

Block Unwanted Web Pages – Limit to only one domain

The web browser performer shipped with Limnor releases provides many ways for you to block unwanted web pages. This article introduces one of such methods. Please read other articles for other methods.

In this sample application, we limit the web browser to only show web pages from the same domain. To do this is very simple. The first two sections describe how this is done. The other sections describe all the other programming details, not related to blocking unwanted web pages.

[How to block pages](#)

[Block Conditions – Same Domain](#)

[Web Navigation](#)

[Show URL on Address Bar](#)

[Make graphic buttons](#)

[Show Time](#)

[Use On-Screen Keyboard](#)

[Automatic resizing of the web performer](#)

[Screen Saver](#)

[Deployment](#)

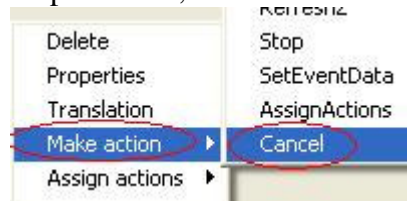
[Disable Design Mode](#)

[Distribute to other computers](#)

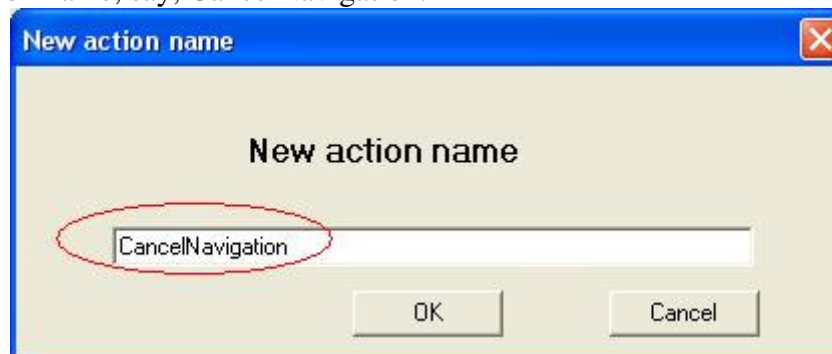
How to block pages

Suppose we already created a web browser performer on a page. A web browser performer has a “Cancel” method. We may use it to make an action to block page navigation.

Right-click on the web browser performer, choose “Make action”, choose “Cancel”:



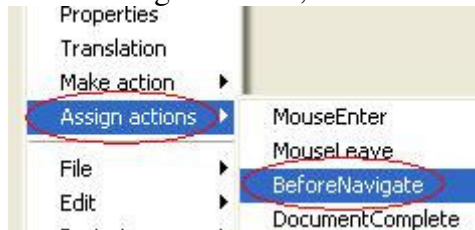
Give an action name, say, CancelNavigation:



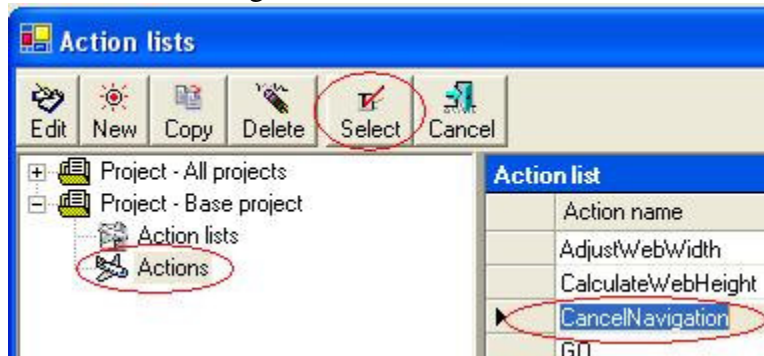
Web Browsing Limit to One Domain

Now this **CancelNavigation** will block page navigation. The next question is the timing: when is this action carried out? The answer is that this action should be carried out at the events **BeforeNavigate** and **NewWindow**. The event **BeforeNavigate** occurs before the web browser going to a new page. The event **NewWindow** occurs before the web browser pops up a new window showing a web page.

Suppose we want to block the web browser from going to a new page. Right-click on the web browser performer, choose “Assign actions”, choose “BeforeNavigate”:



Select Actions, select **CancelNavigation**, click “Select” button:



The above programming assigns action **CancelNavigation** to event **BeforeNavigate**.

This way, we blocks page navigation on the web browser.

To block pop-up window, we just assign action **CancelNavigation** to event **NewWindow**. Right-click on the web browser performer, choose “Assign actions”, Choose “NewWindow”:



Just as we did before, we select action **CancelNavigation** for it. Now all pop-up windows will be blocked.

Block Conditions – Same Domain

If we run the application now, we will see that the web browser will not show any web pages. This is because we blocked all pages. This is certainly not what we want. We just want to block certain pages, or just allow certain pages, or use other criteria to determine which pages to block.

Web Browsing Limit to One Domain

In this sample, we want to block all pages that are **not** from the domain of the first page initially shown on the web browser.

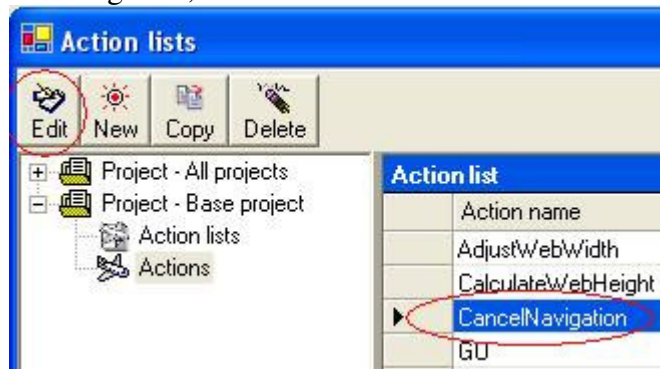
The web browser performer has a property “**SameDomain**”. If this property is True, the new page for navigation or pop-up is from the same domain of the current page, or the current page is empty. Therefore, for our purpose, we should execute action CancelNavigation only when the “SameDomain” property is False. In other words, if “SameDomain” property is True, we should disable action CancelNavigation.

In Limnor, you may enable/disable every action or action list by properties. In this case, we need to use “SameDomain” property to disable CancelNavigation.

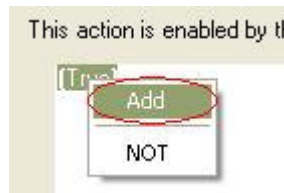
Open “Actions” window by clicking “Actions” or choose “Project | Action lists” menu:



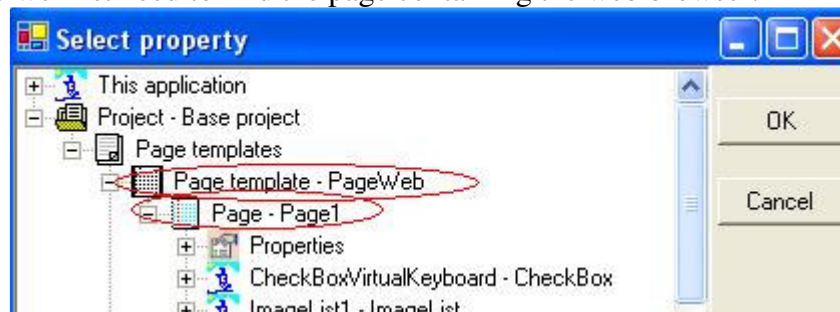
Choose action CancelNavigation, click “Edit” button:



Click “Next” button. It lets you select execute condition for the action. Right-click on the word “True”, choose menu “Add” to add condition:

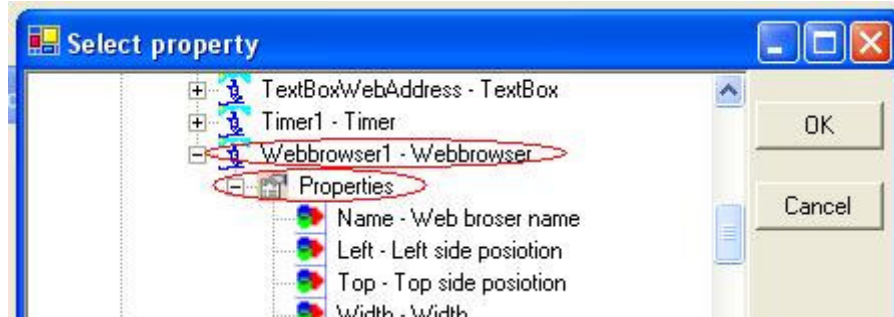


It will ask you to select a property. We want to use “SameDoamin” property of the web browser. So we first need to find the page containing the web browser:

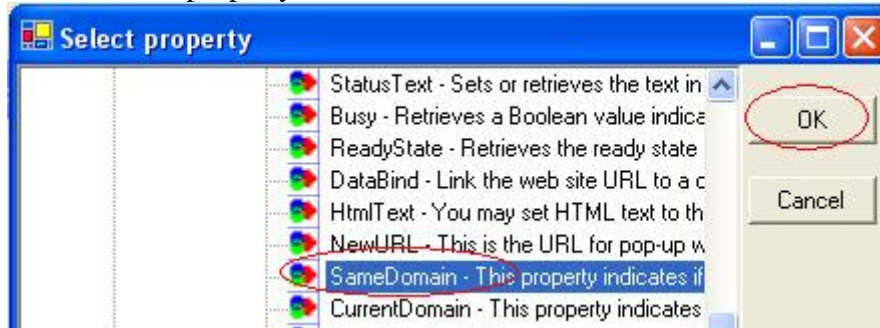


Under the page, we may find the web browser performer:

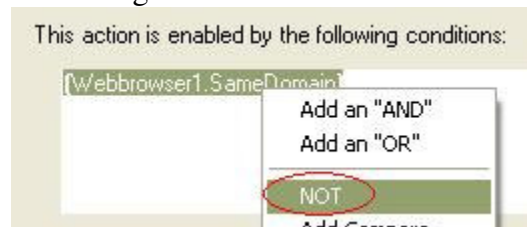
Web Browsing Limit to One Domain



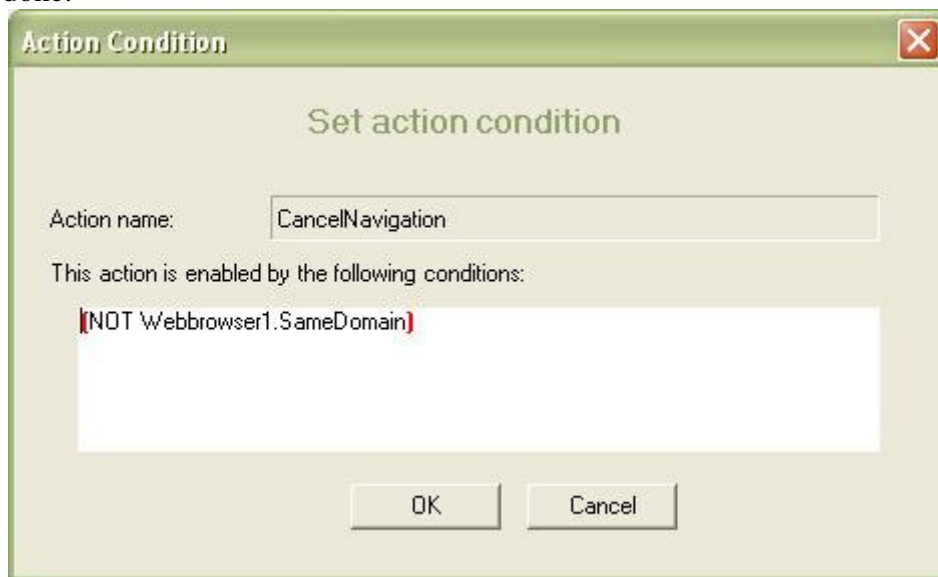
Find its "SameDomain" property, select it and click OK:



Now the property SameDomain is used as the condition. We need the condition to be when SameDomain is not true. Right-click on it and choose "NOT":



We are done:



In this example, we set the URL property of the web browser to <http://www.ibm.com>:

Web Browsing Limit to One Domain

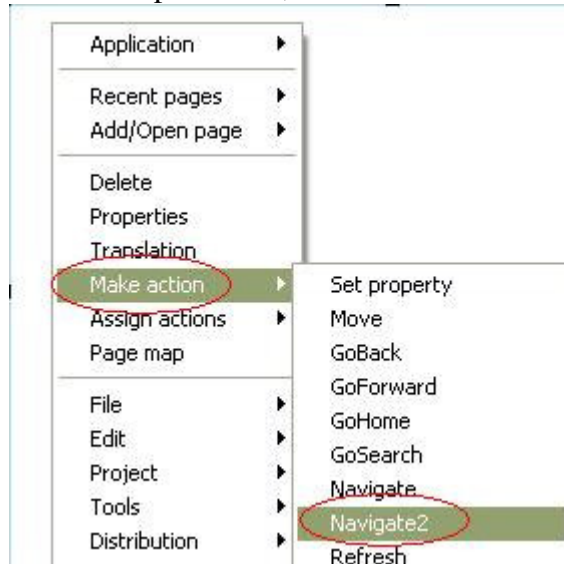


So when you run the application, the web browser will show the home page of the IBM web site. The user may navigate through the IBM web site. But if the user tries to type in a web address outside of IBM web domain, the web browser will refuse to go there.

Web Navigation

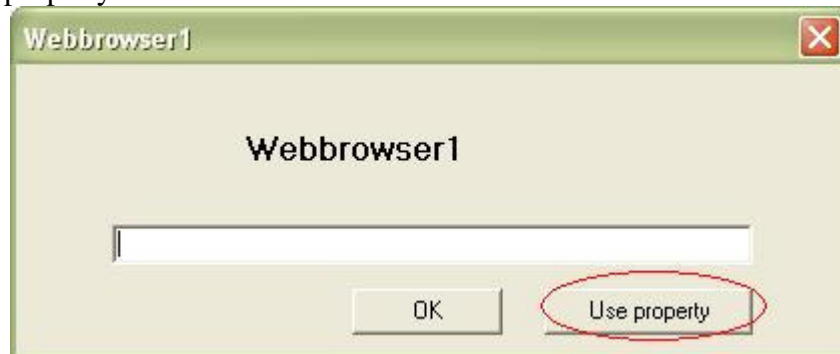
You may use a TextBox to let the user type web address. To navigate to the web address you may create an action using Navigate2 method of the web browser performer.

Right-click on the web browser performer, choose “Make action”, select “Navigate2”:



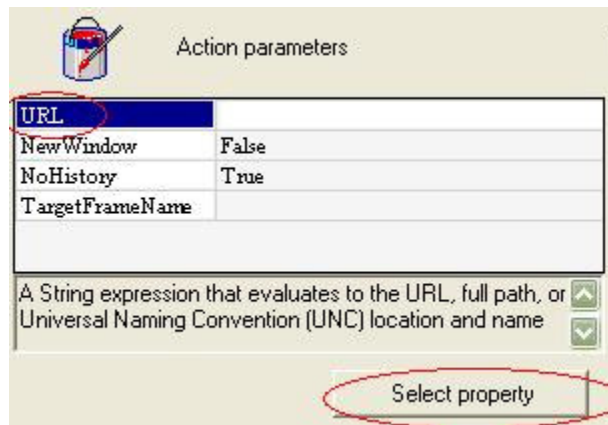
After giving an action name, say, GO, it will ask for the web address to go to. We want to navigate to the web address represented by the Text property of the TextBox performer.

Click “Use property”:

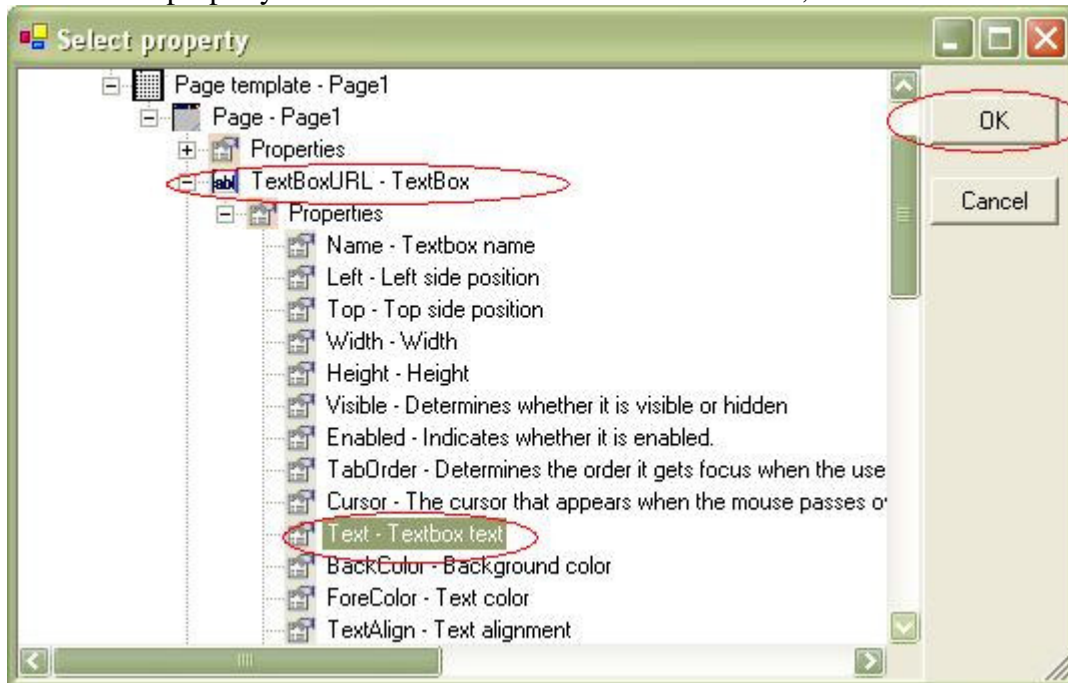


The “Action data” dialogue box appears. Select URL parameter and click “Select property”:

Web Browsing Limit to One Domain



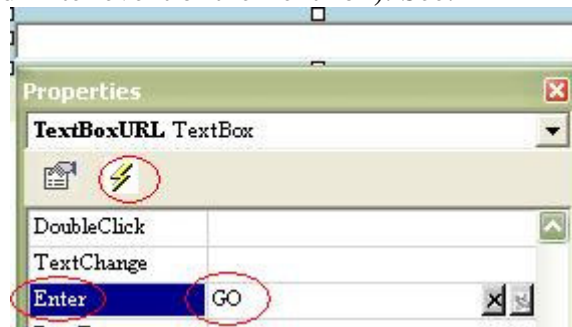
Select the Text property of the TextBox used for the web address, click OK:



Click "Finish" to finish making this action.

We have the action to do navigation. The next question is when to do the navigation.

Typically, we may do the navigation when the user clicks a button (GO button) or when the user presses <Enter> key. We just need to assign the action to these two events (Click event of the button and Enter event of the TextBox). See:



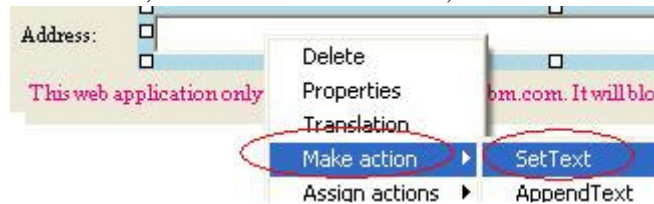
Show URL on Address Bar

Web Browsing Limit to One Domain

When the user navigates the web, the web page address should appear in the address bar. To do this, we may set the URL property of the web performer to the Text property of the address bar. We must do this property setting at the event of `NavigateComplete2` of the web performer. This event occurs when a new web page finishes loading.

Now let's make an action to do this property setting, and assign this action to `NavigateComplete2`.

Right-click on the address bar, choose "Make action", choose "SetText":



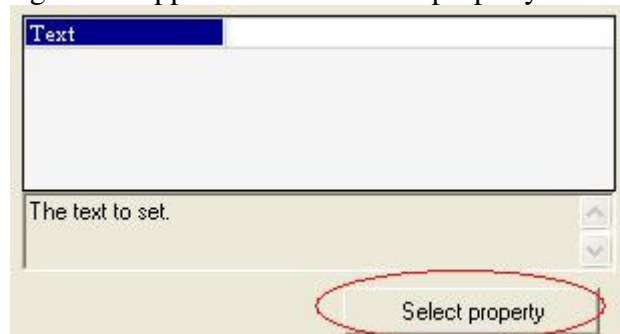
Give an action name, say, **ShowURL**,



It asks you to type in text. Click Cancel because we want to use a property:

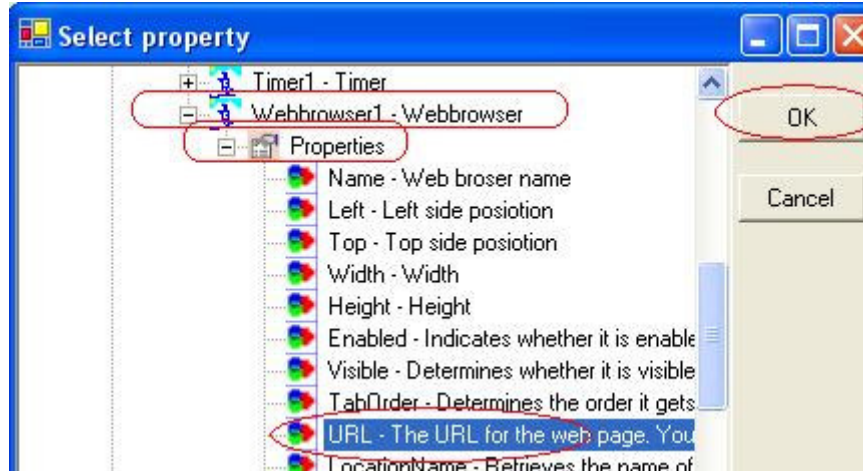


The Action Data dialog box appears. Click "Select property" button:



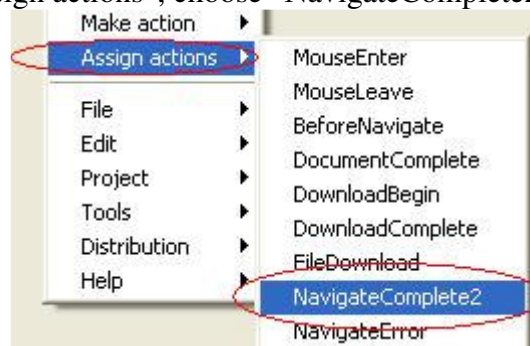
Find the web performer and select its URL property:

Web Browsing Limit to One Domain

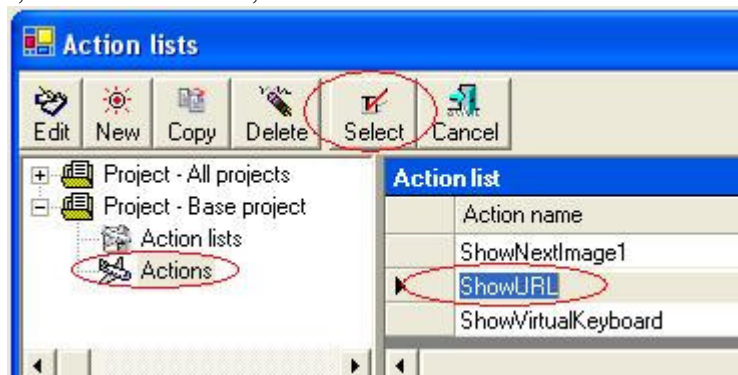


Click OK.

Now we need to assign this action to **NavigateComplete2** event. Right-click on the web performer, choose “Assign actions”, choose “NavigateComplete2”:



Select “Actions”, select ShowURL, click “Select” button:



Make graphic buttons

We may use a label performer to make a graphic button. We may show image on a label. When the user moves mouse pointer into a label, the label fires a “MouseEnter” event. At this event, we may show a different picture on the label to indicate that the label is waiting to be clicked.

Web Browsing Limit to One Domain

When the user moves mouse pointer out of a label, the label fires a “MouseLeave” event. At this event, we may show the original picture on the label to restore the state.

When the user presses mouse button down, the label fires a “MouseDown” event. At this event, we may display a picture on the label to graphically show that the label is pressed down just as a 3D button is pressed.

When the user releases mouse button, the label fires a “MouseUp” event. At this event, we may display the same picture on the label as when mouse move in.

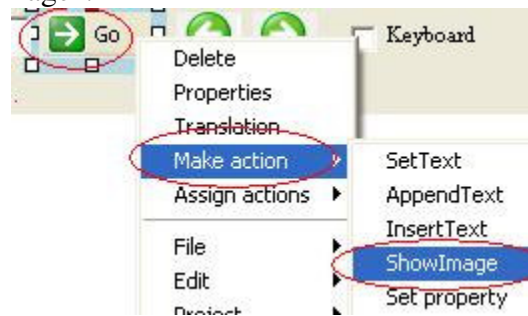
So we need 3 pictures to achieve this animation effects:

Picture 1 for normal button state;

Picture 2 for active button state indicates that the mouse pointer moves in.

Picture 3 for button pressed down state to show that the button is pressed.

To make an action to show a picture on a label, right-click on the label, choose “Make action”, choose “ShowImage”:



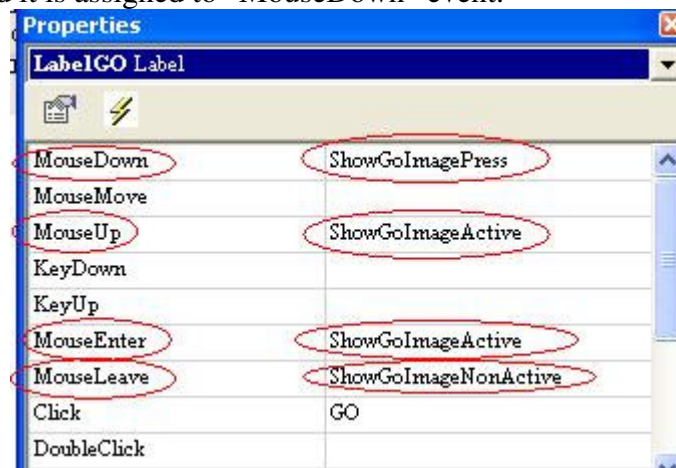
You may create many actions using “Make action” and “ShowImage” operation, and give different action name and different picture file name each time.

In the sample, label performer named LabelGO is programmed in the above way.

Action **ShowGoImageNonActive** is made to show picture for normal button state, and this action is assigned to “MouseLeave” event.

Action **ShowGoImageActive** is made to show a picture to indicate active button when mouse is moved in, and it is assigned to “MouseEnter” event, and “MouseUp” event.

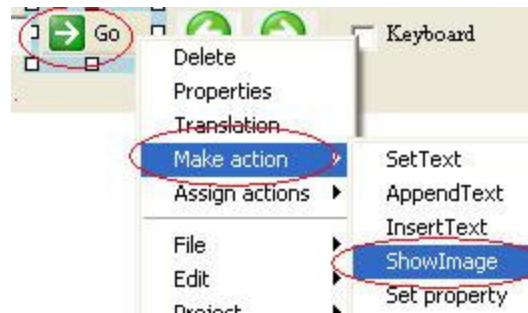
Action **ShowGoImagePress** is made to show a picture to indicate mouse button is pressed down, and it is assigned to “MouseDown” event.



Now the label LabelGO behaves like an animated graphic 3D button. Below, we show you how the action **ShowGoImageNonActive** is made and how it is assigned to the “MouseLeave” event.

Right-click on LabelGO performer, choose “Make action”, choose “ShowImage”:

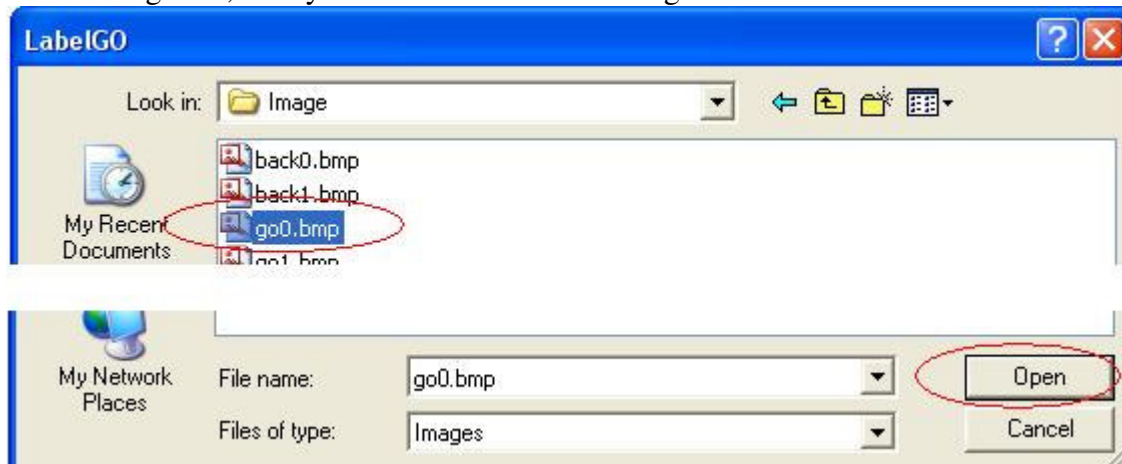
Web Browsing Limit to One Domain



Give an action name. In this case, we name it **ShowGoImageNonActive**:

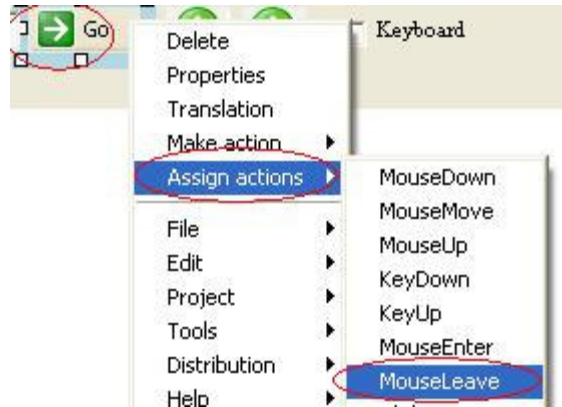


It lets you specify the image file to be used for this action. Suppose you already prepared such an image file, now you need to choose that image file:

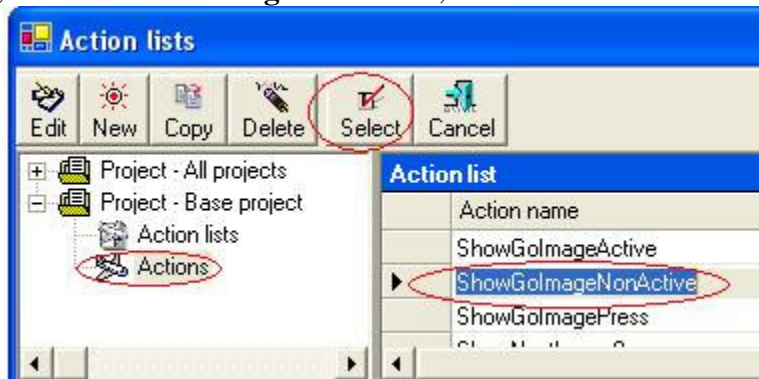


To assign this new action to “MouseLeave” event, right-click LabelGO performer, choose “Assign actions”, choose “MouseLeave”:

Web Browsing Limit to One Domain



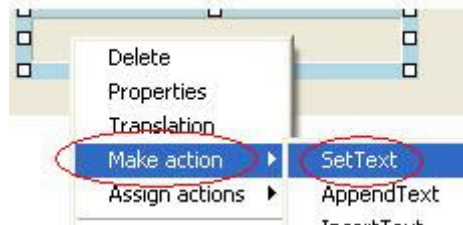
Select Actions, select **ShowGoImageNonActive**, click Select button:



Show Time

We use a label to show time.

The Application performer has a property called “Now”. This property is the current time. We create an action to set this “Now” property to the Text property of the label performer. Right-click on the label performer, choose “Make action”, choose “SetText”:



Give an action name, say, RefreshTime,

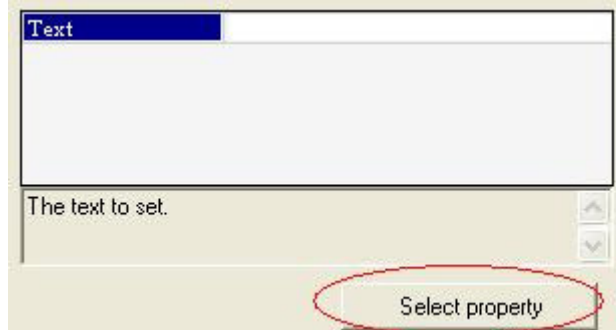
Web Browsing Limit to One Domain



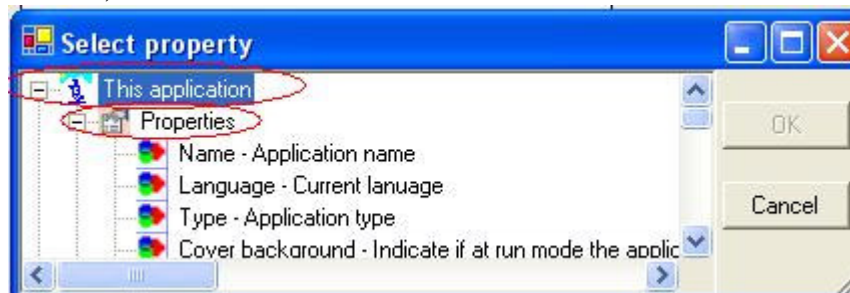
It asks you to type in text. Click Cancel because we want to use a property:



The Action Data dialogue box appears. Click "Select property" button:

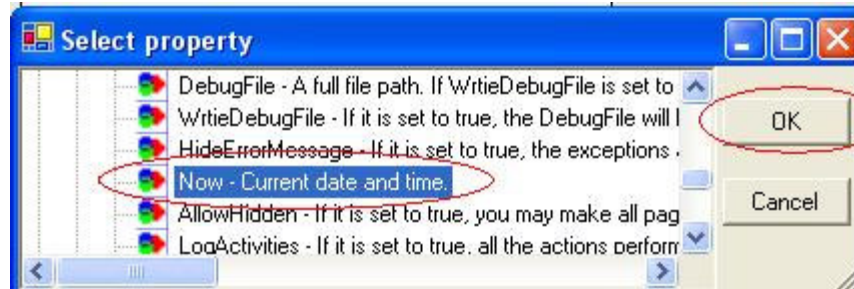


Select "Now" property of the Application performer. The Application performer is the first item in the list;

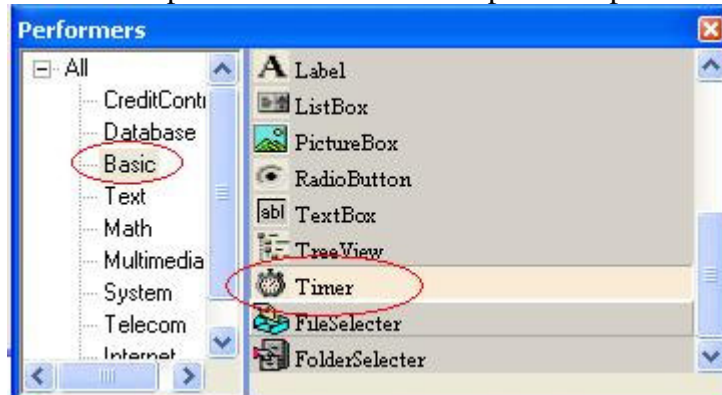


Scroll down to select the "now" property:

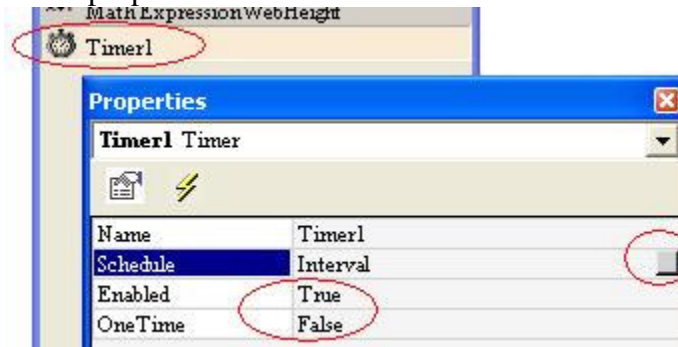
Web Browsing Limit to One Domain



We want to refresh time display every one second. That is, to execute this action every one second. We use a Timer performer to do this. Drop a timer performer to the page:

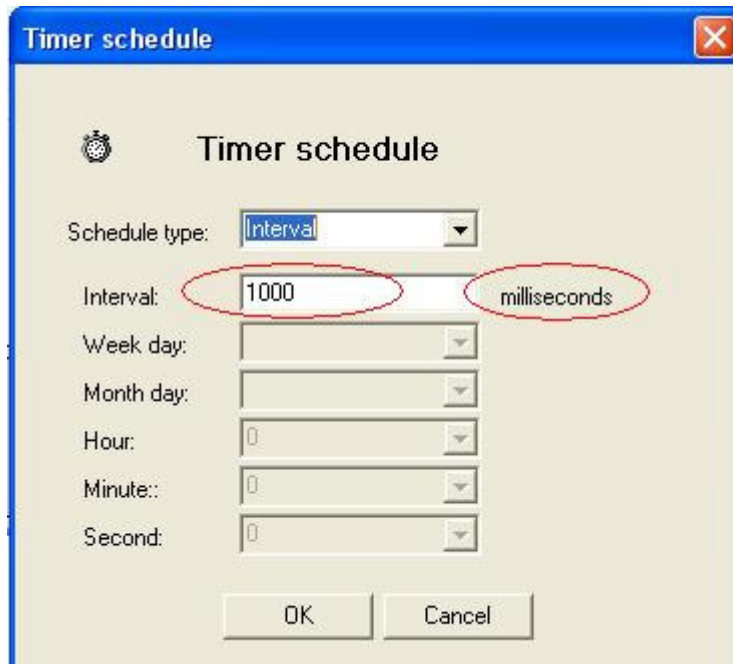


Set the timer performer properties:

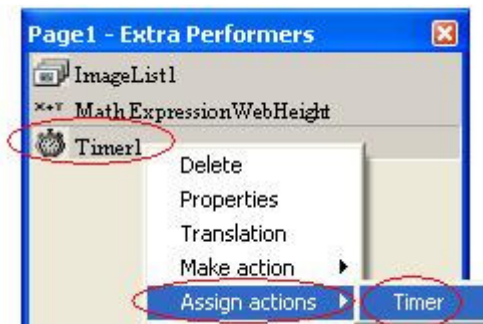


Set the Schedule property:

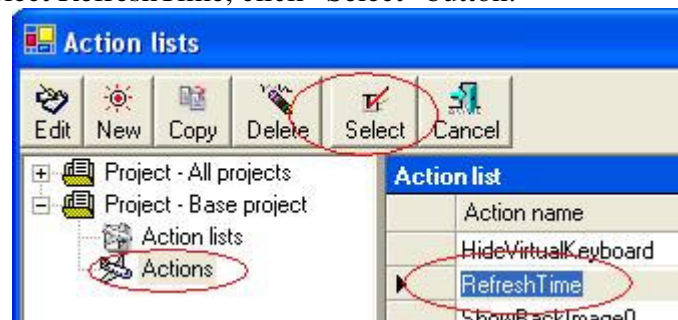
Web Browsing Limit to One Domain



Once we set the timer performer in the above way, it will generate a Timer event every one second. We assign action RefreshTime to Timer event so that this action will be executed every one second. Right-click on the timer performer, choose “Assign actions”, choose “Timer”:



Select Actions, select RefreshTime, click “Select” button:



Use On-Screen Keyboard

For touch-screen kiosks, you may use on-screen keyboard to let users input data. You may create your own on-screen keyboard using Limnor’s button performers. You may

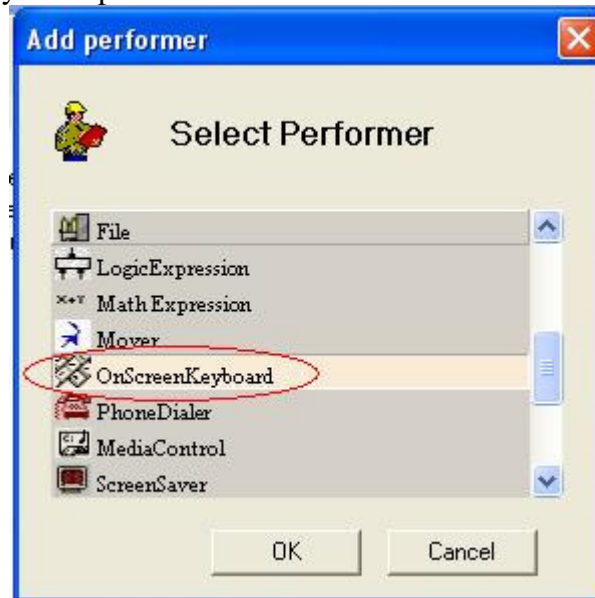
Web Browsing Limit to One Domain

also use ready made on-screen keyboard performer. You need to add on-screen keyboard performer to the Application performer.

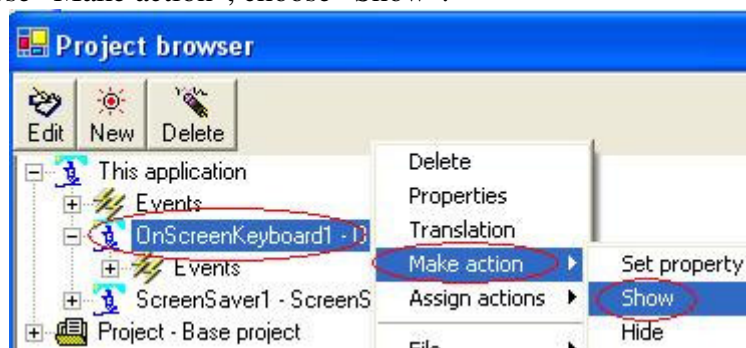
Open Project Explorer, the Application performer is the first object. Select Application, click New button to add performer to the Application performer:



Select On-Screen Keyboard performer:



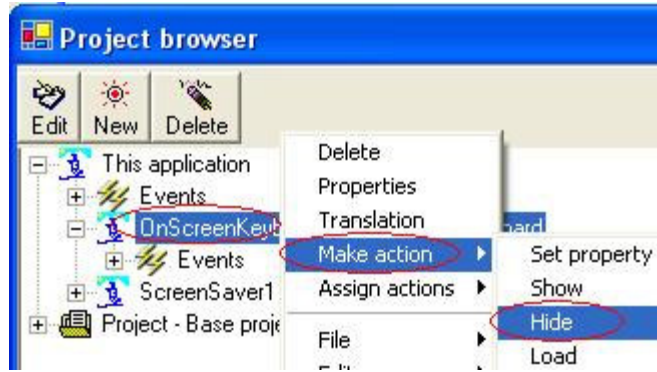
The on-screen keyboard performer appears under Application performer. We will create two actions to show and hide the on-screen keyboard. Right-click on the on-screen keyboard, choose “Make action”, choose “Show”:



Give an action name, say, ShowVirtualKeyboard.

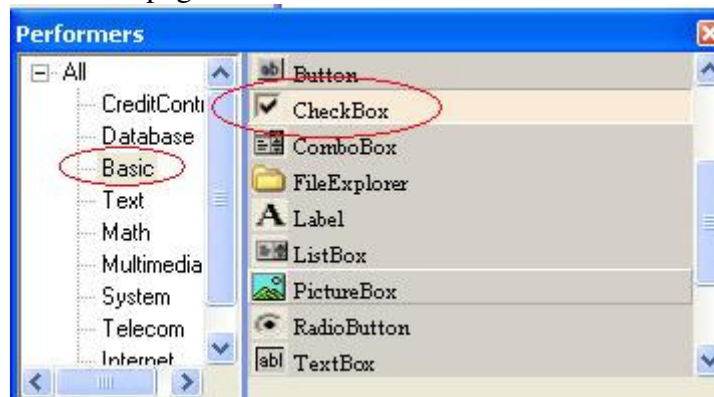
Right-click on the on-screen keyboard, choose “Make action”, choose “Hide”:

Web Browsing Limit to One Domain

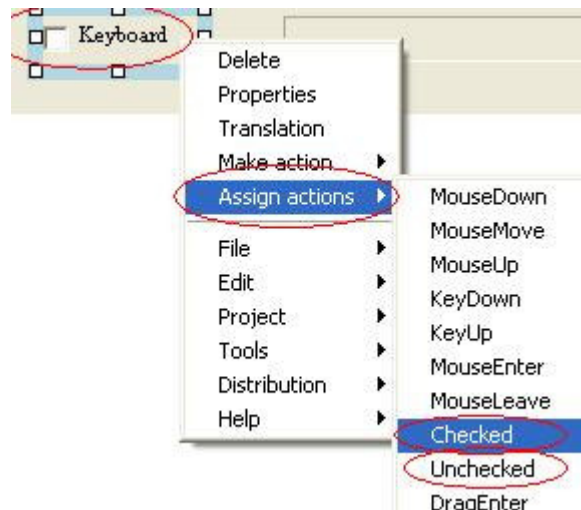


Give an action name, say, HideVirtualKeyboard.

We use a check box performer for the user to show and hide on-screen keyboard. Drop a check box performer to the page:



When the user checks a checkbox, the checkbox fires a “Checked” event. We may assign ShowVirtualKeyBoard action to “Checked” event. When the user un-checks a checkbox, the checkbox fires an “Uncheck” event. We may assign HideVirtualKeyboard action to “Unchecked” event:

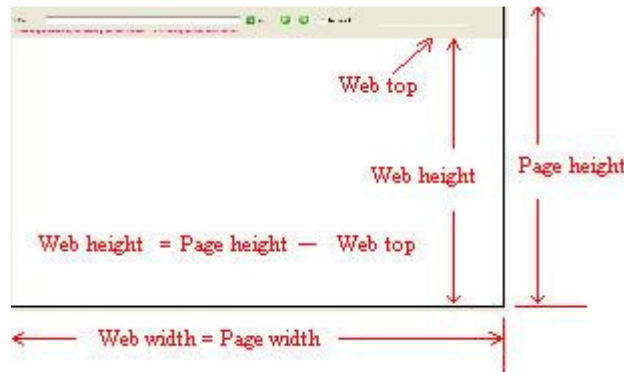


Automatic resizing of the web performer

Web Browsing Limit to One Domain

When you use this application under different screen resolutions, we want the application may automatically resize the web page to use all the screen area. We may adjust the Width and Height properties of the web performer at the page event Resize. Resize event occurs when the page is loaded and when the size of the page is changed.

What value should we set to the Width and Height properties of the web performer? The answer depends on how you design your screen layout. For this ample, we let the web browser fill the whole width of the page. We leave some space on the top of the page for placing address bar, “Go” button and other buttons. The web performer fills the rest of the page all the way to the bottom.




Set Width. Because of this screen layout design, we just use the Width performer of the page to set the Width property of the web performer. Let’s make an action to do that. Right-click on the web performer, choose “Make action”, choose “Set property”:

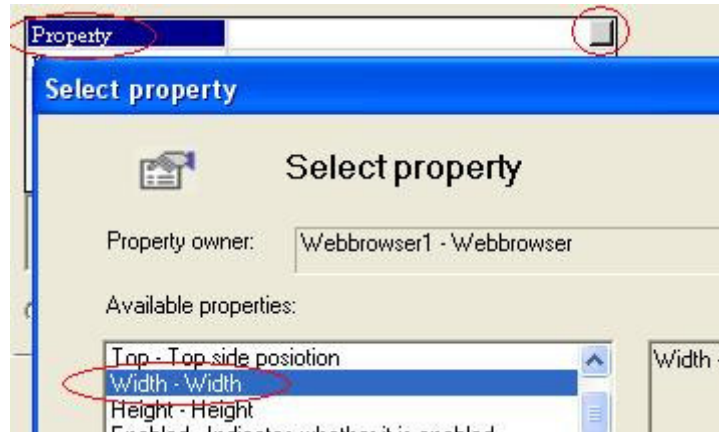


Give an action name, say, AdjustWebWidth:



The Action Data dialogue box appears. For the Property to be set, click  and choose “Width”:

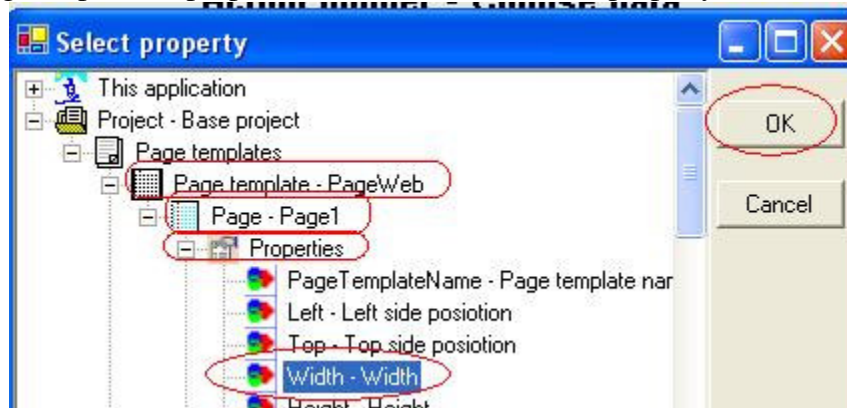
Web Browsing Limit to One Domain



For the Value to be set to the property, we want to use the Width property of the page. So we click “Select property” button:



Find the page, expand its properties node, select its Width property:



Click OK button and then click Finish button. This action is made.

Set Height. Referring to the screen layout design, we know we may calculate the height of the web performer as:

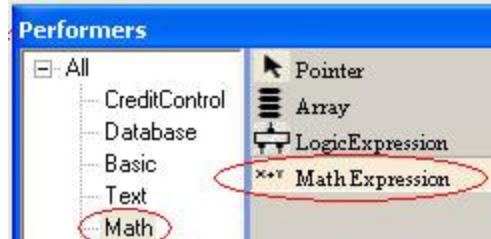
$$\text{Web Height} = \text{Page Height} - \text{Web Top}$$

We need to use a Math Expression performer to use the above formula. Right-click on a page, choose “Tools | Toolbox” menu:

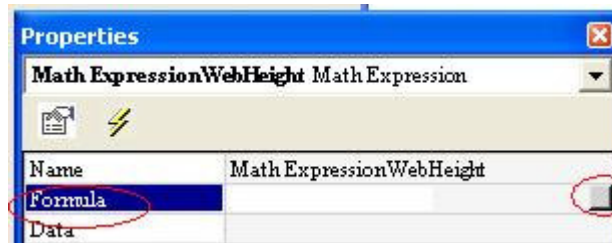


The Toolbox appears, showing all the performers. Drop a Math Expression performer to the page:

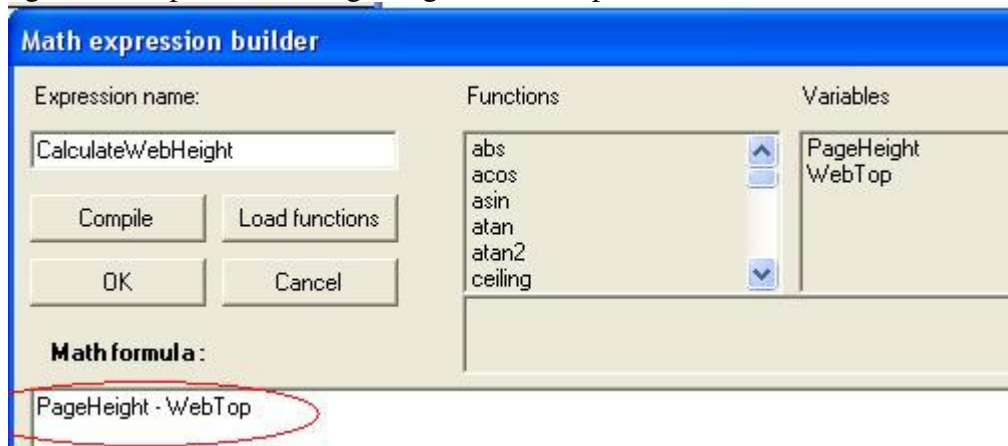
Web Browsing Limit to One Domain



The Math Expression performer appears in the Extra-Performer window for the page. You may right-click on it and choose “Properties” to set its properties. For example, change its name and formula.

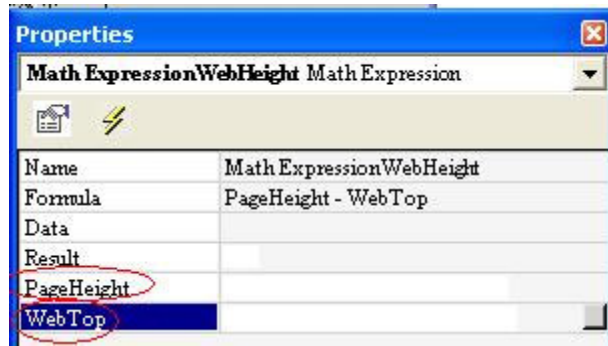


In the Math Expression Builder, type in the math expression we want to use to calculate the height of web performer: $\text{PageHeight} - \text{WebTop}$



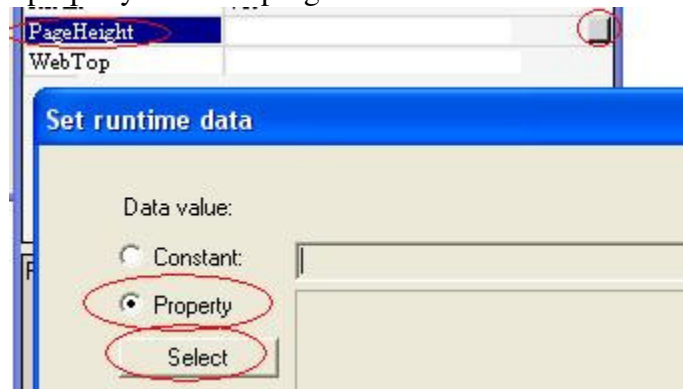
Note that, in this math expression, PageHeight and WebTop are two variable names. We may use any variable names. It does not affect the calculation. Do not use spaces in the variable name. And the first letter of a variable name must not be a number. Click OK button. You will see that all variables in the math expression become properties:

Web Browsing Limit to One Domain

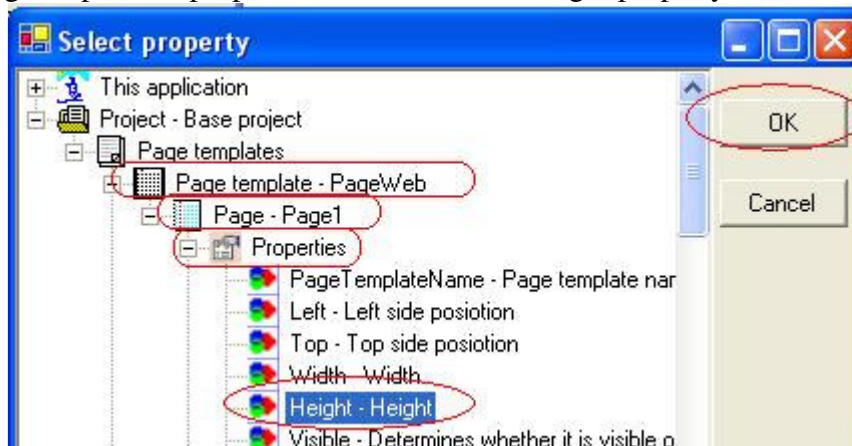


We need to set these properties to the values we want to use.

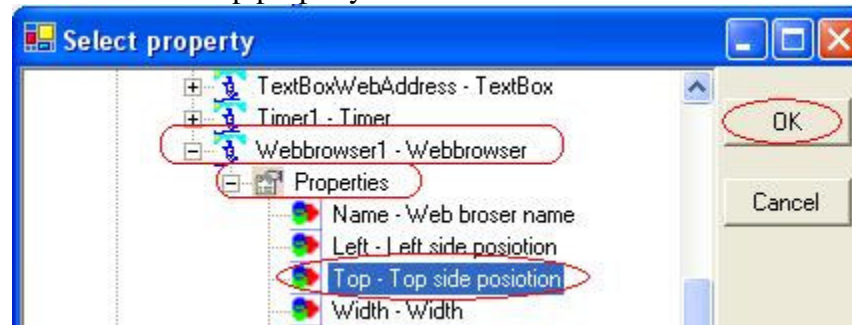
Select "PageHeight" property, click . Choose "Property" as the value, click "Select" button to select the property from the program:



Find the page, expand its properties node, select its Height property:

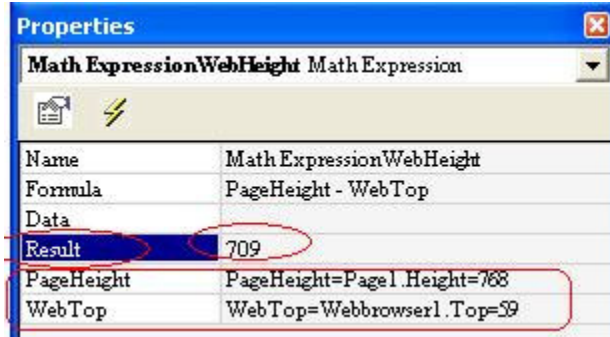


Do the same thing for the WebTop property. This time, find the web performer under and page node, and choose its Top property:

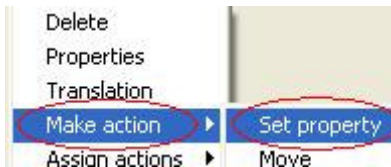


Web Browsing Limit to One Domain

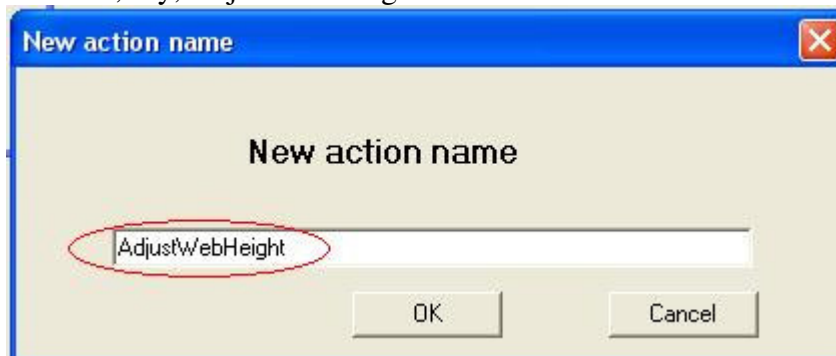
Once we assigned the values to the variables, the Result property of the math expression is the calculation result:



Now we make an action to set the calculation result to the Height property of the web performer. Right-click on the web performer, choose "Make action", choose "Set property":



Give an action name, say, AdjustWebHeight:



The Action Data dialogue box appears. For the Property to be set, choose Height:

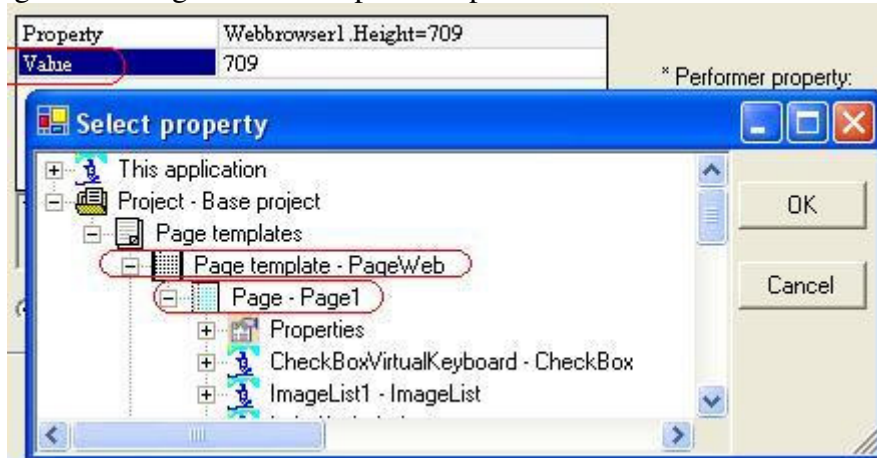


For Value, we want to use the Result property of the Math Expression performer. So we click "Select property" button:

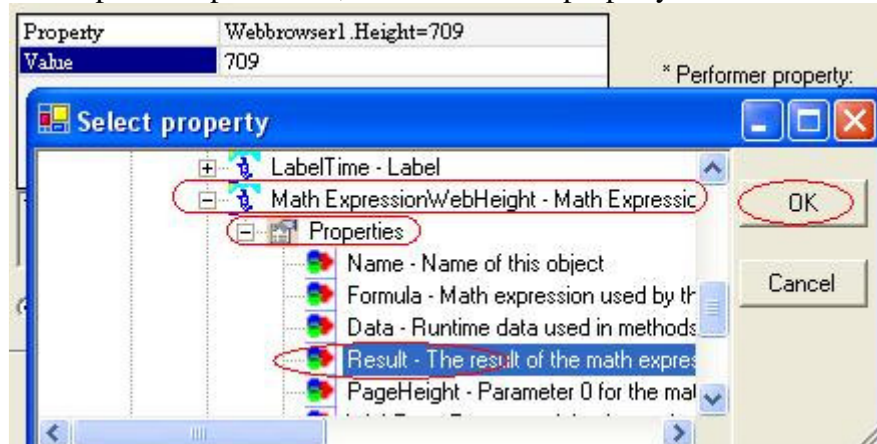
Web Browsing Limit to One Domain



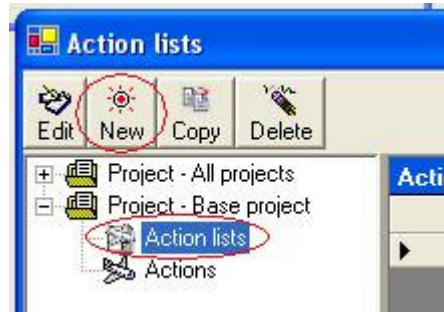
Find the page containing the Math Expression performer:



Find the Math Expression performer, select its Result property:

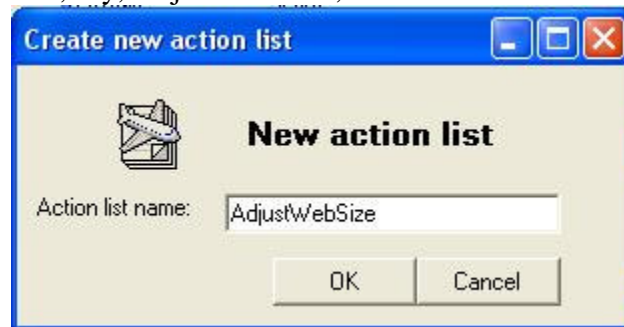


Now we have action AdjustWebHeight to adjust the height of the web performer and AdjustWebWidth to adjust the width of the web performer. We want to gather these two actions into an action list so that they always execute together. Open Actions window, select "Action Lists", click New button:



Web Browsing Limit to One Domain

Give an action list name, say, AdjustWebSize,



Click Add button to add these two actions into this list:

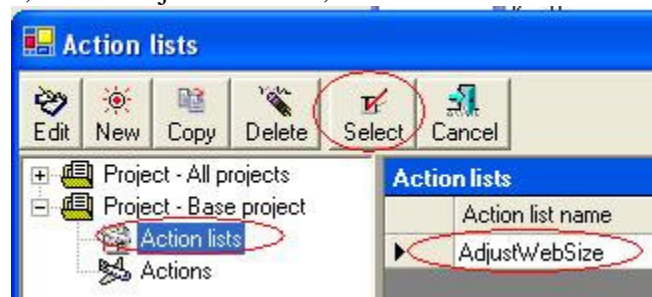


Click Finish button to finish.

Now we need to assign this action list to the Resize event of the page. Right-click on the page, choose “Assign actions”, choose “Resize”:



Select “Action lists”, select AdjustWebSize, click Select button:



Screen Saver

Web Browsing Limit to One Domain

Limnor ships with a Screen Saver performer. You may use it to add screen saver to your kiosk. You create Limnor pages to be used as screen savers. Every half an hour can use a different screen saver page. You may sell screen saver time to advertisers.

For details on how to use the Screen Saver performer, please download the sample application from <http://www.limnor.com/downloads/ScreenSaver.cab>. You may also download the document describing the making of the sample screen saver application: <http://www.limnor.com/downloads/ScreenSaver.doc>.

Deployment

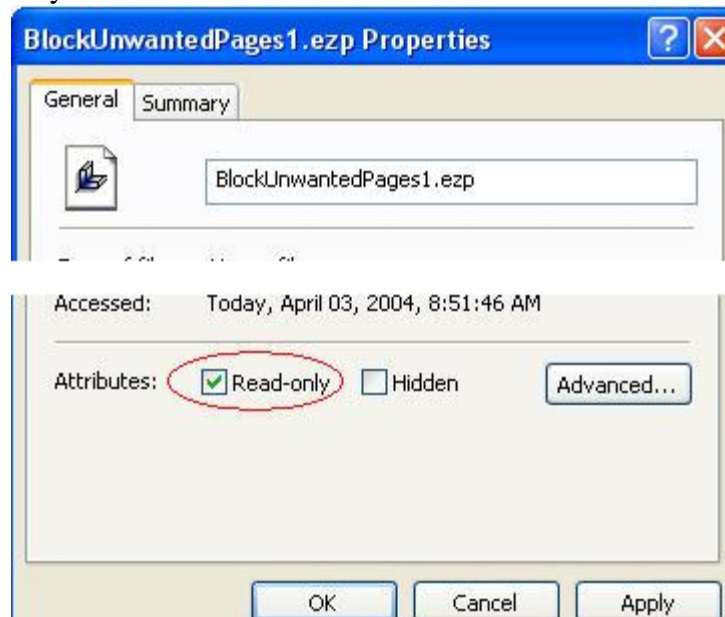
Disable Design Mode

When you press F2 key, your kiosk application goes into run-mode. Press F2 key again, it goes back to design mode. If you do not want it to go to design mode anymore, you may set your application file to read-only.

First, close Limnor. Next, use Window File Explorer to find the application file. The application file is a file with extension EZP. In this example, the application file is named BlockUnwantedPages1.ezp. Right-click on this file name, choose "Properties":



Check its Read Only attribute:



Web Browsing Limit to One Domain

Now this application will only be in run mode. Press F2 will not bring it to design mode, instead, it will bring you to runtime setting screen. You may also shutdown the application from the runtime setting screen. It will ask you for exit code. The default exit code is QUIT. You change it by setting "ExitCode" property of the Application performer.

For more information on application files, please read Lesson 1 of Limnor Tutorial.

Distribute to other computers

You may distribute your kiosk application via a CD-ROM. Limnor collects all files required for the application and automatically creates a distribution CD-ROM image for you.

Please read Lesson 8 of Limnor Tutorial.

©2003 Longflow Enterprises Ltd. All rights reserved.