This page is specific to

Microsoft Visual Studio 2010/.NET Framework 4

.NET Framework Class Library

**ClientScriptManager.RegisterForEventValidation Method (String, String)**

Registers an event reference for validation with a unique control ID and event arguments representing the client control generating the event.

**Namespace:** [System.Web.UI](http://msdn.microsoft.com/en-us/library/system.web.ui.aspx)  
**Assembly:** System.Web (in System.Web.dll)

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifSyntax

Visual Basic

Public Sub RegisterForEventValidation ( \_

uniqueId As String, \_

argument As String \_

)

C#

public void RegisterForEventValidation(

string uniqueId,

string argument

)

Visual C++

public:

void RegisterForEventValidation(

String^ uniqueId,

String^ argument

)

F#

member RegisterForEventValidation :

uniqueId:string \*

argument:string -> unit

**Parameters**

*uniqueId*

Type: [System.String](http://msdn.microsoft.com/en-us/library/system.string.aspx)  
A unique ID representing the client control generating the event.

*argument*

Type: [System.String](http://msdn.microsoft.com/en-us/library/system.string.aspx)  
Event arguments passed with the client event.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifExceptions

|  |  |
| --- | --- |
| **Exception** | **Condition** |
| [InvalidOperationException](http://msdn.microsoft.com/en-us/library/system.invalidoperationexception.aspx) | The method is called prior to the [Render](http://msdn.microsoft.com/en-us/library/20b8wk8d.aspx) method. |

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifExamples

The following code example demonstrates using the [RegisterForEventValidation](http://msdn.microsoft.com/en-us/library/system.web.ui.clientscriptmanager.registerforeventvalidation.aspx) method and the [ValidateEvent](http://msdn.microsoft.com/en-us/library/system.web.ui.clientscriptmanager.validateevent.aspx) method to register a callback for validation and to validate that the callback originated from the page. To improve on the validation shown in the example, you could modify the validation *argument* parameter to contain information specific to the user such as an identity or role

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl36_ctl00_ctl02_code');" \o "Copy Code)

<%@ Page Language="VB" %>

<%@ Implements Interface="System.Web.UI.ICallbackEventHandler" %>

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

<script runat="server">

Dim \_cbMessage As String = ""

' Define method that processes the callbacks on server.

Public Sub RaiseCallbackEvent(ByVal eventArgument As String) \_

Implements System.Web.UI.ICallbackEventHandler.RaiseCallbackEvent

Try

Page.ClientScript.ValidateEvent(button1.UniqueID, Me.ToString())

\_cbMessage = "Correct event raised callback."

Catch ex As Exception

\_cbMessage = "Incorrect event raised callback."

End Try

End Sub

' Define method that returns callback result.

Public Function GetCallbackResult() \_

As String Implements \_

System.Web.UI.ICallbackEventHandler.GetCallbackResult

Return \_cbMessage

End Function

Protected Sub Page\_Load(ByVal sender As Object, ByVal e As System.EventArgs)

If (Not IsPostBack) Then

Dim cs As ClientScriptManager = Page.ClientScript

Dim cbReference As String = cs.GetCallbackEventReference("'" & \_

Page.UniqueID & "'", "arg", "ReceiveServerData", "", \_

"ProcessCallBackError", False)

Dim callbackScript As String = "function CallTheServer(arg, context) {" & \_

cbReference & "; }"

cs.RegisterClientScriptBlock(Me.GetType(), "CallTheServer", \_

callbackScript, True)

End If

End Sub

Protected Overrides Sub Render(ByVal writer As System.Web.UI.HtmlTextWriter)

Page.ClientScript.RegisterForEventValidation(button1.UniqueID, Me.ToString())

MyBase.Render(writer)

End Sub

</script>

<script type="text/javascript">

var value1 = new Date();

function ReceiveServerData(arg, context)

{

Message.innerText = arg;

Label1.innerText = "Callback completed at " + value1;

value1 = new Date();

}

function ProcessCallBackError(arg, context)

{

Message.innerText = 'An error has occurred.';

}

</script>

<html >

<head runat="server">

<title>CallBack Event Validation Example</title>

</head>

<body>

<form id="Form1" runat="server">

<div>

Callback result: <span id="Message"></span>

<br /> <br />

<input type="button"

id="button1"

runat="server"

value="ClientCallBack"

onclick="CallTheServer(value1, null )"/>

<br /> <br />

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

</div>

</form>

</body>

</html>

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl36_ctl00_ctl03_code');" \o "Copy Code)

<%@ Page Language="C#" %>

<%@ Implements Interface="System.Web.UI.ICallbackEventHandler" %>

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

<script runat="server" >

string \_cbMessage = "";

// Define method that processes the callbacks on server.

public void RaiseCallbackEvent(String eventArgument)

{

try

{

Page.ClientScript.ValidateEvent(button1.UniqueID, this.ToString());

\_cbMessage = "Correct event raised callback.";

}

catch (Exception ex)

{

\_cbMessage = "Incorrect event raised callback.";

}

}

// Define method that returns callback result.

public string GetCallbackResult()

{

return \_cbMessage;

}

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

ClientScriptManager cs = Page.ClientScript;

String cbReference = cs.GetCallbackEventReference("'" +

Page.UniqueID + "'", "arg", "ReceiveServerData", "",

"ProcessCallBackError", false);

String callbackScript = "function CallTheServer(arg, context) {" +

cbReference + "; }";

cs.RegisterClientScriptBlock(this.GetType(), "CallTheServer",

callbackScript, true);

}

}

protected override void Render(HtmlTextWriter writer)

{

Page.ClientScript.RegisterForEventValidation(button1.UniqueID, this.ToString());

base.Render(writer);

}

</script>

<script type="text/javascript">

var value1 = new Date();

function ReceiveServerData(arg, context)

{

Message.innerText = arg;

Label1.innerText = "Callback completed at " + value1;

value1 = new Date();

}

function ProcessCallBackError(arg, context)

{

Message.innerText = 'An error has occurred.';

}

</script>

<html >

<head runat="server">

<title>CallBack Event Validation Example</title>

</head>

<body>

<form id="Form1" runat="server">

<div>

Callback result: <span id="Message"></span>

<br /> <br />

<input type="button"

id="button1"

runat="server"

value="ClientCallBack"

onclick="CallTheServer(value1, null )"/>

<br /> <br />

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

</div>

</form>

</body>

</html>

The following code example demonstrates using the [RegisterForEventValidation](http://msdn.microsoft.com/en-us/library/system.web.ui.clientscriptmanager.registerforeventvalidation.aspx) method to register a callback for validation.

Visual Basic

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl36_ctl00_ctl05_code');" \o "Copy Code)

<%@ Page EnableEventValidation="true" Language="VB" %>

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

<html >

<head id="Head1" runat="server">

<title>Register an option for event validation</title>

<script runat="server">

Protected Overrides Sub Render(ByVal writer As System.Web.UI.HtmlTextWriter)

ClientScript.RegisterForEventValidation("DropDownList1", "This is Option 1")

ClientScript.RegisterForEventValidation("DropDownList1", "This is Option 2")

ClientScript.RegisterForEventValidation("DropDownList1", "This is Option 3")

' Uncomment the line below when you want to specifically register the option for event validation.

' ClientScript.RegisterForEventValidation("DropDownList1", "Is this option registered for event validation?")

MyBase.Render(writer)

End Sub

</script>

<script type="text/javascript">

function Initialize()

{

var oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "This is Option 1";

oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "This is Option 2";

oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "This is Option 3";

oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "Is this option registered for event validation?";

}

</script>

</head>

<body onload="Initialize();">

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

<div>

<asp:DropDownList ID="DropDownList1" runat="server">

</asp:DropDownList>

<asp:Button ID="Button1" runat="server" Text="Postback to server for validation" />

</div>

</form>

</body>

</html>

C#

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl36_ctl00_ctl06_code');" \o "Copy Code)

<%@ Page EnableEventValidation="true" Language="C#" %>

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

<html >

<head id="Head1" runat="server">

<title>Register an option for event validation</title>

<script runat="server">

protected override void Render(HtmlTextWriter writer)

{

ClientScript.RegisterForEventValidation("DropDownList1", "This is Option 1");

ClientScript.RegisterForEventValidation("DropDownList1", "This is Option 2");

ClientScript.RegisterForEventValidation("DropDownList1", "This is Option 3");

// Uncomment the line below when you want to specifically register the option for event validation.

// ClientScript.RegisterForEventValidation("DropDownList1", "Is this option registered for event validation?");

base.Render(writer);

}

</script>

<script type="text/javascript">

function Initialize()

{

var oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "This is Option 1";

oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "This is Option 2";

oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "This is Option 3";

oOption = document.createElement("OPTION");

document.all("DropDownList1").options.add(oOption);

oOption.innerText = "Is this option registered for event validation?";

}

</script>

</head>

<body onload="Initialize();">

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

<div>

<asp:DropDownList ID="DropDownList1" runat="server"></asp:DropDownList>

<asp:Button ID="Button1" runat="server" Text="Postback to server for validation" />

</div>

</form>

</body>

</html>