



INTERNET & WEB
APPLICATION DEVELOPMENT
SWE 444

Fall Semester 2008-2009 (081)

Module 5.4: Data Binding

Dr. El-Sayed El-Alfy

Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa

Objectives/Outline

• Objectives

- Learn how to populate server control with data
- How to access and use external database

• Outline

- How to Populate Server Controls?
- What is data binding?
- Scalar expression
- Simple lists and list binding
- Binding to database sources

How to Populate Server Controls?

- Specify the data in the control's tags
 - Not dynamic: can't get data from a database
- Write code that uses the control's object model
 - okay for populating simple values or lists, but quickly gets too complicated for populating sophisticated displays
- Data binding
 - Better solution than the previous two methods
 - Creates an object that holds the data (DataSet, Array, string, int, etc.)
 - Associate that object with the control

What is Data Binding?

- One of the most useful features of ASP.NET
- Allows programmers to easily display and manipulate data without dealing with the underlying data structure
- Provides a single simple, yet powerful way to populate Web Form controls with data
 - Enables clean separation of control from object holding data
- Supports binding to any data source
 - Properties, expressions, method calls
 - Collections (Array, Hashtable, etc.)
 - DataSet, DataTable, DataView, DataReader
 - XML

What is Data Binding? (cont.)

- Controls that support data binding expose a property named `DataSource` and a method called `DataBind`
- When a page is loaded, the user of the control initializes the `DataSource` property to some collection of data like an array, `DataReader` or `DataSet`
- When the `DataBind` method of the control is called, the expression is evaluated and bound
 - `DataBind` for a single control (and sub-controls)
 - `Page.DataBind` binds all controls on a page
- Works for scalars, e.g. Label control
- Works for lists: `asp:RadioButtonList`, `asp:CheckBoxList`, `asp:DropDownList`, `asp:ListBox`
- Enables the use of templates

Scalar Expressions

- Data binding expression: `<%# expression %>`
- Expression is evaluated when `DataBind()` is called

```
<asp:Label id=label1
  Text=<%# "The result is " + (1 + 2) + ", the time is " +
  DateTime.Now.ToLongTimeString() %> runat="server" />
```

.....

```
public void Page_Load(object s, EventArgs e) {
  if (! Page.IsPostBack)
    Page.DataBind();
}
```

Scalar Expressions: Demo

DataBinding1.aspx

```
<html >
<head>
  <script language="C#" runat="server">
    void Page_Load(Object Src, EventArgs E) {
      if (!Page.IsPostBack) {
        ArrayList values = new ArrayList();
        values.Add (0); values.Add (1); values.Add (2); values.Add (3);
        values.Add (4); values.Add (5); values.Add (6);
        DataList1.DataSource = values; DataList1.DataBind();
      }
    }
    String EvenOrOdd(int number) {
      if ((number % 2) == 0)
        return "Even";
      else
        return "Odd";
    }
  </script>
</head>
<body>
  <h3><font face="Verdana">Data binding to Methods and Expressions</font></h3>
  <form runat="server">
    <asp:DataList id="DataList1" runat="server" BorderColor="black"
      BorderWidth="1" GridLines="Both" CellPadding="3" CellSpacing="0" >
      <ItemTemplate>
        Number Value: <%= Container.DataItem %>
        Even/Odd: <%= EvenOrOdd((int) Container.DataItem) %>
      </ItemTemplate>
    </asp:DataList>
  </form>
</body>
</html >
```

KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development

4.7

Simple Lists

- Data binding to a list creates a user interface element for each item in the list
- Each item contains text (displayed to user) and an optional value (not displayed)
- The simple list controls:
 - <asp:ListBox>
 - Single or multiple select
 - <asp:DropDownList>
 - <asp:RadioButtonList>
 - <asp:CheckBoxList>
- The above controls can be populated with
 - ArrayList
 - Hashtable
 - SortedList
 - XML data
 - Database

The image shows three ASP.NET list controls side-by-side, each displaying a list of movie titles: Titanic, Star Wars, Jurassic Park, Jaws, Ghost, Forrest Gump, Ice Age, Shrek, Independence Day, and The Ring.

- CheckBoxList:** A vertical list of checkboxes next to each movie title.
- DropDownList:** A single dropdown menu with 'Titanic' selected.
- ListBox:** A vertical list box with multiple selection arrows and all movie titles listed.
- RadioButtonList:** A vertical list of radio buttons next to each movie title.

KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development

4.8

Simple Lists (cont.)

➤ Example: without data binding

```
<asp:RadioButtonList id="countryList" runat="server">
  <asp:ListItem value="NG" text="Nigeria" />
  <asp:ListItem value="S" text="Sweden" />
  <asp:ListItem value="F" text="France" />
</asp:RadioButtonList>
```

➤ Steps to bind data to a list control

- Declare the list control, e.g.

```
<asp:RadioButtonList id="countryList" runat="server" />
```
- Optionally set `DataValueField` and `DataTextField`
- Set its `DataSource`
- Call `DataBind()` method

➤ Notes:

- Use `ArrayList` to populate controls with `Text` and `Value` fields to have the same name
- Use `HashTable` or `SortedList` object to add values that are different from `Text`

Examples (DataBinding2.aspx)



```
// Databind the checkbox list
ArrayList sArrayList3 = new ArrayList();
sArrayList3.Add("Red");
sArrayList3.Add("Blue");
sArrayList3.Add("Purple");
CheckBoxList1.DataSource = sArrayList3;
CheckBoxList1.DataBind();
```

```
// Databind the list box
string[] sArray2 = new string[] { "2001", "2000",
    "1999", "1998", "1997", "1996", "1995" };
ListBox1.DataSource = sArray2;
ListBox1.DataBind();
```

```
// Databind the list box
Hashtable ht = new Hashtable();
ht.Add(1, "one");
ht.Add(2, "two");
ht.Add(3, "three");
ht.Add(4, "four");
RadioButtonList1.DataSource = ht;
RadioButtonList1.DataValueField = "Key";
RadioButtonList1.DataTextField = "value";
RadioButtonList1.DataBind();
```

Binding to XML : Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<countries>
<country>
<text>Norway</text>
<value>N</value>
</country>
<country>
<text>Sweden</text>
<value>S</value>
</country>
<country>
<text>France</text>
<value>F</value>
</country>
<country>
<text>Italy</text>
<value>I</value>
</country>
</countries>
```

Binding to XML : Example (cont.)

```
<%@ Import Namespace="System.Data" %>
<script runat="server">
void Page_Load(Object Src, EventArgs E) {
if (!Page.IsPostBack) {
Dataset mycountries = new Dataset();
mycountries.ReadXml (MapPath("countries.xml"));
rb.DataValueField="value"; rb.DataTextField="text"; rb.DataBind()
}

void displayMessage(Object s, EventArgs e){
lbl1.text="Your favorite country is: " + rb.SelectedItem.Text;
}
</script>
<html>
<body>
<form runat="server">
<asp:RadioButtonList id="rb" runat="server"
AutoPostBack="True" onSelectedIndexChanged="displayMessage" />
<p><asp:label id="lbl1" runat="server" /></p>
</form>
</body>
</html>
```

Binding to Database Sources

- We have seen that it is possible to bind to many different types of collections in ASP.NET
- The most common type of binding is to bind to a result set retrieved from a database query
- ADO.NET provides two ways of retrieving result sets from a database:
 - Using the streaming `IDataReader` interface (or `DataReader` class)
 - Using the disconnected `DataSet` class
- Binding to database sources requires working with four main categories of objects:
 - **Connections**
 - Establishes a Web page connection to a specified data source.
 - **Commands**
 - Defines a command, normally an SQL statement, to retrieve or update a set of records in the data source.
 - **DataReader**
 - Reads through a recordset retrieved by the Command object.
 - **DataAdapter**
 - Populates an in-memory data store with records from a data source, and manages updates to the source.

List Binding Examples

```
public void Page_Load(object s, EventArgs e) {
    if (! Page.IsPostBack) {

        // Databind a drop down to a DataReader
        DropDownList1.DataSource = GetSampleDataDR();
        DropDownList1.DataValueField = "CategoryID";
        DropDownList1.DataTextField = "CategoryName";
        DropDownList1.DataBind();

        // Databind a radio button list to a DataView
        RadioButtonList1.DataSource = GetSampleDataDV();
        RadioButtonList1.DataValueField = "CategoryID";
        RadioButtonList1.DataTextField = "CategoryName";
        RadioButtonList1.DataBind();
    }
}
```

List Binding Examples (Cont.)

Binding to a DataReader

```
IDataReader GetSampleDataDR() {  
    SqlConnection cxn =  
        new SqlConnection("Data Source=localhost;Initial  
            Catalog=Northwind;Integrated Security=True");  
  
    SqlCommand cmd = new SqlCommand("select CategoryID,  
        CategoryName from Categories", cxn);  
    cxn.Open();  
  
    SqlDataReader dr =  
        cmd.ExecuteReader(CommandBehavior.CloseConnection);  
  
    return dr;  
}
```

List Binding Examples (Cont.)

Binding to DataView of DataSet

```
DataView GetSampleDataDV() {  
    DataSet ds;  
  
    SqlConnection cxn =  
        new SqlConnection("Data Source=localhost;Initial  
            Catalog=Northwind;Integrated Security=True");  
  
    SqlDataAdapter adp =  
        new SqlDataAdapter("select CategoryID,  
            CategoryName from Categories", cxn);  
  
    ds = new DataSet();  
    adp.Fill(ds, "Categories");  
  
    return ds.Tables["Categories"].DefaultView;  
}
```


DataReader Vs DataSet for Data Binding

<u>DataReader</u>	<u>DataSet</u>
Supports a connected, read-only access style	Supports a disconnected access style
Programmer must explicitly open and close the sqlConnection	SqlDataAdapter automatically handles opening/closing connection
Does not cache query results	Makes a local copy of and manipulates query results, then reconcile with the actual DB later
Preferred for Web applications where short operations (displaying data) are usually performed	Suitable for long-running applications
Focuses on data access and presentation	Focuses on data management
Use when you simply need to read data and immediately bind it to a control	Use when binding one set of data to multiple controls using different DataViews control

Binding to a Database Demo

- DataBinding3.aspx
 - Data binding to a database



Q & A



References

- H. M. Deitel, P. J. Deitel, and A. B. Goldberg, *Internet and World Wide Web How to Program*, 4/e, Pearson Education Inc., 2008.
- Some useful links with examples and other resources:
 - <http://msdn.microsoft.com/net/aspnet/default.asp>
 - The Official Microsoft ASP.NET Site www.asp.net
 - ASP.NET QuickStart Tutorial
 - <http://quickstarts.asp.net/QuickstartV20/aspnet/>
 - <http://samples.gotdotnet.com/quickstart/aspplus/>
 - W3School ASP.NET Tutorial
 - <http://www.w3schools.com/ASPNET/default.asp>
 - ASP.NET at wikipedia <http://en.wikipedia.org/wiki/ASP.NET>
 - <http://www.asptoday.com/>
 - <http://www.aspfree.com/>
 - <http://www.devx.com/dotnet/>
 - <http://msdn.microsoft.com>
 - Several Online Presentations