When it comes to data binding, not all ASP.NET controls are created equal. In the previous chapter, you saw how data binding can help you automatically insert single values and lists into all kinds of common controls. In this chapter, you’ll concentrate on three more advanced controls—GridView, DetailsView, and FormView—that allow you to bind entire tables of data.

The rich data controls are quite a bit different from the simple list controls. For one thing, they are designed exclusively for data binding. They also have the ability to display more than one field at a time, often in a table-based layout or according to what you’ve defined. They also support higher-level features such as selecting, editing, and sorting.

The rich data controls include the following:

- **GridView**: The GridView is an all-purpose grid control for showing large tables of information. The GridView is the heavyweight of ASP.NET data controls.

- **DetailsView**: The DetailsView is ideal for showing a single record at a time, in a table that has one row per field. The DetailsView also supports editing.

- **FormView**: Like the DetailsView, the FormView shows a single record at a time and supports editing. The difference is that the FormView is based on templates, which allow you to combine fields in a flexible layout that doesn’t need to be table based.

- **ListView**: The ListView plays the same role as the GridView—it allows you to show multiple records. The difference is that the ListView is based on templates. As a result, using the ListView requires a bit more work and gives you slightly more layout flexibility. The ListView isn’t described in this book, although you can learn more about it in the Visual Studio Help, or in the book *Pro ASP.NET 4.5 in VB* (Apress).

In this chapter, you’ll explore the rich data controls in detail.

**The GridView**

The GridView is an extremely flexible grid control that displays a multicolumn table. Each record in your data source becomes a separate row in the grid. Each field in the record becomes a separate column in the grid.

The GridView is the most powerful of the rich data controls you’ll learn about in this chapter because it comes equipped with the most ready-made functionality. This functionality includes features for automatic paging, sorting, selecting, and editing. The GridView is also the only data control you’ll consider in this chapter that can show more than one record at a time.
Automatically Generating Columns

The GridView provides a DataSource property for the data object you want to display, much like the list controls you saw in Chapter 15. Once you’ve set the DataSource property, you call the DataBind() method to perform the data binding and display each record in the GridView. However, the GridView doesn’t provide properties, such as DataTextField and DataValueField, that allow you to choose what column you want to display. That’s because the GridView automatically generates a column for every field, as long as the AutoGenerateColumns property is True (which is the default).

Here’s all you need to create a basic grid with one column for each field:

```xml
<asp:GridView ID="GridView1" runat="server" />
```

Once you’ve added this GridView tag to your page, you can fill it with data. Here’s an example that performs a query using the ADO.NET objects and binds the retrieved DataSet:

```vbscript
Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load

    ' Define the ADO.NET objects.
    Dim connectionString As String = 
        WebConfigurationManager.ConnectionStrings("Northwind").ConnectionString
    Dim selectSQL As String = 
        "SELECT ProductID, ProductName, UnitPrice FROM Products"
    Dim con As New SqlConnection(connectionString)
    Dim cmd As New SqlCommand(selectSQL, con)
    Dim adapter As New SqlDataAdapter(cmd)

    ' Fill the DataSet.
    Dim ds As New DataSet()
    adapter.Fill(ds, "Products")

    ' Perform the binding.
    GridView1.DataSource = ds
    GridView1.DataBind()

End Sub
```

Remember, in order for this code to work you must have a connection string named Northwind in the web.config file (just as you did for the examples in the previous two chapters).

Figure 16-1 shows the GridView this code creates.