

**School of Information Technology**

Course : Diploma in Business Informatics

Subject : ITP282 - Enterprise Application Development & Project

AY / Sem : 2018 S2

**Lab 5c**: Displaying data using Databinding

OBJECTIVES:

By the end of this Practical students should be able to:

1. Understand what is Data Binding
2. Use Data controls to display data from a Data Source

**Practical 5c Displaying data using Databinding**

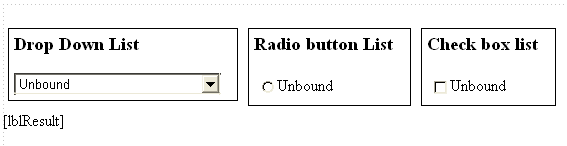
**Data Binding**

Data Binding refers to the process of associating a Data Source such as Data Reader with a Data Consumer such as Drop Down List.

|  |  |
| --- | --- |
| SqlDataReader, acting as a data source , is bind to the Drop Down List control |  |

**Exercise 1: Data Binding**

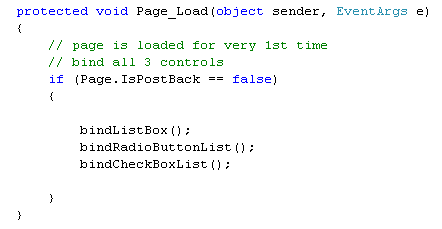
1. Copy **Lab5c** folder into your computer. Open up **Ex1.aspx**. There are three controls which allows you to perform databinding:



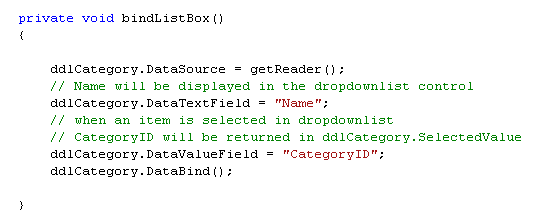
View the properties of the three controls from the Property Window. Notice that AutoPostback = True. You must always ensure this is set if you want the controls to post back to the server each time the control is selected.



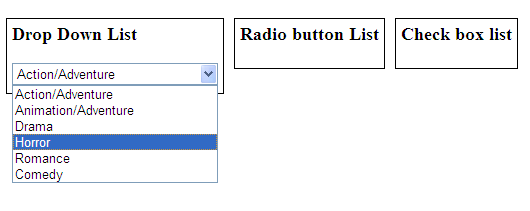
1. Go to code behind. Notice that when the page first loads, a check is made to see if IsPostback is false. If so, it means that the web form is loaded for very first time, and we bind the controls using data reader within the three methods shown below:



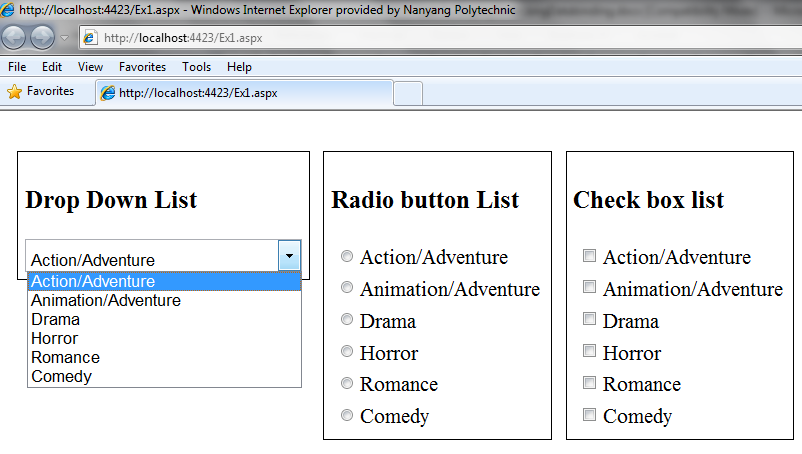
1. The code for bindListBox is already completed for you. Notice that this method calls another method getReader to retrieve data via a Sql Data Reader, the code for which is given to you as well.



1. Build and run your program. You should see a screen similar to the one shown:



1. Using the codes in BindListBox as a guide, complete the codes for BindCheckBoxList and BindRadioButtonList. In both cases, you should assign DataTextField = "Name" and DataValueField="CategoryID". Build and run your program once completed. The web page should look similar to the following:

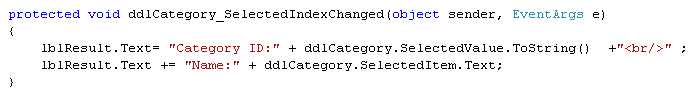


**Displaying Selected Items**

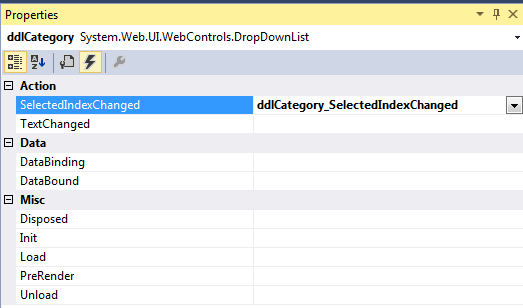
We now want to display an item selected by the user in any of the above controls. When any of the items is selected, an event *SelectedIndexChanged* will be raised. In this event handler, we may retrieve the item selected using SelectedValue and SelectedItem properties of the control. Remember we have assigned DataTextField = "Name" and DataValueField="CategoryID"; Therefore, SelectedValue and SelectedItem properties will give the values of the CategoryID and Name respectively of the item selected. For Drop Down List control the code is as follows:

Retrieves CategoryID

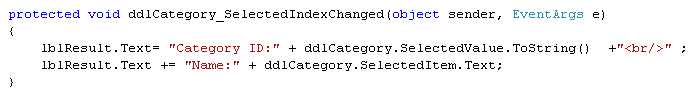
Retrieves Name



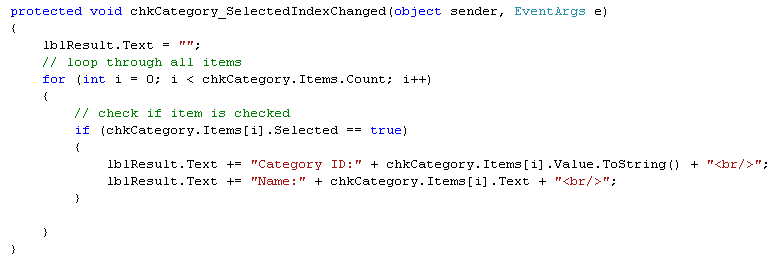
1. To generate the event handler, click on the Drop Down List control. In the Property Window, select. Double click on SelectedIndexChanged.



1. In event handler, add in the following codes.



1. Build and run your program. Select an item in the Drop Down List control to confirm the selected item Category ID and Name is displayed.
2. Complete the codes for both Radio Button List to display the name and value of the item selected. Use the codes for Drop Down List as a guide to complete the codes for the Radio Button List.
3. Complete the codes for both Check Box to display the names and values of the items selected. However, for this control, more than one item may be selected at one time. Hence, we need to modify the code to *loop* through all the items to check whether each item is selected. The code is as follows:



1. Build and run your program.

**Data Controls**

In ASP.NET, there are a number of new Data Controls that allow the developer to bind data from data source control to the web server controls are:

* Data List

Used for displaying data in a format using templates and styles.

* GridView

Like DataList, GridView also displays data in a Table. In addition, it also allows sorting and paging

* Details View

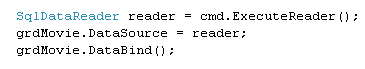
Use for displaying a single record from a data source in a table, where each data row represents a field in the record. . It is often used in combination with a GridView for master-detail scenarios .

* Form View

Used for displaying a single record from a data source in a table. It offers more control over form layout as compared to Details View.

**Exercise 2: Grid View: Displaying data and customisation**

* 1. Open **Ex2.aspx**. Notice there is a Grid View ID=grdMovie already created.
  2. Under Page Load, when the web form *first* loads, call a method BindGridView (Refer to Ex 1 Step 1.2 to guide you if you are not sure).
  3. Create the method BindGridView. In this method, complete the codes which use Sql Data Reader to retrieve ProductID, Title, Price and Image1FileName from DVDShop.mdf. Sql Data Reader is then used to bind to the GridView as follows:



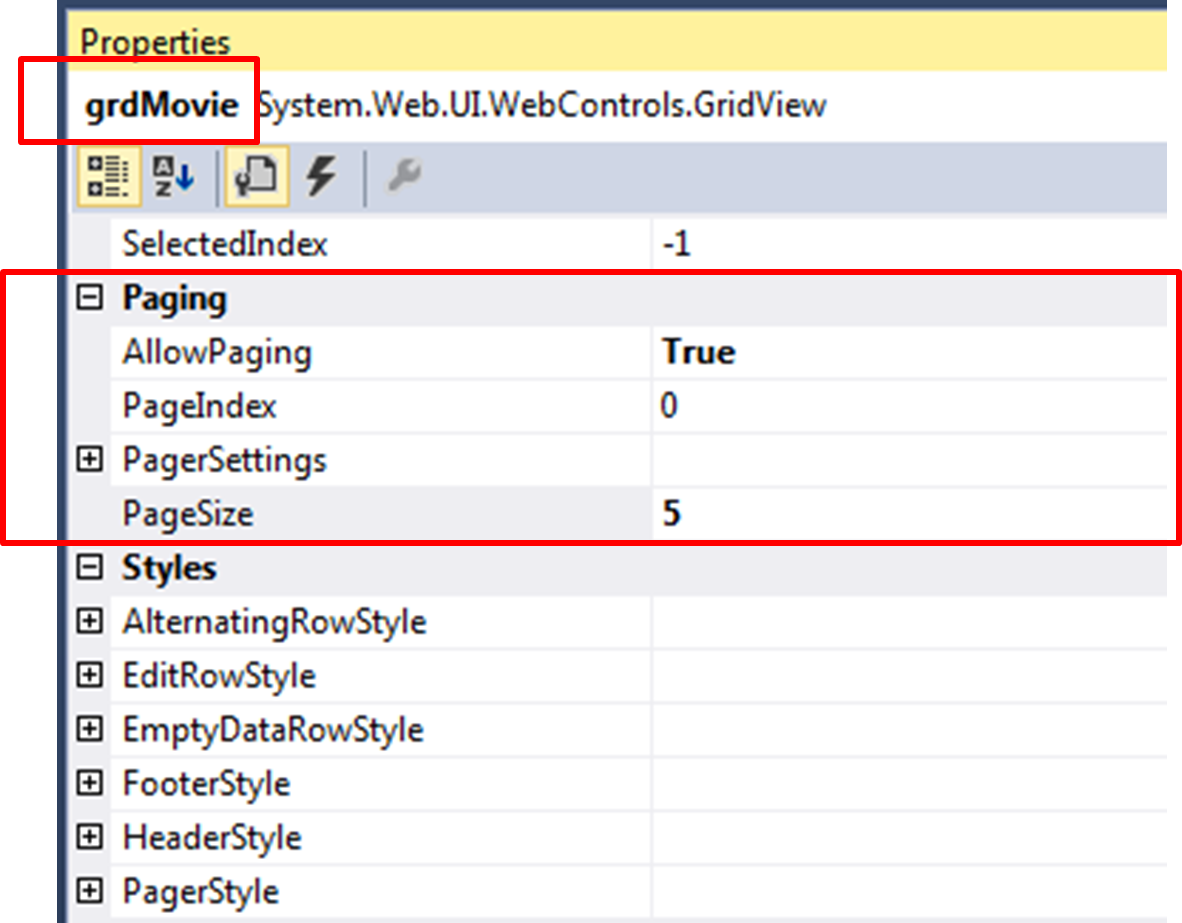
* 1. Build and run your program. It should look similar to the following:



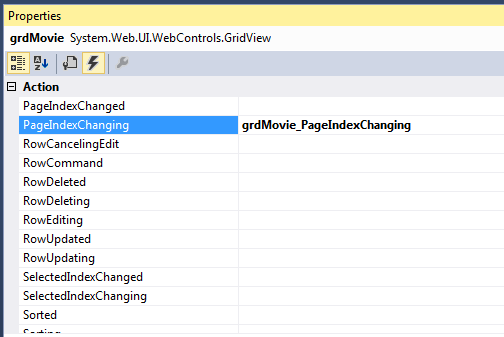
**Paging**

We can easily implement paging in Grid View using the following steps:

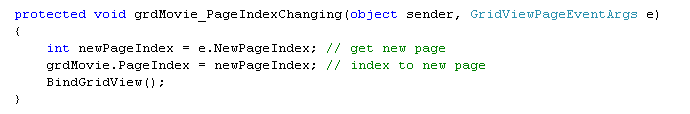
* 1. Under the Properties Window of Grid View, set AllowPaging=True and PageSize=5.



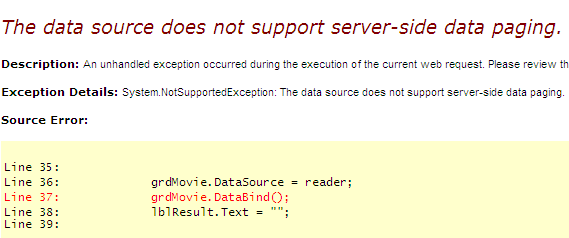
* 1. Select and double click on PageIndexChanging to create the event handler when a page is selected.



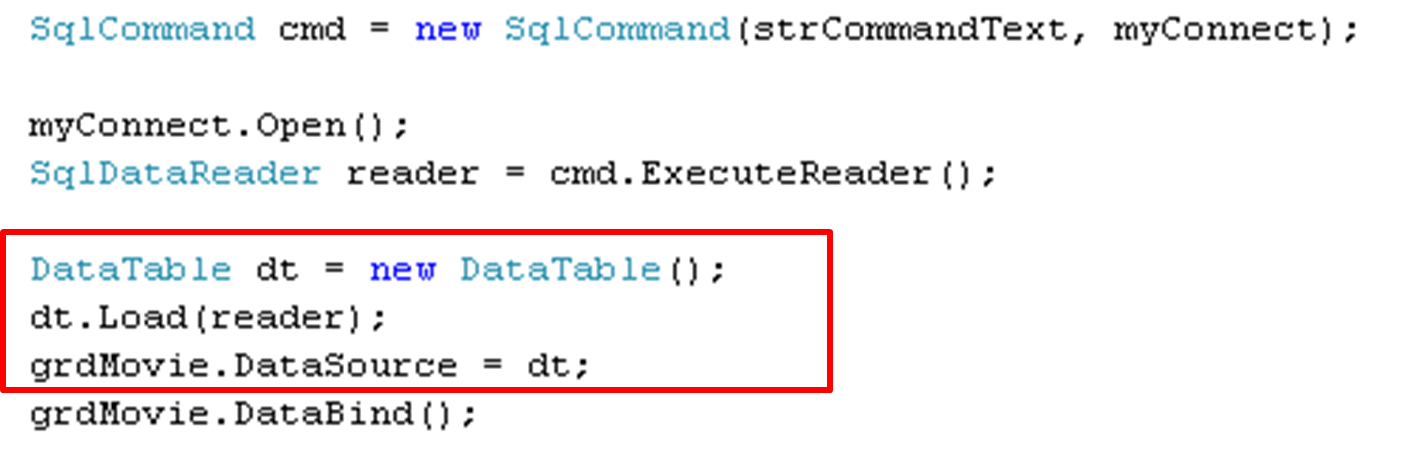
1. In the PageIndexChanging event handler, add in the following codes to support paging:



* 1. Build and run your program. What happens?



* 1. As seen above, unfortunately, Data Reader does not support paging. We can overcome this by loading the results of data reader into a Data Table, as shown below.

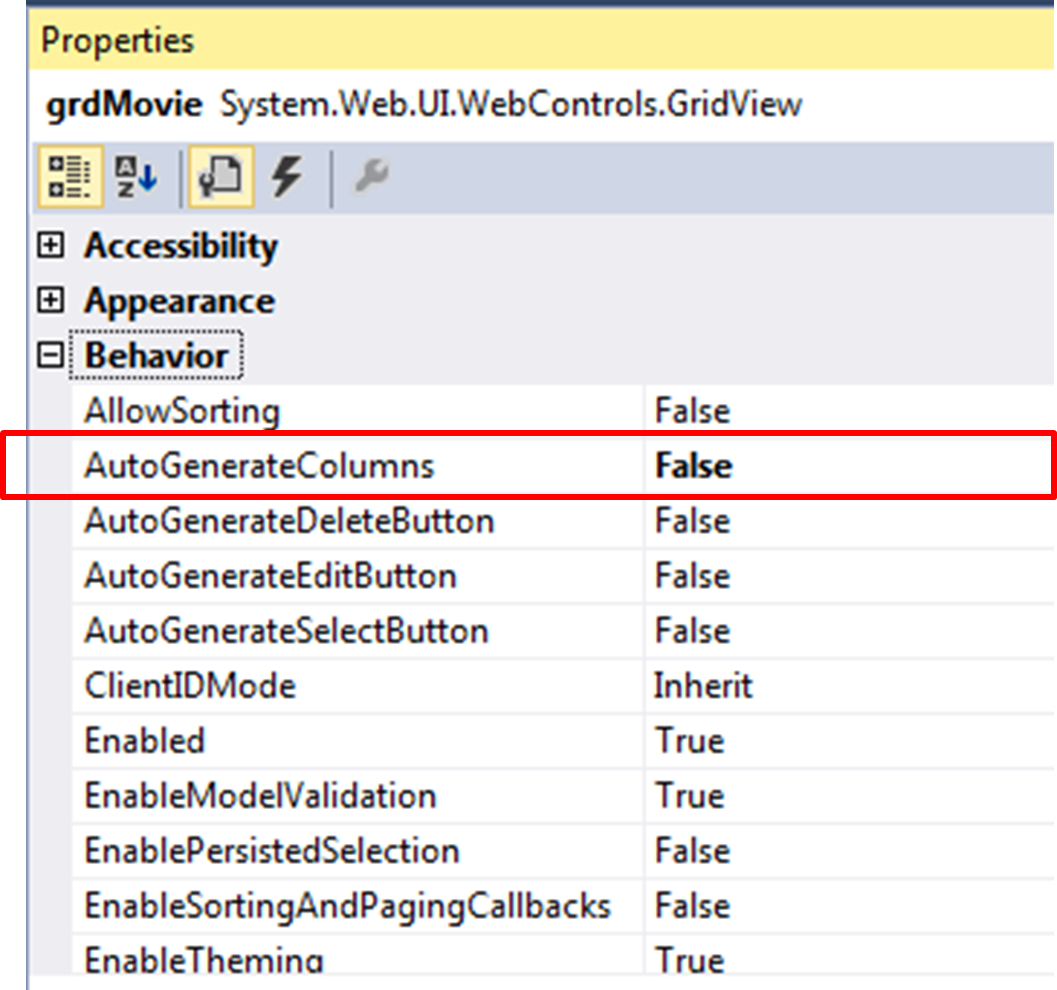


* 1. Build and run your application once again. The program should be working.

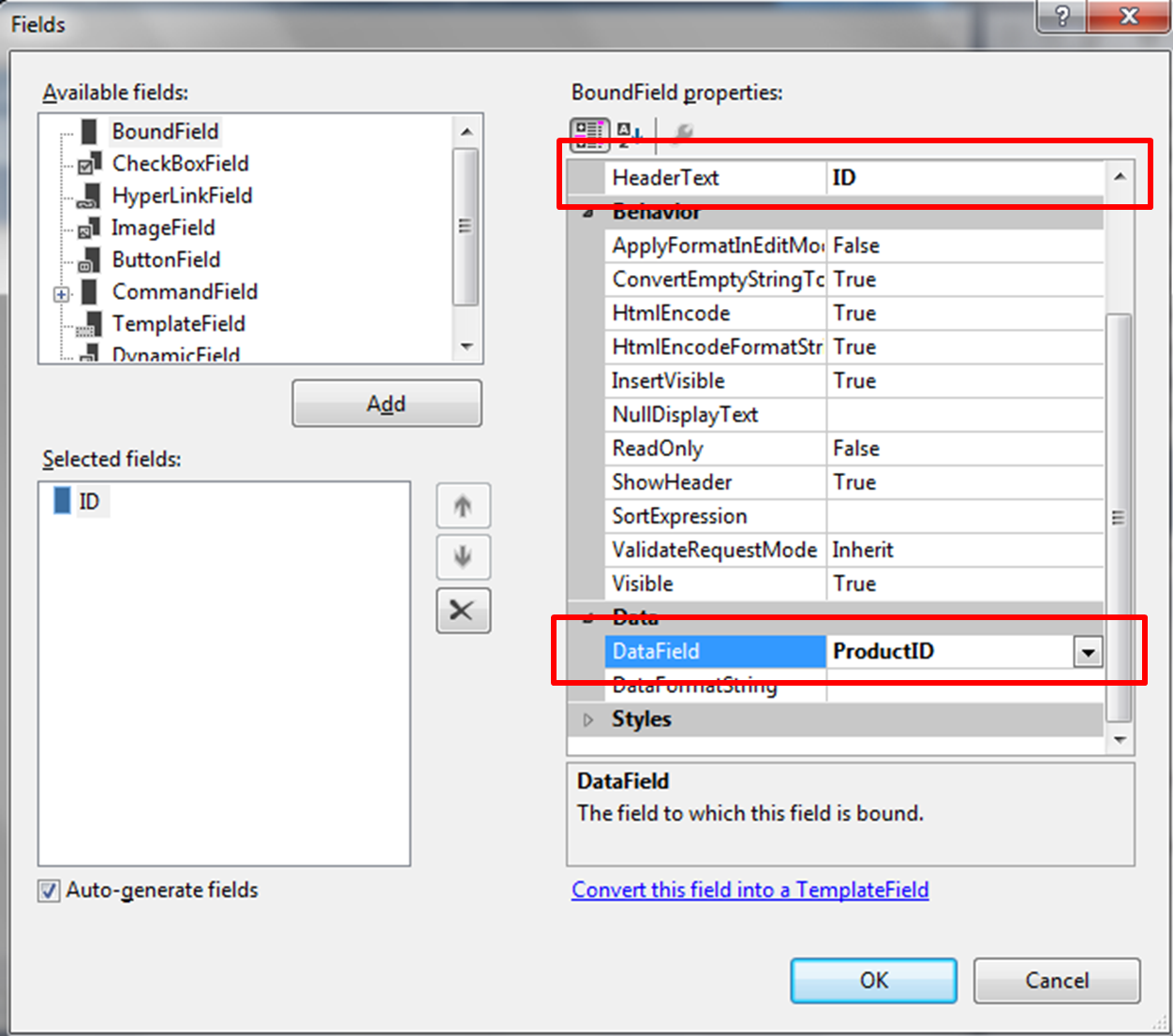
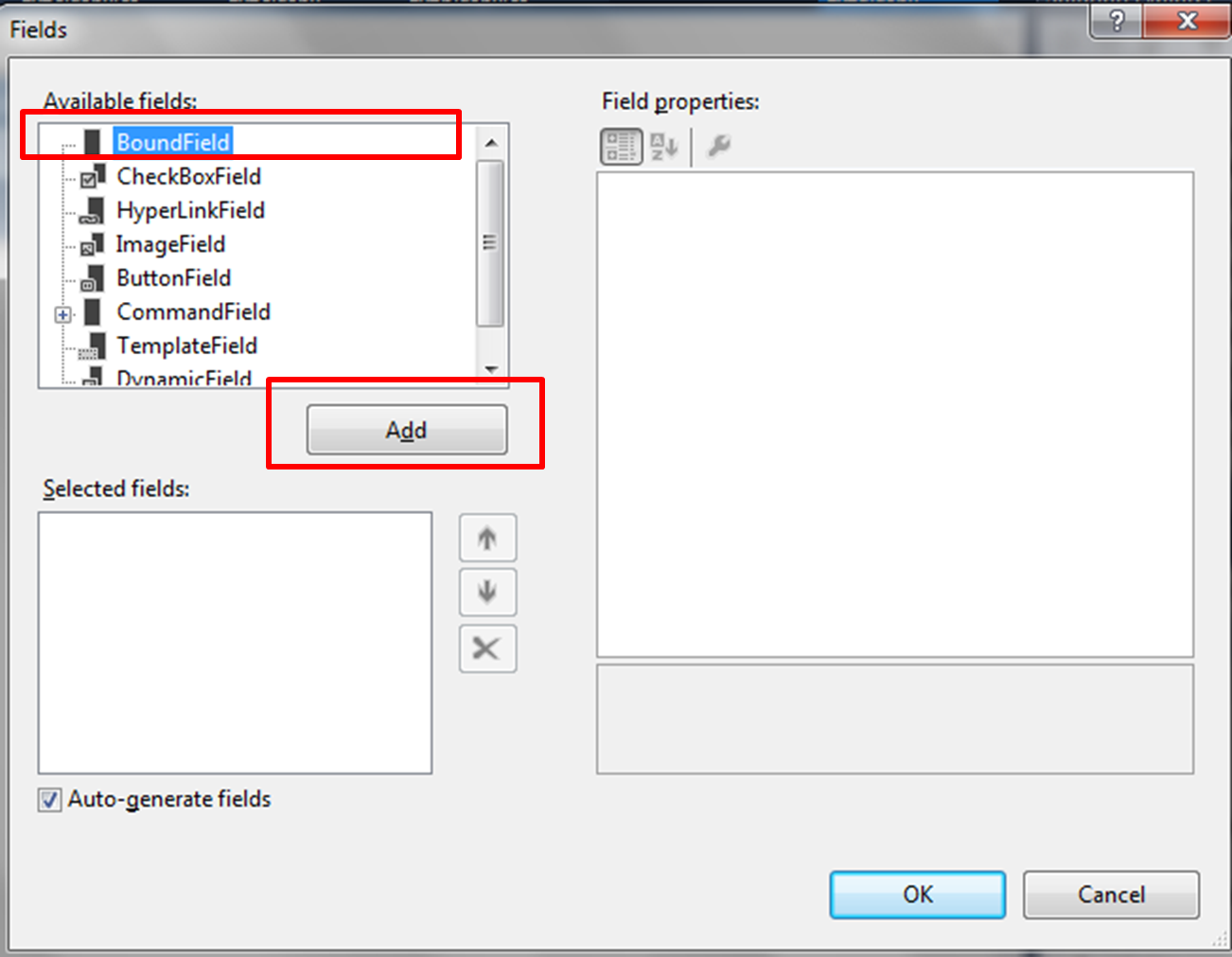
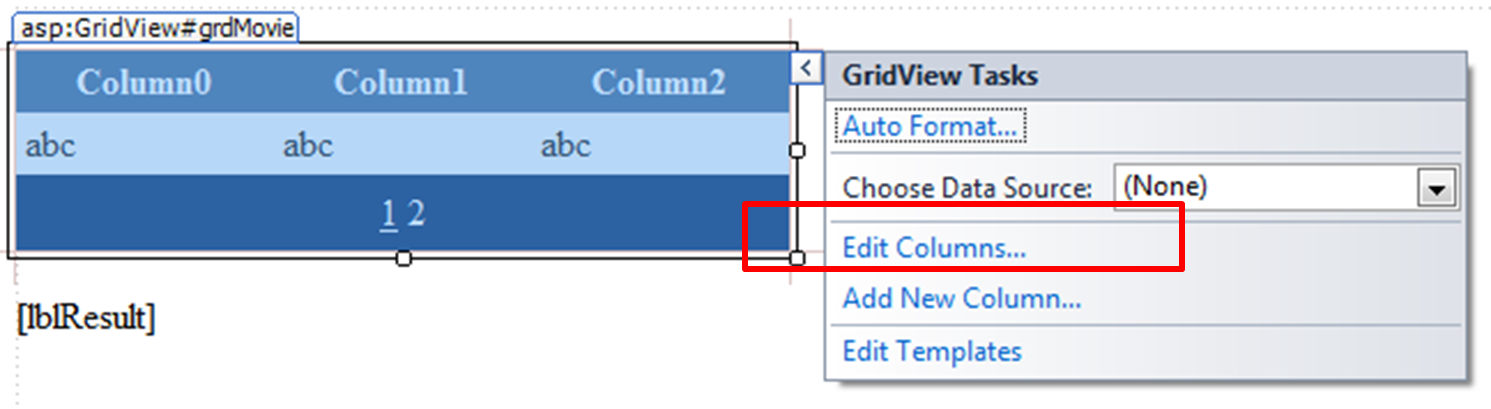
**Customising GridView**

So far, *all* the columns retrieved by data reader are displayed by default in Grid View. We can customise data to display only columns we need.

* 1. Set AutoGenerateColumns = False.



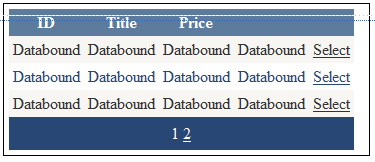
* 1. Add in your own columns you want to display by selecting Edit Columns. Select BoundField and click on Add button. Under Field properties, assign Header Text = ID and DataField=ProductID. Note Header Text is any text entered by you. It will be displayed in the Grid View header. DataField must be assigned the name of column of the records retrieved from DVDShop.mdf.



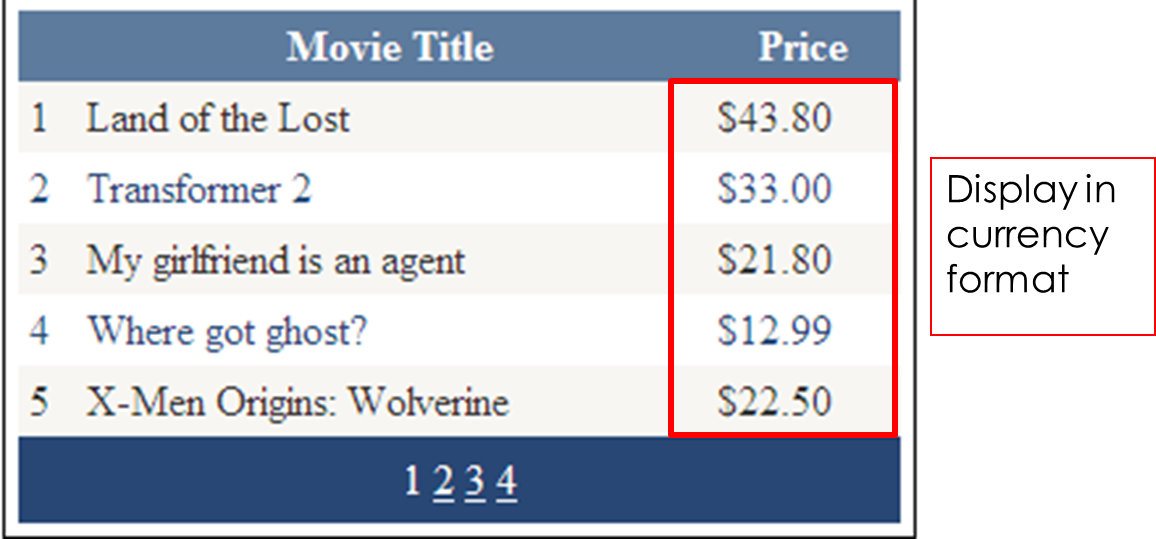
* 1. Repeat by adding 2 more BoundFields with the following information:

|  |  |  |  |
| --- | --- | --- | --- |
| **Header Text** | **DataField** | **DataFormatString** | **Remarks** |
| Movie Title | Title |  |  |
| Price | Price | {0:C} | To display currency |

* 1. The design view for the GridView as following:

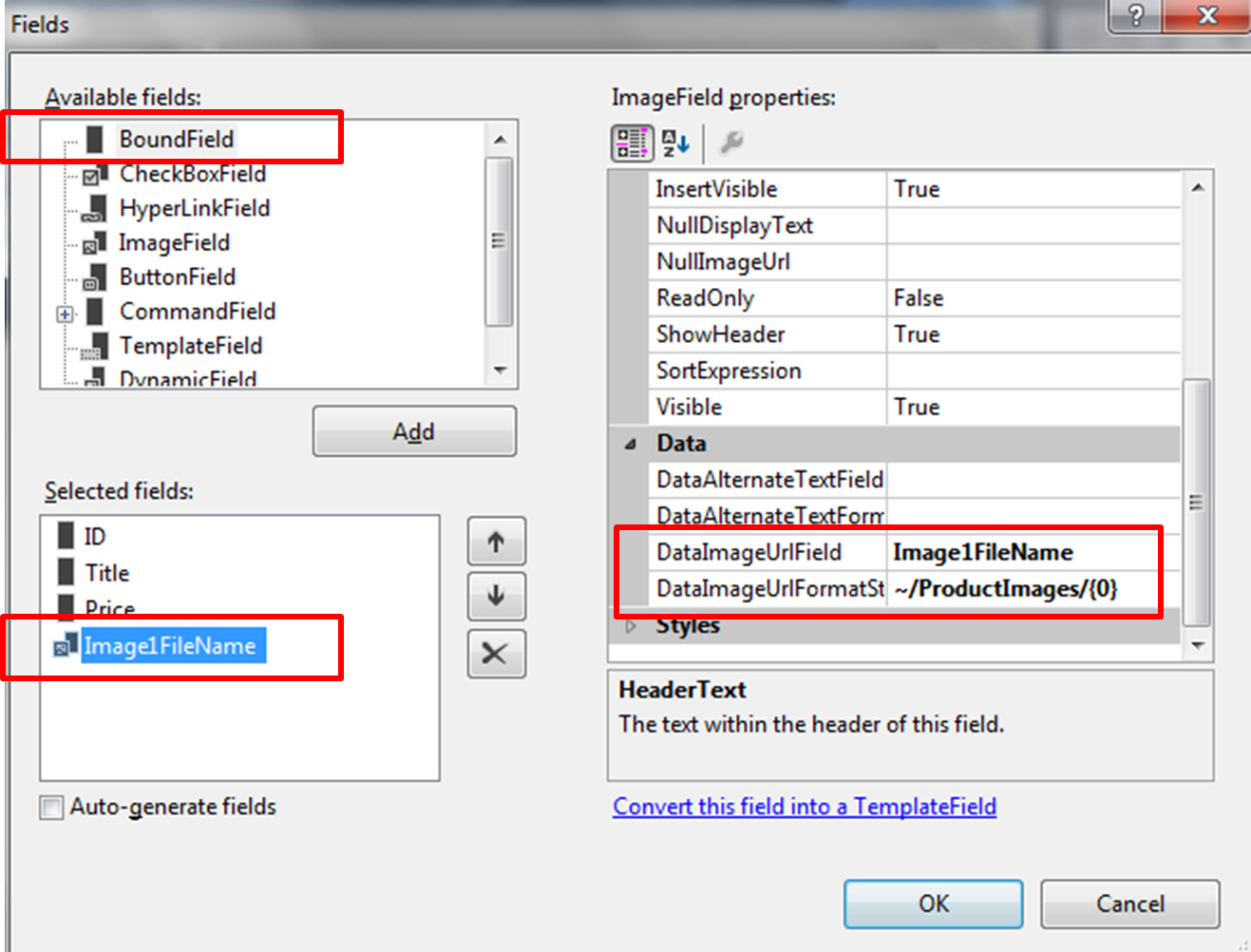


* 1. Build and run your program. It should look as follows:



**Displaying images in Grid View**

* 1. Add in an ImageField into the Grid View. Under DataImageUrlField, assign Image1FileName. Under DataImageUrlFormatString, assign ~/ProductImages/{0}.

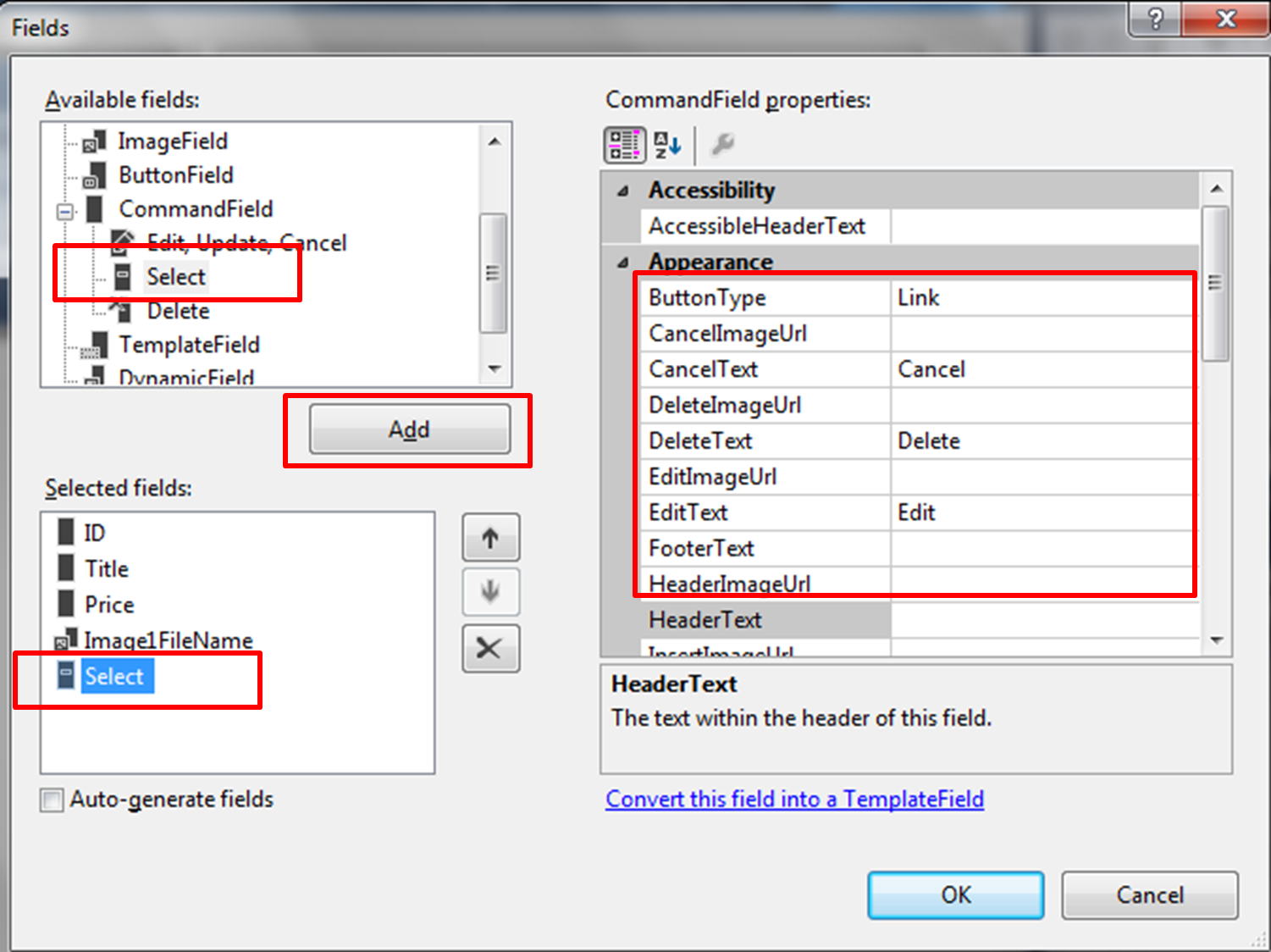


* 1. Build and run your program. The images of the movies should be displayed in the Grid View.

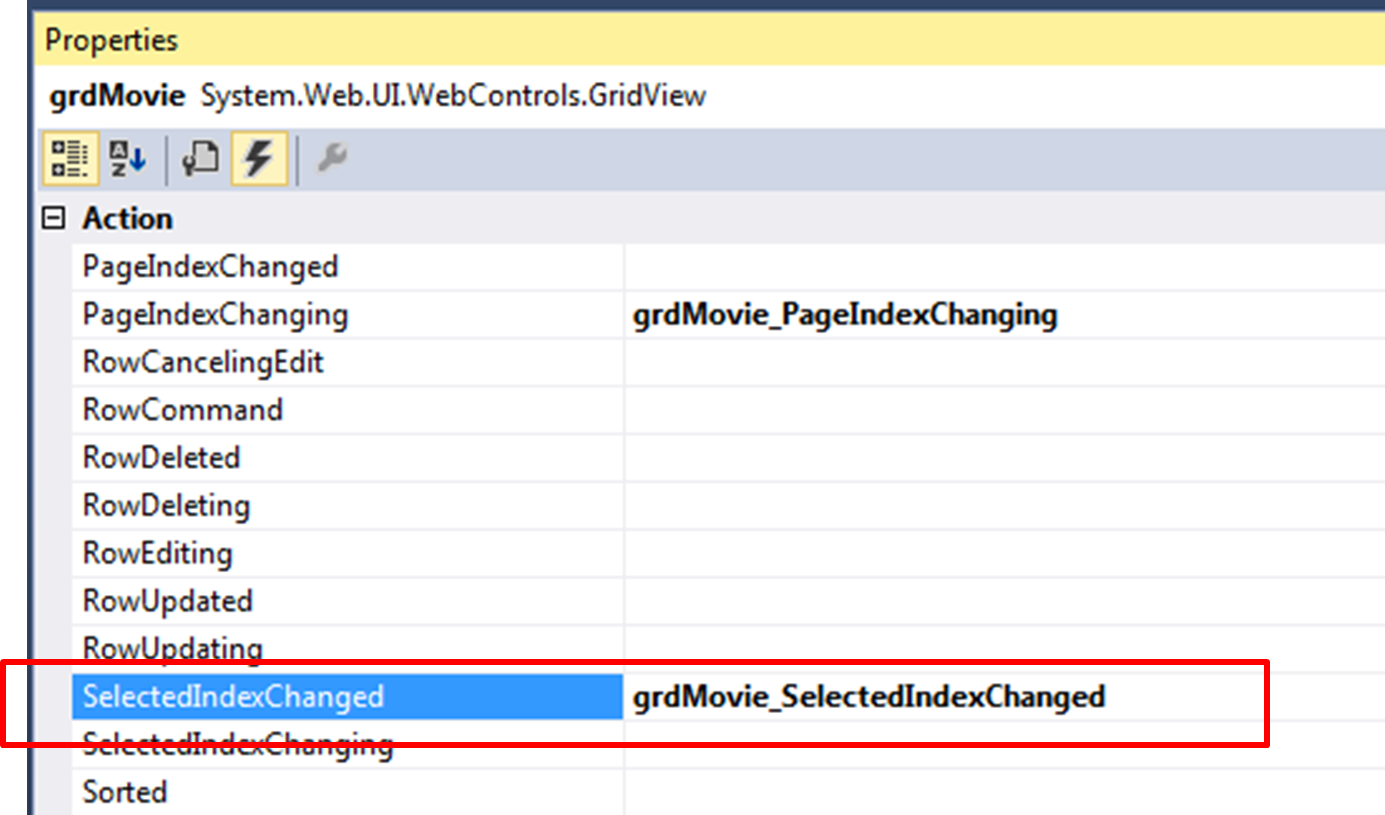
**Displaying items selected in Grid View**

Lastly, we may also display items selected in a Grid View using these steps:

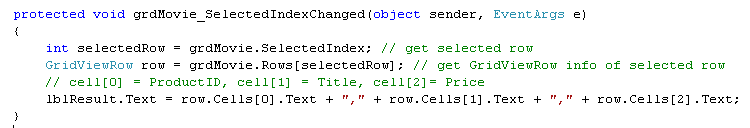
* 1. Add in a Select Command Field into the Grid View:



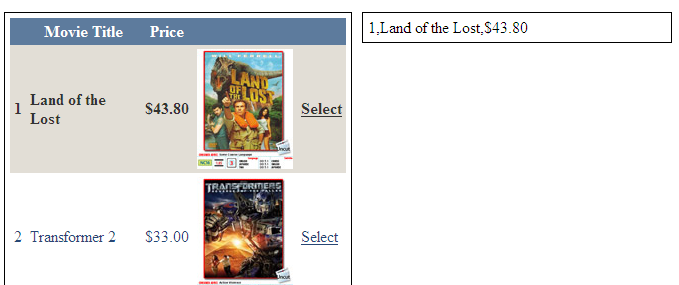
* 1. Double click on SelectedIndexChanged to create the event handler.



* 1. Within the event handler, add in the following codes:



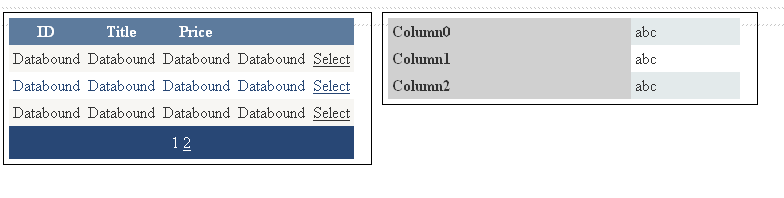
* 1. Build and run your application. The display should be similar to the following:



**Exercise 3: Details View**

Details View is used for displaying a single record from a data source in a table, where each data row represents a field in the record. It is often used in combination with a Grid View for master-detail scenarios.

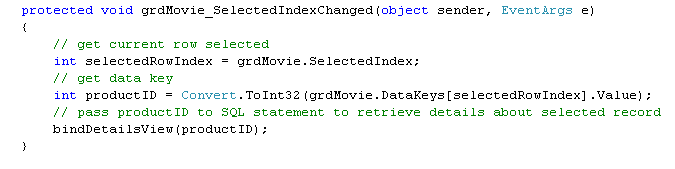
* 1. Copy and paste Ex2.aspx into Solution Explorer and rename it Ex3.aspx.
  2. Replace the label on the right with a Details View ID= DetailsViewMovie as shown:



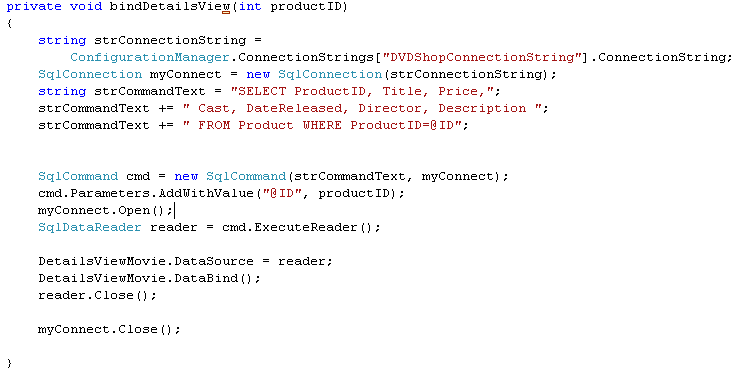
* 1. When a record is selected in Grid View, we need a way to pass the primary key of that record to Details View so that the relevant information of that record can be retrieved from database. One way of doing this is to make use of Data Key property in the Grid View. Enter ProductID in DataKeyNames as shown:



* 1. In the SelectedIndexChanged Event Handler, we modify the code to retrieve the Data Key (i.e. the ProductID of record selected) and pass it to a method bindDetailsView to retrieve the detailed record for that record and bind to Details View.



* 1. The code for bindDetailsView is shown:

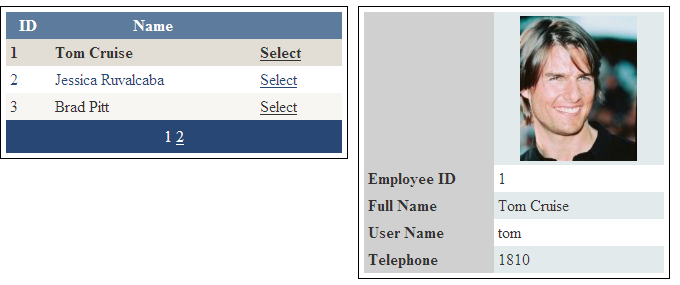


* 1. Build and run your program. When a record is selected, details of the record selected is as shown :



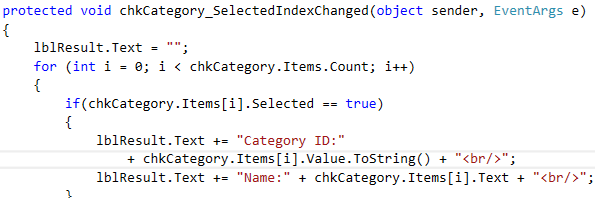
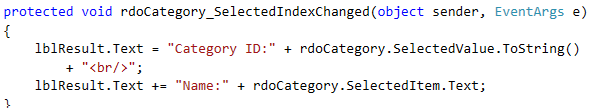
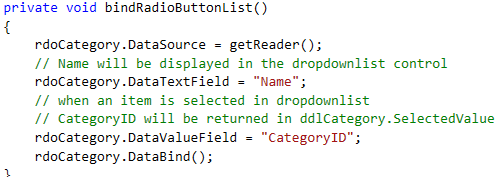
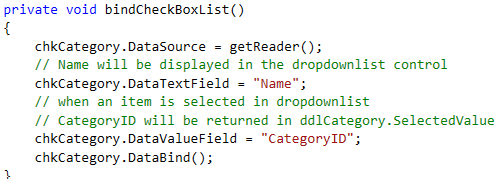
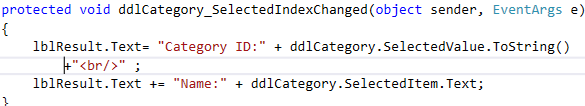
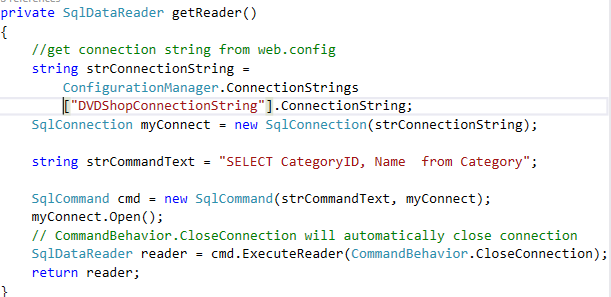
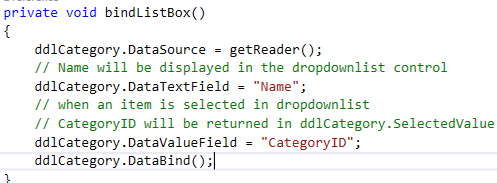
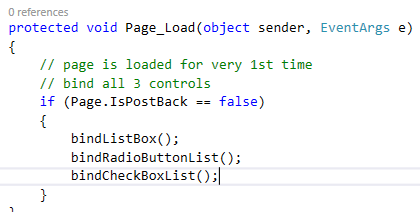
**(Optional) Exercise 4: Applying Grid View and Details View**

* 1. Open up **Ex4.aspx**. Noticed that a Grid View and Details View is created for you
  2. Using EmployeeDB.mdf, write codes so that the Grid View displays the Employee ID and Name. The Grid View allows paging with paging size = 3. When a record in Grid View is selected, the details of that record are displayed in the Details View. The connection string EmployeeConnectionString is already created for you in web.config.

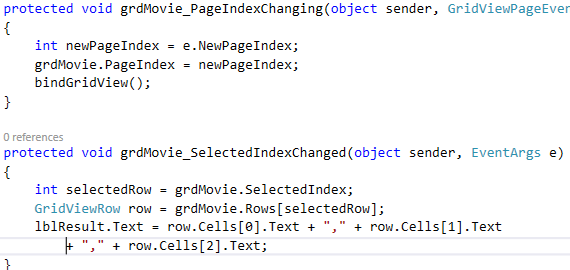
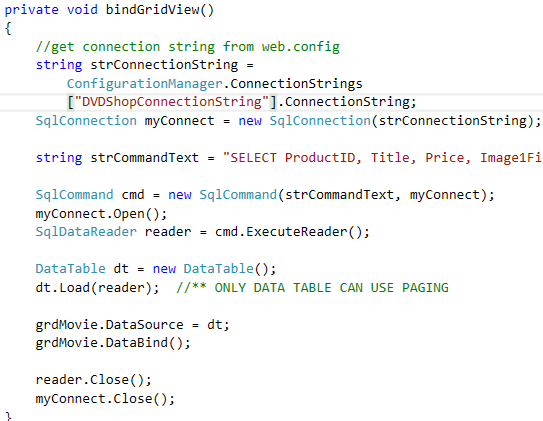
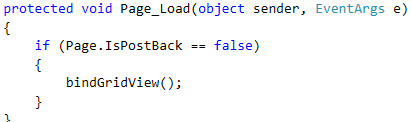


**========== End ==========**

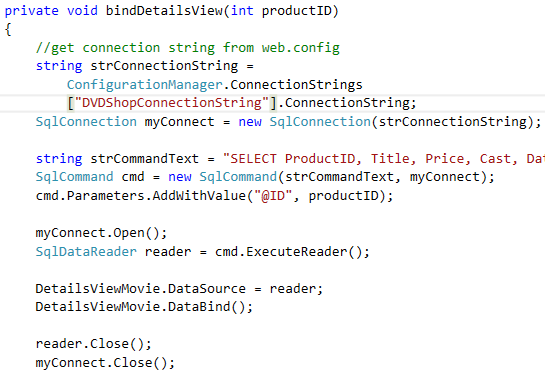
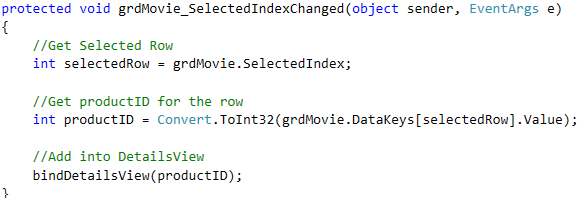
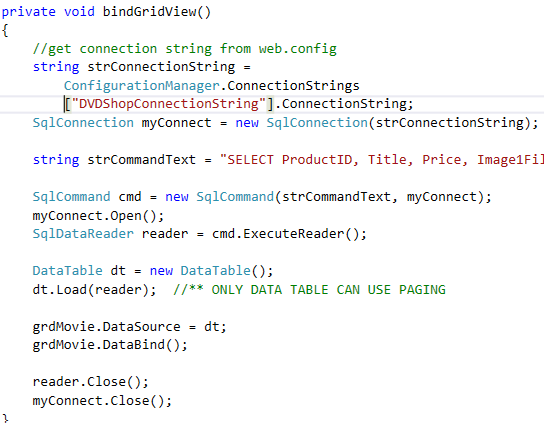
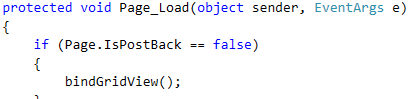
**EX1.aspx.cs**



**EX2.aspx.cs**



**EX3.aspx.cs**



**EX4.aspx.cs**

