Create App Configuration Utility

Contents

Introduction

The sample project created in this document can be downloaded from
http://www.limnor.com/studio/ConfigUtil.zip.

Application Configuration component can be used to remember user preferences and settings. See
http://www.limnor.com/support/UseAppConfig.pdf and
http://www.limnor.com/support/UseAppConfigForBranding.pdf. In those samples, runtime
configuration operations are provided within the program to be configured. Runtime configuration
operations may also be programmed into a separate utility program. Thus you have an option to not
allow the users to change the configurations. You may make the configurations and distribute the
configuration file together with the program to the users. This document shows a sample of such usages.

The functionality of this sample project has been built into the newer versions (build # 5.6.1.637 or
higher) of Limnor Studio. You may access it by menu "Tools | Application Configuration Utility".

Sample UI

A very simply UI is shown below. It uses an OpenFileDialog component for selecting a program to do
configurations. It uses a set of buttons for setting configuration values; creating new set of
configurations; selecting a set of configurations to be used when the program starts. It uses an
Application Configuration component to form configuration operations.
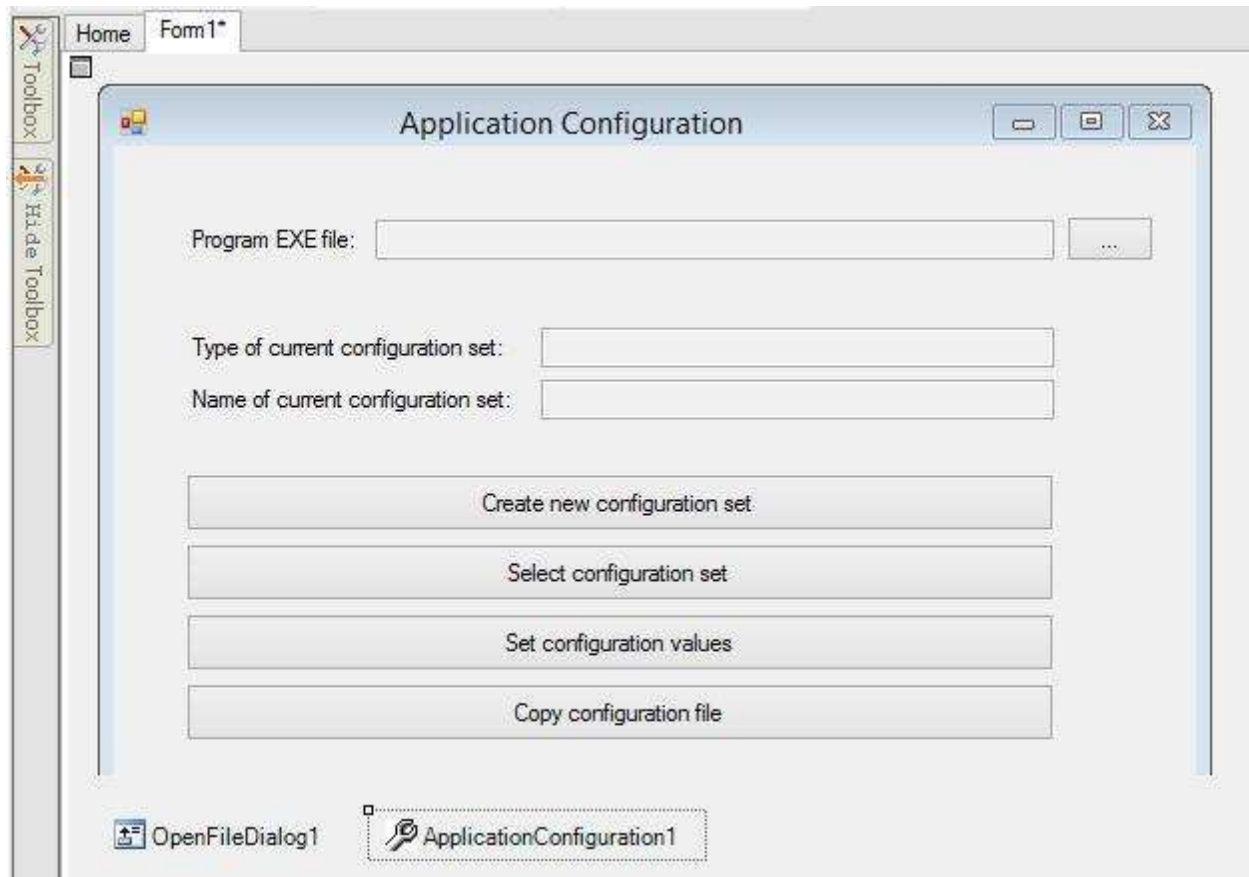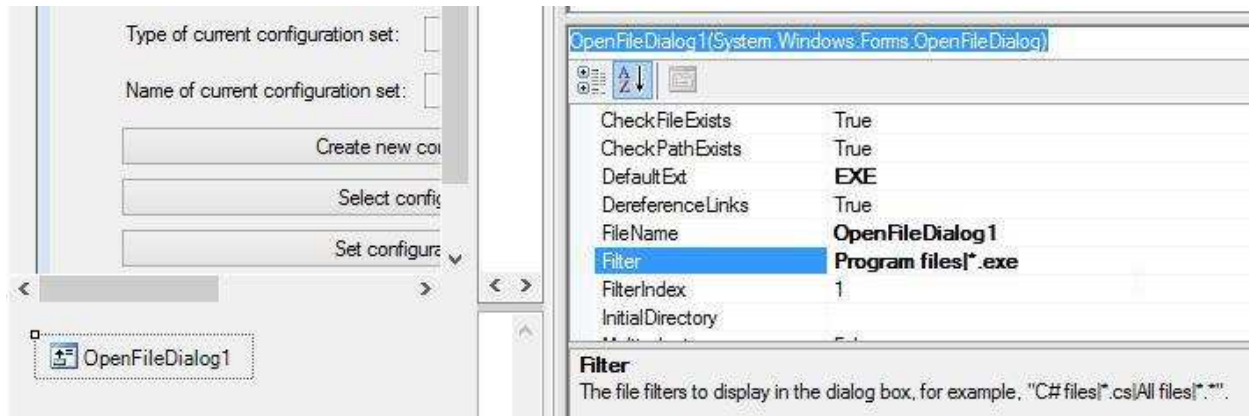
Select Program

If you develop a configuration utility for a specific program then you may hard-code the program name. We develop this utility program for performing configurations for all programs which use Application Configuration component. So, we need to allow the user to select the program to be configured.

Use OpenFileDialog

We use an OpenFileDialog for the user to select a program to be configured.
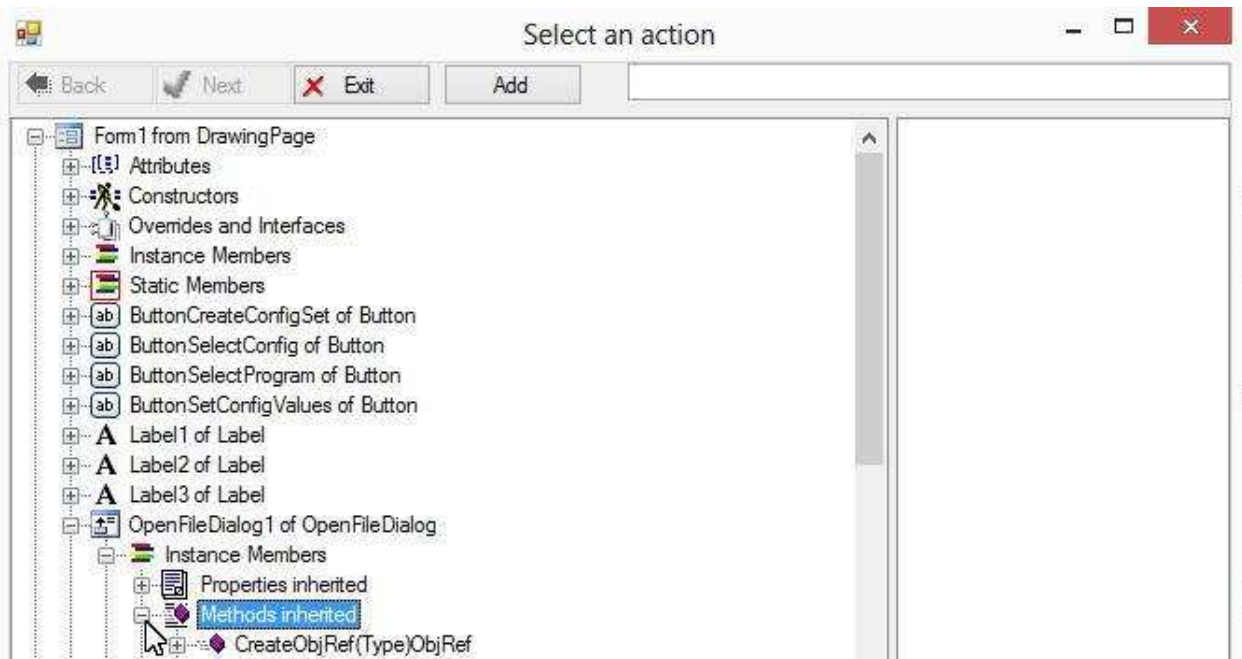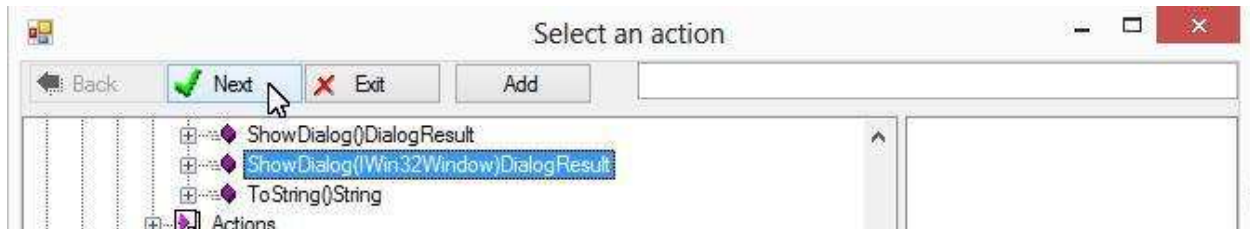
Set properties, DefaultExt, Filter, and Title, for the OpenFileDialog:

Launch OpenFileDialog

We use a button to launch the OpenFileDialog:

Select ShowDialog method:

Select the current form as the dialogue box owner:
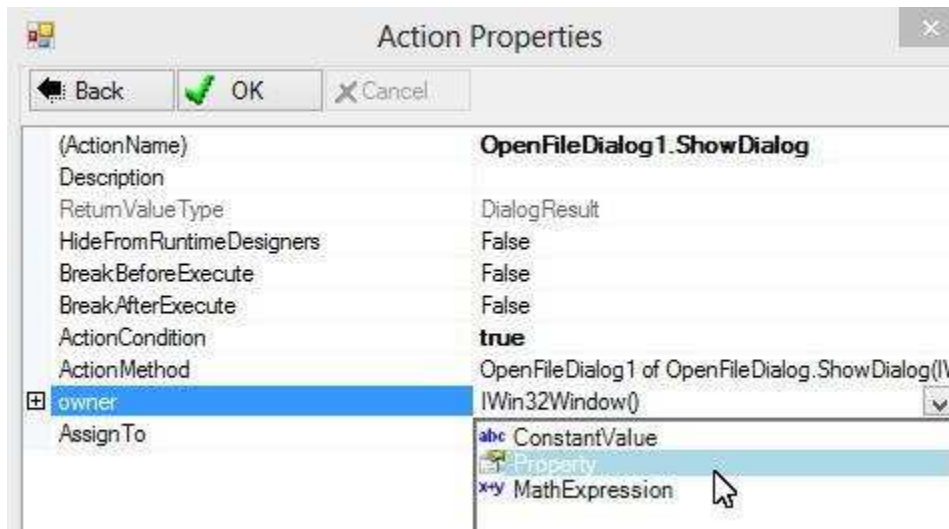
Click OK:
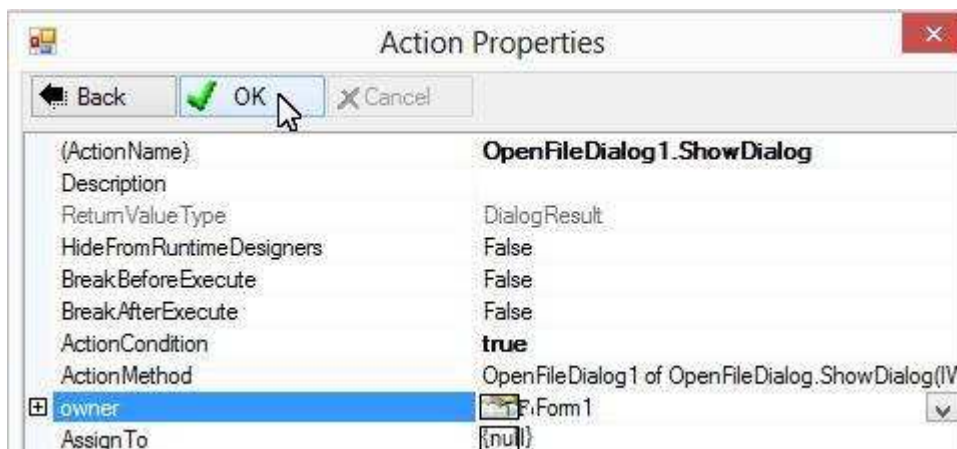
The action is created and assigned to the button:
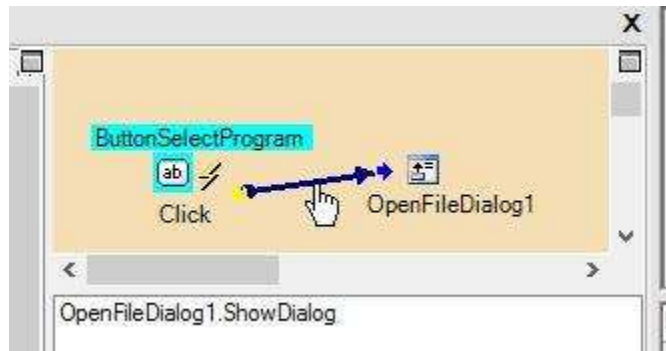
Apply program selection

When the user makes a program selection, event FileOK of the OpenFileDialog occurs. We may handle this event to apply program selection. Right-click the OpenFileDialog; choose "Assign Action"; select "FileOK" event:
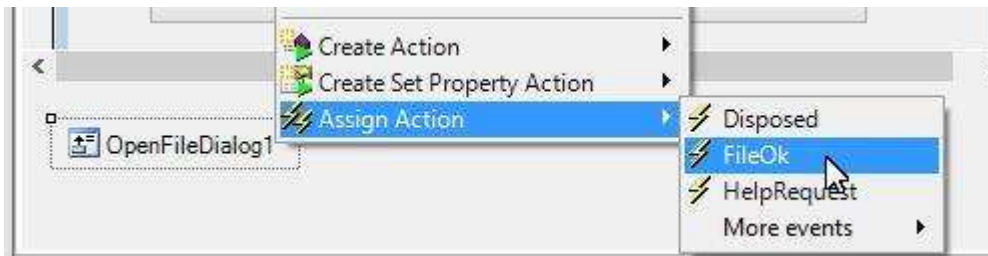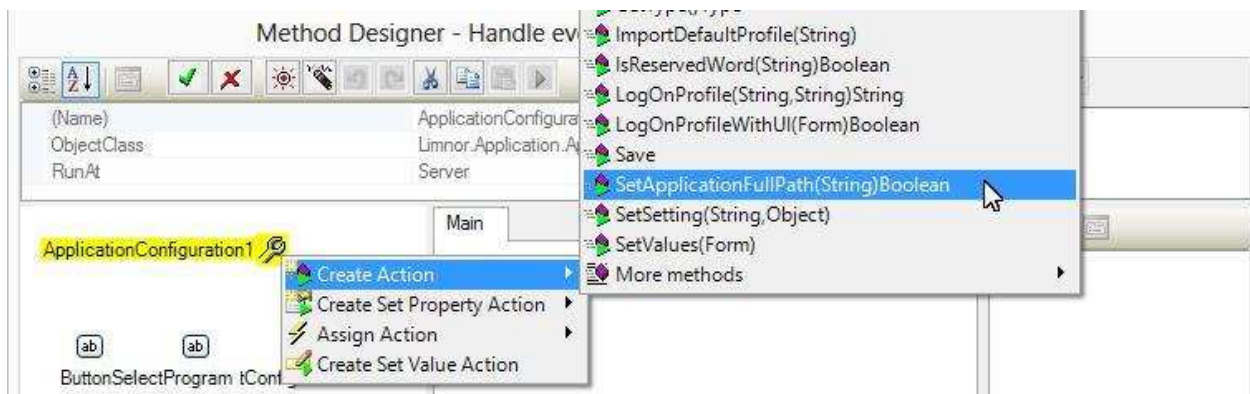

Choose "New event handler method for FileOK":
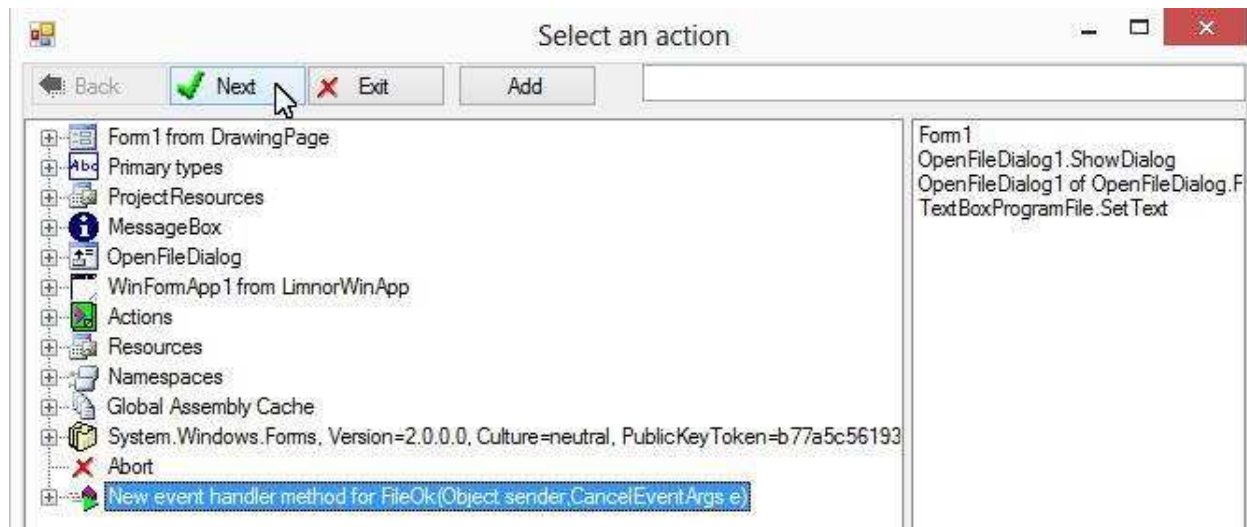
A Method Editor appears for developing the event handler:

Assign selected program to Application Configuration

Create a SetApplicationFullPath action to use the selected program:

Use the file selection for "exePath":
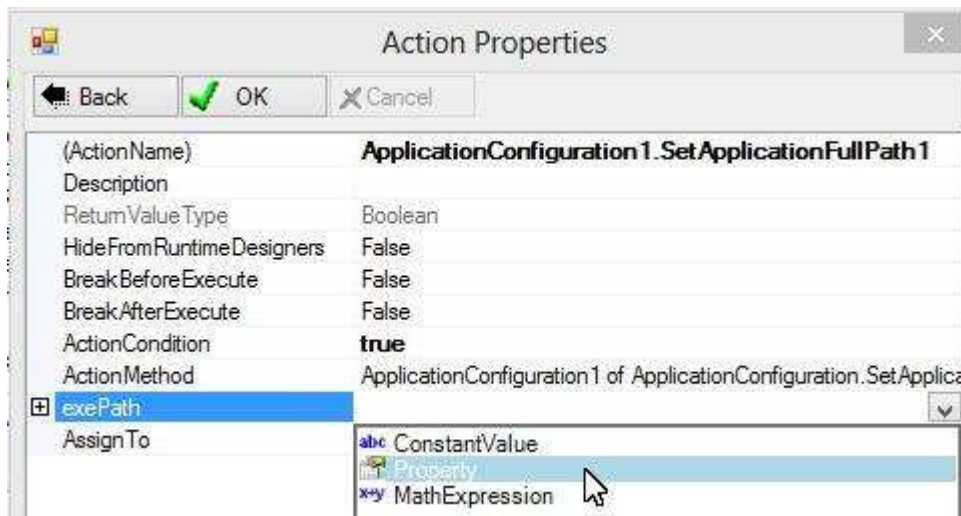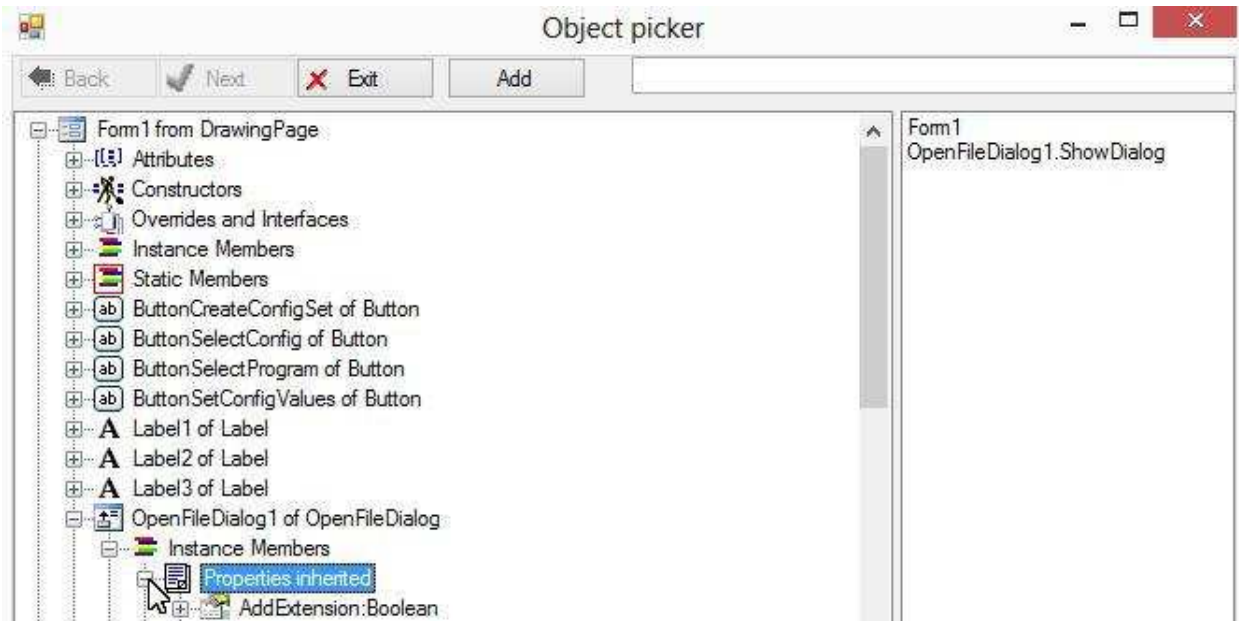
Click OK:
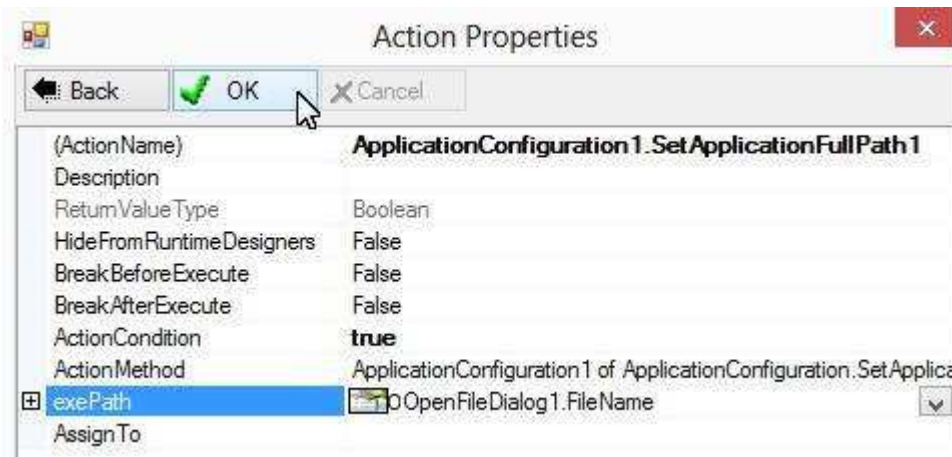
The action is created and appears in the Action Pane:
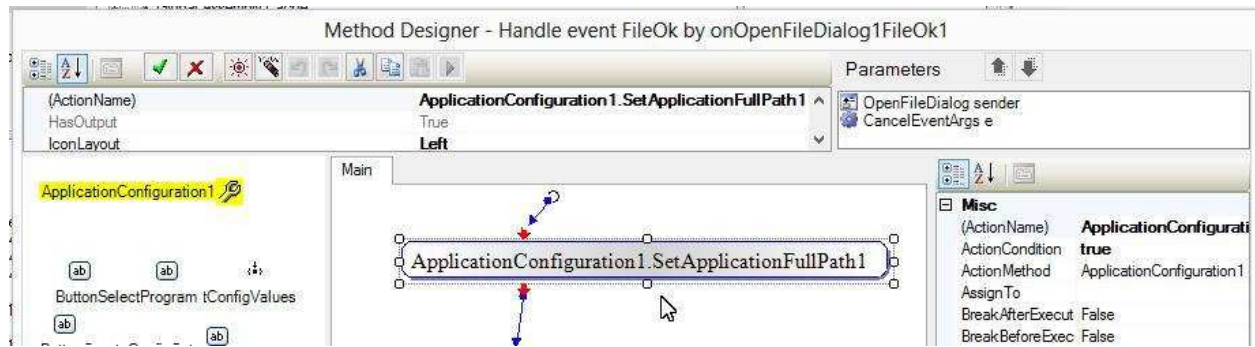
Add a Condition action to check the execution result of the above action:
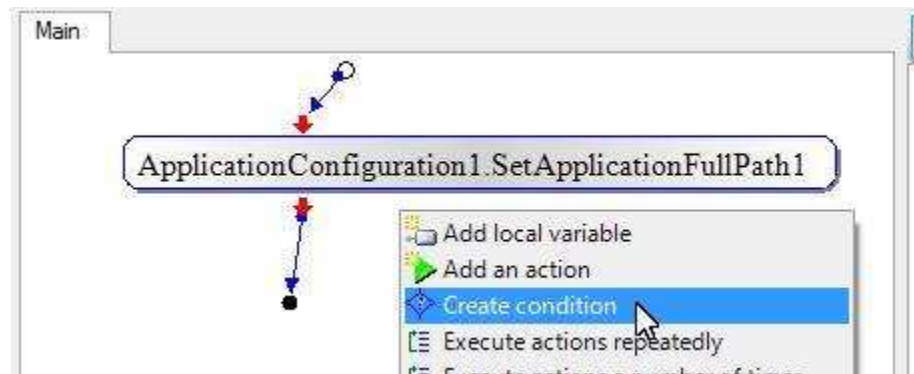
Link the output of the above action to the new Condition action:
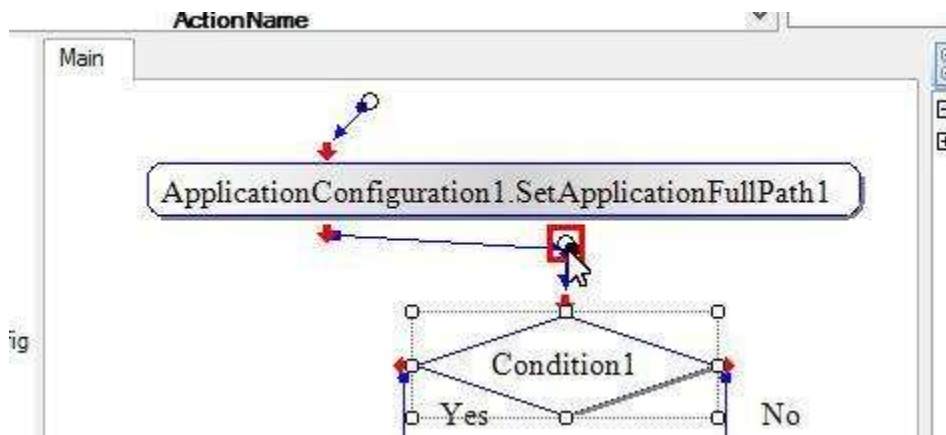
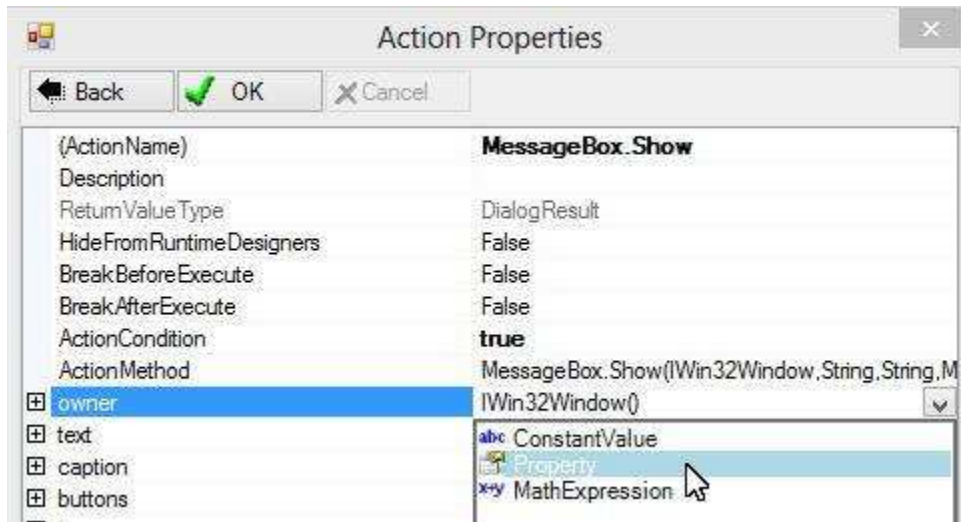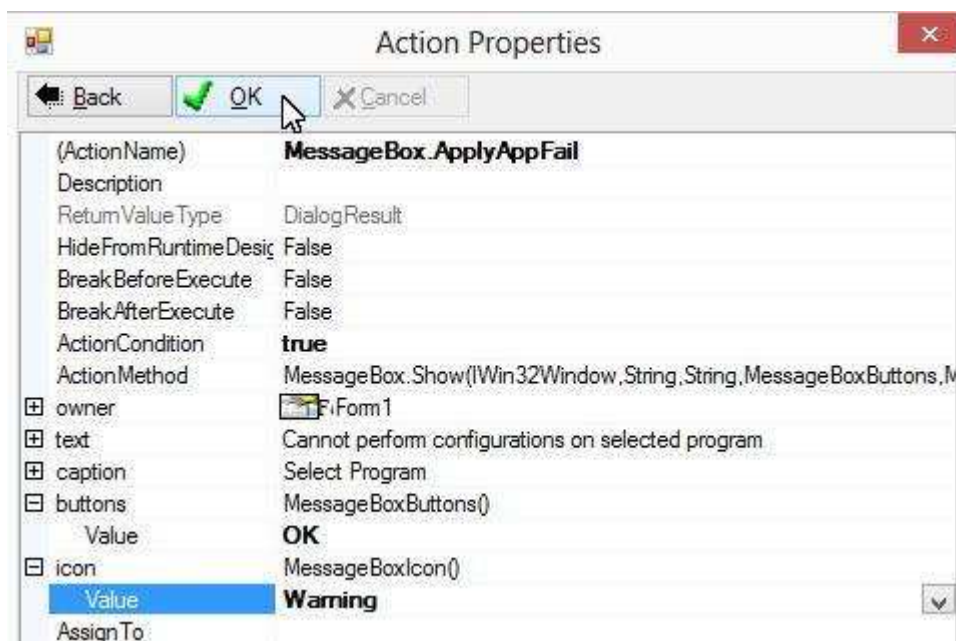Rename the Condition action:
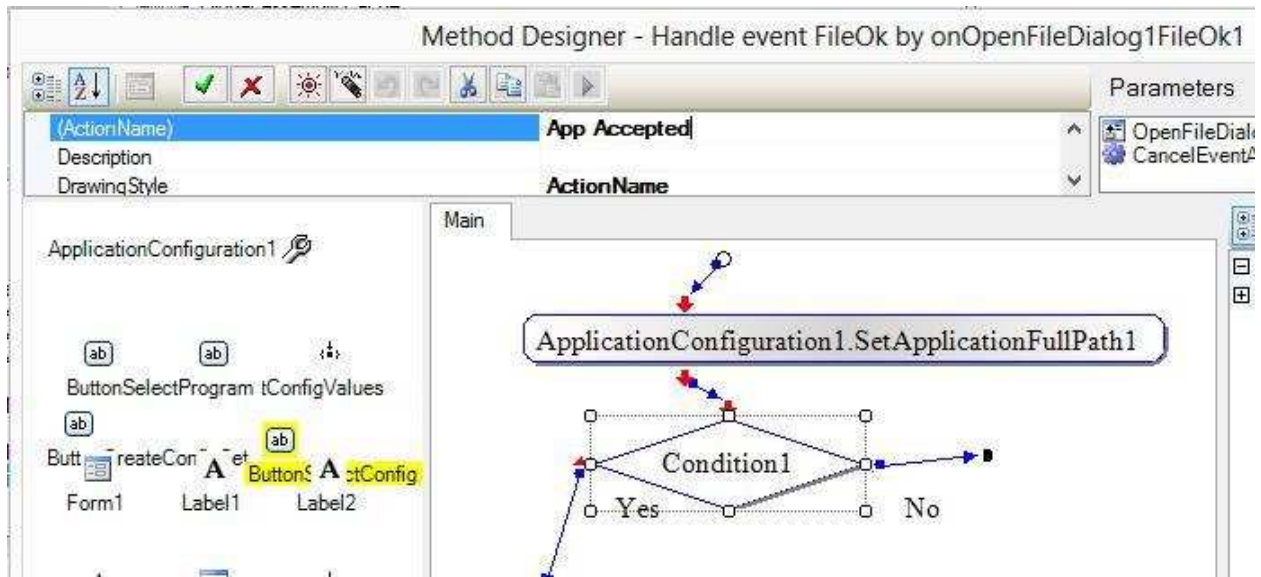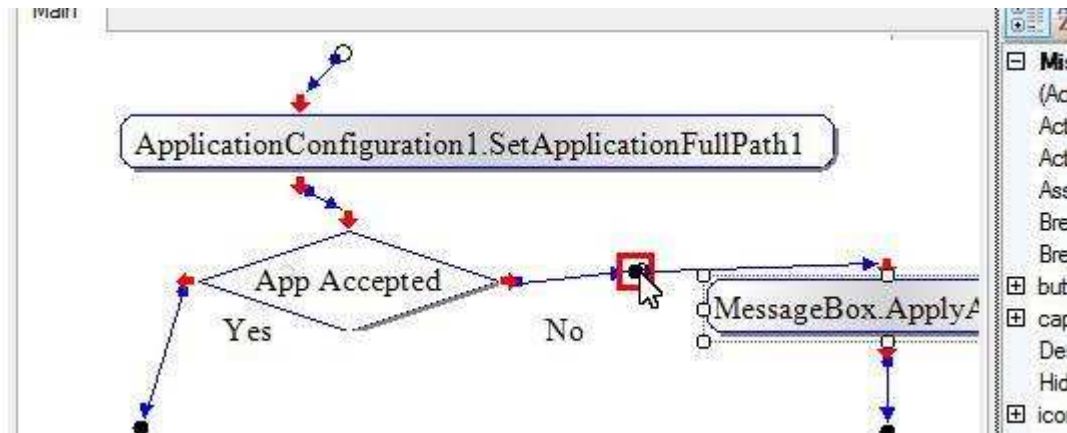
Add a message box to show execution failure:
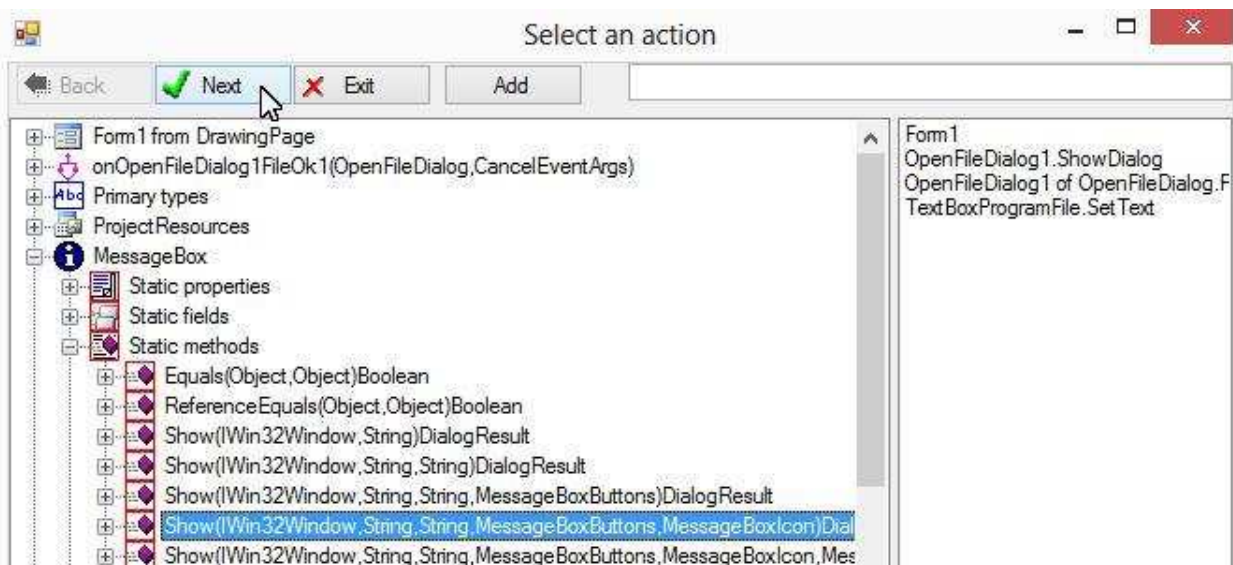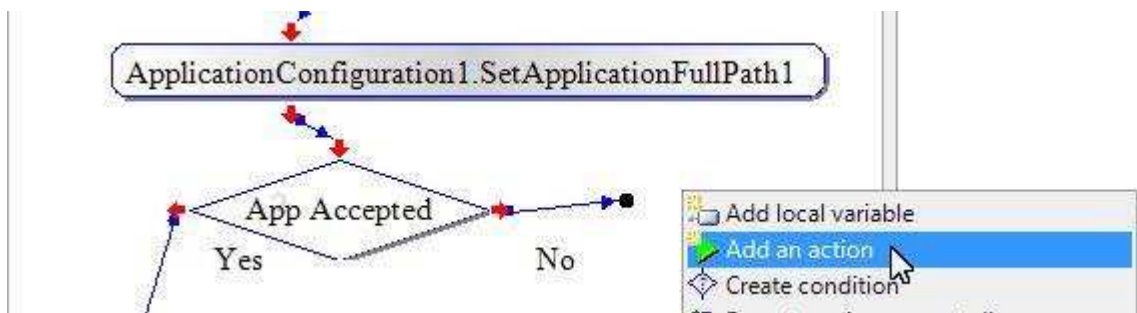
Use the current form as the owner of the message box:

Set other properties of the action and rename the action. Click OK:

Link the new action to the No port of the Condition action:

Show selected program

We want to show the file selection on the text box for program name. Right-click the text box; choose "Create Set Property action"; select Text property of the text box:
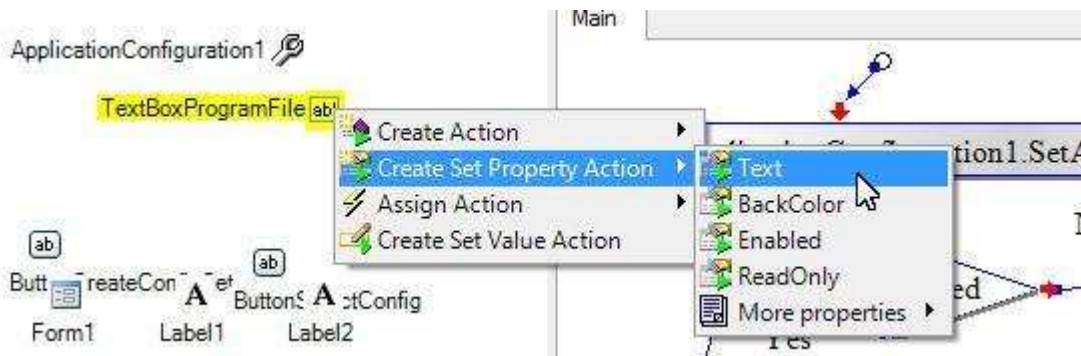
For "value" of the action, choose Filename property of the OpenFileDialog:
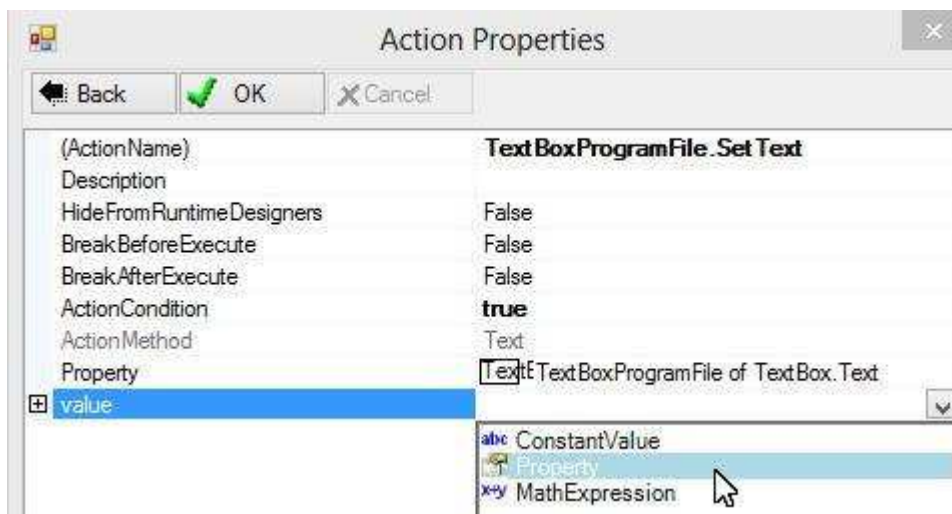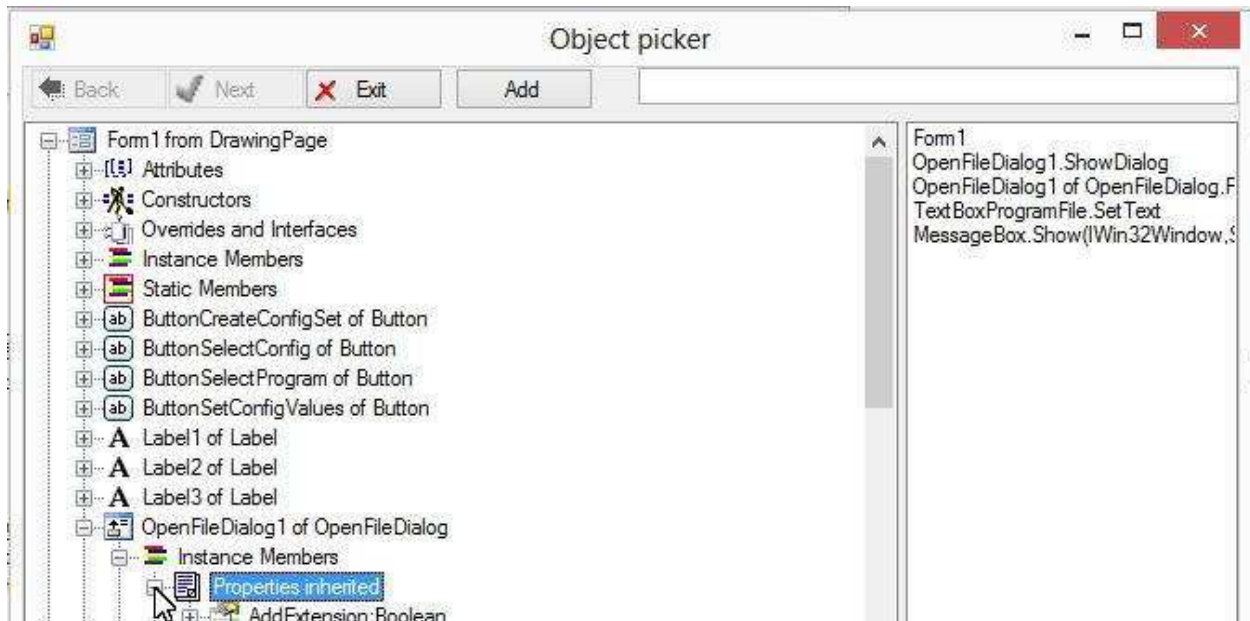
Click OK:
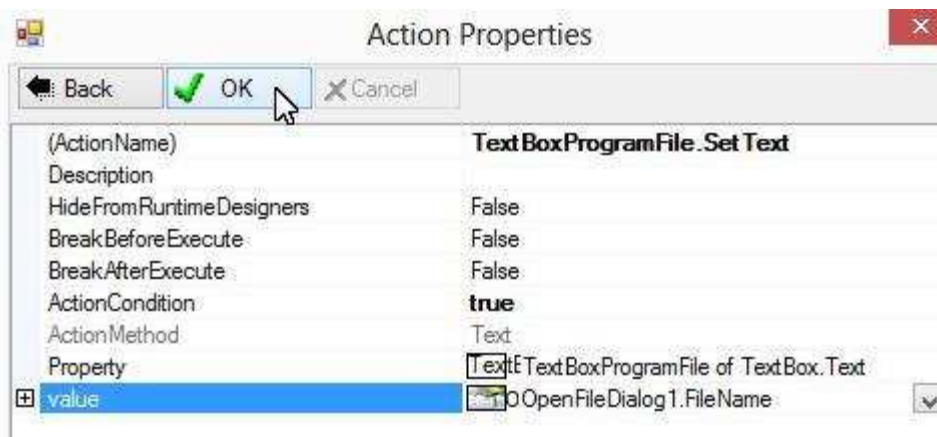
The action is created and appears in the Action Pane. Link it to the Yes port of the Condition action:
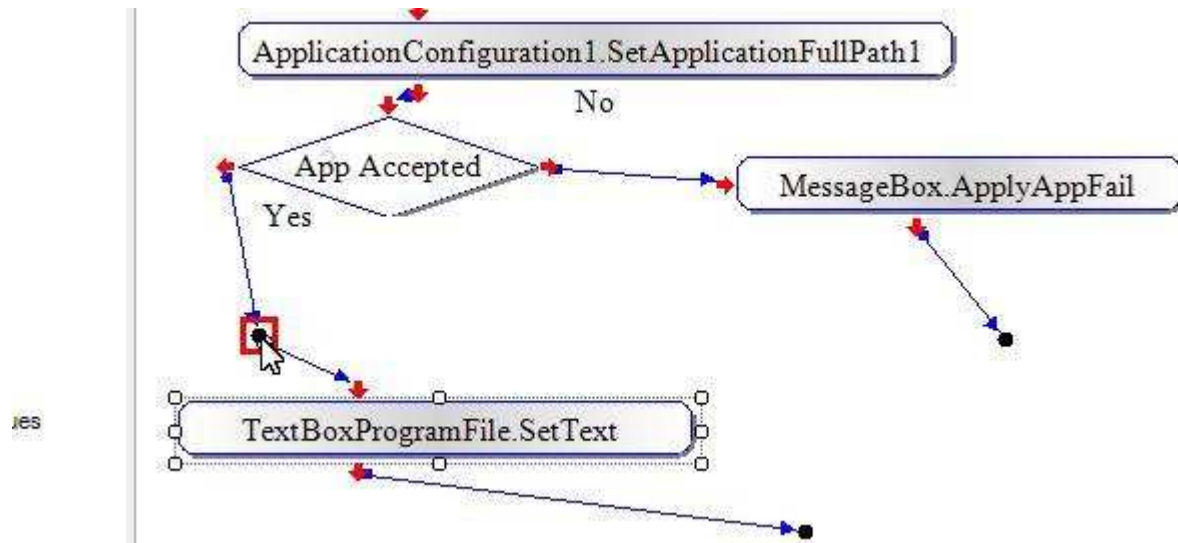
Enable configuration buttons

Initially all configuration buttons are disabled. On applying the selected program to the Application Configuration component, we need to enable these buttons.

Set "value" to True:
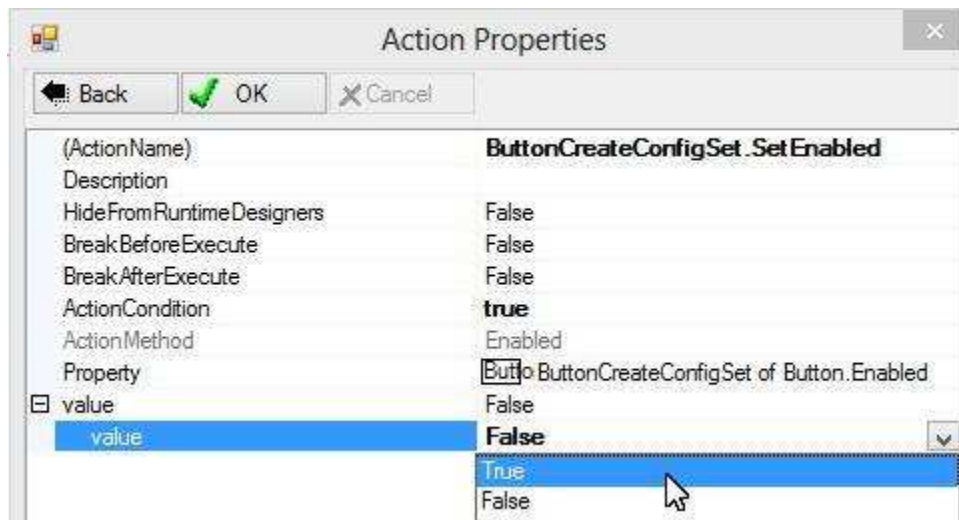
Click OK:
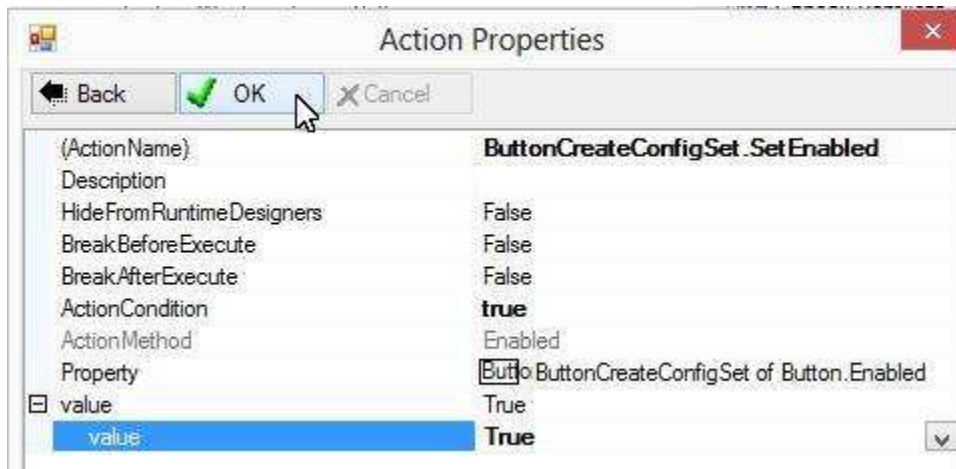
Link the new action to the last action:
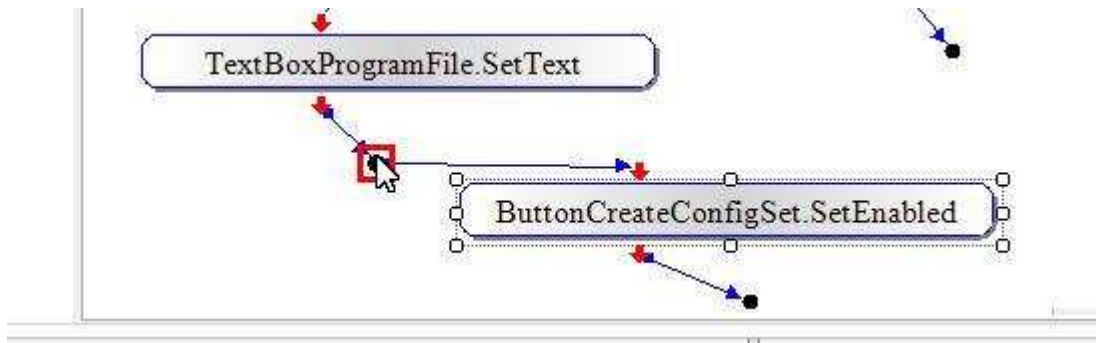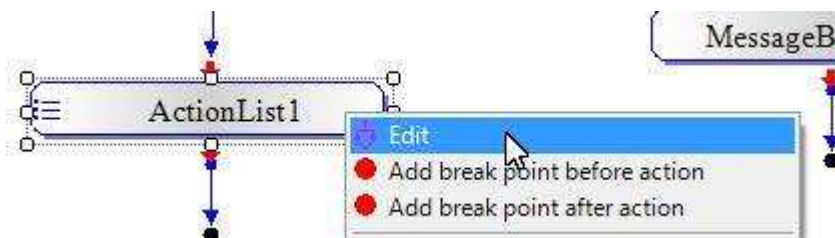

Do the same for the other 2 buttons.
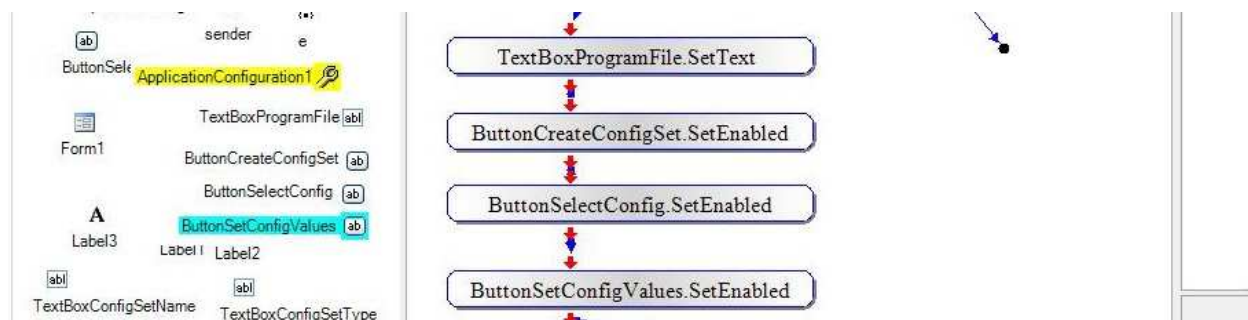
Let's group the actions into an action list to take less space:
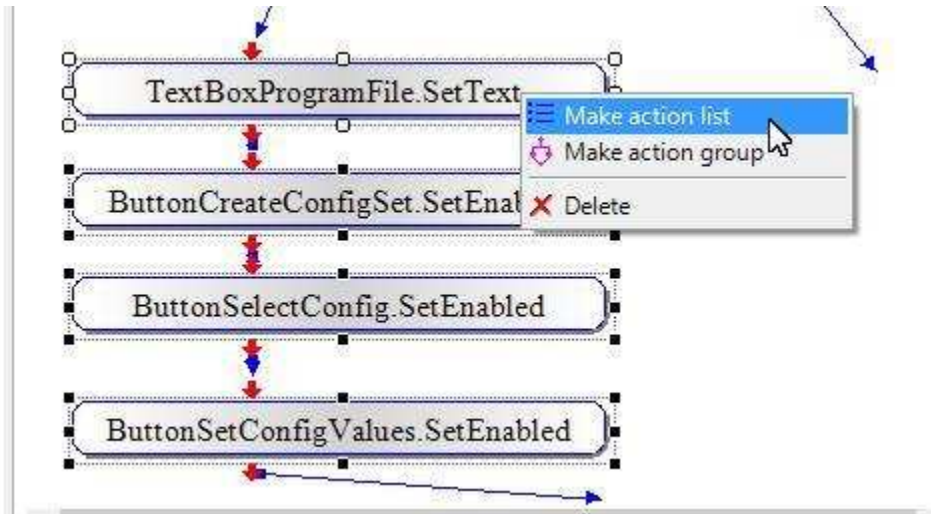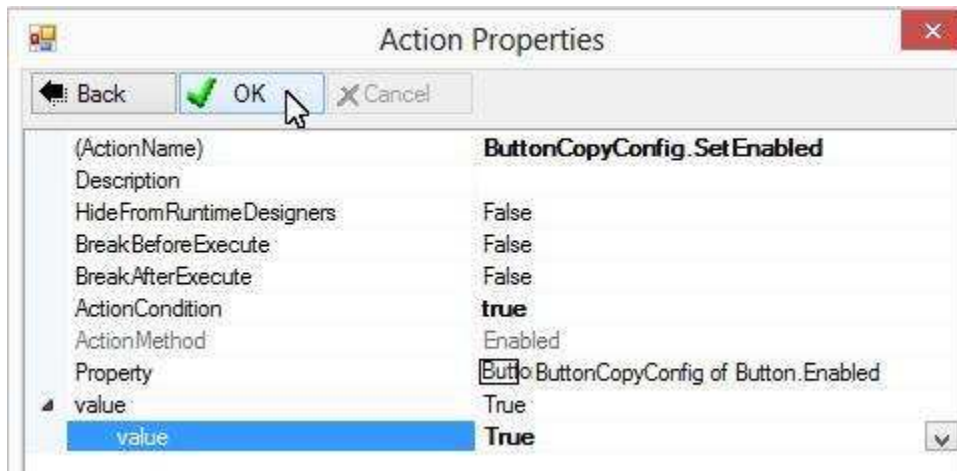
Now it looks cleaner:
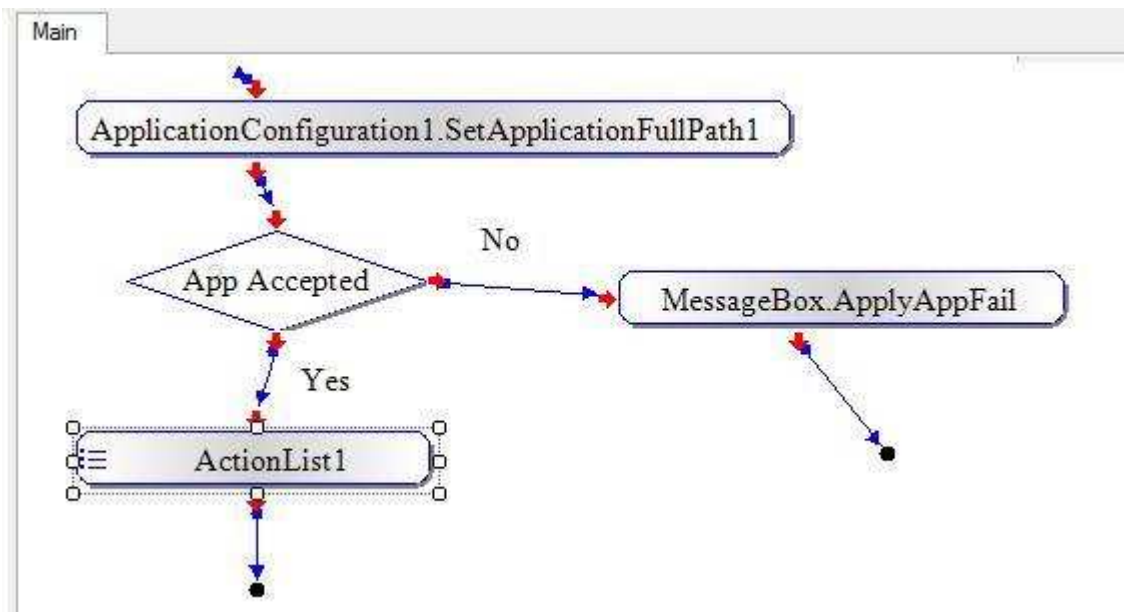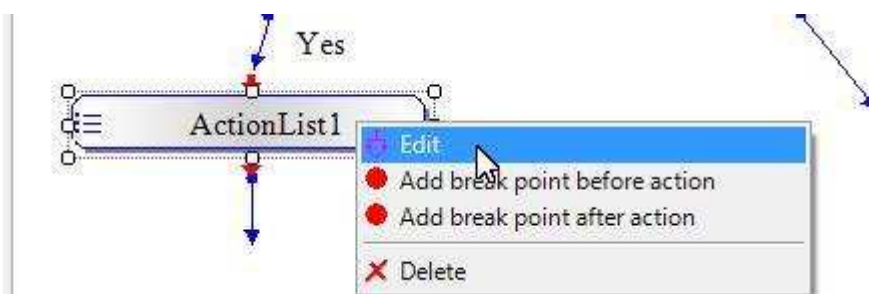

You may edit the action list:

Add a new action to the list:
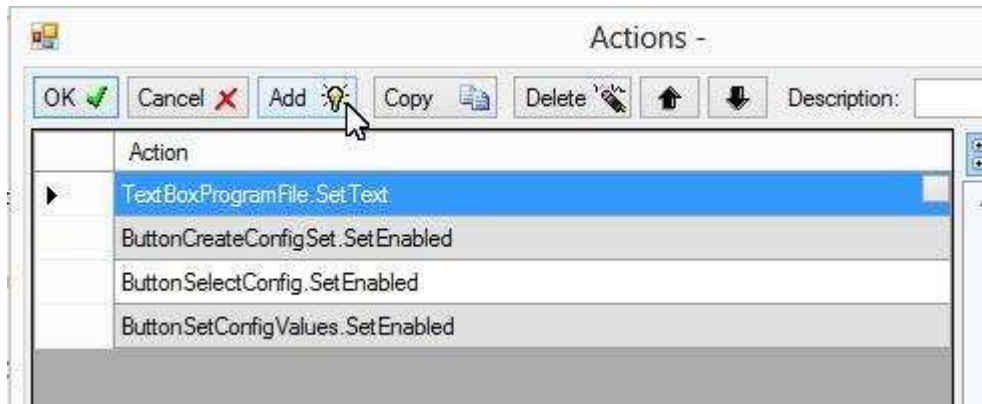
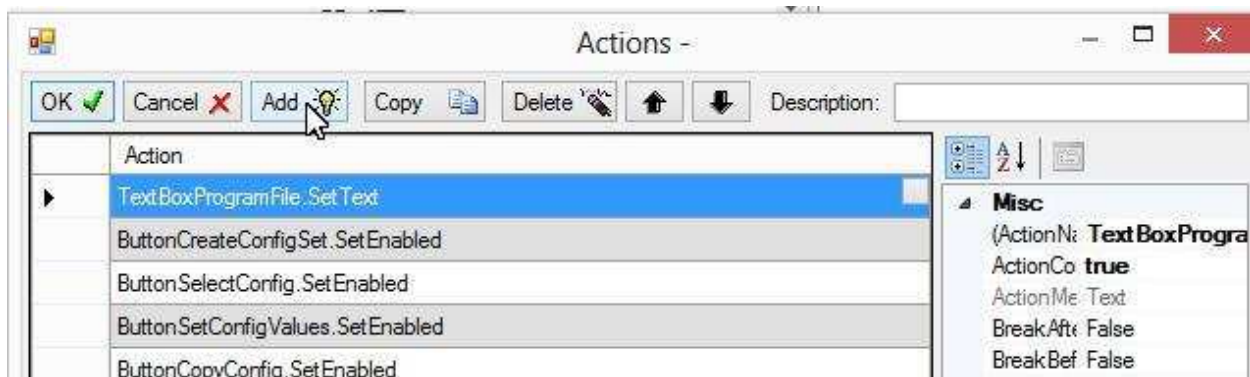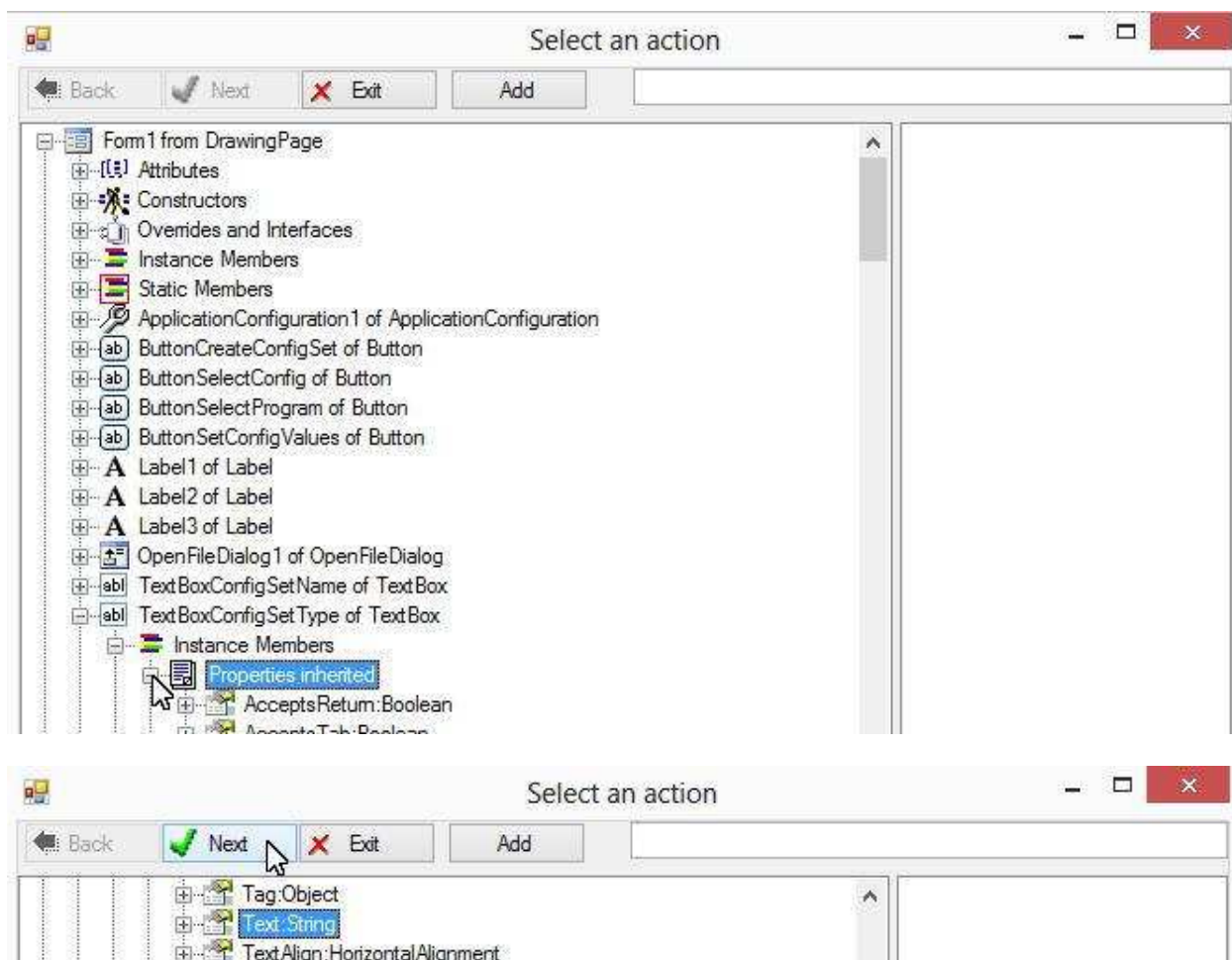We want to enable the copy button:
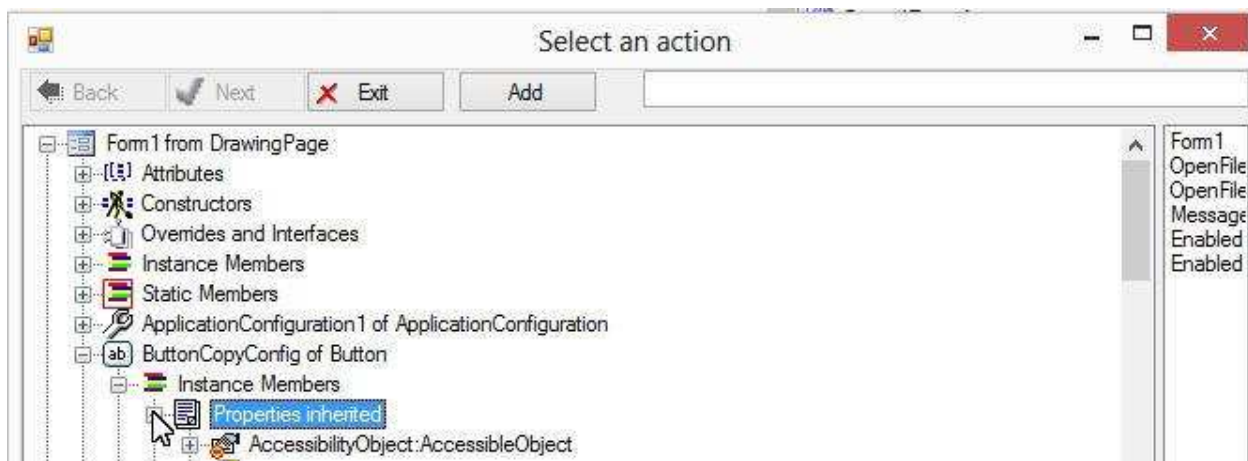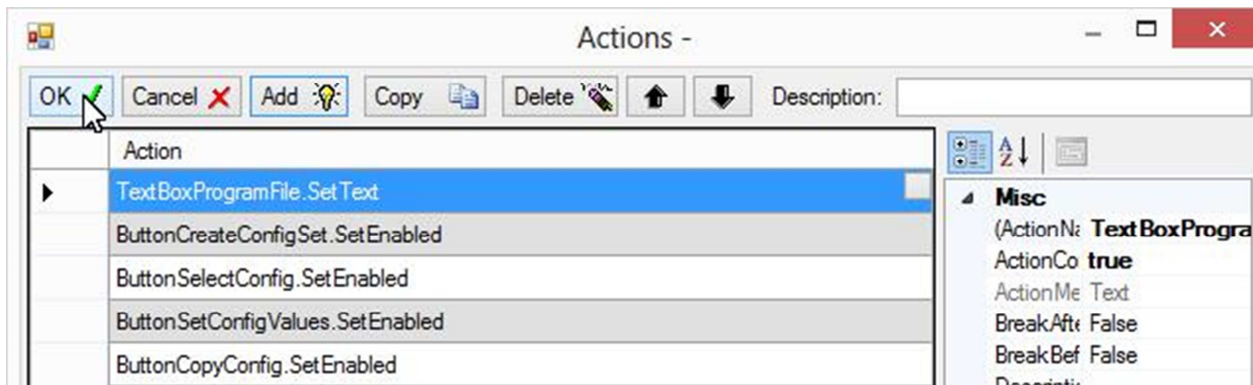
Show Type of Configuration Set

Once a program is selected, we may show the type of its current configuration set. We may do it by adding a new action into the action list:
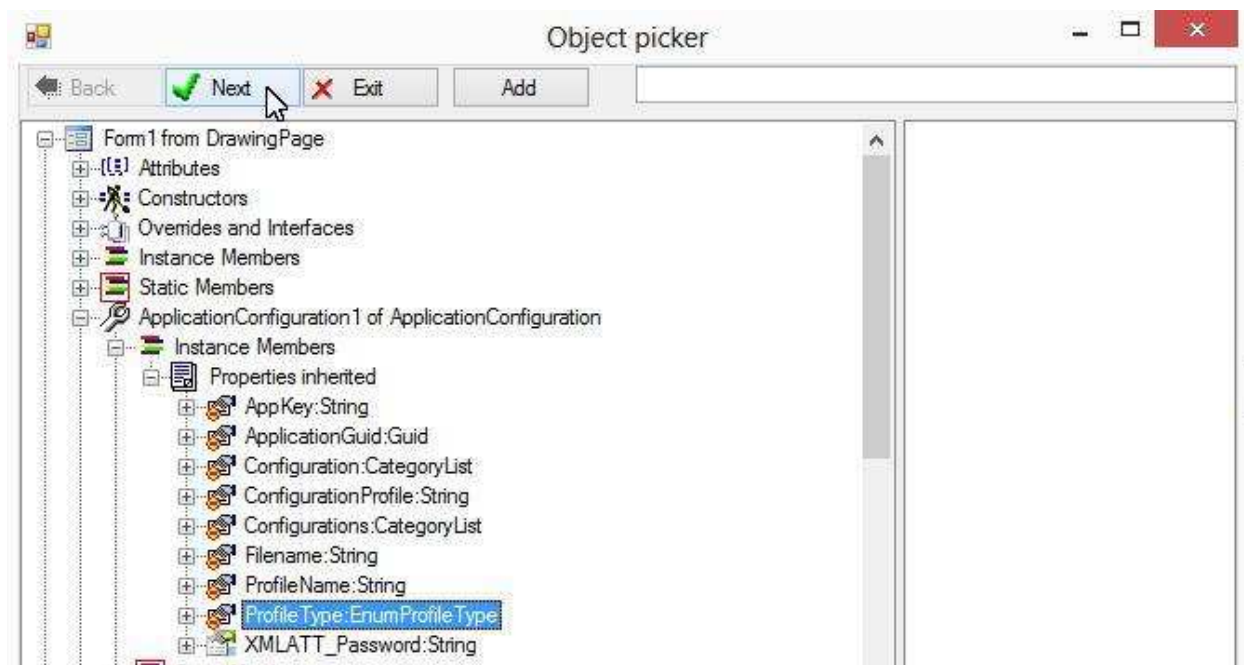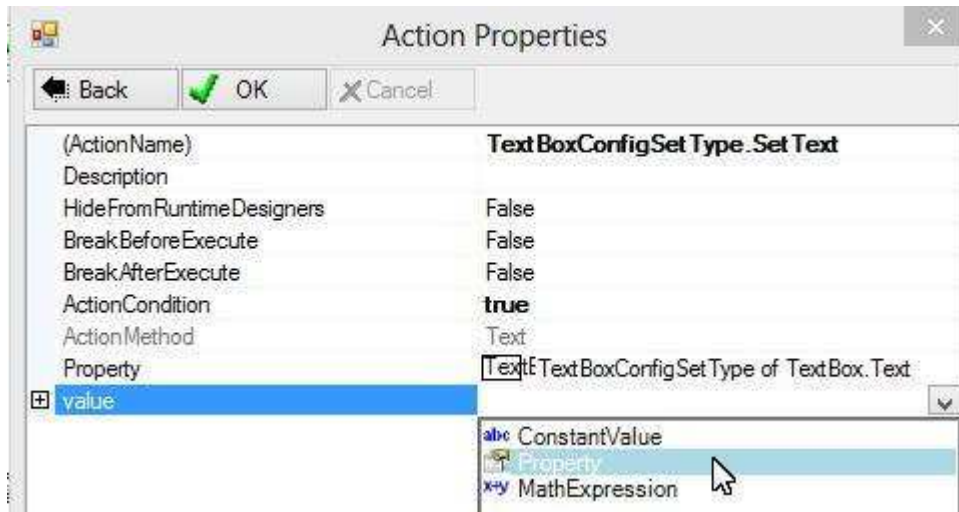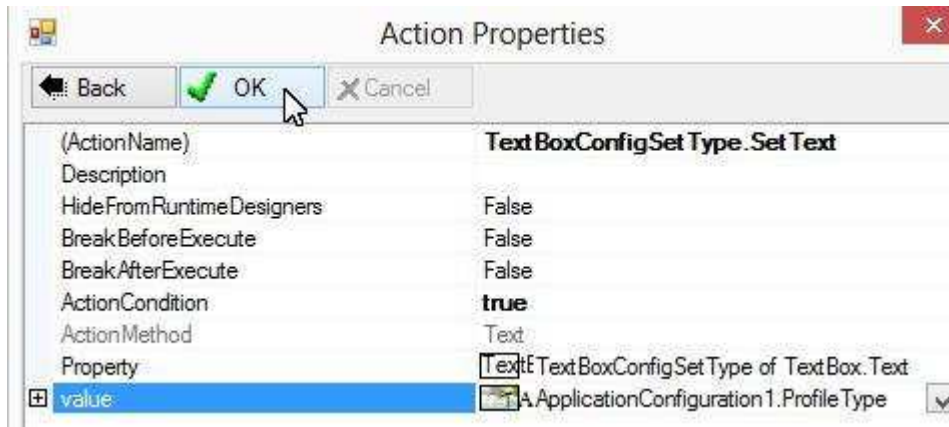
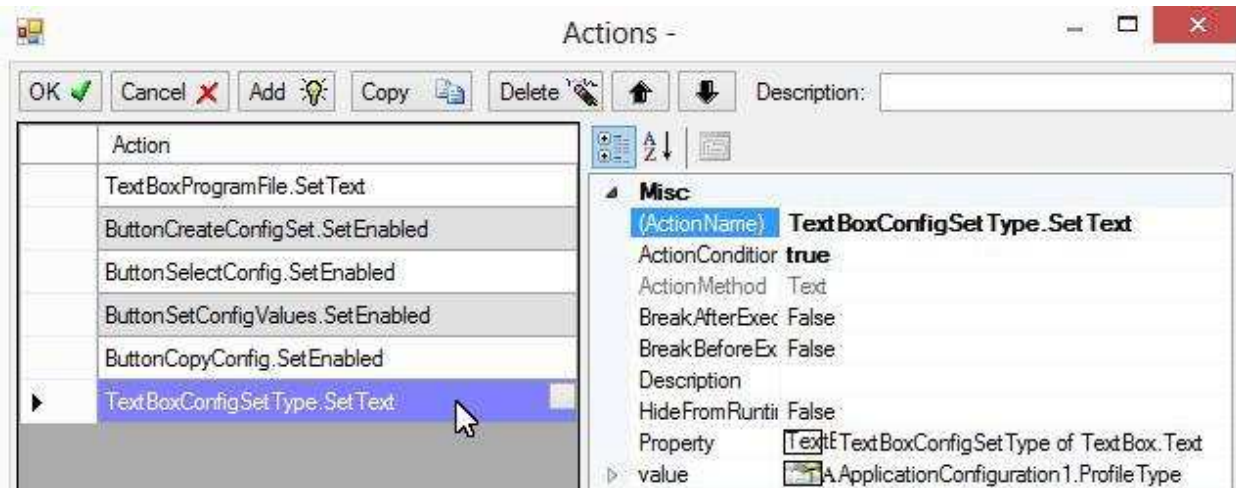Select the Text property of the text box for configuration type:

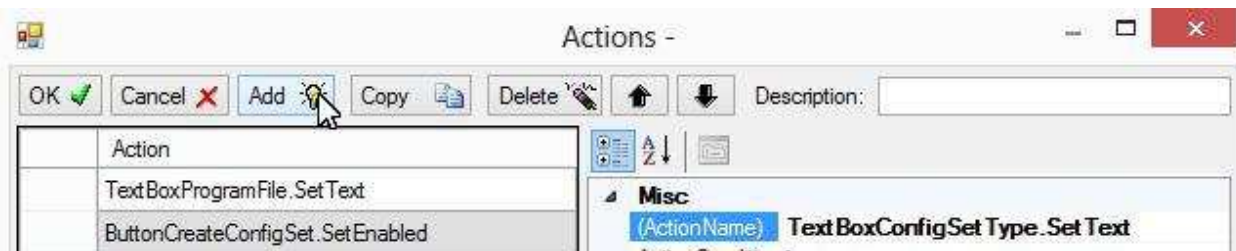Select property ProfileType of the Application Configuraton component:

Click OK:

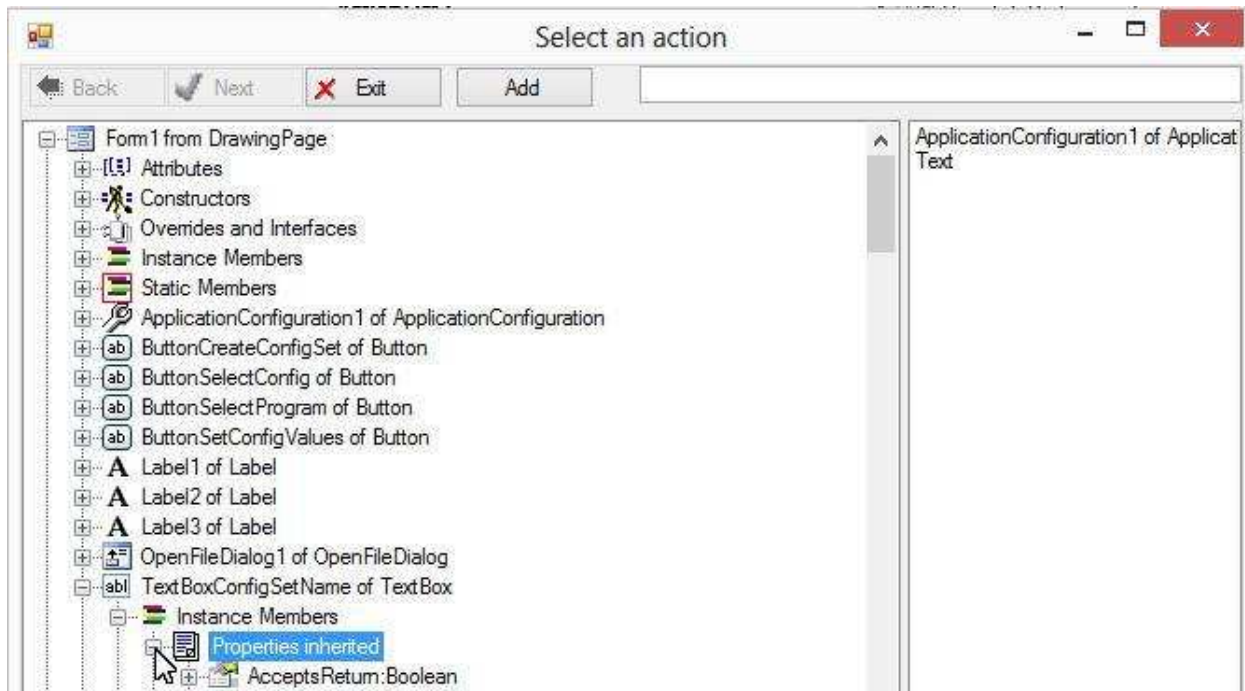The action is created and added to the action list:
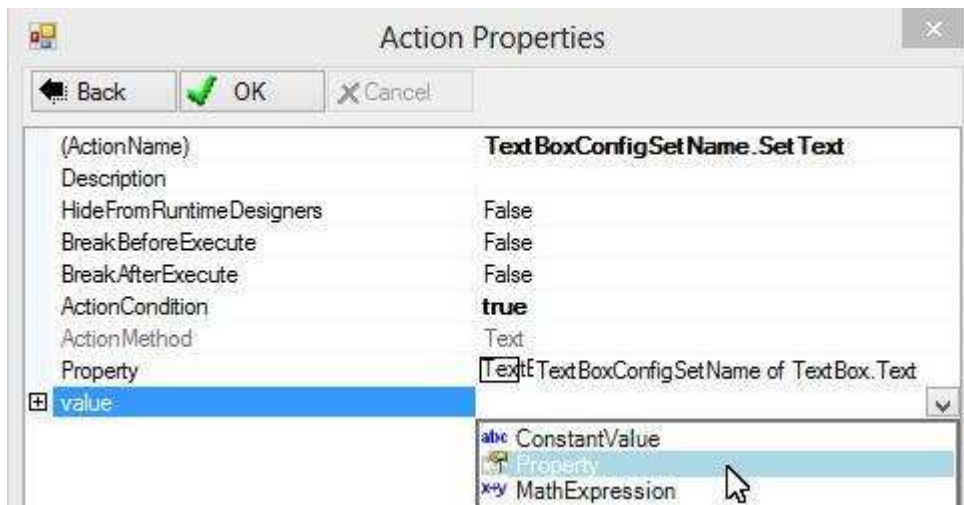

Show name of configuration set

Once a program is selected, we may show the name of its current configuration set, if it is a named configuration set. We may do it by adding a new action into the action list:
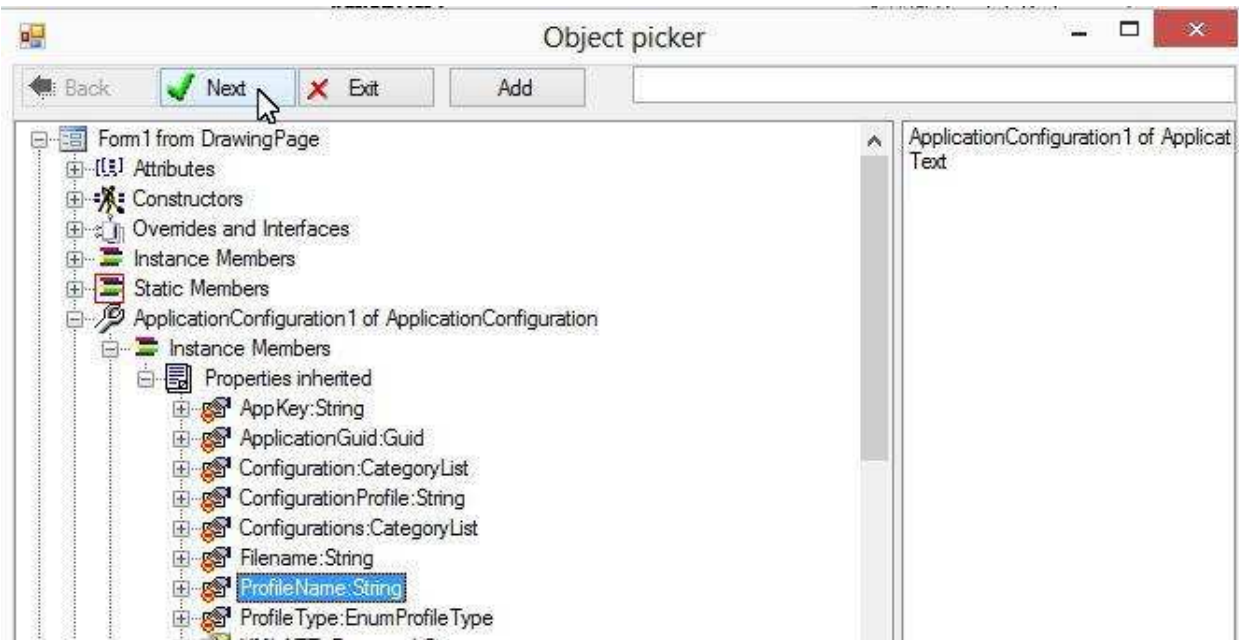

Select the Text property of the text box for showing the name of configuration set:
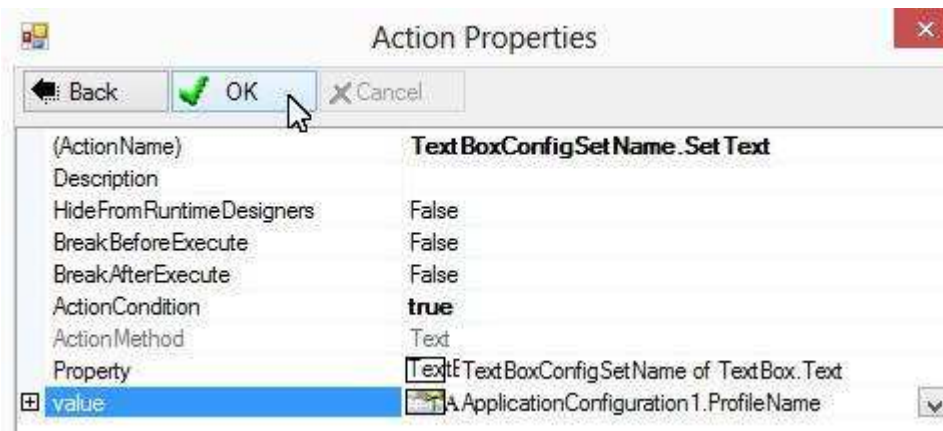
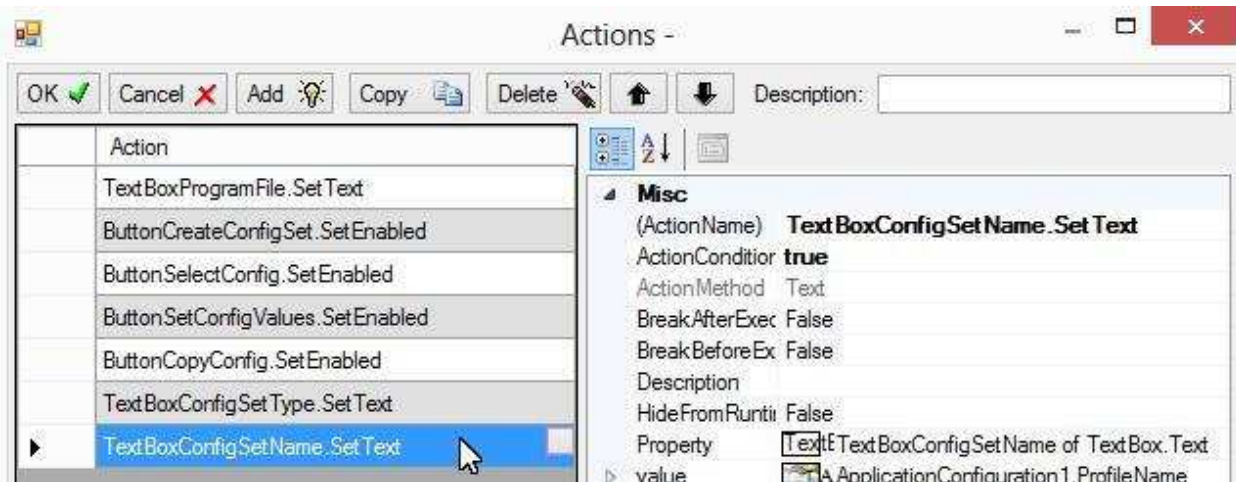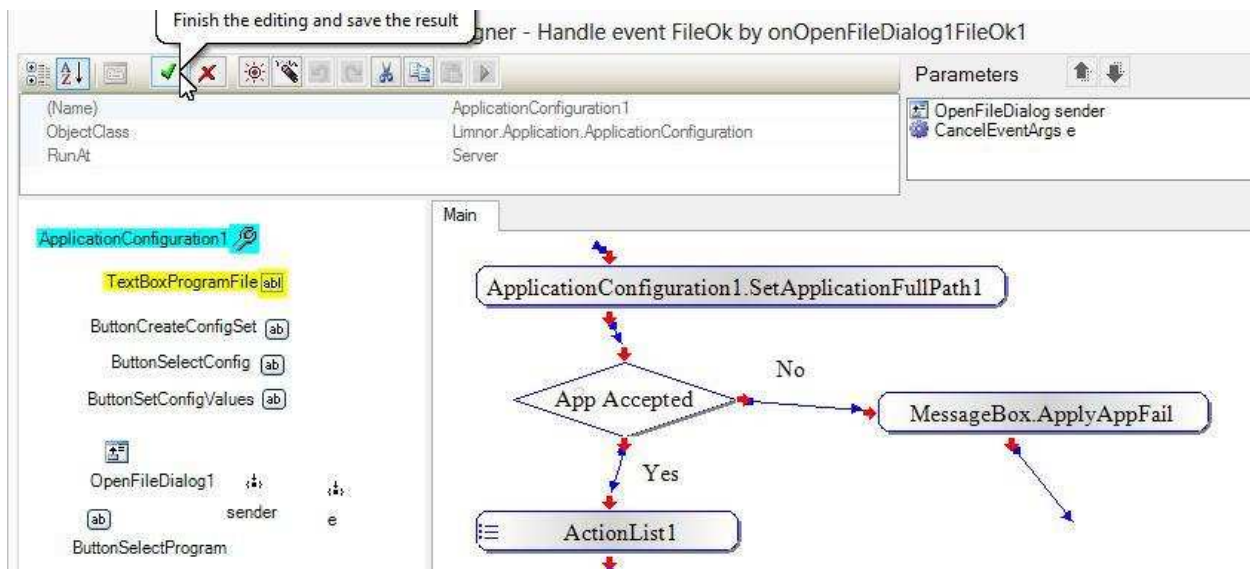Select ProfileName property of the Application Configuration component:
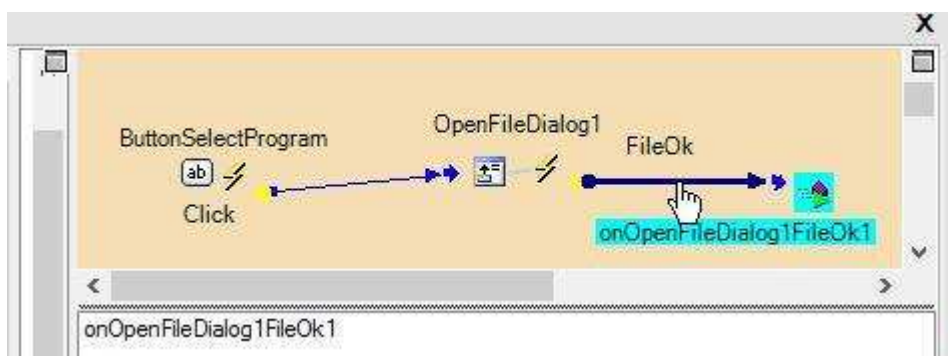
Click OK:

The action is created and added to the action list:

This is our event handler for event FileOK.

On event FileOK, this event handler will be executed:

Make Configurations to Selected Program

Once a program is selected, we can make configurations to it.
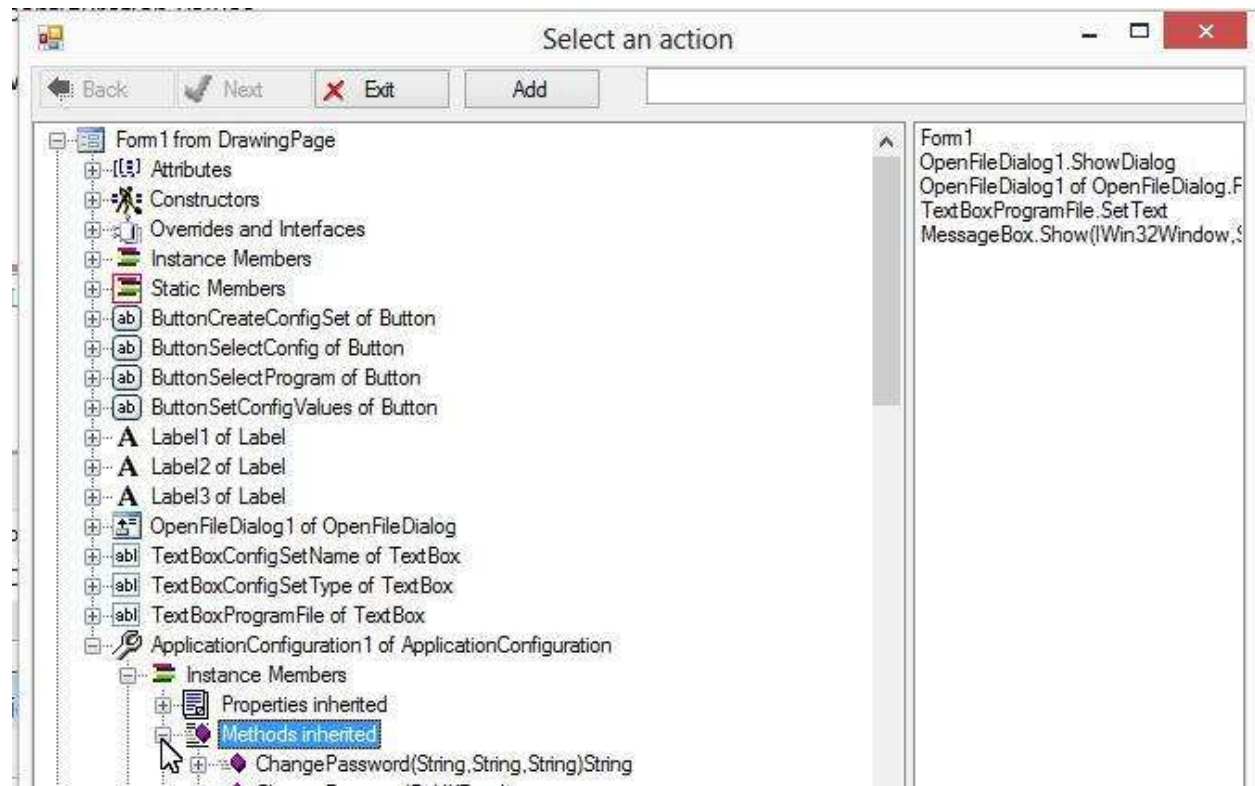
Create Configuration Sets

In other documents, a set of configuration values is called a "Profile" or a "Brand". Multiple sets of configuration values may co-exist for an application. An application may switch between sets of configuration values.

Launch dialogue

We use a button for creating new sets of configuration values:

Choose CreateProfile method:

Use the current form as the dialogue box owner:
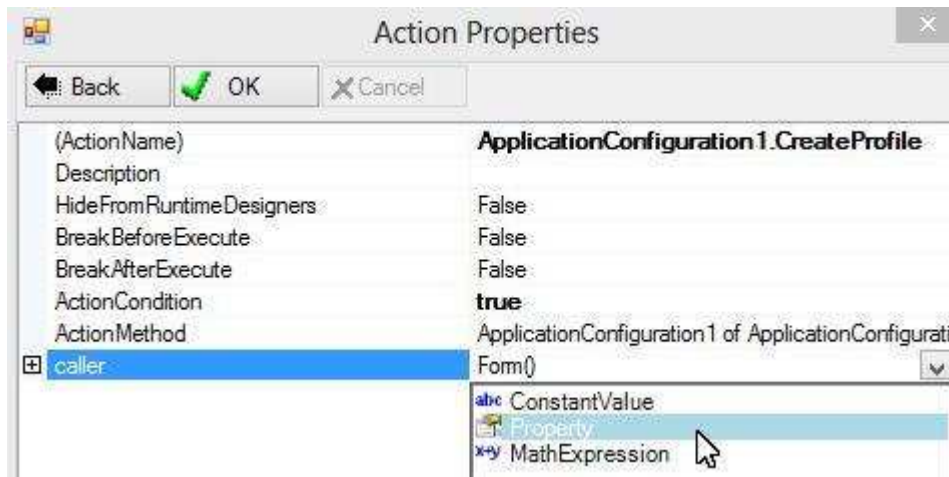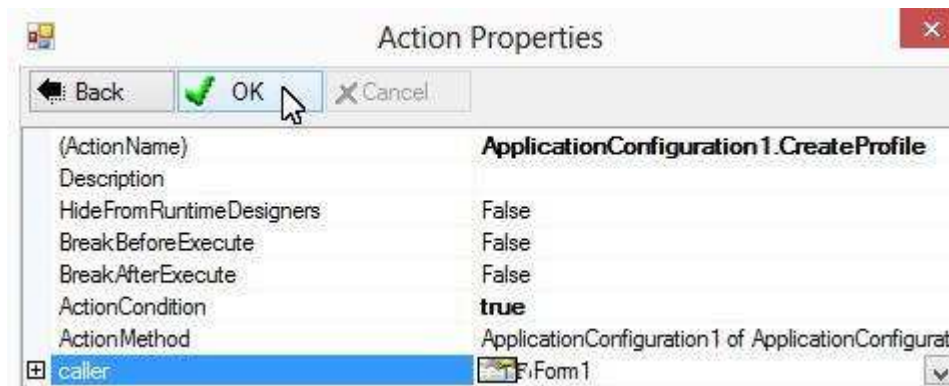
Click OK:

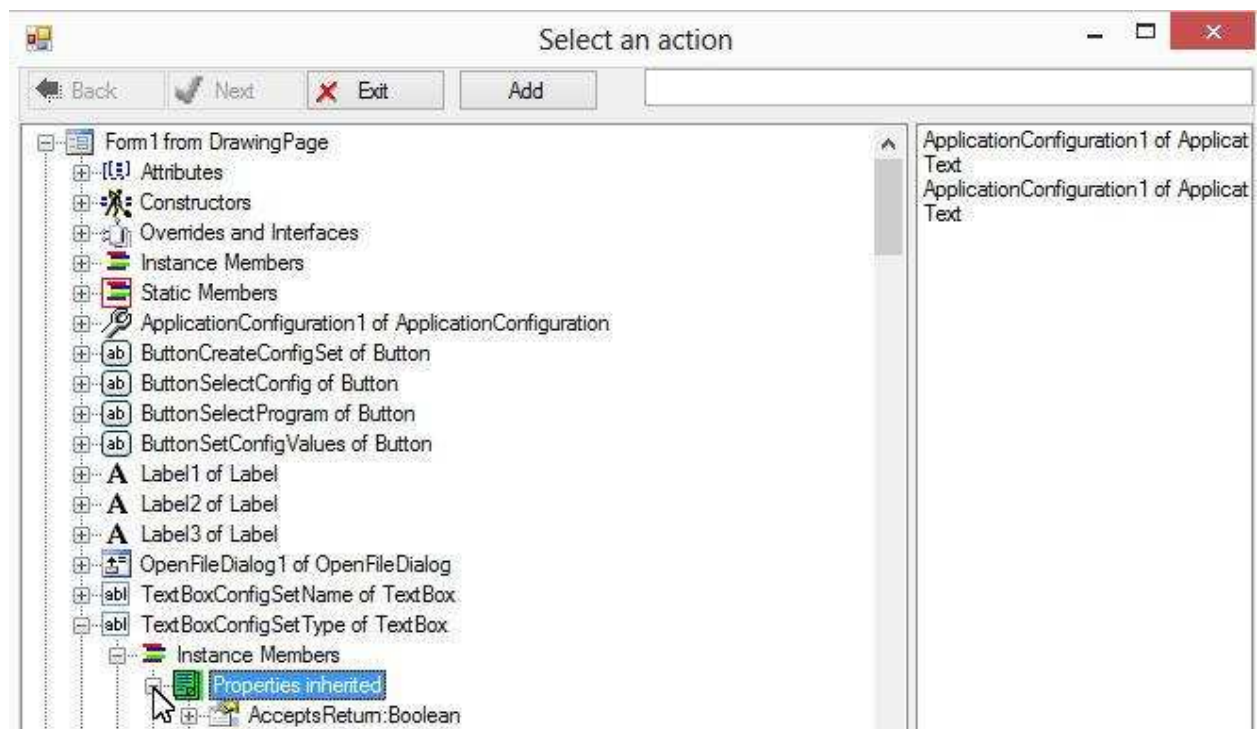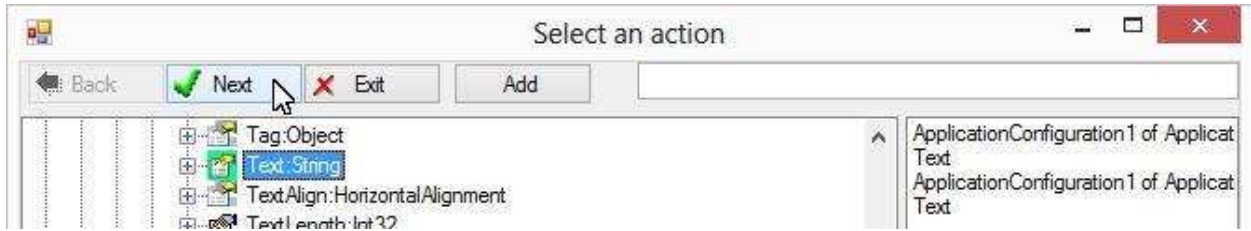The action is created and assigned to the button:
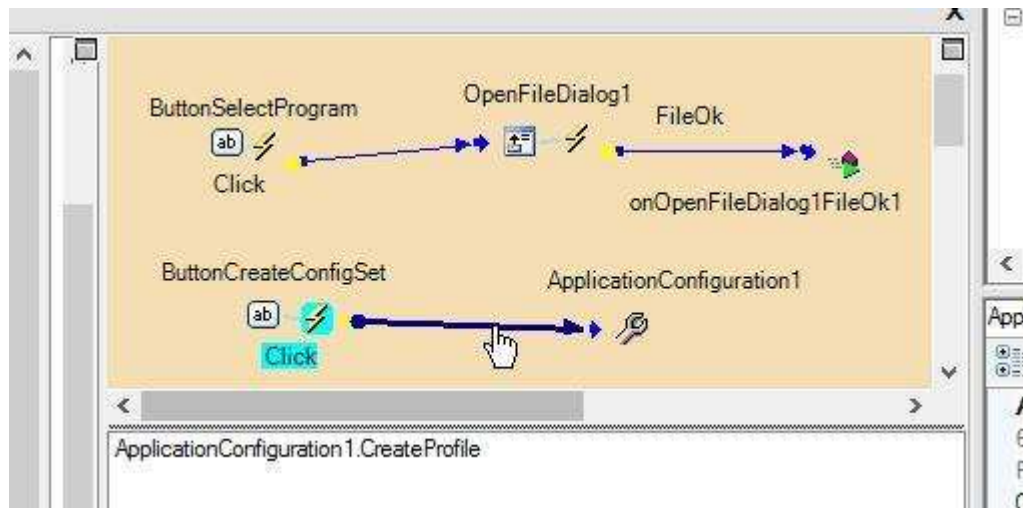
Show type of configuration set

Newly created configuration set will be the current configuration set. Let's show its type after the dialogue box is closed.
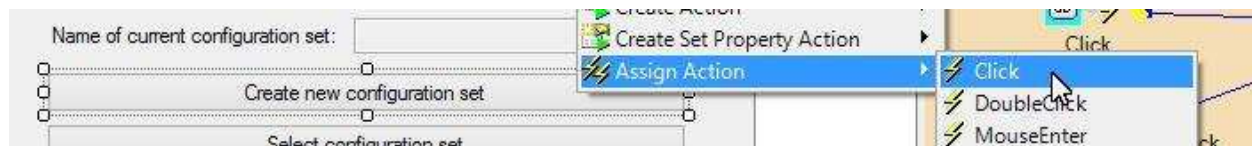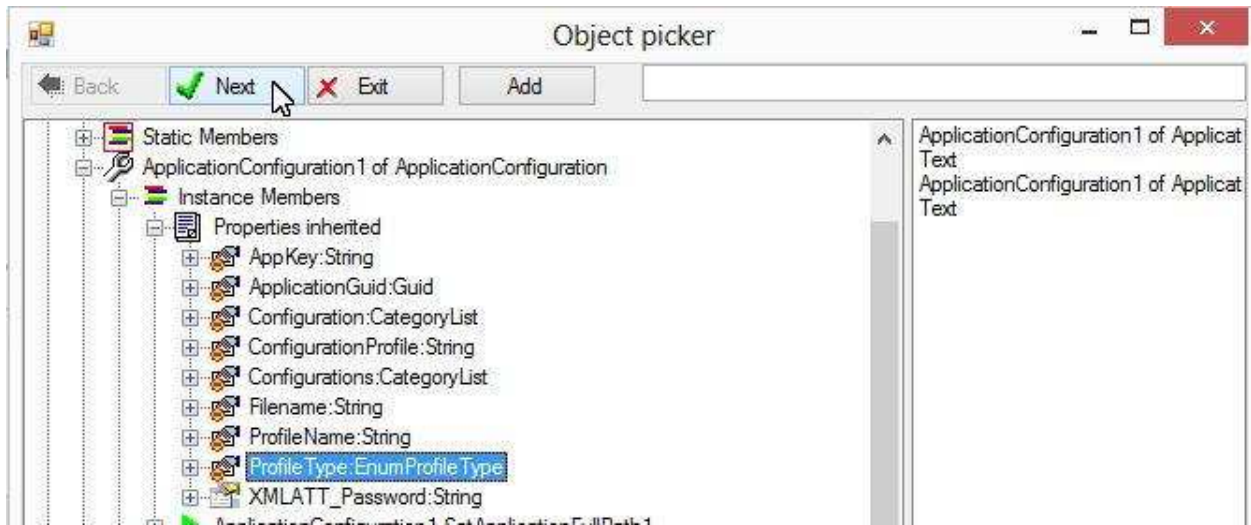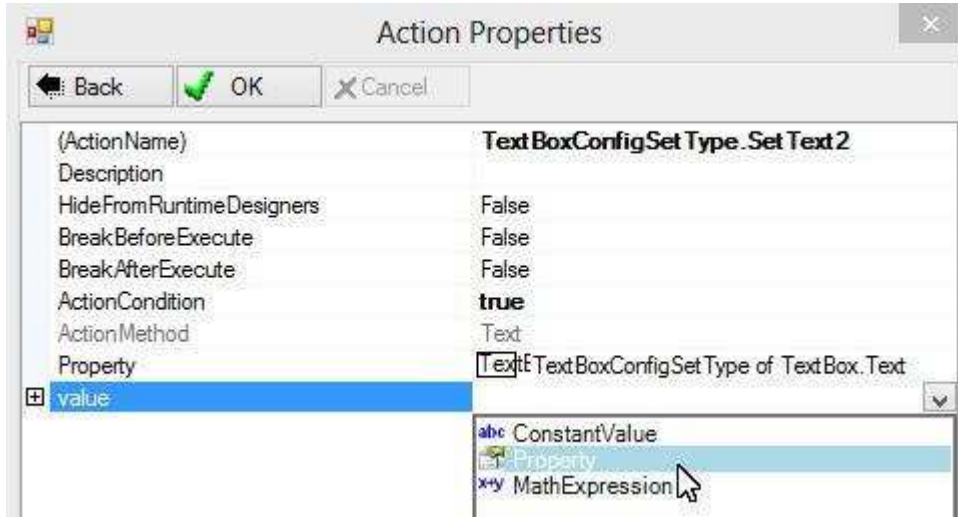

Choose the Text property of the text box for the type of the configuration set:
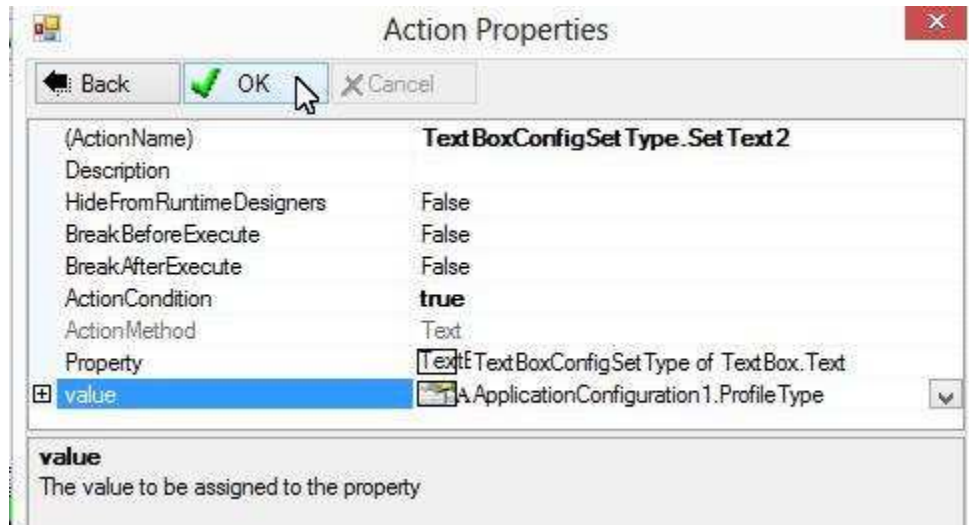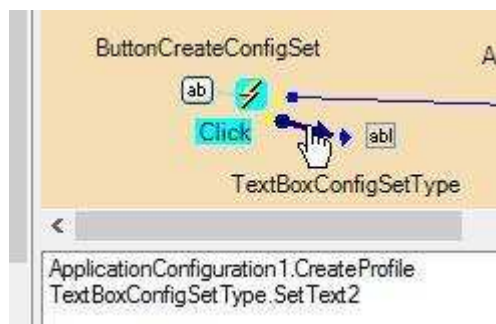
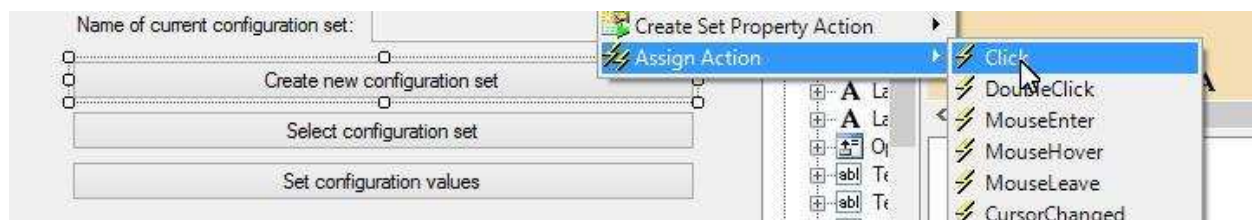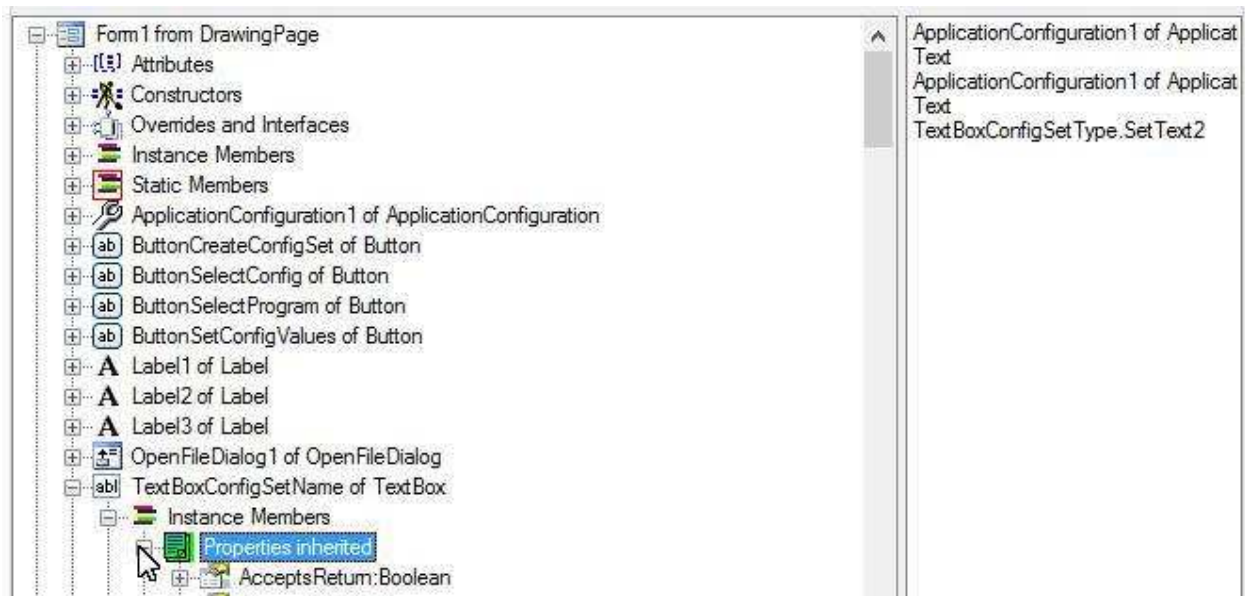Select ProfileType property of the Application Configuration component:

Click OK:

The action is created and assigned to the button:

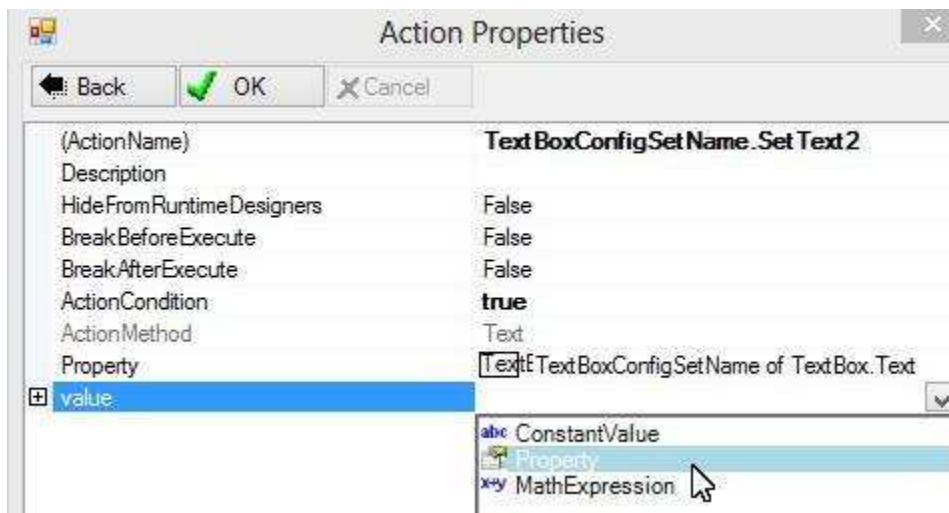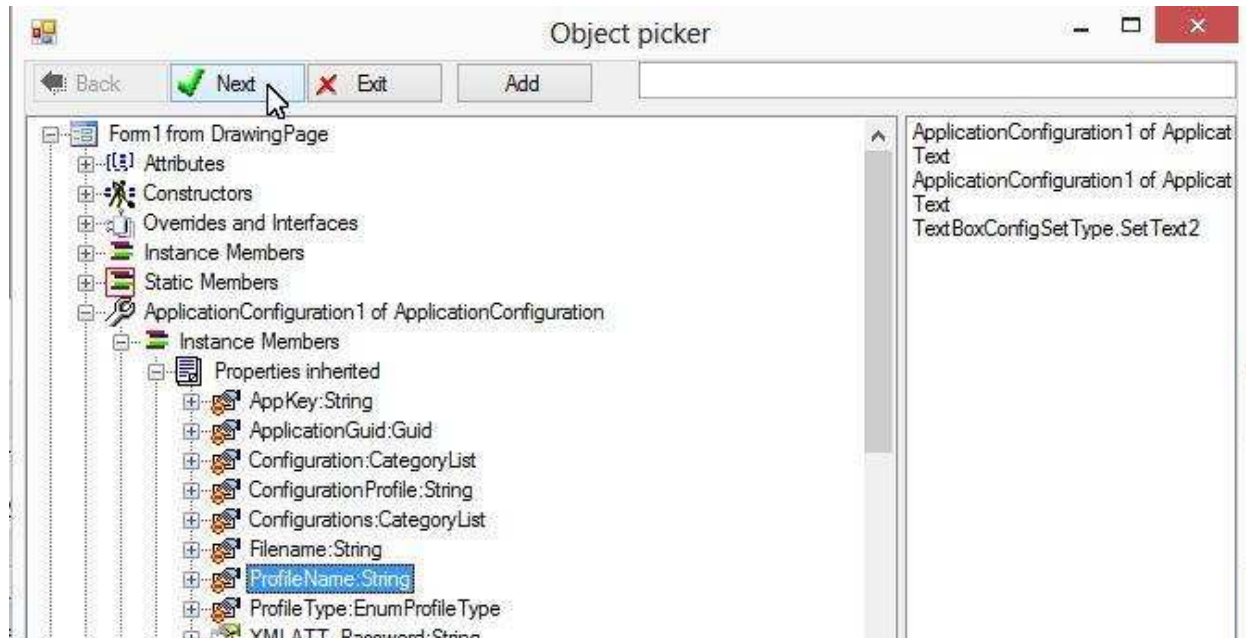

Show name of configuration set
Newly created configuration set will be the current configuration set. Let's show its name after the dialogue box is closed.

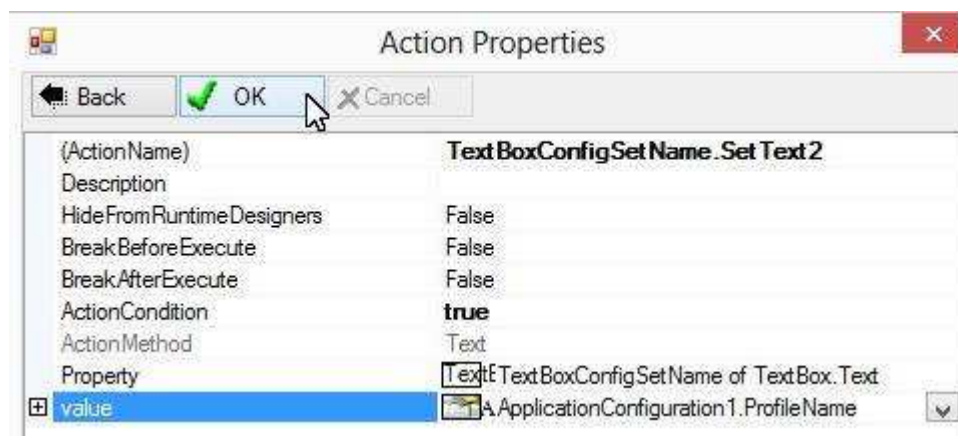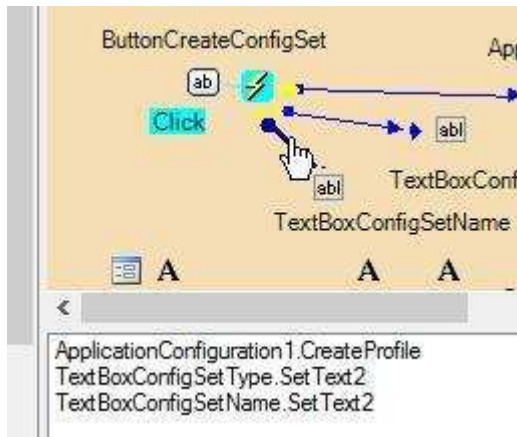Choose the Text property of the text box for showing the name of the configuration set:

Select ProfileName property of the Application Configuration component:

Click OK:

The action is created and assigned to the button:

Select Configuration Set

Configuration values are used and modified for the current selected configuration set. We need to let the user to select configuration set.

Launch dialogue

We use a button for selecting a set of configurations. The selected set of configurations will be used when the selected program is started. Configuration value modifications will be applied to the selected set of configurations.
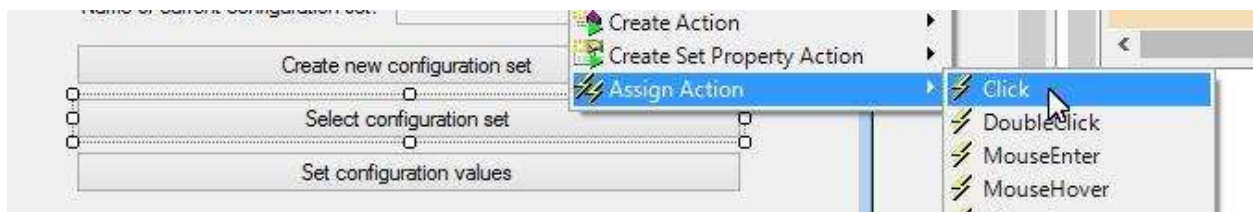
Choose LogOnProfileWithUI method:

The action will launch a dialogue box. Use the current form as the owner of the dialogue box:
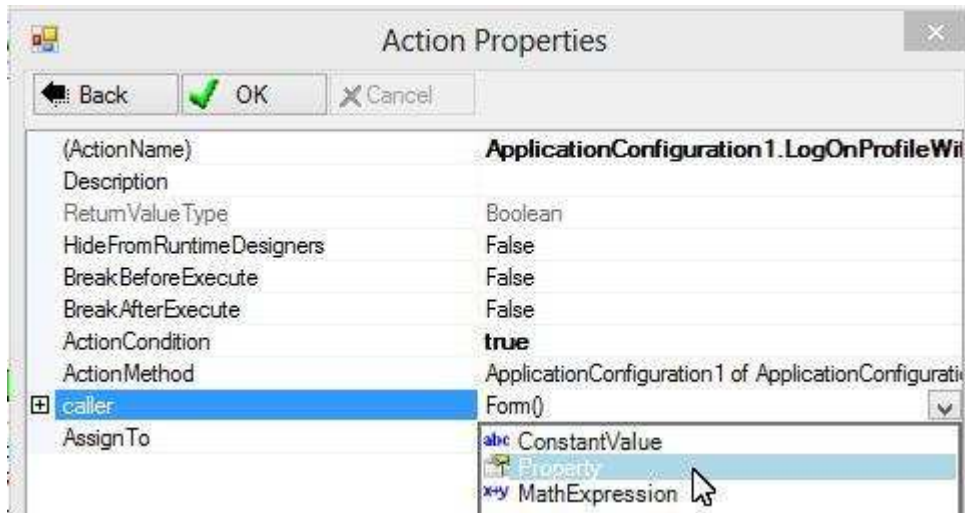
Click OK:
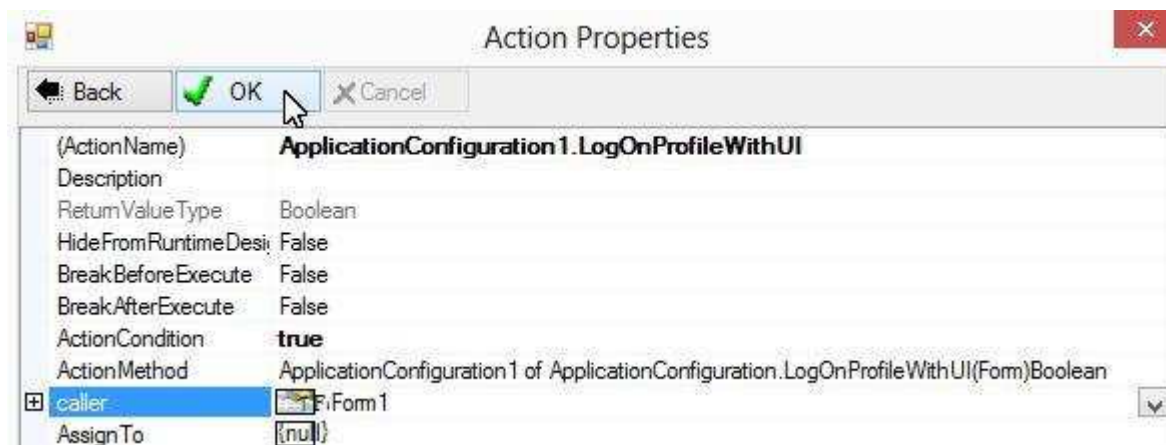
The action is created and assigned to the button:
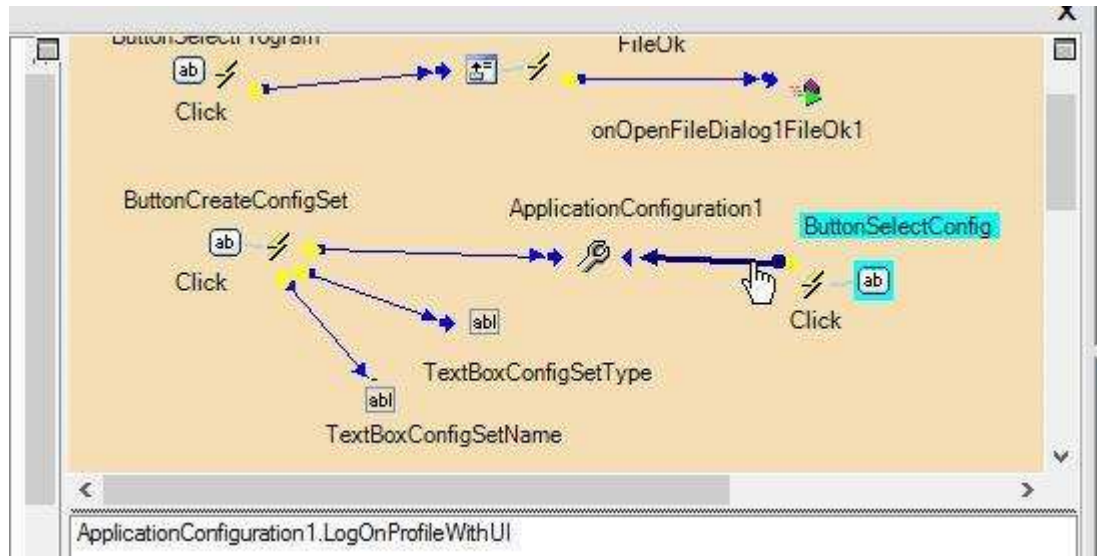
Show type of configuration set

After selecting configuration set, we need to show the type of the selected configuration set:
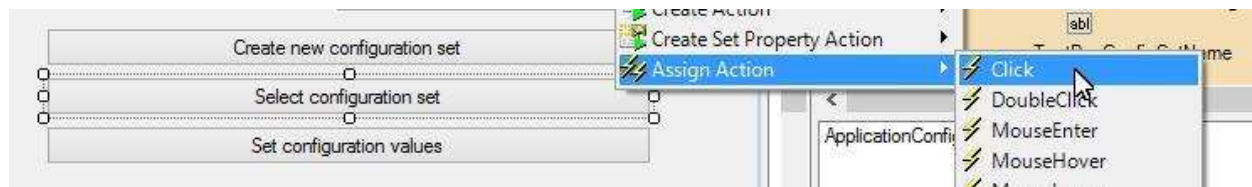
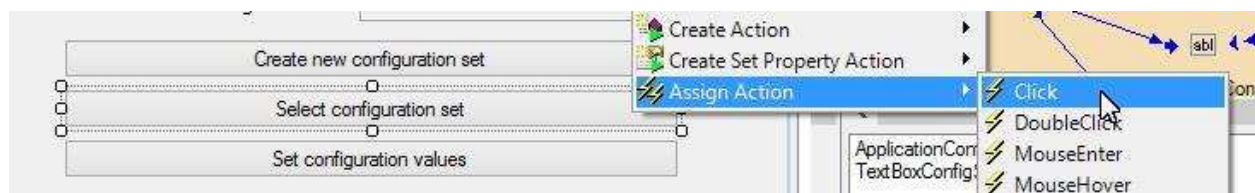Select the action, TextBoxConfigSetType.SetText2, we created previously:
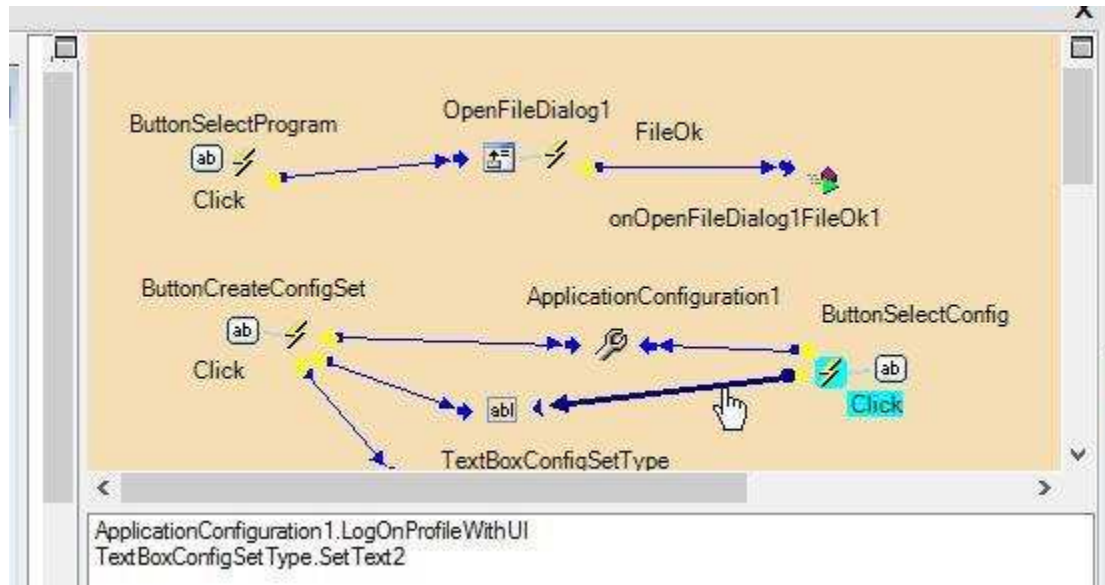
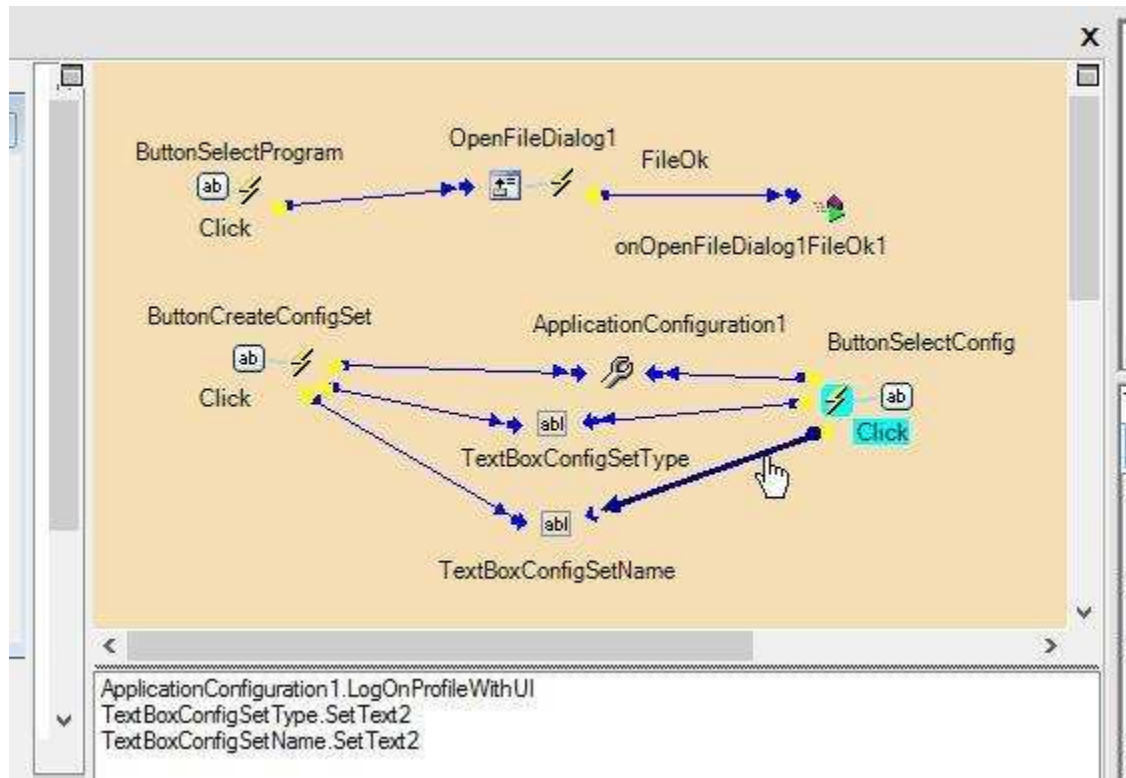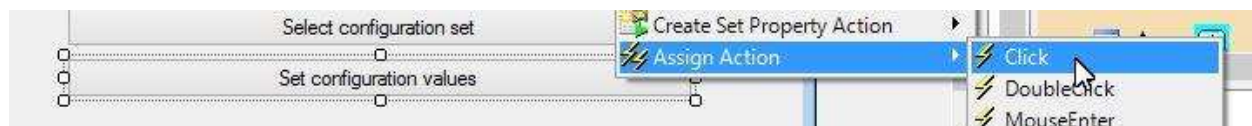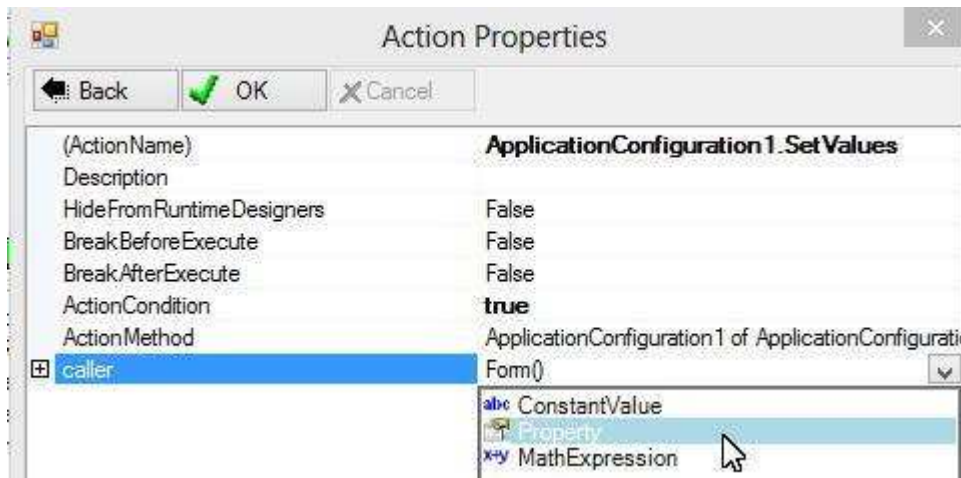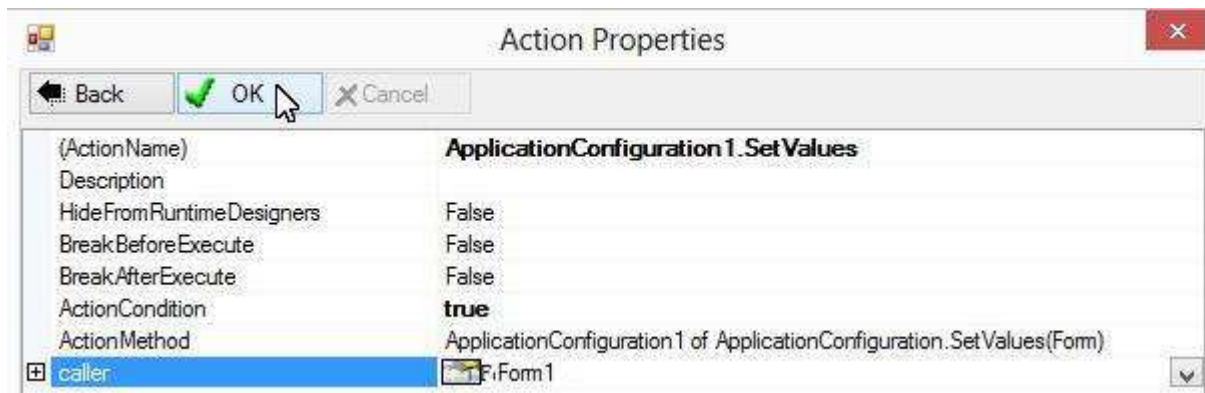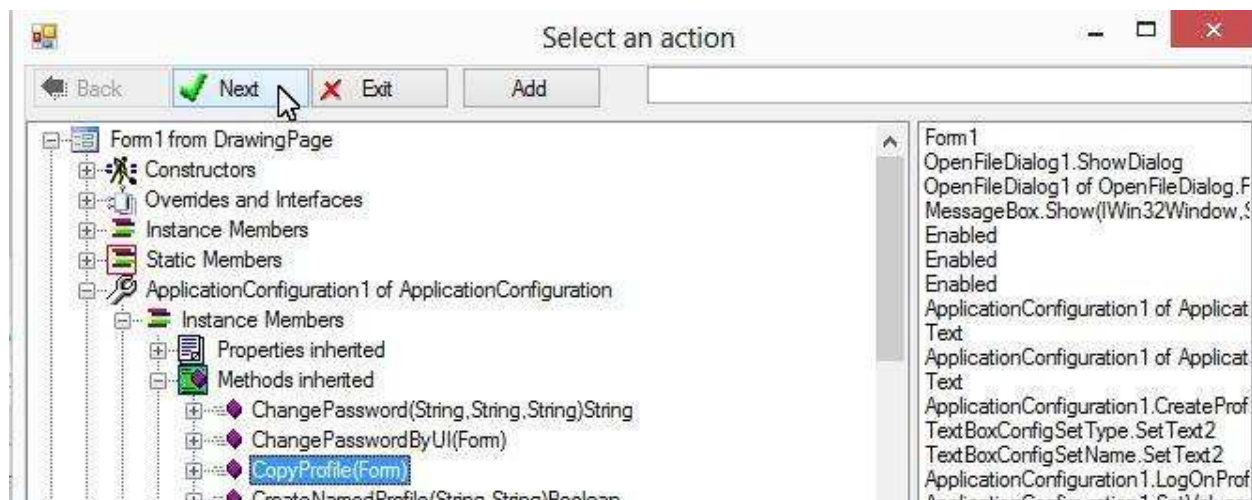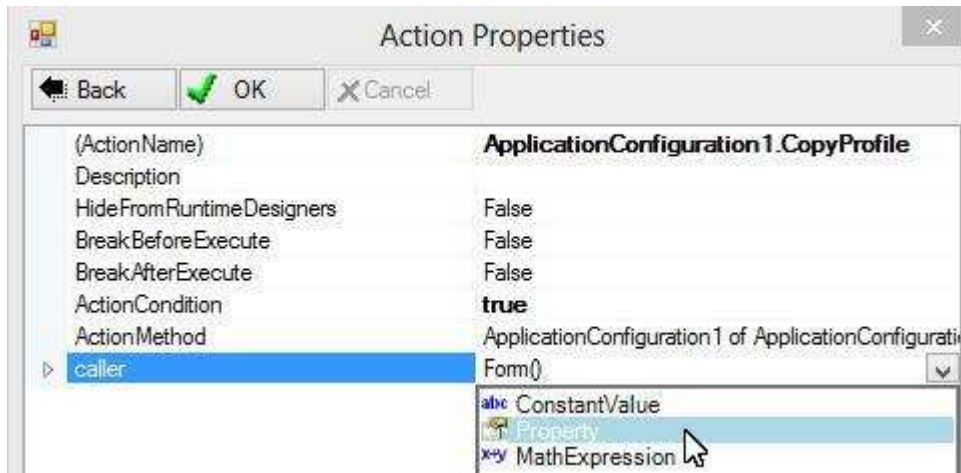The action is assigned to the button:

Show name of configuration set

After selecting configuration set, we need to show the type of the selected configuration set:

Select the actions, TextBoxConfigSetName.SetText2, we created previously:

The action is assigned to the button:

Set Configuration Values

We may use a button to launch a dialogue to modify configuration values. The modifications will be applied to the selected set of configurations.

Select method SetValues:

The action will launch a dialogue box. Use the current form as the owner of the dialogue box:

Click OK:

The action is created and assigned to the button:

ApplicationConfiguration1.SetValues
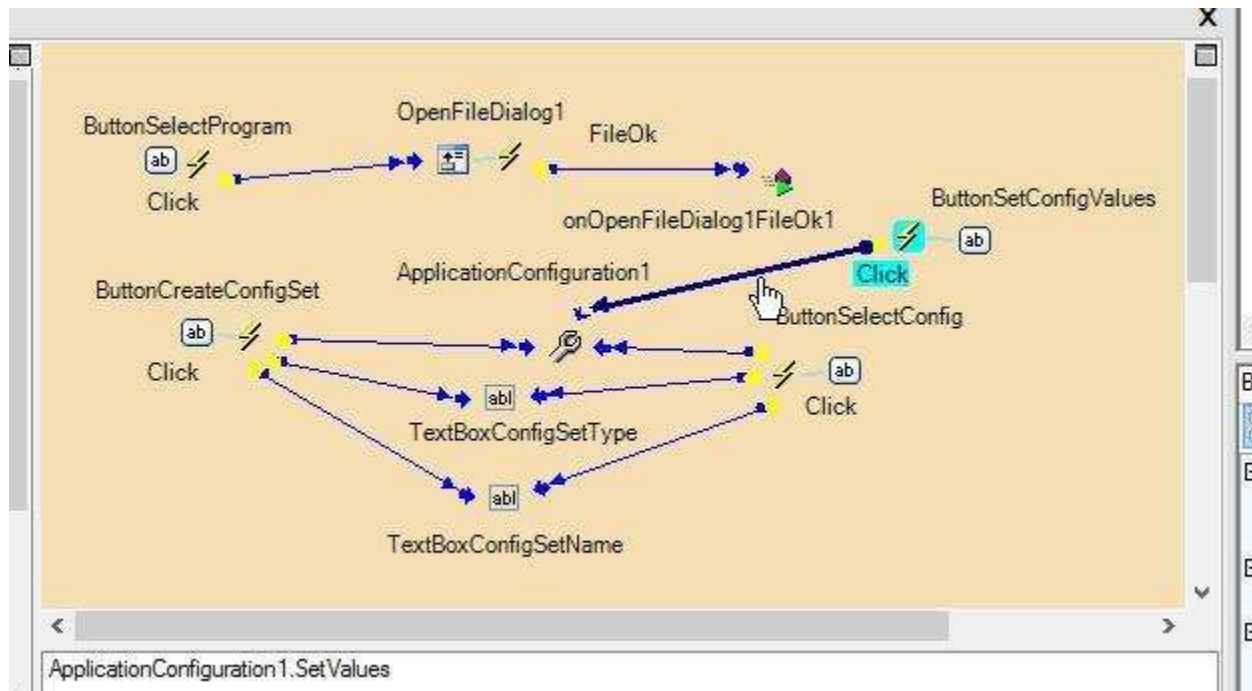
Copy configuration file

You may copy current configuration file to a specified folder. This feature is a convenient way for you to distribute a desired set of configuration values to your users as the factory settings.

Select CopyProfile method:

Use current form as the dialogue box owner:

Click OK to finish creating the action:
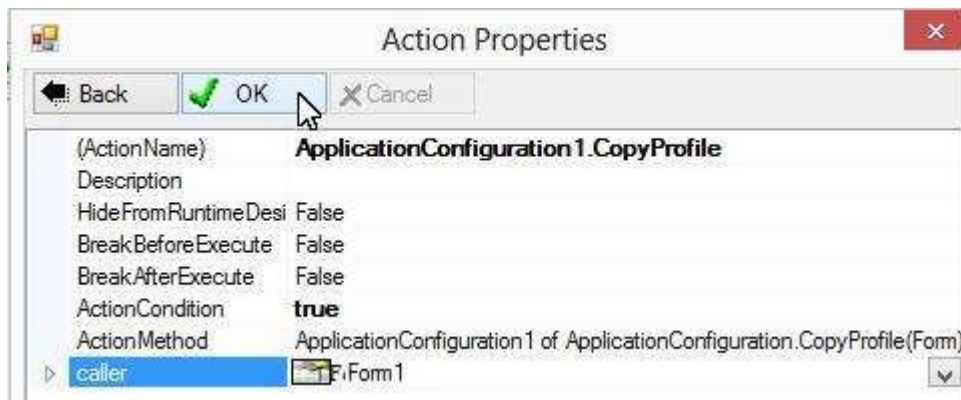
The action is created and assigned to the button:
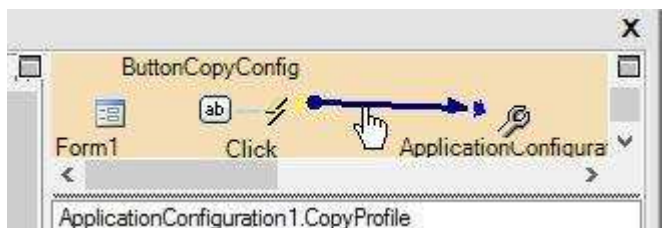
In the Test chapter below we will show you how this feature helps you to distribute desired configurations to your users as the factory settings.

Test

Run the program to start testing:
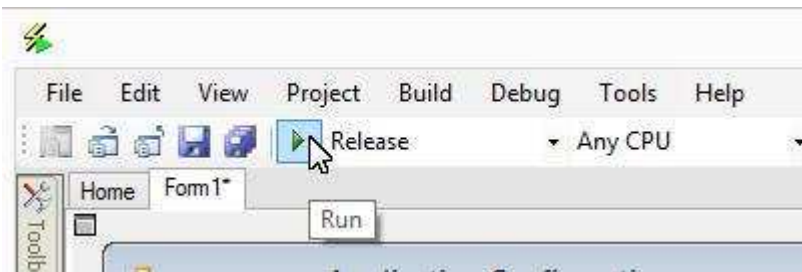
Select a program

Click "..." to select a program:
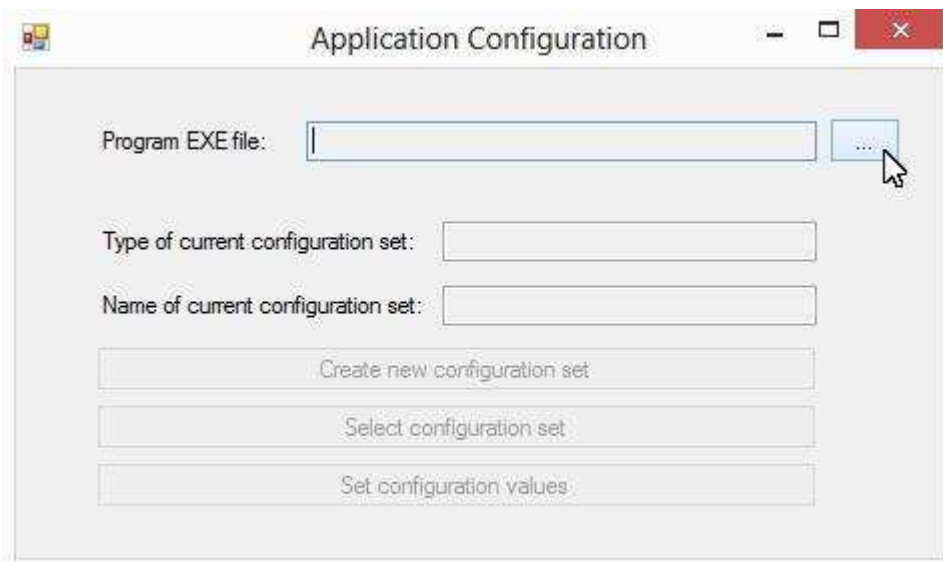
Select a program which has Application Configurations enabled:
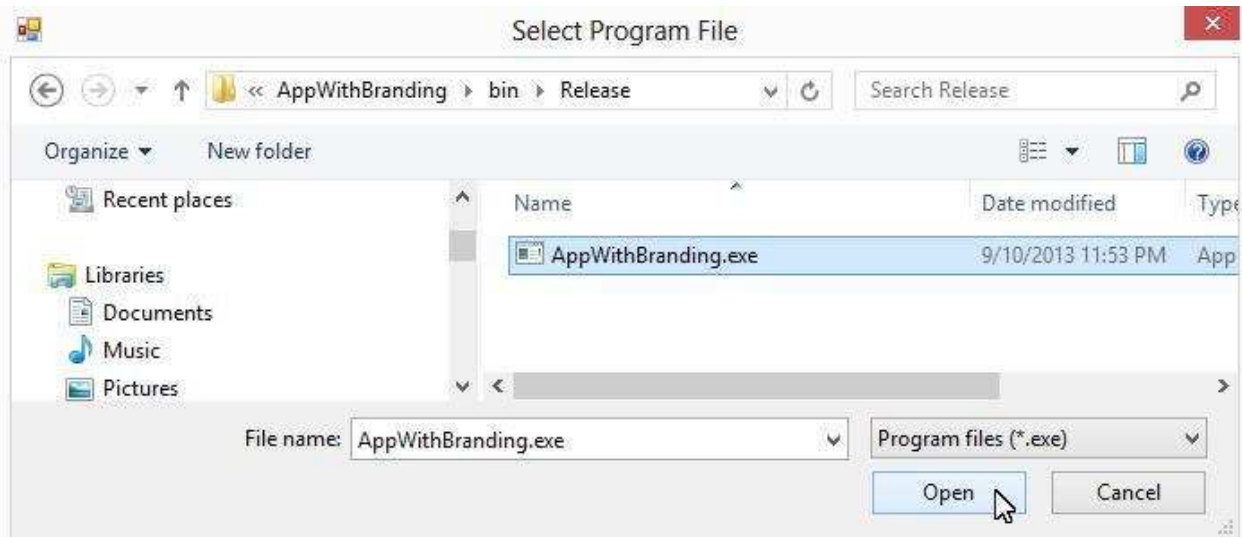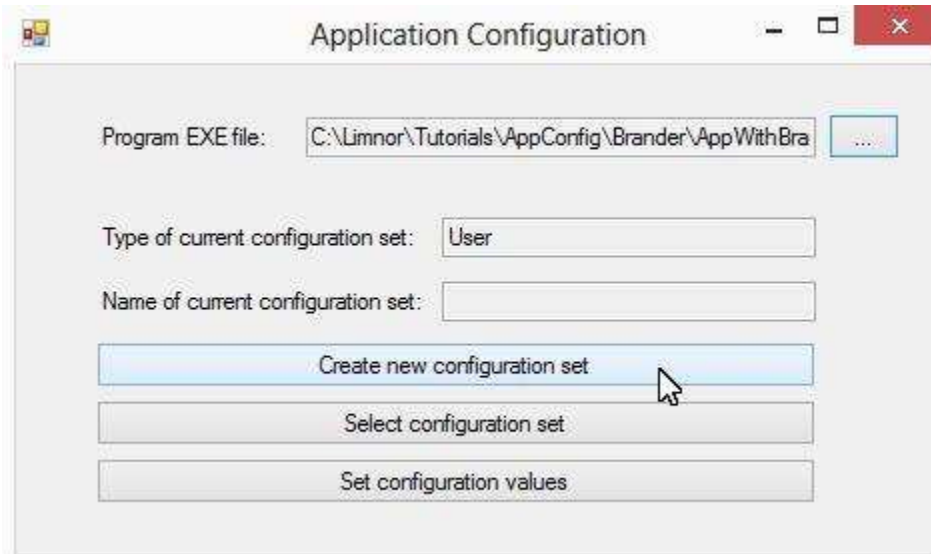
The program is applied to the configuration component:
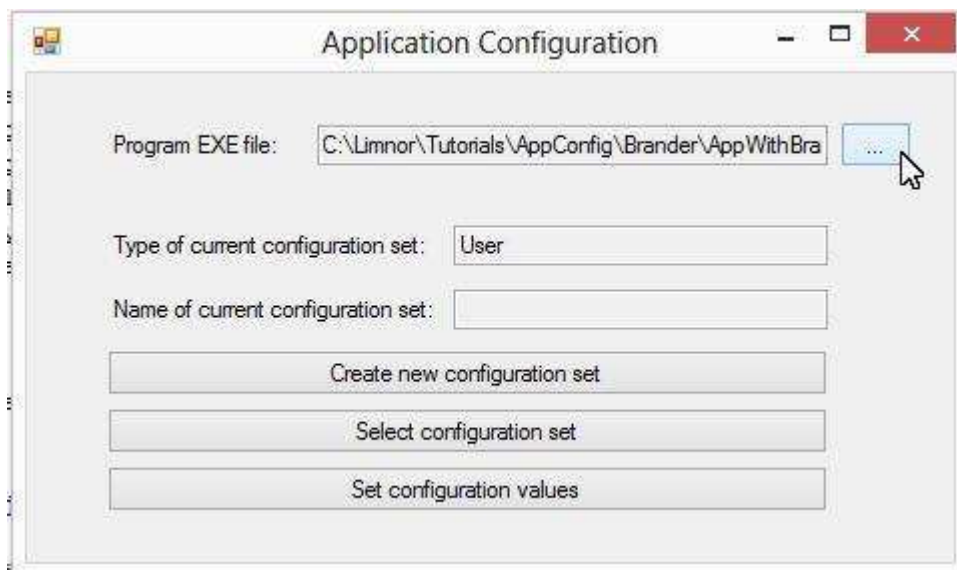
We can see that its current configuration set is user-specific because the type of the current configuration set is User.
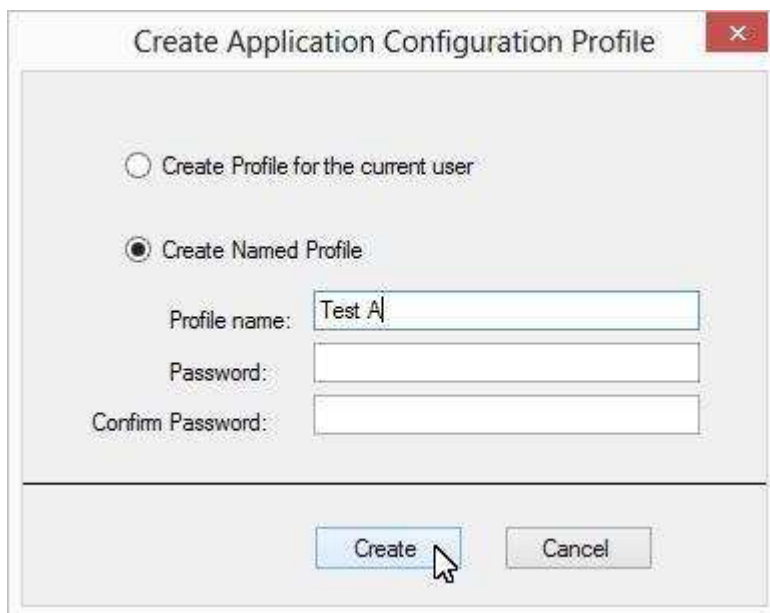
The 3 buttons for performing configurations are enabled. Now we can do configurations for the selected program.

Create configuration set

Note that the new configuration set becomes the current configuration set:

We may create more if we want:

Select configuration set
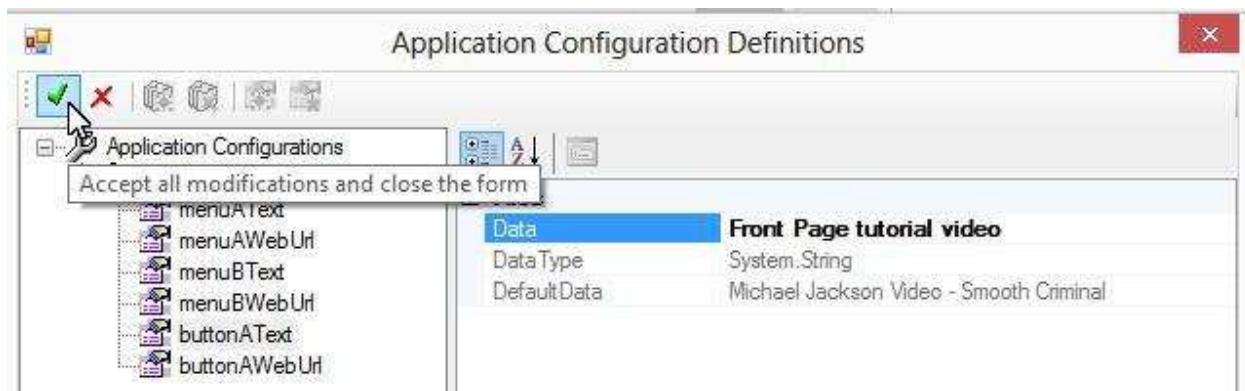
Set configuration values
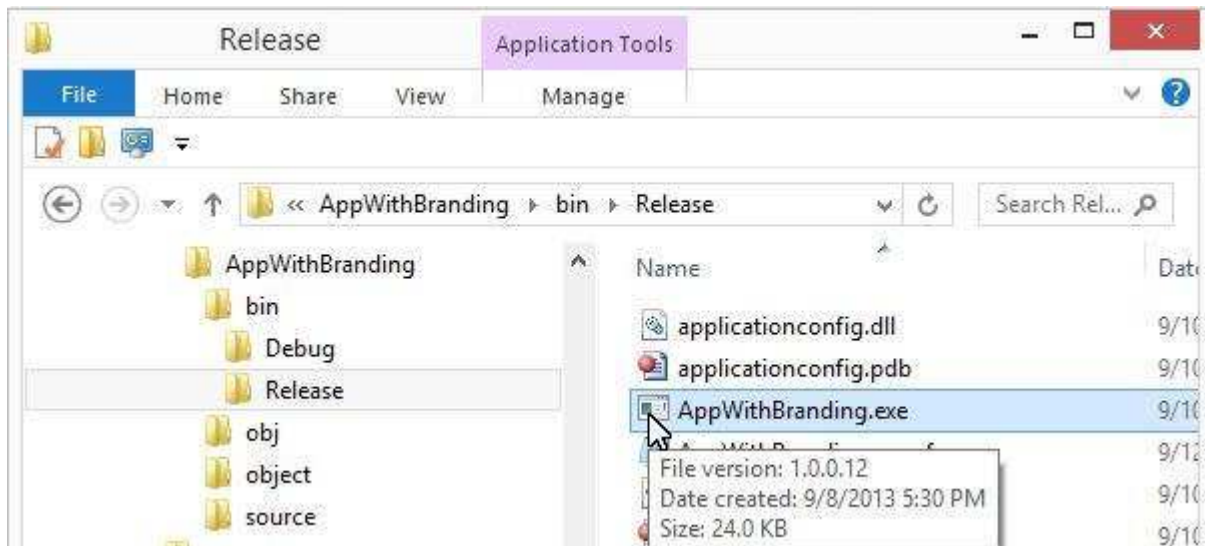

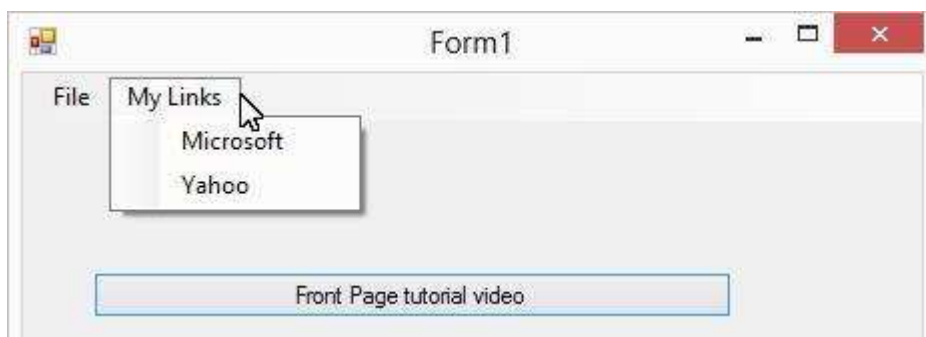Make modifications to configuration values:
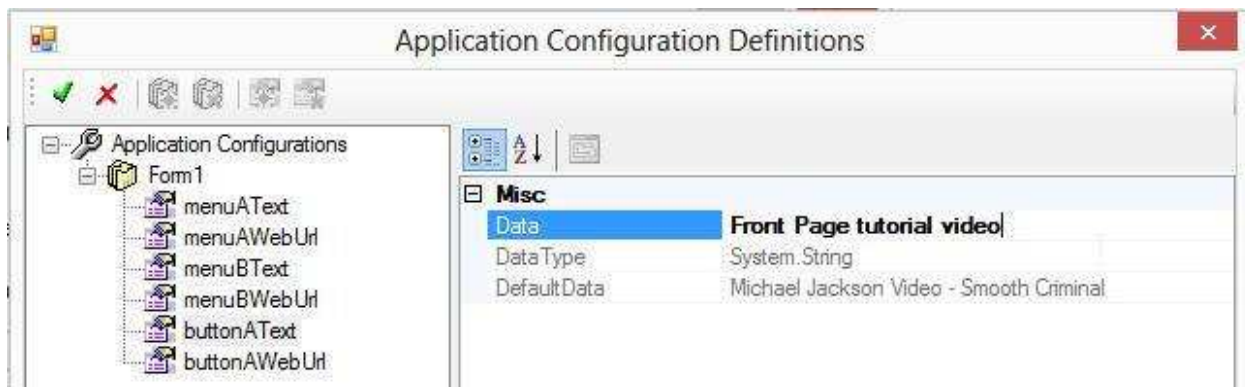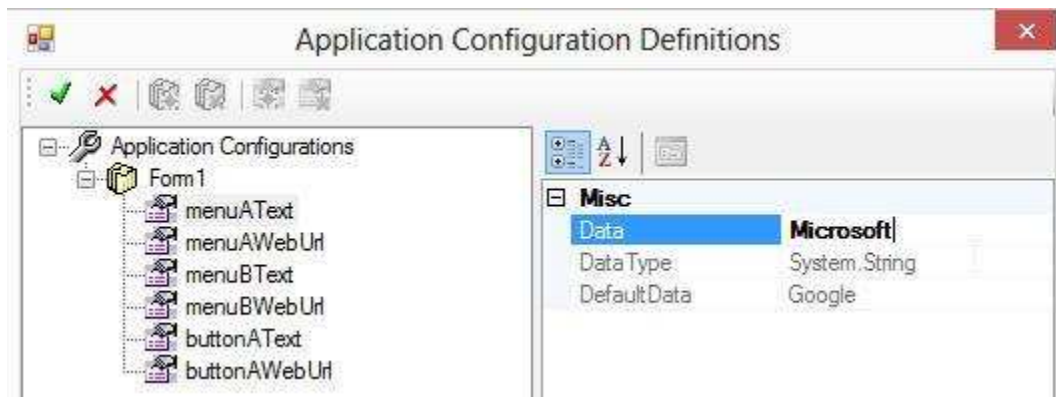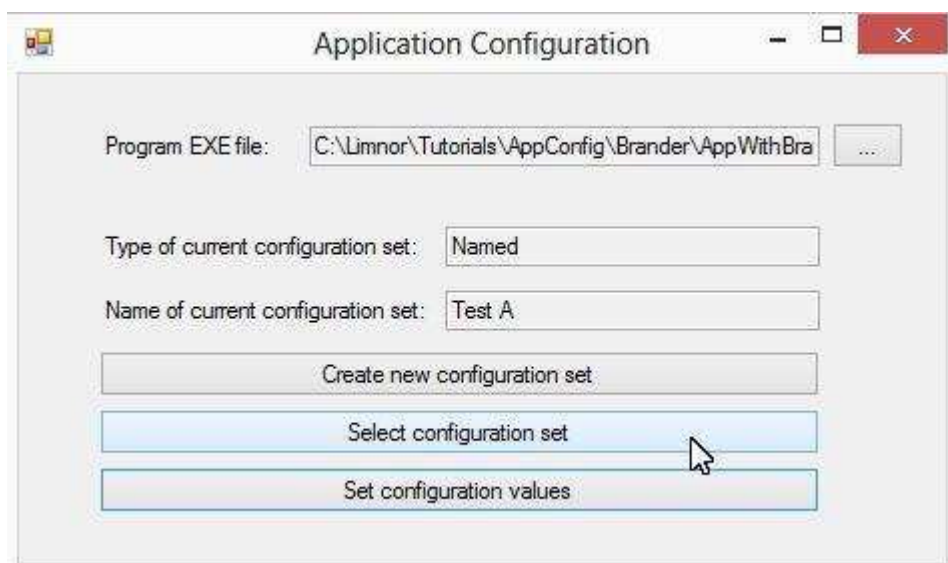
Finish the editing:
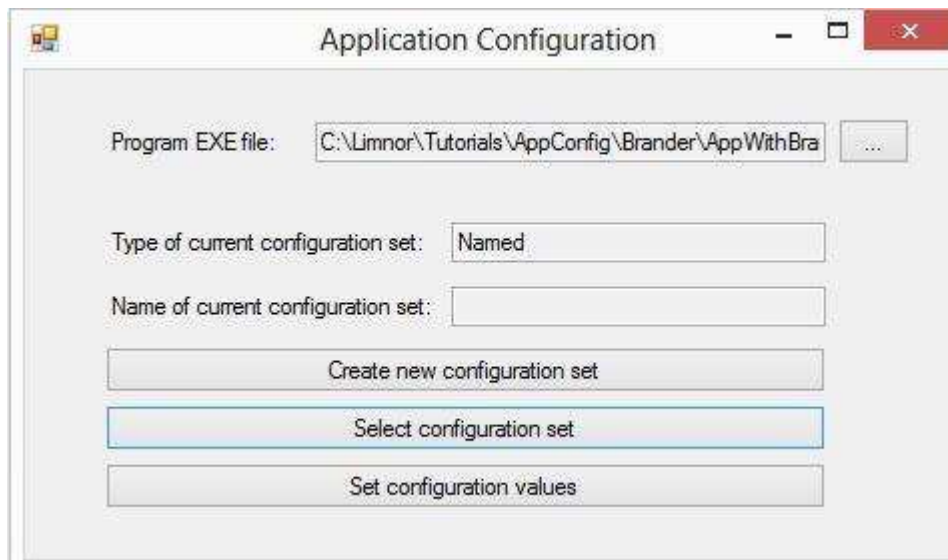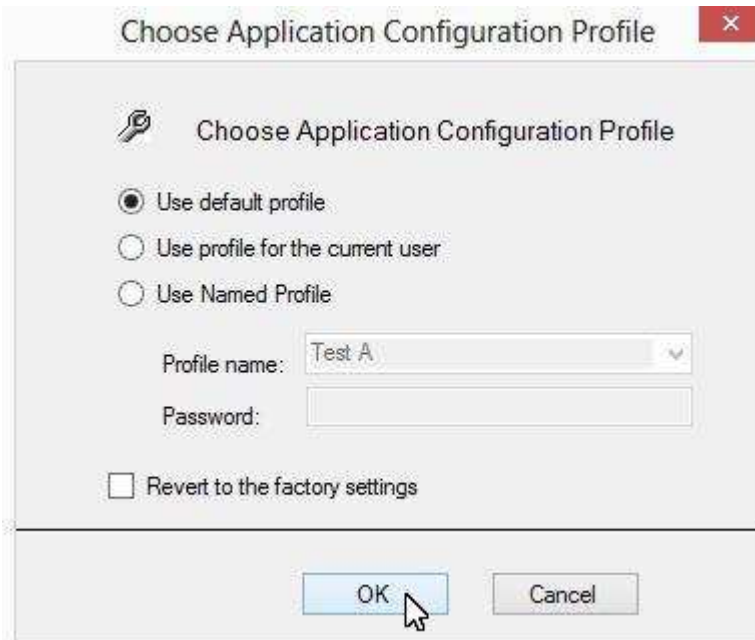

Run program

Now we run the program:

We can see that this program is changed by our configuration utility.
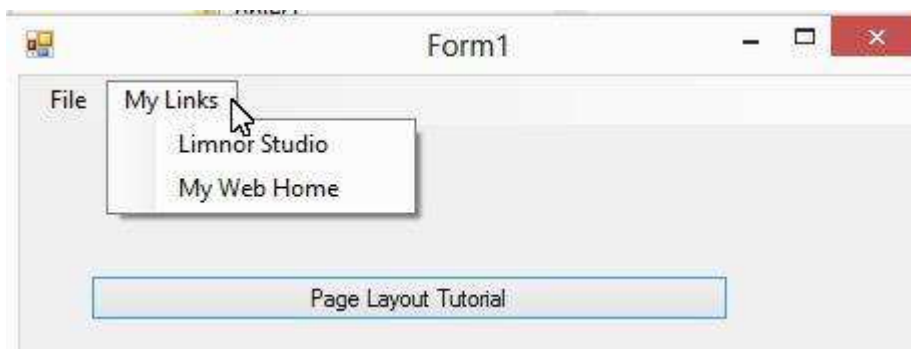

Let's close the program and select a different set of configuration using our utility:
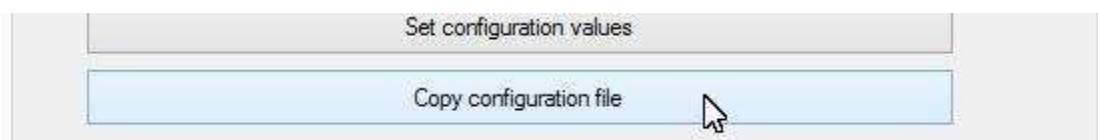
The default profile is the named profile with empty name.

Now run the program. We can see the default configurations are used:
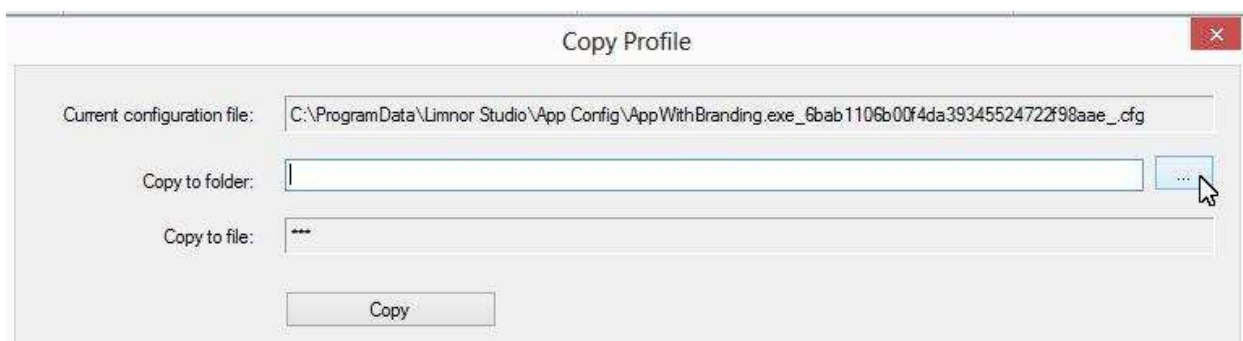
Distribute desired configurations
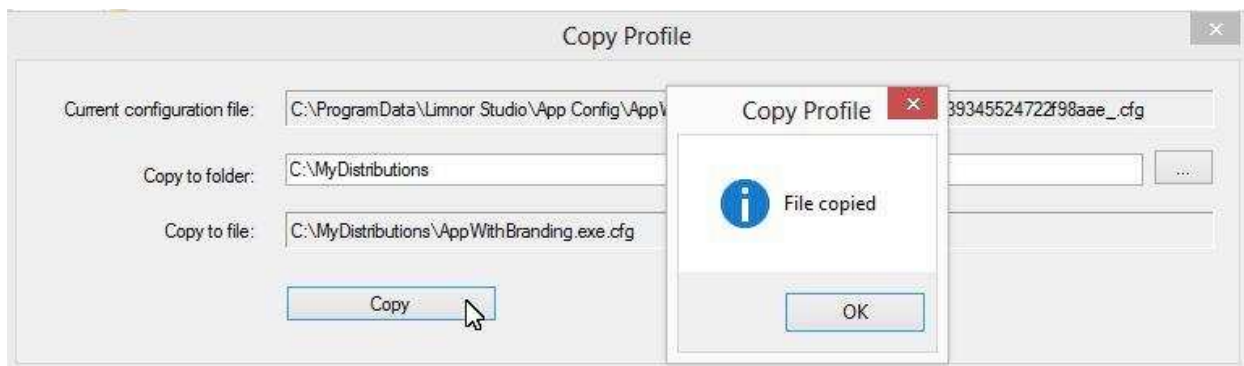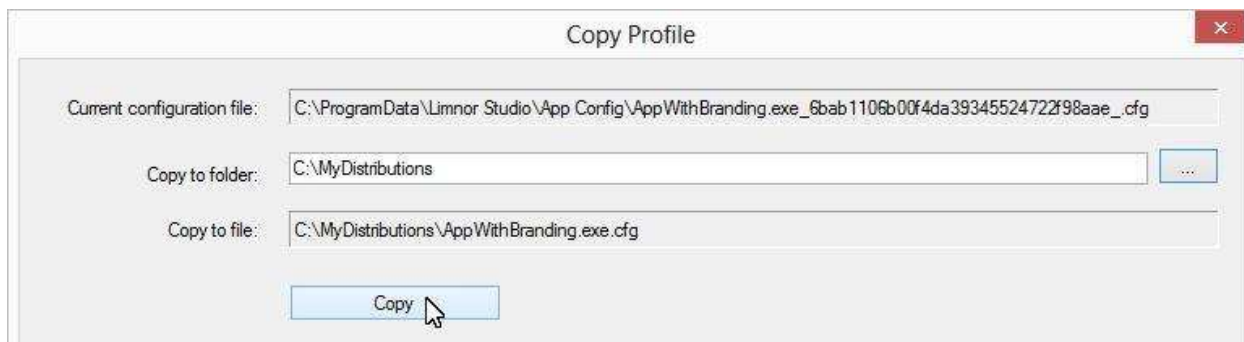
Click button "Copy configuration file":

A dialogue box appears. Click "…" to select Copy-to folder:


Select a folder:

Click Copy button to copy the file:

The file is copied to the selected folder:

Note that the copied file is named as {EXE file name}.cfg. This naming is for the factory settings. For example, we developed the program named AppWithBranding (see http://www.limnor.com/support/UseAppConfigForBranding.pdf), the files to be distributed to our users are in folder Release:
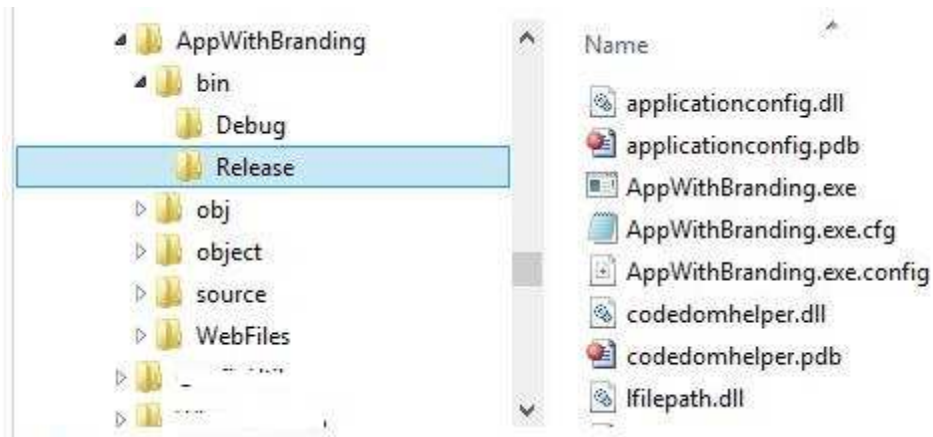
The file AppWithBranding.exe.cfg in the folder is the factory settings. Replace this file to give the users the factory settings we want. You may backup the file being replaced and restore it when needed.

Feedback

Please send your feedback and suggestions to support@limnor.com