Use Web Event Bubbling

Contents

Introduction

Every element in a web page has a parent element. For example, a button's parent element can be a group box, a paragraph, a division, etc. An element can be a parent of many other elements, but it may only have one immediate-level parent element. The top level parent is the web page itself.

Keyboard and mouse events occur in an element can be caught by all levels of parent elements of the event-generating element.

Suppose Button1's parent element is GruopBox1; GroupBox1's parent is GroupBox2.
Suppose we assign an action, Action1, to the onclick event of GroupBox2. Clicking GroupBox2, the action will be executed, as we expected. But clicking Button1 or GroupBox1, Action1 will also be executed because the event will bubble up and reach GroupBox2.
Suppose we assign an action, Action2, to the onclick event of GroupBox1; another action, Action3, to the onclick event of Button1. Click Button1, Action3 is executed first, and then execute Action2, and then Action1. Click GroupBox1, Action2 is executed and then Action1 is executed. Click GroupBox2, only Action1 is executed.
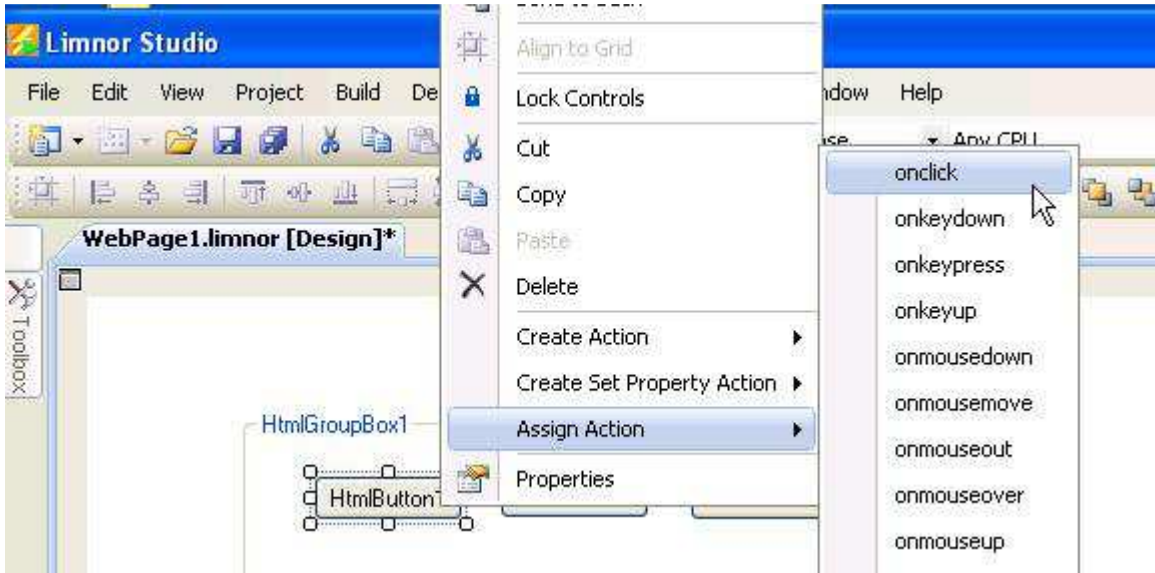
Such behavior is called event bubbling. Below we will use examples to show how to make use of event bubbling.
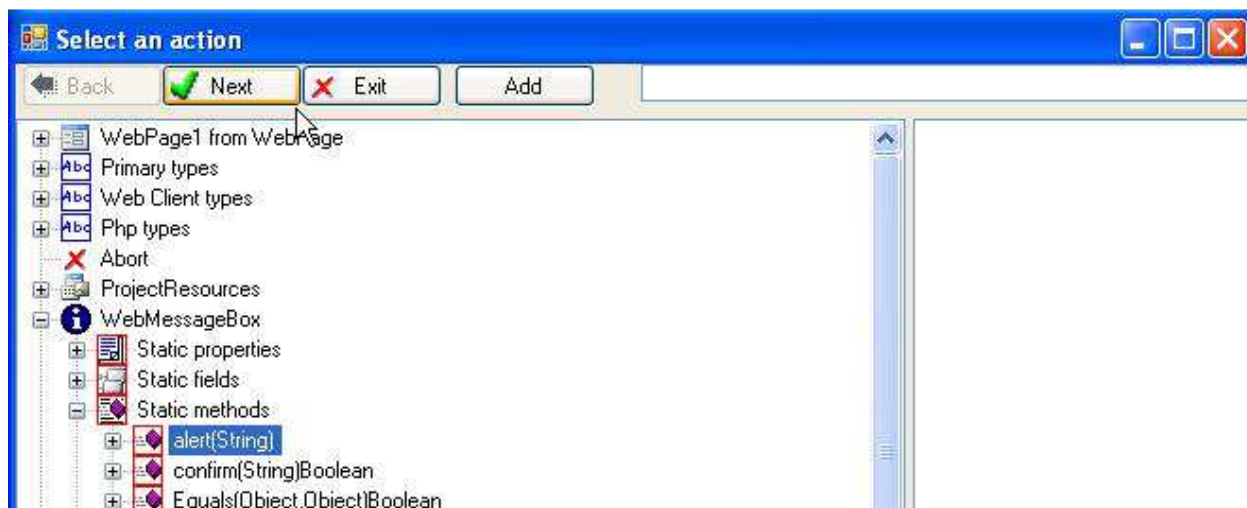
Use event bubbling

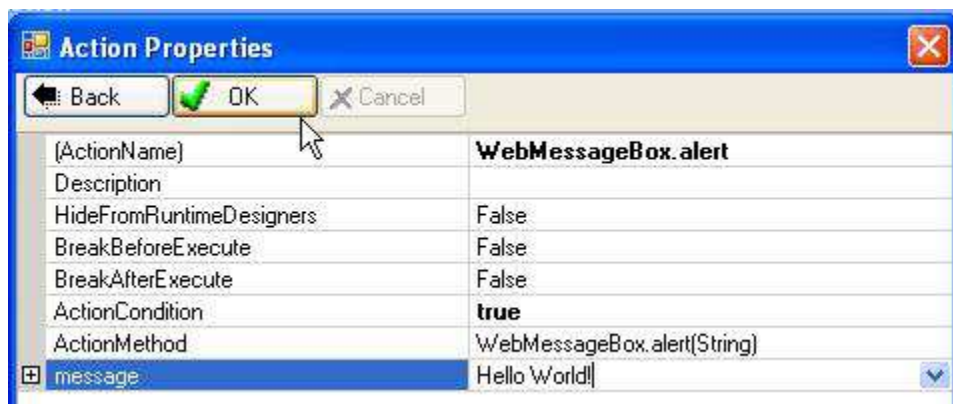Suppose we create a web page to have a group box containing a few buttons.
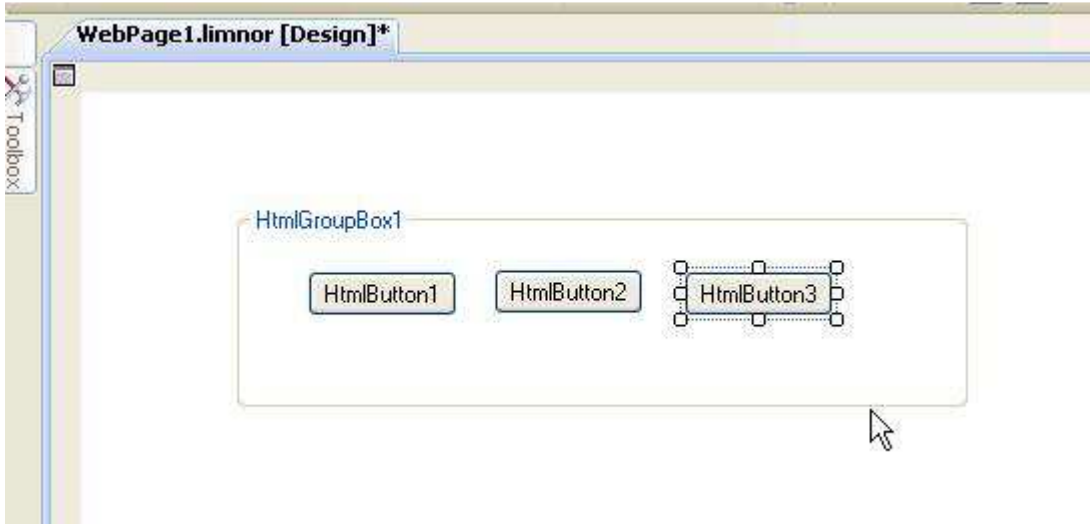
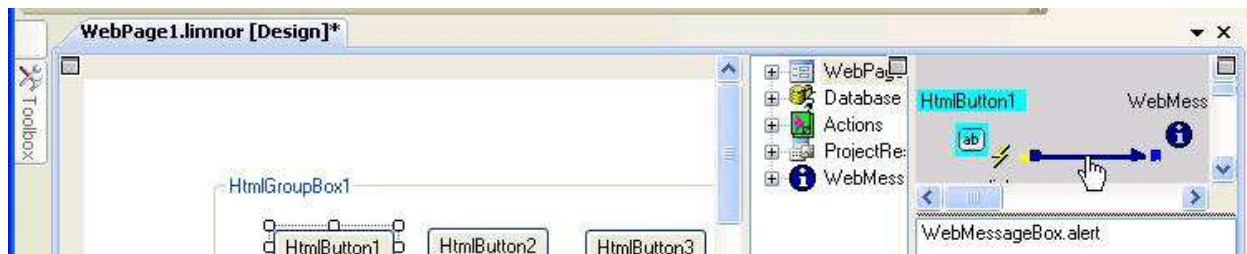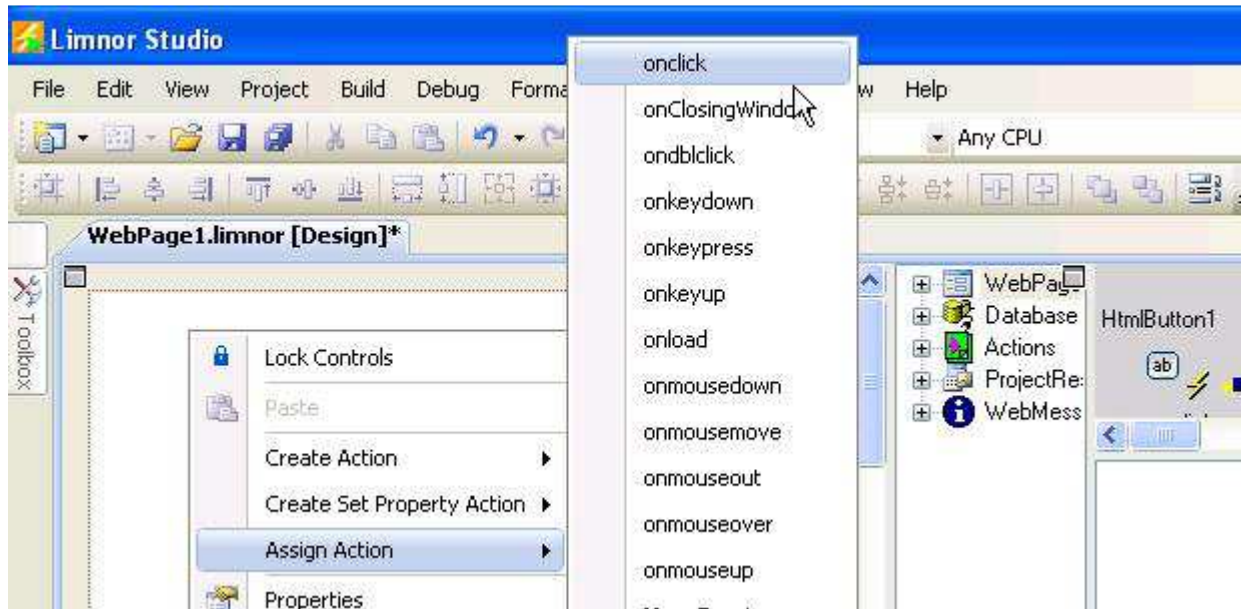Let's assign a message box action to HtmlButton1:

Give a message and click OK:

The action is created and assigned to the onclick event of HtmlButton1:


Let's assign another action to the onclick event of the web page:

Suppose we want to show a message box to display which element generated the onclick event. We need to create an event handler method so that we may use the information contained in the event parameters. In this sample, we select "New client handler actions for onclick":
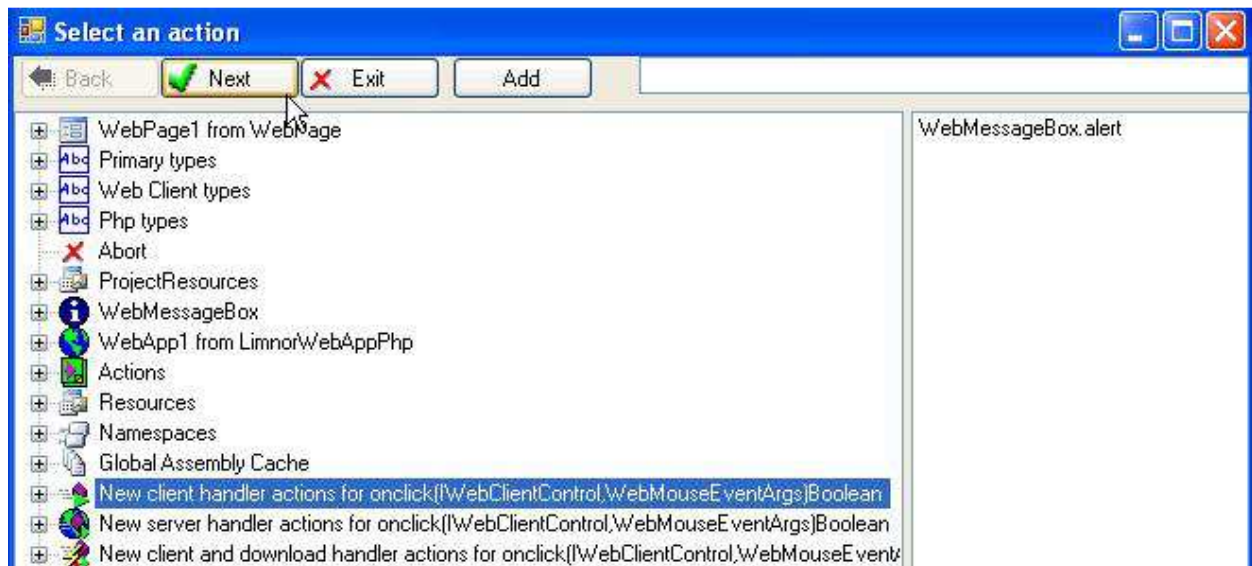
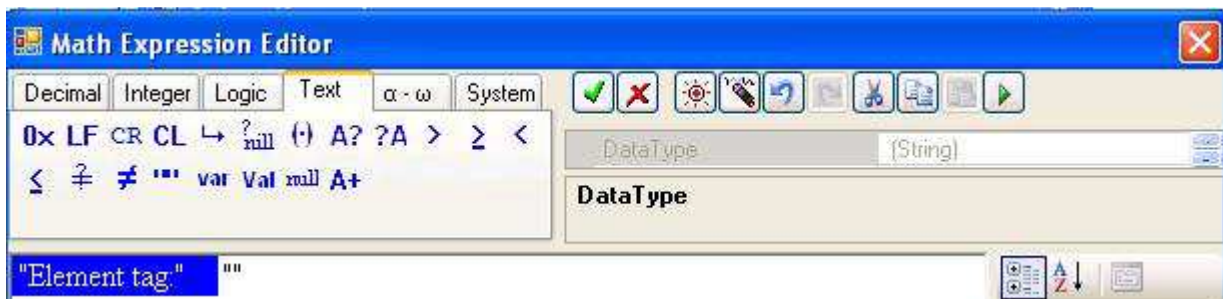The Method Editor appears. Add a message box action to it:
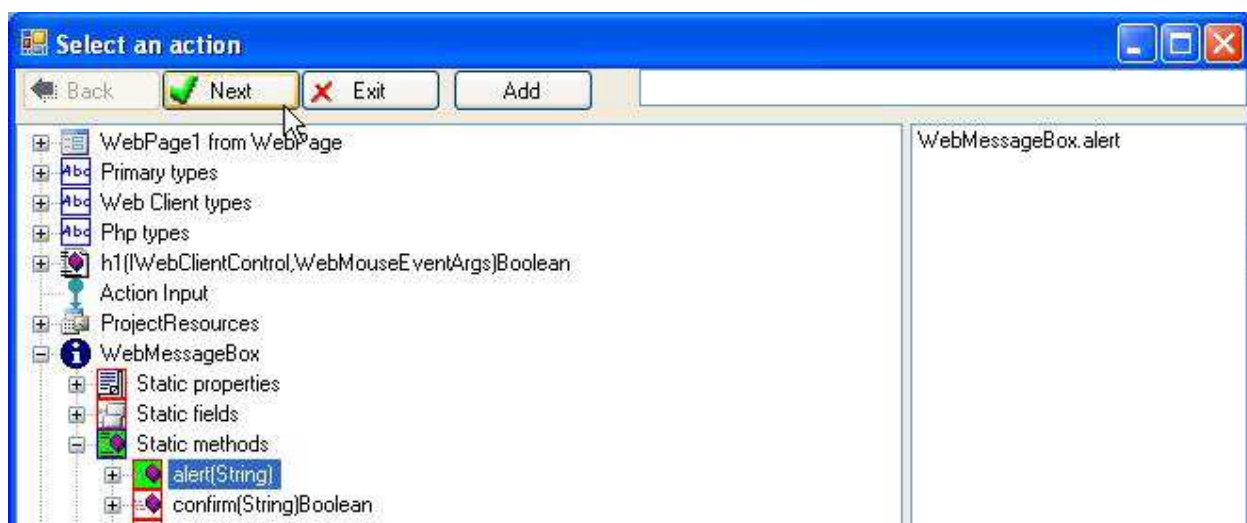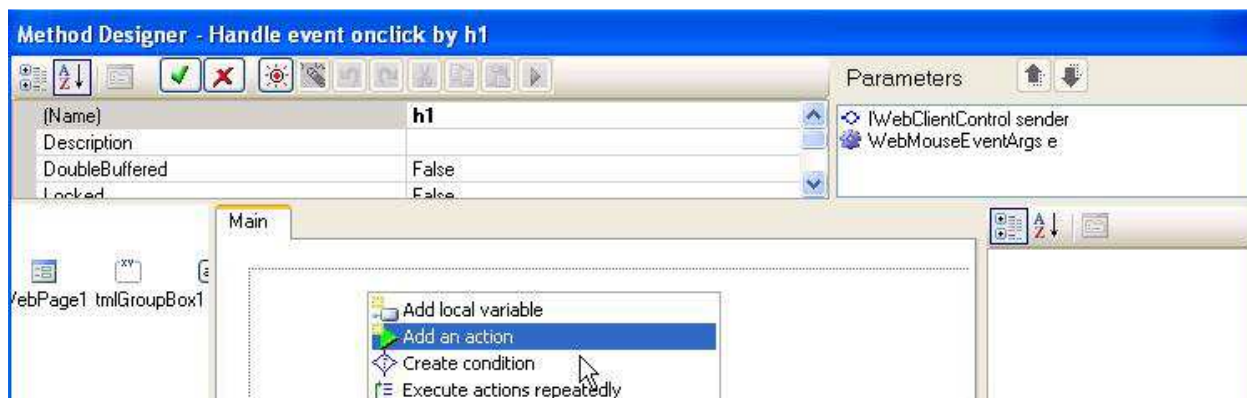
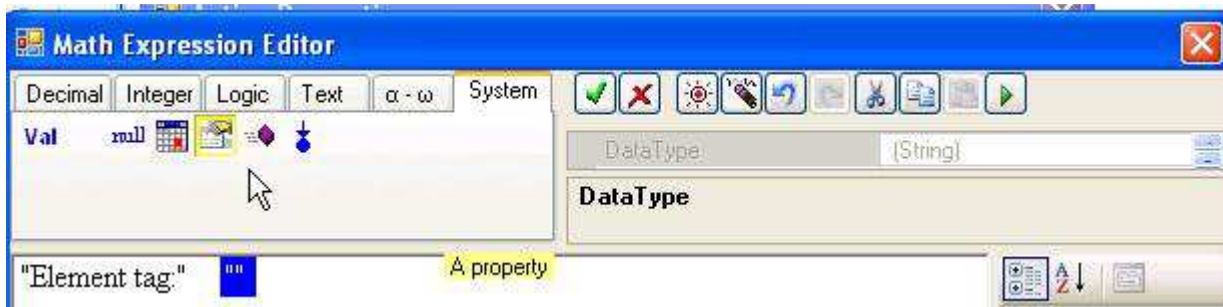Select "Math Expression" for the "message" to form a string expression:
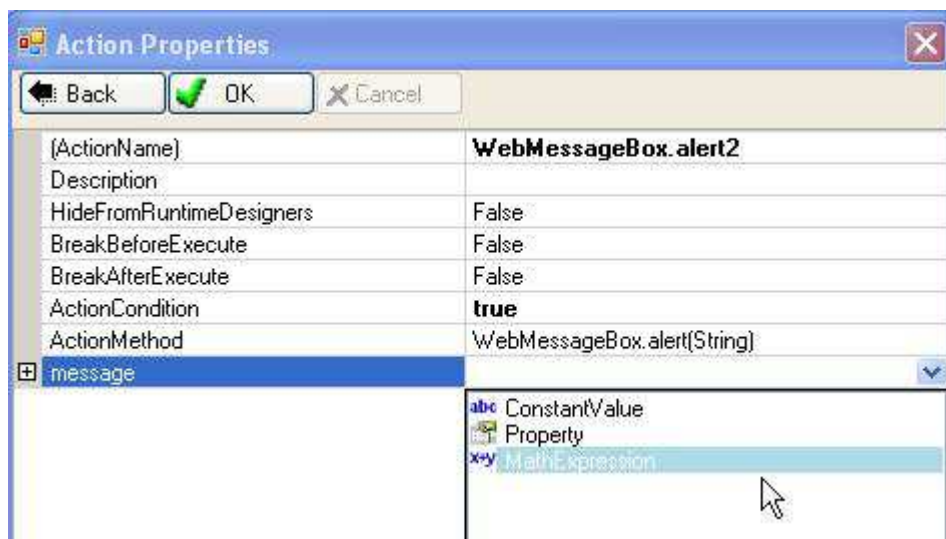
We want to display the tab name and id of the "sender" parameter. "sender" parameter is the html element generating the onclick event. Click A+ to form string concatenation:
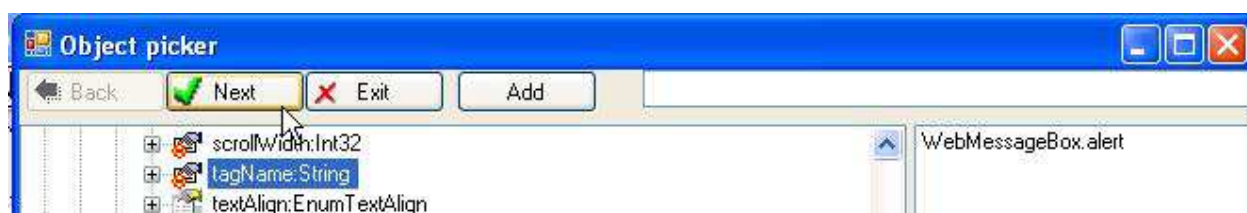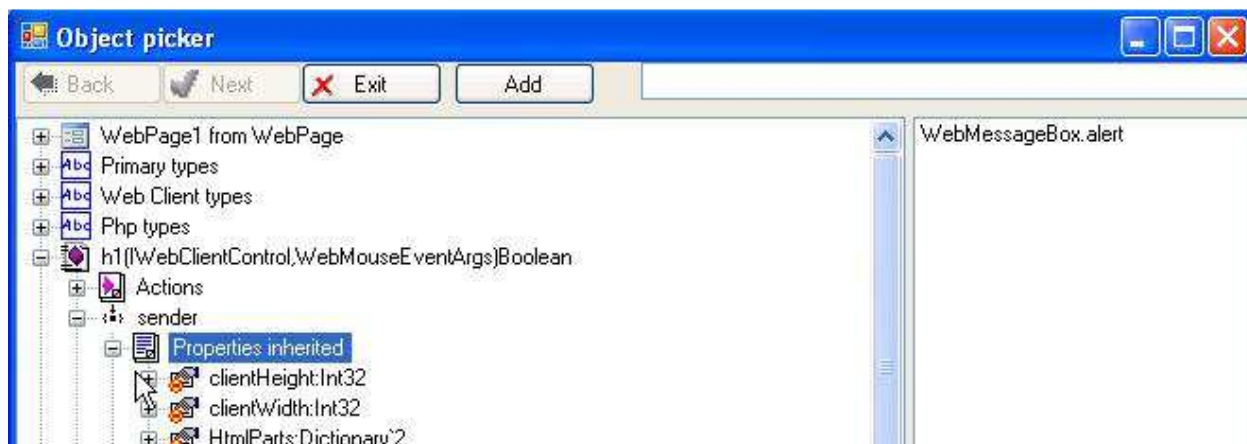
Select the first item and type "Element tag:"

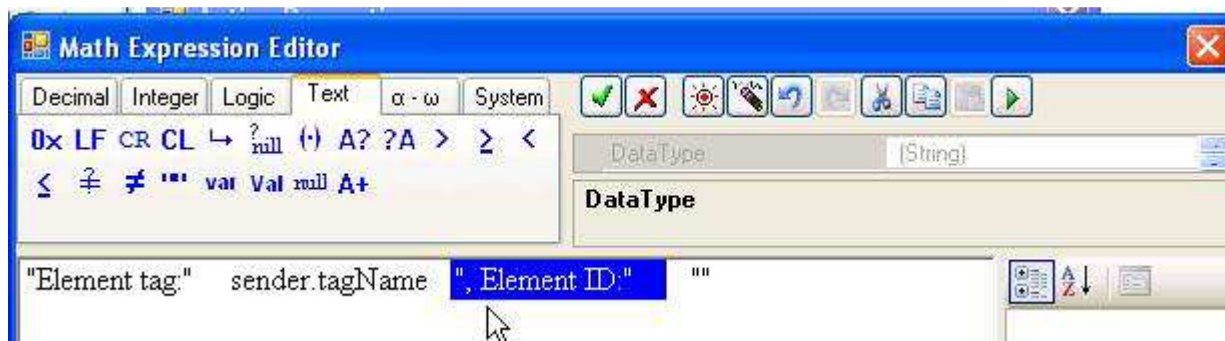Select the second item and click the Property icon:

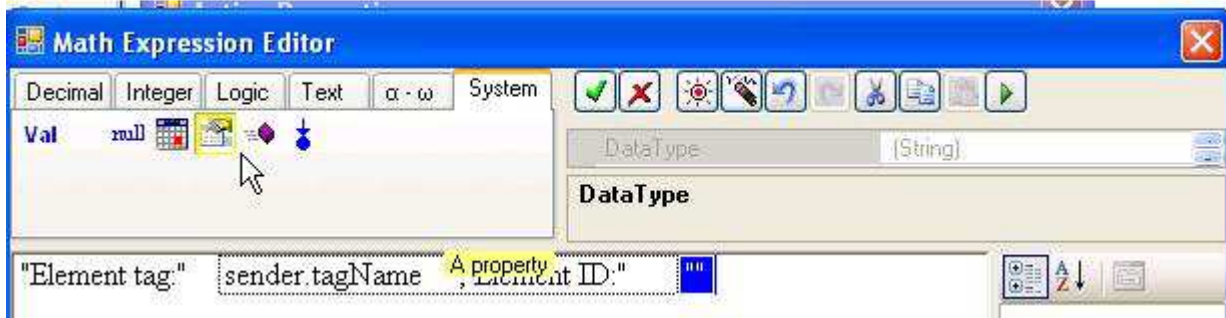Select the "tagName" property of the "sender":
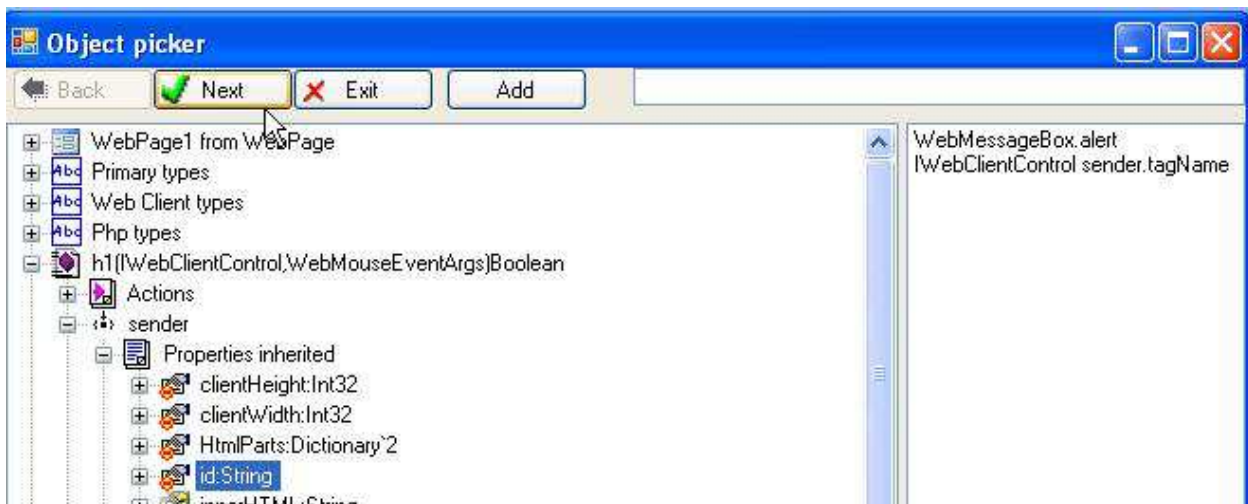
Click A+ twice to add 2 more string items:

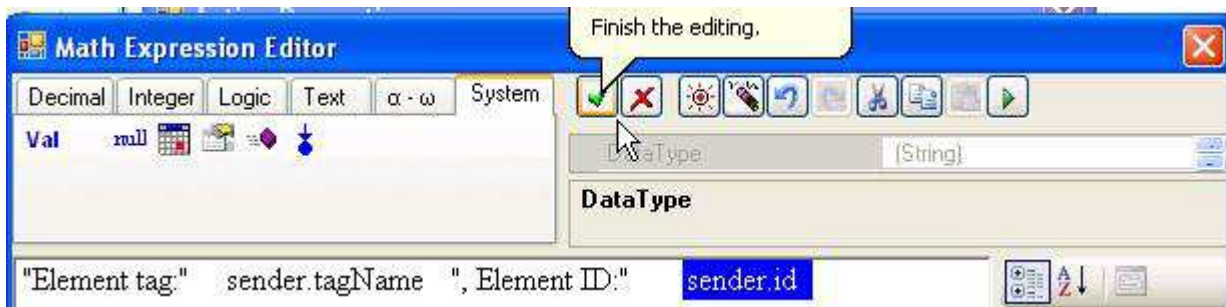Select the third item and type ", Element ID:":

Select the last item and click the Property icon:
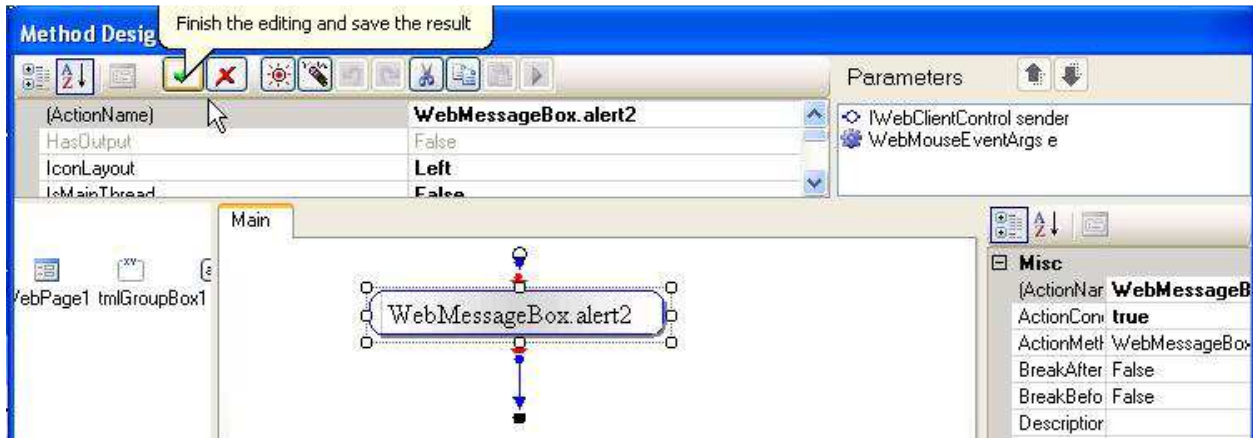
Select property "id" of the "sender":

This is the information we want to display:

Click OK. The action is created and appears in the Method Editor. For this sample, that is all for this event handler method:
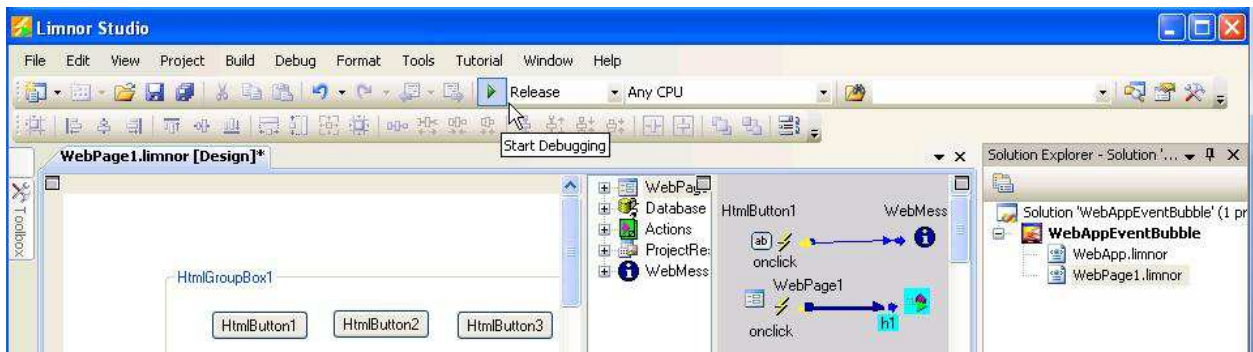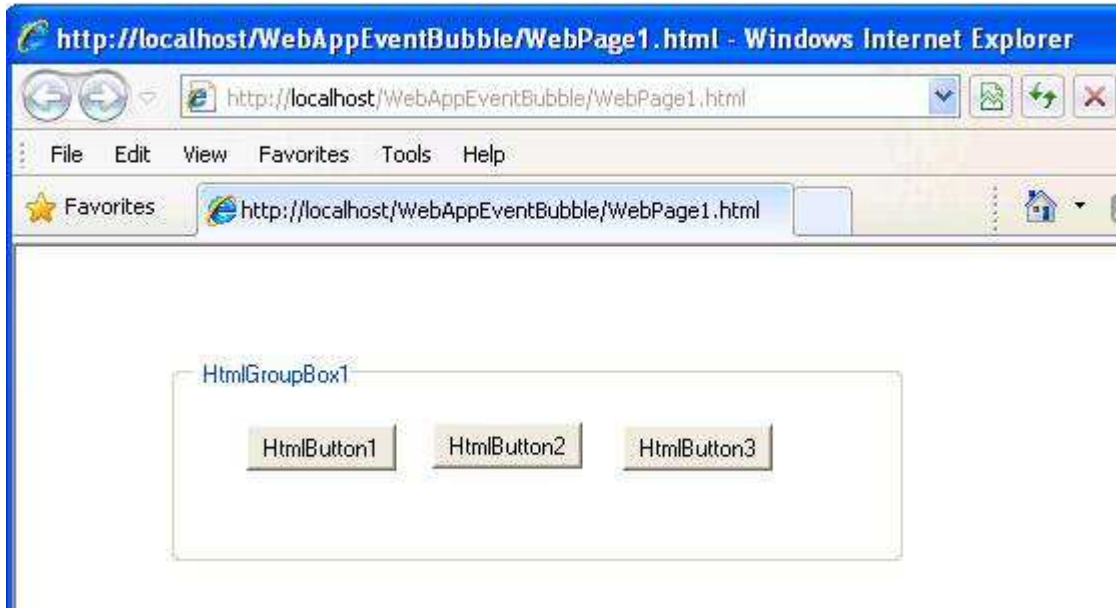
In your real web applications, you may use condition actions using tagName or id to form action execution paths.

Test event bubbling

We may start the web application to test how the event handling is done.

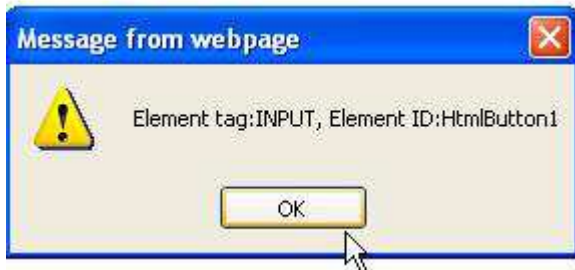The web page appears in the web browser:

Click HtmlGroupBox1; the following message appears as the result of handling web page click:

Click HtmlButton1; the following message appears as the result of handling of the button click:

Click OK, another message appears as the result of handling web page click:

We saw that in this case one click of HtmlButton1 triggers both onclick event of HtmlButton1 and Web Page.

Click another button, HtmlButton2, the following message appears as the result of handling web page click:
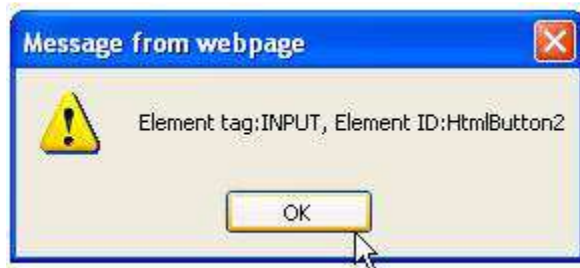
Feedbacks

Please send your feedbacks to support@limnor.com. Thanks!