

ASP.NET 3.5 Data Control Tips and Tricks

Paul Litwin
Deep Training &
Fred Hutchinson Cancer Research Center

paul@deeptesting.com

Paul Litwin

- **Developer**
 - Focus: ASP.NET, ASP, VB, C#, SQL Server, ...
 - MCSD
 - Microsoft MVP
 - Programming Manger with Fred Hutchinson Cancer Research Center (Seattle)
- **Co-Founder and Senior Trainer**
 - Deep Training
 - www.deeptraining.com
- **Conference Chair/Speaker**
 - Chair, Microsoft ASP.NET Connections
 - Member INETA Speakers Bureau
- **Author**
 - Author/co-author of a dozen books, including...
 - *ASP.NET for Developers*
 - *Access Cookbook, 2nd edition*
 - *Access 2002 Desktop/Enterprise Dev Handbook*
 - *AppDev SQL Server 2005 Reporting Services Course*

Slides & Samples Download

- You can download them from:
 - www.deeptraining.com/litwin

Agenda

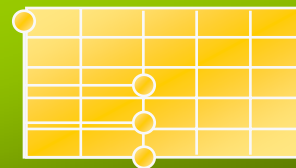
- Programming the Data & Data Source Controls
- Data Control Tips and Tricks (and a few Traps)
- New ListView Control

Programming Data & Data Source Controls

Data Binding in ASP.NET 2.0

Data-bound Control

```
<asp:GridView  
  DataSourceId="MySource"  
  runat="server" />...
```



Data Source Control

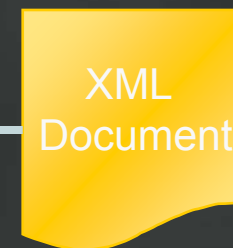
```
<asp:ObjectDataSource  
  Id="MySource"  
  TypeName="System.Xml"  
  SelectMethod="GetCustomers" />
```



Database



Business
Object



XML
Document

Programming Data Controls: Things You Need to Know

1. How to hook in

- What events are most useful
- When to use events of GridView vs. Data Source Control

2. How to programmatically access controls and fields

Tabular Data Controls

- GridView – tabular grid
- DetailsView – form that places fields in a table
 - uses BoundFields by default (with no ID property)
- FormView – form that uses a free-form template
 - uses template fields by default
 - more flexibility
 - Almost always a better choice than DetailsView
- ListView – new very flexible control
 - can look like grid or form
 - new for VS 2008
 - more on this control later in presentation
- All support editing, inserts (not GridView), pagination, template fields

Interesting Data Control Stuff (1 of 2)

- **BoundFields don't have IDs**
 - No easy way to refer to programmatically
- **TemplateFields**
 - Eval provides read-only data-binding
 - Bind provides two-way data-binding
- **Updates not Working?**
 - Check DataSourceId attribute
 - Check DataKeyNames attribute
 - If using ObjectDataSource control, check OldValuesParameterFormatString
 - Defaults to `original_{0}`

Interesting Data Control Stuff (2 of 2)

- **Making Sense of Parameters**
 - Parameters are automatically inferred for update, delete, and insert statements
 - Use explicit parameters when you want to control data types or add additional params
 - Want to avoid passing back a field as a parameter?
 - Use **InsertVisible** property (for inserts)
 - Use **ReadOnly** property
 - Use **Eval** instead of **Bind**

GridView Events of Note

- **DataBound**
 - Use to modify data as it is being loaded into control, e.g., setting a default value
- **RowUpdating**
 - Use to manipulate data before row is updated in database, e.g., saving a timestamp field.
- **RowDeleting**
 - Use to manipulate data before row is deleted from database, e.g., stopping delete.
- **RowUpdated**
 - Use to react to database update, e.g. refreshing another control on page or handling update exceptions.
- **RowDeleted**
 - Use to react to database delete, e.g. refreshing another control on page or handling delete exceptions

Details/Form View Events of Note (1 of 2)

DataBound and the “ing” Events

- **DataBound**
 - Use to modify data as it is being loaded into control, e.g., setting a default value
- **ItemUpdating**
 - Use to manipulate data before row is updated in database, e.g., saving a timestamp field.
- **ItemInserting**
 - Use to manipulate data before row is inserted to database, e.g., saving a timestamp field.
- **ItemDeleting**
 - Use to manipulate data before row is deleted from database, e.g., stopping delete.

Details/Form View Events of Note (2 of 2)

The “ed” Events

- **ItemUpdated**
 - Use to react to database update, e.g. refreshing another control on page or handling update exceptions.
- **ItemInserted**
 - Use to react to database insert, e.g. refreshing another control on page or handling insert exceptions
- **ItemDeleted**
 - Use to react to database delete, e.g. refreshing another control on page or handling delete exceptions
- **DataSourceControl Inserted**
 - Useful for grabbing Identity field value and refreshing page to display new record
 - Note: this is an event of the DataSource control, not the Data control!

Data Control How To's

1. How do I...

- Filter a single data control by multiple controls?
 - Alter parameter collections programmatically

```
protected void txtCompany_TextChanged(object sender, EventArgs e)
{
    sdsCustomers.SelectParameters.Clear();
    sdsCustomers.SelectParameters.Add(new Parameter("CompanyName",
        TypeCode.String, txtCompany.Text + '%'));
}
```

- Example: Filtered.aspx

2. How do I...

- Display a default value of a control during an insert?
- Use DataBound event, checking the CurrentMode property

```
protected void fvCustomers_DataBound(object sender, EventArgs e) {  
    if (fvCustomers.CurrentMode == FormViewMode.Insert) {  
        ((TextBox)fvCustomers.FindControl("CityTextBox")).Text = "Seattle";  
        ((TextBox)fvCustomers.FindControl("CountryTextBox")).Text = "USA";  
    }  
}
```

- Example: FormViewDefault.aspx

3. How do I...

- Refresh one control when another changes?
 - Call DataBind() method of second control from first control's "ed" event
- Example: DrillDown1Page.aspx

4. Trap to Avoid

- Trap: If a page hides and unhides DetailsView/FormView controls, it may produce a bogus “Failed to load viewstate...” error message
 - Fix: Set “EnableViewState” property to False for each of the DetailsView/FormView controls

5. How do I...

- Confirm a record delete for a data control?
 - Use a templated LinkButton control w/ client-side JavaScript and OnClientClick event

```
<asp:TemplateField>
  <ItemTemplate>
    <asp:LinkButton ID="lbDelete" Runat="server"
      OnClientClick="return confirm('OK to delete record?');"
      CommandName="Delete"
      Text = "Delete"></asp:LinkButton>
  </ItemTemplate>
</asp:TemplateField>
```

- Example: [GridViewUpdate2DeleteConfirm.aspx](#)

6. How do I...

- Use a drop-down control for editing with a data control?
 - Use a template field

```
<asp:TemplateField HeaderText="ReportsTo" SortExpression="ReportsTo">
  <EditItemTemplate>
    <asp:DropDownList ID="ddlReportsTo" runat="server"
      SelectedValue='< %# Bind("ReportsTo") %>'
      DataSourceID="sdsReportsTo" DataTextField="EmployeeName"
      DataValueField="EmployeeId"></asp:DropDownList>
  </EditItemTemplate>
  <ItemTemplate>
    <asp:Label ID="lblReportsTo" runat="server"
      Text='< %# Bind("ReportsToName") %>'></asp:Label>
  </ItemTemplate>
</asp:TemplateField>
```

Example: [EmployeeGridWithDropDown.aspx](#)

6. How do I...

- Use a drop-down control for editing with a data control? *(continued)*
 - Don't forget to provide a NULL to the list!

```
<asp:SqlDataSource ID="sdsReportsTo" runat="server" ..."  
    SelectCommand=  
    "SELECT EmployeeId,  
    LastName + ', ' + FirstName AS EmployeeName  
    FROM Employees  
    UNION  
    SELECT NULL, NULL  
    ORDER BY LastName + ', ' + FirstName">  
</asp:SqlDataSource>
```

7. How do I...

- Create cascading drop-down controls within a FormView?
- Should be simple but it's not
 - Why? According to Microsoft's Polita Paulus...

This specific scenario doesn't quite work correctly in a FormView control because when the DataSource raises its DataSourceChanged event, the second dropdown list's Eval statement attempts to evaluate against the container's (FormView's) databinding context, which is only valid at DataBind time. Since this happens outside of DataBind time, the context isn't there and the Eval fails. The best way to work around this is to write some code to manually databind the second (dependent) dropdown list.

7. How do I...

- Create cascading drop-down controls within a FormView? (*continued*)
- Things you need to do
 - *Don't* set SelectedValue of 2nd dropdown declaratively
 - Write code to bind 2nd dropdown in its DataBound event
 - Write code to save SelectedValue during FormView's ItemInserting and ItemUpdating events
- Example: FVCascadingDropdowns.aspx

8. How do I...

- Update fields that aren't shown (e.g., time stamp fields) when a row is updated?
 - Template the fields so you can control updatability/visibility during edits/inserts
 - For Insert
 - Use `ItemInserting` event and `Values` collection
 - For Update
 - Use `ItemUpdating` event and `NewValues` collection
- Example: `DetailsViewHiddenFields.aspx`

9. Trap to Avoid

- Trap: Checkboxes can't be set to Null
 - Fix: Make sure bit field always has a default value of 0 when using a `SqlDataSource`; or convert `DBNull` to 0 when using an `ObjectDataSource`

10. How do I...

- Not pass an identity column during an insert to the database?
 - Set **InsertVisible** property of control to false
- AND...After the insert, retrieve the newly inserted record, complete with identity column?
 - Use **Inserted** event of **DataSource** control and grab the value of the identity column using `e.Command.Parameters` collection
- Example: [EmpDrillDown.aspx/EmpDetails.aspx](#)

11. How do I...

- Handle Exceptions?
 - Use one of “ed” events & retrieve exception:

```
Exception exp = e.Exception;  
if (exp != null)  
{  
    lblMsg.Text = "Insert failed. Details: " + exp.Message;  
    // If using ObjectDataSource use exp.InnerException.Message instead  
    e.ExceptionHandled = true;  
    e.KeepInInsertMode = true; // keeps user editing.  
}
```

- Example: FormViewExp.aspx

12. Trap to Avoid

- Trap: When hooking up stored procedures to DataSource controls containing output parameters, Visual Studio will set the Direction to “inputoutput” instead of “output” which will create problems
 - Fix: Change direction to “output”

13. How do I...

- Hook up an class to a data control?
 - Use the ObjectDataSource control
- Example: GridViewODS.aspx

14. How do I...

- Get at Dataset behind Data control?
 - Use code like this:

```
DataSourceSelectArguments args = new DataSourceSelectArguments();  
DataView dv = (DataView)sdsAbstract.Select(args);
```

- For example, you might use this technique to pass a filtered Dataset off to Excel
- Example: [Filtered2Excel.aspx](#)

ListView Control

Why Another Data Control?

- **GridView, FormView, and DetailView only cover a limited universe**
 - Useful for Grid or Form
 - GridView doesn't support inserts
- **Repeater and DataList limitations**
 - Don't support pagination or sorting
 - Don't support grouping


Enter ListView Control

- Most flexible of *View controls
- More powerful than Repeater or DataList
- Supports templates, grouping, sorting, and paging (with DataPager control)

Basic ListView

- **LayoutTemplate** defines container
 - itemPlaceholder control is replaced by ItemTemplate
 - Doesn't have to be a Placeholder control

```
<asp:ListView ID="lstMovies" DataSourceID="srcMovies"
runat="server">
  <LayoutTemplate>
    <asp:Placeholder ID="itemPlaceholder" runat="server" />
  </LayoutTemplate>
  <ItemTemplate>
    <%# Eval("Title") %>
  </ItemTemplate>
  <ItemSeparatorTemplate>
    <hr />
  </ItemSeparatorTemplate>
</asp:ListView>
```



Very Simple Example

- **VerySimpleListView.aspx**

```
<asp:ListView
  ID="lstMovies"
  DataSourceId="srcMovies"
  runat="server">
  <LayoutTemplate>
    <asp:Placeholder id="itemPlaceholder" runat="server" />
  </LayoutTemplate>
  <ItemTemplate>
    <%# Eval("Title") %>
  </ItemTemplate>
  <ItemSeparatorTemplate>
    <hr />
  </ItemSeparatorTemplate>
</asp:ListView>
```

Templates

- LayoutTemplate – container for entire control
- ItemTemplate – replaces itemPlaceholder for each data item
- AlternatingItemTemplate – every other item
- ItemSeparatorTemplate – separates items (and alternating items)
- GroupTemplate – container for groups
- GroupSeparatorTemplate – separator between groups
- EmptyItemTemplate – used for left over items in group
- EmptyDataTemplate – used when no data items
- SelectedItemTemplate – displayed for selected item
- EditItemTemplate – display for editing item
- InsertItemTemplate – display for inserting a new item

Interesting ListView Attributes

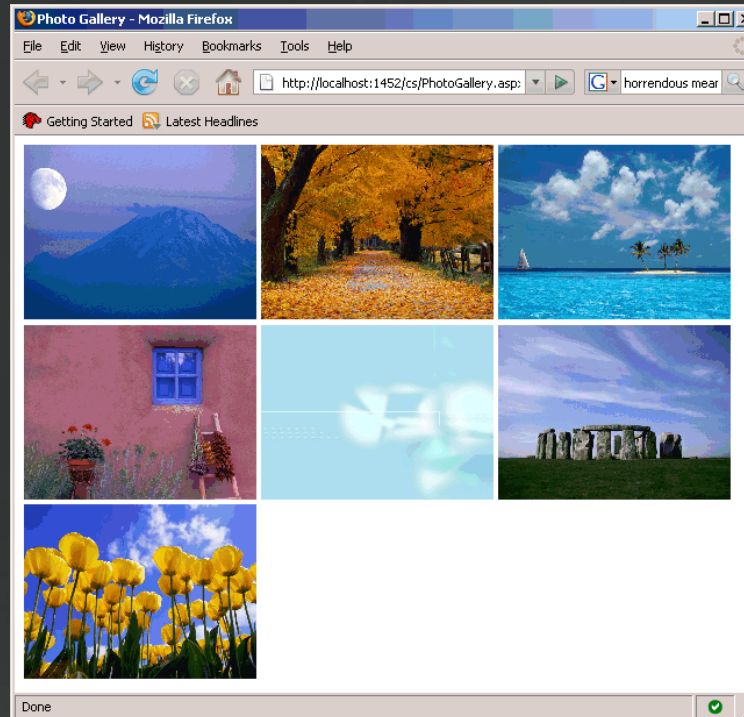
- ItemPlaceHolderId – defaults to itemPlaceholder
- GroupPlaceHolderId – defaults to groupPlaceholder
- GroupItemCount – # of items to display per group; defaults to 1
- InsertItemPosition – can be FirstItem, LastItem, or None

Groups

- Adding a GroupTemplate allows you to create multiple columns
- Basic steps...
 1. Set GroupItemCount attribute of ListView to # of cols
 2. Insert 1st Placeholder ctrl in LayoutTemplate that points to groupPlaceholder control
 3. Insert 2nd Placeholder ctrl in GroupTemplate points to itemPlaceholder control

Group Example

- PhotoGallery.aspx (from ASP.NET 3.5 Unleashed)



- PhotoGallery_EmptyItem.aspx (version that uses EmptyItem template)

Group Example

- **PhotoGallery.aspx**

```
<asp:ListView
  ID="lstPhotos" GroupItemCount="3"  runat="server">
  <LayoutTemplate>
    <asp:Placeholder id="groupPlaceholder" runat="server" />
  </LayoutTemplate>
  <GroupTemplate>
    <div>
      <asp:Placeholder id="itemPlaceholder" runat="server" />
    </div>
  </GroupTemplate>
  <ItemTemplate>
    <asp:Image
      id="imgPhoto" ImageUrl='<%=# Container.DataItem %>'
      Width="200px" runat="server" />
  </ItemTemplate>
</asp:ListView>
```

Sorting

- Add button/link button controls to header with
 - `CommandName="Sort"`
 - `CommandArgument="field_name"`
- Example: `SortedListView.aspx`

Editing Data

- Use `DataKeyNames` attribute
- Add `EditItemTemplate`, `InsertItemTemplate` as needed
- `InsertItemTemplate` always displayed
 - Control placement with `InsertItemPosition` attribute of `ListView`
 - Can be `FirstItem`, `LastItem`
- Add buttons with `CommandName` of
 - Edit – switch to `EditItemTemplate`
 - Delete – delete item
 - Update – save edited item
 - Insert – save new item
 - Cancel – cancel edits for edits/inserts
- Example: `EditMovies.aspx`

Paging ListViews

- **DataPager** pages ListViews
- **Properties**
 - PageSize – number of items on page
 - PagedControlId – id of ListView to page
 - Fields – UI controls for paging
 - NextPreviousPagerField
 - NumericPagerField
 - TemplatePagerField – custom
 - StartRowIndex – index of first item to show
 - MaximumRows – max # of rows to retrieve from data source
 - TotalRowCount – total # of items from data source

Pager Example

- PagedMovies.aspx
- and EditMovies.aspx

Questions?

Thank You!

- Please complete evaluation forms
- Contact: paul@deeptraining.com
- Download slides & samples from
 - www.deeptraining.com/litwin