

## ❄ File Upload Control

- ➔ ASP.NET has two controls that allow users to upload files to the web server.
- ➔ Once the server receives the posted file data, the application can save it, check it, or ignore it. The following controls allow the file uploading

**HtmlInputFile** - an HTML server control

**FileUpload** - and ASP.NET web control

- ➔ The FileUpload control allows the user to browse for and select the file to be uploaded, providing a browse button and a text box for entering the filename.
- ➔ Once, the user has entered the filename in the text box by typing the name or browsing, the SaveAs method of the FileUpload control can be called to save the file to the disk.
- ➔ The fileupload control in asp.net used to upload any file like image, document file, zip file ..etc to asp.net website.

- ➔ The basic syntax of FileUpload is

**<asp:FileUpload ID= "Uploader" runat = "server" />**

- ➔ The FileUpload class is derived from the WebControl class, and inherits all its members. Apart from those, the FileUpload class has the following read-only properties.

### ❏ Important Properties

Properties	Description
FileBytes	Returns an array of the bytes in a file to be uploaded.
FileContent	Returns the stream object pointing to the file to be uploaded.
FileName	Returns the name of the file to be uploaded.
HasFile	Specifies whether the control has a file to upload.
PostedFile	Returns a reference to the uploaded file.

- ➔ The posted file is encapsulated in an object of type HttpPostedFile, which could be accessed through the PostedFile property of the FileUpload class.

➔ The HttpPostedFile class has the following frequently used properties:

### ▮ Important Properties

Properties	Description
ContentLength	Returns the size of the uploaded file in bytes.
ContentType	Returns the MIME type of the uploaded file.
FileName	Returns the full filename.

### Example.

Here we demonstrate a file to be upload and immediately it display on image control

- ➔ First Drag file upload control, image control, button control and label control on the web page
- ➔ Create img directory on the root of the solution and mentioned below code.

```
If (FileUpload1.HasFile) Then

    FileUpload1.SaveAs(Server.MapPath("~/img/") + FileUpload1.FileName)
    Label1.Text = "Image Uploaded Successfully !!"
    Image1.ImageUrl = "~/img/" + FileUpload1.FileName

Else

    Label1.Text = "Select image first !!"

End If
```

### ❄ Image Control

- ➔ Image box control is used to display images on it.
- ➔ The images displayed can be anything varying from Bitmap, JPEG, GIF, PNG or any other image format files. The Image control is based on the Control class.
- ➔ Notable property of the Image Control in the Appearance section of the properties window is the ImageURL property which allows to add the image to be displayed on the control.
- ➔ Images can be added to the image control with the Imageurl property from the Properties window or by code.
- ➔ **ImageUrl** is the method of Image properties to load image in image control dynamically.

### ▮ Important Properties

Properties	Description
AlternateText	Alternate text to be displayed in absence of the image.
ImageAlign	Alignment options for the control.
ImageUrl	Path of the image to be displayed by the control.

### ❄Hyper Link

➔ The HyperLink is used to create a link to another Web page/web server.

### ▮ Important Properties

Properties	Description
ID	To identify in the coding window by the id.
Text	To Specifies the text on a hyperlink
ImageURL	The URL of the image to display for the link
Navigate URL	The target URL of the link
Target	The target frame of the URL <b>Target Means</b> _blank - the target URL will open in a new window _self - the target URL will open in the same frame as it was clicked _parent - the target URL will open in the parent frameset _top - the target URL will open in the full body of the window

### ❄Tables

➔ Table control is used to structure a web pages. In other words to divide a page into several rows and columns to arrange the information or images. When it is rendered on the page, it is implemented through <table> HTML tag.

- ➔ We can simply use HTML <table> control instead of using asp:Table control. However many of one benefits of using asp:Table control is we can dynamically add rows or columns at the runtime or change the appearance of the table.
- ➔ You can skip ID property of the TableRow or TableCell, however it is advisable to write these property otherwise you will not be able to play with these controls.

### ▮ Important Properties

Properties	Description
BackImageUrl	Used to Set background image of the table
Caption	Used to write the caption of the table.
BorderStyle	Style around the border
Gridlines	Which gridlines display around the cells
CaptionAlign	Alignment to the associated caption
Rows	The collection of rows within the table.

## ❄Panel

- ➔ The Panel control works as a container for other controls on the page. It controls the appearance and visibility of the controls it contains. It also allows generating controls programmatically.

### ▮ Important Properties

Properties	Description
BackImageUrl	URL of the background image of the panel.
DefaultButton	Gets or sets the identifier for the default button that is contained in the Panel control.
Direction	Text direction in the panel.
GroupingText	Allows grouping of text as a field.
HorizontalAlign	Horizontal alignment of the content in the panel.
ScrollBars	Specifies visibility and location of scrollbars within the panel.
Wrap	Allows text wrapping.

## ❄ Wizard Control

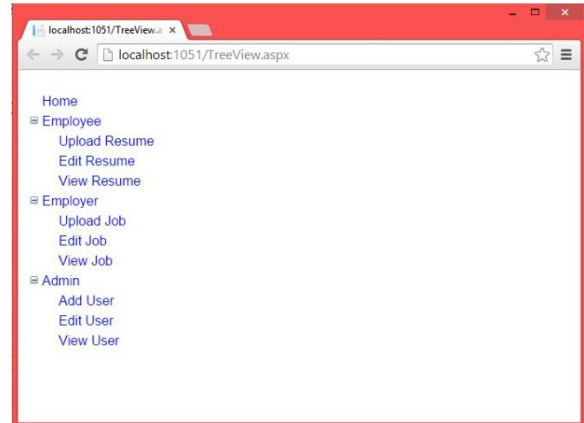
- ➔ The Wizard control is a more advanced version of the MultiView control.
- ➔ It has built-in yet customizable behavior, including sidebar with step links, style, and navigation buttons.
- ➔ Generally, Wizards represent a single task and user moves, from the current step to the next or immediately preceding step in case he/she wants modification.



## Navigation Control [Treeview, Sitemappath & Menu]

### ❄ Tree View Control

- ➔ The TreeView control is the most impressing new control in ASP.NET. It is used to display the hierarchical data in tree view format.
- ➔ It also supports dynamic population of the node on demand without page refresh.
- ➔ You can use the TreeView control to display information from a wide variety of data sources such as an XML file, site-map file, string, or from a database.
- ➔ The top level in a tree view are root nodes that can be expanded or collapsed if the nodes have child nodes.
- ➔ Tree view look like this...
- ➔ The user can expand the TreeNode by clicking the plus sign (+) button, if one is displayed next to the TreeNode, or you can expand the TreeNode by calling the TreeNode.Expand method.



### ▮ Important Properties

Properties	Description
Text	Indicates the text to display in the node.
Tooltip	Indicates the tooltip of the node when you mouse over.
Value	Indicates the nondisplayed value (usually unique id to use in server side events)
NavigateUrl	Indicates the target location to send to the user when node is clicked. If not set you can handle TreeView.SelectedNodeChanged event to decide what to do.
Target	If NavigationUrl property is set, it indicates where to open the target location (in new window or same window).

ImageUrl	Indicates the image that appears next to the node.
ImageToolTip	Indicates the tooltip text to display for image next to the node.
<b>Style Properties</b>	
NodeSpacing	Space (in pixel) between current node and the node above or below it.
VerticalPadding	Space (in pixel) between the top and bottom of the node text.
HorizontalPadding	Space (in pixel) between the left and right of the node text.
ChildNodePadding	Space (in pixel) between the parent node and its child node.

## ❄Menu Control

- ➔ The Menu control is used to create a menu of hierarchical data that can be used to navigate through the pages.
- ➔ The Menu control conceptually contains two types of items. First is StaticMenu that is always displayed on the page, Second is DynamicMenu that appears when opens the parent item.

### ▮ Important Properties

Properties	Description
Text	Indicates the text to display in the menu.
Tooltip	Indicates the tooltip of the menu item when you mouse over.
Value	Indicates the nondisplayed value (usually unique id to use in server side events)
NavigateUrl	Indicates the target location to send the user when menu item is clicked. If not set you can handle MenuItemClick event to decide what to do.
Target	If NavigationUrl property is set, it indicates where to open the target location (in new window or same window).
Selectable	true/false. If false, this item can't be selected. Usually in case of this item has some child.

ImageUrl	Indicates the image that appears next to the menu item.
ImageToolTip	Indicates the tooltip text to display for image next to the item.
PopOutImageUrl	Indicates the image that is displayed right to the menu item when it has some subitems.
Target	If NavigationUrl property is set, it indicates where to open the target location (in new window or same window).
<b>Style Properties</b>	
StaticMenuStyle	Sets the style of the parent box in which all menu items appears.
DynamicMenuStyle	Sets the style of the parent box in which dynamic menu items appears.
StaticMenuItemStyle	Sets the style of the individual static menu items.
DynamicMenuItemStyle	Sets the style of the individual dynamic menu items.
StaticSelectedStyle	Sets the style of the selected static items.
DynamicSelectedStyle	Sets the style of the selected dynamic items.
StaticHoverStyle	Sets the mouse hovering style of the static items.
DynamicHoverStyle	Sets the mouse hovering style of the dynamic items (subitems).

### ✱ Site map path

- ➔ The SiteMapPath control displays the trail (navigation path) to the current page. The path acts as clickable links to previous pages.
- ➔ Unlike the TreeView and Menu control the SiteMapPath control does NOT use a siteMapDataSource. The SiteMapPath control uses the web.sitemap file by default.

### ▮ Important Properties

Properties	Description
PathSeparator	Gets or sets Path separator text.
NodeStyle	Sets the style of all nodes that will be displayed.
CurrentNodeStyle	Sets the style on node that represent the current page.



RootNodeStyle	Sets the style on the absolute root node.
PathSeparatorStyle	Sets the style of path separator.
NodeStyle	Sets the style of all nodes that will be displayed.
Path Direction	The direction of the path to render.

### \*Master Page

- ➔ Master pages allow you to create a consistent look and behavior for all the pages (or group of pages) in your web application.
- ➔ A master page provides a template for other pages, with shared layout and functionality. The master page defines placeholders for the content, which can be overridden by content pages. The output result is a combination of the master page and the content page.
- ➔ The content pages contain the content you want to display.
- ➔ When users request the content page, ASP.NET merges the pages to produce output that combines the layout of the master page with the content of the content page.
- ➔ The extension of MasterPage is **'master'**.
- ➔ MasterPage cannot be directly accessed from the client because it just acts as a template for the other Content Pages.
- ➔ In a MasterPage we can have content either inside **ContentPlaceHolder** or outside it. Only content inside the ContentPlaceHolder can be customized in the Content Page.
- ➔ We can have multiple masters in one web application.
- ➔ A MasterPage can have another MasterPage as Master to it.
- ➔ The content page content can be placed only inside the content tag.
- ➔ Controls of MasterPage can be programmed in the MasterPage and content page but a content page control will never be programmed in MasterPage.
- ➔ The **MasterPageFile** property of a webform can be set dynamically and it should be done either in or before the **Page\_PreInit** event of the WebForm. **Page.MasterPageFile = "MasterPage.master"**. The dynamically set Master Page must have the ContentPlaceHolder whose content has been customized in the WebForm.
- ➔ The order in which events are raised: **Load (Page)** a **Load (Master)** a **LoadComplete (Page)** i.e. if we want to overwrite something already done in Load event handler of Master then it should be coded in the LoadComplete event of the page.
  - ➔ **Page\_Load** is the name of method for event handler for Load event of Master. (it's not Master\_Load).

## **Master and Content Pages**

- ➔ Defining a Master Page is just like defining a normal page. Master Pages can contain markup, controls, or code, or any combination of these elements.
- ➔ However, a Master Page can contain a special type of control, called a ContentPlaceHolder control.
- ➔ A ContentPlaceHolder defines a region of the master page rendering that can be substituted with content from a page associated to the master. A ContentPlaceHolder can also contain default content, just in case the derive page does not need to override this content.
- ➔ To differentiate a Master Page from a normal page, a Master Page is saved under the .master file extension. A page can derive from a Master Page by defining a MasterPageFile attribute on its Page directive, A page that is associated to a Master Page is called a Content Page.

### **Advantage of Master Page**

1. Centralized Updating
2. Common Controls
3. Flexibilit0y
4. Enhanced User Experience
5. Site Management
6. Less Code
7. Less Storage Requirements

## **\*Content Page**

- ➔ The Content Page is a page that is associated to a Master Page. A Content Page will contain only markup and controls inside Content controls and it cannot have any top-level content of its own. Any Content Page can use controls that specifically override content placeholder sections in the Master Page.

## **\*Themes**

- ➔ ASP.NET includes a number of features for customizing the look-and-feel or style of pages and controls in your application.
- ➔ Controls support a Style object model for setting stylistic properties such as fonts, borders, background and foreground colors, width, height, and more.

Controls also fully support Cascading Style Sheets (CSS) for factoring style settings separately from control properties.

- You can define style information as control properties or CSS, or you can also define this information in a separate group of files called a Theme, to be applied to all or a portion of the pages in your application. Individual control styles are specified as a Skin within a Theme.
- Themes implement under the App\_Themes Folder. It contains images, skin & css files.

#### **HOW TO DEFINE THEME**

- To define a Theme, you need to right click the Web Project > Add ASP.NET Folder > Theme. This will by default create a new folder called App\_Themes > Theme1. You are free to rename the Theme1 folder but not the App\_Themes folder. The folder name under App\_Themes folder will be considered as Theme name and all .css files and .skin files under this will be attached with that theme. You can create as many number of theme you want under App\_Themes folder.
- A Theme can have more than one .css and/or .skin files.

#### **WHAT IS .CSS FILE**

- A .css file is a Cascading Style Sheet file that contains the style, behavior about a html page and its elements. These styles are written in the .css file as a block of code and they are referred to as css class.

#### **WHAT IS .SKIN FILE**

- .skin file is a new type of file introduced in ASP.NET 2.0, this can contain style information of any asp.net server control. These information are written in this file as if they are written asp.net page but ID property is not specified

## ❄Validation Control

- ➔ ASP.NET validation controls validate the user input data to ensure that useless, unauthenticated, or contradictory data don't get stored.

### *BaseValidator Class*

- ➔ The validation control classes are inherited from the BaseValidator class hence they inherit its properties and methods. Therefore, it would help to take a look at the properties and the methods of this base class, which are common for all the validation controls

Members Properties	Description
ControlToValidate	Indicates the input control to validate.
Display	Indicates how the error message is shown.
Enabled	Enables or disables the validator.
ErrorMessage	Error string.
Text	Error text to be shown if validation fails.
SetFocusOnError	It indicates whether in case of an invalid control, the focus should switch to the related input control.
ValidationGroup	The logical group of multiple validators, where this control belongs.

### A. Required Field Validators

The RequiredFieldValidator control ensures that the required field is not empty. It is generally tied to a text box to force input into the text box.

### B. Range Validators

- ➔ The RangeValidator control verifies that the input value falls within a predetermined range.
- ➔ It has three specific properties

Properties	Description
Type	it defines the type of the data; the available values are:

	Currency, Date, Double, Integer and String
MinimumValue	it specifies the minimum value of the range
MaximumValue	it specifies the maximum value of the range

### C. Regular Expression Validators

- ➔ The RegularExpressionValidator allows validating the input text by matching a pattern in a regular expression. The regular expression is set in the **ValidationExpression** property.
- ➔ The following table summarizes the commonly used syntax constructs for regular expressions

Character Escapes	Description
\b	Matches a backspace
\t	Matches a tab
\r	Matches a carriage return
\v	Matches a vertical tab
\f	Matches a form feed
\n	Matches a new line
\	Escape character

- ➔ Apart from single character match, a class of characters could be specified that can be matched, called the metacharacters.

Metacharacters	Description
.	Matches any character except \n
[abcd]	Matches any character in the set
[^abcd]	Excludes any character in the set
[2-7a-mA-M]	Matches any character specified in the range
\w	Matches any alphanumeric character and underscore
\W	Matches any non-word character
\s	Matches whitespace characters like, space, tab, new line etc.
\S	Matches any non-whitespace character
\d	Matches any decimal character

\D	Matches any non-decimal character
----	-----------------------------------

→ Quantifiers could be added to specify number of times a character could appear

Quantifier	Description
*	Zero or more matches
+	One or more matches
?	Zero or one matches
{N}	N matches
{N,}	N or more matches
{N,M}	Between N and M matches

E.g.

### Using Server Script : Validation Control

1. 10 Digit Number : Validation Expression : [0-9]{10}
2. Only Digit : ^\d+\$

### Using Client Side Script

```
function isNumber(evt) {  
    evt = (evt) ? evt : window.event;  
    var charCode = (evt.which) ? evt.which :  
    evt.keyCode;  
    if (charCode > 31 && (charCode < 48 ||  
    charCode > 57)) {  
        alert('only number')  
        return false;  
    }  
    return true;  
}
```

```
<asp:TextBox ID="TextBox2"  
runat="server" onkeypress="return  
isNumber(event)" ></asp:TextBox>
```

```
function validatenumerics(key) {  
    //getting key code of pressed key  
    var keycode = (key.which) ? key.which :  
    key.keyCode;  
    //comparing pressed keycodes  
  
    if (keycode > 31 && (keycode < 48 ||  
    keycode > 57)) {  
        alert(" You can enter only characters  
0 to 9 ");  
        return false;  
    }  
    else return true;  
}
```

```
<asp:textbox runat="server"  
id="txtquantity" width="90px"  
onkeypress="return  
validatenumerics(event);" />
```

### D. Compare Validators

- ➔ The CompareValidator control compares a value in one control with a fixed value or a value in another control.
- ➔ It has the following specific properties

Properties	Description
Type	it specifies the data type
ControlToCompare	it specifies the value of the input control to compare with
ValueToCompare	it specifies the constant value to compare with
Operator	it specifies the comparison operator, the available values are: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual and DataTypeCheck

### E. Custom Validators

- ➔ The CustomValidator control allows writing application specific custom validation routines for both the client side and the server side validation.
- ➔ The client side validation is accomplished through the ClientValidationFunction property. The client side validation routine should be written in a scripting language, like JavaScript or VBScript, which the browser can understand.
- ➔ The server side validation routine must be called from the control.s ServerValidate event handler. The server side validation routine should be written in any .Net language, like C# or VB.Net.

### F. Validation Summary

- ➔ The ValidationSummary control does not perform any validation but shows a summary of all errors in the page. The summary displays the values of the ErrorMessage property of all validation controls that failed validation.

**The following two mutually inclusive properties list out the error message:**

- ✓ **ShowSummary:** shows the error messages in specified format.

- ✓ **ShowMessageBox:** shows the error messages in a separate window.

## ❄Ad Rotators

- ➔ The AdRotator control randomly selects banner graphics from a list, which is specified in an external XML schedule file. This external XML schedule file is called the advertisement file.
- ➔ The AdRotator control allows you to specify the advertisement file and the type of window that the link should follow in the AdvertisementFile and the Target property respectively.
- ➔ The basic syntax of adding an AdRotator is as follows

```
<asp:AdRotator runat = "server" AdvertisementFile = "adfile.xml"  
Target = "_blank" />
```

## Advertisement File

- ➔ The advertisement file is an XML file, which contains the information about the advertisements to be displayed.
- ➔ Extensible Markup Language (XML) is a W3C standard for text document markup. It is a text-based markup language that enables you to store data in a structured format by using meaningful tags. The term 'extensible' implies that you can extend your ability to describe a document by defining meaningful tags for the application.
- ➔ XML is not a language in itself, like HTML, but a set of rules for creating new markup languages. It is a meta-markup language. It allows developers to create custom tag sets for special uses. It structures, stores, and transports the information.

Element	Description
Advertisements	Encloses the advertisement file.
Ad	Delineates separate ad.
ImageUrl	The path of image that will be displayed.
NavigateUrl	The link that will be followed when the user clicks the ad.
AlternateText	The text that will be displayed instead of the picture if it cannot be displayed.
Keyword	Keyword identifying a group of advertisements. This is used for filtering.



Impressions	The number indicating how often an advertisement will appear.
Height	Height of the image to be displayed.
Width	Width of the image to be displayed.

➔ Following is an example of XML file:

```
<Advertisements>
  <Ad>
    <ImageUrl>rose1.jpg</ImageUrl>
    <NavigateUrl>http://www.1800flowers.com</NavigateUrl>
    <AlternateText>
      Order flowers, roses, gifts and more
    </AlternateText>
    <Impressions>20</Impressions>
    <Keyword>flowers</Keyword>
  </Ad>

  <Ad>
    <ImageUrl>rose2.jpg</ImageUrl>
    <NavigateUrl>http://www.babybouquets.com.au</NavigateUrl>
    <AlternateText>Order roses and flowers</AlternateText>
    <Impressions>20</Impressions>
    <Keyword>gifts</Keyword>
  </Ad>
</Advertisements>
```

## ❄ Login Control

### 1) Login

➔ Login control provides a ready to use user interface that can be used as a Login interface in the web site.

Properties of the Login Control	
DestinationPageUrl	Indicates the URL to be sent after login attempt successful.
DisplayRememberMe	true/false. Indicates whether to show Remember Me checkbox or not.
CreateUserUrl	Indicates the url of the create user page.
CreateUserText	Indicates the text of the create user link.
PasswordRecoveryUrl	Indicates the url of the password recovery page.
PasswordRecoveryText	Indicates the text of the password recovery link.

### 1) Login View

- ➔ LoginView control is very simple yet very powerful and customizable. It allows user to customize its view for both anonymous user and logged in user.

### 2) Password Recovery

- ➔ PasswordRecovery control is a ready to use control to help user to recover their password, if they forget.
- ➔ PasswordRecovery control uses Membership service to retrieve or reset user's password.

### 3) LoginStatus

- ➔ LoginStatus control is a very simple and compact control that displays either Login (if user is not logged in) or Logout (if user is logged in).

### 4) LoginName

- ➔ LoginName control is used to display current logged in user name

### 5) Create User Wizard

- ➔ CreateUserWizard control is a ready to use control that is used to create a new user to the website.
- ➔ Following are some important events that are very useful.

Events of CreateUserWizard Control	
ContinueButtonClick	Fires when user clicks Continue button in the last wizard step.
CreatingUser	Fires before creating a new user.
CreatedUser	Fires after new user created.
CreateUserError	Fires when creation of user is not successful.

### 6) Change Password

- ➔ ChangePassword control is a ready to use control to help users to change their password.

