



**S. S Jain Subodh P.G. (Autonomous) College**

**SUBJECT - .NET TECHNOLOGY**

**TITLE - ASP.NET AJAX**

**PowerPoint Presentation**

**by**

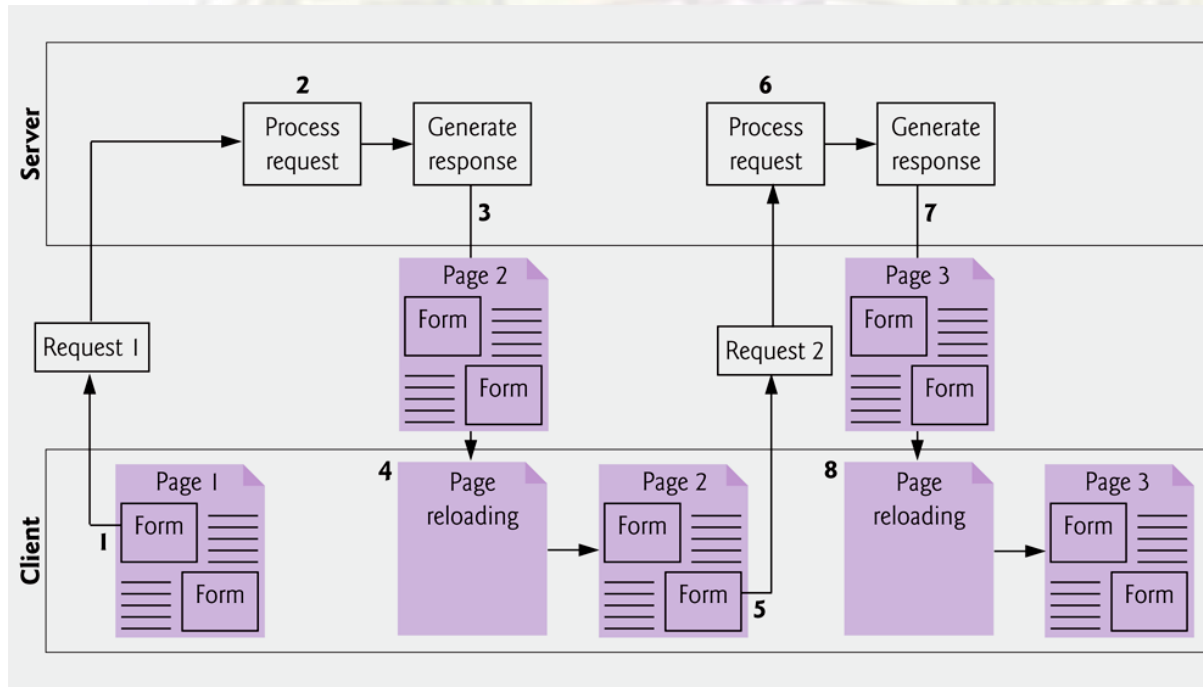
**Mr. Aleem Khan**

**[Department of Computer Science]**



## Traditional Web Applications

- Figure below presents the typical interactions between the client and the server in a traditional web application.



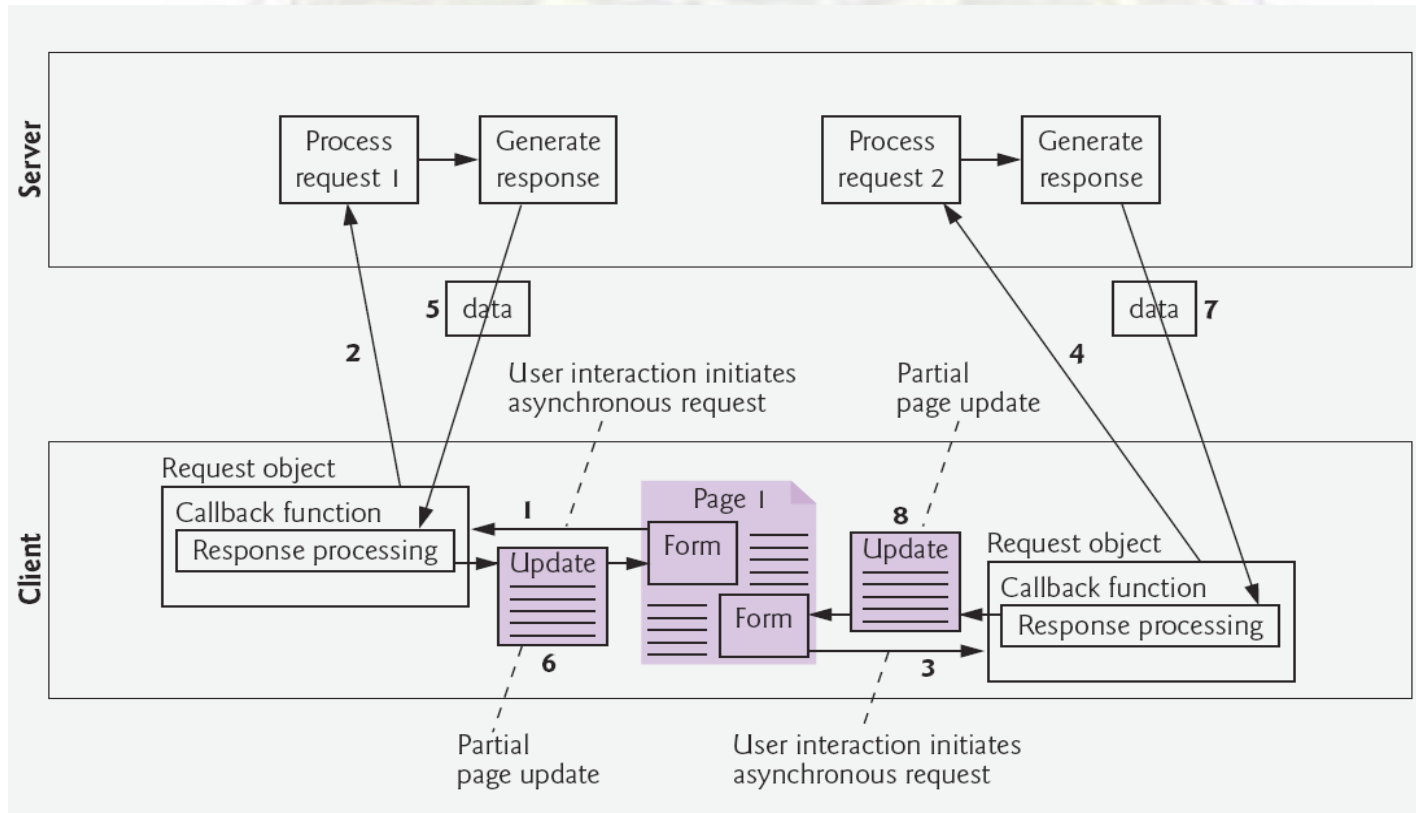


- The user first fills in the form's fields, then submits the form (*Step 1*).
- The browser generates a request to the server, which receives the request and processes it (*Step 2*).
- The server generates and sends a response containing the page that the browser renders (*Step 3*).
- The browser to loads the new page (*Step 4*) and temporarily makes the browser window blank.
- While such a **synchronous request** is being processed on the server, the user cannot interact with the web page.
- If the user interacts with and submits another form, the process begins again (*Steps 5–8*).



## Ajax Web Applications

- Ajax web applications add a layer between the client and the server to manage communication between the two.





- When the user interacts with the page, the client requests information from the server (*Step 1*).
  - The request is intercepted by the ASP.NET AJAX controls and sent to the server as an **asynchronous request** (*Step 2*)
  - The user can continue interacting with the application in the client browser while the server processes the request.
- Other user interactions could result in additional requests to the server (*Steps 3 and 4*)



- Once the server responds to the original request (Step 5), the ASP.NET AJAX control calls a client-side function to process the data returned by the server.
- This function—known as a callback function—uses partial-page updates (Step 6) to display the data without reloading the entire page.
- At the same time, the server may be responding to the second request (Step 7) and the client browser may be starting another partial-page update (Step 8).



- The callback function updates only a designated part of the page.
- Such partial-page updates help make web applications more responsive, making them feel more like desktop applications.
- The web application does not load a new page while the user interacts with it.



## Examining an ASP.NET AJAX Application

- The **AJAX Extensions package** that implements basic Ajax functionality comes preinstalled in Visual Studio.
- There is a tab of basic **AJAX Extensions** controls in the **Toolbox**.
- Microsoft also provides the **ASP.NET AJAX Control Toolkit**, which contains rich, Ajax-enabled GUI controls, which you can download.





- To add the controls to the **Toolbox** in Visual Web Developer:
  - Right click the Toolbox and choose Add Tab.
  - Type **AJAX Control Toolkit** as the name of the new tab.
  - Right click the tab and select Choose Items.
  - Click **Browse...** and select **AjaxControlToolkit.dll** from the folder.
  - Finally, click OK, and a list of available Ajax controls will appear under the **AJAX Control Toolkit** tab when you are in **Design mode**.



- To demonstrate ASP.NET AJAX capabilities we'll enhance the Validation application we did before.
- The only significant modifications to this application appear in its ASPX file.
- We'll use Ajax-enabled controls to add Ajax features to this application.



a) The user enters his or her first and last name and proceeds to the Contact tab.

Demonstrating Validation Controls - Windows Internet Explorer  
http://localhost:494  
Google  
Demonstrating Validatio...  
Please fill out the following form.  
All fields are required and must contain valid information  
Name Contact  
First Name: Mike  
Last Name: Brown  
Submit  
Internet | Protected Mode: On | 100%

b) The user enters an e-mail address in an incorrect format and presses *Tab* to move to the next input field. A callout appears informing the user to enter an e-mail address in a valid format.

Demonstrating Validation Controls - Windows Internet Explorer  
http://localhost:494  
Google  
Demonstrating Validatio...  
Please fill out the following form.  
All fields are required and must contain valid information  
Name Contact  
E-mail address: mbrown  
Phone number: (123) 456-7890  
Please enter a valid e-mail address.  
Submit  
Internet | Protected Mode: On | 100%

validation.aspx

(9 of 10)

**Fig. 1 | Validation application enhanced by ASP.NET AJAX. (Part 9 of 10.)**



## Outline

validation.aspx

(10 of 10)

c) After the user fills out the form properly and clicks the **Submit** button, the submitted data is displayed at the bottom of the page with a partial-page update.

Demonstrating Validation Controls - Windows Internet Explorer

http://localhost:494

Google

Demonstrating Validatio...

Please fill out the following form.  
*All fields are required and must contain valid information*

Name Contact

E-mail address:  e.g., user@domain.com

Phone number:  e.g., (555) 555-1234

Submit

Thank you for your submission.  
We received the following information:

First Name: Mike  
Last Name: Brown  
E-mail address: mbrown@test.com  
Phone number: (123) 456-7890

Internet | Protected Mode: On 100%

**Fig. 1** | Validation application enhanced by ASP.NET AJAX. (Part 10 of 10.)



- To demonstrate ASP.NET AJAX capabilities we'll enhance the `Validation` application we did before.
- The only significant modifications to this application appear in its ASPX file.
- We'll use Ajax-enabled controls to add Ajax features to this application.
- Figure below is a modified `Validation.aspx` file that enhances the application by using several ASP.NET AJAX controls.

## Outline

### `Validation.aspx`

(1 of 10)

```
1 <!-- Fig. 22.68: Validation.aspx -->
2 <!-- Validation application enhanced with ASP.NET AJAX. -->
3 <%@ Page Language="C#" AutoEventWireup="true"
4     CodeFile="Validation.aspx.cs" Inherits="Validation" %>
5 <%@ Register assembly="AjaxControlToolkit"
6     namespace="AjaxControlToolkit" tagprefix="ajax" %>
7
8 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
9     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
10
```

Associate the `AjaxControlToolkit` assembly with a tag prefix, allowing us to use AJAX Control Toolkit elements.

**Fig.1** | Validation application enhanced by ASP.NET AJAX. (Part 1 of 10.)



## Outline

```
11 <html xmlns="http://www.w3.org/1999/xhtml">
12 <head runat="server">
13   <title>Demonstrating Validation Controls</title>
14   <style type="text/css">
15     .style1
16     {
17       width: 100%;
18     }
19     .style2
20     {
21       text-align: right;
22       vertical-align: top;
23     }
24   </style>
25 </head>
26 <body>
27 <form id="form1" runat="server">
28 <ajax:ToolkitScriptManager ID="ToolkitScriptManager1" runat="server">
29 </ajax:ToolkitScriptManager>
```

validation.aspx

(2 of 10)

Load the  
ToolkitScriptManager  
on the page.

**Fig.1** | Validation application enhanced by ASP.NET AJAX. (Part 2 of 10.)



## Outline

```
30 <div>
31   <p>
32     Please fill out the following form.<br /><i>All fields are
33     required and must contain valid information.</i>
34   </p>
35   <ajax:TabContainer ID="TabContainer1" runat="server"
36     ActiveTabIndex="0" width="500px">
37     <ajax:TabPanel runat="server" HeaderText="Name" ID="TabPanel1">
38       <ContentTemplate>
39         <table class="style1">
40           <tr>
41             <td class="style2">First Name:</td>
42             <td>
43               <asp:TextBox ID="firstNameTextBox" runat="server">
44                 </asp:TextBox><br />
45               <asp:RequiredFieldValidator
46                 ID="firstNameExistsValidator" runat="server"
47                 ControlToValidate="firstNameTextBox" Display="None"
48                 ErrorMessage="Please enter your first name.">
49                 </asp:RequiredFieldValidator>
```

validation.aspx

(3 of 10)

**Fig.1** | Validation application enhanced by ASP.NET AJAX. (Part 3 of 10.)



```
50     <ajax:ValidatorCalloutExtender
51         ID="firstNameExistsCallout" runat="server"
52         Enabled="True"
53         TargetControlID="firstNameExistsValidator">
54     </ajax:ValidatorCalloutExtender>
55 </td>
56 </tr>
57 <tr>
58     <td class="style2">Last Name:</td>
59     <td>
60         <asp:TextBox ID="lastNameTextBox" runat="server">
61         </asp:TextBox><br />
62         <asp:RequiredFieldValidator
63             ID="lastNameExistsValidator" runat="server"
64             ControlToValidate="lastNameTextBox" Display="None"
65             ErrorMessage="Please enter your last name.">
66         </asp:RequiredFieldValidator>
67         <ajax:ValidatorCalloutExtender
68             ID="lastNameExistsCallout" runat="server"
69             Enabled="True"
70             TargetControlID="lastNameExistsValidator">
71         </ajax:ValidatorCalloutExtender>
```

## Outline

validation.aspx

(4 of 10)

**ValidatorCalloutExtender controls** display error messages in small yellow callouts next to the input fields.

**Fig.1** | Validation application enhanced by ASP.NET AJAX. (Part 4 of 10.)





## Outline

```
72         </td>
73     </tr>
74 </table>
75 </ContentTemplate>
76 </ajax:TabPanel>
77 <ajax:TabPanel ID="TabPanel2" runat="server" HeaderText="Contact">
78     <ContentTemplate>
79     <table class="style1">
80         <tr>
81             <td class="style2">E-mail address:</td>
82             <td>
83                 <asp:TextBox ID="emailTextBox" runat="server">
84                 </asp:TextBox>&nbsp;e.g., user@domain.com<br />
85                 <asp:RequiredFieldValidator
86                 ID="emailExistsValidator" runat="server"
87                 ErrorMessage="Please enter your e-mail address."
88                 ControlToValidate="emailTextBox" Display="None">
89                 </asp:RequiredFieldValidator>
```

validation.aspx

(5 of 10)

**Fig.1** | Validation application enhanced by ASP.NET AJAX. (Part 5 of 10.)



## Outline

validation.aspx

(6 of 10)

```
90 <ajax:ValidatorCalloutExtender
91     ID="emailExistsCallout" runat="server"
92     Enabled="True"
93     TargetControlID="emailExistsValidator">
94 </ajax:ValidatorCalloutExtender>
95 <asp:RegularExpressionValidator
96     ID="emailFormatValidator" runat="server"
97     ErrorMessage="Please enter a valid e-mail address."
98     ControlToValidate="emailTextBox" Display="None"
99     ValidationExpression=
100         "\w+([-+.']\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*">
101 </asp:RegularExpressionValidator>
102 <ajax:ValidatorCalloutExtender ID="emailFormatCallout"
103     runat="server" Enabled="True"
104     TargetControlID="emailFormatValidator">
105 </ajax:ValidatorCalloutExtender>
106 </td>
107 </tr>
108 <tr>
109     <td class="style2">Phone number:</td>
```

**Fig.1** | Validation application enhanced by ASP.NET AJAX. (Part 6 of 10.)



```
110 <td>
111 <asp:TextBox ID="phoneTextBox" runat="server">
112 </asp:TextBox>&nbsp;e.g., (555) 555-5555<br />
113 <asp:RequiredFieldValidator ID="phoneExistsValidator"
114 <runat="server" ControlToValidate="phoneTextBox"
115 <ErrorMessage="Please enter your phone number."
116 <Display="None">
117 </asp:RequiredFieldValidator>
118 <ajax:ValidatorCalloutExtender ID="phoneExistsCallout"
119 <runat="server" Enabled="True"
120 <TargetControlID="phoneExistsValidator">
121 </ajax:ValidatorCalloutExtender>
122 <asp:RegularExpressionValidator
123 <ID="phoneFormatValidator" runat="server"
124 <ControlToValidate="phoneTextBox"
125 <ErrorMessage="Please enter a valid phone number."
126 <Display="None" ValidationExpression=
127 <"((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}">
128 </asp:RegularExpressionValidator>
129 <ajax:ValidatorCalloutExtender ID="phoneFormatCallout"
130 <runat="server" Enabled="True"
131 <TargetControlID="phoneFormatValidator">
132 </ajax:ValidatorCalloutExtender>
133 </td>
```

## Outline

validation.aspx

(7 of 10)



```
134         </tr>
135     </table>
136     </ContentTemplate>
137 </ajax:TabPanel>
138 </ajax:TabContainer>
139 <p>
140     <asp:Button ID="submitButton" runat="server" Text="Submit" />
141 </p>
142 <asp:UpdatePanel ID="UpdatePanel1" runat="server">
143     <ContentTemplate>
144         <asp:Label ID="outputLabel" runat="server"
145             Text="Thank you for your submission." visible="False">
146         </asp:Label>
147     </ContentTemplate>
148     <Triggers>
149         <asp:AsyncPostBackTrigger ControlID="submitButton"
150             EventName="Click" />
151     </Triggers>
152 </asp:UpdatePanel>
153 </div>
154 </form>
155 </body>
156 </html>
```

## Outline

validation.aspx

(8 of 10)



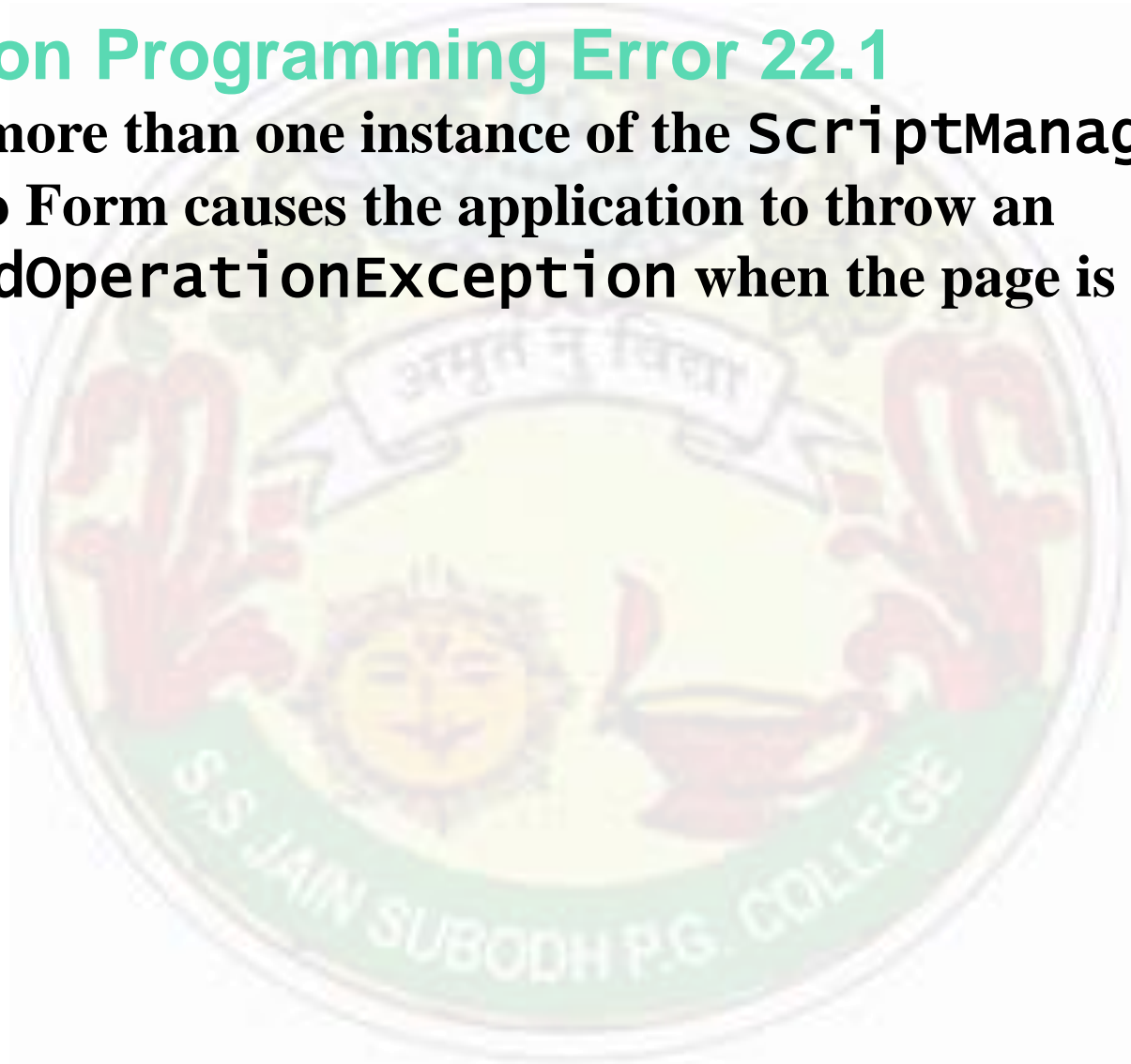
## ToolkitScriptManager Control

- The **ToolkitScriptManager** control manages the client-side scripts that enable asynchronous Ajax functionality.
- There can be only one ToolkitScriptManager per page.
- To incorporate controls from the AJAX Control Toolkit, use the ToolkitScriptManager, which derives from ScriptManager, instead.
- Drag the ToolkitScriptManager from the AJAX Control Toolkit tab in the Toolbox to the top of the page.
- A script manager must appear before any controls that use the scripts it manages.



## Common Programming Error 22.1

Putting more than one instance of the `ScriptManager` control on a Web Form causes the application to throw an `InvalidOperationException` when the page is initialized.





## *Grouping Information in Tabs Using the TabContainer Control*

- The **TabContainer control** enables you to group information into tabs that are displayed only if they are selected.
- To create multiple tabs, drag the TabContainer control from the **AJAX Control Toolkit** tab in the **Toolbox** to your form.
- To add a tab, open the **TabContainer Tasks** smart-tag menu and select **Add Tab Panel**. This adds a **TabPanel object** to the TabContainer.



- In **Design** view, you can navigate between tabs by holding *Ctrl* and clicking the tab header, and you can drag and drop elements into the tab.
- The content of a `TabPane1` must be defined inside its `ContentTemplate` element.





### *Partial-Page Updates Using the UpdatePanel Control*

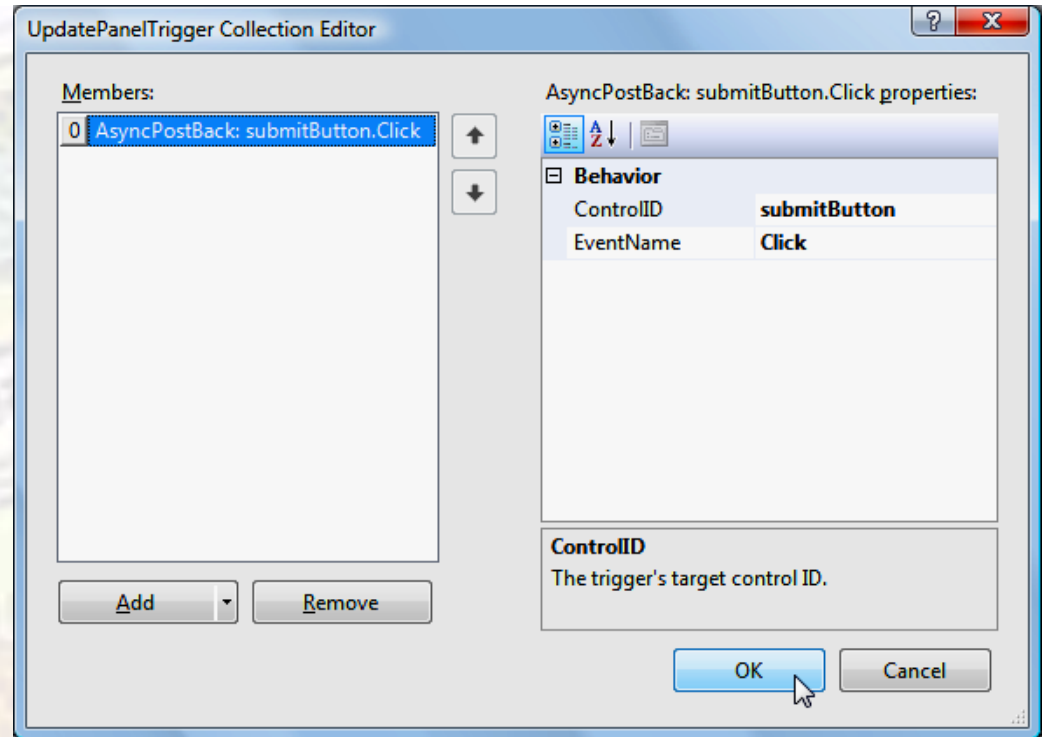
- The **UpdatePanel control** eliminates full-page refreshes by isolating a section of a page for a partial-page update.
- To implement a partial-page update, drag the UpdatePanel into your form, and place controls to be updated inside it.
- To specify when an UpdatePanel should update, you need to define an **UpdatePanel trigger**.



# S. S Jain Subodh P.G. (Autonomous) College

## ASP.NET AJAX

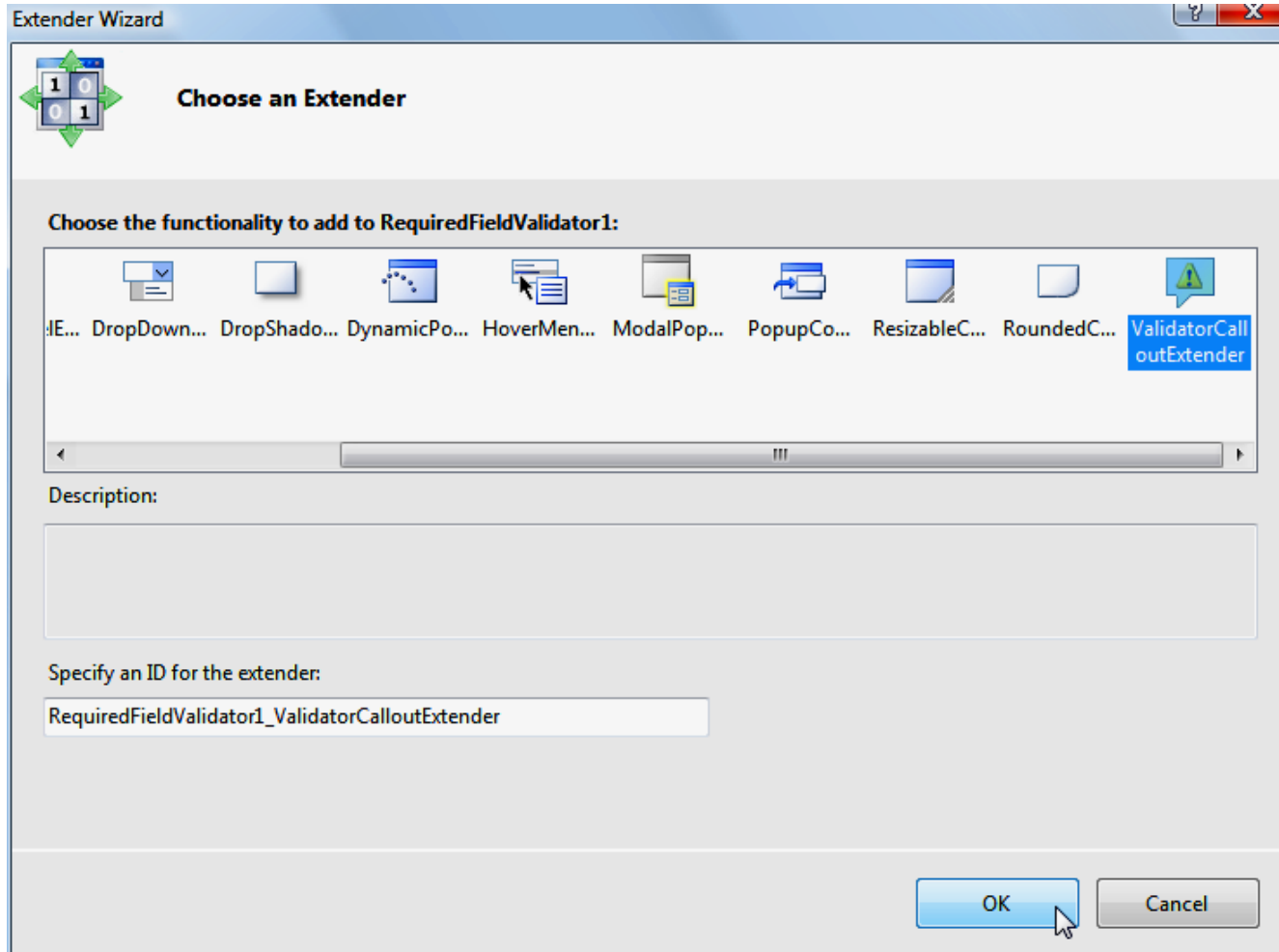
- Click the ellipsis button next to the control's Triggers property in the **Properties** window.
- In the **UpdatePanelTrigger Collection Editor** dialog that appears, click **Add** to add an **AsyncPostBackTrigger**.
- Set the ControlID property and the EventName property to the desired trigger.





## *Adding Ajax Functionality to ASP.NET Validation Controls Using Ajax Extenders*

- Several controls in the Ajax Control Toolkit are **extenders**— components that enhance regular ASP.NET controls.
- **ValidatorCalloutExtender controls** display error messages in small yellow callouts next to the input fields.
- To create a ValidatorCalloutExtender, you can drag and drop it, or you can select the **Add Extender** option in a validator's smart-tag menu.
- In the **Extender Wizard** dialog that displays, choose ValidatorCalloutExtender, specify its ID, and click **OK**.



**Fig.** Creating a control extender using the **Extender Wizard**.



- `ValidatorCalloutExtender`'s `TargetControlID` property indicates the validator control from which the extender should obtain the error message.
- This is automatically determined if you created the extender through the **Extender Wizard**.
- The `ValidatorCalloutExtenders` display error messages with a nicer look-and-feel.



**Thank You....**