

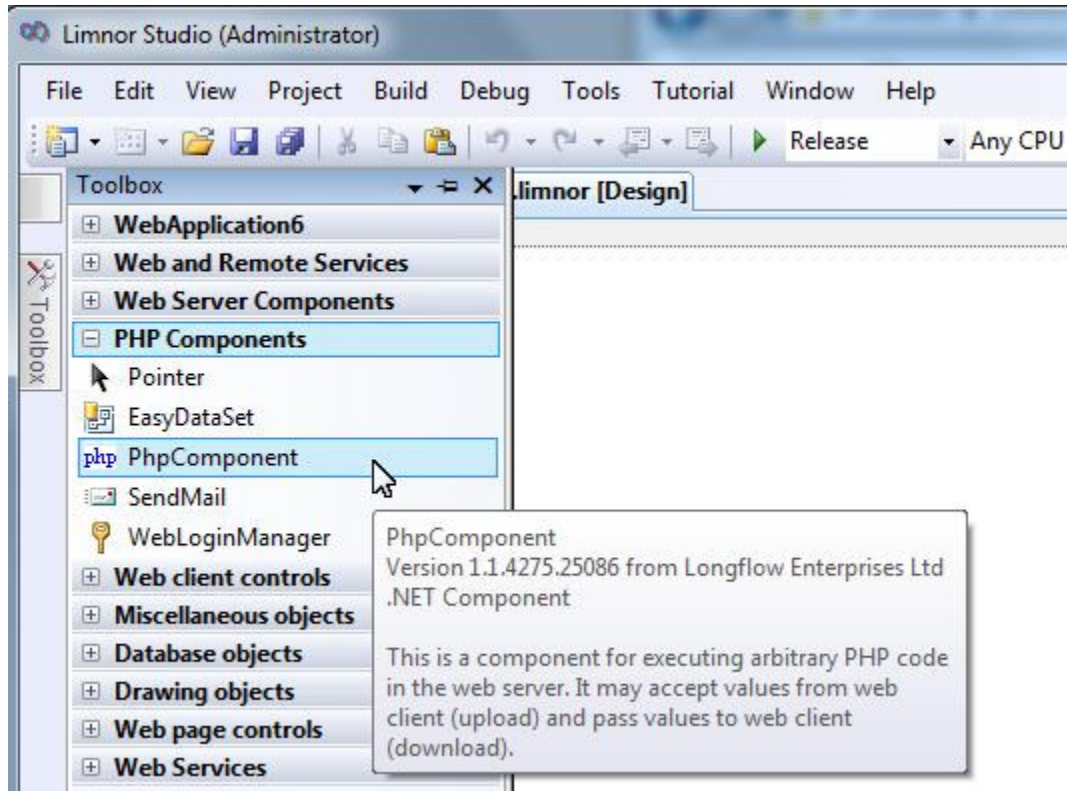
Use Arbitrary PHP Code

Contents

Introduction	1
Execute Arbitrary PHP Code.....	2
Data Exchange between Client and Server.....	6
Create values.....	6
Use value in client.....	7
Use value in server	11
Use PHP Files.....	18

Introduction

For PHP Web Applications, Limnor Studio compiler will generate PHP code for the visual and codeless programming. But there are useful PHP codes you may get from somewhere. You may use arbitrary PHP codes in your PHP web projects through a PhpComponent object.

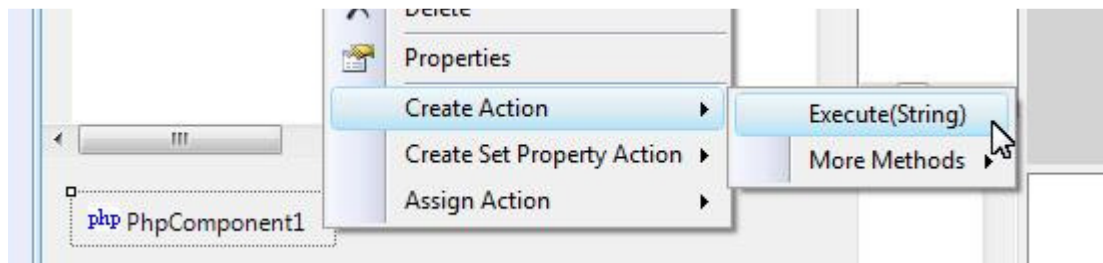


Drop PhpComponent to a web page to start using it.

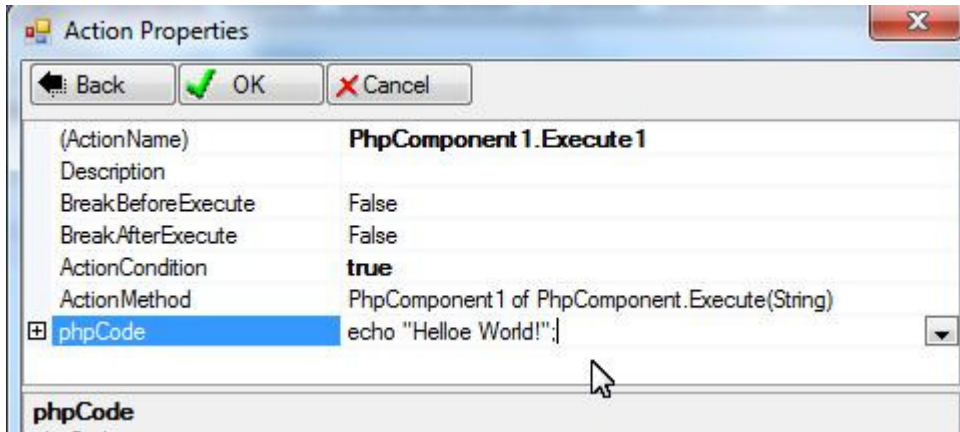
Execute Arbitrary PHP Code

The PhpComponent has an Execute method. The method has a parameter to specify arbitrary PHP code to be executed. We may use this method to create actions to execute various PHP codes.

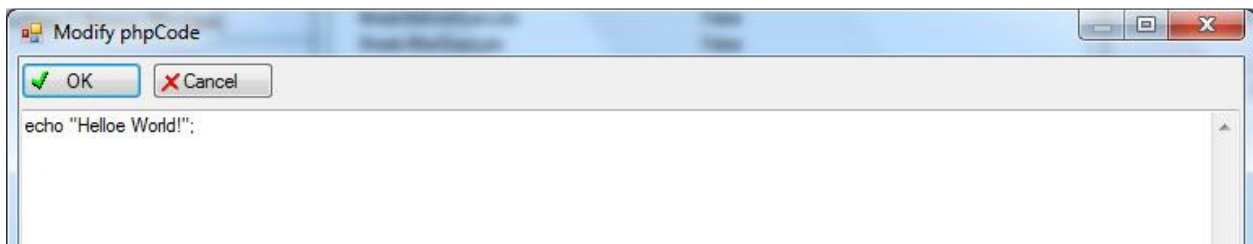
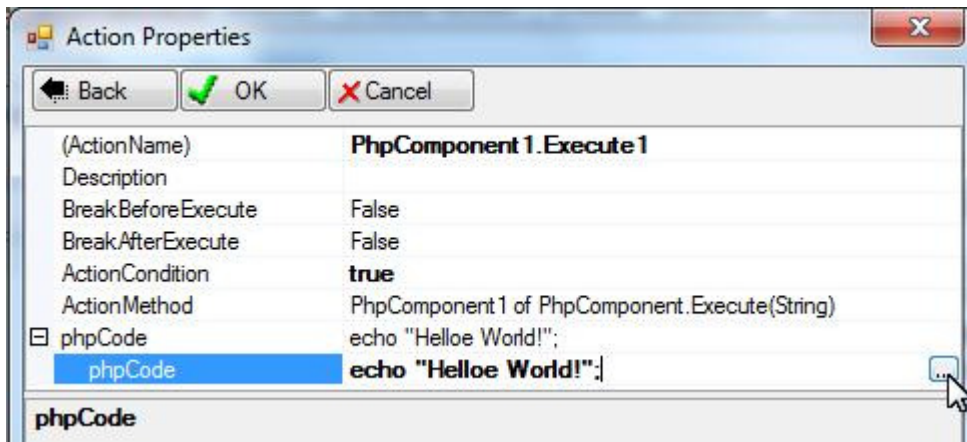
Right-click the PhpComponent; choose "Create Action"; choose "Execute" method:



Parameter "phpCode" of the action is the PHP code to be executed. We may type in some code directly into it:

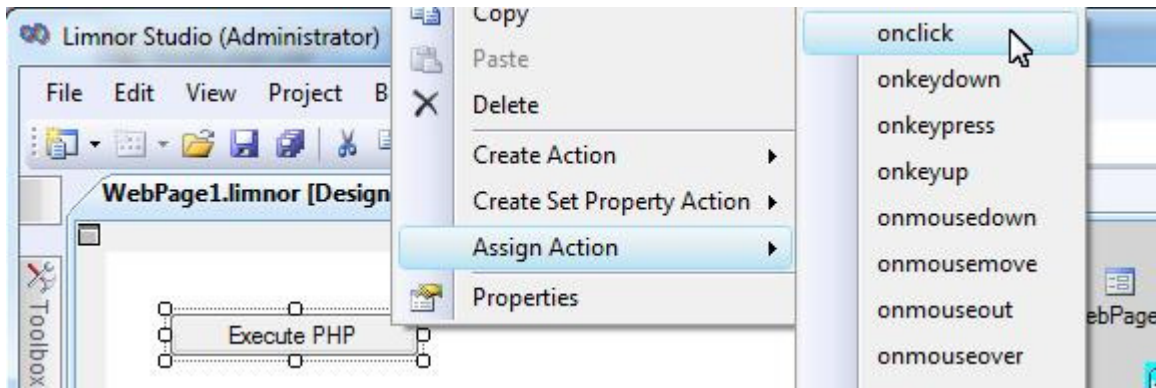


We may also click “...” to launch a dialogue box to edit the PHP code:

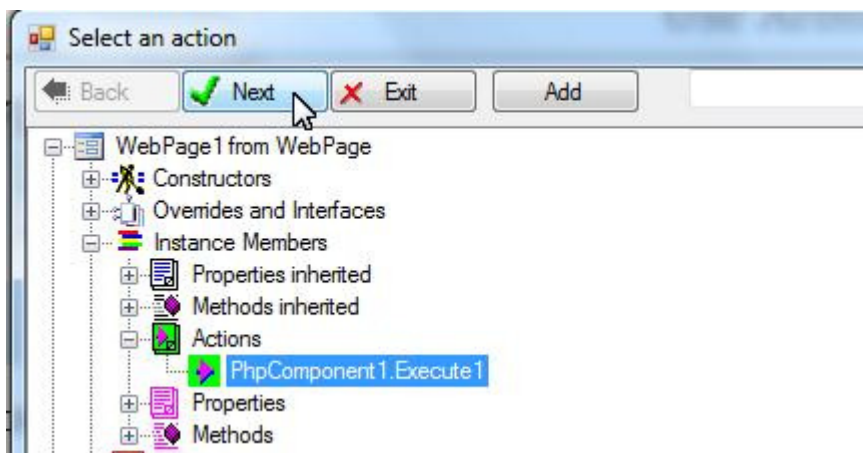


Click OK to finish creating this action.

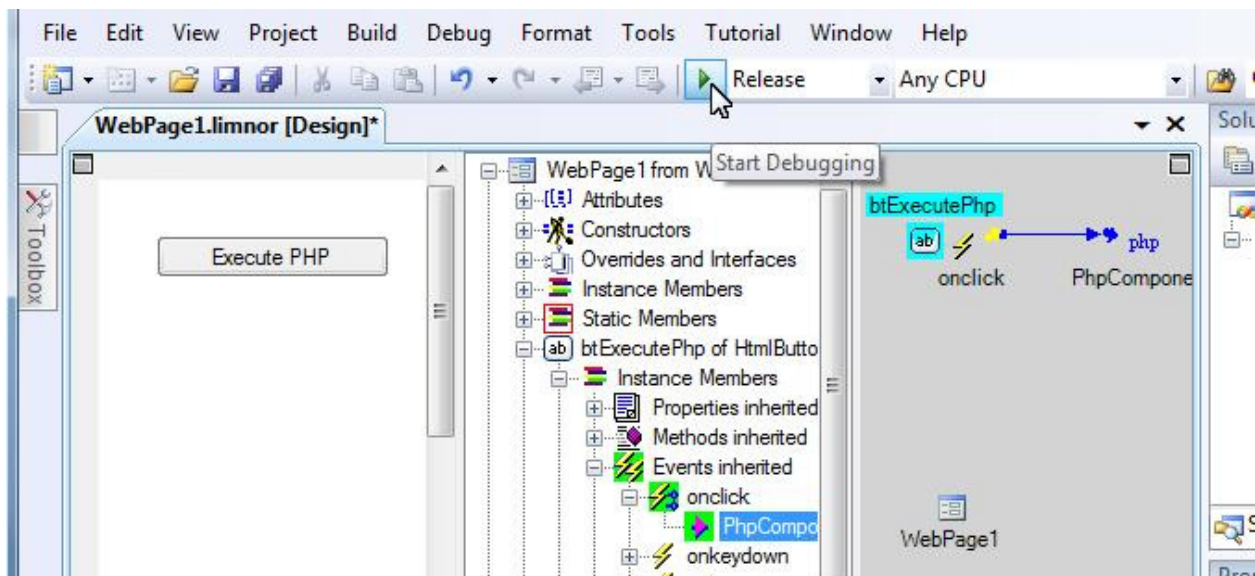
Suppose we want to execute this action when a button is clicked. Right-click the button; choose “Assign Action”; choose “onclick” event:



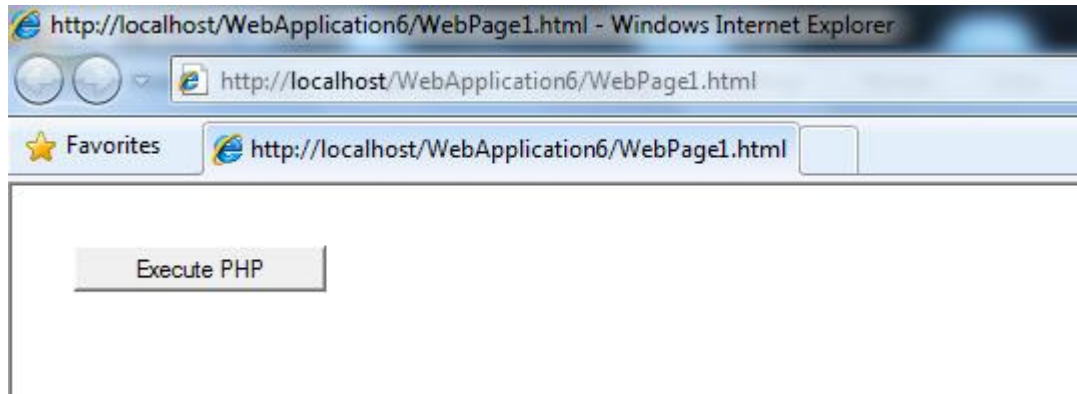
Select the action:



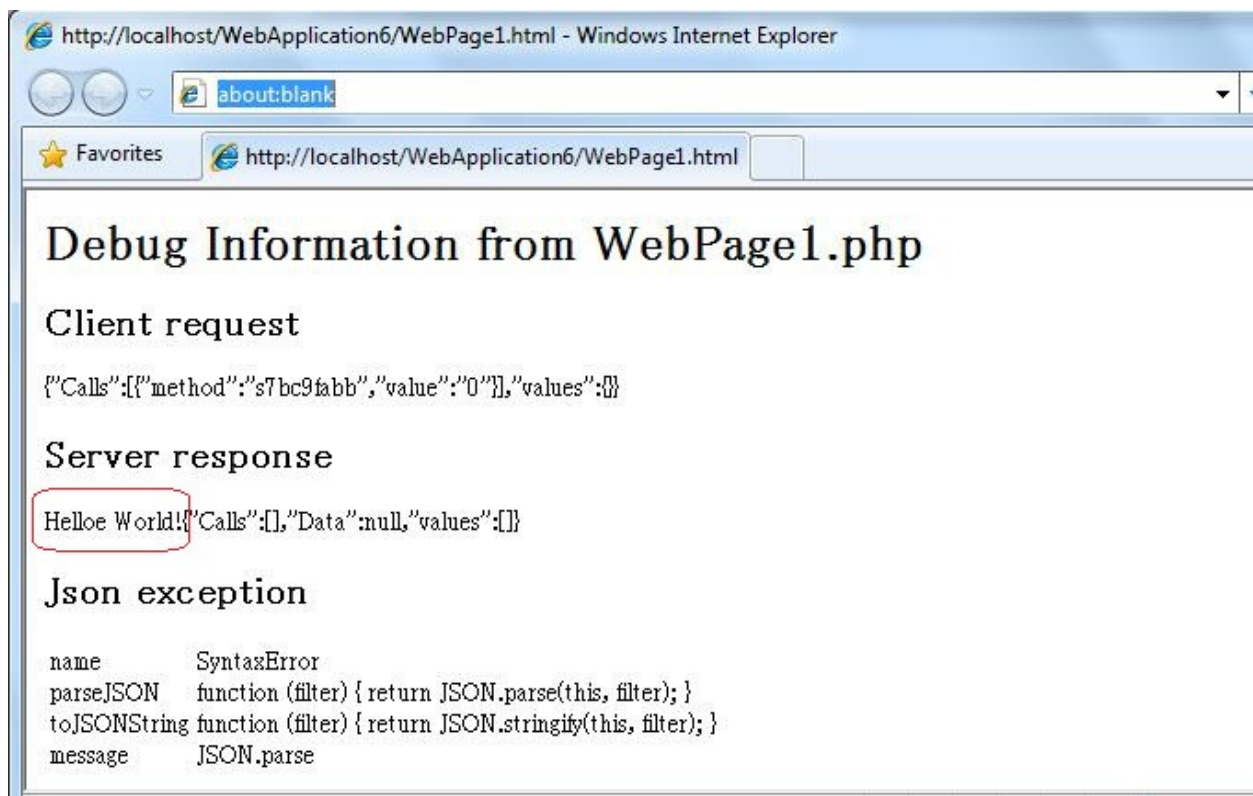
The action is assigned to the button. We may test the web application now:



The web browser shows the web page:



Click the button to execute the PHP code. A debug web page appears because the PHP code uses an “echo” statement to send text to the web client. In Limnor Studio visual and codeless programming, client/server data exchanges are formatted in Json data. Using “echo” to send unformatted text will cause the debug window to appear. In this example, we can see “Hello World!” under the “Server Response”, which is the result of executing the “echo” statement:



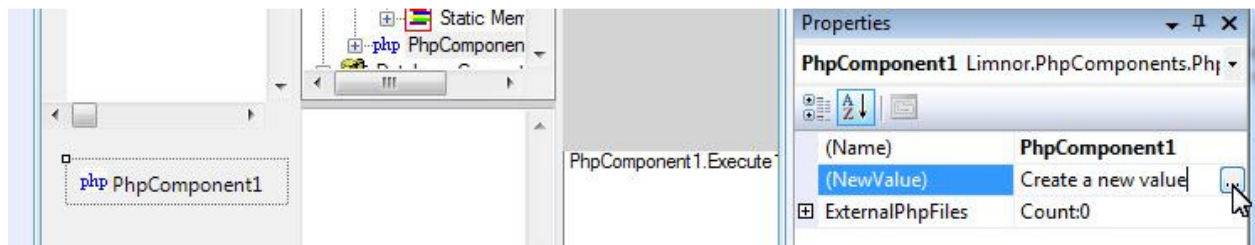
As this example showed, unless you are doing debugging do not use “echo” to pass information to web page. Limnor Studio enables you to easily passing data between client and server in a visual and codeless manner, as shown below.

Data Exchange between Client and Server

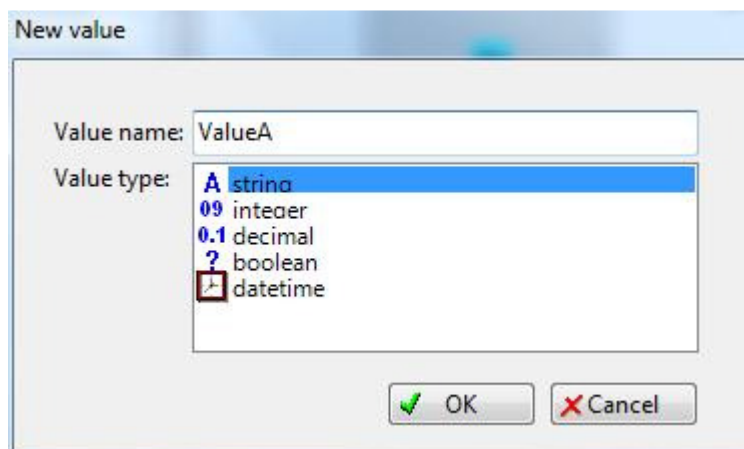
In many cases your PHP codes need values from web pages and pass data to web pages. You may create values for the PhpComponent. Each value can be accessed in both client and server, including your PHP codes.

Create values

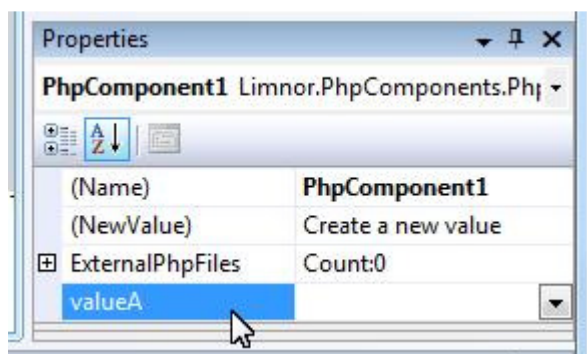
To create a new value, click “...” on the “NewValue” in the Properties Window:



You may give a unique name for the new value. You may specify a value type.

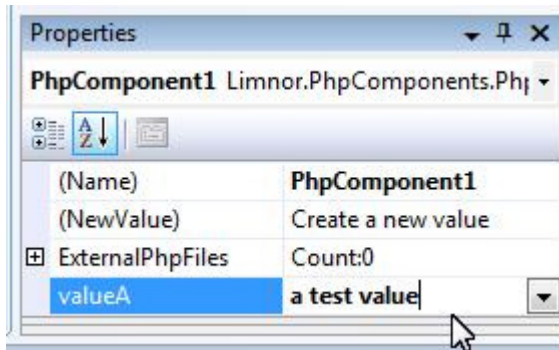


The new value becomes a property of the PhpComponent:



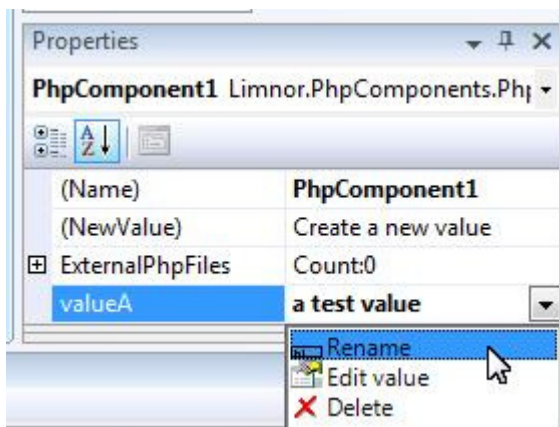
You may use this new property just as using other properties of other components in your programming.

You may give a value for the property:

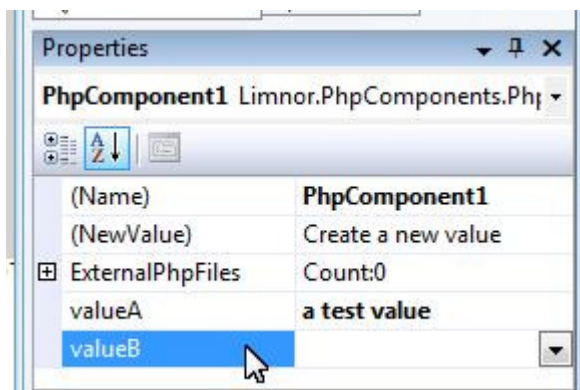


This value will become the initial value of the property when the PHP page is accessed.

You may rename the value, edit value contents in a dialogue box, or delete the value:



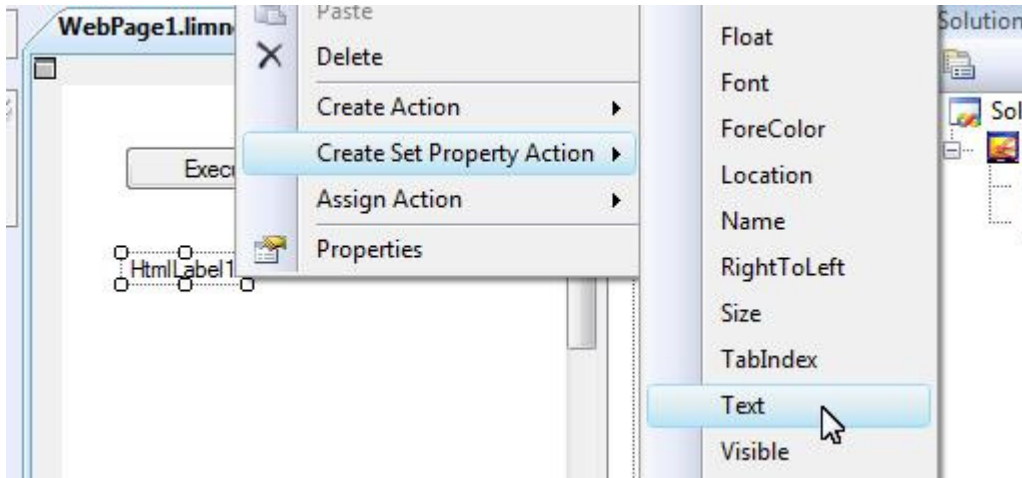
You may create many values for the PhpComponent. Let's create another value named "valueB":



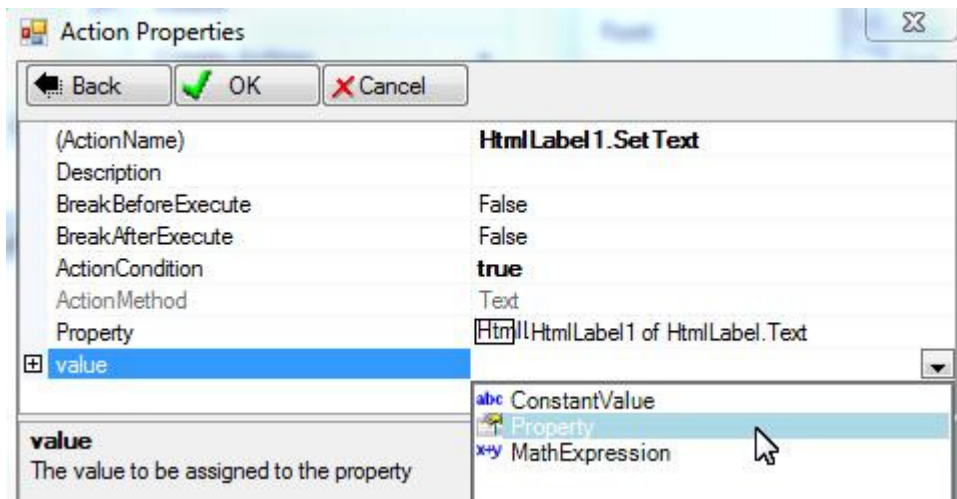
Use value in client

Let's create a label in the web page to show the use of values from the PhpComponent.

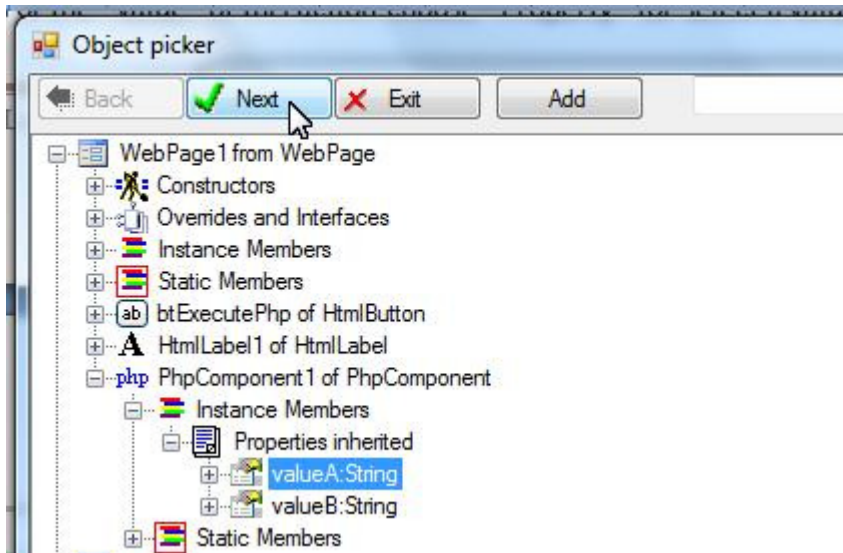
Right-click the label; choose "Create Set Property Action"; choose "Text" property:



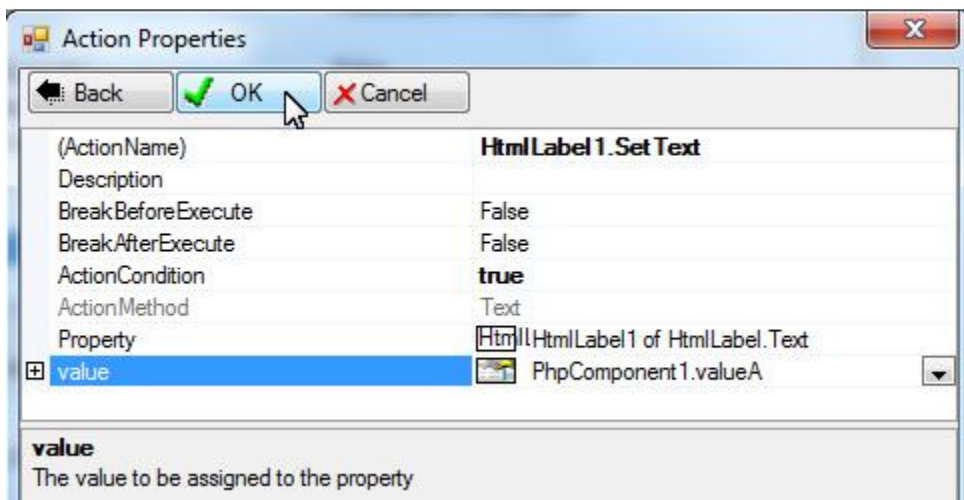
For the “value” of this action choose “Property” for selecting a value from the PhpComponent:



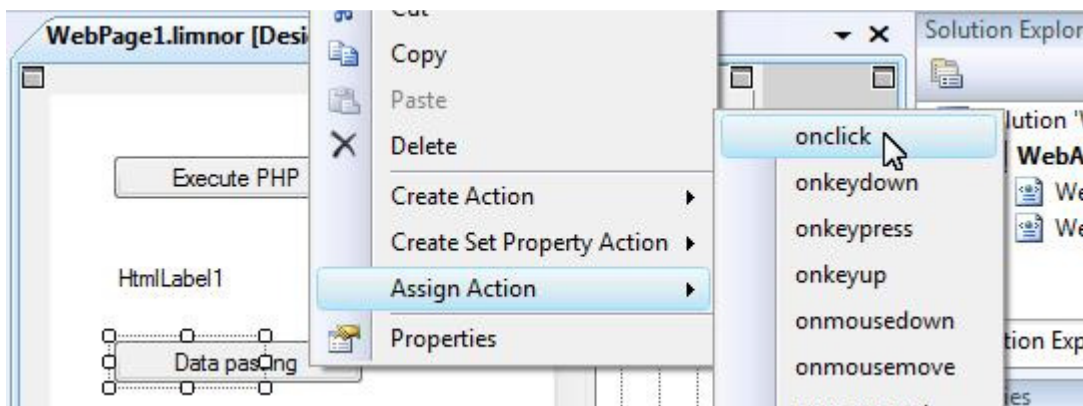
Choose a value from the PhpComponent:



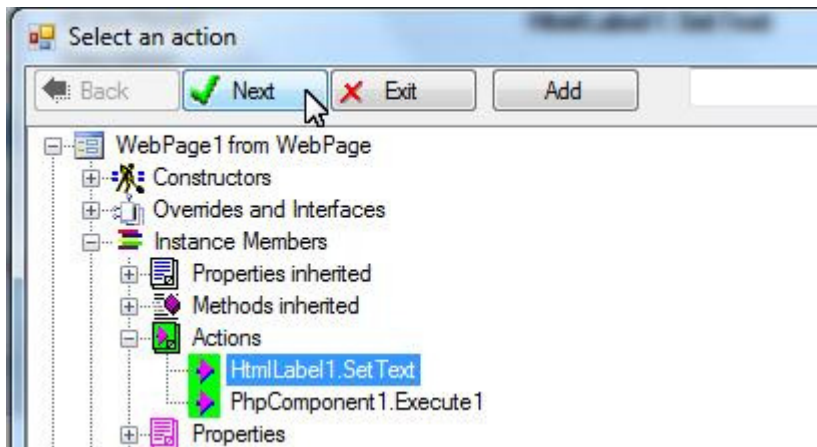
Click OK to finish creating the action:



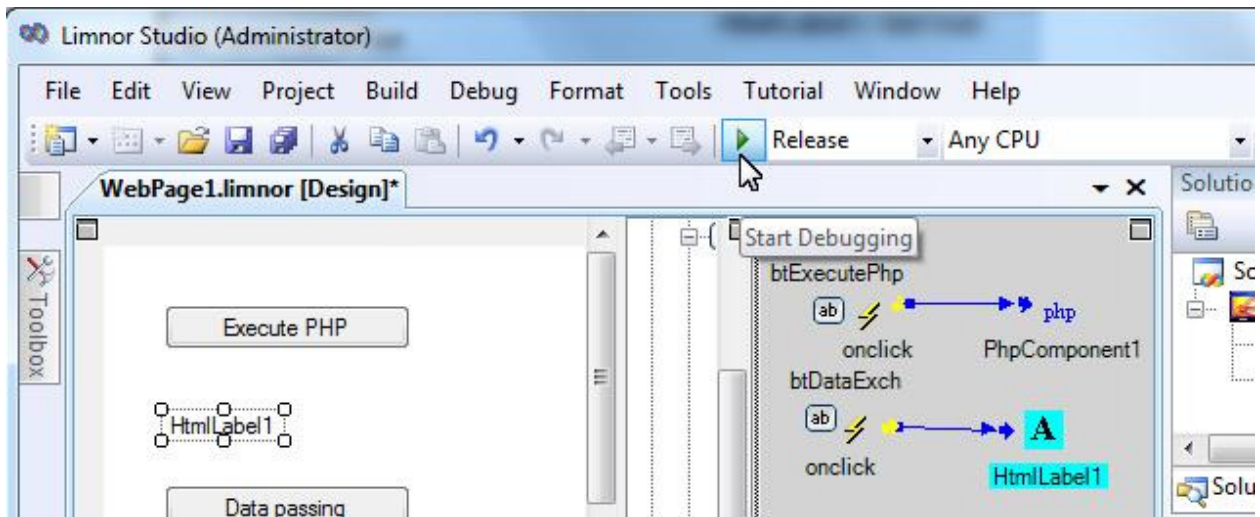
Suppose we want to execute the above action when a button is clicked. Right-click the button; choose “Assign Action”; choose “onclick” event:



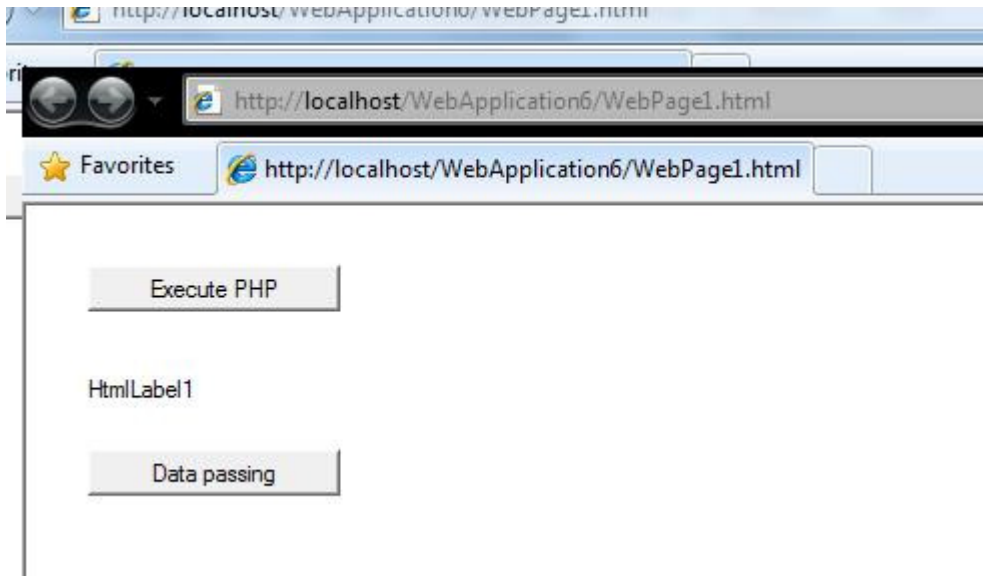
Select the action:



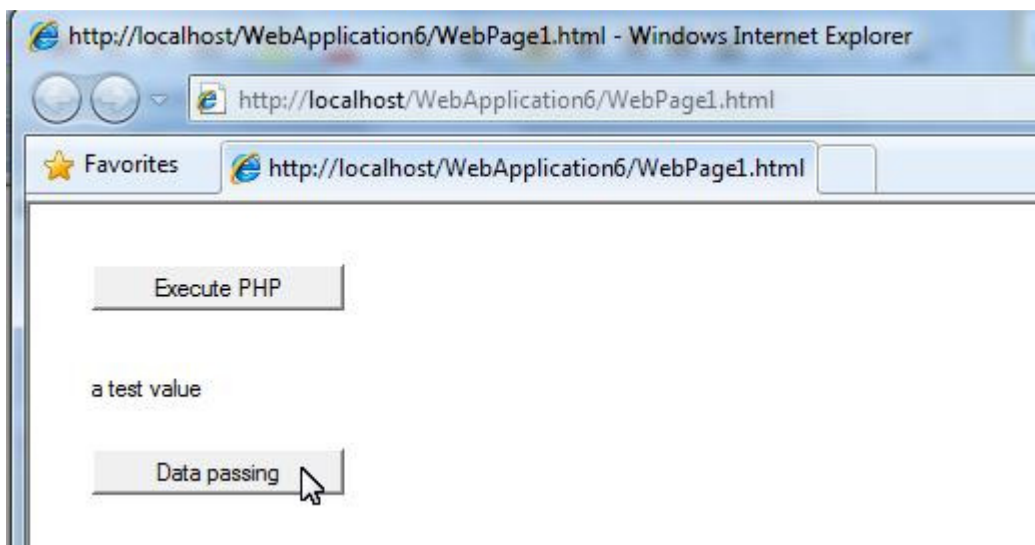
The action is assigned to the button. We may test the web application now:



The web page appears:



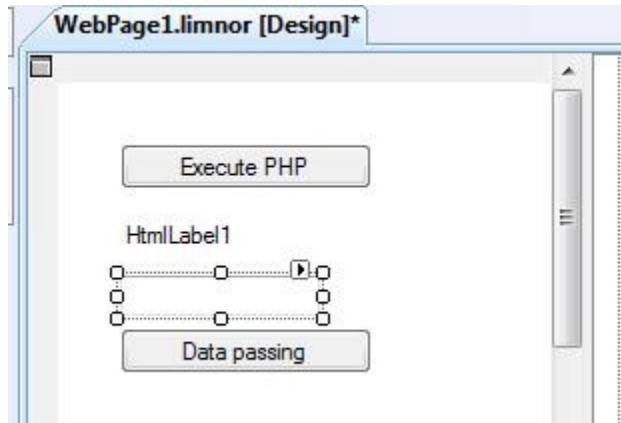
Click the “Data passing” button. The contents of “valueA” appear:



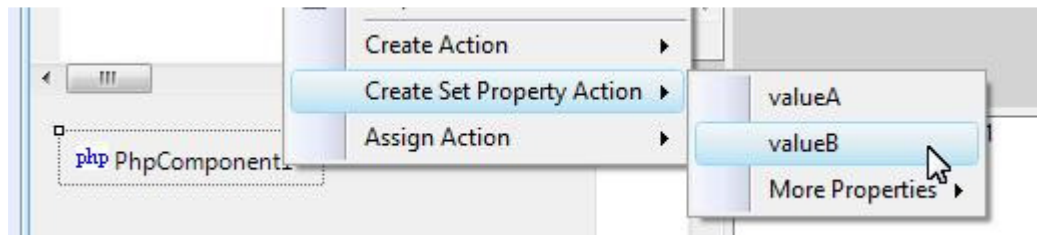
In the example, we saw that the initial value of the PhpComponent was downloaded from the web server to the web page.

Use value in server

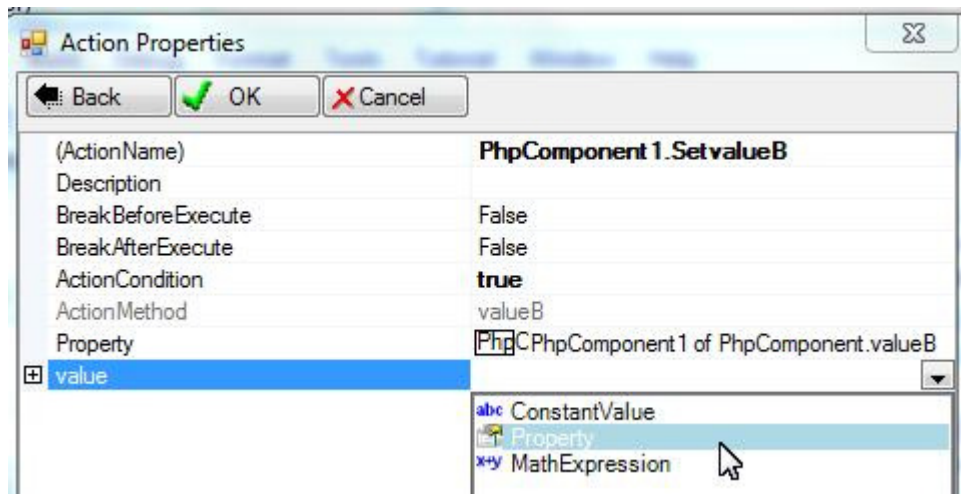
Let's use a text box in the web page to allow the web page visitors to enter text. We then pass the text to a value of the PhpComponent. We then execute PHP code to use the values.



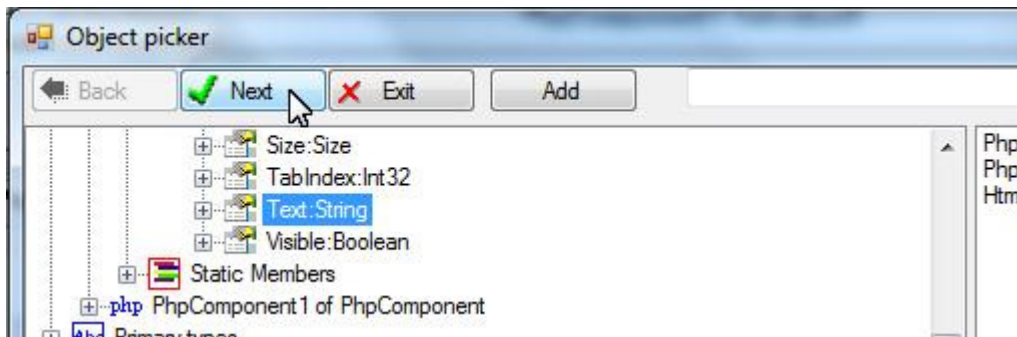
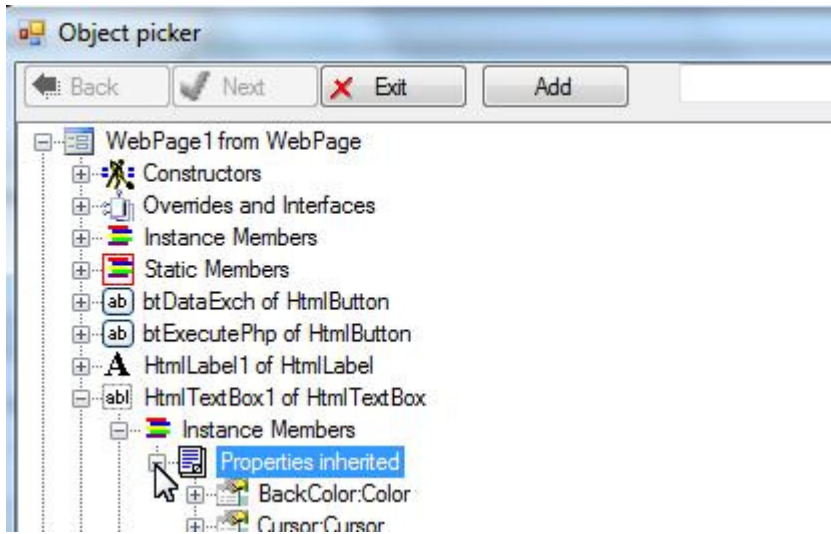
Right-click the PhpComponent; choose "Create Set Property Action"; choose "valueB":



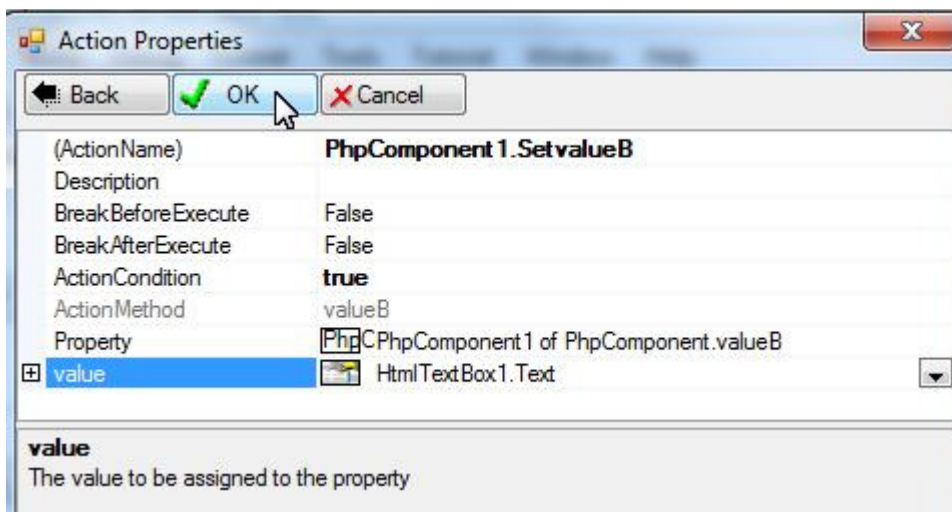
For the "value" parameter of the action, choose "Property" to use the text box:



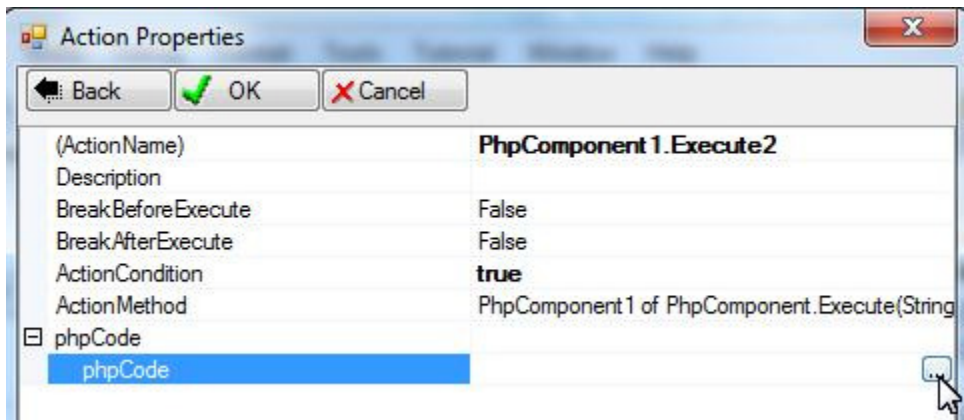
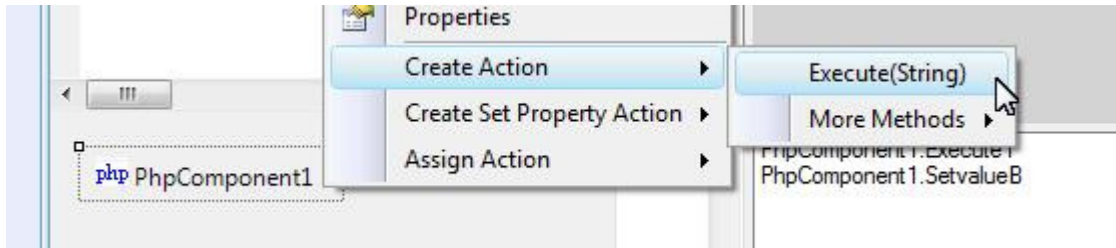
Select the Text property of the text box:



Click OK to finish creating this action:

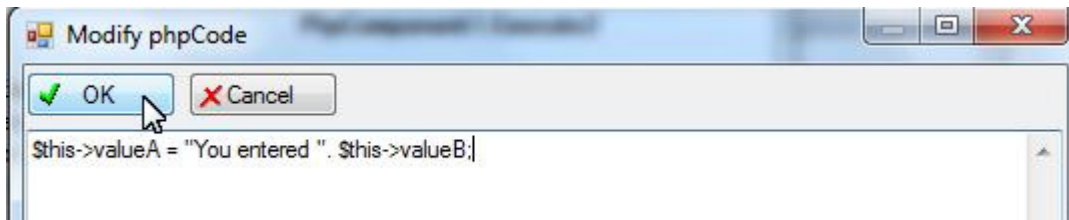


Let's create an action to execute PHP code:

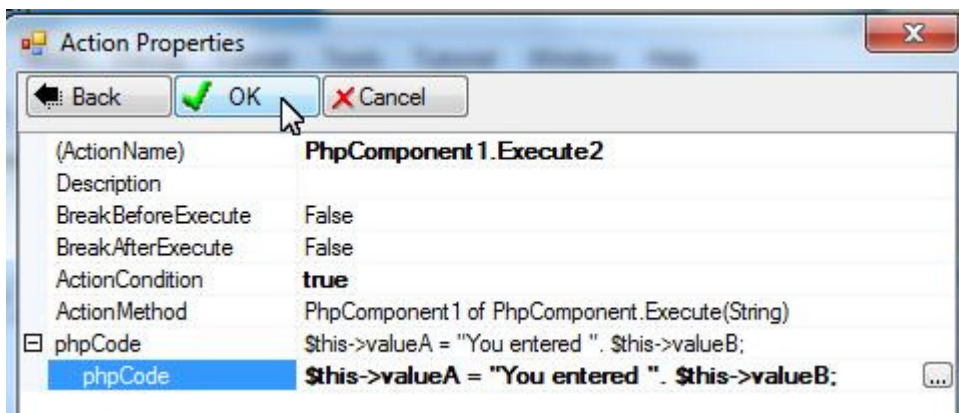


We use such a line of PHP code for this sample:

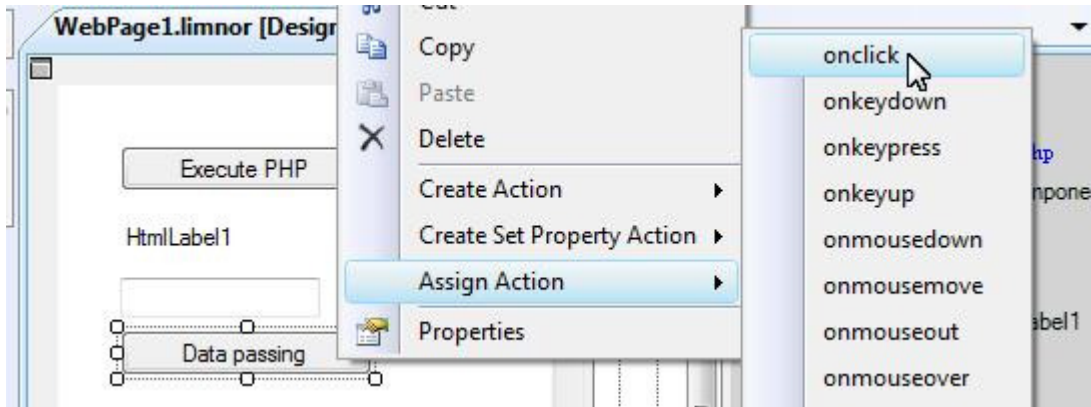
```
$this->valueA = "You entered ". $this->valueB;
```



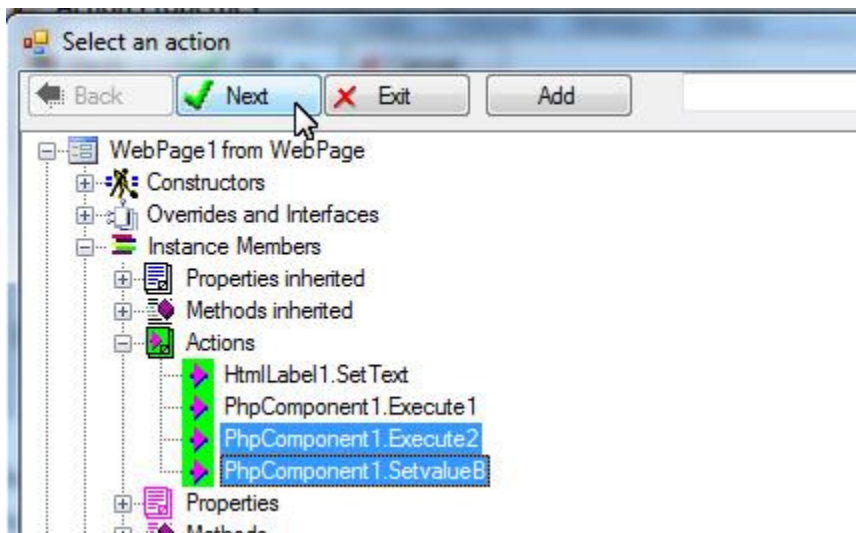
Click OK to finish creating this action:



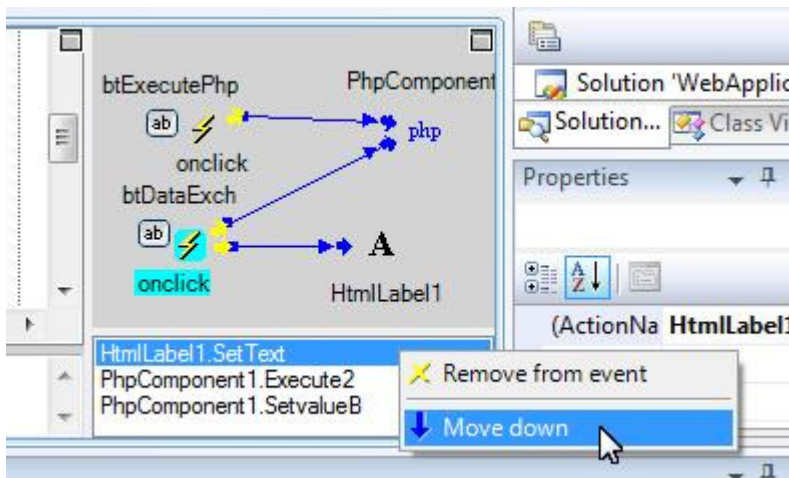
Assign the above two actions to the button "Data passing". Right-click the button; choose "Assign Action"; choose event "onclick":

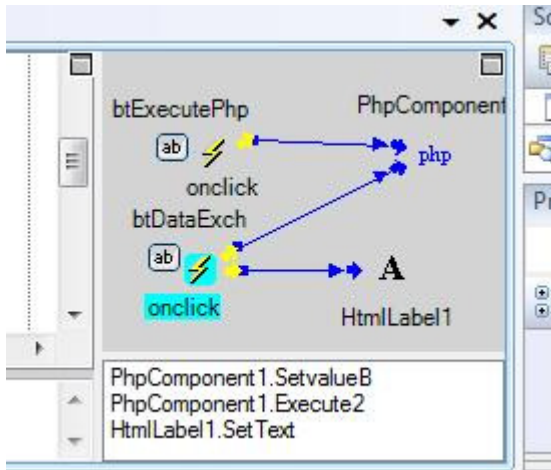


Select the two actions:



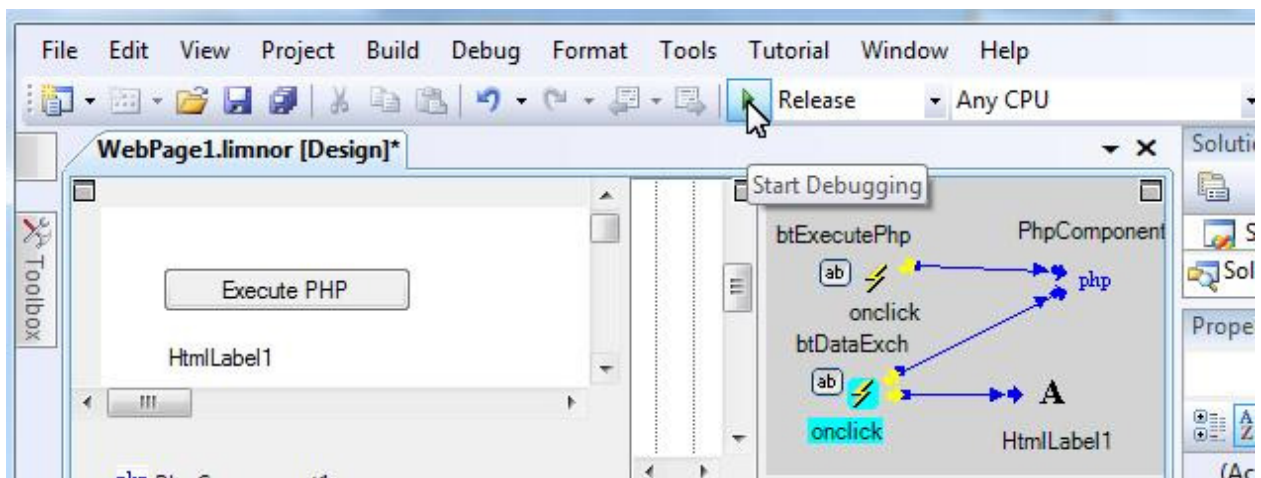
Rearrange the action execution order:



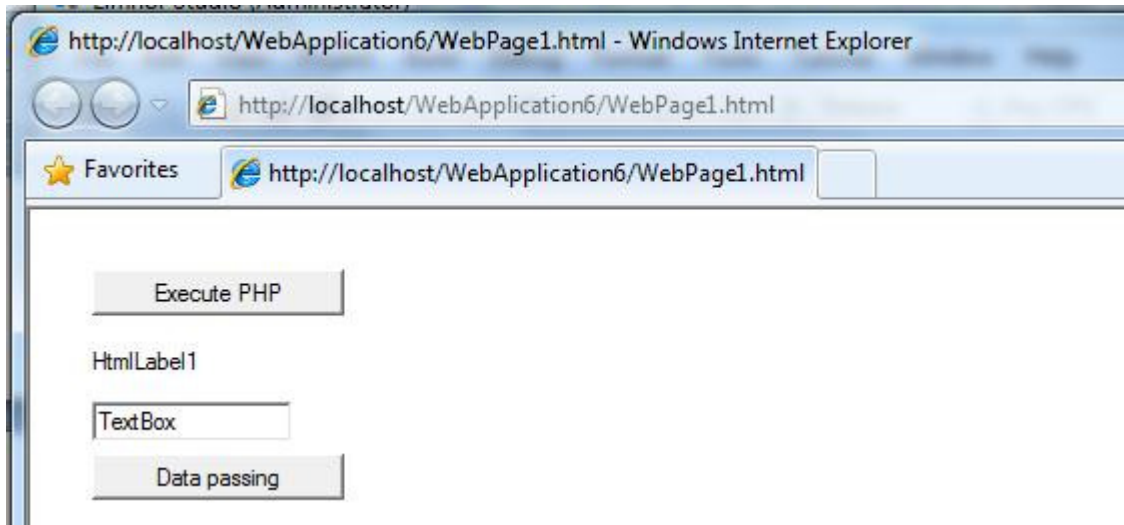


Using the above execution order, a client value in the text box is uploaded and assigned to “valueB” in the server; PHP code is executed in the server to modify “valueA”; “valueA” is downloaded to the web page and assigned to the label.

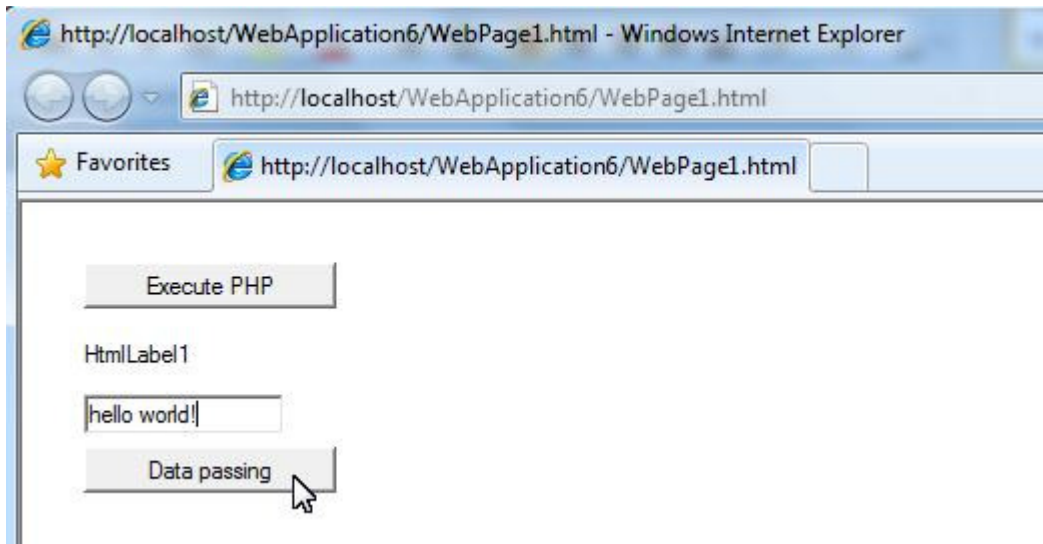
Let’s test it.



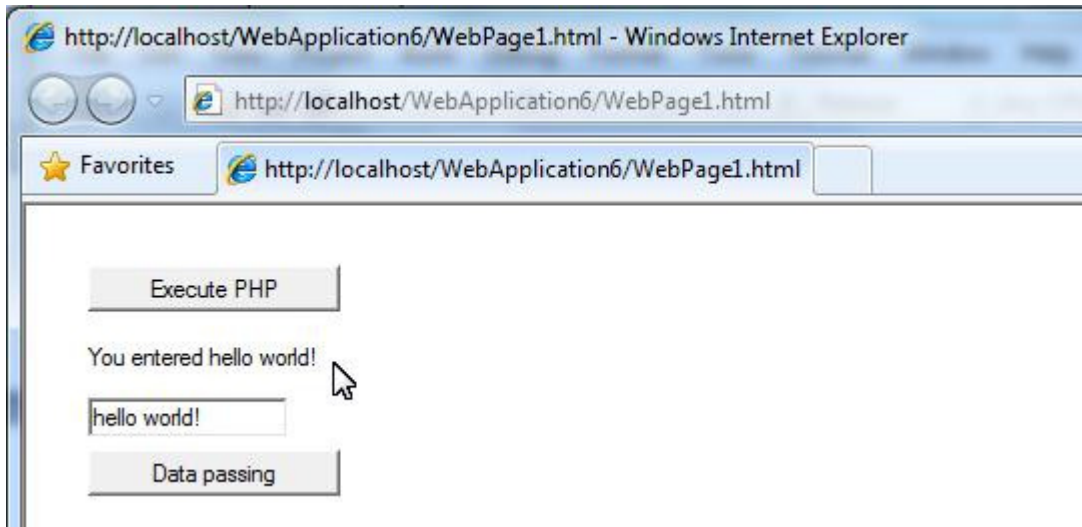
The web page appears:



Type some text in the text box and click the button “Data passing”:



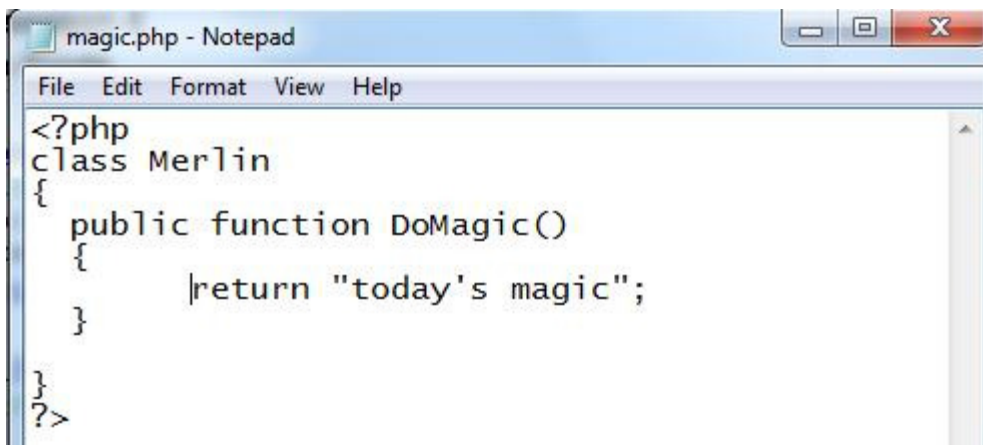
The web page shows the results from the PHP code execution:



Use PHP Files

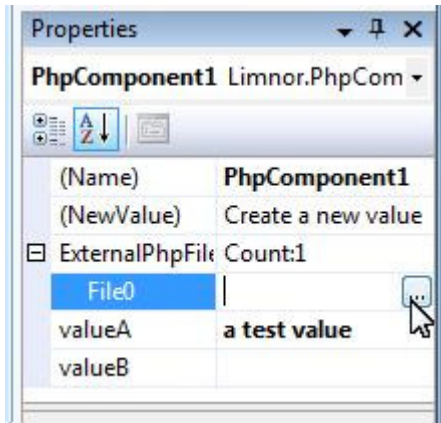
In many cases, you get useful PHP code in PHP files. Your own PHP code is to make use of functionality presented in those PHP files. PhpComponent also allows you to do that.

Suppose we have a PHP file named magic.php. Its contents are shown below:

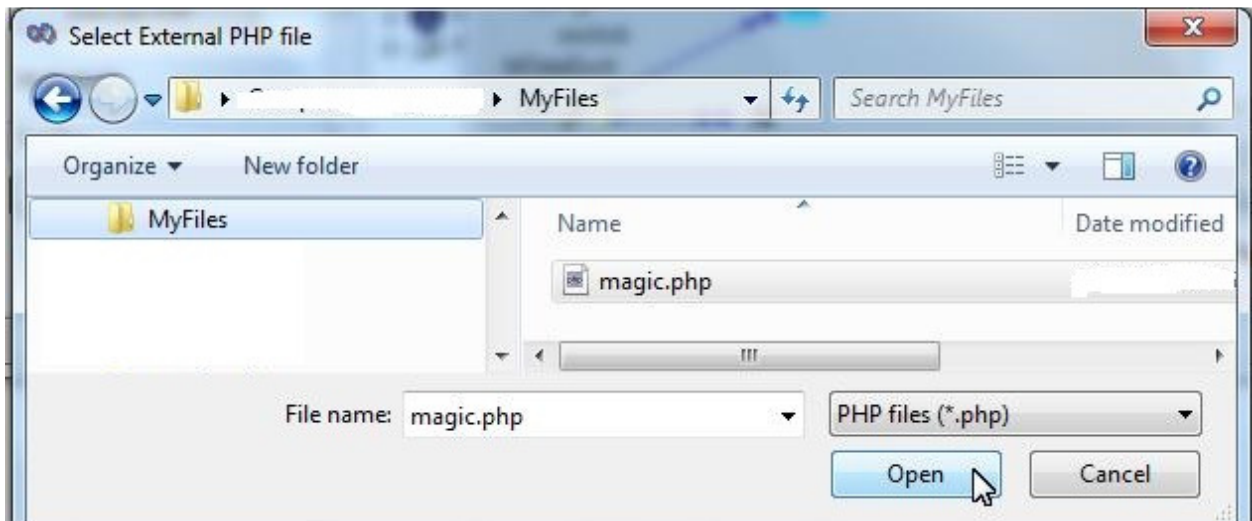


It is a simple file and not useful. But let's pretend it is a complex file and very useful, as yours will be.

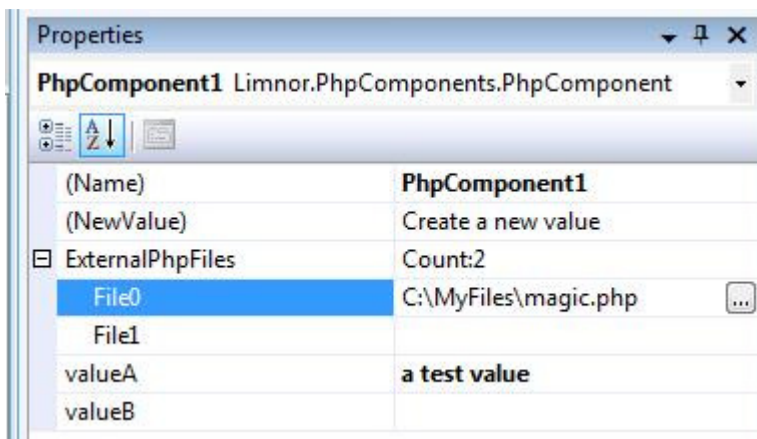
Click "..." for a file name under ExternalPhpFiles in the Properties Window:



Select the file we want:



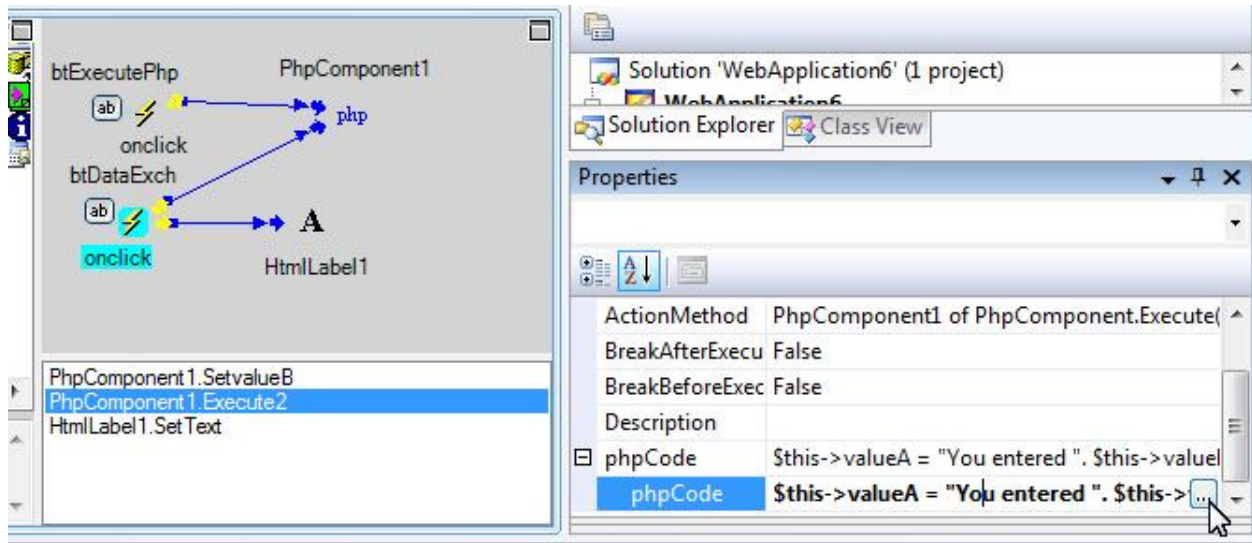
The file path appears under ExternalPhpFiles:



You may add many files. To remove a file, simply clear the file path to empty.

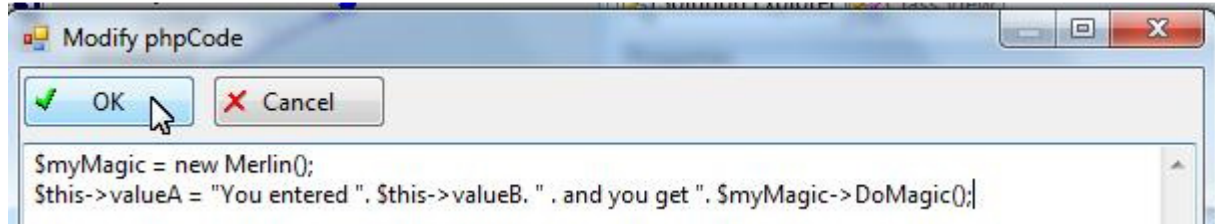
Let's modify the PHP execution action to use the functionality in the PHP file:

Click “...” to launch the editor:

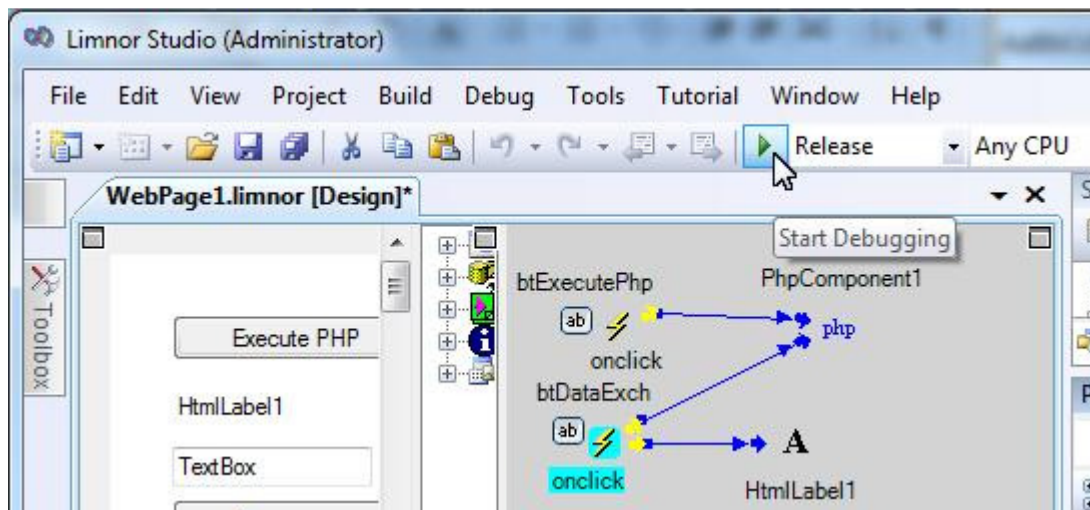


Modify the PHP code to be as following.

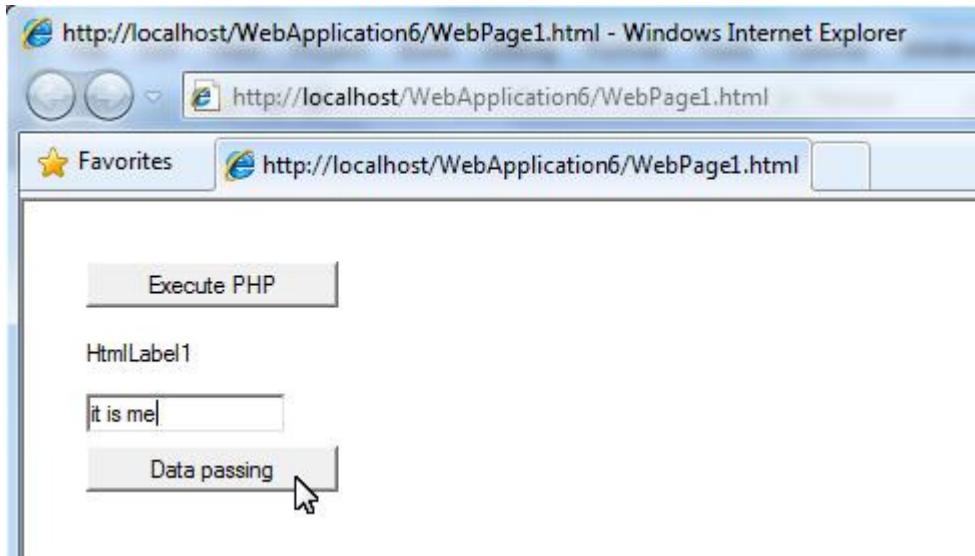
```
$MyMagic = new Merlin();
$this->valueA = "You entered ". $this->valueB. " . and you get ". $MyMagic->DoMagic();
```



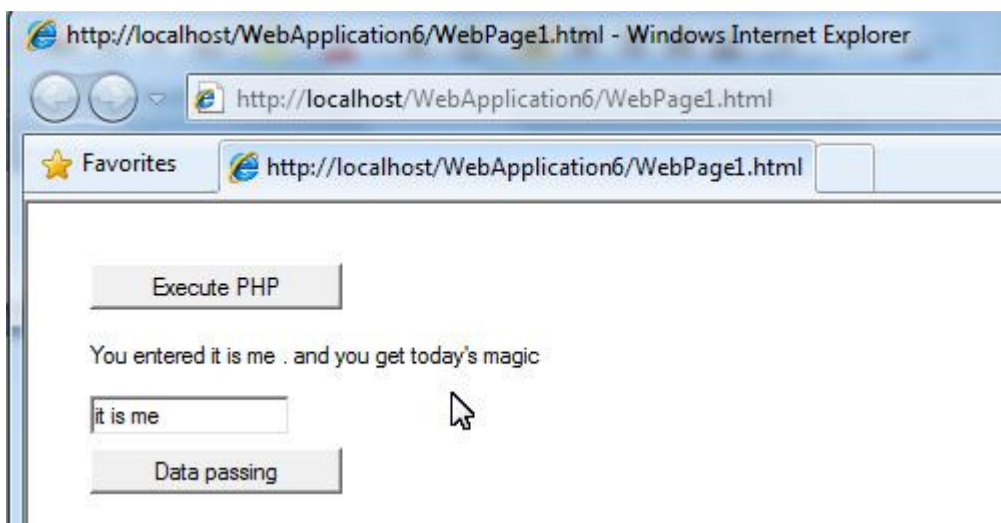
Now we may test the web application again:



The web page appears. Type in some text and click button “Data passing”:



The web page shows the PHP code execution results:



That is it!

Now it is time for you to do your magic with your PHP web applications development.