



INTERNET & WEB APPLICATION DEVELOPMENT SWE 444



Fall Semester 2008-2009 (081)

Module 5.2: Introduction to ASP.NET and Web Forms

Dr. El-Sayed El-Alfy

Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa

Objectives/Outline

• Objectives

- Understand the server controls and how to use them

• Outline

- HTML Server Control
- Web Controls
 - Intrinsic controls
 - List controls
 - Rich controls
 - Validation controls

HTML Controls

- Work well with existing HTML designers
 - i.e., when converting existing ASP 3.0 Web pages to ASP.NET Web pages
 - Easier to change to HTML server controls than to Web server controls
- Properties map 1:1 with HTML
 - `table.bgcolor = "red";`
- Can specify client-side event handlers
- Good when quickly converting existing pages
- Derived from `System.Web.UI.HtmlControls.HtmlControl`
- Supported controls have custom class, others derive from `HtmlGenericControl`
 - `HtmlAnchor` controls the `<a>` element
 - `HtmlButton` controls the `<button>` element
 - `HtmlForm` controls the `<form>` element
 - etc

HTML Controls (cont.)

- Can use controls two ways:
 - Handle everything in action events (e.g. button click)
 - Event code will read the values of other controls (e.g. text, check boxes, radio buttons, select lists)
 - Handle change events as well as action events

Examples

HTMLControls1.aspx

```
<%@ Page Language="c#" %>
<script runat="server">
    public void accumulate (object sender, System.EventArgs e) {
        int a, b;
        if(IsPostBack){
            a = int.Parse(number.Value);
            b = int.Parse(label.InnerText);
            label.InnerText = (a + b).ToString();
        }
    }
</script>
<html >
<head>
</head>
<body>
    <form method="post" runat="server">
        Enter a number: <input type="text" id="number" runat="server"/>
        Sum: <span id="label" runat="server"> 0 </span><br />
        <input type="submit" onserverclick="accumulate" value="Add" runat="server"/>
    </form>
</body>
</html >
```

➤ Another Example: HTMLControls2.aspx

Web Controls

➤ Web controls appear in HTML markup as namespaced tags with prefix asp:

➤ Example

```
<asp:button onclick="button1_click" runat="server">
<asp:textbox onchange="text1_changed" runat="server">
```

➤ Defined in System.Web.UI.WebControls namespace

- Which is automatically mapped to the asp: prefix

➤ Provide extensive properties used to control display and format, e.g.

- Font
- BackColor, ForeColor
- BorderColor, BorderStyle, BorderWidth
- Style, CssClass
- Height, Width
- Visible, Enabled

Types of Web Controls

➤ Four types of Web Controls

- Intrinsic controls
- List controls
- Rich controls
- Validation controls

Intrinsic Controls

➤ Correspond to HTML controls

➤ Supported controls

- <asp:button>
- <asp:imagebutton>
- <asp:linkbutton>
- <asp:hyperlink>
- <asp:textbox>
- <asp:checkbox>
- <asp:radiobutton>
- <asp:image>
- <asp:label>
- <asp:panel>
- <asp:table>

➤ TextBox, CheckBox and their subclasses don't automatically do a postback when their controls are changed

➤ Specify AutoPostBack="true" to make change events cause a postback

Examples

➤ Image

```
20 <asp:Image ID="userInfoOnImage" runat="server"
21 EnableViewState="False" ImageUrl="~/images/user.png" /> &nbsp;
```

➤ TextBox

```
36 <asp:TextBox ID="lastNameTextBox" runat="server"
37 EnableViewState="False"></asp:TextBox>
```

➤ Hyperlink

```
76 <asp:HyperLink ID="booksHyperLink" runat="server"
77 EnableViewState="False" NavigateUrl="http://www.deitel.com"
78 Target="_blank">
79 Click here to view more information about our books
80 </asp:HyperLink>
```

➤ Button

```
99 <asp:Button ID="registerButton" runat="server"
100 EnableViewState="False" Text="Register" />
```

List Controls

➤ Controls that handle repetition

➤ Supported controls

- <asp:dropdownlist>
- <asp:listbox>
- <asp:radiobuttonlist>
- <asp:checkboxlist>
- <asp:repeater>
- <asp:datalist>
- <asp:datagrid>

➤ Repeater, DataList, and DataGrid controls

- Powerful, customizable list controls
- Expose templates for customization
- Can contain other controls
- Provide event bubbling through their OnItemCommand event
- More about these controls and templates later

List Controls (cont.)

- CheckBoxList Provides a collection of check box controls
- Can be populated via data binding

```
<asp:CheckBoxList id="Check1" runat="server">
  <asp:ListItem>Item 1</asp:ListItem>
  <asp:ListItem>Item 2</asp:ListItem>
  <asp:ListItem>Item 3</asp:ListItem>
  <asp:ListItem>Item 4</asp:ListItem>
  <asp:ListItem>Item 5</asp:ListItem>
</asp:CheckBoxList>
```

List Controls (cont.)

➤ RadioButtonList Example

```
89 <asp:RadioButtonList ID="operatingSystemRadioButtonList"
90 runat="server" EnableViewState="False">
91 <asp:ListItem>Windows XP</asp:ListItem>
92 <asp:ListItem>Windows 2000</asp:ListItem>
93 <asp:ListItem>Windows NT</asp:ListItem>
94 <asp:ListItem>Linux</asp:ListItem>
95 <asp:ListItem>Other</asp:ListItem>
96 </asp:RadioButtonList>
```

➤ DropDownList Example

```
64 <asp:DropDownList ID="booksDropDownList" runat="server"
65 EnableViewState="False">
66 <asp:ListItem>Visual Basic 2005 How to Program 3e
67 </asp:ListItem>
68 <asp:ListItem>Visual C# 2005 How to Program 2e
69 </asp:ListItem>
70 <asp:ListItem>Java How to Program 6e</asp:ListItem>
71 <asp:ListItem>C++ How to Program 5e</asp:ListItem>
72 <asp:ListItem>XML How to Program 1e</asp:ListItem>
73 </asp:DropDownList>
```

Intrinsic & Simple List Controls

- Demo 1: WebControls1.aspx
 - Assorted intrinsic and list controls
- Demo 2: WebControls2.aspx
 - Same controls with AutoPostBack



Rich Controls

- Custom controls with rich functionality
- Supported Controls
 - <asp: calendar>
 - <asp: adrotator>
- More will be added
- 3rd party controls are coming
- Demo: RichControls1.aspx

AdRotator control

- Randomly selects an image to display
- Generates a hyperlink to the Web page associated with that image
- Browsers that do not support images display alternate text that is specified in the XML document
- The browser loads the Web page associated with that image when a user clicks the image or substituted text
- Example

```
18 <asp: AdRotator ID="countryRotator" runat="server"
19   DataSourceID="adXmlDataSource" />
20 <asp: XmlDataSource ID="adXmlDataSource" runat="server"
21   DataFile="~/App_Data/AdRotatorInformation.xml">
22 </asp: XmlDataSource>
```

AdRotator control (cont.)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Fig. 25.16: AdRotatorInformation.xml -->
3 <!-- XML file containing advertisement information. -->
4 <Advertisements>
5   <Ad>
6     <ImageUrl>images/france.png</ImageUrl>
7     <NavigateUrl>http://www.cla.gov/library/publications/
8       the-world-factbook/geos/fr.html
9     </NavigateUrl>
10    <AlternateText>France Information</AlternateText>
11    <Impressions>1</Impressions>
12  </Ad>
13
14  <Ad>
15    <ImageUrl>images/germany.png</ImageUrl>
16    <NavigateUrl>http://www.cla.gov/library/publications/
17      the-world-factbook/geos/gm.html
18    </NavigateUrl>
19    <AlternateText>Germany Information</AlternateText>
20    <Impressions>1</Impressions>
21  </Ad>
22 </Advertisements>
```

Validation Controls

- Determines whether the data in another Web control is in the proper format
- Rich, declarative validation
- Validation declared separately from input control
- Extensible validation framework
- Supports validation on client and server
 - Automatically detects uplevel clients
 - Avoids roundtrips for uplevel clients
- For security reasons, server-side validation is always performed for important validation when client-side cannot be trusted
 - Prevents users from spoofing Web Forms

Validation Controls (cont.)

- <asp: RequiredFieldValidator>
 - Ensures that a value is entered
- <asp: RangeValidator>
 - Checks if value is within minimum and maximum values
- <asp: CompareValidator>
 - Compares value against constant, another control or data type
- <asp: RegularExpressionValidator>
 - Tests if value matches a predefined pattern (a regular expression)
 - That expression is assigned to ValidationExpression property
- <asp: CustomValidator>
 - Lets you create custom client- or server-side validation function
- <asp: ValidationSummary>
 - Displays list of validation errors in one place

Validation Controls (cont.)

- Validation controls are derived from System.Web.UI.WebControls.BaseValidator, which is derived from the Label control
- Validation controls contain text which is displayed only if validation fails
- Text property is displayed at control location
- ErrorMessage is displayed in summary if validation fails

Validation Controls (cont.)

- Validation controls are associated with their target control using the ControlToValidate property

```
<asp: TextBox id="TextBox1" runat="server" />  
  
<asp: RequiredFieldValidator id="Req1"  
  ControlToValidate="TextBox1"  
  Text="Required Field" runat="server" />
```

- Can create multiple validation controls with the same target control

Validation Controls (cont.)

- Page.IsValid indicates if all validation controls on the page succeed

```
void Submit_Click(object s, EventArgs e) {  
    if (Page.IsValid) {  
        Message.Text = "Page is valid!";  
    }  
}
```

Validation Controls (cont.)

- Display property controls how the error message is shown
 - Static: fixed layout, display won't change if invalid; Space for the validation message is allocated in the page layout
 - Dynamic: Space for the validation message is dynamically added to the page if validation fails
 - None: Validation message is never displayed; can still use ValidationSummary and Page.IsValid
- Type property specifies expected data type: Currency, Date, Double, Integer, String
- Can force down-level option
 - Only server-side validation

```
<% @ Page Language="c#" ClientTarget="DownLevel" %>
```

Validation Control Demo



```
1 <!-- Fig. 25.17: Validation.aspx -->  
2 <!-- Form that demonstrates using validators to validate user input. -->  
3 <%@ Page Language="VB" AutoEventWireup="false"  
4 CodeFile="Validation.aspx.vb" Inherits="Validation" %>  
5  
6 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
7 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
8  
9 <html xmlns="http://www.w3.org/1999/xhtml" >  
10 <head runat="server">  
11 <title>Demonstrating Validation Controls</title>  
12 </head>  
13 <body>  
14 <form id="form1" runat="server">  
15 <div>  
16 Please fill out the following form.<br /><em>All fields are  
17 required and must contain valid information.</em><br />  
18 <br />  
19 <table>  
20 <tr>  
21 <td style="width: 100px; vertical-align: top">  
22 Name: </td>  
23 <asp:TextBox ID="nameTextBox" runat="server">  
24 </asp:TextBox><br />  
25 <asp:RequiredFieldValidator  
26 ID="nameInputValidator" runat="server"  
27 ControlToValidate="nameTextBox" Display="Dynamic"  
28 ErrorMessage="Please enter your name.">  
29 </asp:RequiredFieldValidator>  
30
```

Outline

Validation.aspx

(1 of 5)

Defines a RequiredFieldValidator which confirms the TextBox is not empty

```

31 </td>
32 </tr>
33 <tr>
34 <td style="width: 100px; valign="top">E-mail address:</td>
35 <td style="width: 450px; valign="top">
36 <asp:TextBox ID="emailTextBox" runat="server">
37 </asp:TextBox>
38 &nbsp;e.g., user@domain.com<br />
39 <asp:RegularExpressionValidator
40 ID="emailInputValidator" runat="server"
41 ControlToValidate="emailTextBox" Display="Dynamic"
42 ErrorMessage="Please enter your e-mail address.">
43 </asp:RegularExpressionValidator>
44 <asp:RegularExpressionValidator
45 ID="emailFormatValidator" runat="server"
46 ControlToValidate="emailTextBox" Display="Dynamic"
47 ErrorMessage=
48 "Please enter an e-mail address in a valid format."
49 ValidationExpression=
50 "[\w+([-+.]|\w+)*@\w+([-.]|\w+)*\.\w+([-.]|\w+)*">
51 </asp:RegularExpressionValidator>
52 </td>
53 </tr>
54 <tr>
55 <td style="width: 100px; height: 21px; valign="top">
56 Phone number:</td>
57 <td style="width: 450px; height: 21px; valign="top">
58 <asp:TextBox ID="phoneTextBox" runat="server">
59 </asp:TextBox>

```

Outline

Defines a RegularExpressionValidator which confirms the TextBox is not empty

Defines a RegularExpressionValidator control to determine whether data is entered in a valid format



```

60 &nbsp;e.g., (555) 555-1234<br />
61 <asp:RegularExpressionValidator
62 ID="phoneInputValidator" runat="server"
63 ControlToValidate="phoneTextBox" Display="Dynamic"
64 ErrorMessage="Please enter your phone number.">
65 </asp:RegularExpressionValidator>
66 <asp:RegularExpressionValidator
67 ID="phoneFormatValidator" runat="server"
68 ControlToValidate="phoneTextBox" Display="Dynamic"
69 ErrorMessage=
70 "Please enter a phone number in a valid format."
71 ValidationExpression=
72 "[(\d{3}-?)?(\d{3}-)\d{3}-\d{4}]"
73 </asp:RegularExpressionValidator>
74 </td>
75 </tr>
76 </table>
77 <br />
78 <asp:Button ID="submitButton" runat="server" Text="Submit" /><br />
79 <br />
80 <asp:Label ID="outputLabel" runat="server"
81 Text="Thank you for your submission." Visible="False">
82 </asp:Label >
83 </div>
84 </form>
85 </body>
86 </html >

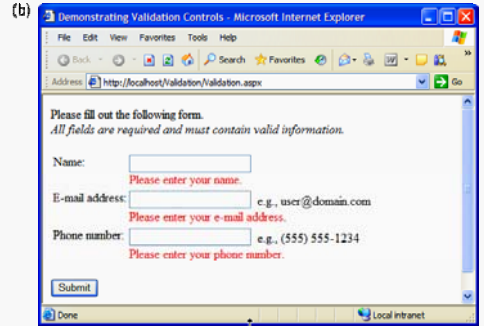
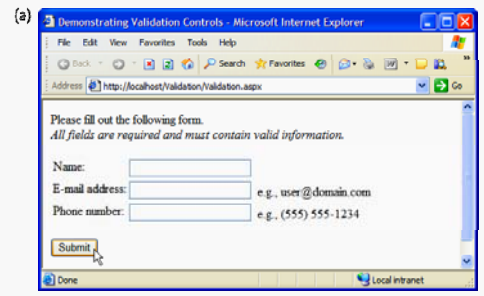
```

Outline

Validation.aspx (3 of 5)

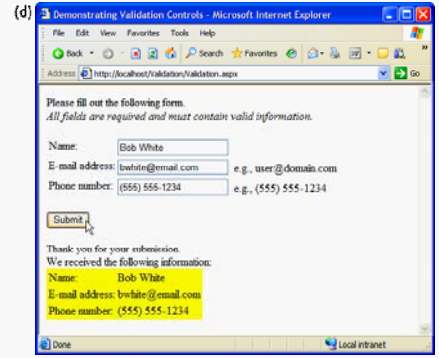
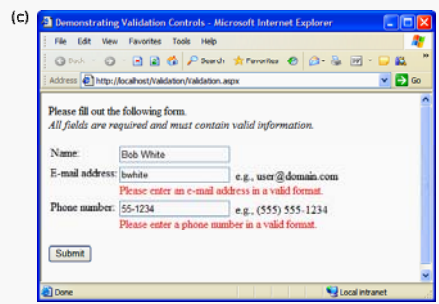
Defines a RegularExpressionValidator which confirms the TextBox is not empty

Defines a RegularExpressionValidator control to determine whether data is entered in a valid format



Outline

Validation.aspx (4 of 5)



Outline

Validation.aspx (5 of 5)



Q & A



References

- H. M. Deitel, P. J. Deitel, and A. B. Goldberg, [*Internet and World Wide Web How to Program*](#), 4/e, Pearson Education Inc., 2008.
- Some useful links with examples and other resources:
 - <http://msdn.microsoft.com/net/aspnet/default.asp>
 - The Official Microsoft ASP.NET Site www.asp.net
 - ASP.NET QuickStart Tutorial
 - <http://quickstarts.asp.net/QuickstartV20/aspnet/>
 - <http://samples.gotdotnet.com/quickstart/aspplus/>
 - W3School ASP.NET Tutorial
 - <http://www.w3schools.com/ASPNET/default.asp>
 - ASP.NET at wikipedia <http://en.wikipedia.org/wiki/ASP.NET>
 - <http://www.asptoday.com/>
 - <http://www.aspfree.com/>
 - <http://www.devx.com/dotnet/>
 - <http://msdn.microsoft.com>
 - Several Online Presentations