.NET Framework Developer's Guide

**Bubbling an Event**

The ASP.NET page framework provides a technique called event bubbling that allows a child control to propagate events up its containment hierarchy. Event bubbling enables events to be raised from a more convenient location in the controls hierarchy and allows event handlers to be attached to the original control as well as to the control that exposes the bubbled event.

Event bubbling is used by the data-bound controls ([Repeater](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.repeater(VS.71).aspx), [DataList](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist(VS.71).aspx), and [DataGrid](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datagrid(VS.71).aspx)) to expose command events raised by child controls (within item templates) as top-level events. While ASP.NET server controls in the .NET Framework use event bubbling for command events (events whose event data class derives from **CommandEventArgs**), any event defined on a server control can be bubbled.

A control can participate in event bubbling through two methods that it inherits from the base class **System.Web.UI.Control**. These methods are **OnBubbleEvent** and **RaiseBubbleEvent**. The following code shows the signatures of these methods.

[C#]

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl04C#');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl04C#');)

protected virtual bool OnBubbleEvent(

object source,

EventArgs args

);

protected void RaiseBubbleEvent(

object source,

EventArgs args

);

[Visual Basic]

Overridable Protected Function OnBubbleEvent( \_

ByVal source As Object, \_

ByVal args As EventArgs \_

) As Boolean

Protected Sub RaiseBubbleEvent( \_

ByVal source As Object, \_

ByVal args As EventArgs \_

)

The implementation of **RaiseBubbleEvent** is provided by **Control** and cannot be overridden. **RaiseBubbleEvent** sends the event data up the hierarchy to the control's parent. To handle or to raise the bubbled event, a control must override the **OnBubbleEvent** method.

A control that has an event bubbled to it does one of the following three things.

* It does nothing, in which case the event is automatically bubbled up to its parent.
* It does some processing and continues to bubble the event. To accomplish this, a control must override **OnBubbleEvent** and invoke **RaiseBubbleEvent** from **OnBubbleEvent**. The following code fragment (from the [Templated Data-Bound Control Sample](http://msdn.microsoft.com/en-us/library/aa720696(VS.71).aspx)) bubbles an event after checking for the type of the event arguments.

[C#]

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl06C#');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl06C#');)

protected override bool OnBubbleEvent(object source, EventArgs e) {

if (e is CommandEventArgs) {

// Adds information about an Item to the

// CommandEvent.

TemplatedListCommandEventArgs args =

new TemplatedListCommandEventArgs(this, source, (CommandEventArgs)e);

RaiseBubbleEvent(this, args);

return true;

}

return false;

}

[Visual Basic]

Protected Overrides Function OnBubbleEvent(source As Object, e As EventArgs) As Boolean

If TypeOf e Is CommandEventArgs Then

' Adds information about an Item to the

' CommandEvent.

Dim args As New TemplatedListCommandEventArgs(Me, source, CType(e, CommandEventArgs))

RaiseBubbleEvent(Me, args)

Return True

End If

Return False

End Function

* It stops bubbling the event and raises and/or handles the event. Raising an event involves invoking the method that dispatches the event to listeners. To raise the bubbled event, a control must override **OnBubbleEvent** to invoke the **On***EventName* method that raises the bubbled event. A control that raises a bubbled event generally exposes the bubbled event as a top-level event. The following code fragment (from the [Templated Data-Bound Control Sample](http://msdn.microsoft.com/en-us/library/aa720696(VS.71).aspx)) raises a bubbled event.

[C#]

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl08C#');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl08C#');)

protected override bool OnBubbleEvent(object source, EventArgs e) {

bool handled = false;

if (e is TemplatedListCommandEventArgs) {

TemplatedListCommandEventArgs ce = (TemplatedListCommandEventArgs)e;

OnItemCommand(ce);

handled = true;

}

return handled;

}

[Visual Basic]

Protected Overrides Function OnBubbleEvent(source As Object, e As EventArgs) As Boolean

Dim handled As Boolean = False

If TypeOf e Is TemplatedListCommandEventArgs Then

Dim ce As TemplatedListCommandEventArgs = CType(e, TemplatedListCommandEventArgs)

OnItemCommand(ce)

handled = True

End If

Return handled

End Function

For samples that demonstrate event bubbling, see [Event Bubbling Control Sample](http://msdn.microsoft.com/en-us/library/aa720044(VS.71).aspx) and [Templated Data-Bound Control Sample](http://msdn.microsoft.com/en-us/library/aa720696(VS.71).aspx).

**Note**   While the method that enables event bubbling, **OnBubbleEvent**, follows the standard .NET Framework naming pattern for methods that raise events, there is no event named BubbleEvent. The bubbled event is exposed as a top-level event in the control that stops event bubbling. For example, the **DataList** control exposes **Command** events from controls in its template as **ItemCommand** events. Note also that the standard signature of **On***EventName* methods in the .NET Framework has one argument (protected void OnEventName (EventArgs e)). However, **OnBubbleEvent** has two arguments because the event originates outside the control; the second argument supplies the source.

The discussion so far shows how a control can respond to an event that is bubbled up to it. The following section shows how to author a control that defines a bubbled event.

**Defining a Bubbled Event**

If you want your control to enable event bubbling for an event that it defines, it must invoke the **RaiseBubbleEvent** from the **On***EventName* method that raises the event. No additional work needs to be done from within the control. The following code fragment shows a control that defines a **Command** event that enables bubbling.

[C#]

[[http://i.msdn.microsoft.com/Global/Images/clear.gif](javascript:CopyCode('ctl00_mainContentContainer_ctl11C#');)Copy Code](javascript:CopyCode('ctl00_mainContentContainer_ctl11C#');)

protected virtual void OnCommand(CommandEventArgs e) {

CommandEventHandler handler = (CommandEventHandler)Events[EventCommand];

if (handler != null)

handler(this,e);

// The Command event is bubbled up the control hierarchy.

RaiseBubbleEvent(this, e);

}

[Visual Basic]

Protected Overridable Sub OnCommand(e As CommandEventArgs)

Dim handler As CommandEventHandler = CType(Events(EventCommand), CommandEventHandler)

If Not (handler Is Nothing) Then

handler(Me, e)

End If

' The Command event is bubbled up the control hierarchy.

RaiseBubbleEvent(Me, e)

End Sub

**Note**   Event bubbling is not limited to command events. You can use the mechanism described here to bubble any event.