**ASP.NET Page Life Cycle Overview**

This page is specific to

**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/ms178472(VS.80).aspx)

[.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/ms178472(VS.85).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4.0](http://msdn.microsoft.com/en-us/library/ms178472(VS.100).aspx)

When an ASP.NET page runs, the page goes through a life cycle in which it performs a series of processing steps. These include initialization, instantiating controls, restoring and maintaining state, running event handler code, and rendering. It is important for you to understand the page life cycle so that you can write code at the appropriate life-cycle stage for the effect you intend. Additionally, if you develop custom controls, you must be familiar with the page life cycle in order to correctly initialize controls, populate control properties with view-state data, and run any control behavior code. (The life cycle of a control is based on the page life cycle, but the page raises more events for a control than are available for an ASP.NET page alone.)

http://i.msdn.microsoft.com/Global/Images/clear.gif General Page Life-cycle Stages

In general terms, the page goes through the stages outlined in the following table. In addition to the page life-cycle stages, there are application stages that occur before and after a request but are not specific to a page. For more information, see [ASP.NET Application Life Cycle Overview for IIS 7.0](http://msdn.microsoft.com/en-us/library/bb470252.aspx).

|  |  |
| --- | --- |
| **Stage** | **Description** |
| Page request | The page request occurs before the page life cycle begins. When the page is requested by a user, ASP.NET determines whether the page needs to be parsed and compiled (therefore beginning the life of a page), or whether a cached version of the page can be sent in response without running the page. |
| Start | In the start step, page properties such as [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) and [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) are set. At this stage, the page also determines whether the request is a postback or a new request and sets the [IsPostBack](http://msdn.microsoft.com/en-us/library/system.web.ui.page.ispostback.aspx) property. Additionally, during the start step, the page's [UICulture](http://msdn.microsoft.com/en-us/library/system.web.ui.page.uiculture.aspx) property is set. |
| Page initialization | During page initialization, controls on the page are available and each control's [UniqueID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.uniqueid.aspx) property is set. Any themes are also applied to the page. If the current request is a postback, the postback data has not yet been loaded and control property values have not been restored to the values from view state. |
| Load | During load, if the current request is a postback, control properties are loaded with information recovered from view state and control state. |
| Validation | During validation, the [Validate](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basevalidator.validate.aspx) method of all validator controls is called, which sets the [IsValid](http://msdn.microsoft.com/en-us/library/system.web.ui.ivalidator.isvalid.aspx) property of individual validator controls and of the page. |
| Postback event handling | If the request is a postback, any event handlers are called. |
| Rendering | Before rendering, view state is saved for the page and all controls. During the rendering phase, the page calls the [Render](http://msdn.microsoft.com/en-us/library/system.web.ui.control.render.aspx) method for each control, providing a text writer that writes its output to the [OutputStream](http://msdn.microsoft.com/en-us/library/system.web.httpresponse.outputstream.aspx) of the page's [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) property. |
| Unload | Unload is called after the page has been fully rendered, sent to the client, and is ready to be discarded. At this point, page properties such as [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) and [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) are unloaded and any cleanup is performed. |

http://i.msdn.microsoft.com/Global/Images/clear.gif Life-cycle Events

Within each stage of the life cycle of a page, the page raises events that you can handle to run your own code. For control events, you bind the event handler to the event, either declaratively using attributes such as **onclick**, or in code.

Pages also support automatic event wire-up, meaning that ASP.NET looks for methods with particular names and automatically runs those methods when certain events are raised. If the **AutoEventWireup** attribute of the [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a.aspx) directive is set to **true** (or if it is not defined, since by default it is **true**), page events are automatically bound to methods that use the naming convention of **Page\_***event*, such as **Page\_Load** and **Page\_Init**. For more information on automatic event wire-up, see [ASP.NET Web Server Control Event Model](http://msdn.microsoft.com/en-us/library/y3bwdsh3.aspx).

The following table lists the page life-cycle events that you will use most frequently. There are more events than those listed; however, they are not used for most page processing scenarios. Instead, they are primarily used by server controls on the ASP.NET Web page to initialize and render themselves. If you want to write your own ASP.NET server controls, you need to understand more about these stages. For information about creating custom controls, see [Developing Custom ASP.NET Server Controls](http://msdn.microsoft.com/en-us/library/zt27tfhy.aspx).

|  |  |
| --- | --- |
| **Page Event** | **Typical Use** |
| [PreInit](http://msdn.microsoft.com/en-us/library/system.web.ui.page.preinit.aspx) | Use this event for the following:   * Check the [IsPostBack](http://msdn.microsoft.com/en-us/library/system.web.ui.page.ispostback.aspx) property to determine whether this is the first time the page is being processed. * Create or re-create dynamic controls. * Set a master page dynamically. * Set the [Theme](http://msdn.microsoft.com/en-us/library/system.web.ui.page.theme.aspx) property dynamically. * Read or set profile property values.   Note**Note:**  If the request is a postback, the values of the controls have not yet been restored from view state. If you set a control property at this stage, its value might be overwritten in the next event. |
| [Init](http://msdn.microsoft.com/en-us/library/system.web.ui.control.init.aspx) | Raised after all controls have been initialized and any skin settings have been applied. Use this event to read or initialize control properties. |
| [InitComplete](http://msdn.microsoft.com/en-us/library/system.web.ui.page.initcomplete.aspx) | Raised by the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) object. Use this event for processing tasks that require all initialization be complete. |
| [PreLoad](http://msdn.microsoft.com/en-us/library/system.web.ui.page.preload.aspx) | Use this event if you need to perform processing on your page or control before the [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) event.  Before the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) instance raises this event, it loads view state for itself and all controls, and then processes any postback data included with the [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) instance. |
| [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) | The [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) calls the [OnLoad](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onload.aspx) event method on the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx), then recursively does the same for each child control, which does the same for each of its child controls until the page and all controls are loaded.  Use the [OnLoad](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onload.aspx) event method to set properties in controls and establish database connections. |
| Control events | Use these events to handle specific control events, such as a [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control's [Click](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.click.aspx) event or a [TextBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.textbox.aspx) control's [TextChanged](http://msdn.microsoft.com/en-us/library/system.web.ui.mobilecontrols.textbox.textchanged.aspx) event.  Note**Note:**  In a postback request, if the page contains validator controls, check the [IsValid](http://msdn.microsoft.com/en-us/library/system.web.ui.page.isvalid.aspx) property of the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) and of individual validation controls before performing any processing. |
| [LoadComplete](http://msdn.microsoft.com/en-us/library/system.web.ui.page.loadcomplete.aspx) | Use this event for tasks that require that all other controls on the page be loaded. |
| [PreRender](http://msdn.microsoft.com/en-us/library/system.web.ui.control.prerender.aspx) | Before this event occurs:   * The [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) object calls [EnsureChildControls](http://msdn.microsoft.com/en-us/library/system.web.ui.control.ensurechildcontrols.aspx) for each control and for the page. * Each data bound control whose [DataSourceID](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.databoundcontrol.datasourceid.aspx) property is set calls its [DataBind](http://msdn.microsoft.com/en-us/library/system.web.ui.control.databind.aspx) method. For more information, see [Data Binding Events for Data-Bound Controls](http://msdn.microsoft.com/en-us/library/ms178472.aspx#databindingevents) later in this topic.   The [PreRender](http://msdn.microsoft.com/en-us/library/system.web.ui.control.prerender.aspx) event occurs for each control on the page. Use the event to make final changes to the contents of the page or its controls. |
| [SaveStateComplete](http://msdn.microsoft.com/en-us/library/system.web.ui.page.savestatecomplete.aspx) | Before this event occurs, [ViewState](http://msdn.microsoft.com/en-us/library/system.web.ui.control.viewstate.aspx) has been saved for the page and for all controls. Any changes to the page or controls at this point will be ignored.  Use this event perform tasks that require view state to be saved, but that do not make any changes to controls. |
| [Render](http://msdn.microsoft.com/en-us/library/system.web.ui.control.render.aspx) | This is not an event; instead, at this stage of processing, the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) object calls this method on each control. All ASP.NET Web server controls have a [Render](http://msdn.microsoft.com/en-us/library/system.web.ui.control.render.aspx) method that writes out the control's markup that is sent to the browser.  If you create a custom control, you typically override this method to output the control's markup. However, if your custom control incorporates only standard ASP.NET Web server controls and no custom markup, you do not need to override the [Render](http://msdn.microsoft.com/en-us/library/system.web.ui.control.render.aspx) method. For more information, see [Developing Custom ASP.NET Server Controls](http://msdn.microsoft.com/en-us/library/zt27tfhy.aspx).  A user control (an .ascx file) automatically incorporates rendering, so you do not need to explicitly render the control in code. |
| [Unload](http://msdn.microsoft.com/en-us/library/system.web.ui.control.unload.aspx) | This event occurs for each control and then for the page. In controls, use this event to do final cleanup for specific controls, such as closing control-specific database connections.  For the page itself, use this event to do final cleanup work, such as closing open files and database connections, or finishing up logging or other request-specific tasks.  Note**Note:**  During the unload stage, the page and its controls have been rendered, so you cannot make further changes to the response stream. If you attempt to call a method such as the **Response.Write** method, the page will throw an exception. |

http://i.msdn.microsoft.com/Global/Images/clear.gif Additional Page Life Cycle Considerations

Individual ASP.NET server controls have their own life cycle that is similar to the page life cycle. For example, a control's [Init](http://msdn.microsoft.com/en-us/library/system.web.ui.control.init.aspx) and [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) events occur during the corresponding page events.

Although both [Init](http://msdn.microsoft.com/en-us/library/system.web.ui.control.init.aspx) and [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) recursively occur on each control, they happen in reverse order. The [Init](http://msdn.microsoft.com/en-us/library/system.web.ui.control.init.aspx) event (and also the [Unload](http://msdn.microsoft.com/en-us/library/system.web.ui.control.unload.aspx) event) for each child control occur before the corresponding event is raised for its container (bottom-up). However the [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) event for a container occurs before the [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) events for its child controls (top-down).

You can customize the appearance or content of a control by handling the events for the control, such as the [Click](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.click.aspx) event for the [Button](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.button.aspx) control and the [SelectedIndexChanged](http://msdn.microsoft.com/en-us/library/system.windows.forms.listbox.selectedindexchanged.aspx) event for the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control. Under some circumstances, you might also handle a control's [DataBinding](http://msdn.microsoft.com/en-us/library/system.web.ui.control.databinding.aspx) or [DataBound](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basedataboundcontrol.databound.aspx) events. For more information, see the class reference topics for individual controls and [Developing Custom ASP.NET Server Controls](http://msdn.microsoft.com/en-us/library/zt27tfhy.aspx).

When inheriting a class from the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) class, in addition to handling events raised by the page, you can override methods from the page's base class. For example, you can override the page's [InitializeCulture](http://msdn.microsoft.com/en-us/library/system.web.ui.page.initializeculture.aspx) method to dynamically set culture information. Note that when creating an event handler using the **Page\_***event* syntax, the base implementation is implicitly called and therefore you do not need to call it in your method. For example, the base page class's [OnLoad](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onload.aspx) method is always called, whether you create a **Page\_Load** method or not. However, if you override the page [OnLoad](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onload.aspx) method with the **override** keyword (**Overrides** in Visual Basic), you must explicitly call the base method. For example, if you override the [OnLoad](http://msdn.microsoft.com/en-us/library/system.web.ui.control.onload.aspx) method on the page, you must call **base.Load** (**MyBase.Load** in Visual Basic) in order for the base implementation to be run.

**Catch-up Events for Added Controls**

If controls are created dynamically at run time or are authored declaratively within templates of data-bound controls, their events are initially not synchronized with those of other controls on the page. For example, for a control that is added at run time, the [Init](http://msdn.microsoft.com/en-us/library/system.web.ui.control.init.aspx) and [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load.aspx) events might occur much later in the page life cycle than the same events for controls created declaratively. Therefore, from the time that they are instantiated, dynamically added controls and controls in templates raise their events one after the other until they have caught up to the event during which it was added to the [Controls](http://msdn.microsoft.com/en-us/library/system.web.ui.control.controls.aspx) collection.

In general, you do not need to be concerned about this unless you have nested data-bound controls. If a child control has been data bound, but its container control has not yet been data bound, the data in the child control and the data in its container control can be out of sync. This is true particularly if the data in the child control performs processing based on a data-bound value in the container control.

For example, suppose you have a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) that displays a company record in each row along with a list of the company officers in a [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control. To fill the list of officers, you would bind the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control to a data source control (such as [SqlDataSource](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.sqldatasource.aspx)) that retrieves the company officer data using the CompanyID in a query.

If the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control's data-binding properties, such as [DataSourceID](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basedatalist.datasourceid.aspx) and [DataMember](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basedatalist.datamember.aspx), are set declaratively, the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control will try to bind to its data source during the containing row's [DataBinding](http://msdn.microsoft.com/en-us/library/system.web.ui.control.databinding.aspx) event. However, the CompanyID field of the row does not contain a value until the [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) control's [RowDataBound](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.rowdatabound.aspx) event occurs. In this case, the child control (the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control) is bound before the containing control (the [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) control) is bound, so their data-binding stages are out of sync.

To avoid this condition, put the data source control for the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control in the same template item as the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control itself, and do not set the data binding properties of the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) declaratively. Instead, set them programmatically at run time during the [RowDataBound](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.rowdatabound.aspx) event, so that the [ListBox](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listbox.aspx) control does not bind to its data until the CompanyID information is available.

For more information, see [Binding to Data Using a Data Source Control](http://msdn.microsoft.com/en-us/library/ms228089.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Data Binding Events for Data-Bound Controls

To help you understand the relationship between the page life cycle and data binding events, the following table lists data-related events in data-bound controls such as the [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx), [DetailsView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.detailsview.aspx), and [FormView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.formview.aspx) controls.

|  |  |
| --- | --- |
| **Control Event** | **Typical Use** |
| [DataBinding](http://msdn.microsoft.com/en-us/library/system.web.ui.control.databinding.aspx) | This event is raised by data-bound controls before the [PreRender](http://msdn.microsoft.com/en-us/library/system.web.ui.control.prerender.aspx) event of the containing control (or of the [Page](http://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) object) and marks the beginning of binding the control to the data.  Use this event to manually open database connections, if required. (The data source controls often make this unnecessary.) |
| [RowCreated](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.rowcreated.aspx) ([GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) only) or [ItemCreated](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist.itemcreated.aspx) ([DataList](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist.aspx), [DetailsView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.detailsview.aspx), [SiteMapPath](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.sitemappath.aspx), [DataGrid](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datagrid.aspx), [FormView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.formview.aspx), [Repeater](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.repeater.aspx), and [ListView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listview.aspx) controls) | Use this event to manipulate content that is not dependent on data binding. For example, at run time, you might programmatically add formatting to a header or footer row in a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) control. |
| [RowDataBound](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.rowdatabound.aspx) ([GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) only) or [ItemDataBound](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist.itemdatabound.aspx) ([DataList](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datalist.aspx), [SiteMapPath](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.sitemappath.aspx), [DataGrid](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.datagrid.aspx), [Repeater](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.repeater.aspx), and [ListView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listview.aspx) controls) | When this event occurs, data is available in the row or item, so you can format data or set the [FilterExpression](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.sqldatasource.filterexpression.aspx) property on child data source controls for displaying related data within the row or item. |
| [DataBound](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basedataboundcontrol.databound.aspx) | This event marks the end of data-binding operations in a data-bound control. In a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview.aspx) control, data binding is complete for all rows and any child controls.  Use this event to format data bound content or to initiate data binding in other controls that depend on values from the current control's content. (For details, see "Catch-up Events for Added Controls" earlier in this topic.) |

http://i.msdn.microsoft.com/Global/Images/clear.gif Login Control Events

The [Login](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.aspx) control can use settings in the Web.config file to manage membership authentication automatically. However, if your application requires you to customize how the control works, or if you want to understand how [Login](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.aspx) control events relate to the page life cycle, you can use the events listed in the following table.

|  |  |
| --- | --- |
| **Control Event** | **Typical Use** |
| [LoggingIn](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.loggingin.aspx) | This event is raised during a postback, after the page's [LoadComplete](http://msdn.microsoft.com/en-us/library/system.web.ui.page.loadcomplete.aspx) event has occurred. It marks the beginning of the login process.  Use this event for tasks that must occur prior to beginning the authentication process. |
| [Authenticate](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.authenticate.aspx) | This event is raised after the [LoggingIn](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.loggingin.aspx) event.  Use this event to override or enhance the default authentication behavior of a [Login](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.aspx) control. |
| [LoggedIn](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.loggedin.aspx) | This event is raised after the user name and password have been authenticated.  Use this event to redirect to another page or to dynamically set the text in the control. This event does not occur if there is an error or if authentication fails. |
| [LoginError](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.login.loginerror.aspx) | This event is raised if authentication was not successful.  Use this event to set text in the control that explains the problem or to direct the user to a different page. |