.NET Framework Developer's Guide

**Generating Client-Side Script for Postback**

Only two HTML form elements (button and image button) cause form postback. If your custom control renders an HTML element that does not cause postback (such as a text box or a link button) and you want the control to initiate postback, you can program this in ASP.NET through an event architecture that relies on client-side script (JScript, JavaScript).

Control developers need to make a few small changes to their controls to enable this postback mechanism. The following code fragment shows (in bold) the code that must be added to the **Render** method of a control to initiate postback.

[C#]

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl01_code');" \o "Copy Code)

protected override void Render(HtmlTextWriter output) {

output.Write("<a id=\"" + this.UniqueID + "\" href=\"javascript:" + Page.GetPostBackEventReference(this) +"\">");

output.Write(" " + this.UniqueID + "</a>");

}

[Visual Basic]

Protected Overrides Sub Render(output As HtmlTextWriter)

output.Write("<a id=""" & Me.UniqueID & \_

""" href=""javascript:" & \_

Page.GetPostBackEventReference(Me) & """>")

output.Write(" " & Me.UniqueID & "</a>")

End Sub

The **GetPostBackEventReference** method emits client-side script that initiates postback and also provides a reference to the control that initiated the postback event.

To understand what the **GetPostBackEventReference** method does, compile the code in the next section, [Postback Using Client-Side Script Sample](http://msdn.microsoft.com/en-us/library/aa720417(VS.71).aspx), and request the accompanying .aspx page in your browser. If you view the source for the rendered page using your browser's View Source command, you will see the following HTML and script. (The actual values of the name and the ID attributes of the form will differ, and so will the value attribute of the hidden variable.)

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl03_code');" \o "Copy Code)

<html>

<body>

<form name="ctrl2" method="POST" action="MyLinkButton.aspx" id="ctrl2">

<input type="hidden" name="\_\_EVENTTARGET" value="" />

<input type="hidden" name="\_\_EVENTARGUMENT" value="" />

<input type="hidden" name="\_\_VIEWSTATE" value="dDwtMjI1NTgwNDE2Ozs+eIZ+AfRvoCj1nWimbh+zPv/rKNg=" />

Here is the custom link button.<br>

<a id ="Link" href="javascript:\_\_doPostBack('Link','')"> Link</a>

<br><br>

<input name="TextBox" type="text" value="Click the link" id="TextBox" style="background-color:Cyan;width:200px;" />

<br>

<script language="javascript">

<!--

function \_\_doPostBack(eventTarget, eventArgument) {

var theform = document.ctrl2

theform.\_\_EVENTTARGET.value = eventTarget

theform.\_\_EVENTARGUMENT.value = eventArgument

theform.submit()

}

// -->

</script>

</form>

</body>

</html>

The elements in bold in this example — two hidden fields and the client-side script method — are added by the ASP.NET page framework. The hidden form fields indicate which server control to post to and optionally specify an argument to pass. The client-side script method is used to set the hidden fields and causes the form to be submitted to the server. For a complete sample, see [Postback Using Client-Side Script Sample](http://msdn.microsoft.com/en-us/library/aa720417(VS.71).aspx).