

Lesson 6. Visual Programming with Event Maps

(The contents applies to Limnor version 3.3.2207.37378 and later)

You have learnt that screen design can be done visually. You have learnt that to make your application do things is to make actions and link actions to events. In this chapter, we will show you how to visually link actions and events.

This is done via windows called “Event Map”.

Every page has one event map associated with it (Page map). The Application performer also has one event map associated with it (Application map).

An event map uses icons to represent performers. It uses lines to link from event firing performers to action performing performers. Thus it shows graphically how your application flows.



At design time, you may draw lines between performers to form application flow paths.

At runtime, you may watch application flow step by step, and watch the properties of all performers.

6.1. Open Event Maps

To open the event map for the Application performer, click the event map button on the Project Explorer window:

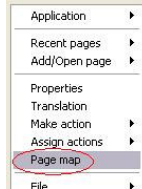


Or you may choose menu “Project | Application Map”:



We will use term “**Application Map**” to refer to the event map for the Application performer.

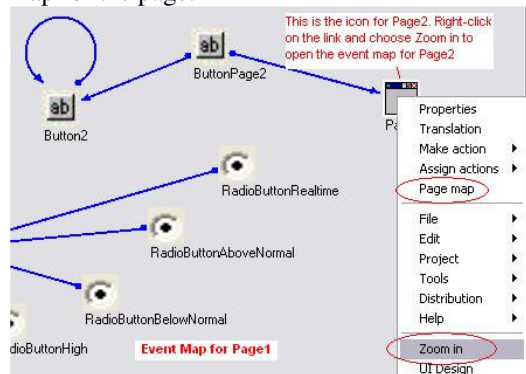
To open an event map for a page, right-click on the page, choose menu “Page map”:



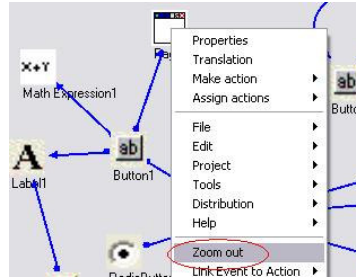
We will use term “**Page Map**” to refer to the event map for a Page performer.

6.2. Event Map Navigations




Every event map shows all pages and the Application. Right-click on the icon for a page and choose menu “Zoom in” to open the event map for the page:



Right-click on a page icon, choose “Zoom out” to show the Application Map:

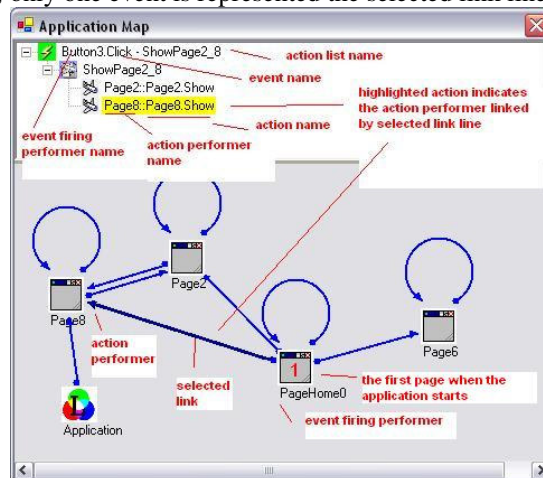


6.3. View the Application Map

This map shows the icon for the Application performer  and icons for all pages . The page icon with a number 1 on it  indicates the first page (home page) of the application.

You may drag the icons to re-position them to make the map easier to view.

You may click on each event link line to select it. The selected link line will be highlighted with thick line and darker color. The events represented by the selected link line will be displayed on the top of the event map. In the example below, only one event is represented the selected link line:

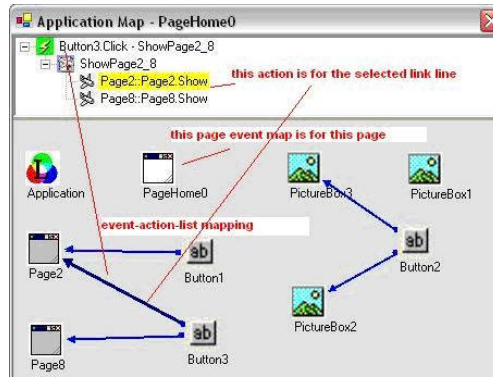


A dot at the end of the link line indicates the performer is the event firing performer. An arrow at the end of the link line indicates the performer is the action performing performer.

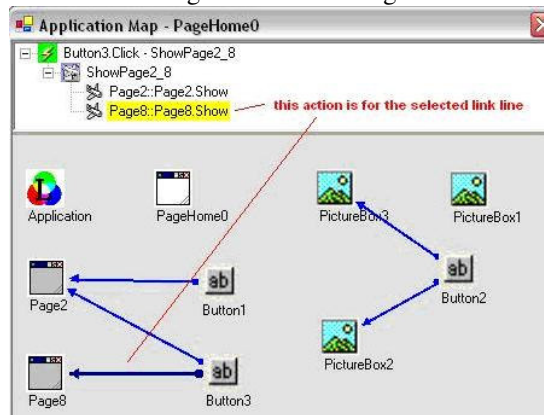
If an action list is linked to the event then all actions will be displayed on the top. Not all actions in the action list are performed by the performer linked by the selected link line. Those actions indeed performed by the performer linked by the selected link line are highlighted by yellow background.

6.4. View the Page Maps

Each page map is for one specific page. On a page map, all page icons will still be displayed but event link lines starting from the page icons will not be displayed except the page icon for the page specific for the page map. All performers for the page will be displayed as icons. Event link lines will be displayed to graphically show the event-action-list mappings. The page icon with white background color indicates the page for this page map.

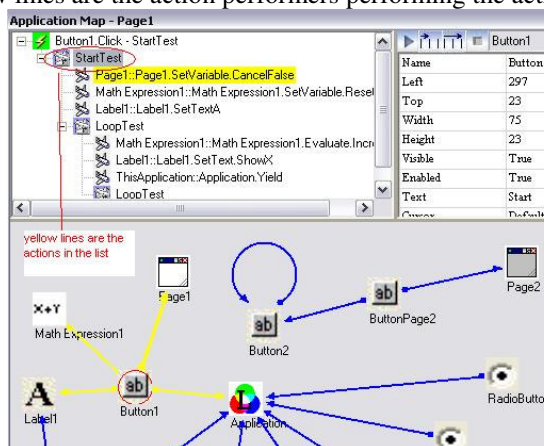


In this example, we can see that the Click event of performer Button3 triggers action list ShowPage_2_8 which has two actions; one action is Page2.Show which is highlighted because Page2 is the action performer and the selected link line is linked to Page2; the other action is Page8.Show which is not highlighted because Page8 is the action performer and the selected link line is not linked to Page8. If we select the link line linked from Button3 to Page8 then action Page8.Show will be highlighted:

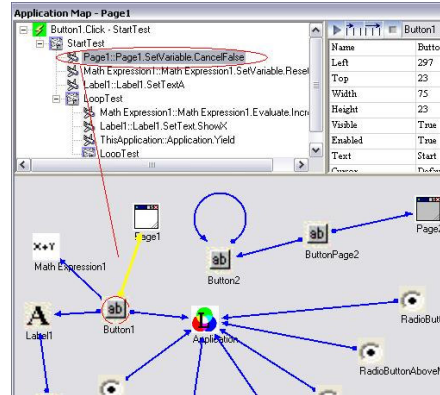


6.5. View the Actions and Linking Lines

With the actions displayed on the top of the event map, we may select an action or action list and see which lines represent the action or the action list. In the example below, the Button's Click event is linked to an action list "StartTest". When we select this action list, 4 lines are shown in yellow color. The 4 performers pointed to by these 4 yellow lines are the action performers performing the actions in this action list:



If we select the first action in the action list, we see one yellow line pointing to the icon Page1:



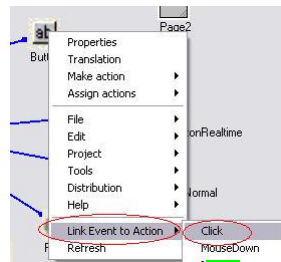
You may select different nodes to see which lines gets highlighted.



6.6. Create/Delete Linking Lines

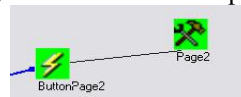
To delete a linking line, right-click on the line and choose menu “Remove Event Handler”:



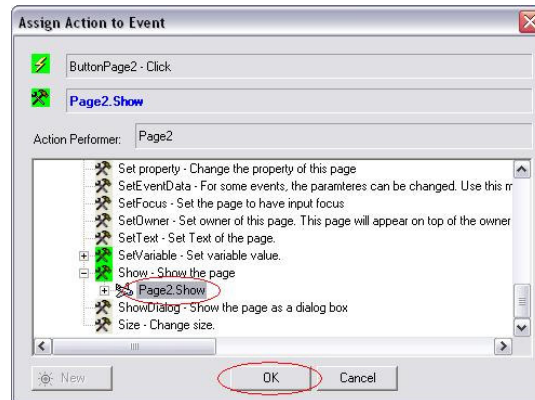
Note that the event name and the action (or action list) name are shown in the menu. One linking line may represent several events. These names help you to choose the right event handling you want to remove. To create a linking line, start from the event firing performer. Suppose when the Button named ButtonPage2 is clicked, we want to show Page2. So, ButtonPage2 is the event firing performer (firing Click event). Right-click on the icon for ButtonPage2, choose menu “Link Event to Action”, and choose event “Click”:



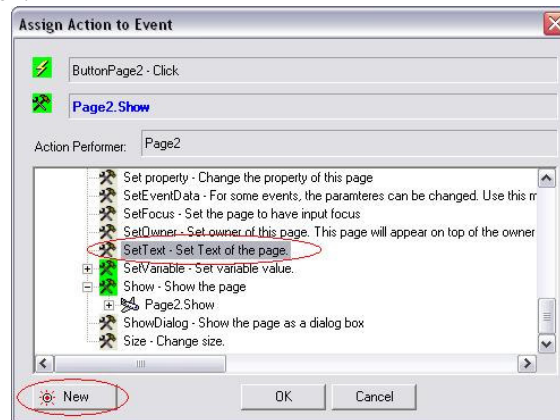
Once you select the event you will see the icon changed to  indicating that it is an event firer; and a line is drawn from the icon to the mouse pointer. When you move the mouse pointer to an icon, the icon changed to  indicating that it is an action performing performer. You may click the icon to confirm it to be the action performer. For this example, we move the mouse pointer to the icon for Page2 and click on it:



A dialogue box appears showing all methods of the action performer and the existing actions under each method. You may select an existing action and click OK:



If the action you want to link does not exist then you may select the method you want to use and click New button to create a new action:



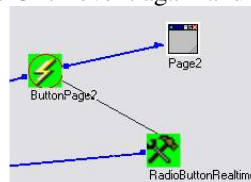
The event will be linked to the new action created.

You will see a new line is created linking the event firing performer and the action performing performer:



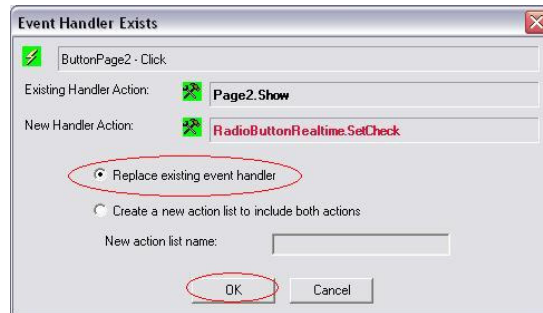
6.7. Modify Event Handler

In the last section, we made a line linking from ButtonPage2 to Page2, which links event Click to action Page2.Show. Now suppose we choose the Click event again and link it to another performer:

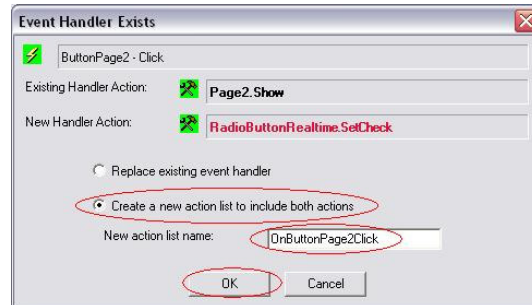


On selecting an action to be linked, a new line is supposed to be drawn from the icon of ButtonPage2 to the new action performer. Since we already have a line linking from ButtonPage2's Click event, we are facing a question of how to deal with this existing link. A dialogue box appears to give you two choices of removing or keeping the existing line.

To remove the existing linking line, choose "Replace existing event handler".

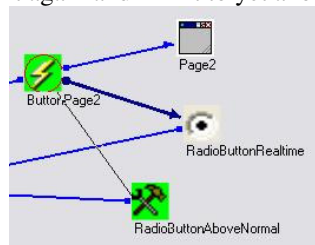


To keep the existing linking line, choose “Create a new action list to include both actions”. You need to give an action list name:



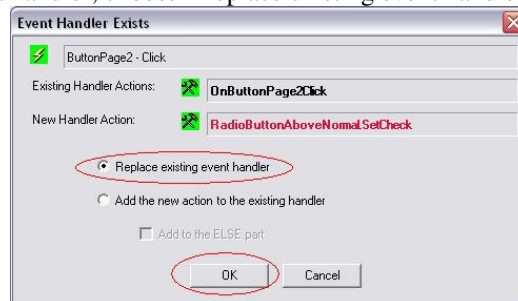
Suppose we choose to keep the existing event handler, then a new action list is created and linked to the Click event.

Now suppose we choose the Click event again and link it to yet another performer:



On selecting an action to be linked, a new line is supposed to be drawn from the icon of ButtonPage2 to the new action performer. Because there is an existing action list linked to the Click event, a dialogue box appears to give you two choices of removing or keeping the existing event handler.

To remove the existing event handler, choose “Replace existing event handler”:



The existing two lines will be removed and a new line created.

To keep the existing event handler, choose “Add the new action to the existing handler”:



The new action will be added to the end of the existing action list. An action list may have action conditions and two parts of action list: IF part and ELSE part. When the action conditions evaluated to True, the IF part is executed. When the action conditions evaluated to False, the ELSE part is executed. Check the option “Add to the ELSE part” if the new action is to be executed when the action conditions is evaluated to False.

The new action is always added to the end of existing action list. To change the action orders, see below.

6.8. Modify Action Lists

When an action list is displayed as an event handler, you may modify it by changing action orders and remove actions from the list.

Right-click on the action to be changed, a popup menu appears:



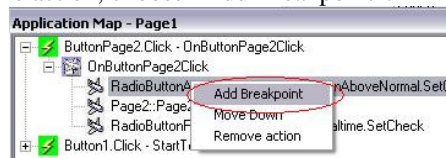
To change the position of the action, choose “Move Up” or “Move Down”. To remove the action from the action list, choose “Remove action”.

6.9. Debug with Event Map

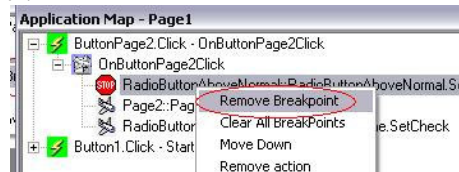
6.9.1. Set Breakpoints

You may specify that when certain events trigger some actions, the executions of those actions are suspended so that you may inspect the performer properties and see that the properties are what you expected. Such suspension points are called breakpoints.

To set a breakpoint, right-click the action, choose “Add Breakpoint”:









A stop sign appears at the action indicating that it is a breakpoint. To remove a breakpoint, right-click on it and choose “Remove Breakpoint”:



To remove all breakpoints, choose menu “Clear All Breakpoints”.

6.9.2. Start Debugging

You may watch your application running from event maps. To do that, you need to run your application from an event map. You may start your application in 3 ways: run ; step into ; and step over 

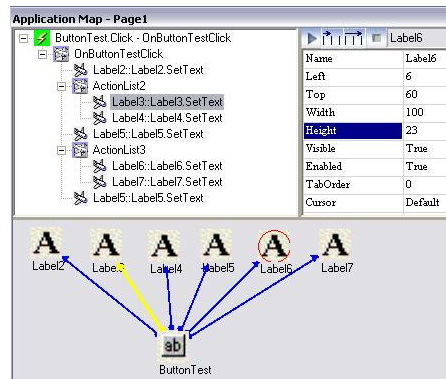
Click on , the application goes into runtime. You need to set some breakpoints in some event handlers. When an event triggers its handler and hits a breakpoint, the event handler is displayed in the event map. From the breakpoint, you may run the actions one by one. Click on  or , the application goes into runtime. When any event triggers any action, it will stop there and display the event handler, as if that action is a breakpoint. You can start debugging from there.


6.9.3. Step by Step Execution

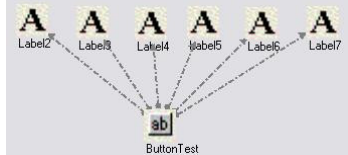
To demonstrate the debug process, we created 6 Label performers and created an action for each of them. We use these actions to create action lists. One action list named OnButtonTestClick is assigned to the Click event of a button. This action list includes two other action lists. The page looks like:




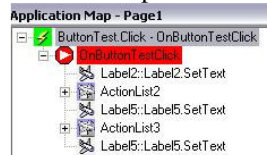
The page map looks like:




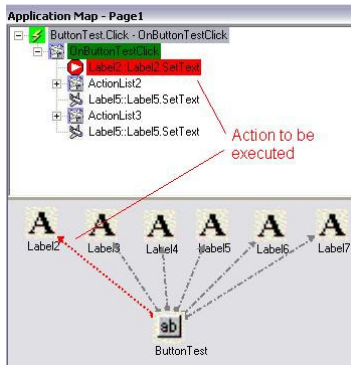
Now click . We can see the application goes into runtime. Once in the runtime, the linking lines in the event maps become grey and dashed, when the actions are not executed.



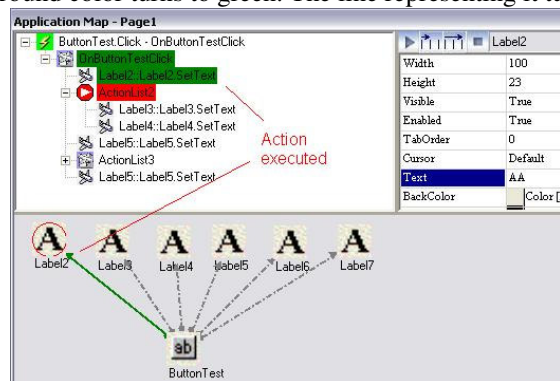
Now click the button. It will trigger the action list OnButtonTestClick. Because we used  to start the debugging, the execution of this action list will be suspended and it will be displayed in the event map:



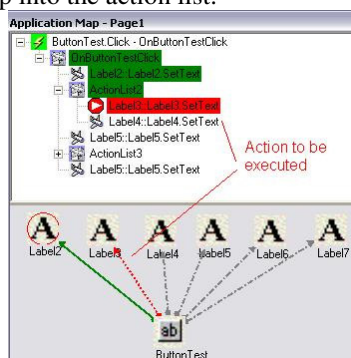
The red background marks the current execution point. Now click  to advance one step forward. The execution point landed on the first action in the list, Label2.SetText. The line representing the action also changes to a red dotted line:



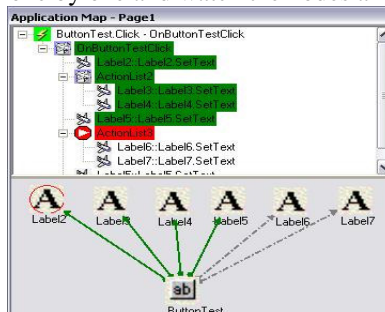
Click to advance one step forward again. The action Label2.SetText will be executed. Once it is executed, the node's background color turns to green. The line representing it turns to solid green.




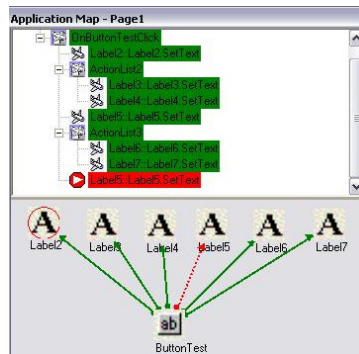
You may click on icons to show performer properties. For example, we click on the icon for Label2, we can see the Text property of Label2 is “AA”. That is the result of executing action Label2.SetText. Now the action to be executed is an action list, ActionList2. Clicking or will make a difference. Clicking will stop at the first action of ActionList2, that is Label3.SetText. Clicking will execute all actions in ActionList2 and stop at the next action following ActionList2, that is Label5.SetText. If you have a long action list and you do not want to step through each action in the list, you may click . For this action list, we click to step into the action list:





Keep clicking to execute actions one by one and watch the nodes and lines turn to green:



When it reaches the action list ActionList3, we may click  to see all actions in the action list are executed:

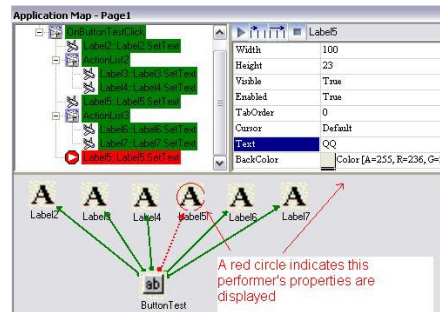


At any time of debugging, if you no longer want to step through all the actions, you may click  to execute all actions without stopping. It will stop when it hits a breakpoint.

At anytime, you may click  to stop the debugging and return to design time.

6.10. View Performer Properties

Click an icon to show the properties of the performer. There is a red circle on the icon indicating that its properties are displayed.



If a performer has not been loaded into memory then clicking its icon will not display its properties.

Suppose we have a Page2 performer. When Page2 is not opened, clicking its icon will show its properties:

