

Manipulate Microsoft Office Word

Contents

Introduction	1
Data Merge Sample	1
Create Word Application Object.....	4
Create Word Application	4
Create Document Object	6
Manipulate Word Document – Search and Replace.....	13
Manipulate Word Document – Modify and Save	23
Execute Method.....	36
Test.....	37
Feedback	39

Introduction

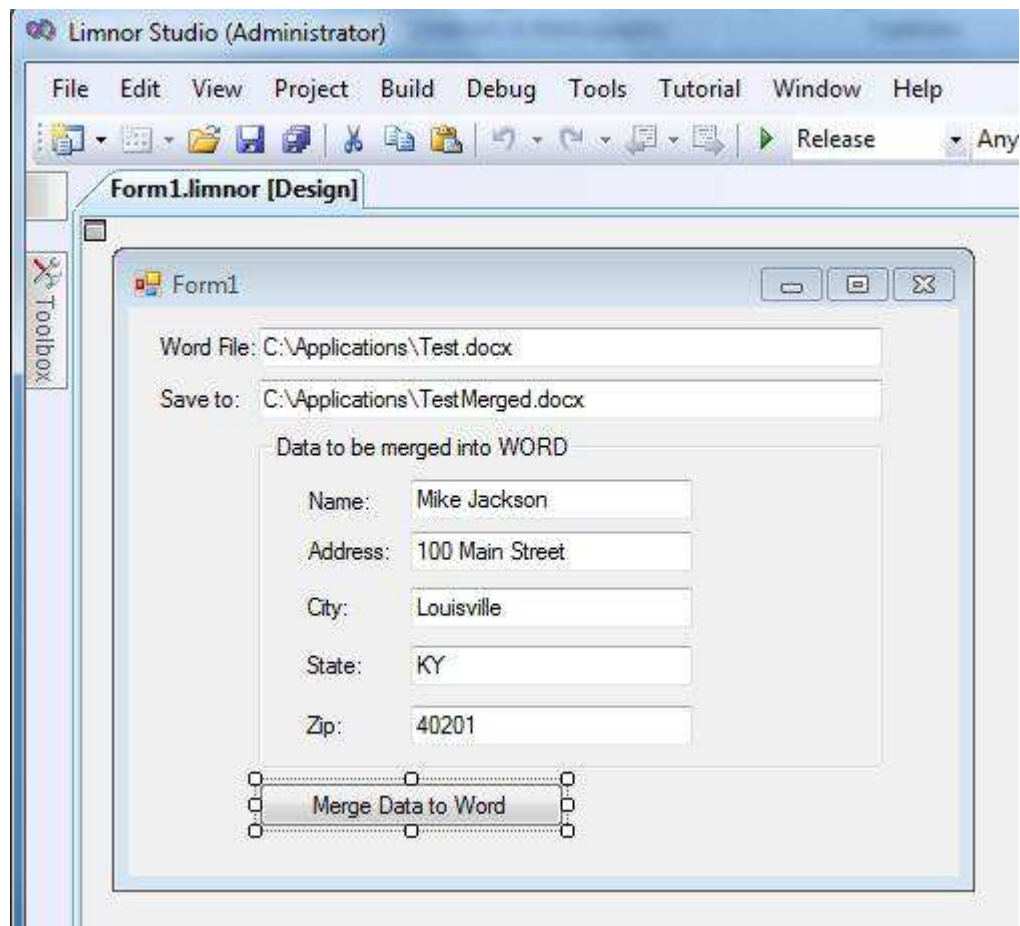
All Microsoft Office applications can be manipulated programmatically using .Net Framework interfaces provided by Microsoft. The interface is called Primary Interop Assemblies (PIA). This sample demonstrates merging data into a Word document and save the document to a new file, using PIA.

PIA is very powerful, enabling total control of the Office. But the PIA is not intuitive. This sample project is trying to point out pitfalls to watch out. The sample uses PIA for Office 2007.

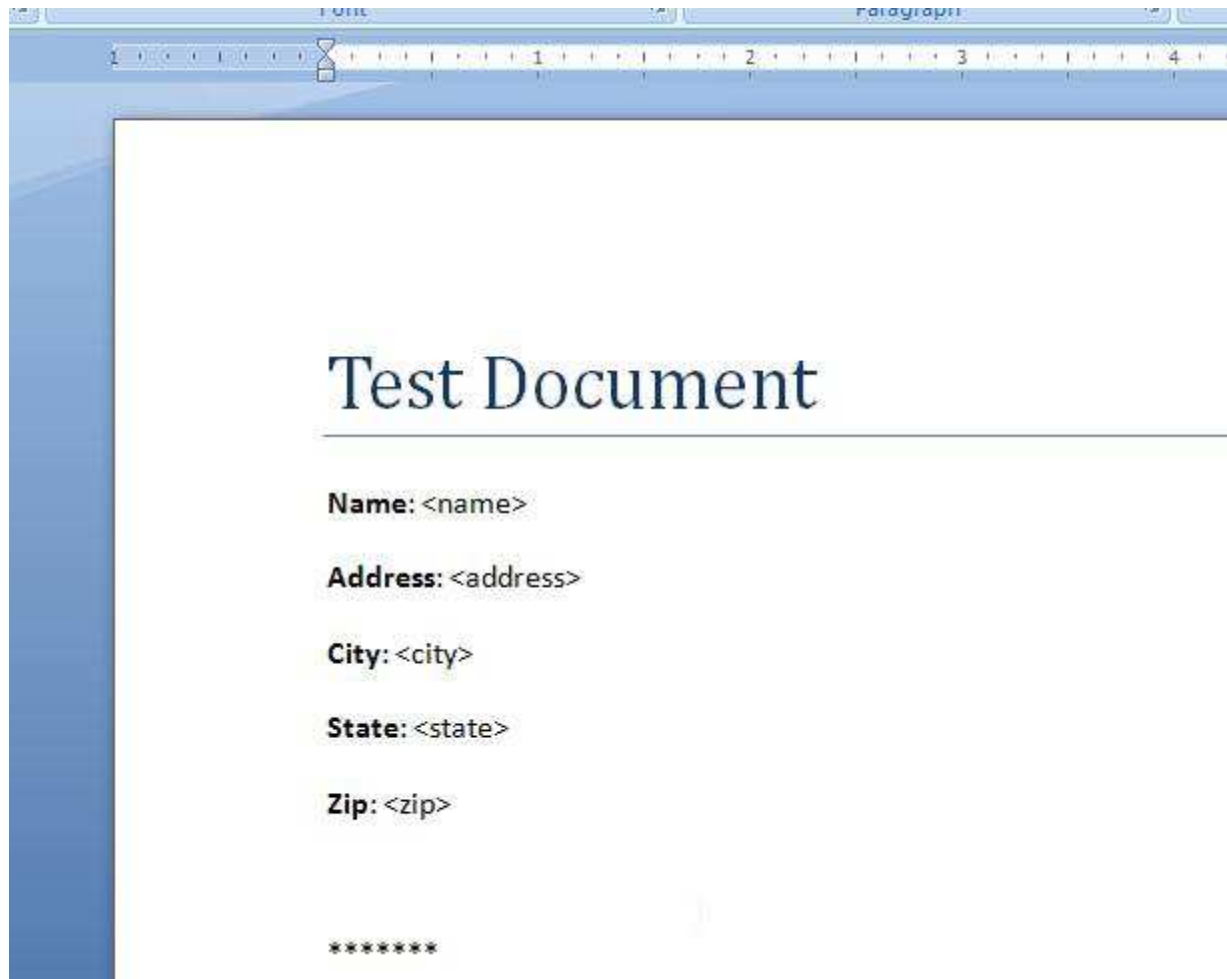
Suppose Microsoft Office is installed and Microsoft Primary Interop Assemblies (PIA) is installed. For downloading PIA for Office 10, see <http://www.microsoft.com/download/en/details.aspx?id=3508> or search the internet for “PIA downloads” for PIA matching the version for your Office.

Data Merge Sample

The sample is a Windows Form application. The data to be merged into the Word document are entered into text boxes. A button is used to initiate the data merging action.

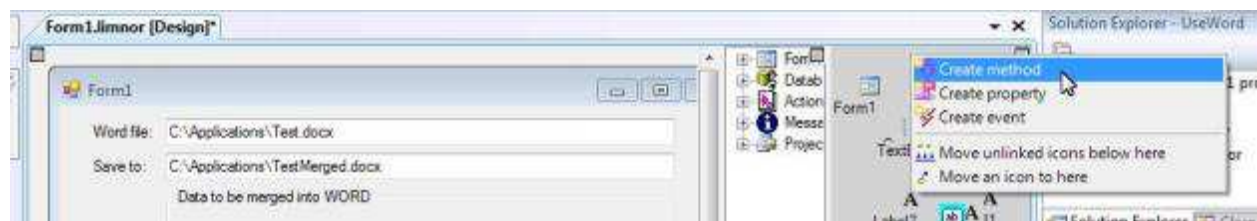


Suppose we use such a test WORD document:



When the button is clicked we want to use the data in the text boxes to replace the place holders marked with "<" and ">".

We may create a method to do the job of merging the data to the Word file.



The Method Editor appears. Rename the method to "MergeToWord":



In this sample, all the programming with using Word will be done in developing methods.

Create Word Application Object

To use Word, we need to at least get a Document object. To get a Document object we must have a Word Application object.

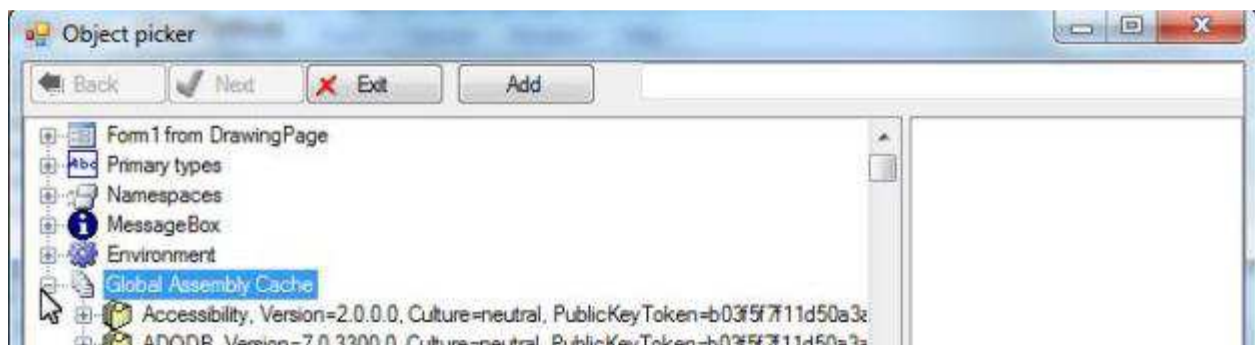
In the sample, we first create a Word Application object. The Application object has a Documents property which has an “Open” method. We use it to create a new Document by opening a Word file.

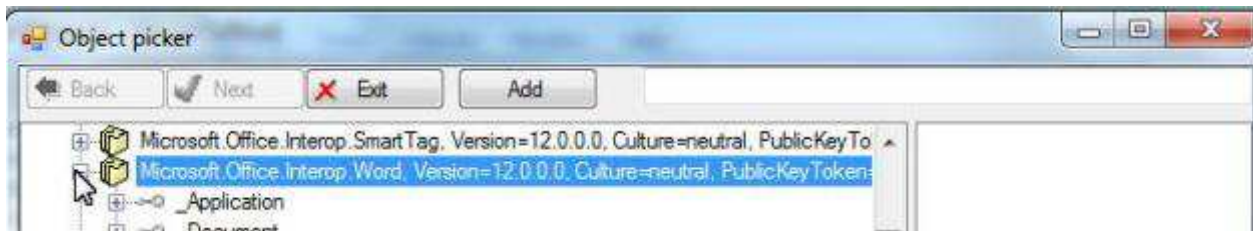
Create Word Application

Right-click the Method Editor; choose “Add local variable”:

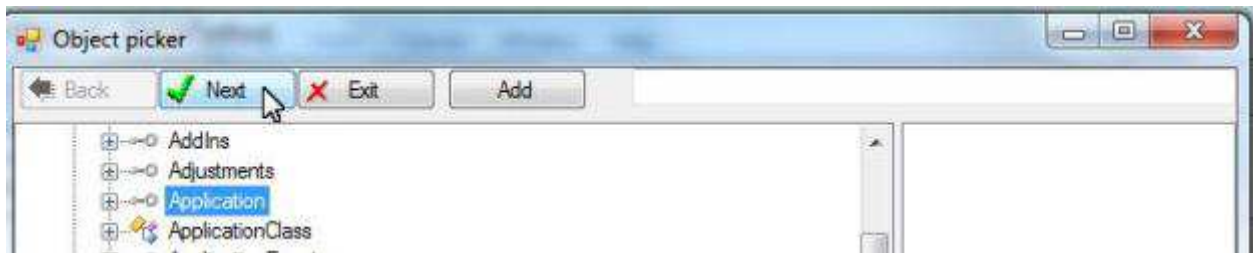


The PIA is under the Global Assembly Cache:

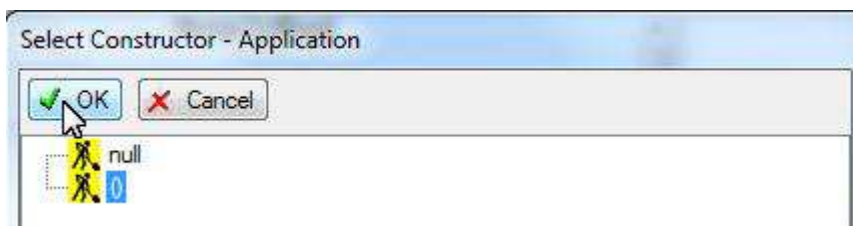




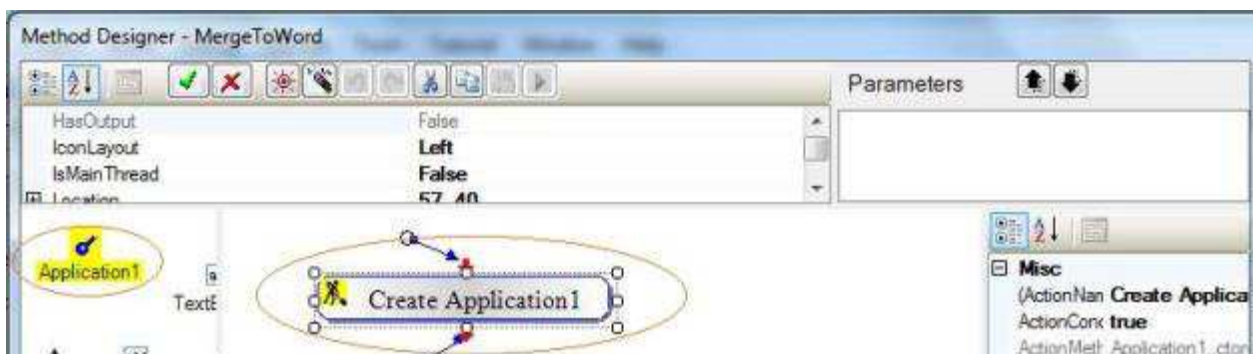
Select Application and click Next:



Select the parameter-less constructor to create an instance of Word Application:



An Application object appears in the Variable Pane. An action appears in the Action Pane to construct the Application object:



Note the following rules

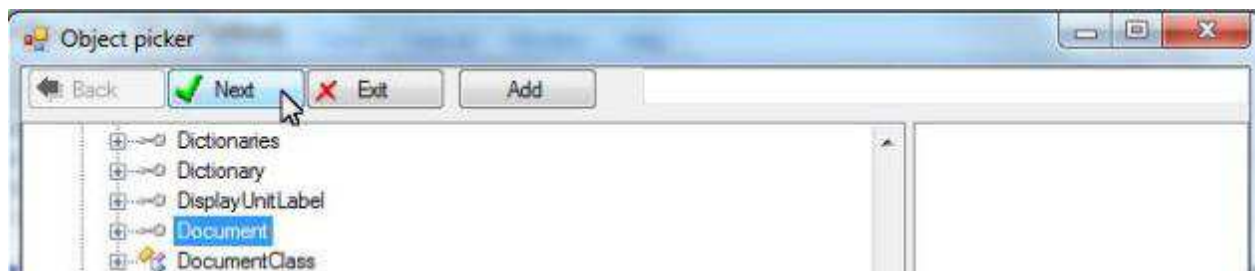
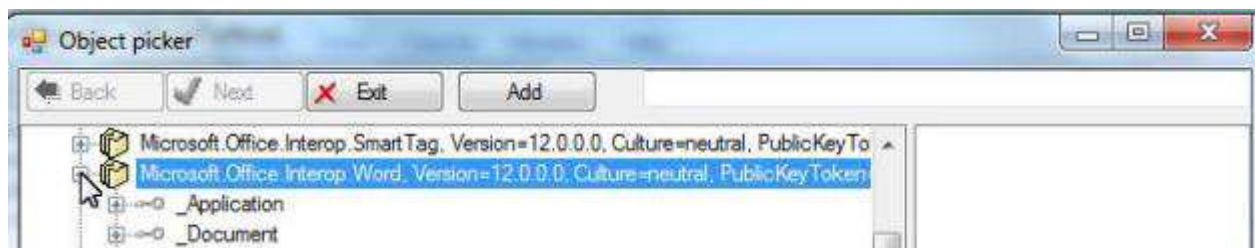
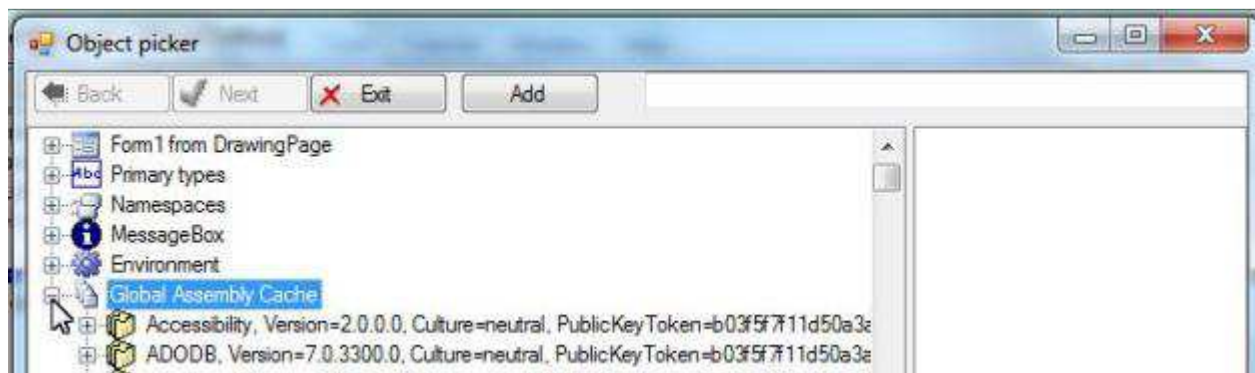
- All PIA objects to be used must be created via “Add local variable” so that their properties, methods and events can be accessed for programming
- Only the Application object should be manually constructed

Create Document Object

We must use “Add local variable” to create a Document variable first. Right-click the Method Editor; choose “Add local variable”:



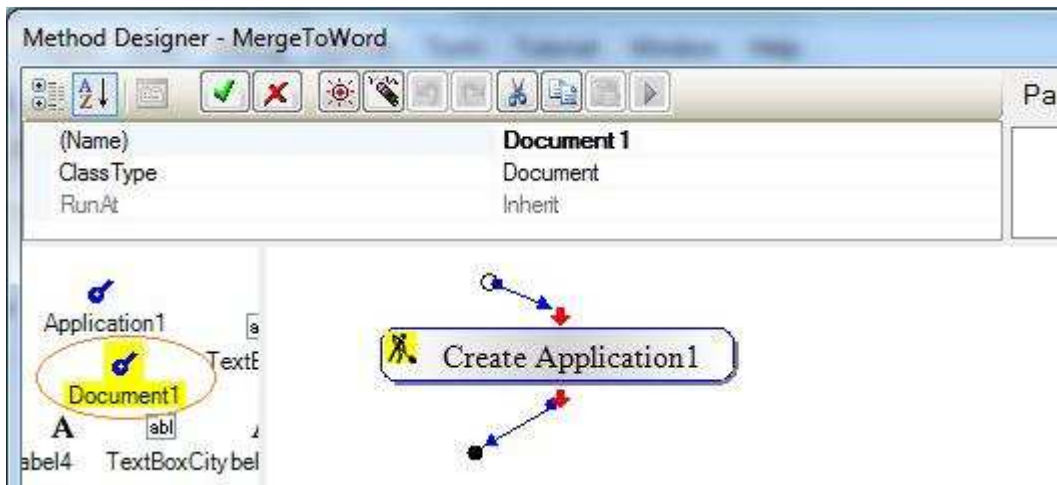
Select Document in the Global Assembly Cache:



Select “null” because we should not construct an instance of Document manually. We need to let the Application object to create Document:

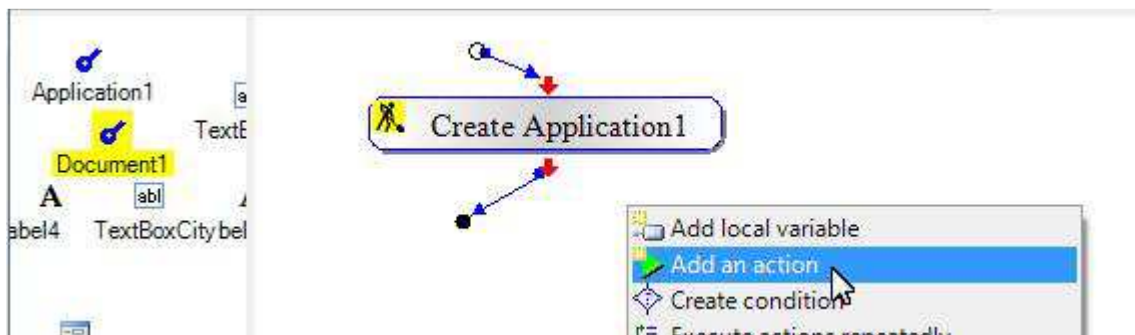


A Document variable appears:

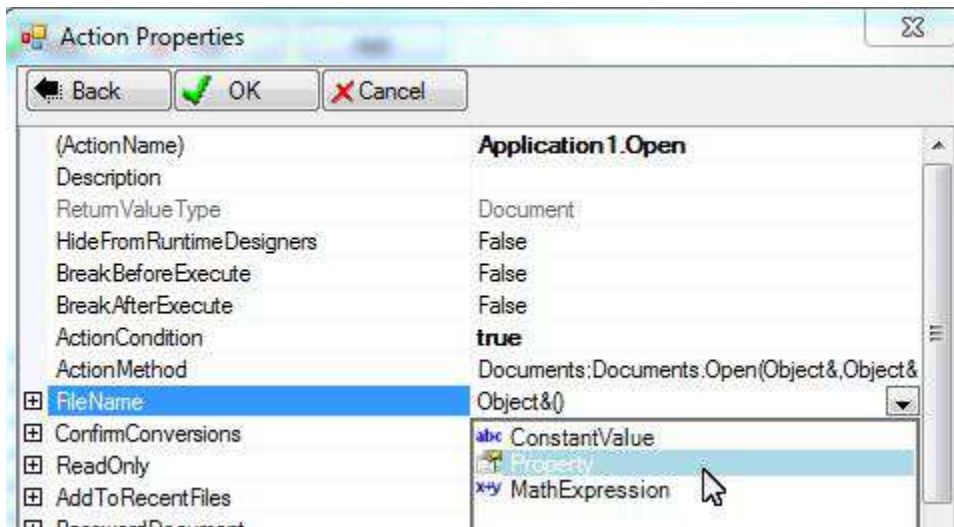


The Word Application has a Documents property. The Documents property has an Open method. We may use it to open a Word document.

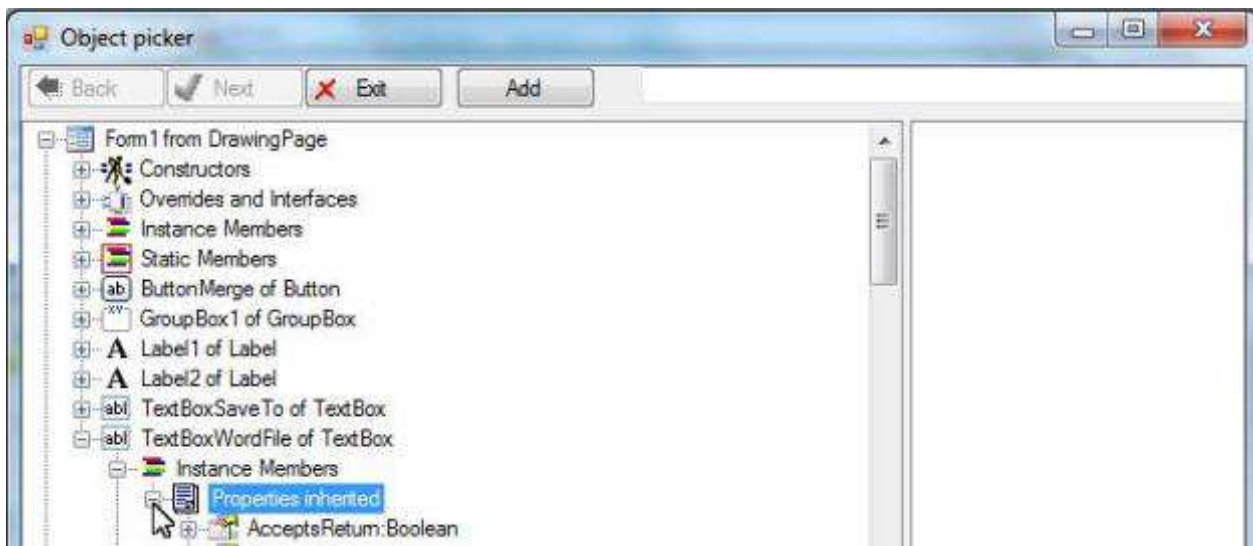
Right-click the Action-Pane; choose "Add an action":



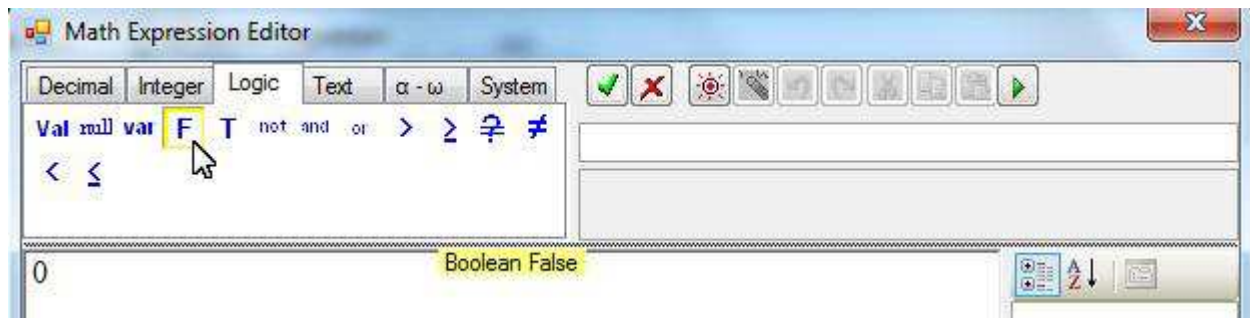
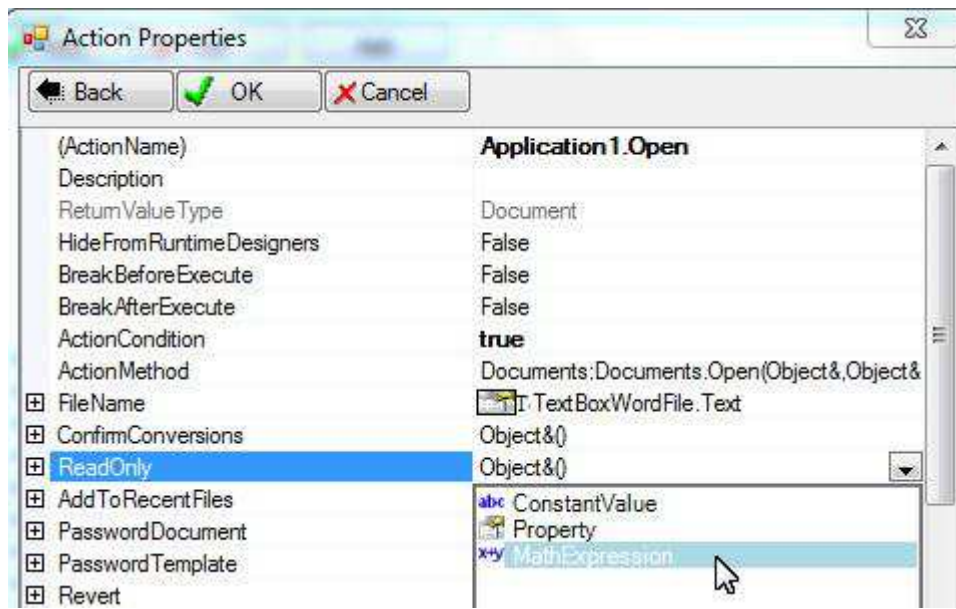
Locate the Documents property of the Word Application and select its Open method:



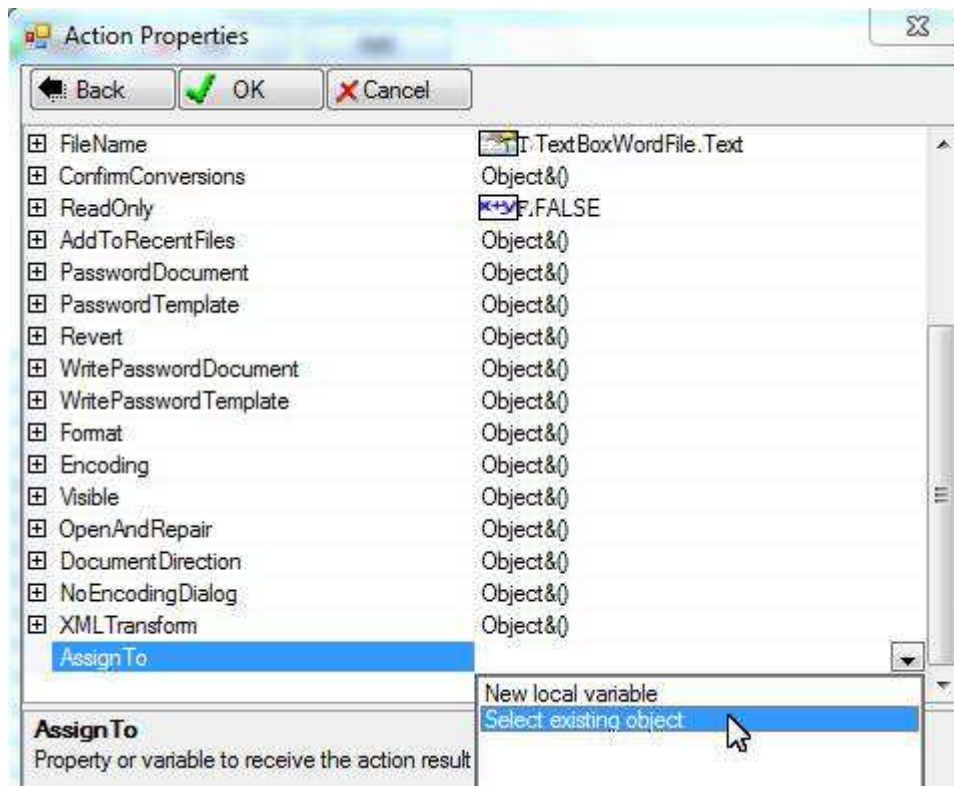
Select the Text property of the text box for the Word document:



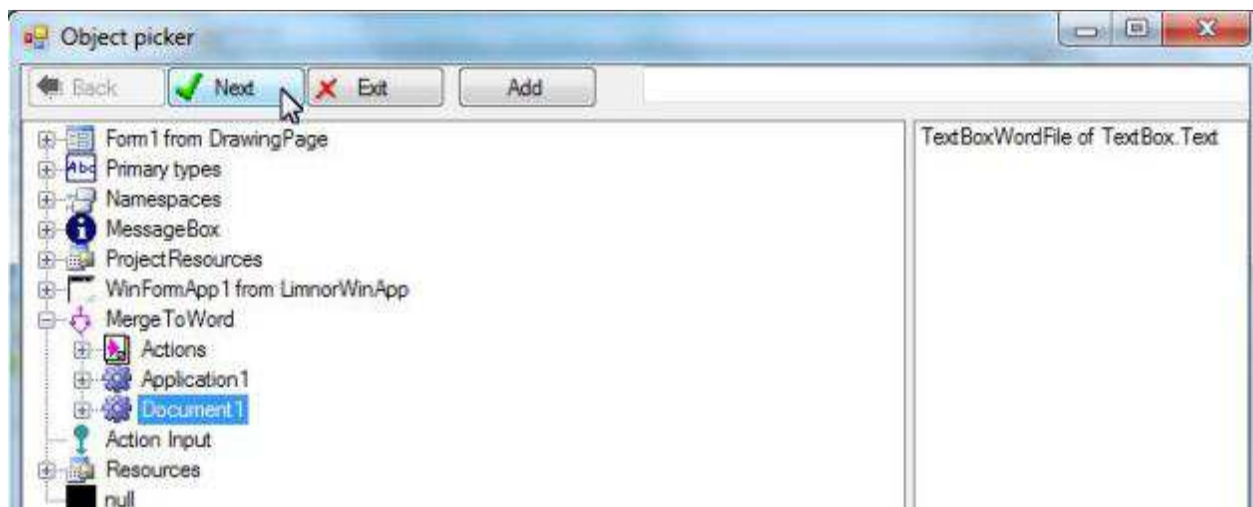
For "ReadOnly", use "MathExpression" to set it to False:



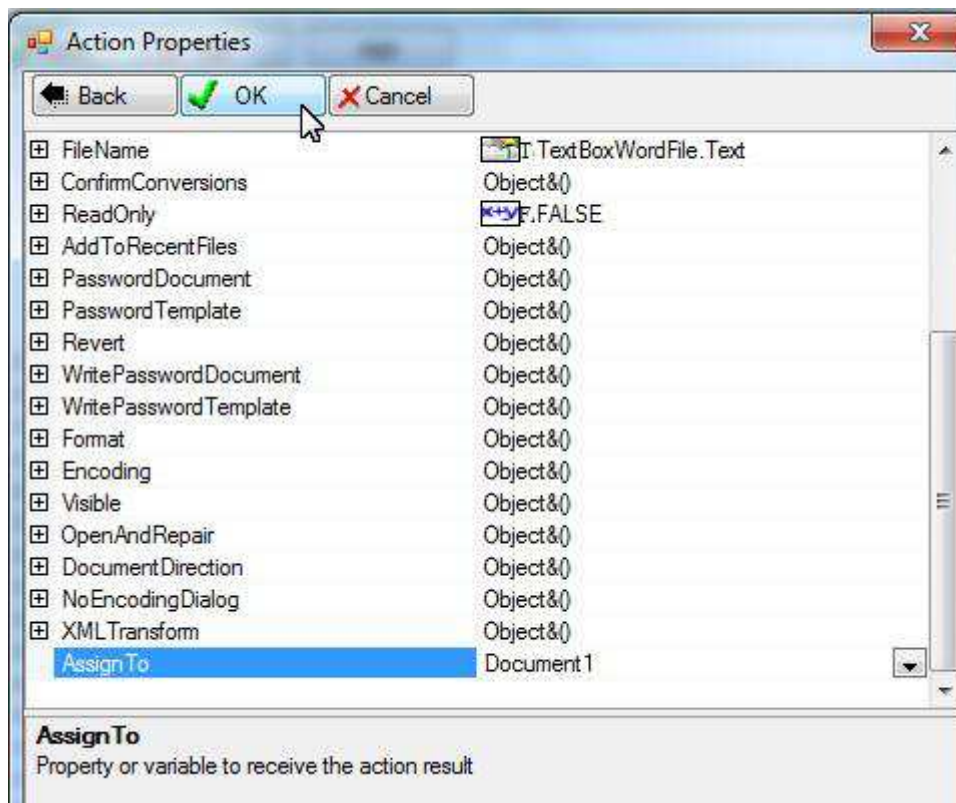
For "AssignTo" of this action, choose "Select existing object":



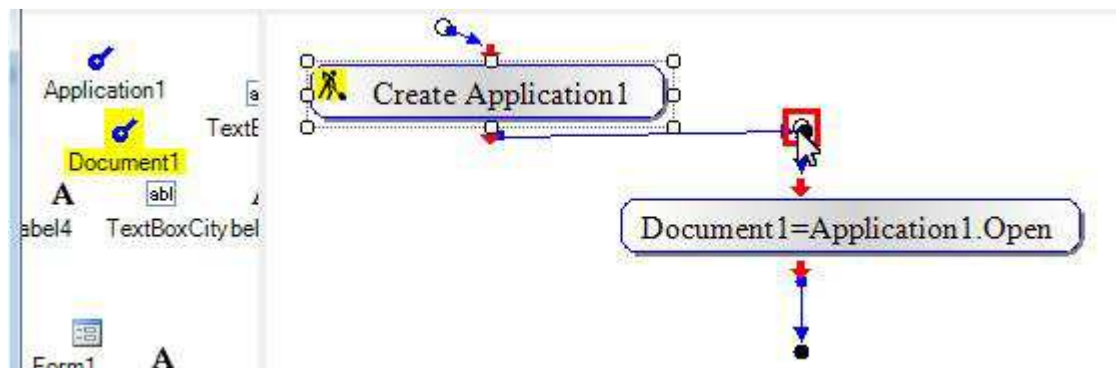
Select the Document variable:



Click OK to finish creating this action:



Link the actions:



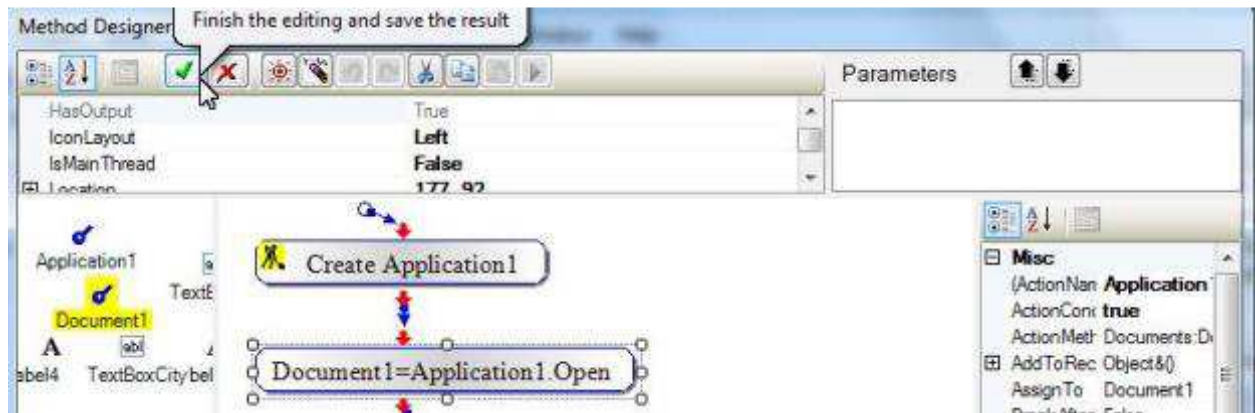
Note the following rules

- Constant parameters in a PIA action should be specified via Math Expression so that correct data types are used.
- Except the Application object, all other PIA objects should not be constructed manually. Some actions can be used to return PIA objects. Specify "AssignTo" property of such an action to a PIA variable to get access to a PIA object.

Manipulate Word Document – Search and Replace

Now we have a Document object. We can do data merge into it. We do it by searching and replacing. Because we are going to do more than one search-replace, we want to create a new method to do it so that our search-and-replace programming can be re-used many times.

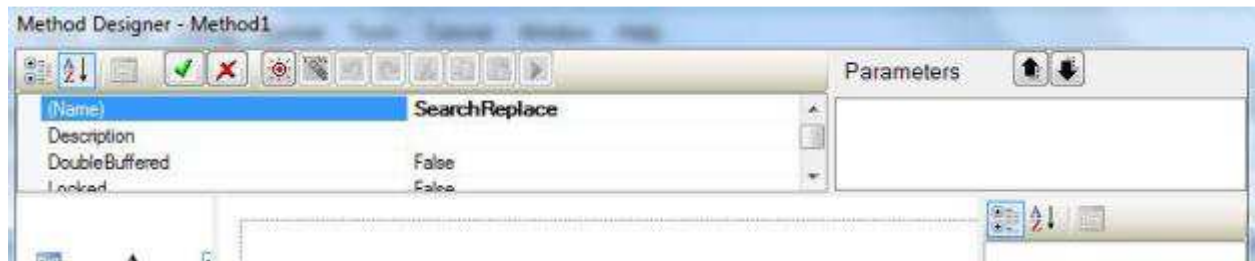
Close the current method so that we may work on a new method:



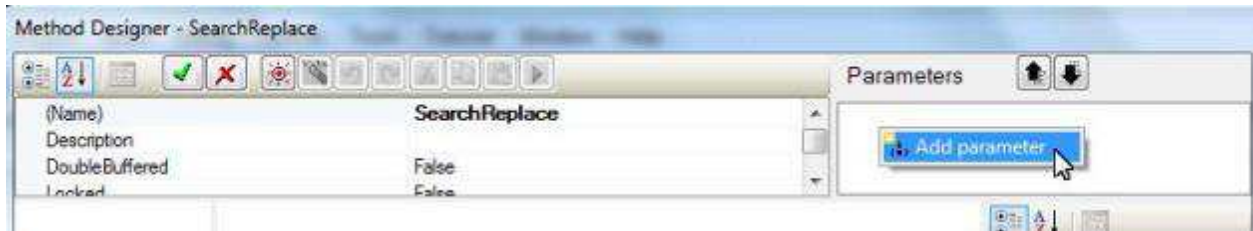
Create a new method to do search and replace in a Word document.



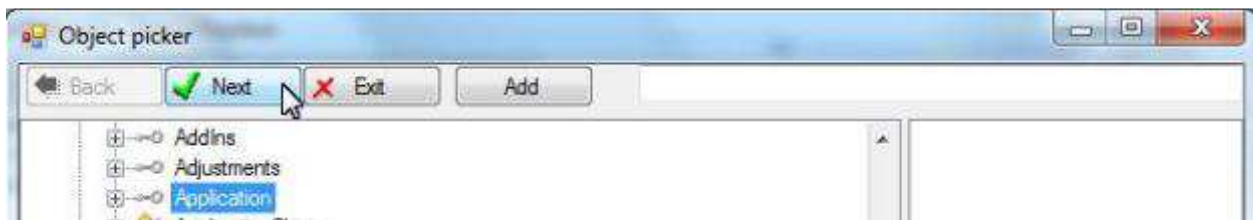
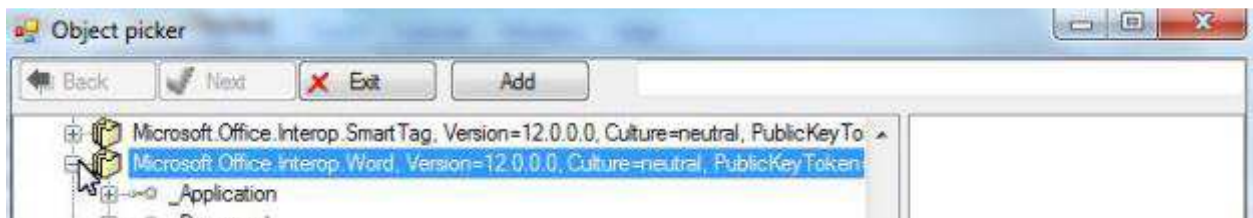
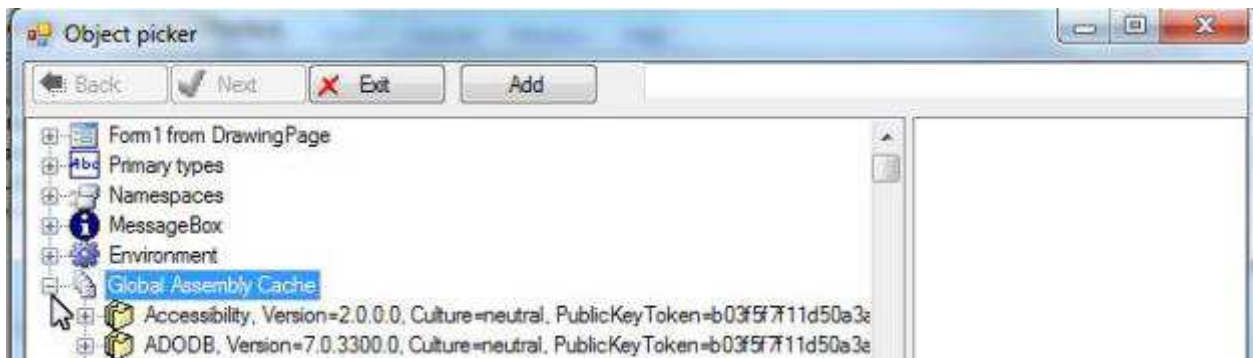
Name it SearchReplace:



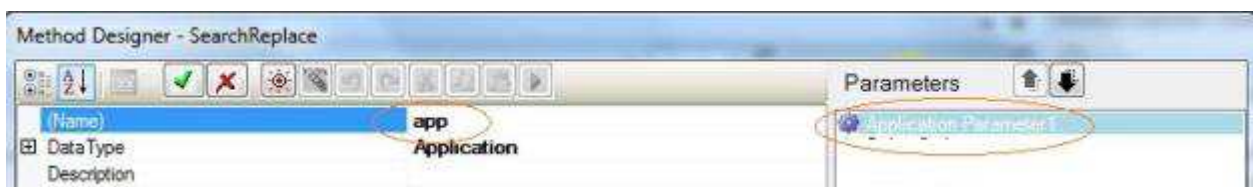
We want to pass the Application object to this method. So, add a parameter and name it “app”:



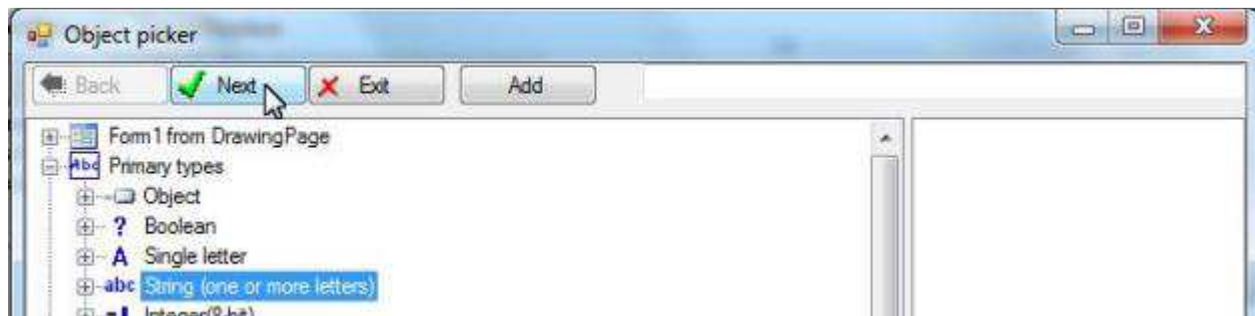
We are asked for the data type for the parameter. Select Word Application from the GAC:



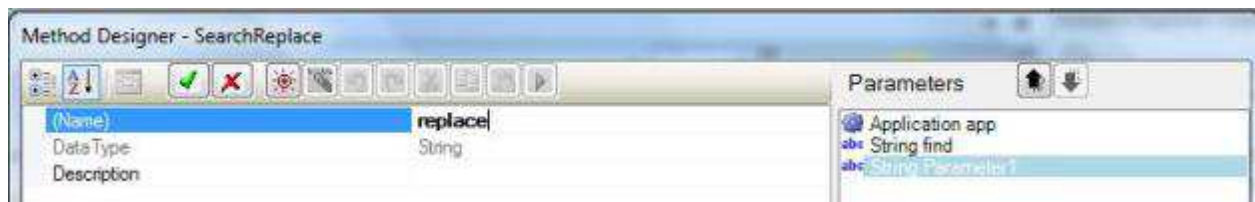
Rename it to "app":



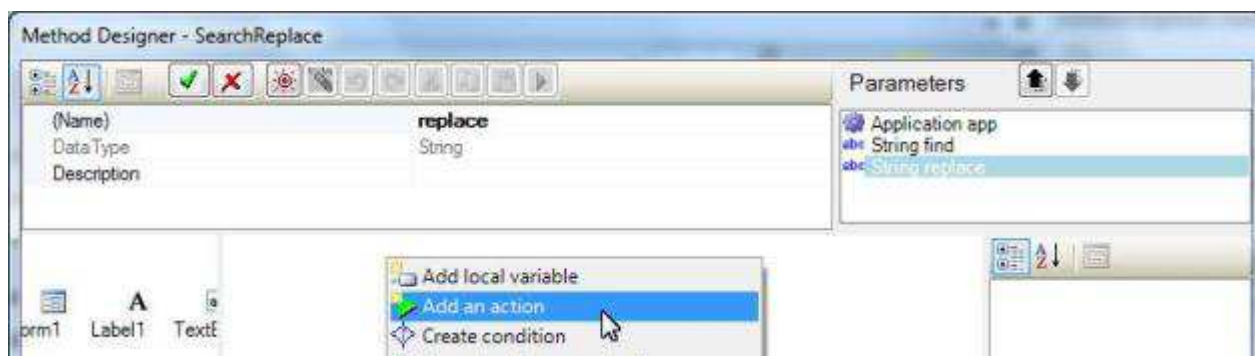
Add two more parameters: find and replace, both are Strings:



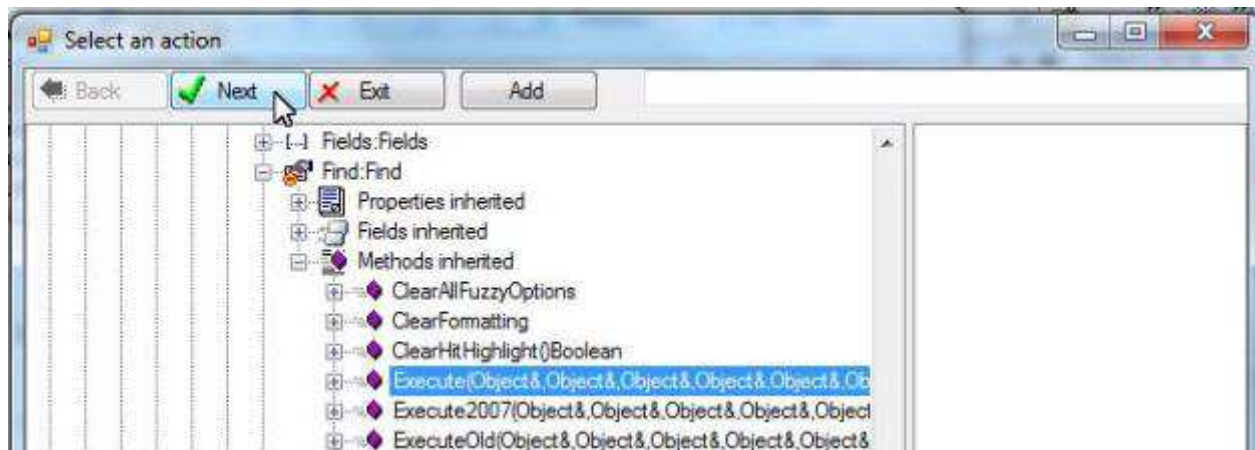
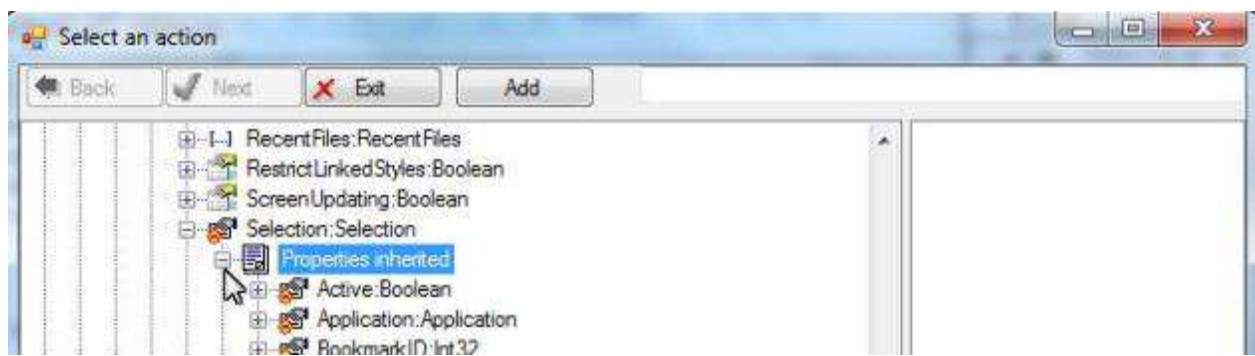
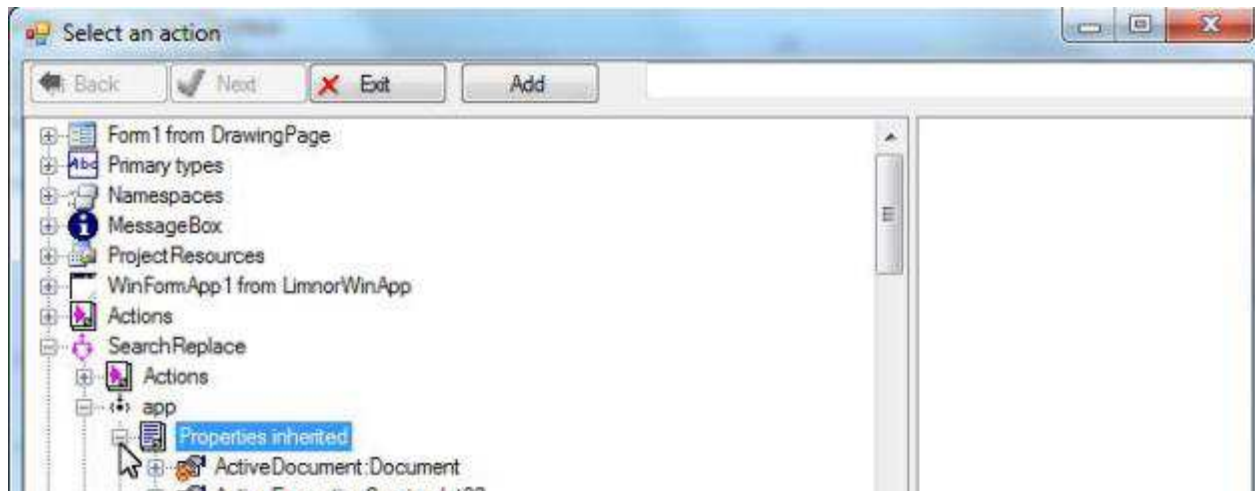
Add "replace":



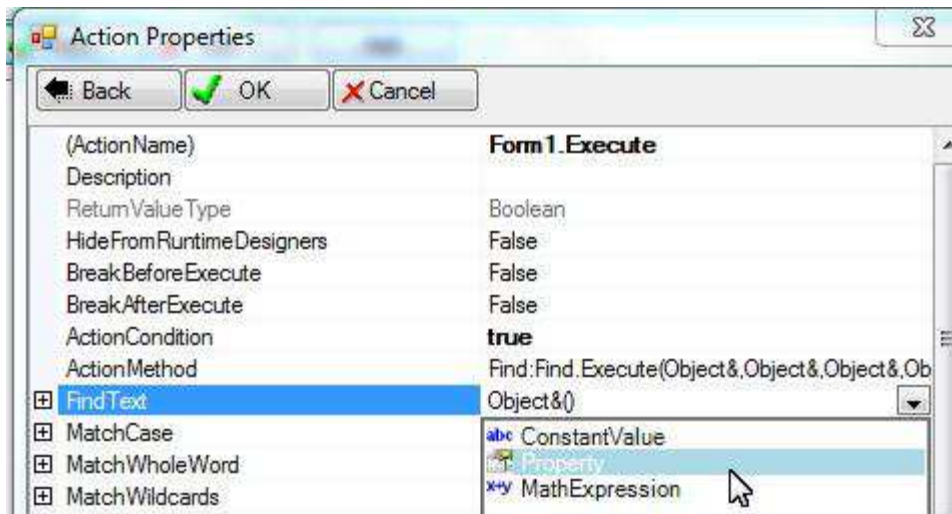
Right-click the Action Pane; choose "Add an action":



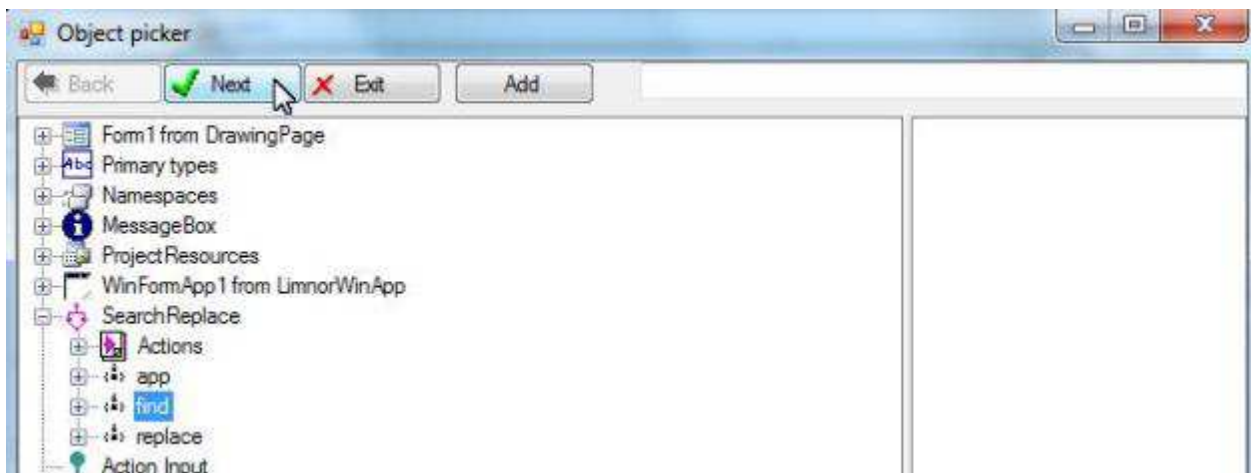
Select Execute method of the Find property of the Selection property of the Word application:



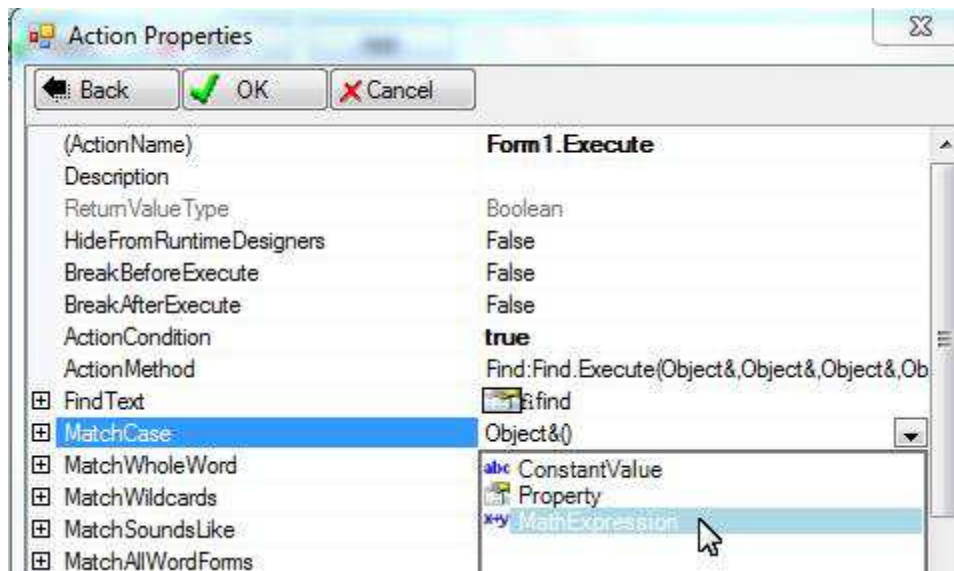
For "FindText" of the action, select "Property":



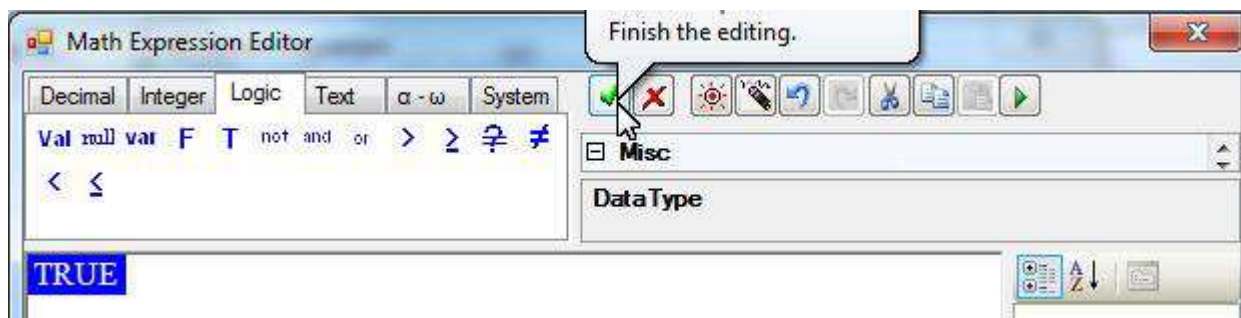
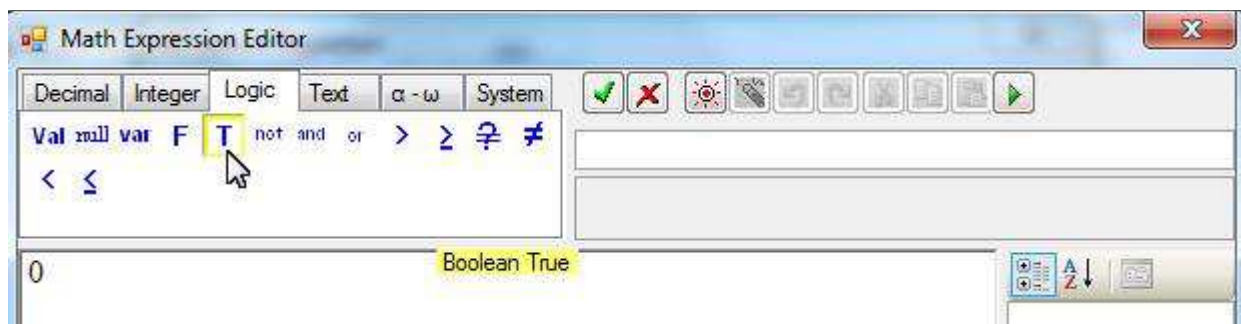
Select "find" parameter of the method; click "Next":



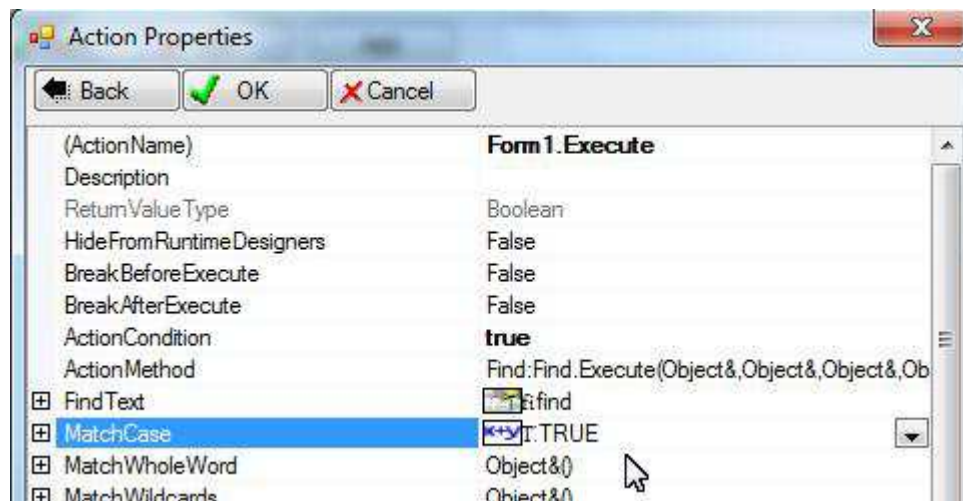
For "MatchCase", select "MathExpression":



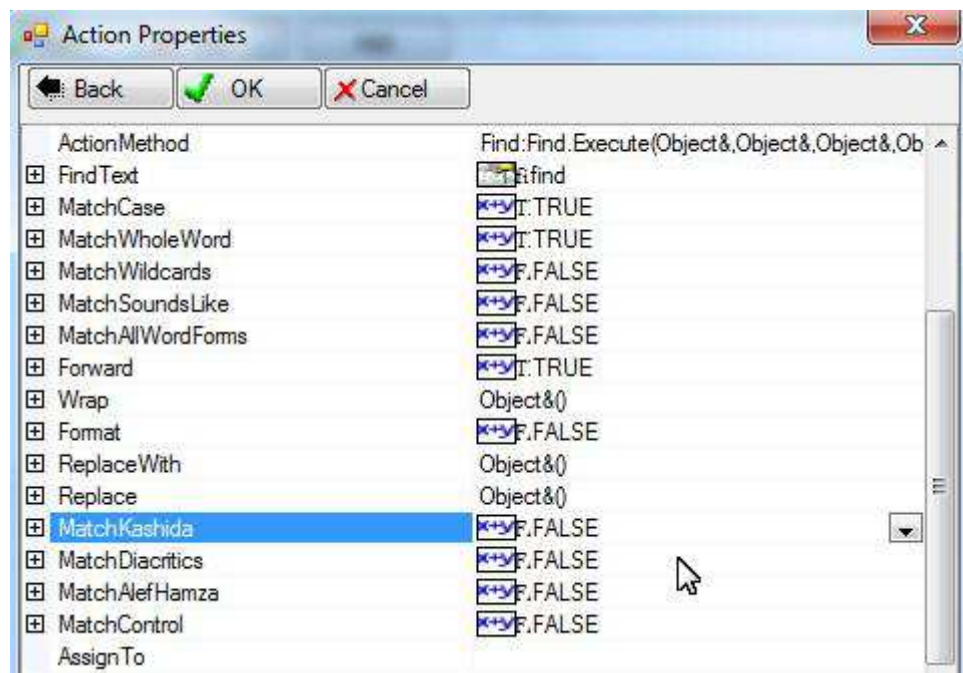
Select logic True:



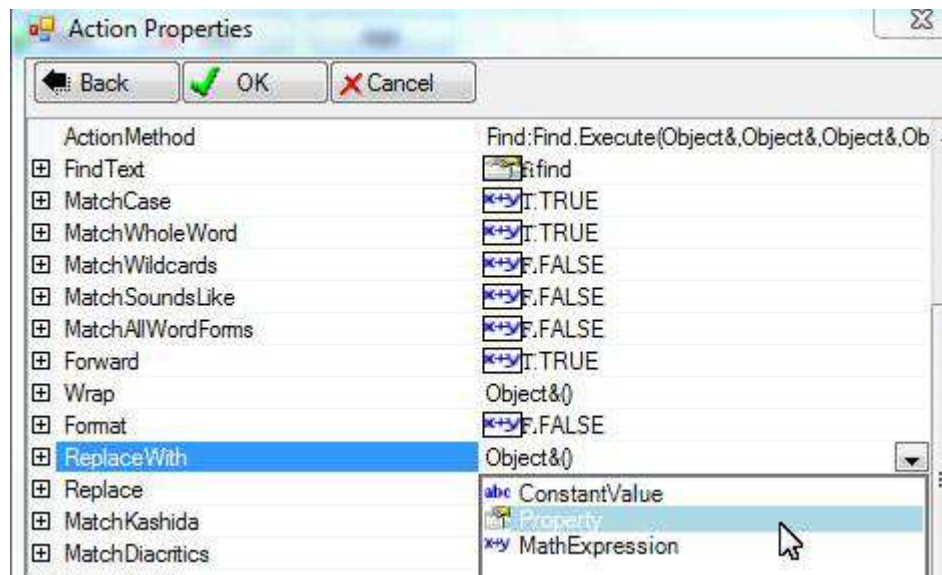
"True" appears as the value for "MatchCase":



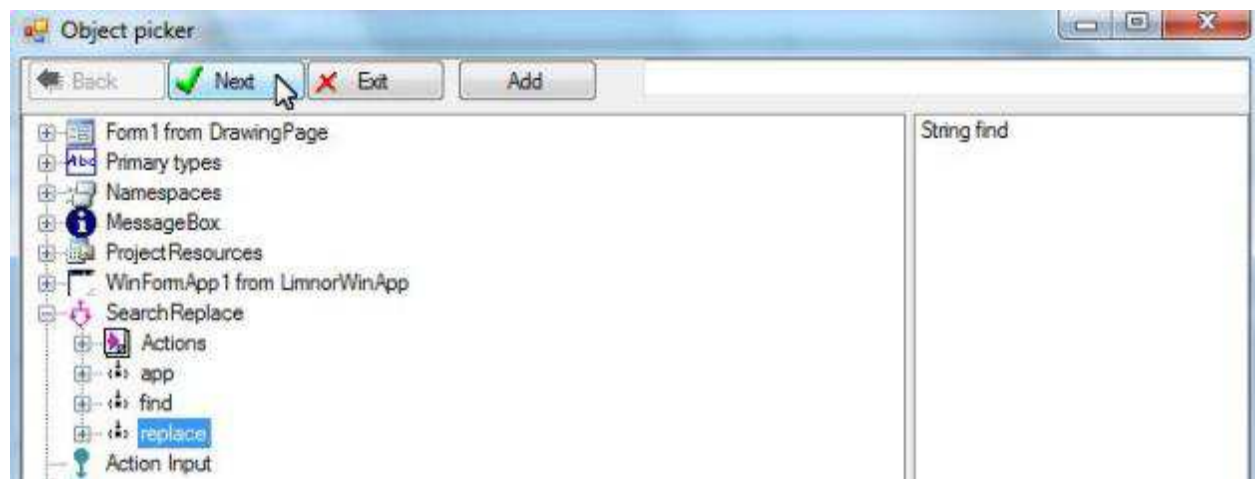
Set other parameters use True or False:



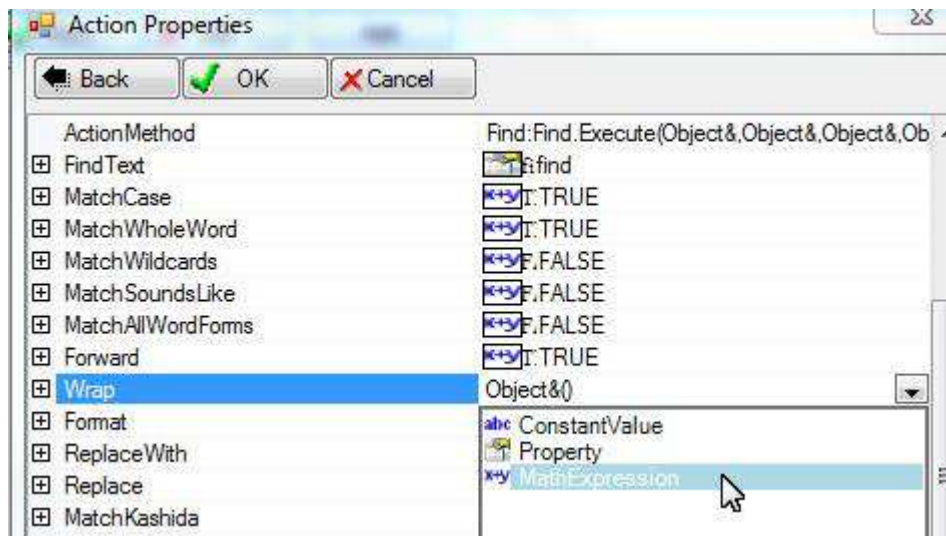
For "ReplaceWith" of the action, select "Property":



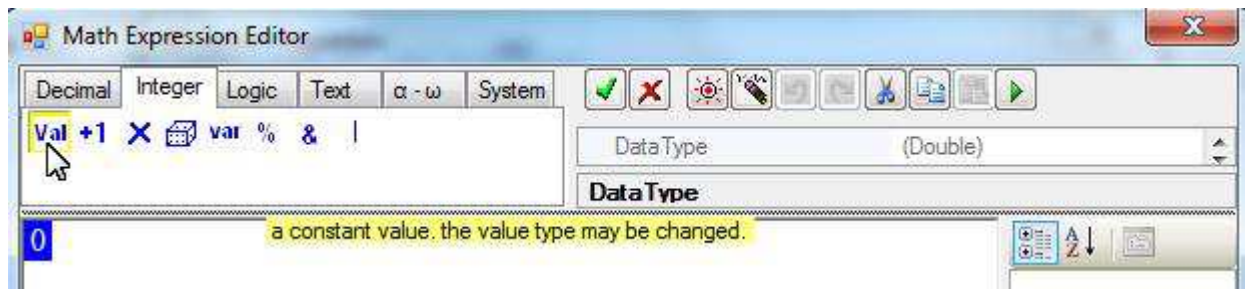
Select "replace" parameter of the method:



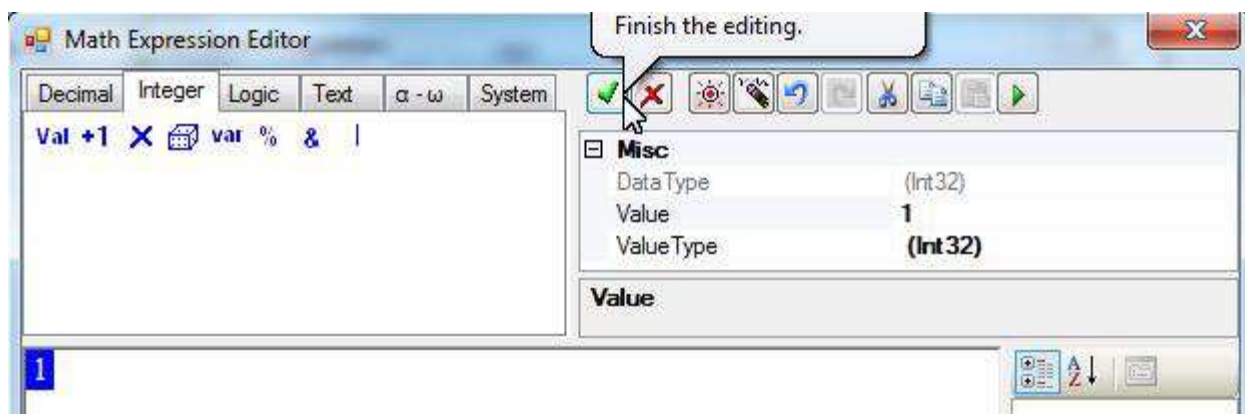
For "wrap", select "MathExpression":

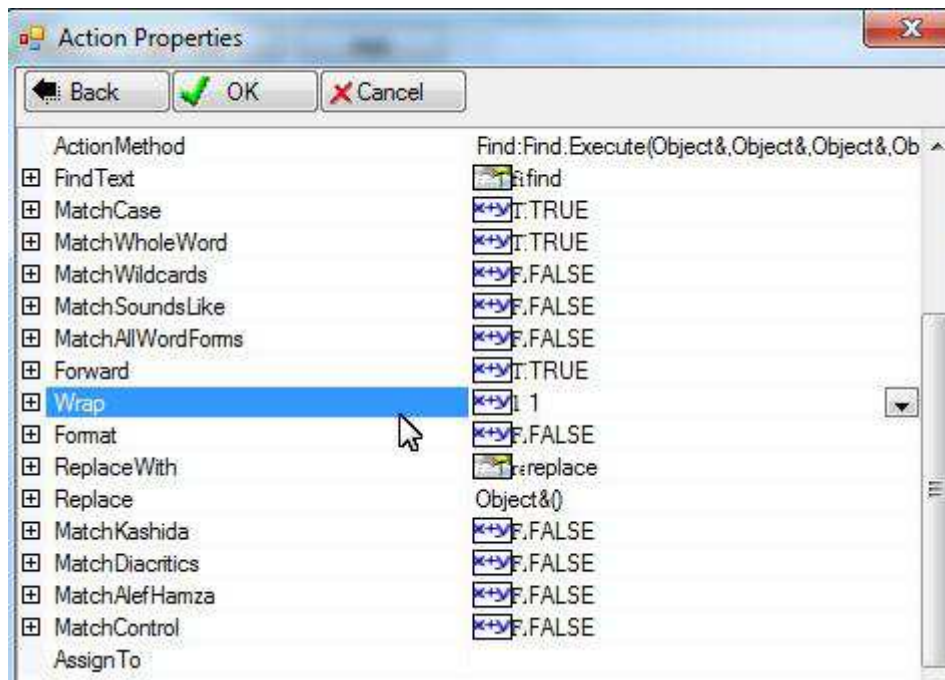


Click integer constant icon:

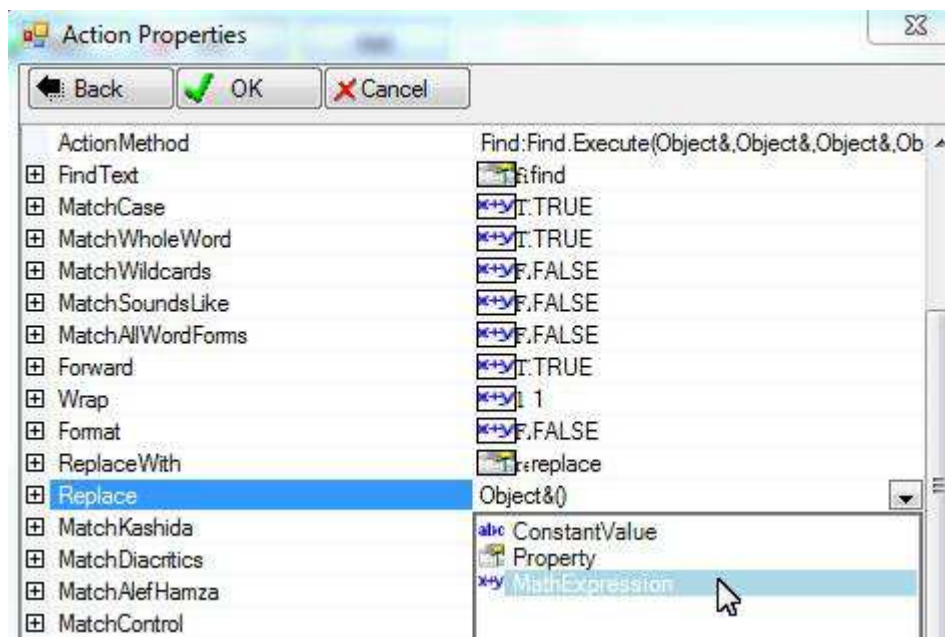


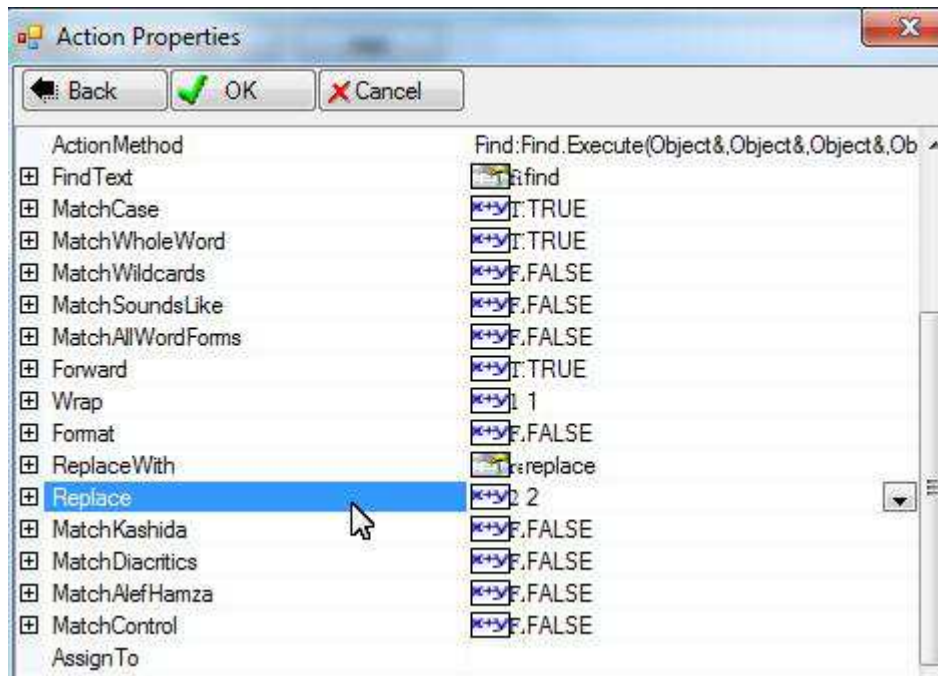
Set Value to 1:





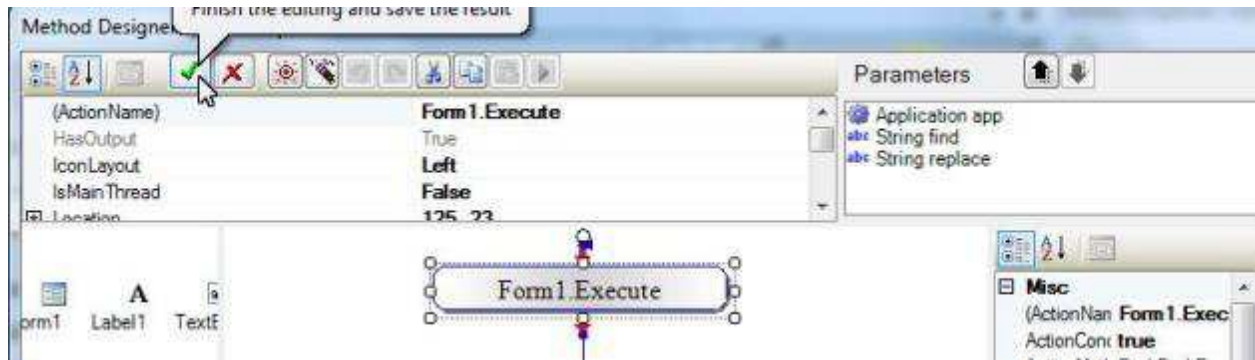
Set "Replace" to 2:





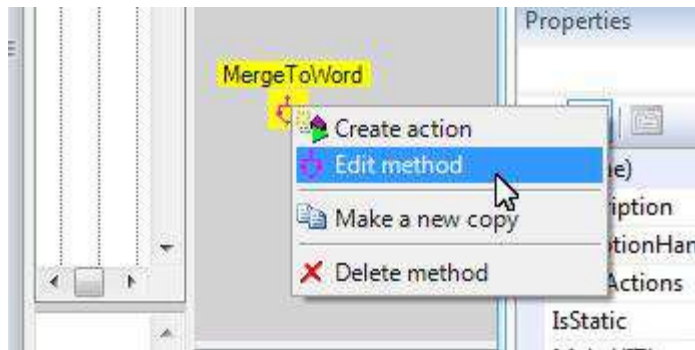
Click OK to finish creating this action.

The action appears in the Action Pane. For this method we just need this one action.

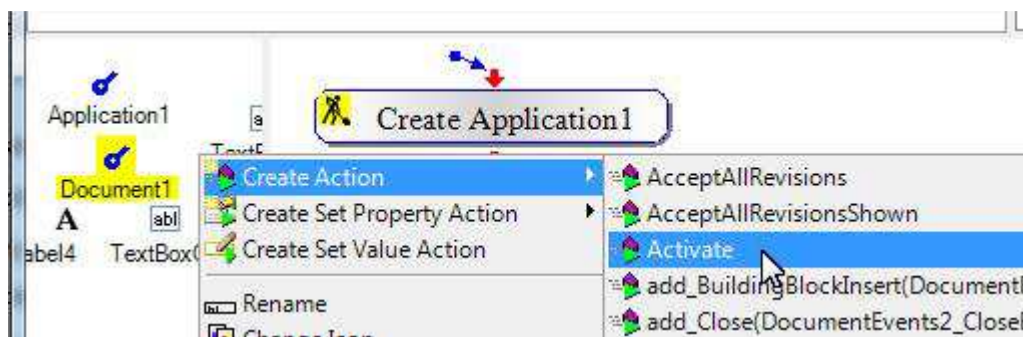


Manipulate Word Document – Modify and Save

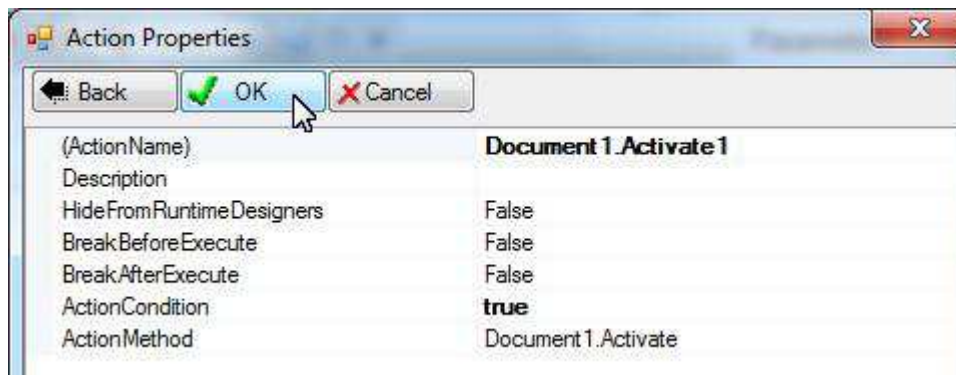
Let's edit the method MergeToWord to add search and replace using text boxes values and save to a new file then close Word document.



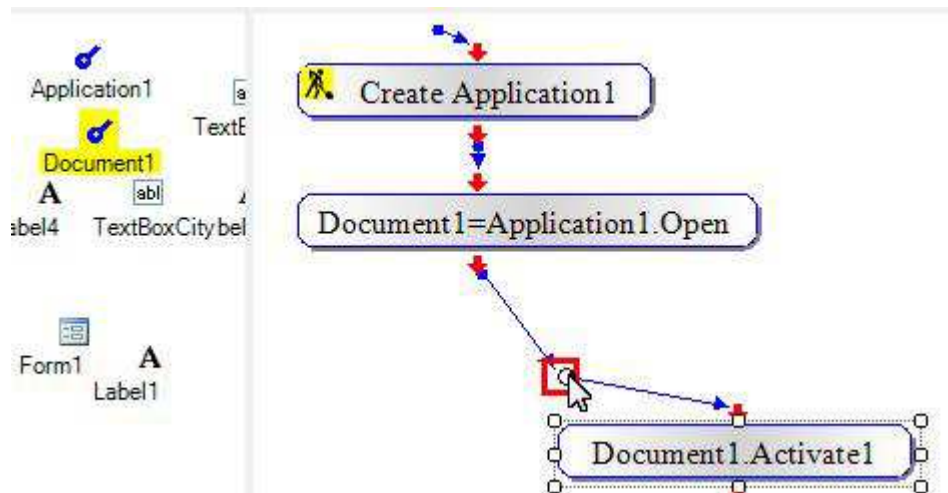
Create an action to activate the newly opened document. Right-click the document object icon; choose “Create Action”; choose “Activate” method:



Click OK:

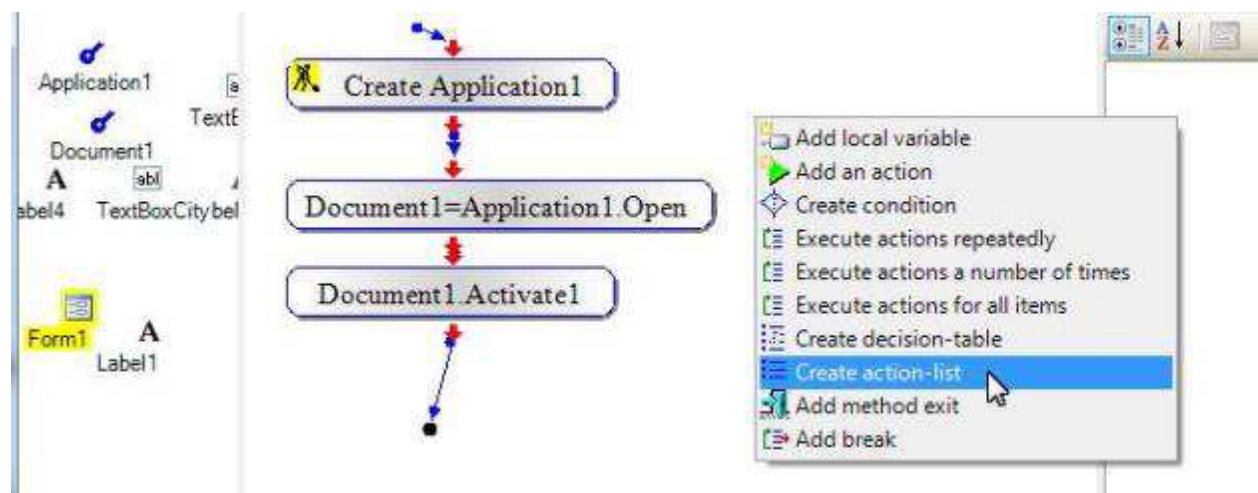


Link it to the last action:

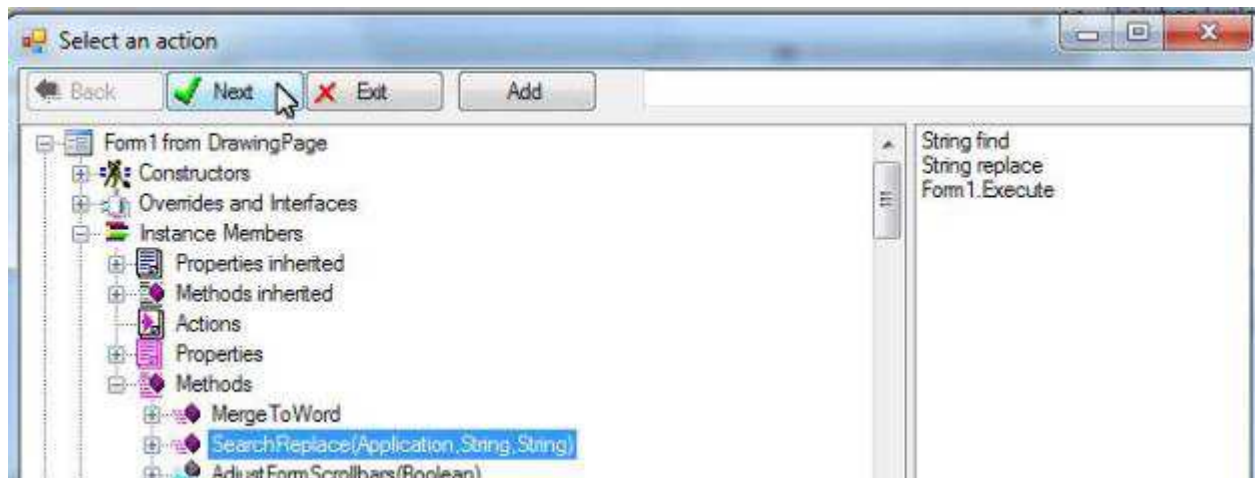


We now may call the SearchReplace method to modify the contents of the document. Because we want to call this method many times by creating many actions, instead of showing all actions individually in the Action-Pane, we may add an action-list to include all such actions.

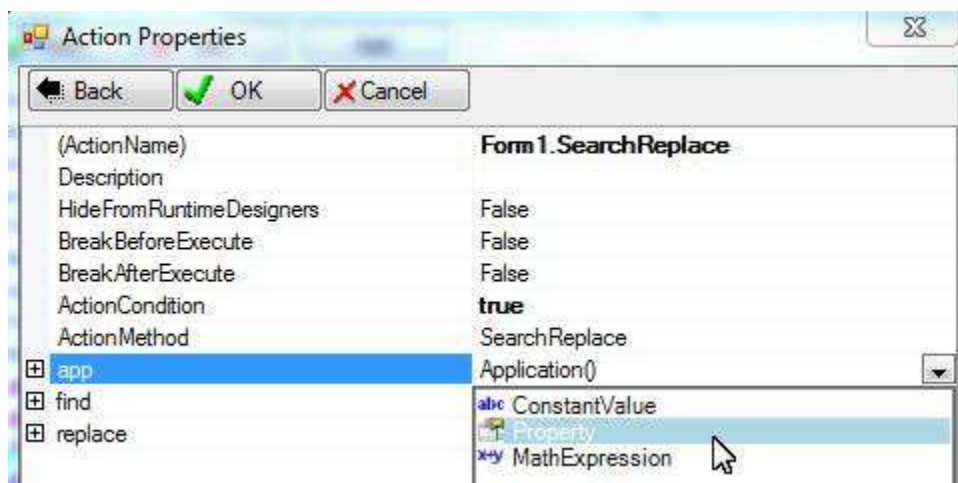
Right-click the Action-Pane; choose “Add action-list”:



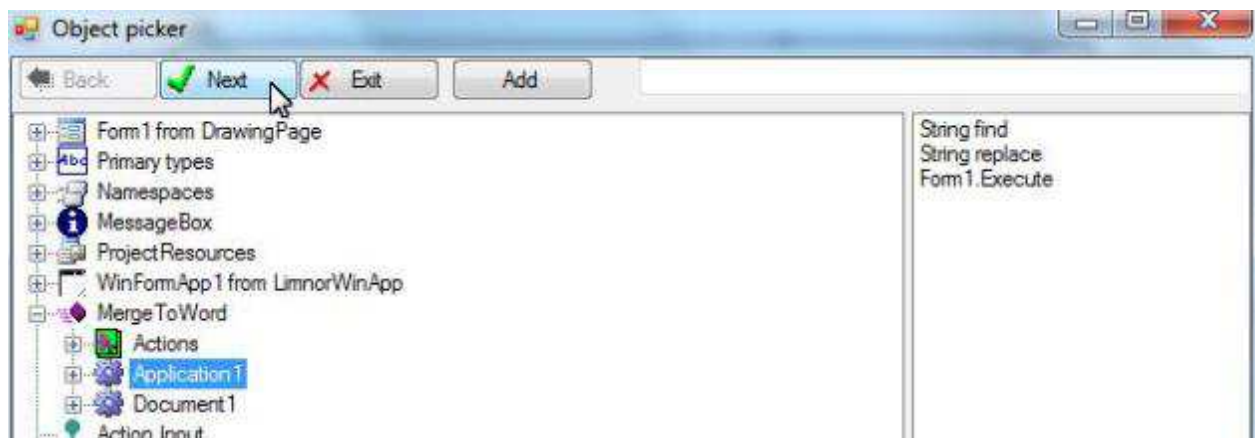
It asks us to select the actions to be included in the action-list. We haven’t created the actions to be included in the action-list. We may create the first action now. Choose “SearchReplace” method; click “Next”:



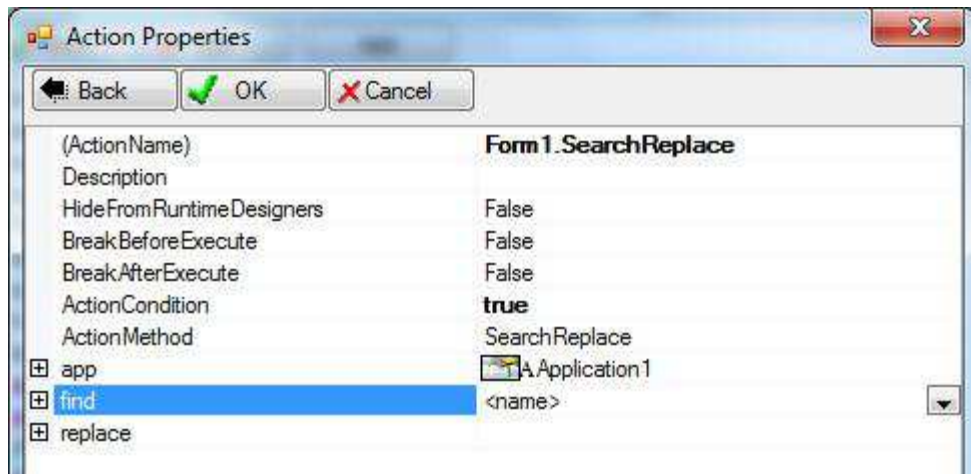
For the “app” parameter of the action, select Property:



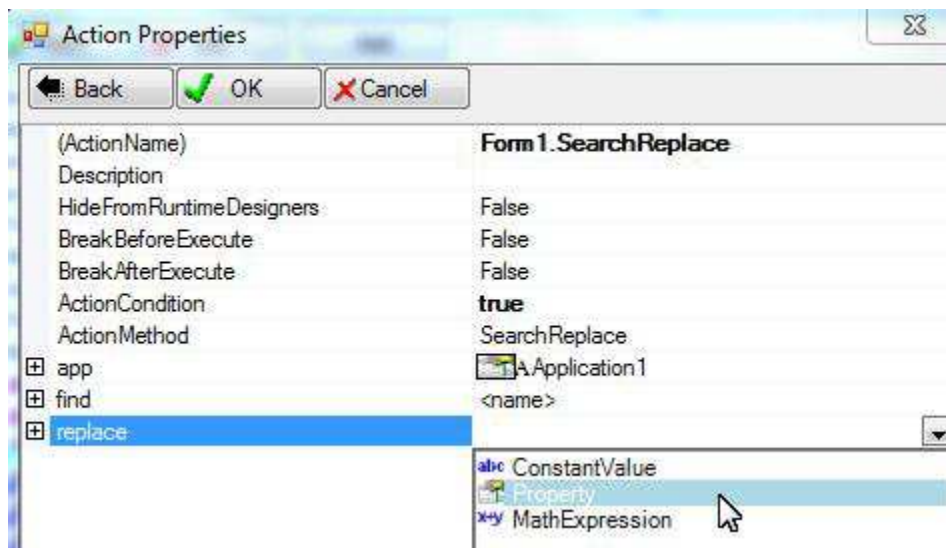
Select the Word Application object:



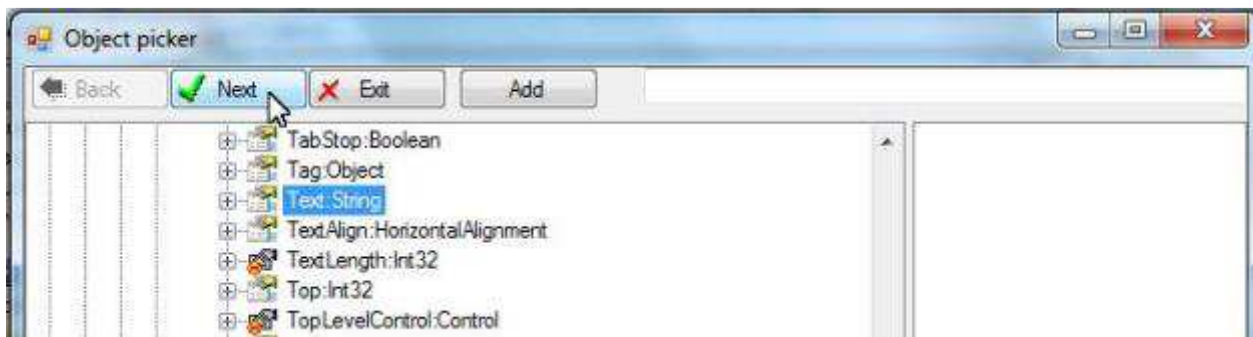
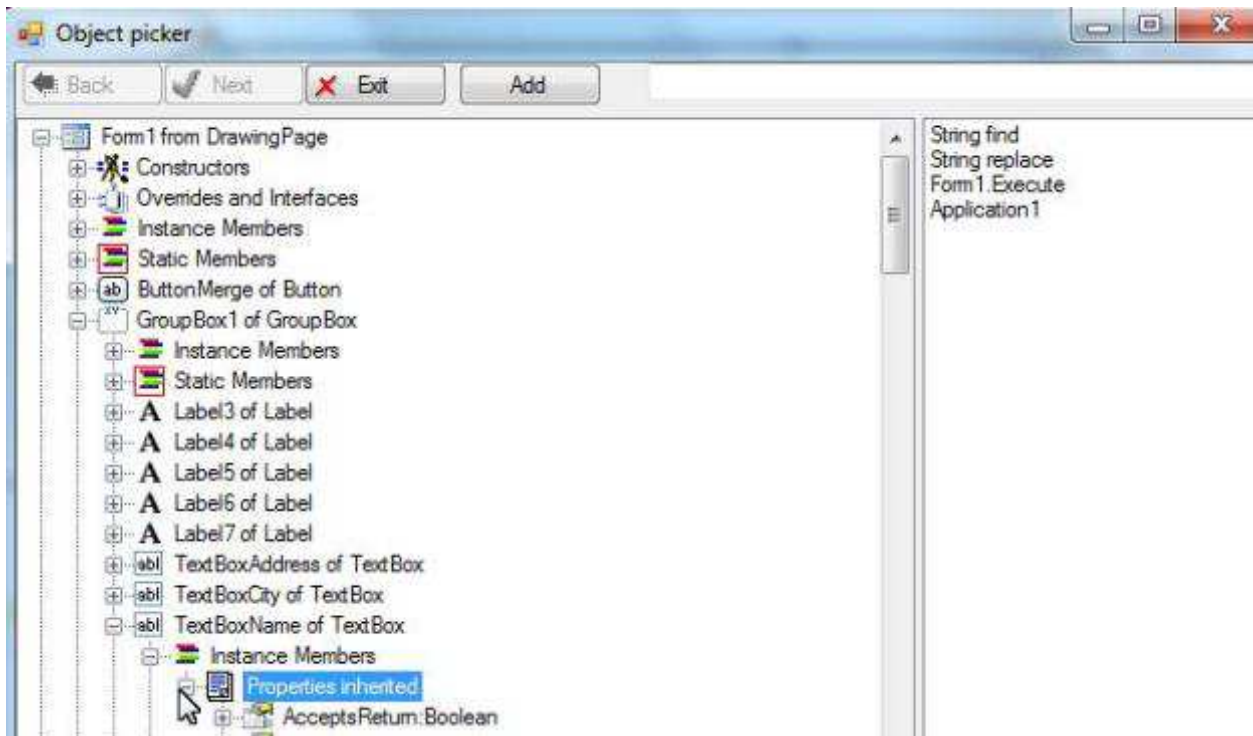
For the “find” parameter of the action, type the text to be replaced. For this action, type “<name>”:



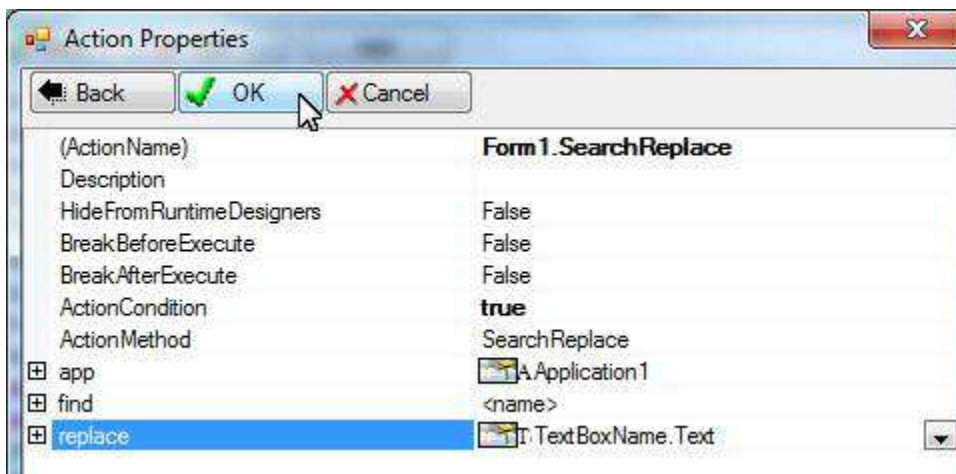
For parameter “replace”, select “Property” because we want to use value from a text box:



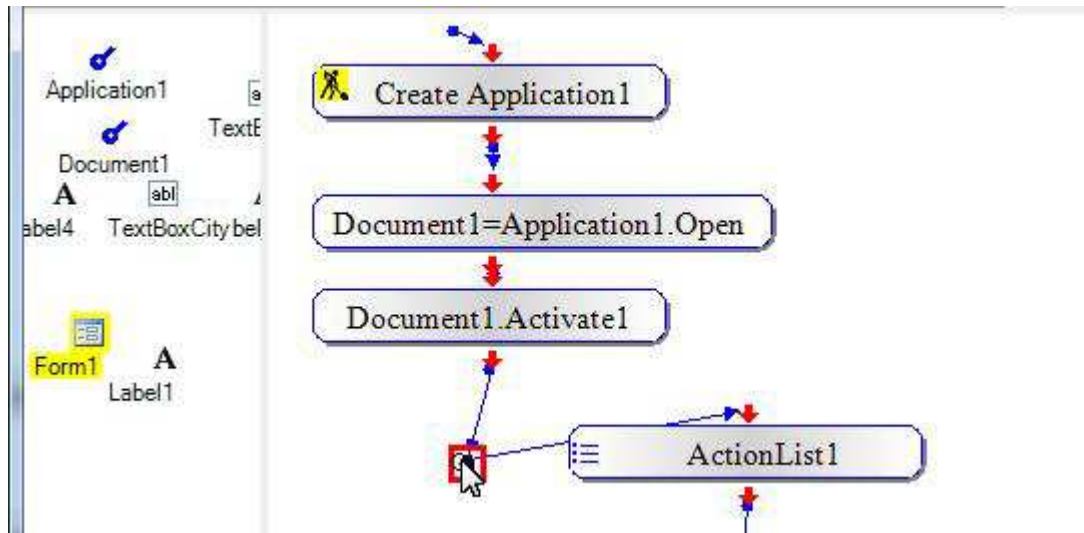
Select the Text property of the text box for name:



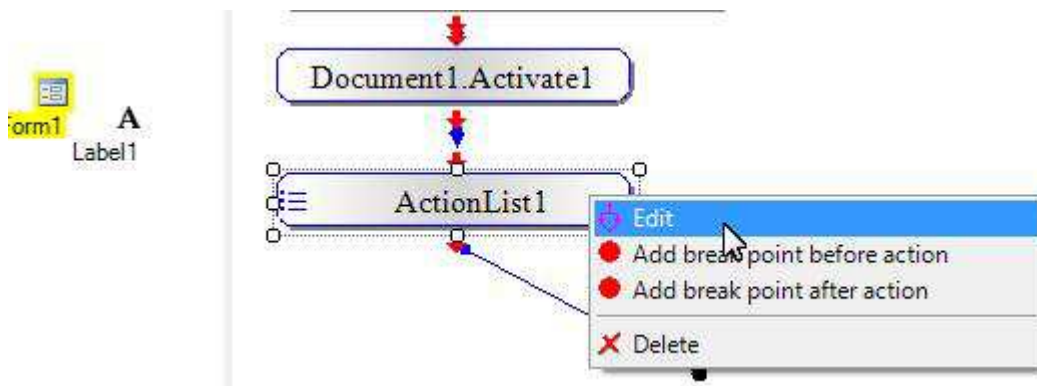
Click OK:



The action-list appears in the Action-Pane. Link it to the previous action:



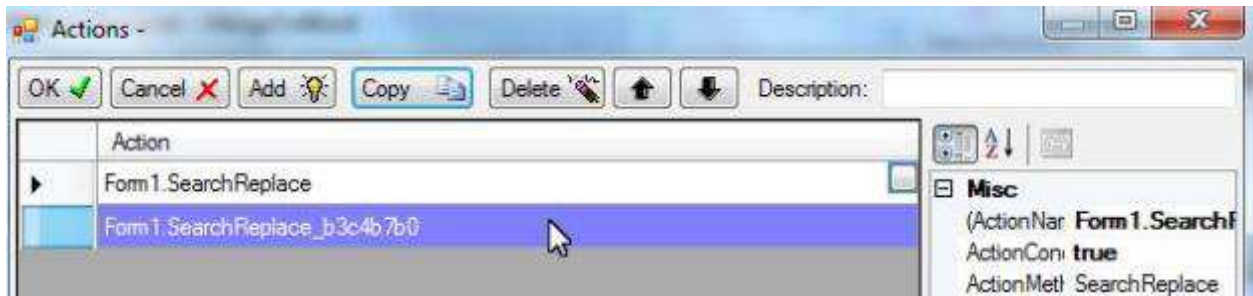
To add more actions into the action-list, right-click the action-list; choose "Edit":



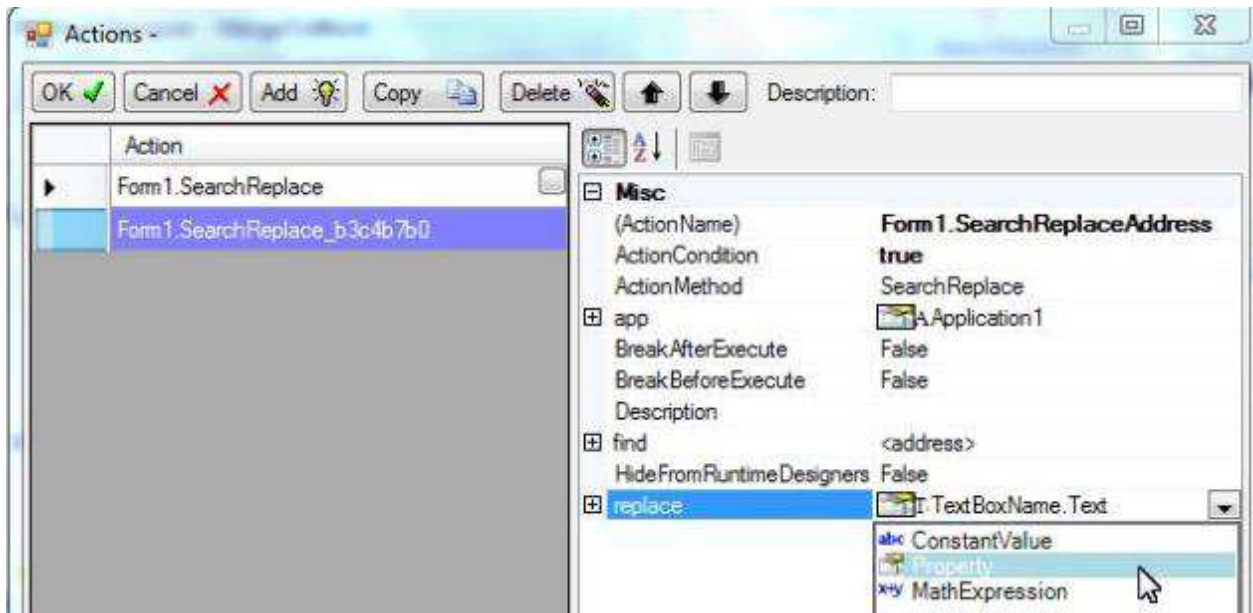
We can see that the search and replace action we created is in the action list. Click "Copy" button to create a new Search and Replace action:



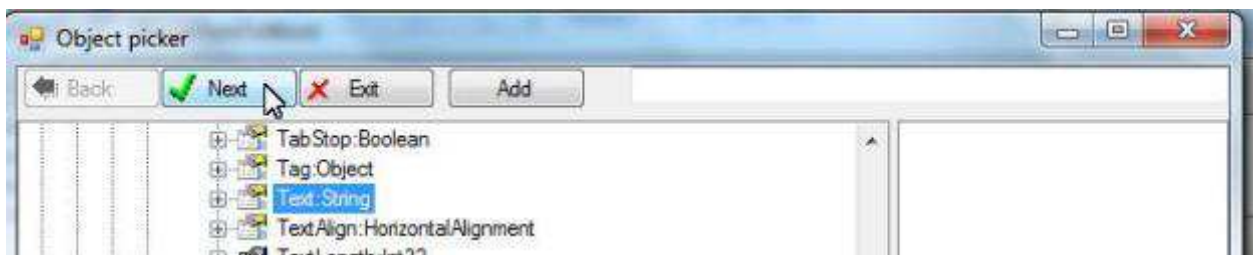
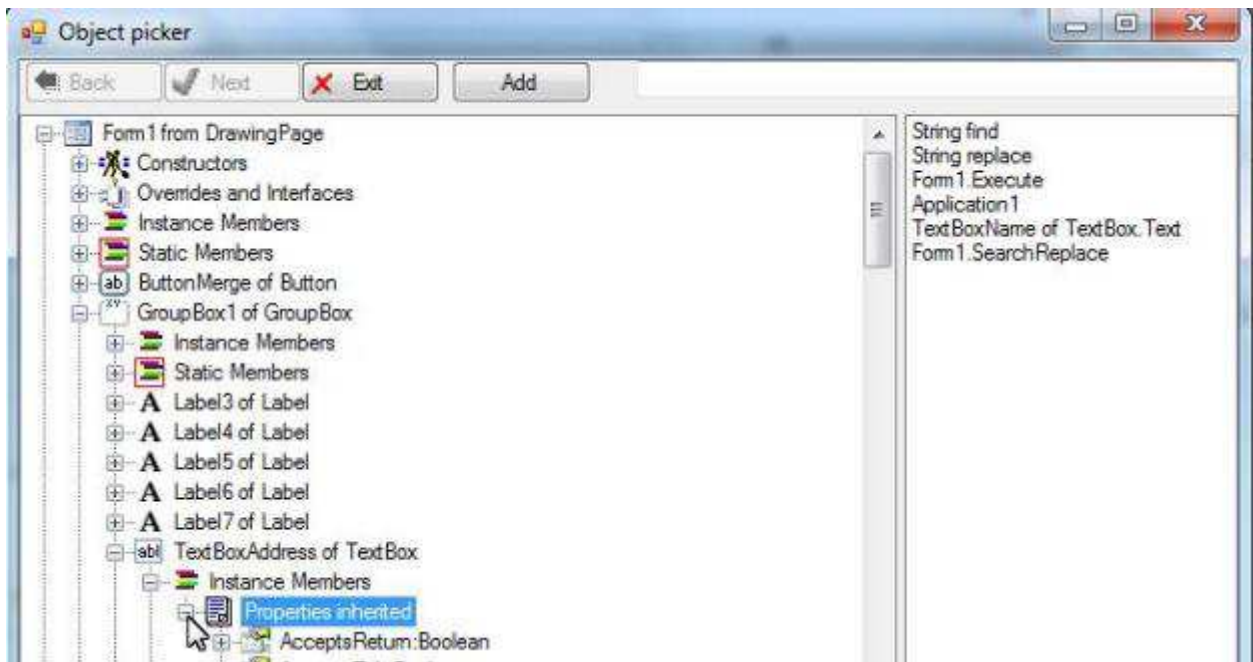
A new action appears in the list:



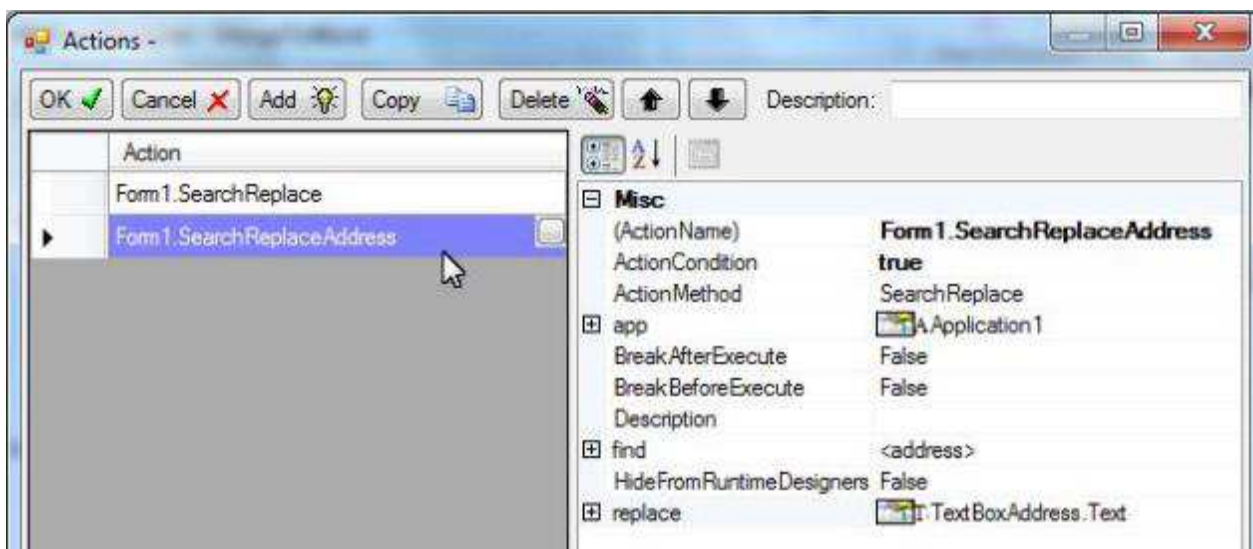
We may rename the action to SearchAndReplaceAddress. Set “find” to <address>. Select “Property” for “replace”:



Select Text property of the text box for address:



The new action will replace <address> with value from the text box:

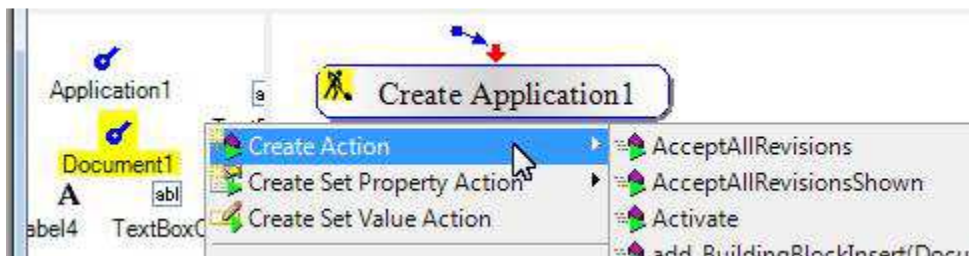


We may copy more actions for other data to be replaced:

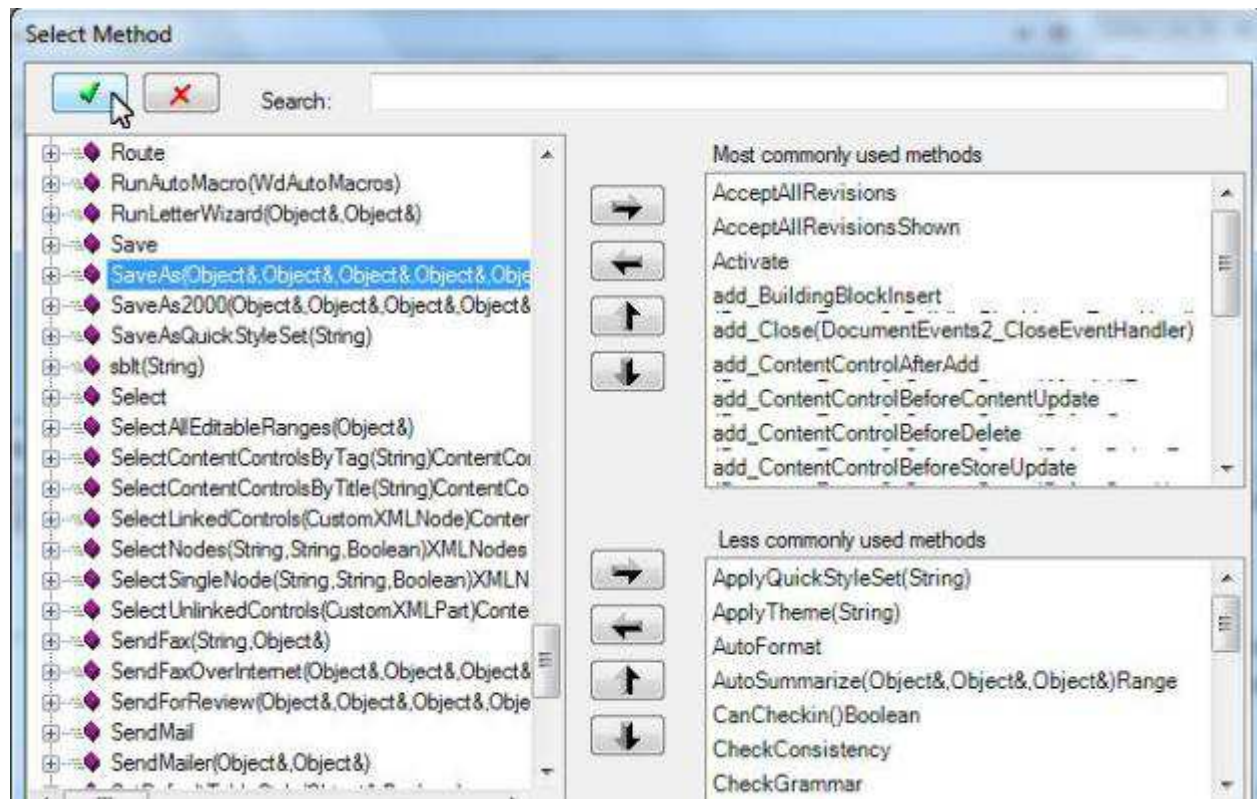


We may create actions to save and close the document.

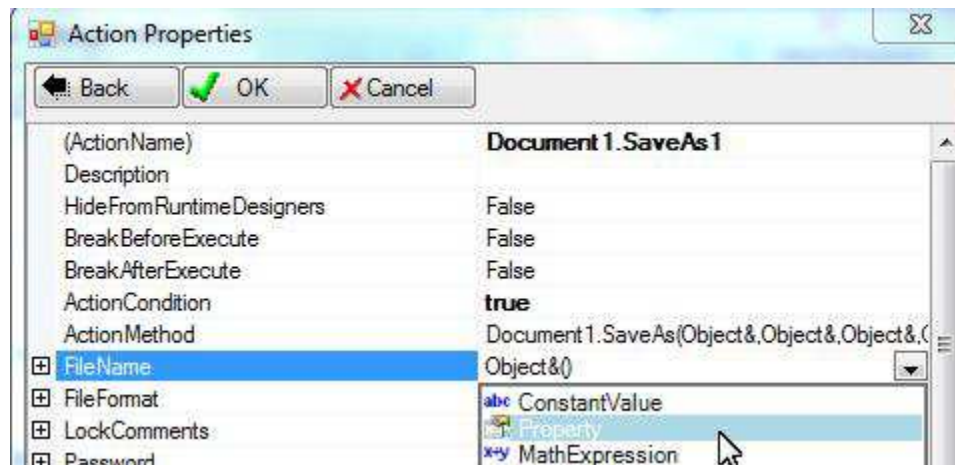
Right-click the document object icon; choose "Create action"; choose "More methods"; choose "*All methods* =>":



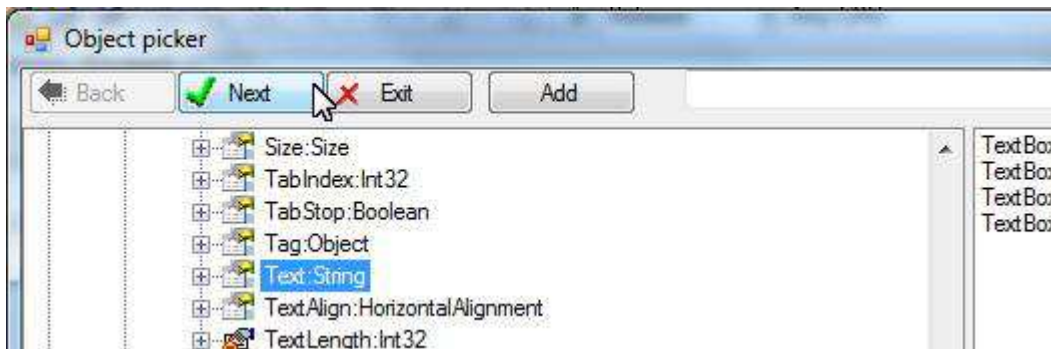
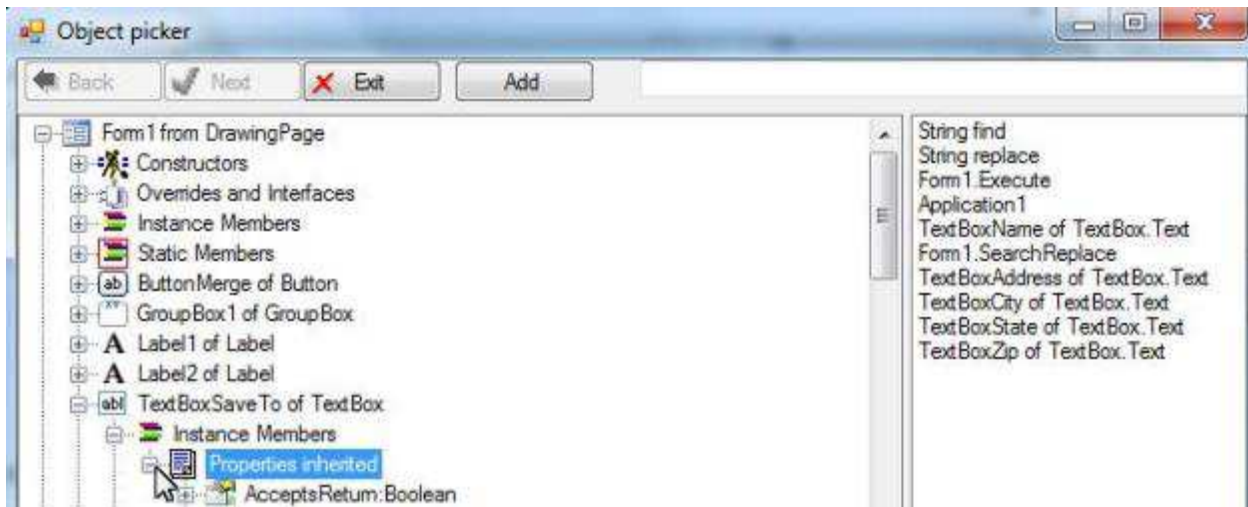
Select "SaveAs" method:



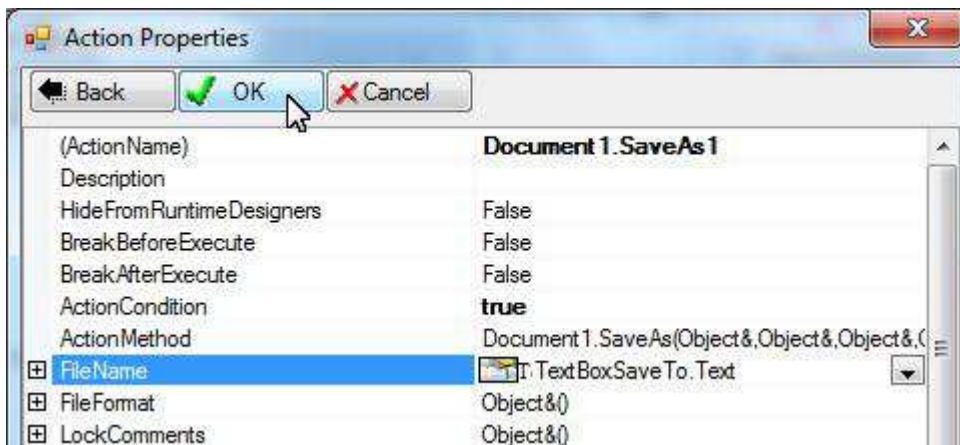
Select "Property" for the "FileName" parameter:



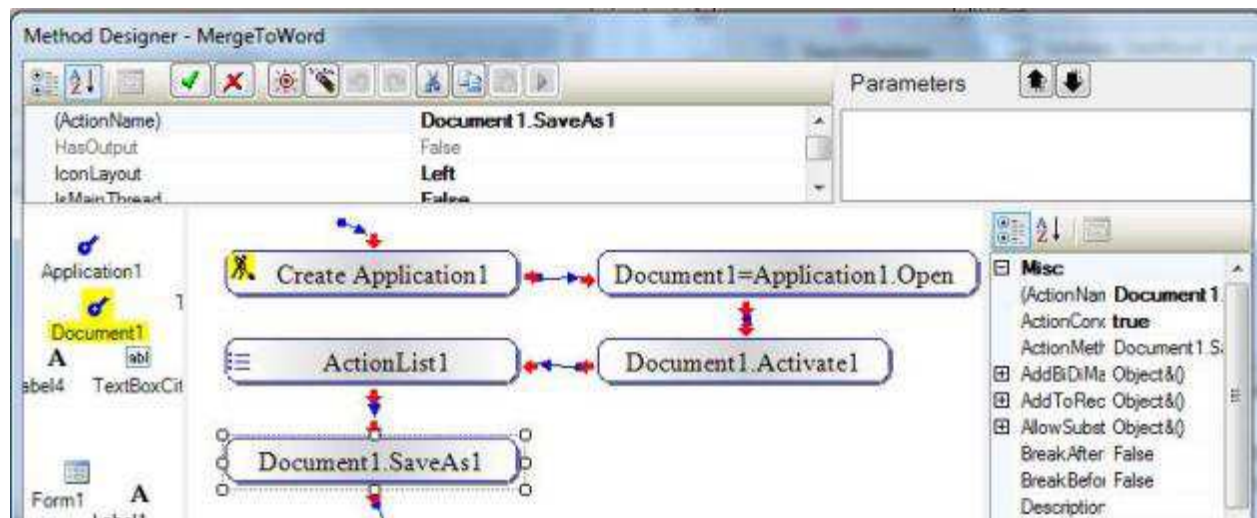
Select Text property of the text box for target file path:



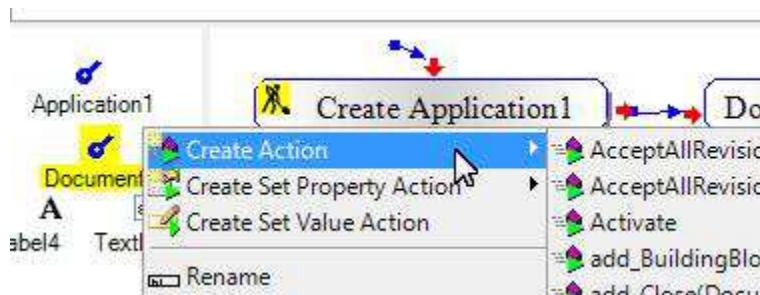
Click OK:



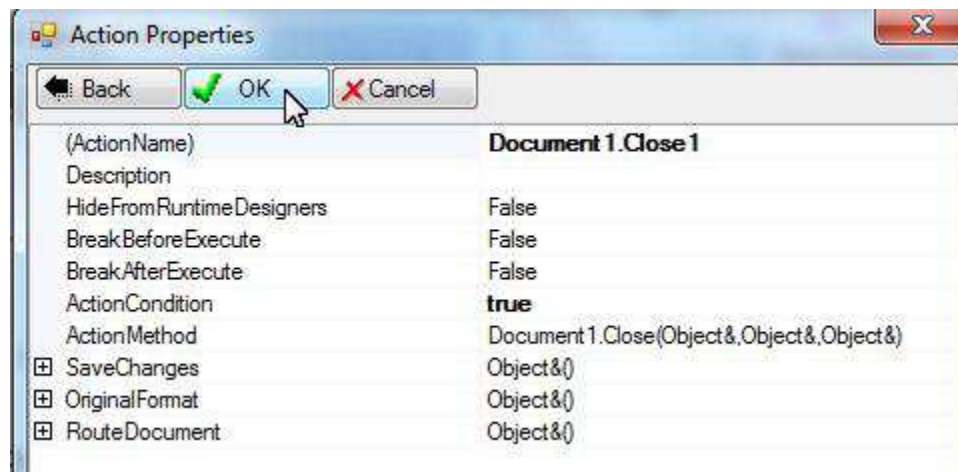
Link the action to the last action:



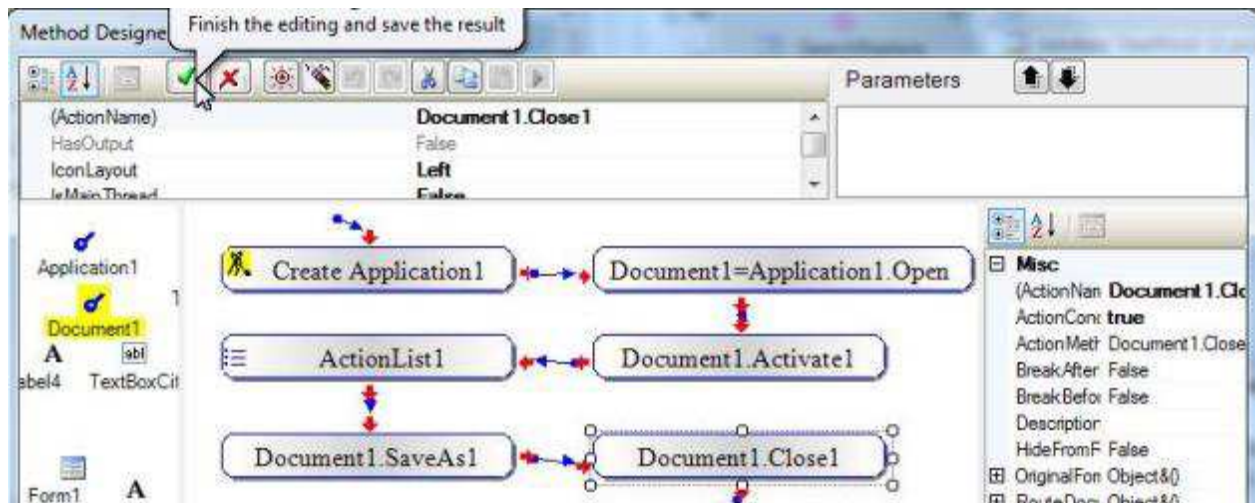
To create a Close document action, right-click the document object icon; choose "Create action"; choose "More actions"; choose "Close" method.



Click OK:

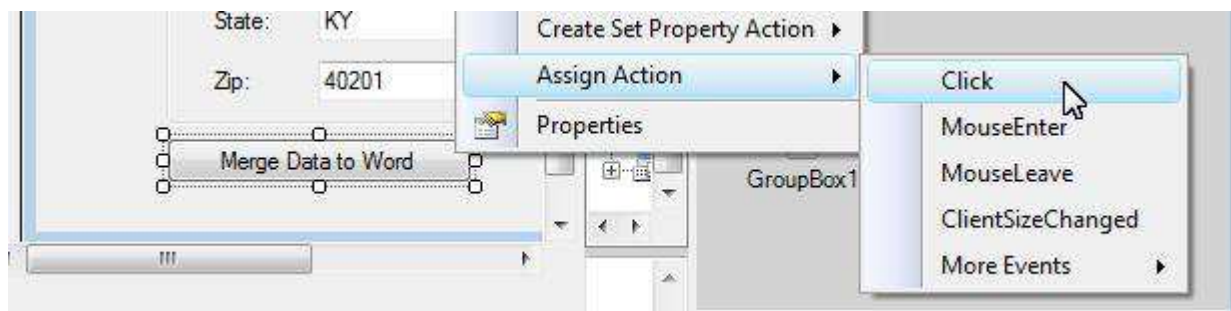


Link it to the last action. We are done creating this method:



Execute Method

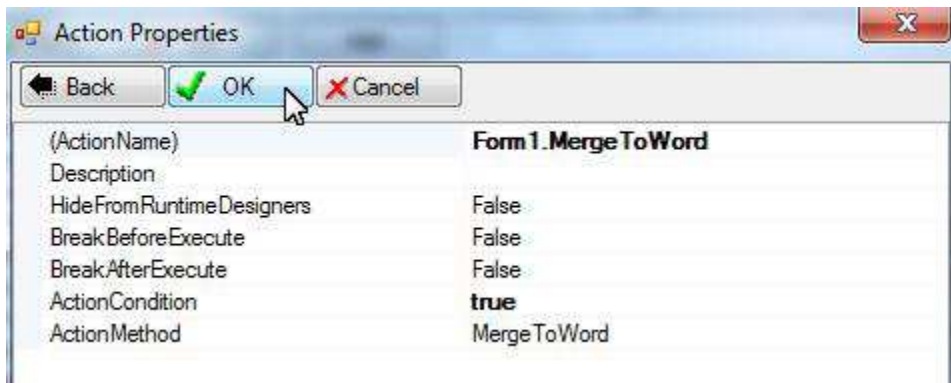
Right-click the button; choose "Assign action"; choose "Click" event:



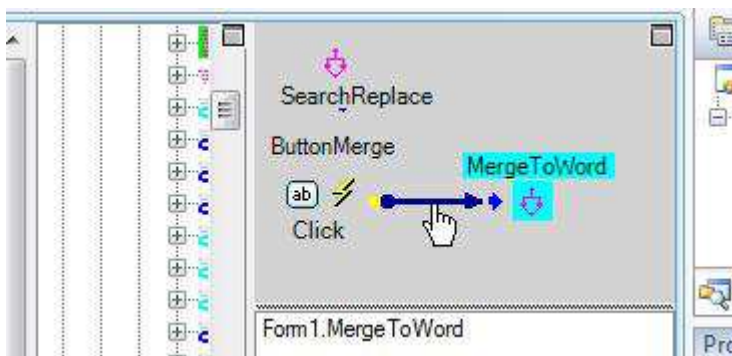
Select MergeToWord method:



Click OK:

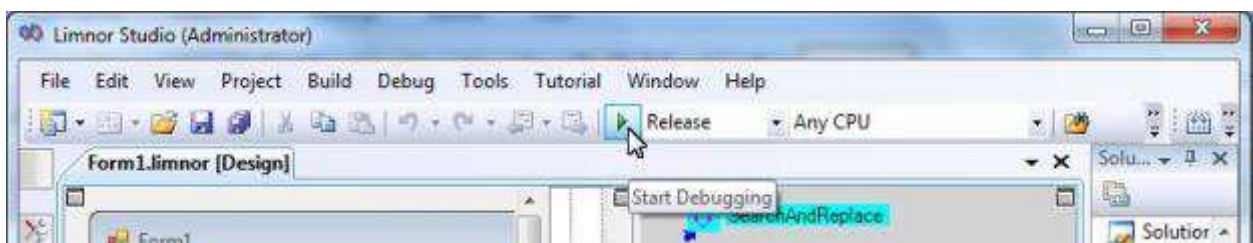


An action is created and assigned to the button:



Test

Click the Run button to compile and run the sample.

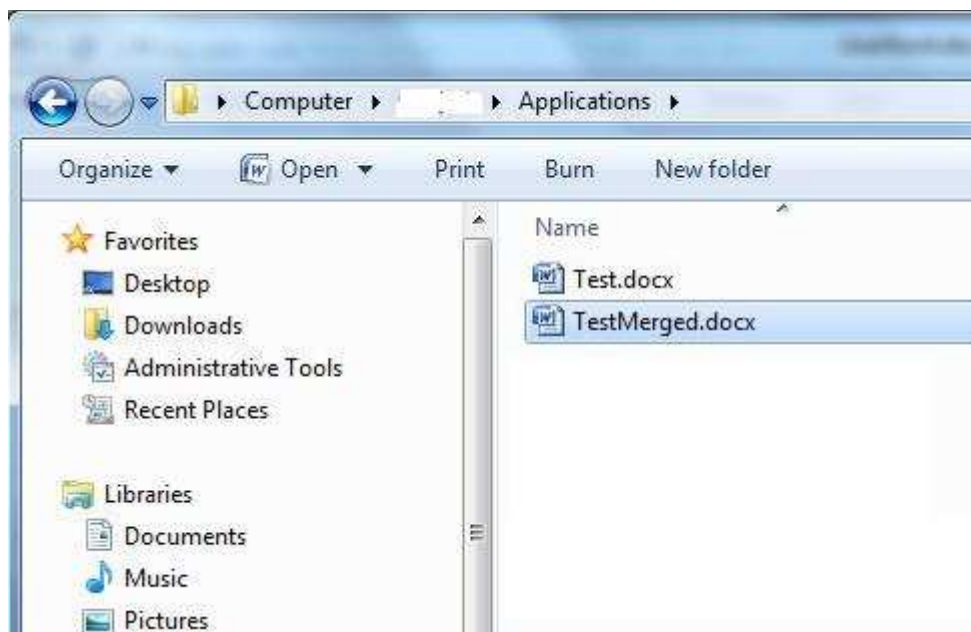


The form appears. Click the button.

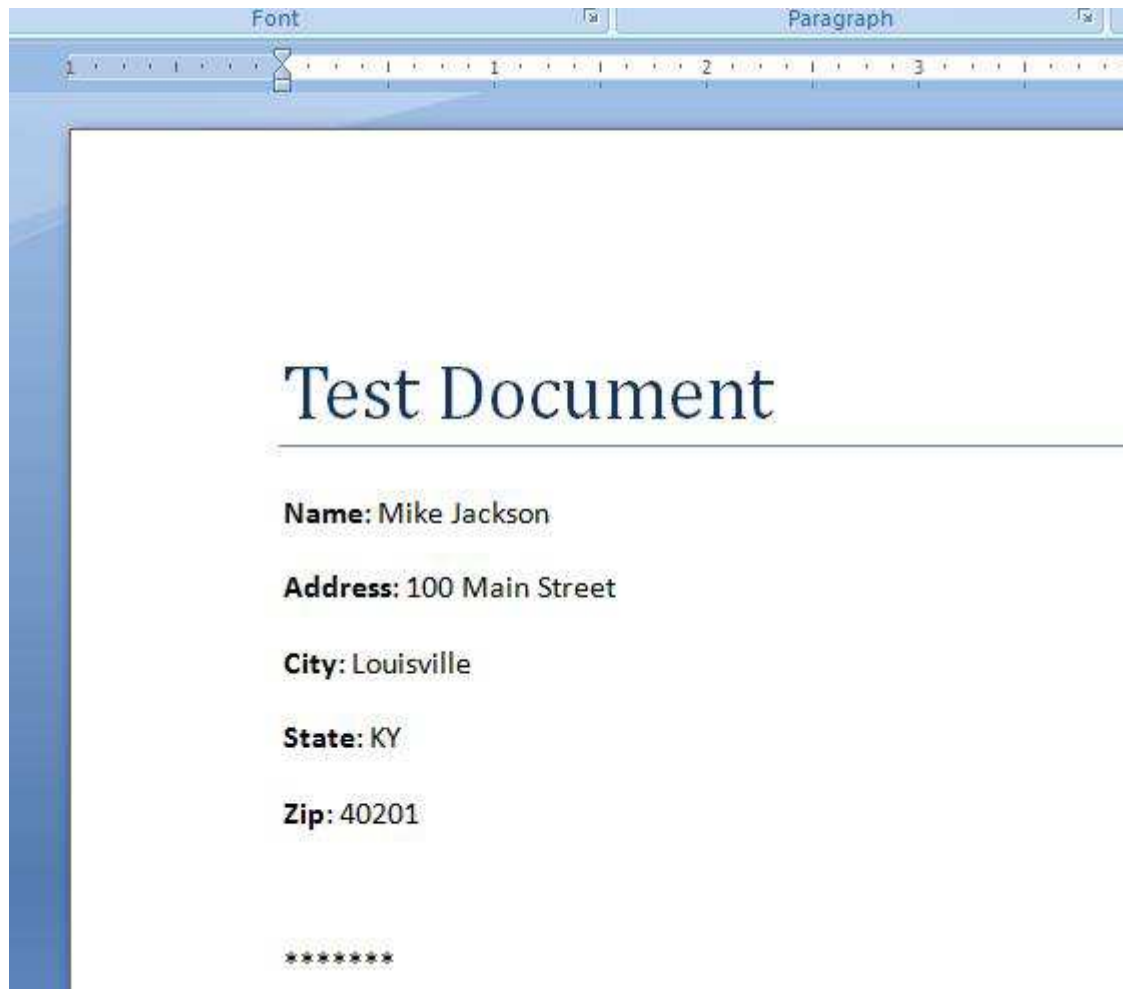
The screenshot shows a window titled "Form1" with the following elements:

- Word File:** A text box containing "C:\Applications\Test.docx".
- Save to:** A text box containing "C:\Applications\TestMerged.docx".
- Data to be merged into WORD:** A group box containing five text boxes:
 - Name:** "Mike Jackson"
 - Address:** "100 Main Street"
 - City:** "Louisville"
 - State:** "KY"
 - Zip:** "40201"
- Buttons:** A single button at the bottom labeled "Merge Data to Word" with a mouse cursor hovering over it.

We can see that the "Save to" file, TestMerged.docx, is created:



Open this document file; we can see that the data from the form are merged into the document:



So, the sample works as we expected.

Feedback

Please send your feedback to support@limnor.com