

## **ASP.NET Web Server Controls (Validation Controls)**

### ➤ **RequiredFieldValidator Control:**

#### ▪ **Definition and Usage**

The RequiredFieldValidator control ensures that the user does not skip an entry. The control fails validation if the value it contains does not change from its initial value when validation is performed. If all the fields in the page are valid, the page is valid.

#### • **Syntax:**

```
<asp:RequiredFieldValidator id="accessID"
  ControlToValidate = "ID of input control to validate"
  InitialValue = ""
  ErrorMessage = "message to display in ValidationSummary control"
  Text = "message to display in control"
  ForeColor = "color value"
  BackColor = "color value"
  runat="server" />
```

#### • **Properties:**

Property	Description
BackColor	The background color of the RequiredFieldValidator control
ControlToValidate	The id of the control to validate
Display	The display behavior for the validation control. Legal values are: <ul style="list-style-type: none"><li>• None (the control is not displayed. Used to show the error message only in the ValidationSummary control)</li><li>• Static (the control displays an error message if validation fails. Space is reserved on the page for the message even if the input passes validation.</li><li>• Dynamic (the control displays an error message if validation fails. Space is not reserved on the page for the message if the input passes validation)</li></ul>
EnableClientScript	A Boolean value that specifies whether client-side validation is enabled or not
Enabled	A Boolean value that specifies whether the validation control is enabled or not

ErrorMessage	The text to display in the ValidationSummary control when validation fails. <b>Note:</b> This text will also be displayed in the validation control if the Text property is not set
ForeColor	The foreground color of the control
Id	A unique id for the control
InitialValue	Specifies the starting value of the input control. Default value is ""
IsValid	A Boolean value that indicates whether the control specified by ControlToValidate is determined to be valid
Runat	Specifies that the control is a server control. Must be set to "server"
Text	The message to display when validation fails

- **Examples**

### **ASP.NET Source Code:**

```

<html>
<body>

    <h3><font face="Verdana">RequiredFieldValidator Example</font></h3>
    <form runat=server>
        Name: <asp:TextBox id=Text1 runat="server"/>
        <asp:RequiredFieldValidator id="RequiredFieldValidator1"
            ControlToValidate="Text1" Font-Names="Arial" Font-Size="11"
            runat="server">
            Required field!
        </asp:RequiredFieldValidator>
        <p>
        <asp:Button id="Button1" runat="server" Text="Validate" />

    </form>

</body>
</html>

```

### **Output:**

## **RequiredFieldValidator Example**

Name:  Required field!

---

## ➤ RangeValidator Control:

### ▪ Definition and Usage

The **RangeValidator** control tests whether an input value falls within a given range. **RangeValidator** uses three key properties to perform its validation: **ControlToValidate** contains the value to validate, **MinimumValue** defines the minimum value of the valid range, and **MaximumValue** defines the maximum value of the valid range. These constants are stored as string values, but are converted to the data type defined by **Type** when the comparison is performed.

### • Syntax:

```
<asp:RangeValidator id="accessID"
  ControlToValidate = "ID of input control to validate"
  MinimumValue = "lower bound value"
  MaximumValue = "upper bound value"
  Type = "datatype to convert to before comparing"
  ErrorMessage = "message to display in ValidationSummary control"
  Text = "message to display in control"
  ForeColor = "color value"
  BackColor = "color value"
  runat="server" />
```

### • Properties:

Property	Description
BackColor	The background color of the RangeValidator control
ControlToValidate	The id of the control to validate
Display	The display behavior for the validation control. Legal values are: <ul style="list-style-type: none"><li>• None (the control is not displayed. Used to show the error message only in the ValidationSummary control)</li><li>• Static (the control displays an error message if validation fails. Space is reserved on the page for the message even if the input passes validation.</li><li>• Dynamic (the control displays an error message if validation fails. Space is not reserved on the page for the message if the input passes validation)</li></ul>
EnableClientScript	A Boolean value that specifies whether client-side validation

	is enabled or not
Enabled	A Boolean value that specifies whether the validation control is enabled or not
ErrorMessage	The text to display in the ValidationSummary control when validation fails. <b>Note:</b> This text will also be displayed in the validation control if the Text property is not set
ForeColor	The foreground color of the control
id	A unique id for the control
IsValid	A Boolean value that indicates whether the control specified by ControlToValidate is determined to be valid
MaximumValue	Specifies the maximum value of the input control
MinimumValue	Specifies the minimum value of the input control
runat	Specifies that the control is a server control. Must be set to "server"
Type	Specifies the data type of the value to check. The types are: <ul style="list-style-type: none"> <li>• Currency</li> <li>• Date</li> <li>• Double</li> <li>• Integer</li> <li>• String</li> </ul>
Text	The message to display when validation fails

## • Examples

### ASP.NET Source Code:

```

<html>
<body>
<form runat="server">
Enter a date between 2005-01-01 and 2005-12-31:
<br />
<asp:TextBox id="tbox1" runat="server" />
<br /><br />
<asp:Button Text="Submit" runat="server" />
<br /><br />
<asp:RangeValidator
ControlToValidate="tbox1"
MinimumValue="2005-01-01"
MaximumValue="2005-12-31"
Type="Date"
EnableClientScript="false"
Text="The date must be between 2005-01-01 and 2005-12-31!"
runat="server" />
</form>

```

</body>  
</html>

### **Output:**

Enter a date between 2005-01-01 and 2005-12-31:

2007-01-25

Submit

The date must be between 2005-01-01 and 2005-12-31!

## ➤ **RegularExpressionValidator Control:**

### ▪ **Definition and Usage**

The **RegularExpressionValidator** control confirms that the entry matches a pattern defined by a regular expression. This type of validation allows you to check for predictable sequences of characters, such as those in social security numbers, e-mail addresses, telephone, numbers, postal, codes, and so on.

**RegularExpressionValidator** uses two key properties to perform its validation: **ControlToValidate** contains the value to validate, and **ValidationExpression** contains the regular expression to match.

### • **Syntax:**

```
<asp:RegularExpressionValidator id="accessID"  
  ControlToValidate = "ID of input control to validate"  
  ValidationExpression = "the regular expression pattern to test against"  
  ErrorMessage = "message to display in ValidationSummary control"  
  Text = "message to display in control"  
  ForeColor = "color value"  
  BackColor = "color value"  
  runat="server" />
```

### • **Properties:**

Property	Description
BackColor	The background color of the RegularExpressionValidator control

ControlToValidate	The id of the control to validate
Display	<p>The display behavior for the validation control. Legal values are:</p> <ul style="list-style-type: none"> <li>• None (the control is not displayed. Used to show the error message only in the ValidationSummary control)</li> <li>• Static (the control displays an error message if validation fails. Space is reserved on the page for the message even if the input passes validation.</li> <li>• Dynamic (the control displays an error message if validation fails. Space is not reserved on the page for the message if the input passes validation)</li> </ul>
EnableClientScript	A Boolean value that specifies whether client-side validation is enabled or not
Enabled	A Boolean value that specifies whether the validation control is enabled or not
ErrorMessage	The text to display in the ValidationSummary control when validation fails. <b>Note:</b> This text will also be displayed in the validation control if the Text property is not set
ForeColor	The foreground color of the control
id	A unique id for the control
IsValid	A Boolean value that indicates whether the control specified by ControlToValidate is determined to be valid
runat	Specifies that the control is a server control. Must be set to "server"
BackColor	The background color of the RegularExpressionValidator control
Text	The message to display when validation fails
ValidationExpression	Specifies the expression used to validate input control. The expression validation syntax is different on the client than on the server. JScript is used on the client. On the server, the language you have specified is used

## • Examples

### ASP.NET Source Code:

```

<html>
<head>
    <script language="C#" runat=server>

        void ValidateBtn_Click(Object Src, EventArgs E) {

```

```

        if (Page.IsValid) {
            lblOutput.Text = "Page is Valid!";
        }
        else {
            lblOutput.Text = "Page is Invalid! :-( ";
        }
    }
}

</script>

</head>
<body>

<h3><font face="Verdana">Simple RegularExpressionValidator
Sample</font></h3>

<form runat="server">

    <table bgcolor="#eeeeee" cellpadding=10>

        <tr valign="top">
            <td colspan=3>
                <asp:Label ID="lblOutput" Text="Enter a 5 digit zip code" Font-
                Names="Verdana" Font-Size="10pt" runat="server"/>
            </td>
        </tr>

        <tr>
            <td colspan=3>
                <font face=Verdana size=2><b>Personal Information</b></font>
            </td>
        </tr>

        <tr>
            <td align=right>
                <font face=Verdana size=2>Zip Code:</font>
            </td>
            <td>
                <ASP:TextBox id=TextBox1 runat=server />
            </td>
            <td>
                <asp:RegularExpressionValidator
                id="RegularExpressionValidator1" runat="server"
                ControlToValidate="TextBox1"
                ValidationExpression="\d{5}"
                Display="Static"
                Font-Names="verdana"
                Font-Size="10pt">
                    Zip code must be 5 numeric digits
                </asp:RegularExpressionValidator>
            </td>
        </tr>

        <tr>
            <td></td>
            <td>

```

```

        <ASP:Button text="Validate" OnClick="ValidateBtn_Click"
        runat=server />
    </td>
    <td></td>
</tr>

</table>

</form>

</body>
</html>

```

### Output:

## Simple RegularExpressionValidator Sample

Enter a 5 digit zip code

### Personal Information

Zip Code:

Zip code must be 5 numeric digits

## ➤ CompareValidator Control:

### ▪ Definition and Usage

The **CompareValidator** control compares the value of one control to another, or to an explicit value in the control's **ValueToCompare** property.

**Note:** For the purpose of validation, a specific property on the control is designated as its "value".

**CompareValidator** uses three key properties to perform its validation. **ControlToValidate** and **ControlToCompare** contain the values to compare. **Operator** defines the type of comparison to perform, for example, Equal or Not Equal.



- **Syntax:**

```
<asp:CompareValidator id="accessID"
  ControlToValidate = "ID of input control to validate"
  ValueToCompare = "constant value or the value of another control"
  ControlToCompare = "the name of the control to compare"
  Type = "datatype to convert to before comparing"
  Operator = "the type of comparison to perform
    ( less than, greater than, equal to, etc. )"
  ErrorMessage = "message to display in ValidationSummary control"
  Text = "message to display in control"
  ForeColor = "color value"
  BackColor = "color value"
  runat="server" />
```

- **Properties:**

Property	Description
BackColor	The background color of the CompareValidator control
ControlToCompare	The name of the control to compare with
ControlToValidate	The id of the control to validate
Display	<p>The display behavior for the validation control. Legal values are:</p> <ul style="list-style-type: none"> <li>• None (the control is not displayed. Used to show the error message only in the ValidationSummary control)</li> <li>• Static (the control displays an error message if validation fails. Space is reserved on the page for the message even if the input passes validation.</li> <li>• Dynamic (the control displays an error message if validation fails. Space is not reserved on the page for the message if the input passes validation)</li> </ul>
EnableClientScript	A Boolean value that specifies whether client-side validation is enabled or not
Enabled	A Boolean value that specifies whether the validation control is enabled or not
ErrorMessage	The text to display in the ValidationSummary control when validation fails. <b>Note:</b> This text will also be displayed in the validation control if the Text property is not set
ForeColor	The foreground color of the control
id	A unique id for the control
IsValid	A Boolean value that indicates whether the control specified by ControlToValidate is determined to be valid

Operator	<p>The type of comparison to perform. The operators are:</p> <ul style="list-style-type: none"> <li>• Equal</li> <li>• GreaterThan</li> <li>• GreaterThanEqual</li> <li>• LessThan</li> <li>• LessThanEqual</li> <li>• NotEqual</li> <li>• DataTypeCheck</li> </ul>
runat	Specifies that the control is a server control. Must be set to "server"
Text	The message to display when validation fails
Type	<p>Specifies the data type of the values to compare. The types are:</p> <ul style="list-style-type: none"> <li>• Currency</li> <li>• Date</li> <li>• Double</li> <li>• Integer</li> <li>• String</li> </ul>
ValueToCompare	A specified value to compare with

## • **Examples**

### **ASP.NET Source Code:**

```

<script runat="server">
void check_operator(Object sender, EventArgs e)

{
    compval.Operator=CType(list.SelectedIndex,ValidationCompareOperator)
    compval.Validate()
}
</script>

<html>
<body>

<form runat="server">
<table border="0" bgcolor="#b0c4de">
    <tr valign="top">
        <td colspan="4"><h4>Compare two values</h4></td>
    </tr>
    <tr valign="top">
        <td><asp:TextBox id="txt1" runat="server" /></td>
        <td>

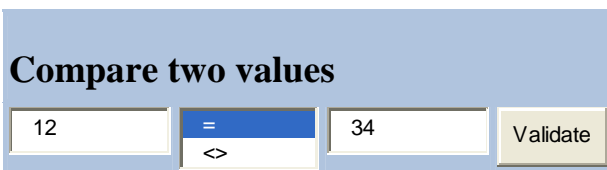
```

```

        <asp:ListBox id="list" rows="2" OnSelectedIndexChanged="check_operator"
runat="server">
            <asp:ListItem value="Equal" selected>=</asp:ListItem>
            <asp:ListItem value="NotEqual"><></asp:ListItem>
        </asp:ListBox>
    </td>
    <td><asp:TextBox id="txt2" runat="server" /></td>
    <td><asp:Button Text="Validate" runat="server" /></td>
</tr>
</table>
<br />
<asp:CompareValidator
id="compval"
Display="dynamic"
ControlToValidate="txt1"
ControlToCompare="txt2"
ForeColor="red"
BackColor="yellow"
Type="String"
EnableClientScript="false"
Text="Validation Failed!"
runat="server" />
</form>
</body>
</html>

```

### **Output:**



Validation Failed!

## ➤ **CustomValidator Control:**

### ▪ **Definition and Usage**

The **CustomValidator** control calls a user-defined function to perform validations that the standard validators can't handle. The custom function can execute on the server or in client-side script, such as JScript or VBScript. For client-side custom validation, the name of the custom function must be identified in the **ClientValidationFunction** property. The custom function must have the form function myvalidator(source, arguments).

Note that **source** is the client-side **CustomValidator** object, and **arguments** is an object with two properties, **Value** and **IsValid**. The **Value** property is the value to be validated and the **IsValid** property is a Boolean used to set the return result of the validation.

- **Syntax:**

```
<asp:CustomValidator id="accessID"  
  ControlToValidate = "ID of input control to validate"  
  onServerValidate = "ServerValidateID"  
  ErrorMessage = "message to display in ValidationSummary control"  
  Text = "message to display in control"  
  ForeColor = "color value"  
  BackColor = "color value"  
  runat="server" />
```

- **Properties:**

Property	Description
BackColor	The background color of the CustomValidator control
ClientValidationFunction	<p>Specifies the name of the client-side validation script function to be executed. <b>Note:</b> The script must be in a language that the browser supports, such as VBScript or JScript</p> <p>With VBScript, the function must be in the form:</p> <p>Sub FunctionName (source, arguments)</p> <p>With JScript, the function must be in the form:</p> <p>Function FunctionName (source, arguments)</p>
ControlToValidate	The id of the control to validate
Display	<p>The display behavior for the validation control. Legal values are:</p> <ul style="list-style-type: none"><li>• None (the control is not displayed. Used to show the error message only in the ValidationSummary control)</li><li>• Static (the control displays an error message if validation fails. Space is reserved on the page for the message even if the input passes validation.</li><li>• Dynamic (the control displays an error message if validation fails. Space is not reserved on the page for the message if the input passes validation)</li></ul>
EnableClientScript	A Boolean value that specifies whether client-side validation is enabled or not

Enabled	A Boolean value that specifies whether the validation control is enabled or not
ErrorMessage	The text to display in the ValidationSummary control when validation fails. <b>Note:</b> This text will also be displayed in the validation control if the Text property is not set
ForeColor	The foreground color of the control
id	A unique id for the control
IsValid	A Boolean value that indicates whether the control specified by ControlToValidate is determined to be valid
OnServerValidate	Specifies the name of the server-side validation script function to be executed
runat	Specifies that the control is a server control. Must be set to "server"
Text	The message to display when validation fails

- **Examples**

**ASP.NET Source Code:**

```

<script runat="server">
void user(object source, ServerValidateEventArgs args)

{
    if( len(args.Value)<8 or len(args.Value)>16 )
        args.IsValid=false
    else
        args.IsValid=true
}
</script>

<html>
<body>

<form runat="server">
<asp:Label runat="server" Text="Enter a username: " />
<asp:TextBox id="txt1" runat="server" />
<asp:Button Text="Submit" runat="server"/>
<br />
<asp:Label id="mess" runat="server"/>
<br />
<asp:CustomValidator
ControlToValidate="txt1"
OnServerValidate="user"
Text="A username must be between 8 and 16 characters!"
runat="server"/>
</form>

</body>
</html>

```

## Output:

Enter a username:

A username must be between 8 and 16 characters!

## ➤ ValidationSummary Control:

### ▪ Definition and Usage

When the user's input is processed (for example, when the form is submitted), the Web Forms framework passes the user's entry to the associated validation control or controls. The validation controls test the user's input and set a property to indicate whether the entry passed the validation test. After all validation controls have been processed, the **IsValid** property on the page is set; if any of the controls shows that a validation check failed, the entire page is set to invalid.

A **ValidationSummary** control is displayed when the **IsValid** property of the page is false. It "polls" each of the validation controls on the page and aggregates the text messages exposed by each. The following sample illustrates displaying errors with a **ValidationSummary**.

### • Syntax:

```
<asp:ValidationSummary id="accessID"
  DisplayMode = "BulletList | List | SingleParagraph"
  EnableClientScript=true | false
  ShowSummary=true | false
  ShowMessageBox=true | false
  HeaderText = "text to display as summary title"
  runat="server" />
```

### • Properties:

Property	Description
DisplayMode	How to display the summary. Legal values are:

	<ul style="list-style-type: none"> <li>• BulletList</li> <li>• List</li> <li>• SingleParagraph</li> </ul>
EnableClientScript	A Boolean value that specifies whether client-side validation is enabled or not
Enabled	A Boolean value that specifies whether the validation control is enabled or not
ForeColor	The fore color of the control
HeaderText	A header in the ValidationSummary control
id	A unique id for the control
runat	Specifies that the control is a server control. Must be set to "server"
ShowMessageBox	A Boolean value that specifies whether the summary should be displayed in a message box or not
ShowSummary	A Boolean value that specifies whether the ValidationSummary control should be displayed or hidden

## • **Examples**

### **ASP.NET Source Code:**

```

<html>
<body>
<form runat="server">
<table>
<tr>
<td>
<table bgcolor="#b0c4de" cellspacing="10">
<tr>
<td align="right">Name:</td>
<td><asp:TextBox id="txt_name" runat="server"/></td>
<td>
<asp:RequiredFieldValidator
ControlToValidate="txt_name"
ErrorMessage="Name"
Text="*"
runat="server"/>
</td>
</tr>
<tr>
<td align="right">Card Type:</td>
<td>
<asp:RadioButtonList id="rlist_type"
RepeatLayout="Flow"
runat="server">
<asp:ListItem>Diners</asp:ListItem>

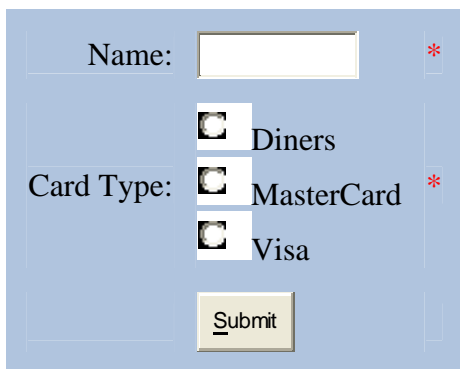
```

```

        <asp:ListItem>MasterCard</asp:ListItem>
        <asp:ListItem>Visa</asp:ListItem>
    </asp:RadioButtonList>
</td>
<td>
    <asp:RequiredFieldValidator
        ControlToValidate="rlist_type"
        ErrorMessage="Card Type"
        InitialValue=""
        Text="*"
        runat="server"/>
</td>
</tr>
<tr>
    <td></td>
    <td><asp:Button id="b1" Text="Submit" runat="server"/></td>
    <td></td>
</tr>
</table>
</td>
</tr>
</table>
<br />
<asp:ValidationSummary
    HeaderText="You must enter a value in the following fields:"
    DisplayMode="BulletList"
    EnableClientScript="true"
    runat="server"/>
</form>
</body>
</html>

```

### **Output:**



The screenshot shows a web form with a light blue background. It contains a text input field for 'Name' and a radio button list for 'Card Type' with options 'Diners', 'MasterCard', and 'Visa'. A 'Submit' button is at the bottom. Red asterisks (\*) are placed to the right of the 'Name' and 'MasterCard' labels, indicating validation errors. The 'Name' field is empty, and 'MasterCard' is selected.

You must enter a value in the following fields:

- Name
- Card Type