

## C# First Assignment

### Background

You are going to implement the menu interface of a DVR (Digital Video Recorder). The interface must be developed using C# and Windows forms.

A TV Program has a title, eg "Doctor Who". A TV program typically (but not always) has an episode title, eg "The Pandorica Opens". A TV program typically (but not always) has a description of the episode, eg "The Doctor is trapped and Amy is dead". A TV program has a channel that it is broadcast on, eg ABC1. A TV program has a broadcast date and time, eg Monday 11/7/2010 7:30 PM. A TV program has a duration, eg 45 minutes.

On a digital video recorder (eg TiVO) program information can be stored in a TV Guide. The guide contains all the programs that are on each channel for a given day. Program information can also be stored in a To Do List. This is a list of programs for a given date ordered by time of programs that the DVR will record. Program information can also be stored in a Season Pass Manager. This list stores an ordered priority list of TV Series that the owner has indicated they would like recorded no matter what time or channel they are on (this saves the owner putting them on the To Do List each week, it is done automatically).

### Information

For the purposes of this assignment you will need an enum to represent the Broadcast Stations.

```
public enum BroadcastStation { ABC1, SCTV, TDT }
```

To display dates and times correctly you will need to know the following (some of this information is incorrect in the text book), assume you have a DateTime object called StartTime:

- To display a day in three letter form you use StartTime.ToString( "ddd")
- To display a time in the format 7:30 PM you use StartTime.ToString( "t")
- To display 11/7 you would use StartTime.Day + "/" + StartTime.Month

### Setup the Database

1. Download the file tbl\_tvguide.sql from module 8 or additional resources.
2. Use PHPMyAdmin to access your MySQL account – username is your CIS username, the password is your six digit student number (include leading zeros) unless you have changed it to something else.
  - a. You can access PhpMyAdmin at the following URL:  
In Hobart:  
<http://alacritas.cis.utas.edu.au/phpMyAdmin/>  
In Launceston:  
<http://lawson.cis.utas.edu.au/phpMyAdmin/>

3. In PHPMyAdmin make sure there are tables tbl\_tvguide, tbl\_SeasonPass and tbl\_ToDoList. tbl\_tvguide should have lots of data for three channels over three days. tbl\_SeasonPass and tbl\_ToDoList should be empty (use the Empty button to remove all data). If this isn't the case do the following:
  - a. In PHPMyAdmin go to the import tab and use the choose file button to select the file tbl\_tvguide.sql. Press the go button. This will create you a table called tbl\_tvguide and fill it with data and also two more empty tables tbl\_SeasonPass and tbl\_ToDoList.
4. Within Visual Studio 2008 (after you have begun a project) add a reference to the MySQL.Data.dll – right click in the solution explorer, add reference, choose MySql.Data.
5. When coding and a connection string is required use the following:
 

```
conn = new MySqlConnection( "Database=xxxx;Data
Source=alacritas.cis.utas.edu.au;User Id=xxxxxx;Password=xxxxxx");
```
6. Edit the connection string so it has Database set to your username, User Id is also set to your username, and password has your password value. If you are a Launceston student change alacritas to lawson.

### Getting Started

1. Create a C# Windows Form project called usernameAssignment1 using Visual Studio 2008.
2. Using the form that is currently displayed rename the form to be DVR.
  - a. In the properties window in the lower right, change Name to DVR and change text to DVR. You may or may not need to save the file as DVR.cs.

### Domain Classes

You will need the following classes to complete this assignment, if you have been doing the module exercises you will have most of these already, but some minor changes are required.

1. Create a class called TVProgram, which has properties: Title( string), EpisodeTitle (string), Description (string), Channel(BroadcastStation), StartTime (this is a date and time), Duration (int). These properties can all have the default get and set methods.
2. Create two constructors for TVProgram one that takes all 6 values and one that takes none.
3. Create a ToString method in the TVProgram class this method should return a string in the following format:
  - Title : Episode Title : Description  
Channel Day Date Time Duration  
Day should be just 3 letters, Date should be simply day/month,

Time should be in the format 3:30 PM, Duration is 30 minutes or 2 hours 30 minutes. There is a line break (or new line) between description and channel, use Environment.NewLine to do this.

EG Doctor Who : The Pandorica Opens : Amy is dead  
ABC1 11/7 7:30 PM 45 minutes

- The string should handle empty episode titles and descriptions, there shouldn't be a : if it is not required.
  - Make sure you display the duration correctly so that it handles hours/hour/minutes correctly depending on the number of minutes eg 90 minutes should display 1 hour 30 minutes, 135 minutes should display 2 hours 15 minutes, 45 minutes should simply be 45 minutes, 60 minutes should simply be 1 hour.
4. Within TVProgram create another property called GuideTitle. This property should just have a get method. It should return a string in the following format:  
StartTime Title : EpisodeTitle
    - a. In the above, only display the time part of the StartTime eg 16:10. Also if there is no EpisodeTitle it shouldn't display a trailing :.
  5. Create a class called ToRecordProgram, which extends TVProgram.
  6. Create a class called SeasonProgram, which extends TVProgram. SeasonProgram should have another property Priority (int).
  7. Create a class called TVProgramIndexer. Give it a constructor and an Indexer. The size of the array is stored in field called arrsize and its value is set in the constructor. Create a method within TVProgramIndexer called Length that returns the size of the array.

## DVR Form

As you get each form working you should compile and run the program. Screen shots of all the forms are shown at the end.

The DVR form simply has a menu bar. You will need to handle all the following menu options:

1. File Menu
  - a. Import – reads in an XML file of the TVGuide into the database. This should use an OpenFileDialog form to get the filename. The instructions on how to read XML were contained in the module 8 exercise.
  - b. Export – creates an XML file of the TVGuide from the database. This should use a SaveFileDialog form to get the filename. The instructions on how to generate XML were contained in the module 8 exercise.

- c. Quit – should simply exit the program. Instructions on how to do this were contained in the module 9 exercise.

## 2. View Menu

- a. To Do List – This is a windows form, instructions on how to add a form to a project were in the module 9 exercise.
- b. Season Pass Manager – This is a windows form.
- c. TV Guide – This is a windows form.

Instructions on how to create a form and give it a menu were contained in the Module 9 exercise. You should complete this exercise now if you haven't already.

### TV Guide

The form has a date picker at the top to allow the user to pick the date to display. Underneath that is a list box that shows all the channels available and the user can select one of these. Next to that is another list box (much wider than the first) that displays all the programs on that day on that channel. Underneath that is a text box and if the user selects a program all the known details about that program are displayed in the box. There are three buttons along the bottom: Record – choosing this adds the program to the tbl\_ToDoList in the database, Season Pass – choosing this adds the program title to the tbl\_SeasonPass and also puts all known showings into the tbl\_ToDoList, Close – closes the form and returns the user to the main screen.

1. Instructions on how to implement all the features above were contained in the Module 9 exercise. You should complete this exercise now if you haven't already.
2. You need to make sure that when a person wants to record a program that it is not already on the tbl\_ToDoList.
3. You still need to implement createSeasonPass that is run when the user selects the Season Pass button and here are some hints:
  - a. Check first that the title is not already in the tbl\_SeasonPass, you will need the following sql command to do that:  
select title from tbl\_SeasonPass where title = ?title  
check that there is nothing returned.
  - b. You need to add it to the existing list so you need to find out what the current maximum priority is, you will need the following sql command for this:  
select max(priority) from tbl\_SeasonPass
  - c. You will need to add the title and priority to tbl\_SeasonPass using the following sql command:  
insert into tbl\_SeasonPass (title,priority) values (?title,?priority)
  - d. You will then need to add all the current showings of this that are in the TVGuide to the tbl\_ToDoList. To do this you will first need to

create a list of the programs using the ToRecordProgram class and the TVProgramIndexer class. You will need the following sql to do that:

```
select * from tbl_tvguide where title = ?title
```

You will need to use a MySqlReader to complete this part.

- e. You will then need to iterate through this list of objects and write them to the tbl\_ToDoList using the following sql command:  
insert into tbl\_ToDoList (title, episodetitle, description, channel, starttime, duration) values  
(?title,?episodetitle,?description,?channel,?starttime,?duration)  
This part is similar code to what is already in recordProgram method.

### Season Pass Manager

The form has a list box down the left hand side that displays the titles of all the programs in tbl\_SeasonPass. The list should be displayed in order of priority starting from 0. Next to that are three buttons: Up, Down and Delete. The up button moves a selected program up the list (changes its priority closer to zero). The down button moves a selected program down the list (increases its priority number). The delete button should remove a program from the list and then change the priority of all programs that were lower in the list. There is a Close button on the bottom of the form. Below are some hints (but they are not as detailed as before as you should be able to do more this yourself).

1. Follow similar steps as were used in TV Guide to create a constructor and SeasonPassManager\_Load and closeClick methods. You will need to use the following sql command to create the DataTable table:  
select \* from tbl\_SeasonPass order by priority
2. Set the DataSource for the list box to be the table, you do not have to use LINQ this time. The DisplayMember should be title.
3. As you will be sending commands to the database that affect the table you will need to create a MySqlCommandBuilder in the Load method.
4. Create a down\_Click and up\_Click method. You will need to use the following code to update the database:  
var updateCommand = new MySqlCommand("update tbl\_SeasonPass set  
priority = ?priority where title = ?title", conn);  
updateCommand.Parameters.AddRange(  
new MySqlParameter[] {  
new MySqlParameter() { ParameterName = "?priority", SourceColumn =  
"priority" },  
new MySqlParameter() { ParameterName = "?title", SourceColumn =  
"title", SourceVersion = DataRowVersion.Original } } );  
dataAdapter.UpdateCommand = updateCommand;  
:  
:  
dataAdapter.Update(table);

5. Before the final line you will need to modify the table so that the priorities and titles are swapped over. You can access a table row using the following (assume Programs is the listbox):  
`table.Rows[Programs.SelectedIndex]["priority"]`
6. For both the up and down methods make sure the user is not trying to move something above the top of the list or off the bottom.
7. Create a deleteProgram method. You will need to use following code to delete something from the database:  

```
var deleteCommand = new MySqlCommand("delete from tbl_SeasonPass
where title = ?title", conn);
deleteCommand.Parameters.Add(new
MySQLParameter(){ParameterName = "?title",SourceColumn = "title",});
dataAdapter.DeleteCommand = deleteCommand;
table.Rows[Programs.SelectedIndex].Delete();
```
8. As well as deleting the item from the list you will need to change the priorities of each item lower in the list, using code very similar to what what used in the up method.

### To Do List

The form has a date picker at the top that allows the user to specify a date – make sure you set the default date to 21/7/2010. Below that is a large DataGridView (similar to a table) that displays the data for the entries in the tbl\_ToDoList that correspond to the chosen date. Underneath this are two buttons, Delete – which deletes a row from the table and Close – which closes the window.

1. Follow similar steps as were used in TV Guide to create a constructor and ToDoList\_Load and closeClick methods. You will need to use the following sql command to create the DataTable table:  
`select * from tbl_ToDoList`
2. Set the DataSource for the DataGridView to be the table, you do not have to use LINQ this time.
3. As you will be sending commands to the database that affect the table you will need to create a MySqlCommandBuilder in the Load method.
4. The table should ONLY display the title, starttime, duration and channel. Use the Programs.Columns[1].Visible = false to turn off other columns.
5. For the starttime column it should only display the time in the format "HH:mm". You need to work out how to do this, but have a look at ValueType for a Column and alsoDefaultCellStyle.Format.
6. Create a deleteProgram method. This is similar to what you have done before.

## Submission

1. Modify ALL files so that they are in a namespace. The namespace should be your usernameAssignment1, eg nherbertAssignment1.
2. When the program gives the correct output you should zip up all the project files and submit using MyLO. You must submit all files so that the assessors can run your program AND review all the code. Remember to submit an electronic assignment coversheet (see the Assessment page) available at:  
[http://www.cis.utas.edu.au/cisview/assign\\_cover.jsp](http://www.cis.utas.edu.au/cisview/assign_cover.jsp)
3. You must submit the assignment by 11:55pm on Monday 27<sup>th</sup> September. All late submissions will be assessed according to the School's Late Assessment policy available at:  
<http://www.cis.utas.edu.au/downloads/ExtensionPolicy.pdf>

## Assessment

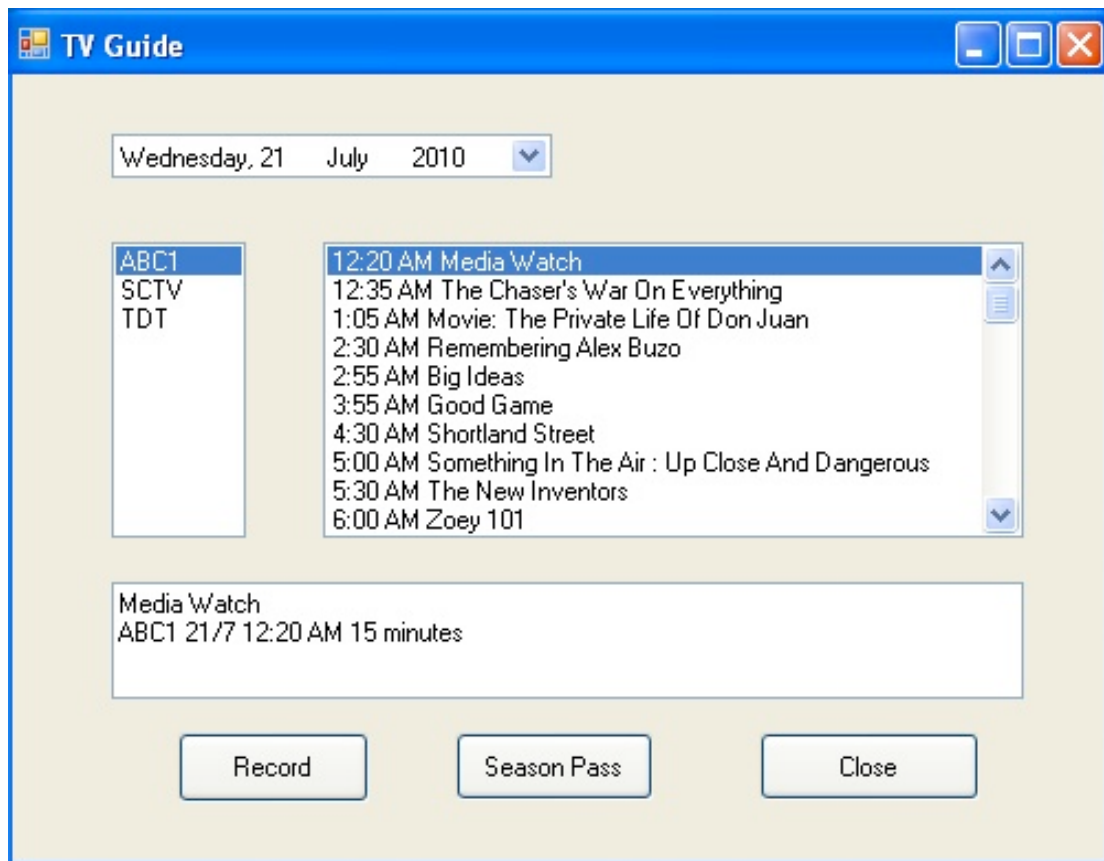
This assignment is worth 15% of your final grade. The assignment will be assessed using Criterion Referenced Assessment. These criteria will be available by August 10<sup>th</sup> on the MyLO site.

## Screen Shots

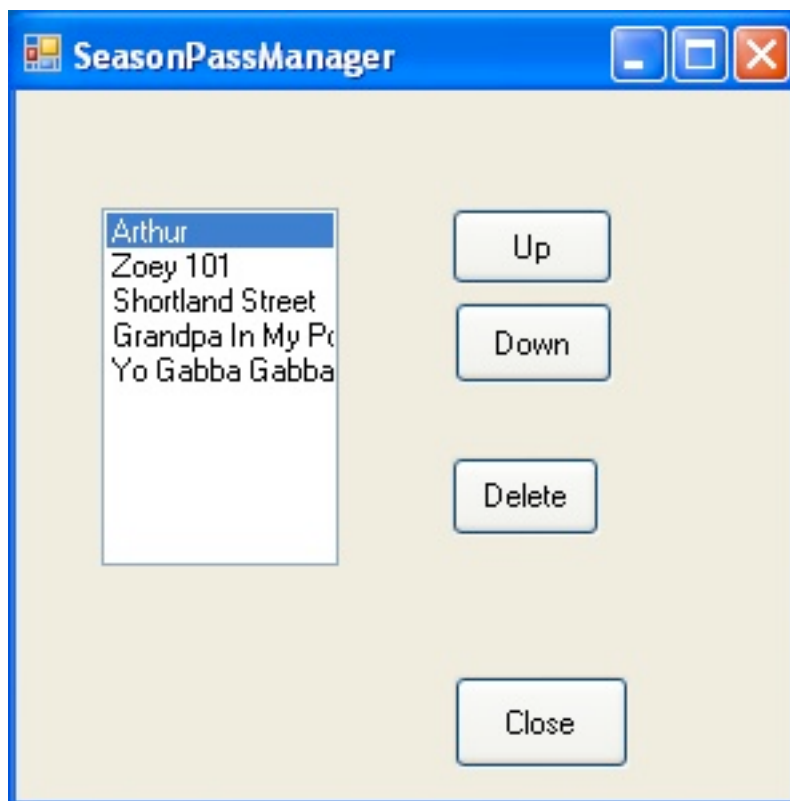
### Main Screen



### TV Guide



### Season Pass Manager





## To Do List

ToDoList

Wednesday, 21 July 2010

	title	channel	starttime	duration
▶	Big Ideas	ABC1	11:00	60
	Toasted Tv	TDT	07:00	5
	Toasted Tv	TDT	07:25	5
	Toasted Tv	TDT	07:55	5
	Toasted Tv	TDT	07:55	5
	Grandpa In My P...	ABC1	08:15	15
	Yo Gabba Gabba!	ABC1	09:00	25
	Yo Gabba Gabba!	ABC1	09:00	25
	The New Inventors	ABC1	05:30	30
	The New Inventors	ABC1	11:30	30
	Shortland Street	ABC1	04:30	30
	Shortland Street	ABC1	04:30	30
	Bananas In Pyja...	ABC1	15:00	5
	Yo Gabba Gabba!	ABC1	09:00	25

Delete

Close