**ASP.NET 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 2003/.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/4w3ex9c2(VS.71).aspx)

[Microsoft Visual Studio 2005/.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/4w3ex9c2(VS.80).aspx)

[.NET Framework 3.0](http://msdn.microsoft.com/en-us/library/4w3ex9c2(VS.85).aspx)

[Microsoft Visual Studio 2010/.NET Framework 4.0](http://msdn.microsoft.com/en-us/library/4w3ex9c2(VS.100).aspx)

ASP.NET is a unified Web development model that includes the services necessary for you to build enterprise-class Web applications with a minimum of coding. ASP.NET is part of the .NET Framework, and when coding ASP.NET applications you have access to classes in the .NET Framework. You can code your applications in any language compatible with the common language runtime (CLR), including Microsoft Visual Basic, C#, JScript .NET, and J#. These languages enable you to develop ASP.NET applications that benefit from the common language runtime, type safety, inheritance, and so on.

ASP.NET includes:

* A page and controls framework
* The ASP.NET compiler
* Security infrastructure
* State-management facilities
* Application configuration
* Health monitoring and performance features
* Debugging support
* An XML Web services framework
* Extensible hosting environment and application life cycle management
* An extensible designer environment

http://i.msdn.microsoft.com/Global/Images/clear.gif Page and Controls Framework

The ASP.NET page and controls framework is a programming framework that runs on a Web server to dynamically produce and render ASP.NET Web pages. ASP.NET Web pages can be requested from any browser or client device, and ASP.NET renders markup (such as HTML) to the requesting browser. As a rule, you can use the same page for multiple browsers, because ASP.NET renders the appropriate markup for the browser making the request. However, you can design your ASP.NET Web page to target a specific browser, such as Microsoft Internet Explorer 6, and take advantage of the features of that browser. ASP.NET supports mobile controls for Web-enabled devices such as cellular phones, handheld computers, and personal digital assistants (PDAs).

ASP.NET Web pages are completely object-oriented. Within ASP.NET Web pages you can work with HTML elements using properties, methods, and events. The ASP.NET page framework removes the implementation details of the separation of client and server inherent in Web-based applications by presenting a unified model for responding to client events in code that runs at the server. The framework also automatically maintains the state of a page and the controls on that page during the page processing life cycle. For more information see [ASP.NET Web Pages Overview](http://msdn.microsoft.com/en-us/library/428509ah.aspx).

The ASP.NET page and controls framework also enables you to encapsulate common UI functionality in easy-to-use, reusable controls. Controls are written once, can be used in many pages, and are integrated into the ASP.NET Web page that they are placed in during rendering.

The ASP.NET page and controls framework also provides features to control the overall look and feel of your Web site via themes and skins. You can define themes and skins and then apply them at a page level or at a control level. For more information, see [ASP.NET Themes and Skins Overview](http://msdn.microsoft.com/en-us/library/ykzx33wh.aspx).

In addition to themes, you can define master pages that you use to create a consistent layout for the pages in your application. A single master page defines the layout and standard behavior that you want for all the pages (or a group of pages) in your application. You can then create individual content pages that contain the page-specific content you want to display. When users request the content pages, they merge with the master page to produce output that combines the layout of the master page with the content from the content page. For more information see [ASP.NET Master Pages Overview](http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif ASP.NET Compiler

All ASP.NET code is compiled, which enables strong typing, performance optimizations, and early binding, among other benefits. Once the code has been compiled, the common language runtime further compiles ASP.NET code to native code, providing improved performance.

ASP.NET includes a compiler that will compile all your application components including pages and controls into an assembly that the ASP.NET hosting environment can then use to service user requests. For more information, see [ASP.NET Compilation Overview](http://msdn.microsoft.com/en-us/library/ms178466.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Security Infrastructure

In addition to the security features of .NET, ASP.NET provides an advanced security infrastructure for authenticating and authorizing user access as well as performing other security-related tasks. You can authenticate users using Windows authentication supplied by IIS, or you can manage authentication using your own user database using ASP.NET forms authentication and ASP.NET membership. Additionally, you can manage the authorization to the capabilities and information of your Web application using Windows groups or your own custom role database using ASP.NET roles. You can easily remove, add to, or replace these schemes depending upon the needs of your application. For more information see the following topics:

* [Securing ASP.NET Web Sites](http://msdn.microsoft.com/en-us/library/91f66yxt.aspx)
* [Managing Users by Using Membership](http://msdn.microsoft.com/en-us/library/tw292whz.aspx)
* [Managing Authorization Using Roles](http://msdn.microsoft.com/en-us/library/9ab2fxh0.aspx)
* [Forms Authentication Provider](http://msdn.microsoft.com/en-us/library/9wff0kyh.aspx)

ASP.NET always runs with a particular Windows identity so you can secure your application using Windows capabilities such as NTFS Access Control Lists (ACLs), database permissions, and so on. For more information on the identity of ASP.NET, see [Configuring ASP.NET Process Identity](http://msdn.microsoft.com/en-us/library/dwc1xthy.aspx) and [ASP.NET Impersonation](http://msdn.microsoft.com/en-us/library/xh507fc5.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif State-Management Facilities

ASP.NET provides intrinsic state management functionality that enables you to store information between page requests, such as customer information or the contents of a shopping cart. You can save and manage application-specific, session-specific, page-specific, user-specific, and developer-defined information. This information can be independent of any controls on the page.

ASP.NET offers distributed state facilities, which enable you to manage state information across multiple instances of the same application on one computer or on several computers. For more information see [ASP.NET State Management Overview](http://msdn.microsoft.com/en-us/library/75x4ha6s.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif ASP.NET Configuration

ASP.NET applications use a configuration system that enables you to define configuration settings for your Web server, for a Web site, or for individual applications. You can make configuration settings at the time your ASP.NET applications are deployed and can add or revise configuration settings at any time with minimal impact on operational Web applications and servers. ASP.NET configuration settings are stored in XML-based files. Because these XML files are ASCII text files, it is simple to make configuration changes to your Web applications. You can extend the configuration scheme to suit your requirements. For more information see [ASP.NET Configuration Overview](http://msdn.microsoft.com/en-us/library/ms178683.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Health Monitoring and Performance Features

ASP.NET includes features that enable you to monitor health and performance of your ASP.NET application. ASP.NET health monitoring enables reporting of key events that provide information about the health of an application and about error conditions. These events show a combination of diagnostics and monitoring characteristics and offer a high degree of flexibility in terms of what is logged and how it is logged. For more information see [ASP.NET Health Monitoring Overview](http://msdn.microsoft.com/en-us/library/bb398933.aspx).

ASP.NET supports two groups of performance counters accessible to your applications:

* The ASP.NET system performance counter group
* The ASP.NET application performance counter group

For more information, see [Monitoring ASP.NET Application Performance](http://msdn.microsoft.com/en-us/library/3xxk09t8.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Debugging Support

ASP.NET takes advantage of the run-time debugging infrastructure to provide cross-language and cross-computer debugging support. You can debug both managed and unmanaged objects, as well as all languages supported by the common language runtime and script languages. For details, see [ASP.NET Debugging](http://msdn.microsoft.com/en-us/library/ms228091.aspx).

In addition, the ASP.NET page framework provides a trace mode that enables you to insert instrumentation messages into your ASP.NET Web pages. For more information see [NIB: What's New in ASP.NET Tracing](http://msdn.microsoft.com/en-us/library/h1ztfx8b.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif XML Web Services Framework

ASP.NET supports XML Web services. An XML Web service is a component containing business functionality that enables applications to exchange information across firewalls using standards like HTTP and XML messaging. XML Web services are not tied to a particular component technology or object-calling convention. As a result, programs written in any language, using any component model, and running on any operating system can access XML Web services. For more information, see [XML Web Services Using ASP.NET](http://msdn.microsoft.com/en-us/library/ba0z6a33.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Extensible Hosting Environment and Application Life-Cycle Management

ASP.NET includes an extensible hosting environment that controls the life cycle of an application from when a user first accesses a resource (such as a page) in the application to the point at which the application is shut down. While ASP.NET relies on a Web server (IIS) as an application host, ASP.NET provides much of the hosting functionality itself. The architecture of ASP.NET enables you to respond to application events and create custom HTTP handlers and HTTP modules. For more information see [ASP.NET Application Life Cycle Overview for IIS 5.0 and 6.0](http://msdn.microsoft.com/en-us/library/ms178473.aspx).

http://i.msdn.microsoft.com/Global/Images/clear.gif Extensible Designer Environment

ASP.NET includes enhanced support for creating designers for Web server controls for use with a visual design tool such as Visual Studio. Designers enable you to build a design-time user interface for a control, so that developers can configure your control's properties and content in the visual design tool. For more information see [ASP.NET Control Designers Overview](http://msdn.microsoft.com/en-us/library/wxh45wzs.aspx).