# Bubbling an Event

**.NET Framework 1.1**

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(v=vs.71).aspx), [DataList](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist(v=vs.71).aspx), and [DataGrid](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datagrid(v=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#

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(v=vs.71).aspx)) bubbles an event after checking for the type of the event arguments.

C#

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(v=vs.71).aspx)) raises a bubbled event.

C#

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(v=vs.71).aspx) and [Templated Data-Bound Control Sample](http://msdn.microsoft.com/en-us/library/aa720696(v=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#

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.

**Control.RaiseBubbleEvent Method**

**.NET Framework 4.5**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [.NET Framework 4](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.100).aspx)
* [.NET Framework 3.5](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.90).aspx)
* [.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.85).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.80).aspx)
* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.71).aspx)

Assigns any sources of the event and its information to the control's parent.

**Namespace:**  [System.Web.UI](http://msdn.microsoft.com/en-us/library/system.web.ui(v=vs.110).aspx)  
**Assembly:**  System.Web (in System.Web.dll)

[Syntax](javascript:void(0))

C#

[C++](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=cpp#code-snippet-1)

[F#](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=fsharp#code-snippet-1)

[VB](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

protected void RaiseBubbleEvent(

Object source,

EventArgs args

)

**Parameters**

source

Type: [System.Object](http://msdn.microsoft.com/en-us/library/system.object(v=vs.110).aspx)

The source of the event.

args

Type: [System.EventArgs](http://msdn.microsoft.com/en-us/library/system.eventargs(v=vs.110).aspx)

An [EventArgs](http://msdn.microsoft.com/en-us/library/system.eventargs(v=vs.110).aspx) object that contains the event data.

[Remarks](javascript:void(0))

ASP.NET server controls such as the [Repeater](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.repeater(v=vs.110).aspx), [DataList](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist(v=vs.110).aspx) and [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx) Web controls can contain child controls that raise events. For example, each row in a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx) control can contain one or more buttons created dynamically by templates. Rather than each button raising an event individually, events from the nested controls are "bubbled"—that is, they are sent to the control's parent. The parent in turn raises a generic event called [RowCommand](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.rowcommand(v=vs.110).aspx) with parameter values. These values allow you to determine which individual control that raised the original event. By responding to this single event, you can avoid having to write individual event-handling methods for child controls.

While you cannot override this method, controls you author can handle or raise bubbled events by overriding the [OnBubbleEvent](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(v=vs.110).aspx) method.

[Examples](javascript:void(0))

The following code example demonstrate how to create a custom class, ChildControl, overriding the [Button.OnClick](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.onclick(v=vs.110).aspx) method to call the RaiseBubbleEvent method that sends the [Button.Click](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.click(v=vs.110).aspx) event to its parent ASP.NET server control. When the user clicks a button in an ASP.NET page that includes an instance of ChildControl, it raises the [OnBubbleEvent](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(v=vs.110).aspx) method on the parent control that contains the instance of ChildControl and writes the string "The ChildControl class OnClick method is called" to the page.

C#

[VB](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

public class ChildControl : Button

{

[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Demand, Name="FullTrust")]

protected override void OnClick(EventArgs e)

{

base.OnClick(e);

Context.Response.Write("<br><br>ChildControl's OnClick called.");

// Bubble this event to parent.

RaiseBubbleEvent(this, e);

}

}

**Control.OnBubbleEvent Method**

**.NET Framework 4.5**

[Other Versions](javascript:;)

Description: http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [.NET Framework 4](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.100).aspx)
* [.NET Framework 3.5](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.90).aspx)
* [.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.85).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.80).aspx)
* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.71).aspx)

Determines whether the event for the server control is passed up the page's UI server control hierarchy.

**Namespace:**  [System.Web.UI](http://msdn.microsoft.com/en-us/library/system.web.ui(v=vs.110).aspx)  
**Assembly:**  System.Web (in System.Web.dll)

[Syntax](javascript:void(0))

C#

[C++](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=cpp#code-snippet-1)

[F#](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=fsharp#code-snippet-1)

[VB](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

protected virtual bool OnBubbleEvent(

Object source,

EventArgs args

)

**Parameters**

source

Type: [System.Object](http://msdn.microsoft.com/en-us/library/system.object(v=vs.110).aspx)

The source of the event.

args

Type: [System.EventArgs](http://msdn.microsoft.com/en-us/library/system.eventargs(v=vs.110).aspx)

An [EventArgs](http://msdn.microsoft.com/en-us/library/system.eventargs(v=vs.110).aspx) object that contains the event data.

**Return Value**

Type: [System.Boolean](http://msdn.microsoft.com/en-us/library/system.boolean(v=vs.110).aspx)  
true if the event has been canceled; otherwise, false. The default is false.

[Remarks](javascript:void(0))

ASP.NET server controls such as the [Repeater](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.repeater(v=vs.110).aspx), [DataList](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist(v=vs.110).aspx) and [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx) Web controls can contain child controls that raise events. For example, each row in a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx) control can contain one or more buttons created dynamically by templates. Rather than each button raising an event individually, events from the nested controls are "bubbled"—that is, they are sent to the naming container. The naming container in turn raises a generic event called [RowCommand](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.rowcommand(v=vs.110).aspx) with parameter values. These values allow you to determine which individual control that raised the original event. By responding to this single event, you can avoid having to write individual event-handling methods for child controls.

[Examples](javascript:void(0))

The following example overrides the OnBubbleEvent method in a custom ASP.NET server control, ParentControl. This method is invoked when a child control of ParentControl calls the [RaiseBubbleEvent](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(v=vs.110).aspx) method. When this happens, the ParentControl class writes two strings to the containing ASP.NET page, the first stating that its OnBubbleEvent method has been called, the second identifying the source control of the [RaiseBubbleEvent](http://msdn.microsoft.com/en-us/library/system.web.ui.control.raisebubbleevent(v=vs.110).aspx) method.

C#

[VB](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onbubbleevent(d=printer,v=vs.110).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

public class ParentControl : Control

{

[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Demand, Name="FullTrust")]

protected override bool OnBubbleEvent(object sender, EventArgs e)

{

Context.Response.Write("<br><br>ParentControl's OnBubbleEvent called.");

Context.Response.Write("<br>Source of event is: " + sender.ToString());

return true;

}

[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Demand, Name="FullTrust")]

protected override void Render( HtmlTextWriter myWriter)

{

myWriter.Write("ParentControl");

RenderChildren(myWriter);

}

}