**ASP.NET** is a [web application framework](http://en.wikipedia.org/wiki/Web_application_framework) developed and marketed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) to allow [programmers](http://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](http://en.wikipedia.org/wiki/Web_site), [web applications](http://en.wikipedia.org/wiki/Web_application) and [web services](http://en.wikipedia.org/wiki/Web_service). It was first released in January 2002 with version 1.0 of the [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework), and is the successor to Microsoft's [Active Server Pages](http://en.wikipedia.org/wiki/Active_Server_Pages) (ASP) technology. ASP.NET is built on the [Common Language Runtime](http://en.wikipedia.org/wiki/Common_Language_Runtime) (CLR), allowing programmers to write ASP.NET code using any supported [.NET language](http://en.wikipedia.org/wiki/.NET_Languages).

|  |
| --- |
| Contents [[hide](javascript:toggleToc())]   * [1 History](http://en.wikipedia.org/wiki/ASP.NET#History#History) * [2 Characteristics](http://en.wikipedia.org/wiki/ASP.NET#Characteristics#Characteristics)   + [2.1 Pages](http://en.wikipedia.org/wiki/ASP.NET#Pages#Pages)   + [2.2 Code-behind model](http://en.wikipedia.org/wiki/ASP.NET#Code-behind_model#Code-behind_model)     - [2.2.1 Example](http://en.wikipedia.org/wiki/ASP.NET#Example#Example)   + [2.3 User controls](http://en.wikipedia.org/wiki/ASP.NET#User_controls#User_controls)   + [2.4 Rendering technique](http://en.wikipedia.org/wiki/ASP.NET#Rendering_technique#Rendering_technique)   + [2.5 State management](http://en.wikipedia.org/wiki/ASP.NET#State_management#State_management)     - [2.5.1 Application state](http://en.wikipedia.org/wiki/ASP.NET#Application_state#Application_state)     - [2.5.2 Session state](http://en.wikipedia.org/wiki/ASP.NET#Session_state#Session_state)     - [2.5.3 View state](http://en.wikipedia.org/wiki/ASP.NET#View_state#View_state)     - [2.5.4 Other](http://en.wikipedia.org/wiki/ASP.NET#Other#Other)   + [2.6 Template engine](http://en.wikipedia.org/wiki/ASP.NET#Template_engine#Template_engine)   + [2.7 Other files](http://en.wikipedia.org/wiki/ASP.NET#Other_files#Other_files)   + [2.8 Directory structure](http://en.wikipedia.org/wiki/ASP.NET#Directory_structure#Directory_structure) * [3 Performance](http://en.wikipedia.org/wiki/ASP.NET#Performance#Performance) * [4 Extension](http://en.wikipedia.org/wiki/ASP.NET#Extension#Extension) * [5 ASP.NET compared to ASP classic](http://en.wikipedia.org/wiki/ASP.NET#ASP.NET_compared_to_ASP_classic#ASP.NET_compared_to_ASP_classic) * [6 Criticism](http://en.wikipedia.org/wiki/ASP.NET#Criticism#Criticism) * [7 Development tools](http://en.wikipedia.org/wiki/ASP.NET#Development_tools#Development_tools) * [8 Frameworks](http://en.wikipedia.org/wiki/ASP.NET#Frameworks#Frameworks) * [9 Versions](http://en.wikipedia.org/wiki/ASP.NET#Versions#Versions) * [10 Notes](http://en.wikipedia.org/wiki/ASP.NET#Notes#Notes) * [11 References](http://en.wikipedia.org/wiki/ASP.NET#References#References) * [12 Further reading](http://en.wikipedia.org/wiki/ASP.NET#Further_reading#Further_reading) * [13 External links](http://en.wikipedia.org/wiki/ASP.NET#External_links#External_links) |



## Characteristics

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=3)] Pages

ASP.NET pages, known officially as "web forms", are the main building block for application development.[[8]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-macdonald63-7#cite_note-macdonald63-7) Web forms are contained in files with an **ASPX** extension; in programming jargon, these files typically contain static ([X](http://en.wikipedia.org/wiki/XHTML))[HTML](http://en.wikipedia.org/wiki/HTML) markup, as well as markup defining server-side Web Controls and User Controls where the developers place all the required static and dynamic content for the web page. Additionally, dynamic code which runs on the server can be placed in a page within a block **<% -- dynamic code -- %>** which is similar to other web development technologies such as [PHP](http://en.wikipedia.org/wiki/PHP), [JSP](http://en.wikipedia.org/wiki/JavaServer_Pages), and [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages), but this practice is generally discouraged except for the purposes of [data binding](http://en.wikipedia.org/wiki/Binding_(computer_science)) since it requires more calls when rendering the page.[[*citation needed*](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Note that this sample uses code "inline", as opposed to code behind.

**<%**@ Page Language="C#" **%>**

<!DOCTYPE html **PUBLIC** "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">

protected void Page\_Load**(**object sender, EventArgs e**)**

**{**

Label1.Text = DateTime.Now.ToLongDateString**()**;

**}**

**</script>**

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Sample page</title>

</head>

<body>

<form id="form1" runat="server">

<div>

The current time **is**: <asp:Label runat="server" id="Label1" />

</div>

</form>

</body>

</html>

**Default.aspx (Visual Studio 2008)**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="WebApplication1.\_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title>Untitled Page</title>

</head>

<body>

<form id="form1" runat="server">

<div>

</div>

</form>

</body>

</html>

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=4)] Code-behind model

It is recommended by Microsoft for dealing with dynamic program code to use the code-behind model, which places this code in a separate file or in a specially designated script tag. Code-behind files typically have names like *MyPage.aspx.cs* or *MyPage.aspx.vb* based on the ASPX file name (this practice is automatic in [Microsoft Visual Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio) and other [IDEs](http://en.wikipedia.org/wiki/Integrated_development_environment)). When using this style of programming, the developer writes code to respond to different events, like the page being loaded, or a control being clicked, rather than a procedural walk through the document.

ASP.NET's code-behind model marks a departure from Classic ASP in that it encourages developers to build applications with [separation of presentation and content](http://en.wikipedia.org/wiki/Separation_of_presentation_and_content) in mind. In theory, this would allow a web designer, for example, to focus on the design markup with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in [model-view-controller](http://en.wikipedia.org/wiki/Model-view-controller) frameworks.

#### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=5)] Example

**<%**@ Page Language="C#" CodeFile="SampleCodeBehind.aspx.cs" Inherits="Website.SampleCodeBehind"

AutoEventWireup="true" **%>**

The above tag is placed at the beginning of the ASPX file. The *CodeFile* property of the *@ Page* directive specifies the file (.cs or .vb) acting as the code-behind while the *Inherits* property specifies the Class the Page derives from. In this example, the *@ Page* directive is included in SamplePage.aspx, then SampleCodeBehind.aspx.cs acts as the code-behind for this page:

using System;

namespace Website

{

public partial class SampleCodeBehind : System.Web.UI.Page

{

protected override void Page\_Load(EventArgs e)

{

base.OnLoad(e);

}

}

}

In this case, the Page\_Load() method is called every time the ASPX page is requested. The programmer can implement event handlers at several stages of the page execution process to perform processing.

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=6)] User controls

ASP.NET supports creating reusable components through the creation of User Controls. A User Control follows the same structure as a Web Form, except that such controls are derived from the System.Web.UI.UserControl class, and are stored in **ASCX** files. Like ASPX files, a ASCX contains static [HTML](http://en.wikipedia.org/wiki/HTML) or [XHTML](http://en.wikipedia.org/wiki/XHTML) markup, as well as markup defining web control and other User Controls. The code-behind model can be used.

Programmers can add their own properties, methods,[[9]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-8#cite_note-8) and event handlers.[[10]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-9#cite_note-9) An event bubbling mechanism provides the ability to pass an event fired by a user control up to its containing page.

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=7)] Rendering technique

ASP.NET uses a *visited composites* rendering technique. During compilation, the template (.aspx) file is compiled into initialization code which builds a control tree (the composite) representing the original template. Literal text goes into instances of the Literal control class, and server controls are represented by instances of a specific control class. The initialization code is combined with user-written code (usually by the assembly of multiple partial classes) and results in a class specific for the page. The page doubles as the root of the **control tree**.

Actual requests for the page are processed through a number of steps. First, during the initialization steps, an instance of the page class is created and the initialization code is executed. This produces the initial control tree which is now typically manipulated by the methods of the page in the following steps. As each node in the tree is a control represented as an instance of a class, the code may change the tree structure as well as manipulate the properties/methods of the individual nodes. Finally, during the rendering step a visitor is used to visit every node in the tree, asking each node to render itself using the methods of the visitor. The resulting HTML output is sent to the client.

After the request has been processed, the instance of the page class is discarded and with it the entire control tree.

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=8)] State management

ASP.NET applications are hosted in a [web server](http://en.wikipedia.org/wiki/Web_server) and are accessed over the [stateless](http://en.wikipedia.org/wiki/Stateless_server) [HTTP](http://en.wikipedia.org/wiki/HTTP) protocol. As such, if the application uses stateful interaction, it has to implement [state management](http://en.wikipedia.org/wiki/State_management) on its own. ASP.NET provides various functionality for state management in ASP.NET applications.

#### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=9)] Application state

Application state is a collection of user-defined variables that are shared by an ASP.NET application. These are set and initialized when the Application\_OnStart event fires on the loading of the first instance of the applications and are available till the last instance exits. Application state variables are accessed using the Applications collection, which provides a wrapper for the application state variables. Application state variables are identified by names.[[11]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-state-10#cite_note-state-10)

#### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=10)] Session state

Session state is a collection of user-defined session variables, which are persisted during a user session. These variables are unique to different instances of a user session, and are accessed using the Session collection. Session variables can be set to be automatically destroyed after a defined time of inactivity, even if the session does not end. At the client end, a user session is identified either by a [cookie](http://en.wikipedia.org/wiki/HTTP_cookie) or by encoding the session ID in the URL itself.[[11]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-state-10#cite_note-state-10)

ASP.NET supports three modes of persistence for session variables:[[11]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-state-10#cite_note-state-10)

In Process Mode

The session variables are maintained within the ASP.NET [process](http://en.wikipedia.org/wiki/Process_(computing)). This is the fastest way, however, in this mode the variables are destroyed when the ASP.NET process is recycled or shut down. Since the application is recycled from time to time this mode is not recommended for critical applications.

ASPState Mode

In this mode, ASP.NET runs a separate [Windows service](http://en.wikipedia.org/wiki/Windows_service) that maintains the state variables. Because the state management happens outside the ASP.NET process, this has a negative impact on performance, but it allows multiple ASP.NET instances to share the same state server, thus allowing an ASP.NET application to be load-balanced and scaled out on multiple servers. Also, since the state management service runs independent of ASP.NET, variables can persist across ASP.NET process shutdowns.

SqlServer Mode

In this mode, the state variables are stored in a [database server](http://en.wikipedia.org/wiki/Database_server), accessible using [SQL](http://en.wikipedia.org/wiki/SQL). Session variables can be persisted across ASP.NET process shutdowns in this mode as well. The main advantage of this mode is it would allow the application to balance load on a server cluster while sharing sessions between servers.

#### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=11)] View state

View state refers to the page-level state management mechanism, which is utilized by the HTML pages emitted by ASP.NET applications to maintain the state of the web form controls and [widgets](http://en.wikipedia.org/wiki/GUI_widget). The state of the controls are encoded and sent to the server at every form submission in a hidden field known as \_\_VIEWSTATE. The server sends back the variable so that when the page is re-rendered, the controls render at their last state. At the server side, the application might change the viewstate, if the processing results in updating the state of any control. The states of individual controls are decoded at the server, and are available for use in ASP.NET pages using the ViewState collection.[[12]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-11#cite_note-11) [[13]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-12#cite_note-12)

#### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=12)] Other

Other means of state management that are supported by ASP.NET are [cookies](http://en.wikipedia.org/wiki/Cookie), [caching](http://en.wikipedia.org/wiki/Caching), and using the [query string](http://en.wikipedia.org/wiki/Query_string).

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=13)] Template engine

When first released, ASP.NET lacked a [template engine](http://en.wikipedia.org/wiki/Template_engine). Because the [.NET framework](http://en.wikipedia.org/wiki/.NET_framework) is [object-oriented](http://en.wikipedia.org/wiki/Object_oriented) and allows for [inheritance](http://en.wikipedia.org/wiki/Inheritance_(computer_science)), many developers would define a new base class that inherits from "System.Web.UI.Page", write [methods](http://en.wikipedia.org/wiki/Method_(computer_science)) here that render HTML, and then make the pages in their application inherit from this new class. While this allows for common elements to be reused across a site, it adds complexity and mixes [source code](http://en.wikipedia.org/wiki/Source_code) with [markup](http://en.wikipedia.org/wiki/Markup_language). Furthermore, this method can only be visually tested by running the application - not while designing it. Other developers have used [include files](http://en.wikipedia.org/wiki/Header_file) and other tricks to avoid having to implement the same navigation and other elements in every page.

ASP.NET 2.0 introduced the concept of "master pages", which allow for [template](http://en.wikipedia.org/wiki/Web_template)-based page development. A web application can have one or more master pages, which can be nested.[[14]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-13#cite_note-13) Master templates have place-holder controls, called *ContentPlaceHolders* to denote where the dynamic content goes, as well as [HTML](http://en.wikipedia.org/wiki/HTML) and [JavaScript](http://en.wikipedia.org/wiki/JavaScript) shared across child pages.

Child pages use those ContentPlaceHolder controls, which must be mapped to the place-holder of the master page that the content page is populating. The rest of the page is defined by the shared parts of the master page, much like a [mail merge](http://en.wikipedia.org/wiki/Mail_merge) in a [word processor](http://en.wikipedia.org/wiki/Word_processor). All markup and [server](http://en.wikipedia.org/wiki/Server_(computing)) controls in the content page must be placed within the ContentPlaceHolder control.

When a request is made for a content page, ASP.NET merges the output of the content page with the output of the master page, and sends the output to the user.

The master page remains fully accessible to the content page. This means that the content page may still manipulate headers, change title, configure caching etc. If the master page exposes public properties or methods (e.g. for setting copyright notices) the content page can use these as well.

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=14)] Other files

Other file extensions associated with different versions of ASP.NET include:

|  |  |  |
| --- | --- | --- |
| **Extension** | **Required version** | **Description** |
| asax | 1.0 | [Global.asax](http://en.wikipedia.org/wiki/Global.asax), used for application-level logic [[15]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-14#cite_note-14) |
| ascx | 1.0 | Web UserControls: custom [controls](http://en.wikipedia.org/wiki/Widget_(computing)) to be placed onto web pages. |
| ashx | 1.0 | custom [HTTP](http://en.wikipedia.org/wiki/HTTP) handlers. |
| asmx | 1.0 | [web service](http://en.wikipedia.org/wiki/Web_service) pages. |
| axd | 1.0 | when enabled in web.config requesting trace.axd outputs application-level tracing. Also used for the special webresource.axd handler which allows control/component developers to package a component/control complete with images, script, css etc. for deployment in a single file (an 'assembly') |
| browser | 2.0 | browser capabilities files stored in [XML](http://en.wikipedia.org/wiki/XML) format; introduced in version 2.0. ASP.NET 2 includes many of these by default, to support common web browsers. These specify which browsers have which capabilities, so that ASP.NET 2 can automatically customize and optimize its output accordingly. Special .browser files are available for free download to handle, for instance, the W3C Validator, so that it properly shows standards-compliant pages as being standards-compliant. Replaces the harder-to-use BrowserCaps section that was in machine.config and could be overridden in [web.config](http://en.wikipedia.org/wiki/Web.config) in ASP.NET 1.x. |
| config | 1.0 | [web.config](http://en.wikipedia.org/wiki/Web.config) is the only file in a specific Web application to use this extension by default (machine.config similarly affects the entire Web server and all applications on it), however ASP.NET provides facilities to create and consume other config files. These are stored in [XML](http://en.wikipedia.org/wiki/XML) format. |
| cs/vb | 1.0 | Code files (cs indicates C#, vb indicates Visual Basic). Code behind files (see above) predominantly have the extension ".aspx.cs" or ".aspx.vb" for the two most common languages. Other code files (often containing common "library" classes) can also exist in the web folders with the cs/vb extension. In ASP.NET 2 these should be placed inside the App\_Code folder where they are dynamically compiled and available to the whole application. |
| dbml | 3.5 | [LINQ to SQL](http://en.wikipedia.org/wiki/Language_Integrated_Query) data classes file |
| master | 2.0 | master page file |
| resx | 1.0 | resource files for [internationalization](http://en.wikipedia.org/wiki/Internationalization) and [localization](http://en.wikipedia.org/wiki/Localization). Resource files can be global (e.g. messages) or "local" which means specific for a single aspx or ascx file. |
| sitemap | 2.0 | sitemap configuration files |
| skin | 2.0 | theme skin files. |
| svc | 3.0 | [Windows Communication Foundation](http://en.wikipedia.org/wiki/Windows_Communication_Foundation) service file |

### [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=15)] Directory structure

In general, the ASP.NET directory structure can be determined by the developer's preferences. Apart from a few reserved directory names, the site can span any number of directories. The structure is typically reflected directly in the urls. Although ASP.NET provides means for intercepting the request at any point during processing, the developer is not forced to funnel requests through a central application or front controller.

The special directory names (from ASP.NET 2.0 on) are [[16]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-15#cite_note-15):

App\_Browsers

holds site-specific browser definition files.

App\_Code

This is the "raw code" directory. The ASP.NET server automatically compiles files (and subdirectories) in this folder into an assembly which is accessible in the code of every page of the site. App\_Code will typically be used for data access abstraction code, model code and business code. Also any site-specific http handlers and modules and web service implementation go in this directory. As an alternative to using App\_Code the developer may opt to provide a separate assembly with precompiled code.

App\_Data

default directory for [databases](http://en.wikipedia.org/wiki/Database), such as Access mdb files and SQL Server mdf files. This directory is usually the only one with write access for the application.

App\_LocalResources

Contains localized resource files for individual pages of the site. E.g. a file called CheckOut.aspx.fr-FR.resx holds localized resources for the french version of the CheckOut.aspx page. When the UI culture is set to french, ASP.NET will automatically find and use this file for localization.

App\_GlobalResources

Holds **resx** files with localized resources available to every page of the site. This is where the ASP.NET developer will typically store localized messages etc. which are used on more than one page.

App\_Themes

holds alternative themes of the site.

App\_WebReferences

holds discovery files and [WSDL](http://en.wikipedia.org/wiki/Web_Services_Description_Language) files for references to [web services](http://en.wikipedia.org/wiki/Web_service) to be consumed in the site.

Bin

Contains compiled code (.dll files) for controls, components, or other code that you want to reference in your application. Any classes represented by code in the Bin folder are automatically referenced in your application.

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=16)] Performance

ASP.NET aims for performance benefits over other script-based technologies (including Classic ASP) by compiling the server-side code to one or more [DLL](http://en.wikipedia.org/wiki/Dynamic_link_library) [files](http://en.wikipedia.org/wiki/Computer_file) on the [web server](http://en.wikipedia.org/wiki/Web_server).[[17]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-macdonald7-8-16#cite_note-macdonald7-8-16) This compilation happens automatically the first time a page is requested (which means the developer need not perform a separate compilation step for pages). This feature provides the ease of development offered by scripting languages with the performance benefits of a compiled binary. However, the compilation might cause a noticeable but short delay to the web user when the newly-edited page is first requested from the web server, but won't again unless the page requested is updated further.

The ASPX and other resource files are placed in a virtual host on an [Internet Information Services](http://en.wikipedia.org/wiki/Internet_Information_Services) server (or other compatible ASP.NET servers; see Other Implementations, below). The first time a client requests a page, the .NET framework parses and compiles the file(s) into a .NET assembly and sends the response; subsequent requests are served from the DLL files. By default ASP.NET will compile the entire site in batches of 1000 files upon first request. If the compilation delay is causing problems, the batch size or the compilation strategy may be tweaked.

Developers can also choose to pre-compile their code before deployment, eliminating the need for just-in-time compilation in a production environment.

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=17)] Extension

Microsoft has released some extension frameworks that plug into **ASP.NET** and extend its functionality. Some of them are:

[ASP.NET AJAX](http://en.wikipedia.org/wiki/ASP.NET_AJAX)

An extension with both client-side as well as server-side components for writing ASP.NET pages that incorporate [AJAX](http://en.wikipedia.org/wiki/AJAX_(programming)) functionality.

[ASP.NET MVC Framework](http://en.wikipedia.org/wiki/ASP.NET_MVC_Framework)

An extension to author ASP.NET pages using the [MVC](http://en.wikipedia.org/wiki/Model_View_Controller) architecture.

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=18)] ASP.NET compared to ASP classic

ASP.NET attempts to simplify developers' transition from [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) application development to web development by offering the ability to build pages composed of [*controls*](http://en.wikipedia.org/wiki/Widget_(computing)) similar to a Windows [user interface](http://en.wikipedia.org/wiki/User_interface). A web control, such as a *button* or *label*, functions in very much the same way as its Windows counterpart: code can assign its properties and respond to its events. Controls know how to render themselves: whereas Windows controls draw themselves to the screen, web controls produce segments of [HTML](http://en.wikipedia.org/wiki/HTML) and [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which form part of the resulting page sent to the end-user's browser.

ASP.NET encourages the programmer to develop applications using an [event-driven](http://en.wikipedia.org/wiki/Event-driven_programming) [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) model, rather than in conventional web-[scripting](http://en.wikipedia.org/wiki/Scripting_programming_language) environments like ASP and [PHP](http://en.wikipedia.org/wiki/PHP). The framework attempts to combine existing technologies such as JavaScript with internal components like "[ViewState](http://en.wikipedia.org/wiki/ViewState)" to bring persistent (inter-request) state to the inherently [stateless](http://en.wikipedia.org/wiki/Stateless_server) web environment.

Other differences compared to ASP classic are:

* Compiled code means applications run faster with more design-time errors trapped at the development stage.
* Significantly improved run-time error handling, making use of [exception handling](http://en.wikipedia.org/wiki/Exception_handling) using try-catch blocks.
* Similar metaphors to Microsoft Windows applications such as controls and events.
* An extensive set of controls and class libraries allows the rapid building of applications, plus user-defined controls allow commonly-used [web template](http://en.wikipedia.org/wiki/Web_template), such as menus. Layout of these controls on a page is easier because most of it can be done visually in most editors.
* ASP.NET leverages the multi-language capabilities of the .NET [Common Language Runtime](http://en.wikipedia.org/wiki/Common_Language_Runtime), allowing web pages to be coded in VB.NET, C#, J#, Delphi.NET, Chrome etc.
* Ability to cache the whole page or just parts of it to improve performance.
* Ability to use the [code-behind](http://en.wikipedia.org/wiki/Code-behind) development model to separate business logic from presentation.
* If an ASP.NET application [leaks memory](http://en.wikipedia.org/wiki/Memory_leak), the ASP.NET runtime unloads the AppDomain hosting the erring application and reloads the application in a new AppDomain.
* [Session state](http://en.wikipedia.org/wiki/Session_state) in ASP.NET can be saved in a [Microsoft SQL Server](http://en.wikipedia.org/wiki/Microsoft_SQL_Server) database or in a separate process running on the same machine as the web server or on a different machine. That way session values are not lost when the web server is reset or the ASP.NET worker process is recycled.
* Versions of ASP.NET prior to 2.0 were criticized for their lack of standards compliance. The generated HTML and JavaScript sent to the client browser would not always validate against [W3C](http://en.wikipedia.org/wiki/W3C)/ECMA standards. In addition, the framework's browser detection feature sometimes incorrectly identified web browsers other than Microsoft's own [Internet Explorer](http://en.wikipedia.org/wiki/Internet_Explorer) as "downlevel" and returned HTML/JavaScript to these clients with some of the features removed, or sometimes crippled or broken. However, in version 2.0, all controls generate valid HTML 4.0, XHTML 1.0 (the default) or XHTML 1.1 output, depending on the site configuration. Detection of standards-compliant web browsers is more robust and support for [Cascading Style Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) is more extensive.
* Web Server Controls: these are controls introduced by ASP.NET for providing the UI for the web form. These controls are state managed controls and are [WYSIWYG](http://en.wikipedia.org/wiki/WYSIWYG) controls.

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=19)] Criticism

On [IIS](http://en.wikipedia.org/wiki/Internet_Information_Services) 6.0 and lower, pages written using different versions of the ASP framework can't share [Session State](http://en.wikipedia.org/wiki/Session_State) without the use of third-party libraries. This criticism does not apply to ASP.NET and ASP applications running side by side on [IIS](http://en.wikipedia.org/wiki/Internet_Information_Services) 7. With [IIS](http://en.wikipedia.org/wiki/Internet_Information_Services) 7, modules may be run in an integrated pipeline that allows modules written in any language to be executed for any request.[[18]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-17#cite_note-17)[[*citation needed*](http://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

ASP.NET 2.0 Web Forms produces markup that passes W3C validation, but it is debatable as to whether this increases [accessibility](http://en.wikipedia.org/wiki/Web_accessibility), one of the benefits of a semantic [XHTML](http://en.wikipedia.org/wiki/XHTML) page + [CSS](http://en.wikipedia.org/wiki/CSS) representation. Several controls, such as the [*Login*](http://en.wikipedia.org/wiki/Login) controls and the *Wizard* control, use [HTML](http://en.wikipedia.org/wiki/HTML) [tables](http://en.wikipedia.org/wiki/Table_(HTML)) for layout by default. [Microsoft](http://en.wikipedia.org/wiki/Microsoft) has solved this problem by releasing the [ASP.NET 2.0 CSS Control Adapters](http://www.asp.net/cssadapters/), a free add-on that produces compliant accessible [XHTML](http://en.wikipedia.org/wiki/XHTML)+[CSS](http://en.wikipedia.org/wiki/CSS) markup.

ASP.NET is not truly cross-browser compatible. Repositioning of pages requires features only available in Internet Explorer.

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=20)] Development tools

Several available software packages exist for developing ASP.NET applications:

* [Delphi 2006](http://en.wikipedia.org/wiki/Delphi_programming_language)
* Macromedia Dreamweaver MX, [Macromedia Dreamweaver MX 2004](http://en.wikipedia.org/wiki/Macromedia_Dreamweaver_MX_2004), or [Macromedia Dreamweaver 8](http://en.wikipedia.org/wiki/Macromedia_Dreamweaver_8) (doesn't support ASP.NET 2.0 features, and produces very inefficient code for ASP.NET 1.x: also, code generation and ASP.NET features support through version 8.0.1 was little if any changed from version MX: version 8.0.2 does add changes to improve security against SQL injection attacks)
* [Macromedia HomeSite](http://en.wikipedia.org/wiki/Macromedia_HomeSite) 5.5 (For ASP Tags)
* [Microsoft Expression Web](http://en.wikipedia.org/wiki/Microsoft_Expression_Web), part of the [Microsoft Expression Studio](http://en.wikipedia.org/wiki/Microsoft_Expression_Studio) application suite.
* [Microsoft SharePoint Designer](http://en.wikipedia.org/wiki/Microsoft_SharePoint_Designer)
* [MonoDevelop](http://en.wikipedia.org/wiki/MonoDevelop) (Free/[Open Source](http://en.wikipedia.org/wiki/Open_Source))
* [SharpDevelop](http://en.wikipedia.org/wiki/SharpDevelop) (Free/[Open Source](http://en.wikipedia.org/wiki/Open_Source))
* [Visual Studio .NET](http://en.wikipedia.org/wiki/Visual_Studio_.NET) (for ASP.NET 1.x)
* [Visual Web Developer 2005 Express Edition](http://en.wikipedia.org/wiki/Visual_Web_Developer_Express_Edition) (free) or [Visual Studio 2005](http://en.wikipedia.org/wiki/Visual_Studio_2005) (for ASP.NET 2.0)
* [Visual Web Developer 2008 Express Edition](http://en.wikipedia.org/wiki/Visual_Web_Developer_Express_Edition) (free) or [Visual Studio 2008](http://en.wikipedia.org/wiki/Visual_Studio_2008) (for ASP.NET 2.0/3.5)[[19]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-18#cite_note-18)
* [Eiffel for ASP.NET](http://www.eiffel.com/downloads/asp.net.html) (free)

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=21)] Frameworks

It is not essential to use the standard webforms development model when developing with ASP.NET. Noteworthy frameworks designed for the platform include:

* [Castle Monorail](http://en.wikipedia.org/wiki/Monorail_(.Net)), an open-source [MVC](http://en.wikipedia.org/wiki/Model-view-controller) framework with an execution model similar to [Ruby on Rails](http://en.wikipedia.org/wiki/Ruby_on_Rails). The framework is commonly used with [Castle ActiveRecord](http://en.wikipedia.org/wiki/Castle_ActiveRecord), an ORM layer built on [NHibernate](http://en.wikipedia.org/wiki/NHibernate).
* Spring.NET, a port of the Spring framework for Java.
* Skaffold.NET, A simple framework for .NET applications, used in enterprise applications.

## [[edit](http://en.wikipedia.org/w/index.php?title=ASP.NET&action=edit&section=22)] Versions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Remarks** | **New features** |
| January 16, 2002 | 1.0 | First version  released together with [Visual Studio .NET](http://en.wikipedia.org/wiki/Visual_Studio_.NET) | * [Object oriented](http://en.wikipedia.org/wiki/Object_oriented) web application development supporting [Inheritance](http://en.wikipedia.org/wiki/Inheritance_(computer_science)), [Polymorphism](http://en.wikipedia.org/wiki/Polymorphism_in_object-oriented_programming) and other standard OOP features   + Developers are no longer forced to use Server.CreateObject(...), so early-binding and type safety are possible. * Based on [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) programming; the developer can make use of DLL class libraries and other features of the web server to build more robust applications that do more than simply rendering HTML ( i.e. [exception handling](http://en.wikipedia.org/wiki/Exception_handling) ) |
| April 24, 2003 | 1.1 | released together with [Windows Server 2003](http://en.wikipedia.org/wiki/Windows_Server_2003)  released together with [Visual Studio .NET 2003](http://en.wikipedia.org/wiki/Visual_Studio_.NET_2003) | * Mobile controls * Automatic input validation |
| November 7, 2005 | 2.0 | codename [Whidbey](http://en.wikipedia.org/wiki/Whidbey) released together with [Visual Studio 2005](http://en.wikipedia.org/wiki/Visual_Studio_2005) and [Visual Web Developer Express](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Express) and [SQL Server 2005](http://en.wikipedia.org/wiki/Microsoft_SQL_Server) | * New data controls (GridView, FormView, DetailsView) * New technique for declarative data access (SqlDataSource, ObjectDataSource, XmlDataSource controls) * Navigation controls * [Master pages](http://en.wikipedia.org/wiki/ASP_master_pages) * Login controls * Themes * Skins * Web parts * Personalization services * Full pre-compilation * New localization technique * Support for 64-bit processors * Provider class model |
| November 19, 2007 | 3.5 | Released with [Visual Studio 2008](http://en.wikipedia.org/wiki/Visual_Studio_2008) and [Windows Server 2008](http://en.wikipedia.org/wiki/Windows_Server_2008) | * New data controls (ListView, DataPager) * [ASP.NET AJAX](http://en.wikipedia.org/wiki/ASP.NET_AJAX) included as part of the framework * Support for [LINQ](http://en.wikipedia.org/wiki/Language_Integrated_Query) with a new LinqDataSource control |
| August 11, 2008 | 3.5 Service Pack 1[[20]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-19#cite_note-19) | Released with Visual Studio 2008 Service Pack 1 | * Incorporation of [ASP.NET Dynamic Data](http://en.wikipedia.org/wiki/ASP.NET_Dynamic_Data) * Support for controlling browser history in an ASP.NET AJAX application * Capability to combine multiple Javascript files into a single file for more efficient downloading |