

ACCP 17.1 – SEMESTER 3

DATABASE HANDLING IN ASP.NET

Session 1 - Database Handling with ASP.NET and
Data Bound Controls

Course Overview



- Introduction to Database Handling with ASP.NET
- Data-Bound Controls
- More on Database Handling
- Data Source Controls in ASP.NET
- Data Caching and Page Caching
- Data Handling with Mobile Applications

Session 1 Objectives



- Describe ADO.NET
- Explain the advantages of ADO.NET
- Explain the different .NET data providers
- Data-Bound Controls
 - ▣ DataList
 - ▣ DataGrid

An Overview of ADO.NET in ASP.NET

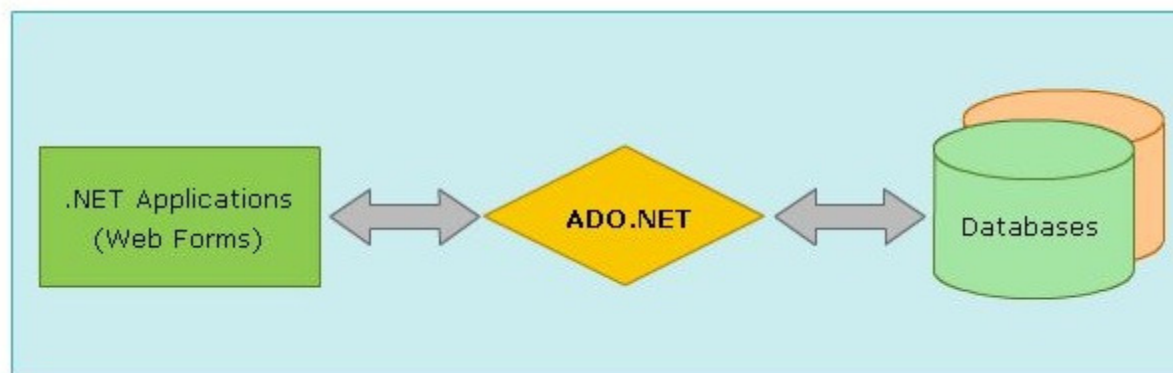
ADO.NET Features

- ❑ Extensive XML support
- ❑ Bridge between applications and databases

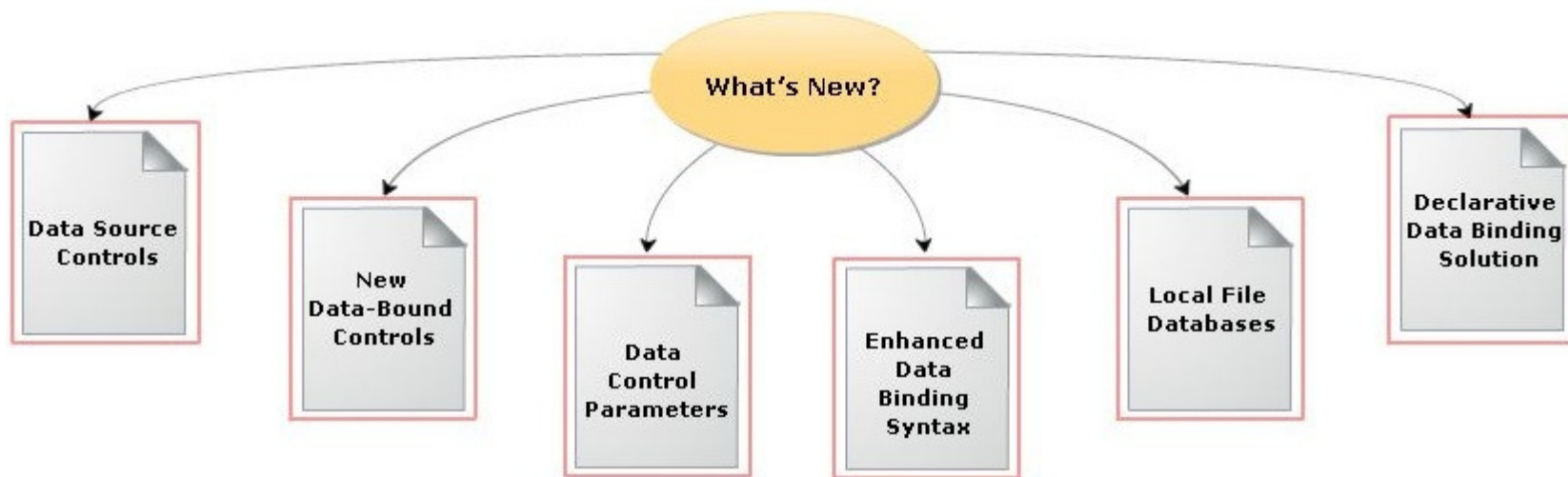
❑ *Dis*
Arc

ADO.NET Advantages

- ❑ Programmability
- ❑ Scalability
- ❑ Interoperability




An Overview of ADO.NET in ASP.NET(continued...)



Connection Strings

- ❑ information in a connection string includes database server, database to be used, authentication information based on the data provider
- ❑ You can either hard-code the connection string within your application or store it in the Web.config file
- ❑ The values stored in the <connectionStrings> element in the Web.config file can be retrieved using the WebConfigurationManager class in the code-behind .cs file

Connection Strings(cond...)



```
// Retrieving connection string from Web.config.
```

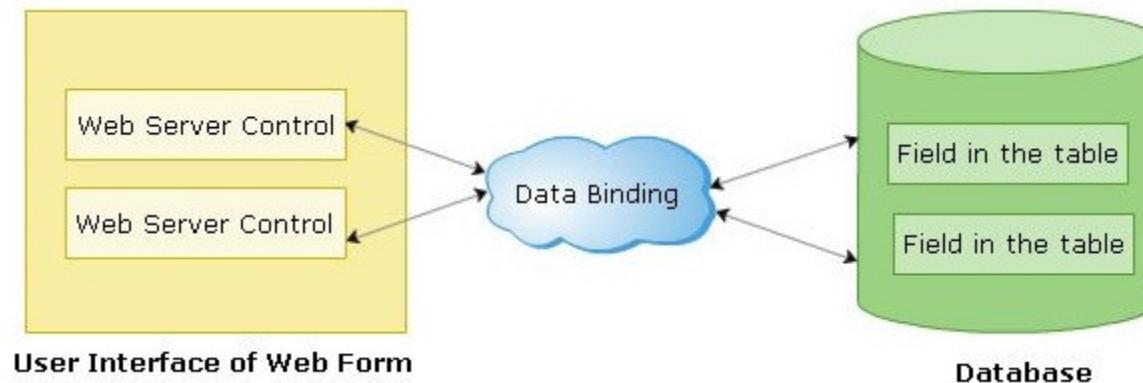
```
String conStr =
```

```
    WebConfigurationHanager.ConnectionStrings  
    ["ConnectionString"].ToString();
```

```
SqlConnection con = new SqlConnection(conString);
```

Data Binding in ASP.NET

- ❑ Data binding = linking data in the UI controls to data in the database
- ❑ The controls that support data binding are called data-bound controls.
- ❑ Data binding minimizes code for displaying data in Web controls.



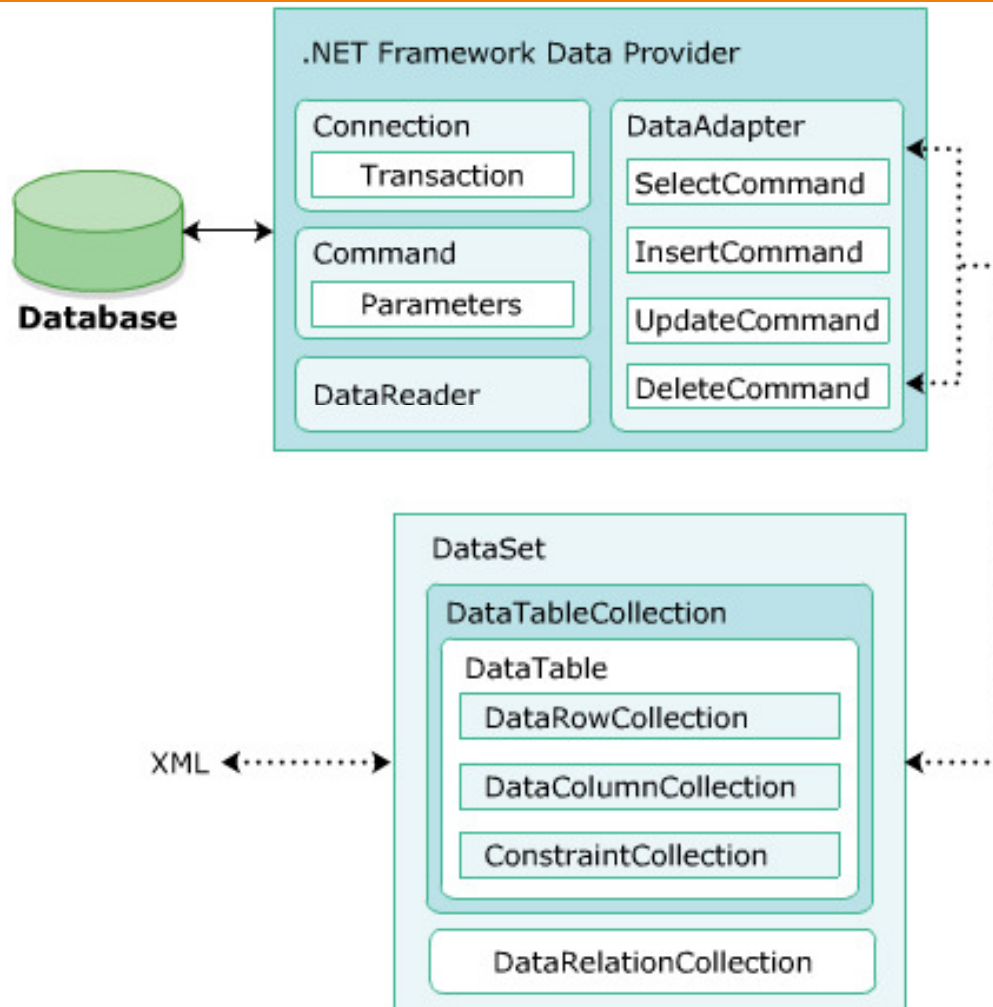
Data Binding Expressions

- ASP.NET data binding syntax: `<%# .%>`
- **Eval()** and **Bind()** : bind the expressions with the controls
- The binding expressions are resolved when the **DataBind()** method is called on a control in the code-behind file
- **GridView**, **FormView**, and **DetailsView** call the **DataBind()** method implicitly to perform binding when they are bound to a data source control

Expression	Description
<code><%# Eval("PropertyName") %></code>	Performs one-way data binding between the field in the table and the property specified.
<code><%# Bind("PropertyName") %></code>	Performs two way data binding between the field in the table and the property specified.
<code>DataRowBinder.Eval(Container.DataItem, "PropertyName") %></code>	Uses the static method of DataRowBinder class to perform Eval operation.

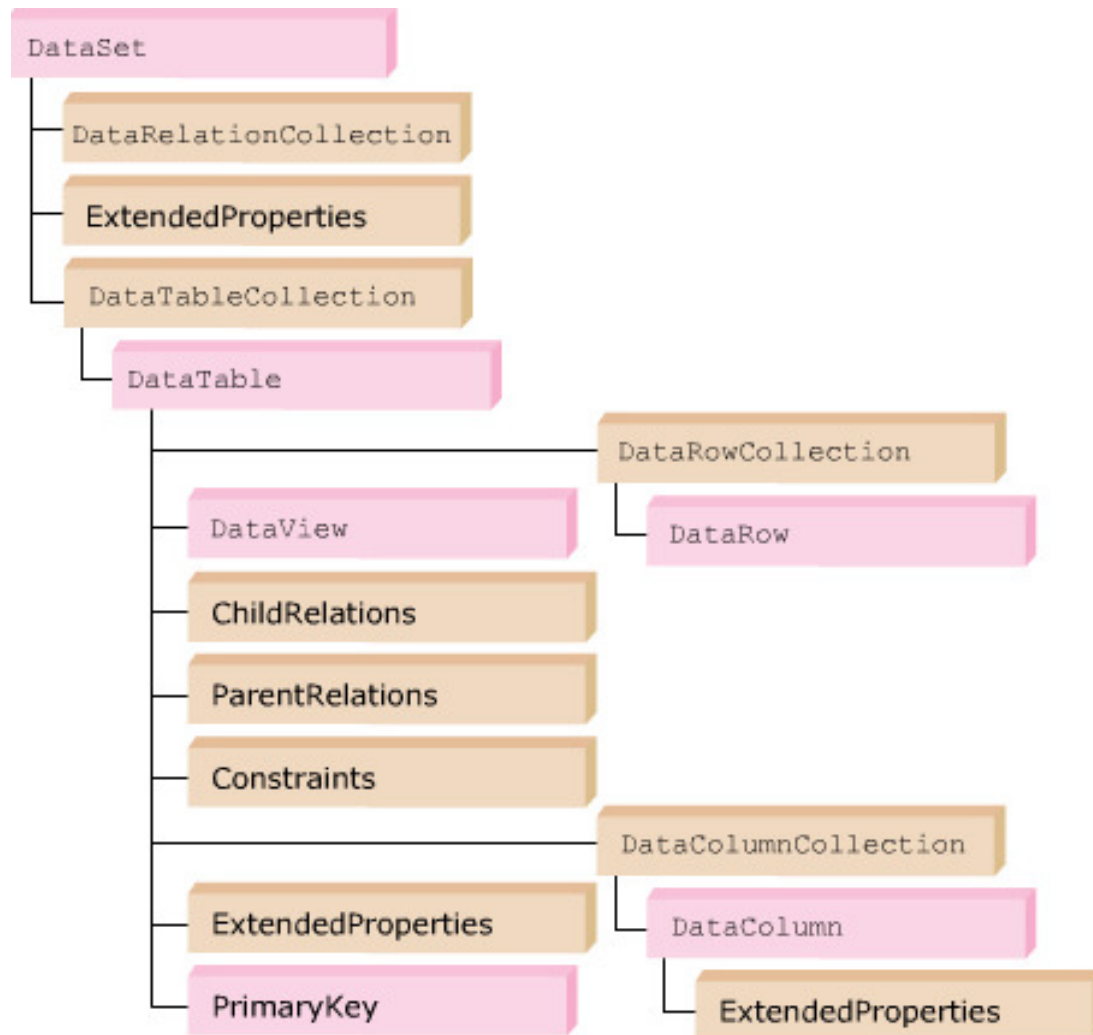
Data Access Components

10



DataSet Architecture

11



TableAdapter

- Using DataSet Designer, VS walks you through the Table Adapter Configuration Wizard to create an instance of TableAdapter
- This feature provides a simple method of binding data to ASP.NET controls

Table Adapter created using Dataset Designer



DataList

- DataList is used to present data in a list using templates
- The templates can be defined for all data items, alternate items, selected item, editable item, and so on
- Most data list controls use at least the ItemTemplate. This is the default template used if no other template is defined.

DataList with
ItemTemplate

Product ID: 8
Name:
<input type="text" value="7"/>
Color:
<input type="text" value="7"/>
Size:
<input type="text" value="7"/>
Price:
<input type="text" value="7"/>
Quantity:
<input type="text" value="7"/>
Update Cancel

DataList with
EditItemTemplate

Product ID: 1
Product ID: 2
Product ID: 3
Product ID: 4
Product ID: 5
Product ID: 6
Product ID: 7
Product ID: 8

Binding DataList Control

- DataList control is a data-bound control.
- **4 steps** for Binding data source to a DataList:
 1. The typed data table instance, dtFootwear, is filled using the typed TableAdapter instance, taFootwear.
 2. The DataList, dlstFootwear is bound to the data table dtFootwear using the DataSource property.
 3. The DataBind() method is invoked on the DataList control to populate the control with data from the data table.
 4. You invoke the user-defined method BindDataList() method in the Page_Load event.

```
public void BindDataList()
{
    // Step 1
    DataSet1 dsetFootwear = new DataSet1();

    DataSet1.FootwearDataTable dtFootwear
        = dsetFootwear.Footwear;

    DataSet1TableAdapters.FootwearTableAdapter taFootwear
        = new DataSet1TableAdapters.FootwearTableAdapter();

    taFootwear.Fill(dtFootwear);

    // Step 2
    dlstFootwear.DataSource = dtFootwear;

    // Step 3
    dlstFootwear.DataBind();
}

protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        // Step 4
        BindDataList();
    }
}
```

DataList Control Properties

Property	Description	Sample Property Values
BackColor	Specifies or retrieves the background color of the control	#DEBA84, Lightgreen
BorderColor	Specifies or retrieves the border color of the control	#DEBA84, blue
BorderStyle	Specifies or retrieves the border style of the control	Solid, Dotted, Dashed, Double, Groove, Ridge, Inset and Outset
BorderWidth	Specifies or retrieves the border width of the control	1px
CellPadding	Specifies or retrieves the amount of space between the cell's content and the cell's border	3
CellSpacing	Specifies or retrieves the space between the cells	2
GridLines	Specifies or retrieves the grid line style of the control when the RepeatLayout property is set to Table	None, Horizontal, Vertical, Both

Content Repetition Properties

- DataList control is a repeater control
- The repetition of DataList layout is controlled by three properties:
 - ▣ RepeatLayout
 - ▣ RepeatDirection
 - ▣ RepeatColumns

Product ID: 1
Product ID: 2
Product ID: 3
Product ID: 4

RepeatLayout=Flow

Product ID: 1
Product ID: 2
Product ID: 3
Product ID: 4

RepeatLayout=Table

Product ID: 1	Product ID: 3
Product ID: 2	Product ID: 4

**RepeatLayout=Table
RepeatColumns=2**

Product ID: 1	Product ID: 2
Product ID: 3	Product ID: 4

**RepeatLayout=Table
RepeatColumns=2
RepeatDirection=Horizontal**

GridView

- Server-side control
GridView is used to display data in tabular format
- GridView control support data source manipulation:
 - ▣ editing
 - ▣ deleting
 - ▣ updating
 - ▣ adding
- GridView supports paging and sorting

Columns with Sorting Support

	<u>ID</u>	<u>Name</u>	<u>Size</u>	<u>Color</u>	<u>Price</u>	<u>Quantity</u>
Edit Delete	1	Reebok 132057 NFL Thorpe Mid D Detachable Adult Football Cleats	6 1/2 - 13, 14	Black	35.99	100
Edit Delete	2	Reebok 136786 NFL 4-Speed II Mid 3/4 Detachable Adult Football Cleats	9-13	Black	41.99	100
Edit Delete	3	Reebok 131930 NFL Thorpe Mid MR7 Youth Football Cleats	1-6	White	38.55	50
Edit Delete	4	Reebok 136811 NFL Thorpe Mid Turf Rat Adult Football Cleats	6+-16	White	39.95	100
Next Last						

Navigation Buttons

Binding GridView control

□ **4 Steps** for binding GridView control to a data source:

1. First, you create an instance of SqlDataAdapter class to fetch from and send data to the Footwear table in pubs database on SQL Server 2005. The adapter instance uses the sqlconPubs instance to establish a connection.
2. You create an instance named dsetFootwear of untyped dataset and fill it with data from Footwear table.
3. You set the DataSource property of GridView control, qvwFootwear, to the Footwear table instance retrieved from the Tables collection of dataset dsetFootwear.
4. You populate the GridView control, qvwFootwear, with the data from the Footwear table.

```
protected void Page_Load(object sender, EventArgs e)
{
    // Step 1
    sqlconPubs = new SqlConnection("Data Source=10.2.1.51;
    Initial
    Catalog=pubs;User ID=sa;Password=playware");
    sdaFootwear = new SqlDataAdapter("select * from Footwear",
    sqlconPubs);

    // Step 2
    dsetFootwear = new DataSet();
    sdaFootwear.Fill(dsetFootwear, "Footwear");

    if (!Page.IsPostBack)
    {
        // Step 3
        qvwFootwear.DataSource = dsetFootwear.Tables["Footwear"];

        // Step 4
        qvwFootwear.DataBind();
    }
}
```

Image 1

ID	Name	Size	Color	Price	Quantity
1	Reebok 132057 NFL Thorpe Mid D Detachable Adult Football Cleats	6 1/2 - 13, 14	Black	35.99	100

Image 2

Paging GridView Properties

- ❑ DataList control provides several properties to change the default appearance of a DataList control:

Property	Description	Sample Property Values
AllowPaging	Specifies or retrieves whether paging is enabled or not	True, False
PagerSettings.FirstPageText	Specifies or retrieves the text to be displayed for the first page button	First, 1
PagerSettings.NextPageText	Specifies or retrieves the text for the next page button	Next
PagerSettings.PreviousPageText	Specifies or retrieves the text for the previous page button	Previous
PagerSettings.LastPageText	Specifies or retrieves the text for the last page button	Last
PagerSettings.Visible	Specifies or retrieves whether the paging buttons are displayed	True, False
PagerSettings.Position	Specifies or retrieves the position of paging buttons for navigating pages	Bottom, Top, TopAndBottom
PagerSettings.Mode	Specifies or retrieves the mode in which to display the paging buttons	Numeric, NumericFirstLast, NextPrevious, NextPreviousFirstLast
PageIndex	Specifies or retrieves the index of currently displayed page	0, 1

Sorting GridViews

- ❑ Sorting is enabled by setting the AllowSorting property to true.
- ❑ Image 2 shows the step-by-step code of Sorting event handler:
 1. You set the SelectCommand property of SqlDataAdapter instance, sdaFootwear, to retrieve records from Footwear table.
 2. You clear the dataset and fill it with the sorted rows.
 3. You re-populate the GridView control, gvwFootwear, with sorted rows.

ID	Name	Size	Color	Price	Quantity
1	Reebok 132057 NFL Thorpe Mid D Detachable Adult Football Cleats	6 1/2 - 13, 14	Black	35.99	100

Image 1

```
protected void gvwFootwear_Sorting(object sender,
GridViewSortEventArgs e)
{
    // Step 1
    sdaFootwear.SelectCommand = new SqlCommand("select
* from Footwear order by " + e.SortExpression,
sqlconPubs);
    // Step 2
    dsetFootwear.Tables["Footwear"].Clear();
    sdaFootwear.Fill(dsetFootwear, "Footwear");
    // Step 3
    gvwFootwear.DataSource =
dsetFootwear.Tables["Footwear"];
    gvwFootwear.DataBind();
}
```

Image 2

Summary and Lab activity notes

