Introduction to Web Programming using ASP.NET

Lecture Notes

Final Test

- Mechanical Engineering :
  - 15th Jan 2014, 10:30 h, room ?

- Computer Science :
  - 16th Jan 2014, 11:30 h, room ?

Information and lectures at **www.tasheva.info/en**

About the Course

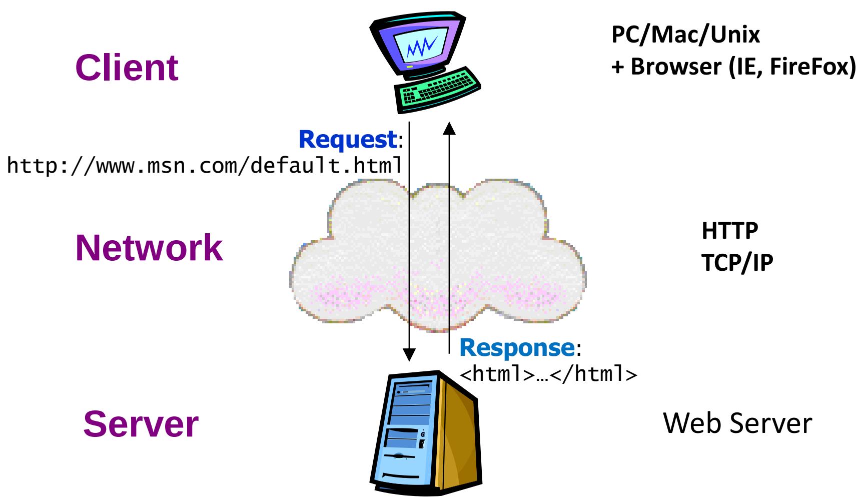
- Objective
  - Introduction to web programming using the Microsoft® ASP.NET technology and Microsoft® IIS (Internet Information Server).
- Prerequisite
  - The programming language will be C#

Outline of the Course

- Introduction to web programming and ASP.NET
- Create web application using Visual Studio® 2010 and C#
- Create and add code-behind file to an ASP.NET web form
- Examine common ASP.NET Controls
- Connecting to a Database in an ASP.NET application and ASP.NET Data Controls
- Session management
- Validation controls
- Master pages
- Configuring and deploying an ASP.NET web application on an IIS server
- Securing an ASP.NET web application
- Introduction to ASP.NET AJAX

Internet Technologies

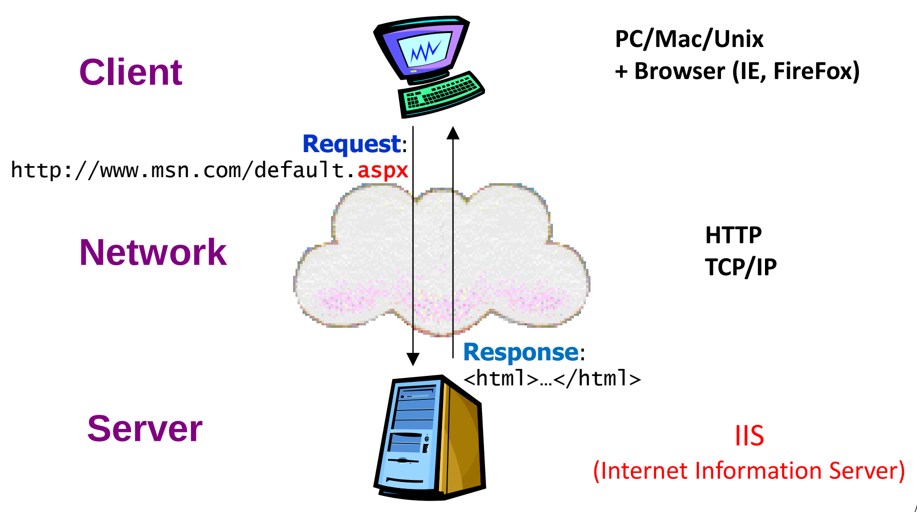
WWW Architecture



Web Technologies

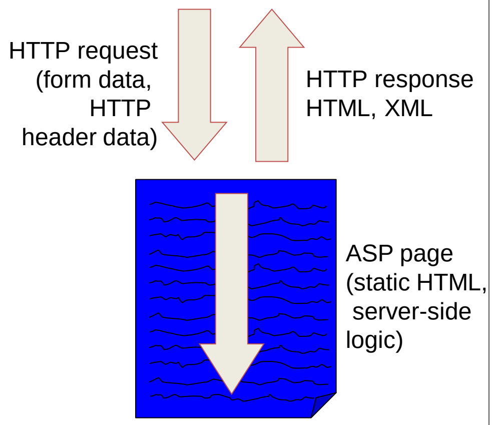
- HTTP / HTTPS (URL, GET/POST)
- Client-side:
  - HTML / XHTML (Extensible HyperText Markup Language)
  - JavaScript / VBScript (client-side scripting)
  - Applets / ActiveX controls
- Server-side:
  - PHP
  - Phython
  - JSP (Java Server Pages)
  - ASP (Active Server Pages)
  - ASP.NET (next generation of ASP)

ASP Architecture



Server-Side Code

- What is server-side code?
  - Software that runs on the server, not the client
  - Receives input from
    - URL parameters
    - HTML form data
  - Can access server-side databases, e-mail servers, files, mainframes, etc.
  - Dynamically builds a custom HTML response for a client



ASP.NET Overview and Features

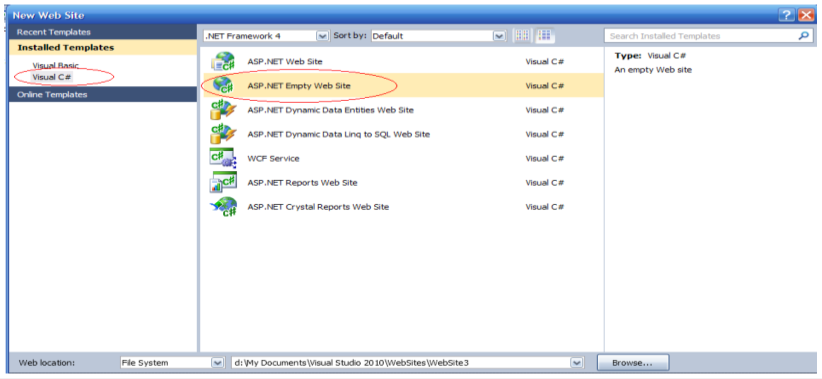
- ASP.NET provides services to allow the creation, deployment, and execution of Web Applications and Web Services
- Web Applications are built using Web Forms
- Web Forms are designed to make building web-based applications as easy as building Visual Basic applications
- Built on .NET Framework: any .NET programming language can be used (C#, Visual Basic)
- Complete object model
- Separation of code and UI
- Maintains page state
- Session management
- Caching, Debugging, Extensibility

`WebTime.aspx` Example

Creating an ASP.NET Web Application using Visual Studio

Step 1: Creating the Web Application Project

- Select **File > New Web Site…** and choose **ASP.NET Empty Web Site** in the **Templates** pane.
- Select **File System** from the drop-down list closest to **Location**.
- Set the **Language** to Visual C#, and click **OK**.



.NET Framework 4 Sort by: Default

ASP.NET Web Site	Visual C#
ASP.NET Empty Web Site	Visual C#
ASP.NET Dynamic Data Entities Web Site	Visual C#
ASP.NET Dynamic Data Linq to SQL Web Site	Visual C#
WCF Service	Visual C#
ASP.NET Reports Web Site	Visual C#
ASP.NET Crystal Reports Web Site	Visual C#

Type: Visual C#
An empty Web site

Web location: File System d:\My Documents\Visual Studio 2010\WebSites\WebSite3

`WebTime.aspx` Example

- Add n ASPX file (i.e., Web Form), default named Default.aspx is created for each new project.
- Visual Web Developer creates a code-behind file named Default.aspx.cs.
- The **View Designer** button opens the Web Form in **Design** mode.
- The **Copy Web Site** button allows you to copy the project's files to another location, such as a remote web server.
- Finally, the **ASP.NET Configuration** button takes you to the **Web Site Administration Tool.**
- Look at **Toolbox** displayed in the IDE when the project loads.
  - **Standard** and **Data** list of web controls.

Editing the `WebTime.aspx`

- When the project loads for the first time, the Web Forms Designer displays the autogenerated ASPX file in **Source** mode.
- **Design** mode indicates the XHTML element where the cursor is currently located.
- You can also view both the markup and the web-page design at the same time by using **Split** mode
- Right click the ASPX file in the **Solution Explorer** and select **View Code** to open the code-behind file.

`WebTime.aspx` Example

- Let's create our first ASP.NET page using Visual Studio
  1. Modify title of the page
  2. Add a heading <h2>
  3. Look at the page in Design and Split modes
  4. Add a **`Label`** control from the *Toolbox*
  5. Change ID of the **`Label`** control
  6. Change some physical properties of the **`Label`** control
  7. Go to `WebTime.aspx.cs` file and add `Page_Init` function to set `Text` property of the **`Label`** control

`WebTime.aspx` Example

Changing the Title of the Page

- We change the page's title from the default Untitled Page to A Simple Web Form Example.
- Open the ASPX file in **Source** mode and modify the text between the <title> tags.
- Alternatively, you can modify the Web Form's Title property in the **Properties** window.
- To view the Web Form's properties, select DOCUMENT from the drop-down list in the **Properties** window.

Designing the Page

- To add controls to the page, you can drag and drop them from the **Toolbox** onto the Web Form in **Design** mode.
- Like the Web Form itself, each control is an object that has properties, methods and events.
- You can type text directly on a Web Form at the cursor location or insert XHTML elements using menu commands.

Renaming the `WebTime.aspx`

Renaming the ASPX File

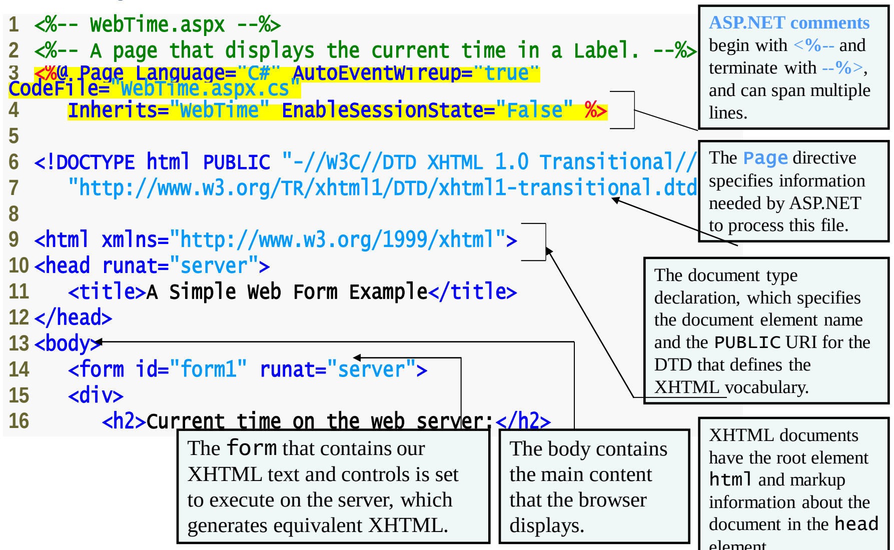
- Right click the ASPX file in the **Solution Explorer** and select **Rename**.
- Enter the new file name WebTime.aspx and press *Enter*. Both the ASPX file and the code-behind file are updated.

Renaming the Class in the Code-Behind File and Updating the ASPX File

- Visual Studio's refactoring tool, which automatically updates the existing references to this class in the rest of the project to reflect this change.
- Right click the class name in the partial class's declaration and select **Refactor > Rename…** to open the **Rename** dialog.

Visual Studio generates the markup shown when you create the GUI.

`WebTime.aspx` (1 of 2)



```
<%-- WebTime.aspx --%>
<%-- A page that displays the current time in a Label. --%>
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="WebTime.aspx.cs"
   Inherits="WebTime" EnableSessionState="False" %>

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>A Simple Web Form Example</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      <h2>Current time on the web server:</h2>
```

ASP.NET comments begin with <%-- and terminate with --%>, and can span multiple lines.

The `Page` directive specifies information needed by ASP.NET to process this file.

The document type declaration, which specifies the document element name and the `PUBLIC` URI for the DTD that defines the XHTML vocabulary.

XHTML documents have the root element `html` and markup information about the document in the `head` element.

The body contains the main content that the browser displays.

The `form` that contains our XHTML text and controls is set to execute on the server, which generates equivalent XHTML.

ASPX file that displays the web server's time.

WebTime.aspx

(2 of 2)

```
      <p>
         <asp:Label ID="timeLabel" runat="server" BackColor="Black"
            Font-Size="XX-Large" ForeColor="Yellow"
            EnableViewState="False"></asp:Label>
      </p>
   </div>
   </form>
</body>
</html>
```

The **asp: tag prefix** indicates that the label is an ASP.NET web control, not an XHTML element.

Markup for a label web control.

- In an ASPX file a **directive** is delimited by **<%@** and **%>**.

ASPX file that displays the web server’s time. (Part 2 of 2.)

WebTime.aspx Example

Examining an ASPX File

- The Page directive’s **Language** attribute specifies the code-behind file’s language.
- The **CodeFile** attribute specifies the code-behind filename.
- When **AutoEventWireup** is true, ASP.NET automatically treats a method of name Page_*eventName* as an event handler.
- When AutoEventWireup is set to false, you specify event handlers using attributes in the Page directive just as you would any other web control.
- The **Inherits** attribute (line 4) specifies the class in the code-behind file from which this ASP.NET class inherits.

WebTime.aspx Example

- The document type declaration, which specifies the document element name and the PUBLIC URI for the DTD that defines the XHTML vocabulary.
- XHTML documents have the root element html and markup information about the document in the head element.
- Setting the **runat** attribute to **"server"** indicates that ASP.NET processes the element and its nested elements and generates the corresponding XHTML.
- The body contains the main content that the browser displays.
- The form that contains our XHTML text and controls is set to execute on the server, which generates equivalent XHTML.

WebTime.aspx Example

- The **ID** attribute assigns a name to a control, used as an identifier in the code-behind file.
- The **asp: tag prefix** indicates that the label is an ASP.NET web control, not an XHTML element.
- Each web control maps to a corresponding XHTML element or group of elements.

`WebTime.aspx` Example

- The asp:Label control is written as an XHTML **span** element.
- A span element contains text with formatting styles.
- This control is processed on the server so that the server can translate the control into XHTML.
- If this is not supported, the asp:Label element is written as text to the client.

The code-behind file (`WebTime.aspx.cs`)

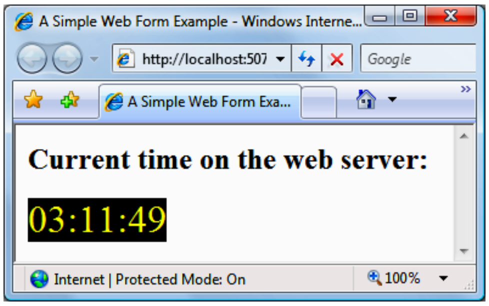
```
// WebTime.aspx.cs
// Code-behind file for a page that displays the current time.
using System;

public partial class WebTime : System.Web.UI.Page
{
   // initializes the contents of the page
   protected void Page_Init( object sender, EventArgs e )
   {
      // display the server's current time in timeLabel
      timeLabel.Text = DateTime.Now.ToString( "hh:mm:ss" );
   } // end method Page_Init
} // end class WebTime
```

The `Page_Init` method handles the page's `Init` event, which indicates that the page is ready to be initialized.

Retrieve the current time and formats it as `hh:mm:ss`.

Code-behind file for a page that displays the web server's time. (Part 1 of 2.)

`WebTime.aspx` Example Run


A Simple Web Form Example - Windows Interne...
http://localhost:507
Google
A Simple Web Form Exa...
Internet | Protected Mode: On
100%

Current time on the web server:

03:11:49

- The Page_Init method handles the page's Init event, which indicates that the page is ready to be initialized.

`WebTime.aspx` Example
Relationship Between an ASPX File and a Code Behind File

- The code-behind file inherits from Page, which defines the general functionality of a web page.
- The code-behind file contains a partial class.
- ASP.NET generates another partial class that defines the remainder of that class, based on the markup in the ASPX file.
- The first time the web page is requested, this class is compiled, and an instance is created.
- This instance represents our page—it creates the XHTML that is sent to the client.
- Once an instance of the web page has been created, multiple clients can use it to access the page—no recompilation is necessary.

How the Code in an ASP.NET Web Page Executes

- When an instance of the page is created, the **PreInit** event occurs first, invoking method **Page_PreInit**, which can be used to set a page's theme.
- The Init event occurs next, invoking method Page_Init, which is used to initialize objects and other aspects of the page.
- Next, the **Load** event occurs, and the **Page_Load** event handler executes.
  - The `Init` event is raised only once (when the page is first requested).
  - The `Load` event is raised with every request.
- The page then processes any events that are generated by the page's controls.
- Once a response has been generated and sent, an `Unload` event occurs, which calls `Page_Unload`, which typically releases resources used by the page.

`WebTime.aspx` Example

- To view the XHTML generated by ASP.NET, select **View Source** from the **Page** menu Page ▾ in Internet Explorer (or **View > Page Source** if you are using Firefox).
- Nonvisual form components, called **hidden inputs**, store data that the user doesn't need to see.
- Attribute **method** of the form element specifies the request method (usually get or post). The **action** attribute identifies the resource that will be requested when a form is submitted.

- Figure shows the XHTML generated by ASP.NET when a web browser requests `WebTime.aspx`.

`WebTime.html`

(1 of 2)

```
<!-- WebTime.html -->
<!-- The XHTML generated when WebTime.aspx is loaded. -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
   <title>A Simple Web Form Example</title>
</head>
<body>
   <form method="post" action="WebTime.aspx" id="form1">
      <div>
         <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value=
            "/wEPDwUJODExMDE5NzY5ZGQ4n4mht8D7Eqxn73tM5LDnstPlCg==" />
      </div>
```

Attribute `method` of the `form` element specifies the request method (usually `get` or `post`). The `action` attribute identifies the resource that will be requested when a form is submitted.

Nonvisual form components, called **hidden inputs**, store data that the user doesn't need to see.

Fig. | XHTML response when the browser requests `WebTime.aspx`. (Part 1 of 2.)

`WebTime.html`

(2 of 2)

```

      <div>
         <h2>Current time on the web server:</h2>
         <p>
            <span id="timeLabel" style="color:Yellow;
               background-color:Black;font-size:XX-Large;">
               03:11:49
            </span>
         </p>
      </div>
   </form>
</body>
</html>
```

The `form` contains a `span` element to represent the text in the label. Formatting properties of `timeLabel` are converted into the `style` attribute of the `span` element.

Fig. | XHTML response when the browser requests `WebTime.aspx`. (Part 2 of 2.)

`WebTime.aspx` Example

- When the form is processed on the server, the `runat` attribute is removed.
- Only those elements marked in the ASPX file with `runat="server"` are modified or replaced in the generated XHTML.

`WebTime.aspx` Example

- The positions of controls and other elements are relative to the Web Form's upper-left corner. This type of layout is known as **relative positioning**.
- An alternate type of layout is known as **absolute positioning**, in which controls are located exactly where they are dropped on the Web Form.
- You can enable absolute positioning in **Design** mode in the **HTML Designer > CSS Styling** node of the **Options** dialog.
- **Absolute positioning is discouraged, because pages designed in this manner may not render correctly in different browsers or on computers with different screen resolutions and font sizes.**

Running `WebTime.aspx` Example

Running the Program

- You can view the Web Form several ways.
  - You can select **Debug > Start Without Debugging**, which runs the application by opening it in a browser window.
  - To debug your application, you can select **Debug > Start Debugging**. You cannot debug a web application unless debugging is explicitly enabled by the **web.config** file.
  - To view a specific ASPX file, you can right click either the Web Forms Designer or the ASPX file name and select **View In Browser**.
  - Finally, you can run your application by opening a browser window and typing the web page's URL in the **Address** field.

Event Handling

- GUIs are **event driven**.
- When the user interacts with a GUI component, the **event** drives the program to perform a task.
- A method that performs a task in response to an event is called an **event handler**.

Event Handling Example (`HelloWorld`)

- Let’s create another ASP.NET page using Visual Studio
  1. Add a `Button` and a `Label` control
  2. To create this click event handler, double click the `Button` on the `Form`.
  3. The following empty event handler is declared:
  4. Set the `Text` property of the `Label` control with the current time in this function.

```
protected void Button1_Click(object sender,
                             EventArgs e)
{

}
```

Event Handling Example (`HelloWorld`)

- To add an event handler, alternatively in markup (aspx) file:
  1. Add a `onclick="BClick"` property to the `Button` control.
  2. Add a function `BClick` to the page class in the code behind.

HelloWorld Example

ASP.NET comments begin with <%-- and terminate with --%>, and can span multiple lines.

```
<%-- Hello World page that also displays the current time. --%>
<%@ Page Language="C#" AutoEventWireup="true"
  CodeFile="HelloWorld.aspx.cs" Inherits="HelloWorldPage

<!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>Hello World Web Form</title>
</head>
<body>
    <form id="form1" runat="server">
    <asp:Button ID="buttonClick" runat="server" Font-Size="Medium"
                Width="102px" Text="Click Me" onclick="BClick" />
    <br />
    <asp:Label ID="labelHello" runat="server"></asp:Label>
    </form>
</body>  </html>
```

The Page directive specifies information needed by ASP.NET to process this file.

XHTML documents have the root element html and markup information about the document in the head element.

The body contains the main content that the browser displays.

The form that contains our XHTML text and controls is set to execute on the server, which generates equivalent XHTML.

Markup for label & button web controls.

The asp: tag prefix indicates that the label is an ASP.NET web control, not an XHTML element.

ASPX Code Behind File

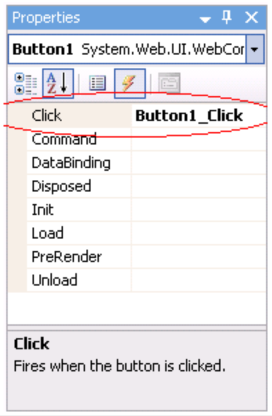
```
public partial class HelloWorldPage :
  System.Web.UI.Page
{
    protected void BClick(object sender, EventArgs e)
    {
        labelHello.Text = "Hello World! Time is " +
                          DateTime.Now;
    }
}
```

Event Handling

- By convention, C# names the event-handler method as *objectName_eventName* (e.g., `Button1_Click`).
- Each event handler receives two parameters when it is called:
  - An object reference named sender—a reference to the object that generated the event.
  - A reference to an object of type `EventArgs`, which contains additional information about the event.

Other Ways to Create Event Handlers

- Typically, controls can generate many different types of events.
- Clicking the **Events** icon (the lightning-bolt icon) in the **Properties** window, displays all the events for the selected control.



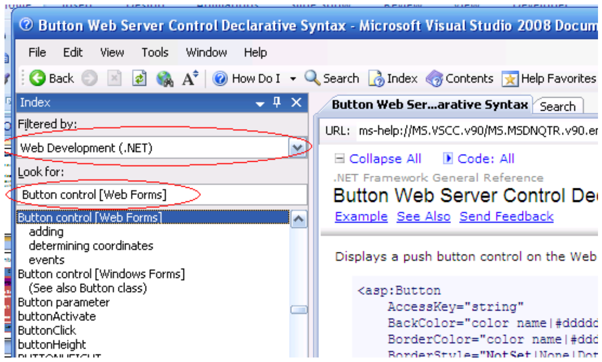
Button1 System.Web.UI.WebCor

Click	**Button1_Click**
Command	
DataBinding	
Disposed	
Init	
Load	
PreRender	
Unload	

Click
Fires when the button is clicked.

Locating Event Information

- To learn about the events raised by a control, select **Help > Index**.
- In the window, select **Web Development (.NET)** in the **Filtered by** drop-down list and enter the name of the control's class in the **Index** window.



Button Web Server Control Declarative Syntax - Microsoft Visual Studio 2008 Docum

File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites
Index
Button Web Ser...arative Syntax Search

Filtered by: Web Development (.NET)

Look for: Button control [Web Forms]

Button control [Web Forms]
  adding
  determining coordinates
  events
Button control [Windows Forms]
  (See also Button class)
Button parameter
buttonActivate
ButtonClick
buttonHeight

URL: ms-help://MS.VSCC.v90/MS.MSDNQTR.v90.er

Collapse All Code: All

.NET Framework General Reference

Button Web Server Control De

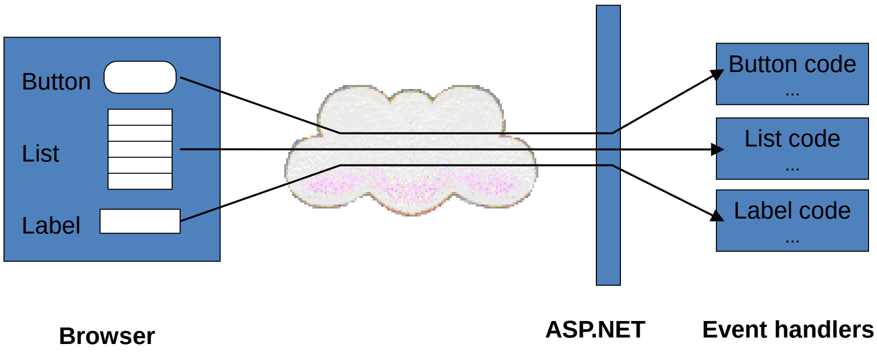
Example See Also Send Feedback

Displays a push button control on the Web

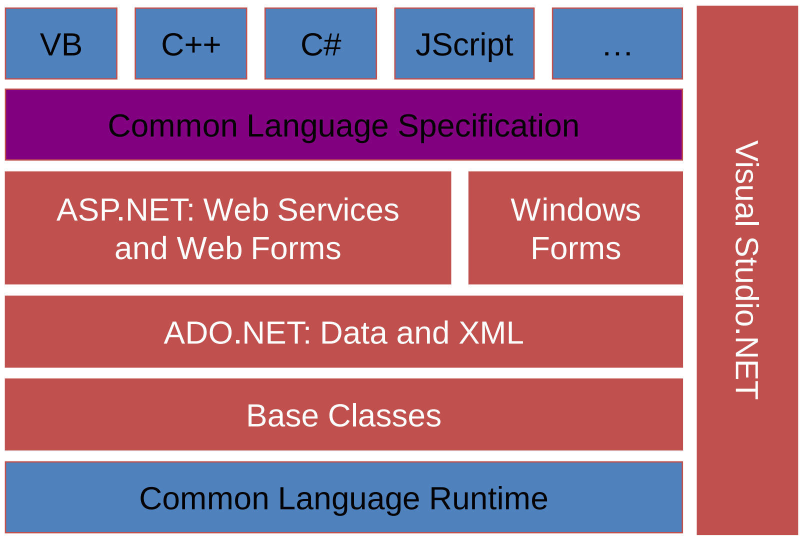
```
<asp:Button
    AccessKey="string"
    BackColor="color name|#dddd
    BorderColor="color name|#ddc
```

39

Programming Model
Controls and Events



ASP.NET Architecture



Programming Model
ASP.NET Object Model

- Controls are objects, available in server-side code
  - Derived from `System.Web.UI.Control`
- The web page is an object too
  - Derived from `System.Web.UI.Page`
- User code executes on the web server in page or control event handlers

Resources

- http://msdn.microsoft.com/en-us/aa336522.aspx
- http://www.asp.net/
- http://www.aspfree.com/
- http://www.devx.com/dotnet/