

ASP.NET Validation

Madhuri Sawant

Validation

- A validation control is a type of ASP.NET control that's used to validate input data.
- Use validation controls to test user input and produce error messages.
- The validation is performed when the focus leaves the control that's being validated and also when the user clicks on a button control whose Causes Validation property is set to True
- Each validation control is associated with a specific server control, but one or more validation control can be associated with a single server control .
- Validation tests are typically done on the client before the page is posted to the server

Common Validator properties

Property	Description
ControlToValidate	The ID of the control to be validated
Display	Determines how the error message is to be displayed .Specify static to allocate space for the message in the page layout ,Dynamic to have space allocated only when an error occurs ,or None to display errors only in a validation summary controls .
ErrorMessage	The message that's displayed in the validator and/or the validation summary control when the validation fails
Text	The message that's displayed in the validator when you use the ErrorMessage property to display a message in the validation summary control.
Enabled	If set to True ,The validation test for the validator is performed
EnableClientScript	If set to True ,Client Side script for the validation is generated.If this property is set to False ,the validation is only done on the server .
IsValid	Indicates whether the control specifies in the ControlToValidate property passed the validation
ValidationGroup	Indicates which group the validation control is part of .

RequiredFieldValidator

- ➔ Required field validator checks that the user entered a value into an input control . If the user doesn't enter a value ,the validator ' s error message is displayed .
- ➔ Typically used with text box controls ,but can also be used with list controls.
- Additional property of the required field Validator

Property	Description
InitialValue	The initial value of the control that's validated .If this value isn't changed ,the validation fails .The default is an empty string

A required filed Validator that checks for a required entry

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

```
<asp:RequiredFieldValidator  
ID="RequiredFieldValidator1"  
runat="server"  
ControlToValidate="TextBox1"  
ErrorMessage="You must enter a name">  
</asp:RequiredFieldValidator>
```

A required field validator that checks that an initial value is changed

```
<asp:TextBox ID="TextBox2" runat="server">mm/dd/yyyy</asp:TextBox>  
<asp:RequiredFieldValidator  
ID="RequiredFieldValidator2"  
runat="server"  
ControlToValidate="TextBox2"  
InitialValue="mm/dd/yyyy"  
ErrorMessage="You must enter a birth date">  
</asp:RequiredFieldValidator>
```

A required field validator that forces an option to be chosen from a list box

```
<asp:ListBox ID="ListBox1" runat="server">  
    <asp:ListItem Selected ="True" Value="None"></asp:ListItem>  
    <asp:ListItem>Red</asp:ListItem>  
    <asp:ListItem>Green</asp:ListItem>  
    <asp:ListItem>Blue</asp:ListItem>  
</asp:ListBox>  
<asp:RequiredFieldValidator ID="RequiredFieldValidator3"  
runat="server"  
    ControlToValidate="ListBox1" InitialValue="None" ErrorMessage="You  
must select a colour"></asp:RequiredFieldValidator>
```

Compare Validator

- The compare validator compares the value entered into a control with a constant value or with the value entered into another control. Also can be used to check that the user entered a specific data type.

Property	Description
ValueToCompare	The value that the control specified in the ControlToValidate property should be compared to
Operator	The type of comparison to perform (Equal,NotEqual, GreaterThan,GreaterThanEqual,LessThan,LessThanEqual, DataType)
Type	The data type to use for comparison (String, Integer ,Double, Date or Currency).
ControlToCompare	The ID of the control that the value of the control specified in the ControlToValidate property should be compared to

A compare validator that checks for a value greater than zero

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>  
    <asp:CompareValidator  
ID="CompareValidator1"  
runat="server"  
ControlToValidate="TextBox1"  
Type="Integer"  
Operator="GreaterThan"  
ValueToCompare="0"  
ErrorMessage="Value should be greater than zero">  
</asp:CompareValidator>
```

Compare validator that checks for an integer value

```
<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>  
  <asp:CompareValidator  
ID="CompareValidator2"  
runat="server"  
ControlToValidate="TextBox2"  
Type="Integer"  
Operator="DataTypeCheck"  
ErrorMessage="Value should be an integer">  
</asp:CompareValidator>
```

A compare validator that compares the values of two text boxes

```
<asp:TextBox ID="txtstartdate" runat="server"></asp:TextBox>
    <asp:TextBox ID="txtenddate" runat="server"></asp:TextBox>
<asp:CompareValidator
ID="CompareValidator3"
runat="server"
ControlToValidate="txtenddate"
Operator="GreaterThan"
Type="Date"
ControlToCompare="txtstartdate"
ErrorMessage="End Date should be greater than start Date">
</asp:CompareValidator>
```

Range Validator

- The range validator checks that the user enters a value that falls within the range specified by the MinimumValue and MaximumValue properties . These properties can be set when the range validator is created or when the page is loaded.

Property	Description
Minimum Value	The minimum value allowed for the control.
Maximum Value	The maximum value allowed for the control.
Type	The data type to use for the comparison(String,Integer ,Double,Date or Currency)

Range Validator that checks for a numeric range

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>  
<asp:RangeValidator ID="RangeValidator1" runat="server"  
ControlToValidate="TextBox1"  
Type="Integer"  
MinimumValue="1"  
MaximumValue="15"  
ErrorMessage="Days must be between 1 and 15">  
</asp:RangeValidator>
```

Regular Expression Validator

- The regular expression validator matches the input entered by the user with the pattern supplied by the ValidationExpression property. If the input doesn't match the pattern the validation fails.

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>  
<asp:RegularExpressionValidator  
ID="RegularExpressionValidator1"  
runat="server"  
ControlToValidate="TextBox1"  
ErrorMessage="Must be a five digit US zip code"  
ValidationExpression="\d{5}">  
</asp:RegularExpressionValidator>
```

Regular expression elements

Element	Description
Ordinary Character	Matches any character other than <code>.,\$,^,[,{,(, ,)*,+,,?</code> or <code>\</code>
<code>\</code>	Matches the character that follows
<code>\d</code>	Matches any decimal digit(0-9)
<code>\D</code>	Matches any character other than a decimal digit
<code>\w</code>	Matches any word character (a-z ,A-Z and 0-9)
<code>\W</code>	Matches any character other than a word character
<code>\s</code>	Matches any white space character(space,tab,newline ,etc)
<code>\S</code>	Matches any character other than s whitespace character
<code>[abcd]</code>	Matches any character included between the brackets
<code>[^abcd]</code>	Matches any character that is not included between the brackets
<code>[a-z]</code>	Matches any character in the indicated range

Regular expression elements

Element	Description
{n}	Matches exactly n occurrences of the preceding element or group
{n,}	Matches at least n occurrences of the preceding element or group
{n,m}	Matches at least n but no more than m occurrences of the preceding element or group
*	Matches zero or more occurrences of the preceding element
?	Matches zero or one occurrence of the preceding element
+	Matches one or more occurrences of the preceding element
	Matches any of the elements separated by the vertical bar
()	Groups the elements that appear between the parentheses

Validation Summary Control

- The validation summary control displays a summary of the error messages that were generated by the page's validators .The summary can be displayed on the web page or in a separate message box.
- The error messages displayed in the validation summary control come from the ErrorMessage property of the page's validators .
- If you want to display a different message in the validator ,set the Text property of the validator

Validation Summary Control

```
<asp:ValidationSummary  
ID="ValidationSummary1" runat="server"  
HeaderText="Please correct the following errors"  
DisplayMode="list"  
ShowSummary=true  
ShowMessageBox=True/>
```

Validation Summary Control

Property	Description
DisplayMode	Specifies how the error messages from the validation controls are to be displayed. The available values are BulletList, List, or SingleParagraph. The default is BulletList.
HeaderText	The text that's displayed before the list of error messages
ShowSummary	A boolean value that determines whether the validation summary should be displayed on the web page. The default is True
ShowMessageBox	A boolean value that determines whether the validation summary should be displayed in a message box. The default is false

Custom Validator

- If none of the other validators provide the data validation program requires , can use a custom validator
- Code own validation routine that's executed when the page is submitted to the server.
- Code the validation tests within an event handler for the ServerValidate event of the custom validator. This event is raised when ever validation is performed on the server. The form must be submitted before the validation can be done.
- Use the properties of the args argument that's passes to the ServerValidate event handler to test the input data (e.value) and indicate whether the data passed the validation test (e.IsValid) . If you set the IsValid property of the args argument to False,the error message specified for the custom validator is displayed .

Custom Validator

Property	Description
Value	The text string to be validated
IsValid	A boolean property that you set to True if the value passes the validation test or to False otherwise .

Custom Validator

- Aspx code for the textbox and a custom validator

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>  
    <asp:CustomValidator ID="CustomValidator1" runat="server"  
        ControlToValidate="TextBox1" ErrorMessage="Type a string that starts with  
014"></asp:CustomValidator>
```

Visual basic code for the custom validator

```
Protected Sub CustomValidator1_ServerValidate(ByVal Sender As Object, ByVal e As  
    ServerValidateEventArgs) Handles CustomValidator1.ServerValidate  
    'check if the first three digits are divisible by 7  
    If Val(e.Value) <> 0 And Val(e.Value.Substring(0, 3)) Mod 7 = 0 Then  
        e.IsValid = True  
    Else  
        e.IsValid = False  
    End If  
End Sub
```