**What is ASP.Net?**

ASP.Net is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web application for PC, as well as mobile devices.ASP.Net works on top of the HTTP protocol and uses the HTTP commands and policies to set a browser-to-server two-way communication and cooperation.ASP.Net is a part of Microsoft .Net platform. ASP.Net applications are complied codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

The ASP.Net application codes could be written in either of the following languages:

* C#
* Visual Basic .Net
* Jscript
* J#

ASP.Net is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls like text boxes, buttons and labels for assembling, configuring and manipulating code to create HTML pages.

**ASP.Net Web Forms Model:**

ASP.Net web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for stateful processing. The server processes the output of the client actions and triggers the reactions.

Now, HTTP is a stateless protocol. ASP.Net framework helps in storing the information regarding the state of the application, which consists of:

* Page state
* Session state

The page state is the state of the client, i.e., the content of various input fields in the web form. The session state is the collective obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take up an example of a shopping cart as follows.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown in a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.Net session state and server side infrastructure keeps track of the information collected globally over a session.

The ASP.Net runtime carries the page state to and from the server across page requests while generating the ASP.Net runtime codes and incorporates the state of the server side components in hidden fields.

This way the server becomes aware of the overall application state and operates in a two-tiered connected way.

**ASP.Net Component Model:**

The ASP.Net component model provides various building blocks of ASP.Net pages. Basically it is an object model, which describes:

* Server side counterparts of almost all HTML elements or tags, like <form> and <input>.
* Server controls, which help in developing complex user-interface for example the Calendar control or the Gridview control.

ASP.Net is a technology, which works on the .Net framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.Net web application is made of pages. When a user requests an ASP.Net page, the IIS delegates the processing of the page to the ASP.Net runtime system.

The ASP.Net runtime transforms the .aspx page into an instance of a class, which inherits from the base class Page of the .Net framework. Therefore, each ASP.Net page is an object and all its components i.e., the server-side controls are also objects.

**Components of .Net Framework 3.5**

Before going to the next session on Visual Studio.Net, let us look at the various components of the .Net framework 3.5. The following table describes the components of the .Net framework 3.5 and the job they perform:

|  |
| --- |
| **Components and their Description** |
| **(1) Common Language Runtime or CLR** It performs memory management, exception handling, debugging, security checking, thread execution, code execution, code safety, verification and compilation.Those codes which are directly managed by the CLR are called the managed code. When the managed code is compiled, the compiler converts the source code into a CPU independent intermediate language (IL) code. A Just in time compiler (JIT) compiles the IL code into native code, which is CPU specific. |
| **(2) .Net Framework Class Library** It contains a huge library of reusable types . classes, interfaces, structures and enumerated values, which are collectively called types. |
| **(3) Common Language Specification** It contains the specifications for the .Net supported languages and implementation of language integration. |
| **(4) Common Type System** It provides guidelines for declaring, using and managing types at runtime, and cross-language communication. |
| **Metadata and Assemblies** Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory. Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code and set of resources like image files etc. |
| **(5) Windows Forms** This contains the graphical representation of any window displayed in the application. |
| **(6) ASP.Net and ASP.Net AJAX** ASP.Net is the web development model and AJAX is an extension of ASP.Net for developing and implementing AJAX functionality. ASP.Net AJAX contains the components that allow the developer to update data on a website without a complete reload of the page. |
| **(7) ADO.Net** It is the technology used for working with data and databases. It provides accesses to data sources like SQL server, OLE DB, XML etc. The ADO .Net allows connection to data sources for retrieving, manipulating and updating data. |
| **(8) Windows Workflow Foundation (WF)** It helps in building workflow based applications in Windows. It contains activities, workflow runtime, workflow designer and a rules engine. |
| **(9)Windows Presentation Foundation** It provides a separation between the user interface and the business logic. It helps in developing visually stunning interfaces using documents, media, two and three dimensional graphics, animations and more. |
| **(10) Windows Communication Foundation (WCF)** It is the technology used for building and running connected systems. |
| **(11) Windows CardSpace** It provides safety of accessing resources and sharing personal information on the internet. |
| **(12) LINQ** It imparts data querying capabilities to .Net languages using a syntax which is similar to the tradition query language SQL. |

**Structure of an ASP.NET Program**

An ASP.Net page is made of number of server controls along with the HTML controls, text and images. Sensitive data from the page and the states of different controls on the page are stored in hidden fields and forms the context of that page request.

ASP.Net runtime controls all association between a page instance and its state. An ASP.Net page is an object of the Page Class or inherited from it.

All the controls on the pages are also objects of the related control class inherited from a parent Control class. When a page is run an instance of the page object is created along with all its content controls.

An ASP.Net page is also a server side file saved with the .aspx extension. It is modular in nature and can be divided into the following core sections:

* Page directives
* Code Section
* Page Layout

**Page directives:**

The page directives set up the environments for the page to run. The @Page directive defines page-specific attributes used by the ASP.Net page parser and compiler. Page directives specify how the page should be processed, and which assumptions are to be taken about the page.

It allows importing namespaces, loading assemblies and registering new controls with custom tag names and namespace prefixes. We will discuss all of these concepts in due time.

**Code Section:**

The code section provides the handlers for the page and control events along with other functions required. We mentioned that, ASP.Net follows an object model. Now, these objects raises events when something happens on the user interface, like a user clicks a button or moves the cursor. How these events should be handled? That code is provided in the event handlers of the controls, which are nothing but functions bound to the controls.

The code section or the code behind file provides all these event handler routines, and other functions used by the developer. The page code could be precompiled and deployed in the form of a binary assembly.

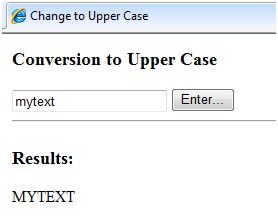
**Page Layout:**

The page layout provides the interface of the page. It contains the server controls, text, inline JavaScript and HTML tags:

The following code snippet provides a sample ASP.Net page explaining pafe directives, code section and page layout written in C#:

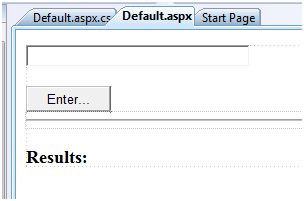
|  |
| --- |
| <!-- directives -->  <% @Page Language="C#" %>  <!-- code section -->  <script runat="server">  private void convertoupper(object sender, EventArgs e)  {  string str = mytext.Value;  changed\_text.InnerHtml = str.ToUpper();  }  </script>  <!-- Layout -->  <html>  <head> <title> Change to Upper Case </title> </head>  <body>  <h3> Conversion to Upper Case </h3>  <form runat="server">  <input runat="server" id="mytext" type="text" />  <input runat="server" id="button1" type="submit"  value="Enter..." OnServerClick="convertoupper"/>  <hr />  <h3> Results: </h3>  <span runat="server" id="changed\_text" />  </form>  </body>  </html> |

Copy this file to the web server's root directory. Generally it is c:\inetput\wwwroot. Open the file from the browser to run it and it should generate following result:



Using Visual Studio IDE:

Let us develop the same example using Visual Studio IDE. Instead of typing the code, you can just drag the controls into the design view:



The content file is automatically developed. All you need to add is the Button1\_Click routine, which is as follows:

|  |
| --- |
| protected void Button1\_Click(object sender, EventArgs e)  {  string buf = TextBox1.Text;  changed\_text.InnerHtml = buf.ToUpper();  } |

The content file code is:

|  |
| --- |
| <%@ Page Language="C#" AutoEventWireup="true"  CodeBehind="Default.aspx.cs"  Inherits="firstexample.\_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>  <asp:TextBox ID="TextBox1" runat="server" Width="224px">  </asp:TextBox>  <br />  <br />  <asp:Button ID="Button1" runat="server" Text="Enter..."  Width="85px" onclick="Button1\_Click" />  <hr />  <h3> Results: </h3>  <span runat="server" id="changed\_text" />  </div>  </form>  </body>  </html> |

Run the example either from Debug menu, or by pressing Ctrl-F5 or by right clicking on the design view and choosing 'View in Browser' from the popup menu. This should generate following result:

