

Chapter 9

Customizing Data with Web Controls



Table of Contents

- Overview
- Objectives
- Instructor Notes
- Quick Quizzes
- Discussion Questions
- Key Terms

Overview

In this chapter, you will learn how to allow the user to select a record in a GridView control, and retrieve and display the results in the DetailsView control. Lastly, you will learn about ways to create your web pages efficiently by using custom templates.

Chapter Objectives

- In this chapter, students will:
- Learn about the various Data controls to display data
 - Modify the Data controls using templates and styles
 - Use the GridView control to display data
 - Build a master-detail page with the GridView control and DetailsView controls
 - Create custom templates

Instructor Notes

Templates and Styles Available to the Data Controls

In this section, you will learn to use different types of columns, templates, and column style properties in order to modify the content and appearance of the data within the Data controls.

Teaching Tip	Students will also learn how to apply templates and styles to the DataList control. Many of the techniques can also be applied to the other controls because they inherit the same classes. Take the opportunity to show students common styles and templates.
--------------	--

Templates Available to Data Controls

When you bound data to a DropDownList control, the data simply was displayed in the control. With other types of controls, however, you can modify how the data is displayed. For example, if you display the data in a GridView control, you may want to modify the column headings using the HeaderText property. To be able to control how the data is displayed, you can use a template. A template is used to format Data controls. The **templates** in your Data control are the parts of the control that determine how to display specific data.

Details of these templates are as follows:

- **HeaderTemplate** is used to configure the data contained within the header section. Instead of using the HeaderText property for the column control, you will use HeaderTemplate to modify contents of the header cell. Therefore, you can use these templates to display images as well as textual content.
- **FooterTemplate** is used to configure the data contained within the footer section.
- **ItemTemplate** is used to configure how the row will appear in display mode. The data is actually displayed in read-only mode in this template. The GridView and DetailsView controls allow you to format the rows of data using columns. In the ItemTemplate, the data is generally bound to the field in the database. Instead of inserting the name of a graphic, you can insert an image control and set the src property to the name of the graphic. Then, the image would be displayed in the row in display mode.
- **EditItemTemplate** allows you to configure the columns in edit mode. Therefore, the user can change the values of the data. Instead of editing the data using text boxes, you can use other types of controls such as check boxes and drop-down lists to edit the data.
- **AlternatingItemTemplate** allows you to configure every other row of data.
- **SeparatorTemplate** allows you to configure the row that is displayed between each row of data.
- **SelectedTemplate** allows you to configure the row that is currently selected.

- **PagerTemplate** allows you to configure a custom template for the page navigation controls. The Data control may display one or more records on a page. The page navigation controls allow you to change which page of records is displayed. By default, for a GridView control, 10 records are displayed on the page. If there were 20 records, you could view the first 10 records on the first page and then use the page navigation controls to move to the next set of 10 records.
- **EmptyItemTemplate** allows you to provide alternative content in case your query returns no records.

Teaching Tip

Not all of the templates are available for all of the Data controls. If you have modified the GridView control to contain custom columns, then you may have access to additional templates.

You might encourage students to build a chart to display which Data controls support which features, templates, styles, properties, and methods.

Styles Available to Data Controls

Once you have a template created, you can format the Data control using styles. The following list describes the styles that can be used to modify the DataList controls:

- *AlternatingItemStyle*—Identifies the style for alternating rows.
- *EditItemStyle*—Identifies the style for a row being edited; the *EditItemStyle* is used in conjunction with the *EditItemTemplate* in the Data control.
- *FooterStyle*—Identifies the style for the footer row.
- *HeaderStyle*—Identifies the style for the header row.
- *ItemStyle*—Identifies the style for individual items within the list or control.
- *SelectedItemStyle*—Identifies the style for the currently selected item.
- *SeparatorStyle*—Identifies the style for content that is between each item.

The GridView control allows you to configure the style of the Pager controls. **Pager controls** are used when the number of rows exceeds the number of rows that can be displayed on the web page. The numbers of rows that are displayed on the web page are configured using the *PageSize* property of the Data control. The following list describes other styles that can be used to modify the GridView controls:

- *AlternatingRowStyle*—Identifies the style for alternating rows.
- *EditRowStyle*—Identifies the style for a row being edited.
- *EmptyDataRowStyle*—Identifies the style applied when there is no data in the row.
- *PagerStyle*—Identifies the style for the page navigation controls with GridView and DataList controls.
- *RowStyle*—Identifies the style for content that is between each item.
- *SelectedRowStyle*—Identifies the style for content that is between each item.

Quick Quiz

1. The _____ in your Data control are the parts of the control that determine how to display specific data.
Answer: templates
2. _____ is used to configure the data contained within the footer section.
Answer: FooterTemplate
3. _____ allows you to configure every other row of data.
Answer: AlternatingItemTemplate
4. _____ allows you to provide alternative content in case your query returns no records.
Answer: EmptyItemTemplate

5. _____ are used when the number of rows exceeds the number of rows that can be displayed on the web page.

Answer: Pager controls

Discussion Questions

Templates

- Discuss when to use each type of template. What part of the control is modified by the template?

Styles

- List the styles with each Data control. Identify the properties associated with that style that are different from other controls.

Modifying Data Controls Using Templates

In this section, you will learn to use different types of columns, templates, and column style properties in order to modify the content and appearance of the data. Many of the techniques can also be applied to the other controls because they inherit the same class. You will learn how to apply templates to the Repeater and DataList controls.

Using Templates to Modify the Repeater Control

You can use templates to modify the Repeater control. Remember that the Repeater control repeats only once for each row of data. You cannot configure the Repeater control in Design view. You must modify the control manually in the Source code view of the web page. The Repeater control is used when you want to have the most flexibility for displaying and managing your data.

Using Templates to Modify the DataList Control

Table tags were used within the ItemTemplate to format the output of the Repeater control. One of the advantages of using the DataList control rather than the Repeater control is that the DataList control can be used to create multiple columns of data without using the table tags in the ItemTemplate.

Discussion Questions

Templates

- Compare the different templates and properties used for the Repeater and the DataList controls.

Using the GridView Control to Display Data

The GridView control is a Data control that allows you to display data in a table. The GridView control allows you to bind data to individual columns or to create custom columns. In this section, you will learn about the different types of columns, the column properties, the data in the columns, and what templates and styles are available to configure for the GridView control. You will also learn how to customize filtering, paging, and sorting of data in the GridView control.

GridView Columns

The GridView control contains a collection of columns. Each column can contain different types of information. Some columns are bound to specific fields in the database in order to display and manage data from the database.

Columns can be bound to data in the record or unbound. When a column is bound to data, it means that the data is retrieved from the data source for the current record, and displayed in that column. If the GridView control allows you to edit the record, a bound column will automatically update the database with the new value for the current record.

Teaching Tip

The book contains examples with many types of columns, but not all in the same exercise. You may want to request students to use different columns in their projects.

--	--

Bound columns are referred to as **BoundFields**. A BoundField is used to retrieve and display the data. In the previous chapter, you used BoundFields to display the data in display mode. **Display mode** is a state where you are only displaying the data. **Edit mode** allows you to modify data.

Teaching Tip

In this chapter, students will work with display mode. In the next chapter, they will learn how to edit the GridView control in edit mode. This allows students to accomplish one task before learning a second task. Because editing will be a place where students will have more questions, this is covered in a separate chapter, after they have mastered using the GridView control to display data.

Column Properties

BoundFields are columns used to manage the display and updating of data in the GridView control. All BoundField columns also allow you to specify the **DataField** property, which identifies the column from the data source bound to the column, and the **DataFormatString** property, which identifies the formatting rules for the contents of the BoundField object. You must set the ReadOnly property to true for fields where you modify the DataFormatString property. The **ReadOnly property** is used with BoundField objects to stop the user from editing a column when the GridView control is in edit mode.

All columns within the GridView control allow you to configure the appearance of the column, along with the **HeaderText**, **HeaderImageUrl**, and **FooterText** properties. You can also set the **Visible** property for a column, which allows you to show or hide the column. The **SortExpression** property can be used to identify the column from the data source that is used when sorting the column.

Teaching Tip

The column properties are specific to the type of column and data contained. You should point out the properties each time you work with a new type of column.

Column Data

It is important to understand how the GridView control refers to the data in case you need to access the data programmatically. In the update procedure, when you pass a request to retrieve the contents of a cell in a bound column, the request is passed as an argument. So, when you update a BoundField column, the contents of the columns are sent using **GridViewCommandEventArgs**; the arguments in the column are represented with the variable *e*. When you want to access a specific cell within a column, you must use the **Item** property of the *e* object. The *e.Item* property contains an indexed Cells collection that contains one Cell object for each column in the GridView control.

Because there are multiple controls in the column, you may not know which control represents the data. You can use the **FindControl** method and the control's name to retrieve the data within the cell.

Teaching Tip

Advanced students usually create template columns with more than one data item. They will appreciate being referred to this section, which covers how to reference the data within the column.

Quick Quiz

- _____ are used to insert a button that can activate row-specific procedures.
Answer: BoundFields
- _____ is raised when any button is clicked within a GridView control.
Answer: ItemCommand event
- _____ is a state where you are only displaying the data.
Answer: Display mode

4. The data in the column can be set to _____ so that the data cannot be modified when the other fields are updated.

Answer: read-only

5. When you want to access a specific cell within a column, you must use the _____ of the e object.

Answer: Item property

Discussion Questions

Columns

- Compare the different types of columns in the GridView control, and identify the properties available and type of data contained in the column.

Building a Master-Detail Page with GridView and DetailsView Controls

It is very common that in a web database application, a user will want to search for a particular record, and then display details about that record. The type of web page that supports this type of display is known as a **master-detail** page. Once you learn the technique to bind a control to the value in another Data control or other source, you easily can modify the steps to work with a variety of data sources and Data controls. In a master-detail page, you configure the master control to pass the value to the detail control. Often the master control is the GridView control and the detail control is the DetailsView control.

Teaching Tip

The master-detail page is an effective way to select a subset of data. You can use this method to create filters using parameters. You can pass parameters to a control using other controls, programming variables, session variables, static values, and other methods.

Try creating multiple master-detail pages. You can use any combination of Data controls.

Creating Custom Templates

It is important to be efficient when you build your web sites. You have learned how to use snippets and the Toolbox to store reusable code. Another tool to improve your efficiency with web programming is custom templates. In order to understand how to customize templates, you need to understand how templates are created and where they are stored. Visual Studio allows the user to create item templates and project templates.

Visual Studio stores user defined templates separate from the application templates. User templates are templates that are only available to the individual user. When Visual Studio was installed, a folder was created for the user; this folder contains user-specific files such as the user templates. Visual Studio stores the user defined templates at C:\Documents and Settings\[*UserName*]\My Documents\Visual Studio 2005\Templates.

**Teaching
Tip**

This is included because you can build templates for your class or have your class build templates in their groups.

Key Terms

- **AlternatingItemTemplate:** Allows the configuration of every other row of data
- **BoundFields:** Columns used to manage the display and updating of data in the GridView control
- **EditItemTemplate:** Allows the configuration of the columns in edit mode
- **EmptyItemTemplate:** Allows alternative content to be provided in case the query returns no records
- **ItemCommand event:** Raised when any button is clicked within a GridView control
- **ItemTemplate:** Used to configure how the row will appear in display mode
- **Pager controls:** Used when the number of rows exceeds the number of rows that can be displayed on the web page
- **PagerTemplate:** Allows the configuration of a custom template for the page navigation controls
- **ReadOnly property:** Used with BoundField objects to stop the user from editing a column when the GridView control is in edit mode
- **SelectedTemplate:** Allows the configuration of the row that is currently selected
- **SortExpression property:** Used to identify the column from the data source that is used when sorting the column
- **Templates:** The parts of the data control that determine how to display specific data