



asp:EntityDataSource#EntityDataSource...

EntityDataSource - EntityDataSourceCustomers

EntityDataSource Tasks

Configure Data Source...

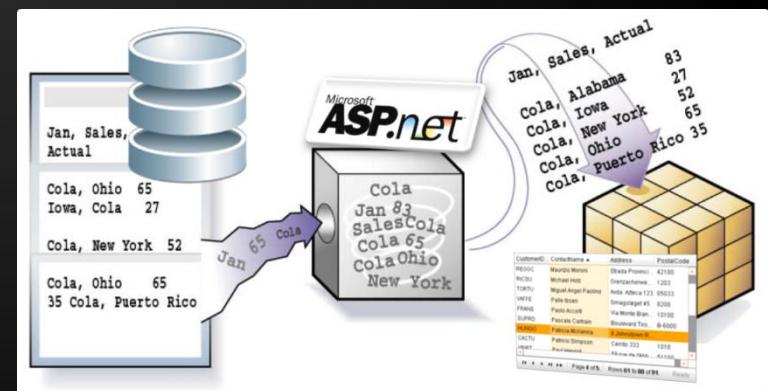
Refresh Schema

ASP.NET

Data Source Controls

Building Data-Driven ASP.NET Web Forms Apps

ASP.NET Web Forms
Telerik Software Academy
<http://academy.telerik.com>



1. ASP.NET Data Source Controls

- **SqlDataSource**
- **EntityDataSource**
- **ObjectDataSource**

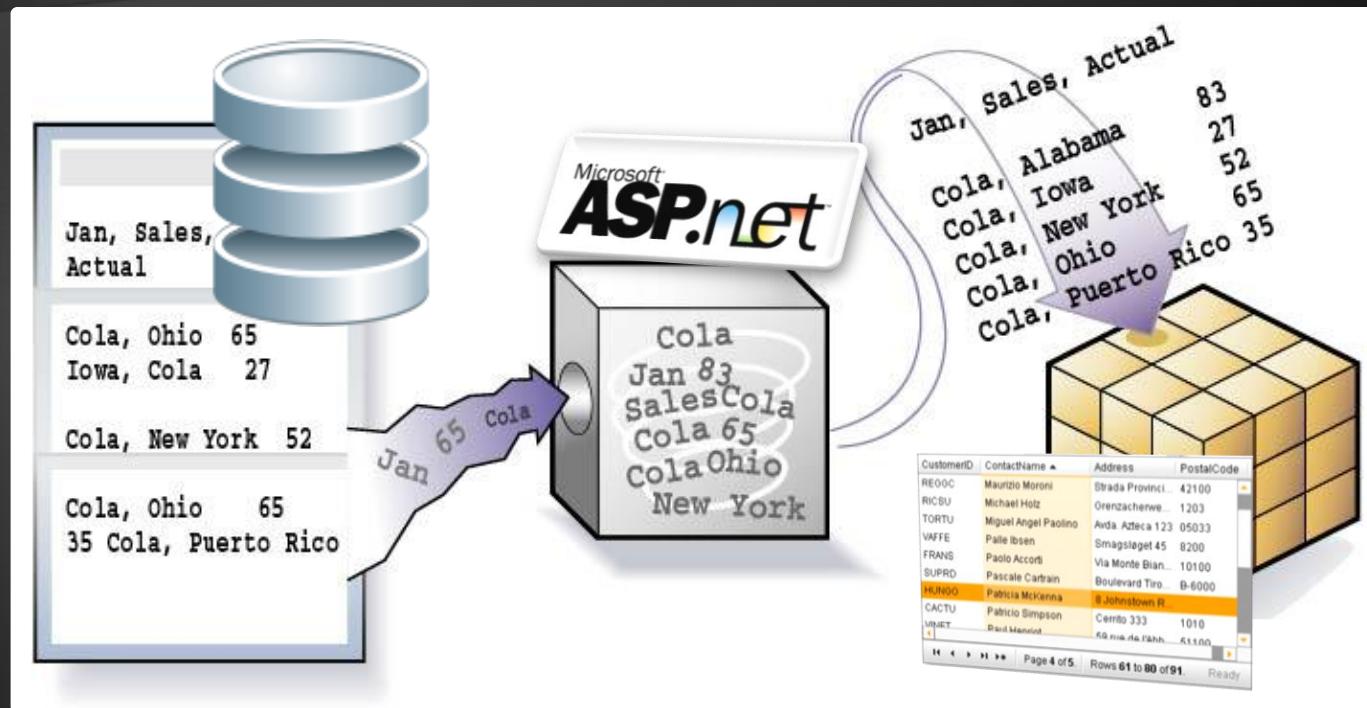
asp:EntityDataSource#EntityDataSource

EntityDataSource - EntityDataSourceCustomers

2. Accessing ADO.NET Entity Framework with EntityDataSource

- **Working with Editable Controls**
- **Master-Details Navigation**

3. Model Binding in Web Forms

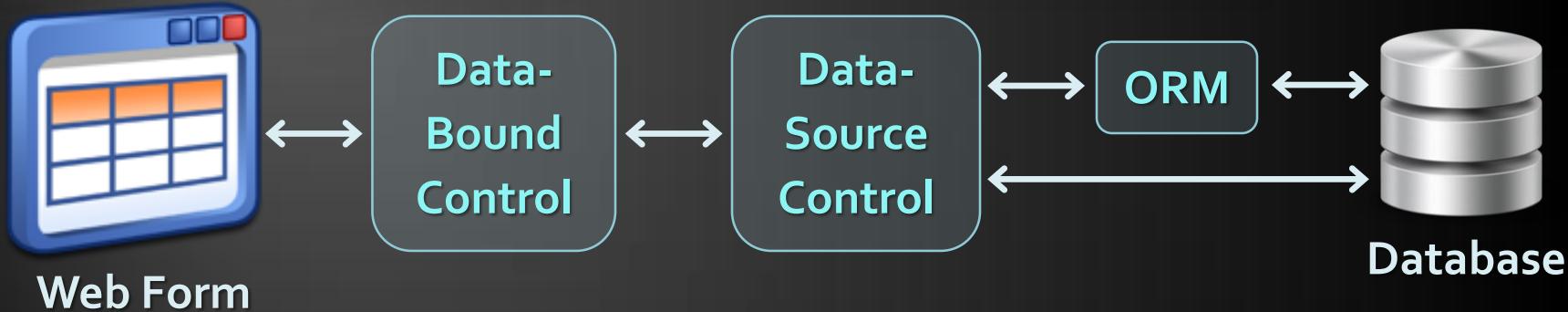


ASP.NET Data Source Controls

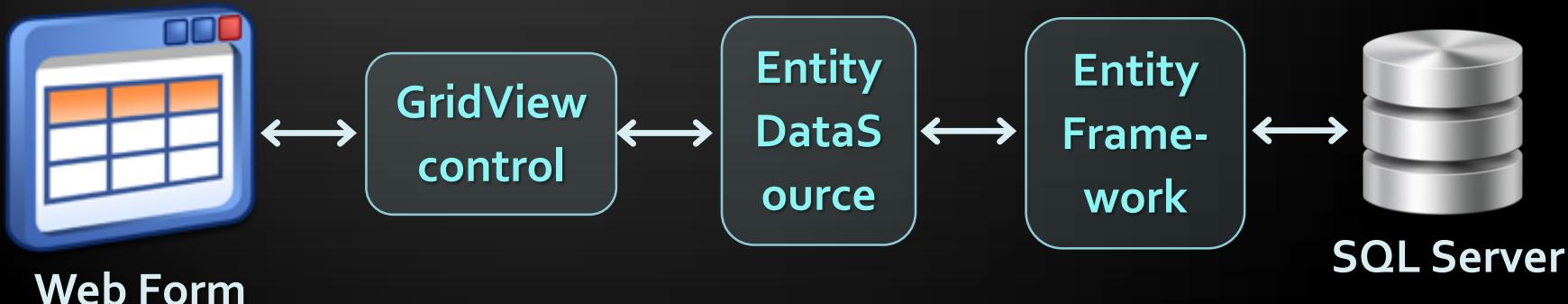
- ◆ ASP.NET provides server controls that take care of data binding details
 - ◆ Known as data source controls
 - ◆ **SqlDataSource, EntityDataSource, ObjectDataSource, XmlDataSource, ...**
- ◆ They are an abstraction over the data source
- ◆ Data-bound server controls can be associated to a data source control
 - ◆ Through the **DataSourceID** property

ASP.NET Data Source Controls

- ◆ ASP.NET data source controls connect the data-bound controls with the database / ORM:



- ◆ Typical scenario:



SqlDataSource

- ◆ **SqlDataSource provides connection to a relational DB (SQL Server, Oracle, OLE DB, ...)**
- ◆ **Data is manipulated by using commands**
 - ◆ **Select, Update, Insert and Delete commands**
 - ◆ **Commands can be either SQL queries or names of stored procedures**
- ◆ **Data is processed with a DataSet (by default)**
 - ◆ **The DataSourceMode property specifies whether to use DataSet or DataReader**
- ◆ **Old-fashioned, better use EntityDataSource**

SqlDataSource – Example

```
<asp:SqlDataSource ID="DSCategories" runat="server"  
ConnectionString="<%$ ConnectionStrings:Northwind %>"  
SelectCommand="SELECT * FROM [Categories]" />  
  
<asp:GridView ID="GridViewCategories" runat="server"  
DataSourceID="DSCategories" DataKeyNames="CategoryID"  
AutoGenerateColumns="True" AllowPaging="True"  
AllowSorting="True">  
</asp:GridView>
```

<u>CategoryID</u>	<u>CategoryName</u>	<u>Description</u>
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal

1 [2](#)

The screenshot shows the Visual Studio IDE interface. A `GridView` control is selected, displaying five rows of data with three columns: "Databound Col0", "Databound Col1", and "Databound Col2". The data consists of the letters "abc" in the first column, the numbers "0" through "4" in the second, and "abc" again in the third. Below the grid, a tooltip indicates it is bound to "SqlDataSource - DSCategories". A context menu titled "GridView Tasks" is open, listing options like "Auto Format...", "Choose Data Source:" (set to "DSCategories"), "Configure Data Source", "Refresh Schema", "Edit Columns...", "Add New Column...", and checkboxes for "Enable Paging" and "Enable Sorting".

The screenshot shows a web browser window with the URL `localhost:55440/Categories.aspx`. The page displays a `GridView` control with data from the "Categories" table. The columns are "CategoryID", "CategoryName", and "Description". The data includes:

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal

Pagination links "1 2" are visible at the bottom of the grid.

SqlDataSource and GridView

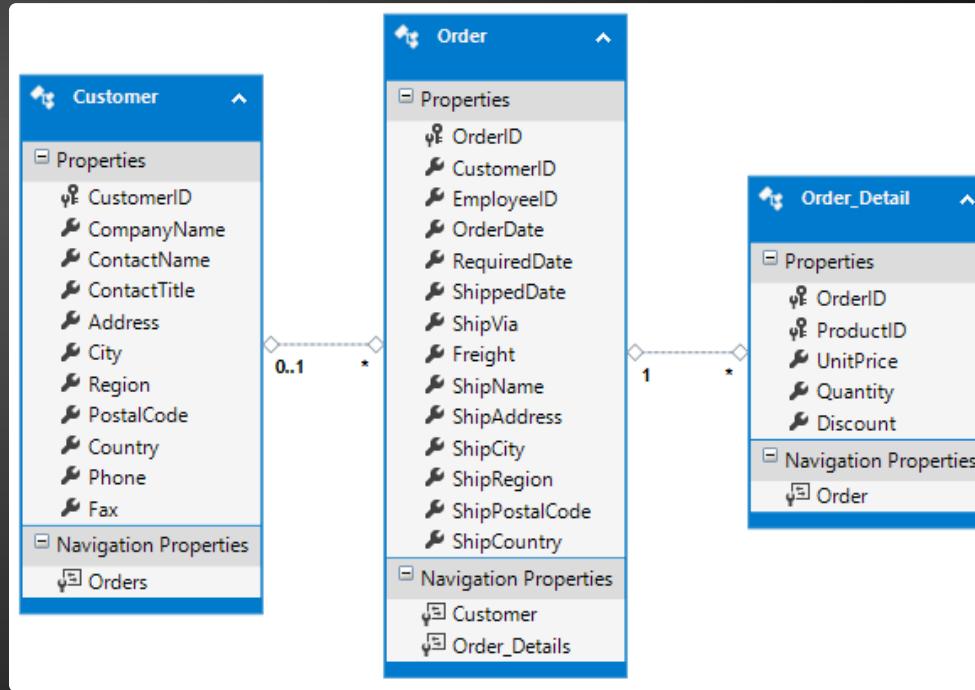
Live Demo

◆ The EntityDataSource

- ◆ Provides data binding in Web applications that use the ADO.NET Entity Framework (EF)
- ◆ Implements CRUD (create, read, update, and delete) operations
 - ◆ Against a database mapped through EF
 - ◆ On behalf of data-bound controls
- ◆ Supports complex queries, sorting and paging
- ◆ The Entity Data Model Designer in VS is used to create the EF mappings (.edmx)

EntityDataSource – Example

1. Define the Entity Data Model (.edmx)

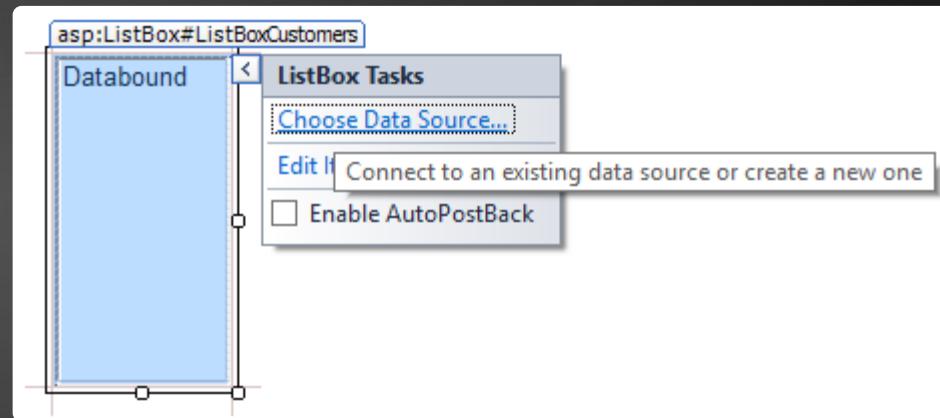


2. Put a list control in the form (e.g. ListBox)

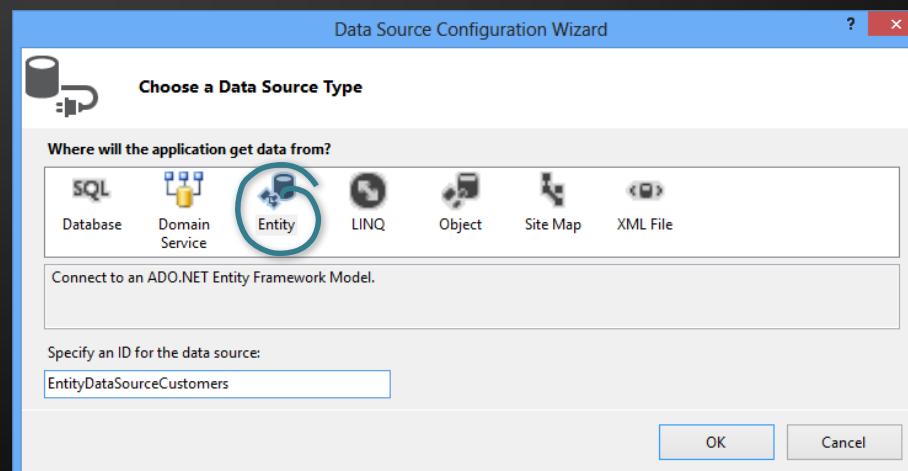
```
<asp:ListBox ID="ListBoxCustomers" runat="server">
</asp:ListBox>
```

EntityDataSource – Example (2)

3. Bind the ListBox to the data model

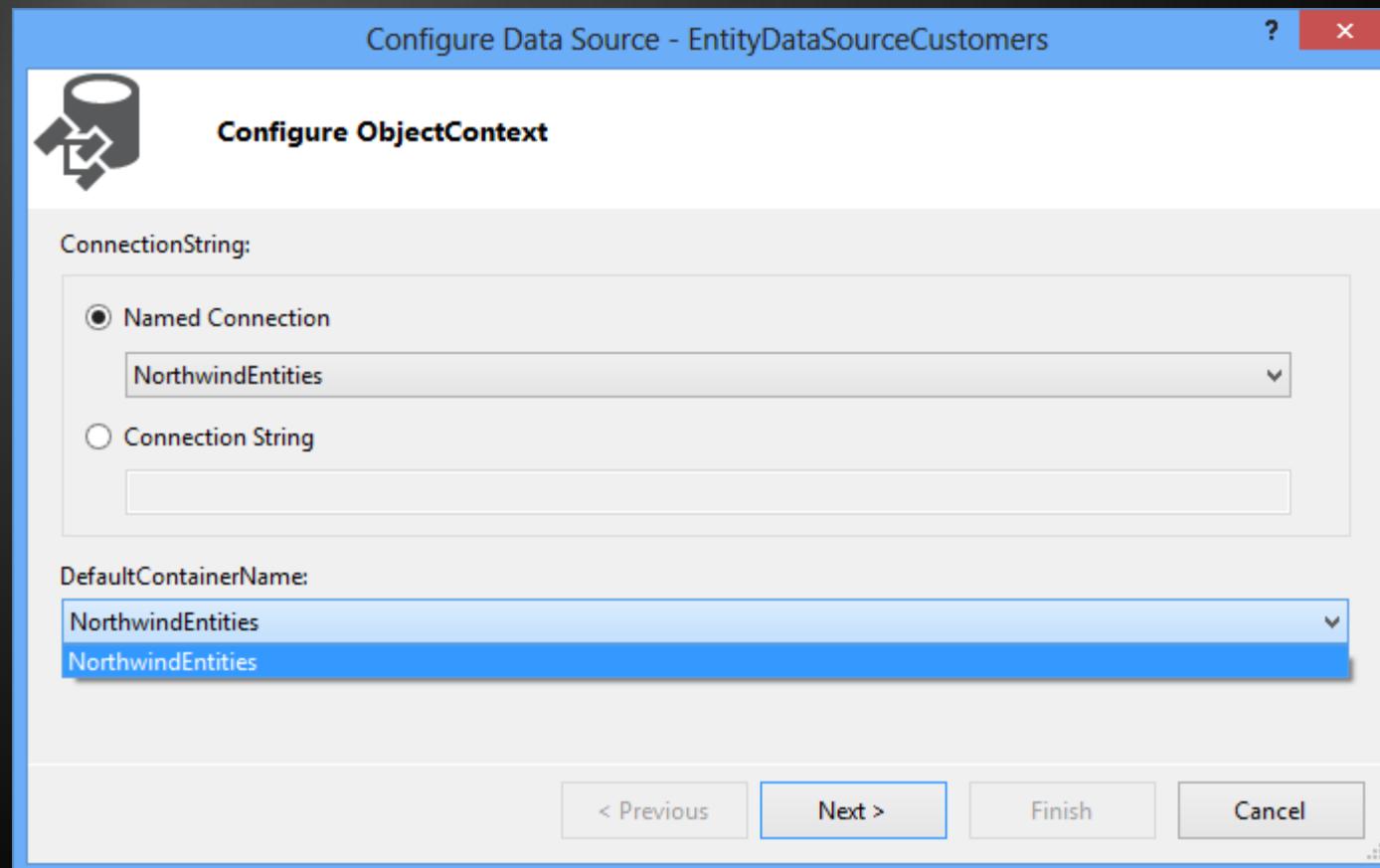


4. Select the new "Entity" option in the dialog box



EntityDataSource – Example (3)

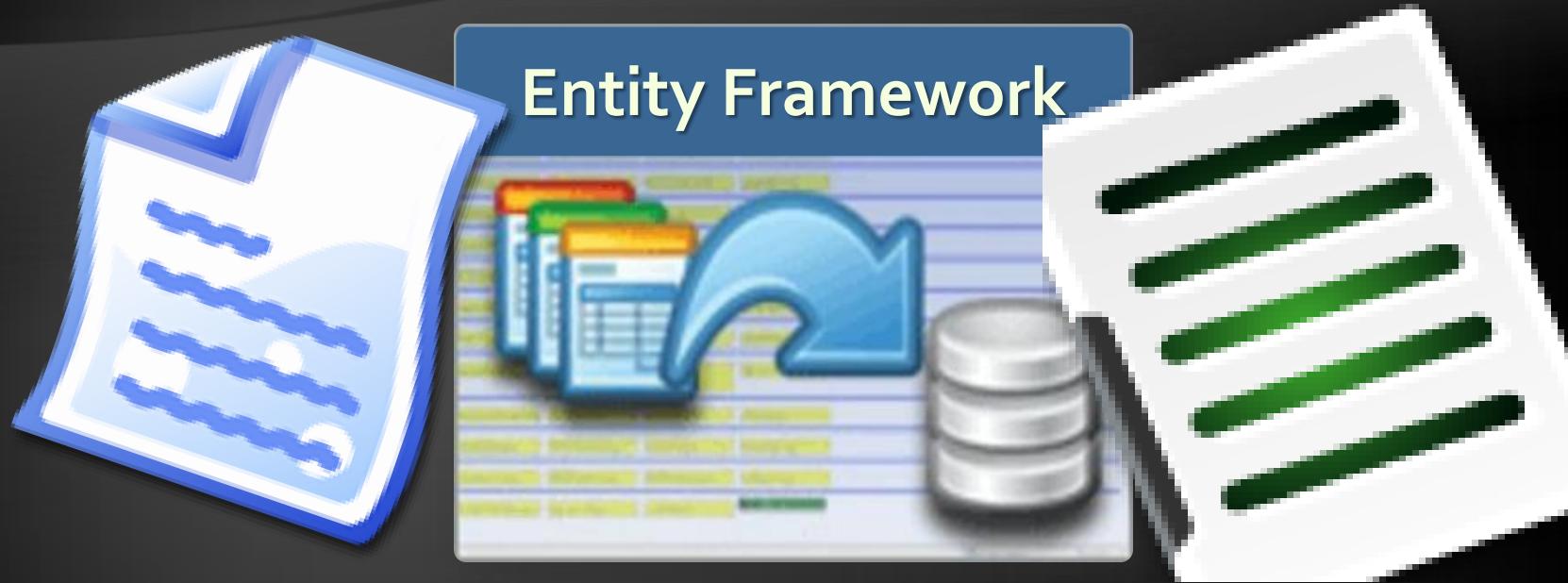
5. The VS Designer will then display the available Entity Containers



EntityDataSource – Example (4)

- ◆ **ListBox control bound an EntityDataSource:**

```
<asp:EntityDataSource  
    ID="EntityDataSourceCustomers" runat="server"  
    ConnectionString="name=NorthwindEntities"  
    DefaultContainerName="NorthwindEntities"  
    EntitySetName="Customers" />  
  
<asp:ListBox ID="ListBoxCustomers" runat="server"  
    DataSourceID="EntityDataSourceCustomers"  
    DataTextField="CompanyName" Rows="10"  
    DataValueField="CustomerID" />
```



ListBox Bound to EF through EntityDataSource

Live Demo

Master-Detail Navigation

- ◆ Master-details navigation could be implemented through filtering:

```
<asp:EntityDataSource ID="DSCustomers" runat="server"  
    EntitySetName="Customers" ... />  
  
<asp:ListBox ID="ListBoxCustomers" runat="server"  
    DataSourceID="EntityDataSourceCustomers" ... />  
  
<asp:EntityDataSource ID="DSOrders" runat="server"  
    EntitySetName="Orders" Where="it.CustomerID=@CustID">  
    <WhereParameters>  
        <asp:ControlParameter Name="CustID" Type="String"  
            ControlID="ListBoxCustomers" />  
    </WhereParameters>  
</asp:EntityDataSource>
```

EntityDataSource: Hints

- ◆ To access navigational properties in EntityDataSource use "Include"

```
<asp:EntityDataSource ID="DataSourceOrderDetails"
    ConnectionString="name=NorthwindEntities"
    DefaultContainerName="NorthwindEntities"
    EntitySetName="Order_Details" Include="Product"
    runat="server" ... />
```

- ◆ To use strongly-typed binding (ItemType) use EnableFlattening="false"

```
<asp:EntityDataSource ID="DataSourceOrderDetails"
    runat="server" EnableFlattening="false" ... />
```

Orders

	Order Date	Shipped Date	Ship Via	Freight	Ship Name
Select	1996-09-18 12:00:00 AM	1996-09-24 12:00:00 AM	3	1.6100	Ana Trujillo Emparedados
Select	1997-08-08 12:00:00 AM	1997-08-14 12:00:00 AM	1	43.9000	Ana Trujillo Emparedados
Select	1997-11-28 12:00:00 AM	1997-12-12 12:00:00 AM	3	11.9900	Ana Trujillo Emparedados
Select	1998-03-04 12:00:00 AM	1998-03-11 12:00:00 AM	3	39.9200	Ana Trujillo Emparedados

Order Details

Product: Tofu
Price: \$23.25
Quantity: 3
Discount: 0

Product: Singaporean Hokkien Fried Mee
Price: \$14.00
Quantity: 5
Discount: 0

Product: Camembert Pierrot
Price: \$34.00
Quantity: 10
Discount: 0

Master-Details Navigation (EntityDataSource + Filters)

Live Demo

Editable EntityDataSource

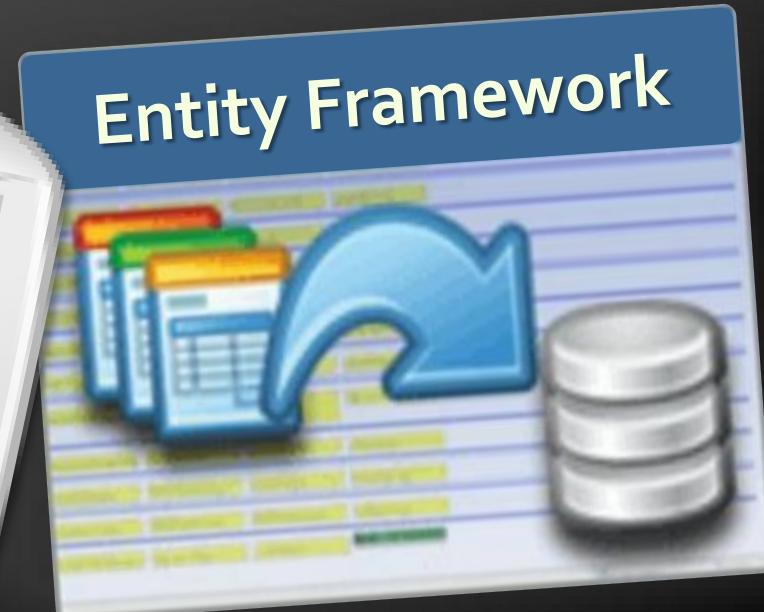
```
<asp:EntityDataSource ID="EntityDataSourceCustomers"
    runat="server" ConnectionString="name=..."
    DefaultContainerName="NorthwindEntities"
    EnableInsert="True" EnableUpdate="True"
    EnableDelete="True" EntitySetName="Customers">
</asp:EntityDataSource>

<asp:ListView ID="ListViewCustomers" runat="server"
    DataKeyNames="CustomerID"
    DataSourceID="EntityDataSourceCustomers">
    <LayoutTemplate>...</LayoutTemplate>
    <ItemTemplate>...</ItemTemplate>
    <EditTemplate>...</EditItemTemplate>
    <InsertTemplate>...</InsertItemTemplate>
</asp:ListView>
```

BindItem and Bind(...)

- ◆ Editable controls require a two-way binding (data field \leftrightarrow UI control)
 - Use `BindItem.Field` for strongly-typed binding instead of `Item.Field`
 - Use `Bind("PropertyName")` for standard binding instead of `Eval("PropertyName")`

```
<EditItemTemplate>
    Company Name: <asp:TextBox ID="TextBoxCompanyName"
        runat="server" Text='<%# BindItem.CompanyName %>' />
    ...
</EditItemTemplate>
```



Editable ListView with EF and EntityDataSource

Live Demo

ObjectDataSource

- ◆ ObjectDataSource enables data-binding of UI control to collection of objects
 - ◆ Instead of directly binding to a database
- ◆ Needs a middle-tier business object class
 - ◆ A class providing the Select, Update, Insert, Delete methods (CRUD operations)
- ◆ ObjectDataSource properties
 - ◆ TypeName – name of the business object class
 - ◆ SelectMethod, UpdateMethod, ...

ObjectDataSource – Example

```
<asp:ObjectDataSource ID="dsProducts"
    runat="server" TypeName="ObjectDataSourceProducts"
    SelectMethod="GetAll" InsertMethod="Insert"
    UpdateMethod="Update" DeleteMethod="Delete">
</asp:ObjectDataSource>

<asp:GridView ID="GridViewProducts" runat="server"
    DataSourceID="dsProducts" DataKeyNames="ProductID">
</asp:GridView>

public class ObjectDataSourceProducts
{
    public IEnumerable<Product> GetAll() { ... }
    public int UpdateProduct(Product p) { ... }

    ...
}
```

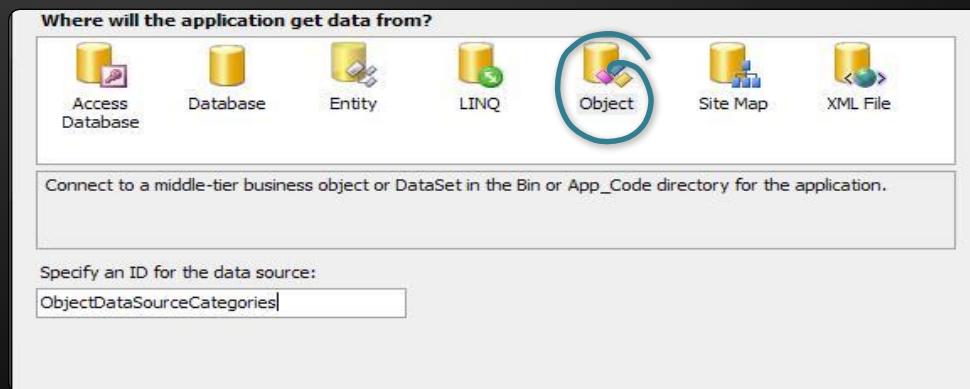
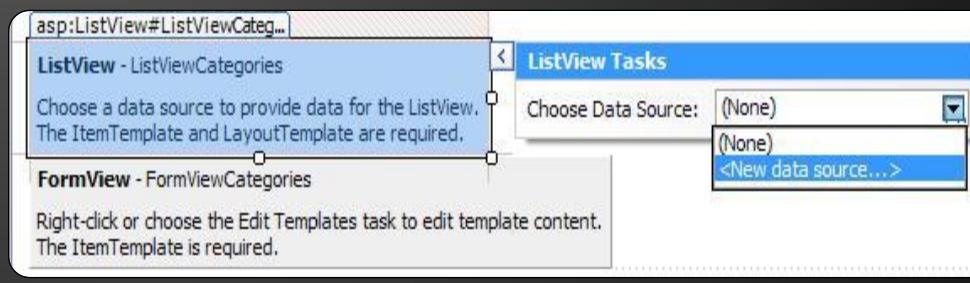
Building File System Explorer with ObjectDataSource

1. Define a class to get all files from given folder
2. Add GetAllFiles() method to return a collection of FileInfo objects

```
public class FileSystemManager
{
    public static IEnumerable<FileInfo>
        GetAllFiles(string folder)
    {
        var files = Directory.GetFiles(folder);
        return files;
    }
}
```

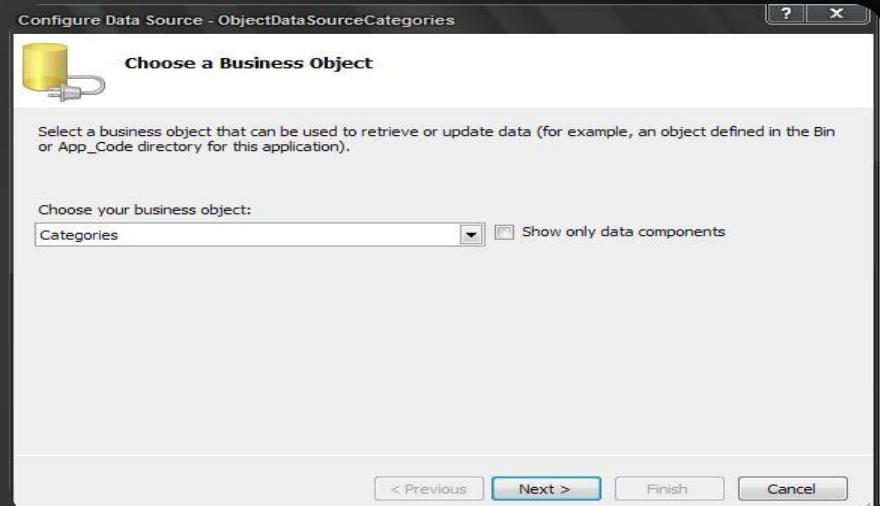
Building File System Explorer with ObjectDataSource (2)

3. Add a ListView control in the form
4. Bind the ListView to the ObjectDataSource

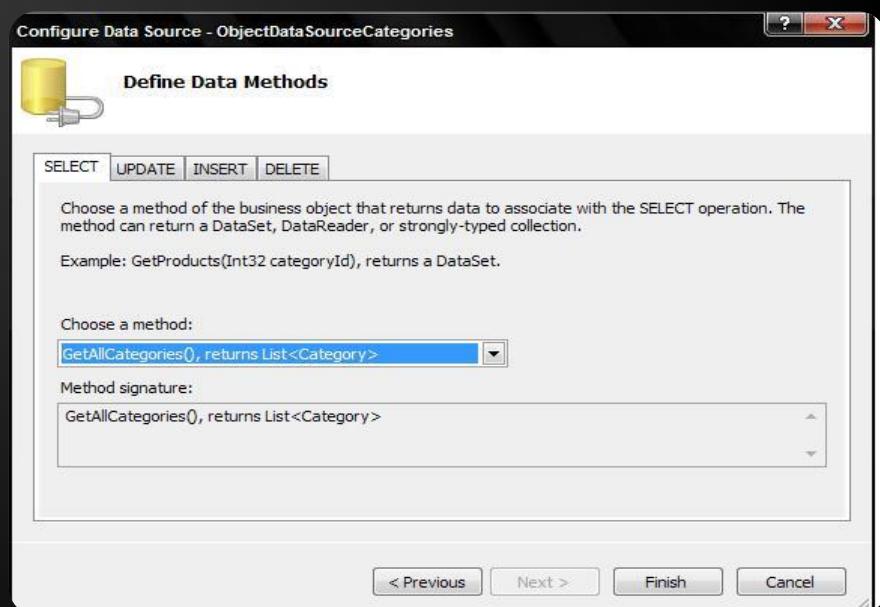


Building File System Explorer with ObjectDataSource (3)

5. Next choose your custom class
FilesManager



6. And choose the
GetAllFiles(...) method as "SELECT
Method"

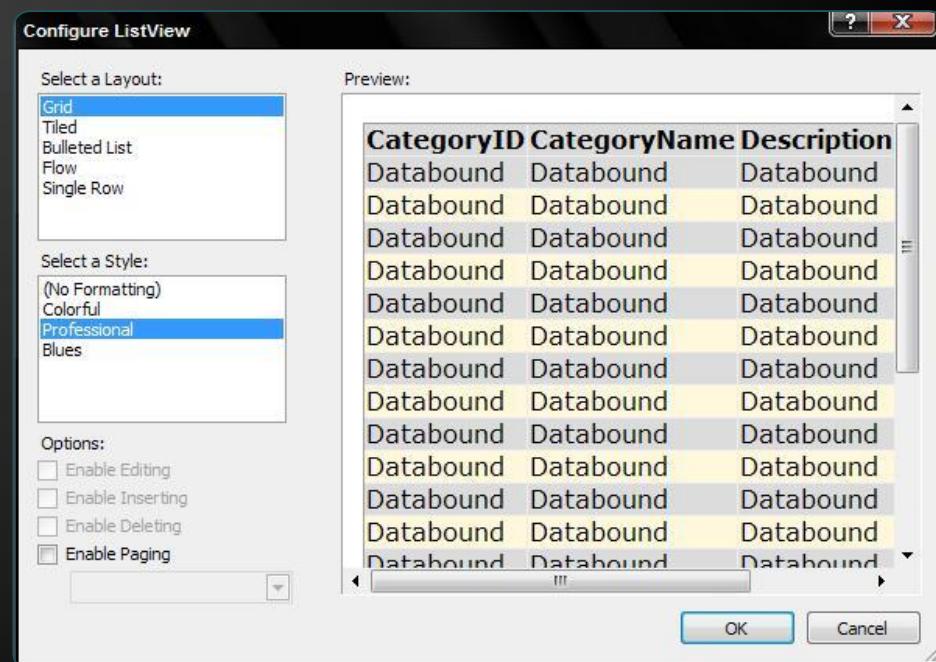


Building File System Explorer with ObjectDataSource (4)

7. Click to configure the ListView control



8. Optionally choose layout and style



Building File System Explorer with ObjectDataSource (5)

9. Customize the ListView to show only rows you like

10. Add a parameter for the SELECT method

```
<title></title>
- </head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:ListView ID="ListViewCategories" runat="server"
        DataSourceID="ObjectDataSourceCategories">
        <ItemTemplate>
          <tr style="background-color:#DCDCDC;color: #000000;">
            <td>
              <asp:Label ID="CategoryIDLabel" runat="server"
                Text='<%# Eval("CategoryID") %>' />
            </td>
            <td>
              <asp:Label ID="CategoryNameLabel" runat="server"
                Text='<%# Eval("CategoryName") %>' />
            </td>
            <td>
              <asp:Label ID="DescriptionLabel" runat="server"
                Text='<%# Eval("Description") %>' />
            </td>
            <td>
              <asp:Label ID="PictureLabel" runat="server" Text='<%#
```

```
<SelectParameters>
  <asp:Parameter DefaultValue="C:\WINDOWS"
    Name="rootFolder" Type="String" />
</SelectParameters>
```

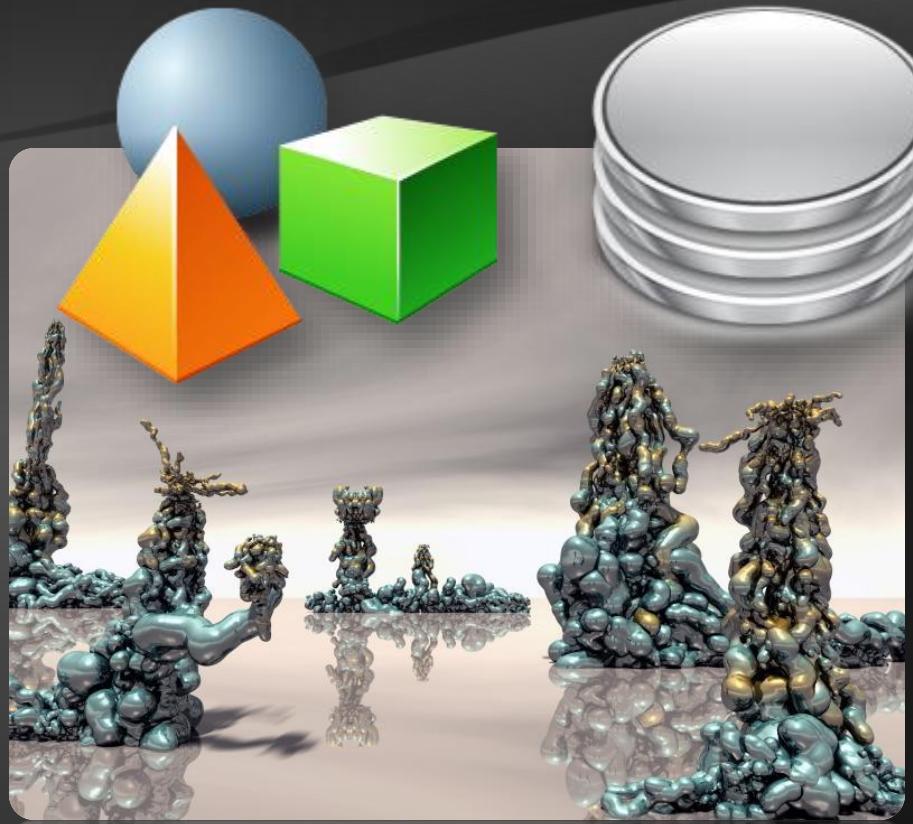
Building File System Explorer with ObjectDataSource (6)

10. The result is:

The screenshot shows a web browser window titled "ObjectDataSource: File Sys" with the URL "localhost:29978/FileSystemExplorer.aspx". The page displays a table of files from the root folder "C:\WINDOWS". The table has columns: Name, Length, DirectoryName, IsReadOnly, and CreationTime. The data is as follows:

Name	Length	DirectoryName	IsReadOnly	CreationTime
regedit.exe	159232	C:\WINDOWS	<input type="checkbox"/>	26.07.2012 4:14:51
RtlExUpd.dll	1698408	C:\WINDOWS	<input type="checkbox"/>	09.09.2012 18:50:03
setupact.log	37656	C:\WINDOWS	<input type="checkbox"/>	09.02.2013 23:17:05
setuperr.log	0	C:\WINDOWS	<input type="checkbox"/>	09.02.2013 23:17:05
splwow64.exe	126464	C:\WINDOWS	<input type="checkbox"/>	26.07.2012 4:27:54
Starter.xml	31537	C:\WINDOWS	<input type="checkbox"/>	26.07.2012 10:59:47
system.ini	219	C:\WINDOWS	<input type="checkbox"/>	26.07.2012 8:26:52
TSSysprep.log	1313	C:\WINDOWS	<input type="checkbox"/>	12.09.2010 5:35:37
twain_32.dll	50176	C:\WINDOWS	<input type="checkbox"/>	26.07.2012 4:19:02
vmgcoinstall.log	1585	C:\WINDOWS	<input type="checkbox"/>	26.07.2012 10:21:16

At the bottom of the table, there are navigation links: 1 2 3 4.



Using ObjectDataSource

Live Demo

- ◆ **LinqDataSource**
 - ◆ For LINQ-to-SQL ORM mappings (legacy)
- ◆ **Hierarchical**
 - ◆ **XmlDataSource**
 - ◆ Establishes a connection to an XML source of data (files, documents)
 - ◆ **DataFile, TransformFile, XPath**
 - ◆ **SiteMapDataSource**
- ◆ **MS Access – AccessDataSource**
 - ◆ Derives from **SqlDataSource**



Model Binding

CRUD Operations Made Easy

What is Model Binding?

- ◆ What is "model"?
 - ◆ In data-driven applications "models" are classes that hold the data behind the UI controls
- ◆ What is "model binding" in ASP.NET?
 - ◆ Bind data controls directly to methods that provide CRUD functionality
 - ◆ `SelectMethod` → returns `IQueryable<T>`
 - ◆ `InsertMethod`, `UpdateMethod`, `DeleteMethod` → edit the model (entity)

Model Binding - Demo

localhost:1407/ModelBindingExample.aspx

Product	Supplier	Category	Quantity Per Unit	Unit Price	Units in Stock	Units on Order	Reorder Level	Discontinued
Chai	Exotic Liquids	Beverages	10 boxes x 20 bags	\$18.00	39	0	10	<input type="checkbox"/>
Chang	Exotic Liquids	Beverages	24 - 12 oz bottles	\$19.00	17	40	25	<input type="checkbox"/>
Aniseed Syrup	Exotic Liquids	Condiments	12 - 550 ml bottles	\$10.00	13	70	25	<input type="checkbox"/>
Chef Anton's Cajun Seasoning	New Orleans Cajun Delights	Condiments	48 - 6 oz jars	\$22.00	53	0	0	<input type="checkbox"/>
Chef Anton's Gumbo Mix	New Orleans Cajun Delights	Condiments	36 boxes	\$21.35	0	0	0	<input checked="" type="checkbox"/>
Grandma's Boysenberry Spread	Grandma Kelly's Homestead	Condiments	12 - 8 oz jars	\$25.00	120	0	25	<input type="checkbox"/>
Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead	Produce	12 - 1 lb pkgs.	\$30.00	15	0	10	<input type="checkbox"/>
Northwoods Cranberry Sauce	Grandma Kelly's Homestead	Condiments	12 - 12 oz jars	\$40.00	6	0	0	<input type="checkbox"/>
Mishi Kobe Niku	Tokyo Traders	Meat/Poultry	18 - 500 g pkgs.	\$97.00	29	0	0	<input checked="" type="checkbox"/>
Ikura	Tokyo Traders	Seafood	12 - 200 ml jars	\$31.00	31	0	0	<input type="checkbox"/>

1 2 3 4 5 6 7 8

Model Binding

Live Demo

ASP.NET Data Source Controls

Questions?

- 1. Create a database holding continents, countries and town. Countries have name, language, population and continent. Towns have name, population and country. Implement an ASP.NET Web application that shows the continents in a ListBox, countries in a GridView and the towns in a ListView and allows master-detail navigation. Use Entity Framework and EntityDataSource. Provide paging and sorting for the long lists. Use HTML escaping when needed.**

- 2. Implement add / edit / delete for the continents, countries, towns and languages. Handle the possible errors accordingly. Ensure HTML special characters handled correctly (correctly escape the HTML).**

3. Add a flag for each country which should be a PNG image, stored in the database, displayed along with the country data. Implement "change flag" functionality by uploading a PNG image.
4. Write a "TODO List" application in ASP.NET. It should be able to list / create / edit / delete TODOs. Each TODO has title, body (rich text) and date of last change. The TODOs are in categories. Categories should also be editable (implement CRUD).
5. * Write ASP.NET Web Forms application to display in a GridView all tweets from given Twitter user. Load the tweet though the Twitter API. Bind the tweets to the GridView thorugh ObjectDataSource.

Free Trainings @ Telerik Academy

- ◆ "Web Design with HTML 5, CSS 3 and JavaScript" course @ Telerik Academy



- ◆ html5course.telerik.com

- ◆ Telerik Software Academy

- ◆ academy.telerik.com

Telerik Academy

- ◆ Telerik Academy @ Facebook

- ◆ [facebook.com/TelerikAcademy](https://www.facebook.com/TelerikAcademy)



- ◆ Telerik Software Academy Forums

- ◆ forums.academy.telerik.com

