

Validation Groups

In more complex pages, you might have several distinct groups of controls, possibly in separate panels. In these situations, you may want to perform validation separately. For example, you might create a form that includes a box with login controls and a box underneath it with the controls for registering a

new user. Each box includes its own submit button, and depending on which button is clicked, you want

to perform the validation just for that section of the page.

ASP.NET enables this scenario with a feature called validation groups. To create a validation group,

you need to put the input controls and the CausesValidation button controls into the same logical group. You do this by setting the ValidationGroup property of every control with the same descriptive string (such as "Login" or "NewUser"). Every button control that provides a CauseValidation property also includes the ValidationGroup property. All validators acquire the ValidationGroup by inheriting from the BaseValidator class.

For example, the following page defines two validation groups, named Group1 and Group2:

```
<form id="form1" runat="server">
  <div>
    <asp:Panel ID="Panel1" runat="server">
      <asp:TextBox ID="TextBox1" ValidationGroup="Group1"
runat="server" />
      <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
ErrorMessage="*Required" ValidationGroup="Group1" runat="server"
ControlToValidate="TextBox1" />
      <asp:Button ID="Button1" Text="Validate" Group1"
ValidationGroup="Group1" runat="server" />
    </asp:Panel>
    <br />
    <asp:Panel ID="Panel2" runat="server">
      <asp:TextBox ID="TextBox2" ValidationGroup="Group2"
runat="server" />
      <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
ErrorMessage="*Required" ValidationGroup="Group2"
ControlToValidate="TextBox2" runat="server" />
      <asp:Button ID="Button2" Text="Validate Group2"
ValidationGroup="Group2" runat="server" />
    </asp:Panel>
  </div>
</form>
```

```
</asp:Panel>  
</div>  
</form>
```

Figure 4-14 shows the page. If you click the first button, only the first text box is validated. If you click the second button, only the second text box is validated.

An interesting scenario is if you add a new button that doesn't specify any validation group. In this case, the button validates every control that isn't explicitly assigned to a named validation group. In this case, no controls fit the requirement, so the page is posted back successfully and deemed to be valid. If you want to make sure a control is always validated, regardless of the validation group of the button that's clicked, you'll need to create multiple validators for the control, one for each group (and one with no validation group). Alternatively, you might choose to manage complex scenarios such as these using server-side code.