How to Get Application Folder
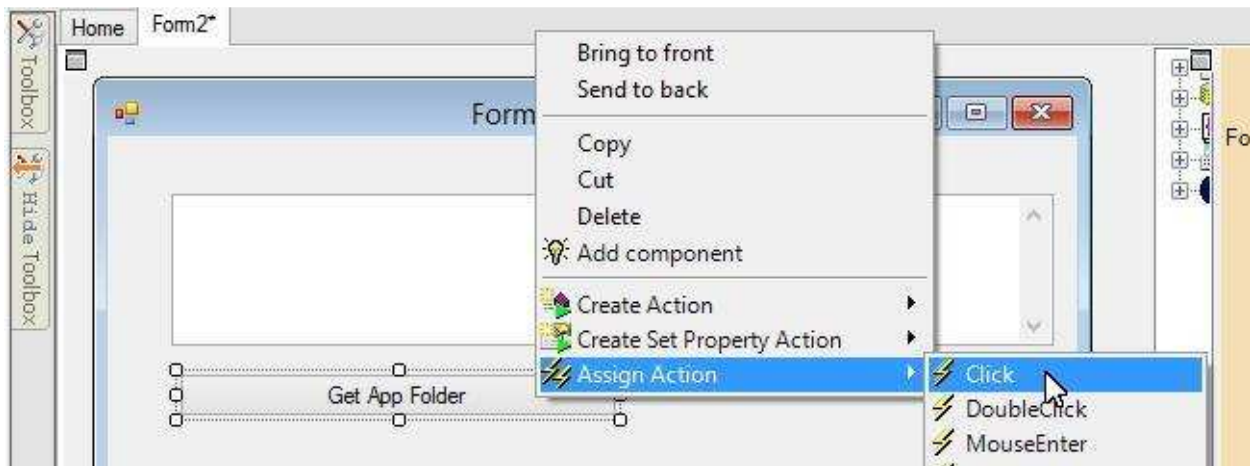
Contents

Use Application Folder in an Action

Suppose when the user clicks a button, we want to display following string in a text box:

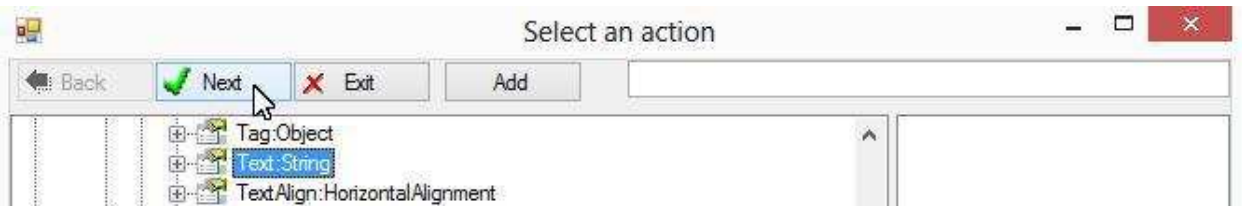Application folder is {Application folder} .

Where {Application folder} is the full path of the application folder, its value depends on where the user installs the application. For example, it can be c:\Program Files\My Company\ProductX
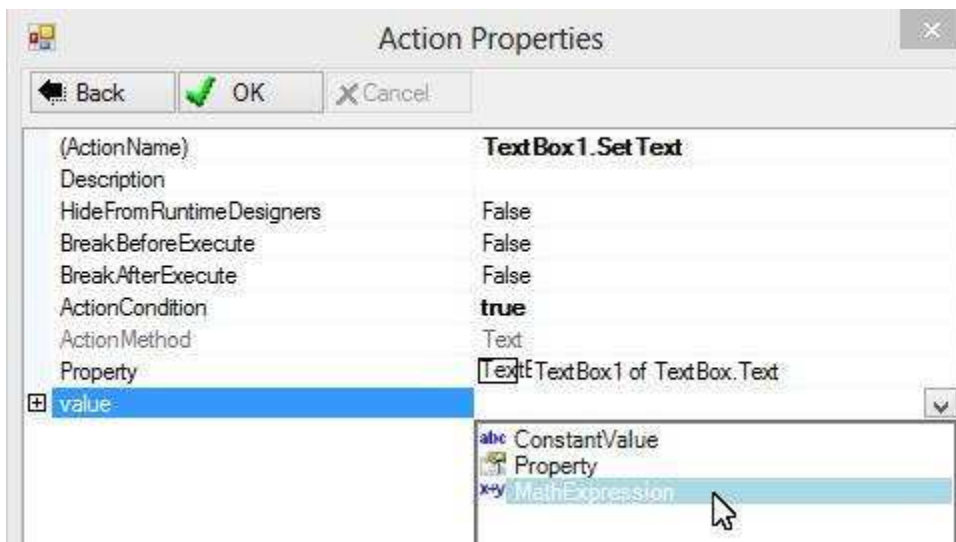
Let's program it.

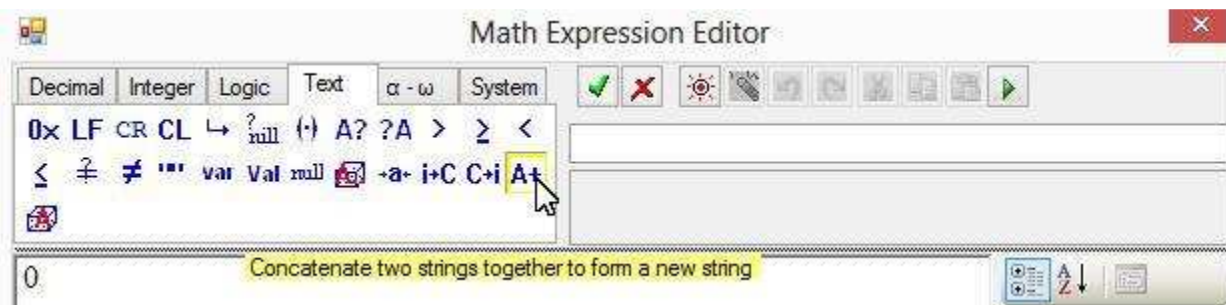Right-click the button, choose "Assign Action", choose "Click":

Select the Text property of the text box, because we want to change the value of this property:
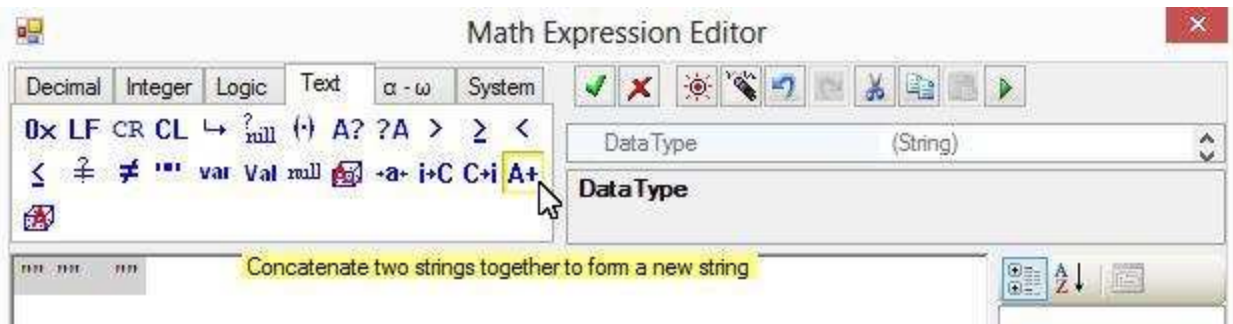
Choose "Math Expression" to construct desire string value:
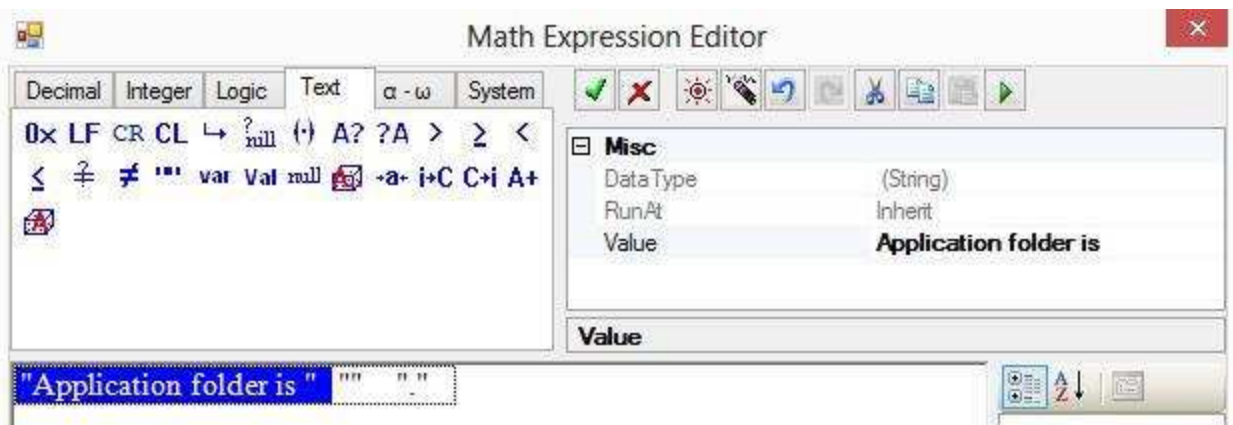

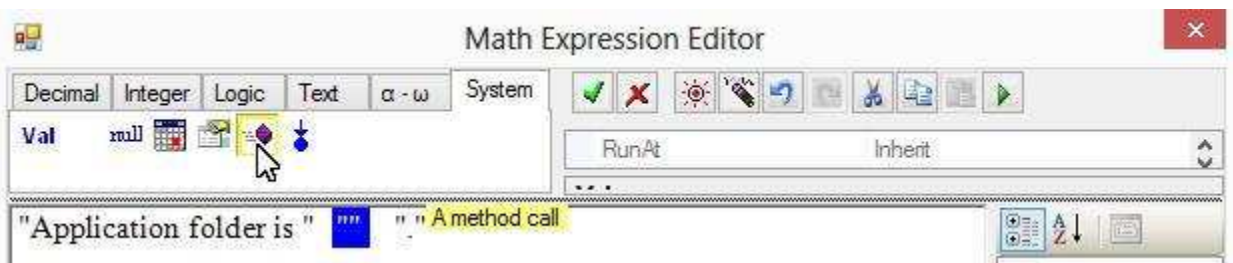Click A+ twice to create 3 string parts:

The first and last parts are constant strings:

For the second part of the string, we want to use the application folder. Here we use following approach to get it: ExecutablePath property of the Application object is the full file name and path of the application EXE file, we may use GetDirectoryName method of the Path object to extract the folder from ExecutablePath. For example, suppose ExecutablePath is c:\Program Files\My Company\ProductX\app1.exe, GetDirectoryName will return c:\Program Files\My Company\ProductX

To use GetDirectoryName method for the second part, select the second part, click method icon:
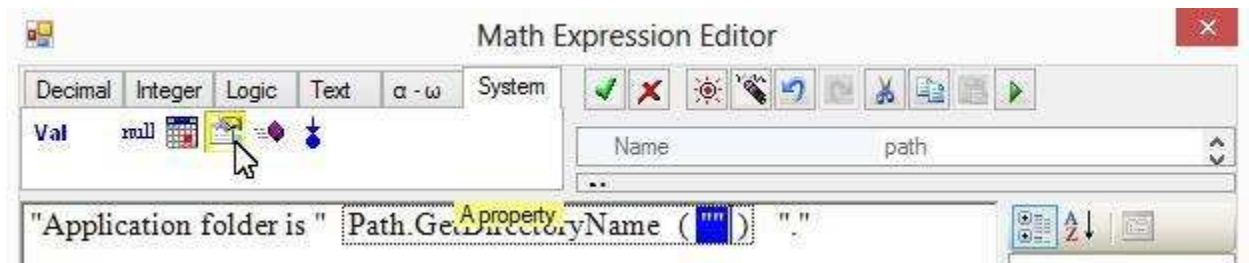

GetDirectoryName belongs to Path; Path is under System.IO namespace. Let's expand Namespaces to find it:

Select GetDirectoryName under Path; click Next:

GetDirectoryName requires a parameter. In our case we want to use ExecutablePath of Application for this parameter. Select the parameter, click property icon to use the property:

The Application is under System.Windows.Forms namespace. Expand Namespaces to find it:
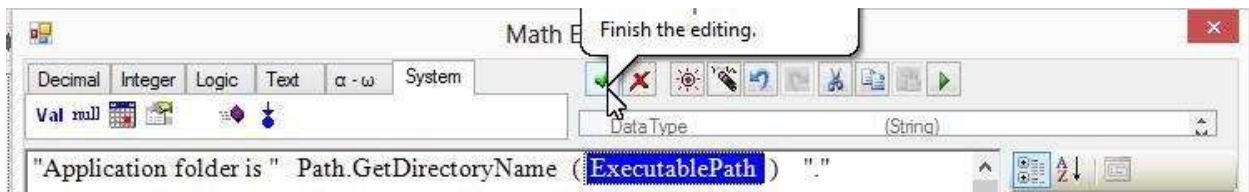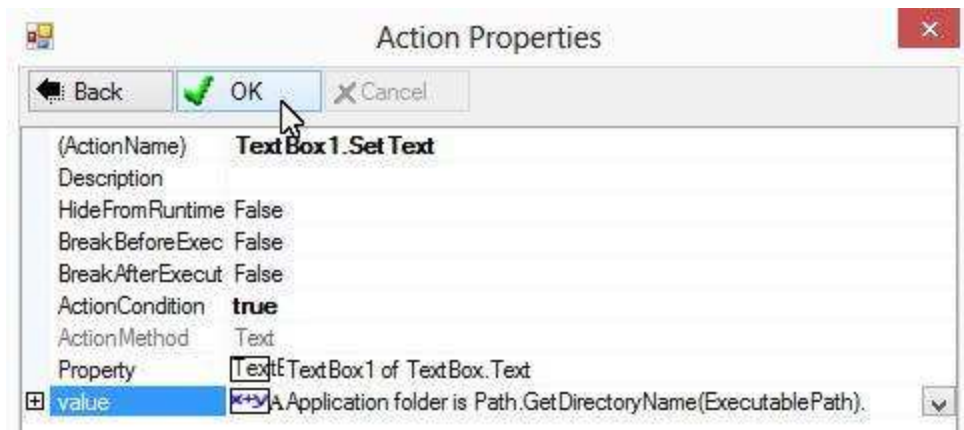
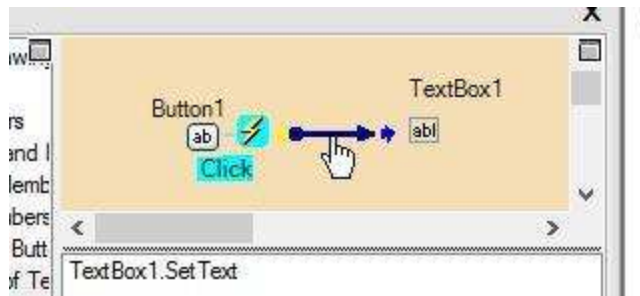Select ExecutablePath under Application; click Next:
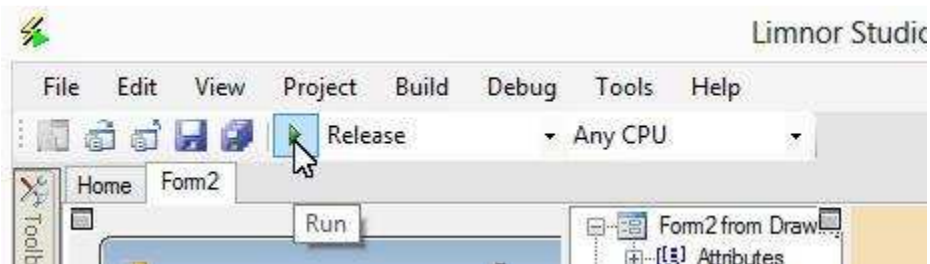

Here is the string expression we formed:

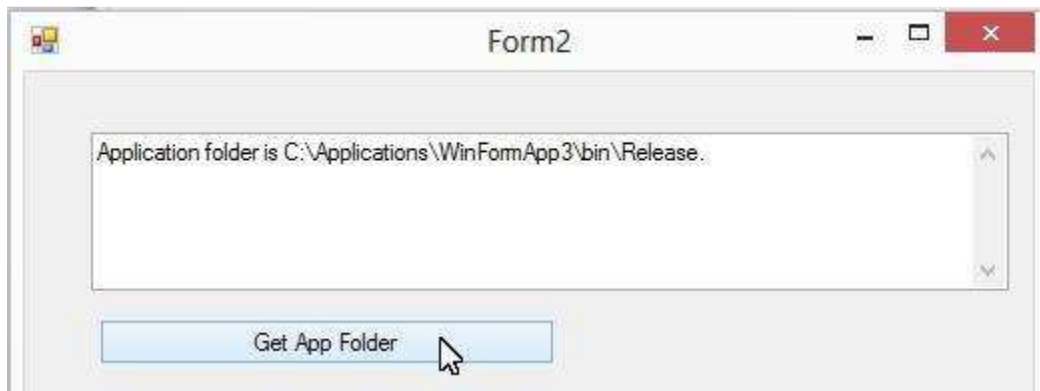Click OK to finish creating the action:

The action is created and assigned to the Click event of the button:

Let's test the application now

The form appears. Click the button. The information appears in the text box:
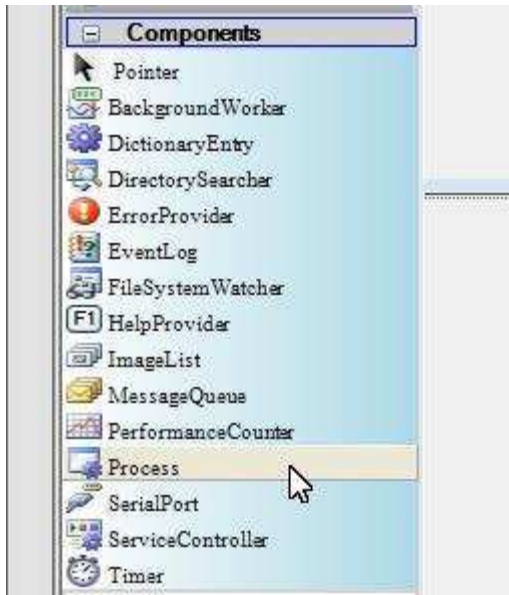

In the above example, we used Application class and Path class. Note that these classes are provided by Microsoft through .Net Framework. You may visit .Net Framework web sites for all the classes it provides. For example, http://msdn.microsoft.com/en-us/library/w0x726c2%28v=VS.90%29.aspx
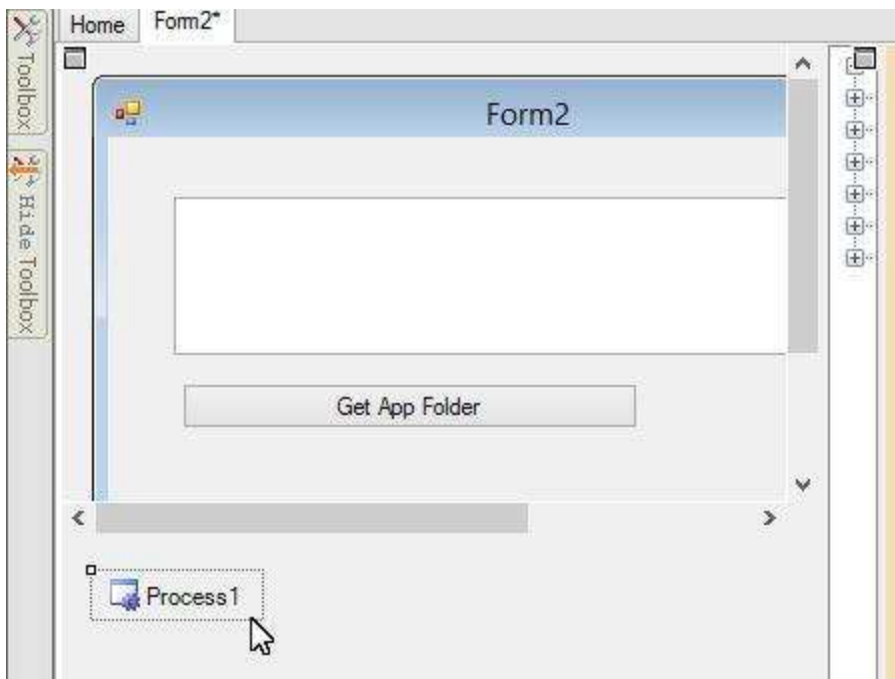
Run Exe from Application Folder

The above approach of using the application folder can be applied to most situations where the application folder is needed.   For example, we may create an action to run an EXE file which we know it will be in the application folder.
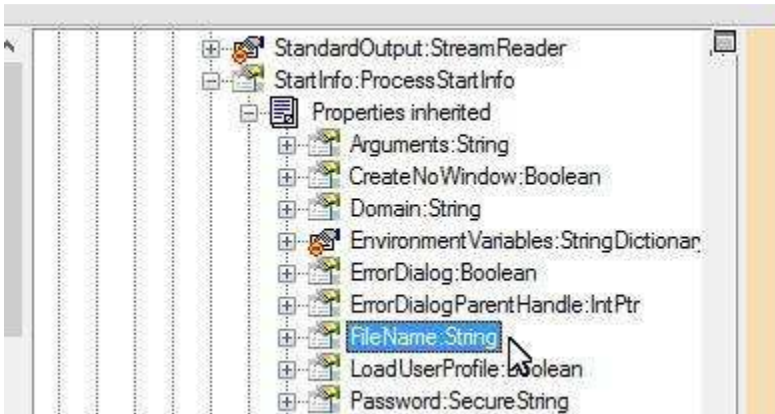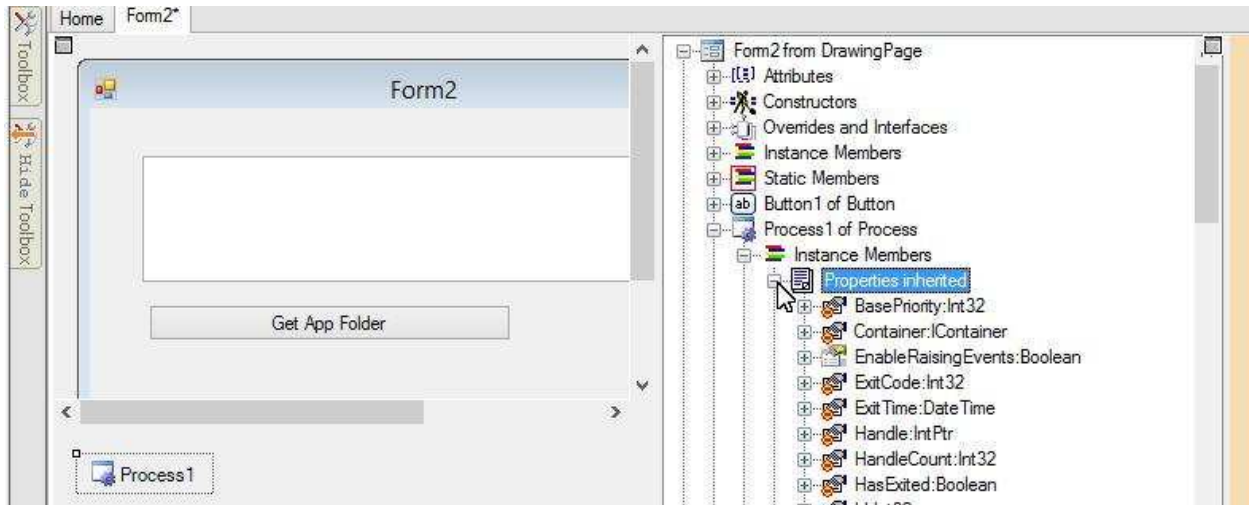
To run an EXE file, we need a Process component. Let's add it to our form. For your convenience you can find Process component in the toolbox. Let's drop it from the toolbox to the form:
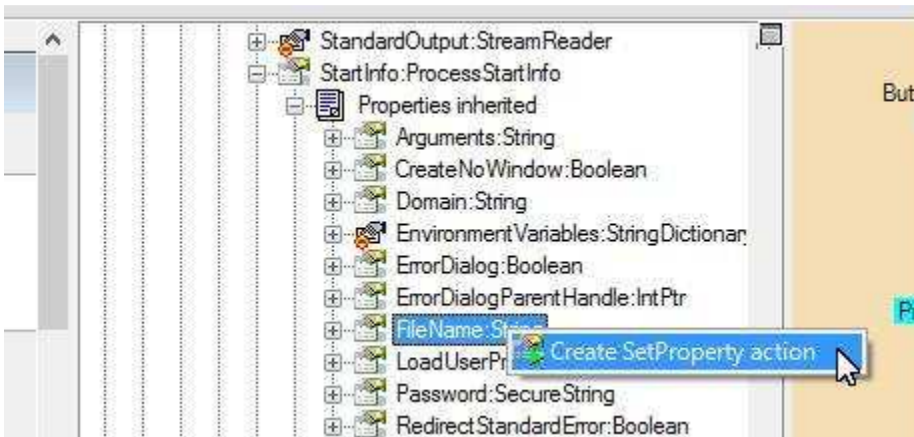
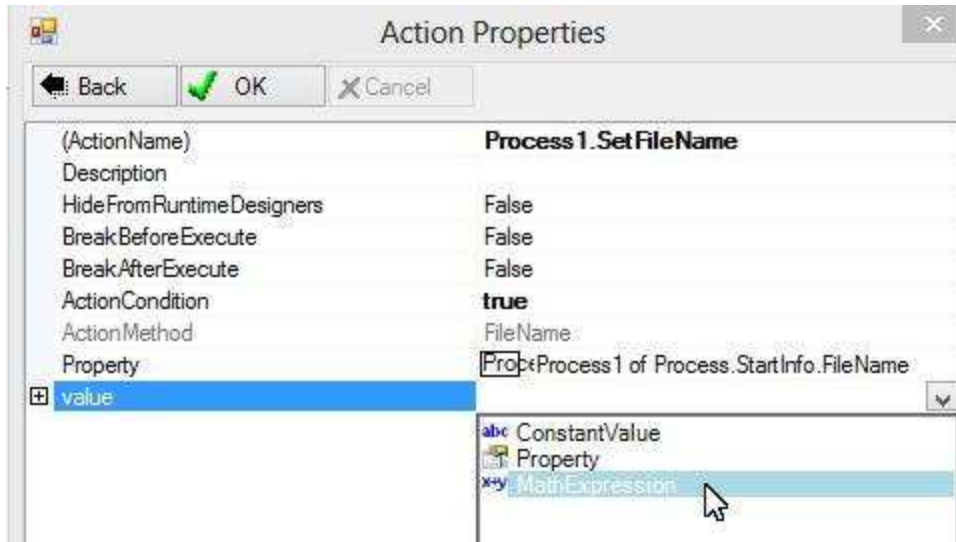It appears in the component tray of the form:

To tell the Process component which program to execute, we need to set its Filename property. The Filename property is under StartInfo property:
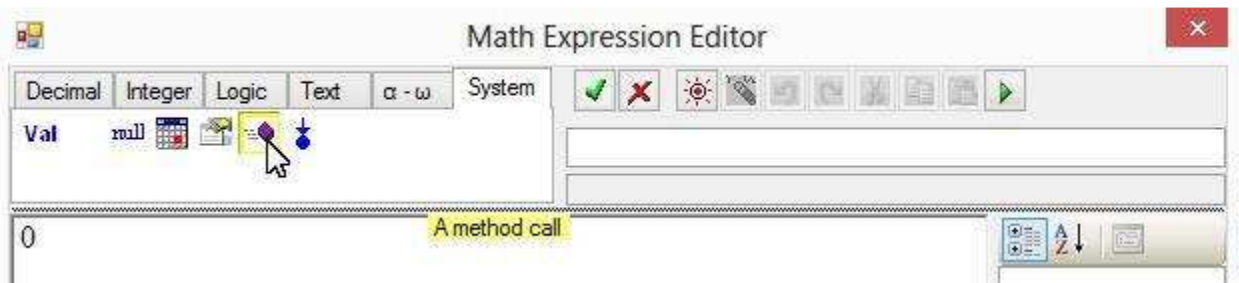
We may create an action to set Filename to the file we want to run. Suppose we want to run an EXE file named **test.exe** under the application folder. To create such an action, right-click Filename property; choose "Create Set Property action":

Select "MathExpression" to form a full path for the EXE file we want to run:
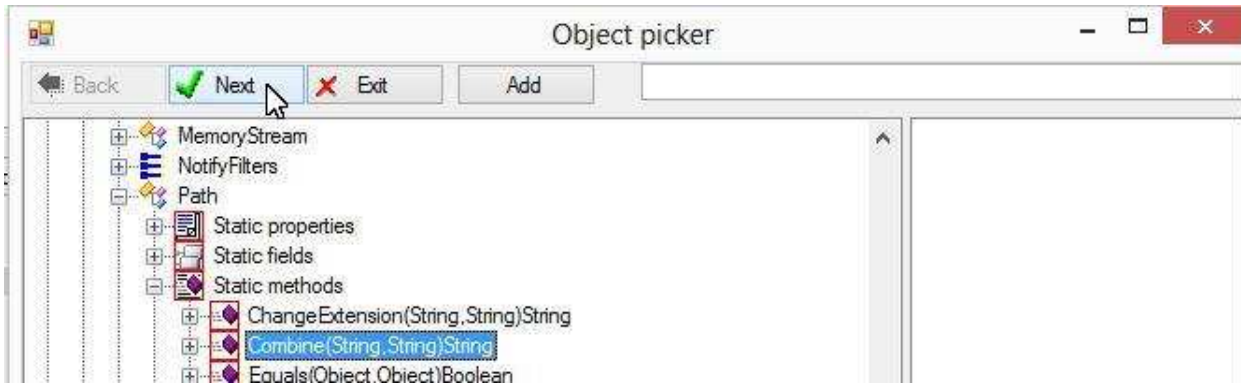
In our situation, we want to run an EXE file named test.exe which is in the application folder. We already know how to get the application folder as shown in the last example; here we need to combine the application folder with the file name, test.exe, to form a full file path. This job, combining a folder and a file name to form a full path, can be done by Combine method of the Path class. Let's click the method icon to select Combine method:
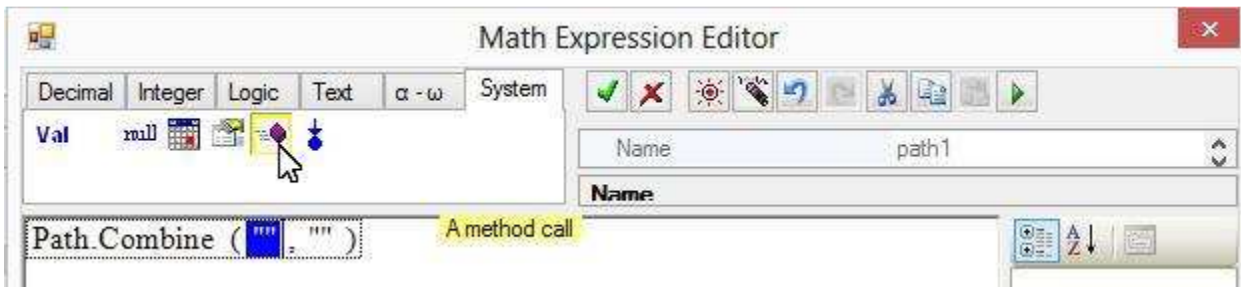

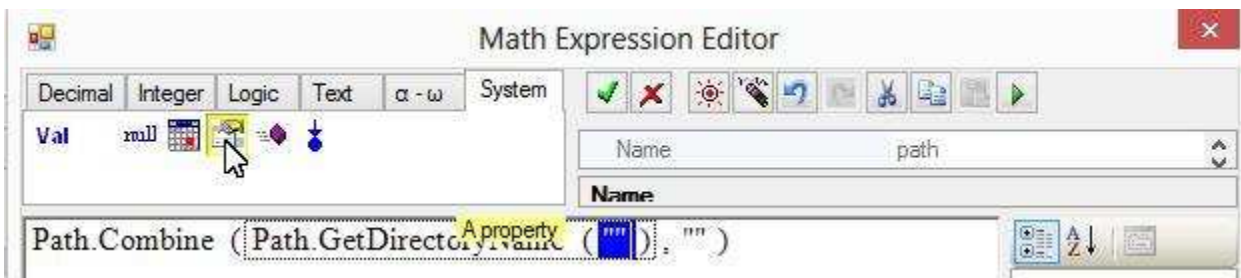Find Path class under System.IO namespace:

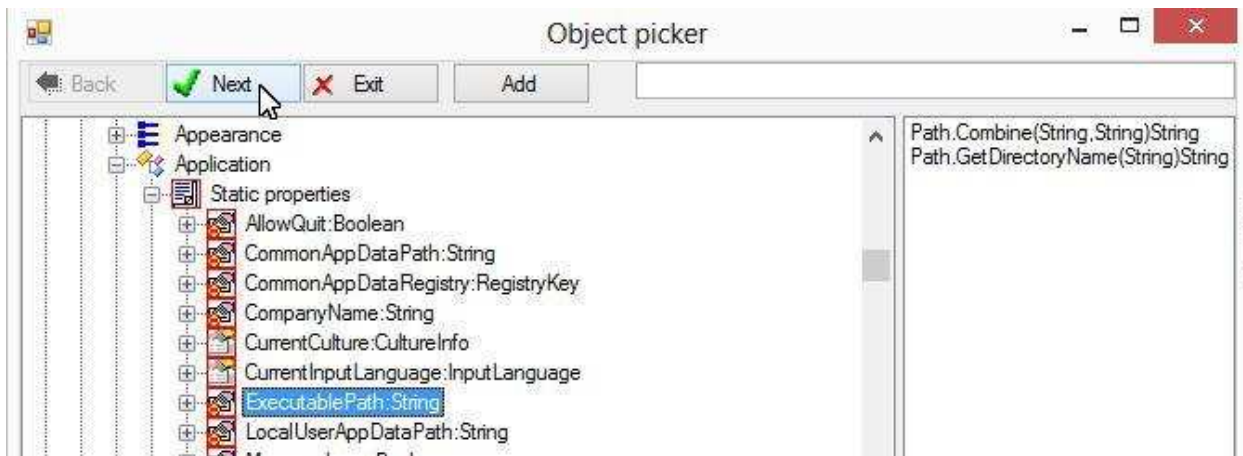Select Combine method under Path class; click Next:

We want to use the application folder for the first parameter; we may use the approach shown in the previous sample to get the application folder. Select the first parameter; click Method icon to choose GetDirectoryName method of the Path class:
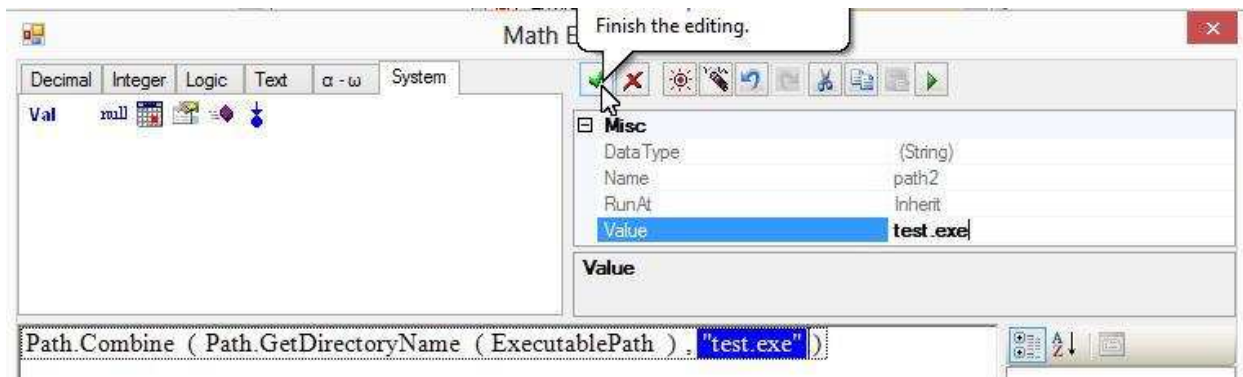

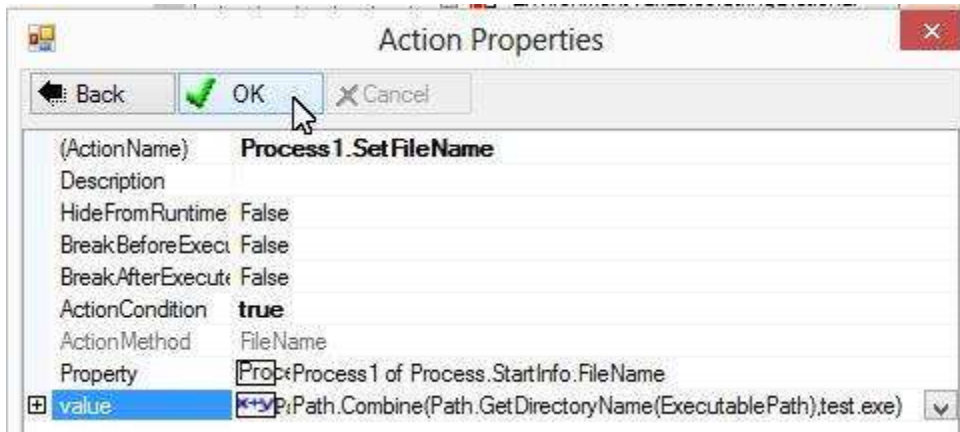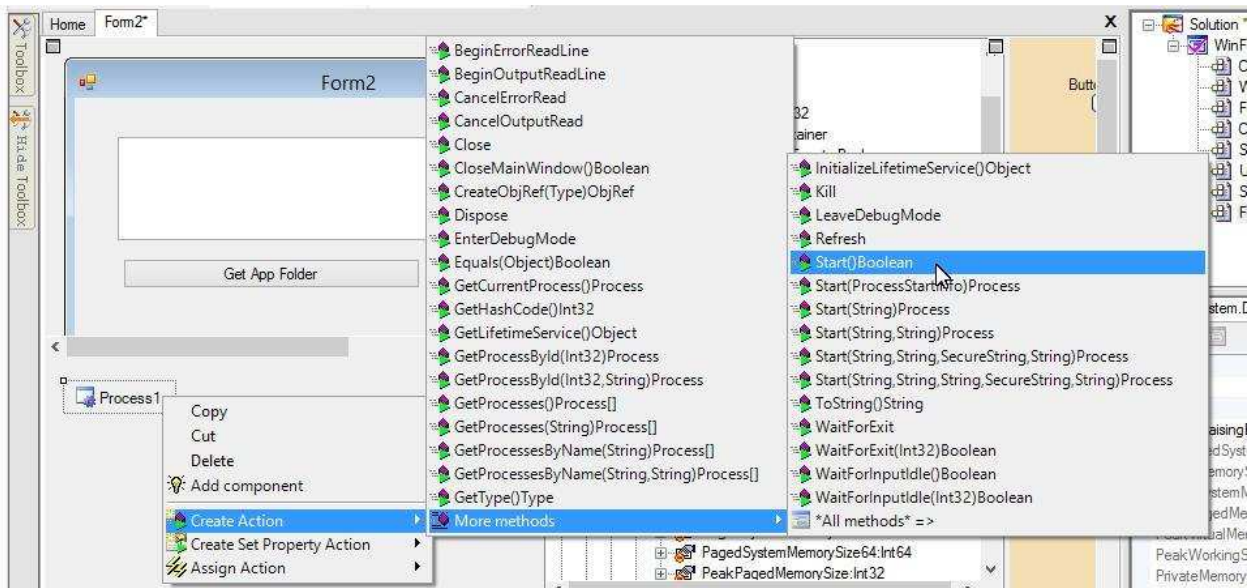Select ExecutablePath property of the Application class for the parameter :

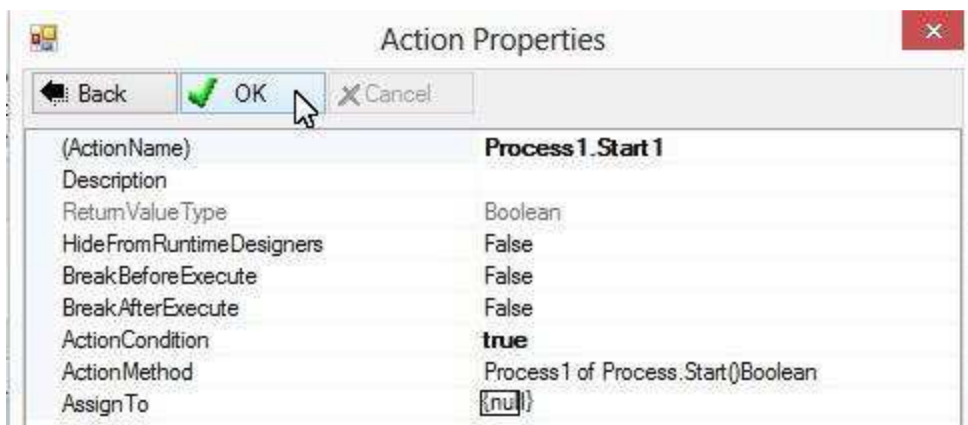For the second parameter of Combine, enter test1.exe. We are done forming the full path:
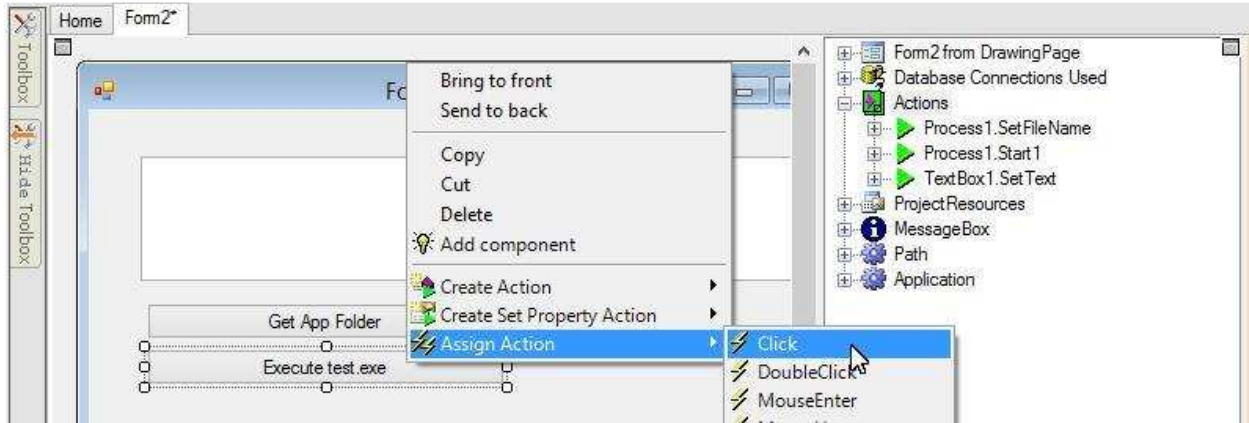

Click OK to finish creating this action:

This action tells the Process component which file to run. To actually run the file, we may create another action using a Start method of the Process component. Right-click the Process component; choose "Create Action"; choose a "Start" method:

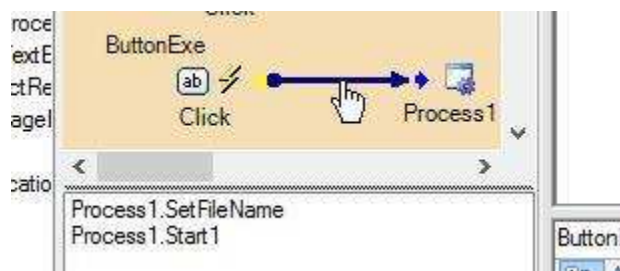Click OK to finish creating this action:

We have created two actions. When these two actions are executed, an EXE file named test.exe under the application folder will be launched. Now the question is that when these two actions should be executed. The answer depends on your programming logic. In this example, we execute these two actions when a button is clicked. So, let's assign these two actions to the Click event of the button. Right-click the button; choose "Assign Action"; choose "Click" event:

Select the two actions we just created:

The two actions are assigned to the button:

That is all for this example.

Feedback

Please send your feedback to support@limnor.com