The Data Controls

Today you will learn

- The GridView
- Formatting the GridView
- Selecting a GridView Row
- Editing with the GridView

The Data Controls

- Rich Data Controls designed for Data Binding are:
  - GridView: Shows large tables of information.
  - DetailsView: Shows a single record at a time.
  - FormView: Shows a single record at a time and supports editing.
  - ListView: show multiple records.

The GridView

- The GridView is an extremely flexible grid control that displays a multicolumn table.
- Ready-made Functionality includes features for automatic paging, sorting, selecting and editing

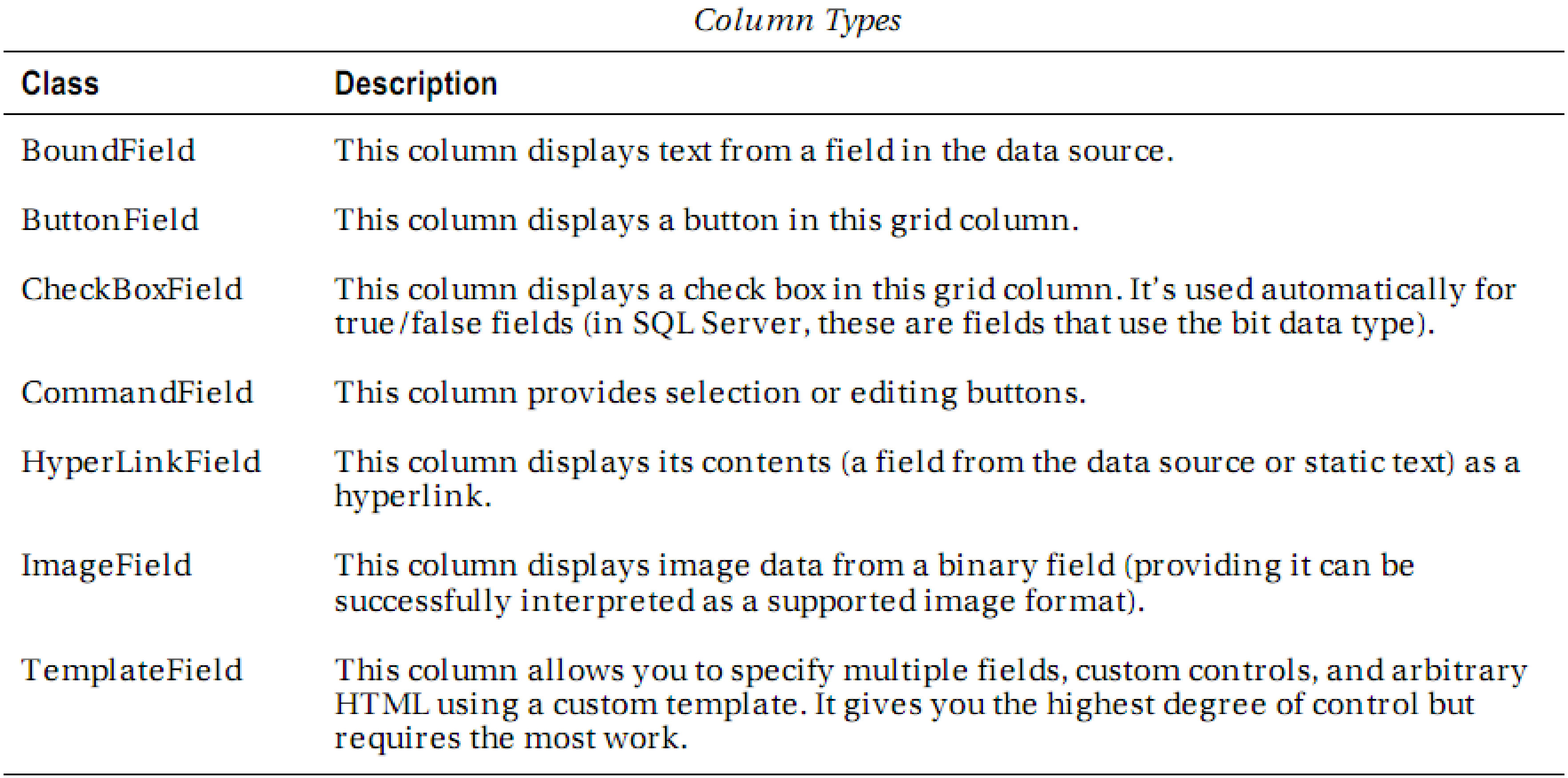
Automatically Generating Columns

- GridView support for DataSource property
- Does not provide DataTextField and DataValueField

(Demo)

Defining Columns

- Set AutoGenerateColumns to False and define the columns in the <Columns> section of the GridView control tag.



Column Types

Class	Description
BoundField	This column displays text from a field in the data source.
ButtonField	This column displays a button in this grid column.
CheckBoxField	This column displays a check box in this grid column. It's used automatically for true/false fields (in SQL Server, these are fields that use the bit data type).
CommandField	This column provides selection or editing buttons.
HyperLinkField	This column displays its contents (a field from the data source or static text) as a hyperlink.
ImageField	This column displays image data from a binary field (providing it can be successfully interpreted as a supported image format).
TemplateField	This column allows you to specify multiple fields, custom controls, and arbitrary HTML using a custom template. It gives you the highest degree of control but requires the most work.

Defining Columns

- Set AutoGenerateColumns to False and define the columns in the <Columns> section of the GridView control tag. (Demo)
- **Advantages of Explicitly Defining Columns:**
- You can easily fine-tune your column order , column headings, and other details by tweaking the properties of your column object.
- You can hide columns you don’t want to show by removing the column tag.
- You’ll see your columns in the design environment.
- You can add extra columns to the mix for selecting, editing, and more.

Generating Columns with Visual Studio (Demo)

Formatting the GridView

- Add DataFormatString property
- Apply the perfect mix of colors, fonts, borders, and alignment options to each aspect of the grid, from headers to data items.

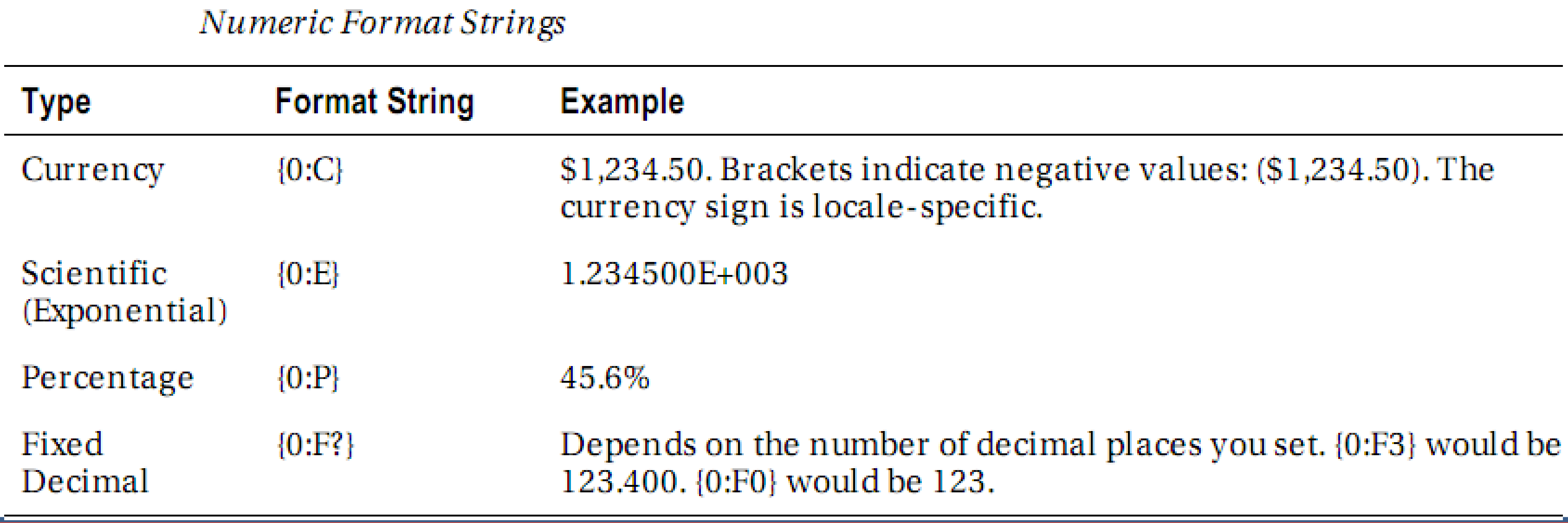
Formatting Fields

- Format strings generally consist of a placeholder and a format indicator, which are wrapped inside curly brackets. A typical format string looks something like this:

{0:C}

- In this case, the 0 represents the value that will be formatted, and the letter indicates a predetermined format style. (C = Currency Style)

```
<asp:BoundField DataField="UnitPrice" HeaderText="Price" DataFormatString="{0:C}" />
```

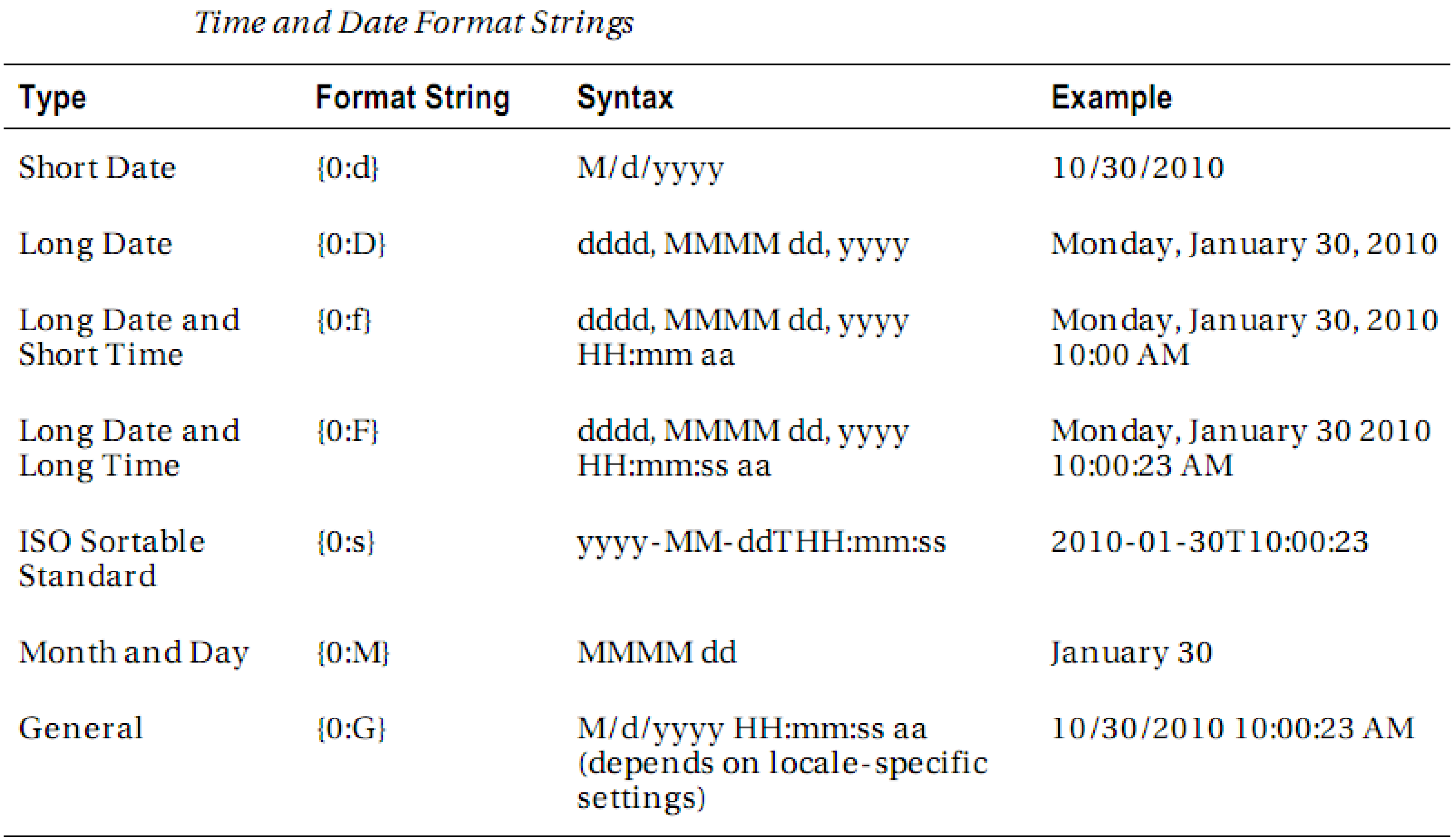


Numeric Format Strings

Type	Format String	Example
Currency	{0:C}	$1,234.50. Brackets indicate negative values: ($1,234.50). The currency sign is locale-specific.
Scientific (Exponential)	{0:E}	1.234500E+003
Percentage	{0:P}	45.6%
Fixed Decimal	{0:F?}	Depends on the number of decimal places you set. {0:F3} would be 123.400. {0:F0} would be 123.

Formatting Fields

```
<asp:BoundField DataField="BirthDate" HeaderText="Birth Date"
DataFormatString="{0:MM/dd/yy}" />
```

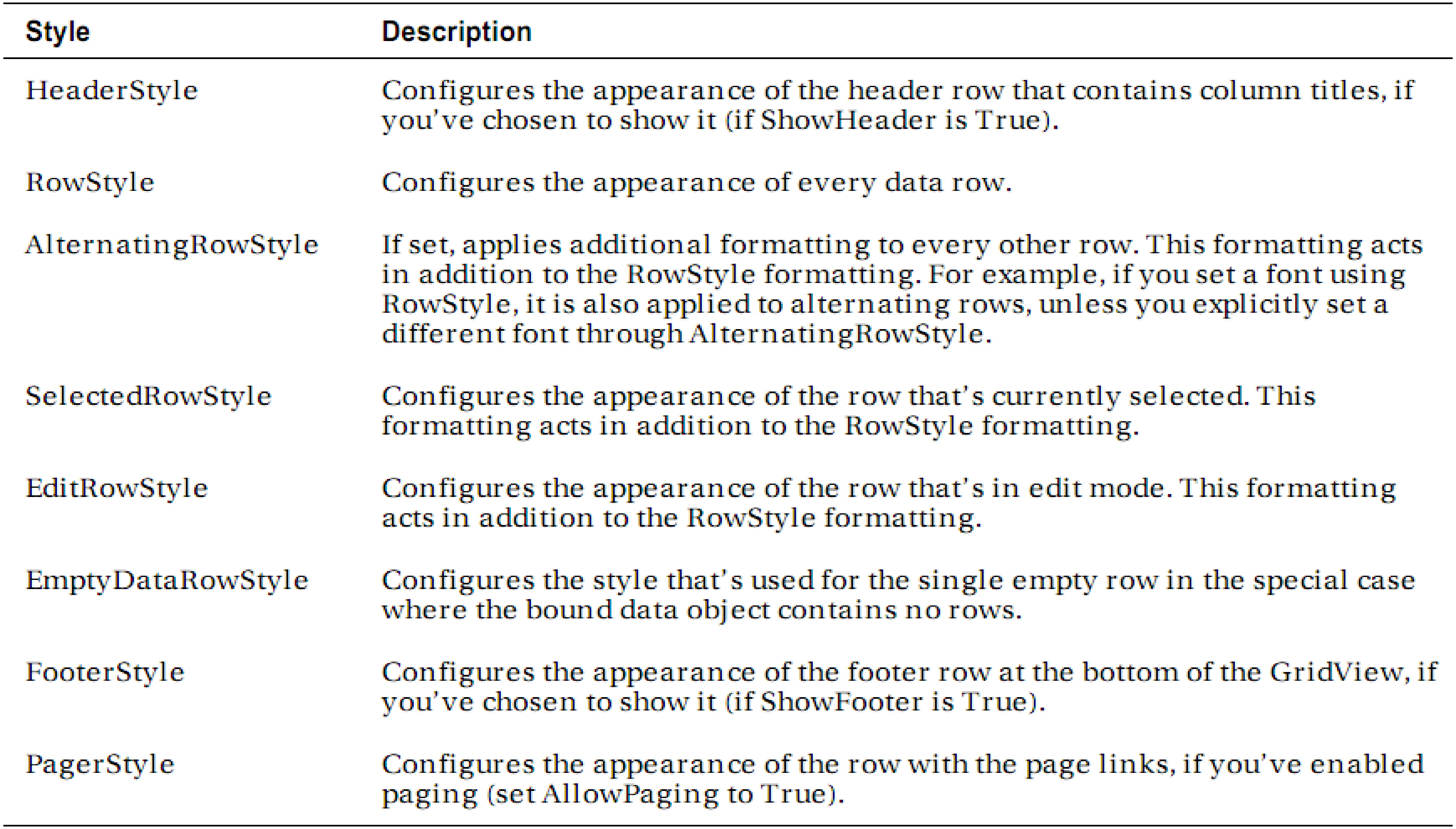


Time and Date Format Strings

Type	Format String	Syntax	Example
Short Date	{0:d}	M/d/yyyy	10/30/2010
Long Date	{0:D}	dddd, MMMM dd, yyyy	Monday, January 30, 2010
Long Date and Short Time	{0:f}	dddd, MMMM dd, yyyy HH:mm aa	Monday, January 30, 2010 10:00 AM
Long Date and Long Time	{0:F}	dddd, MMMM dd, yyyy HH:mm:ss aa	Monday, January 30 2010 10:00:23 AM
ISO Sortable Standard	{0:s}	yyyy-MM-ddTHH:mm:ss	2010-01-30T10:00:23
Month and Day	{0:M}	MMMM dd	January 30
General	{0:G}	M/d/yyyy HH:mm:ss aa (depends on locale-specific settings)	10/30/2010 10:00:23 AM

Using Styles

- GridView supports basic eight styles.



Style	Description
HeaderStyle	Configures the appearance of the header row that contains column titles, if you've chosen to show it (if ShowHeader is True).
RowStyle	Configures the appearance of every data row.
AlternatingRowStyle	If set, applies additional formatting to every other row. This formatting acts in addition to the RowStyle formatting. For example, if you set a font using RowStyle, it is also applied to alternating rows, unless you explicitly set a different font through AlternatingRowStyle.
SelectedRowStyle	Configures the appearance of the row that's currently selected. This formatting acts in addition to the RowStyle formatting.
EditRowStyle	Configures the appearance of the row that's in edit mode. This formatting acts in addition to the RowStyle formatting.
EmptyDataRowStyle	Configures the style that's used for the single empty row in the special case where the bound data object contains no rows.
FooterStyle	Configures the appearance of the footer row at the bottom of the GridView, if you've chosen to show it (if ShowFooter is True).
PagerStyle	Configures the appearance of the row with the page links, if you've enabled paging (set AllowPaging to True).

Using Styles

```
<asp:GridView ID="GridView1" runat="server" DataSourceID="sourceProducts"
 AutoGenerateColumns="False">
   <RowStyle BackColor="#E7E7FF" ForeColor="#4A3C8C" />
   <HeaderStyle BackColor="#4A3C8C" Font-Bold="True" ForeColor="#F7F7F7" />
   <Columns>
     <asp:BoundField DataField="ProductID" HeaderText="ID" />
     <asp:BoundField DataField="ProductName" HeaderText="Product Name" />
     <asp:BoundField DataField="UnitPrice" HeaderText="Price" />
   </Columns>
</asp:GridView>
```

- **Column Specific Style** (Demo)
- **Configuring Styles with Visual Studio** (Demo – Property (AutoFormat)

Formatting-Specific Values

- Write Code after GridView.RowDataBound event fire. (Demo)

decimal totalAmt = (decimal)DataBinder.Eval(e.Row.DataItem, "total_amt");

Selecting a GridView Row

- SelectedRowStyle determines how the selected row or cell will appear.
- Default it uses GridView.SelectedIndex property which is -1(No item Selected).
- Use GrideView_SelectedIndexChaged() event.

Adding a Select Button

```
<asp:CommandField ShowSelectButton="True" ButtonType="Button"
SelectText="Select" />
```

(Demo 4)

Using a Data Field As a Select Button

```
<asp:ButtonField CommandName="Select" ButtonType="Button"
 DataTextField="ProductID" />
```

(Demo 4)

END OF LECTURE