# 1.WER SERVER:

## ➢ What is a web server?

A Web server is a software program which serves web pages to web users (browsers).

A web server delivers requested web pages to users who enter the URL in a web browser. Every computer on the Internet that contains a web site must have a web server program.

The computer in which a web server program runs is also usually called a "web server". So, the term "web server" is used to represent both the server program and the computer in which the server program runs.

## ➢ Characteristics of web servers

A web server computer is just like any other computer. The basic characteristics of web servers are:

- It is always connected to the internet so that clients can access the web pages hosted by the web server.
- It has an application called 'web server' running always.

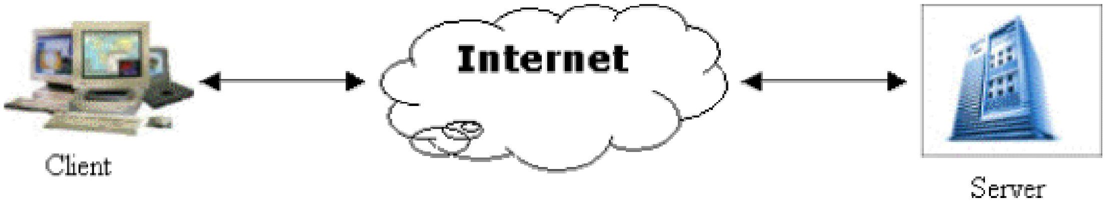
In short, a 'web server' is a computer which is connected to the internet/intranet and has a software called 'web server'. The web server program will be always running in the computer. When any user try to access a website hosted by the web server, it is actually the web server program which delivers the web page which client asks for.

All web sites in the internet are hosted in some web servers sitting in different parts of the world.

### Web Server is a hardware or a software ?

From the above definition, you must have landed up in confusion “Web server is a hardware or a software”

Mostly, Web server refers to the software program, which serves the clients request. But as we mentioned earlier in this chapter, the computer in which the web server program is also called 'web server".



### Web Server – Behind the Scene

Now that you are reading this page, have you ever had a thought how the page is made available to the browser?

Your answer would be, “I typed in the URL http://www.aspspider.com and clicked on some link, I dropped into this page.”

But what happed behind the scenes to bring you to this page and make you read this line of text.

So now, lets see what is actually happening behind the scene. The first you did is, you typed the URL http://www.aspspider.com in the address bar of your browser and pressed your return key.

We could break this URL into two parts,

1. The protocol we are going to use to connect to the server (http)
2. The server name (www.aspspider.com)

The browser breaks up the URL into these parts and then it tries to communicate with the server looking up for the server name. Actually, server is identified through an IP address but the alias for the IP address is maintained in the DNS Server or the Naming server. The browser looks up these naming servers, identifies the IP address of the server requested and gets the site and gets the HTML tags for the web page. Finally it displays the HTML Content in the browser.

## ➢ Where is my web server?

When you try to access a web site, you don't really need to know where the web server is located. The web server may be located in another city or country, but all you need to do is, type the URL of the web site you want to access in a web browser. The web browser will send this information to the internet and find the web server. Once the web server is located, it will request the specific web page from the webserver program running in the server. Web server program will process your request and send the resulting web page to your browser. It is the responsibility of your browser to format and display the webpage to you.

## ➢ How many web servers are needed for a web site?

Typically, there is only one web server required for a web site. But large web sites like Yahoo, Google, MSN etc will have millions of visitors every minute. One computer cannot process such huge numbers of requests. So, they will have hundreds of servers deployed in different parts of the world so that can provide a faster response.

## ➢ How many websites can be hosted in one server?

A web server can hosted hundreds of web sites. Most of the small web sites in the internet are hosted on shared web servers. There are several web hosting companies who offer shared web hosting. If you buy a shared web hosting from a web hosting company, they will host your web site in their web server along with several other web sites for a Fee.
Examples of web server applications
1. IIS
2. Apache

# 2.WEB BROWSER

### *Definitions for browser*

- Browser is an application, which helps us to view the Web sites and the web content.
- Web sites are located in some remote systems, which needs a special kind of program or an application to access them, such an application is called **browser**.
- A browser is an application which you can use to retrieve web pages from web sites and view.
- When you type a URL in the browser, the browser will convert it into a *web request* which web server can understand. Browser will send the request to web server using the HTTP protocol.
- When a web server returns a web page as a *Response*, the browser will understand the response and display the body of the response to the user in browser.

- In simple terms a browser can be defined as “A software application used to locate and display Web pages”
- You can create your own simple browser application using C++, C#, VB.NET or any other language you like. All you need to know is, how to compose a request which web server can understand, how to parse and display the response from web server and communicate with webserver using HTTP protocol.
- Even though the basic job of browser is just send requests to web server and receive response from the server, modern browser provide several other enhanced features including Back/Forward buttons, save viewed files to disk so that they can be viewed later, cache images so that the same images need not be downloaded again and again etc.
- **Internet Explorer, Netspace, Mozilla** are some of the popular web browsers currently available in the market.
- Netscape was the most popular web browser till 4-5 years back, but currently more than 90% of the internet users use "Internet Explorer" to browse the websites.

# 3.Windows Application and Web Application

Developers are often confused about the difference between Windows Applications and Web Applications. This chapter gives brief information about various types of computer applications and provides an introductions to web applications.

## ➢ Different types of computer applications

It is very hard to divide applications into any strict categories. There is no clear definition exists to categorize computer applications. However, here is a small list of different types I can think of:

1. Embedded Systems
2. Windows applications (also called 'Desktop applications')
3. Web Applications

## ➢ Embedded Systems

Have you ever used a digital diary (also called 'digital organiser') or a mobile phone ? Do you know when you save a name and address, how does it get saved in it?

It is a small computer program 'Embedded' in that device. It is similar to any small address book computer program that you can write using your favorite visual basic or c++. The only difference is, it is written using soem special language and 'embedded' into a chip in the device inside the mobile phone or digital diary.

If you find a bug in a regular computer program, you can simply write a better program, compile it and copy to your computer. But if you find a bug in your mobile phone's embedded program, you cannot copy it! The manufacturer has to embed the new program in a new chip and replace the chip in your mobile phone!!

## ➢ Windows Applications

If you don't know what is a 'Windows Application', probably you have never seen a computer. Almost any application you see on a desktop computer is called 'Windows Application'.

It is also called 'desktop applications' since they are mostly used in desktop computers.
Some common examples of desktop applications are:

1. Paint Brush program
2. Calculator program

3. MSN Messenger
4. Yahoo Messenger
The first three windows applications are written by some programmers sitting in Microsoft office and they give it free to all who buy Windows operating system. The Yahoo messenger is written by Yahoo programmers and they give it free to download from their web site.

If your neighbour ask you to write small 'Address book' application for his personal use, you are going to write a 'windows application'.

Now you must have a good idea of what is windows application.

## ➢ Web Applications

I am sure you have seen atleast one web application! Do you know how I guessed it?

It was easy to guess. This tutorial is a web application and you are currently reading this tutorial from our web site (unless you copied it to somewhere...)

### So, what is a web application ?

A web application is also called 'web site'. A web site is a collection of web pages hosted on a special computer called 'web server'.

Now you are reading this tutorial. This chapter is a page among several other pages part of our web application. The name of our web application is 'AspSpider.com'. This web site (web application) is running in our web server, which is located in a safe place in USA. You are a 'visitor' to our site and you are accessing our web application using a tool called 'Internet Explorer' (or, some other browser like Netscape etc). We don't know where you are (we have several ways to find it, which we will explain in some other chapter)

So, here is some interesting points about a web application:

- A web application is a collection of web pages.
- A web application needs a web server to run.
- Web server can be located anywhere and visitors need not be even in the same country of the web server.
- Visitors can access the web application using a tool called 'browser'. There are many browsers exists. Most widely used browser is 'Internet Explorer'. This is provided by Microsoft and it is free. Another famous free browser is 'Netscape'.

# 4. Infrastructure to Develop a Web Application

Any one who want to develop a web application must have the following systems:

1. A web server.
2. An editor to develop the web pages.
3. A browser to view the web page you develop.
4. A database program like MS Access, SQL Server etc, if your web site need to save data into a database.

In the real world situation, a web server will be hosted on a secure server, located in a safe place and will be always connected to high speed internet. However, to develop a web application, you don't need to worry about security and internet connectivity. You can use your own development computer as the 'Web Server'.

So, your development computer must have all the 4 systems mentioned above.

Since this tutorial is guiding you to develop web sites using ASP.NET, all the tools we are going to talk here will be the ASP.NET specific tools.

### ➢ Web Server

There are several types of web servers. But if you like to develop ASP.NET web applications, you need a specific web server called 'Internet Information Server' (IIS).

IIS comes as part of Windows. But it is not installed by default, when you install Windows. Please see the chapter 'Installing IIS' to find more about installing IIS.

### ➢ Editor to develop web pages

Ideally, you do not need any special editor to develop a web application. If you are an expert, you can simply use ntoepad to type HTML and the code for the web pages. However, who want to hand-wash the vessels when there is a dish washer ?!

You don't need to make your hands dirty! Microsoft gives a tool called 'Visual Studio .NET' to edit web pages and write code for ASP.NET.

### ➢ Visual Studio .NET (VS.NET)

Visual Studio .NET allows to easily create web pages. Some of the benefits in using Visual Studio .NEt are:

- You can simply drag and drop html controls to the web page and VS.NET will automatically write the HTML tags for you.
- Start typing an HTML tag and VS.NET will complete it! When you start typing a tag, VS.NET will show you the HTML tags starting with the characters you typed. So, you don't need to even remember all the tags.
- If you type any HTML tags wrong, VS.NET will highlight the errors and tell you how to correct it.

So, even if you are not an expert, VS.NET can help you develop great web pages.

### ➢ Browser

You need a browser to view the web pages you create. If you have any windows operating system in your computer, you will already have a free browser (called 'Internet Explorer')

### ➢ Database program

A database program like MS Access or SQL Server is required only if you need to save data into database. It is not mandatory that all web sites need a database program.

## 5. Installing & Configuring IIS

### ➢ Internet Information Services (IIS):-

If you have Windows 2000 or Windows XP Professional, you may have IIS already installed in your computer. IIS is part of the Windows operating system, but it is not selected by default. So, if you choose default options when you installed the operating system, you may not have the IIS installed.

To find if you already have IIS installed, go to the "Control Panel" and select "Administrative Tools". Check if there is a shortcut for "Internet Information Services" exists as part of the "Administrative Tools". If it does not exist, you will need to install IIS.

To install IIS, go to the "Control Panel" and select "Add Remove Programs". Select the option "Add/Remove Windows Components". Make sure the option "Internet Information Services (IIS)" is selected. If it is already selected, that means you already have IIS ! If the checkbox for IIS is not checked, select it and press the "Next" button to proceed to install.

The Windows Components Installation Wizard may prompt you to insert the CD for WIndows. So, keep the Windows CD handy. Or, if you have the I386 folder copied in your hard drive, you can use that when you are prompted to insert the CD.

## 6. Working With IIS And Virtual Directory

Internet Information Services (IIS) helps you manage the web applications and web sites in your computer.

### ➢ How to open IIS

There are several ways you can open the Internet Information Services tool.

Method 1: Open the 'Control Panel', select 'Administrative Tools' and then double click the shortcut 'Internet Information Services'

Method 2: Go to the "start" menu, select "Run" and then type "inetmgr". Press "OK" to open the IIS tool.

The IIS admin tool look like this:

In the above image, the name "MANJU" represents the computer name. Under the node "Web Sites", it displays all the web sites. When you install IIS, it creates a default web site for you. Under the "Default Web Site" node, you can see all virtual directories listed.

Only on server operating systems like 'Windows 2000 Server' or 'Windows 2003 Server', you can create multiple Web Sites. In other systems like Windows XP, Windows 2000 Professional etc, you can have only 1 web site (which is the 'Default Web Site' created by the system). But you can have any number of web applications under this default web site. Each web application will need to be created as a 'virtual directory' under any one of the 'web sites'. If you do not have a server operating system, you have to always create your web applications under the 'Default Web Site'.

Each web application you create in your machine needs a virtual directory. In the above image, we have two web applications (named 'WebApplication1' and 'WebApplication2')

When you create a new ASP.NET project using Visual Studio .NET, it will automatically create a virtual directory for you.

For example, if you create a new ASP.NET project using Visual Studio .NET called "Shopping Cart", it will create a new virtual directory with the name "ShoppingCart". After you create the ASP.NET project, you can see the virtual directory listed in the IIS.

You can right click on the virtual directory name in IIS and select 'Properties' to view various properties of the virtual directory. One important property is "Local Path". The "Local Path" property represents the actual location of the web application.

By default, when you create a new ASP.NET project, visual studio creates the project under the folder "C:\Inetpub\WWWRoot". For example, if you create a new ASP.NET project called "Shop Cart", VS.NET creates a folder called "C:\Inetpub\WWWRoot\ShopCart" and all files related to the project will be placed inside this folder. This folder will be set as a "Virtual Folder" so that you can access the web site using the URL "http://localhost/ShopCart".

If you are working on several projects, you may want to organize your projects in some specific folder instead of "C:\Inetpub\WWWRoot". What you need to do is, create a virtual folder manually before you create the project.

For example, if you need to create an ASP.NET application called "Shop cart" under the folder "C:\MyProjects", first create the folder "C:\MyProjects\Shopcart". Now convert this folder as a virtual folder. Now you are ready to create the project using VS.NET. If the virtual folder already exists, VS.NET will NOT create a new virtual folder under the WWWRoot folder. It will use the existing virtual folder.

### ➢ What is "localhost"?

"localhost" represents "current machine". If you are accessing any web applications in your own machine, you can use the url http://localhost/XXXXXXX. If your web site is in another machine, you can use the IP Address of the computer instead of "localhost". For example, if the "Shopcart" web application is in a computer with the IP Address "128.32.65.21", then you can access the site from your computer using the url "http://128.32.65.21/Shopcart".

### ➢ Stop and start IIS

You may never need to stop and start your IIS. When your computer is started, IIS is automatically started.

Actually, IIS is a 'windows service'. What you see as 'Internet Information Services' is an admin tool provided by Microsoft to manage the real IIS which works in the background. IIS runs as a windows service in the background and what you see is only a tool to manage the IIS service.

If you like to stop or start IIS, select the "Default Web Site" in IIS tool and select "stop" or "start" from the right click menu. If you stop IIS, you cannot access your web sites anymore. Even, if you try to open the ASP.NET project in Visual Studio, it will complain that web site is currently not available.

## 7. Virtual Directory

### ➢ What is a virtual directory?

A virtual directory represents a web application and it points to a physical folder in your computer.

A web application is accessed using a virtual directory name instead of a physical folder name. For example, if you have a web application called "Shopcart" in your machine, you will have a virtual directory for this web application. You will access your web application using the URL httP://localhost/Shopcart. If your virtaul directory name is "Test", then your web application url will be "http://localhost/Test".

Assume you have a web application called "Shopcart", created under the physical folder

"C:\MyProjects\Shopcart".

You can go to IIS and see this virtual directory listed. Right click on this virtual directory name in IIS and see the properties. You can see that this virtual directory is pointing to the physical location "C:\MyProjects\Shopcart".

If you have a file called "File1.aspx" under the folder "C:\MyProjects\Shopcart\", then you can access this file using Internet Explorer with the URL "http://localhost/Shopcart/File1.aspx"

### ➢ How to create a virtual directory?

When you create a new web project using, a new virtual directory will be created automatically for you. This virtual directory will point to a new folder created under C:\Inetpub\wwwroot.

If you like to better organize your projects and files in your favorite folder, you must manually create a new folder for each project in your preferred location and convert it into a virtual folder manually.

There are couple of ways you can do this.

Method 1: Open the IIS. Right click on the node "Default Web Site" and select "new Virtual Directory". When it prompt you to enter the "alias", enter the virtual directory name you want (E.g.: Shopcart). In the prompt for "directory", select the folder which you want to make a virtual directory (E.g.: C:\MyProjects\Shopcart). Select other default values and press "Finish". Now you should be able to see your new virtual directory in IIS.

Method 2: In the explorer, go to the folder (E.g.: C:\MyProjects\Shopcart) which you want to make a "virtual directory". Right click on the folder name and select "Properties". Select the tab "Web sharing" and select the option "Share this folder". It will prompt you with a default Alias name same as the folder name (E.g.: Shop cart). Simply select the default values and press "OK".

## 8. How Web Application Work

Web applications work quite different from regular windows applications. There are several computers involved when you view a web page.

### ➢ Life Cycle of a Web Request

Viewing a web page is a simple process for a visitor. Just type the URL in a web browser like Internet Explorer or click on a hyper link in any existing web page. The web browser will display the page instantly to you.

But do you know that there are several computers involved in this process? Even though it is a very complex process, we can summarize the process as shown below:

1. You type the web page address (URL) in a browser. For example, consider the current page http://www.aspspider.com/tutorials.aspx. This URL has 3 parts:

- The protocol - http:
- The server name - www.aspspider.com
- The file name - tutorials.aspx

2. Browser communicates with a computer in internet called 'Domain Name Server' to find out the IP Address of the server (Eg: www.aspspider.com).

3. Browser established a connection to the web server at that IP Address.

4. Server composes a 'Request' for the specified URL and sends the request to the web server to which it has established a connection.

5. The web server identifies the type of the page requested. If it is an asp.net web page, then browser knows that needs some processing by the asp.net service running as part of the web server. The request is handed over to the asp.net service. The asp.net service processes the asp.net page and generates the html output.

6. Web server sends the requested page to the browser.

7. When a response is received by the browser, it displays the web page to the user who typed the URL.

After you typed the URL in the browser, the request sent by the browser may go through several computers in the internet before it reaches the actual web server.

You must be surprised to know that so many things happen and several computers are involved before a simple web page is displayed to you. Most of the steps in the above process happens behind the screen. A visitor need not worry about how a web page is processed and served to the browser.

So, by now you must have got a better picture about how a windows application is different from a web application. When you run a windows application (desktop application), only one computer is involved in the whole process. You start an application in your computer and it runs in your computer. But when you request a web page from your home computer, the request goes through several servers in the internet and finally it reaches a computer called 'Web Server'. The actual web page is processed in the web server.

# 9. ASP.NET

Many new programmers are confused about how ASP.NET is different from C# or VB.NET. Many new programmers’ misunderstand that ASP.NET is just another programming language.

## ➢ What is ASP.NET?

- ASP.NET is the name of the Microsoft technology used for web site development.
- ASP.NET is NOT a programming language like C# or VB.NET
- ASP.NET development requires a programming language like C# or VB.NET to write code.
- ASP stands for Active Server Pages.
- There are several other technologies exist for web development (Eg: PHP). ASP.NET is the technology from Microsoft and it he widely used one.
- ASP.NET technology comes with a rich set of components and controls that make the web development very easy.
- Visual Studio .NET is the editor from Microsoft which helps you develop ASP.NET web sites faster and easily.
- IIS is the web server from Microsoft which supports ASP.NET. To develop ASP.NET web sites, you must have IIS installed in your computer.

In ASP.NET programming, a web page is developed using HTML and a .NET programming language like C#, VB.NET or J#. You can choose your favourite .NET language to develop ASP.NET pages.

So, now you must be clear that ASP.NET is not a programming language and it requires a language like C# or VB.NET to develop ASP.NET web sites.

You must have read in the earlier chapters that ASP.NET is a technology which includes several other technologies.

## ➢ What you should know to become an ASP.NET web developer?

So, unlike C# or VB.NET windows programmers who need to know only the programming language, an ASP.NET developer must learn several things.

### ➢ HTML

HTML is a very simple language used to develop web pages. It may not take more than a couple of days to to become a reasonable good HTML programmer.

### ➢ Programming language - C#, VB.NET or your favorite language

You can develop a simple, static web site using plain HTML. You don't need to know C# or VB.NET to develop a web site.

But, to display dynamic content in the ASP.NET website, you must use a programming language like C# or VB.NET. For exampe, if you want to display data from database, you will need to write C# or VB.NET code in the web page to access the database and retrieve database.

So, you must learn atleast one programming language like C# or VB.NET to become a good ASP.NET developer.

### ➢ Client side scripting language

In addition to HTML and C#/VB.NET, you may need to learn one more programming language like Javascript or Vb script for client side scripting. Javascript is the most popular and widely used client side scripting language. You can read more about client side scripting in another chapter.

### ➢ Internet Information Services - IIS

IIS is the engine that drives the ASP.NET web sites. But just like a car driver need to know much about the engine of the car, you need not be an IIS expert to develop web sites. Just like your car starts when you turn the key, the IIS is started when you start windows. There are separate chapters for IIS in this tutorials.

### ➢ Visual Studio .NET - VS.NET

Visual Studio .NET is the editor from Microsoft which allows you to write code easily. Ideally, you do not need VS.NET to develop ASP.NET web sites. You can simply use notepad to type your code and use command line options to compile the code.

But you may not want to use notepad to write web pages, if you want to finish your learning in the next couple of years!!

VS.NET makes the life of a programmer very easy. It does lot of work in the backgroud for you. When you create a new web project, it automatically creates a virtual directory for you in the IIS. If you use notepad to develop web site, you will have to manually create virtual folder in IIS, which requires some knowledge about IIS. Also, it is 1-click operation in VS.NET to compile and build all web pages. In addition, VS.NET creates lot of default files and code for you, which you will have to do manually if you are not using VS.NET. If you want to display a text box or a button control, just drag and drop the control in VS.NET. There is no need to type even a single line!!

# 10. Create First Web Page

Visual Studio .NET is tool that help you automate lot of work related to creating a web site or web application. However, in this chapter we are going to see how to create a small web site manually, without using VS.NET. This help you understand some of the background work done VS.NET when you create a new project.

Let us create a new web site called "ShoppingCart".

**Step 1: Create folder**

The first step is, creating a folder for this web site. So, go to explorer and create the following folder:

C:\ShoppingCart

**Step 2: Create virtual directory**

Create a new virtual directory called "ShoppingCart" pointing to the folder we just created. Refer to this chapter for more information about creating virtual directories and working with IIS.

**Step 3: Create a web page**

Let us create a simple page to get started. Open notepad and copy the following content:

```
<html>

<head>
<title>This is my first web page</title>
</head>

<body>

<br><br><br><br>

<center>

<font size=5 color=red>Welcome to SpiderShop !</font>

<BR><BR>

<font size=3 color=darkgreen>Buy softwares and tools online at very low
rates....</font>

</center>

</body>
</html>
```

Save the above text into a file called "index.html" into the folder "C:\ShoppingCart".

You are done with your first web site. Since you have configured your folder as a virtual directory, you can access all html files in that folder using a web browser.

Open internet explorer and type the URL

```
http://localhost/ShoppingCart/index.html
```

You can see the following output in Internet Explorer:

Welcome to SpiderShop !

Buy softwares and tools online at very low rates....

Try to alter the html in the file and see how it appears in internet explorer. Also, you can create more html files in the same folder (like index2.html, AboutMe.html etc) and access them using the url http://localhost/FileName (Eg: http://localhost/ShoppingCart/AboutMe.html)

If your file is already open in Internet Explorer (IE) and you make some changes in the html file, the changes will not be visible in IE until you refresh the page in IE. Just hit F5 button in IE to reload the page after you make any changes to the file using notepad.

## 11. Moving From HTML To ASP.NET

In this chapter, you learned how to create a simple website with an html page. Our aim is to learn ASP.NET. So, let us see how to develop ASP.NET pages.

An html file can have the extensions **.htm** or **.html**. An ASP.NET page has the extension **.aspx**. So, the very basic difference between an html page and an ASP.NET page is, they have different file extensions.

By looking at the file extension, the web server understands what type of file it is and decides how to process them. An html file and ASP.NET need to be processed differently by the server.

We said the basic difference is, html and ASP.NET pages have different extensions. So, you may ask "what if simply change the extension of the file?"

You are right. You can simply change the extension of an html file to make it an ASP.NET page!

In an earlier chapter, you created a simple web site called "ShoppingCart" with an html page "index.html". Go to the folder "C:\ShoppingCart" and rename the file to "index.aspx". You have created your first ASP.NET page!

### ➢ How to test the ASP.NET page?

Just like the way you opened your html page in Internet Explorer, open the ASP.NET page. Open Internet Explorer and type the following URL:

```
http://localhost/index.aspx
```

You can see the same output as the html file.

### ➢ How did it work?

If you want to change your small grocery store to a big super market, what will you do? You can simply put a big board in front of your store saying "Super Market" and claim that you have a super market.

This is exactly what we did with our html page. We simply renamed it. But changing the name is not enough to convert a grocery store to a super market. There is a lot more to go!

In the coming chapters, you will learn more about how to convert a grocery store into a super market (well, we mean how to switch from simple html to complex asp.net pages)

# 12. How Browser Understand ASP.NET Webpage

There are several several technologies exist to develop dynamic web sites and pages. Some of them are ASP, ASP.NET, PHP etc.

Whatever technology you use to develop the dynamic web pages, a standard browser should be able to display the page to the visitors.

## ➢ How browsers understand ASP.NET or PHP pages?

If visit any web site, no matter what technology is used to develop the site, your browser will be able to display the page for you.

The only thing a browser can understand is "HTML". It does not know ASP.NET or PHP. So, even if your web site is developed using ASP.NET, still your browser can understand only HTML.

This is how it works:

You type a URL in your browser. (Eg: http://www.aspspider.com/tutorials/Tutorials.aspx?TutorialId=61)

Your browser will compose a request for this page and send to the web server in internet.

The web server analyzes the request and it understands that the request is for an ASP.NET page called "Tutorials.aspx". So, the web server hand over the request to the aspnet service running as part of the web server. (If the page is a .php file, then there must be a php service running on the webserver).

The aspnet service loads the page "Tutorials.aspx". Inside this page, we have written code to read the TutorialId passed as a parameter (parameters are called "Query String"). Our code gets this tutorial id and then retrieves the corresponding content from the database. Then our code embeds this content into the page and returns the dynamically modified page content to the web server.

Web server returns the dynamically generated page to the browser. This dynamically generated page has only HTML in it, even though this html came from database.

When the browser receives the page, it has only HTML. So, as far as a browser is concerned, it does not care what type of web site it is. It can be any technology like ASP.NET or PHP. It is the responsibility of the web server to generate dynamic content from database or wherever and give only HTML page content to the browser.

## ➢ Other than HTML, what else browsers understand?

We said that browsers understand only HTML. That is not fully true. Modern browsers understand something more than HTML, like Javascript, Flash etc. We will talk about these in the coming chapters.

# 13. How ASP.NET Converted To HTML

In an earlier chapter you learned that browsers can understand only HTML (and some client side scripts like javascript).

So, what does that mean? Does it mean you cannot develop ASP.NET pages to view in a browser?

This is the concept:

**You can use web development technologies like ASP.NET, PHP etc to develop web sites. But what is happening behind the scenes is, the above technologies are used to dynamically generate html. What is sent from web server to the browser is just plain html. That why browsers are able to display web pages developed using ASP.NET.**

Whatever technology you use to develop web pages, what is sent from server to browser is just plain html. Depdning on various conditions, ASP.NET generate different html. I will prove this with an example.

If you have NOT logged In to this site, you can see a link called "Login" in the top left corner. If you are logged in, you will see Welcome TONY. (You will see your name instead of TONY, but I am using my name for this example)

We use ASP.NET to determine wether user has logged in or not. If logged in, then our ASP.NET code will generate the following html:

```
<a href=ViewProfile.aspx?UserId=tony>Welcome TONY !</a>
```

If the user is NOT logged in, then our ASP.NET code will generate the following HTML:

```
<a href=Login.aspx>Login</a>
```

Did you notice that the above samples show plain html? Depending on various conditions, ASP.NET generate appropriate HTML. This is what sent to browsers. You can right click on in any web page and select View Source to see the actual HTML. Try to see the source html of this page.

Browsers always see plain html (the same html you see by Right click -< View Source) . They dont care how the html is generated, what is the technology or programming language (ASP or ASP.NET or PHP) used to generate those html etc. . Even if some other company comes up with a new technology for web development, all browsers will still work as long as those technologies generate proper html.

This is the fundamental concept behind web development. It is very important to understand this concept clearly before you proceed with ASP.NET programming.

# 14. Static and Dynamic Page

We can broadly classify web sites and web pages into two categories:

1. Static web pages
2. Dynamic web pages

### ➢ *Static Web Pages*

A static web page is a page which has the same content always.

In case of static web pages, content is written in the page itself as plain html. Until the author of the web page updates the content, the content remains the same in the static pages.

Static web pages are meant for providing information which does not change often. For example, visit http://www.google.com/intl/en/about.html. This page is a static page. The content is always the same (until they update the content by uploading a new html file to the web server).

HTML files are used to create static web pages.

### ➢ *Dynamic Web Pages*

Dynamic web pages get content from database. Content is NOT hard-coded in the page itself.

Dynamic pages are created using "serverside code" when the page is loaded every time.

An example for a dynamic page is this tutorial page itself. See the file name in the URL. The file name is "Tutorial8.aspx". We have used only one file to display any tutorial chapter. The chapter number is there as part of the URL (the chapter number is 8 for this chapter).

When you type the URL in the browser, this page is dynamically created from database, based on the chapter id in the URL. The content of each chapter is stored in our database, not in the file itself. When you access the page, our server side code will check what is the TutorialId in the URL. Based on the TutorialId, it will retrieve appropriate chapter content from the database and dynamically create the web page. (You can try to change the TutorialId in the URL to some very large number and try. The page will give an error because our code will fail to get the corresponding chapter content from the database). This entire site has only very few files (like index.aspx, Tutorial.aspx etc)

Dynamic web pages are created using technologies like ASP, ASP.NET, PHP etc.

HTML pages cannot be dynamic. All HTML files are static pages. If you want to write dynamic pages, you must use some technologies like ASP.NET.

# 15. Client Side Scripting and Server Side Scripting

Ever wondered the what is the difference between client side code and server side code? If your answer is YES, this chapter is for you.

### ➢ *Server side code*

In a previous chapter, you learned that dynamic pages are created using code in asp.net web pages. When you request a page, the web server executes the code in the web page and generates an HTML content for that page. It is this HTML content that is sent back to the browser so that it can be displayed to the user.

The code that is executed by the web server to generate the dynamic page is called "server side code". It is called "server side code" because it is executed by the web server.

When you develop ASP.NET pages, you will use C# or VB.NET (or any other .NET compatible code) code to write the server side code. Server side code is used to retrieve and generate content for the dynamic pages. You may be using the code to retrieve the content from database or something like that.

When a dynamic page is requested, the server side code is executed on the server and a page is generated. Once the generated page comes to the browser, there is connection with the server. You cannot do anything from the browser to communicate with the server, othen than requesting another page or same page again. So, if you want to access the database or something like that once the page is displayed, it is not possible.

### ➢ ***Client side code***

Client side code is used to do some kind of basic operations in the browser when the page is displayed to the user.

As you read above, it is not possible to access the server once the page is displayed. Morover, the browser does not knwo anything about ASP.NET or .NET. The browser undersands only HTML and client side scripting languages.

Client side coding is used to do basic operations on the browser. It cannot be used to access the server or database on the server etc.

Client side coding is done using scripting languages like Javascript, VbScript, JScript etc. These scripting languages are easy to learn (they are different from vb.net and C#). The scripting languages provide only very minimal functionality.

The main use of client side scripting is to validate user input before submitting a page to server. For example, suppose you have a "Registration" page. Once user enter all data and press the "submit" button, all the user input will be sent to server. In the server side, you may have written vb.net or C# code to validate all user inputs like Name cannot be empty etc. If the user do not enter a name and press submit, your server side code will generate an error message and return the page to you.

In this case, the page was sent to the server and came back with an error message. It was an expensive operation. It consumed lot of resources to send a page to the server and get it back to the browser with an error message.

If we can catch this validation error in the browser itself, instead of sending to server and coming back with an error, that will save lot of time and resources. User need not wait to send the page to server and come back.

Since, a browser does not know VB.NET or C#, you cannot write such code to validate it in the browser.

Client side scripting comes here to help. You can write Javascript, VBScript or JScript code to validate the name field in the browser itself and submit it to the server only if all user inputs are correct.

You can learn more about client side scripting in coming chapters.

## **15. First Dynamic Webpage**

In an earlier chapter, you learned how to create a small Shopping Cart web site with a simple HTML page in it. In another chapter you larned how to convert the simple html page to an ASP.NET page. That was fun! You simply renamed the html page to an aspx page.

In this chapter you are going to do some real work by writing server side code, which make it a real dynamic page.

To write server side code, you must choose a .NET language. It can be any .NET language like C#, VB.NET, C++.NET or J#. For this chapter, let us choose VB.NET.

Go back to your index.aspx page in your folder "C:\ShoppingCart\". Open the file using notepad. You can see the following content in the index.aspx file.

```
<html>

<head>
<title>This is my first web page</title>
</head>

<body>

<br><br><br><br>

<center>

<font size=5 color=red>Welcome to SpiderShop !</font>

<BR><BR>

<font size=3 color=darkgreen>Buy softwares and tools online at very low
rates....</font>

</center>

</body>
</html>
```

As you can see, it is plain HTML even though the page is an aspx page. Yes, you can have plain HTML inside an ASP.NET page. In this case, there is no server side code and the server has no processing to do, other than simply returning the HTML content as it is, to the browser. Only when there is some server side code is present in the code, the server will process it and return the output embedded in other HTML content.

## ➢ How to write server side code in a page ?

Our index.aspx page has only plain HTML now. We are going to embed some server side code inside it so that the server will process it and generate some output dynamically.

Anything you write inside a special tag as shown below will be trated as server side code:

```
<%

%>
```

You can write your favourite C# or VB.NET code inside the <% %> tags and it will be processed by the .NET runtime to generate html output.

Change the content of your index.aspx as shown below:

```
<html>

<head>
<title>This is my first web page</title>
</head>

<body>
```

```
<br><br><br><br>

<center>

<font size=5 color=red>Welcome to SpiderShop !</font>

<BR><BR>

<font size=3 color=darkgreen>Buy softwares and tools online at very low
rates....</font>

<br><br>

Now the server time is :
<%
Response.Write (DateTime.Now.ToString)
%>

</center>

</body>
</html>
```

Now access your page using the URL http://localhost/ShoppingCart/Index.aspx. You will see the following output in browser:

Buy softwares and tools online at very low rates....

Now the server time is :
4/28/2005 6:33:45 PM

You can see the actual server time displayed in your browser. (Since you are using your own machine as the server and client, you will see your computer time. )

Let us see how it worked. You have writtena piece of code as shown below:

```
Now the server time is :
<%
Response.Write (DateTime.Now.ToString)
%>
```

Since this page has an extension .aspx, the page will be given to the aspnet service to process it. The aspnet service will use the .NET runtime to process all lines enclosed between <% and %>.

In the above case, there is only one line of server side code.

```
Response.Write (DateTime.Now.ToString)
```

The code `Response.Write` is used to generate output.

`DateTime.Now.ToString` will return the current date and time.

So, the resule of the following code is the current date and time and Response.Write() will return that result to the asp.net page which embeds the code.

In the page, we have

```
Now the server time is :
<%
Response.Write (DateTime.Now.ToString)
%>
```

In the above lines, `Now the server time is :` is a plain HTML. The next part gets executes in the server and returns the current time. So, the final output is:

```
Now the server time is :
4/28/2005 6:33:45 PM
```

What a browser gets back from the server is just the above lines. Your server side code is not there anymore. It is processed by the server to produce some output!

Now you must have got a better picture about how the server side code works.

In the coming chapers, we will explain how to write advanced code to write a more dynamic page, which access database and retrieve content just like this Tutorial page.
you learned how to display server time using ASP.NET. The code you wrote was,

```
Now the server time is :
<%
Response.Write (DateTime.Now.ToString)
%>
```

Now let us make some minor changes to the code so that we can explain some concepts better. Change the server side cide in your file Index.aspx under the folder C:\ShoppingCart as shown below:

```
Now the server time is :
<%
dim currentTime as string = DateTime.Now.ToString
Response.Write ( currentTime )
%>
```

We have declared a string variable and assigned the current time to it. Then we are doing a Response.Write ( currentTime ) so that the value is sent to the page output.

Let us view the new page now in browser. Type the URL `http://localhost/ShoppingCart/index.aspx`. You will see the same old result.

You may have noticed that we are using VB.NET syntax to write server side code.

In one of the previous chapters we mentioned that ASP.NET supports any programming language. Let us try it.

Change your server side code to C# syntax:

```
Now the server time is :
<%
string currentTime = DateTime.Now.ToString();
Response.Write ( currentTime );
%>
```

Try to view the page. You will get an error. What happens is, ASP.NET is trying to compile the server side code using vb compiler and it is giving an error (because it does not understand C# syntax).

### ➢ How to use C# in ASP.NET ?

VB.NET is the default language for ASP.NET. This does not mean that VB.NET is a better choice than C#. Both VB.NET and C# are equally good for ASP.NET.

If you like to use a language other than the default VB.NET language, you must explicitely specify what language you are using so that ASP.NET can use appropriate compiler to compile your server side code.

Now we have changed our code to C# syntax. So, we must specify that we want to use C# compiler. What you need to do is, go to the top of the index.aspx page in notepad and type the following line:

```
<%@Language="C#"%>
```

The above line must be the first line in the aspx file. This statement tells the aspx page that it has to use C# compiler to compile the page.

We cannot mix more than one language in one page. If a page uses C# syntax, then you cannot use VB.NET syntax for some portion of the code in the same page.

You will learn about User Controls in one of the coming chapters. You can develop each User Control using a different language and place all those controls in one page.

### ➢ Supported languages for ASP.NET

You can use any .NET programming language to write server side code in ASP.NET. The following are the .NET languages provided by Microsoft:

- C#
- VB.NET
- C++
- J#

Other companies are developing more .NET languages, which you will be able to use with ASP.NET. However, the widely used programming languages for ASP.NET are VB.NET and C#

## 15. Difference between ASP and ASP.NET

ASP stands for **A**ctive **S**erver **P**ages. ASP.NET is the next generation of ASP. After the introduction of ASP.NET, old ASP is called 'Classic ASP'.

Classic ASP uses vb script for server side coding. Vb Script is not supported any more in ASP.NET. Instead, ASP.NET supports more languages including C#, VB.NET, J# etc. VB.NET is very similar to vb script, so it should be easy for old Visual Basic or ASP programmers to switch to VB.NET and ASP.NET

VB Script is a simple scripting language, where as VB.NET or C# are modern, very powerfull, object oriented programming languages. Just for that reason, you will be able to write much more robust and reliable programs in ASP.NET compared to ASP.

In classic ASP, there was no server controls. You have to write all html tags manually. ASP.NET offers a very rich set of controls called Server Controls and Html Controls. It is very easy to drag and drop any controls to a web form. The VS.NET will automatically write the required HTML tags automatically for you.

## ➢ ASP is interpreted, ASP.NET is compiled

Since ASP uses vb script, there is no compilation. All ASP pages are interpreted when th page is executed.

ASP.NET uses modern .NET languages like C#, VB.NET etc. They can be compiled to efficient Microsoft Intermediate Language (MSIL). When you compile an ASP.NET application, the server side code is compiled to assemblies. These assemblies are loaded at run time which makes the ASP.NET pages perform better than classic ASP.

## ➢ ADO and ADO.NET

Classic ASP uses a technology called ADO to connect and work with databases. ASP.NET uses the ADO.NET technology (which is the next generation of ADO).

## ➢ Event driven programming

If you have written Visual Basic programs before, you would love the event driven programming approach. It is so easy to drag and drop a button control and double click on the button to write the event handler for the button click event. When you click on the button at turn time, it will execute whatever code you have written in the event handler.

This type of event driven programming was not available with classic ASP. You cannot drag and drop a button and write a 'on click' event handler with ASP.

With ASP.NET, this is changed. It works pretty much like your Visual Basic program. You can write event handlers for several events like button click event, text changed event etc.

However, there is big difference between the way event handling works in regular Visual basic and ASP.NET. In ASP.NET, a page is loaded in the client browser. And the server may be in another location (may be in another country). When you click on the button in an ASP.NET page, the 'click event' handler has to be executed on the server, not on the client.

How does the server know when you click on a button your browser? This is a tricky thing in ASP.NET. When you write an event handler for a button lick or something like that in ASP.NET, lot of things happens behind the screens. ASP.NET will produce lot of client side javascript code to handle this and embed this javascript in the html page it sends to the browser. When you click on the button in the browser, the client side javascript will get executed. This javascript will generate some information required for the server and and submit the page request to the server. There is enough information embedded in this request so that the server will understand that user has clicked a specific button in the browser and it has to execute some 'specific event handler' code in the server side. So, when the user clicks on a button, the page is submitted automatically to the server with some special information. In the server side, it will process the event handler for the button click event and send back the output page to the browser again. Similary, you can write other event handlers like text changed event for Textboxes etc.

As a user of the web page, you will not even know what happened in the background. All you can see is, when you clicked the button, it executed the button click event handler in the server side and you got the result. However, you may notice a delay because the page has to be submitted to the server to execute the event handler.

NOTE: The event handling we just discussed is server side event handling. In addition to that, you can handle any events in the client side using Javascript. This is supported even in classic ASP. The client side event handling is used for simpel client side validation, displaying messages to the user etc. You cannot do any server side programming in client side java script (like accessing the database etc).