

Action Conditions

1	Introduction.....	1
2	Action Conditions	1
3	Action List Conditions	3
4	Use Properties in Action Conditions	4
5	Use Condition Builder	6
6	Sample Application – Add Conditions to Action	10
7	Sample Application – Add Conditions to Action List	16
8	Questions and Feedback	23

1 Introduction

Limnor has enhanced programming of conditional action execution, making it more intuitive. The purpose is to make codeless programming easier, and make building complex programs easier.

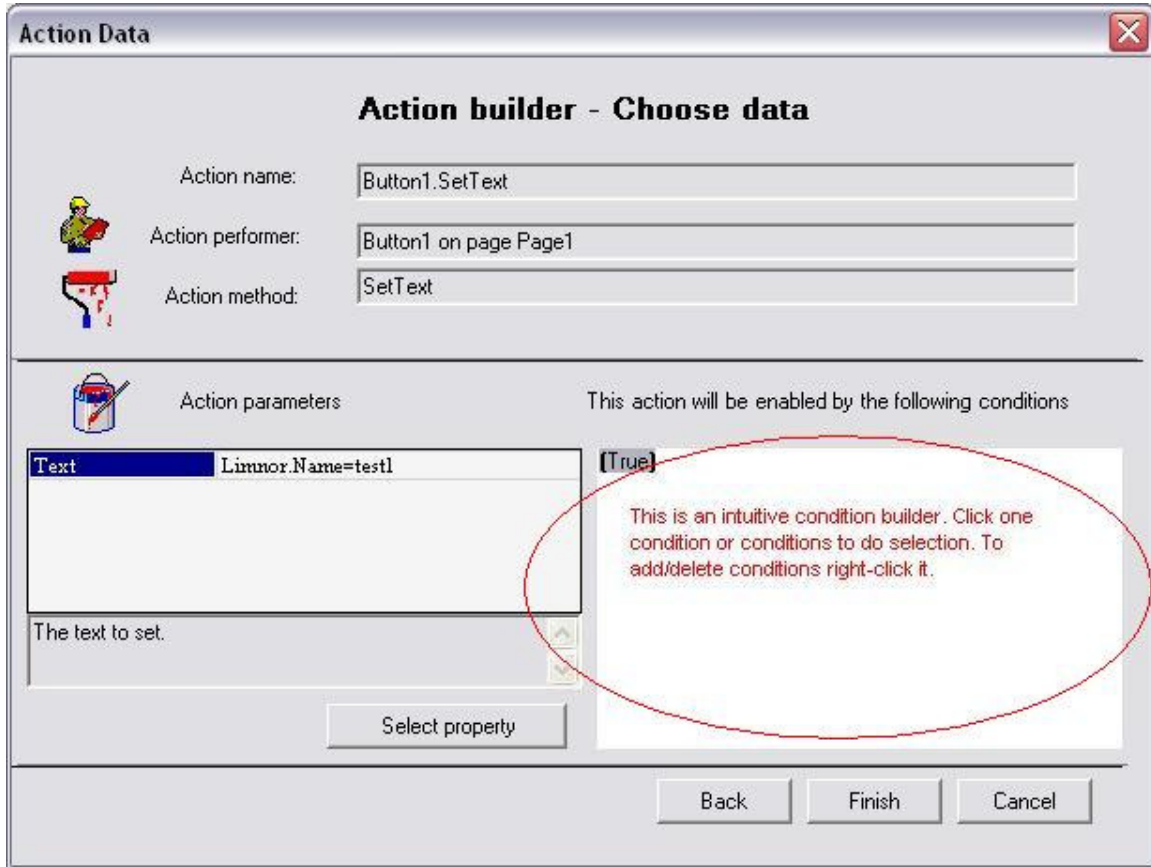
This sample application demonstrates such new capabilities.

2 Action Conditions

You may add conditions to every action. When it is time to execute the action, the conditions will be checked. If the conditions are all met then the action will be executed. If the conditions are not met then the action will be ignored. When you are creating an action, it will allow you to set the action conditions.

When an action needs parameters, the condition builder is beside the parameter list:

Action Conditions



When the action does not have parameters, the condition builder will come up by its own:



By default the condition shows “(True)” meaning unconditional execution.
We will show you how to use the condition builder later.


3 Action List Conditions

An action list has two parts formed as “IF (conditions met) THEN (execute action list 1) ELSE (execute action list 2)”. The action list builder lets you build these 3 elements: “conditions”, “action list 1”, and “action list 2”:



Click  to build action list 2:



Click  to bring action list 1 again.

4 Use Properties in Action Conditions

The action conditions are built with properties of Performers. What is a property of a Performer?

Every Performer has several properties. The value of a property represents a state of the Performer. When the application runs, all Performers have their initial values for all properties. While the application is running, many values of properties will be changed by actions you created or by the Performers owning the properties.

For example, a Checkbox Performer has a property named “Checked”. It indicates whether the checkbox is checked. At runtime when the user checks the checkbox the value of “Checked” property becomes True; when the user un-checks the checkbox the value of “Checked” property becomes False. Therefore your application may use the “Checked” property in the action conditions for such programming logics “IF (the user checks the Checkbox ...) THEN ...”.

The “Checked” property may only have one of two possible values: True or False. This kind of value

Action Conditions

is called Boolean type.

Other kinds of properties may have other kinds of values, not Boolean type. For example, “RowCount” property of a DataQuery Performer is an integer which can be 0 or greater than 0. “Text” property of a TextBox Performer is a string of characters.

All properties used in action conditions will be translated into Boolean values. That is, every value will be translated into one of two values: True or False. The following table shows how the translation is done.

Property value type	Translation rules	Example
Number	0 is translated to False; all other values is translated as True	RowCount property of a DataQuery Performer is the number of records found. 0 means no records found. Logic “IF (data found) THEN ...” can be expressed as “IF (RowCount) THEN ...”, because when data found Rowcount will be greater than 0 and will be translated as True.
String	If the string can be converted to a number then the translation rules for Number apply. If the string cannot be converted to a number then the following rules apply: An empty string is translated to False. The following strings are translated to False: “no”, “false”, “off”. It is case-insensitive. So “No” and “NO” are also translated to False. All other strings are translated into True.	

For example, when you use DataQuery Performer to search a database, if the search succeeds then its “RowCount” property is the number of records fetched from the database. If the search does not find any thing then its “RowCount” property is 0. According to the above translation rules we can use “RowCount” in action conditions to make such programming logics: “IF (search of the database found something...) THEN ...”.

5 Use Condition Builder

In this section we show you how to use the condition builder to build action conditions using performer properties.

Section 2 and section 3 show you where the action condition builder appears. When the action builder is empty you see (True):



That means unconditional execution.

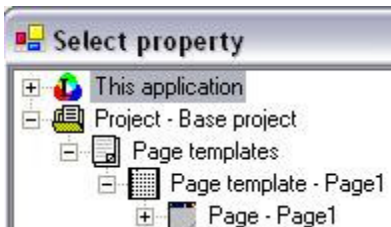
To start adding action conditions, click on the word “True”, it will be highlighted:



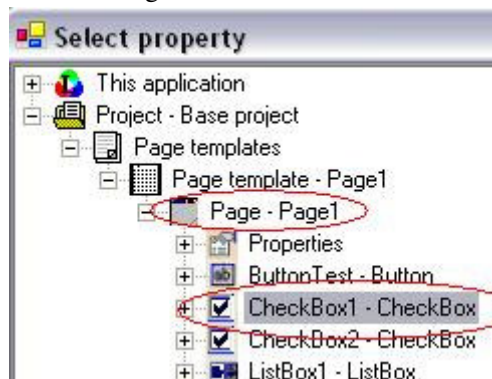
Right-click it, a context menu appears. Choose Add:



It will let you select a property as the condition:

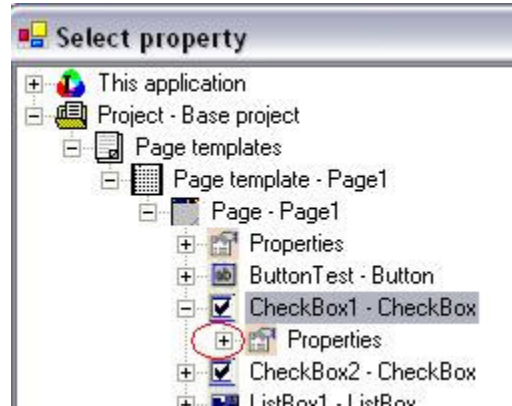


To select a property, first find the Performer that owns the property. For example, we want to use the “Checked” property of CheckBox1 on Page1. First, we find Performer CheckBox1 on Page1:

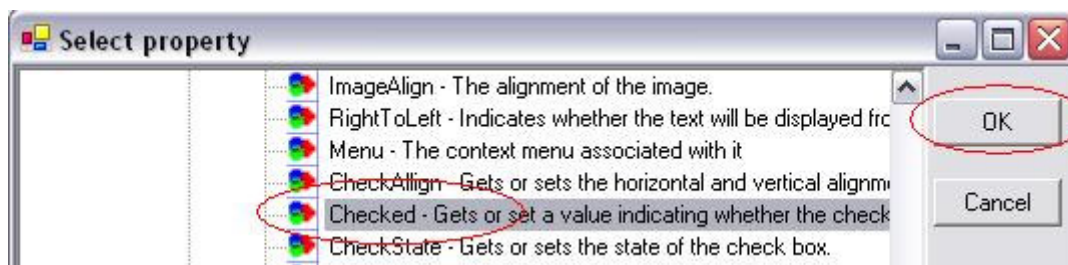


Click  besides CheckBox1 to expand the node. Click  besides Properties to expand the node:

Action Conditions



Under “Properties”, find and select “Checked” property and click “OK” button:



This property appears in the action condition builder:

[CheckBox1.Checked]

For this condition if the CheckBox1 is checked then the condition is True (met); if the CheckBox1 is not checked then the condition is False (not met).

If an action is controlled by this condition then the action can be executed only when CheckBox1 is checked.

What if we want the action can be executed only when the CheckBox1 is NOT checked? Click “CheckBox1.Checked” to select it (highlighted):

[CheckBox1.Checked]

Right-click it while it is highlighted, choose “NOT” from the context menu:

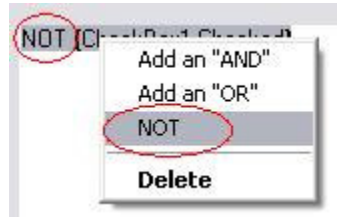


Now the condition is True only when CheckBox1 is NOT checked:

NOT [CheckBox1.Checked]

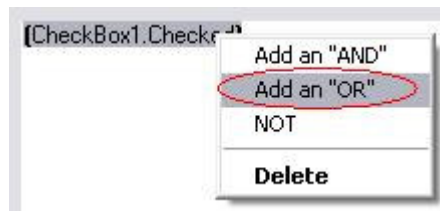
To remove the “NOT” condition, just right-click it and choose NOT again:

Action Conditions

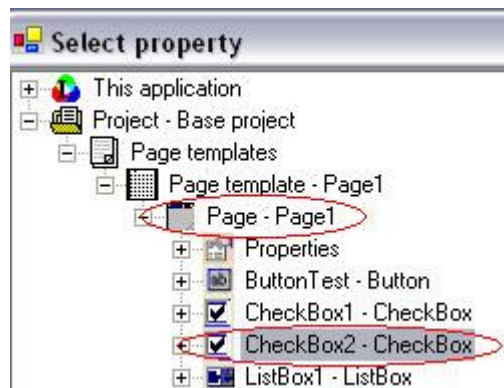


You can see that in the context menu there is an item “Delete”. Choosing it will remove the property from the condition.

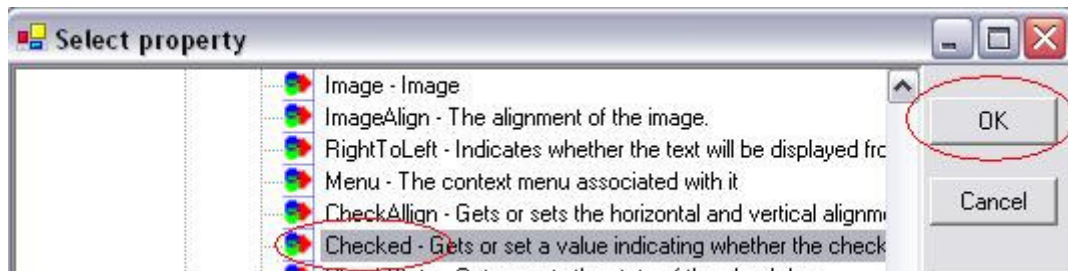
Now let’s make the conditions a little complex. Suppose we want the action can be executed when CheckBox1 **OR** CheckBox2 are checked. To do it, right-click “CheckBox1.Checked” while it is highlighted, choose menu Add an “OR”:



It lets you select the property wanted. We choose performer CheckBox2:



Select its Checked property:



Now the conditions become:

```
{CheckBox1.Checked OR CheckBox2.Checked}
```

We thus built the conditions we wanted.

Action Conditions

How to change the conditions? You need to highlight the parts of the conditions you want to change. For example, click CheckBox1.Checked to highlight it:



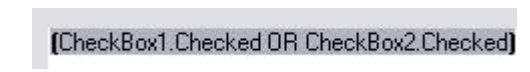
[CheckBox1.Checked OR CheckBox2.Checked]

Click CheckBox2.Checked to highlight it:



[CheckBox1.Checked OR CheckBox2.Checked]

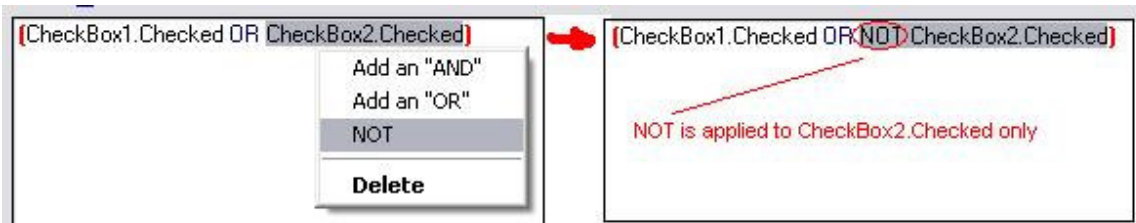
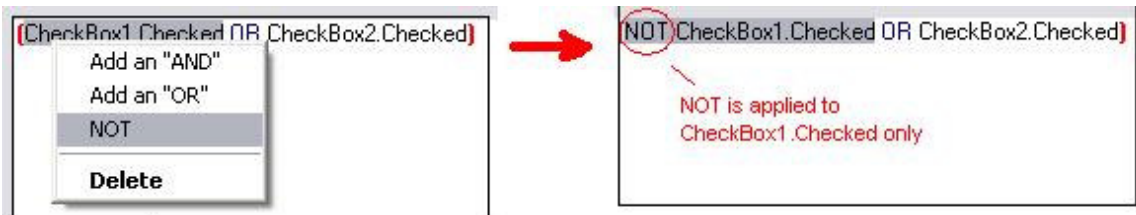
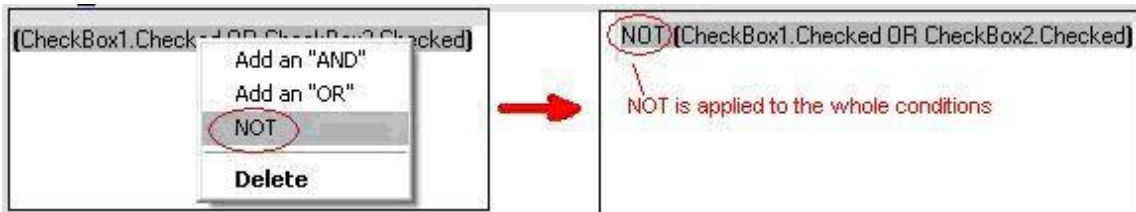
Click the word **OR** to highlight both conditions:



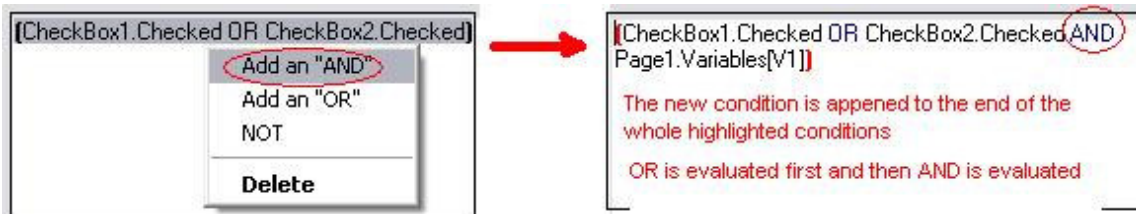
[CheckBox1.Checked OR CheckBox2.Checked]

With the wanted part(s) highlighted, right-click it and choose the context menu to do the editing. For example, choosing menu “Delete” will delete the highlighted part(s).

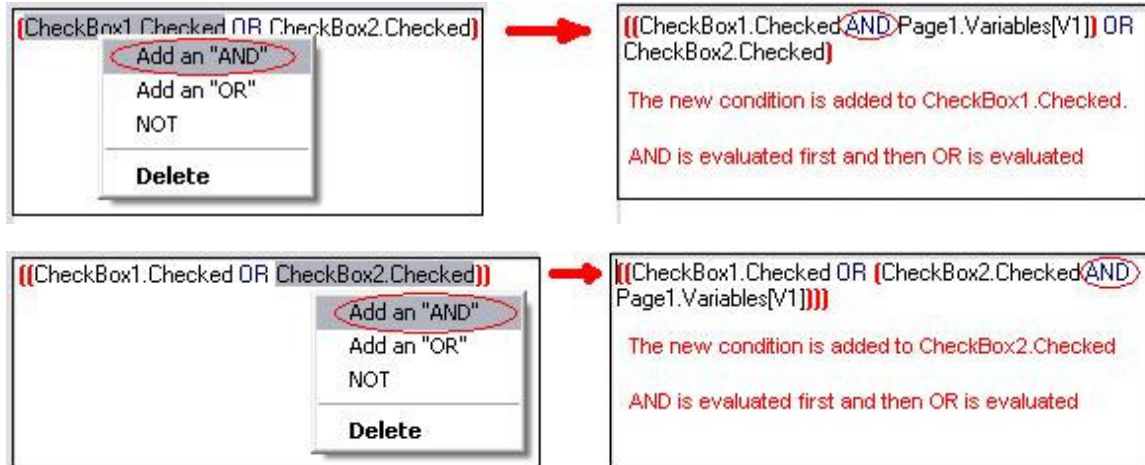
Choosing menu “NOT” will apply NOT to the highlighted part(s). Below are some examples:



The logic of adding a new condition will also be based on how you highlight the part(s). We created a variable named V1 under Page1. We now add this variable as a new condition to the existing conditions. Based on the part(s) we highlighted, V1 will be added in different logics:



Action Conditions



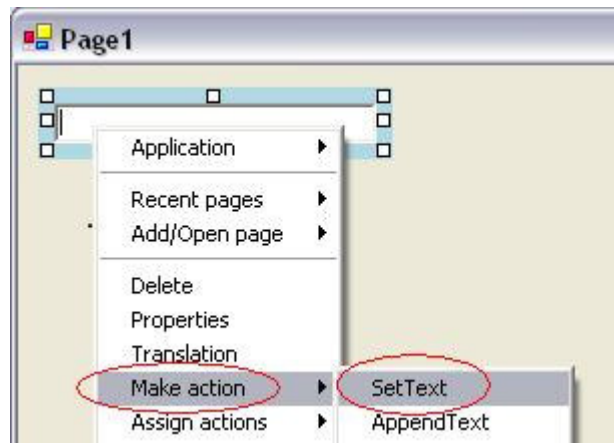
This process can go on and on, for example:

```
[[CheckBox1.Checked OR (CheckBox2.Checked AND  
(Page1.Variables[V1] OR Page1.Variables[V2])]]]
```

You can see that in this way you can visually build logics of any complexity.

6 Sample Application – Add Conditions to Action

We use a Text Box Performer for creating an action to show a sentence on it. Right-click on the Text Box performer, choose "Make action", choose "SetText":

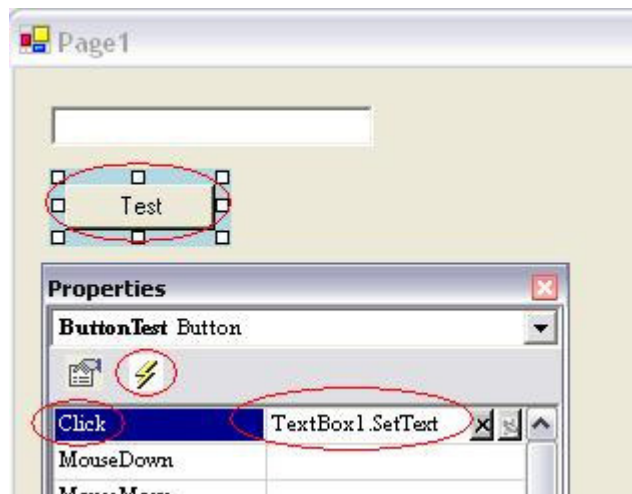


Accept the default action name, "TextBox1.SetText". Type in a sentence:

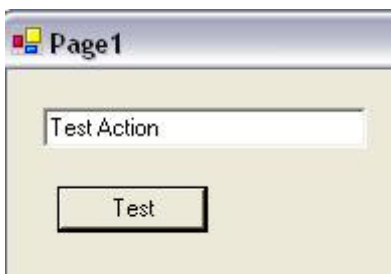
Action Conditions



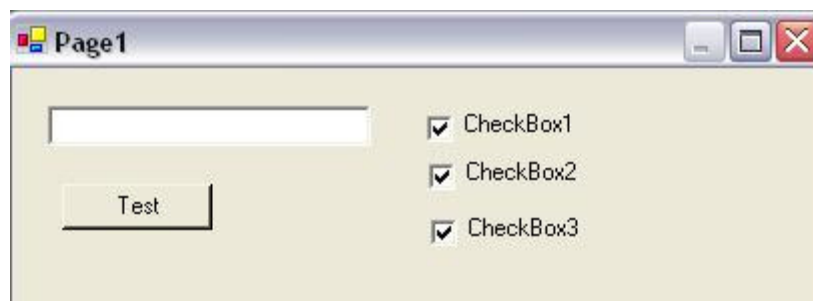
We assign this action to a button's Click event:



Now we can run this simple application and click Test button and see the sentence "Test Action" appears on the TextBox:

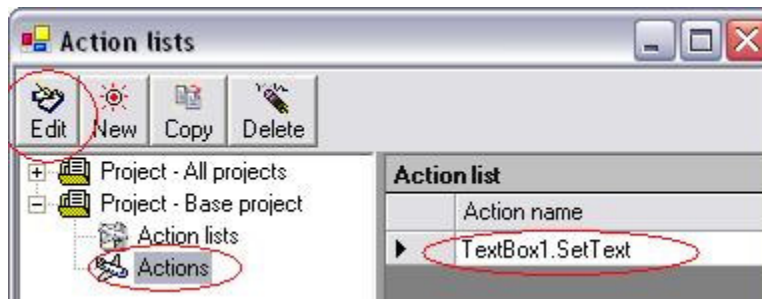


Now let's go back to design mode and add conditions to this action. We add 3 CheckBox performers the page. We want this action can be executed only when all the 3 CheckBox performers are checked.

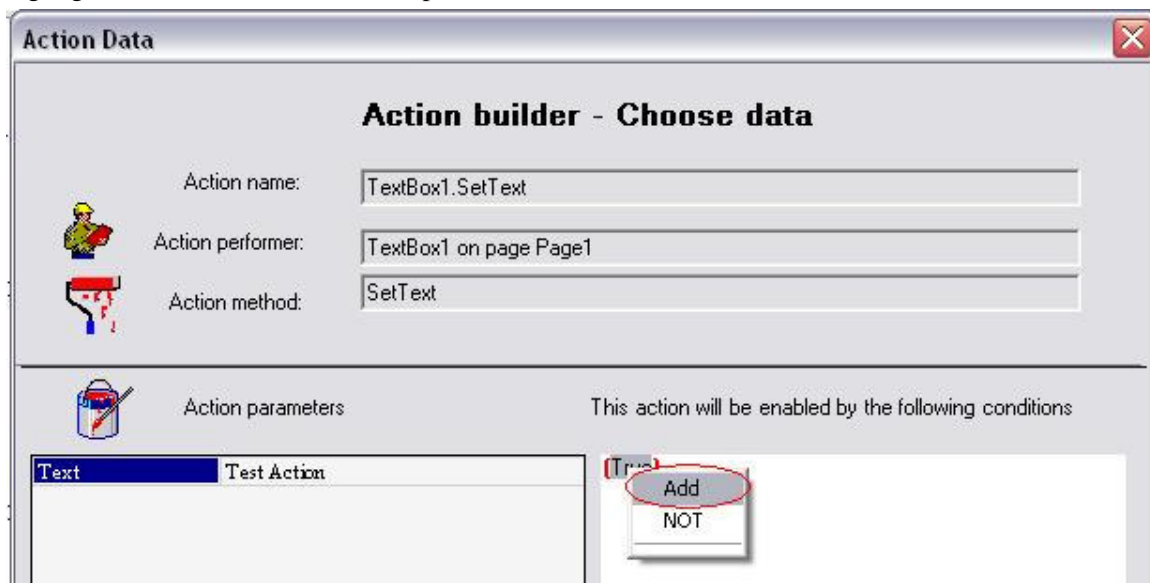


Action Conditions

To add conditions to an action, select the action, click Edit:



Click Next button, you can see the Action Data dialogue box. You can see word (True). It indicates an empty action condition. Click on (True), you can see it is highlighted. Right-click it while it is highlighted. A context menu comes up. Choose “Add”:

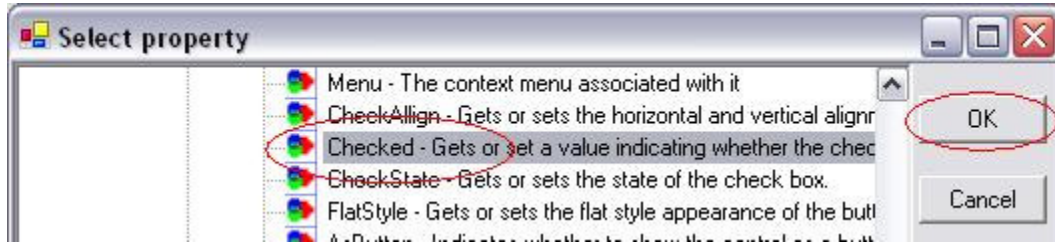


Find performer CheckBox1:

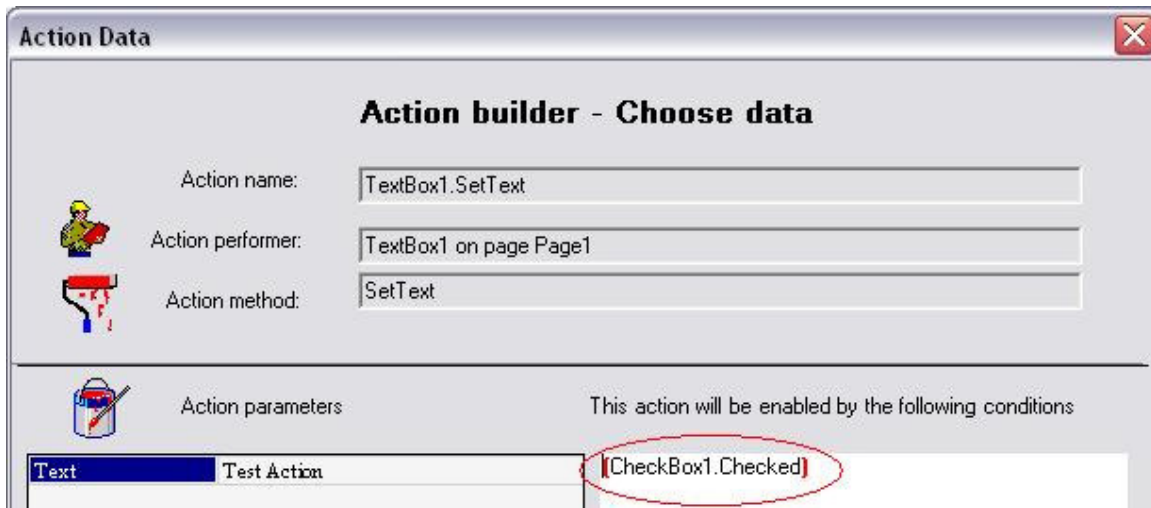


Find and select its “Checked” property:

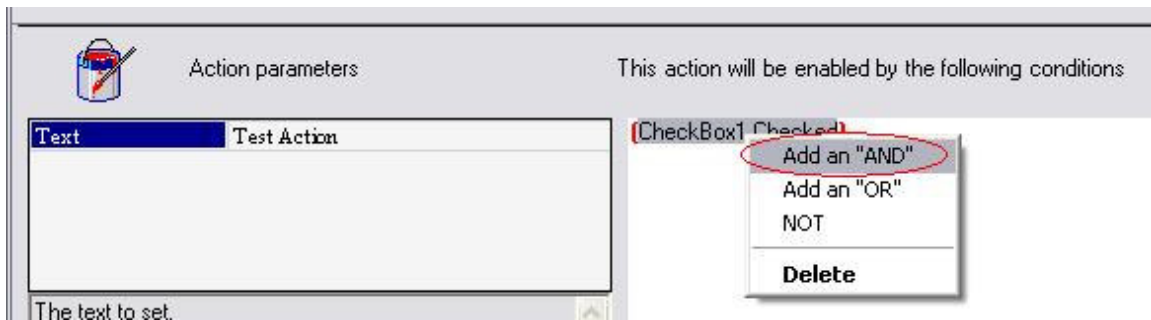
Action Conditions



The property “Checked” of CheckBox1 becomes the action condition:

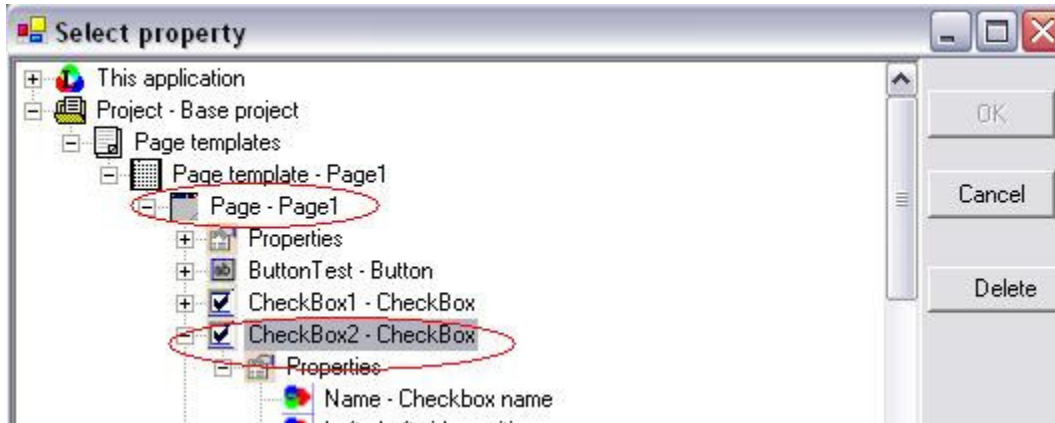


Now we want to add property “Checked” of CheckBox2 to the above condition. Click “CheckBox1.Checked” to highlight it, right-click it, select [Add an “AND”]:

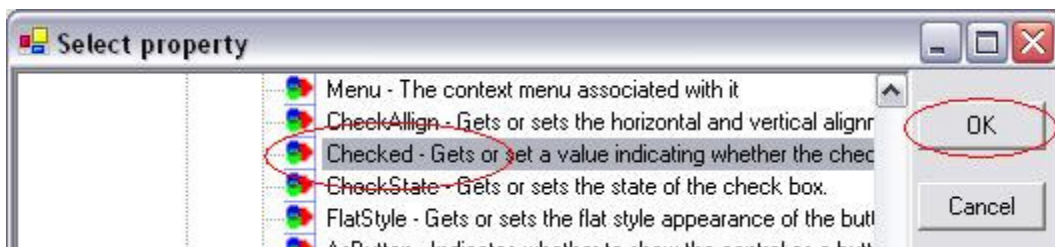


Select performer CheckBox2:

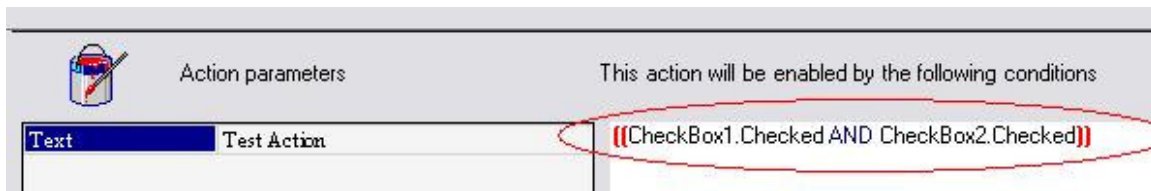
Action Conditions



Find and select its “Checked” property:



Now the action conditions become:



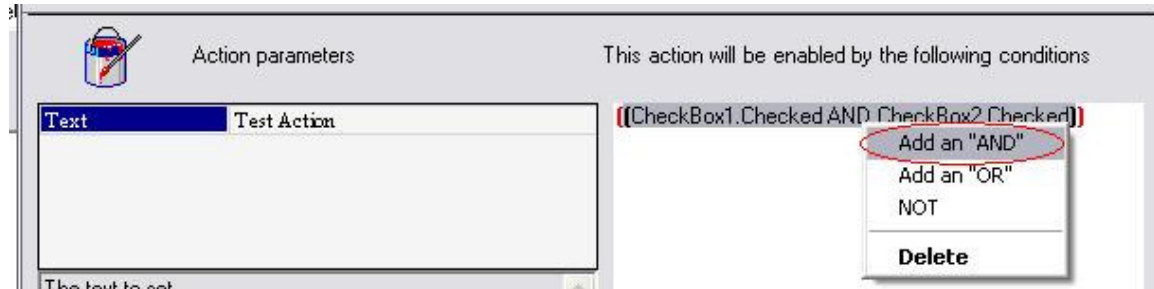
With the above conditions added this action can only be executed when both CheckBox1 and CheckBox2 are checked.

Now we want to add “Checked” property of CheckBox3 performer to the conditions. Click the word “AND” to select the whole conditions:

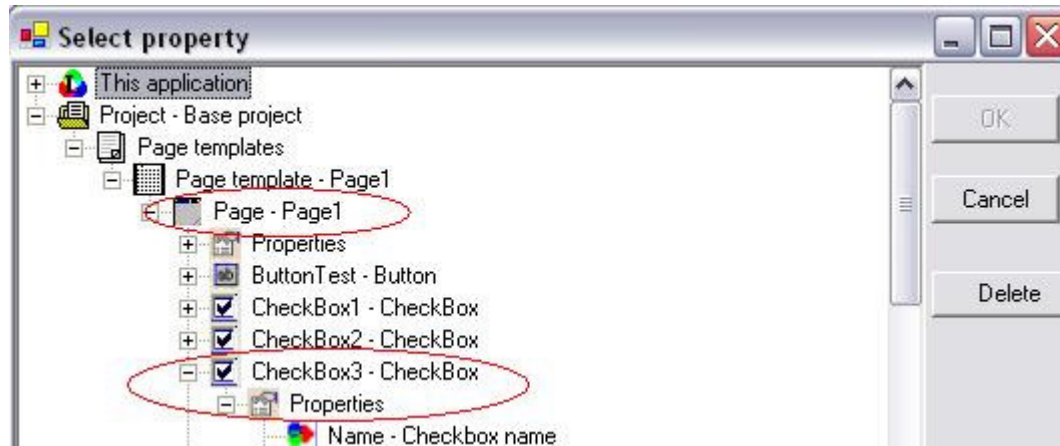


Right-click it while it is highlighted, choose [Add an “AND”]:

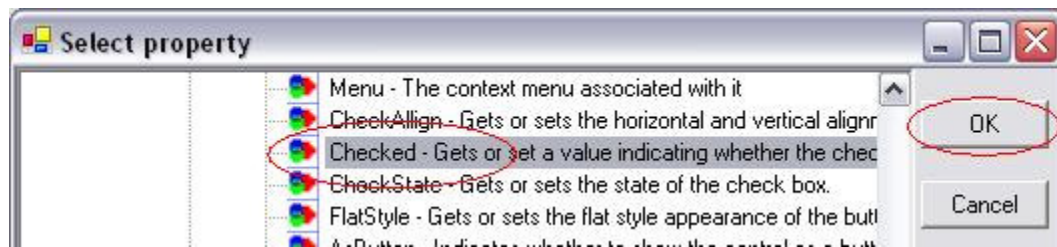
Action Conditions



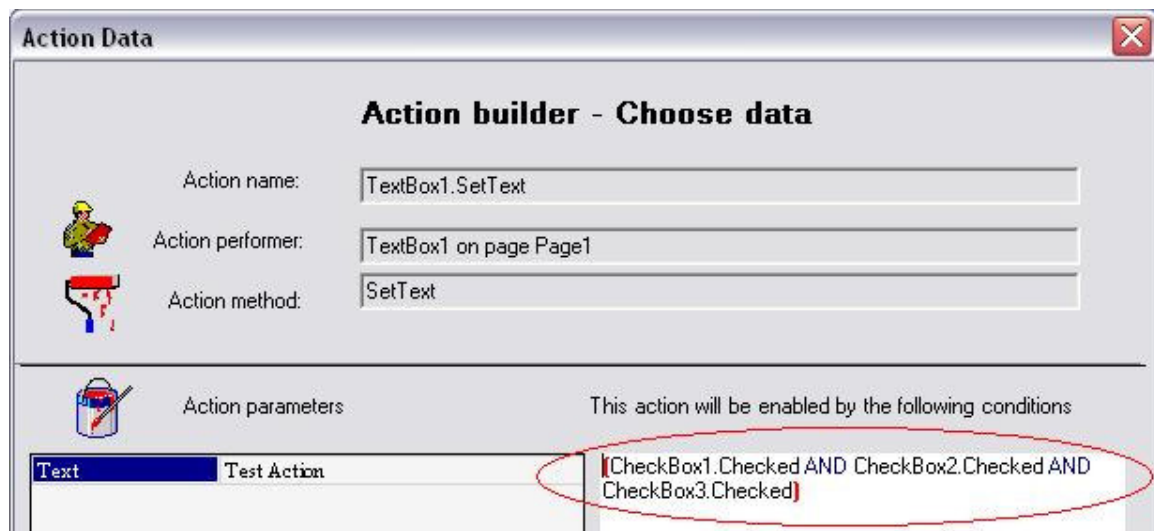
Select performer CheckBox3:



Find and select its "Checked" property:



Now we finish building the conditions:



With these conditions the action can be executed only when all 3 check boxes are checked.

Action Conditions

We may run the application and test it. You can see that if any one of the 3 check boxes is un-checked then clicking the button will do nothing. If all 3 check boxes are checked then clicking the button will display “Test Action” on the text box.

7 Sample Application – Add Conditions to Action List

Suppose we want to make the text box “flashing” by changing its background color and text color every second. The background color of a text box performer is indicated by its “BackColor” property. The text color of a text box performer is indicated by its “ForeColor” property. We may use “Set property” method to make actions to change these color properties.

We make following actions to change colors:

```
TextBox1.SetBackColorToRed  
TextBox1.SetForeColorToWhite
```

The above two actions make the text box has red background and white text.

```
TextBox1.SetBackColorToWhite  
TextBox1.SetForeColorToBlue
```


The above two actions make the text box has white background and blue text.

We assume you know how to create the above actions using “Set property” method. If you need help please contact support@limnor.com.

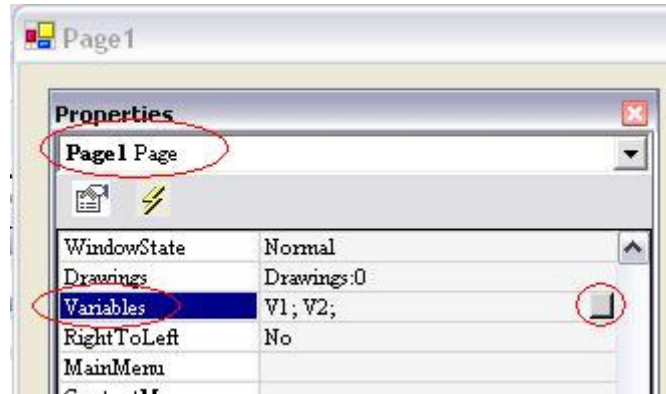
We will use a timer to call these actions to change the colors. The timer fires Timer event every second. Suppose at the first second, the first two actions are executed. The text shows red background and white text. The next second, the other two actions are executed, and the text shows white background and blue text.

Now we have a question: how do we control which set of actions to execute? The answer is to add a condition.

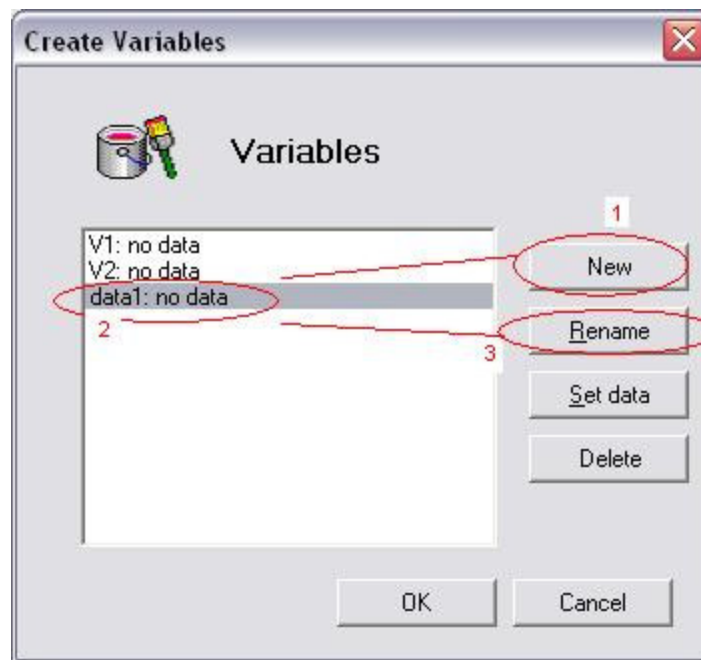
We use a variable to indicate whether the text box has red background color.

To create a variable under a page, bring up the properties window for the page, select “Variables” property, click :

Action Conditions



Click “New”, a new variable appears. Select the new variable, click “Rename” to give it a name of your choice:

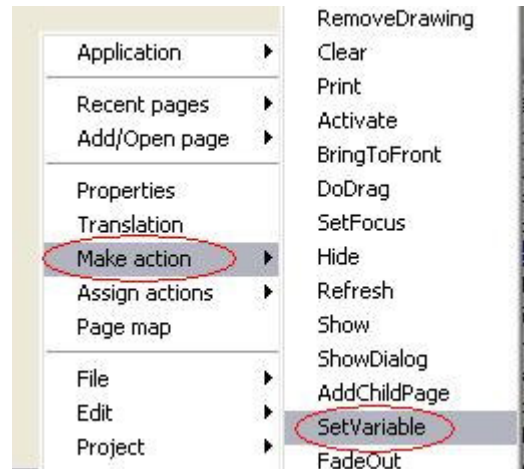


Let's name it “IsRed”:




Below we create two actions to set the value of IsRed to True and False, respectively. Right-click on the page, choose “Make action”, choose “Set variable”:

Action Conditions

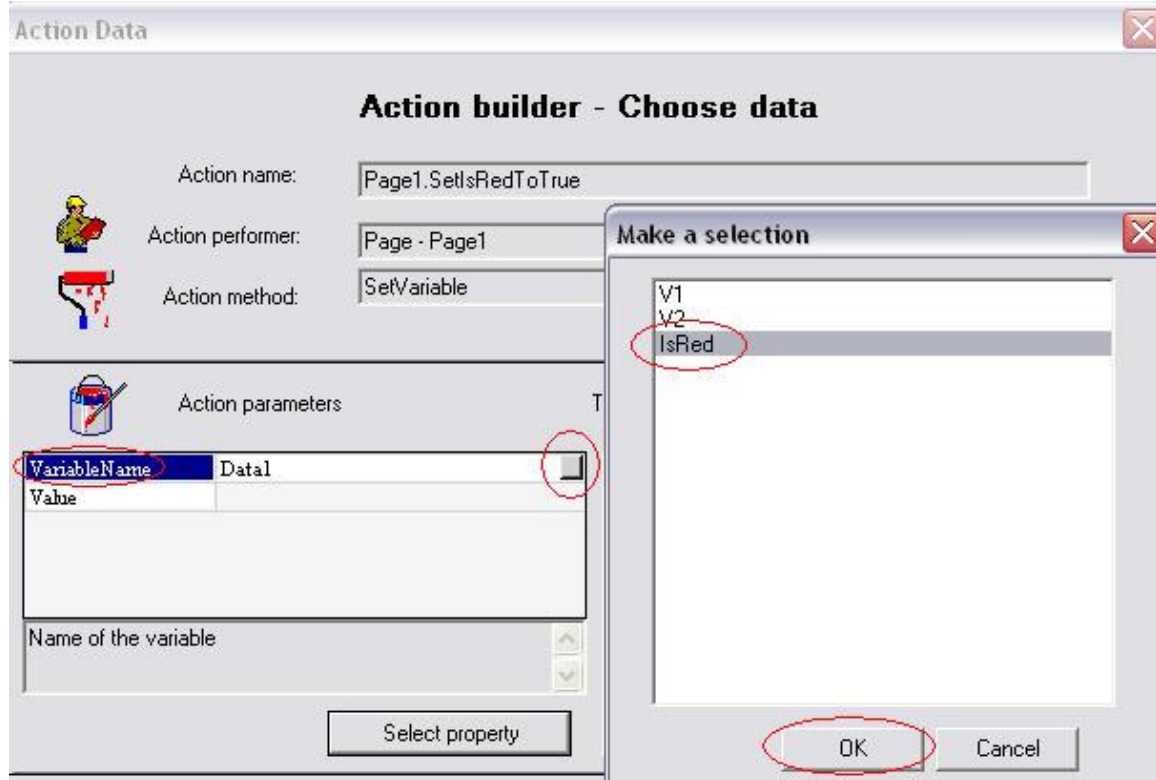


Give an action “Page1.SetIsRedToTrue”:

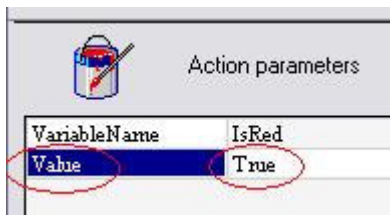


Select “VariableName”, click , select IsRed:

Action Conditions



Select "Value", type in True:



In the same way, we create another action named Page1.SetIsRedToFalse to set variable IsRed to False:



Now we have all the actions we needed. Our programming logic is:

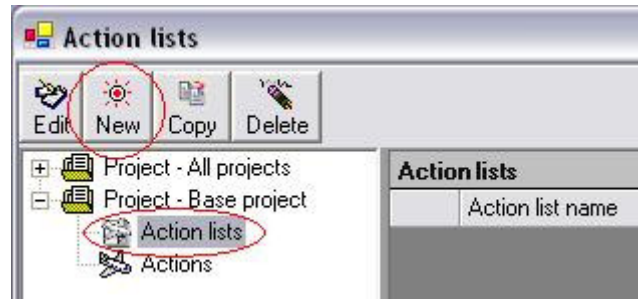
```
IF (IsRed is True) THEN
    TextBox1.SetBackColorToWhite
    TextBox1.SetForeColorToBlue
    Page1.SetIsRedToFalse
ELSE
    TextBox1.SetBackColorToRed
```

Action Conditions

```
TextBox1.SetForeColorToWhite
```

```
Page1.SetIsRedToTrue
```

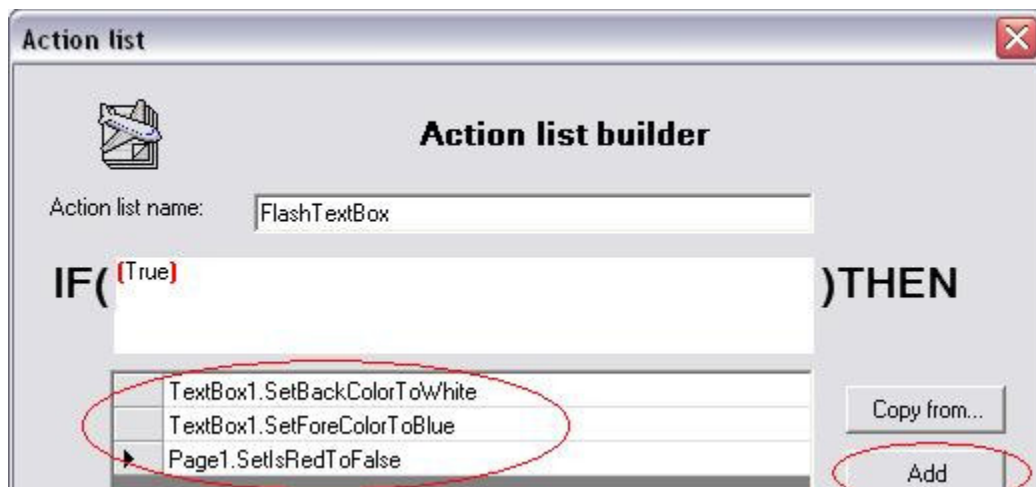
The above logic can be created by an action list. Select “Action List”, click “New” to create a new action list:




Give an action list name, say, FlashTextBox:



Click Add button to add actions:

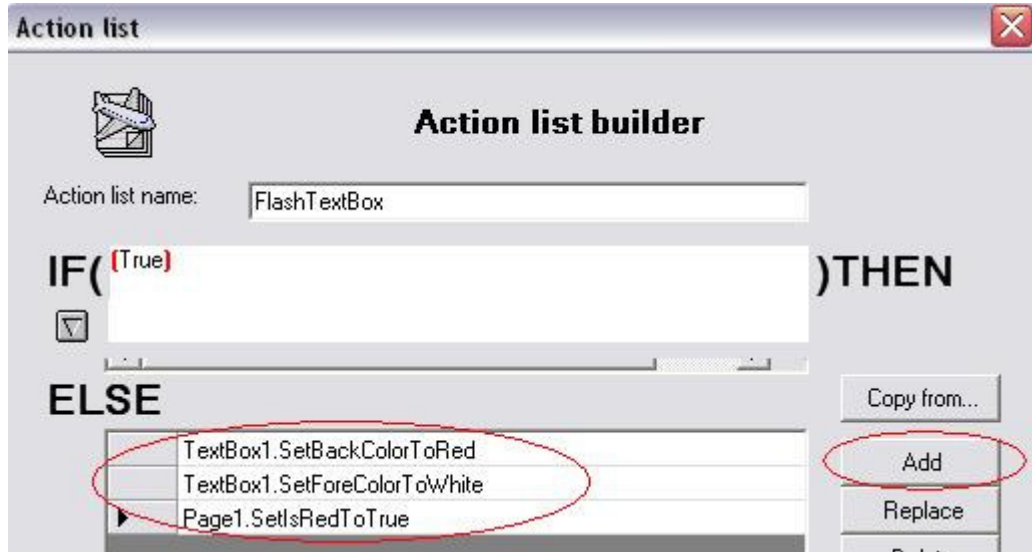


Click  to add actions to the ELSE part:

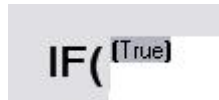


Click Add button to actions:

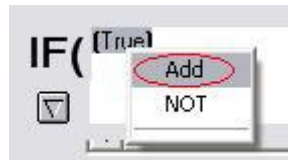
Action Conditions



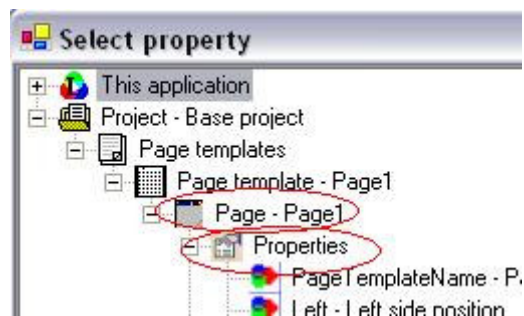
Now click (True) to highlight it:



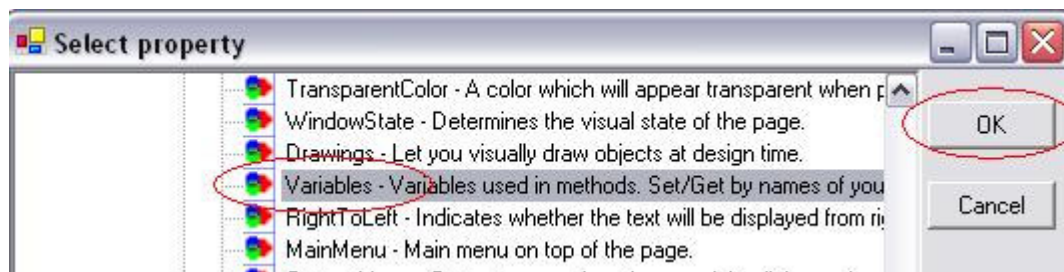
Right-click on it, choose "Add":




Find the page node and expand its properties node:

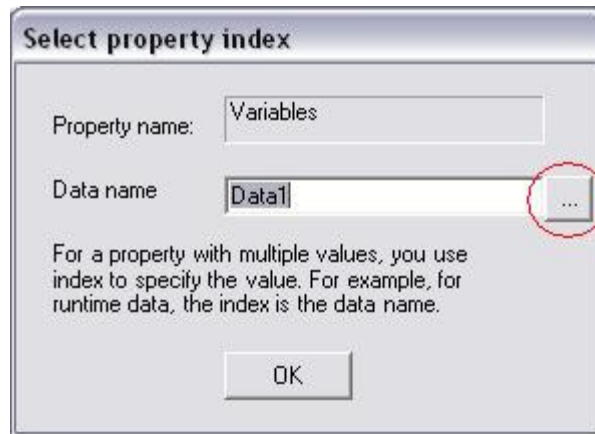


Find and select the page's Variable property, click OK:

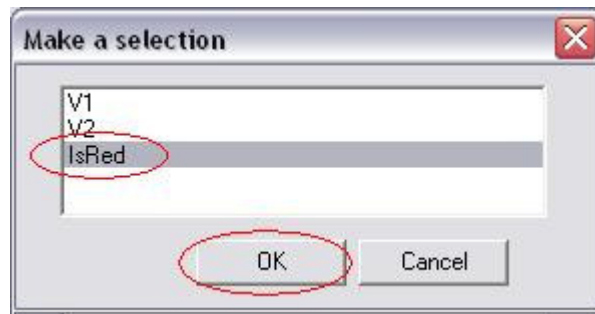


Action Conditions

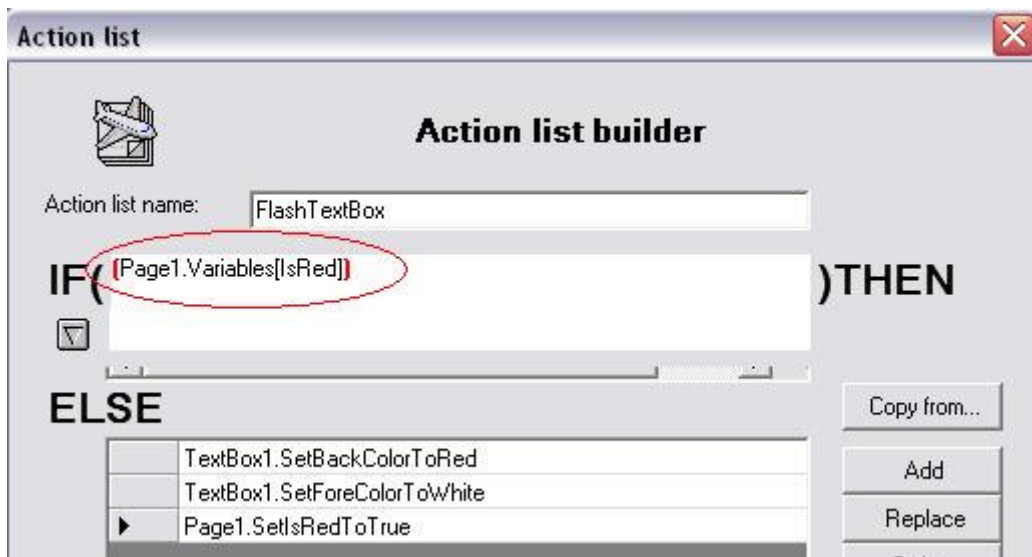
A dialogue box appears to let you select a variable. Click  to choose a variable from variable list:



Select IsRed and click OK:

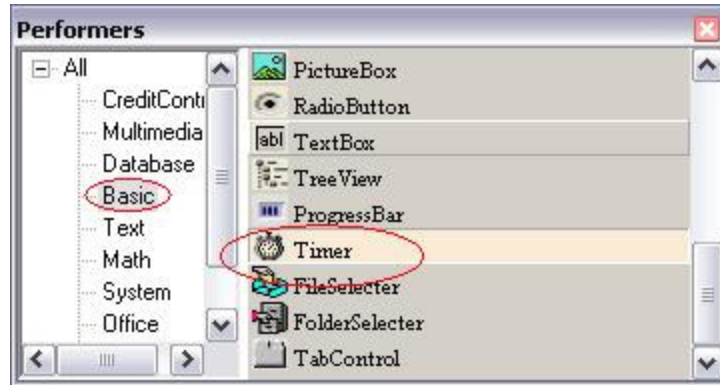


We are done making this action list:

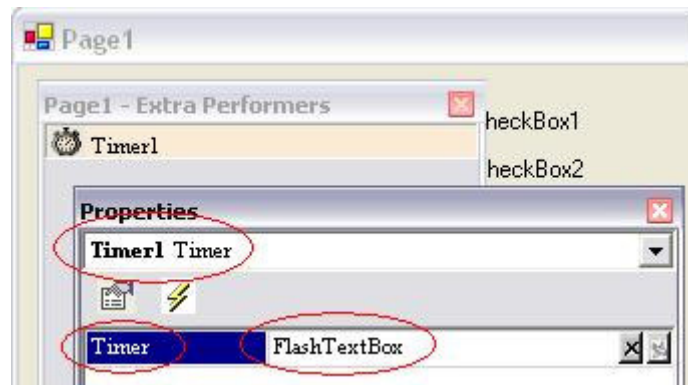


We add a Timer performer to the page:

Action Conditions



We assign the action list FlashTextBox to the Timer event:



We are done. Now run the application. You will see the text box changes its background color and text color every second.

8 Questions and Feedback

For questions and feedbacks please contact support@limnor.com

You may also get information on Limnor Community Portal at <http://limnorcommunity.jrwebb.net>