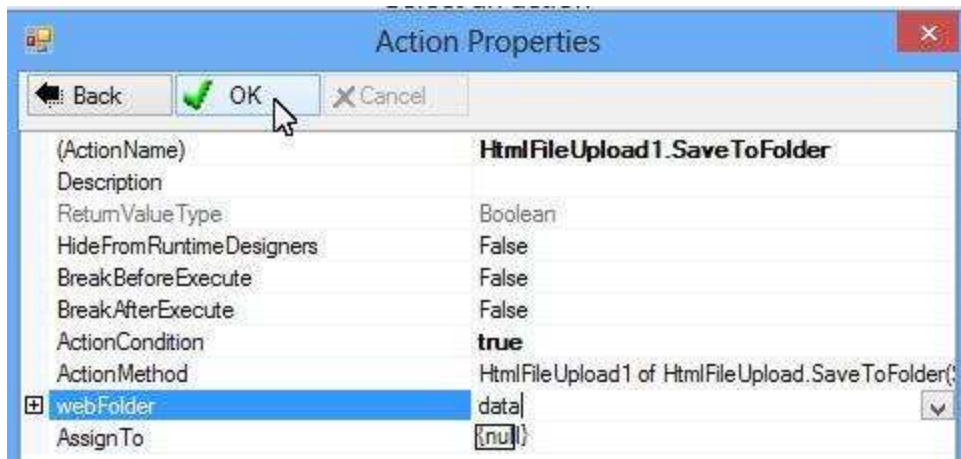
Click OK to create the action and assign it to the button:



After uploading the file we must save the file to a folder so that the file can be used via property FilePathSaved. We also do it at the Click event of the button:

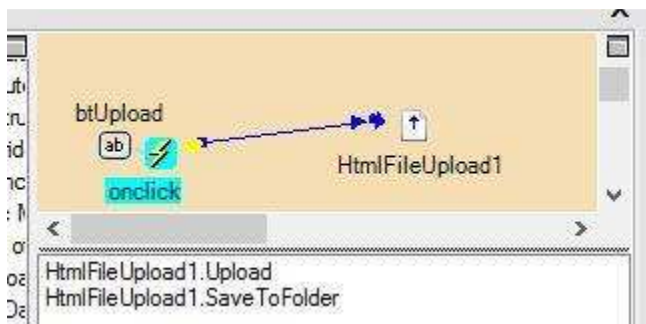


Suppose we want to save the file to a folder named data then we need to set the “webFolder” parameter to “data”:



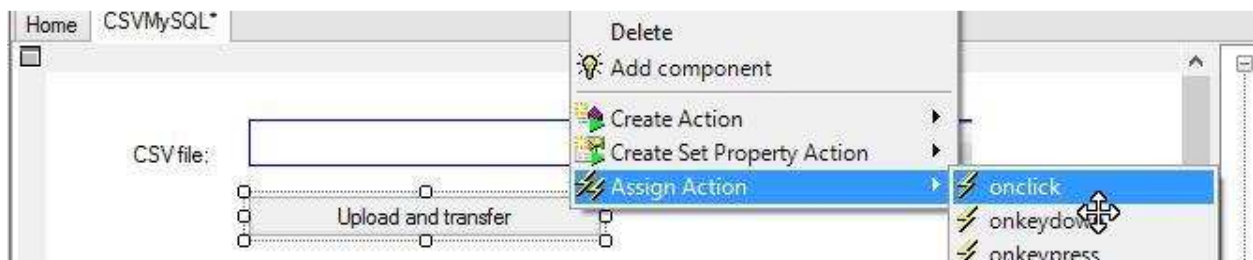
Note that the folder is under the WebFiles folder of your project. For example, if your project folder is c:\myprojects\webDataTransfer then the above setting refers to a folder c:\myprojects\webDataTransfer\WebFiles\data

By this programming, clicking the button will upload the selected file and save it to data folder:

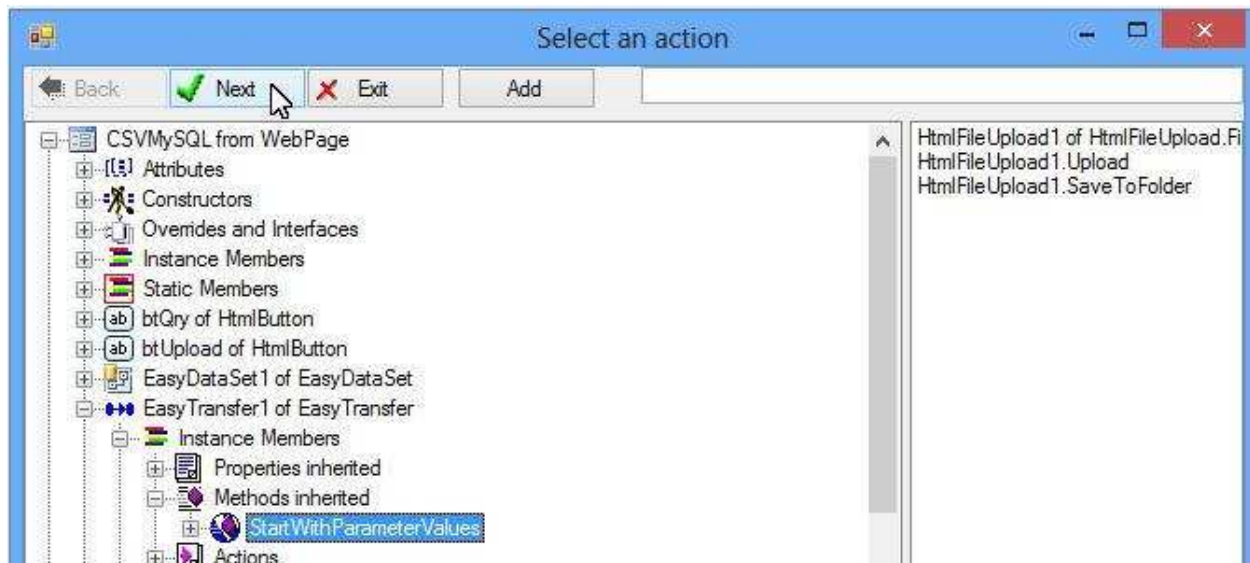


Execute data transfer

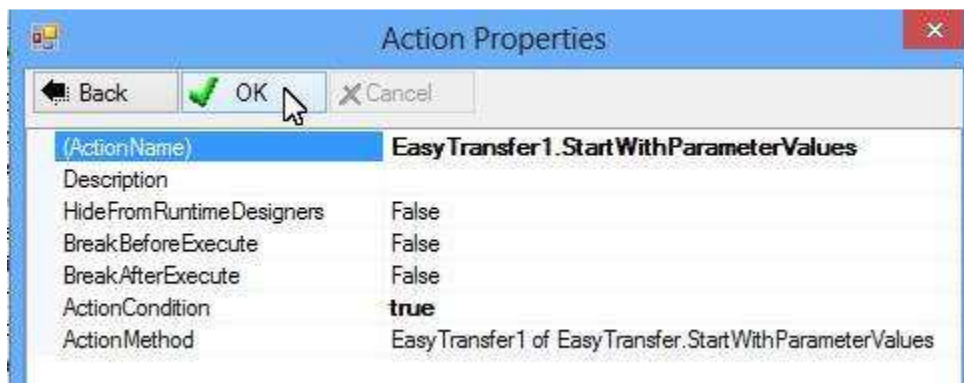
After file uploading, we may execute data transfer:



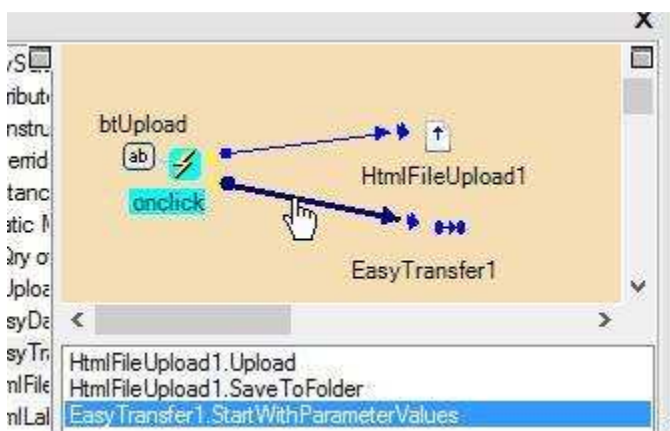
Select method StartWithParameterValues:



Click OK to create the action:

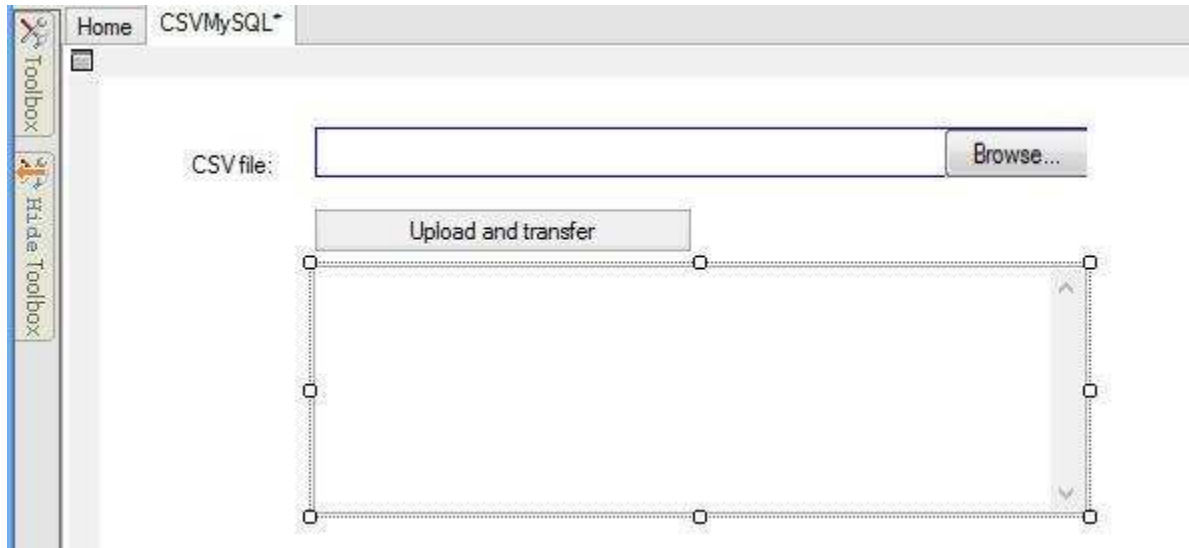


The action is created and assigned to the button:



Check data transfer errors

Data transfer may not be successful and we need to know the reasons. In this sample, we simply display ErrorMessage property of EasyTransfer on the web page. We use an HtmlTextArea to display error message:



Create web client method

It is usually a good idea to create a web client method for processing web server errors. Let's show a typical way of doing it:



Rename the new method processServerError:



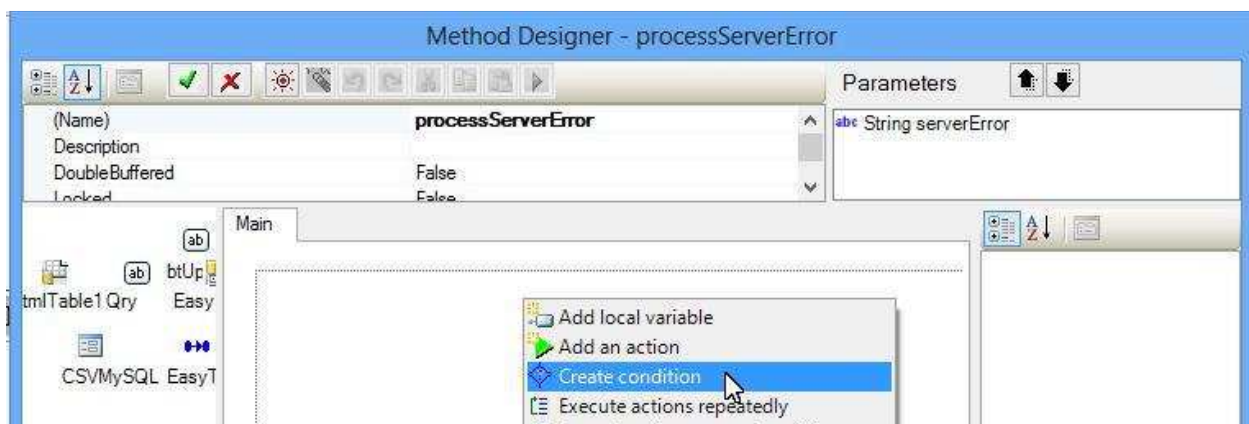
Add a string parameter for passing server error message into the method:



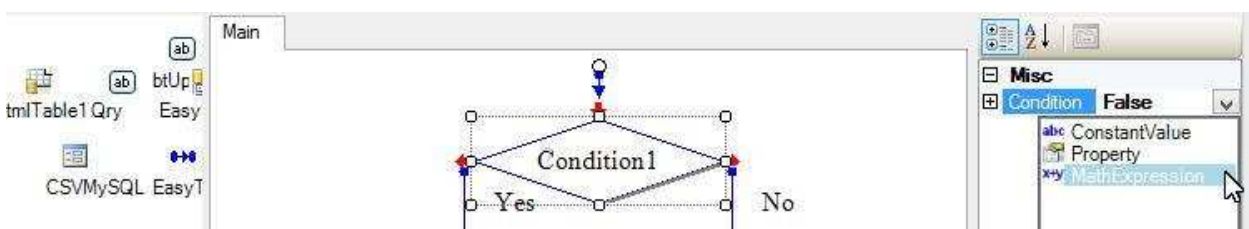
Rename the parameter to serverError:

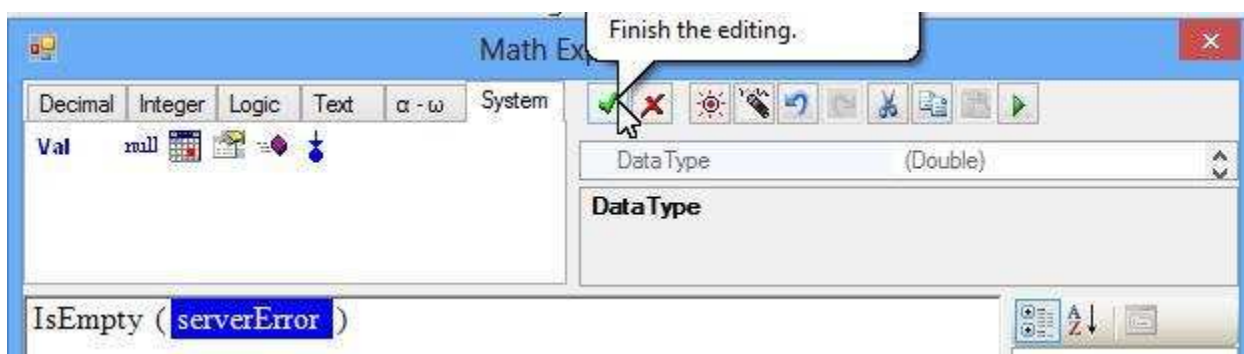
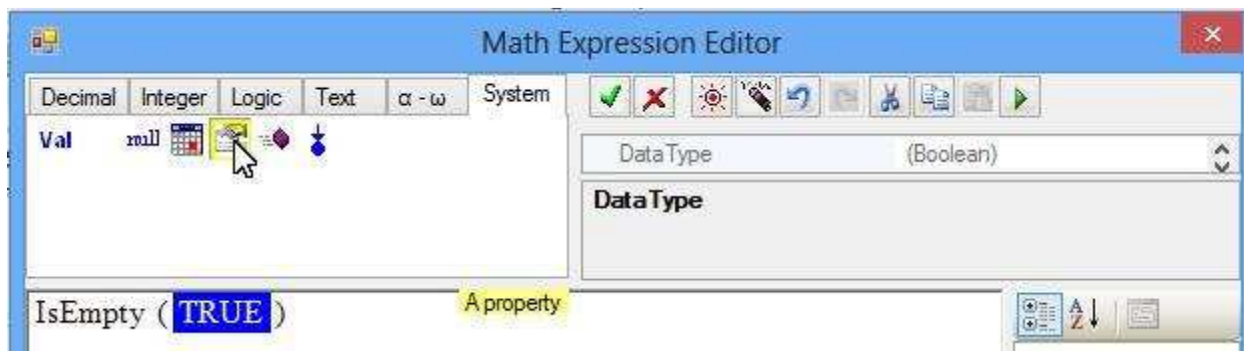
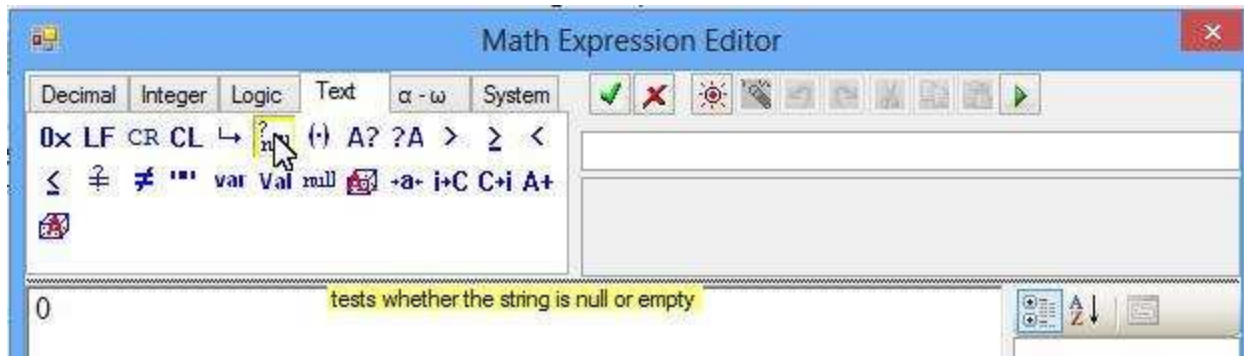


Add a Condition action to check serverError:

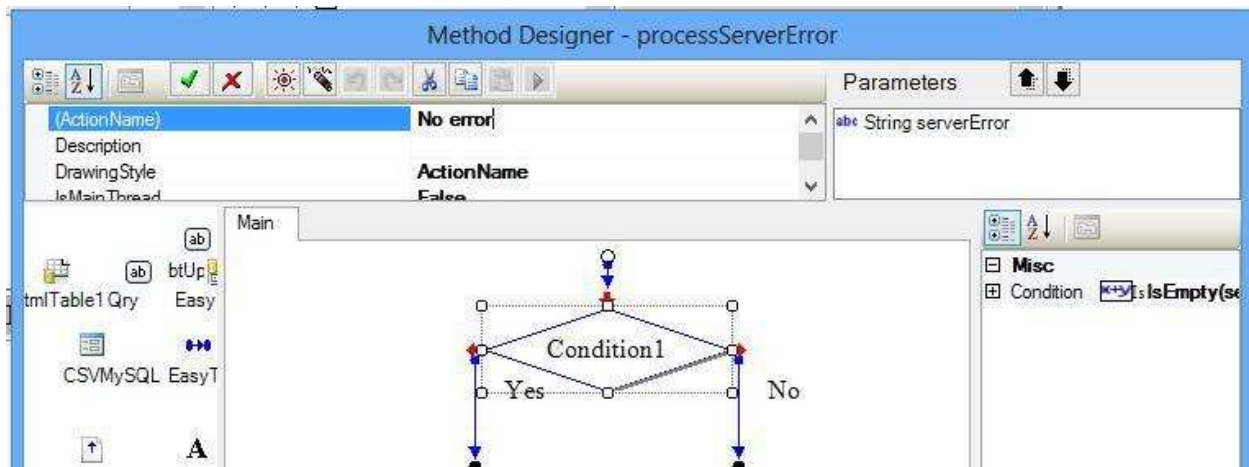


Set its condition to check whether the server error is empty:

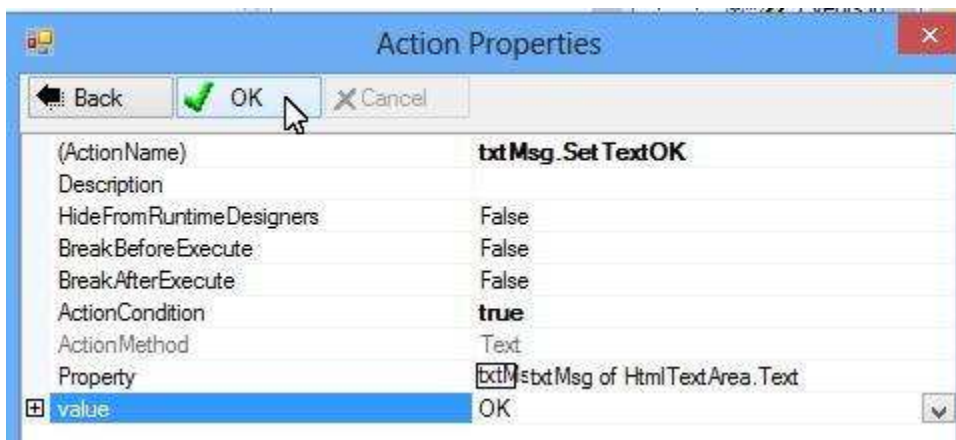
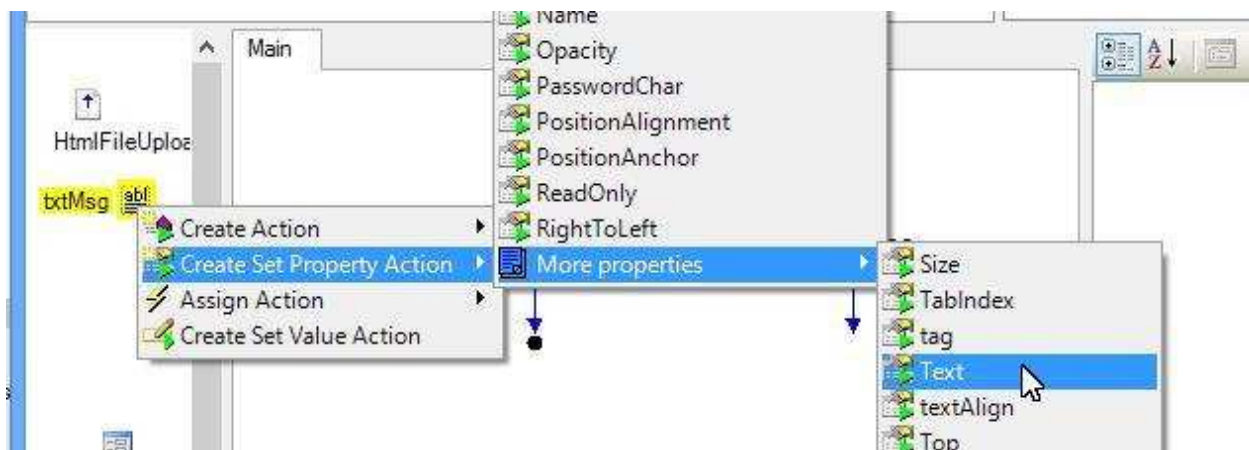




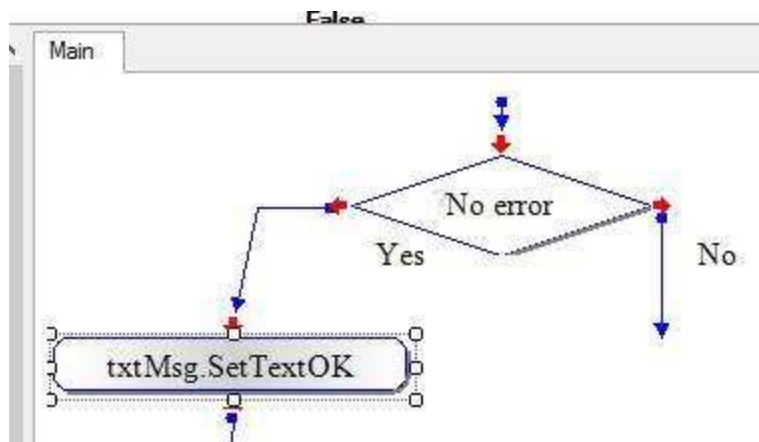
Rename the condition:



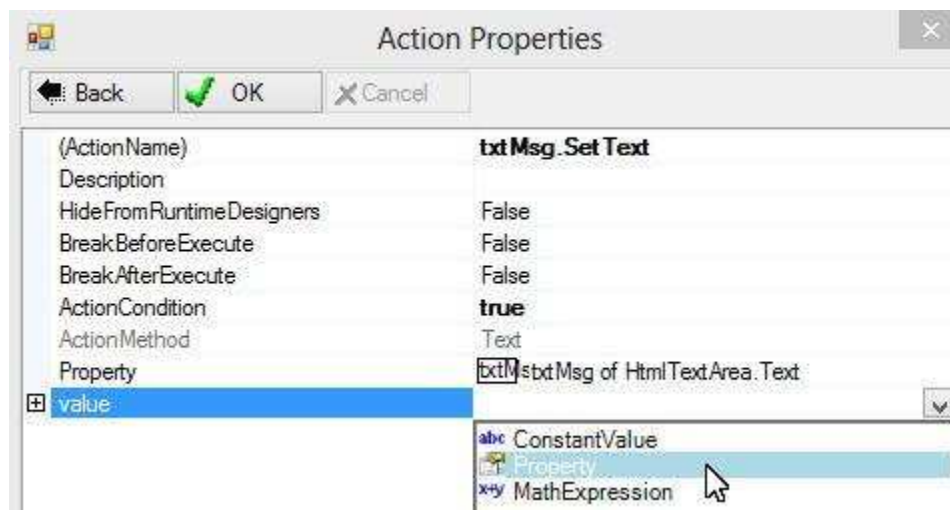
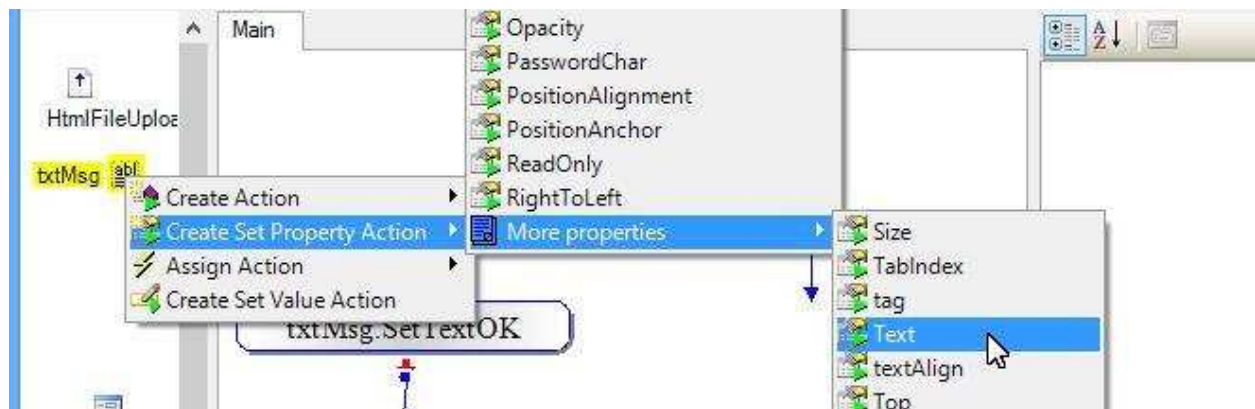
Show OK if the error is empty:

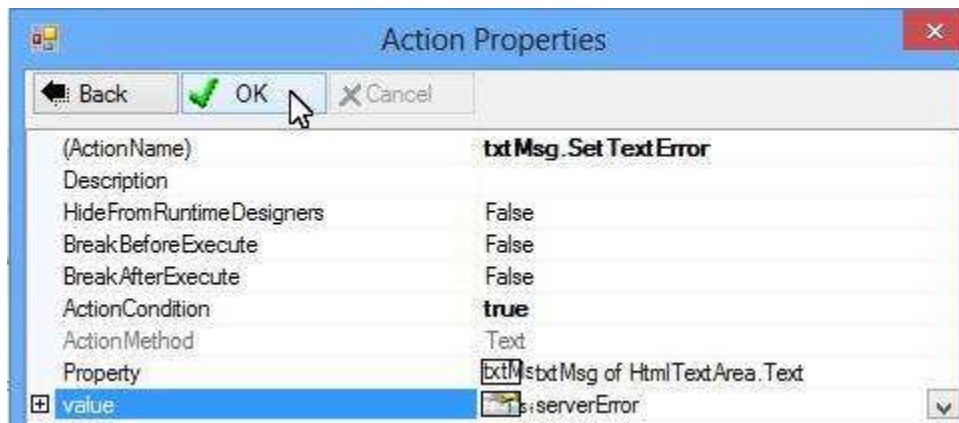
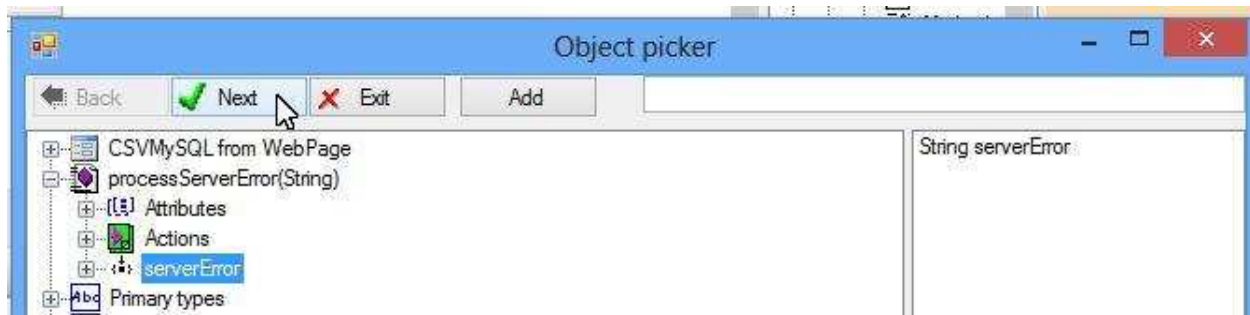


Link the action to the Yes port of the condition:

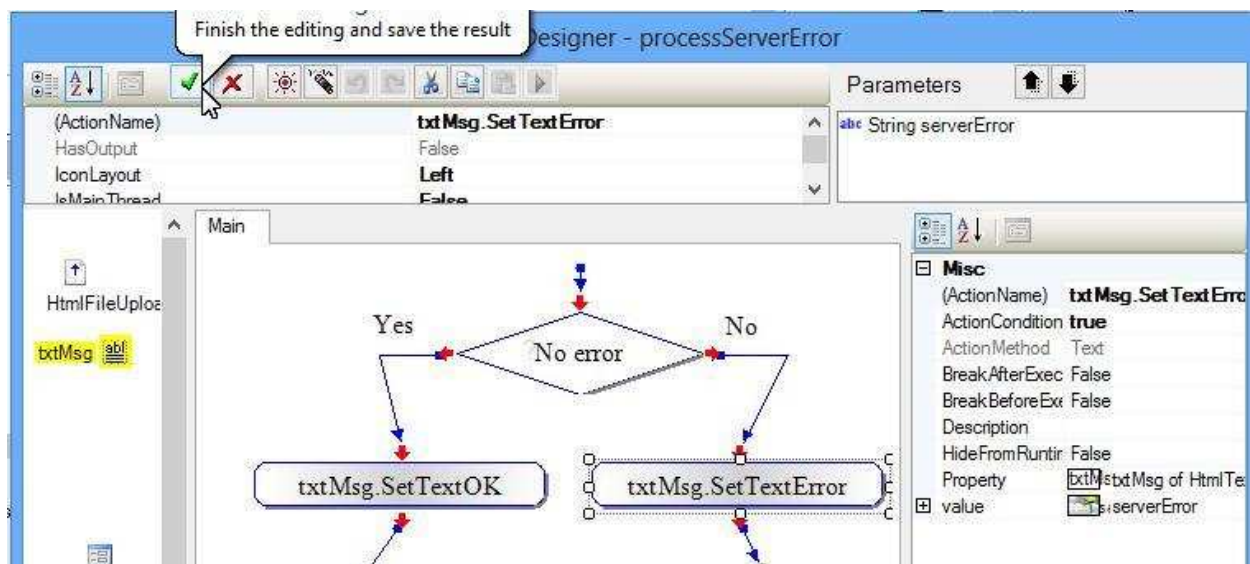


Show the error message if it is not empty:



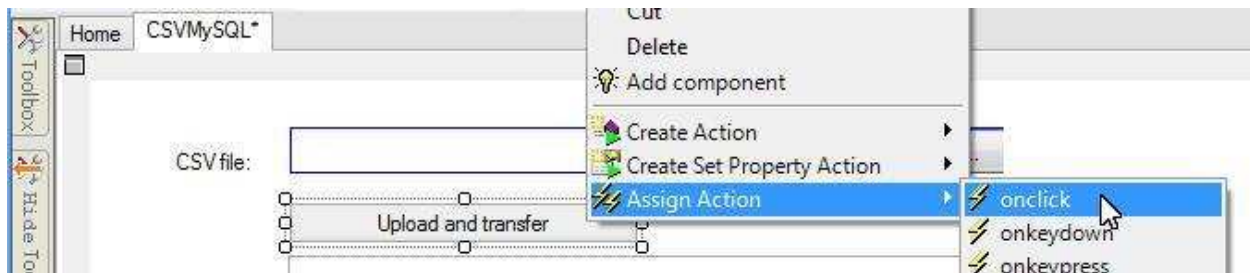


Link the action to the NO port of the condition:



Execute the method

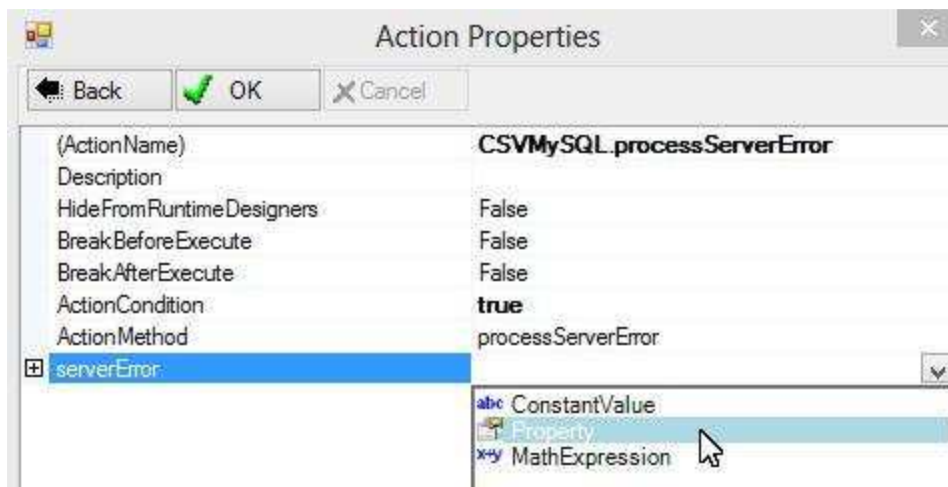
Execute this method at the button click:

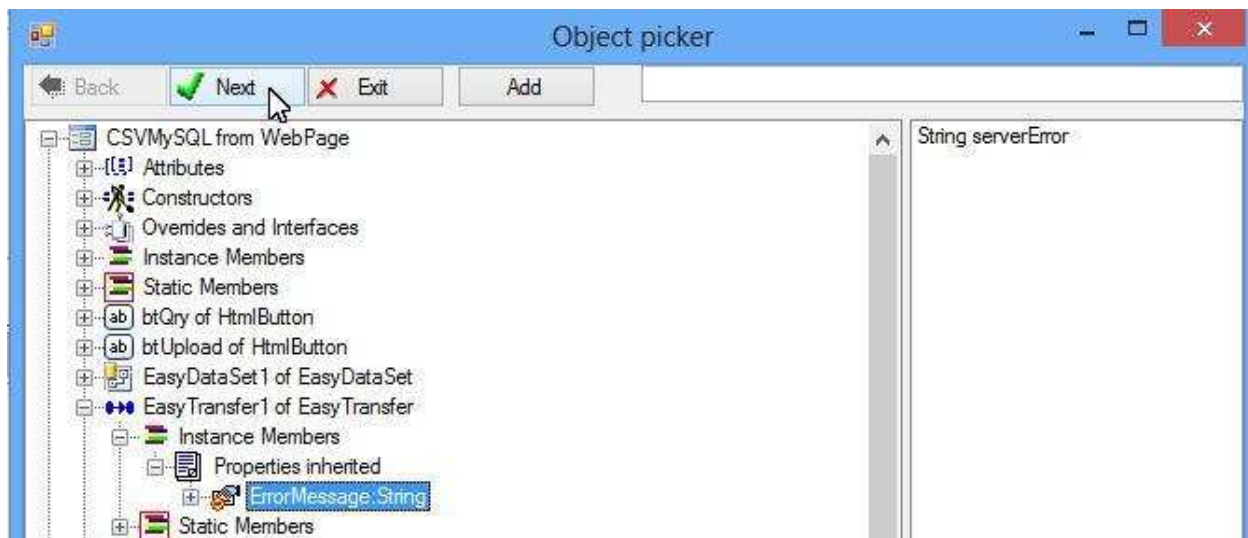


Select the method:

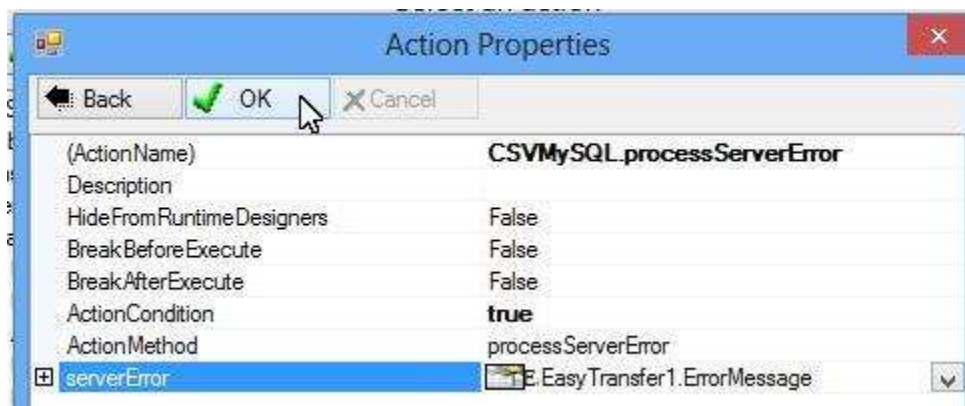


Pass the error message from EasyTransfer1 to the action:

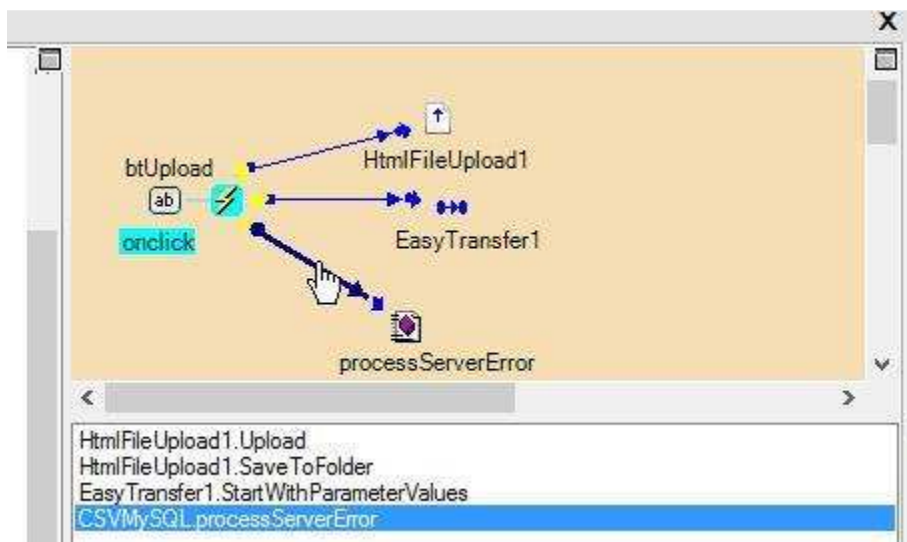




Click OK:

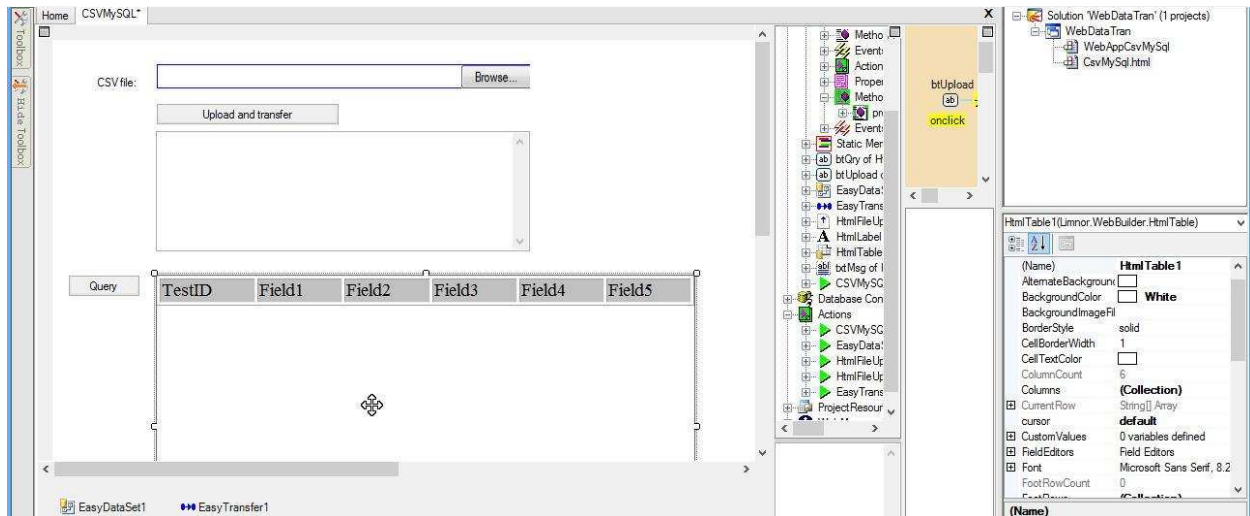


The action is created and assigned to the action:

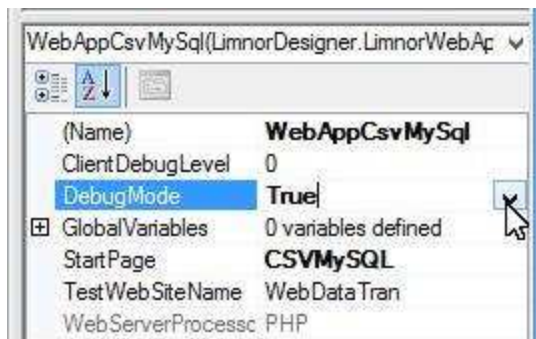


Test

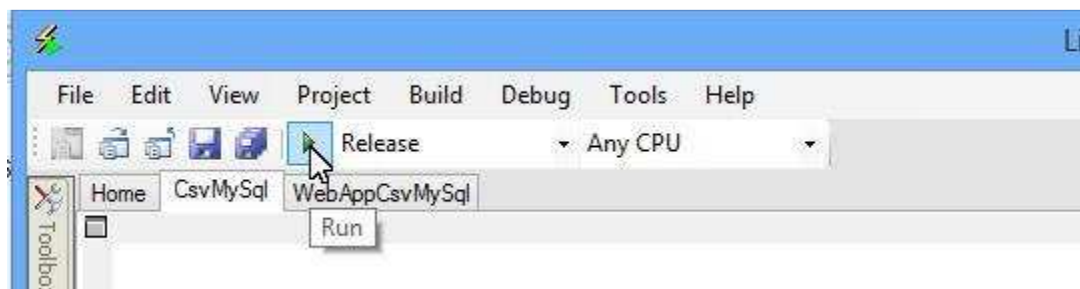
To show the data transfer results, we use an HtmlTable bound to an EasyDataSet to display the records from the table:



Enable debugging for showing client server communications:



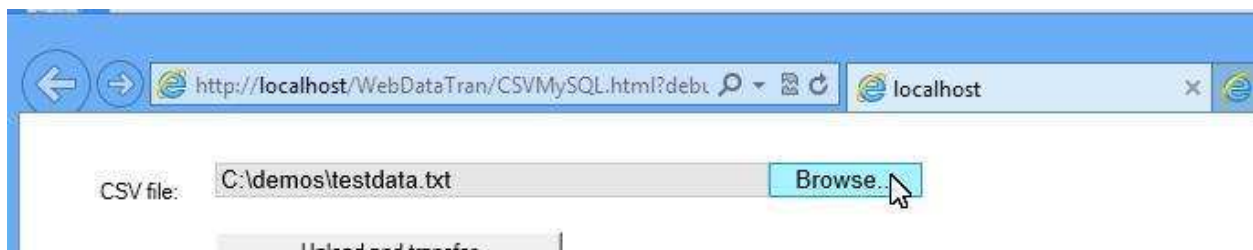
Run the project:



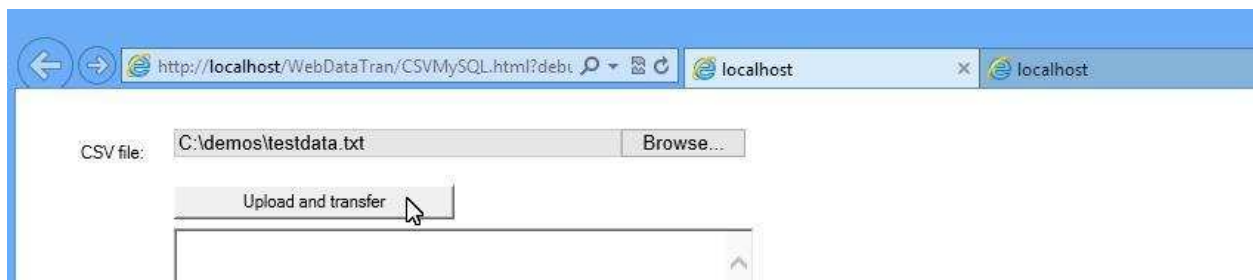
The web page appears. Click Query button to show existing data. In our sample, it is an empty table:



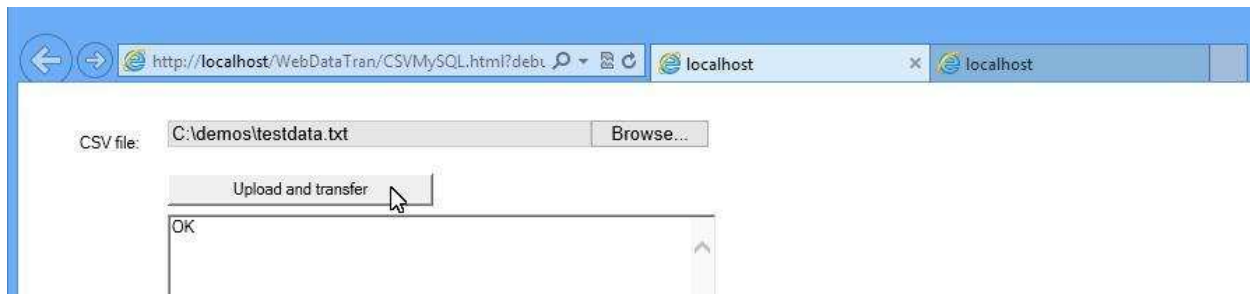
Locate the text file from the local computer:



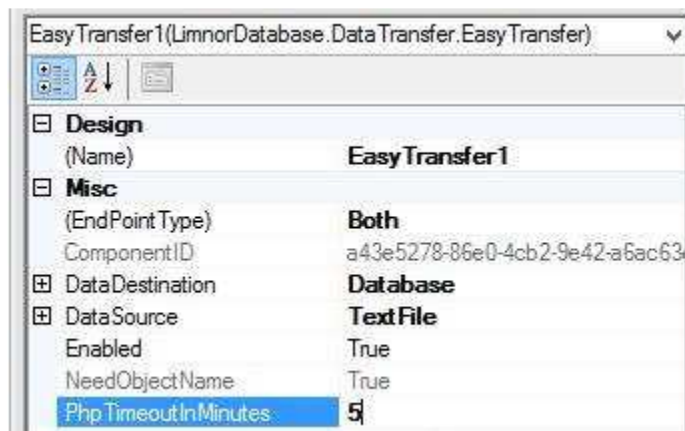
Click "Update and transfer":



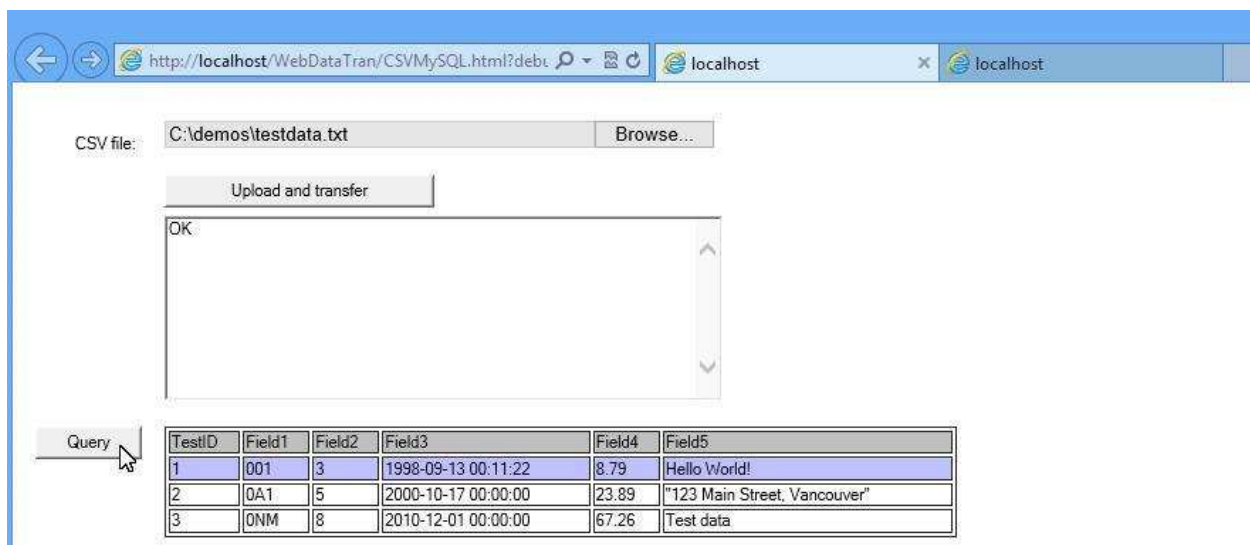
The file is uploaded and its contents are transferred into the table in MySQL database. The text area shows OK after a little while:



Note that if the text file is large then it may take a long time for OK to appear. You may want to set timeout value larger enough to avoid web page timeout error:



On showing OK, we may click "Query" to see the data in the table:



Since we enabled debugging, we can see client/server communications in a separate web page:



If there are errors or if after a long time nothing appears in the HtmlTextArea then you may find error messages in the debug web page. If you cannot figure out the meaning of the error messages then please send the messages to support@limnor.com to see if the Limnor Studio support team may figure them out.

Feedback

Please send questions and suggestions to support@limnor.com