

LINQ & ADO.Net Entity Framework

Asst. Prof. Dr. Özgü Can

LINQ

- Language-Integrated Query
- Enables you to **query data from within .NET programming languages** similar to how SQL enables you to query data in a database.
- LINQ has keywords such as **Select**, **From**, and **Where** to get data from a data source.
- Integrated into .NET
 - Available in all kinds of projects including web applications.

ADO.NET Entity Framework

- *Take a bunch of database objects like tables and turn them into .NET objects* that you can access in your code.
- Use these objects in queries or use them directly in data-binding scenarios.
- Entity Framework also enables you to do the reverse:
 - *Design an object model first and then let EF create the necessary database structure* for you.

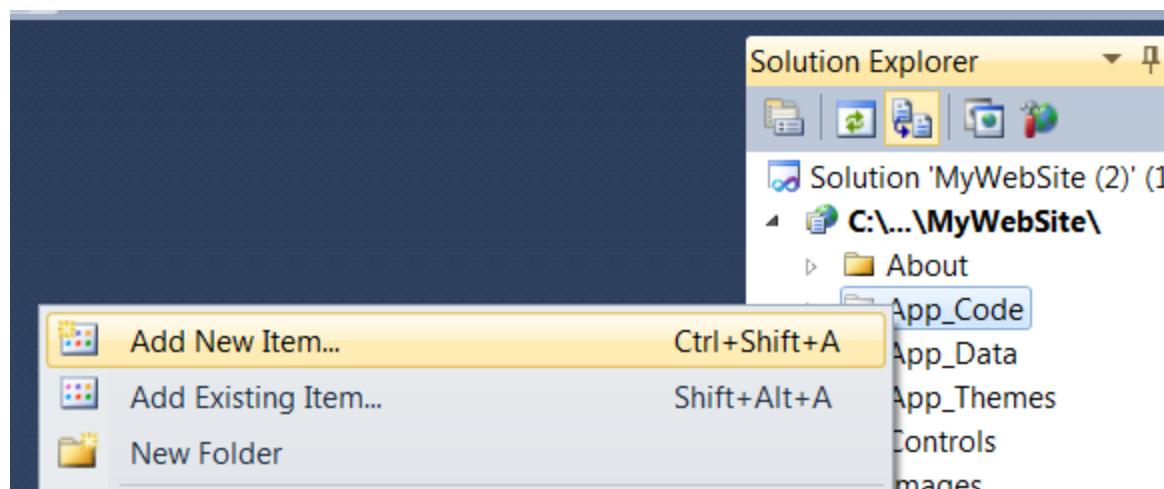
NOTE

- Use → **System.Linq** namespace
- ADO.NET is used under the hood by the **SqlDataSource** control.
- Entity Framework **creates a layer between .NET application and SQL Server database.**

Example

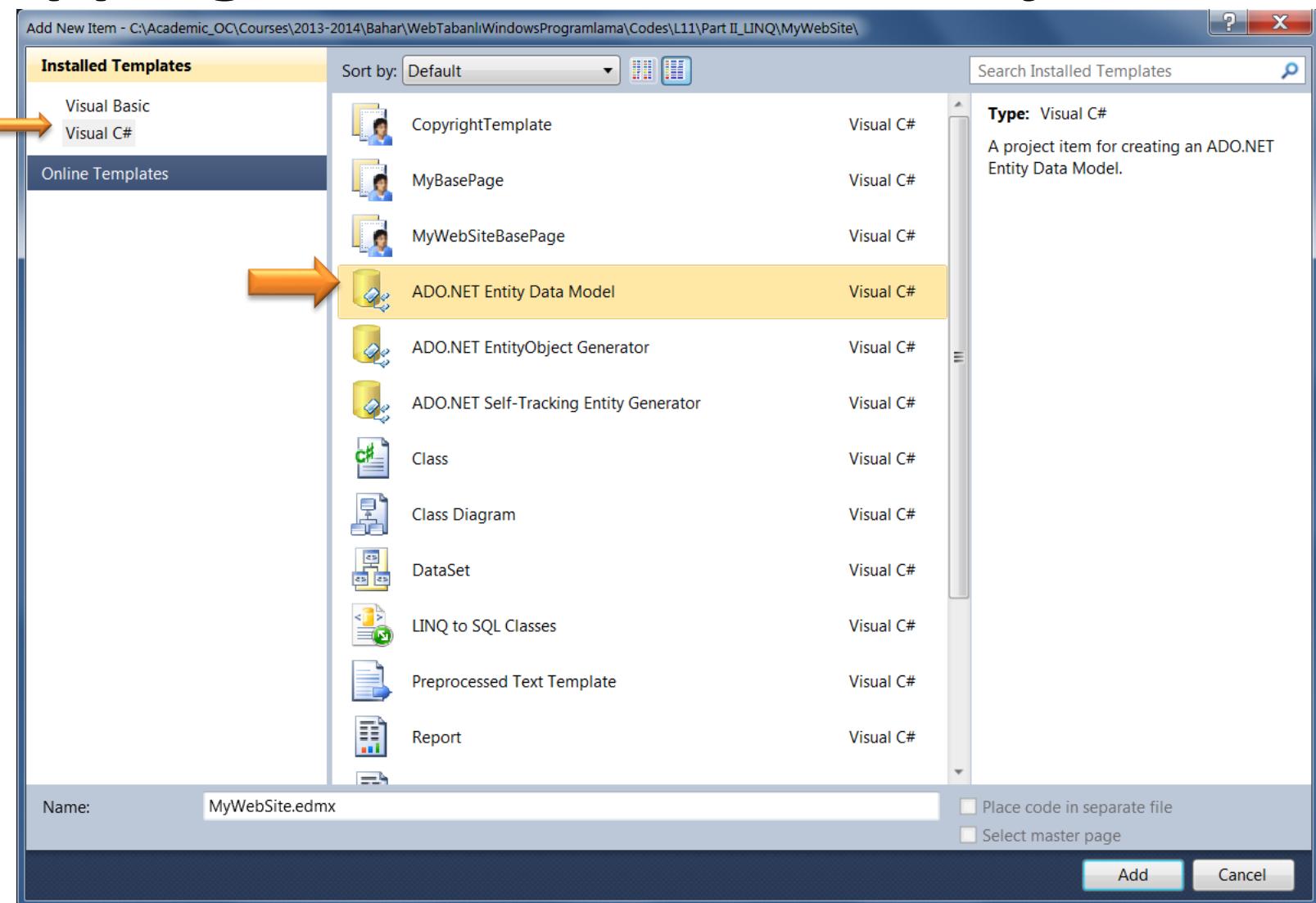
Mapping Data Model to an Object Model

Right-click **App_Code** folder → Add New Item



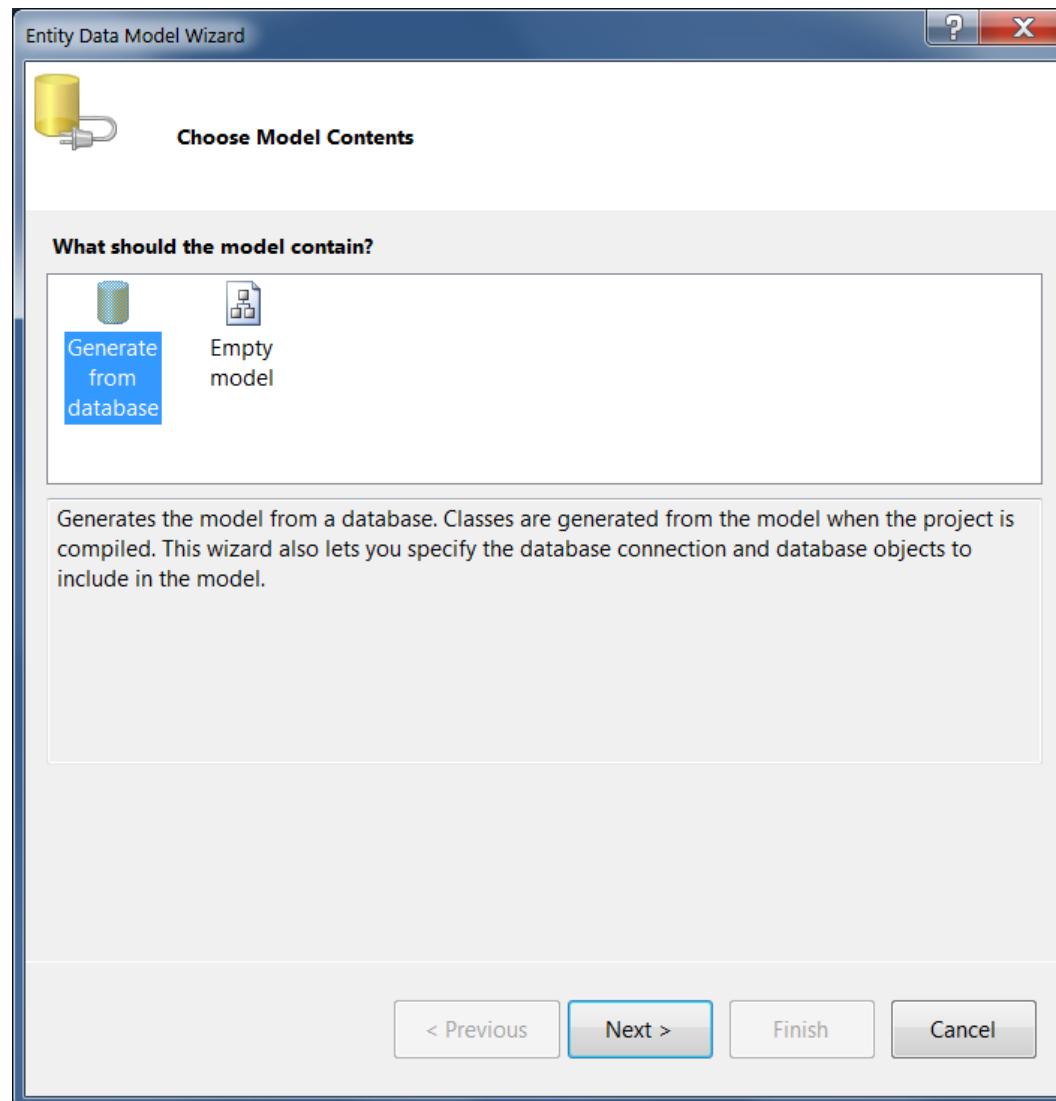
Example

Mapping Data Model to an Object Model



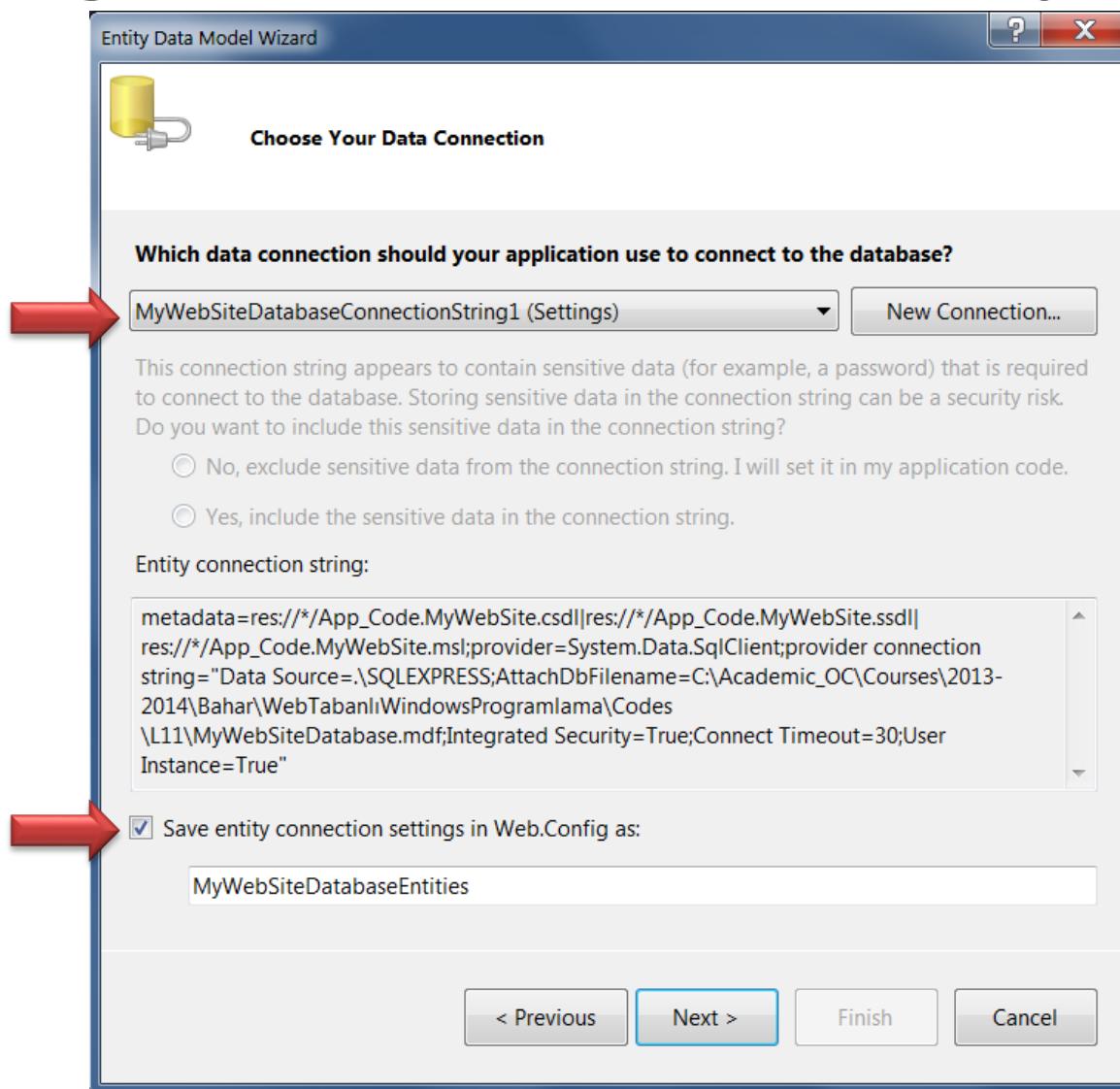
Example

Mapping Data Model to an Object Model



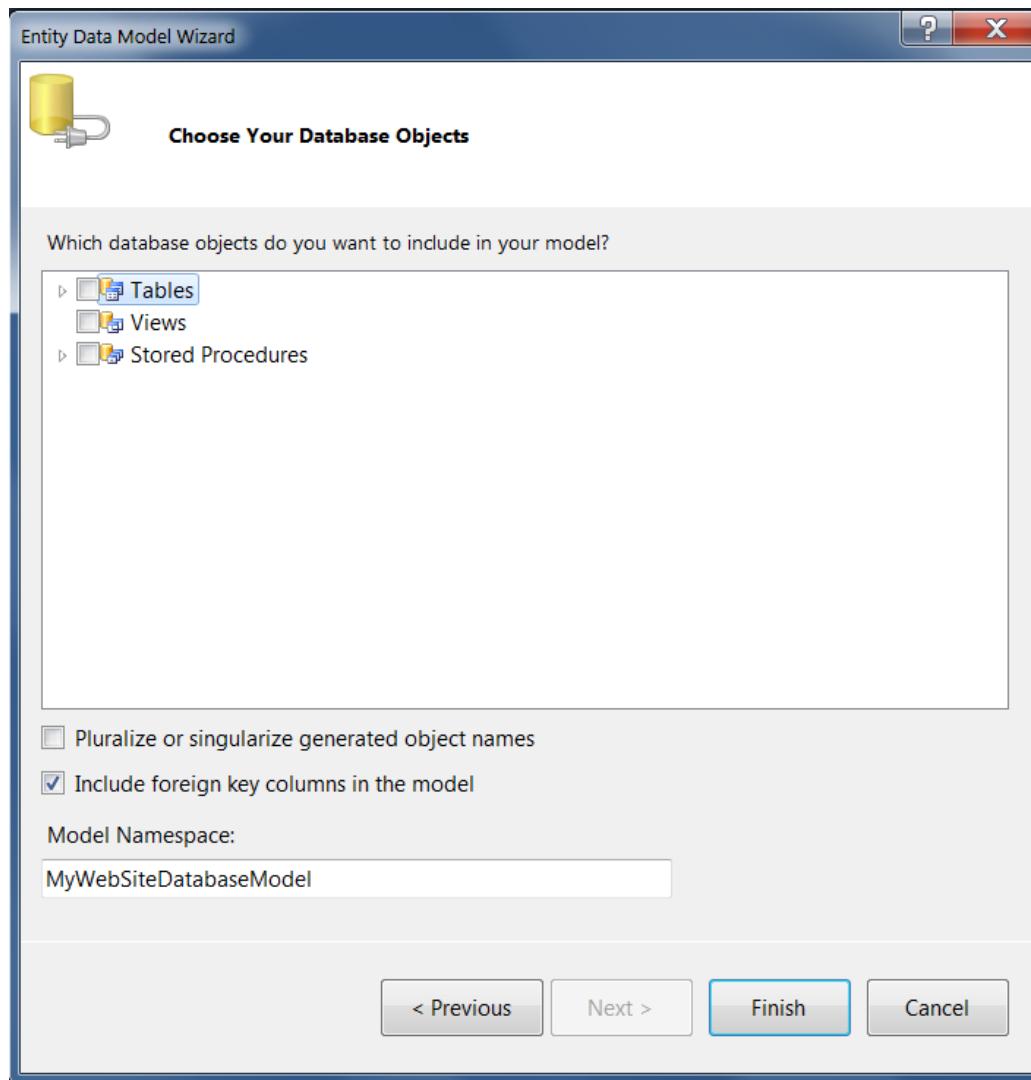
Example

Mapping Data Model to an Object Model



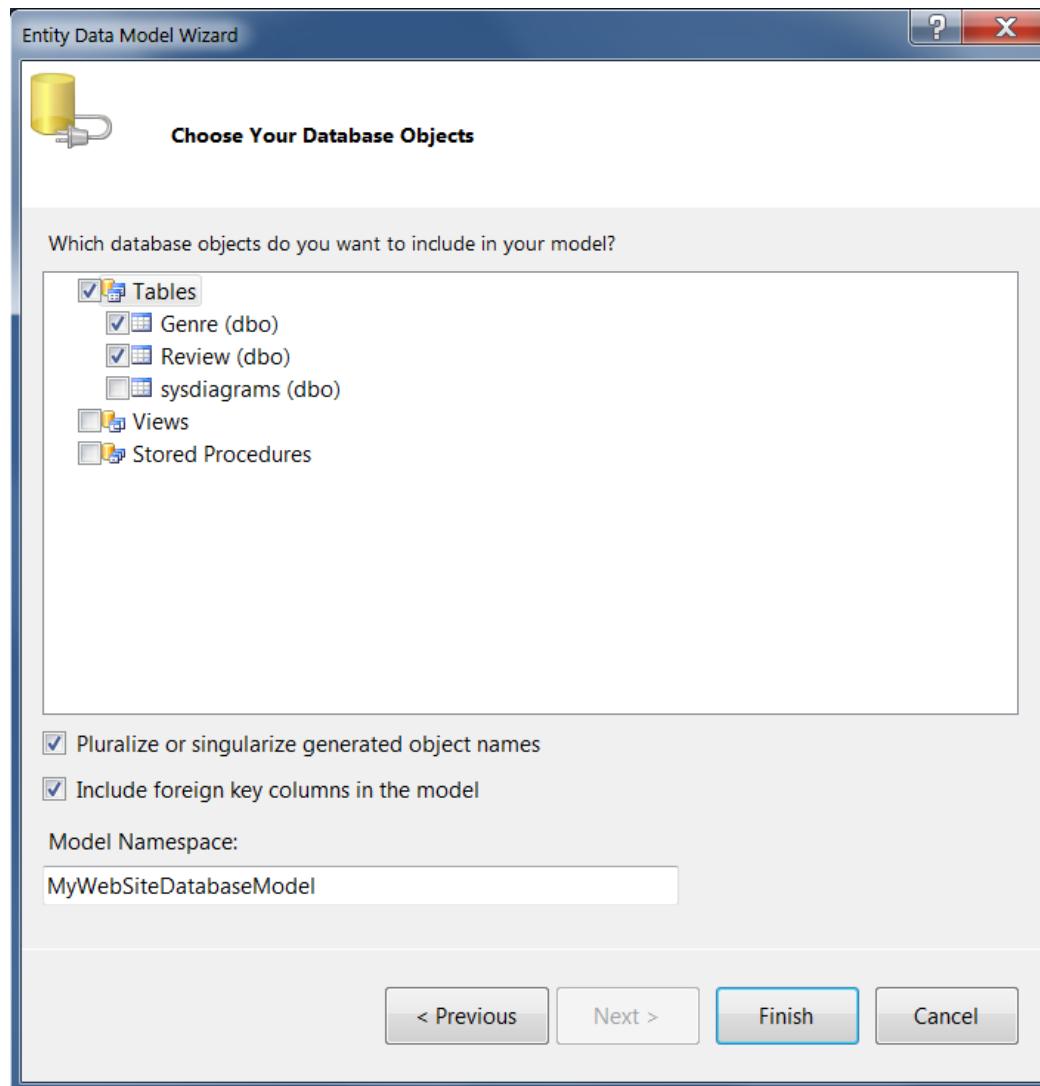
Example

Mapping Data Model to an Object Model



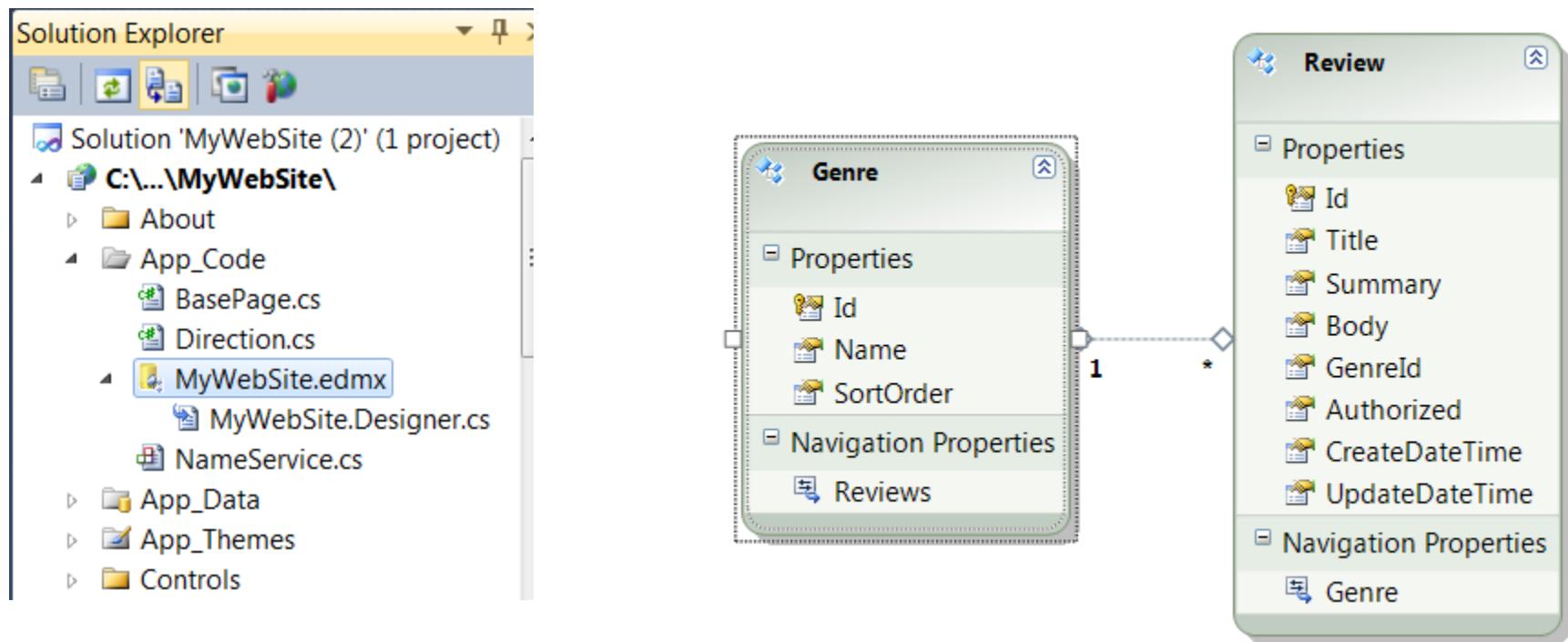
Example

Mapping Data Model to an Object Model



Example

Mapping Data Model to an Object Model



Example

Mapping Data Model to an Object Model

The image displays two side-by-side EntityDataSource Properties windows, illustrating the mapping of database entities to object models.

Left Window (Genre Entity):

Abstract	False
Access	Public
Base Type	(None)
Documentation	
Entity Set Name	Genres
Name	Genre

Right Window (Review Entity):

Abstract	False
Access	Public
Base Type	(None)
Documentation	
Entity Set Name	Reviews
Name	Review

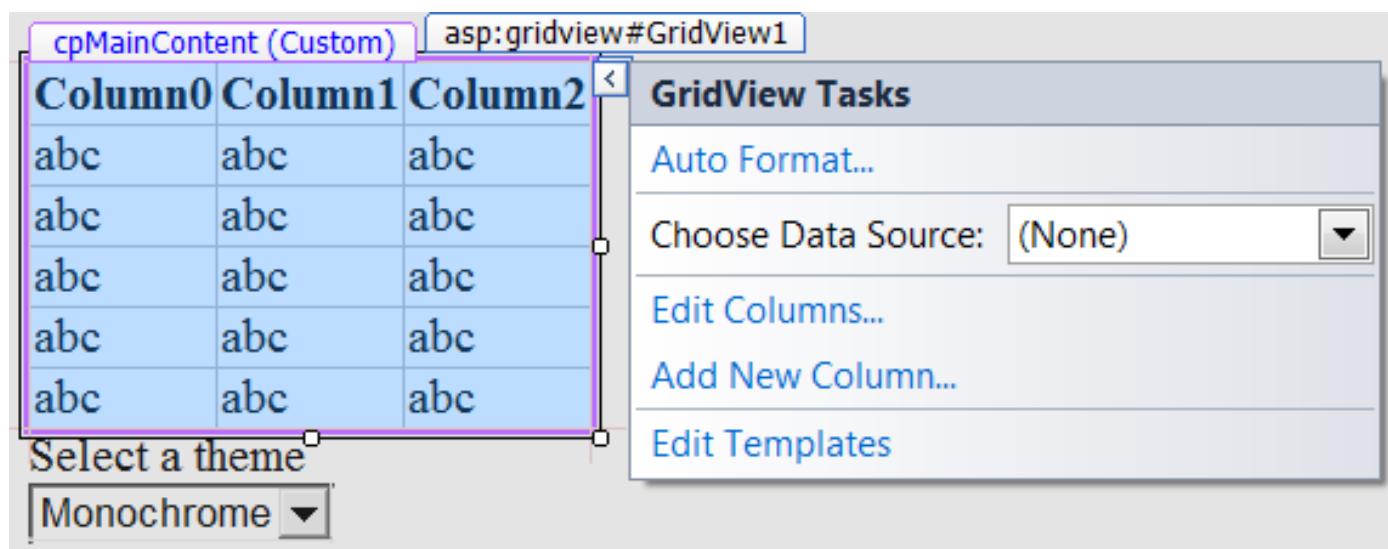
Common Bottom Section (Name):

Name
The name of the entity.

Example

Mapping Data Model to an Object Model

- Open **Reviews > All.aspx** in Design View
 - Drag a **GridView** control



Example

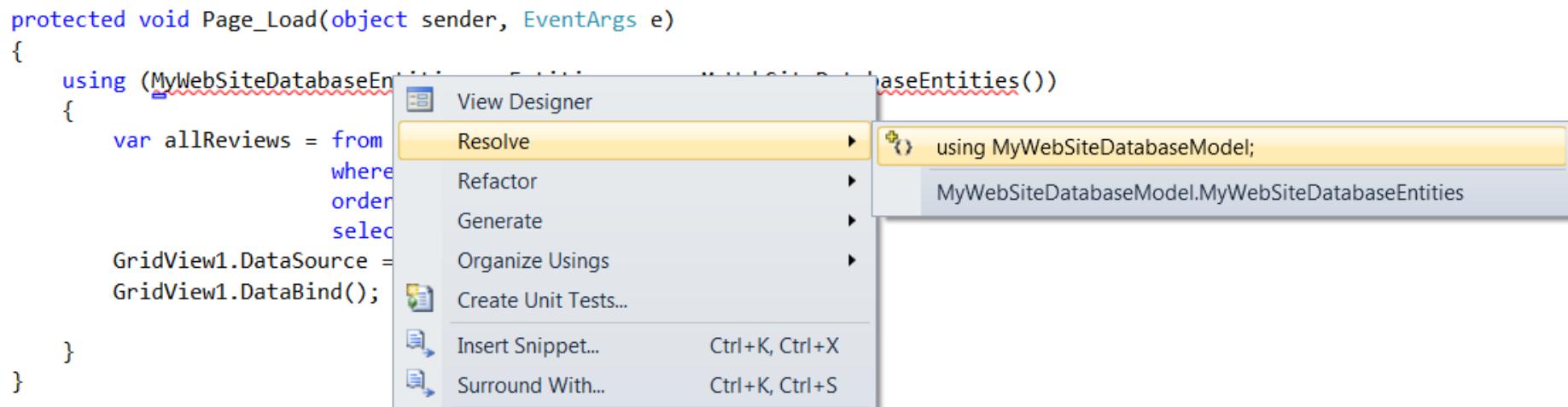
Mapping Data Model to an Object Model

- Double-click the grey area:

```
protected void Page_Load(object sender, EventArgs e)
{
    using (MyWebSiteDatabaseEntities myEntities = new MyWebSiteDatabaseEntities())
    {
        var allReviews = from review in myEntities.Reviews
                         where review.Authorized == true
                         orderby review.CreateDateTime descending
                         select review;
        GridView1.DataSource = allReviews;
        GridView1.DataBind();
    }
}
```

Example

Mapping Data Model to an Object Model



Example

Mapping Data Model to an Object Model

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MyWebSiteDatabaseModel;

public partial class Reviews_Default : BasePage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        using (MyWebSiteDatabaseEntities myEntities = new MyWebSiteDatabaseEntities())
        {
            var allReviews = from review in myEntities.Reviews
                            where review.Authorized == true
                            orderby review.CreateDateTime descending
                            select review;
            GridView1.DataSource = allReviews;
            GridView1.DataBind();
        }
    }
}
```

Example

Mapping Data Model to an Object Model

- View All.aspx in browser

Home Reviews About Login

Home > Reviews > All Reviews

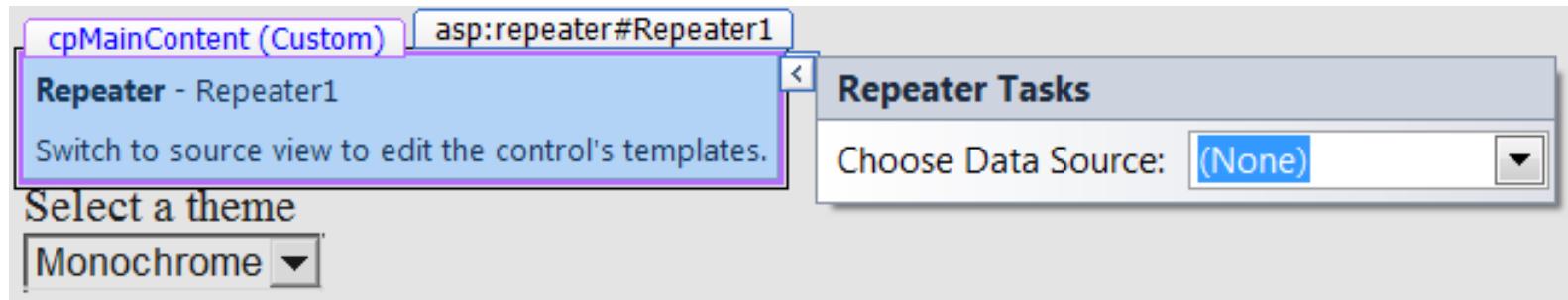
ID	Title	Summary	Body	GenreID	Authorized	CreateDateTime	UpdateDateTime
6	21 by Adele	It's a great album.		4	<input checked="" type="checkbox"/>	01.05.2013 21:19:10	01.05.2013 21:19:10
5	This is the title	This is the summary	This is the body	4	<input checked="" type="checkbox"/>	01.05.2013 19:17:48	01.05.2013 00:00:00
4	Beethoven's 5 Secrets-The Piano Guys	Beethoven		8	<input checked="" type="checkbox"/>	29.04.2013 17:09:15	29.04.2013 17:09:15
3	Random Access Memories-Daft Punk	The fourth studio album from French		6	<input checked="" type="checkbox"/>	29.04.2013 17:04:46	29.04.2013 17:04:46
2	Doo-Wops & Hooligans-Bruno Mars	Bruno Mars is a musical renaissance man		4	<input checked="" type="checkbox"/>	29.04.2013 17:02:40	29.04.2013 17:02:40
1	21-Adele	Adele "the next Amy Winehouse"		4	<input checked="" type="checkbox"/>	29.04.2013 17:00:30	29.04.2013 17:00:30

Select a theme
Monochrome ▾



Example Queries

- Open **Reviews > AllByGenre.aspx** in Design View.
 - Drag a **Repeater** control



Example Queries

- In Source View:

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:Repeater ID="Repeater1" runat="server">
        </asp:Repeater>
</asp:Content>
```



```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:Repeater ID="Repeater1" runat="server">
        <ItemTemplate>
            <h3>
                <asp:Literal ID="Literal1" Text='<%# Eval("Name") %>' runat="server">
                    </asp:Literal>
                </h3>
            </ItemTemplate>
        </asp:Repeater>
    </asp:Content>
```

Example Queries

- Below the <h3>:
 - Drag a **BulletedList** control

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:Repeater ID="Repeater1" runat="server">
        <ItemTemplate>
            <h3>
                <asp:Literal ID="Literal1" Text='<%# Eval("Name") %>' runat="server">
                </asp:Literal>
            </h3>
            <asp:BulletedList ID="BulletedList1" runat="server">
                </asp:BulletedList>
        </ItemTemplate>
    </asp:Repeater>
</asp:Content>
```

Example Queries

- Change properties of **BulletedList** control:
 - **ID = reviewListBulletedList**
 - **DataSource = <%#Eval("Reviews")%>**
 - **DataTextField = Title**
 - **DisplayMode = Text**

```
<asp:BulletedList ID="reviewListBulletedList" runat="server" DataSource='<%#Eval("Reviews")%>' DataTextField="Title">
</asp:BulletedList>
```

Example Queries

- Double-click the grey area in Design View:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using MyWebSiteDatabaseModel;

public partial class Reviews_Default : BasePage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        using (MyWebSiteDatabaseEntities myEntities = new MyWebSiteDatabaseEntities())
        {
            var allGenres = from genre in myEntities.Genres
                           orderby genre.Name
                           select new { genre.Name, genre.Reviews };
            Repeater1.DataSource = allGenres;
            Repeater1.DataBind();
        }
    }
}
```

Example Queries

- View **AllByGenre.aspx** in browser.

The screenshot shows a web application interface for a music review site. At the top, there is a navigation bar with four items: "Home", "Reviews", "About", and "Login". Below the navigation bar, the URL "Home > Reviews > By Genre" is displayed. On the right side of the page, there is a "Select a theme" dropdown menu set to "Monochrome". A large red hand icon is visible on the right side of the main content area. The main content area lists various music genres:

- Blues**
- Chicago Jazz**
- Classical**
 - Beethoven's 5 Secrets-The Piano Guys
- Country**
- Jazz**
- Pop**
 - 21-Adele
 - Doo-Wops & Hooligans-Bruno Mars
 - This is the title
 - 21 by Adele
- Pop Rock**
- Rock**
 - Random Access Memories-Daft Punk

Using Data Controls with Entity Framework

CONTROL	DESCRIPTION
EntityDataSource	EntityDataSource works as a bridge between your data-bound controls and the underlying data source: EF in this case.
ListView	The ListView control provides a visual interface that <u>enables you to display, insert, edit, and delete items in a database, providing full CRUD services.</u>
DataPager	The DataPager is used together with the ListView and enables you to <u>paginate the data</u> in the data source, feeding data to users in bite-sized chunks instead of all records at once.

EntityDataSource

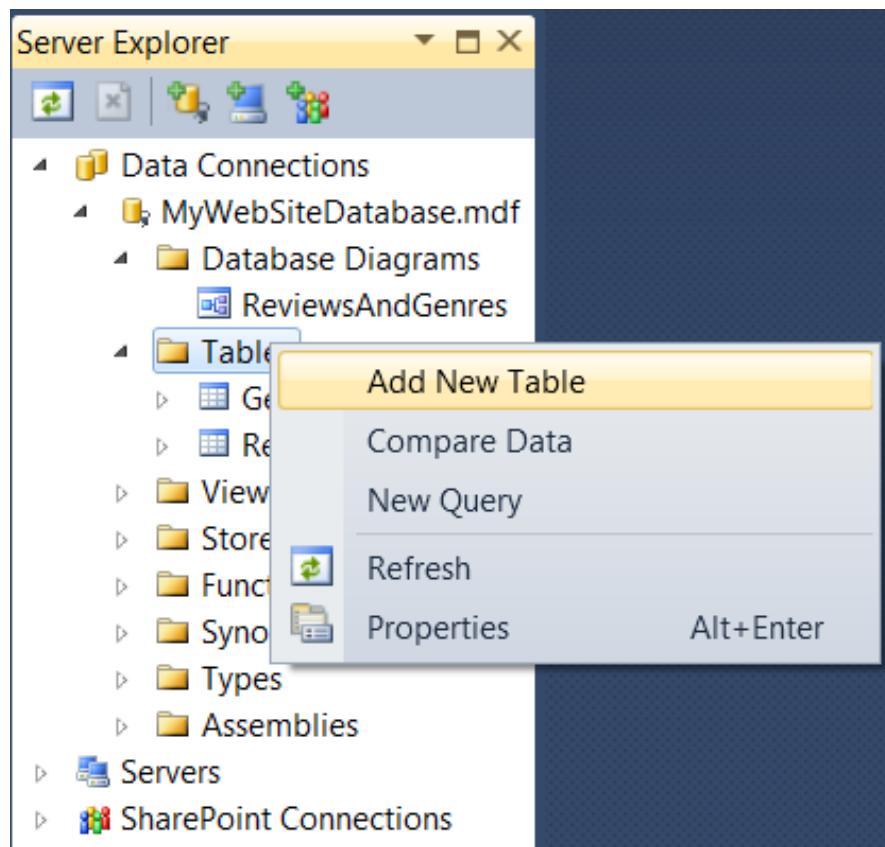
PROPERTY	DESCRIPTION
EnableDelete	Determine whether the control provides automatic insert, update, and delete capabilities. When enabled, you can use the control together with data-bound controls like the GridView or ListView to support data management.
EnableInsert	
EnableUpdate	
ContextTypeName	The name of the ObjectContext class that the control should use.
EntitySetName	The name of the entity set from the EF diagram you want to use, such as Reviews.
Select	Enable you to define the query that the EntityDataSource control fires against the model. Each of these properties maps to one of the query operators.
OrderBy	
Where	

ContextTypeName = MyWebSiteDatabaseEntities

Example

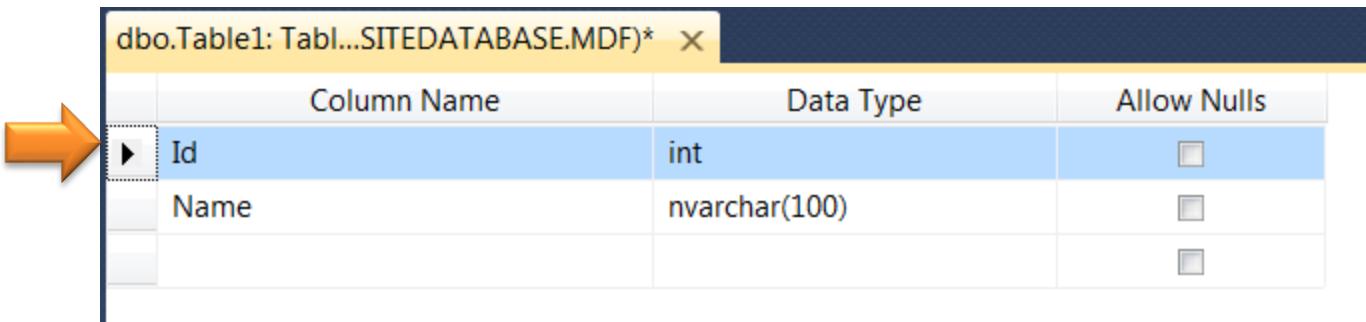
EntityDataSource

- Open **MyWebSiteDatabase.mdf**

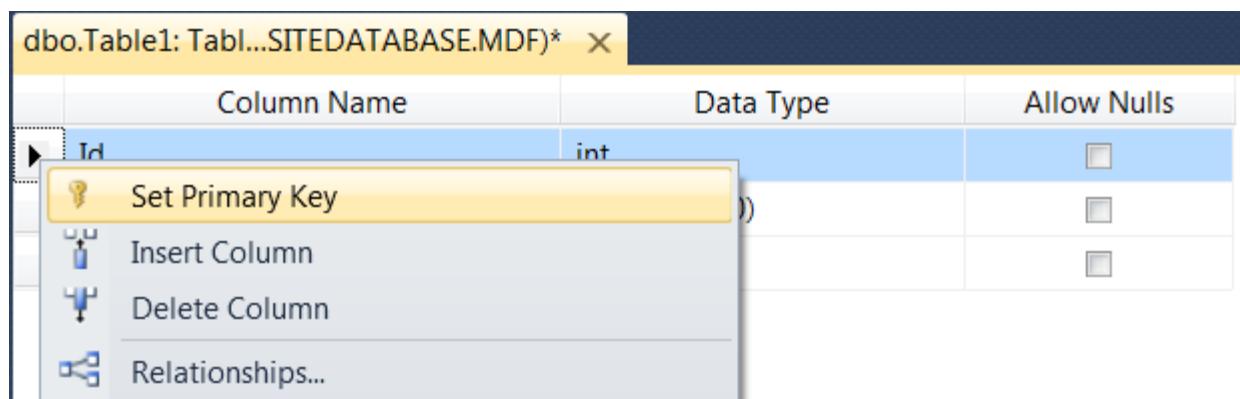


Example

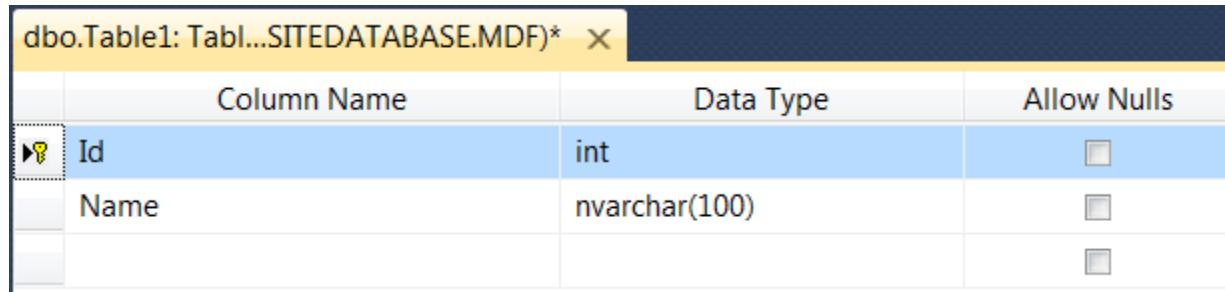
EntityDataSource



	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	Name	nvarchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>



	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
Set Primary Key			
Insert Column			
Delete Column			
Relationships...			



	Column Name	Data Type	Allow Nulls
▶	Id	int	<input type="checkbox"/>
	Name	nvarchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>

Example

EntityDataSource

dbo.PhotoAlbum:...SITEDATABASE.MDF)* X dbo.Picture: Tabl...SITEDATABASE.MDF)*

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Name	nvarchar(100)	<input type="checkbox"/>
		<input type="checkbox"/>

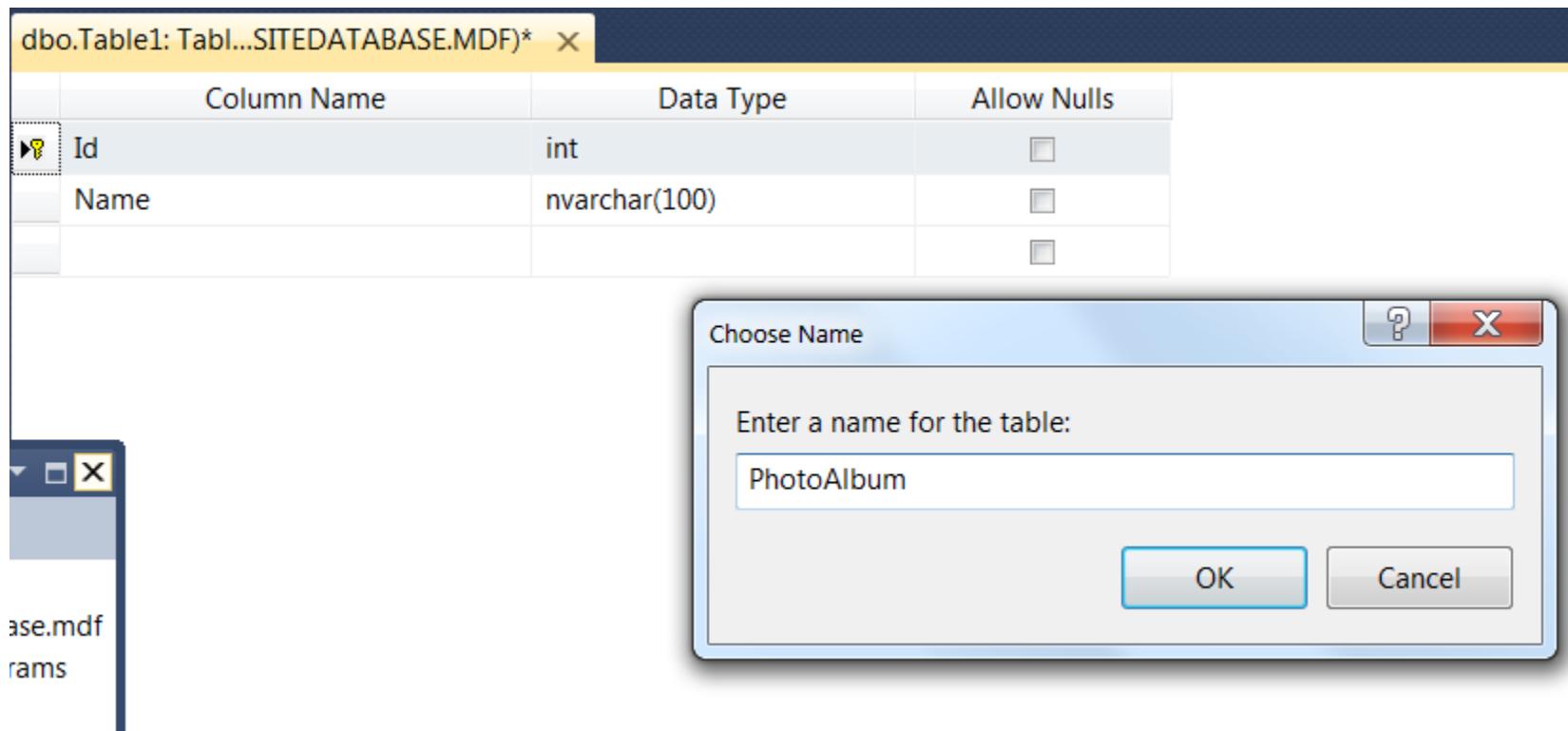
Column Properties

Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
To Columnsort	No

Example

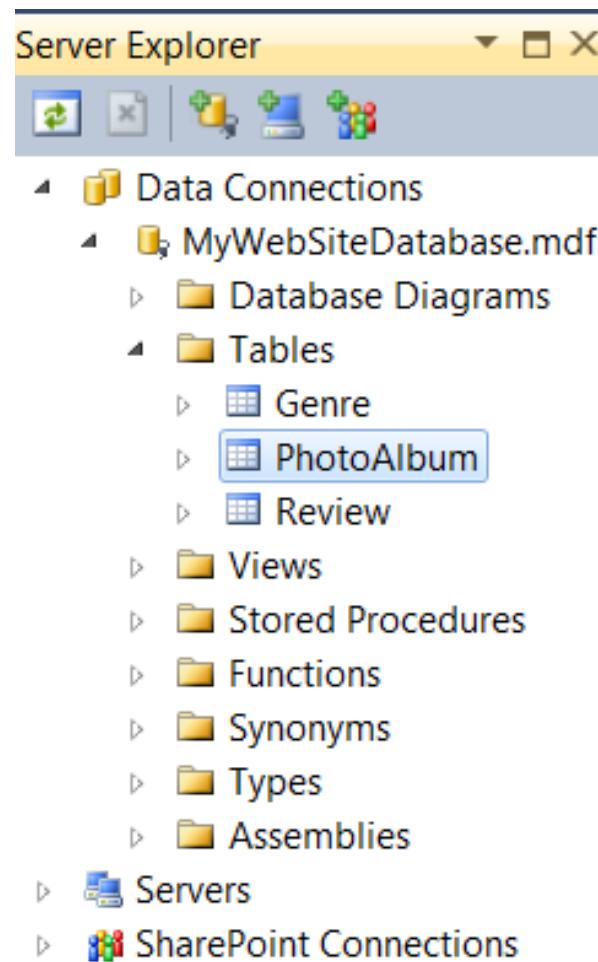
EntityDataSource

- Save as **PhotoAlbum**



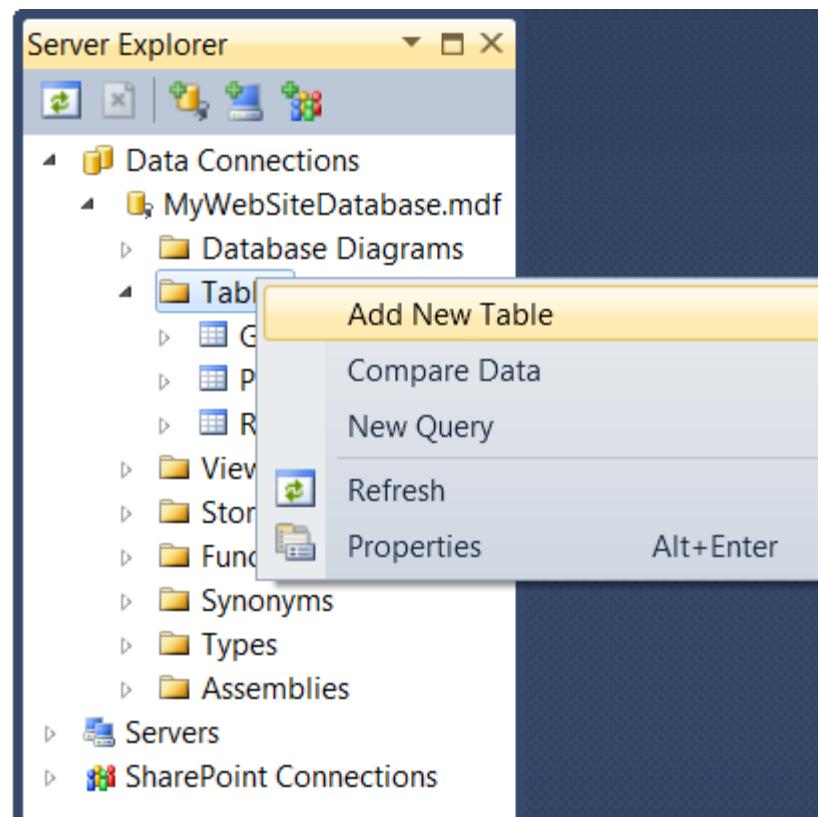
Example

EntityDataSource



Example

EntityDataSource



dbo.Table2: Tabl...SITEDATABASE.MDF)* X

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Description	nvarchar(300)	<input type="checkbox"/>
ToolTip	nvarchar(50)	<input type="checkbox"/>
ImageUrl	nvarchar(200)	<input type="checkbox"/>
PhotoAlbumId	int	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

dbo.Table2: Tabl...SITEDATABASE.MDF)* X

Column Name	Data Type	Allow Nulls
Set Primary Key		<input type="checkbox"/>
Insert Column		<input type="checkbox"/>
Delete Column		<input type="checkbox"/>
Relationships...		<input type="checkbox"/>
Indexes/Keys...		<input type="checkbox"/>
Fulltext Index		<input type="checkbox"/>

dbo.Table2: Tabl...SITEDATABASE.MDF)* X

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Description	nvarchar(300)	<input type="checkbox"/>
ToolTip	nvarchar(50)	<input type="checkbox"/>
ImageUrl	nvarchar(200)	<input type="checkbox"/>
PhotoAlbumId	int	<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>



Example

EntityDataSource

dbo.Picture: Tabl...SITEDATABASE.MDF)* X

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Description	nvarchar(300)	<input type="checkbox"/>
ToolTip	nvarchar(50)	<input type="checkbox"/>
ImageUrl	nvarchar(200)	<input type="checkbox"/>
PhotoAlbumId	int	<input type="checkbox"/>
		<input type="checkbox"/>

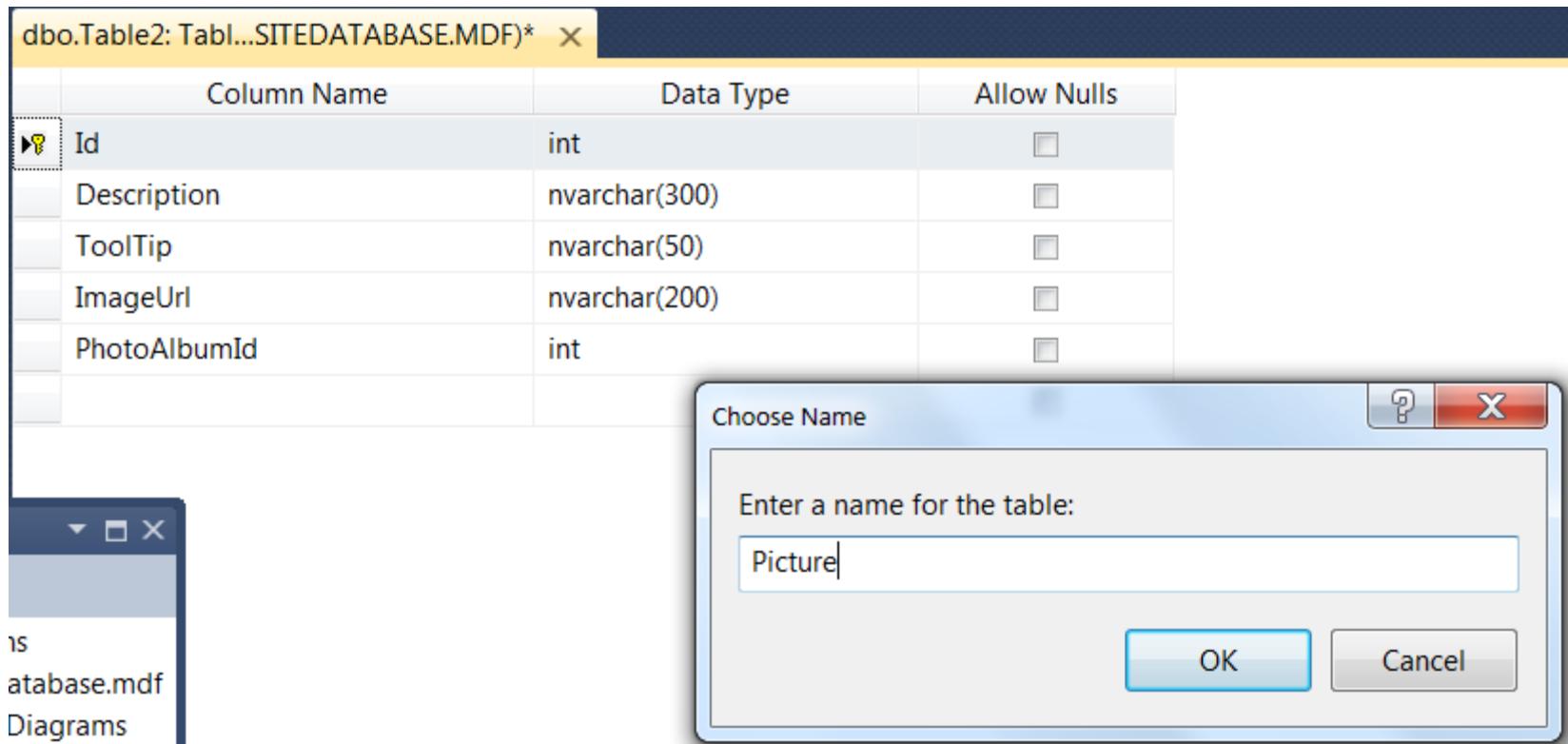
Column Properties

Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
Is Identity	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes

Example

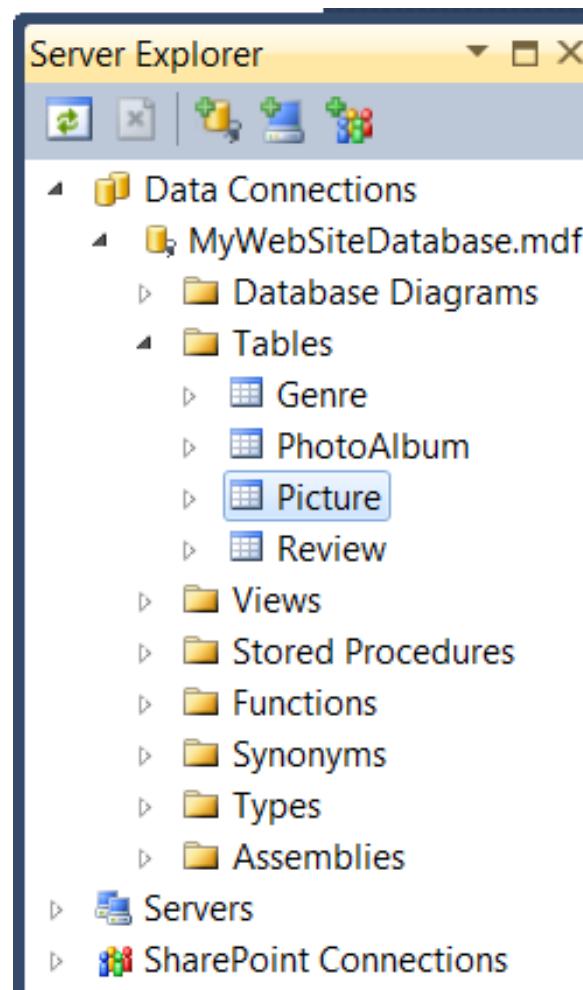
EntityDataSource

- Save as **Picture**



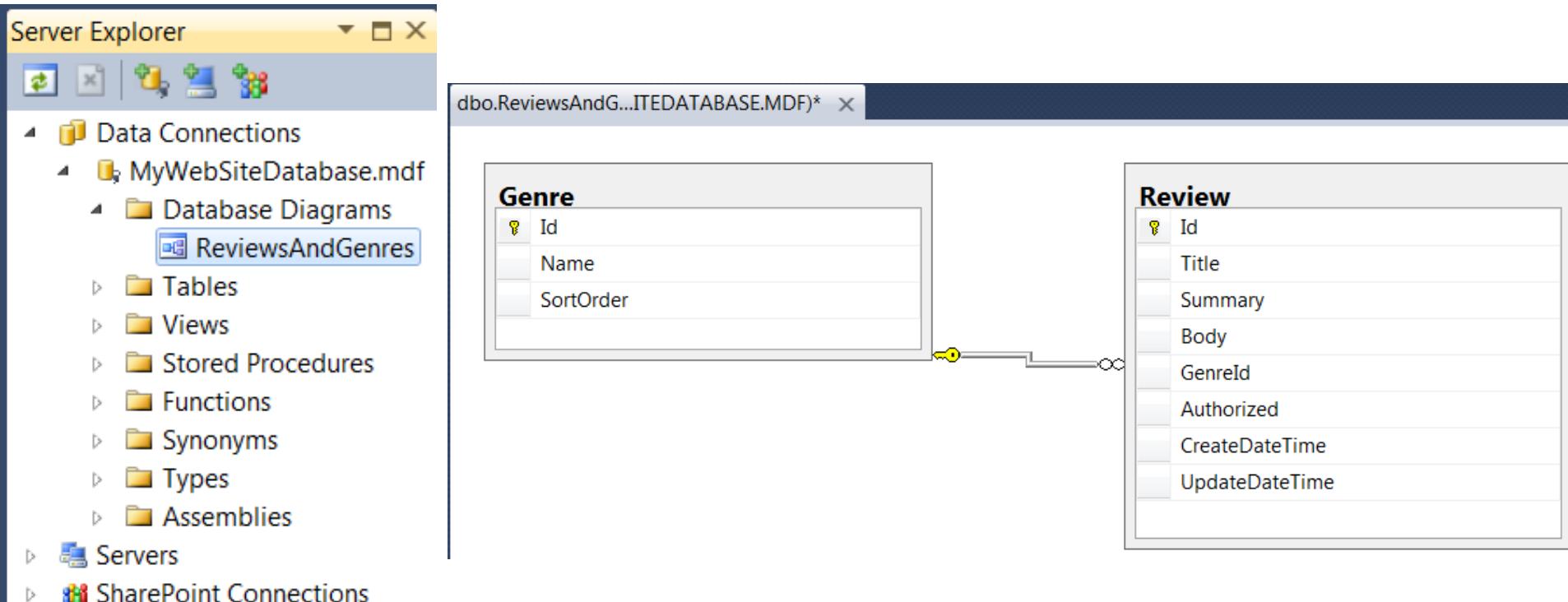
Example

EntityDataSource



Example

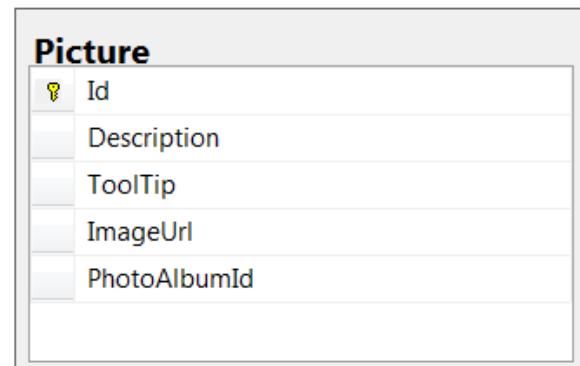
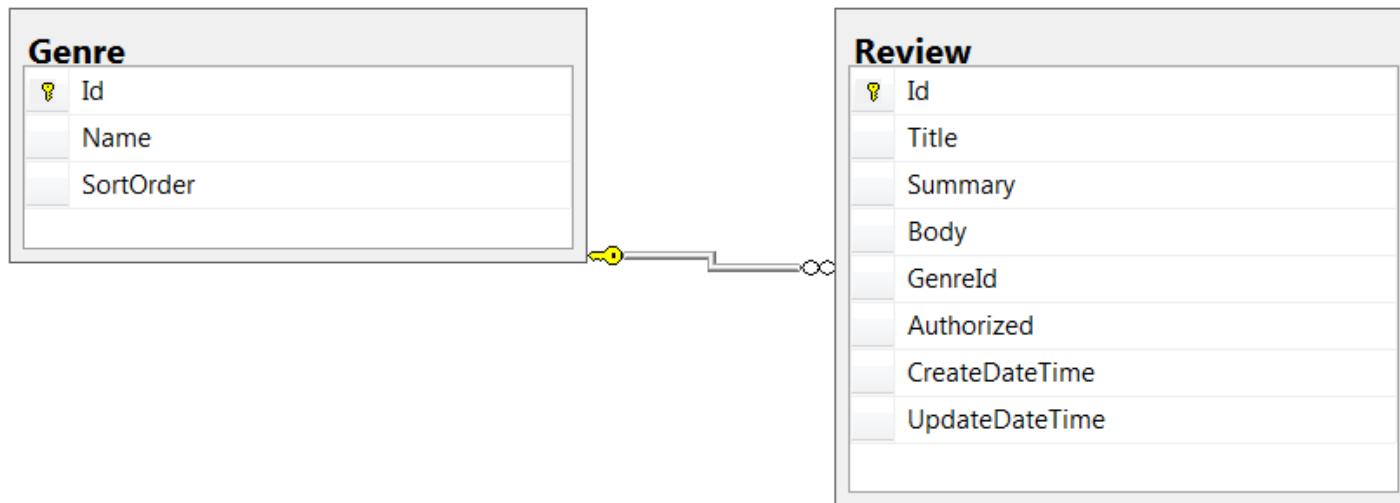
EntityDataSource



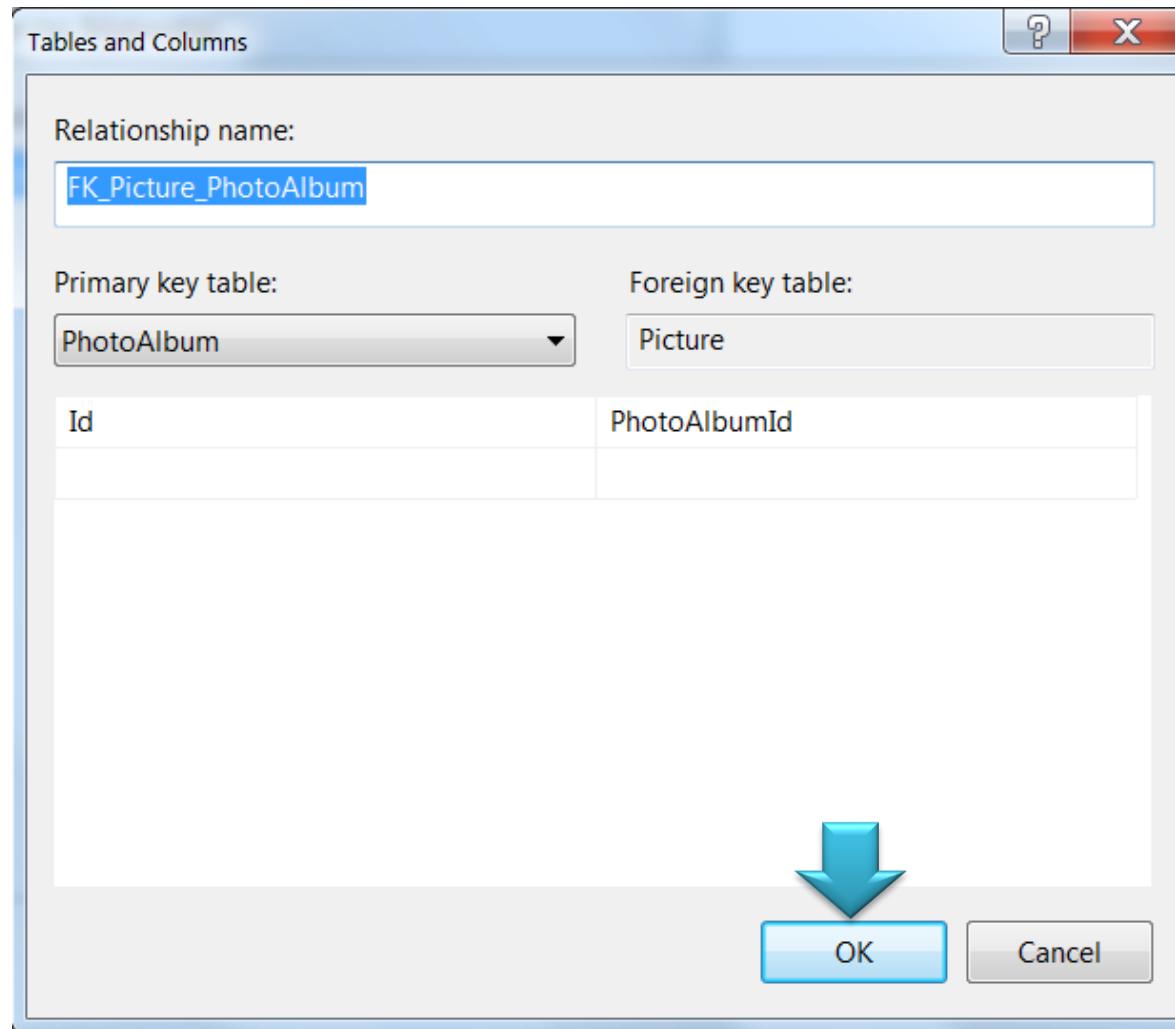
Example

EntityDataSource

- Drag **PhotoAlbum** and **Picture**

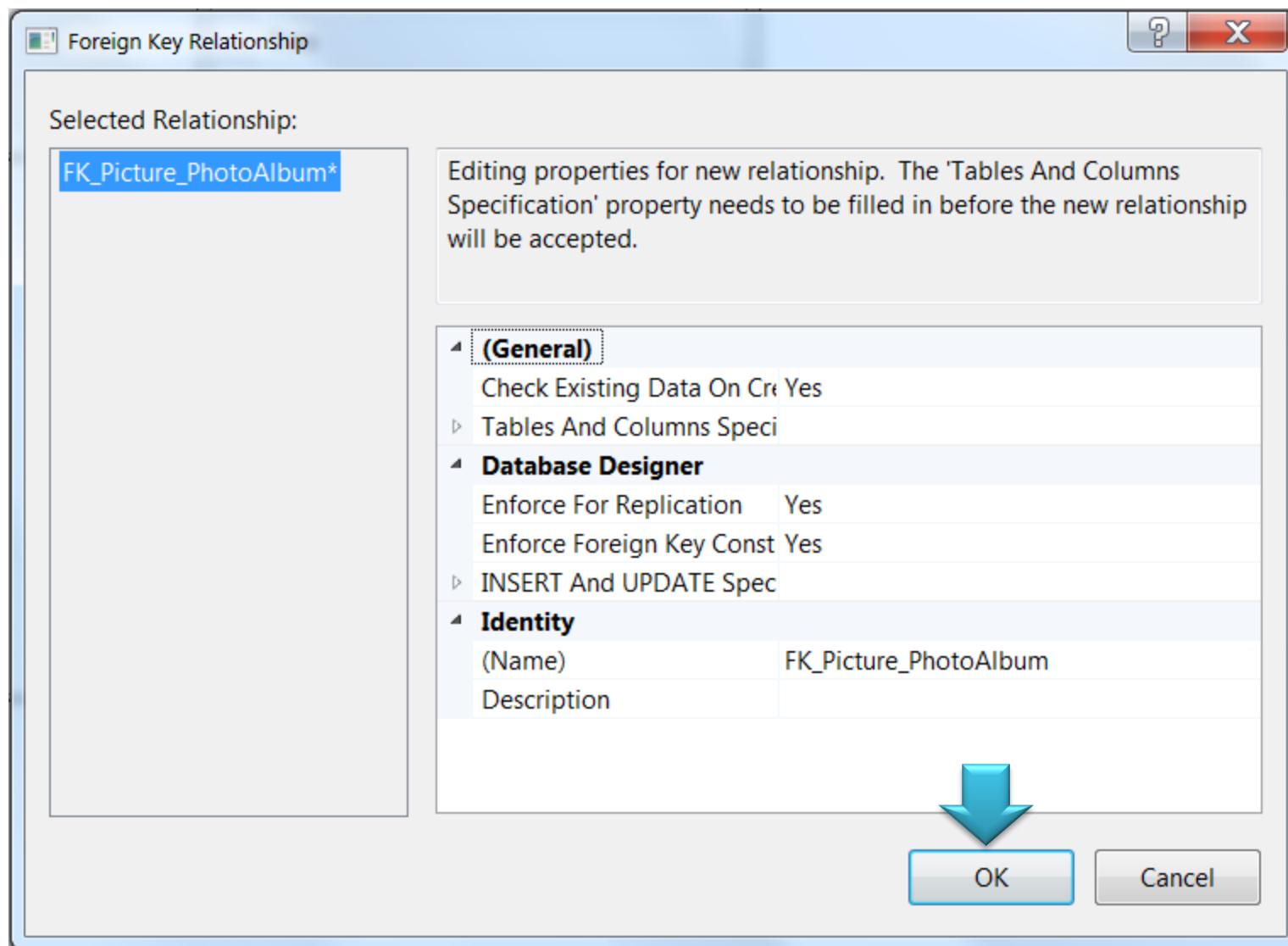


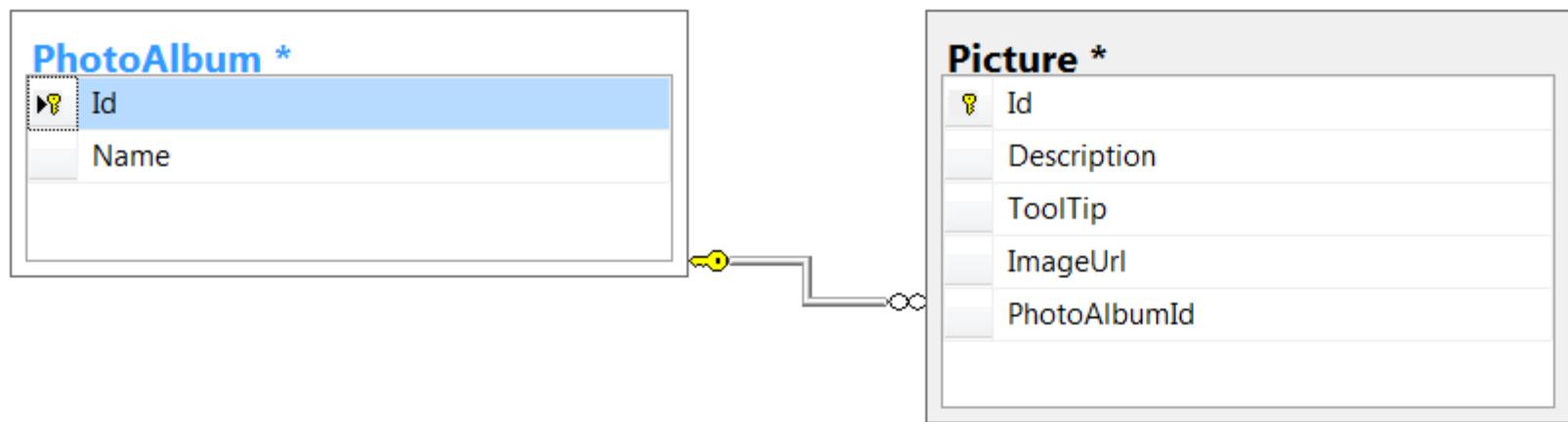
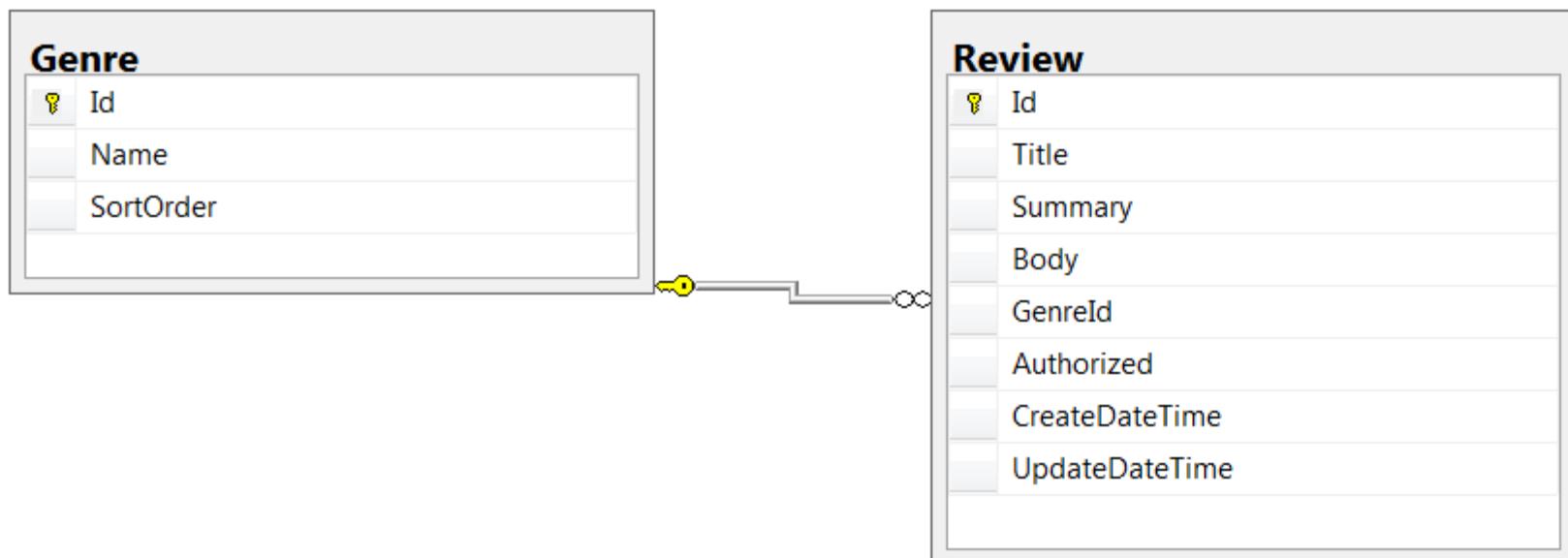
- Drag the **Id** column from the **PhotoAlbum** table onto the **PhotoAlbumId** column of the **Picture** table.



Example

EntityDataSource

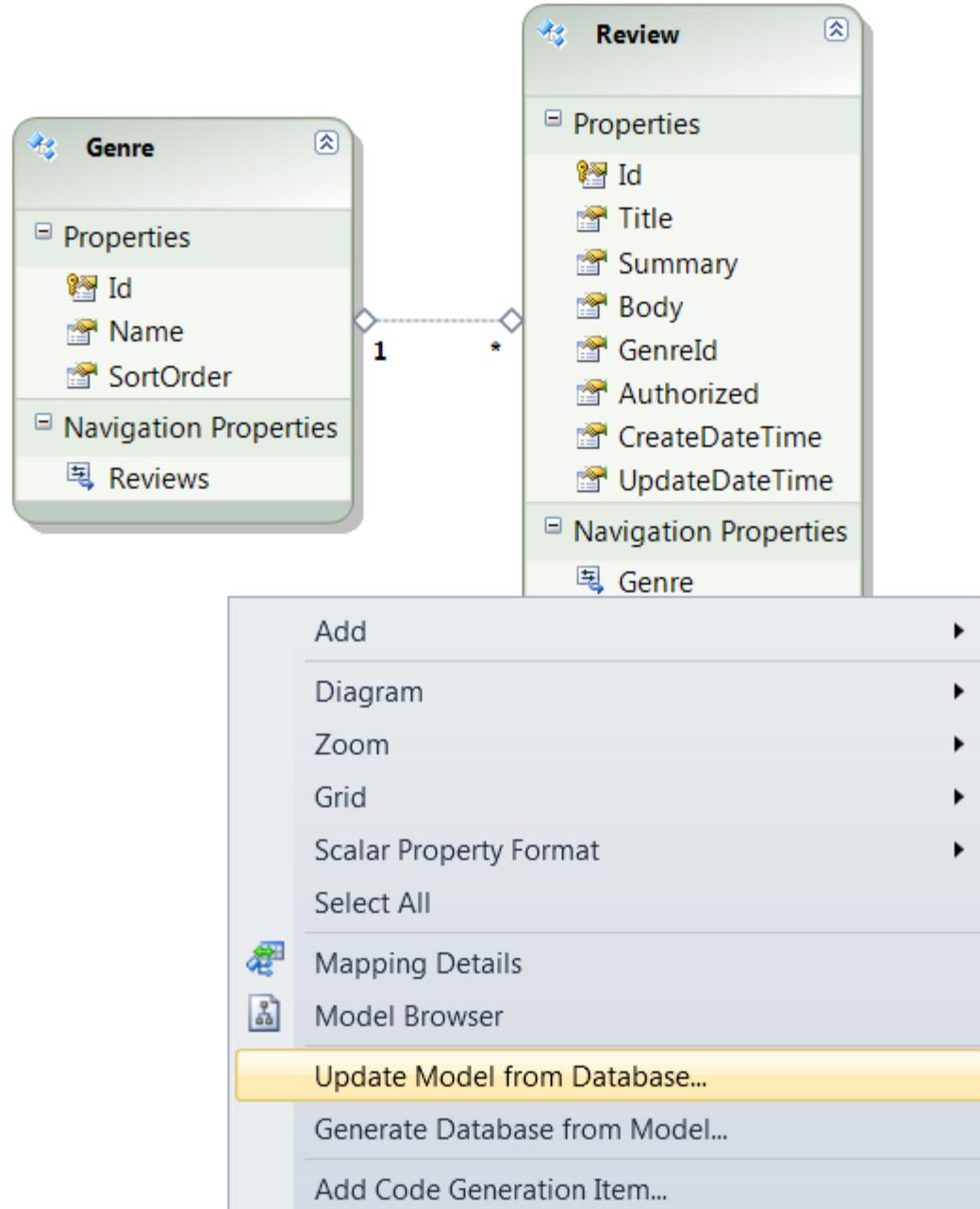




Example

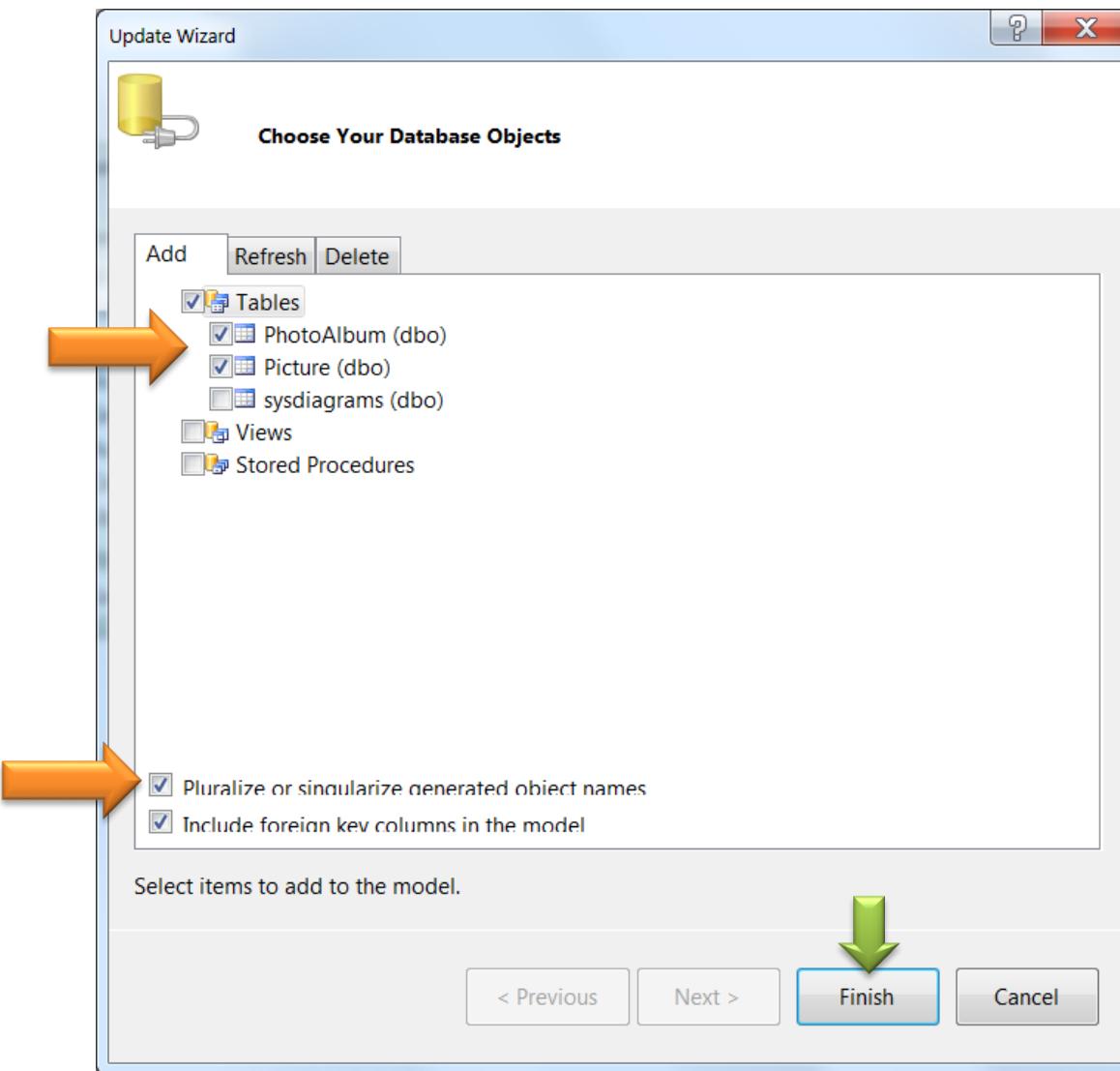
EntityDataSource

- Open **App_Code > MyWebSite.edmx**
- Right-click the empty area:
 - **Update Model from Database**



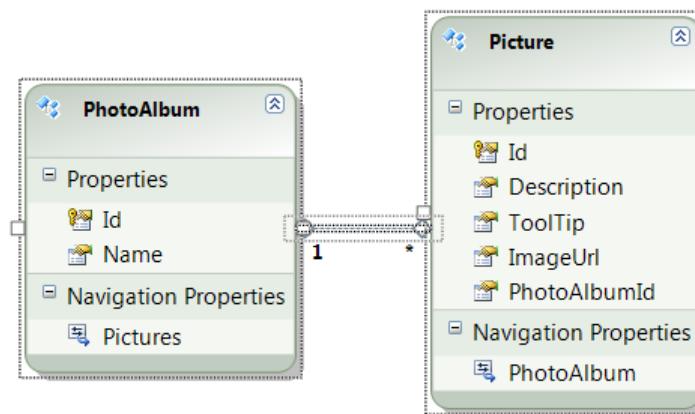
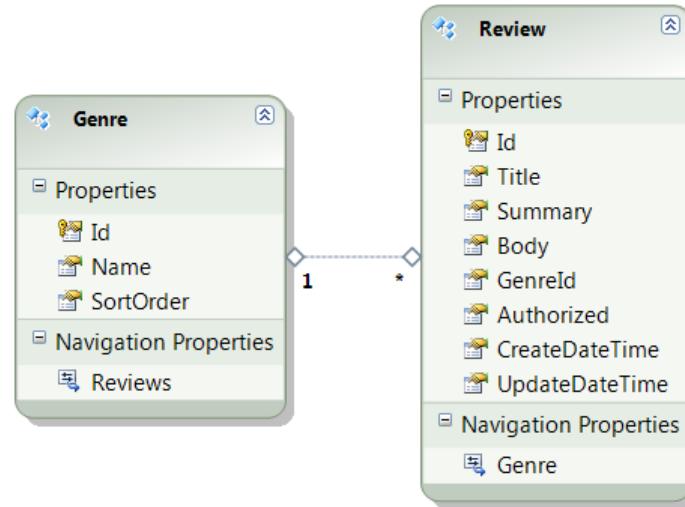
Example

EntityDataSource



Example

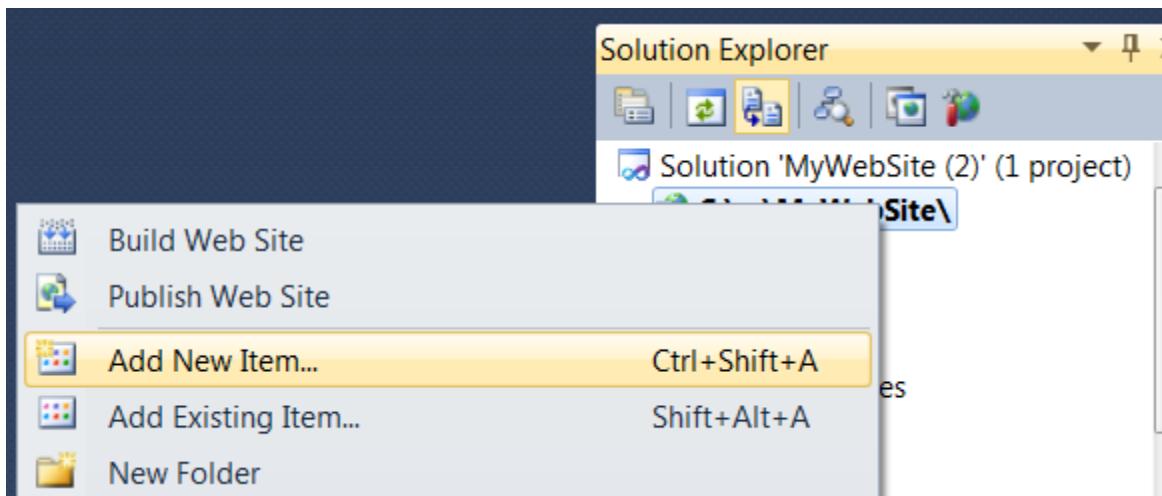
EntityDataSource



Example

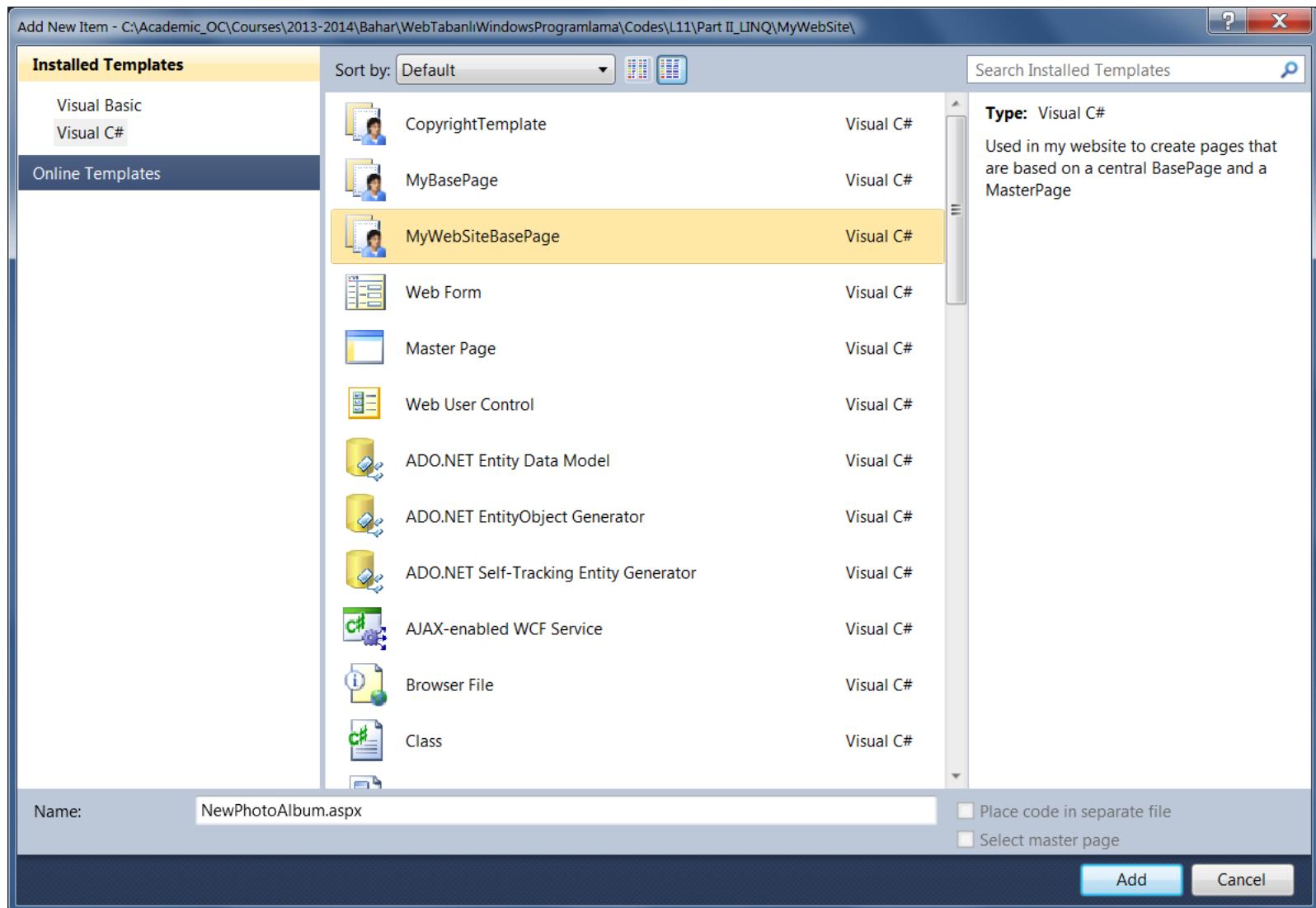
EntityDataSource

- Add New Item to the project



Example

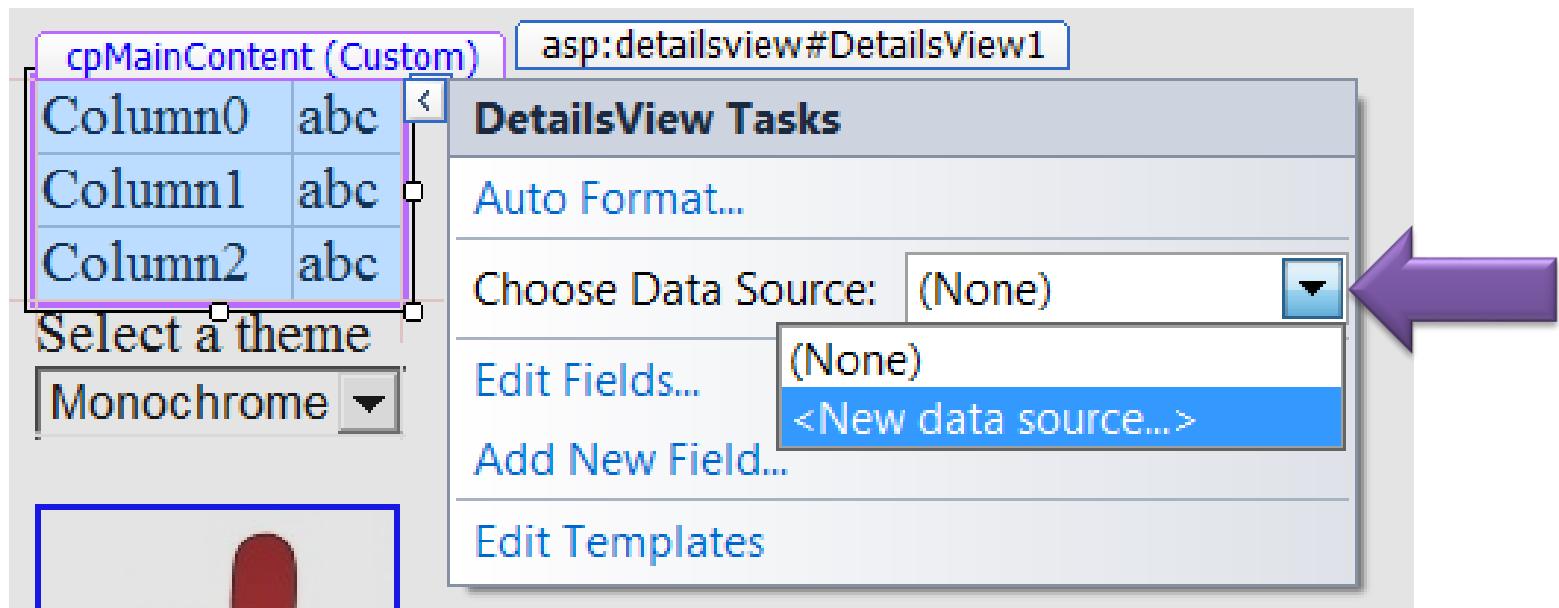
EntityDataSource



Example

EntityDataSource

- Title = Create New Photo Album
- In Design View → Drag a DetailsView control





Choose a Data Source Type

Where will the application get data from?



Access
Database



Database



Entity



LINQ



Object



Site Map



XML File

Connect to an ADO.NET Entity Framework Model.

Specify an ID for the data source:

EntityDataSource1

OK

Cancel



Configure ObjectContext

ConnectionString:

Named Connection

MyWebSiteDatabaseEntities

Connection String

DefaultContainerName:

MyWebSiteDatabaseEntities

< Previous

Next >

Finish

Cancel



Configure Data Selection

EntitySetName:

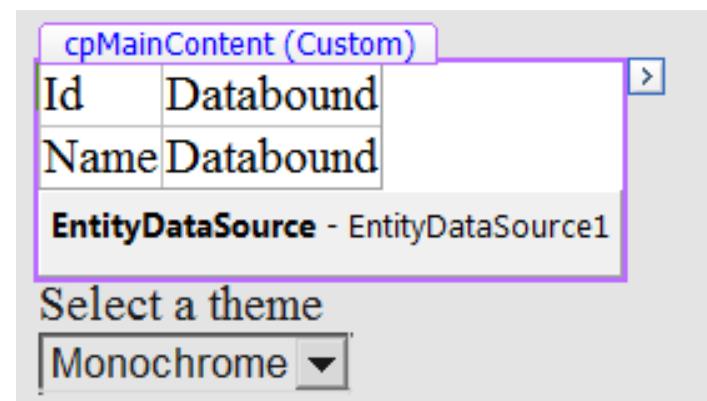
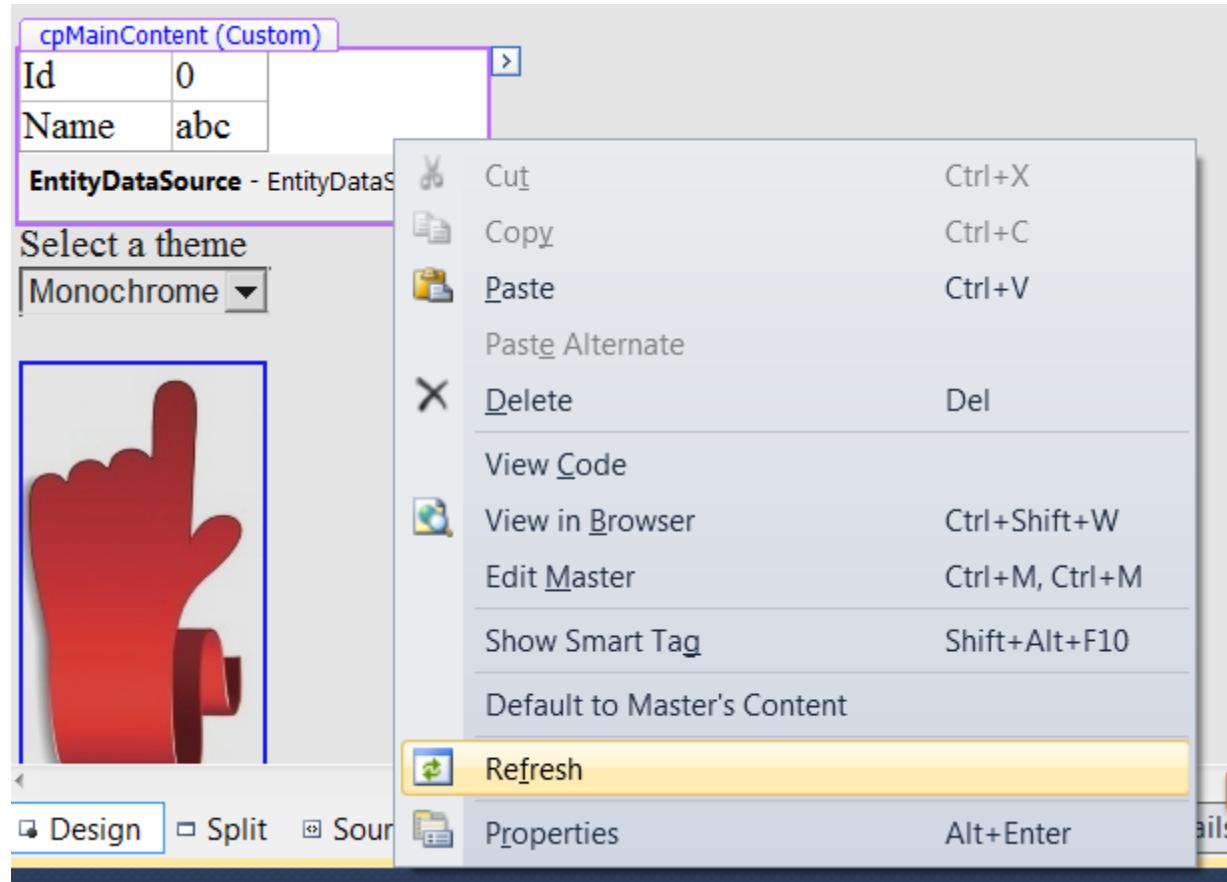
EntityTypeFilter:

Select:

- Select All (Entity Value)
- Id
- Name

- Enable automatic inserts
- Enable automatic updates
- Enable automatic deletes

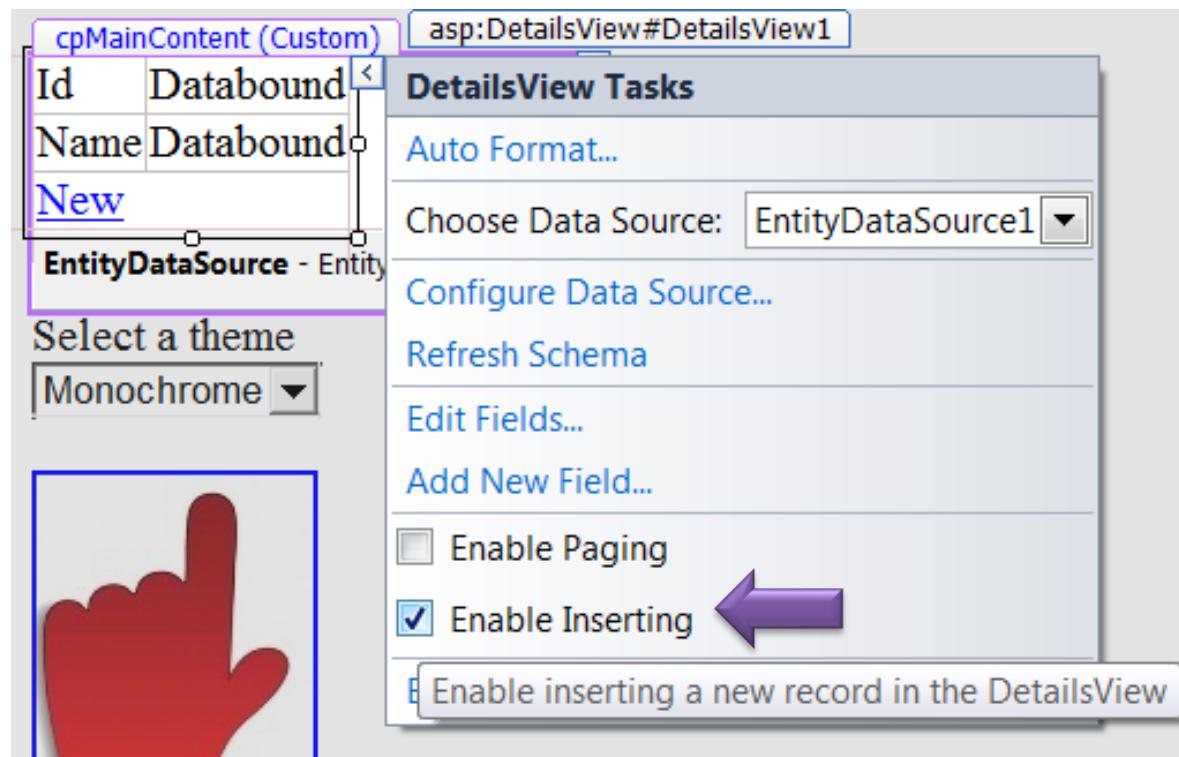
[< Previous](#)[Next >](#)[Finish](#)[Cancel](#)



Example

EntityDataSource

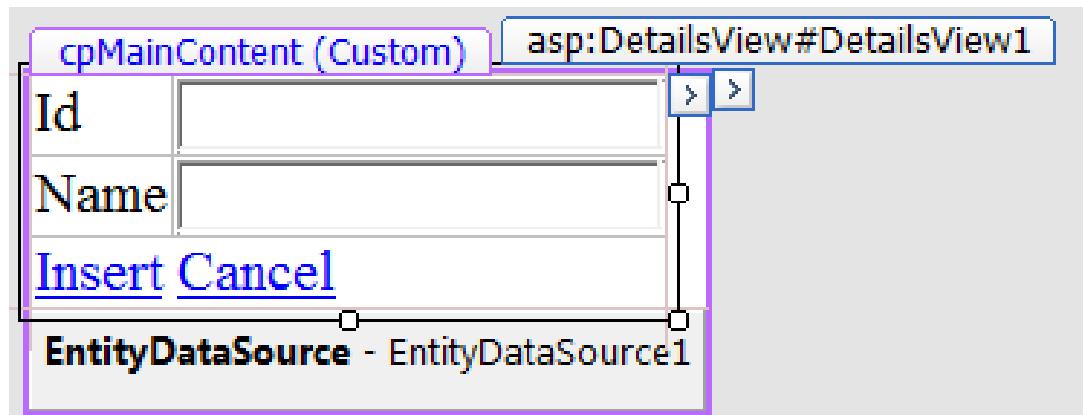
- For the **DetailsView** control



Example

EntityDataSource

- Properties of the **DetailsView** control:
 - **DefaultMode = Insert**



Example

EntityDataSource

- In Source View:

```
<asp:BoundField DataField="Id" HeaderText="Id" ReadOnly="True"  
    SortExpression="Id" />
```

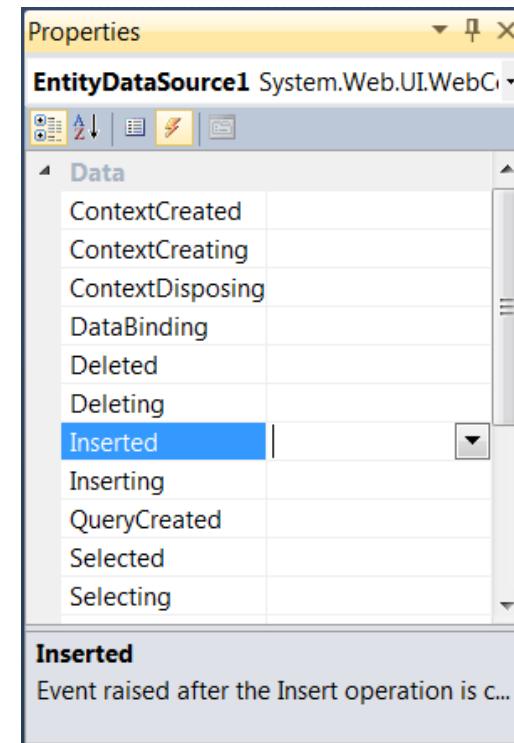
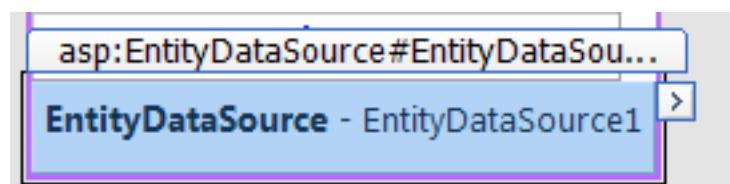


```
<asp:BoundField DataField="Id" HeaderText="Id" ReadOnly="True"  
    SortExpression="Id" InsertVisible="false"/>
```

Example

EntityDataSource

- In Design View:
 - Double-click **Inserted** event of **EntityDataSource**



Example

EntityDataSource

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MyWebSiteDatabaseModel;

public partial class _NewPhotoAlbum : BasePage
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void EntityDataSource1_Inserted(object sender, EntityDataSourceChangedEventArgs e)
    {
        PhotoAlbum myPhotoAlbum = (PhotoAlbum)e.Entity;
        Response.Redirect(string.Format("ManagePhotoAlbum.aspx?PhotoAlbumId={0}", myPhotoAlbum.Id.ToString()));
    }
}
```

Example

EntityDataSource

- View **NewPhotoAlbum.aspx** in browser.



ListView Templates

TEMPLATE	DESCRIPTION
<LayoutTemplate>	Serves as the container for the control. It enables you to define a location where the individual data items are placed. The data items, presented through the <code>ItemTemplate</code> and <code>AlternatingItemTemplate</code> , are then added as children of this container.
<ItemTemplate> <AlternatingItemTemplate>	Define the read-only mode for the control. When used together, they enable you to create a “zebra effect,” where odd and even rows have a different appearance (usually a different background color).
<SelectedItemTemplate>	Enables you to define the look and feel of the currently active, or selected, item.
<InsertItemTemplate> <EditItemTemplate>	These two templates enable you to define the user interface for inserting and updating items in the list. You typically place controls like text boxes, drop-down lists, and other server controls in these templates and bind them to the underlying data source.
<ItemSeparatorTemplate>	Defines the markup that is placed between the items in the list. Useful if you want to add a line, an image, or any other markup between the items.
<EmptyDataTemplate>	Displayed when the control has no data to display. You can add text or other markup and controls to it to tell your users there's no data to display.
<GroupTemplate> <GroupSeparatorTemplate> <EmptyItemTemplate>	Used in advanced presentation scenarios where data can be presented in different groups.

ListView Properties

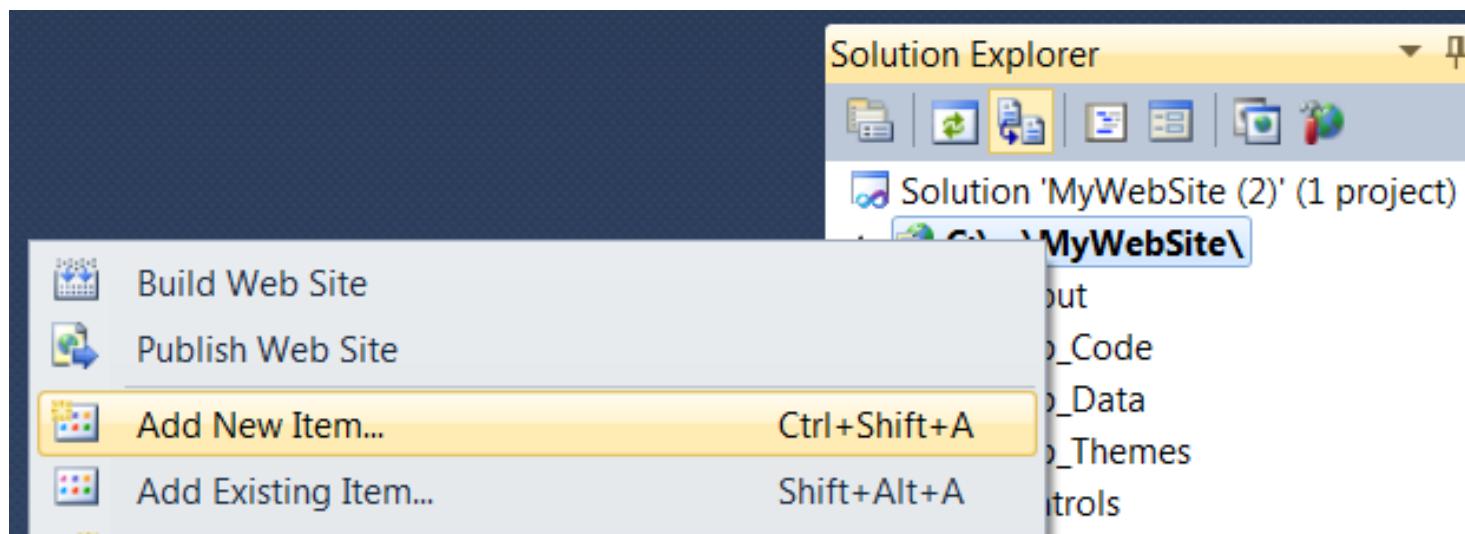
PROPERTY	DESCRIPTION
ItemPlaceholderID	The ID of a server-side control placed within the <code>LayoutTemplate</code> . The control referred to by this property is <i>replaced</i> by all the repeated data items when the control is displayed onscreen. It can be a true server control like an <code><asp:PlaceHolder></code> or a simple HTML element with a valid ID and its <code>runat</code> attribute set to <code>server</code> (for example, <code><ul runat="server" id="MainList"></code>). If you don't set this property, ASP.NET tries to find a control with an ID of <code>itemPlaceholder</code> and uses that control instead.
DataSourceID	The ID of a data source control on the page, such as an <code>EntityDataSource</code> or a <code>SqlDataSource</code> control.
InsertItemPosition	The enumeration for this property contains three values — <code>None</code> , <code>FirstItem</code> , and <code>LastItem</code> — to determine the position of the <code>InsertItemTemplate</code> : either at the beginning or end of the list, or not visible at all.

For more → [http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listview\(VS.100\).aspx](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.listview(VS.100).aspx)

Example

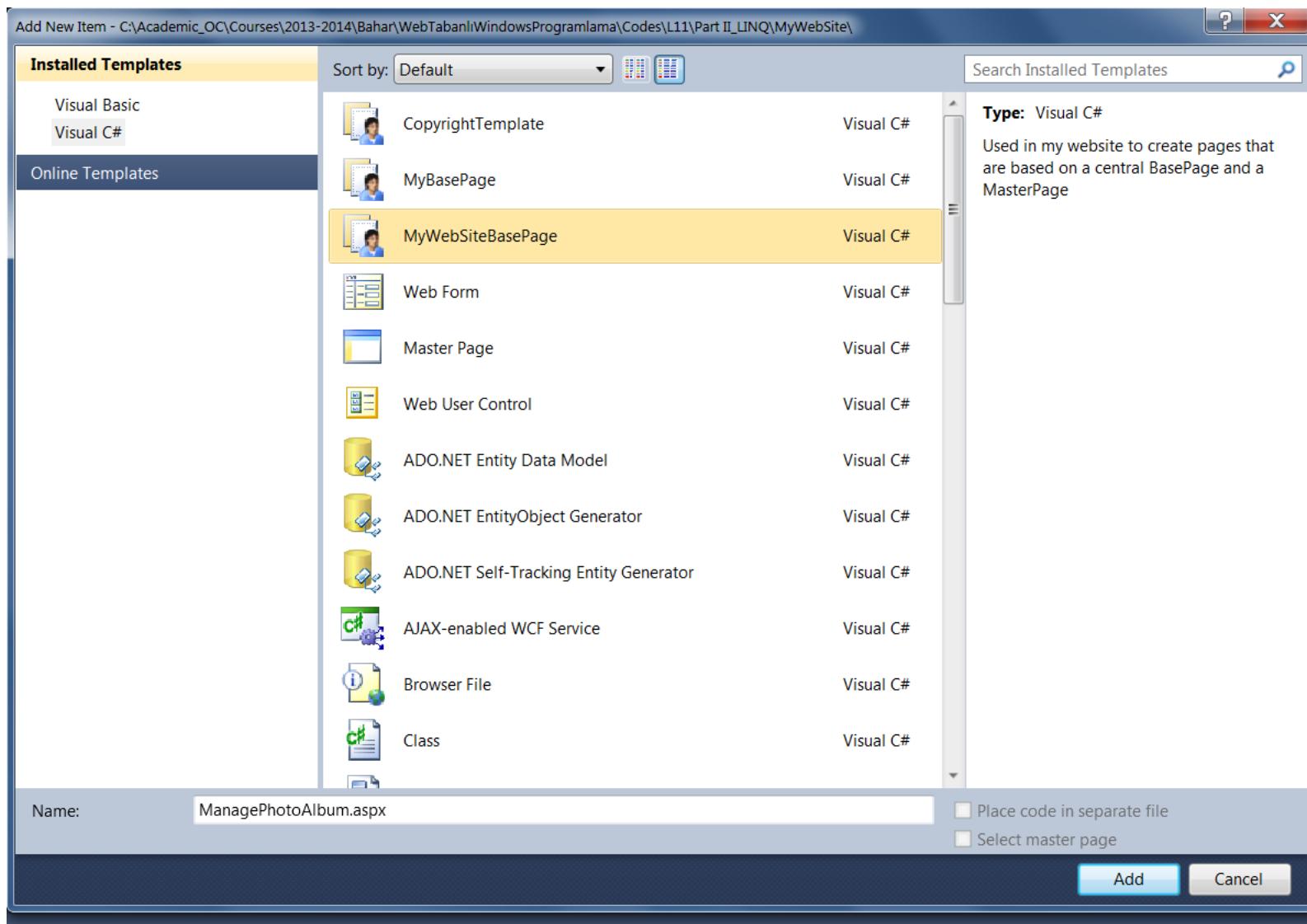
ListView

- Add New Item



Example

ListView



Example

ListView

- **Title = Manage Photo Album**
- Drag a **ListView** control





Choose a Data Source Type

Where will the application get data from?



Access
Database



Database



Entity



LINQ



Object



Site Map



XML File

Connect to an ADO.NET Entity Framework Model.

Specify an ID for the data source:

EntityDataSource1

OK

Cancel



Configure ObjectContext

ConnectionString:

Named Connection

MyWebSiteDatabaseEntities

Connection String

DefaultContainerName:

MyWebSiteDatabaseEntities

< Previous

Next >

Finish

Cancel



Configure Data Selection

EntitySetName:

▼

EntityTypeFilter:

▼

Select:

 Select All (Entity Value) Id Description ToolTip ImageUrl PhotoAlbumId Enable automatic inserts Enable automatic updates Enable automatic deletes

< Previous

Next >

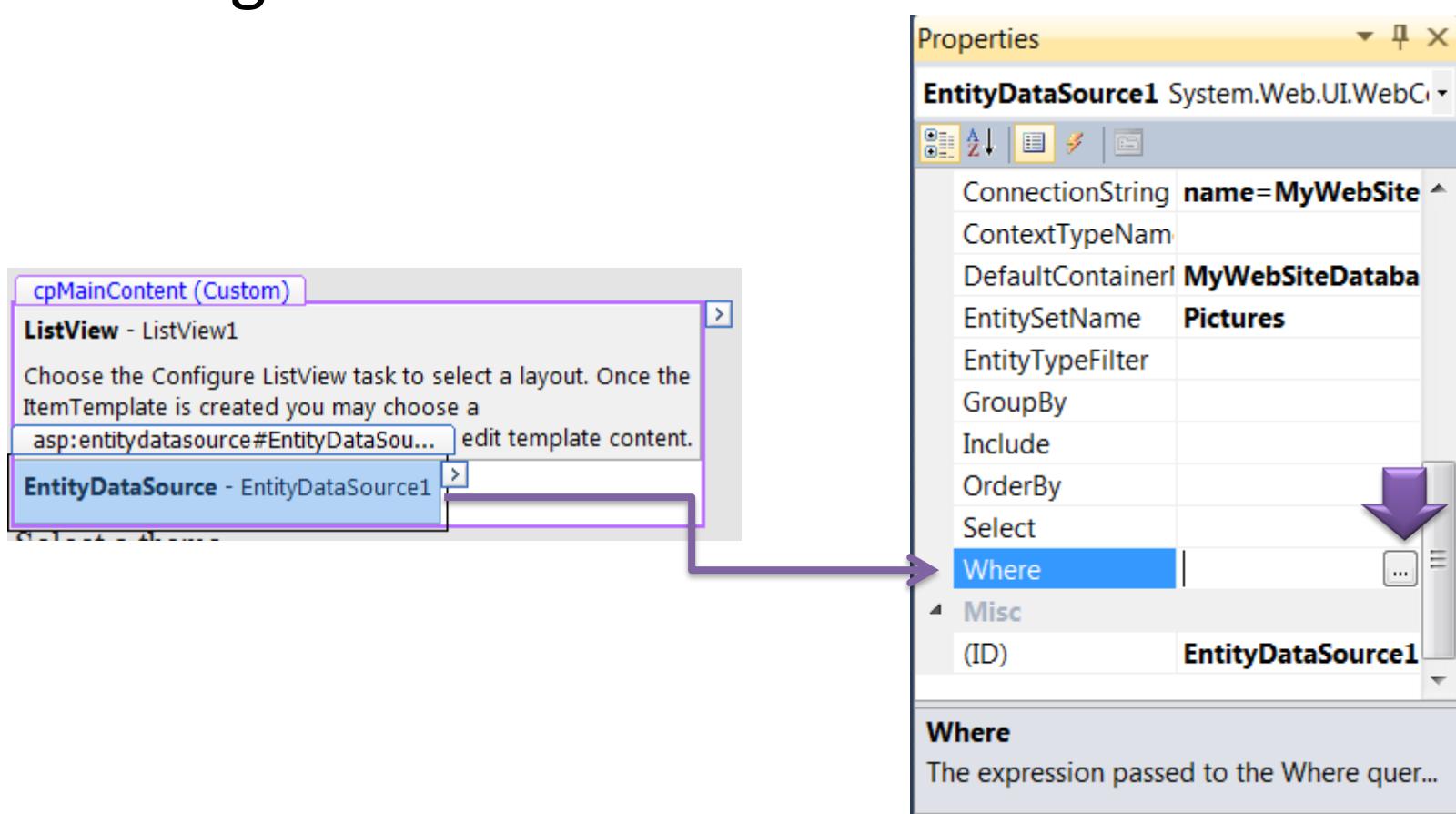
Finish

Cancel

Example

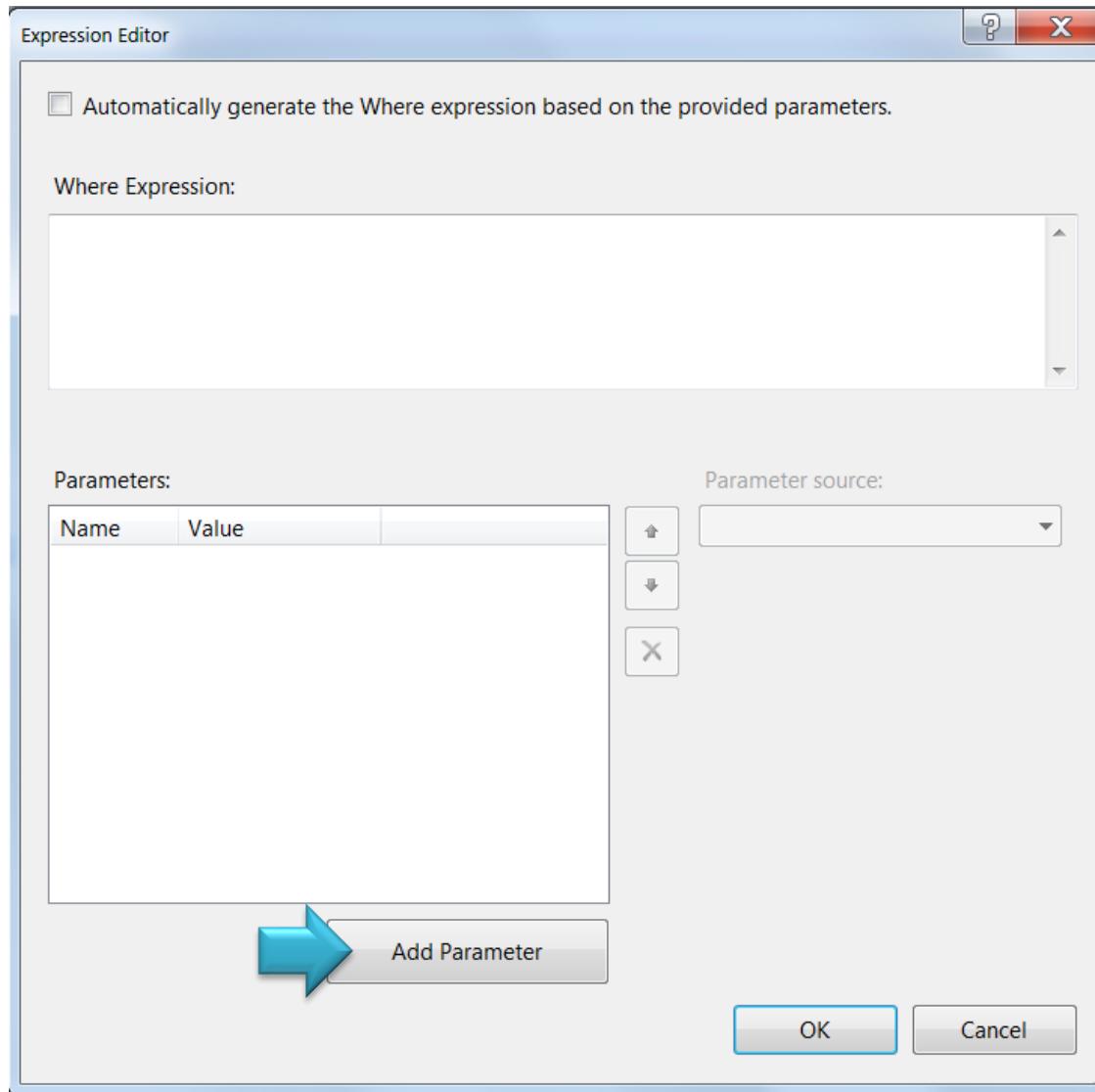
ListView

- In Design View:



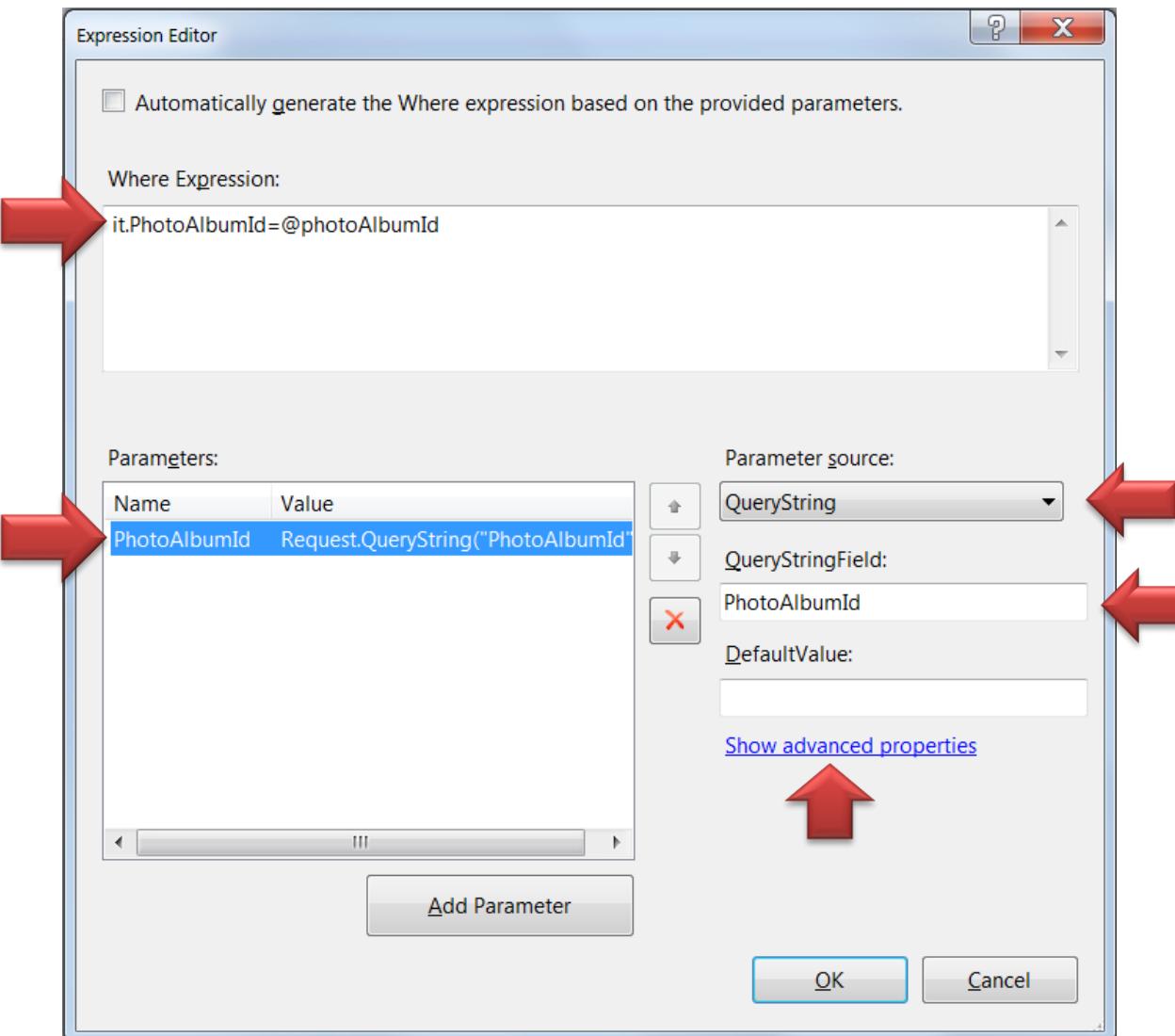
Example

ListView



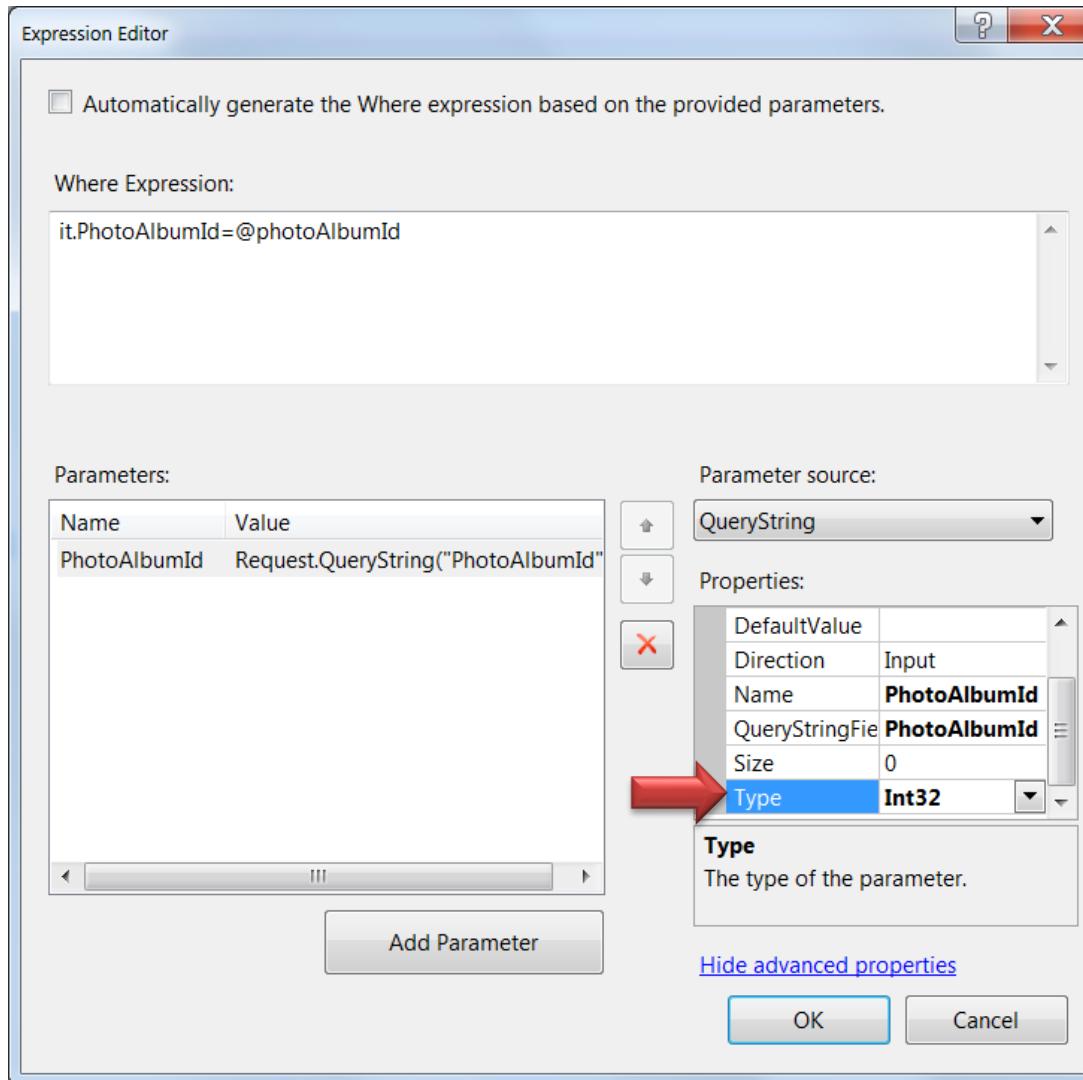
Example

ListView



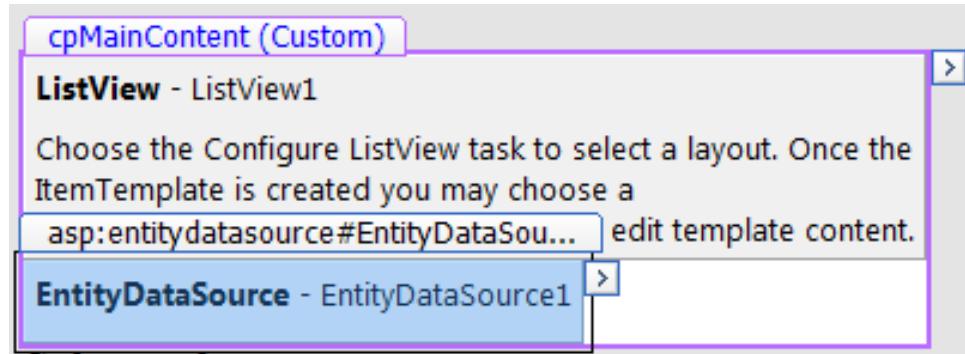
Example

ListView



Example

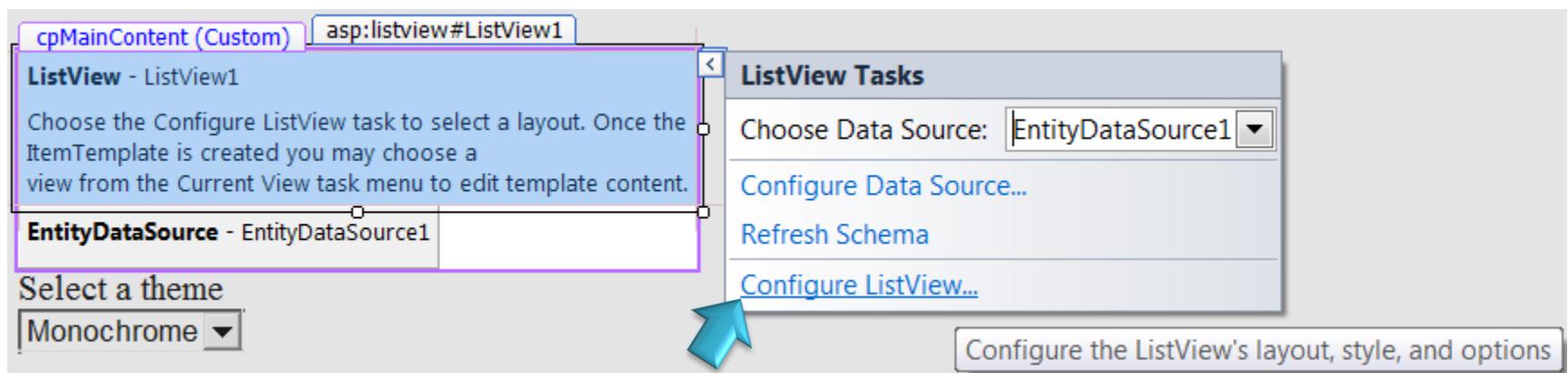
ListView



```
<asp:EntityDataSource ID="EntityDataSource1" runat="server"
    ConnectionString="name=MyWebSiteDatabaseEntities"
    DefaultContainerName="MyWebSiteDatabaseEntities" EnableDelete="True"
    EnableFlattening="False" EnableInsert="True" EntitySetName="Pictures"
    Where="it.PhotoAlbumId=@photoAlbumId">
    <WhereParameters>
        <asp:QueryStringParameter Name="PhotoAlbumId" QueryStringField="PhotoAlbumId"
            Type="Int32" />
    </WhereParameters>
</asp:EntityDataSource>
```

Example

ListView



Configure ListView

Select a Layout:

- Grid
Tiled
Bulleted List
Flow
Single Row

Preview:

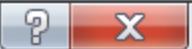
Options:

- Enable Editing
 - Enable Inserting
 - Enable Deleting
 - Enable Paging
 - Enable Dynamic

OK

Cancel

Configure ListView



Select a Layout:

Grid
Tiled
Bulleted List

Flow
Single Row

Select a Style:

(No Formatting)
Colorful
Professional
Blues

Options:

- Enable Editing
 Enable Inserting
 Enable Deleting
 Enable Paging
 Enable Dynamic Data

Preview:

- Id: Databound
Description: Databound
ToolTip: Databound
ImageUrl: Databound
PhotoAlbumId: Databound
PhotoAlbum: Databound

Delete

- Id: Databound
Description: Databound
ToolTip: Databound
ImageUrl: Databound
PhotoAlbumId: Databound
PhotoAlbum: Databound

Delete

- Id: Databound
Description: Databound
ToolTip: Databound
ImageUrl: Databound
PhotoAlbumId: Databound

OK

Cancel

Example

ListView

cpMainContent (Custom) | asp:listview#ListView1 | Login_V1.aspx

The screenshot shows the Visual Studio IDE with a web form named 'Login_V1.aspx'. A 'ListView' control is placed on the page, with its ID set to 'ListView1'. The 'Properties' window on the right is open, showing the 'ListView Tasks' section. Under 'Current View: Runtime View', the 'DataSourceID' property is set to 'EntityDataSource1'. The 'Misc' section shows the 'EditIndex' property is set to '-1'. The 'ListView1' item template displays two items from the data source:

- Id: 0
Description: abc
ToolTip: abc
ImageUrl: abc
PhotoAlbumId: 0
PhotoAlbum: abc
[Delete](#)
- Id: 1
Description: abc
ToolTip: abc
ImageUrl: abc
PhotoAlbumId: 1
PhotoAlbum: abc

Example

ListView

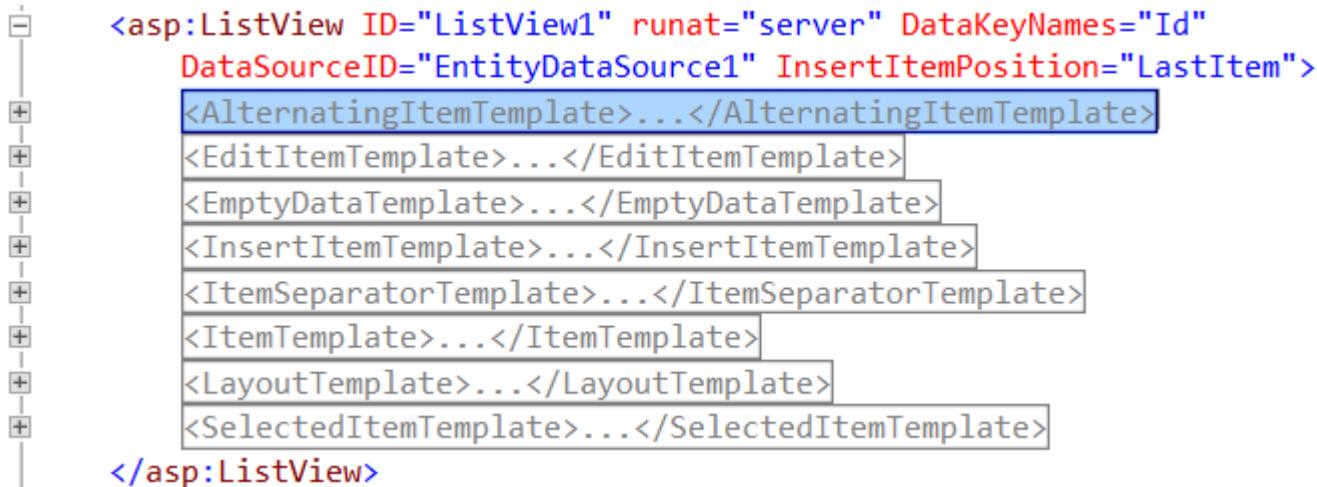
- In Source View:

```
<asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
    DataSourceID="EntityDataSource1" InsertItemPosition="LastItem">
    <AlternatingItemTemplate>...</AlternatingItemTemplate>
    <EditItemTemplate>...</EditItemTemplate>
    <EmptyDataTemplate>...</EmptyDataTemplate>
    <InsertItemTemplate>...</InsertItemTemplate>
    <ItemSeparatorTemplate>...</ItemSeparatorTemplate>
    <ItemTemplate>...</ItemTemplate>
    <LayoutTemplate>...</LayoutTemplate>
    <SelectedItemTemplate>...</SelectedItemTemplate>
</asp:ListView>
```

Example

ListView

- Remove

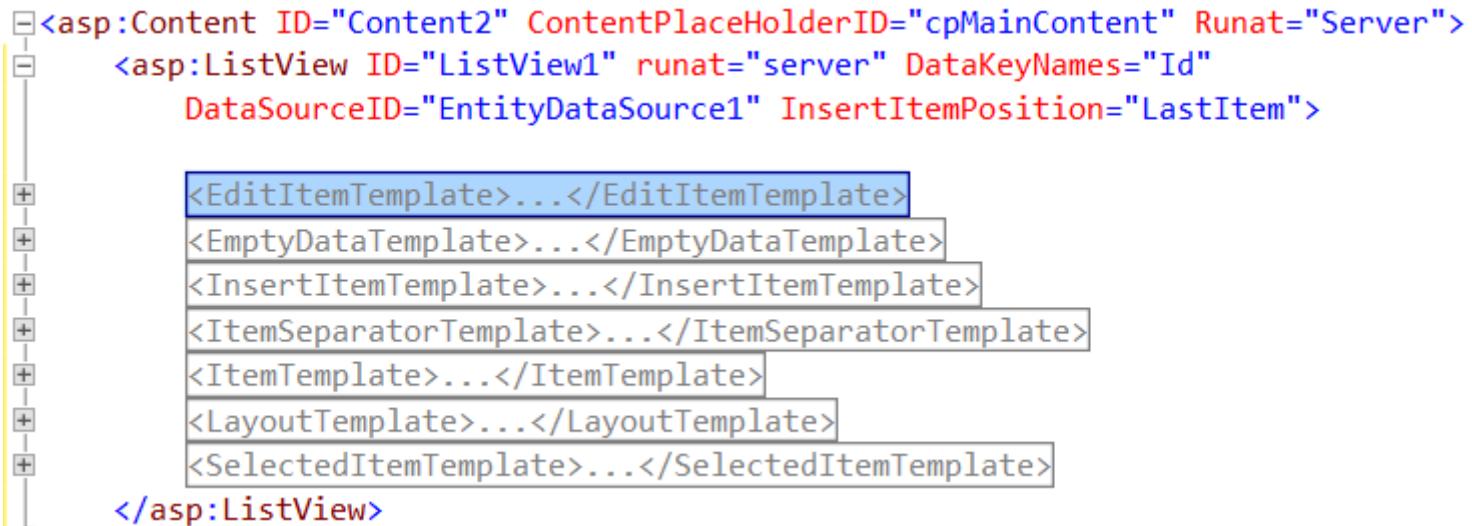


```
<asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
    DataSourceID="EntityDataSource1" InsertItemPosition="LastItem">
    <AlternatingItemTemplate>...</AlternatingItemTemplate>
    <EditItemTemplate>...</EditItemTemplate>
    <EmptyDataTemplate>...</EmptyDataTemplate>
    <InsertItemTemplate>...</InsertItemTemplate>
    <ItemSeparatorTemplate>...</ItemSeparatorTemplate>
    <ItemTemplate>...</ItemTemplate>
    <LayoutTemplate>...</LayoutTemplate>
    <SelectedItemTemplate>...</SelectedItemTemplate>
</asp:ListView>
```

Example

ListView

- Remove



The image shows a portion of an ASP.NET page's source code. A yellow vertical bar on the left indicates the current line of focus. The code is as follows:

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
        DataSourceID="EntityDataSource1" InsertItemPosition="LastItem">

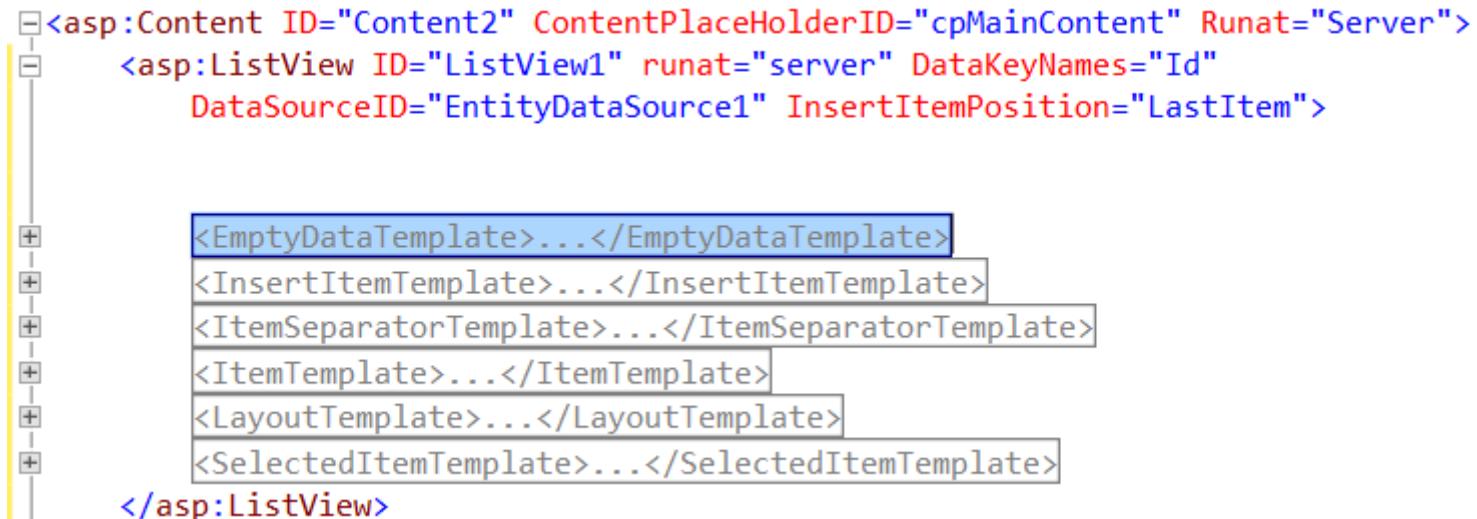
        <EditItemTemplate>...</EditItemTemplate>
        <EmptyDataTemplate>...</EmptyDataTemplate>
        <InsertItemTemplate>...</InsertItemTemplate>
        <ItemSeparatorTemplate>...</ItemSeparatorTemplate>
        <ItemTemplate>...</ItemTemplate>
        <LayoutTemplate>...</LayoutTemplate>
        <SelectedItemTemplate>...</SelectedItemTemplate>

    </asp:ListView>
```

Example

ListView

- Remove



The image shows a screenshot of a Visual Studio code editor. On the left, there is a tree view of the page structure. The main content area contains the following ASPX code:

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
        DataSourceID="EntityDataSource1" InsertItemPosition="LastItem">

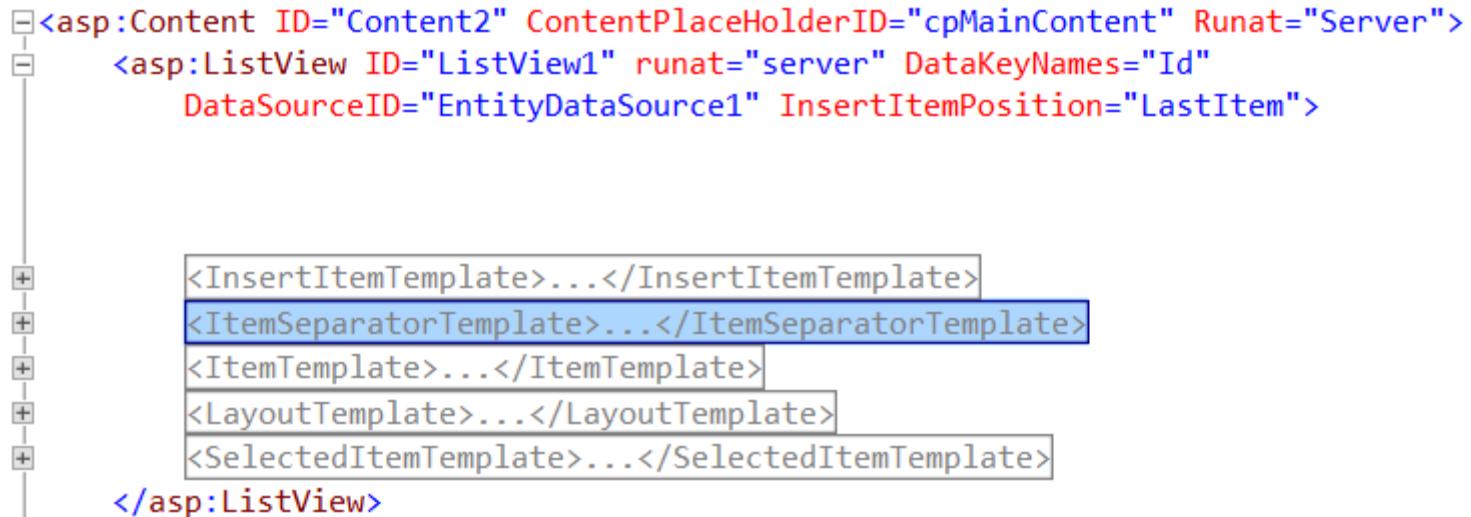
        <EmptyDataTemplate>...</EmptyDataTemplate>
        <InsertItemTemplate>...</InsertItemTemplate>
        <ItemSeparatorTemplate>...</ItemSeparatorTemplate>
        <ItemTemplate>...</ItemTemplate>
        <LayoutTemplate>...</LayoutTemplate>
        <SelectedItemTemplate>...</SelectedItemTemplate>
    </asp:ListView>
```

The code editor highlights several template sections: `<EmptyDataTemplate>...`, `<InsertItemTemplate>...`, `<ItemSeparatorTemplate>...`, `<ItemTemplate>...`, `<LayoutTemplate>...`, and `<SelectedItemTemplate>...`. The entire `<asp:ListView>` tag is also highlighted.

Example

ListView

- Remove



The image shows a portion of an ASPX page from a Visual Studio code editor. On the left, there's a tree view with a Content placeholder node expanded, showing the ListView control and its child nodes. The code itself is as follows:

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
        DataSourceID="EntityDataSource1" InsertItemPosition="LastItem">

        <InsertItemTemplate>...</InsertItemTemplate>
        <ItemSeparatorTemplate>...</ItemSeparatorTemplate>
        <ItemTemplate>...</ItemTemplate>
        <LayoutTemplate>...</LayoutTemplate>
        <SelectedItemTemplate>...</SelectedItemTemplate>

    </asp:ListView>
```

The `<ItemSeparatorTemplate>...` tag is highlighted with a blue selection rectangle, indicating it is the target for removal.

Example

ListView

- Remove



The image shows a portion of an ASP.NET page's source code. It includes the following structure:

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
        DataSourceID="EntityDataSource1" InsertItemPosition="LastItem">

        <InsertItemTemplate>...</InsertItemTemplate>

        <ItemTemplate>...</ItemTemplate>
        <LayoutTemplate>...</LayoutTemplate>
        <SelectedItemTemplate>...</SelectedItemTemplate>

    </asp:ListView>
```

The code editor highlights several template sections: `<InsertItemTemplate>...`, `<ItemTemplate>...`, `<LayoutTemplate>...`, and `<SelectedItemTemplate>...`. The `<SelectedItemTemplate>...` section is highlighted with a blue rectangle.

Example

ListView

Locate the **** element in the **LayoutTemplate**

```
<LayoutTemplate>
    <ul ID="itemPlaceholderContainer" runat="server"
        style="font-family: Verdana, Arial, Helvetica, sans-serif;">
        <li runat="server" id="itemPlaceholder" />
    </ul>
    <div style="text-align: center;background-color: #5D7B9D;font-family: Verdana, Arial, Helvetica, sans-serif;color: #FFFFFF;">
    </div>
</LayoutTemplate>
```



```
<LayoutTemplate>
    <ul class="ItemContainer">
        <li runat="server" id="itemPlaceholder" />
    </ul>
</LayoutTemplate>
```

Locate the ItemTemplate

```
<ItemTemplate>
    <li style="background-color: #E0FFFF;color: #333333;">Id:
        Remove <asp:Label ID="IdLabel" runat="server" Text='<%# Eval("Id") %>' />
        <br />
        Description:
        <asp:Label ID="DescriptionLabel" runat="server"
            Text='<%# Eval("Description") %>' />
        <br />
        ToolTip:
        <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />
        <br />
        ImageUrl:
        <asp:Label ID="ImageUrlLabel" runat="server" Text='<%# Eval("ImageUrl") %>' />
        <br />
        PhotoAlbumId:
        Remove <asp:Label ID="PhotoAlbumIdLabel" runat="server"
            Text='<%# Eval("PhotoAlbumId") %>' />
        <br />
        PhotoAlbum:
        Remove <asp:Label ID="PhotoAlbumLabel" runat="server"
            Text='<%# Eval("PhotoAlbum") %>' />
        <br />
        <asp:Button ID="DeleteButton" runat="server" CommandName="Delete"
            Text="Delete" />
    </li>
</ItemTemplate>
```

Example

ListView

```
<ItemTemplate>
    <li style="background-color: #E0FFFF;color: #333333;">
        Description:
        <asp:Label ID="DescriptionLabel" runat="server"
            Text='<%# Eval("Description") %>' />
        <br />
        ToolTip:
        <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />
        <br />
        ImageUrl:
        <asp:Label ID="ImageUrlLabel" runat="server" Text='<%# Eval("ImageUrl") %>' />
        <br />
        <asp:Button ID="DeleteButton" runat="server" CommandName="Delete"
            Text="Delete" />
    </li>
</ItemTemplate>
```

Locate the InsertItemTemplate

Remove

```
<InsertItemTemplate>
    <li style="">
        Id:
        {
            <asp:TextBox ID="IdTextBox" runat="server" Text='<%# Bind("Id") %>' />
            <br />
        Description:
            <asp:TextBox ID="DescriptionTextBox" runat="server"
                Text='<%# Bind("Description") %>' />
            <br />
        ToolTip:
            <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
            <br />
        ImageUrl:
            <asp:TextBox ID="ImageUrlTextBox" runat="server"
                Text='<%# Bind("ImageUrl") %>' />
            <br />
        PhotoAlbumId:
        {
            <asp:TextBox ID="PhotoAlbumIdTextBox" runat="server"
                Text='<%# Bind("PhotoAlbumId") %>' />
            <br />
        PhotoAlbum:
        {
            <asp:TextBox ID="PhotoAlbumTextBox" runat="server"
                Text='<%# Bind("PhotoAlbum") %>' />
            <br />
            <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
                Text="Insert" />
            <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
                Text="Clear" />
        }
    </li>
</InsertItemTemplate>
```

Remove

Remove

Example

ListView

```
<InsertItemTemplate>
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        ImageUrl:
        <asp:TextBox ID="ImageUrlTextBox" runat="server"
            Text='<%# Bind("ImageUrl") %>' />
        <br />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" />
    </li>
</InsertItemTemplate>
```

Example

ListView

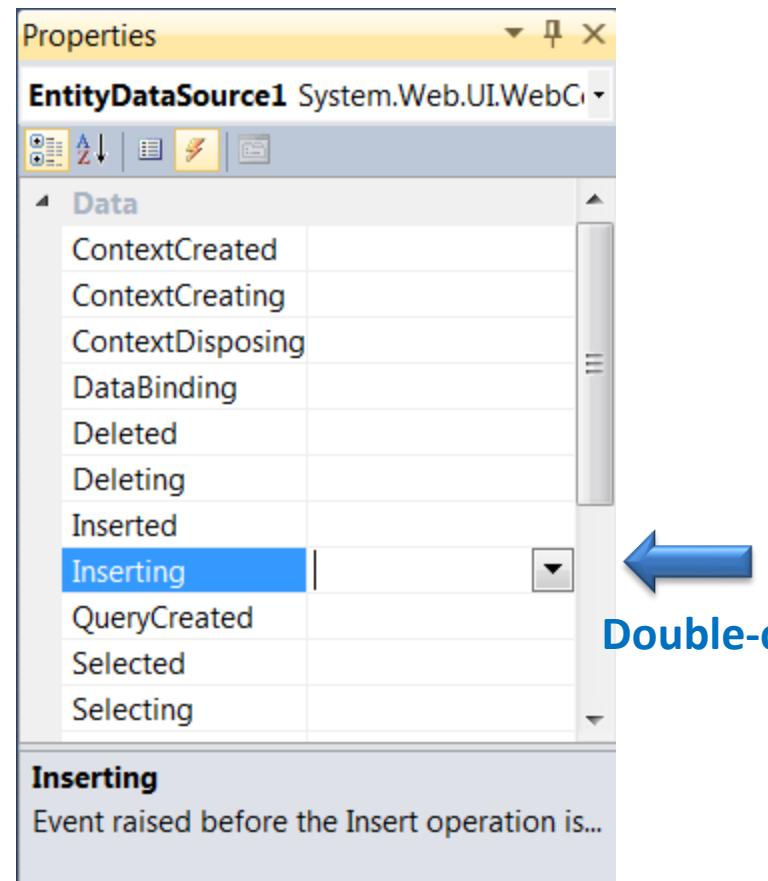
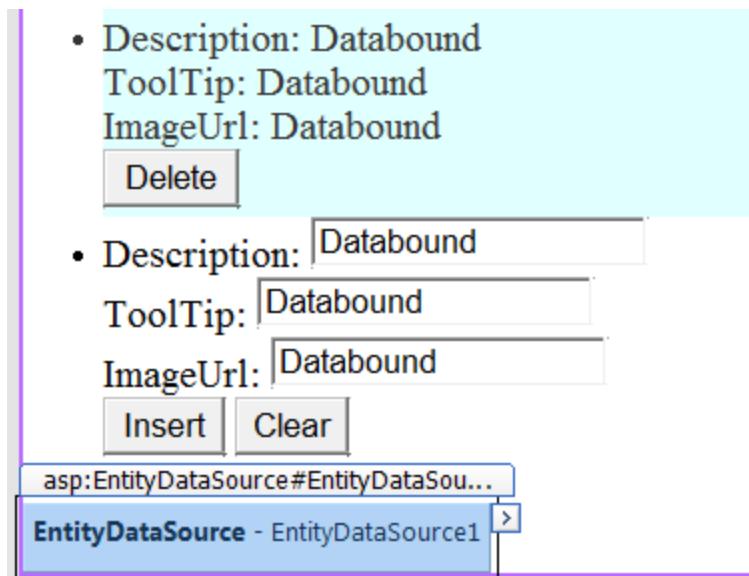
cpMainContent (Custom) asp:ListView#ListView1

• Description: Databound ToolTip: Databound ImageUrl: Databound Delete	
• Description: Databound ToolTip: Databound ImageUrl: Databound Delete	
• Description: Databound ToolTip: Databound ImageUrl: Databound Delete	
• Description: Databound ToolTip: Databound ImageUrl: Databound Delete	

Example

ListView

- Select **EntityDataSource** control



Example

ListView

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using MyWebSiteDatabaseModel; ←

public partial class _ManagePhotoAlbum : BasePage
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void EntityDataSource1_Inserting(object sender, EntityDataSourceChangingEventArgs e)
    {
        int photoAlbumId = Convert.ToInt32(Request.QueryString.Get("PhotoAlbumId"));
        Picture myPicture = (Picture)e.Entity;
        myPicture.PhotoAlbumId = photoAlbumId;
    }
}
```

Example

ListView

- Open → **Styles > Styles.css**

```
.ItemContainer
{
    width: 600px;
    list-style-type: none;
    clear: both;
}
.ItemContainer li
{
    height: 300px;
    width: 200px;
    float: left;
}
.ItemContainer li img
{
    width: 180px;
    margin: 10px 20px 10px 0;
}
```

Example ListView

- Open → Frontend.master



Example

ListView

- Drag **Styles.css** into **Frontend.master**

ScriptManager - ScriptManager1

Home Reviews About Login

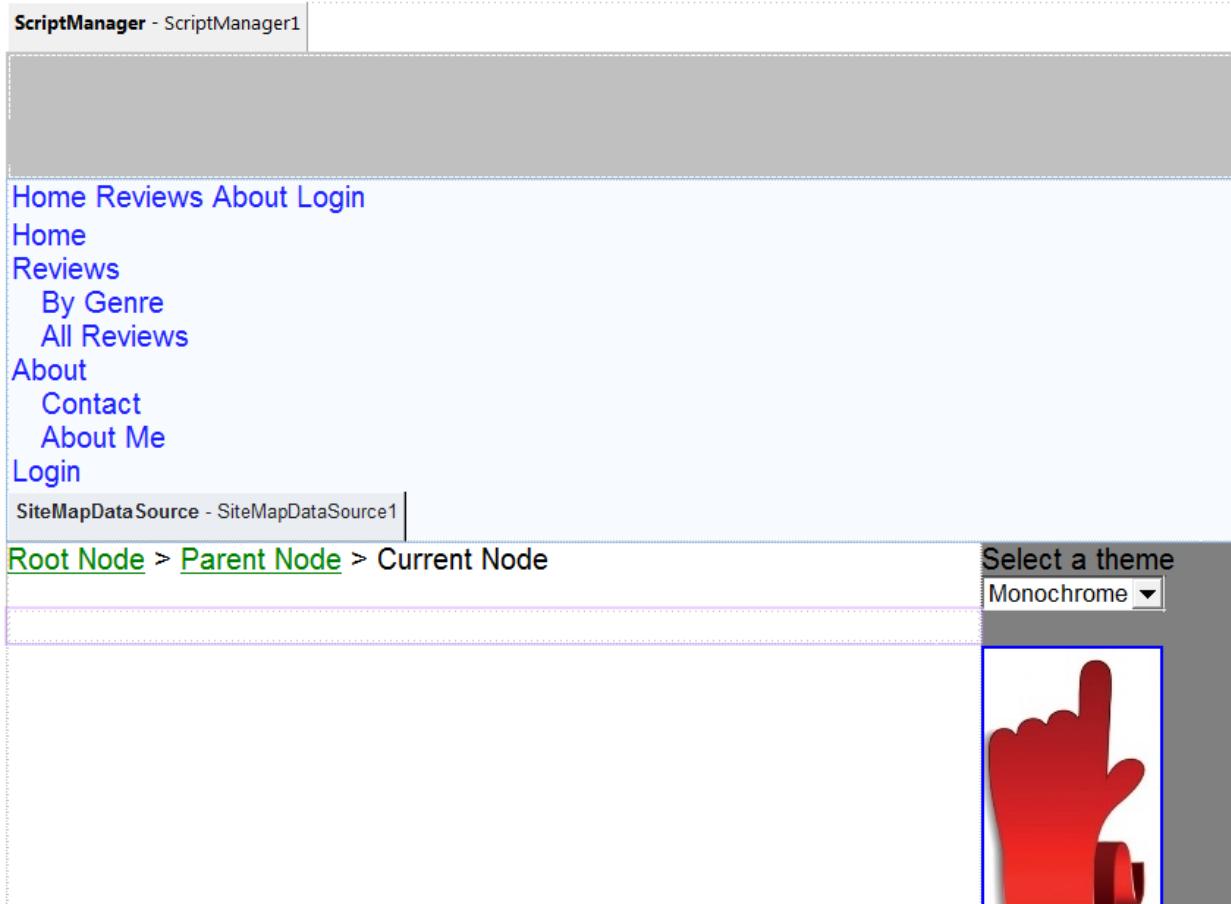
Home
Reviews
By Genre
All Reviews
About
Contact
About Me
Login

SiteMapDataSource - SiteMapDataSource1

Root Node > Parent Node > Current Node

Select a theme

Monochrome ▾



Example ListView

- View **NewPhotoAlbum.aspx** in browser.

The screenshot shows a web page with a yellow header bar. On the left is a 'Home' link, and on the right is a 'Reviews' link. Below the header is a large orange rectangular area containing a form. The form has a single input field labeled 'Name' with an empty white text box. Below the input field are two green hyperlinks: 'Insert' and 'Cancel'.

The screenshot shows the same web page as above, but the input field now contains the text 'Santana Concert'. The 'Insert' and 'Cancel' buttons remain at the bottom of the form.

Example ListView

Description:

ToolTip:

ImageUrl:

Description:

ToolTip:

ImageUrl:

Home Reviews About

Description: Santana on the stage Description:
ToolTip: Santana on the stage
ImageUrl:

Example ListView

The screenshot shows a web application interface with a navigation bar at the top and a list view below it.

Navigation Bar:

- Home
- Reviews
- About
- Login

List View:

Description	ToolTip	ImageUrl
Description: Santana on the stage ToolTip: Santana on the stage ImageUrl: http://concertpics.com/santana.jpg		
Description: Carlos Santana fall on stage ToolTip: Carlos Santana fall on stage ImageUrl: http://concertpics.com/santana_fall.jpg		

Add Item Form:

Description: Description:
ToolTip: ToolTip:
ImageUrl: ImageUrl:

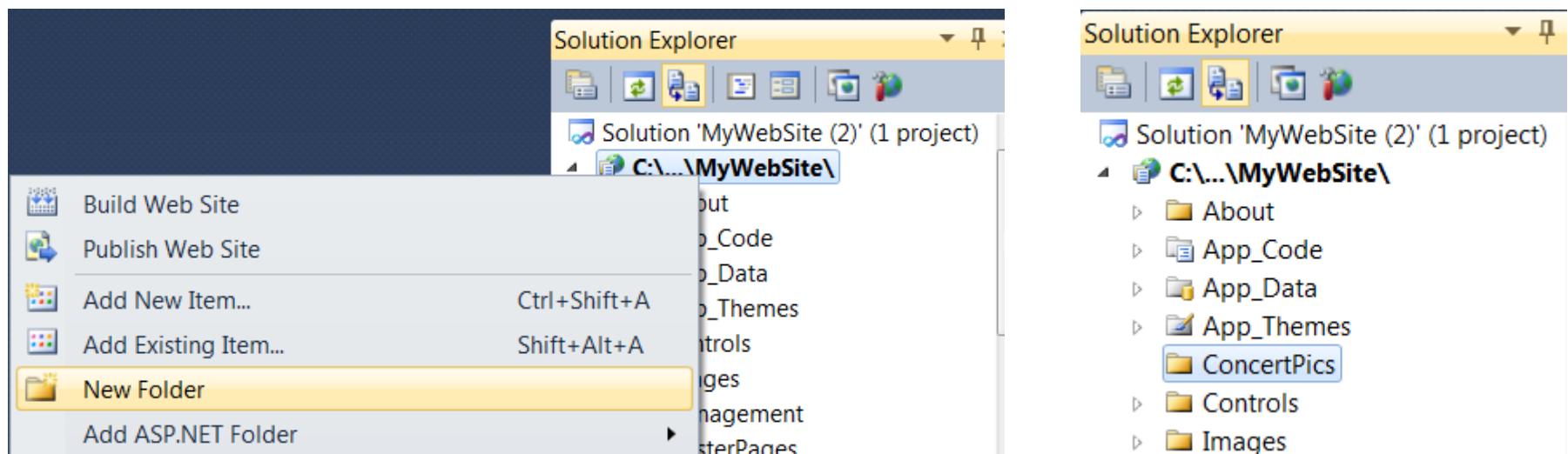
Buttons:

Delete **Delete** **Insert** **Clear**

Example

Customizing Templates of the ListView

- Create a new folder → **ConcertPics**



Example

Customizing Templates of the ListView

- Open ManagePhotoAlbum.aspx

```
<ItemTemplate>
    <li style="">
        Description:
        <asp:Label ID="DescriptionLabel" runat="server"
            Text='<%# Eval("Description") %>' />
        <br />
        ToolTip:
        <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />
        <br />
        Remove
        ImageUrl:
        <img alt='Image Url' src='<%# Eval("ImageUrl") %>' />
        <br />
        <asp:Button ID="DeleteButton" runat="server" CommandName="Delete"
            Text="Delete" />
    </li>
</ItemTemplate>
```

Example

Customizing Templates of the ListView

- Drag an **Image** control

```
<ItemTemplate>
    <li style="">
        Description:
        <asp:Label ID="DescriptionLabel" runat="server"
            Text='<%# Eval("Description") %>' />
        <br />
        ToolTip:
        <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />
        <br />
        ImageUrl:
        <asp:Image ID="ImageUrl" runat="server" ImageUrl= '<%# Eval("ImageUrl") %>' />
        <br />
        <asp:Button ID="DeleteButton" runat="server" CommandName="Delete"
            Text="Delete" />
    </li>
</ItemTemplate>
```

Example

Customizing Templates of the ListView

- In the **InsertItemTemplate**:

```
<InsertItemTemplate>
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        ImageUrl:
        <asp:TextBox ID="ImageUrlTextBox" runat="server"
            Text='<%# Bind("ImageUrl") %>' />
        <br />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" />
    </li>
</InsertItemTemplate>
```

Remove {

Example

Customizing Templates of the ListView

- Drag a **FileUpload** control.

```
<InsertItemTemplate>
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" />
    </li>
</InsertItemTemplate>
```

```
<InsertItemTemplate> ←
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" />
    </li>
</InsertItemTemplate>
```



```
<InsertItemTemplate>
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" CausesValidation="false"/> ↓
    </li>
</InsertItemTemplate>
```

```
<ItemTemplate> ←
  <li style="">
    Description:
    <asp:Label ID="DescriptionLabel" runat="server"
      Text='<%# Eval("Description") %>' />
    <br />
    ToolTip:
    <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />
    <br />
    ImageUrl:
    <asp:Image ID="ImageUrl" runat="server" ImageUrl= '<%# Eval("ImageUrl") %>' />
    <br />
    <asp:Button ID="DeleteButton" runat="server" CommandName="Delete"
      Text="Delete" />
  </li>
</ItemTemplate>
```



```
<ItemTemplate>
  <li style="">
    Description:
    <asp:Label ID="DescriptionLabel" runat="server"
      Text='<%# Eval("Description") %>' />
    <br />
    ToolTip:
    <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />
    <br />
    ImageUrl:
    <asp:Image ID="ImageUrl" runat="server" ImageUrl= '<%# Eval("ImageUrl") %>' />
    <br />
    <asp:Button ID="DeleteButton" runat="server" CommandName="Delete"
      Text="Delete" CausesValidation="false"/>
  </li>
</ItemTemplate>
```

Example

Customizing Templates of the ListView

- Open → **ManagePhotoAlbum.aspx.cs**

```
protected void EntityDataSource1_Inserting(object sender, EntityDataSourceChangedEventArgs e)
{
    int photoAlbumId = Convert.ToInt32(Request.QueryString.Get("PhotoAlbumId"));
    Picture myPicture = (Picture)e.Entity;
    myPicture.PhotoAlbumId = photoAlbumId;

    FileUpload FileUpload1 = (FileUpload)ListView1.InsertItem.FindControl("FileUpload1");
    string virtualFolder = "~/ConcertPics/";
    string physicalFolder = Server.MapPath(virtualFolder);
    string fileName = Guid.NewGuid().ToString();
    string extension = System.IO.Path.GetExtension(FileUpload1.FileName);
    FileUpload1.SaveAs(System.IO.Path.Combine(physicalFolder, fileName + extension));
    myPicture.ImageUrl = virtualFolder + fileName + extension;
}
```

Example

Customizing Templates of the ListView

- In Source View:
 - Add 2 **RequiredFieldValidator** and 1 **CustomValidator**

```
<InsertItemTemplate>
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" CausesValidation="false"/>
    </li>
</InsertItemTemplate>
```

```
<InsertItemTemplate>
    <li style="">
        Description:
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" CausesValidation="false"/>
    </li>
</InsertItemTemplate>
```



```
<InsertItemTemplate>
    <li style="">
        Description:
        {<asp:RequiredFieldValidator ID="reqDesc" ControlToValidate="DescriptionTextBox" runat="server" ErrorMessage="Enter a description.">
        </asp:RequiredFieldValidator>
        <asp:TextBox ID="DescriptionTextBox" runat="server"
            Text='<%# Bind("Description") %>' />
        <br />
        ToolTip:
        {<asp:RequiredFieldValidator ID="reqToolTip" ControlToValidate="ToolTipTextBox" runat="server" ErrorMessage="Enter a tooltip.">
        </asp:RequiredFieldValidator>
        <asp:TextBox ID="ToolTipTextBox" runat="server" Text='<%# Bind("ToolTip") %>' />
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <asp:CustomValidator ID="cusValImage" runat="server" ErrorMessage="Select a valid image file.">
        </asp:CustomValidator>
        <asp:Button ID="InsertButton" runat="server" CommandName="Insert"
            Text="Insert" />
        <asp:Button ID="CancelButton" runat="server" CommandName="Cancel"
            Text="Clear" CausesValidation="false"/>
    </li>
</InsertItemTemplate>
```

Example

Customizing Templates of the ListView

- Set up an event handler for **ListView** control:
 - **ItemInserting**

```
protected void ListView1_ItemInserting(object sender, ListViewInsertEventArgs e)
{
    FileUpload FileUpload1 = (FileUpload)ListView1.InsertItem.FindControl("FileUpload1");
    if (!FileUpload1.HasFile || !FileUpload1.FileName.ToLower().EndsWith(".jpg"))
    {
        CustomValidator cusValImage = (CustomValidator)ListView1.InsertItem.FindControl("cusValImage");
        cusValImage.IsValid = false;
        e.Cancel = true;
    }
}
```

Example

Customizing Templates of the ListView

- View **NewPhotoAlbum.aspx** in browser.

A screenshot of a web page titled "Home". Below the title is a text input field labeled "Name". Underneath the input field are two green links: "Insert" and "Cancel". The entire form is set against a light orange background.

A screenshot of the same web page after a value has been entered into the "Name" input field. The field now contains the text "Santana feat. Rob Thomas". The "Insert" and "Cancel" links remain below the input field, which is still set against a light orange background.

Example

Customizing Templates of the ListView

Description:

ToolTip:

Description: Santana feat. Rob Thomas
ToolTip: @NewYork



Description: Santana feat. Rob Thomas
ToolTip: @New Orleans
C:\Academic_OC\Course

Description:
Santana feat Rob Thoma

ToolTip:
@New York

C:\Academic_OC\Course

Description: Santana feat. Rob Thomas
ToolTip: @NewYork



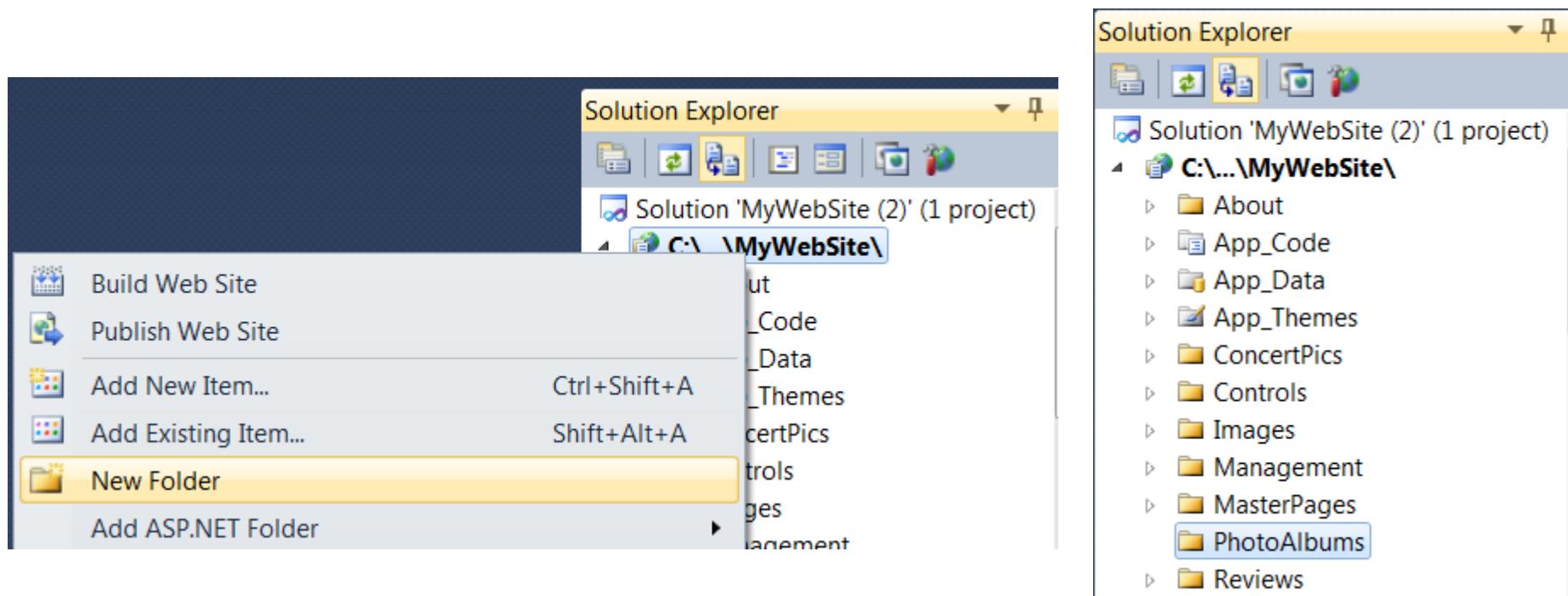
Description: Santana feat. Rob Thomas
ToolTip: @New Orleans



Description:
ToolTip:

Example DataPager

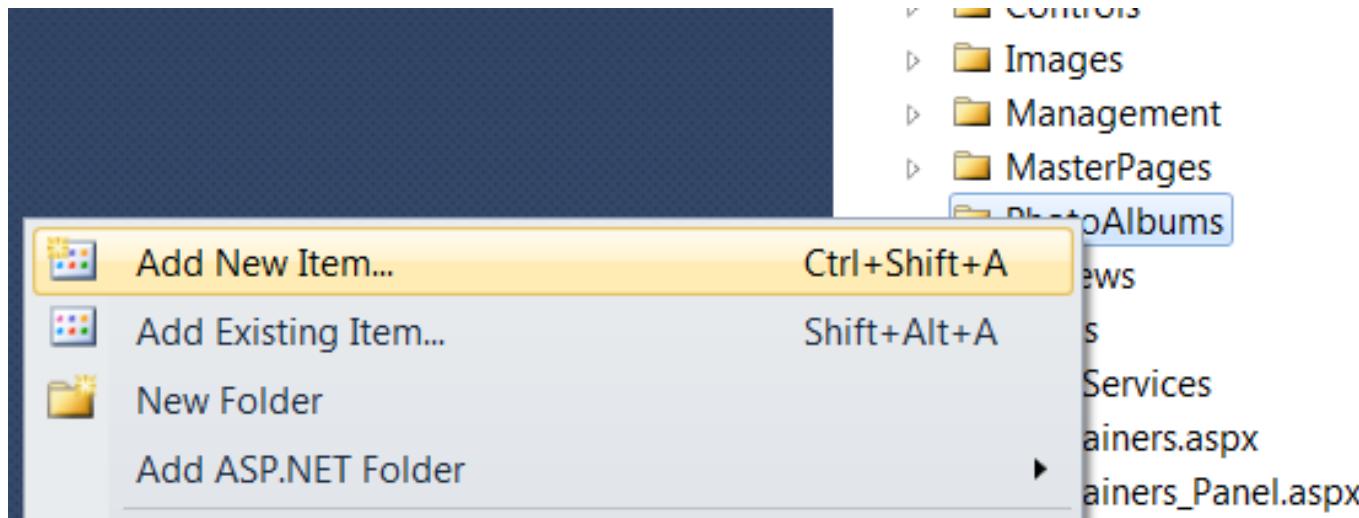
- Add New Folder → PhotoAlbums



Example

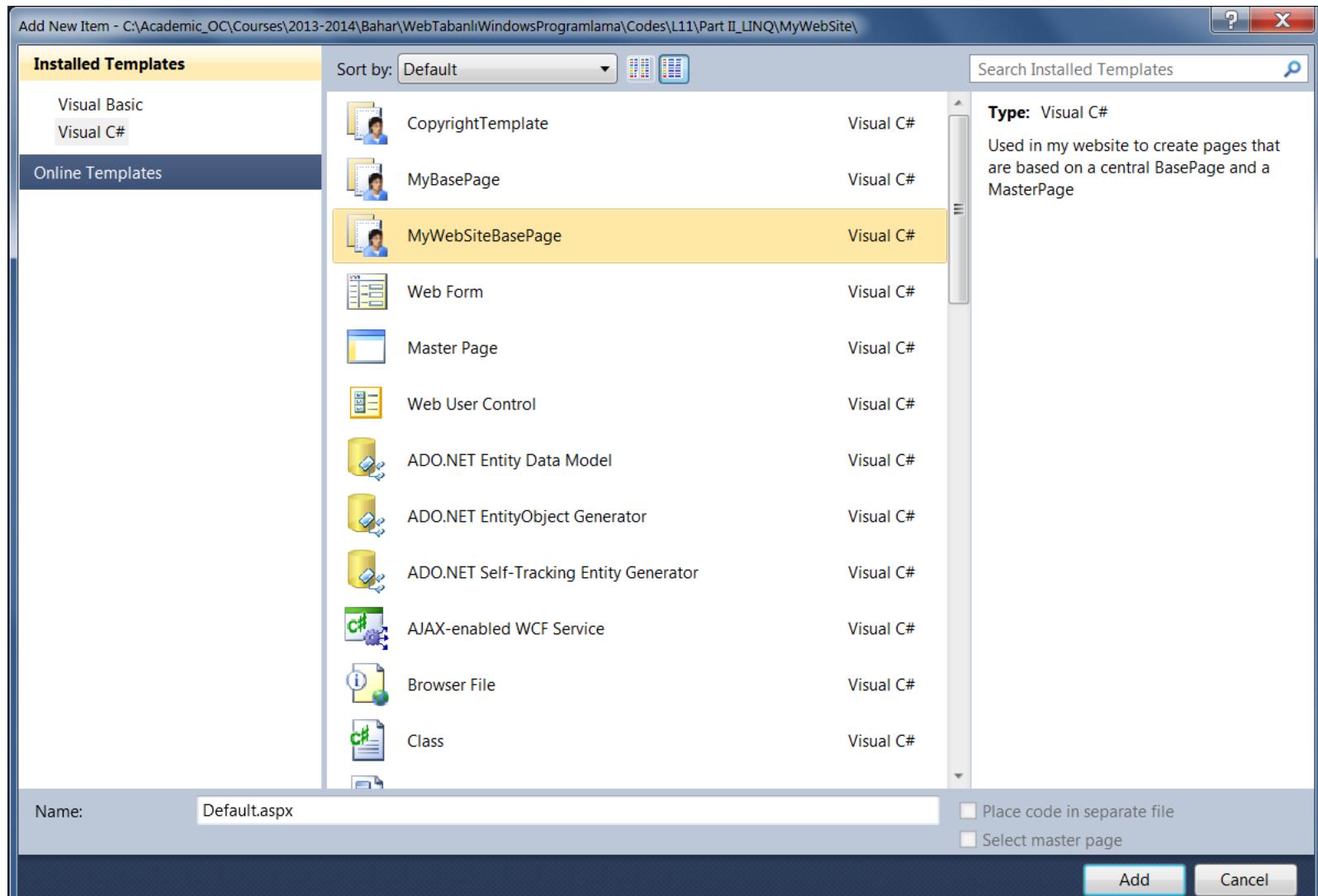
DataPager

- Add New Item → PhotoAlbums



Example

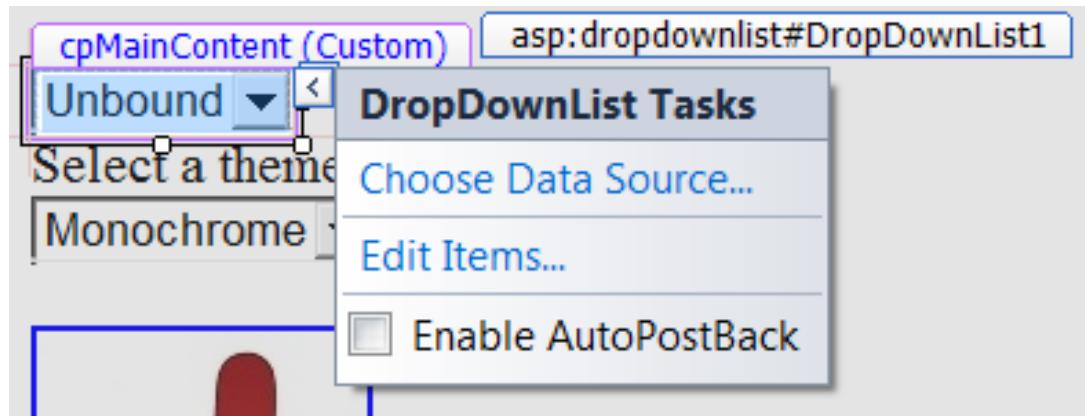
DataPager



Example

DataPager

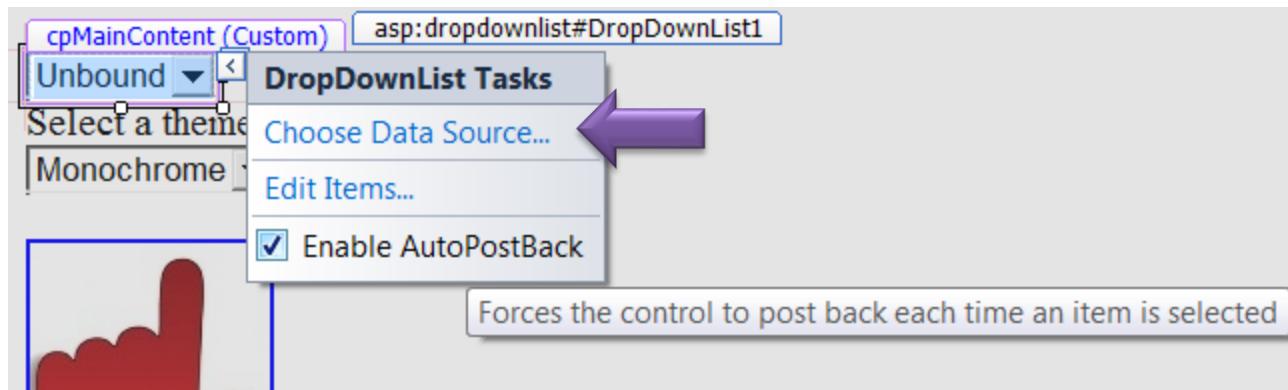
- Title = All Photo Albums
- In Design View:
 - Drop a DropDownList



Example

DataPager

- Enable **AutoPostBack**





Choose a Data Source

Select a data source:

<New data source...>

(None)

<New data source...>

ownList:

<New data source...>

Select a data field for the value of the DropDownList:

<New data source...>

OK

Cancel



Choose a Data Source Type

Where will the application get data from?



Access
Database



Database



Entity



LINQ



Object



Site Map



XML File

Connect to an ADO.NET Entity Framework Model.

Specify an ID for the data source:



Configure ObjectContext

ConnectionString:

Named Connection

MyWebSiteDatabaseEntities

Connection String

DefaultContainerName:

MyWebSiteDatabaseEntities

< Previous

Next >

Finish

Cancel



Configure Data Selection

EntitySetName:

EntityTypeFilter:

Select:

 Select All (Entity Value) Id Name Enable automatic inserts Enable automatic updates Enable automatic deletes

< Previous

Next >

Finish

Cancel



Choose a Data Source

Select a data source:

Select a data field to display in the DropDownList:

Select a data field for the value of the DropDownList:

[Refresh Schema](#)

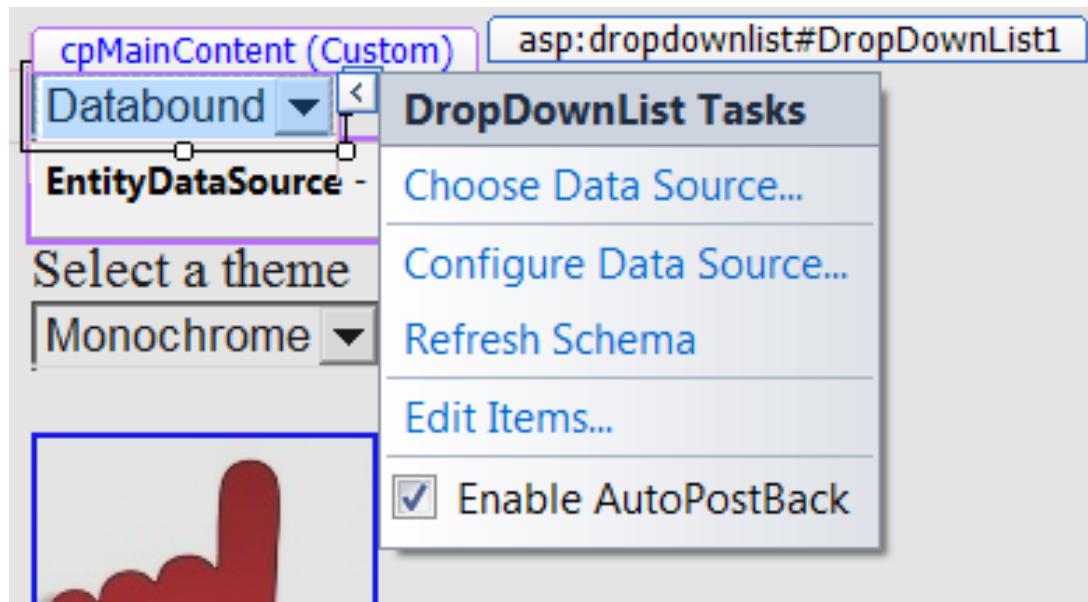


OK

Cancel

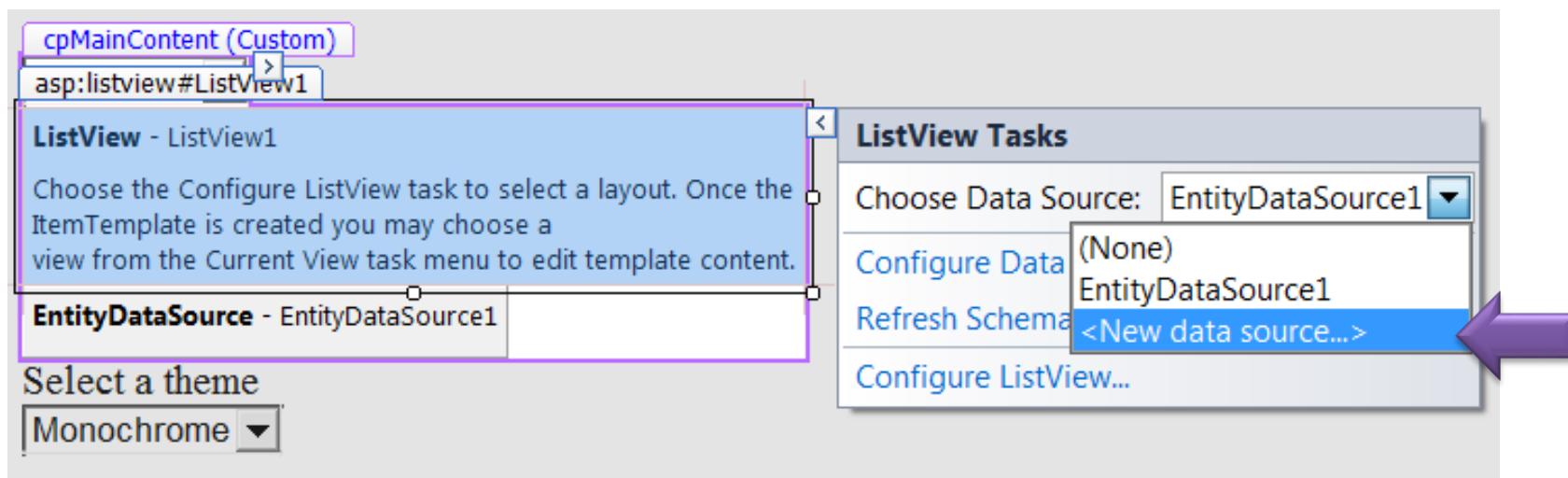
Example

DataPager



Example DataPager

- Below the **DropDownList**, **Listview** control.





Choose a Data Source Type

Where will the application get data from?



Access
Database



Database



Entity



LINQ



Object



Site Map



XML File

Connect to an ADO.NET Entity Framework Model.

Specify an ID for the data source:

EntityDataSource2

OK

Cancel



Configure ObjectContext

ConnectionString:

Named Connection

MyWebSiteDatabaseEntities

Connection String

DefaultContainerName:

MyWebSiteDatabaseEntities

< Previous

Next >

Finish

Cancel



Configure Data Selection

EntitySetName:

EntityTypeFilter:

Select:

 Select All (Entity Value) Id Description ToolTip ImageUrl PhotoAlbumId Enable automatic inserts Enable automatic updates Enable automatic deletes

< Previous

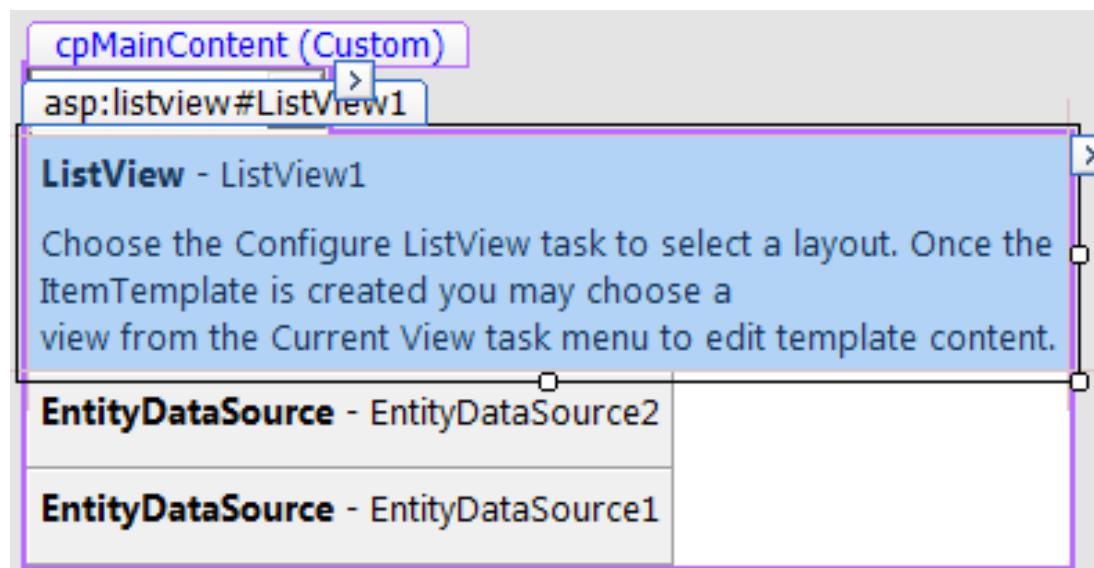
Next >

Finish

Cancel

Example

DataPager



Example DataPager

- Select EntityDataSource2

The screenshot shows the Visual Studio IDE interface. On the left, the Solution Explorer displays two projects: "MyWebSite" and "MyWebSite.csproj". In the center, the Properties window is open for "EntityDataSource2". The "ConnectionString" property is set to "name=MyWebSiteDatabase". The "DefaultContainerName" property is set to "MyWebSiteDatabase". The "EntitySetName" property is set to "Pictures". The "Where" property is currently selected. At the bottom of the Properties window, there is a "Where" section with the placeholder text: "The expression passed to the Where query buil...". On the right side of the IDE, the code editor shows the "Default.aspx.cs" file with the following code:

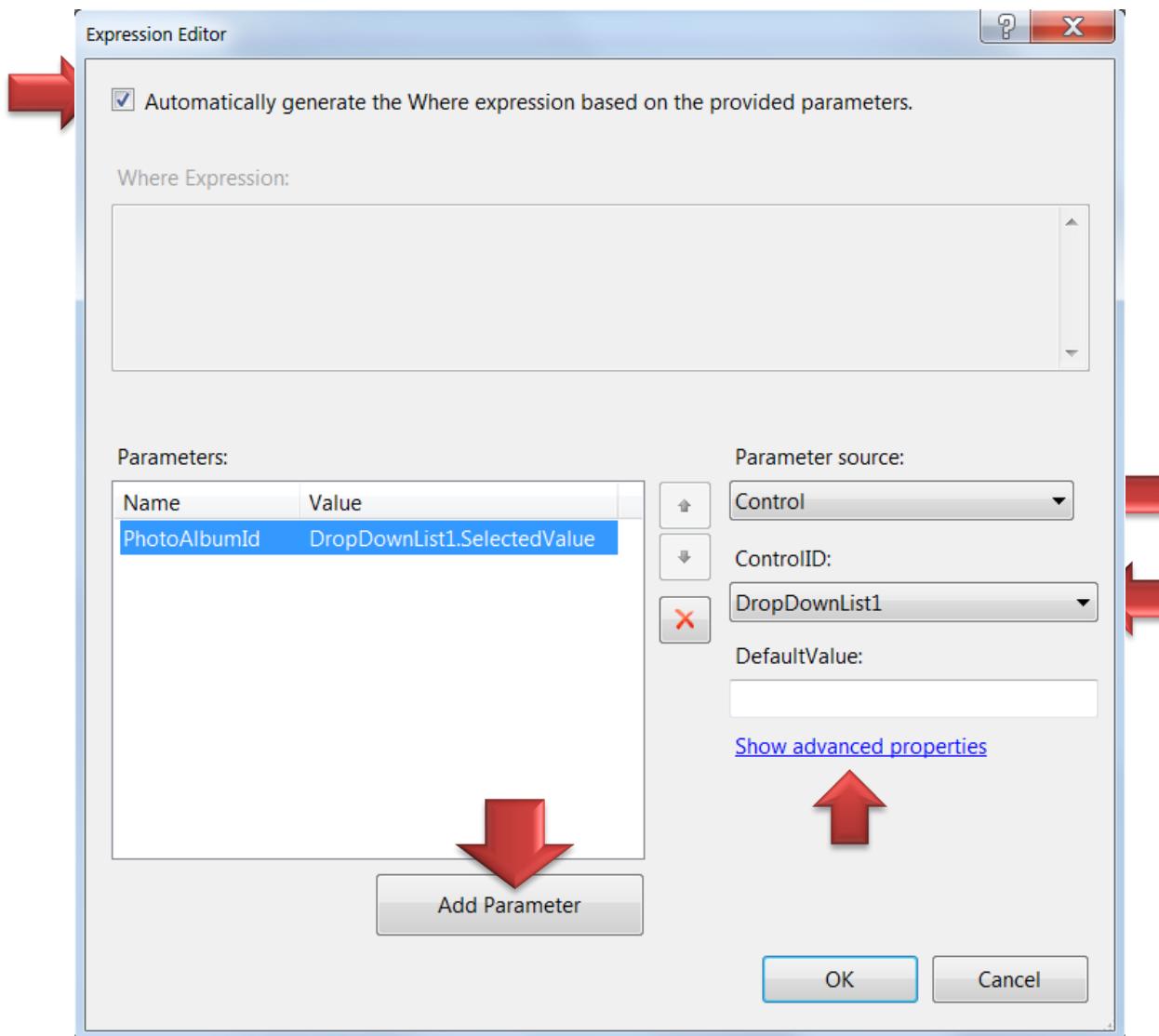
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Entity;

public partial class Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            EntityDataSource21.DataBind();
        }
    }

    protected void EntityDataSource21_Selecting(object sender, EntityDataSourceSelectingEventArgs e)
    {
        e.WhereClause = "CategoryID = 1";
    }
}
```

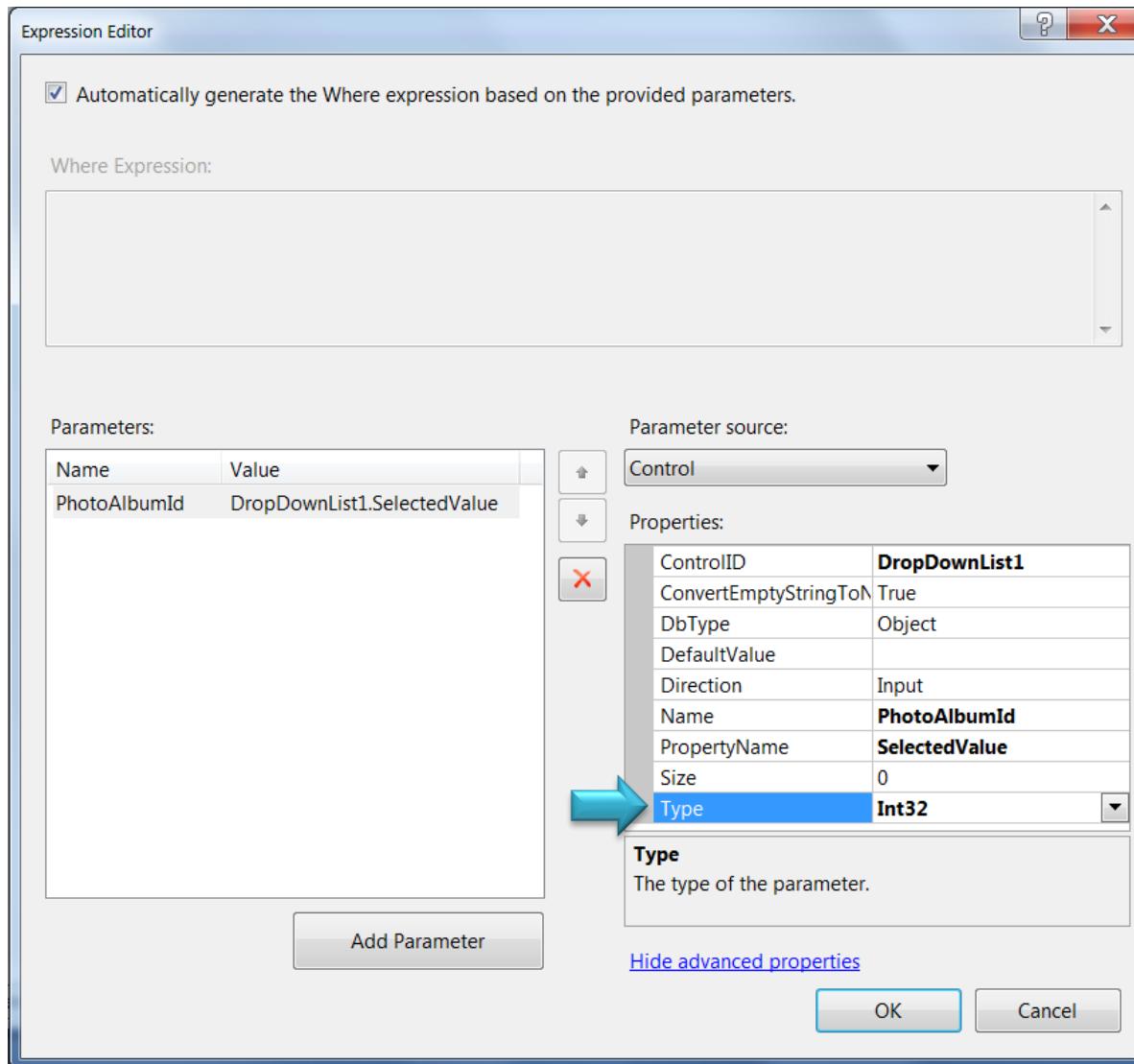
Example

DataPager



Example

DataPager

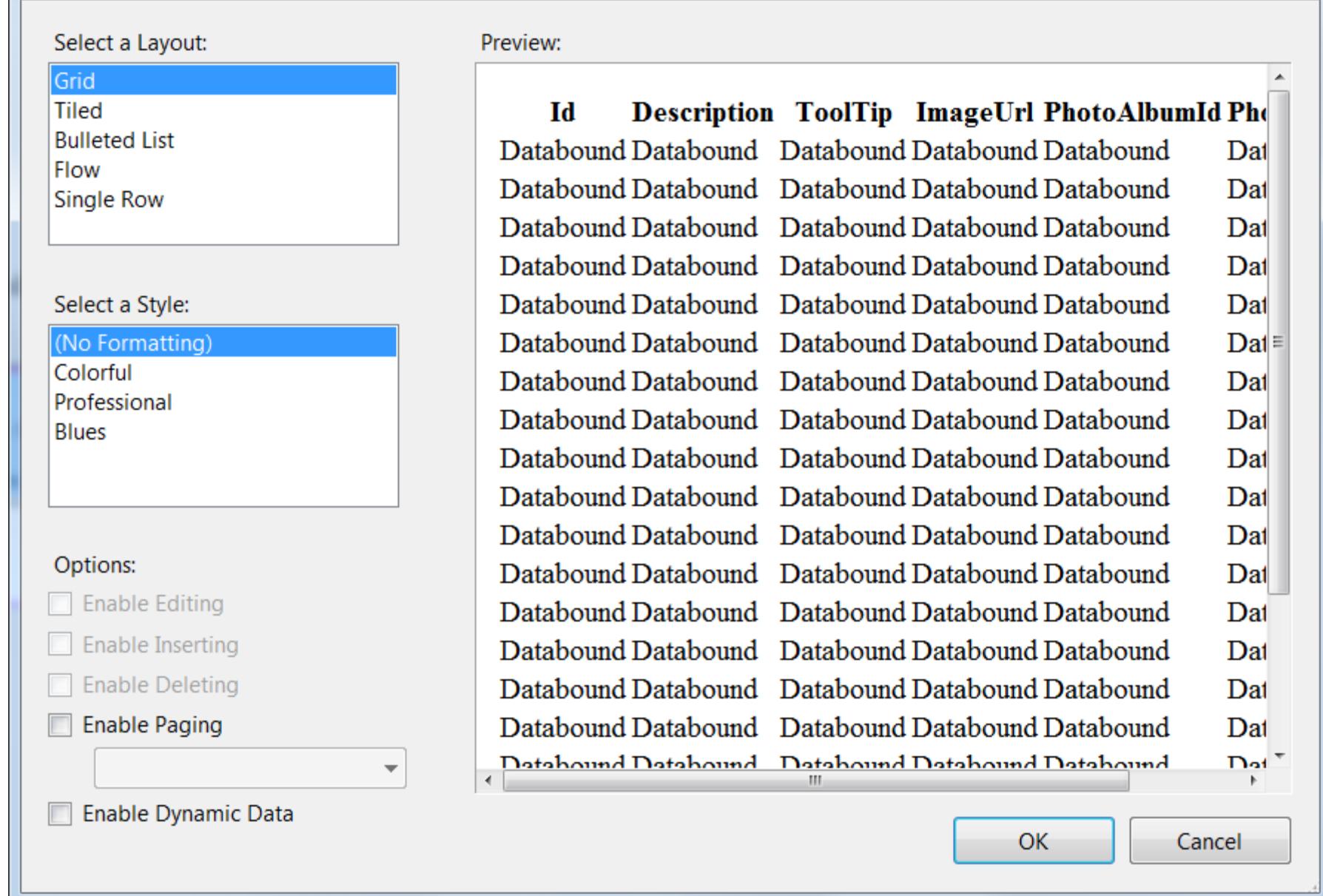


Example DataPager

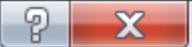
The screenshot shows the Sitefinity CMS interface. On the left, there's a content editor window with a purple border containing a **ListView - ListView1** control. Below it are two **EntityDataSource** controls: **- EntityDataSource2** and **- EntityDataSource1**. To the right of the content editor is a floating **ListView Tasks** panel. This panel has a dropdown for "Choose Data Source" set to **EntityDataSource2**. It contains three buttons: **Configure Data Source...**, **Refresh Schema** (which has a purple arrow pointing to it), and **Configure ListView...**.

This screenshot is similar to the one above, showing the Sitefinity CMS interface with a content editor window and a floating **ListView Tasks** panel. The panel's "Choose Data Source" dropdown is set to **EntityDataSource2**. The **Configure ListView...** button is highlighted with a purple arrow. A tooltip box appears below the panel, stating: **Configure the ListView's layout, style, and options**.

Configure ListView



Configure ListView



Select a Layout:

- Grid
- Tiled
- Bulleted List
- Flow
- Single Row

Select a Style:

- (No Formatting)
- Colorful
- Professional
- Blues

Options:

- Enable Editing
- Enable Inserting
- Enable Deleting
- Enable Paging

Next/Previous Pager ▾

- Enable Dynamic Data

Preview:

- Id: Databound
Description: Databound
ToolTip: Databound
ImageUrl: Databound
PhotoAlbumId: Databound
PhotoAlbum: Databound
- Id: Databound
Description: Databound
ToolTip: Databound
ImageUrl: Databound
PhotoAlbumId: Databound
PhotoAlbum: Databound
- Id: Databound
Description: Databound
ToolTip: Databound
ImageUrl: Databound
PhotoAlbumId: Databound
PhotoAlbum: Databound
- Id: Databound

OK

Cancel

Example

DataPager

The screenshot shows the Visual Studio IDE interface with a design view of a web page. The page contains a ListView control with two items. The first item has Id: 0 and the second item has Id: 1. Both items have Description: abc, ToolTip: abc, ImageUrl: abc, PhotoAlbumId: 0, and PhotoAlbum: abc.

On the right side, the 'ListView Tasks' window is open, showing options like Choose Data Source (set to EntityDataSource2), Configure Data Source..., Refresh Schema, and Configure ListView... . It also shows Current View: Runtime View and a table for DataKeyNames, DataMember, and DataSourceID. The DataKeyNames column has 'Id' selected, and the DataMember and DataSourceID columns have 'EntityDataSource2' selected.

At the top left, there's a breadcrumb navigation bar: cpMainContent (Custom) > Databound > asp:listview#ListView1.

Example

DataPager

- In Source View:

The diagram illustrates the structure of an ASP.NET ListView control. On the left, vertical labels 'R', 'E', 'M', 'O', 'V', and 'E' are aligned with red arrows pointing to specific template sections within the ListView's source code. The code itself is highlighted in blue boxes, and the template sections are shown in grey boxes.

```
<asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"  
    DataSourceID="EntityDataSource2">  
        <AlternatingItemTemplate>...</AlternatingItemTemplate>  
        <EditItemTemplate>...</EditItemTemplate>  
        <EmptyDataTemplate>...</EmptyDataTemplate>  
        <InsertItemTemplate>...</InsertItemTemplate>  
        <ItemSeparatorTemplate>...</ItemSeparatorTemplate>  
        <ItemTemplate>...</ItemTemplate>  
        <LayoutTemplate>...</LayoutTemplate>  
        <SelectedItemTemplate>...</SelectedItemTemplate>  
</asp:ListView>
```

Example

DataPager

```
<asp:ListView ID="ListView1" runat="server" DataKeyNames="Id"
    DataSourceID="EntityDataSource2">

    <EmptyDataTemplate>...</EmptyDataTemplate>

    <ItemTemplate>...</ItemTemplate>
    <LayoutTemplate>...</LayoutTemplate>

</asp:ListView>
```

```
→ <LayoutTemplate>
    <ul ID="itemPlaceholderContainer" runat="server" style="">
        <li runat="server" id="itemPlaceholder" />
    </ul>
    <div style="">
        <asp:DataPager ID="DataPager1" runat="server">
            <Fields>
                <asp:NextPreviousPagerField ButtonType="Button" ShowFirstPageButton="True"
                    ShowLastPageButton="True" />
            </Fields>
        </asp:DataPager>
    </div>
</LayoutTemplate>
```



```
→ <LayoutTemplate>
    <ul class="ItemContainer">
        <li runat="server" id="itemPlaceholder" />
    </ul>
    → <div style="clear:both">
        → <asp:DataPager ID="DataPager1" runat="server" PageSize="3">
            <Fields>
                <asp:NextPreviousPagerField ButtonType="Button" ShowFirstPageButton="True"
                    ShowLastPageButton="True" />
            </Fields>
        </asp:DataPager>
    </div>
</LayoutTemplate>
```

→<ItemTemplate>

```
<li style="">Id:  
    <asp:Label ID="IdLabel" runat="server" Text='<%# Eval("Id") %>' />  
    <br />  
    Description:  
    <asp:Label ID="DescriptionLabel" runat="server"  
        Text='<%# Eval("Description") %>' />  
    <br />  
    ToolTip:  
    <asp:Label ID="ToolTipLabel" runat="server" Text='<%# Eval("ToolTip") %>' />  
    <br />  
    ImageUrl:  
    <asp:Label ID="ImageUrlLabel" runat="server" Text='<%# Eval("ImageUrl") %>' />  
    <br />  
    PhotoAlbumId:  
    <asp:Label ID="PhotoAlbumIdLabel" runat="server"  
        Text='<%# Eval("PhotoAlbumId") %>' />  
    <br />  
    PhotoAlbum:  
    <asp:Label ID="PhotoAlbumLabel" runat="server"  
        Text='<%# Eval("PhotoAlbum") %>' />  
    <br />  
  </li>  
</ItemTemplate>
```



```
<ItemTemplate>  
  <li>  
    <asp:Image ID="Image1" runat="server" ImageUrl='<%# Eval("ImageUrl") %>' ToolTip='<%# Eval("ToolTip") %>' />  
    <asp:Label ID="DescriptionLabel" runat="server" Text='<%# Eval("Description") %>' />  
  </li>  
</ItemTemplate>
```

- Wrap the entire code in the **cpMainContent** content block in an **UpdatePanel** with a **ContentTemplate** element to avoid page flicker.

```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:DropDownList ID="DropDownList1" ...>...</asp:DropDownList>
    <asp:ListView ID="ListView1" ...>...</asp:ListView>
    <asp:EntityDataSource ID="EntityDataSource2" ...>...</asp:EntityDataSource>
    <asp:EntityDataSource ID="EntityDataSource1" ...>...</asp:EntityDataSource>
</asp:Content>
```



```
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <asp:UpdatePanel ID="UpdatePanel1" runat="server">
        <ContentTemplate>
            <asp:DropDownList ID="DropDownList1" ...>...</asp:DropDownList>
            <asp:ListView ID="ListView1" ...>...</asp:ListView>
            <asp:EntityDataSource ID="EntityDataSource2" ...>...</asp:EntityDataSource>
            <asp:EntityDataSource ID="EntityDataSource1" ...>...</asp:EntityDataSource>
        </ContentTemplate>
    </asp:UpdatePanel>
</asp:Content>
```

Example

DataPager

- Open → **Web.sitemap**

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
| <siteMapNode url="/" title="Home" description="Home" >
|   <siteMapNode url("~/Default.aspx" title="Home" description="Go to the homepage" />
|   <siteMapNode url "~/Reviews/Default.aspx" title="Reviews" description="Reviews published on this site" >
|     <siteMapNode url "~/Reviews/AllByGenre.aspx" title="By Genre" description="All reviews grouped by genre" />
|     <siteMapNode url "~/Reviews/All.aspx" title="All Reviews" description="All reviews published on this site" />
|   </siteMapNode>
|
|   { <siteMapNode url "~/PhotoAlbums/" title="Concert Pics" description="All Concert Pics">
|     <siteMapNode url "~/PhotoAlbums/Default.aspx" title="Concert Pics" description="All Concert Pics" />
|     <siteMapNode url "~/NewPhotoAlbum.aspx" title="New Album" description="Create a new Photo Album with Concert Pics" />
|   </siteMapNode>
|
|   <siteMapNode url "~/About/Default.aspx" title="About" description="About my site" >
|     <siteMapNode url "~/About/Contact.aspx" title="Contact" description="Contact me" />
|     <siteMapNode url "~/About/AboutMe.aspx" title="About Me" description="About myself" />
|   </siteMapNode>
|
|   <siteMapNode url "~/Login.aspx" title="Login" description="Log in to the web site" />
| </siteMapNode>
</siteMap>
```

Example DataPager

- Open → **Monochrome.css**

```
.MainMenu ul li
{
    /*
        Gives the four menu items in the main menu a width of 200px each
    */
    width: 200px;
}
```



```
.MainMenu ul li
{
    /*
        Gives the four menu items in the main menu a width of 200px each
    */
    width: 160px;
}
```

Example DataPager

- View Default.aspx in browser.

Home Reviews

Home > Concert Pics > Concert Pics

Santana feat. Rob Thomas ▾ No data was returned.

Home > Concert Pics > Concert Pics

Santana feat. Rob Thomas ▾



Santana

First Previous Next Last