

The Data Controls

Today you will learn

- Editing with the GridView
- Sorting and Paging the GridView
- Using GridView Templates
- The DetailsView and FormView

Editing with the GridView

```
<asp:GridView ID="gridProducts" runat="server" DataSourceID="sourceProducts" AutoGenerateColumns="False" DataKeyNames="ProductID">
  <Columns>
    <asp:BoundField DataField="ProductID" HeaderText="ID" ReadOnly="True" />
    <asp:BoundField DataField="ProductName" HeaderText="Product Name" />
    <asp:BoundField DataField="UnitPrice" HeaderText="Price" />
    <asp:CommandField ShowEditButton="True" />
  </Columns>
</asp:GridView>

<asp:SqlDataSource id="sourceProducts" runat="server" ConnectionString="<%= $ConnectionStrings:Northwind %>" SelectCommand="SELECT ProductID,
ProductName, UnitPrice FROM Products"
UpdateCommand="UPDATE Products SET ProductName=@ProductName, UnitPrice=@UnitPrice WHERE ProductID=@ProductID" />
```

(Demo)

Sorting and Paging the GridView

- Set **GridView.AllowSorting** property to True for Sorting.
- Define a **SortExpression** for each column that can be sorted.

```
<asp:BoundField DataField="ProductName" HeaderText="Product Name"
SortExpression="ProductName" /> (Demo)
```

Sorting and Selecting

- If you select the second row and perform a sort, the second row will still be selected in the new page, even though this isn't the record you selected.
- This problem will solve by setting `GridView.EnablePersistedSelection` property True.

Paging

- In automatic paging, the full results are retrieved from the data source and placed into a DataSet. Once the DataSet is bound to the GridView, however, the data is subdivided into smaller groupings.

Paging Members of the GridView

Property	Description
AllowPaging	Enables or disables the paging of the bound records. It is False by default.
PageSize	Gets or sets the number of items to display on a single page of the grid. The default value is 10.
PageIndex	Gets or sets the zero-based index of the currently displayed page, if paging is enabled.

Paging

PagerSettings	Provides a PagerSettings object that wraps a variety of formatting options for the pager controls. These options determine where the paging controls are shown and what text or images they contain. You can set these properties to fine-tune the appearance of the pager controls, or you can use the defaults.
PagerStyle	Provides a style object you can use to configure fonts, colors, and text alignment for the paging controls.
PageIndexChanging and PageIndexChanged events	Occur when one of the page selection elements is clicked, just before the PageIndex is changed (PageIndexChanging) and just after (PageIndexChanged).

```
<asp:GridView ID="GridView1" runat="server" DataSourceID="sourceProducts"  
    PageSize="10" AllowPaging="True" >
```

...

```
</asp:GridView>
```

(Demo)

Using GridView Templates

Using GridView Templates

- The TemplateField allows you to define a completely **customized template** for a column.
- Inside the template you can add control tags, arbitrary HTML elements, and data binding expressions.

(Demo)

Using Multiple Templates

TemplateField Templates

Mode	Description
HeaderTemplate	Determines the appearance and content of the header cell.
FooterTemplate	Determines the appearance and content of the footer cell (if you set ShowFooter to True).
ItemTemplate	Determines the appearance and content of each data cell.
AlternatingItemTemplate	Determines the appearance and content of even-numbered rows. For example, if you set the AlternatingItemTemplate to have a shaded background color, the GridView applies this shading to every second row.

Using Multiple Templates

EditItemTemplate	Determines the appearance and controls used in edit mode.
InsertItemTemplate	Determines the appearance and controls used in edit mode. The GridView doesn't support this template, but the DetailsView and FormView controls (which are described later in this chapter) do.

Editing Templates in Visual Studio

(Demo Edit Template)

Editing with a Template

Limitations of Standard Editing GridView Records (Using GridView.EditIndex="True" and add CommandField column, set ShowEditButton="True") :

- It's not always appropriate to edit values using a text box.
- You get no validation.
- The visual appearance is often ugly.

Using Multiple Templates

```
<asp:TemplateField HeaderText="Status">
    <ItemStyle Width="100px" />
    <ItemTemplate>
        <b>In Stock:</b> <%# Eval("UnitsInStock") %><br />
        <b>On Order:</b> <%# Eval("UnitsOnOrder") %><br />
        <b>Reorder:</b> <%# Eval("ReorderLevel") %>
    </ItemTemplate>
    <EditItemTemplate>
        <b>In Stock:</b> <%# Eval("UnitsInStock") %><br />
        <b>On Order:</b> <%# Eval("UnitsOnOrder") %><br /><br />
        <b>Reorder:</b>
        <asp:TextBox Text="<%# Bind("ReorderLevel") %>" Width="25px" runat="server" id="txtReorder" />
    </EditItemTemplate>
</asp:TemplateField>
```



	ID	Product Name	Price	Status
Update Cancel	1	Chai	18.0000	In Stock: 36 On Order: 0 Reorder: 10

Editing with Validation

```
<asp:TemplateField HeaderText="Status">
    <ItemStyle Width="100px" />
    <ItemTemplate>
        <b>In Stock:</b> <%# Eval("UnitsInStock") %><br />
        <b>On Order:</b> <%# Eval("UnitsOnOrder") %><br />
        <b>Reorder:</b> <%# Eval("ReorderLevel") %>
    </ItemTemplate>
    <EditItemTemplate>
        <b>In Stock:</b> <%# Eval("UnitsInStock") %><br />
        <b>On Order:</b> <%# Eval("UnitsOnOrder") %><br /><br />
        <b>Reorder:</b>
        <asp:TextBox Text='<%# Bind("ReorderLevel") %>' Width="25px" runat="server" id="txtReorder" />
        <asp:RangeValidator id="rngValidator" MinimumValue="0" MaximumValue="100" ControlToValidate="txtReorder"
            runat="server" ErrorMessage="Value out of range." Type="Integer"/>
    </EditItemTemplate>
</asp:TemplateField>
```

	ID	Product Name	Price	Status
Update Cancel	1	Chai	18.0000	In Stock: 36 On Order: 0 Reorder: -5 Value out of range.

Editing Without a Command Column

- LinkButton is used to do editing. (Demo)

The DetailsView and FormView

- Multiple pages can be displayed by numbering by setting AllowPaging=“True” in DetailsView. (Demo)

Defining fields in DetailsView:

- Set AutoGenerateRows=“False” in DetailsView
- Adding <Fields></Fields>
- Add Column Types

(Demo)

- The beauty of the FormView template model is that it matches the model of the TemplateField in the GridView quite closely.

(Demo)

END OF LECTURE