Validation Controls for ListView CRUD

Validation Controls

Validation controls for a ListView are implemented in the same fashion as a form with individual fields. You will need a control for each validation you wish to preform against a specific control. The validation control must have a unique name. The control to validate must have a unique name (more on this below). You will possible still want the Display=”None”. Still include your personalized ErrorMessage. **One additional parameter needed will be ValidationGroup.** This parameter will allow you to group the validations per **template**. Your EditTemplate and your InsertTemplate will need their **own** set of validation controls. You will need one ValidationSummary for each validation group. Add the **ValidationGroup** parameter to the ValidationSummary control with a parameter value equal to the group name used for the validation controls. The validation controls **must** be coded **within** the template.

Template Changes

Validation controls need to be associated with a control. This is done via the ControlToValidate on the validation control. This parameter points to the uniquely named webform control with which it is associated. **Unfortunately**, the default id parameter names used in the InsertTemplate and EditTemplate for a specific column (field) are the **same.** If you plan to validate a control on one of these templates, you will need to ensure that the id parameter name is unique for the **entire webform**. One easy way to accomplish this is to add the suffix **I (for insert)** or **E (for edit)** to the end of the id parameter name.

HTML5 validation

In addition to Validation controls, you could use HTML TextMode. For a numeric field you could add the TextMode=”Number” parameter to the asp control. Check out the numerous TextMode possibilities. Remember however, HTML5 validation is only client side.

ListView Buttons

The command buttons on your ListView are treated as any button on your webform would be treated You will need to add another parameter to the InsertButton and UpdateButton of your ListView. This parameter identifies the validation group: **ValidationGroup=”xxxx”** where xxxx is the value of the parameter that you used on the validation control. If you need to prompt the user for confirmation, you still can add the **OnClientClick** parameter to your button.

Setup of ValidationSummary Controls

<%--control to display validation control errors

separate controls are needed for each validation group

HearderText on--%>

<asp:ValidationSummary ID="ValidationSummaryInsert" runat="server"

HeaderText="Following are concerns with your data:" ValidationGroup="igroup"/>

<asp:ValidationSummary ID="ValidationSummaryUpdate" runat="server"

HeaderText="Following are concerns with your data:" ValidationGroup="egroup"/>

OnClientClick event for a button

<AlternatingItemTemplate>

<tr style="background-color: #FFFFFF; color: #284775;">

<td>

<asp:Button runat="server" CommandName="Delete" Text="Remove" ID="DeleteButton"

OnClientClick="return confirm('Are you sure to delete?')"/>

<asp:Button runat="server" CommandName="Edit" Text="Edit" ID="EditButton"/>

</td>

Validation control setup for EditTemplate

|  |  |  |
| --- | --- | --- |
| Field | Validation Control |  |
| TitleTextBoxE | Required |  |
|  | RegularExpression | Limited to 160 characters |
| ReleaseYear | Required |  |
|  | Range | 1950 to today |
| ReleaseLabel | RegularExpression | Limited to 50 characters |

<EditItemTemplate>

<asp:RequiredFieldValidator ID="RequiredTitleTextBoxE" runat="server"

ErrorMessage="Title is required" Display="None"

ControlToValidate="TitleTextBoxE" ValidationGroup="egroup" >

</asp:RequiredFieldValidator>

<asp:RegularExpressionValidator ID="RegExTitleTextBoxE" runat="server"

ErrorMessage="Title is limited to 160 characters" Display="None"

ControlToValidate="TitleTextBoxE" ValidationGroup="egroup"

ValidationExpression="^.{1,160}$">

</asp:RegularExpressionValidator>

<asp:RequiredFieldValidator ID="RequiredReleaseYearTextBoxE" runat="server"

ErrorMessage="ReleaseYear is required" Display="None"

ControlToValidate="ReleaseYearTextBoxE" ValidationGroup="egroup">

</asp:RequiredFieldValidator>

<asp:RangeValidator ID="RangeReleaseYearTextBoxE" runat="server"

ErrorMessage="Release Year must be between 1950 and today."

MinimumValue="1950" MaximumValue="<%# DateTime.Today.Year %>"

Display="None" ControlToValidate="ReleaseYearTextBoxE" Type="Integer"

ValidationGroup="egroup">

</asp:RangeValidator>

<asp:RegularExpressionValidator ID="RegExReleaseLabelE" runat="server"

ErrorMessage="Release Label is limited to 50 characters" Display="None"

ControlToValidate="ReleaseLabelTextBoxE" ValidationGroup="egroup"

ValidationExpression="^.{1,50}$">

</asp:RegularExpressionValidator>

<tr style="background-color: #999999;">

<td>

<asp:Button runat="server" CommandName="Update" Text="Update" ID="UpdateButton"

ValidationGroup="egroup" />

<asp:Button runat="server" CommandName="Cancel" Text="Cancel"  
 ID="CancelButton"/>

</td>

<td>

<asp:TextBox Text='<%# Bind("AlbumId") %>' runat="server" ID="AlbumIdTextBox"  
 Enabled="false" width="50px"/></td>

<td>

<asp:TextBox Text='<%# Bind("Title") %>' runat="server" ID="TitleTextBoxE" /></td>

<td>

<asp:DropDownList ID="ArtistList" runat="server" DataSourceID="ArtistListODS"

DataTextField="Name"

DataValueField="ArtistId"

SelectedValue='<%# Bind("ArtistId") %>' width="300px">

</asp:DropDownList>

</td>

<td>

<asp:TextBox Text='<%# Bind("ReleaseYear") %>' runat="server"  
 ID="ReleaseYearTextBoxE" Width="70px" TextMode="Number"/></td>

<td>

<asp:TextBox Text='<%# Bind("ReleaseLabel") %>' runat="server"   
 ID="ReleaseLabelTextBoxE" /></td>

</tr>

</EditItemTemplate>