

RangeValidator.MinimumValue Property

.NET Framework (current version)

Gets or sets the minimum value of the validation range.

Namespace: [System.Web.UI.WebControls](#)

Assembly: System.Web (in System.Web.dll)

Syntax

C#

```
[ThemeableAttribute(false)]  
public string MinimumValue { get; set; }
```

Property Value

Type: [System.String](#)

The minimum value of the validation range. The default value is [String.Empty](#).

Remarks

Use the `MinimumValue` property to specify the minimum value of the validation range. If the value specified by this property fails to convert to the data type specified by the [BaseCompareValidator.Type](#) property, an exception is thrown.

Note

If you specify **ValidationDataType.Date** for the [BaseCompareValidator.Type](#) property without programmatically setting the culture for the application, you should use a culture-neutral format, such as YYYY/MM/DD, for the [MaximumValue](#) and `MinimumValue` properties. Otherwise, the date may not be interpreted correctly.

This property cannot be set by themes or style sheet themes. For more information, see [ThemeableAttribute](#) and [ASP.NET Themes and Skins](#).

Examples

The following example demonstrates how to use the `MinimumValue` property to specify the minimum value of the validation range.



Security Note

This example has a text box that accepts user input, which is a potential security threat. By default, ASP.NET Web pages validate that user input does not include script or HTML elements. For more information, see [Script Exploits Overview](#).

C#

```
<%@ Page Language="C#" AutoEventWireup="True" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >

<head>
    <title>RangeValidator Example</title>
    <script runat="server">

        void ButtonClick(Object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                Label1.Text="Page is valid.";
            }
            else
            {
                Label1.Text="Page is not valid!!";
            }
        }
    }

</script>

</head>

<body>

    <form id="form1" runat="server">

        <h3>RangeValidator Example</h3>

        Enter a number from 1 to 10:

        <br />

        <asp:TextBox id="TextBox1"
            runat="server"/>

        <br />

        <asp:RangeValidator id="Range1"
            ControlToValidate="TextBox1"
            MinimumValue="1"
            MaximumValue="10"
            Type="Integer"
            EnableClientScript="false"
            Text="The value must be from 1 to 10!"
```

```
        runat="server"/>

<br /><br />

<asp:Label id="Label1"
    runat="server"/>

<br /><br />

<asp:Button id="Button1"
    Text="Submit"
    OnClick="ButtonClick"
    runat="server"/>

</form>

</body>
</html>
```

Version Information

.NET Framework

Available since 1.1

See Also

[MaximumValue](#)

[Type](#)

[IsValid](#)

[String.Empty](#)

[RangeValidator Class](#)

[System.Web.UI.WebControls Namespace](#)

[Return to top](#)

© 2017 Microsoft