

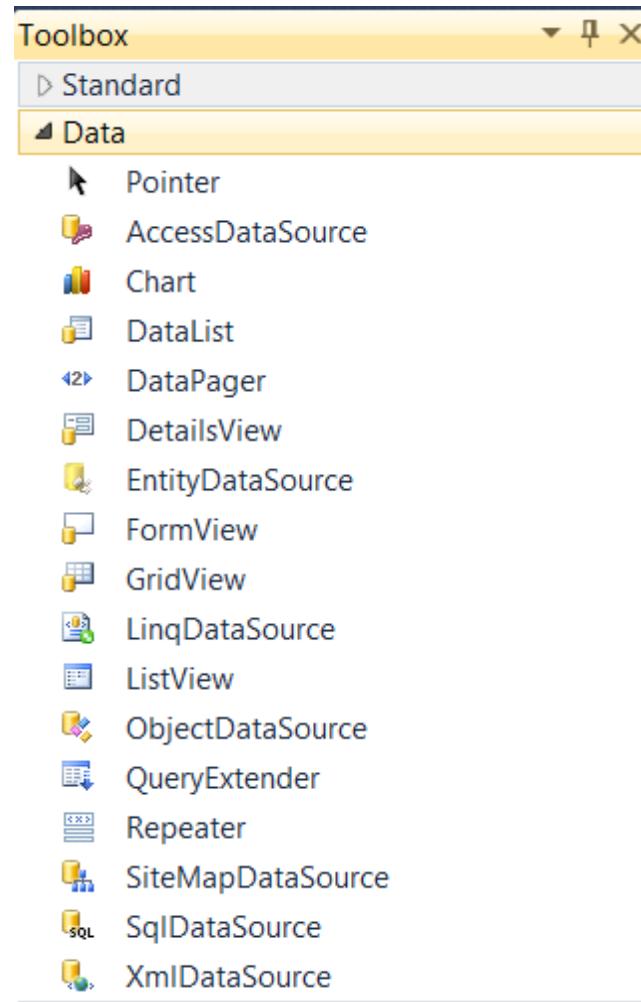
Displaying and Updating Data

Asst. Prof. Dr. Özgü Can

Data Controls

- *Data-Bound*
 - Display and edit data
- *Data Source*
 - Retrieve data from a data source, like a database or an XML file, and then offer this data to the data-bound controls.

Data Controls



Data-Bound Controls

- **DownList**
- **DataPager**
- **DetailsView**
- **FormView**
- **GridView**
- **ListView**
- **Repeater**

GridView

- Supports:
 - Automatic paging (where records are spread out over multiple “pages”)
 - Sorting / Editing / Deleting / Selecting
- It **renders its data** like a spreadsheet **with rows and columns** where ***each row contains a complete record.***
- Although many possibilities exist to style the looks of these rows and controls, you **cannot** radically **change the way the data is presented.**
- Does **not** allow you to **insert records** in the underlying data source directly.

DataList

- Enables you to **present records of data not only in rows** as with the **GridView**, but **in columns** as well, enabling you to create a ***matrix-like presentation of data***.
- Enables you to **define** the look and feel of the data through a set of **templates**.
- Does **not support paging** and **sorting** natively, and **doesn't allow** you to **insert new records** or **update or delete** existing ones.

Repeater

- Gives the **greatest flexibility** in terms of the HTML.
 - Because the control by itself **does not add** any HTML to the page output.
- Flexibility comes at a price:
 - The control has **no built-in capabilities** to **page, sort, or modify** data.

ListView

- Best combination of the **GridView**, the **DataList**, and the **Repeater**.
- Supports **editing**, **deleting**, and **paging** of data, similar to the **GridView**.
- Supports **multi-column** and **multi-row layouts** like the **DataList** offers.
- Enables you to completely **control** the markup generated by the control, just as the **Repeater** does.

Single Item Controls

- Display a single record at a time.
 - **DetailsView**
 - Uses a built-in tabular format.
 - **FormView**
 - Let you define the look and feel of your data.

DataPager

- Enables paging on other controls.
- *For the time being, it can only be used to extend the ListView control, but that might change with future versions of the .NET Framework.*

Data Source Controls

- For the *data-bound* controls to display something useful, you need to assign them a *data source*.
- To bind data source to the control:
 - You can assign data to the control's **DataSource** property.
 - or
 - You can use one of the separate data source controls.

Data Source Controls

- **AccessDataSource**
- **EntityDataSource**
- **LinqDataSource**
- **ObjectDataSource**
- **QueryExtender**
- **SqlDataSource**
- **XmlDataSource**

SqlDataSource

- Create web pages that enable you to perform all four operations (CRUD) on data:
 - **C**reate
 - **R**ead
 - **U**pdate
 - **D**elete
- **Although** its name may seem to imply that the control can **access** only **Microsoft's SQL Server**, the control can **access** other databases, such as **Oracle** or **MySQL**, as well.

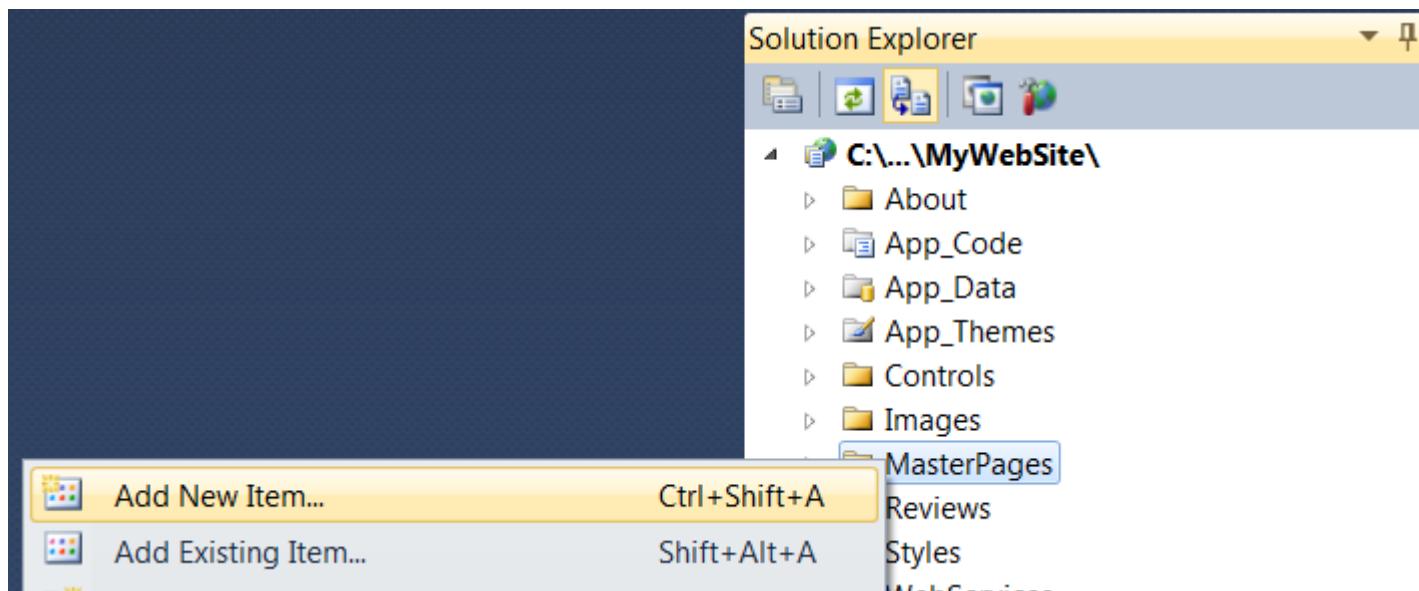
Other Data Controls

- Chart
 - Designed to render chart graphics ranging from simple bar charts to 3-D pie charts and fancy line diagrams.

Example

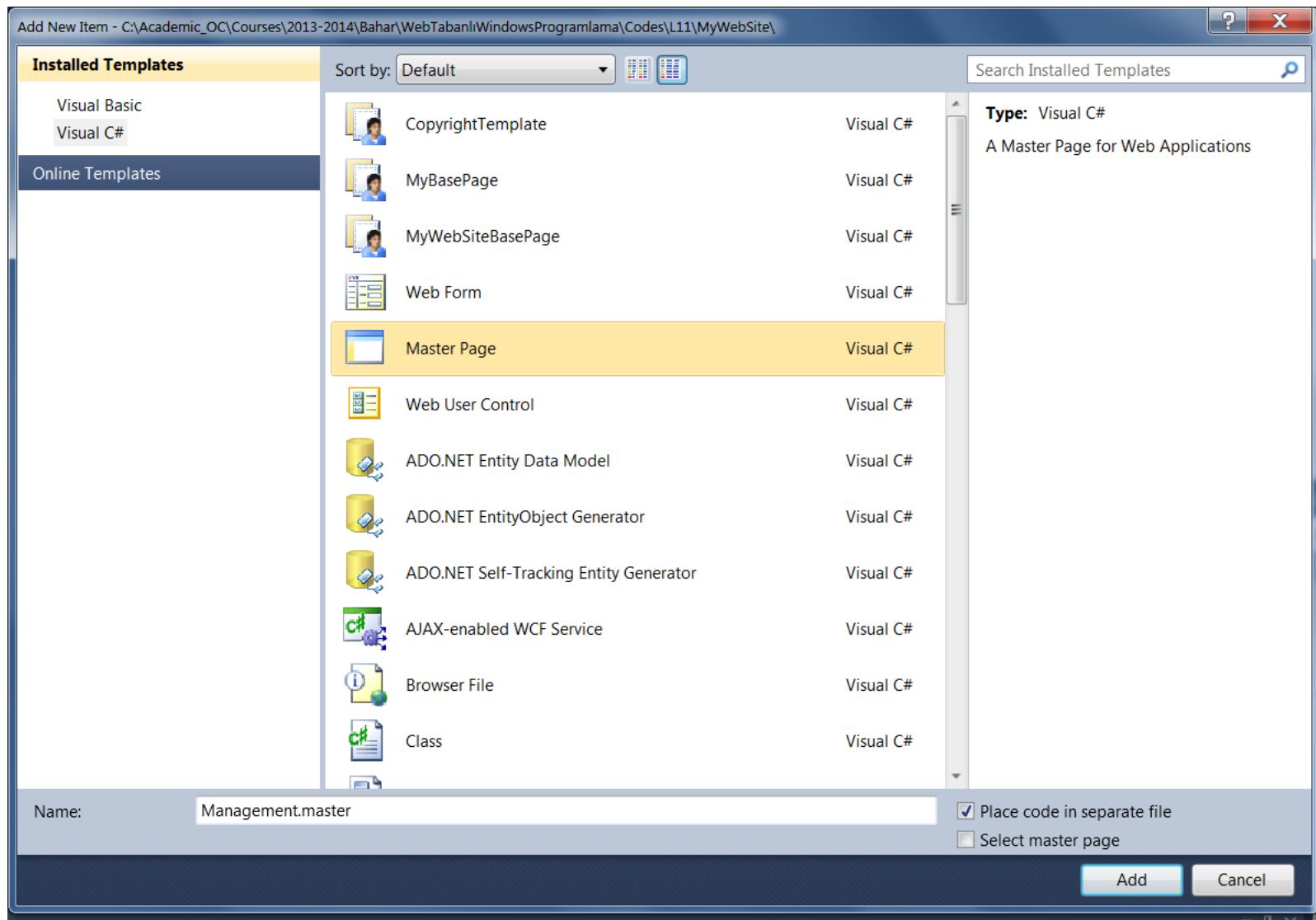
GridView and SqlDataSource

- Add New Item to MasterPages folder



Example

GridView and SqlDataSource



Example

GridView and SqlDataSource

```
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
                </asp:ContentPlaceHolder>
            </div>
        </form>
    </body>
```

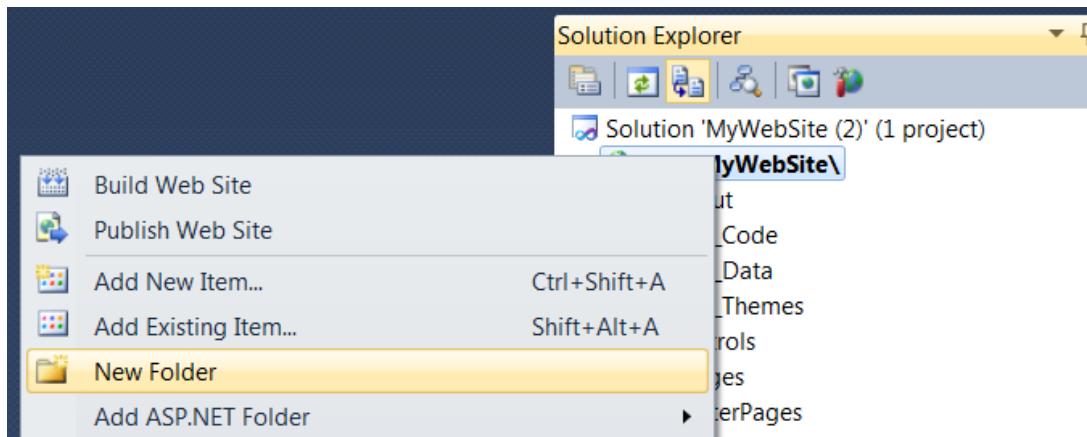


```
<body>
    <form id="form1" runat="server">
        <div>
            <div style="width: 200px; float: left;">
                <ul>
                    <li><a id="A1" href="~/Management/Default.aspx" runat="server">Management Home</a></li>
                    <li><a id="A2" href="~/Management/Genres.aspx" runat="server">Manage Genres</a></li>
                </ul>
            </div>
            <div style="width: 750px; float: left;">
                <asp:ContentPlaceHolder ID="cpMainContent" runat="server">
                </asp:ContentPlaceHolder>
            </div>
        </div>
    </form>
</body>
```

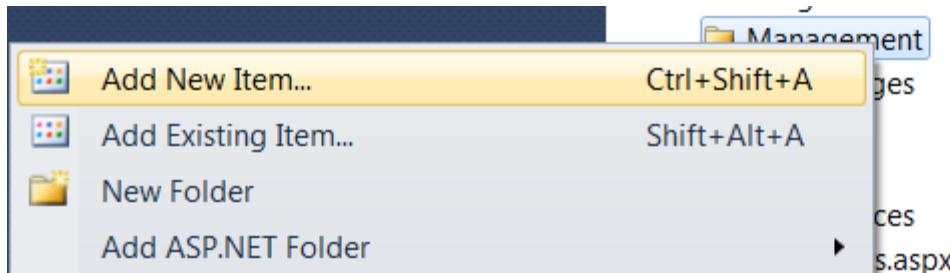
Example

GridView and SqlDataSource

- Create a new folder → **Management**

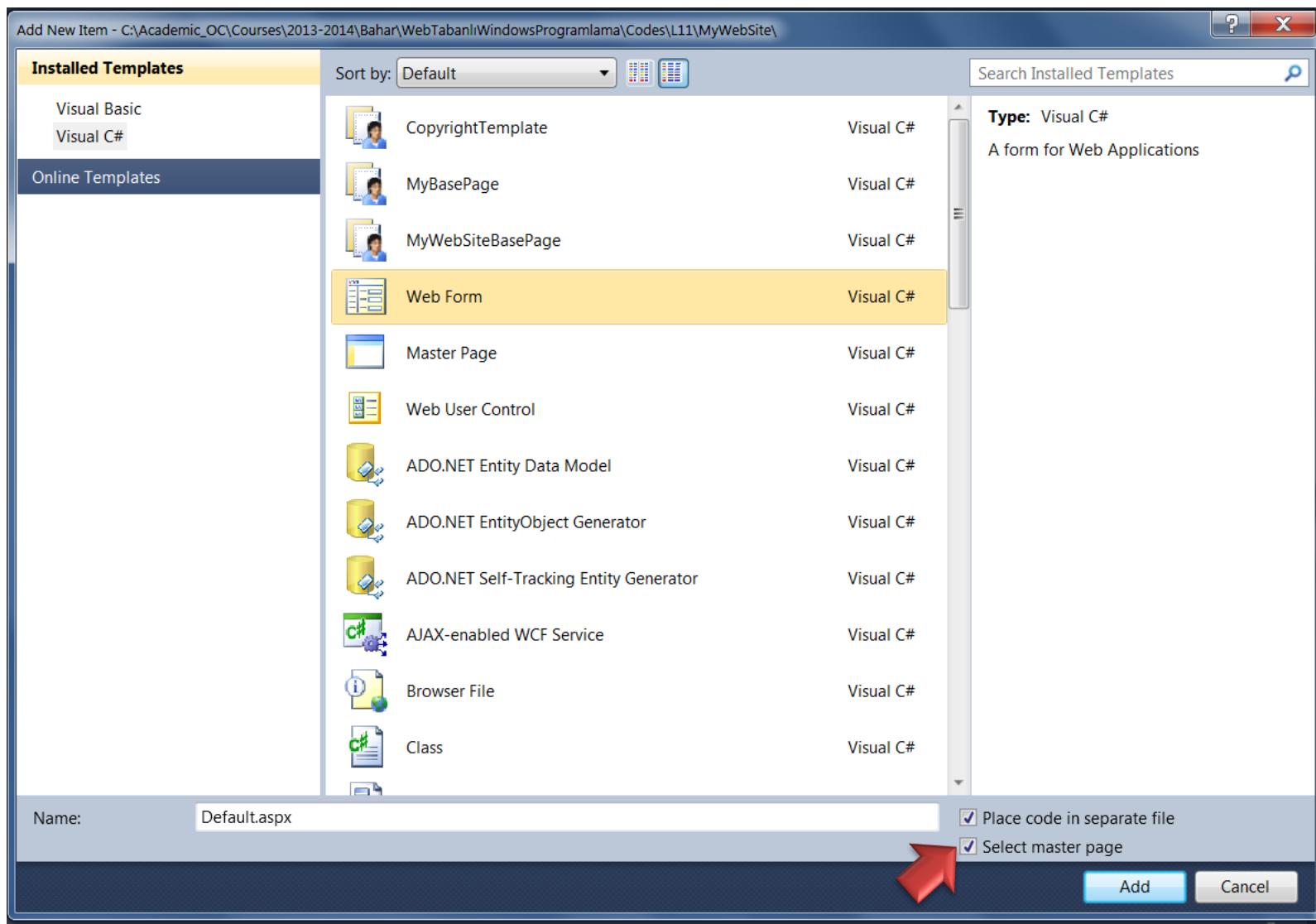


- Add New Item to Management folder



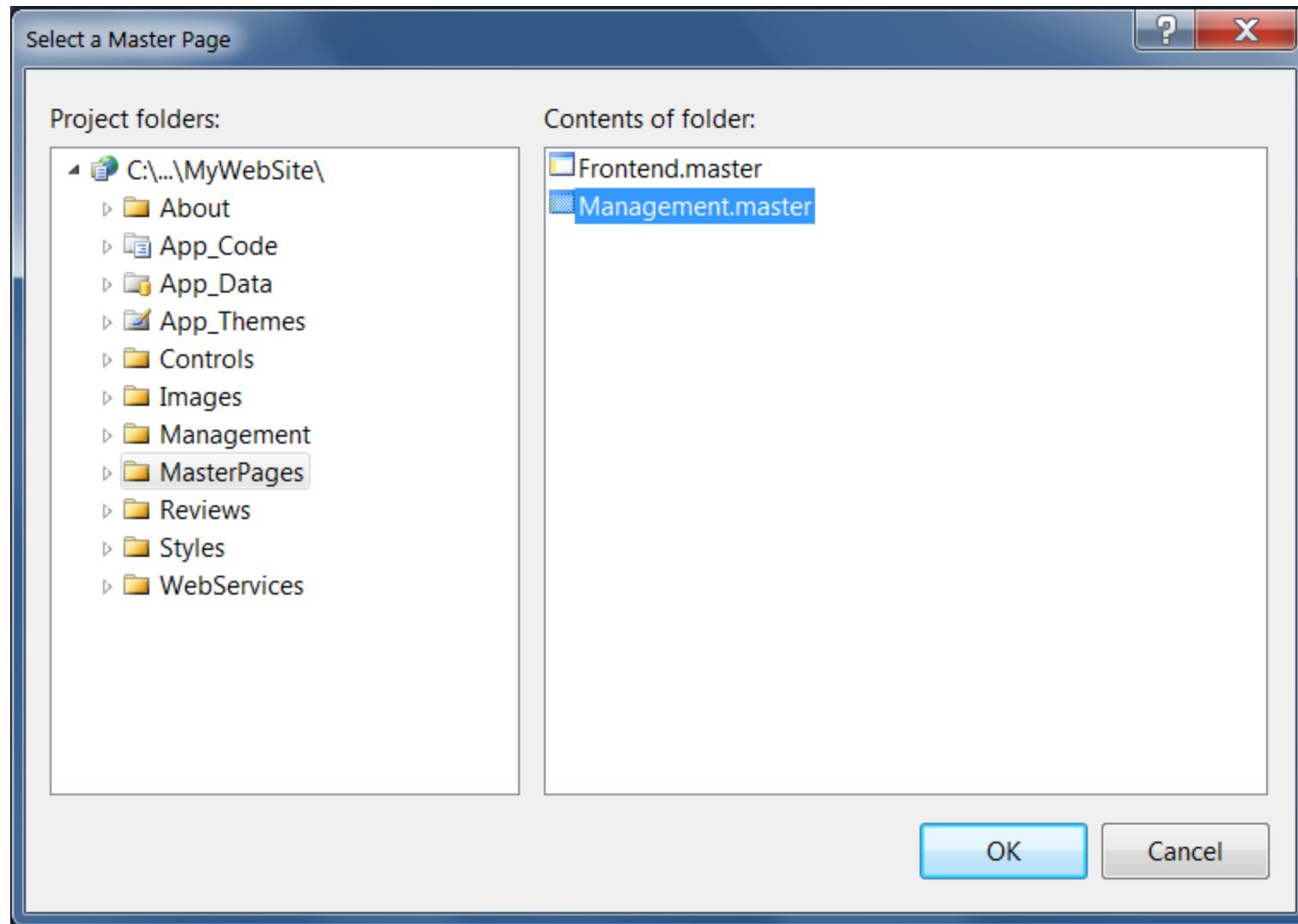
Example

GridView and SqlDataSource



Example

GridView and SqlDataSource



Example

GridView and SqlDataSource

- Change **Title = Management-Home**
- Add some text to **cpMainContent**

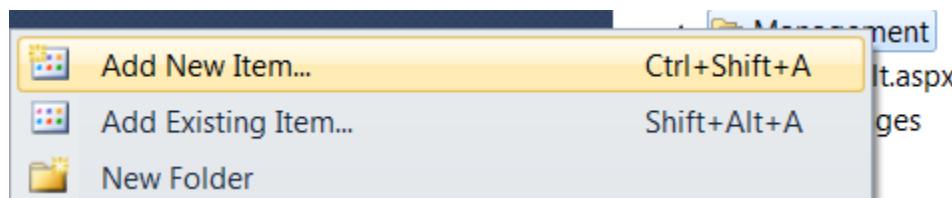
```
<%@ Page Title="Management - Home" Language="C#" MasterPageFile="~/MasterPages/Management.master" AutoEventWireup="true" Codebehind="Management.aspx.cs" Inherit="false" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
    <h1>Management Section</h1>
    <p>Welcome to the Management section of this web site. Please choose an item from the menu on the left to continue.</p>
</asp:Content>
```

Example

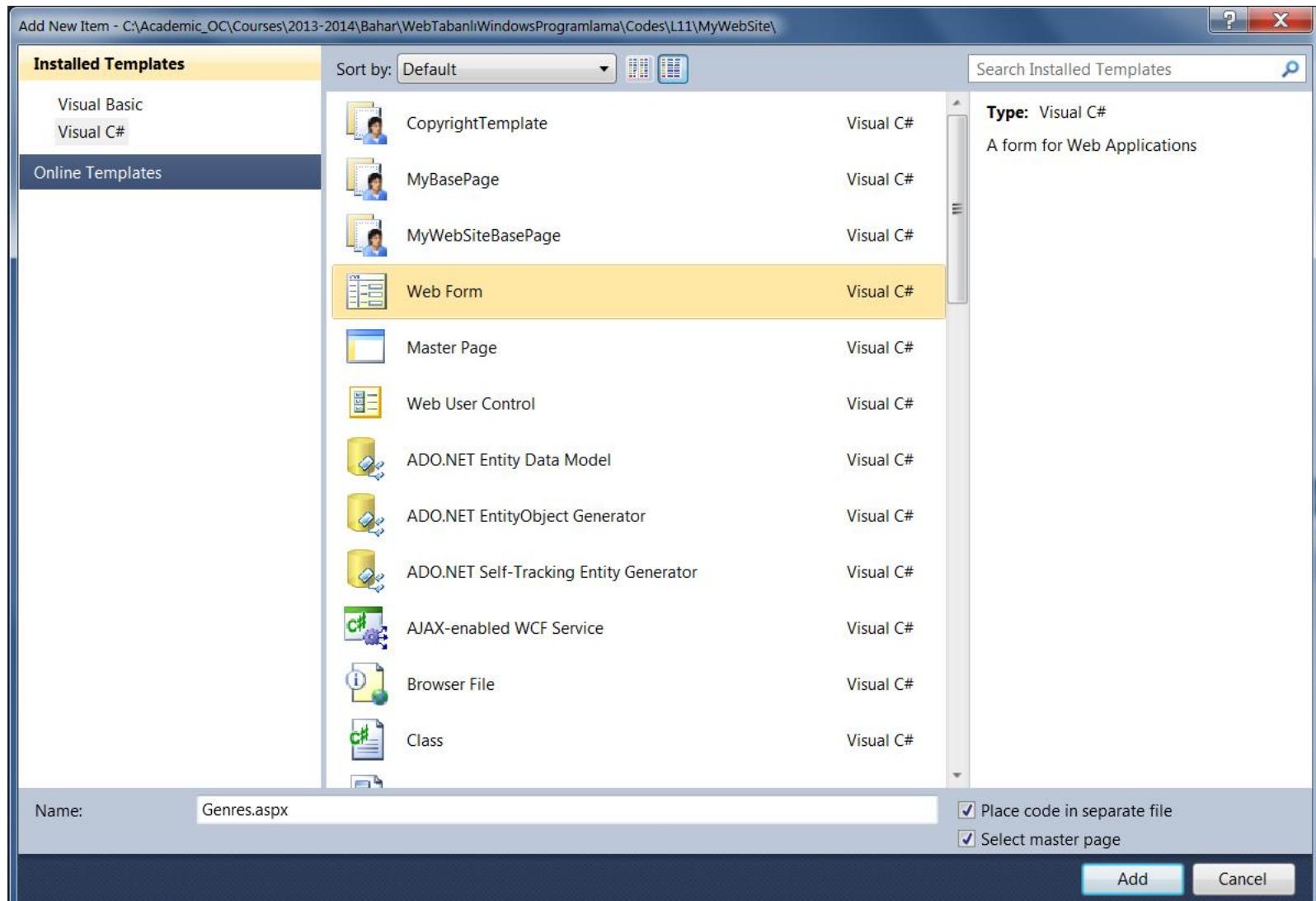
GridView and SqlDataSource

- Add New Item → **Management** folder



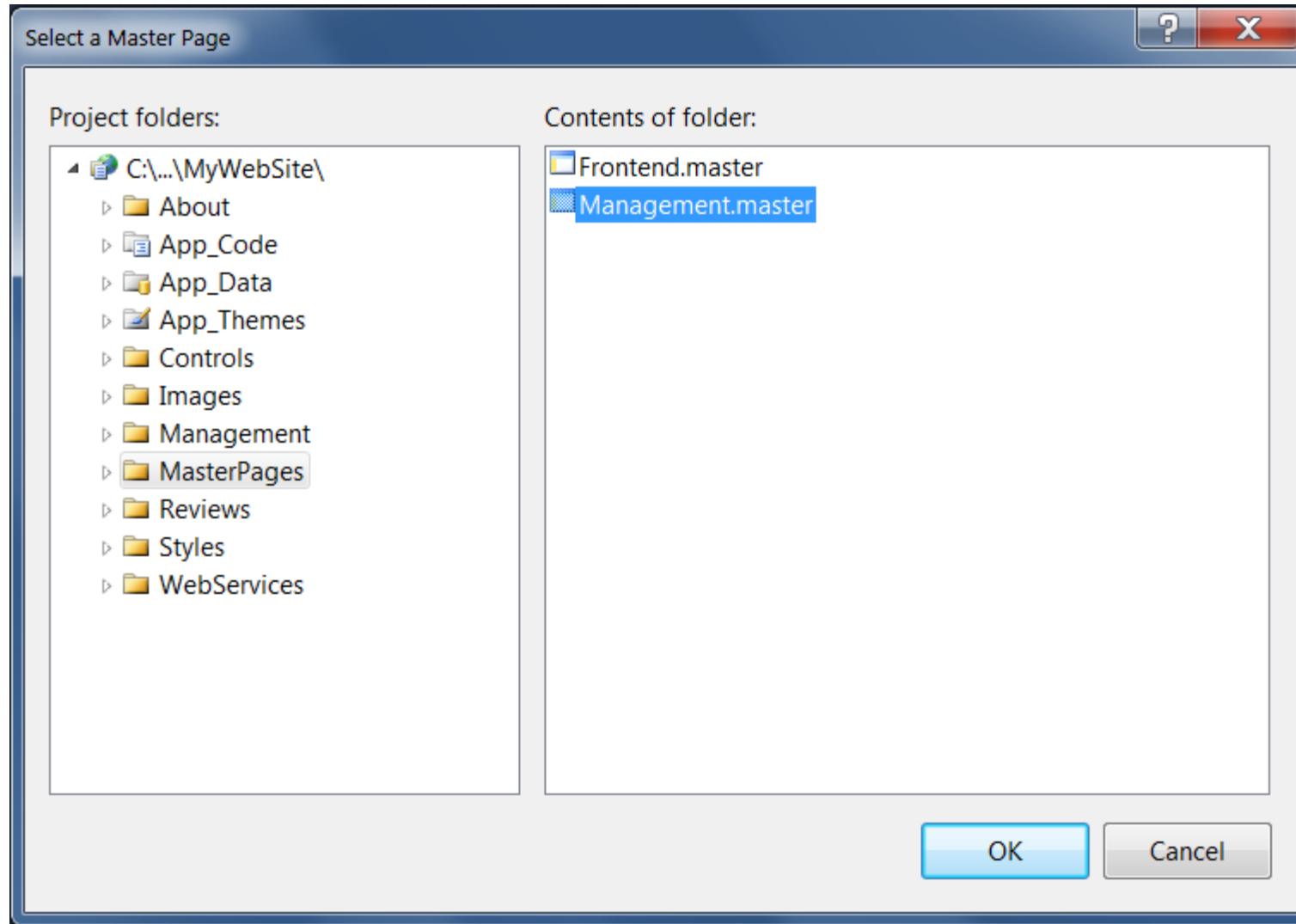
Example

GridView and SqlDataSource



Example

GridView and SqlDataSource



Example

GridView and SqlDataSource

- Change **Title = Genres**

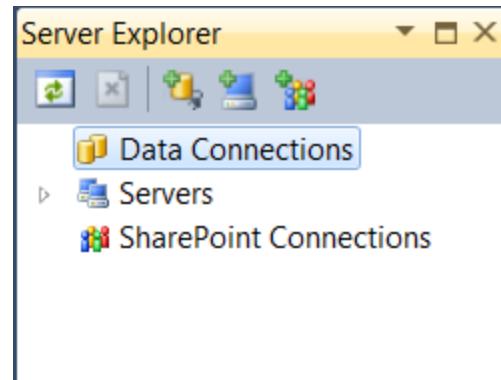
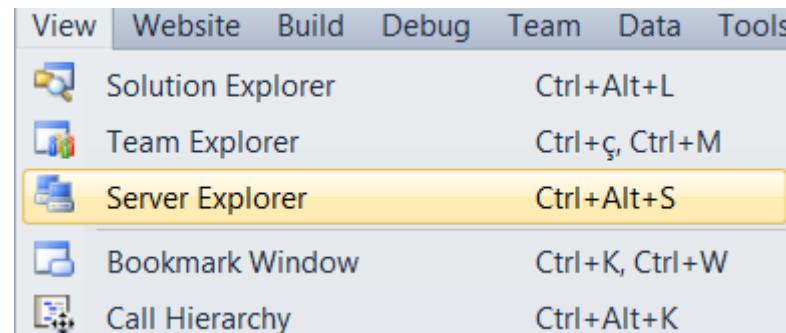
```
<%@ Page Title="Management - Genres" Language="C#" MasterPageFile="~/MasterPages/Management.master"

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
</asp:Content>
```

Example

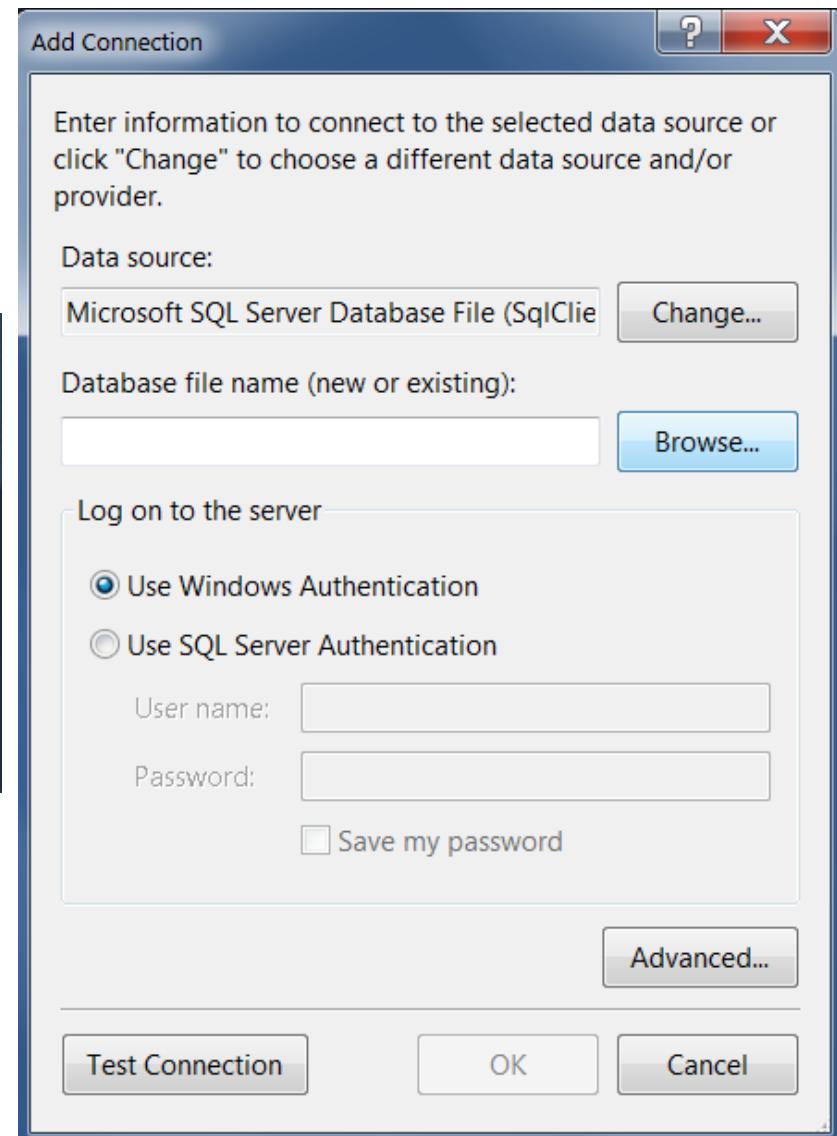
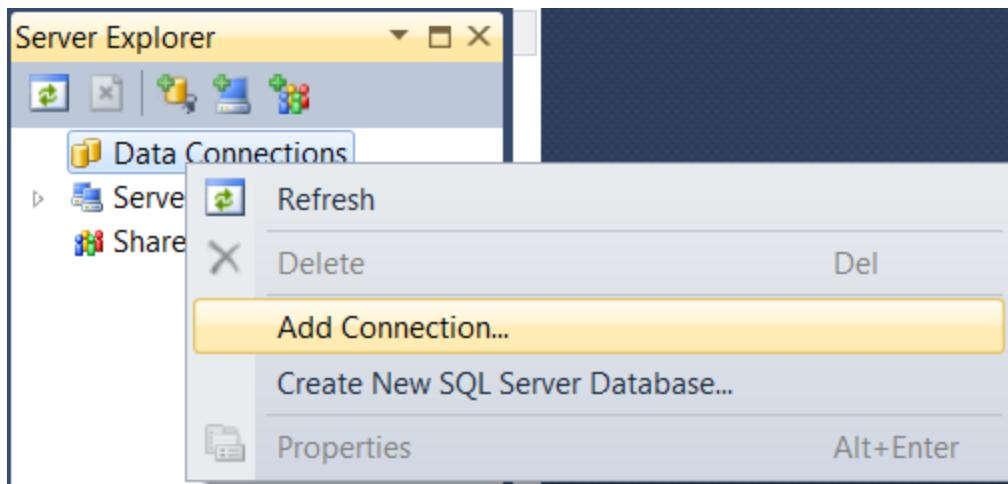
GridView and SqlDataSource

- In Design View



Example

GridView and SqlDataSource



Example

GridView and SqlDataSource

The image shows two windows side-by-side. On the left is a Windows file browser window titled 'Name' with a list of items. The item 'MyWebSiteDatabase.mdf' is selected and highlighted with a blue selection bar. On the right is a 'Add Connection' dialog box for a Microsoft SQL Server Database File (SqlClient). The dialog box contains fields for 'Database file name (new or existing)' set to 'C:\Academic_OC\Courses\2015-2016\Bahar', 'Log on to the server' authentication options ('Use Windows Authentication' selected), and connection save settings. At the bottom are 'Test Connection', 'OK', and 'Cancel' buttons.

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:

Microsoft SQL Server Database File (SqlClient)

Database file name (new or existing):
C:\Academic_OC\Courses\2015-2016\Bahar

Log on to the server

Use Windows Authentication

Use SQL Server Authentication

User name:

Password:

Save my password

Advanced...

Test Connection Cancel

Favorites

Name

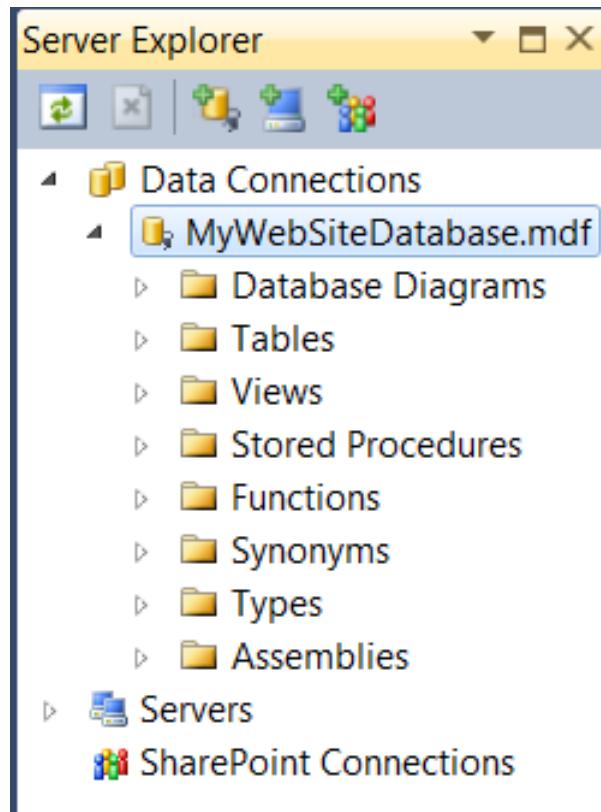
- Downloads
- Dropbox
- Recent Places
- Desktop

MyWebSite

MyWebSiteDatabase.mdf

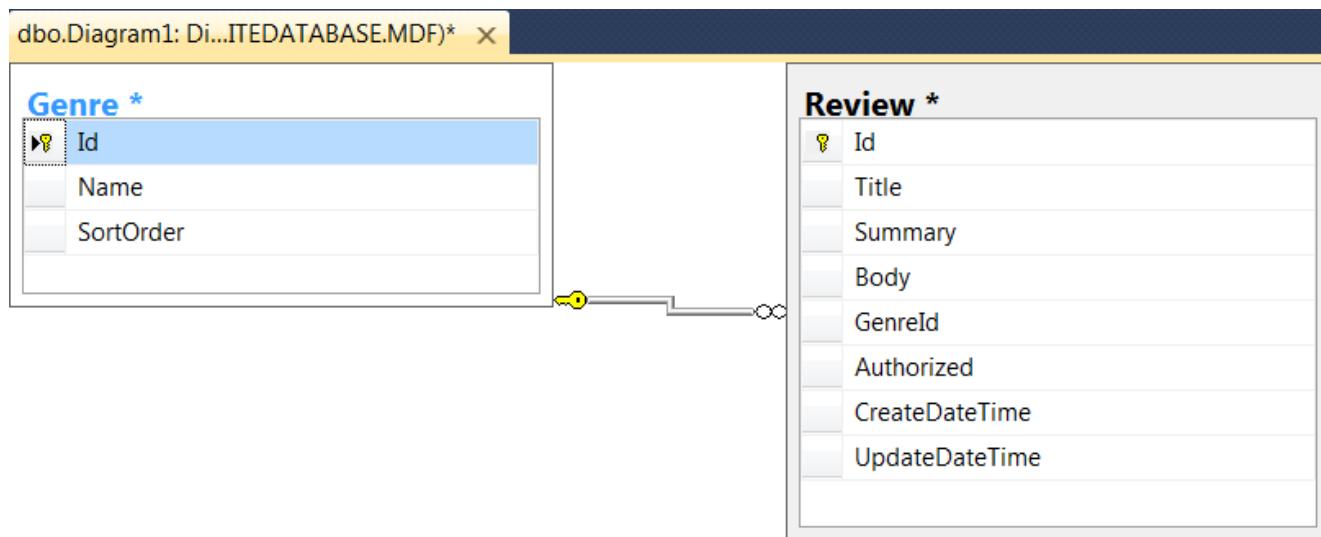
Example

GridView and SqlDataSource



NOTE

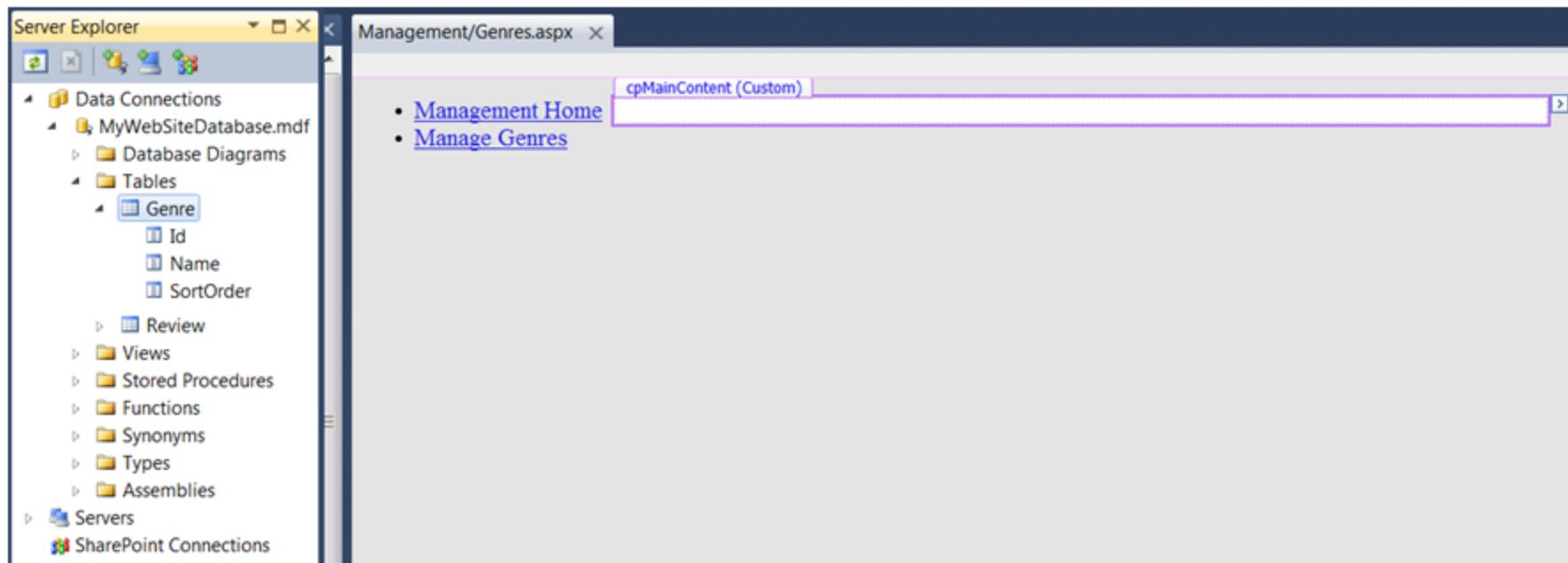
MyWebSiteDatabase.mdf



Example

GridView and SqlDataSource

- In **Genres.aspx**:
 - Drag **Genre** table into the file.



Example

GridView and SqlDataSource

Management/Genres.aspx*

- [Management Home](#)
- [Manage Genres](#)

Id	Name	SortOrder
0	abc	0
1	abc	1
2	abc	2
3	abc	3
4	abc	4

SqlDataSource - SqlDataSource

GridView Tasks

Auto Format...

Choose Data Source: SqlDataSource2 ▾

Configure Data Source...

Refresh Schema

Edit Columns...

Add New Column...

Enable Paging

Enable Sorting

Enable Editing

Enable Deleting

Enable Selection

Edit Templates

Example

GridView and SqlDataSource

- Select all available options of **GridView** control.

The screenshot shows the Visual Studio IDE interface with a web form titled "cpMainContent (Custom)". On the left, there is a navigation menu with links: "Management Home" and "Manage Genres". The main content area contains a "GridView" control with the following data:

	Edit	Delete	Select	Id	Name	SortOrder
	Edit	Delete	Select	0	abc	0
	Edit	Delete	Select	1	abc	1
	Edit	Delete	Select	2	abc	2
	Edit	Delete	Select	3	abc	3
	Edit	Delete	Select	4	abc	4
	Edit	Delete	Select	5	abc	5
	Edit	Delete	Select	6	abc	6
	Edit	Delete	Select	7	abc	7
	Edit	Delete	Select	8	abc	8
	Edit	Delete	Select	9	abc	9

Below the grid, there are page navigation links: "1" and "2". At the bottom of the grid's container, there is a label: "SqlDataSource - SqlDataSource2".

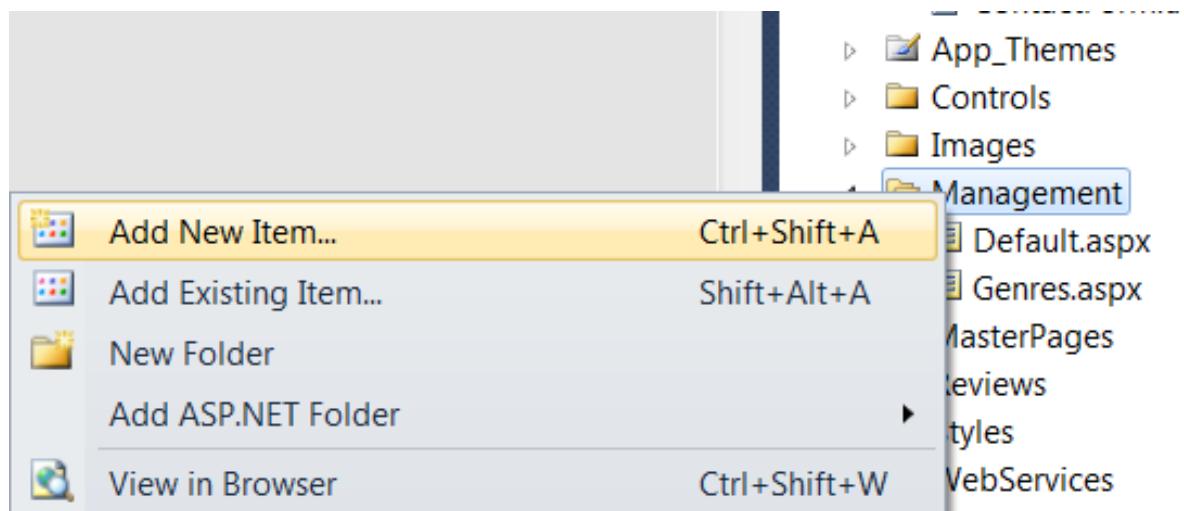
To the right of the grid, a "GridView Tasks" pane is open, listing various configuration options:

- Auto Format...
- Choose Data Source: **SqlDataSource2** (dropdown)
- Configure Data Source...
- Refresh Schema
- Edit Columns...
- Add New Column...
- Enable Paging
- Enable Sorting
- Enable Editing
- Enable Deleting
- Enable Selection
- Edit Templates

Example

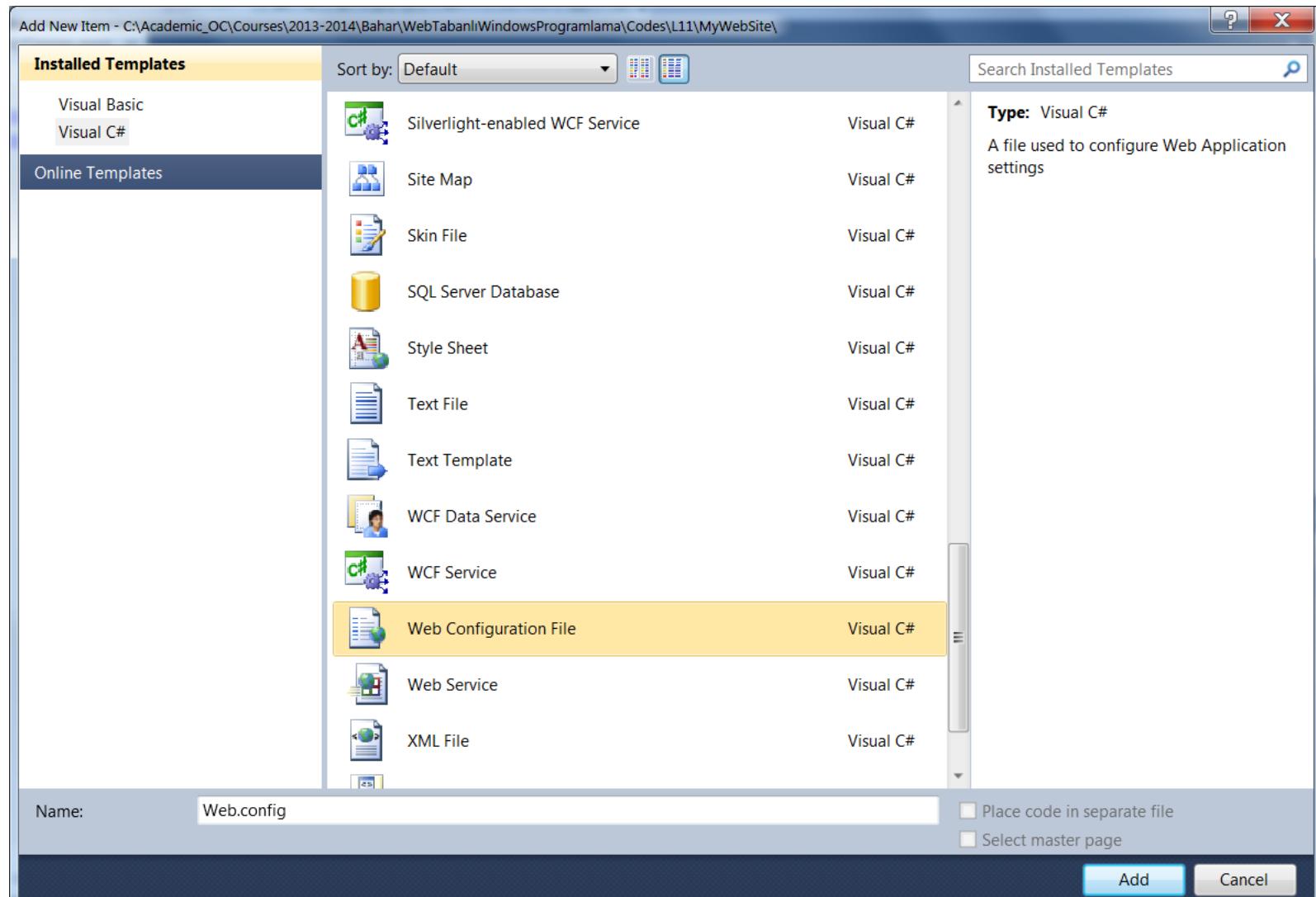
GridView and SqlDataSource

- Add New Item → Management folder



Example

GridView and SqlDataSource



Example

GridView and SqlDataSource

- Add a <pages> element
 - Set theme to an empty string → Disable the theme for the *entire Management section* of the web site.

```
<?xml version="1.0"?>
<configuration>
    <system.web>
        <pages theme=""></pages>
    </system.web>
</configuration>
```

Example

GridView and SqlDataSource

- View **Genres . aspx** in browser.

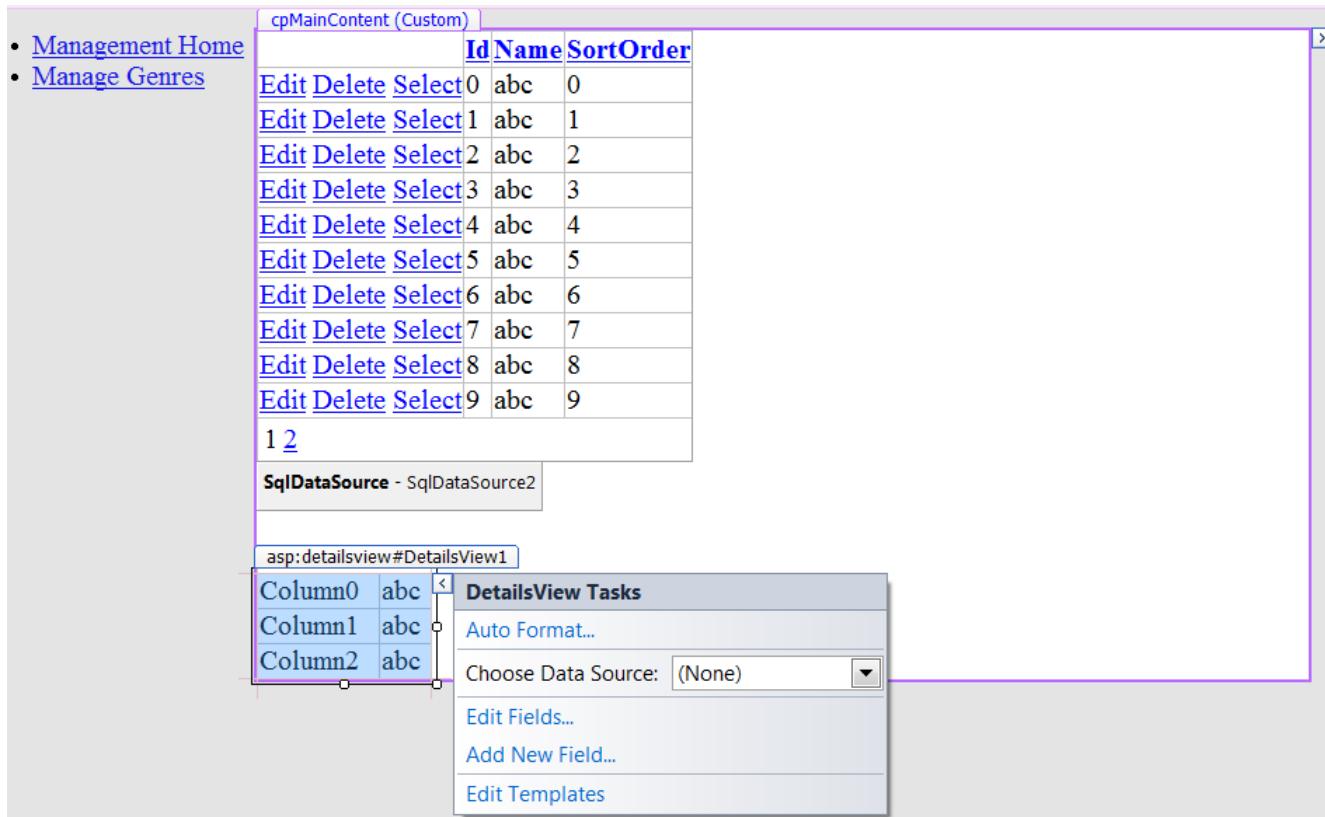
-
- [Management Home](#)
 - [Manage Genres](#)

	Id	Name	SortOrder
Edit	4	Pop	12
Delete	5	Jazz	4
Select	6	Rock	2
Edit	7	Pop Rock	7
Delete	8	Classical	6
Select	9	Blues	1
Edit	10	Chicago Jazz	3
Delete	11	Country	5

Example

Inserting Data

- In Design View of **Genres . aspx**:
 - Drag a **DetailsView** below the **GridView**



Example

Inserting Data

cpMainContent (Custom)

	<u>Id</u>	<u>Name</u>	<u>SortOrder</u>
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
Edit Delete Select	Databound	Databound	Databound
1 2			

SqDataSource - SqDataSource1

asp:detailsview#DetailsView1

Column0	abc
Column1	abc
Column2	abc

DetailsView Tasks

Auto Format...

Choose Data Source: (None)

(None)
SqDataSource1
<New data source...>

Edit Fields...
Add New Field...
Edit Templates



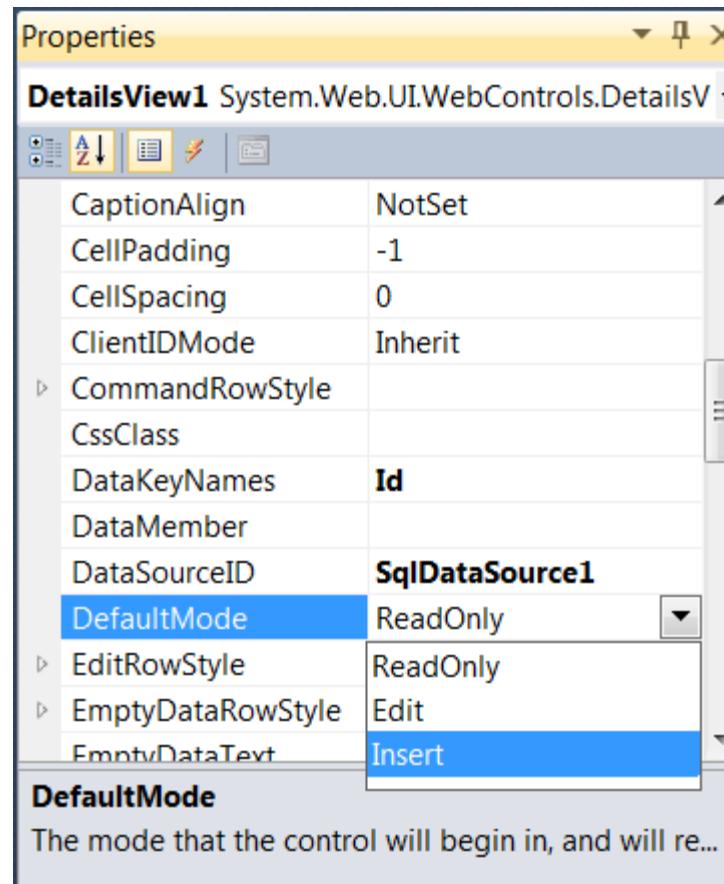
Example

Inserting Data

Example

Inserting Data

DefaultMode = Insert



Example

Inserting Data

cpMainContent (Custom)

	<u>Id</u>	<u>Name</u>	<u>SortOrder</u>
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
<u>Edit</u> <u>Delete</u> <u>Select</u>	Databound	Databound	Databound
	1	<u>2</u>	

SqIDataSource - SqIDataSource1

asp:detailsview#DetailsView1

Name	<input type="text"/>	>
SortOrder	<input type="text"/>	□
<u>Insert</u> <u>Cancel</u>		

Example

Inserting Data

```
<asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False"
    DataKeyNames="Id" DataSourceID="SqlDataSource1" DefaultMode="Insert"
    Height="50px" Width="125px">
    <Fields>
        <asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
            ReadOnly="True" SortExpression="Id" />
        <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
        <asp:BoundField DataField="SortOrder" HeaderText="SortOrder"
            SortExpression="SortOrder" />
        <asp:CommandField ShowInsertButton="True" />
    </Fields>
</asp:DetailsView>
```

Example

Inserting Data

- View **Genres .aspx** in browser

-
- [Management Home](#)
 - [Manage Genres](#)

	<u>Id</u>	<u>Name</u>	<u>SortOrder</u>
Edit Delete Select	4	Pop	12
Edit Delete Select	5	Jazz	4
Edit Delete Select	6	Rock	2
Edit Delete Select	7	Pop Rock	7
Edit Delete Select	8	Classical	6
Edit Delete Select	9	Blues	1
Edit Delete Select	10	Chicago Jazz	3

Name	<input type="text"/>
SortOrder	<input type="text"/>
Insert Cancel	

Example

Inserting Data

- [Management Home](#)
- [Manage Genres](#)

	<u>Id</u>	<u>Name</u>	<u>SortOrder</u>
Edit	4	Pop	12
Delete	5	Jazz	4
Select	6	Rock	2
Edit	7	Pop Rock	7
Delete	8	Classical	6
Select	9	Blues	1
Edit	10	Chicago Jazz	3

Name	Latin
SortOrder	8
Insert Cancel	

- [Management Home](#)
- [Manage Genres](#)

	<u>Id</u>	<u>Name</u>	<u>SortOrder</u>
Edit	4	Pop	12
Delete	5	Jazz	4
Select	6	Rock	2
Edit	7	Pop Rock	7
Delete	8	Classical	6
Select	9	Blues	1
Edit	10	Chicago Jazz	3
Select	12	Latin	8

Name	
SortOrder	
Insert Cancel	

NOTE

- Visual Studio creates:
 - A **SqlDataSource** control for what database to access
 - A connection string in the **web.config** file.

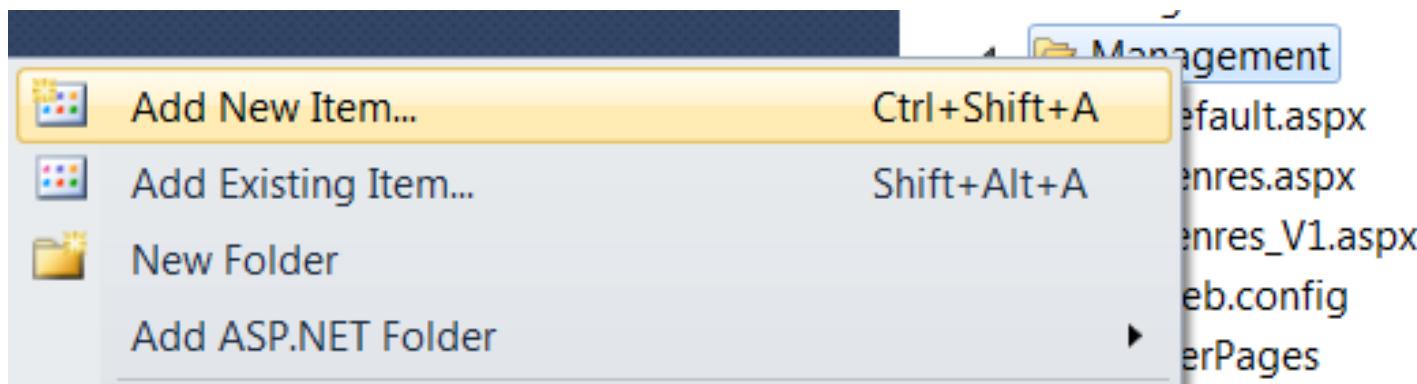
```
<configuration>

<connectionStrings>
    <add name="MyWebSiteDatabaseConnectionString1" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=C:\\
        providerName="System.Data.SqlClient" />
</connectionStrings>
<system.web>
    <compilation debug="false" targetFramework="4.0" />
    <pages theme="Monochrome">
        <controls>
            <add tagPrefix="Ege" tagName="Banner" src("~/Controls/Banner.ascx")/>
        </controls>
    </pages>
</system.web>
```

Example

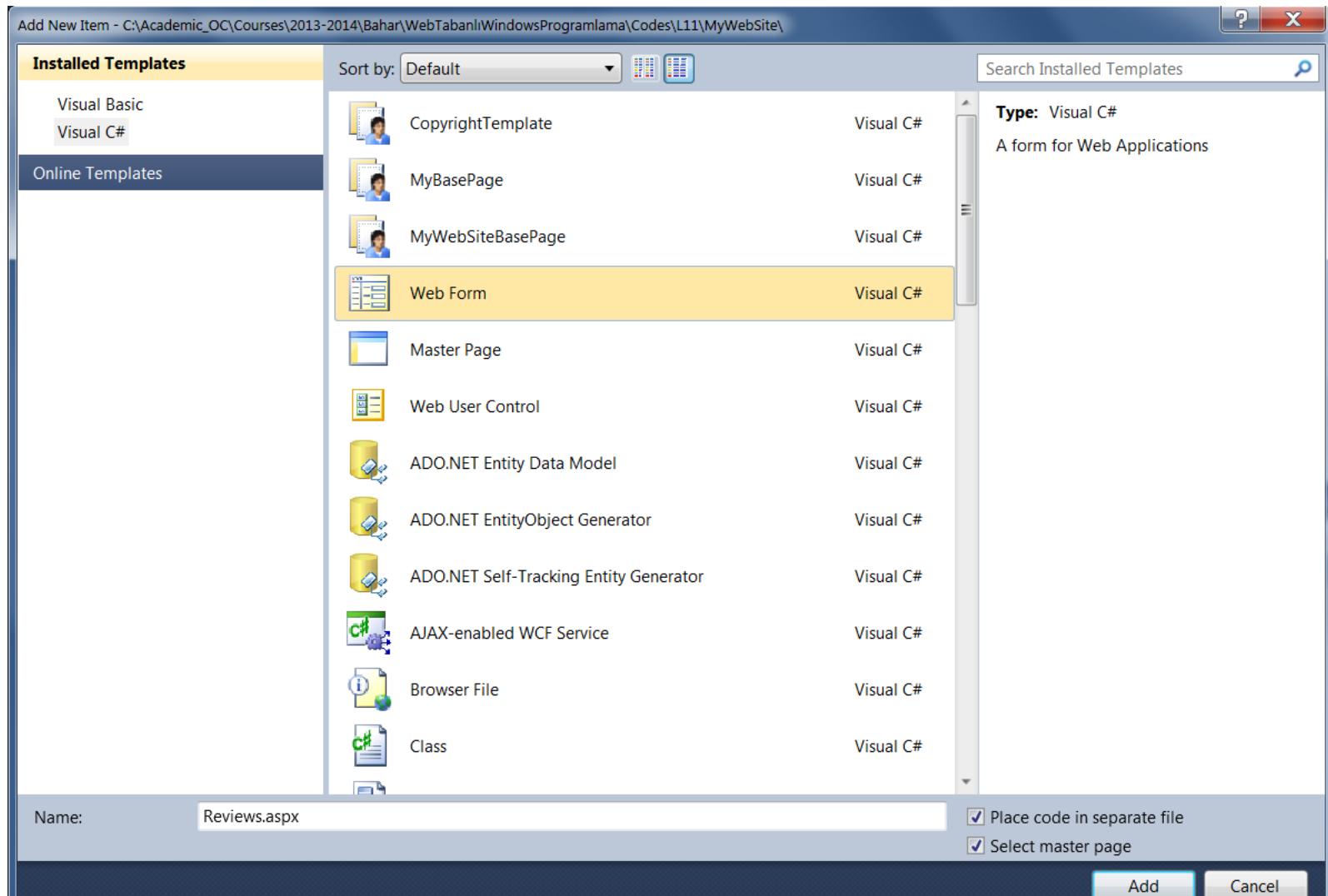
Filtering Data

- Add New Item → Management folder



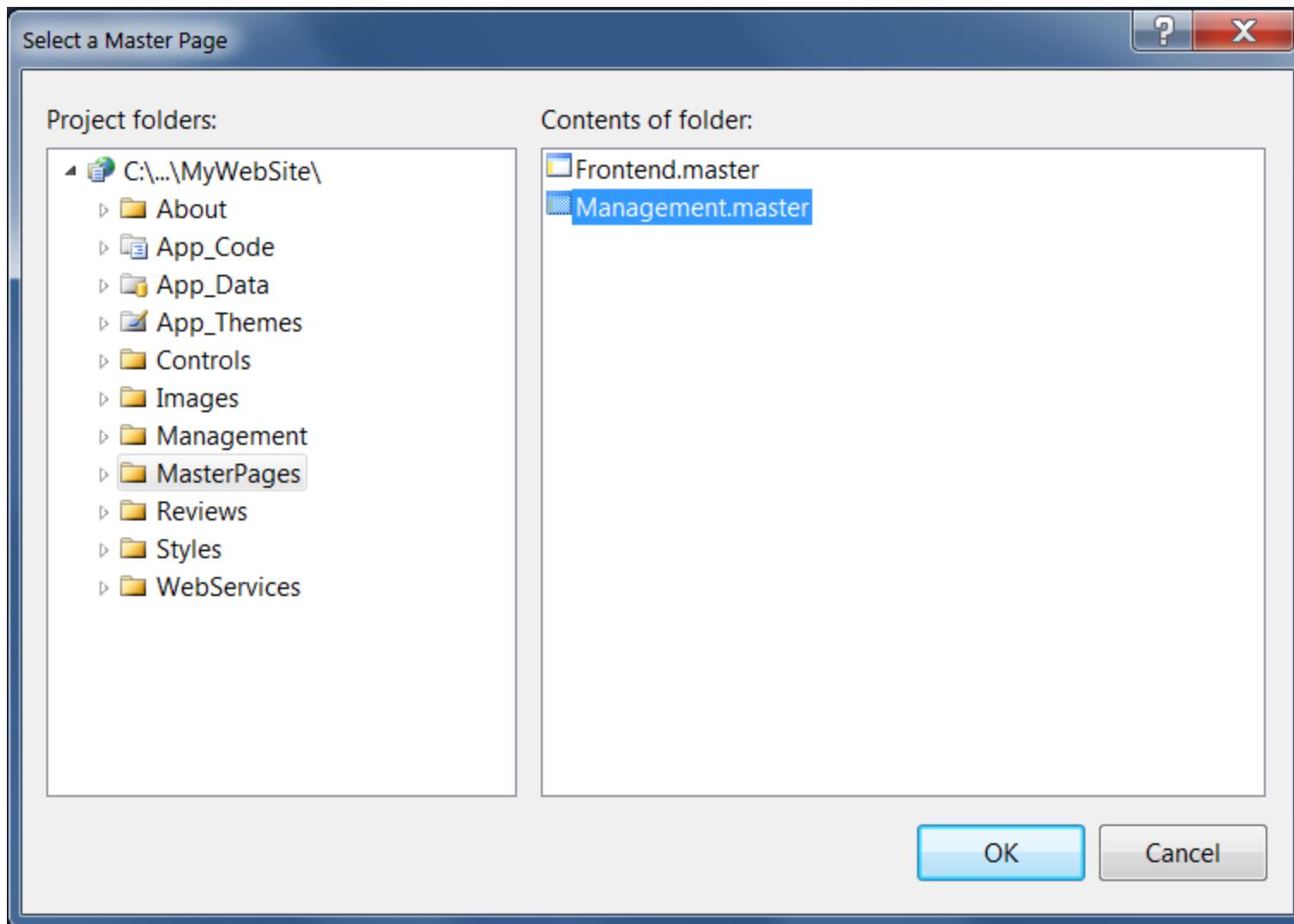
Example

Filtering Data



Example

Filtering Data



Example

Filtering Data

- Change **Title = Management-Reviews**

```
<%@ Page Title="Management - Reviews" Language="C#" MasterPageFile="~/MasterPages/Management.master"

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
</asp:Content>
```

Example

Filtering Data

- Open → **Management.master**

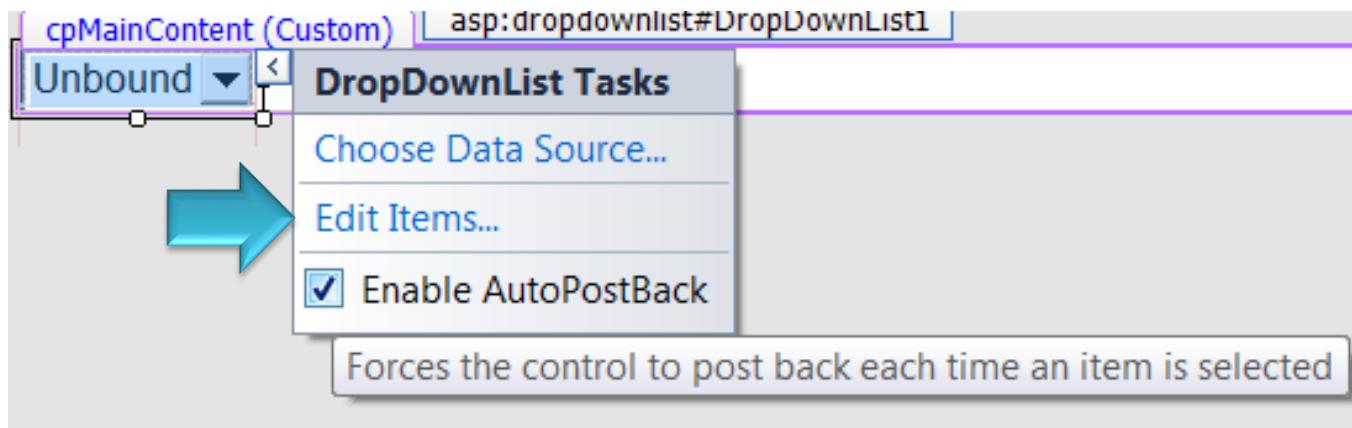
```
<ul>
  <li><a id="A1" href="~/Management/Default.aspx" runat="server">Management Home</a></li>
  <li><a id="A2" href="~/Management/Genres.aspx" runat="server">Manage Genres</a></li>
  <li><a id="A3" href="~/Management/Reviews.aspx" runat="server">Manage Reviews</a></li>
</ul>
```



Example

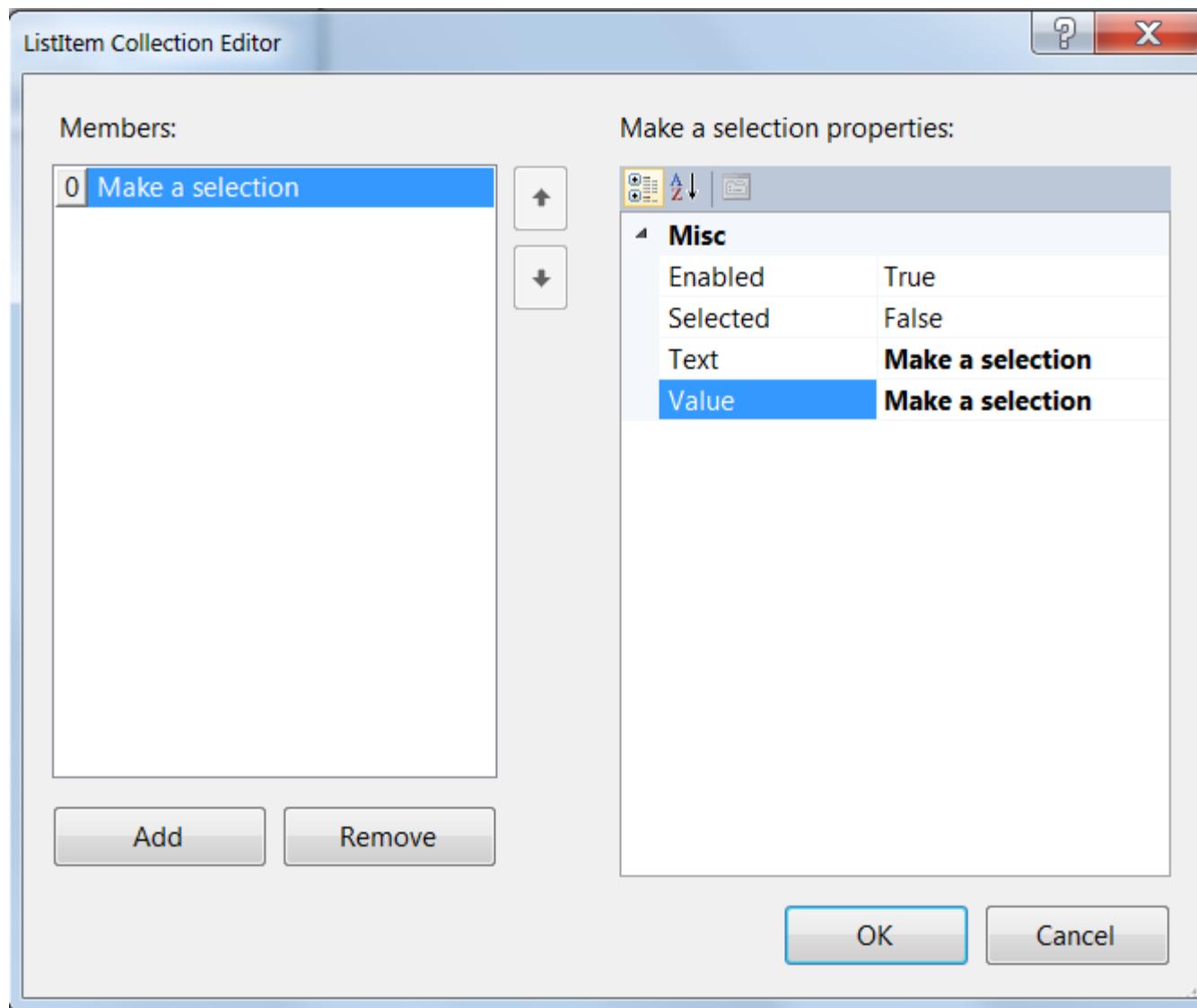
Filtering Data

- In Design View of **Reviews.aspx**:
 - Drag a **DropDownList**
 - **Enable AutoPostBack**
 - Click **Edit Items**



Example

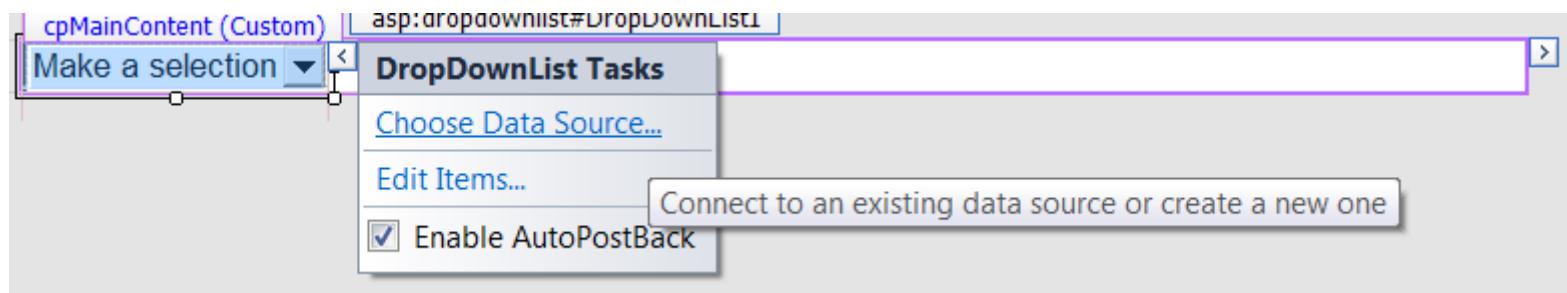
Filtering Data



Example

Filtering Data

Choose Data Source





Choose a Data Source

Select a data source:

(None) ▾

(None)

<New data source...>

ownList:

Select a data field for the value of the DropDownList:

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 any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.

Specify an ID for the data source:

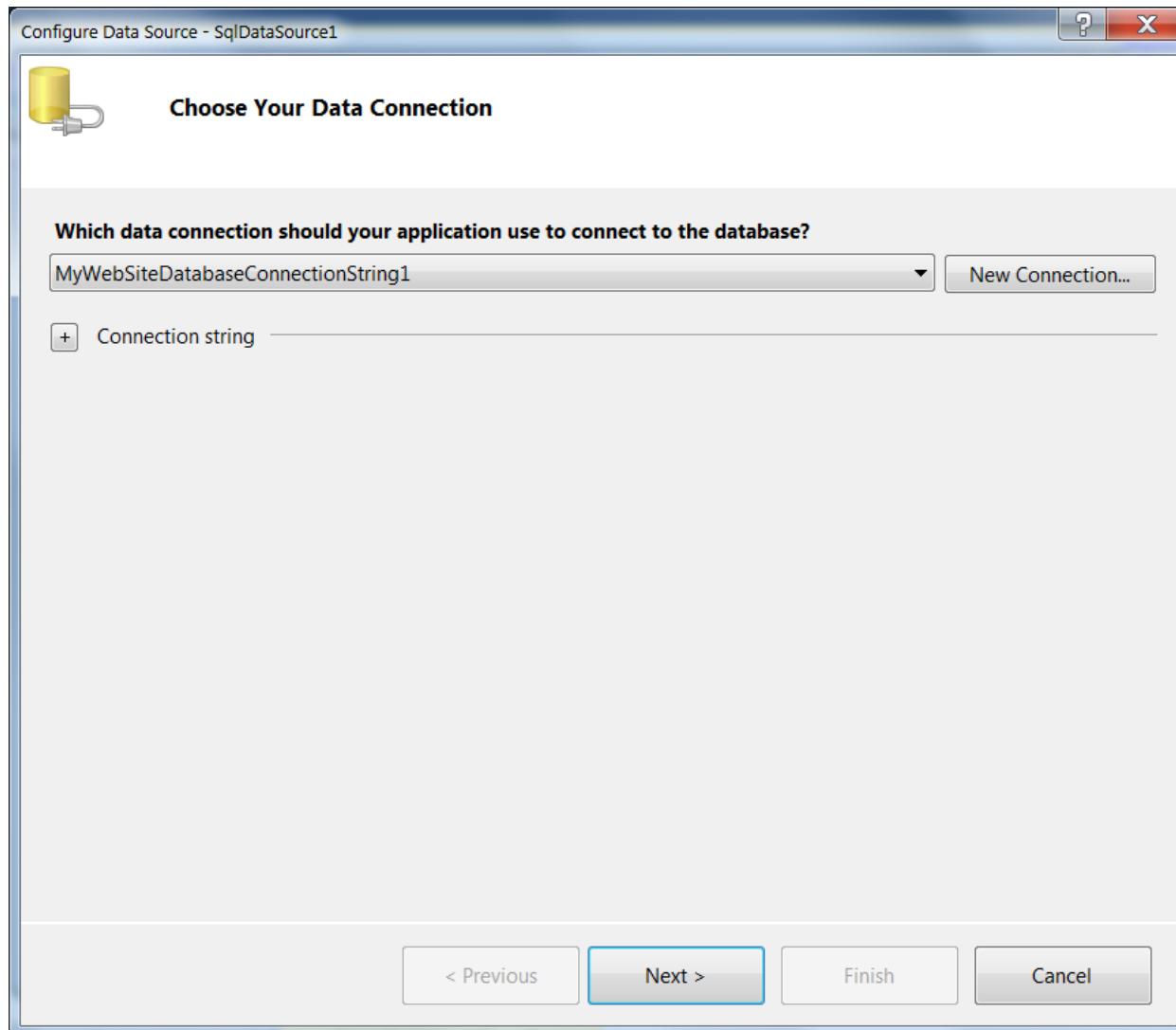
SqlDataSource1

OK

Cancel

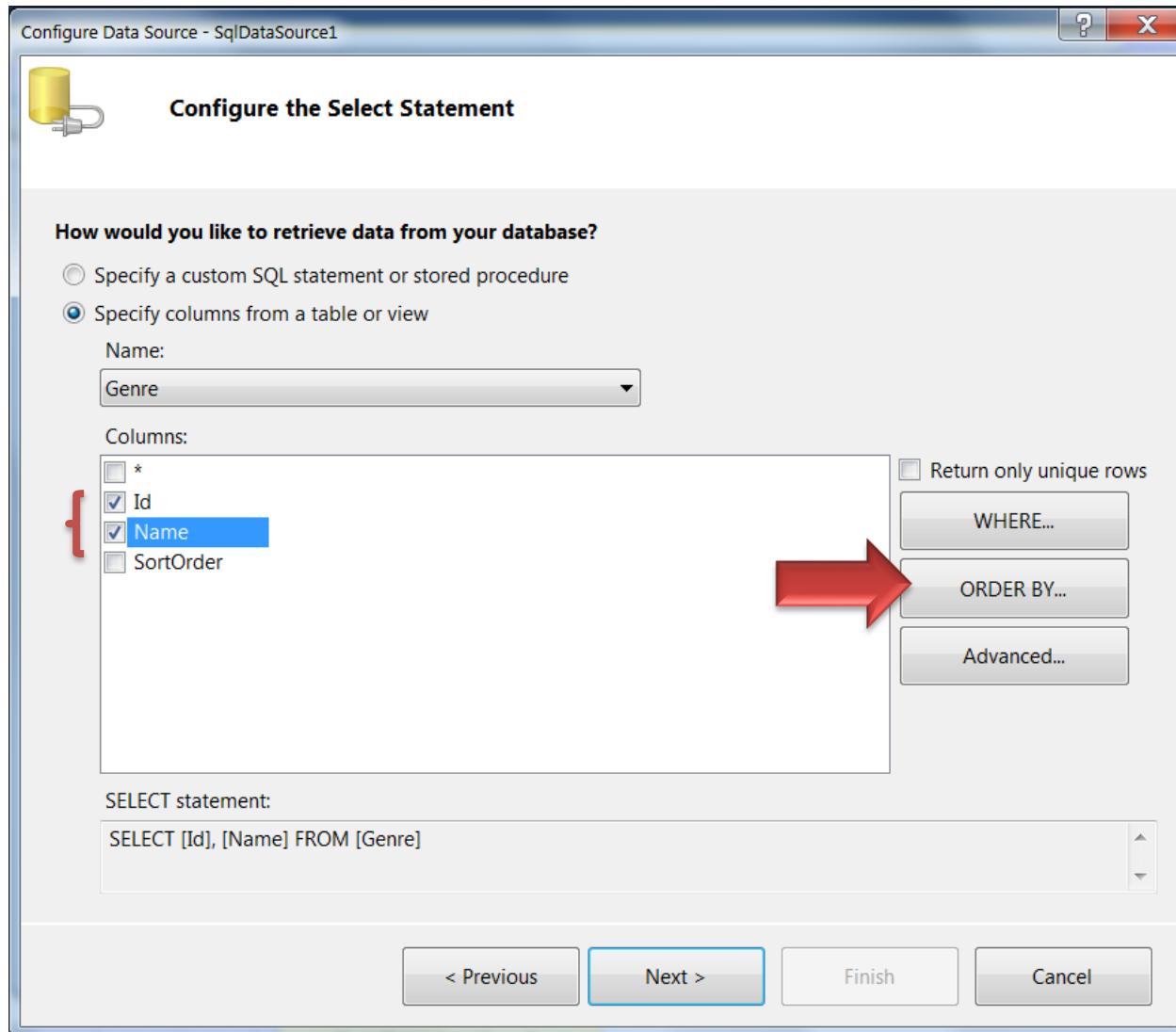
Example

Filtering Data



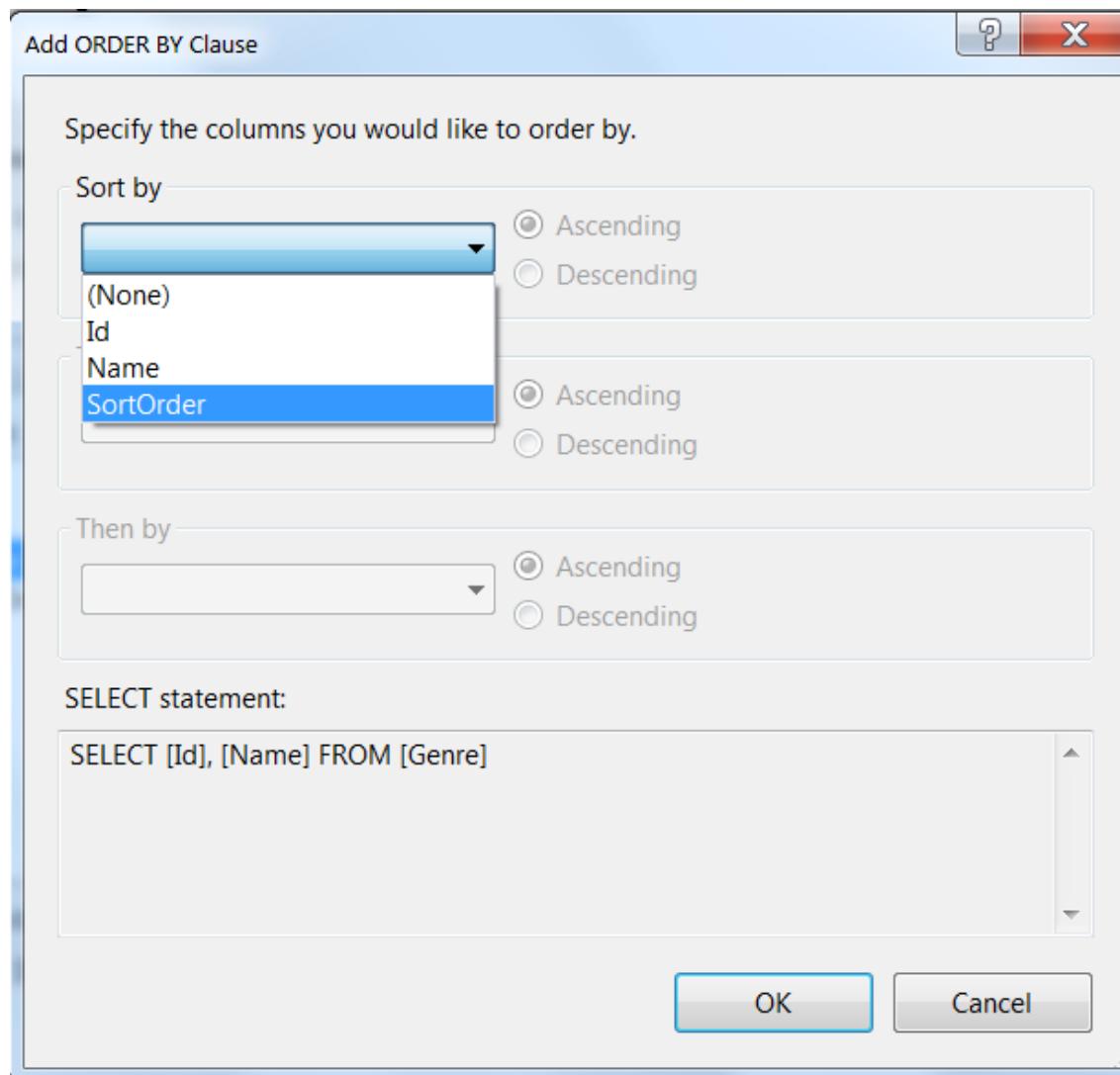
Example

Filtering Data



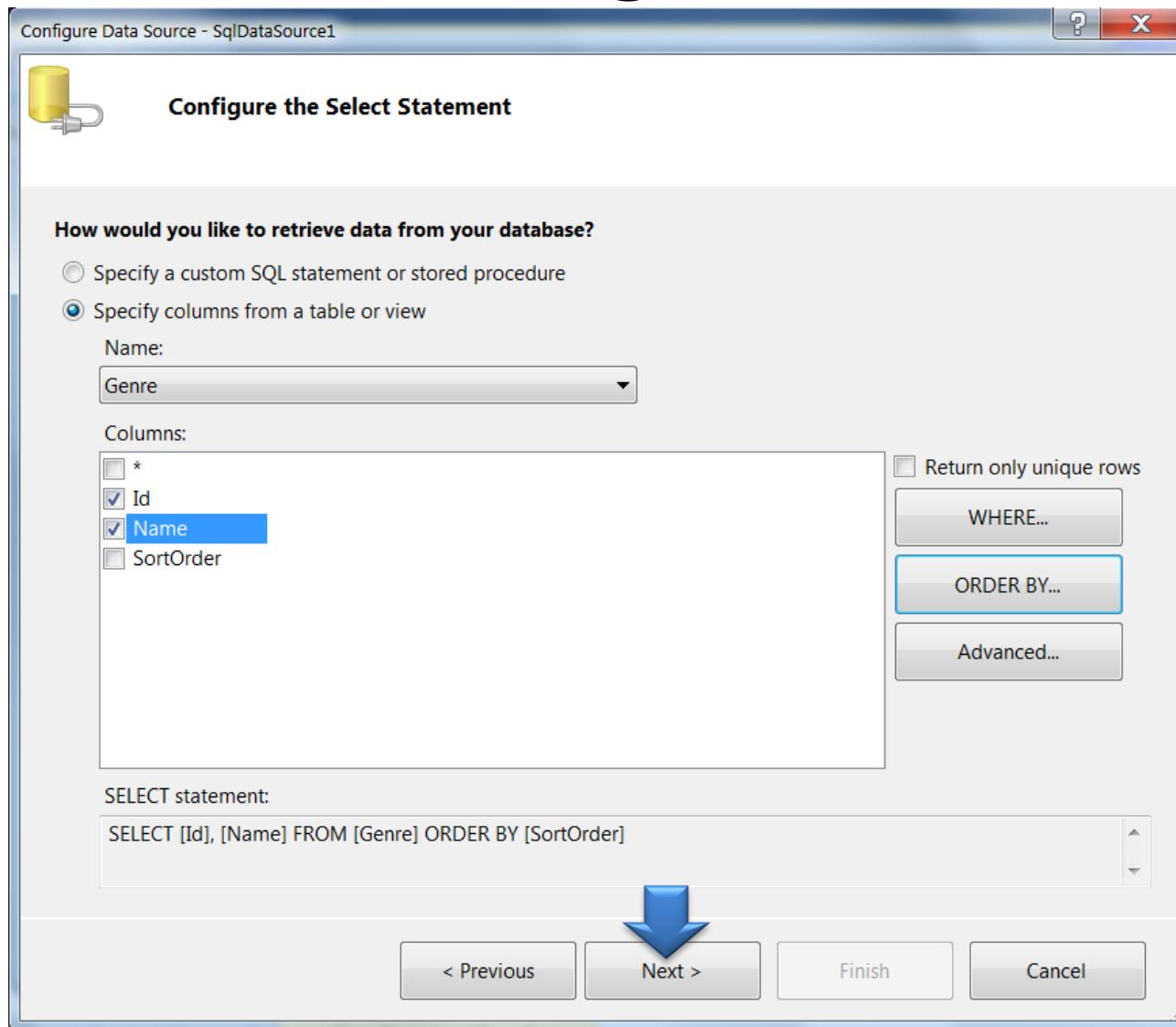
Example

Filtering Data



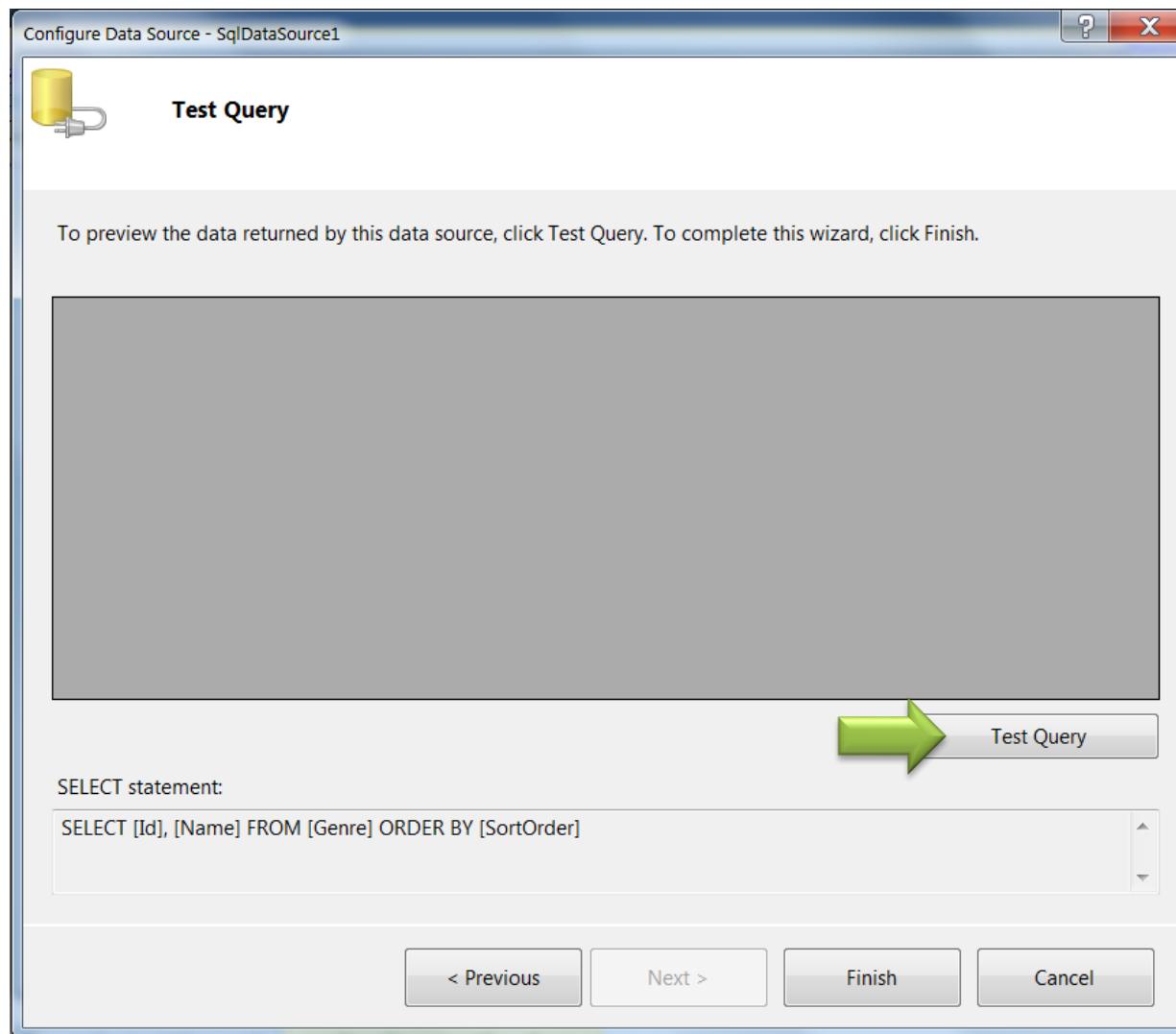
Example

Filtering Data



Example

Filtering Data



Example

Filtering Data

Configure Data Source - SqlDataSource1

 **Test Query**

To preview the data returned by this data source, click Test Query. To complete this wizard, click Finish.

Id	Name
9	Blues
6	Rock
10	Chicago Jazz
5	Jazz
13	Country
8	Classical
7	Pop Rock
12	Latin
4	Pop

Test Query

SELECT statement:

```
SELECT [Id], [Name] FROM [Genre] ORDER BY [SortOrder]
```

< 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](#)

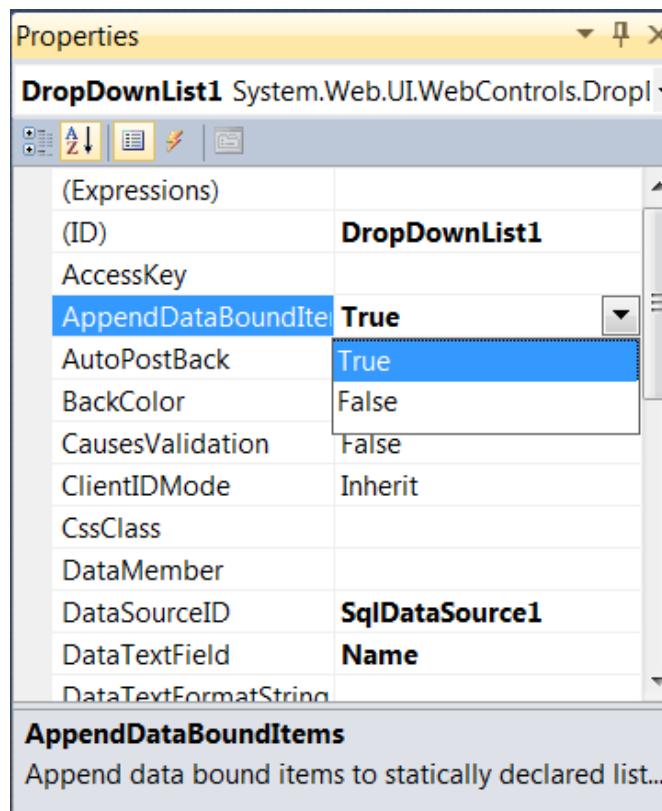
OK

Cancel

Example

Filtering Data

- In DropDownList properties:
 - **AppendDataBoundItems = True**



Example

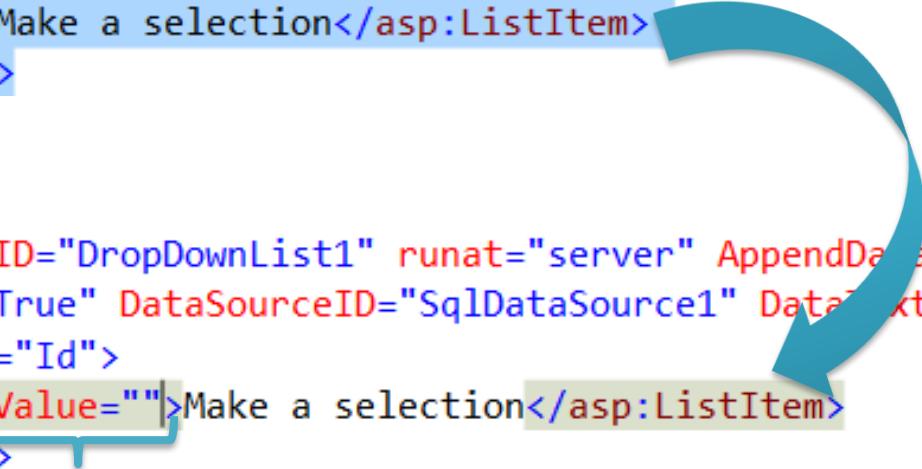
Filtering Data

- In Source View:

```
<asp:DropDownList ID="DropDownList1" runat="server" AppendDataBoundItems="True"  
    AutoPostBack="True" DataSourceID="SqlDataSource1" DataTextField="Name"  
    DataValueField="Id">  
    <asp:ListItem>Make a selection</asp:ListItem>  
</asp:DropDownList>
```



```
<asp:DropDownList ID="DropDownList1" runat="server" AppendDataBoundItems="True"  
    AutoPostBack="True" DataSourceID="SqlDataSource1" DataTextField="Name"  
    DataValueField="Id">  
    <asp:ListItem Value="">Make a selection</asp:ListItem>  
</asp:DropDownList>
```

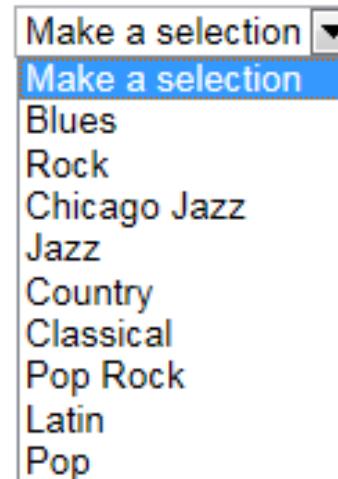


Example

Filtering Data

- View **Reviews.aspx** in browser.

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)



Example

Applying Filter

- In Design View of **Reviews.aspx**:
 - On top of the existing **DropDownList**, drag a **GridView**

The screenshot shows the Visual Studio Design View for a web page. On the left, there is a navigation menu with links: "Management Home", "Manage Genres", and "Manage Reviews". The main area contains a **GridView** control with three columns labeled "Column0", "Column1", and "Column2". Each row contains the value "abc" in all three columns. Below the grid, a status bar shows "Databound" and "SqlDataSource - SqlDataSource1". A context menu titled "GridView Tasks" is open over the grid, listing options: "Auto Format...", "Choose Data Source: (None)", "Edit Columns...", "Add New Column...", and "Edit Templates".

Column0	Column1	Column2
abc	abc	abc

Databound ▾
SqlDataSource - SqlDataSource1

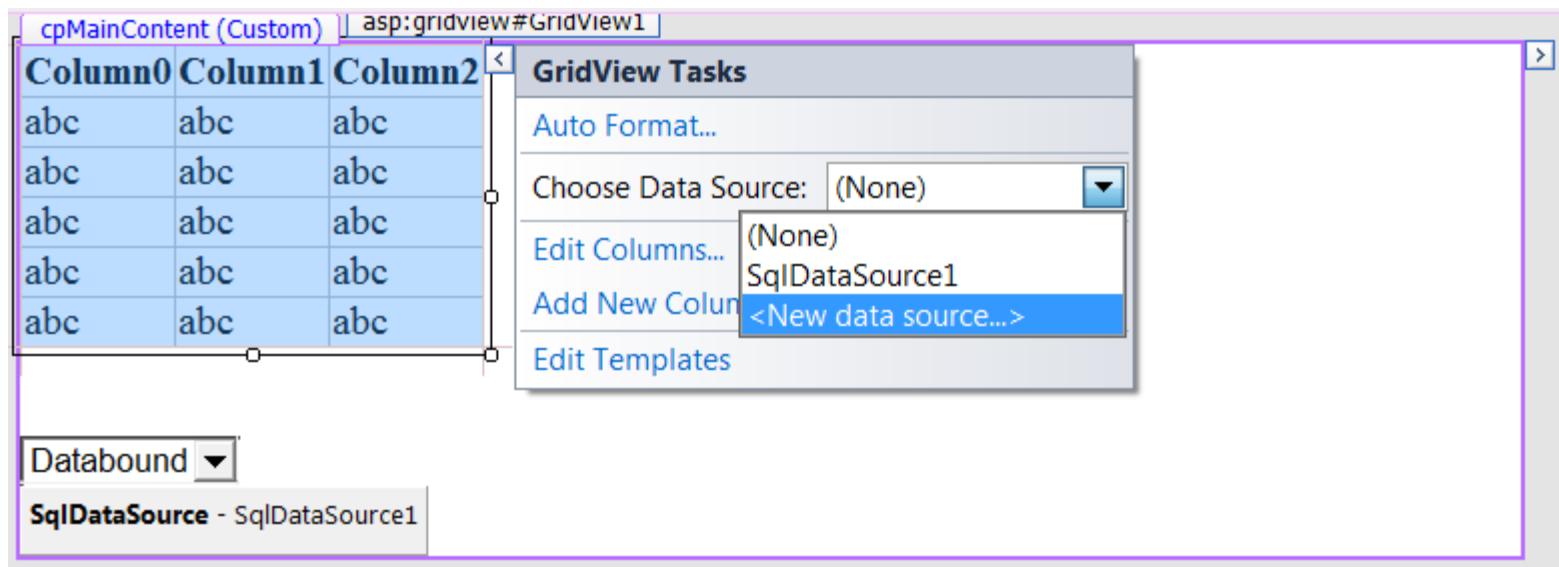
GridView Tasks

- Auto Format...
- Choose Data Source: (None)
- Edit Columns...
- Add New Column...
- Edit Templates

Example

Applying Filter

Choose Data Source → <New data source>





Choose a Data Source Type

Where will the application get data from?



Access
Database



Database



Entity



LINQ



Object



Site Map



XML File

Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.

Specify an ID for the data source:

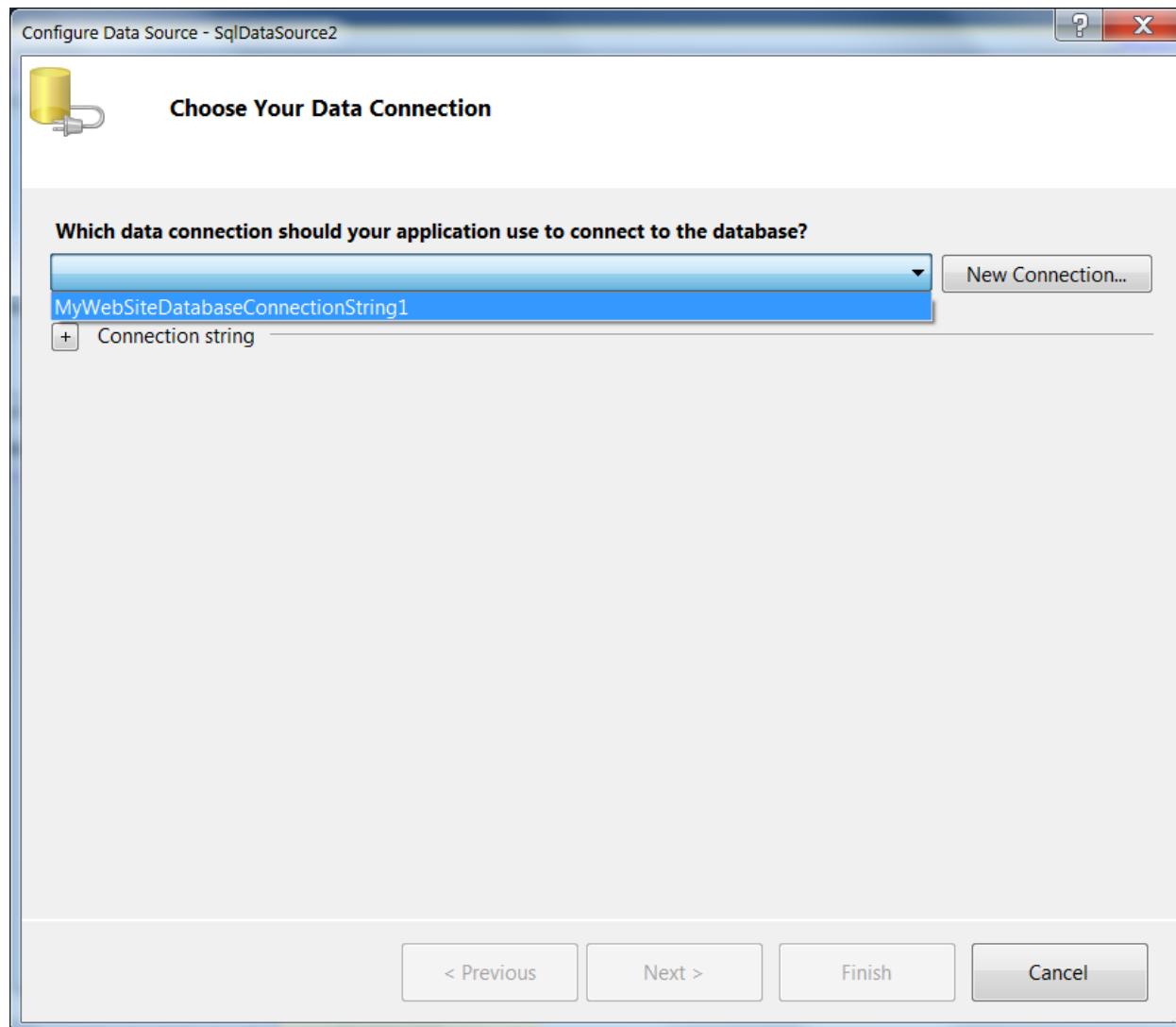
SqlDataSource2

OK

Cancel

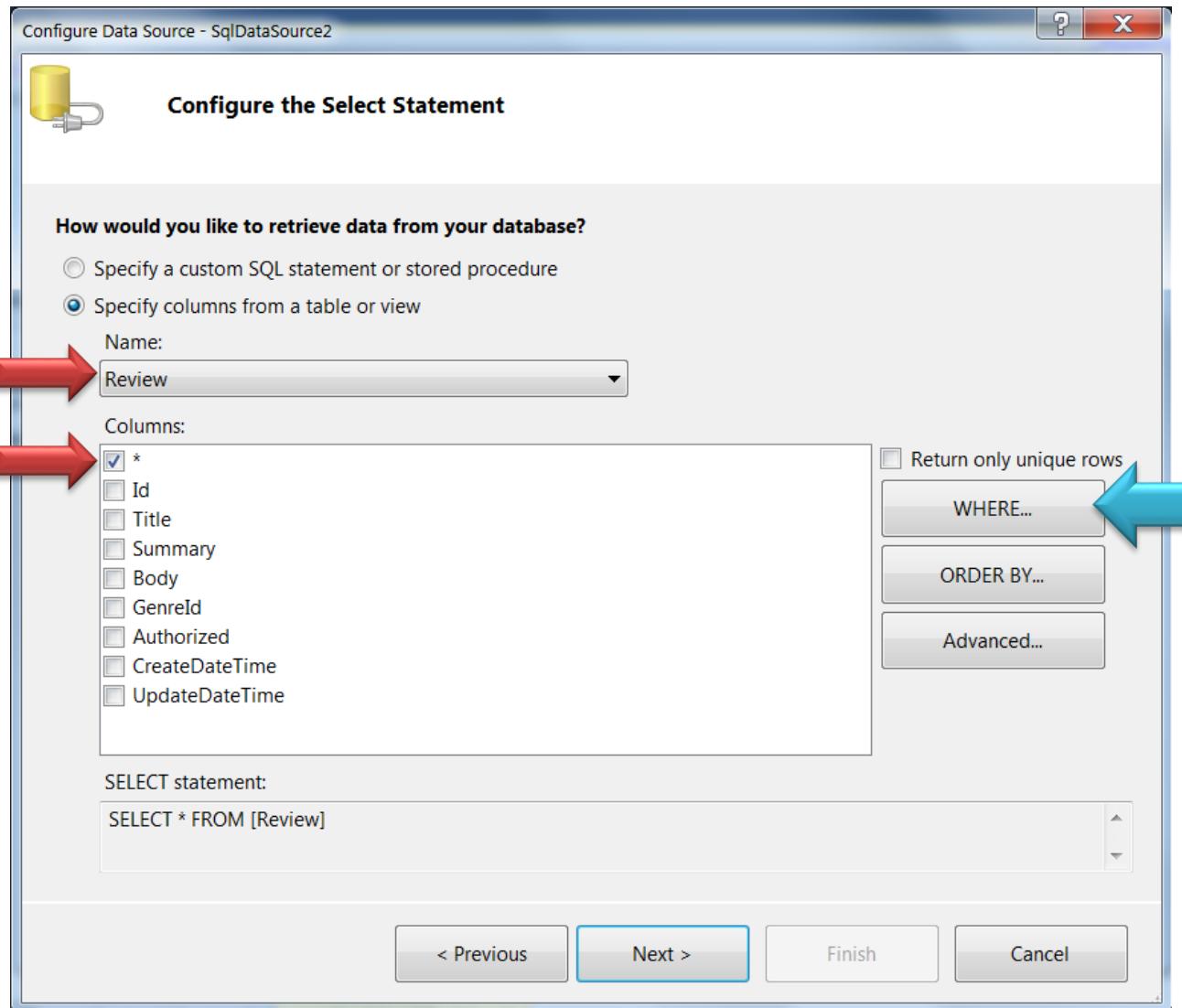
Example

Applying Filter

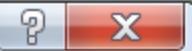


Example

Applying Filter



Add WHERE Clause



Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column:

GenreId

Operator:

=

Source:

Control

SQL Expression:

[GenreId] = @GenreId

Parameter properties

Control ID:

DropDownList1

Default value:

(empty)

Value:

DropDownList1.SelectedValue

Add



WHERE clause:

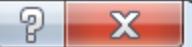
SQL Expression	Value
(empty)	(empty)

Remove

OK

Cancel

Add WHERE Clause



Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column:

Parameter properties

Operator:

Source:

SQL Expression:

Value:

Add

WHERE clause:

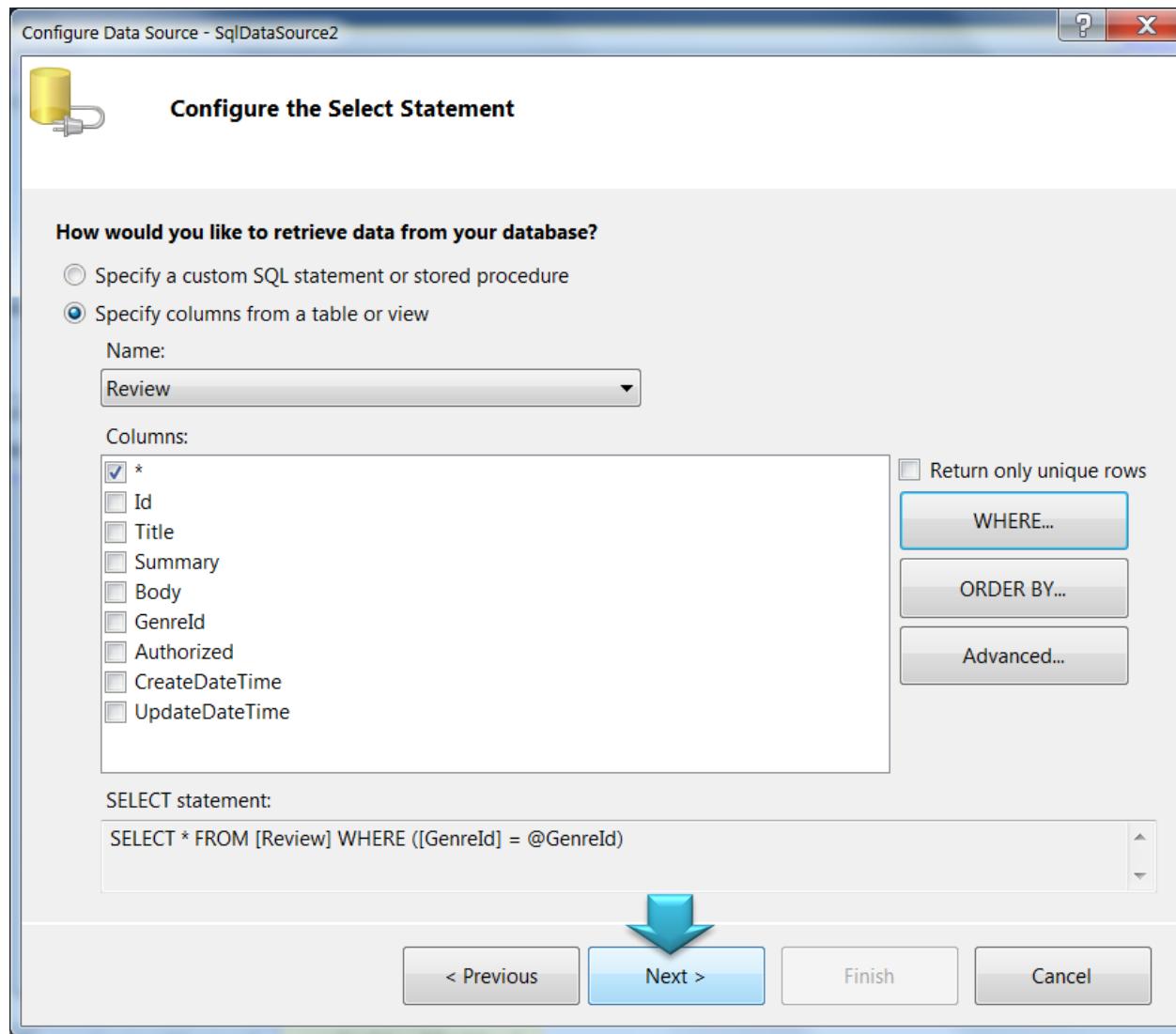
SQL Expression	Value	
[GenreId] = @GenreId	DropDownList1.Sele...	<button>Remove</button>

OK

Cancel

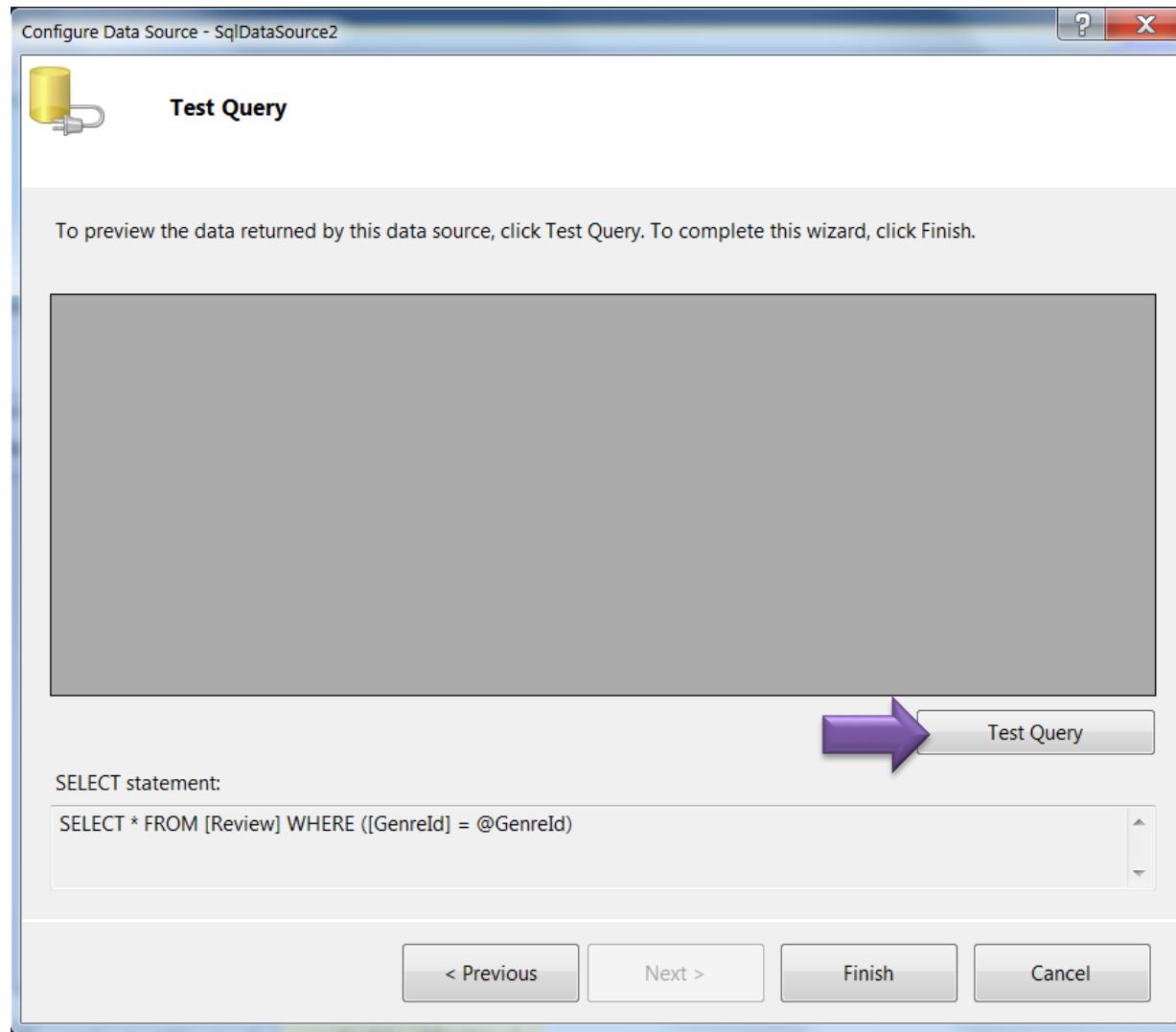
Example

Applying Filter



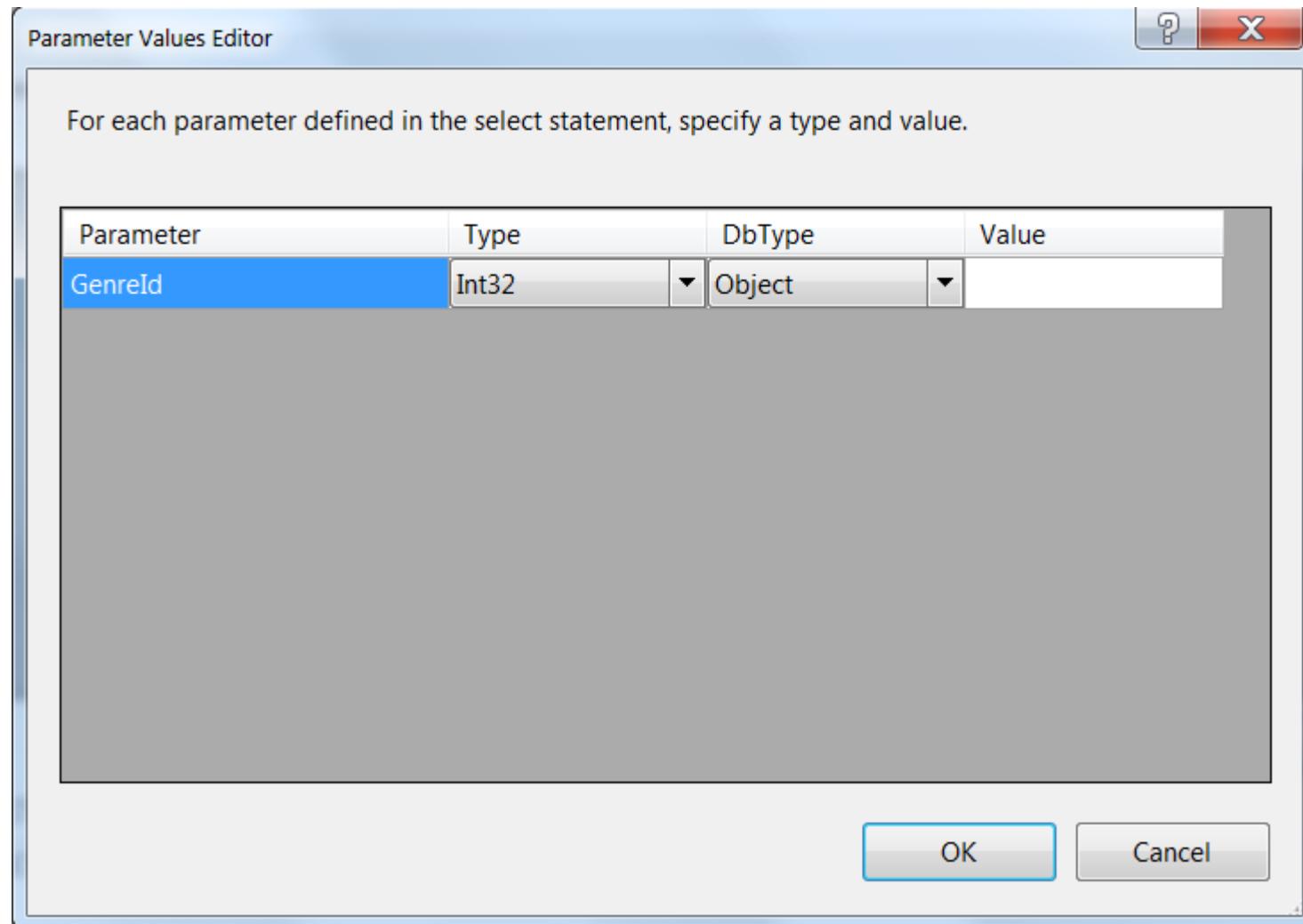
Example

Applying Filter



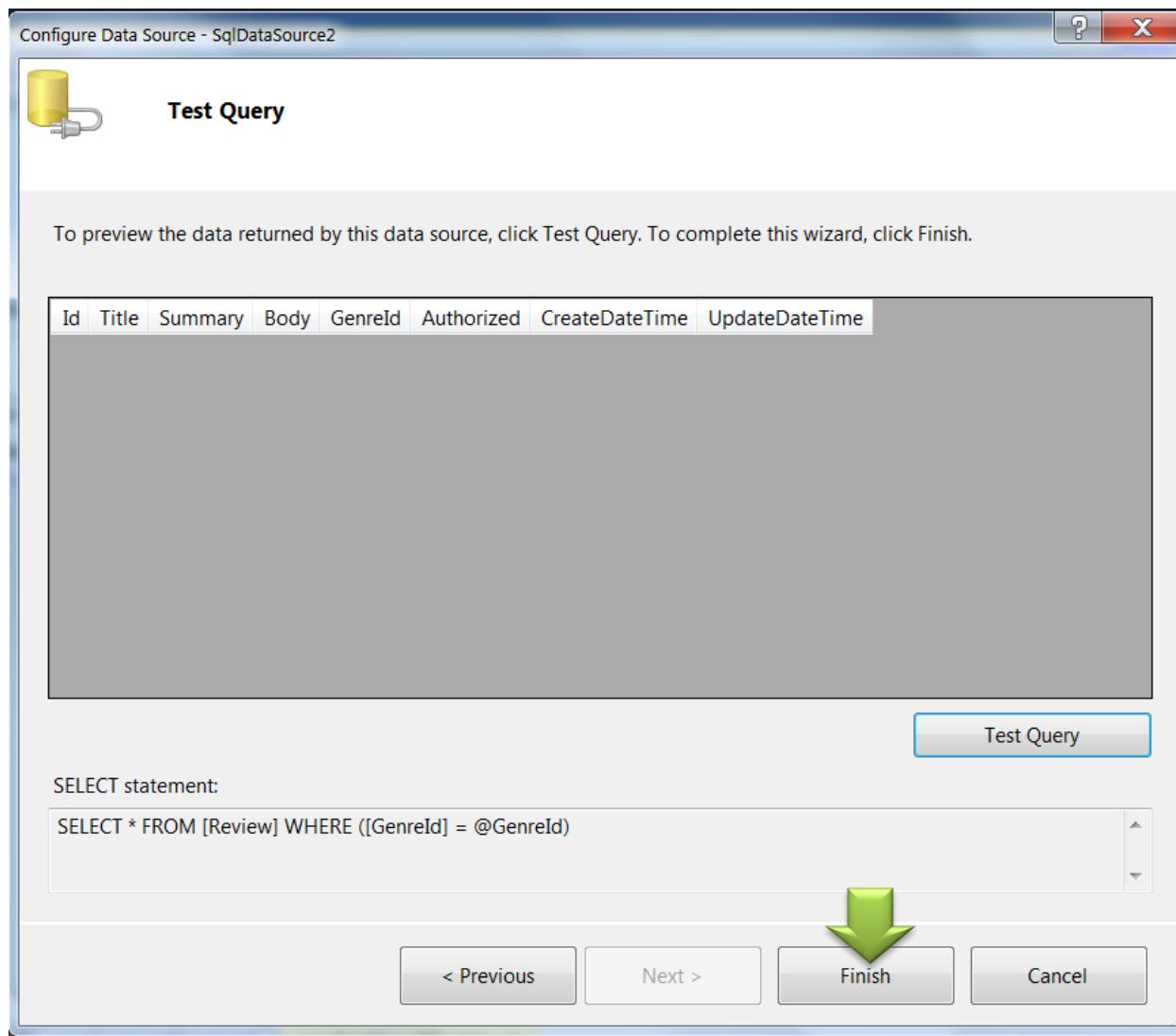
Example

Applying Filter



Example

Applying Filter



Example

Applying Filter

• [Management Home](#)

• [Manage Genres](#)

• [Manage Reviews](#)

cpMainContent (Custom)	asp:gridview#GridView1						
Id	Title	Summary	Body	GenreId	Authorized	CreateDateTime	UpdateDateTime
0	abc	abc	abc	0	<input type="checkbox"/>	14.05.2014 00:00:00	14.05.2014 00:00:00
1	abc	abc	abc	1	<input checked="" type="checkbox"/>	14.05.2014 00:00:00	14.05.2014 00:00:00
2	abc	abc	abc	2	<input type="checkbox"/>	14.05.2014 00:00:00	14.05.2014 00:00:00
3	abc	abc	abc	3	<input checked="" type="checkbox"/>	14.05.2014 00:00:00	14.05.2014 00:00:00
4	abc	abc	abc	4	<input type="checkbox"/>	14.05.2014 00:00:00	14.05.2014 00:00:00

SqIDataSource - SqIDataSource2

Databound ▾

SqIDataSource - SqIDataSource1

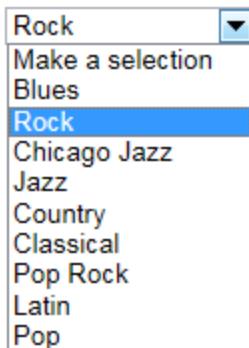
Example

Applying Filter

- View **Reviews.aspx** in browser

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Id	Title	Summary	Body	GenreId	Authorized	CreateDateTime	UpdateDateTime
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

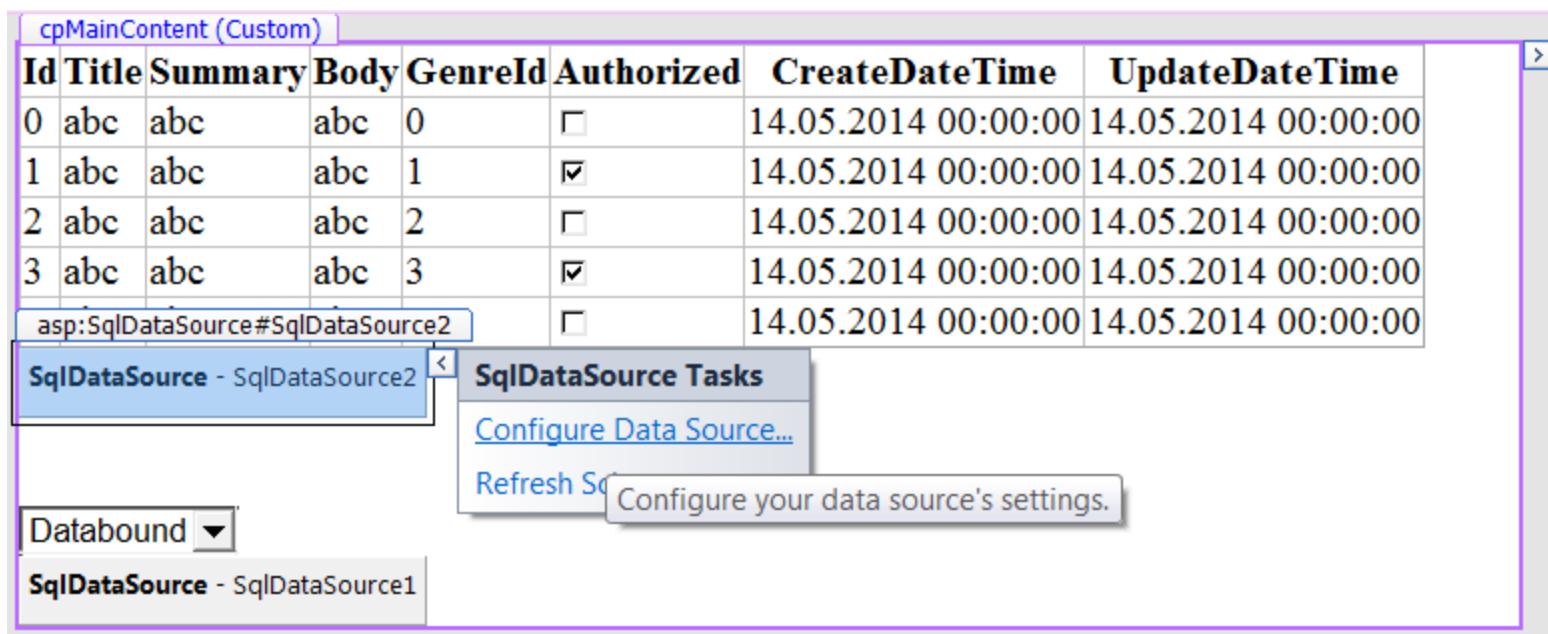


Example

Customizing GridView

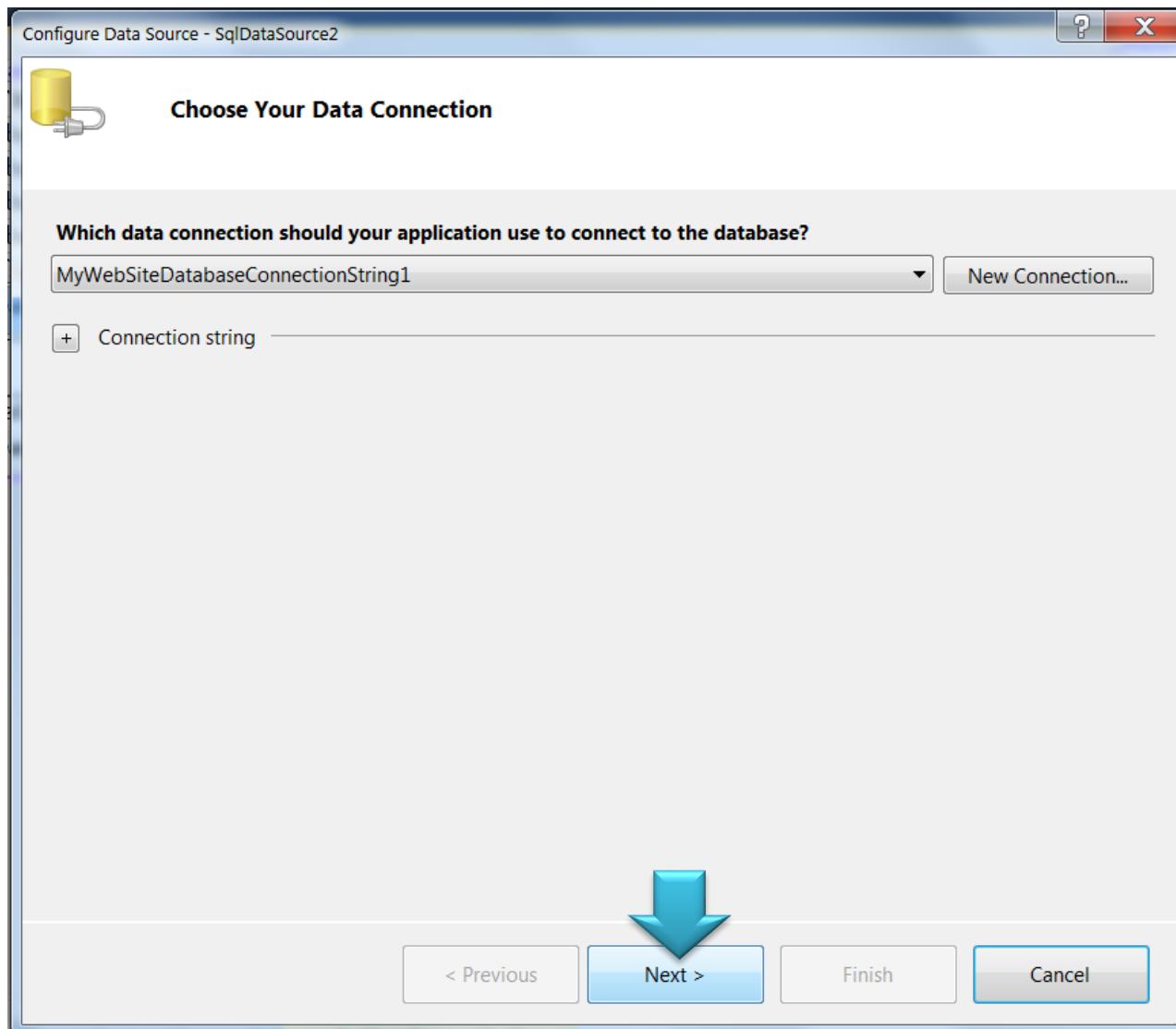
- In Design View of **Reviews . aspx**:

– Click **Configure Data Source** of
SqlDataSource2



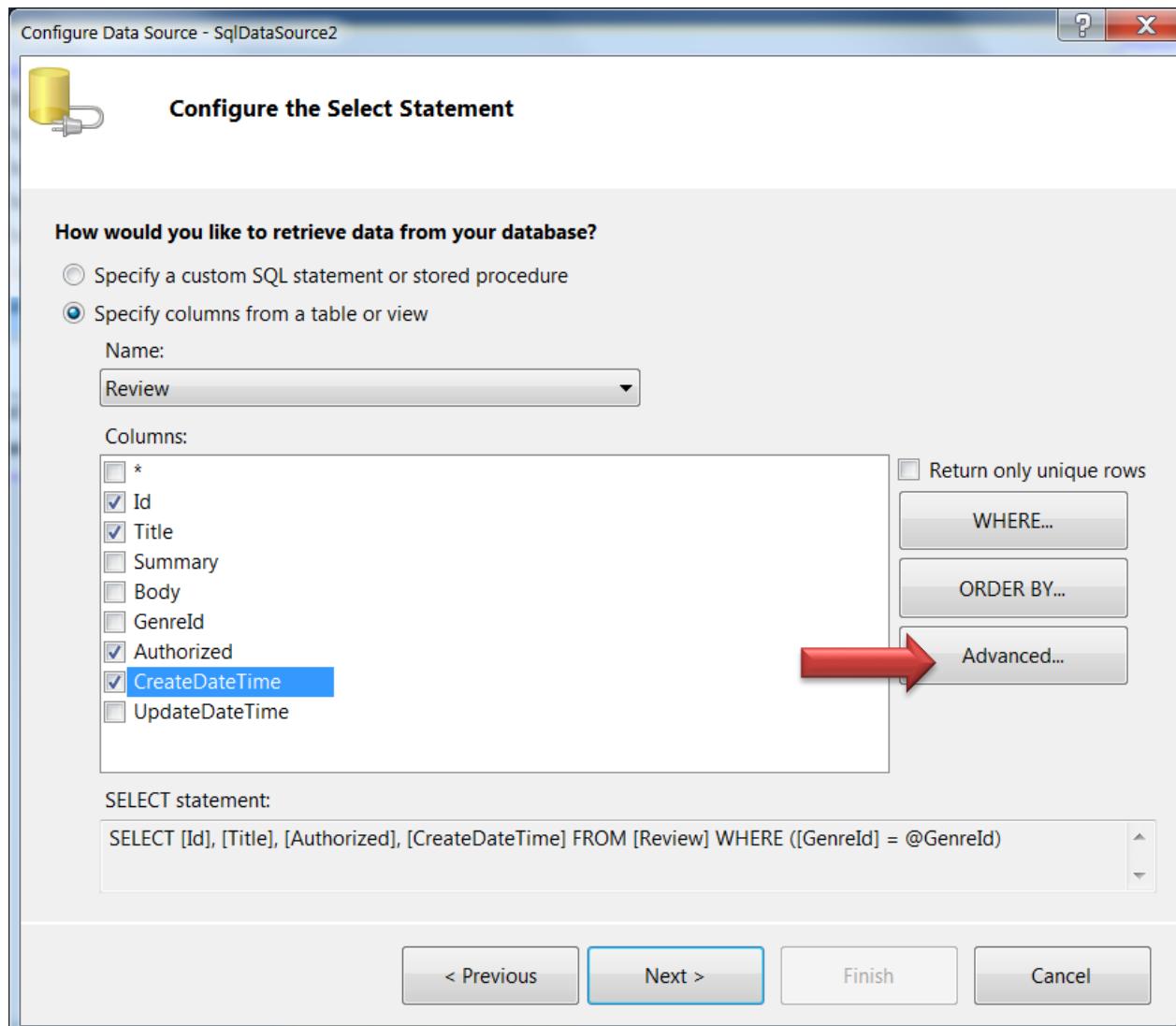
Example

Customizing GridView



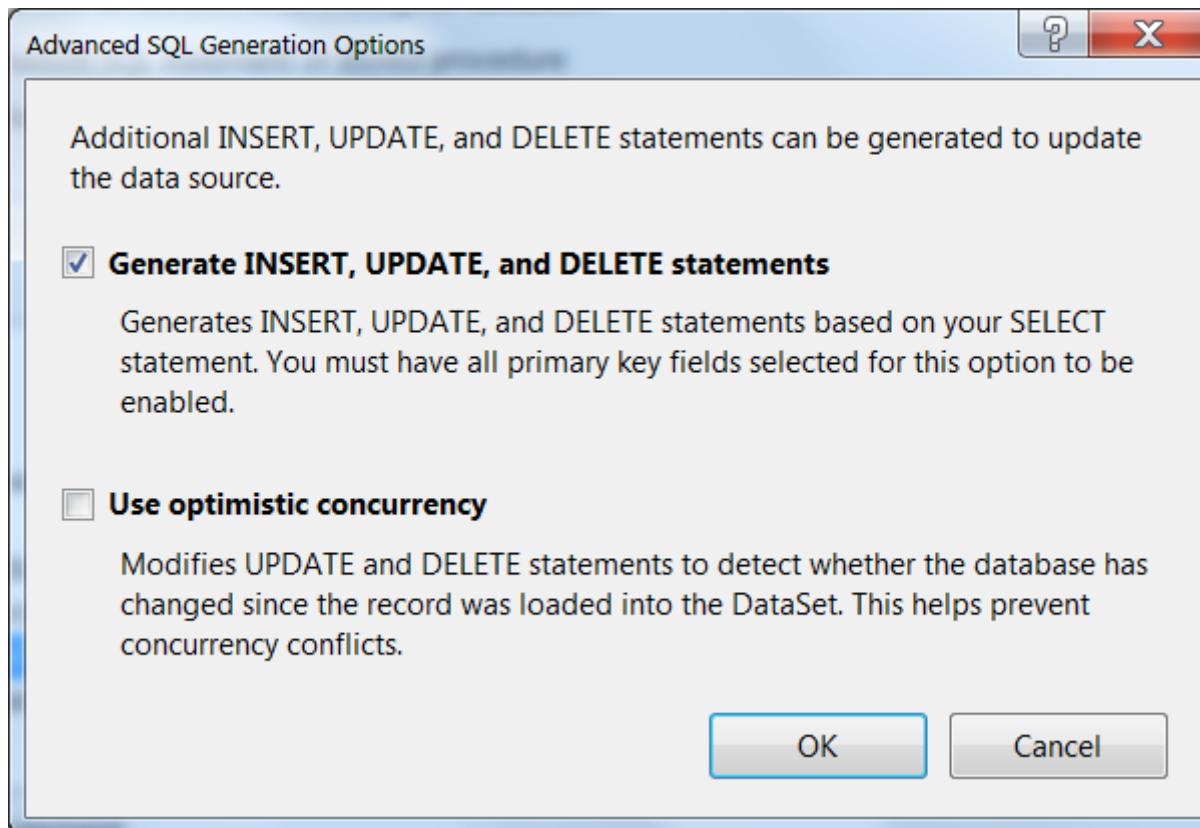
Example

Customizing GridView



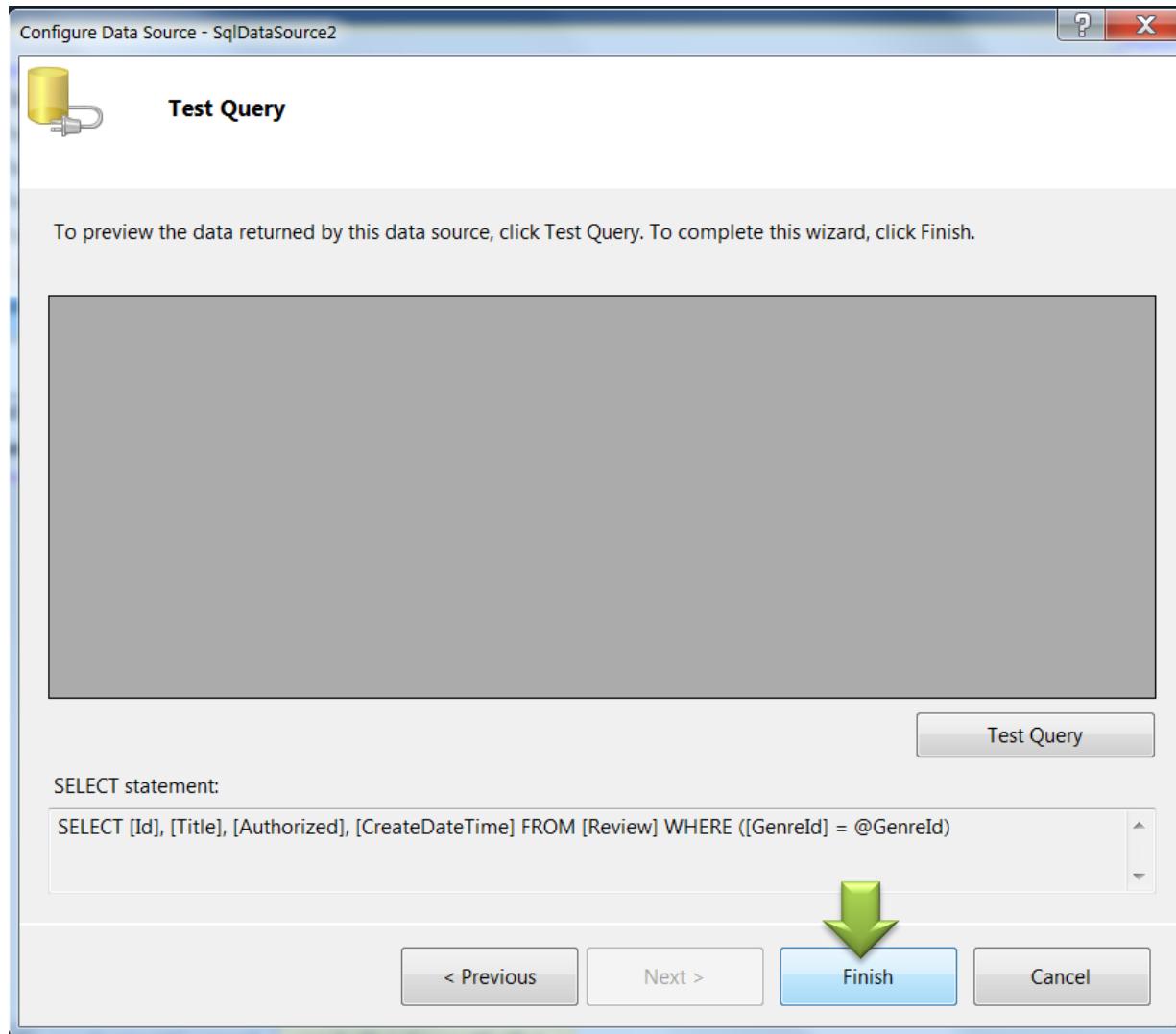
Example

Customizing GridView



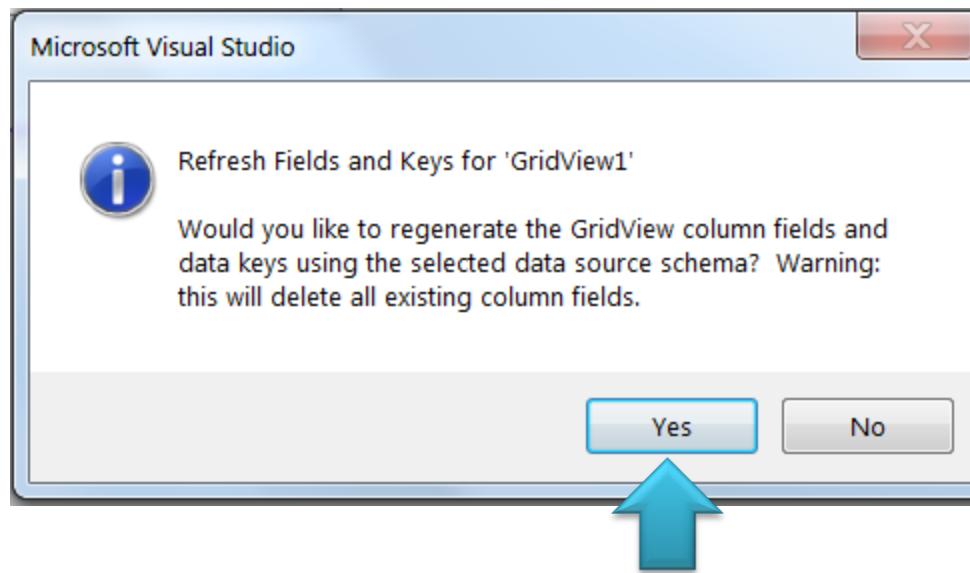
Example

Customizing GridView



Example

Customizing GridView



Example

Customizing GridView

The screenshot shows a Windows application's design view. At the top, a tab labeled "cpMainContent (Custom)" is selected. Below it is a **GridView** control displaying four rows of data:

Id	Title	Authorized	CreateDateTime
0	abc	<input type="checkbox"/>	14.05.2014 00:00:00
1	abc	<input checked="" type="checkbox"/>	14.05.2014 00:00:00
2	abc	<input type="checkbox"/>	14.05.2014 00:00:00
3	abc	<input checked="" type="checkbox"/>	14.05.2014 00:00:00

Below the GridView, two **SqlDataSource** controls are visible: "SqlDataSource - SqlDataSource2" and "SqlDataSource - SqlDataSource1". A context menu is open over the "SqlDataSource - SqlDataSource2" control, titled "SqlDataSource Tasks". The menu contains the following options:

- Configure Data Source...
- Refresh Schema

Example

Customizing GridView

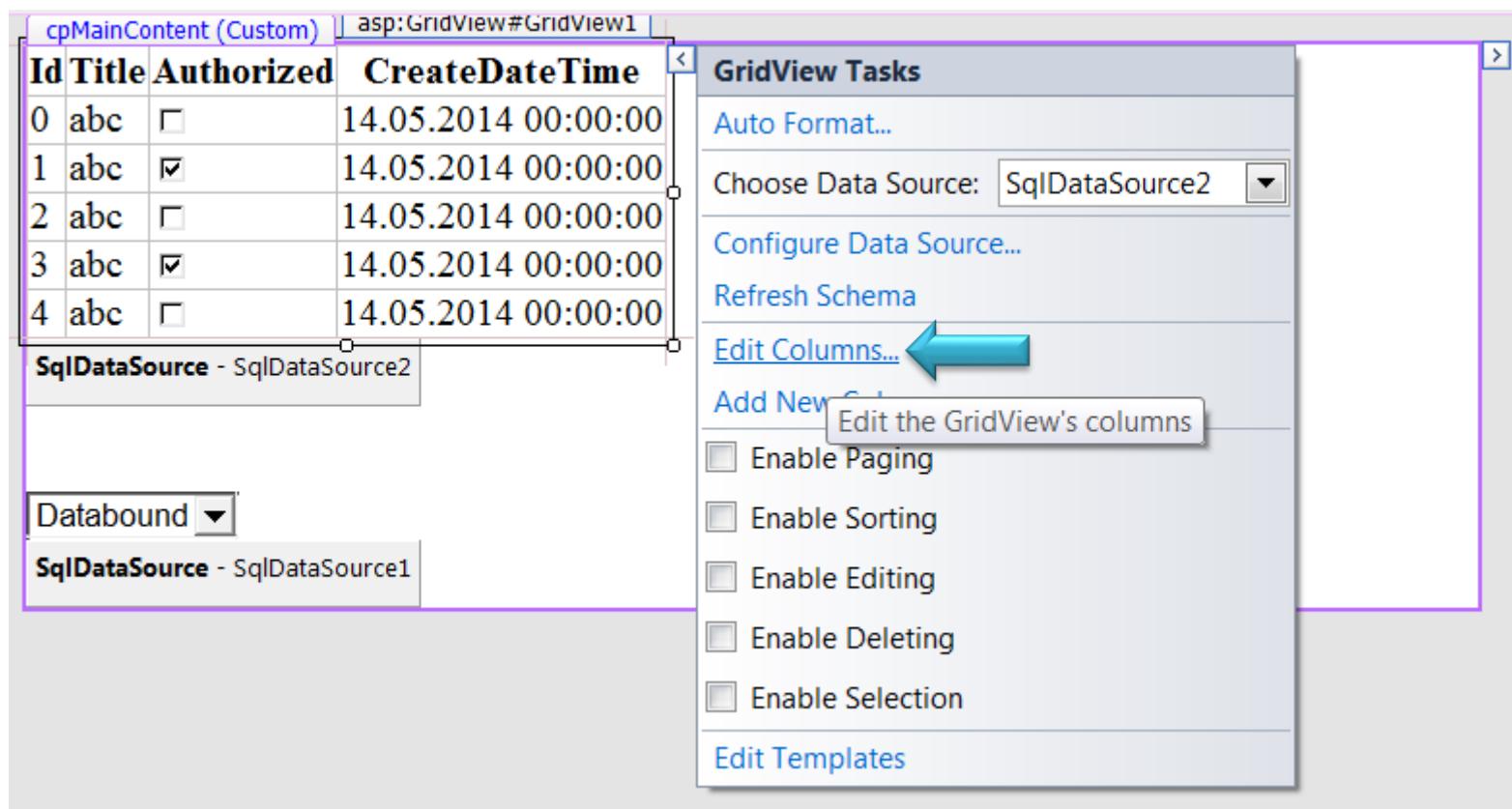
- In Source View:

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="Id" DataSourceID="SqlDataSource2">
    <Columns>
        <asp:BoundField DataField="Id" HeaderText="Id" ReadOnly="True"
            SortExpression="Id" />
        <asp:BoundField DataField="Title" HeaderText="Title" SortExpression="Title" />
        <asp:CheckBoxField DataField="Authorized" HeaderText="Authorized"
            SortExpression="Authorized" />
        <asp:BoundField DataField="CreateDateTime" HeaderText="CreateDateTime"
            SortExpression="CreateDateTime" />
    </Columns>
</asp:GridView>
```

Example

Customizing GridView

- In Design View
 - To remove those items:



Fields

Available fields:

- + CommandField
- + TemplateField
- DynamicField
 - Id
 - Title
 - Authorized
 - CreateDateTime

Add

Selected fields:

- Id
- Title
- Authorized
- CreateDateTime

Auto-generate fields

BoundField properties:

Convert this field into a TemplateField

OK Cancel

Fields

Available fields:

- CommandField
- TemplateField
- DynamicField
 - Id
 - Title
 - Authorized
 - CreateDateTime

Add

Selected fields:

- Id
- Title
- Authorized
- CreateDateTime**

Clear all of them

BoundField properties:

	A Z	
Accessibility		
AccessibleHeaderText		
Appearance		
FooterText		
HeaderImageUrl		
HeaderText	CreateDateTime	
Behavior		
ApplyFormatInEditMode	False	
ConvertEmptyStringToNull	True	
HtmlEncode	True	
HtmlEncodeFormatString	True	
InsertVisible	True	
NullDisplayText		
ReadOnly	False	
ShowHeader	True	

HeaderText

The text within the header of this field.

[Convert this field into a TemplateField](#)

Refresh Schema

OK Cancel

Fields

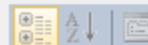


Available fields:

- + CommandField
- + TemplateField
- + DynamicField
 - Id
 - Title
 - Authorized
 - CreateDateTime

Add

Field properties:



Selected fields:

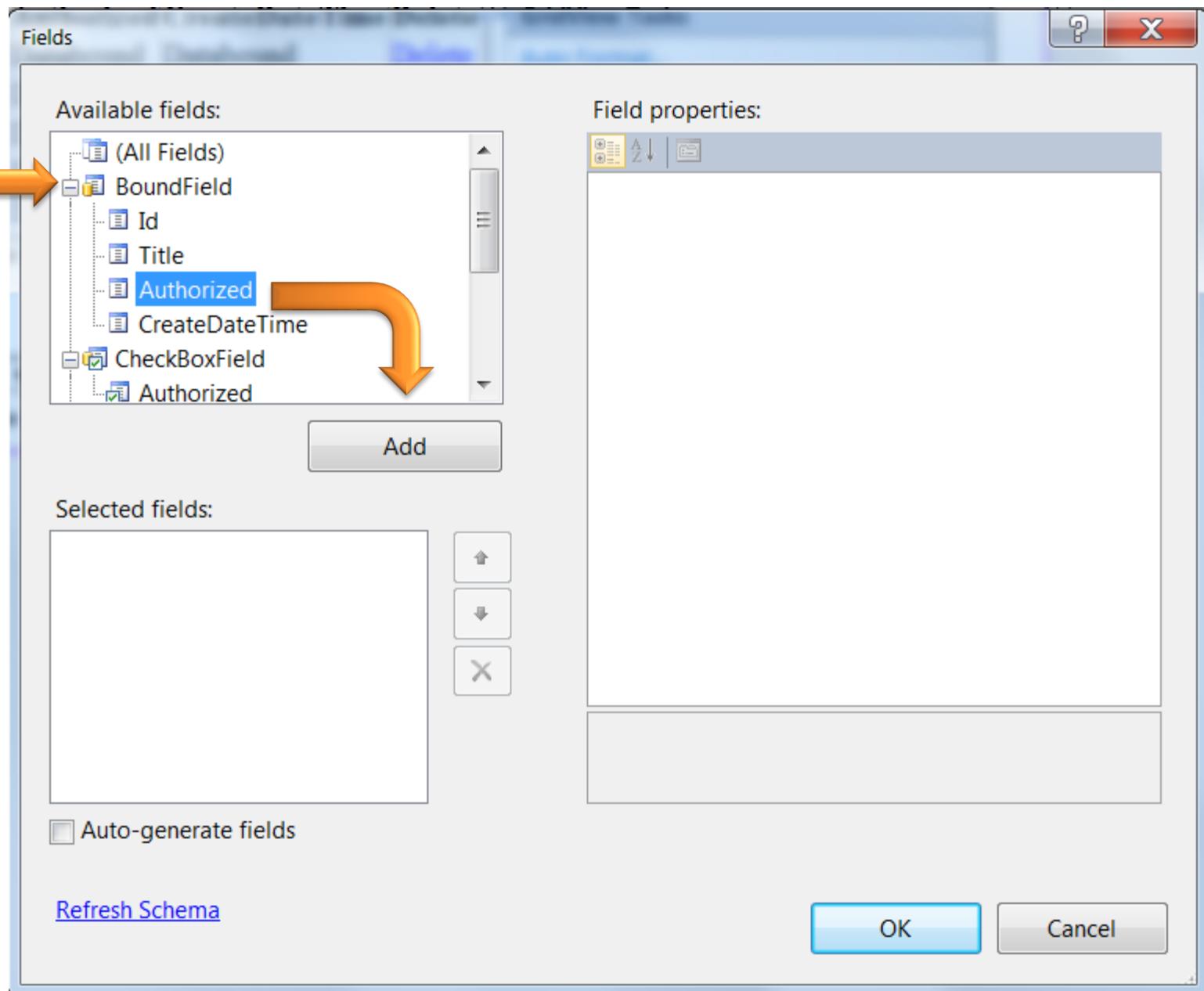


Auto-generate fields

[Refresh Schema](#)

OK

Cancel



Fields

Available fields:

- (All Fields)
- BoundField
 - Id
 - Title
 - Authorized
 - CreateDateTime
- CheckBoxField
 - Authorized

Add

Selected fields:

- Authorized

Auto-generate fields

Refresh Schema

BoundField properties:

Accessibility	
AccessibleHeaderText	
Appearance	
FooterText	
HeaderImageUrl	
HeaderText	Authorized
Behavior	
ApplyFormatInEditMode	False
ConvertEmptyStringToNull	True
HtmlEncode	True
HtmlEncodeFormatString	
InsertVisible	True
NullDisplayText	
ReadOnly	False
ShowHeader	True

HeaderText
The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Fields

Available fields:

- (All Fields)
- BoundField
 - Id
 - Title
 - Authorized
 - CreateDateTime
- CheckBoxField
 - Authorized

Add

Selected fields:

- Authorized
- CreateDateTime

Auto-generate fields

Refresh Schema

BoundField properties:

A **Z** icon is visible above the properties grid.

Accessibility	
AccessibleHeaderText	
Appearance	
FooterText	
HeaderImageUrl	
HeaderText	CreateDateTime
Behavior	
ApplyFormatInEditMode	False
ConvertEmptyStringToNull	True
HtmlEncode	True
HtmlEncodeFormatString	True
InsertVisible	True
NullDisplayText	
ReadOnly	False
ShowHeader	True

HeaderText

The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DvnamicField

Add

Selected fields:

- Authorized
- CreateDateTime
- HyperLinkField

Auto-generate fields

Refresh Schema

HyperLinkField properties:

A  icon points from the Selected fields list to the properties panel.

Accessibility	
AccessibleHeaderText	
Appearance	
FooterText	
HeaderImageUrl	
HeaderText	
Text	
Behavior	
InsertVisible	True
NavigateUrl	
ShowHeader	True
SortExpression	
Target	
Visible	True
Data	
HeaderText	
The text within the header of this field.	

Convert this field into a [TemplateField](#)

OK Cancel

Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DynamicField

Add

Selected fields:

- HyperLinkField
- Authorized
- CreateDateTime

Auto-generate fields

[Refresh Schema](#)

HyperLinkField properties:

A Z	
Accessibility	
AccessibleHeaderText	
Appearance	
FooterText	
HeaderImageUrl	
HeaderText	
Text	
Behavior	
InsertVisible	True
NavigateUrl	
ShowHeader	True
SortExpression	
Target	
Visible	True
Data	
HeaderText	
The text within the header of this field.	

[Convert this field into a TemplateField](#)

OK Cancel

Example

Customizing GridView

Change properties:

- **HeaderText = Title**
- **DataNavigateUrlFields = Id**
- **DataNavigateUrlFormatString = AddEditReview.aspx?Id={0}**
- **DataTextField = Title**

Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DvnyamicField

Add

Selected fields:

- Title
- Authorized
- CreateDateTime

Auto-generate fields

Refresh Schema

HyperLinkField properties:

A Z	
HeaderImageUrl	
HeaderText	Title
Text	
Behavior	
InsertVisible	True
NavigateUrl	
ShowHeader	True
SortExpression	
Target	
Visible	True
Data	
DataNavigateUrlField	Id
DataNavigateUrlForm	AddEditReview.aspx?Id=
DataTextField	Title
DataTextFormatString	

DataTextField
The field bound to the text property of the hyperlink.

[Convert this field into a TemplateField](#)

OK Cancel

Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DvnamicField

Add

Selected fields:

- Title
- Authorized
- CreateDateTime
- CommandField

Auto-generate fields

Refresh Schema

CommandField properties:

Accessibility	
Appearance	
AccessibleHeaderText	
ButtonType	Link
CancelImageUrl	
CancelText	Cancel
DeleteImageUrl	
DeleteText	Delete
EditImageUrl	
EditText	Edit
FooterText	
HeaderImageUrl	
HeaderText	
InsertImageUrl	
InsertText	Insert

HeaderText
The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Q X

Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DynamicField

Add

Selected fields:

- Title
- Authorized
- CreateDateTime
- ~~Delete~~

Auto-generate fields

Refresh Schema

CommandField properties:

HeaderText	Delete
InsertImageUrl	
InsertText	Insert
NewImageUrl	
NewText	New
SelectImageUrl	
SelectText	Select
UpdateImageUrl	
UpdateText	Update
Behavior	
CausesValidation	True
InsertVisible	True
ShowCancelButton	True
ShowDeleteButton	True
ShowEditButton	False
ShowDeleteButton	
Whether the field should display a delete button to ...	

[Convert this field into a TemplateField](#)

OK Cancel

Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DynamicField

Add

Selected fields:

- Title
- Authorized
- CreateDateTime
- Delete

Auto-generate fields

Refresh Schema

Field properties:

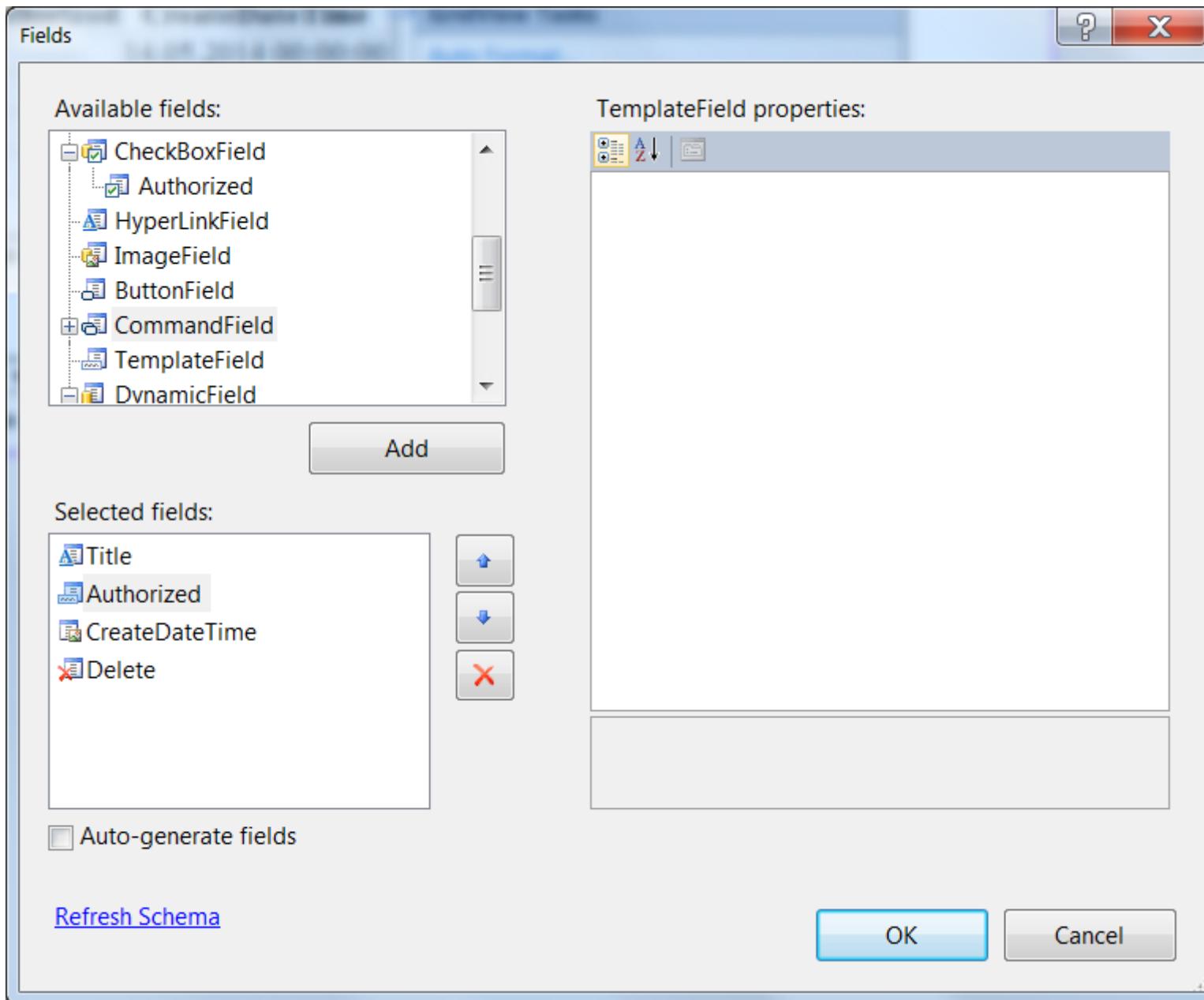
A **Z** icon is present above the properties grid.

Accessibility	AccessibleHeaderText
Appearance	FooterText
	HeaderImageUrl
	HeaderText Authorized
Behavior	ApplyFormatInEditMode False
	ConvertEmptyStringToNull False
	HtmlEncode True
	InsertVisible True
	NullDisplayText
	ReadOnly False
	ShowHeader True
	SortExpression

HeaderText
The text within the header of this field.

Convert this field into a TemplateField

OK Cancel



Fields

Available fields:

- CheckBoxField
 - Authorized
- HyperLinkField
- ImageField
- ButtonField
- CommandField
- TemplateField
- DynamicField

Add

Selected fields:

- Title
- Authorized
- CreateDateTime
- Delete

Auto-generate fields

Refresh Schema

Field properties:

	A Z ↴	↶
ConvertEmptyStringT	False	
HtmlEncode	True	
InsertVisible	True	
NullDisplayText		
ReadOnly	False	
ShowHeader	True	
SortExpression		
UIHint		
ValidationGroup		
Visible	True	

Data

DataField	CreateDateTime
DataFormatString	{0:g}

Styles

ControlStyle	
--------------	--

DataFormatString

The formatting that is applied to the bound value. F...

Convert this field into a TemplateField

OK Cancel

Convert this field into a TemplateField

OK Cancel

Example

Customizing GridView

The screenshot shows a Microsoft Visual Studio interface with a sidebar on the left containing navigation links:

- Management Home
- Manage Genres
- Manage Reviews

The main area displays a `cpMainContent (Custom)` page. On the page, there is a `asp:GridView#GridView1` control. The grid view has the following data:

Title	Authorized	CreateDate	Time	Delete
abc	False	14.05.2014	00:00	Delete
abc	True	14.05.2014	00:00	Delete
abc	False	14.05.2014	00:00	Delete
abc	True	14.05.2014	00:00	Delete
abc	False	14.05.2014	00:00	Delete

Below the grid view, there are two `SqlDataSource` controls: `SqlDataSource2` and `SqlDataSource1`. A context menu, titled "GridView Tasks", is open over the grid view. The menu includes the following options:

- Auto Format...
- Choose Data Source: `SqlDataSource2` (dropdown menu)
- Configure Data Source...
- Refresh Schema
- Edit Columns...
- Add New Column...
- Enable Paging
- Enable Sorting
- Enable Editing
- Enable Deleting
- Enable Selection
- Edit Templates

Example

Customizing GridView

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="Id" DataSourceID="SqlDataSource2">
    <Columns>
        <asp:HyperLinkField DataNavigateUrlFields="Id"
            DataNavigateUrlFormatString="AddEditReview.aspx?Id={0}" DataTextField="Title"
            HeaderText="Title" />
        <asp:TemplateField HeaderText="Authorized"
            SortExpression="Authorized">
            <EditItemTemplate>
                <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Authorized") %>'></asp:TextBox>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="Label1" runat="server" Text='<%# Bind("Authorized") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="CreateDateTime" DataFormatString="{0:g}"
            HeaderText="CreateDateTime" SortExpression="CreateDateTime" />
        <asp:CommandField HeaderText="Delete" ShowDeleteButton="True" />
    </Columns>
</asp:GridView>
```

Remove

Example

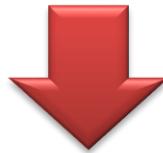
Customizing GridView

```
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="Id" DataSourceID="SqlDataSource2">
    <Columns>
        <asp:HyperLinkField DataNavigateUrlFields="Id"
            DataNavigateUrlFormatString="AddEditReview.aspx?Id={0}" DataTextField="Title"
            HeaderText="Title" />
        <asp:TemplateField HeaderText="Authorized"
            SortExpression="Authorized">
            <ItemTemplate>
                <asp:Label ID="Label1" runat="server" Text='<%# Bind("Authorized") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="CreateDateTime" DataFormatString="{0:g}"
            HeaderText="CreateDateTime" SortExpression="CreateDateTime" />
        <asp:CommandField HeaderText="Delete" ShowDeleteButton="True" />
    </Columns>
</asp:GridView>
```

Example

Customizing GridView

```
<ItemTemplate>
    <asp:Label ID="Label1" runat="server" Text='<%# Bind("Authorized") %>'></asp:Label>
</ItemTemplate>
```



```
<ItemTemplate>
    <asp:Label ID="AuthorizedLabel" runat="server" Text='<%# GetBooleanText(Eval("Authorized")) %>'></asp:Label>
</ItemTemplate>
```

Example

Customizing GridView

- Open → **Reviews.aspx.cs**

```
public partial class Management_Reviews : System.Web.UI.Page
{
    protected string GetBooleanText(object booleanValue)
    {
        bool authorized = (bool)booleanValue;
        if (authorized)
        {
            return "Yes";
        }
        else
        {
            return "No";
        }
    }

    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

Returns Yes or No depending on the Boolean value that you pass

Example

Customizing GridView

- View **Reviews.aspx** in browser.

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Title	Authorized	CreateDateTime	Delete
21-Adele	Yes	29.04.2013 17:00	Delete
Doo-Wops & Hooligans-Bruno Mars	Yes	29.04.2013 17:02	Delete
This is the title	Yes	01.05.2013 19:17	Delete
21 by Adele	Yes	01.05.2013 21:19	Delete

Insert and Update Data

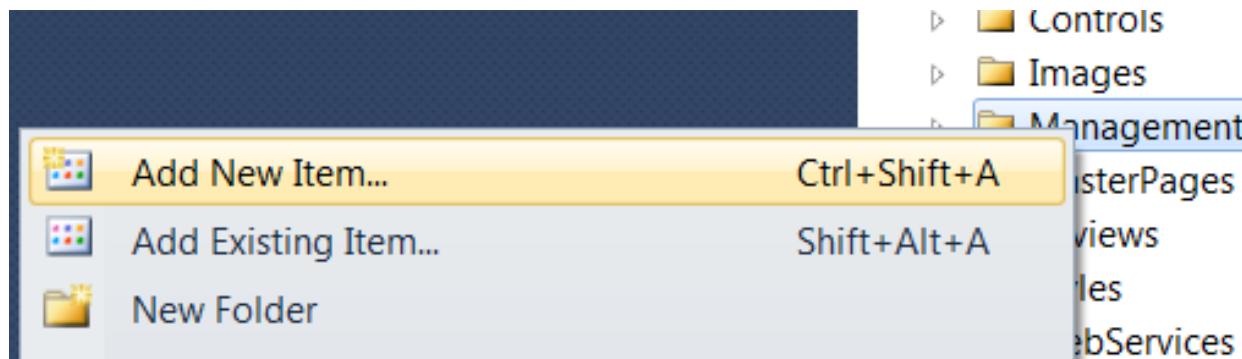
- Some of the events that the **DetailsView**, the **FormView**, and the **ListView** expose and raise during their lifetime:

EVENT	DESCRIPTION
ItemInserting	Fires right before the <code>Insert</code> command is executed against the data source. This is an ideal location to change the data that is about to be sent to the database.
ItemInserted	Fires right after the <code>Insert</code> command has been executed against the data source.
ItemUpdating	Fires right before the <code>Update</code> command is executed against the data source. This is an ideal location to change the data that is about to be sent to the database.
ItemUpdated	Fires right after the <code>Update</code> command has been executed against the data source.
ItemDeleting	Fires right before the <code>Delete</code> command is executed against the data source.
ItemDeleted	Fires right after the <code>Delete</code> command has been executed against the data source.

Example

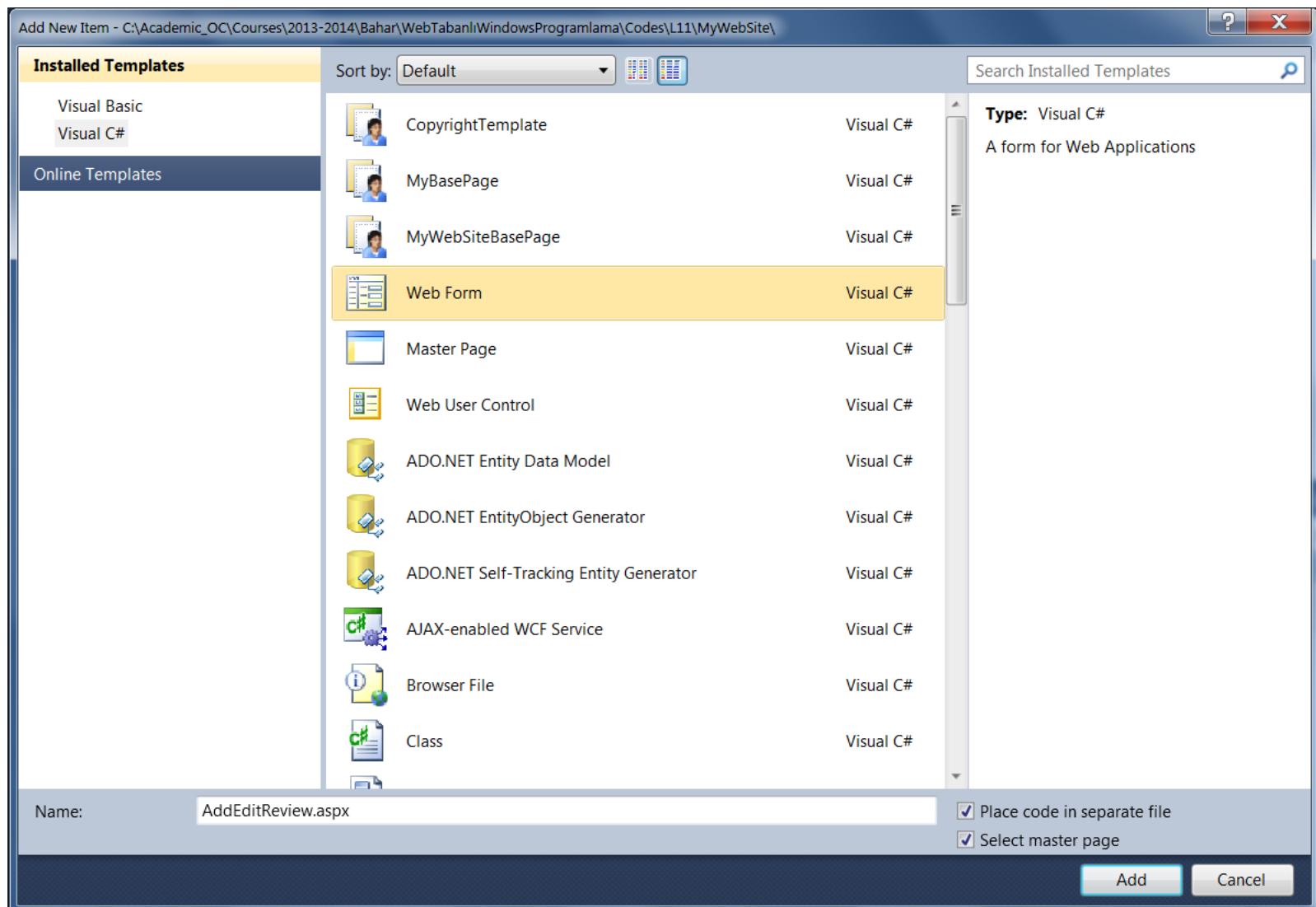
DetailsView

- Add New Item → Management folder



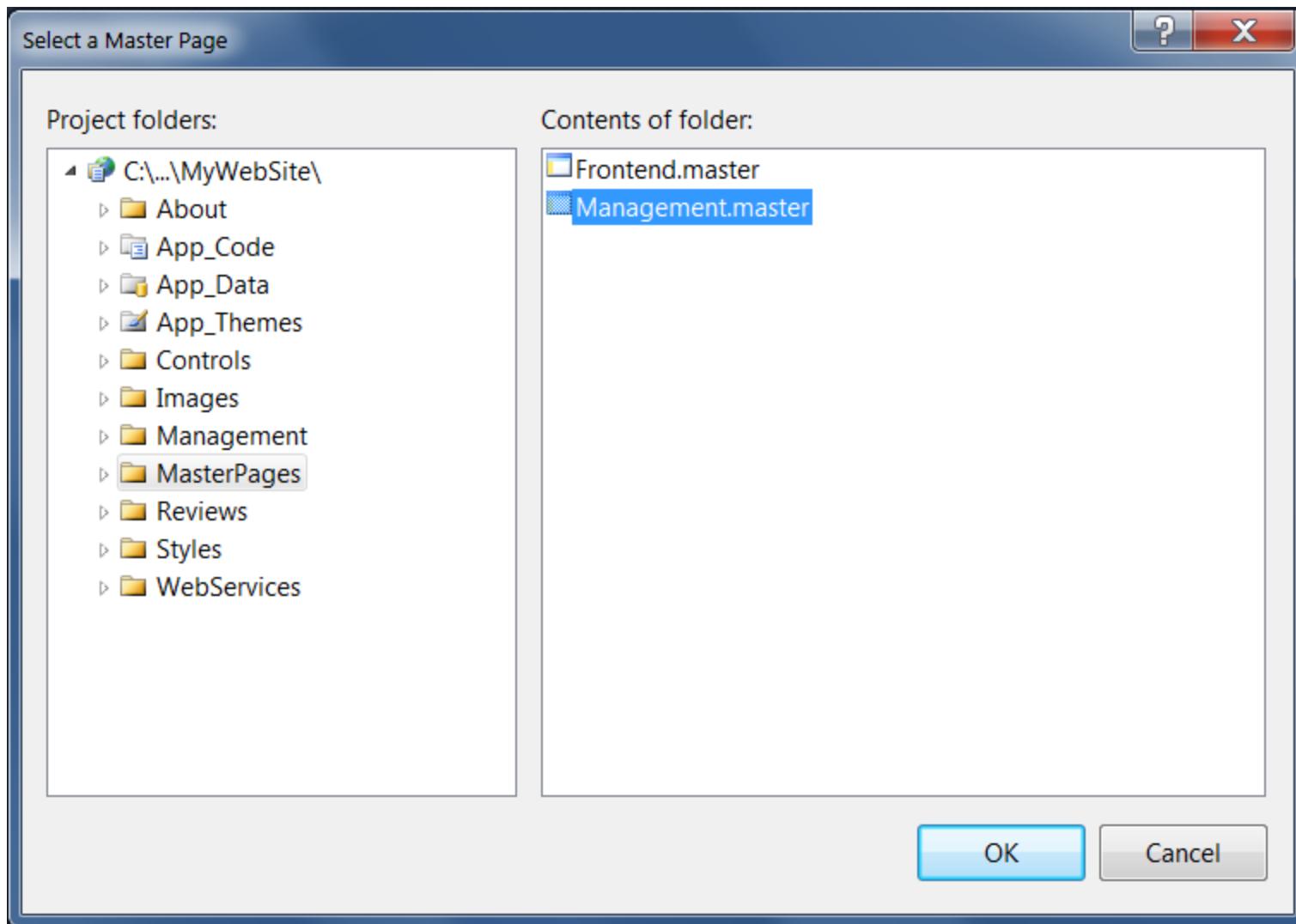
Example

DetailsView



Example

DetailsView



Example

DetailsView

- Change **Title = Management - Insert and Update Reviews**

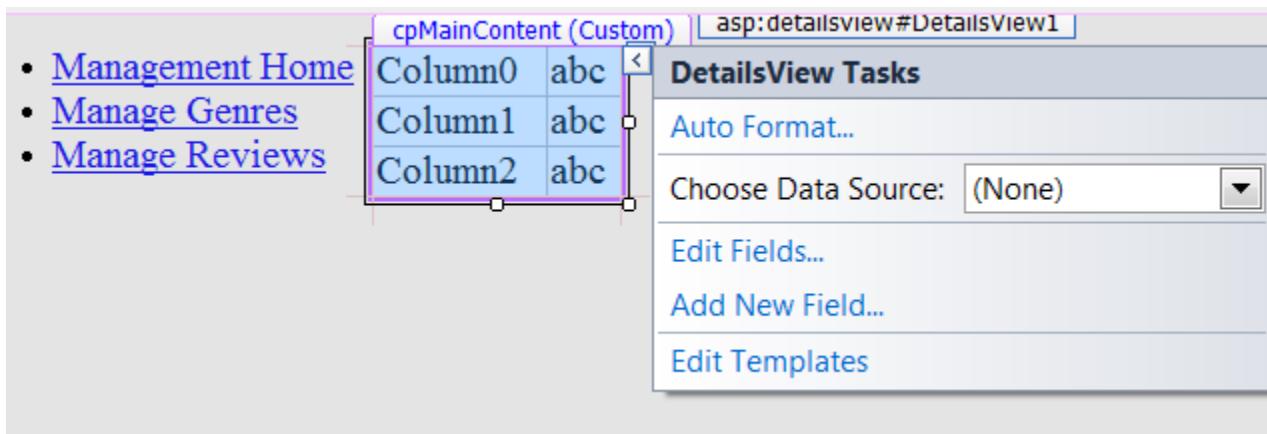
```
<%@ Page Title="Management - Insert and Update Reviews" Language="C#" MasterPageFile="~/MasterPages/Management.master"

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="cpMainContent" Runat="Server">
</asp:Content>
```

Example

DetailsView

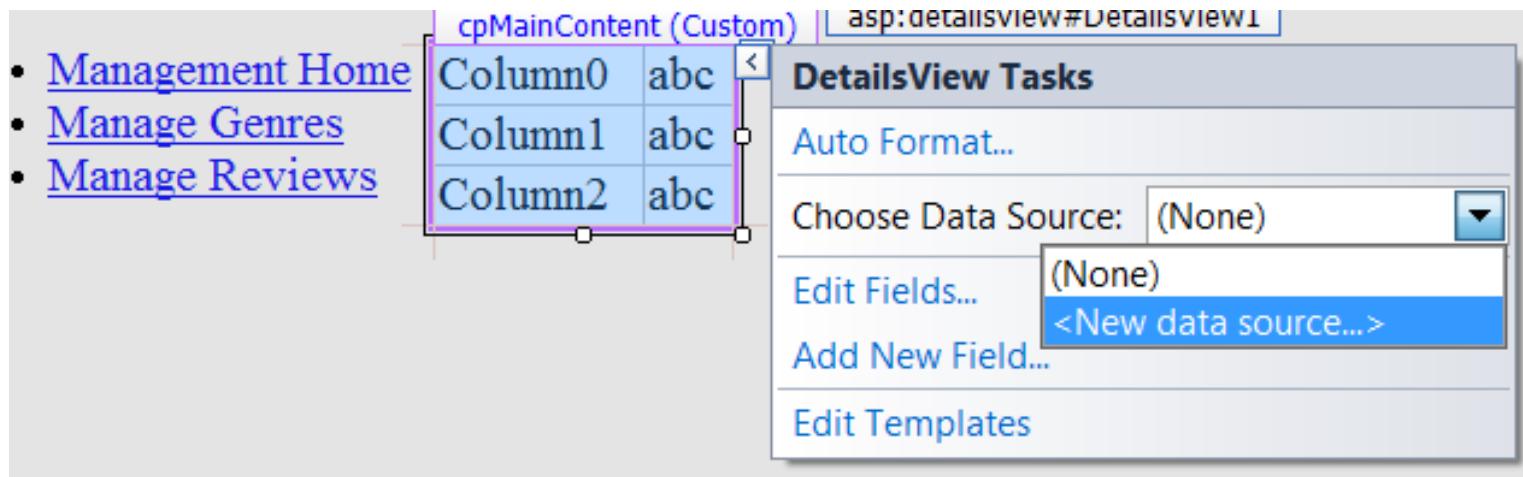
- In Design View:
 - Drop a **DetailsView** control



Example

DetailsView

Choose Data Source → <New data source>





Choose a Data Source Type

Where will the application get data from?



Access
Database



Database



Entity



LINQ



Object



Site Map



XML File

Connect to any SQL database supported by ADO.NET, such as Microsoft SQL Server, Oracle, or OLEDB.

Specify an ID for the data source:

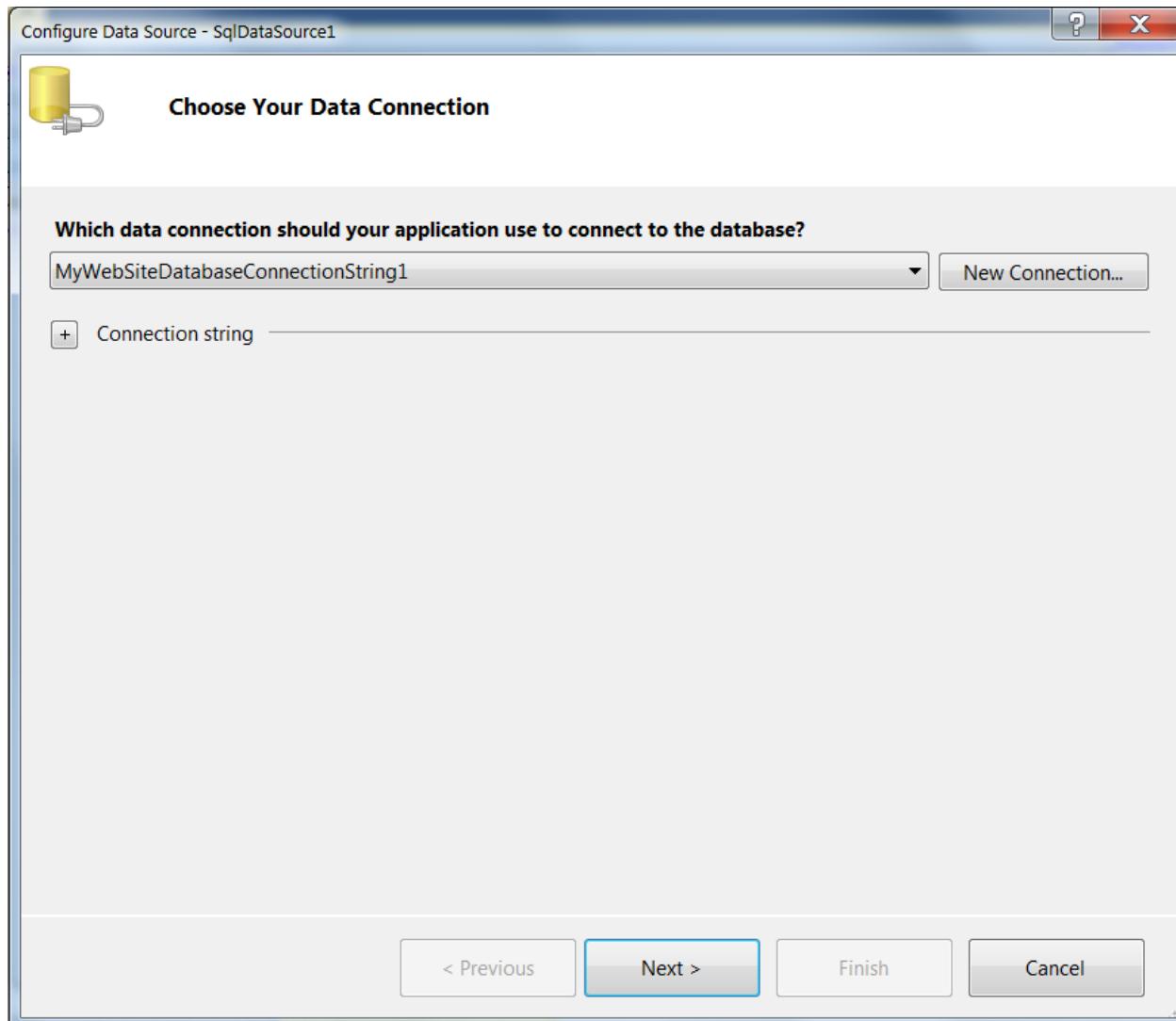
SqlDataSource1

OK

Cancel

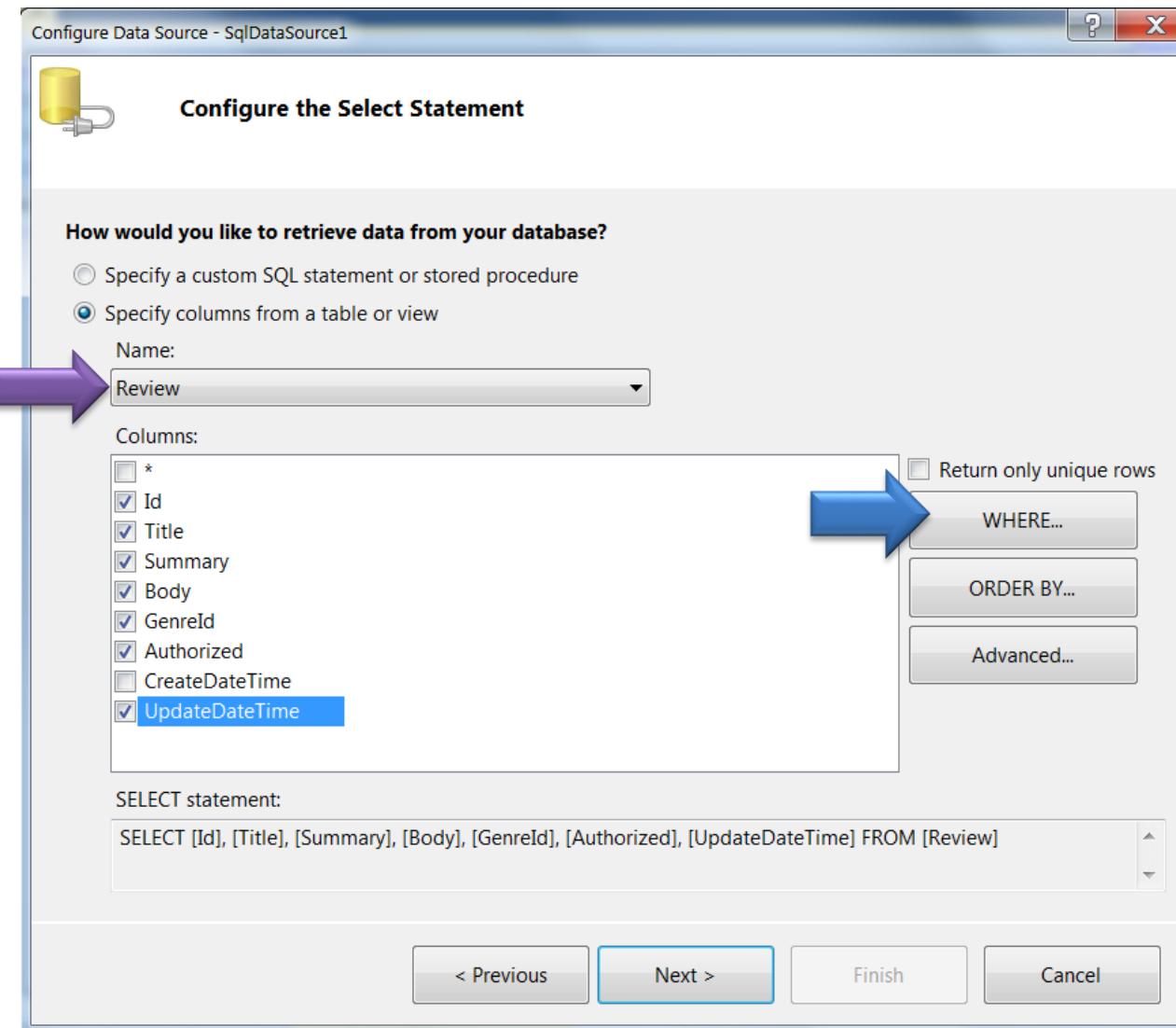
Example

DetailsView



Example

DetailsView



Add WHERE Clause

Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column: 

Operator: 

Source: 

SQL Expression: 

WHERE clause:

SQL Expression	Value

Parameter properties

QueryString field: 

Default value:

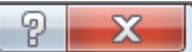
Value: 

Add 

Remove

OK Cancel

Add WHERE Clause



Add one or more conditions to the WHERE clause for the statement. For each condition you can specify either a literal value or a parameterized value. Parameterized values get their values at runtime based on their properties.

Column:

Parameter properties

Operator:

Source:

SQL Expression:

Value:

Add

WHERE clause:

SQL Expression	Value	Remove
[Id] = @Id	Request.QueryString(...)	

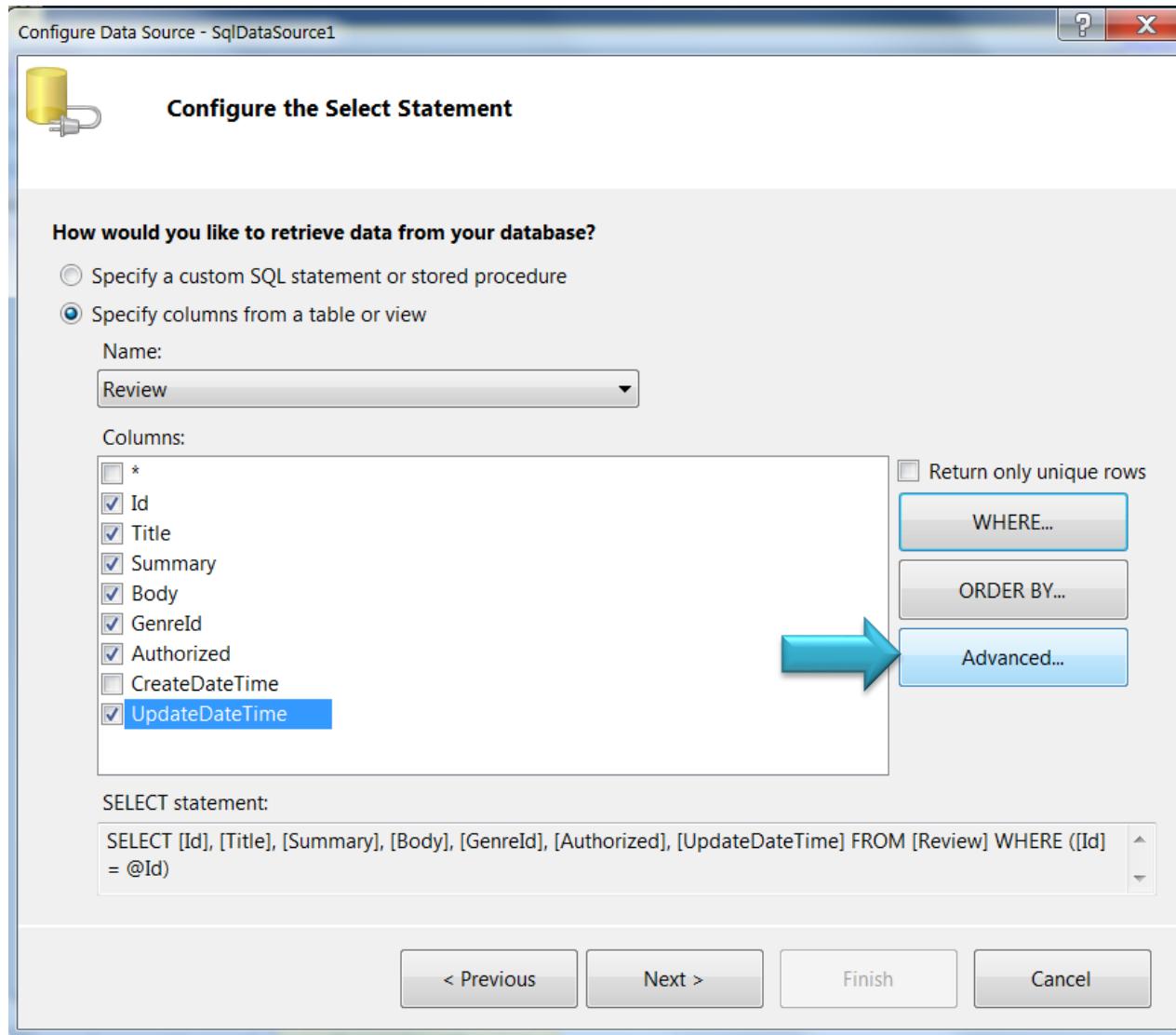


OK

Cancel

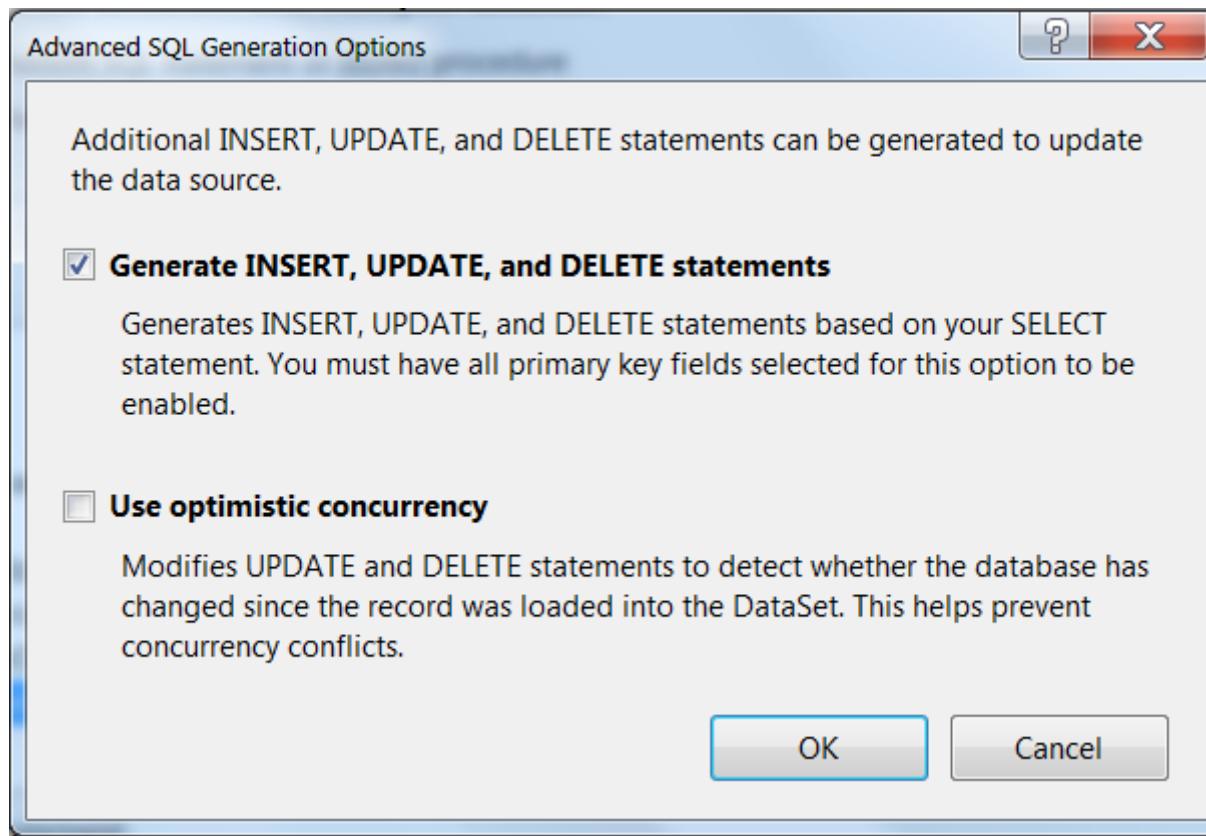
Example

DetailsView



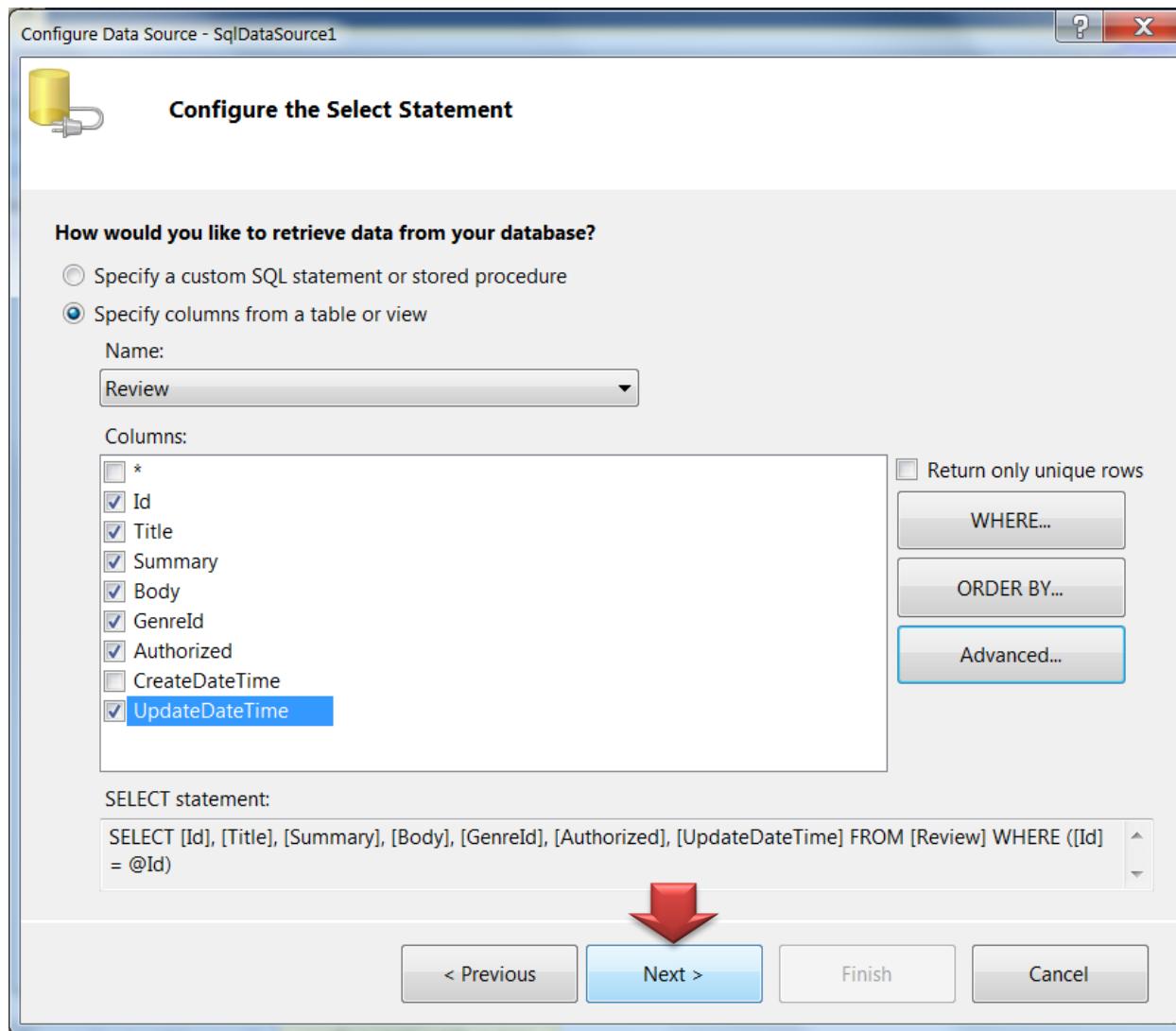
Example

DetailsView



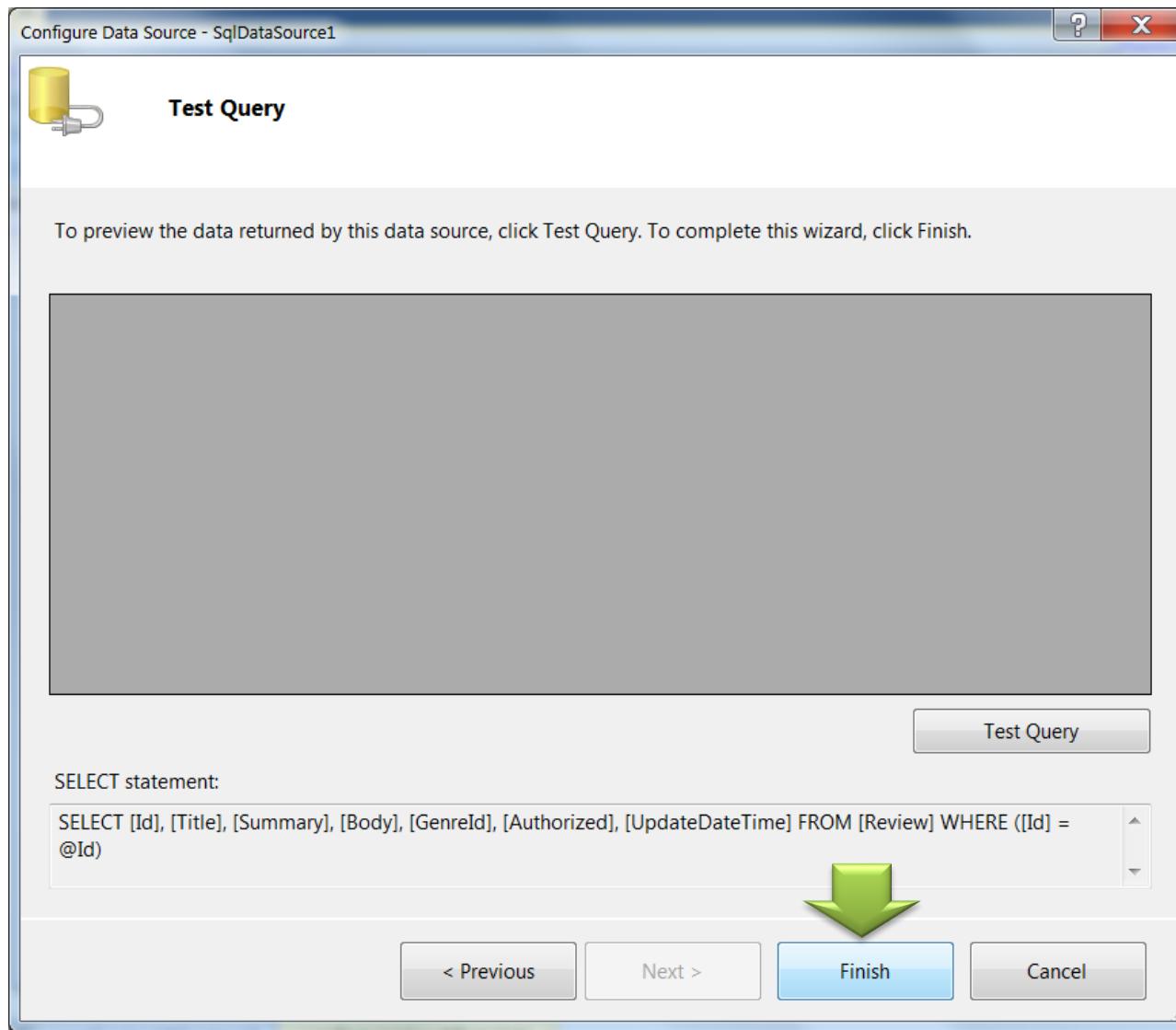
Example

DetailsView



Example

DetailsView



Example

DetailsView

The screenshot shows a Windows Forms application window titled "cpMainContent (Custom)". Inside the window, there is a DetailsView control with the header "asp:detailsview#DetailsView1". The DetailsView displays a single record with the following data:

Id	0
Title	abc
Summary	abc
Body	abc
GenreId	0
Authorized	<input type="checkbox"/>
UpdateDateTime	14.05.2014 00:00:00

Below the DetailsView, a SqlDataSource control is visible with the label "SqlDataSource - SqlDataSource1". To the left of the main content area, there is a sidebar with the following navigation links:

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Example

DetailsView

The screenshot shows the Visual Studio IDE in design mode. A navigation menu on the left lists:

- Management Home
- Manage Genres
- Manage Reviews

The main area contains a **cpMainContent (Custom)** container holding a **asp:detailsview#DetailsView1**. The **DetailsView** displays the following data:

Id	0
Title	abc
Summary	abc
Body	abc
GenreId	0
Authorized	<input type="checkbox"/>
UpdateDateTime	14.05.2014 00:00:00

[Edit New](#)

A **SqlDataSource - SqlDataSource1** is connected to the DetailsView.

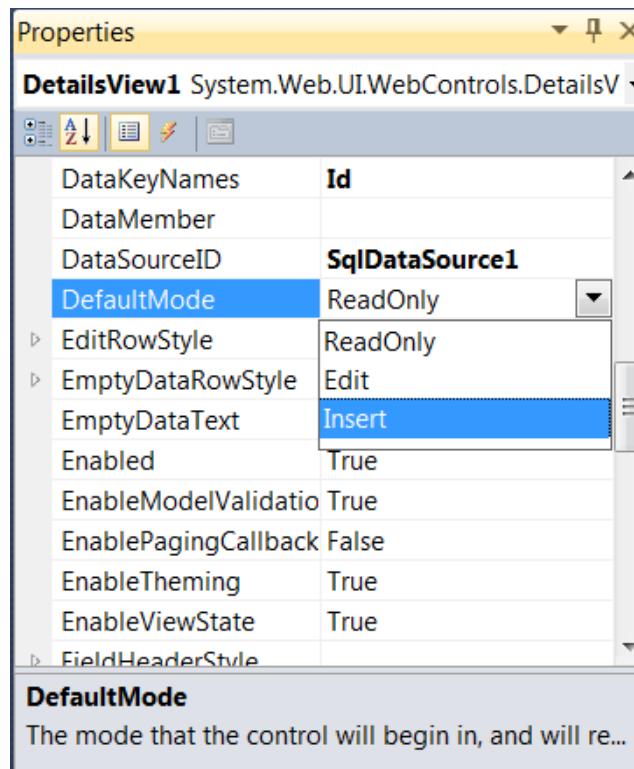
The **DetailsView Tasks** pane is open on the right, showing the following options:

- Auto Format...
- Choose Data Source: **SqlDataSource1**
- Configure Data Source...
- Refresh Schema
- Edit Fields...
- Add New Field...
- Enable Paging
- Enable Inserting ←
- Enable Editing ←
- Enable Deleting
- Edit Templates

Example

DetailsView

- **DetailsView** properties:
 - **DefaultMode = Insert**



Example

DetailsView

- Open → AddEditReview.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.QueryString.Get("Id") != null)
    {
        DetailsView1.DefaultMode = DetailsViewMode.Edit;
    }
}
```

Example

DetailsView

- Open **Reviews.aspx**
 - Drag the page **AddEditReview.aspx** from the Solution Explorer onto the page below the **GridView**.

The screenshot shows the Visual Studio IDE interface with the following components:

- Navigation Menu:** On the left, there is a sidebar with links: "Management Home", "Manage Genres", and "Manage Reviews".
- Main Content Area:** The main content area is titled "cpMainContent (Custom)". It contains a **GridView** with the following structure:

Title	Authorized	CreateDateTime	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete
Databound	Databound	Databound	Delete

- SqlDataSource:** Below the GridView, there is a **SqlDataSource** control with the ID "SqlDataSource2".
- Link:** At the bottom of the page, there is a blue link labeled "AddEditReview.aspx".

Example

DetailsView

- In Source View:

```
<a href="AddEditReview.aspx">AddEditReview.aspx</a>
```



```
<a href="AddEditReview.aspx">Insert new review</a>
```

Example

DetailsView

- View **AddEditReview.aspx** in browser.

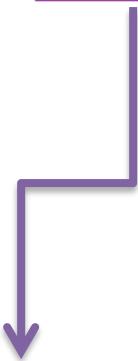
- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Id	
Title	
Summary	
Body	
GenreId	
Authorized	<input type="checkbox"/>
UpdateDateTime	
Insert	Cancel

Example

DetailsView

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)



Id	17
Title	Xscape
Summary	Album of Michael Jackson
Body	New album
GenreId	4
Authorized	<input checked="" type="checkbox"/>
UpdateDateTime	2014/05/14
Insert Cancel	

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Title	Authorized	CreateDateTime	Delete
21-Adele	Yes	29.04.2013 17:00	Delete
Doo-Wops & Hooligans-Bruno Mars	Yes	29.04.2013 17:02	Delete
This is the title	Yes	01.05.2013 19:17	Delete
21 by Adele	Yes	01.05.2013 21:19	Delete
Xscape	Yes	14.05.2014 18:05	Delete

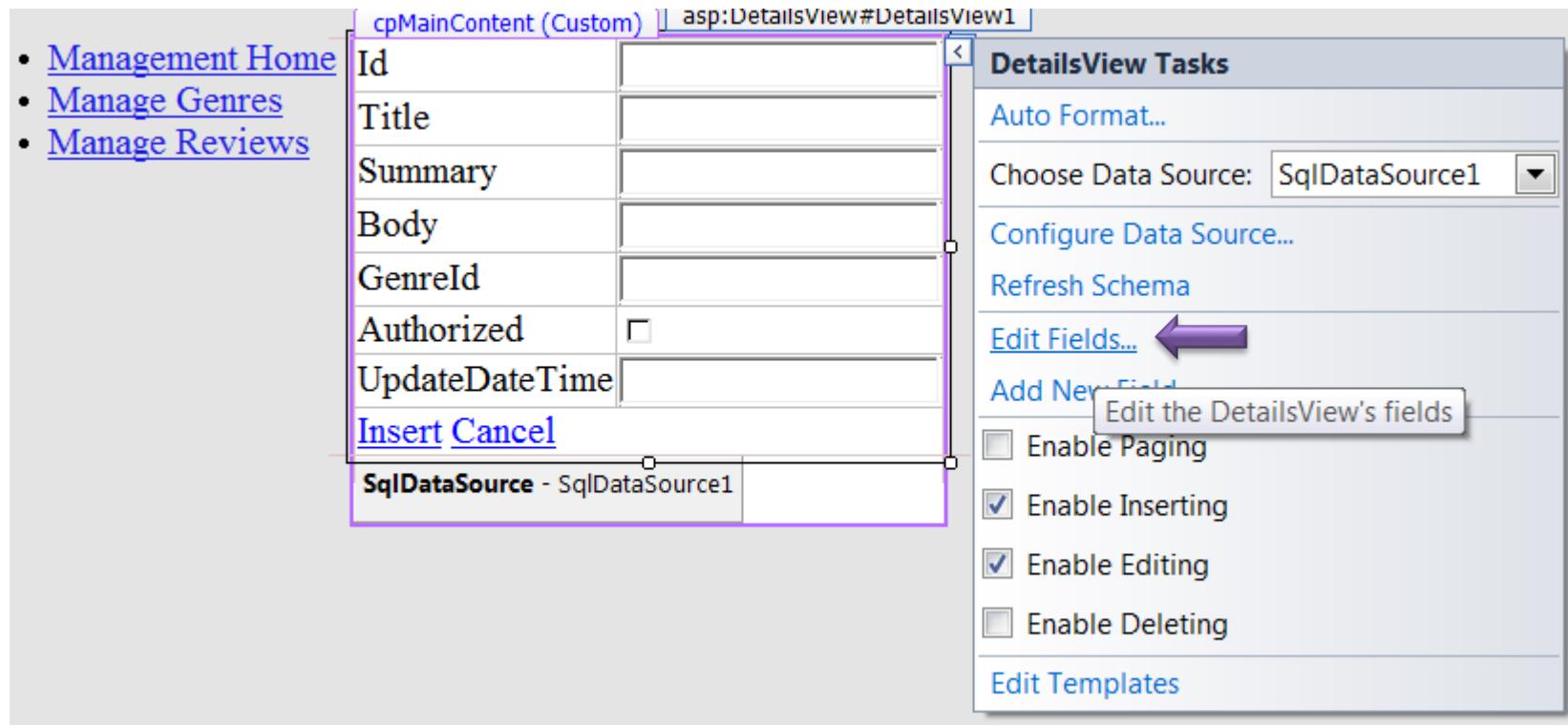
Pop ▼

[Insert new review](#)

Example

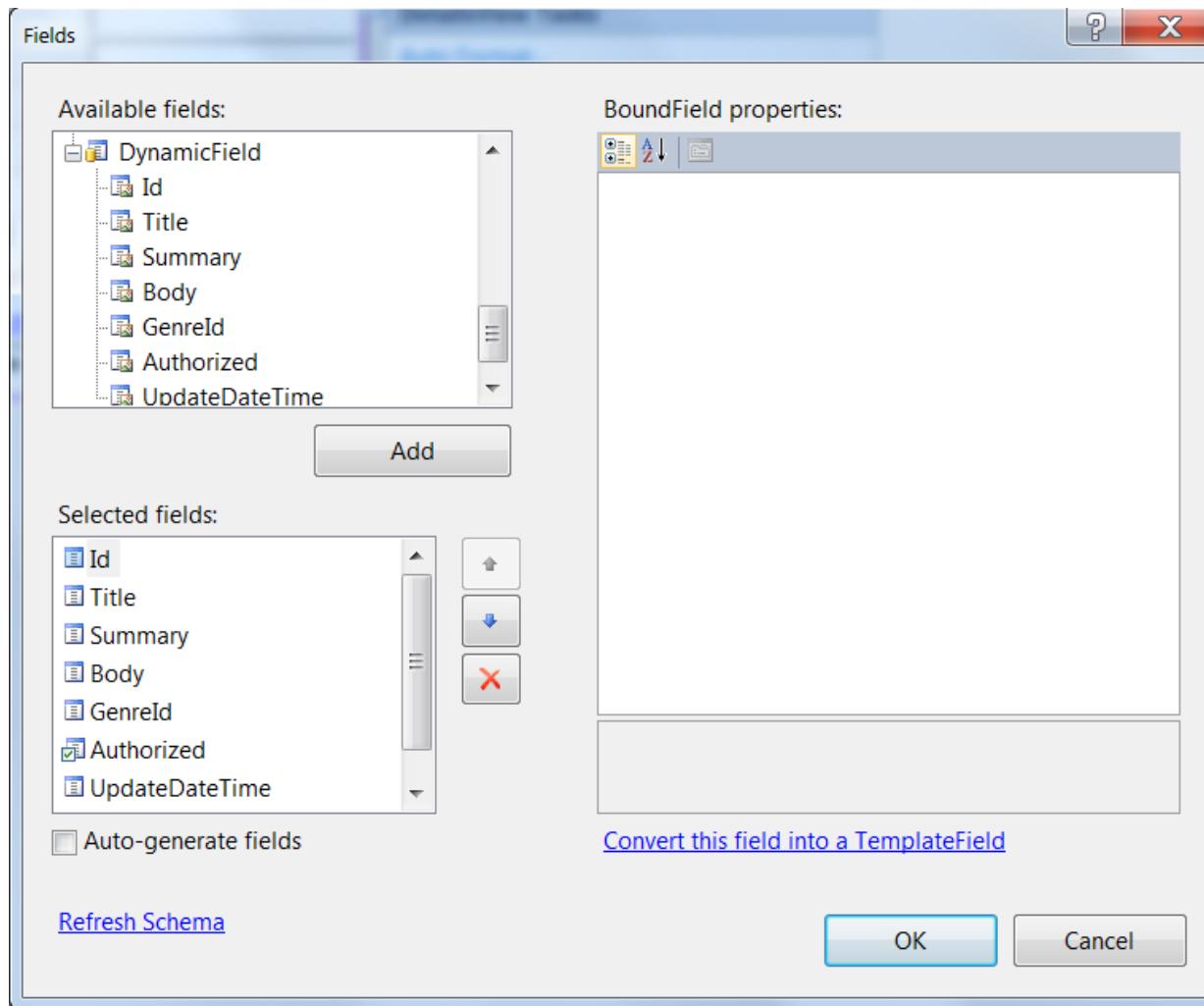
Customizing DetailsView

- Open → AddEditReview.aspx in Desing View



Example

Customizing DetailsView



Selected Fields → UpdateDateTime

Visible = False

Fields

Available fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Add

Selected fields:

- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime
- CommandField

Auto-generate fields

BoundField properties:

ApplyFormatInEditMode	False
ConvertEmptyStringToNull	True
HtmlEncode	True
HtmlEncodeFormatString	True
InsertVisible	True
NullDisplayText	
ReadOnly	False
ShowHeader	True
SortExpression	UpdateDateTime
Visible	False

Data

DataField	UpdateDateTime
DataFormatString	

Styles

ControlStyle	
--------------	--

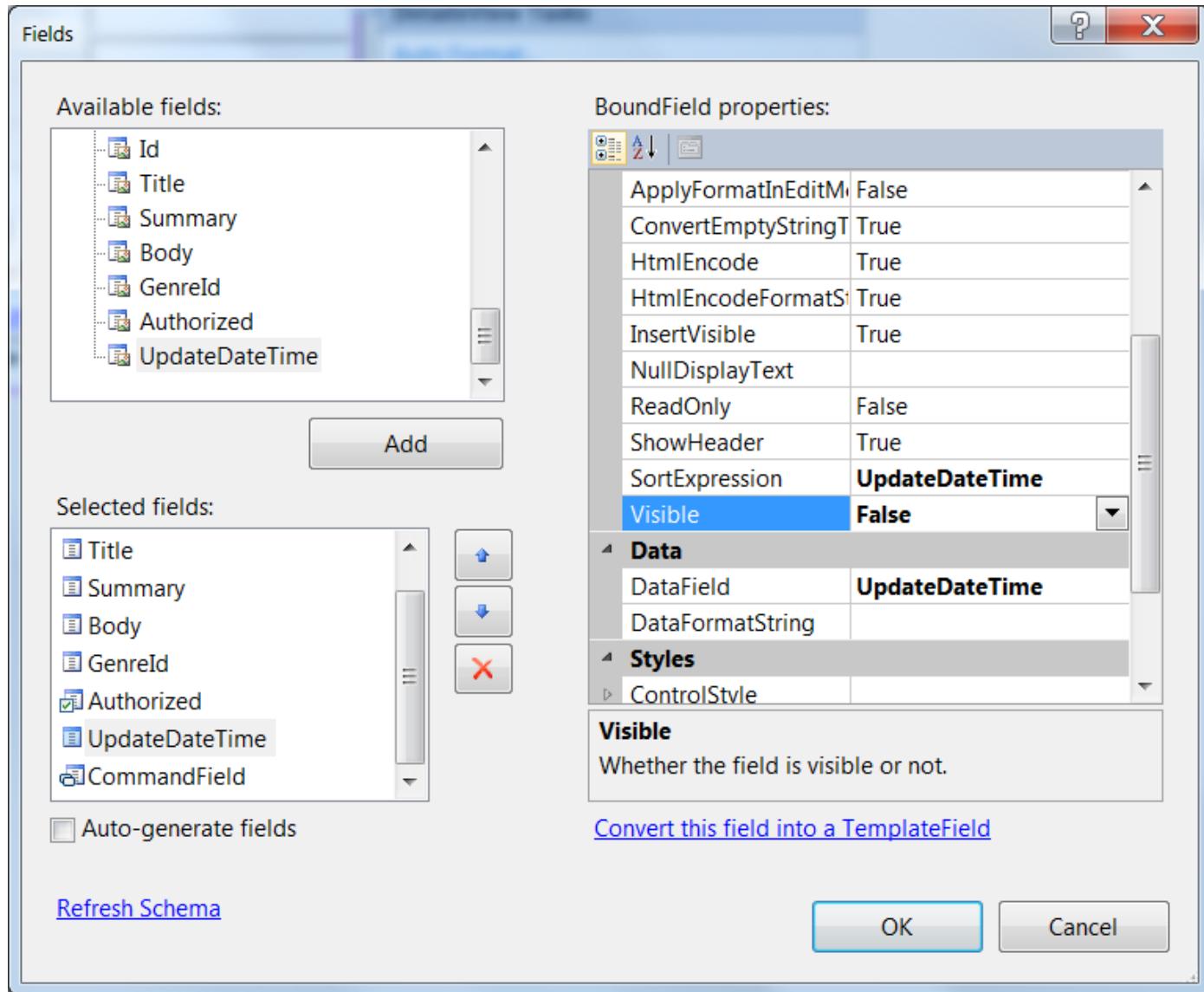
Visible

Whether the field is visible or not.

[Convert this field into a TemplateField](#)

Refresh Schema

OK Cancel



Fields

Available fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Add

Selected fields:

- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime
- CommandField

Auto-generate fields

Refresh Schema

BoundField properties:

Accessibility
AccessibleHeaderText

Appearance
FooterText
HeaderImageUrl
HeaderText

Behavior	
ApplyFormatInEditMode	False
ConvertEmptyStringToNull	True
HtmlEncode	True
HtmlEncodeFormatString	True
InsertVisible	True
NullDisplayText	
ReadOnly	False
ShowHeader	True

HeaderText
The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Fields

Available fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Add

Selected fields:

- Id
- Title
- Summary**
- Body
- GenreId
- Authorized
- UpdateDateTime

Auto-generate fields

Refresh Schema

BoundField properties:

Accessibility

AccessibleHeaderText

Appearance

FooterText	
HeaderImageUrl	
HeaderText	Summary

Behavior

ApplyFormatInEditMode	<input type="checkbox"/> False
ConvertEmptyStringToNull	<input checked="" type="checkbox"/> True
HtmlEncode	<input checked="" type="checkbox"/> True
HtmlEncodeFormatString	<input checked="" type="checkbox"/> True
InsertVisible	<input checked="" type="checkbox"/> True
NullDisplayText	
ReadOnly	<input type="checkbox"/> False
ShowHeader	<input checked="" type="checkbox"/> True

HeaderText

The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Available fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Add

Selected fields:

- Id
- Title
- Summary**
- Body
- GenreId
- Authorized
- UpdateDateTime

Auto-generate fields

Refresh Schema

BoundField properties:

Accessibility

AccessibleHeaderText

Appearance

FooterText	
HeaderImageUrl	
HeaderText	Summary

Behavior

ApplyFormatInEditMode	<input type="checkbox"/> False
ConvertEmptyStringToNull	<input checked="" type="checkbox"/> True
HtmlEncode	<input checked="" type="checkbox"/> True
HtmlEncodeFormatString	<input checked="" type="checkbox"/> True
InsertVisible	<input checked="" type="checkbox"/> True
NullDisplayText	
ReadOnly	<input type="checkbox"/> False
ShowHeader	<input checked="" type="checkbox"/> True

HeaderText

The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Fields

Available fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Add

Selected fields:

- Id
- Title
- Summary
- Body**
- GenreId
- Authorized
- UpdateDateTime

Auto-generate fields

Refresh Schema

BoundField properties:

Accessibility

AccessibleHeaderText

Appearance

FooterText

HeaderImageUrl

HeaderText **Body**

Behavior

ApplyFormatInEditMode False

ConvertEmptyStringToNull True

HtmlEncode True

HtmlEncodeFormatString

InsertVisible True

NullDisplayText

ReadOnly False

ShowHeader True

HeaderText

The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

This screenshot shows the 'Fields' dialog box, likely from a Microsoft Visual Studio interface. It displays a list of available fields and a list of selected fields, along with their properties. A blue arrow points from the 'Selected fields' list to the 'HeaderText' property in the properties grid, highlighting the conversion of the 'Body' field into a template field. Another blue arrow points to a link labeled 'Convert this field into a TemplateField'.

Fields

Available fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Add

Selected fields:

- Id
- Title
- Summary
- Body
- GenreId
- Authorized
- UpdateDateTime

Auto-generate fields

Refresh Schema

BoundField properties:

Accessibility

AccessibleHeaderText

Appearance

FooterText

HeaderImageUrl

HeaderText **GenreId**

Behavior

ApplyFormatInEditMode False

ConvertEmptyStringToNull True

HtmlEncode True

HtmlEncodeFormatString

InsertVisible True

NullDisplayText

ReadOnly False

ShowHeader True

HeaderText

The text within the header of this field.

[Convert this field into a TemplateField](#)

OK Cancel

Example

Customizing DetailsView

The screenshot shows the Visual Studio IDE in design mode. On the left, there is a navigation menu with three items: "Management Home", "Manage Genres", and "Manage Reviews". The main area contains a "cpMainContent (Custom)" placeholder containing an "asp:DetailsView#DetailsView1" control. This control displays a table with seven columns corresponding to the fields: Id, Title, Summary, Body, GenreId, Authorized, Insert, and Cancel. Below the table is a "SqlDataSource - SqlDataSource1" component. To the right of the control is a "DetailsView Tasks" context menu. The "Choose Data Source:" dropdown is set to "SqlDataSource1". Other options in the menu include "Auto Format...", "Configure Data Source...", "Refresh Schema", "Edit Fields...", "Add New Field...", and checkboxes for "Enable Paging" (unchecked), "Enable Inserting" (checked), "Enable Editing" (checked), and "Enable Deleting" (unchecked). At the bottom of the menu is an "Edit Templates" link.

Id	Databound
Title	Databound
Summary	Databound
Body	Databound
GenreId	Databound
Authorized	<input type="checkbox"/>
Insert Cancel	

SqlDataSource - SqlDataSource1

DetailsView Tasks

Choose Data Source: SqlDataSource1

Auto Format...
Configure Data Source...
Refresh Schema
Edit Fields...
Add New Field...
Enable Paging
Enable Inserting
Enable Editing
Enable Deleting
Edit Templates

In Source View:

```
<asp:TemplateField HeaderText="Summary" SortExpression="Summary">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox2" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
    </EditItemTemplate>
    <InsertItemTemplate>
        <asp:TextBox ID="TextBox2" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label2" runat="server" Text='<%# Bind("Summary") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```



```
<asp:TemplateField HeaderText="Summary" SortExpression="Summary">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox2" TextMode="MultiLine" Width="500" Height="100" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
    </EditItemTemplate>
    <InsertItemTemplate>
        <asp:TextBox ID="TextBox2" TextMode="MultiLine" Width="500" Height="100" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label2" runat="server" Text='<%# Bind("Summary") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```

In Source View:



```
<asp:TemplateField HeaderText="Summary" SortExpression="Summary">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox2" TextMode="MultiLine" Width="500" Height="100" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
    </EditItemTemplate>
    <InsertItemTemplate>
        <asp:TextBox ID="TextBox2" TextMode="MultiLine" Width="500" Height="100" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label2" runat="server" Text='<%# Bind("Summary") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```



```
<asp:TemplateField HeaderText="Summary" SortExpression="Summary">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox2" TextMode="MultiLine" Width="500" Height="100" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1" ControlToValidate="TextBox2" runat="server" ErrorMessage="Enter a summary"></asp:Req
    </EditItemTemplate>
    <InsertItemTemplate>
        <asp:TextBox ID="TextBox2" TextMode="MultiLine" Width="500" Height="100" runat="server" Text='<%# Bind("Summary") %>'></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2" ControlToValidate="TextBox2" runat="server" ErrorMessage="Enter a summary"></asp:Req
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label2" runat="server" Text='<%# Bind("Summary") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```

In Source View:

```
<asp:TemplateField HeaderText="Title" SortExpression="Title">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Title") %>'></asp:TextBox>
    </EditItemTemplate>
    <InsertItemTemplate>
        <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Title") %>'></asp:TextBox>
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label1" runat="server" Text='<%# Bind("Title") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```



```
<asp:TemplateField HeaderText="Title" SortExpression="Title">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Title") %>'></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator3" ControlToValidate="TextBox1" runat="server" ErrorMessage="Enter a title"></asp:Req
    </EditItemTemplate>
    <InsertItemTemplate>
        <asp:TextBox ID="TextBox1" runat="server" Text='<%# Bind("Title") %>'></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator4" ControlToValidate="TextBox1" runat="server" ErrorMessage="Enter a title"></asp:Req
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label1" runat="server" Text='<%# Bind("Title") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```

Example

Customizing DetailsView

The screenshot shows a Microsoft Visual Studio interface with a web form. On the left, there is a navigation menu with three items:

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

The main area contains a `asp:DetailsView#DetailsView1` control. This control displays a table with the following columns and rows:

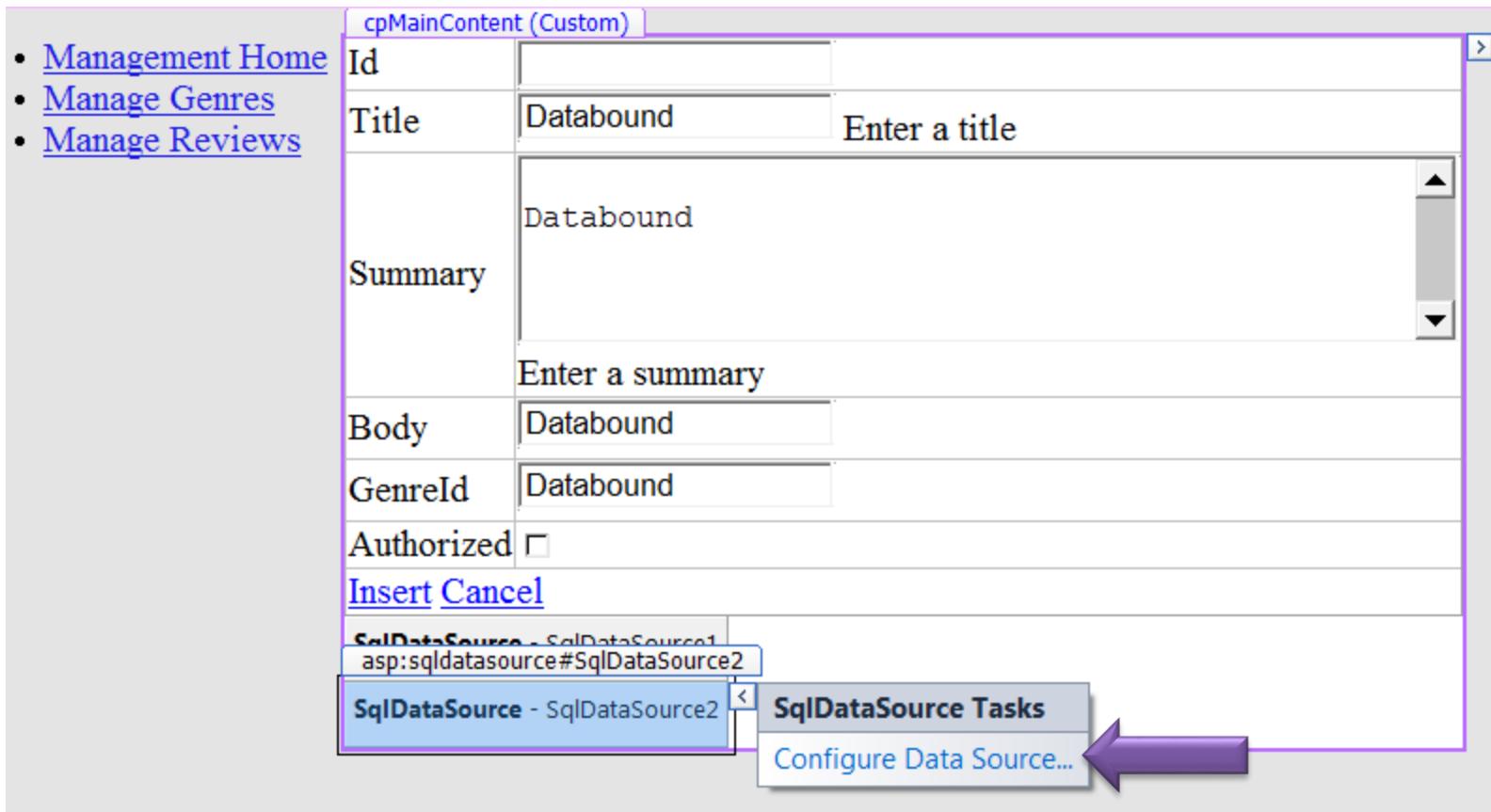
Id	Databound	
Title	Databound	Enter a title
Summary	Databound	Enter a summary
Body	Databound	
GenreId	Databound	
Authorized	<input type="checkbox"/>	
Insert Cancel		
SqlDataSource - SqlDataSource1		

The `SqlDataSource` is named `- SqlDataSource1`. The `DetailsView` is identified by the ID `DetailsView1`.

Example

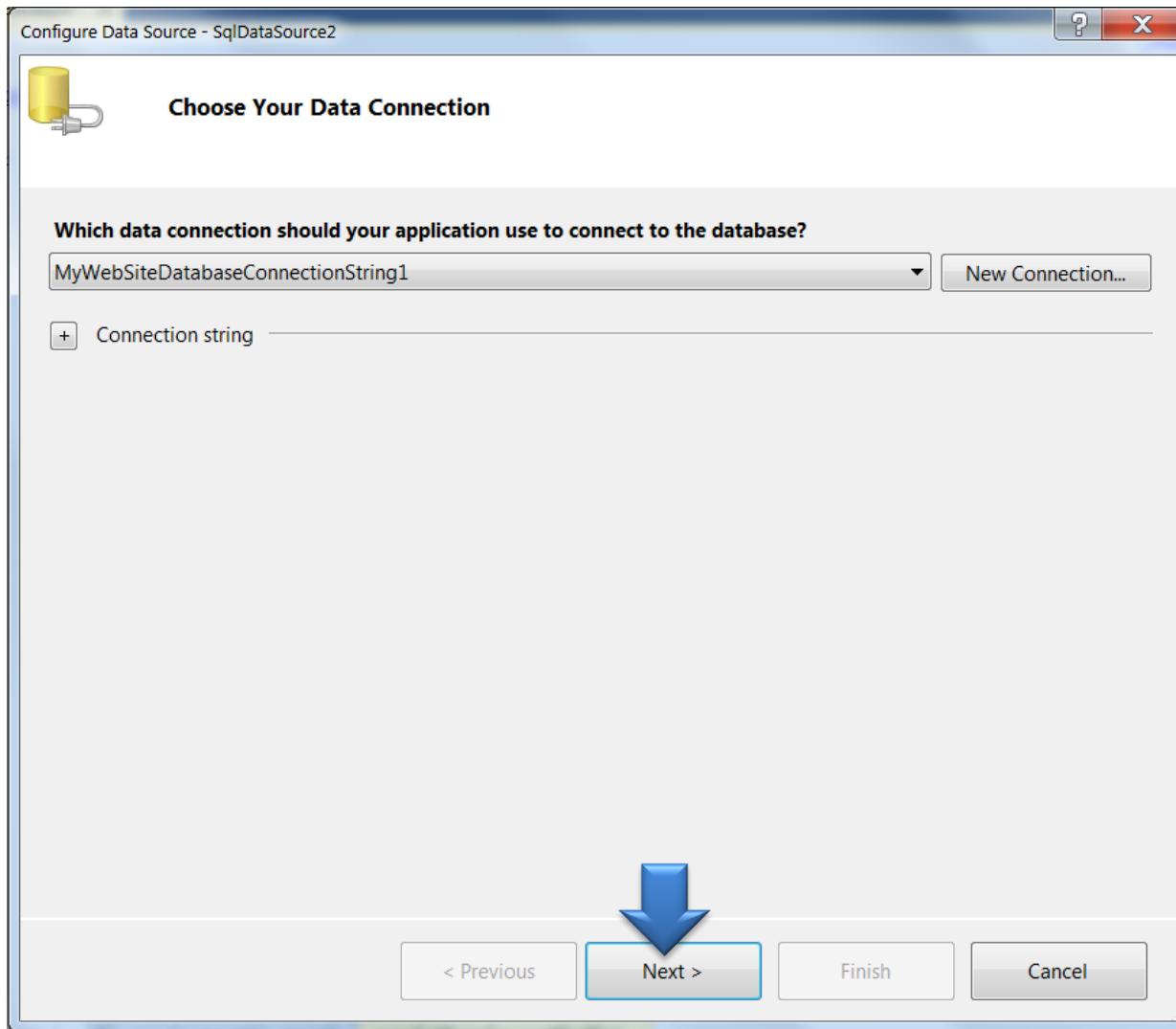
Customizing DetailsView

- In Design View:
 - Drag a **SqlDataSource** control next to **SqlDataSource1**



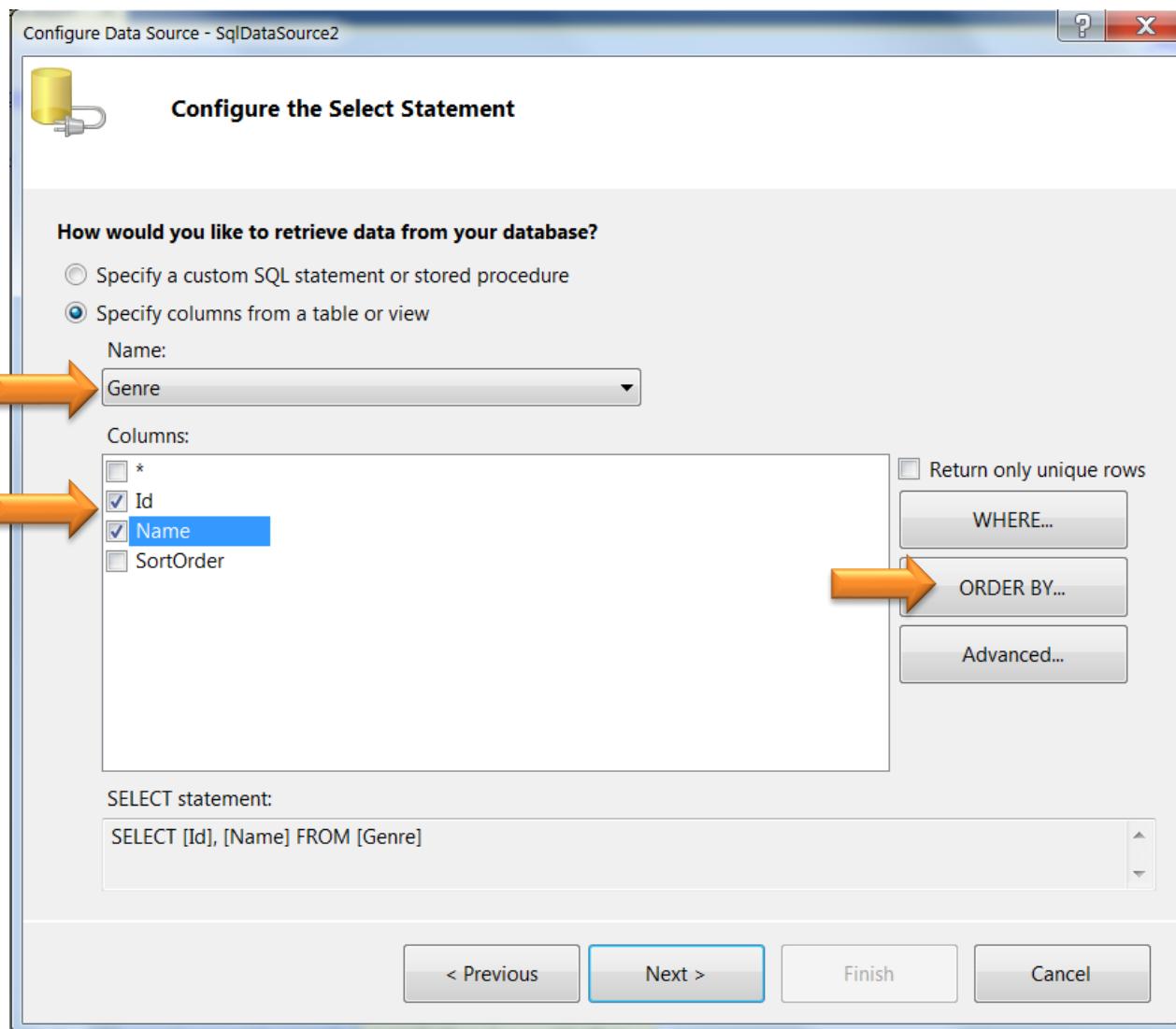
Example

Customizing DetailsView



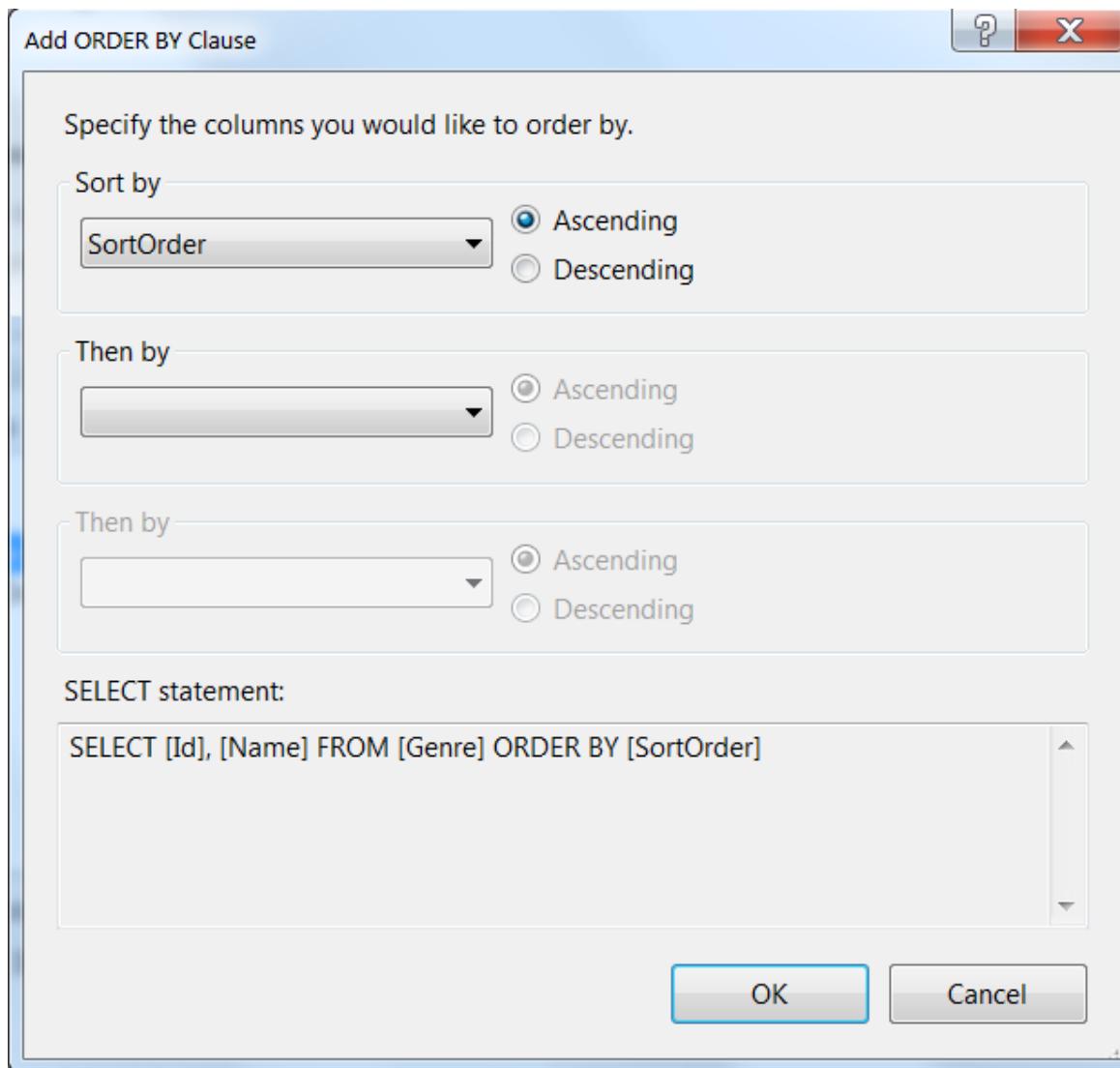
Example

Customizing DetailsView



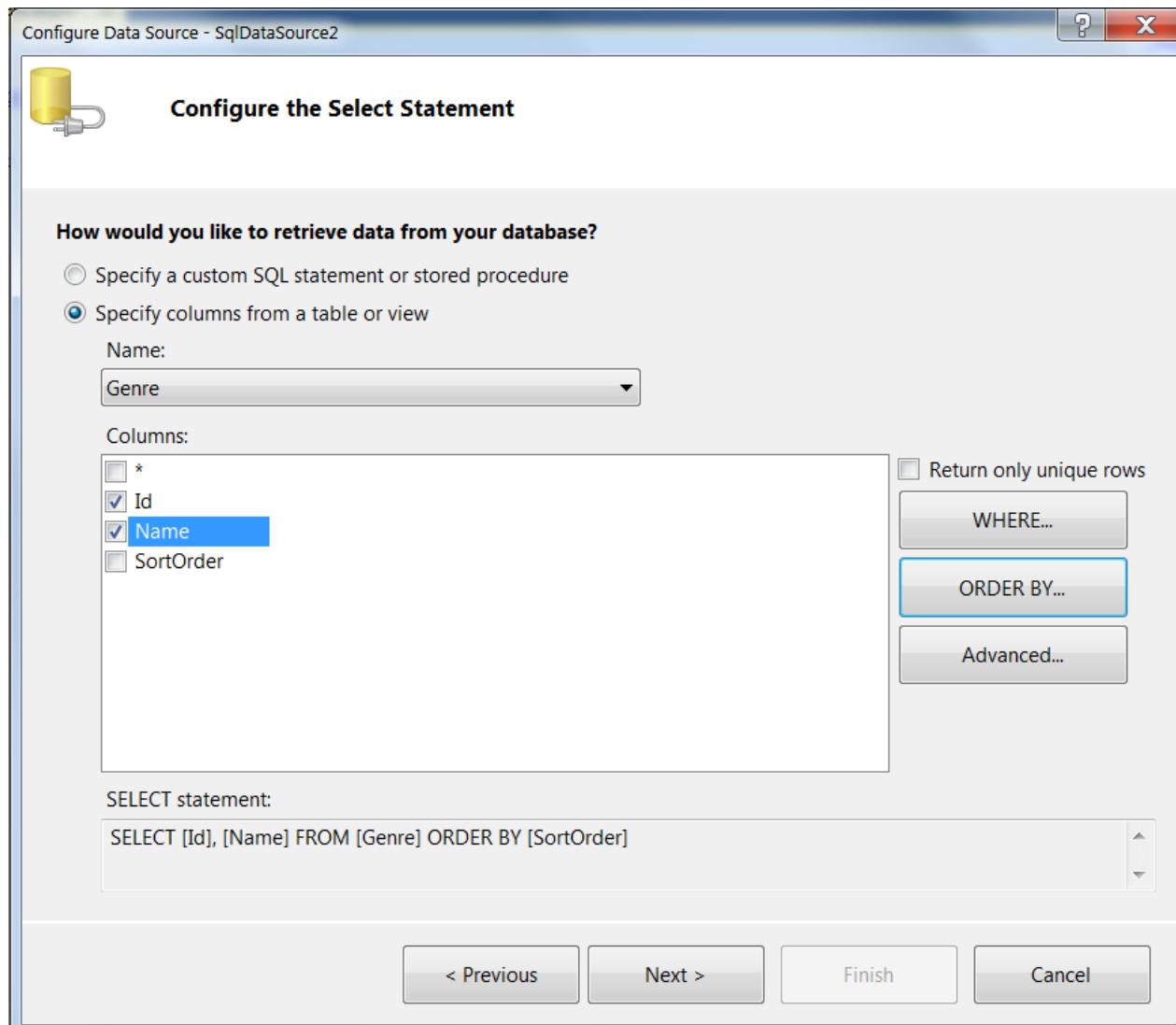
Example

Customizing DetailsView



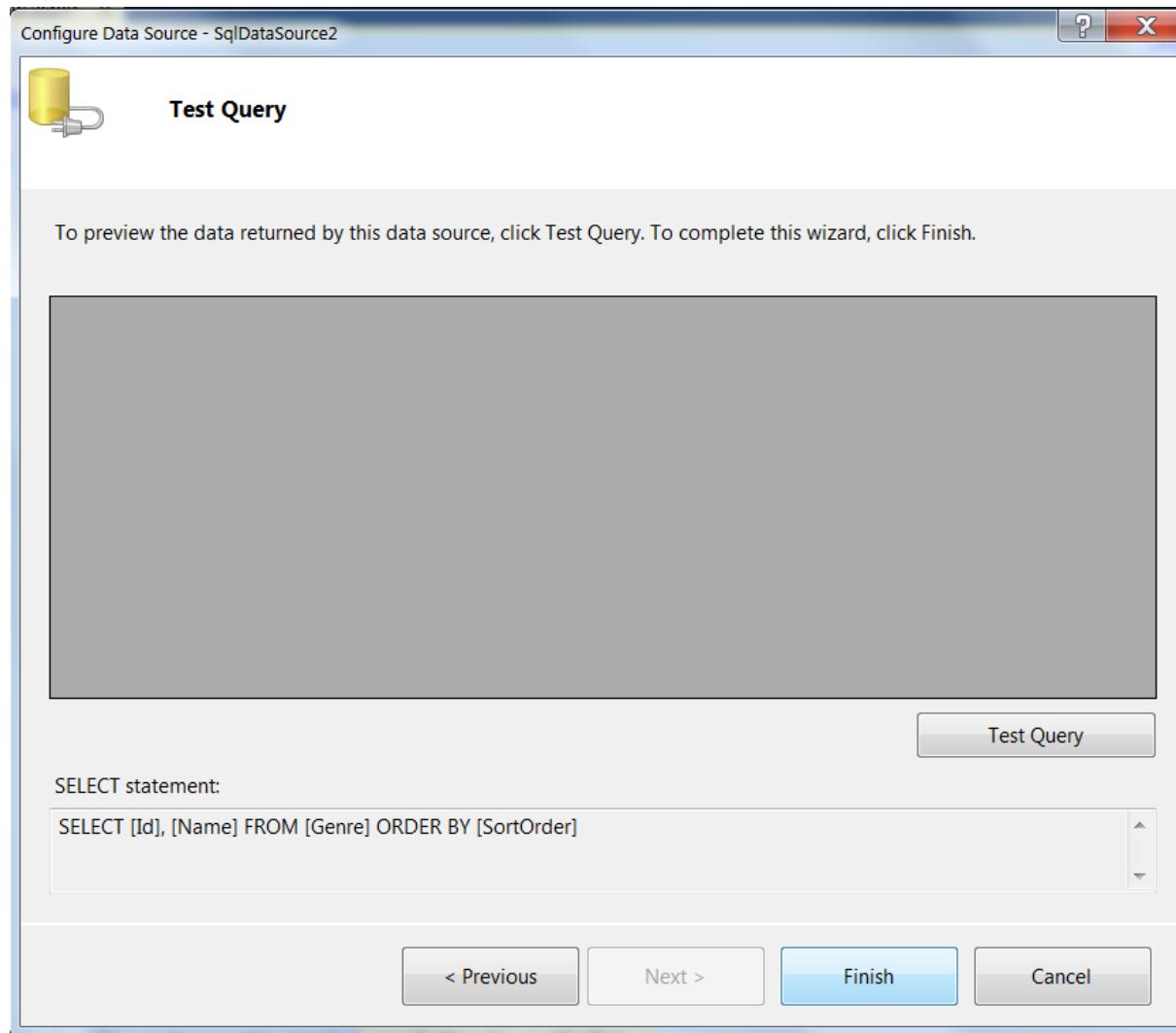
Example

Customizing DetailsView



Example

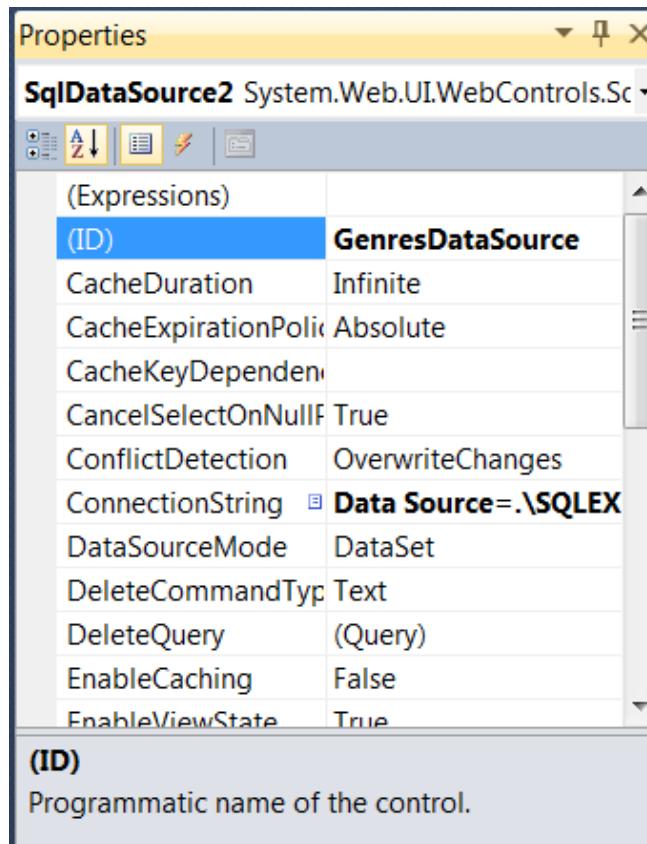
Customizing DetailsView



Example

Customizing DetailsView

- Change ID of `SqlDataSource2`
 - `ID = GenresDataSource`



In Source View:

```
<asp:TemplateField HeaderText="GenreId" SortExpression="GenreId">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox4" runat="server" Text='<%# Bind("GenreId") %>'></asp:TextBox>
    </EditItemTemplate>
    Remove <InsertItemTemplate>
    → <asp:TextBox ID="TextBox4" runat="server" Text='<%# Bind("GenreId") %>'></asp:TextBox>
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label4" runat="server" Text='<%# Bind("GenreId") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```

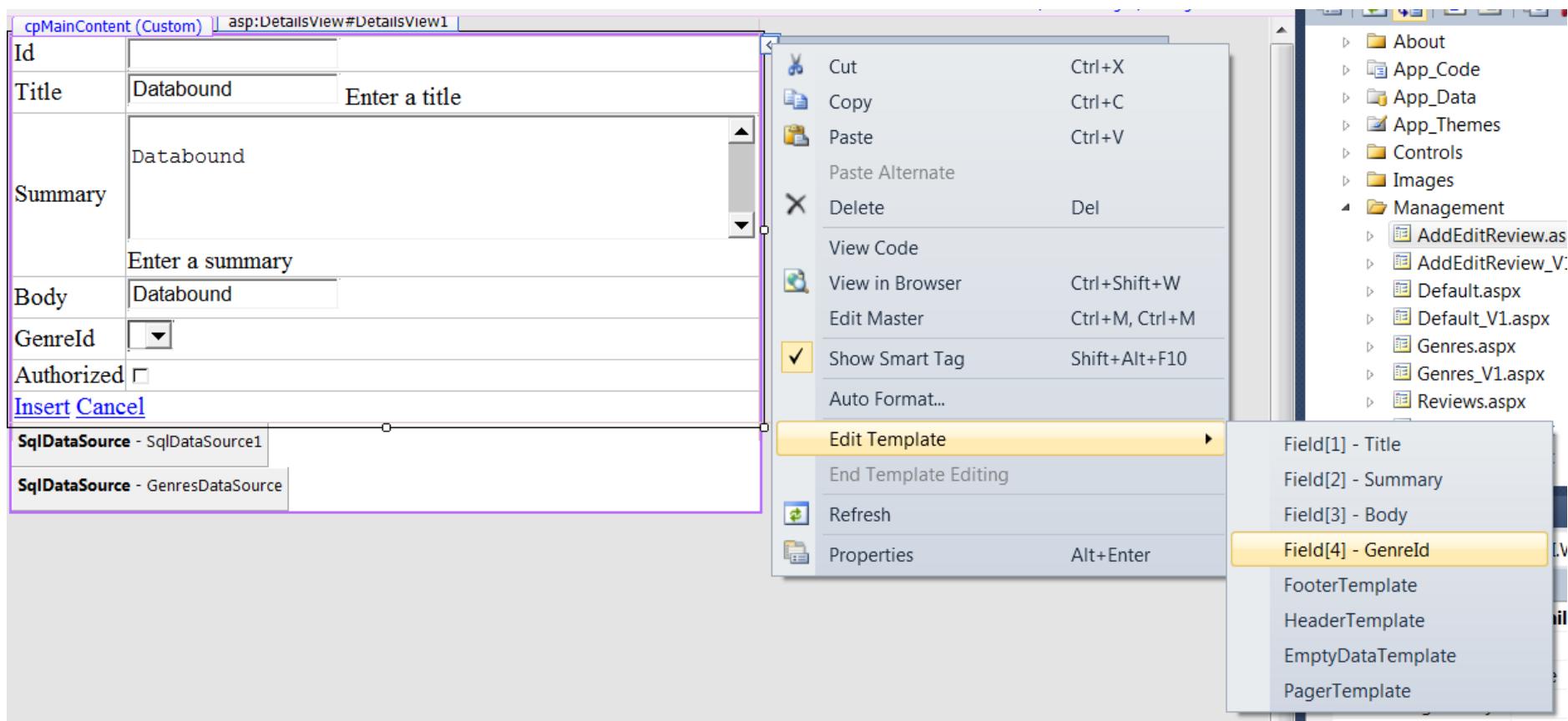


```
<asp:TemplateField HeaderText="GenreId" SortExpression="GenreId">
    <EditItemTemplate>
        <asp:TextBox ID="TextBox4" runat="server" Text='<%# Bind("GenreId") %>'></asp:TextBox>
    </EditItemTemplate>
    <InsertItemTemplate>
        Drag → <asp:DropDownList ID="DropDownList1" runat="server">
        </asp:DropDownList>
    </InsertItemTemplate>
    <ItemTemplate>
        <asp:Label ID="Label4" runat="server" Text='<%# Bind("GenreId") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```

Example

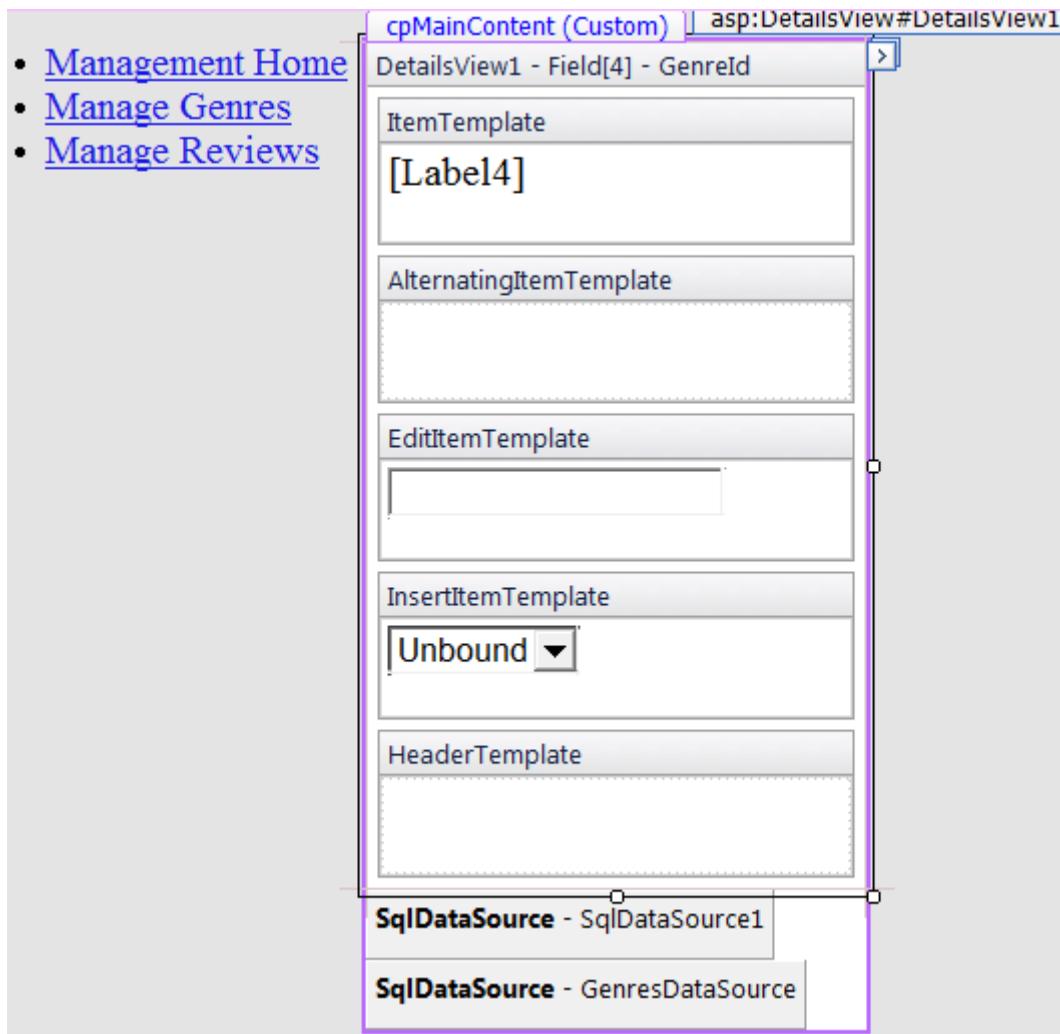
Customizing DetailsView

In Design View → Right-click the DetailsView

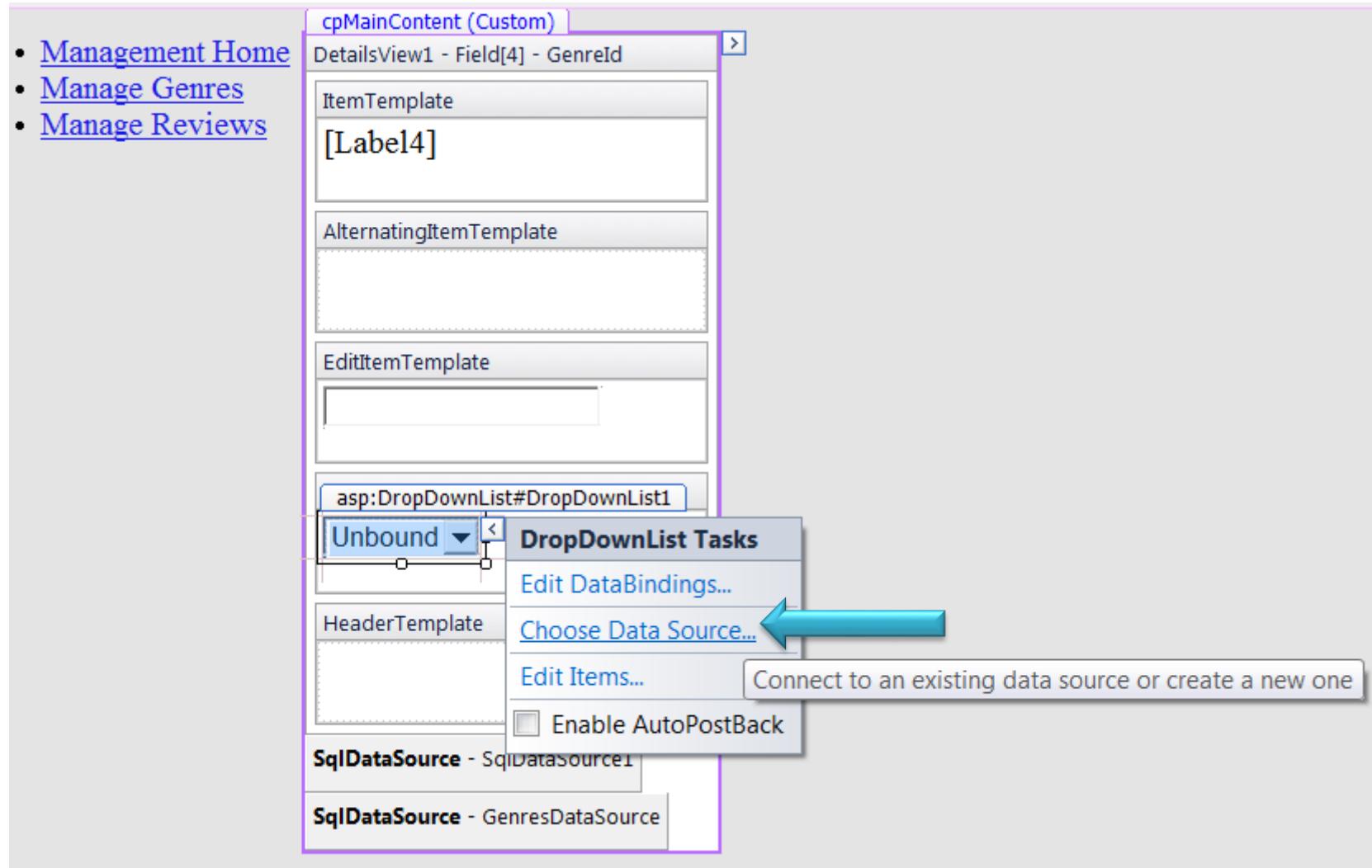


Example

Customizing DetailsView



Open → DropDownList control's Smart Tasks panel





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:



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:



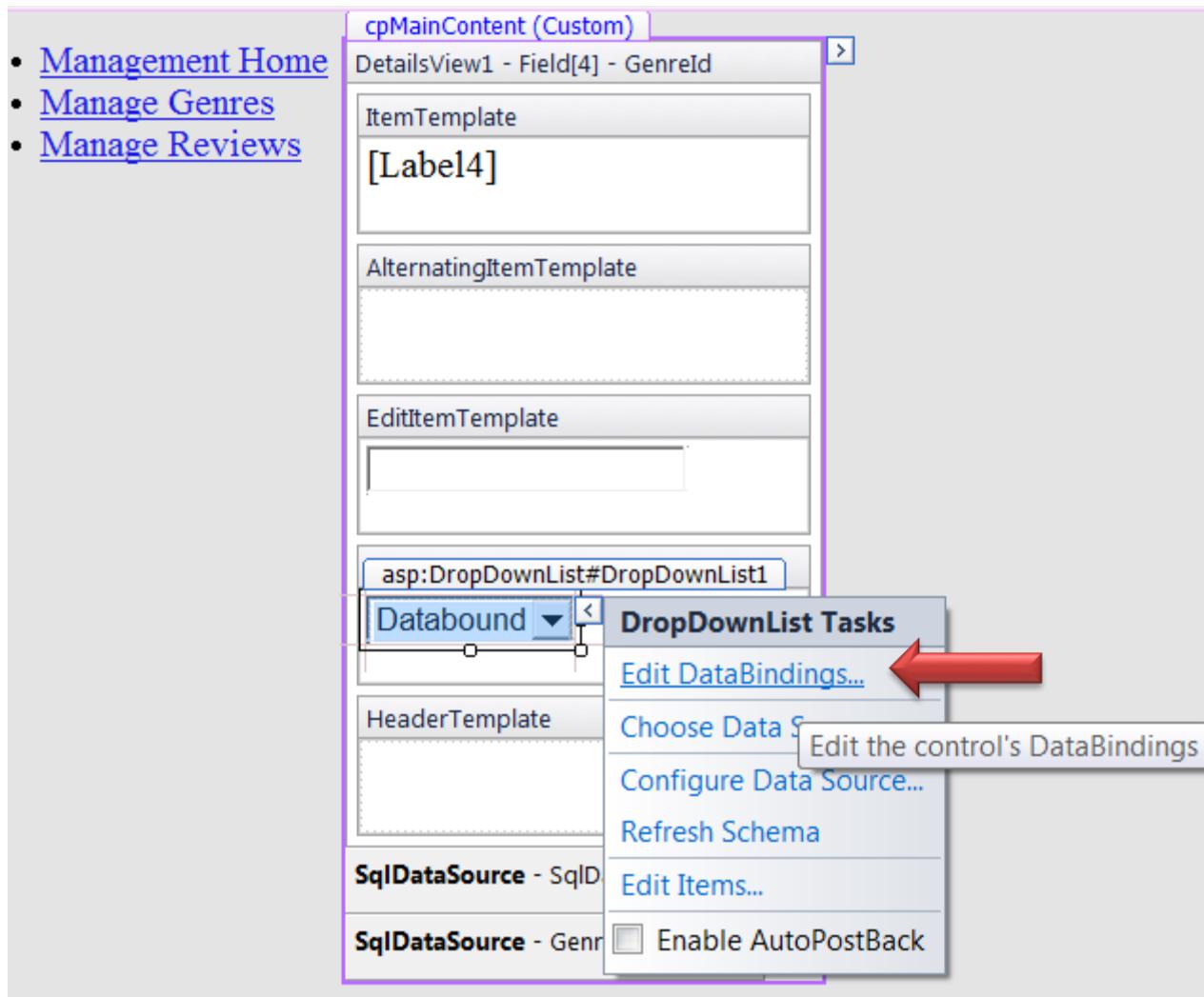
[Refresh Schema](#)

OK

Cancel

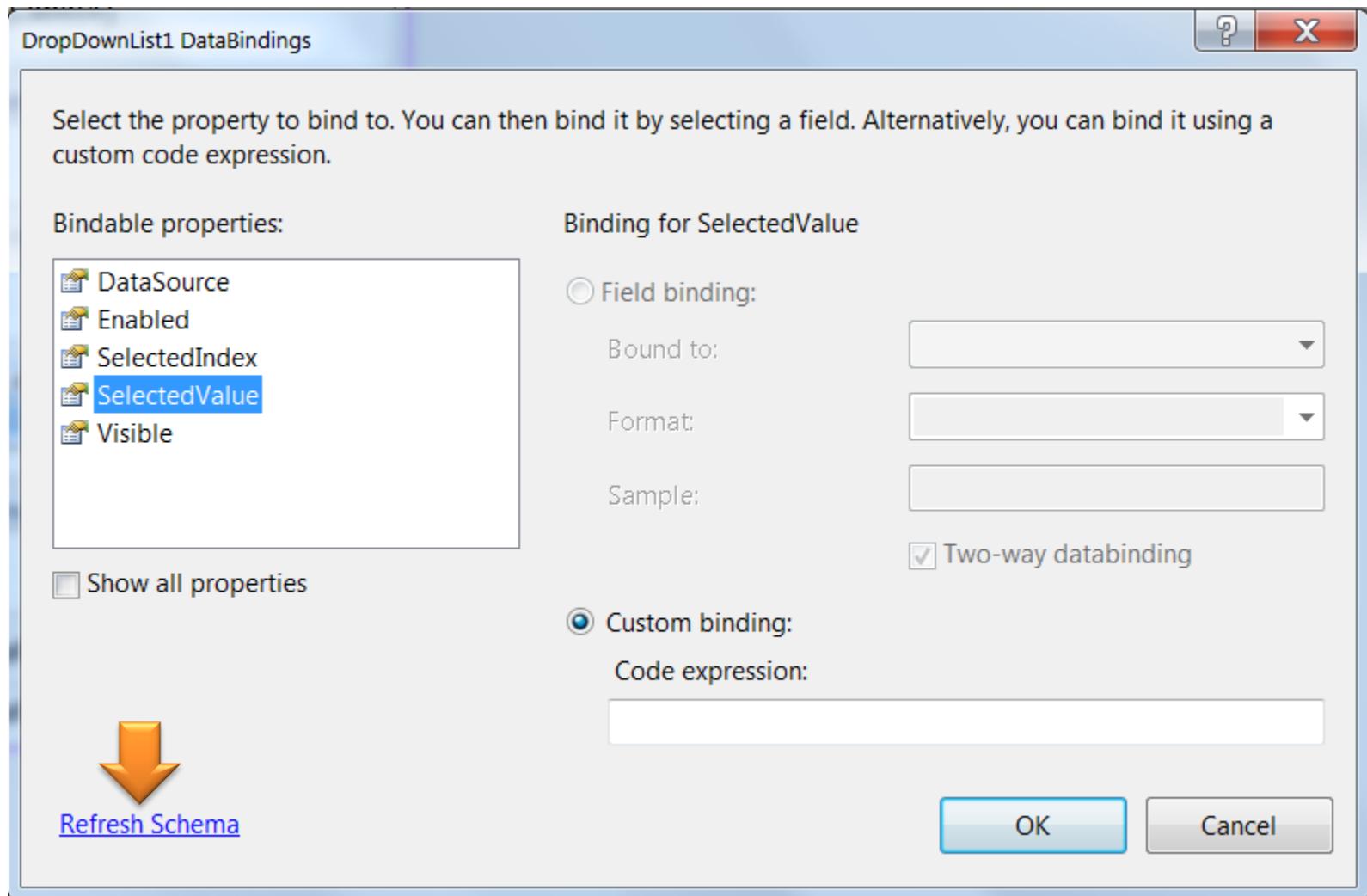
Example

Customizing DetailsView



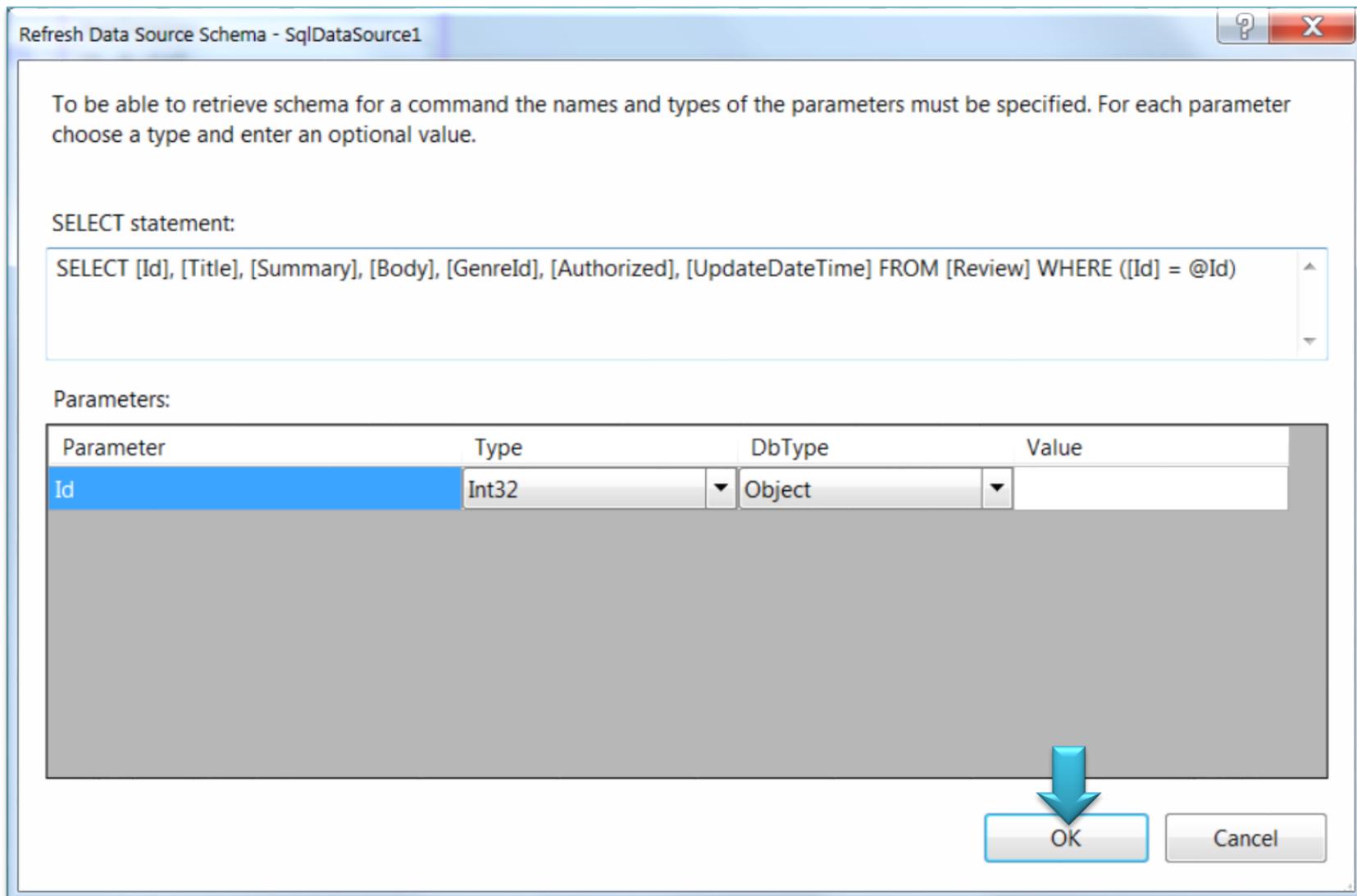
Example

Customizing DetailsView



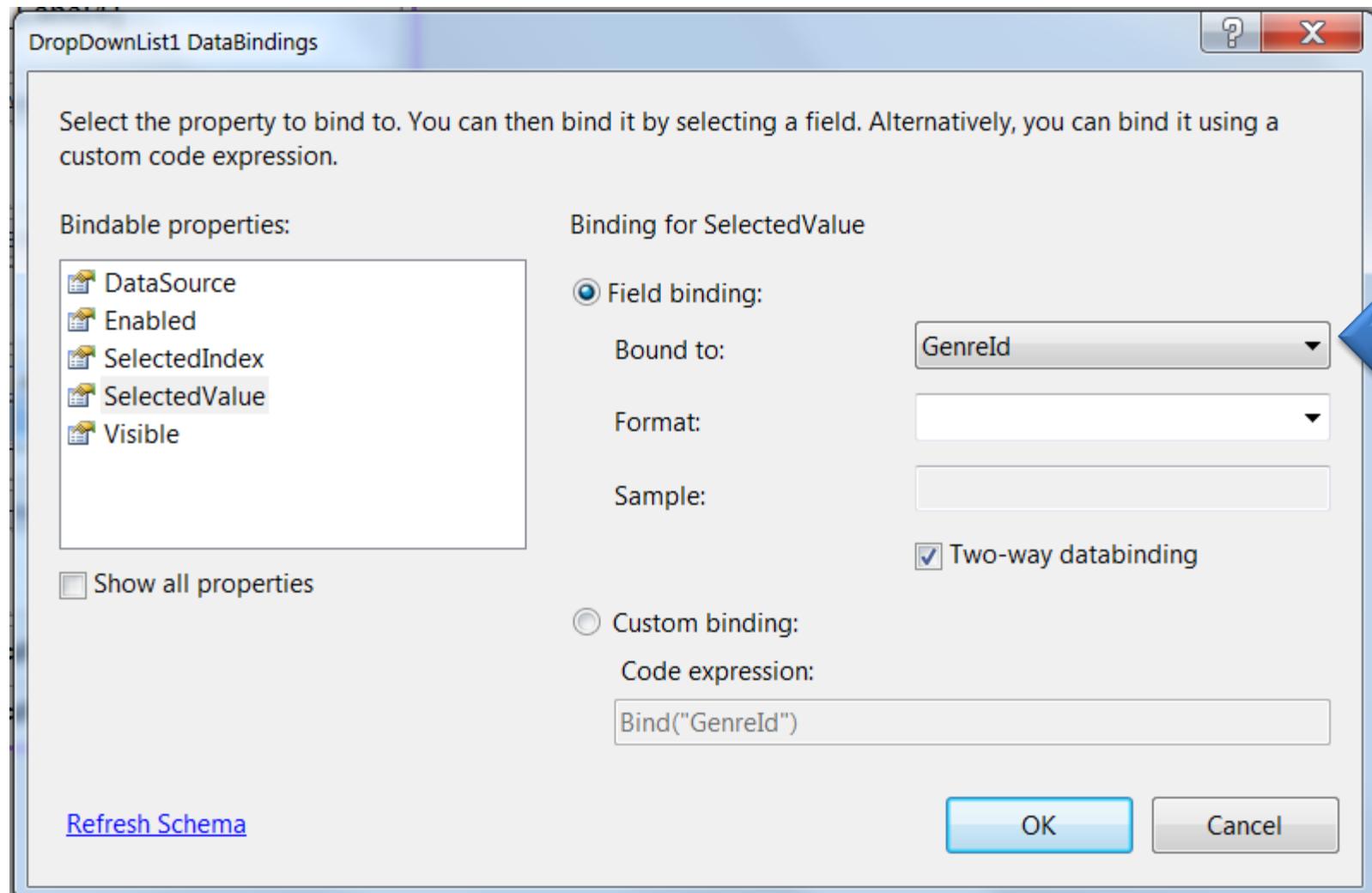
Example

Customizing DetailsView



Example

Customizing DetailsView



Example

Customizing DetailsView

Copy

```
<InsertItemTemplate>
  <asp:DropDownList ID="DropDownList1" runat="server"
    DataSourceID="GenresDataSource" DataTextField="Name" DataValueField="Id"
    SelectedValue='<%# Bind("GenreId") %>'>
  </asp:DropDownList>
</InsertItemTemplate>
```



Paste

```
<EditItemTemplate>
  <asp:TextBox ID="TextBox4" runat="server" Text='<%# Bind("GenreId") %>'></asp:TextBox>
</EditItemTemplate>
```



```
<EditItemTemplate>
  <asp:DropDownList ID="DropDownList1" runat="server"
    DataSourceID="GenresDataSource" DataTextField="Name" DataValueField="Id"
    SelectedValue='<%# Bind("GenreId") %>'>
  </asp:DropDownList>
</EditItemTemplate>
```

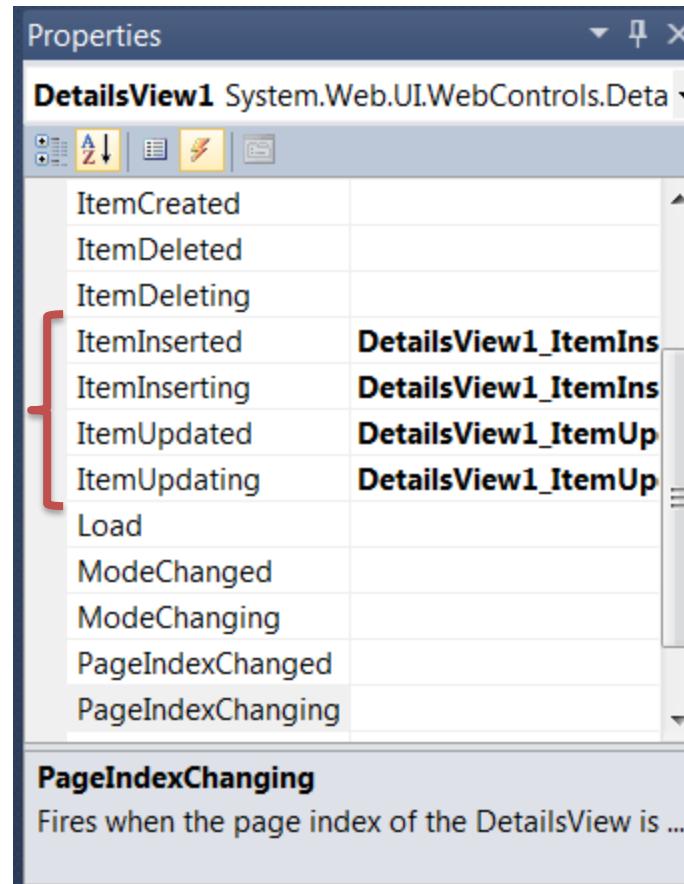
Example

Customizing DetailsView

- In Design View → Click **DetailsView**

Double-click events:

- **ItemInserted**
- **ItemInserting**
- **ItemUpdated**
- **ItemUpdating**



Example

Customizing DetailsView

```
protected void DetailsView1_ItemInserted(object sender, DetailsViewInsertedEventArgs e)
{
    EndEditing();
}
protected void DetailsView1_ItemInserting(object sender, DetailsViewInsertEventArgs e)
{
    e.Values["UpdateDateTime"] = DateTime.Now;
}
protected void DetailsView1_ItemUpdated(object sender, DetailsViewUpdatedEventArgs e)
{
    EndEditing();
}
protected void DetailsView1_ItemUpdating(object sender, DetailsViewUpdateEventArgs e)
{
    e.NewValues["UpdateDateTime"] = DateTime.Now;
}
private void EndEditing()
{
    Response.Redirect("Reviews.aspx");
}
```

Example

Customizing DetailsView

- View **AddEditReview.aspx** in browser.

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Id	
Title	
Summary	
Body	
GenreId	Blues ▼
Authorized	<input type="checkbox"/>
Insert Cancel	

Example

Customizing DetailsView

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Id	15
Title	Kaas Chante Piaf
Summary	By Patricia Kaas
Body	
GenreId	Jazz <input type="button" value="▼"/>
Authorized	<input checked="" type="checkbox"/>
Insert Cancel	

Example

Customizing DetailsView

- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)



- [Management Home](#)
- [Manage Genres](#)
- [Manage Reviews](#)

Title	Authorized	CreateDateTime	Delete
Kaas Chante Piaf	Yes	15.05.2014 16:44	Delete

Jazz

[Insert new review](#)