

Server controls

An introduction to server controls

Types of server controls

ASP.NET 2.0 uses seven different types of server controls. These seven categories correspond to the group in Visual Studio Toolbox. Below is a list of these controls, presented by categories:

Standard

- Commonly used user input controls such as labels, text boxes, and drop-down lists
- Many can be bound to a data source
- The *MulView* and *Wizard* controls

Data

- Databound user-interface controls that display data via a data source control
- Data source controls that access data from a variety of databases, XML data sources, and business objects

Validation

- Used to validate user input
- Work by running client-side script
- Can handle most validation requirements

Navigation

- Controls that provide menus and path maps for navigating a web site

Login

- Controls that provide user authentication

WebParts

- Controls that let you create a page from user-selectable components

HTML

- Standard HTML controls that can be converted to HTML server controls
- Not commonly used in ASP.NET applications

Handling control events

Code that handles a *Click* event using the *OnClick* attribute

The asp tag for a button control

```
<asp:Button id="btnCancel" runat="server" Text="Cancel Order" OnClick="btnCancel_Click" />
```

The event handler for the *Click* event of the control

```
Protected void btnCancel_Click(object sender, EventArgs e)
{
    Session.Remove("Cart");
    Response.Redirect("Order");
}
```

The asp tags for two button controls that use the same event handler

```
<asp:Button id="btnPrevious" runat="server" Text="Previous" OnClick="NavigationButtons_Click" />
<asp:Button id="btnNext" runat="server" Text="Next" OnClick="NavigationButtons_Click" />
```

Common control events

Event	Attribute	Controls
<i>Click</i>	<i>OnClick</i>	Button, image button, link button, image map
<i>Command</i>	<i>OnCommand</i>	Button, image button, link button
<i>TextChanged</i>	<i>OnTextChanged</i>	Text box
<i>SelectedIndexChanged</i>	<i>OnSelectedIndexChanged</i>	Drop-down list, list box, radio button, check box list
<i>CheckedChanged</i>	<i>OnCheckedChanged</i>	Check box, radio button

An event handler must be declared with *Public* or *Protected* scope.

Access keys with web server controls

Most Windows applications use **access keys** (also **called accelerator keys**) that let the user select controls by using keyboard shortcuts. If, for example, you designate *F* as the access key for input field that accepts a customer's first name, the user can move the focus directly to this field by pressing *Alt+F*.

A form that uses access keys and default focus and button attribute

Please enter your contact information

Name:

Email:

The asp code for the form

```
<form id="form1" runat="server" defaultfocus="txtName" defaultbutton="btnNext">
<div>
    Please enter your contact information:<br /><br />
    <table>
        <tr>
            <td style="width: 75px"><span style="text-decoration:underline">N</span>ame:</td>
            <td style="width: 100px">
                <asp:TextBox ID="txtName" runat="server"
                AccessKey="N"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td style="width: 75px"><span style="text-decoration:underline">E</span>mail:</td>
            <td style="width: 100px">
                <asp:TextBox ID="txtEmail" runat="server"
                AccessKey="E"></asp:TextBox>
            </td>
        </tr>
    </table><br />
    <asp:Button ID="btnPrevious" runat="server" AccessKey="P" Text="Previous"
        OnClick="btnPrevious_Click /">&nbsp;
    <asp:Button ID="btnNext" runat="server" AccessKey="N" Text="Next"
        OnClick="btnNext_Click /"><br />
</div>
</form>
```

```
txtEmail.Focus();
```

A button, a link button, and an image button

[illegible]

Attribute	Description
<i>Text</i>	(<i>Button</i> and <i>LinkButton</i> controls only) The text displayed by the button. For a <i>LinkButton</i> control, the text can be coded as content between the start and end tags or as the value of the <i>Text</i> attribute.
<i>ImageUrl</i>	(<i>ImageButton</i> control only) The image to be displayed for the button.
<i>AlternateUrl</i>	(<i>ImageButton</i> control only) The text to be displayed if the browser can't display the image.
<i>CausesValidation</i>	Determines whether page validation occurs when you click the button. The default is <i>True</i> .
<i>CommandName</i>	A string value that's passed to the <i>Command</i> event when a user clicks the button.
<i>CommandArgument</i>	A string value that's passed to the <i>Command</i> event when a user clicks the button.
<i>PostBackUrl</i>	The URL of the page that should be requested when the user clicks the button.

An event handler for the *Click* event of the button control

```
protected void btnAdd_Click(object sender, EventArgs e)
{
    if (Page.IsValid)
    {
        CartItem item = new CartItem();

        item.Product = selectedProduct;
        item.Quantity = Convert.ToInt32(txtQuantity.Text);

        this.AddToCart(item);

        Response.Redirect("Cart.aspx");
    }
}
```

Using *e* argument of an image button control

An image used for an image button control



The asp tag for the control

```
<asp:ImageButton ID="ibtnNavigate" runat="server"
    ImageUrl="~/Image/navbuttons.gif" Height="24px" Width="96px"
    OnClick="ibtnNavigate_Click" />
```

An event handler for the *Click* event of the control

```
Protected void ibtnNavigate_Click(object sender, ImageClickEventArgs e)
{
    If (e.X >= 0 & e.X <= 23)
        this.GoToFirstRow();
    If (e.X >= 24 & e.X <= 47)
        this.GoToPreviousRow();
    If (e.X >= 48 & e.X <= 71)
        this.GoToNextRow();
    If (e.X >= 72 & e.X <= 95)
        this.GoToLastRow();
}
```

Properties of the *ImageClickEventArgs* class

Property	Description
<i>X</i>	An integer that represents the <i>X</i> coordinate where the user clicked the image button.
<i>Y</i>	An integer that represents the <i>Y</i> coordinate where the user clicked the image button.

Using button controls that use the *CommandName* attribute

Four controls that use the *CommandName* attribute

```
<asp:Button ID="btnFirst" runat="server" Text="<<" Width="25px" CommandName="First"
    OnCommand="NavigationButtons_Command" />
```

```
<asp:Button ID="btnPrevious" runat="server" Text="<" Width="25px" CommandName="Previous"
    OnCommand="NavigationButtons_Command" />
```

```
<asp:Button ID="btnNext" runat="server" Text=">>" Width="25px" CommandName="Next"
    OnCommand="NavigationButtons_Command" />
```

```
<asp:Button ID="btnLast" runat="server" Text=">" Width="25px" CommandName="Last"
    OnCommand="NavigationButtons_Command" />
```

An event handler for the *Command* events of the buttons

```
Protected void NavigationButtons_Command(object sender, CommandEventArgs e)
```

```
{
    Case "First":
        this.GoToFirstRow();
        break;
    Case "Previous":
        this.GoToPreviousRow();
        break;
    Case "Next":
        this.GoToNextRow();
        break;
    Case "Last":
        this.GoToLastRow();
        break;
}
```

Properties of the *CommandEventArgs* class

Property	Description
<i>CommandName</i>	The value specified in the <i>CommandName</i> property for the control that generated the <i>Command</i> event.
<i>CommandArgument</i>	The value specified in the <i>CommandArgument</i> property for the control that generated the <i>Command</i> event.

Text boxes, labels, checkboxes, and radio buttons

The tag for the text box

```
<asp:TextBox ID="txtQuestion" runat="server" Rows="5" TextMode="MultiLine" Width="296px">
</asp:TextBox>
```

Common text box attributes

Attribute	Description
<i>TextMode</i>	The type of text box. <i>SingleLine</i> creates a standard text box, <i>MultiLine</i> creates a text box that accepts more than one line of text, and <i>Password</i> causes the characters that are entered to be masked. The default is <i>SingleLine</i> .
<i>Text</i>	The text content of the text box.
<i>MaxLength</i>	The maximum number of characters that can be entered into the text box.
<i>Wrap</i>	Determines whether or not text wraps automatically when it reaches the end of a multi-line text box. The default is True.
<i>ReadOnly</i>	Determines whether the user can change the text in the text box. The default value is False, which means that the text can be changed.
<i>Columns</i>	The width of the text box in characters. The actual width is determined based on the font that's used for the text entry.
<i>Rows</i>	The height of a multi-line text box in lines. The default value is 0, which sets the height to a single line.

The asp tag for the label

```
<asp:Label ID="lblConfirm" runat="server" ></asp:Label>
```

Common label attribute

Attribute	Description
<i>Text</i>	The text displayed by the label

The asp code for the check box and radio buttons

```
<asp:CheckBox ID="chkMail1" runat="server" Checked="True"
    Text="Add me to your mailing list" /><br /><br />
Contact me about:<br />
<asp:CheckBox ID="chkSpecial" runat="server" Text="Special offers" /><br />
<asp:CheckBox ID="chkNew" runat="server" Text="New products" /><br />
<asp:CheckBox ID="chkRelated" runat="server" Text="Related products" /><br />
Contact me by:<br />
<asp:RadioButton ID="rdoEMail" runat="server" Checked="True" GroupName="Contact"
    Text="Email" />&nbsp;
<asp:RadioButton ID="rdoPostal" runat="server" GroupName="Contact"
    Text="Postal mail" />&nbsp;
```

Common check box and radio buttons attributes

Attribute	Description
<i>Text</i>	The text that's displayed next to the check box or radio button.
<i>Checked</i>	Indicates whether the check box or radio button is selected. The default value is <i>False</i> .
<i>GroupName</i>	The name of the group that the control belongs to (radio button only)

Code that retrieves the text entered by user

```
string question = txtQuestion.Text;
```

Code that changes the Text property of the label

```
lblConfirm.Text = "Thank you for your question.<br />" + "We will respond within 2 business days.";
```

Code that processes the first check box and radio button

```
protected void btnContinue_Click(object sender, EventArgs e)
{
    if (chkMail.Checked)
        customer.Mail = true;
    else
        customer.Mail = false;

    if (rdoEmail.Checked)
        customer.MailType = "Email";
    else
        customer.MailType = "Postal";
}
```


Another way to process the check box and radio buttons

```
protected void chkMail_CheckChanged(object sender, EventArgs e)
{
    if (chkMail.Checked)
        customer.Mail = true;
    else
        customer.Mail = false;
}

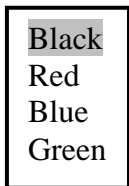
protected void rdoEmail_CheckChanged(object sender, EventArgs e)
{
    customer.MailType = "Email";
}

protected void rdoPostal_CheckChanged(object sender, EventArgs e)
{
    customer.MailType = "Postal";
}
```

List controls

A list box displayed in a browser

Color



The asp tag for the list box

```
<asp:ListBox ID="lstColor" runat="server">
    <asp:ListItem Value="Black" Selected="True">Black</asp:ListItem>
    <asp:ListItem Value="Red">Red</asp:ListItem>
    <asp:ListItem Value="Blue">Blue</asp:ListItem>
    <asp:ListItem Value="Green">Green</asp:ListItem>
</asp:ListBox>
```

A drop-down list box displayed in a browser

Day of week:



The asp tag for the drop-down list

```
<asp:DropDownList id="ddlDay" runat="server">
    <asp:ListItem Value="1">Sunday</asp:ListItem>
    <asp:ListItem Value="2">Monday</asp:ListItem>
    <asp:ListItem Value="3">Tuesday</asp:ListItem>
    <asp:ListItem Value="4">Wednesday</asp:ListItem>
    <asp:ListItem Value="5">Thursday</asp:ListItem>
    <asp:ListItem Value="6">Friday</asp:ListItem>
    <asp:ListItem Value="7">Saturday</asp:ListItem>
</asp:DropDownList>
```

Common properties of list box and drop-down list controls

Property	Description
<i>Items</i>	The collection of <i>ListItem</i> objects that represents the items in the control. This property returns an object of type <i>ListItemCollection</i> .
<i>Rows</i>	The number of items that are displayed in a list box at one time. If the list contains more rows than can be displayed, a scroll bar is added automatically.
<i>SelectedItem</i>	The <i>ListItem</i> object for the currently selected item or the <i>ListItem</i> object for the item with the lowest index if more than one item is selected in a list box.
<i>SelectedIndex</i>	The index of the currently selected item, or the index of the first selected item if more than one item is selected in a list box. If no item is selected in a list box, the value of this property is -1.
<i>SelectedValue</i>	The value of the currently selected item, or the value of the first selected item if more than one item is selected in a list box. If no item is selected in a list box, the value of this property is an empty string ("").
<i>SelectionMode</i>	Indicates whether a list box allows single selections (<i>Single</i>) or multiple selections.

Common properties of list item objects

Property	Description
<i>Text</i>	The text that's displayed for the list item.
<i>Value</i>	A string value associated with the list item.
<i>Selected</i>	Indicates whether the item is selected.

Code that retrieves the value of a selected item in a drop-down list

```
Int dayNumber = Convert.ToInt32(ddlDay.SelectedValue);
```

Code that retrieves the text for a selected item in a drop-down list

```
string dayName = ddlDay.selectedItem.Text;
```

Code that uses the *SelectedIndexChanged* event of a drop-down list

```
Protected void ddlDay_SelectedIndexChanged(object sender, EventArgs e)
{
    int dayNumber = Convert.ToInt32(ddlDay.SelectedValue);
}
```

Common members of list item collection objects

Property	Description
<i>Count</i>	The number of items in the collection.
Indexer	Description
<i>[index]</i>	A list object that represents the item at the specified index.
Method	Description
<i>Add(string)</i>	Adds a new item to the end of the collection, and assigns the specified string value to both the <i>Text</i> and <i>Value</i> properties of the item.
<i>Add(ListItem)</i>	Adds the specified list item to end of the collection.
<i>Insert(index, string)</i>	Inserts an item at the specified index location in the collection, and assigns the specified string value to the <i>Text</i> property of the item.
<i>Insert(index, ListItem)</i>	Inserts the specified list item at the specified index location in the collection.
<i>Remove(string)</i>	Removes from the collection the item whose <i>Text</i> property is equal to the specified string value.
<i>Remove(ListItem)</i>	Removes the specified list from the collection.
<i>RemoveAt(index)</i>	Removes the item at the specified index location from the collection.
<i>Clear()</i>	Removes all the items from the collection.
<i>FindByValue(string)</i>	Returns the list item whose <i>Value</i> property has the specified value.
<i>FindByText(string)</i>	Returns the list item whose <i>Text</i> property has the specified value.

Code that loads items into a list box using strings

```
lstColor.Items.add("Black");  
lstColor.Items.add("Red");  
lstColor.Items.add("Blue");  
lstColor.Items.add("Green");
```

Code that loads items into a drop-down list using ListItem objects

```
ddlDay.Items.Add(new ListItem("Sunday", "1"));  
ddlDay.Items.Add(new ListItem("Monday", "2"));  
ddlDay.Items.Add(new ListItem("Tuesday", "3"));  
ddlDay.Items.Add(new ListItem("Wednesday", "4"));  
ddlDay.Items.Add(new ListItem("Thursday", "5"));  
ddlDay.Items.Add(new ListItem("Friday", "6"));  
ddlDay.Items.Add(new ListItem("Saturday", "7"));
```

Radio button lists and check box lists

The asp tag for the check box list

```
<asp:CheckBoxList id="cblContact" runat="server" Width="305px" RepeatColumns="2">  
    <asp:ListItem Value="Special">Special offers</asp:ListItem>  
    <asp:ListItem Value="New">New product</asp:ListItem>  
    <asp:ListItem Value="Related">Related products</asp:ListItem>  
    <asp:ListItem Value="Events">Local events</asp:ListItem>  
</asp:CheckBoxList>
```

The asp tag for the radio button list

```
<asp:RadioButtonList id="rblMail" runat="server" Width="346px" RepeatDirection="Horisontal">  
    <asp:ListItem Value="Email">Email</asp:ListItem>  
    <asp:ListItem Value="Postal">Postal</asp:ListItem>  
    <asp:ListItem Value="Both" Selected="True">Both</asp:ListItem>  
</asp:RadioButtonList>
```

Attributes for formatting radio buttons and check box lists

Attribute	Description
<i>RepeatLayout</i>	Specifies whether ASP.NET should use table tags (<i>Table</i>) or normal HTML flow (<i>Flow</i>) to format the list when it rendered the control. The default is <i>Table</i> .
<i>RepeatDirection</i>	Specifies the direction in which the control should be presented. The available values are <i>Horizontal</i> and <i>Vertical</i> . The default is <i>Vertical</i> .
<i>RepeatColumns</i>	Specifies the number of columns to use when repeating the controls. The default is 0.

A statement that gets the value of the selected item in a radio button list

```
customer.MailType = rblMail.Selected.Value;
```

A statement that checks if the list item in a check box list is selected

```
if (cblContact.Items[0].Selected) ...
```

Bulleted lists and numbered lists

A bulleted list and numbered list displayed in a browser

Materials you will need:	Selected a project type:
<ul style="list-style-type: none">• Styrofoam panel• Gray and black latex paint• Stone texture paint• Rotary tool	<ol style="list-style-type: none">1. Customers2. Static props3. Animated props

Common attributes of the bulleted list control

Attribute	Description
<i>BulletStyle</i>	Specifies the bullet style. For a bulleted list, allowable values are <i>Disc</i> , <i>Circle</i> , <i>Square</i> , or <i>CustomImage</i> . For a numbered list, allowable values are <i>Numbered</i> , <i>LowerAlpha</i> , <i>UpperAlpha</i> , <i>LowerRoman</i> , or <i>UpperRoman</i> .
<i>BulletImageUrl</i>	Specifies the URL of the image used to display the bullets if the <i>BulletStyle</i> attribute is set to <i>CustomImage</i> .
<i>FirstBulletNumber</i>	Specifies the starting numbers if number are displayed.
<i>DisplayMode</i>	Specifies how the text for each item should be displayed. Allowable values are <i>Text</i> , <i>HyperLink</i> , or <i>LinkButton</i> . <i>Text</i> is default.

The asp code for the bulleted list shown above

```
<asp:BulletedList ID="BulletedList1" runat="server" BulletedStyle="Disc">
    <asp:ListItem>Styrofoam panel</asp:ListItem>
    <asp:ListItem>Gray and black latex paint</asp:ListItem>
    <asp:ListItem>Stone texture paint</asp:ListItem>
    <asp:ListItem>Rotary tool</asp:ListItem>
</asp:BulletedList>
```

The asp code for the bulleted list shown above

```
<asp:BulletedList ID="BulletedList2" runat="server" BulletedStyle="Numbered"
DisplayMode="HyperLink">
    <asp:ListItem Value="Customer.aspx">Customer</asp:ListItem>
```

```

    <asp:ListItem Value="StaticProps.aspx">Static props</asp:ListItem>
    <asp:ListItem Value="AnimatedPropes.aspx">Animated props</asp:ListItem>
</asp:BulletedList>

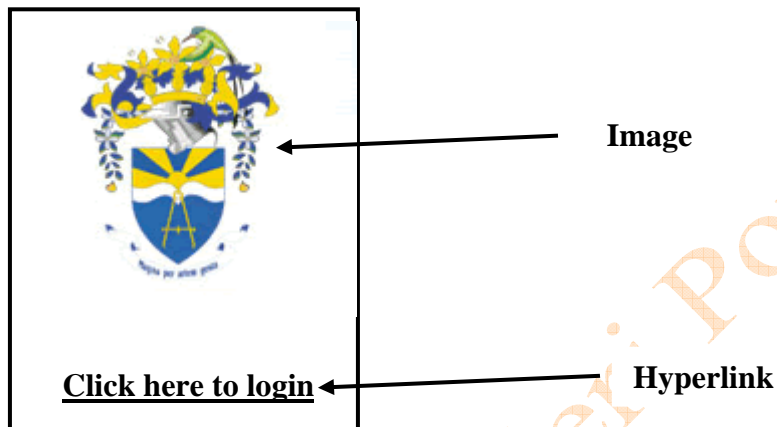
```

A statement that checks if the first link button in a bulleted list was clicked

If (blCategories.Item[0].Selected) . . .

Image and hyperlink controls

An image control and hyperlink control displayed in a browser



The asp tag for the image control

```

<asp:Image ID="Image1" runat="server" ImageUrl="~/Image/UtechLogo.jpg" AlternateText="Utech
logo" />

```

Code that sets the URL of an Image control

```
imgProduct.ImageUrl = "Image/Product/cat01.jpg";
```

Common image attributes

Attribute	Description
<i>ImageUrl</i>	The absolute or relative URL of the image.
<i>AlternateText</i>	The text that's used in place of the image if the browser can't display the image.
<i>ImageAlign</i>	The alignment of the image relative to the web page or other elements on the page.
<i>Width</i>	The width of the image.
<i>Height</i>	The height of the image.

The asp tag for the hyperlink control

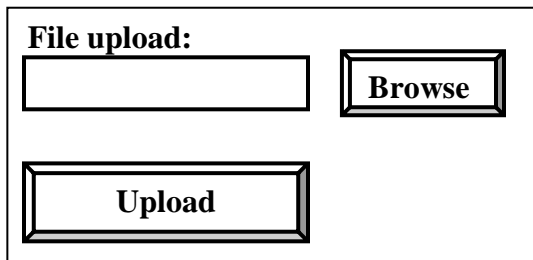
```
<asp:Hyperlink ID="Hyperlink1" runat="server" NavigateUrl=http://www.utech.edu/">Click here to login</asp:Hyperlink>
```

Common hyperlink attributes

Attribute	Description
<i>NavigateUrl</i>	The absolute or relative URL of the page that's displayed when the control is clicked.
<i>Text</i>	The text that's displayed for the control.
<i>ImageUrl</i>	The absolute or relative URL of the image that's displayed for the control

A file upload control

A file upload control displayed in a browser



File upload:

The asp code used to implement the file upload

```
File upload:<br /><br />
<asp:FileUpload ID="uplCustList" runat="server" /><br /><br />
<asp:Button ID="btnUpload" runat="server" Text="Upload" OnClick="btnUpload_Click" /><br /><br />
<asp:Label ID="lblMessage" runat="server"></asp:Label>
```

The Click event handler for the Upload button

```
protected void btnUpload_Click(object sender, EventArgs e)
{
    int sizeLimit = 5242880; // 5,242,880 is 5MB

    if (FileUpload1.PostedFile.ContentLength <= sizeLimit)
    {
        string path = "C:\\uploads\\" + FileUpload1.FileName;

        FileUpload1.SaveAs(path);
    }
}
```

```

        lblMessage.Text = "File upload to " + path;
    }
    else
        lblMessage.Text = "File exceeds size limit";
}

```

For this code to work, the user account that ASP.NET runs must have write access to the directory that the file is saved to. To grant that access, use Window Explorer to navigate to the directory, right-click it, and chose Properties. Then, click the Security tab, click the Add button, add the ASP.NET *Machine Account (machinename/ASPNET)*, and grant *Modify* and *Write* access to the account. The event handler uses *PostedFile.ContentLength* property to determine the size of the uploaded file. Then, if this value exceeds the limit set by the *sizeLimit* variable, the file isn't saved. Instead, an error message is displayed.

Properties and methods of the *FileUpload* class

Property	Description
<i>HasFile</i>	If <i>True</i> , the user has selected a file to upload.
<i>FileName</i>	The name of the file to be uploaded.
<i>PostedFile</i>	The <i>HttpPostedFile</i> object that represents the file that was posted. You can use this object's <i>ContentLength</i> property to determine the size of the posted file.
Method	Description
<i>SaveAs(string)</i>	Saves the posted file to the specified path.