**Label control**

Label control is used to place a static, non clickable (can't fire onclick event) piece of text on the page. When it is rendered on the page, it is implemented through <span></span> HTML tag. Its properties like BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height etc. are implemented through style properites of <span>. You can set its Text property either by setting Text properties in the .aspx page or from server side page. (other properties can also be set from both pages)   
  
Following are few properties of the Label that are very useful.

|  |  |
| --- | --- |
| EnableViewState | true/false. If false ViewState will not be maintained. |
| Visible | true/false. If false control will not be rendered to the page |

|  |
| --- |
| DEMO : Label |
| Write something into the TextBox |

|  |
| --- |
| Ex. Example of Label Control |
| // Label control code  <asp:Label ID="Label2" runat="server" BackColor="Coral"  ForeColor="blue" BorderColor="ActiveBorder"  BorderStyle="dashed" BorderWidth="1" Height="20"  Text="Example of Label Control" Width="200"  ></asp:Label> | |

**TextBox Control**

TextBox control is used to enter data into the form that can be sent to the webserver by posting the form.

|  |
| --- |
| DEMO : TextBox |
| |  |  | | --- | --- | | TextMode is Singleline |  | | TextMode is Multiline |  | | PostBack the form | It will Postback the page when cursor leaves this box. | |  |  | |

|  |
| --- |
| Ex. TextBox value will be written here |
| // Singleline TextBox code  <asp:TextBox ID="TextBox1" runat="Server"  Width="300"></asp:TextBox> | |

**Button control**

Button control is generally used to post the form or fire an event either client side or server side. When it is rendered on the page, it is generally implemented through <input type=submit> HTML tag. However, if UserSubmitBehavior property is set to false then control will render out as <input type=button>.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| UserSubmitBehavior | true/false. If true, the button will be used as client browser submit mechanism else asp.net postback mechanism. |
| CausesValidation | Value can be set as true/false. This indicates whether validation will be performed when a button is clicked. |
| PostBackUrl | Indicates the URL on which the Form will be posted back. |
| ValidationGroup | Gets or Sets the name of the validation group that the button belongs to. This is used to validate only a set of Form controls with a Button. |
| OnClick | Attach a server side method that will fire when button will be clicked. |

|  |  |
| --- | --- |
| OnClientClick | Attach a client side (javascript) event that will fire when button will be clicked. |

**LinkButton control**

It implements an anchor <a/> tag that uses only ASP.NET postback mechanism to post the data on the server. Despite being a hyperlink, you can't specify the target URL. There is no UserSubmitBehavior property like Button control with LinkButton control.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| CausesValidation | Value can be set as true/false. This indicates whether validation will be performed when a button is clicked. |
| PostBackUrl | Indicates the URL on which the Form will be posted back. |
| ValidationGroup | Gets or Sets the name of the validation group that the button belongs to. This is used to validate only a set of Form controls with a Button. |
| OnClick | Attach a server side method that will fire when button will be clicked. |
| OnClientClick | Attach a client side (javascript) method that will fire when button will be clicked. |

**ImageButton control**

ImageButton control is generally used to post the form or fire an event either client side or server side. When it is rendered on the page, generally it is implemented through <input type=image > HTML tag.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| ImageUrl | Gets or Sets the location of the image to display. |
| CausesValidation | Value can be set as true/false. This indicates whether validation should be performed when a button is clicked. |
| PostBackUrl | Indicates the URL on which the Form will be posted back. |
| ValidationGroup | Gets or Sets the name of the validation group that the button belongs to. This is used to validate only a set of Form controls with a Button. |
| OnClientClick | Attach a client side (javascript) method that will fire when button will be clicked. |
| OnClick | Attach a server side method that will fire when button will be clicked. |

**Hyperlink control**

Hyperlink control is used to jump to another location or to execute the script code. When rendered on the page, it implements an anchor <a/> tag.

Following are some important properties that are useful.

|  |  |
| --- | --- |
| NavigateUrl | Used to specify the location to jump to. |
| ImageUrl | Used to place an image instead of text as Hyperlink. |

|  |
| --- |
| **DropDownList control** |
| DropDownList control is used to give a single select option to the user from multiple listed items. |

You can specify its height and width in pixel by setting its height and width but you will not be able give mutliple select option to the user. When it is rendered on the page, it is implemented through <select/> HTML tag. It is also called as Combo box.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| SelectedValue | Get the value of the Selected item from the dropdown box. |
| SelectedIndex | Gets or Sets the index of the selected item in the dropdown box. |
| SelectedItem | Gets the selected item from the list. |
| Items | Gets the collection of items from the dropdown box. |
| DataTextField | Name of the data source field to supply the text of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataValueField | Name of the data source field to supply the value of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataSourceID | ID of the datasource component to provide data. (Only used when you have any DataSource component on the page, like SqlDataSource, AccessDataSource etc.) |
| DataSource | The datasource that populates the items in the dropdown box. (Generally used when you are dynamically generating the items from Database.) |
| AutoPostBack | true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection. It will also fire OnSelectedIndexChanged method. |
| AppendDataBoundItems | true or false. If true, the statically added item (added from .aspx page) is maintained when adding items dynamically (from code behind file) or items are cleared. |
| OnSelectedIndexChanged | Method name that fires when user changes the selection of the dropdown box. (Fires only when AutoPostBack=true.) |

<asp:DropDownList ID="DropDownList1" runat="server">

<asp:ListItem Text="Red" Value="red"></asp:ListItem>

<asp:ListItem Text="Blue" Value="blue"></asp:ListItem>

<asp:ListItem Text="Green" Value="green"></asp:ListItem>

</asp:DropDownList>

|  |
| --- |
| **ListBox control** |
| ListBox control is used to give a single or multiple select options to the user from multiple listed items. |

All properties and its working resembles DropDownList box. However, ListBox has two extra properties called Rows and SelectionMode. ListBox control is used to give a single or multiple select option to the user (based on the property set) from multiple listed items. You can specify its height and width in pixel by setting its height and width but you will not be able give mutliple select option to the user. When it is rendered on the page, it is implemented through <select/> HTML tag. It is also called as Combo box.

You can add its option items by directly writing into .aspx page directly or dynamically add at run time or bind through database.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| Rows | No. of rows (items) can be set to display in the List. |
| SelectionMode | Single or Multiple. If multiple, it allows user to select multiple items from the list by holding Ctrl or Shift key. |
| SelectedValue | Get the value of the Selected item from the dropdown box. |
| SelectedIndex | Gets or Sets the index of the selected item in the dropdown box. |
| SelectedItem | Gets the selected item from the list. |
| Items | Gets the collection of items from the dropdown box. |
| DataTextField | Name of the data source field to supply the text of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataValueField | Name of the data source field to supply the value of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataSourceID | ID of the datasource component to provide data. (Only used when you have any DataSource component on the page, like SqlDataSource, AccessDataSource etc.) |
| DataSource | The datasource that populates the items in the listbox box. (Generally used when you are dynamically generating the items from Database.) |
| AutoPostBack | true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection. It will also fire OnSelectedIndexChanged method. |
| AppendDataBoundItems | true or false. If true, the statically added item (added from .aspx page) is maintained when adding items dynamically (from code behind file) or items are cleared. |
| OnSelectedIndexChanged | Method name that fires when user changes the selection of the dropdown box. (Fires only when AutoPostBack=true.) |

<asp:ListBox ID="ListBox1" runat="server">

<asp:ListItem Text="Red" Value="red"></asp:ListItem>

<asp:ListItem Text="Blue" Value="blue"></asp:ListItem>

<asp:ListItem Text="Green" Value="green"></asp:ListItem>

</asp:ListBox>

**CheckBox control**

CheckBox control is used to give option to the user.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| AutoPostBack | Form is automatically posted back when CheckBox is checked or Unchecked. |
| CausesValidation | true/false. If true, Form is validated if Validation control has been used in the form. |
| Checked | true/false. If true, Check box is checked by default. |
| OnCheckedChanged | Fires when CheckBox is checked or Unchecked. This works only if AutoPostBack property is set to true. |
| ValidationGroup | Used to put a checkbox under a particular validation group. It is used when you have many set of form controls and by clicking a paricular button you want to validate a particular set of controls only. |

<asp:CheckBox ID="checkbox2" runat="Server" Text="Click, if Office address is same as Home address" AutoPostBack="True" OnCheckedChanged="PutHomeAddressAsOfficeAddress" BorderColor="brown" BorderWidth="1" CausesValidation="True" />

**CheckBoxList control**

CheckBoxList control is a single control that groups a collection of checkable list items, all are rendered through an individual <input type=checkbox></input>.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| SelectedValue | Gets the value of first selected item. |
| SelectedIndex | Gets or Sets the index of the first selected item. |
| SelectedItem | Gets the first selected item |
| TextAlign | Gets or Sets the alignment of the checkbox text. |
| DataTextField | Name of the data source field to supply the text of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataValueField | Name of the data source field to supply the value of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataSourceID | ID of the datasource component to provide data. (Only used when you have any DataSource component on the page, like SqlDataSource, AccessDataSource etc.) |
| DataSource | The datasource that populates the items in the checkboxlist box. (Generally used when you are dynamically generating the items from Database.) |
| AutoPostBack | true/false. If true, the form is automatically posted back to the server when user click any of the checkbox. It will also fire OnSelectedIndexChanged method. |
| AppendDataBoundItems | true/false. If true, the statically added item (added from .aspx page) is maintained when adding items dynamically (from code behind file) or items are cleared. |
| OnSelectedIndexChanged | Method name that fires when user click any of the checkbox in the list. (Fires only when AutoPostBack=true.) |
| Items | Gets the colleciton of the items from the list. |
| RepeatLayout | table/flow. Gets or Sets the layout of the chekboxes when rendered to the page. |
| RepeatColumns | Gets or Sets the no. of columns to display when the control is rendered. |
| RepeatDirection | Horizontal/Vertical. Gets or Sets the the value to indicate whether the control will be rendered horizontally or vertically. |

<asp:CheckBoxList ID="CheckBoxList1" runat="Server">

<asp:ListItem Text="Red" Value="red"></asp:ListItem>

<asp:ListItem Text="Blue" Value="blue"></asp:ListItem>

<asp:ListItem Text="Green" Value="green"></asp:ListItem>

</asp:CheckBoxList>

**RadioButton control**

RadioButton control is used to give single select option to the user from multiple items.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| AutoPostBack | Form is automatically posted back when Radio button selection is changed. |
| CausesValidation | true/false. If true, Form is validated if Validation control has been used in the form. |
| Checked | true/false. If true, Radio button is selected by default. |
| OnCheckedChanged | Fires when Radio button selection changes. This works only if AutoPostBack property is set to true. |
| ValidationGroup | Used to put a radio button under a particular validation group. It is used when you have many set of form controls and by clicking a paricular button you want to validate a particular set of controls only. |
| GroupName | It is used a group a set of radion buttons so only one of them can be selected at a time. |

<asp:RadioButton ID="RadioButton7" runat="Server" GroupName="1stGroup" Text="Red" Checked="True" />

<asp:RadioButton ID="Radio8" runat="Server" GroupName="1stGroup" Text="Blue" />

**RadioButtonList control**

RadioButtonList control is a single control that groups a collection of radiobuttons, all are rendered through an individual <input type=radio></input>.

Following are some important properties that are very useful.  
(RadioButtonList controls supports the same set of properties as the CheckBoxList control does.

|  |  |
| --- | --- |
| SelectedValue | Get the value first selected item. |
| SelectedIndex | Gets or Sets the index of the first selected item. |
| SelectedItem | Gets the first selected item |
| TextAlign | Gets or Sets the alignment of the radiobutton text. |
| DataTextField | Name of the data source field to supply the text of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataValueField | Name of the data source field to supply the value of the items. (No need to set when you are adding items directly into .aspx page.) |
| DataSourceID | ID of the datasource component to provide data. (Only used when you have any DataSource component on the page, like SqlDataSource, AccessDataSource etc.) |
| DataSource | The datasource that populates the items in the radiobuttonlist. (Generally used when you are dynamically generating the items from Database.) |
| AutoPostBack | true/false. If true, the form is automatically posted back to the server when user click any of the radiobutton. It will also fire OnSelectedIndexChanged method. |
| AppendDataBoundItems | true/false. If true, the statically added item (added from .aspx page) is maintained when adding items dynamically (from code behind file) or items are cleared. |
| OnSelectedIndexChanged | Method name that fires when user click any of the radiobutton in the list. (Fires only when AutoPostBack=true.) |
| Items | Gets the colleciton of the items from the list. |
| RepeatLayout | table/flow. Gets or Set the layout of the radiobuttons when rendered to the page. |
| RepeatColumns | Get or Sets the no. of columns to display when the control is rendered. |
| RepeatDirection | Horizontal/Vertical. Gets or Sets the the value to indicate whether the control will be rendered horizontally or vertically. |

<asp:RadioButtonList ID="RadioButtonList1" runat="Server">

<asp:ListItem Text="Red" Value="red"></asp:ListItem>

<asp:ListItem Text="Blue" Value="blue"></asp:ListItem>

<asp:ListItem Text="Green" Value="green"></asp:ListItem>

</asp:RadioButtonList>

**Image control**

Image control is used to place an image on the page.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| ImageUrl | Url of image location. |
| AlternetText | Appears if image not loaded properly or if image is missing in the specified location. |
| Tooltip | Text message Appearing on mouse over the image |
| ImageAlign | Used to align the Text beside image. |

<asp:Image ID="Image2" runat="Server" ImageUrl="~/images/Dot.gif" AlternateText="Dot Logo"ImageAlign="textTop" ToolTip="Go to Dot Home page" />

**ImageMap control**

ImageMap control is used to create an image that contains clickable hotspot region.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| ImageUrl | Url of image location. |
| AlternetText | Appears if image not loaded properly |
| Tooltip | Appears when on mouse over the image |
| ImageAlign | Used to align the Text beside image. |
| HotSpotMode | PostBack/Navigate .... When Navigate, the user is navigated to a different URL. In case of PostBack, the page is posted back to the server. |
| OnClick | Attach a server side event that fires after clicking on image when HostSpotMode is PostBack. |
| PostBackValue | You can access it in the server side click event through ImageMapEventArgs. (eg. e.PostBackValue) |

<asp:ImageMap ID="ImageMap1" runat="Server" ImageUrl="controldata/gotocontrols.gif" OnClick="FireImageMapClick">

<asp:RectangleHotSpot AlternateText="Label" Left="10" Top="33" Right="75" Bottom="10" NavigateUrl="~/tutorials/controls/label.aspx" />

<asp:RectangleHotSpot AlternateText="Button" Left="80" Top="33" Right="150" Bottom="10" NavigateUrl="~/tutorials/controls/button.aspx" />

<asp:RectangleHotSpot AlternateText="ImageButton" Left="155" Top="33" Right="275" Bottom="10" NavigateUrl="~/tutorials/controls/imagebutton.aspx" />

</asp:ImageMap>

**Asp: Table control**

Table control is used to structure a web pages. In other words to divide a page into several rows and colums to arrange the information or images.

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

Its properties like BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height etc. are implemented through style properites of <table> 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.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| BackImageUrl | Used to Set background image of the table |
| Caption | Used to write the caption of the table. |

<asp:Table ID="Table2" runat="Server" CellPadding="2" CellSpacing="1"

BorderColor="CadetBlue" Caption="Demo of asp:Table control" BorderWidth="1" BorderStyle="Dashed">

<asp:TableRow ID="TableRow2" runat="Server" BorderWidth="1">

<asp:TableCell ID="TableCell4" runat="Server" BorderWidth="1">

Row 1 - Cell 1 </asp:TableCell>

<asp:TableCell ID="TableCell5" runat="Server">

Row 1 - Cell 2 </asp:TableCell> </asp:TableRow>

<asp:TableRow ID="TableRow3" runat="Server">

<asp:TableCell ID="TableCell6" runat="Server">

Row 2 - Cell 1 </asp:TableCell>

<asp:TableCell ID="TableCell7" runat="Server">

Row 2 - Cell 2 </asp:TableCell> </asp:TableRow> </asp:Table>

**BulletedList control**

BulletedList control is used to display the data in a list prefixed with bullet characters.

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| DisplayMode | HyperLink/LinkButton/Text. Determines how to display the items. |
| FirstBulletNumber | Sets a starting number for Bulleted list when BulletStyle is set to Numbering. |
| Items | Gets the colleciton of the items in the list control. |
| BulletStyle | Circle/CustomImage/Disc/LowerAlpha/LowerRoman/Numbered/Square/UpperAlpha/UpperRoman. Determines the style of the bullet. |
| AppendDataBoundItems | Determines whether statically defined items should remain and shown when adding items dynamically. |
| DataTextField | Name of the field to set as items text. Used when DisplayMode is Hyperlink or LinkButton. |
| DataValueField | Name of the field to set as items value. Used when DisplayMode is Hyperlink or LinkButton. |
| BulletImageUrl | Used to set the Bullet Image when BulletStyle is CustomImage. |

<asp:BulletedList ID="BulletedList3" runat="Server" BorderColor="Blue" BorderWidth="1">

<asp:ListItem Text="Item 1"></asp:ListItem>

<asp:ListItem Text="Item 2"></asp:ListItem>

<asp:ListItem Text="Item 3"></asp:ListItem

</asp:BulletedList>

**Literal control**

Literal control is the rarely used control which is used to put static text on the web page.

Ideally Literal control is the rarely used control which is used to put static text on the web page.

When it is rendered on the page, it is implemented just as a simple text.

Unlike asp:Label control, there is no property like BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height etc. of Literal control. That makes it more powerful, you can even put a pure

HTML contents into it.

Select color to change the background color the cell Ex. Just a text inside Literal Control

// CODE BEHIND

// Fires when Button is clicked

protected void ChangeBackColor(object sender, EventArgs e)

{ Literal1.Text = " bgcolor='" + dropStatic.SelectedValue + "'";

litText.Text = "<div style='background-color:white;color:#000000'>Literl Control is powerful</div>";}

**Calendar control**

Calendar control is used to display one month calendar and allows to navigate backword & forward through dates, and months.

There are many properties of Calendar control to customize the functionality and appearance. However, these are some important properties that are very useful.

|  |  |
| --- | --- |
| **Properties** | **Description** |
| Caption | Gets or sets the caption for the calendar control. |
| CaptionAlign | Gets or sets the alignment for the caption. |
| CellPadding | Gets or sets the number of spaces between the data and the cell border. |
| CellSpacing | Gets or sets the space between cells. |
| DayHeaderStyle | Gets the style properties for the section that displays the day of the week. |
| DayNameFormat | Gets or sets format of days of the week. |
| DayStyle | Gets the style properties for the days in the displayed month. |
| FirstDayOfWeek | Gets or sets the day of week to display in the first column. |
| NextMonthText | Gets or sets the text for next month navigation control. The default value is >. |
| NextPrevFormat | Gets or sets the format of the next and previous month navigation control. |
| OtherMonthDayStyle | Gets the style properties for the days on the Calendar control that are not in the displayed month. |
| PrevMonthText | Gets or sets the text for previous month navigation control. The default value is <. |
| SelectedDate | Gets or sets the selected date. |
| SelectedDates | Gets a collection of DateTime objects representing the selected dates. |
| SelectedDayStyle | Gets the style properties for the selected dates. |
| SelectionMode | Gets or sets the selection mode that specifies whether the user can select a single day, a week or an entire month. |
| SelectMonthText | Gets or sets the text for the month selection element in the selector column. |
| SelectorStyle | Gets the style properties for the week and month selector column. |
| SelectWeekText | Gets or sets the text displayed for the week selection element in the selector column. |
| ShowDayHeader | Gets or sets the value indicating whether the heading for the days of the week is displayed. |
| ShowGridLines | Gets or sets the value indicating whether the gridlines would be shown. |
| ShowNextPrevMonth | Gets or sets a value indicating whether next and previous month navigation elements are shown in the title section. |
| ShowTitle | Gets or sets a value indicating whether the title section is displayed. |
| TitleFormat | Gets or sets the format for the title section. |
| Titlestyle | Get the style properties of the title heading for the Calendar control. |
| TodayDayStyle | Gets the style properties for today's date on the Calendar control. |
| TodaysDate | Gets or sets the value for today's date. |
| UseAccessibleHeader | Gets or sets a value that indicates whether to render the table header <th> HTML element for the day headers instead of the table data <td> HTML element. |
| VisibleDate | Gets or sets the date that specifies the month to display. |
| WeekendDayStyle | Gets the style properties for the weekend dates on the Calendar control. |

The Calendar control has the following three most important events that allow the developers to program the calendar control. They are:

|  |  |
| --- | --- |
| **Events** | **Description** |
| SelectionChanged | It is raised when a day, a week or an entire month is selected. |
| DayRender | It is raised when each data cell of the calendar control is rendered. |
| VisibleMonthChanged | It is raised when user changes a month. |

|  |
| --- |
| **Panel control** |
| Panel control is generally used to keep a set of controls into it. |

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| GroupingText | Its used to set the caption of the group of controls inside the panel. |
| Visible | true/false. Used to hide or show the panel. |

**Login control**

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

Following are some important properties that are very useful.

|  |  |
| --- | --- |
| **Properties of the Login Control** | |
| TitleText | Indicates the text to be displayed in the heading of the control. |
| InstructionText | Indicates the text that appears below the heading of the control. |
| UserNameLabelText | Indicates the label text of the username text box. |
| PasswordLabelText | Indicates the label text of the password text box. |
| FailureText | Indicates the text that is displayed after failure of login attempt. |
| UserName | Indicates the initial value in the username text box. |
| LoginButtonText | Indicates the text of the Login button. |
| LoginButtonType | Button/Link/Image. Indicates the type of login button. |
| DestinationPageUrl | Indicates the URL to be sent after login attempt successful. |
| DisplayRememberMe | true/false. Indicates whether to show Remember Me checkbox or not. |
| VisibleWhenLoggedIn | true/false. If false, the control is not displayed on the page when the user is logged in. |
| 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. |
| **Style of the Login Control** | |
| CheckBoxStyle | Indicates the style property of the Remember Me checkbox. |
| FailureStyle | Indicates the style property of the failure text. |
| TitleTextStyle | Indicates the style property of the title text. |
| LoginButtonStyle | Indicates the style property of the Login button. |
| TextBoxStyle | Indicates the style property of the TextBox. |
| LabelStyle | Indicates the style property of the labels of text box. |
| HyperLinkStyle | Indicates the style property of the hyperlink in the control. |
| InstructionTextStyle | Indicates the style property of the Instruction text that appears below the heading of the control. |
| **Events of the Login Control** | |
| LoggingIn | Fires before user is going to authenticate. |
| LoggedIn | Fires after user is authenticated. |
| LoginError | Fires after failure of login attempt. |
| Authenticate | Fires to authenticate the user. This is the function where you need to write your own code to validate the user. |

|  |  |
| --- | --- |
| **Log In** | |
| User Name: |  |
| Password: |  |
| Remember me next time. | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | | |  | | --- | | [Register User](http://www.dotnetfunda.com/tutorials/controls/createuser.aspx) [Forget password?](http://www.dotnetfunda.com/tutorials/controls/receoverypassword.aspx) | | |  |
| // Login Control ////////////////////////////  <asp:Login ID="Login1" runat="server" BackColor="#F7F6F3" BorderColor="#E6E2D8" BorderPadding="4"  BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em" ForeColor="#333333" OnAuthenticate="Login1\_Authenticate" OnLoginError="Login1\_LoginError">  <TitleTextStyle BackColor="#5D7B9D" Font-Bold="True" Font-Size="0.9em" ForeColor="White" />  <LoginButtonStyle BackColor="#FFFBFF" BorderColor="#CCCCCC" BorderStyle="Solid" BorderWidth="1px"  Font-Names="Verdana" Font-Size="0.8em" ForeColor="#284775" />  </asp:Login> | |

|  |
| --- |
| **LoginView control** |
| 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. |

LoginView Control ////////////////////////////

<asp:LoginView ID="LoginView1" runat="Server">

<AnonymousTemplate>

<span style="font-family: Arial; font-size: 10pt;">Welcome, Guest

<asp:LoginStatus ID="LoginStatus1" runat="Server" />

</span>

</AnonymousTemplate>

<LoggedInTemplate>

Welcome,

<asp:LoginName ID="LoginName1" runat="Server" />

<asp:LoginStatus ID="LoginStatus1" runat="Server" />

</LoggedInTemplate>

**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

Both controls allow file uploading, but the FileUpload control automatically sets the encoding of the form, whereas the HtmlInputFile does not do so.

In this tutorial, we use the FileUpload 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 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:

|  |  |
| --- | --- |
| **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:

|  |  |
| --- | --- |
| **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. |
| InputStream | Returns a stream object pointing to the uploaded file. |

For Example

Dim strname, strpath, strfullpath As String

strname = ""

If FileUpload1.HasFile Then

strname = FileUpload1.FileName

strpath = Server.MapPath("~/image/")

strfullpath = strpath + strname

FileUpload1.SaveAs(strfullpath)

End If