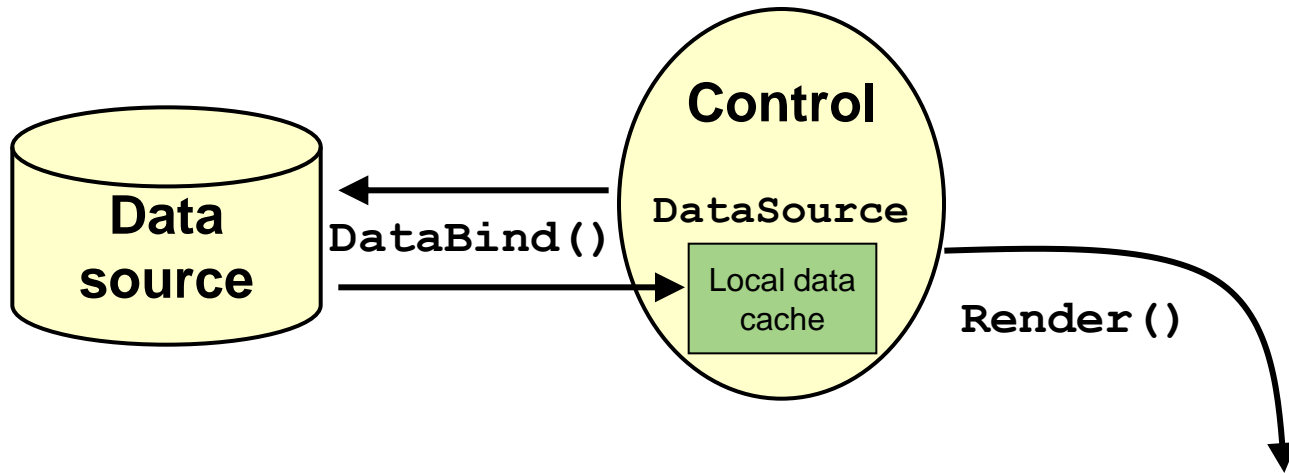# Data Binding I

Outline

- **Databinding in ASP.NET**
  - Imperative data binding
  - Controls
- **Page-level binding**
- **Binding text and values**
- **Declarative data sources**
- **SqlDataSource**
- **Connection strings**
- **GridView/DetailsView**
- **Data binding templates**
  - Repeater
  - DataList

pluralsight
see what you can learn

Databinding in ASP.NET

- **Data binding is the process of populating a control with data from a data source**
  - Controls that support data binding expose
    - a property called `DataSource`
    - a method called `DataBind()`
  - To bind data to a control imperatively, initialize its `DataSource` to a data source and then invoke its `DataBind()` method

Data binding process

Data binding example

```
<%@Page Language="C#" %>

<html>
<script language="C#" runat="server">
protected void Page_Load(Object src, EventArgs e)
{
  string dsn = "server=.;trusted_connection=yes;database=pubs";
  using (IDbConnection conn = new SqlConnection(dsn))
  {
    IDbCommand cmd = conn.CreateCommand();
    cmd.CommandText = "SELECT * FROM Authors";
    conn.Open();
    IDataReader reader = cmd.ExecuteReader();
    gv1.DataSource = reader;
    gv1.DataBind();
  }
}
</script>
<body>
<form runat=server>
  <asp:GridView id="gv1" runat="server" />
</form>
</body></html>
```

Controls that support Data Binding

- **Several controls in ASP.NET support data binding**
  - □ Can be bound to any data source supporting `IEnumerable` as well as `DataTable` and `DataSet` classes

Databinding Controls

```
HtmlSelect
CheckBoxList
BulletedList
GridView
DetailsView
FormView
ListView
DataGrid
DataList
Repeater
DropDownList
ComboBox
ListBox
RadioButtonList
```

Bind to →

```
Array
ArrayList
HashTable
Queue
SortedList
Stack
StringCollection
DataView
DataTable
DataSet
IDataReader
IEnumerable
```
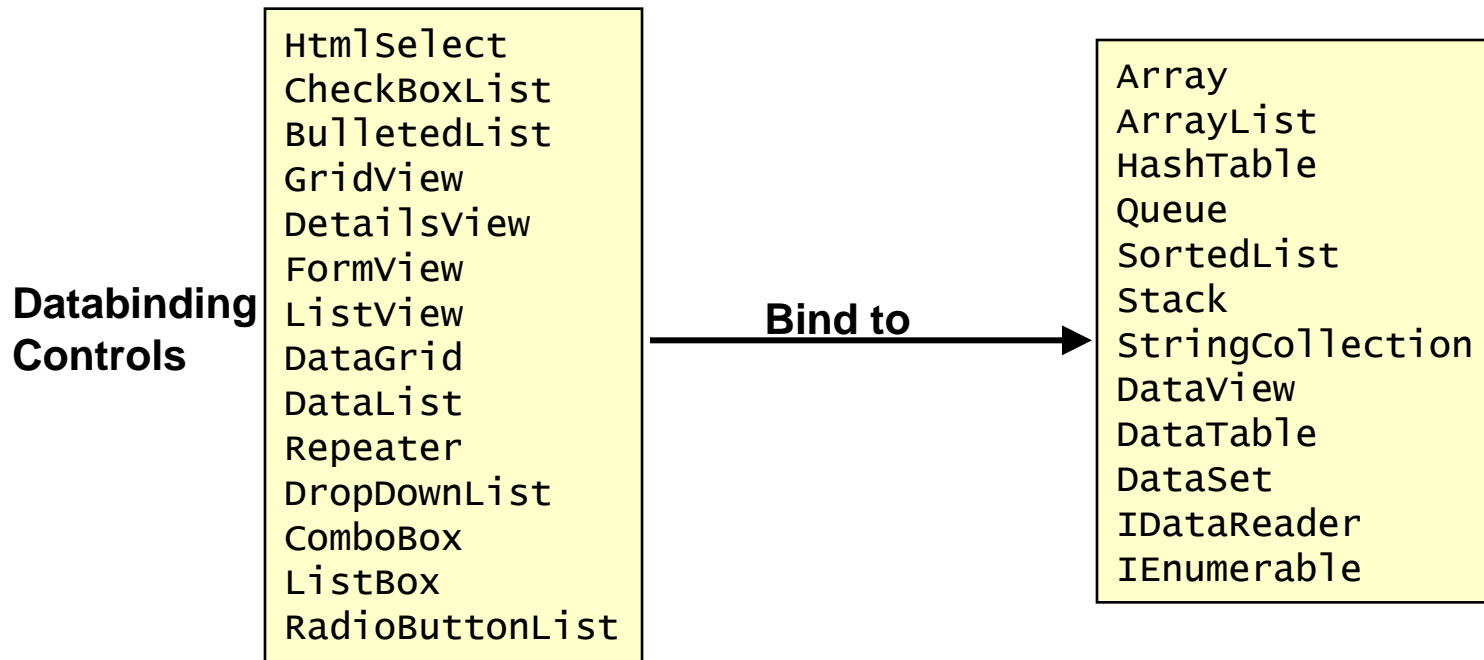
Page-level binding

```
<%<%@ Page Language="C#" %>
<html><body><head>
<script language="C#" runat="server">
  void Page_Load(Object sender, EventArgs e) {
    if (!Page.IsPostBack) {
      string[] vals = new string[] {"v1", "v2", "v3", "v4"};

      s1.DataSource  = vals;
      cbl1.DataSource= vals;
      dd1.DataSource = vals;
      lb1.DataSource = vals;
      rbl1.DataSource= vals;
      DataBind();
    }
  }
</script></head>
<form runat="server">
  <Select id="s1" runat="server" /> <br/>
  <asp:DropDownList id="dd1" runat="server" /> <br/>
  <asp:CheckBoxList id="cbl1" runat="server" /> <br/>
  <asp:ListBox id="lb1" runat="server" /> <br/>
  <asp:RadioButtonList id="rbl1" runat="server" /> <br/>
</form></body> </html>
```
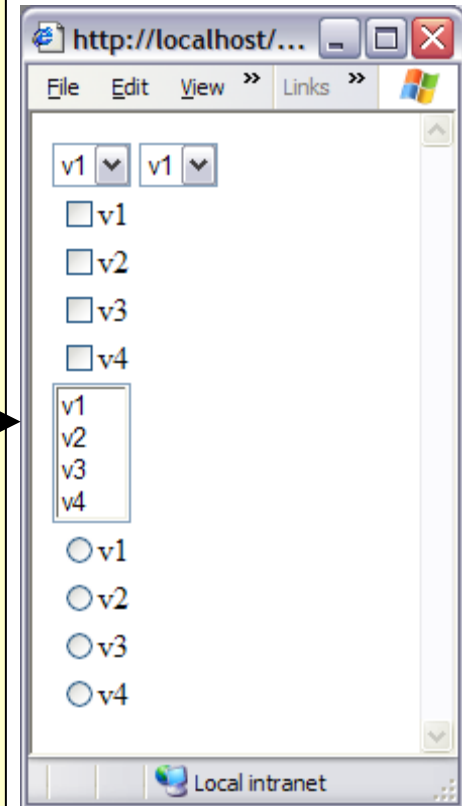
Binding Text and Values

- **For single-dimensional controls (ListBox, DropDownList, etc.) bound to rectangular data sources**
  - Initialize `DataTextField` to the field in the query you want displayed
  - Initialize `DataValueField` to the field in the query you want associated with the value of an item

```csharp
string dsn = "server=.;trusted_connection=yes;database=pubs";
string sql = "SELECT au_id, au_lname FROM Authors";

using (SqlConnection conn = new SqlConnection(dsn))
using (SqlCommand cmd = new SqlCommand(sql, conn))
{
  conn.Open();

  dd1.DataSource = cmd.ExecuteReader();
  dd1.DataTextField = "au_lname";
  dd1.DataValueField = "au_id";
  dd1.DataBind();
}
```

Declarative data sources

- **ASP.NET 2.0 introduced the declarative data source**
  - SqlDataSource and friends enable much richer designer support for data-bound controls
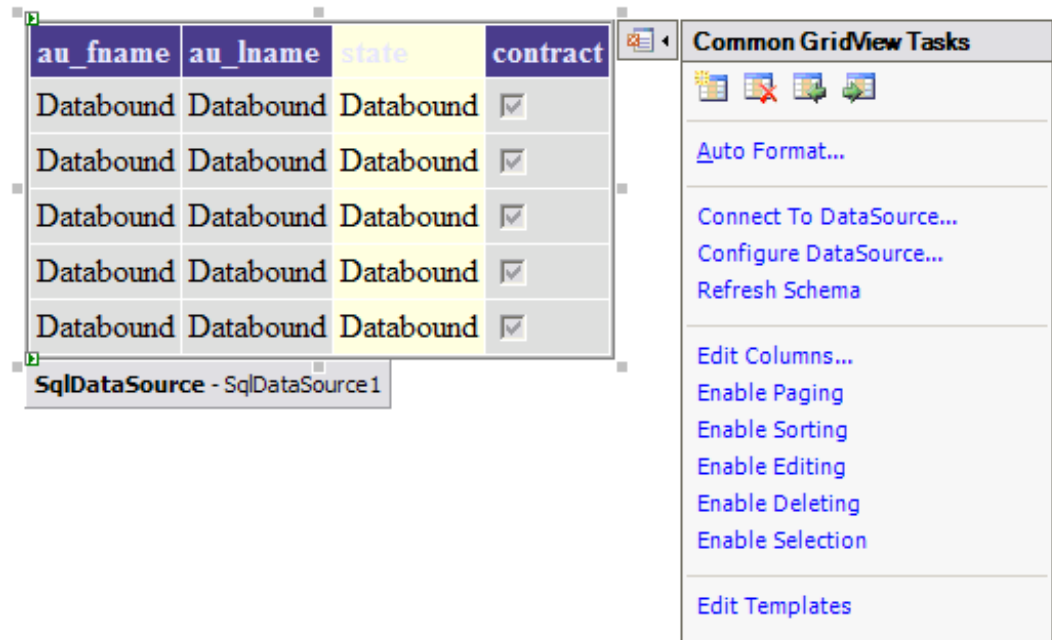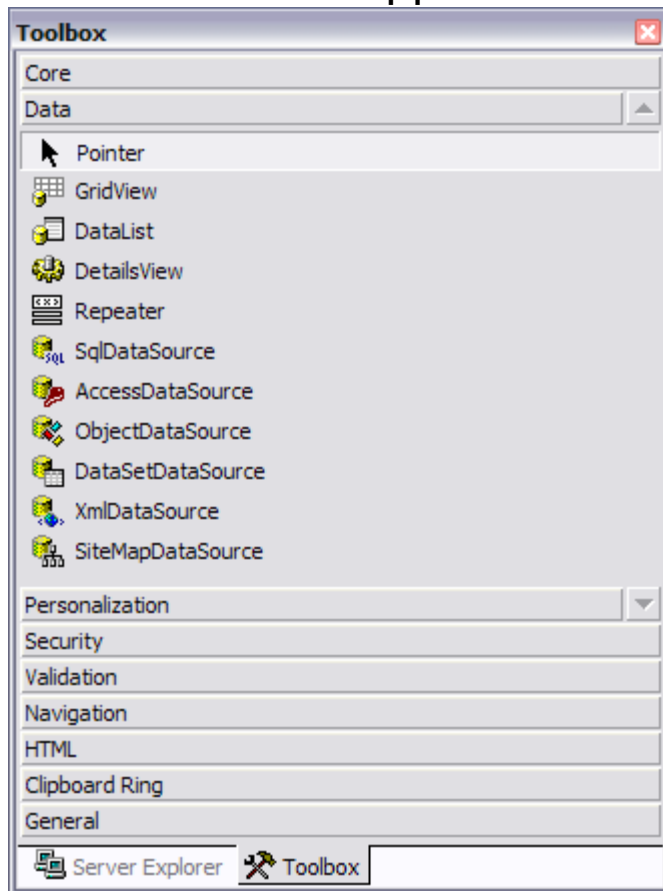
Binding to declarative data sources

- **Data binding can now be accomplished with no code**
  - Extensive designer support
  - Integration with server explorer
  - Standard binding model easily customized

```
<asp:GridView ID="myGridView" runat="Server"
                DataSourceId="pubsDs" />
<asp:SqlDataSource ID="pubsDs" runat="server"
    ConnectionString="server=.;database=pubs;trusted_connection=yes"
    SelectCommand="SELECT * FROM Authors" />
```

How declarative data binding works

Request

GridView

CreateChildControls()
EnsureDataBound()
DataBind()

IDataSource

SqlDataSource

database

Select

local cache of data

Render()

data
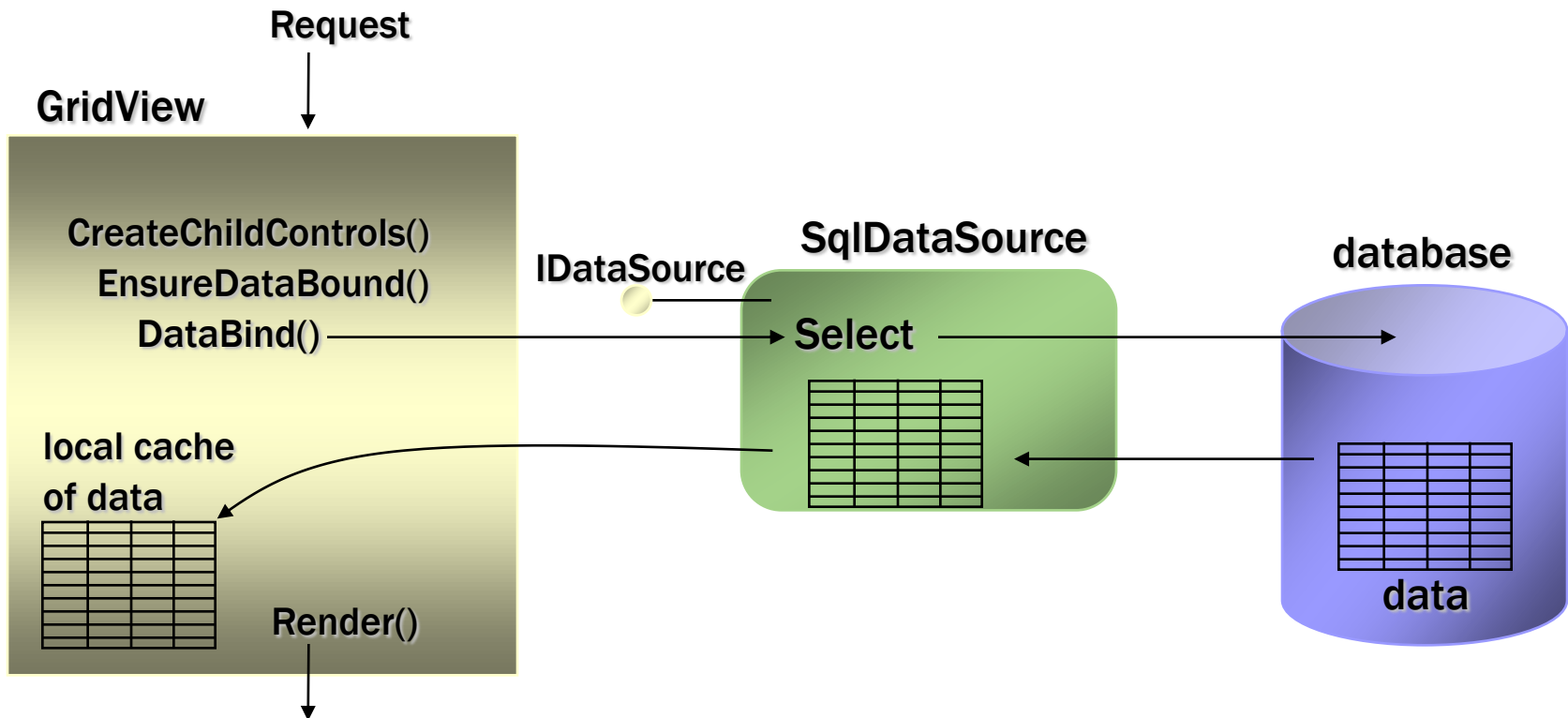
title	filed_by	assigned_to	severity	status	date_filed	
Reactor controller times out	Author Pewty	Homer Simpson	Cosmetic	Open	4/25/2005	Details
Icon logo wrong color	Pointy-Haired Boss	Albert Einstein	Disastrous	Open	4/26/2005	Details
Not cat-compliant	Pointy-Haired Boss	Henry Kissenger	Disastrous	Closed	4/5/2005	Details
Needs dangly string	Boss's Cat	Albert Einstein	Embarrassing	Open	4/26/2005	Details
Anon bug # 0	Anonymous Coward	<unassigned>	Pretty serious	Open	3/12/2005	Details
Anon bug # 1	Anonymous Coward	<unassigned>	Pretty serious	Open	3/12/2005	Details
Anon bug # 2	Anonymous Coward	<unassigned>	Pretty serious	Open	3/12/2005	Details
Anon bug # 3	Anonymous Coward	<unassigned>	Pretty serious	Open	3/12/2005	Details
Anon bug # 4	Anonymous Coward	<unassigned>	Pretty serious	Open	3/12/2005	Details
Anon bug # 5	Anonymous Coward	<unassigned>	Pretty serious	Open	3/12/2005	Details

1 2 3 4 5 6 7 8 9 10 ...

Data binding

- **With data source controls, you no longer explicitly bind data to a control**
  - Occurs implicitly now when child controls are created

```
public abstract class CompositeDataBoundControl : DataBoundControl {
  protected internal override void CreateChildControls() {
    //...
    base.EnsureDataBound();
    //...
  }
}
```

```
public abstract class BaseDataBoundControl : WebControl {
  protected void EnsureDataBound()
  {
    if (this.RequiresDataBinding && (this.DataSourceID.Length > 0))
      this.DataBind();
  }
}
```

SqlDataSource

- **The SqlDataSource is defined for all SQL-based data retrieval / manipulation**
  - ProviderName specifies SQL provider to use (defaults to SqlClient)
  - Supports DataReader and DataSet based retrieval (with caching)
  - Select, Insert, Update, Delete commands each exposed as properties with associated parameter collections
  - FilterExpression with parameters for adding WHERE constraints
  - SortParameterName for adding ORDER BY sort

SqlDataSource control

```csharp
public class SqlDataSource : DataSourceControl
{
  public virtual string ProviderName { get; set; }

  public virtual string ConnectionString { get; set; }
  public SqlDataSourceMode DataSourceMode { get; set; }

  public string SelectCommand { get; set; }
  public ParameterCollection SelectParameters { get; }
  public string InsertCommand { get; set; }
  public ParameterCollection InsertParameters { get; }
  public string UpdateCommand { get; set; }
  public ParameterCollection UpdateParameters { get; }
  public string DeleteCommand { get; set; }
  public ParameterCollection DeleteParameters { get; }

  public string FilterExpression { get; set; }
  public ParameterCollection FilterParameters { get; }
  public string SortParameterName { get; set; }
  // ...
}
```

SqlDataSource example

```
<asp:SqlDataSource ID="patientDataSource" Runat="server"
  UpdateCommand= "UPDATE patients SET firstname = @firstname ...
                  WHERE patientid = @original_patientid"
  InsertCommand="INSERT INTO patients (firstname, ...)
                  VALUES (@firstname ...)"
  DeleteCommand="DELETE FROM patients WHERE patientid =
                  @original_patientid"
  SelectCommand="SELECT patientid, firstname... FROM patients"
  ConnectionString="...">
  <DeleteParameters>
    <asp:Parameter Type="Int32" Name="patientid"></asp:Parameter>
  </DeleteParameters>
  <UpdateParameters>
    <asp:Parameter Type="String" Name="firstname"></asp:Parameter>
  </UpdateParameters>
  <InsertParameters>
    <asp:Parameter Type="String" Name="firstname"></asp:Parameter>
  </InsertParameters>
</asp:SqlDataSource>
```

Connection string storage

- **Instead of hard-coding connection strings in data source controls prefer to retrieve them from your config file**
  - New ConnectionStrings section
  - New support for encrypting portions of config files
  - New <%$ %> syntax for retrieving config file values

web.config

```
<configuration>
      <appSettings/>
      <connectionStrings>
   <add name="pubs_dsn" connectionString="server=.;..."
         providerName="System.Data.SqlClient"/>
 </connectionStrings>
</configuration>
```

somepage.aspx

```
<asp:SqlDataSource
     ConnectionString="<%$ ConnectionStrings:pubs_dsn %>"
...
```

GridView

- **Renders tabular grid of data**
  - Implicit or explicit column generation
  - Special fields for image/checkbox/hyperlink
    - ImageField
    - CheckBoxField
    - HyperLinkField
  - Intelligent interaction with data source controls
    - Automatic pagination, editing, deleting, and sorting

GridView sample

```
<asp:SqlDataSource ID="authorsDataSource" Runat="server"
       SelectCommand="SELECT ..." InsertCommand="INSERT ..."
       SelectCommand="UPDATE ..." SelectCommand="DELETE ..."
       ConnectionString="<%$ ConnectionStrings:pubs_dsn %>" />
<asp:GridView ID="authorsGridView" Runat="server"
              DataSourceID="authorsDataSource"
              EmptyDataText="There are no data records to display."
              AllowPaging="True" AllowSorting="True" />
```

au_id	au_lname	au_fname	phone	address	city	state	zip	contract
172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025	☐
213-46-8915	Blue	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618	☑
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	☑
267-41-2394	O'Leary	Michael	408 286-2428	22 Cleveland Av. #14	San Jose	CA	95128	☑
274-80-9391	Straight	Dean	415 834-2919	5420 College Av.	Oakland	CA	94609	☑
341-22-1782	Smith	Meander	913 843-0462	10 Mississippi Dr.	Lawrence	KS	66044	☐
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705	☑
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	☑
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428	☑
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av.	San Francisco	CA	94130	☑

1 2 3

DetailsView

- **Supports single-row display/editing**
  - Useful for master/detail scenarios
 - Supports paging, editing, deleting, and *inserting*!

```
<asp:SqlDataSource ID="authorsDataSource" Runat="server"
        SelectCommand="SELECT ..." InsertCommand="INSERT ..."
        SelectCommand="UPDATE ..." SelectCommand="DELETE ..."
        ConnectionString="<%$ ConnectionStrings:pubs_dsn %>" />
<asp:DetailsView ID="authorDetailView" Runat="server"
                DataSourceID="authorsDataSource"
                DefaultMode="edit" />
```

au_id	172-32-1176
au_lname	White
au_fname	Johnson
phone	408 496-7223
address	10932 Bigge Rd.
city	Menlo Park
state	CA
zip	94025
contract	☐

Templates

- **The `GridView, DetailsView, FormView, ListView, Repeater,` and `DataList` controls provide complete appearance customization through templates**
  - Provide a shell for displaying data
  - Let you specify exactly how that data should be displayed using a template
  - Handle layout and data binding for you
  - `Eval` syntax used to extract data from current row

Binding Model for Templated Controls
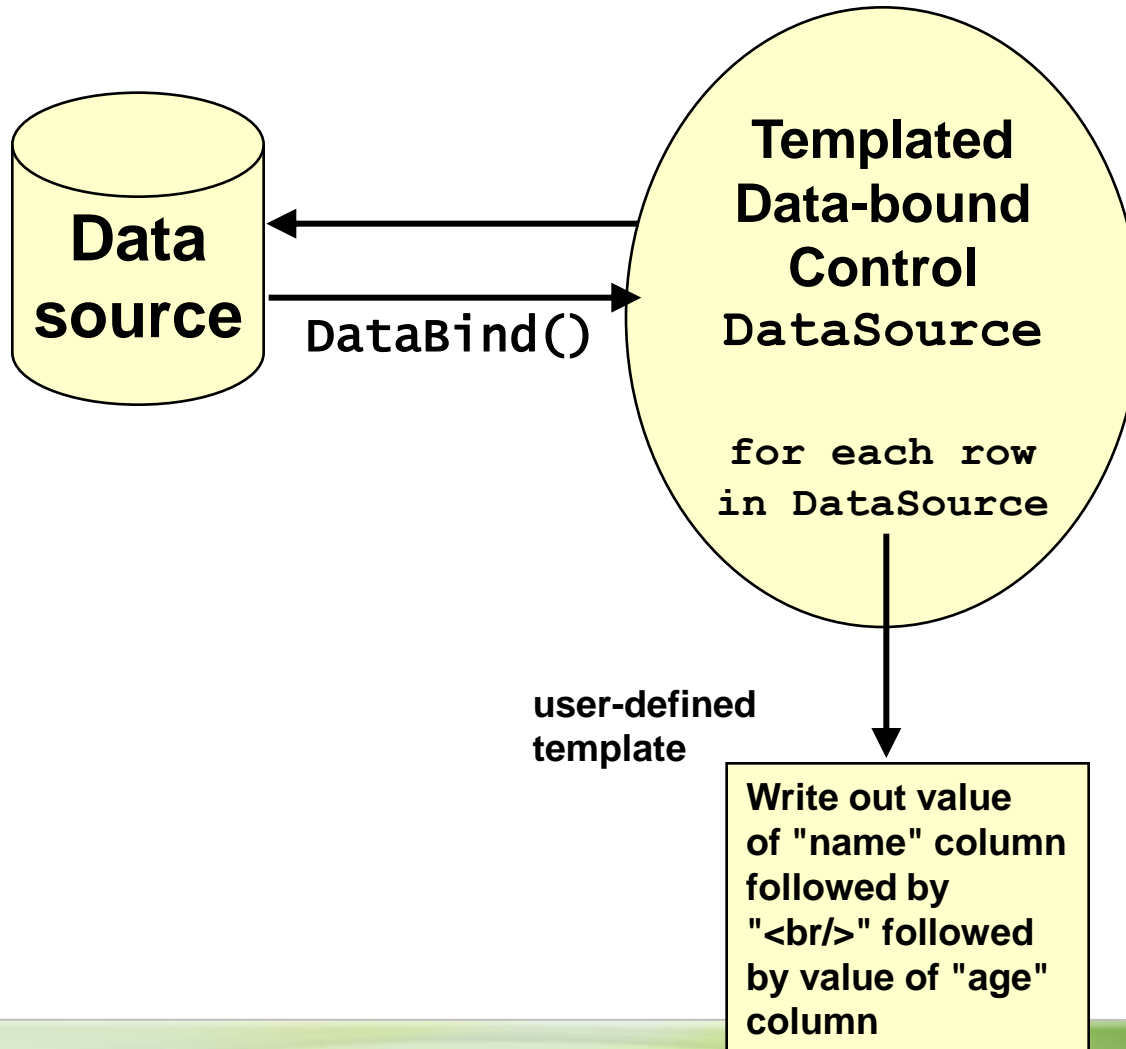
Databinding evaluation syntax

- **Templated controls need a way to specify access to the currently bound row of the data source**
  - Special syntax introduced
    ```
    <%# expression %>
    ```
  - Evaluated when the DataBinding event of the containing control is issued (which will be once per row of a data source for data bound controls)
  - `Eval` syntax used to extract column values

Repeater example

```
<asp:Repeater runat="server" ID="rp1" DataSourceID="dogsDataSource">
  <ItemTemplate>
    <%# Eval("Name") %> is a <%# Eval("Breed") %>
    and looks like this:
    <asp:Image runat="server" ID="dogImage"
              ImageUrl='<%# Eval("Image") %>' />
  </ItemTemplate>
</asp:Repeater>
```

Fred is a Dalmation and looks like this:

Fido is a Labrador and looks like this:

Rex is a Golden Retriever and looks like this:

Dal is a Mutt and looks like this:

Joe is a Black Labrador Retriever and looks like this:

DataList

- **The `DataList` control displays data items in a repeating list**
  - The `DataList` is useful for laying out data items in a list-like format
  - Renders each row in the data source to a single table cell
  - Requires that you define at least an `ItemTemplate` describing how to render each item

DataList example

```
<asp:DataList runat="server" ID="dl1" RepeatColumns="2"
              RepeatDirection="Horizontal" DataSourceID="dogsDataSource">
  <ItemTemplate>
    <div style="padding:15,15,15,15">
      <asp:Image runat="server" ImageUrl='<%# Eval("Image")%>'/>
      <br />
      <b>Name: </b><%# Eval("Name") %><br />
      <b>Breed: </b><%# Eval("Breed")%>
    </div>
  </ItemTemplate>
</asp:DataList>
```


Name: Fred
Breed: Dalmation

Name: Fido
Breed: Labrador

Name: Rex
Breed: Golden Retriever

Name: Dal
Breed: Mutt

Name: Joe
Breed: Black Labrador Retriever

Summary

- **Data binding**
  - Mapping data to controls
- **Declarative data sources**
  - Reduce code, move data access code into framework
- **SqlDataSource**
  - Powerful abstraction over ADO.NET queries
- **Connection strings**
  - Storage location in web.config
- **Data bound controls**
  - GridView, DetailsView, Repeater, DataList