INN570 2013 Week 2 Prac Exercises

Creating Windows Applications in .NET

In this exercise, we will create a windows application in .NET, which will be used throughout the practicals in the following weeks.

1 Set up the interface

Open up Visual Studio, and create a new project, of type Visual C# Windows Application. Name the project *BBookWk2*. In the Properties, you will need to name the main form to be called *BBook* and set its "Text" property as "*Birthday Book*". Place controls onto the form as follows:

- Place a *DataGridView* control under the title bar. In the Properties, name it "*BBookGrid*". Set the "Dock" property to "Top".
- Place a Button control. Name it "*cmdExit*". Set the "Text" property to be "*Exit*", and the "Anchor" property to anchor bottom and right only.
- In the Solution Explorer, rename the Form1.cs file to be "*BBook*", and check that the *Application.Run*() call in *program.cs* also calls *BBook* rather than *Form1*.

You should now be able to run the application by clicking on "Start" and obtain a form as shown in Figure 1.You can observe its behaviour as you resize. The *DataGridView* should fill the width of the form, and the exit button should remain fixed relative to the bottom right corner of the form when resizing the form.

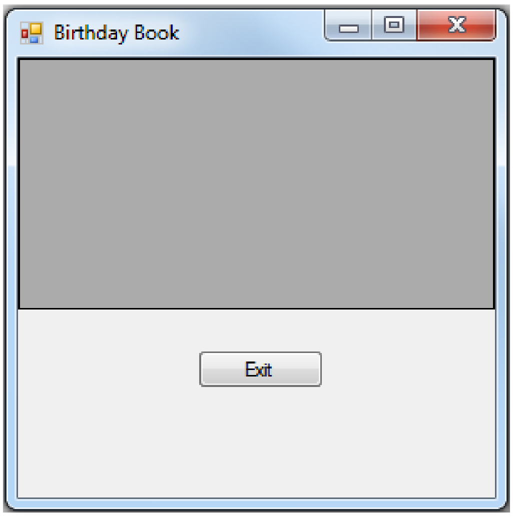
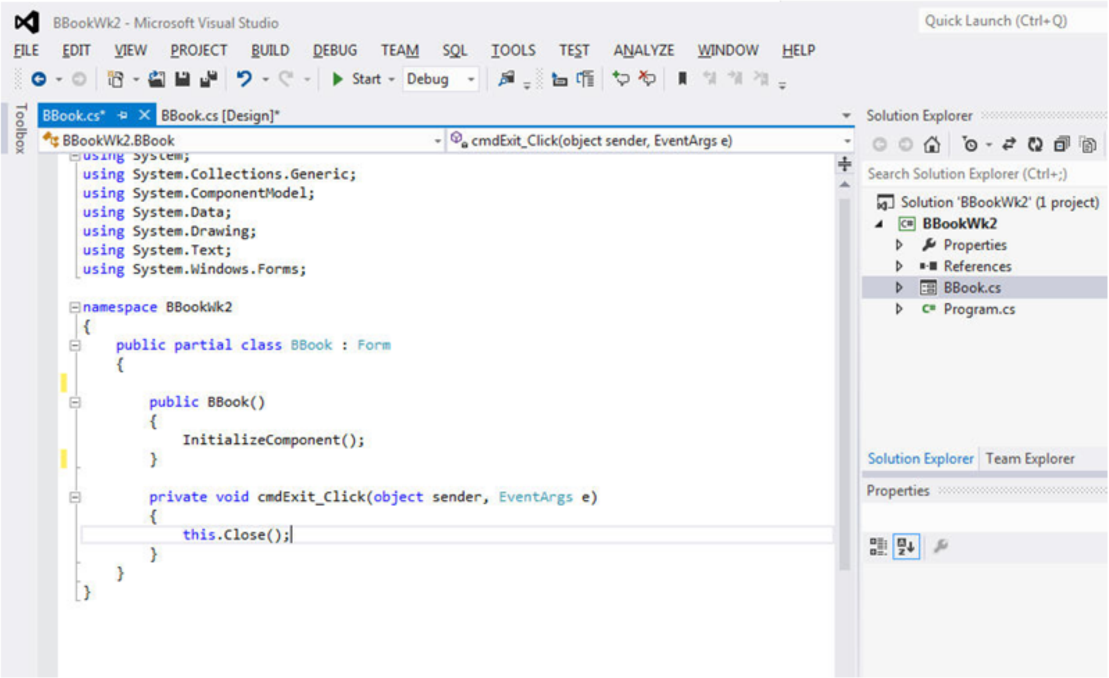


Figure 1: The Birthday Book Form

2 Set up some code

In the Design view, double click on the exit button. This will put you in code view, and entering code for an event handler. Add the code: "*this.Close();*" in the event handler `cmdExit_Click`. If you run the application, you should now find that the Exit button works. See the screenshot of Figure 2.



BBookWk2 - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ANALYZE WINDOW HELP
Start
Debug
Toolbox
BBook.cs*
BBook.cs [Design]*
BBookWk2.BBook
cmdExit_Click(object sender, EventArgs e)
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'BBookWk2' (1 project)
BBookWk2
Properties
References
BBook.cs
Program.cs
Solution Explorer
Team Explorer
Properties

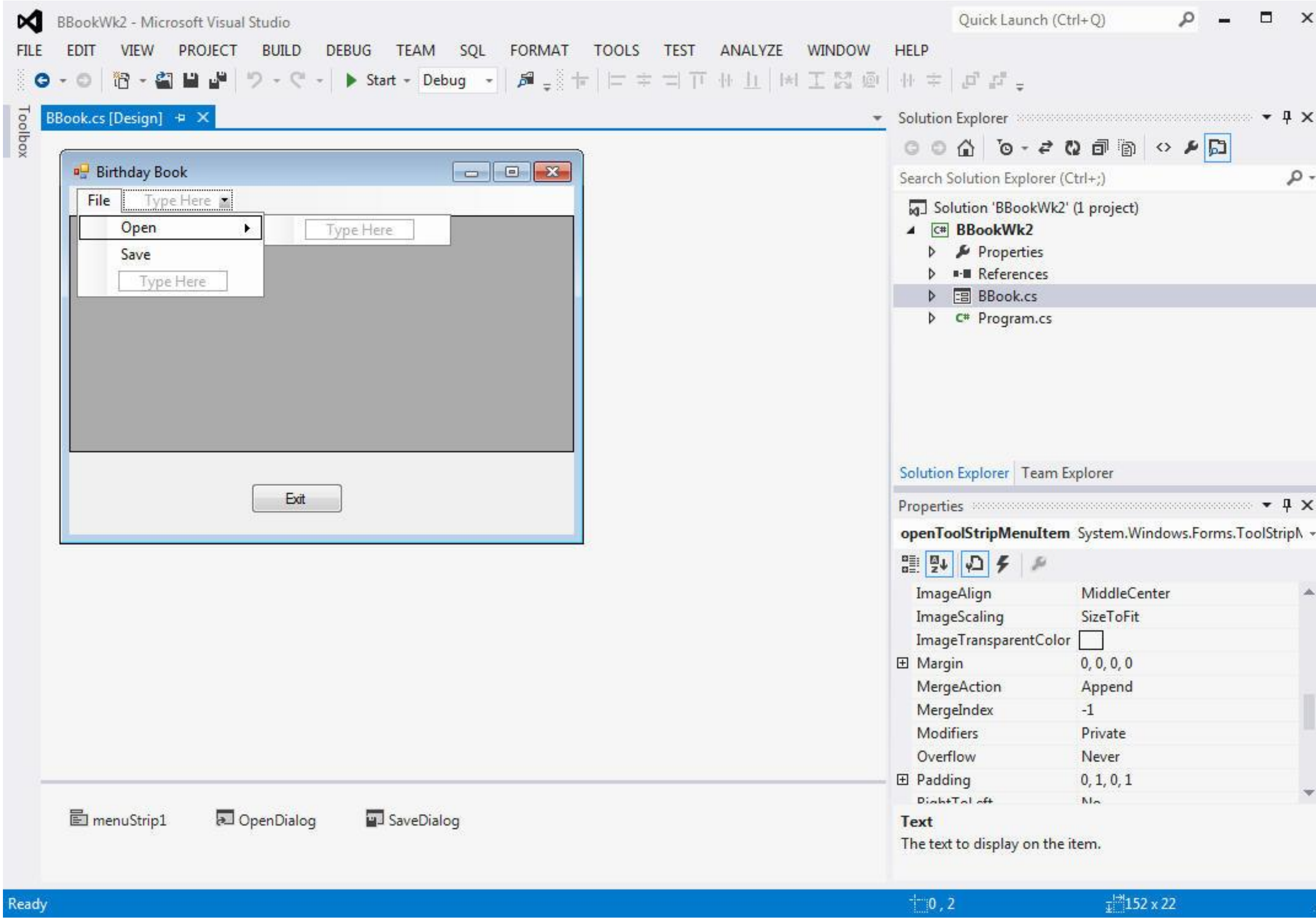
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace BBookWk2
{
    public partial class BBook : Form
    {

        public BBook()
        {
            InitializeComponent();
        }

        private void cmdExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

Figure 2: The event handler for the exit button

3 *Menus and Dialogs*

Add more controls to the form. The Visual Studio Toolbox also has a dialogs section. As with the standard controls, drop these onto the form, although they will appear in a panel below the form itself.

- Add an *OpenFileDialog*, and name it *OpenDialog*.
- Add a *SaveFileDialog* and name it *SaveDialog*.
- Add a *MenuStrip*. You will find that you can type in names for the menu items very easily. Provide one menu called "File", with submenus called "Open", and "Save", as shown in Figure 3.
- Visual Studio uses a standard naming convention for these submenus, so we will preserve this. For example, the name of the "Save" submenu becomes saveToolStripMenuItem.



openToolStripMenuItem System.Windows.Forms.ToolStripM

ImageAlign	MiddleCenter
ImageScaling	SizeToFit
ImageTransparentColor	□
⊞ Margin	0, 0, 0, 0
MergeAction	Append
MergeIndex	-1
Modifiers	Private
Overflow	Never
⊞ Padding	0, 1, 0, 1

Text
The text to display on the item.

Figure 3: The menu creation process

4 *Adding code for the DataTable*

In the Design view, double click on the form. This will put you in code of the form. Add the following declarations in the class *BBook*.

```
using System.IO;

.. . .
public partial class BBook : Form
{
      const string TableName = "Birthdays";
      const string NameCol = "Name";
      const string DateCol = "Birthday";
      const int GridLineWidth = 1;
      const int NumColumns = 2;              // Number of columns
      const string Separator = ";";          // Field separator


      private DataTable m_BBookTable;      //stores the birthdays
.. ..
```

This basically establishes some useful constants, and a *DataTable* variable which will hold the birthday data. Now we must add initialisation code to set up the grid and the table when the application starts. In the design view, select the Events view in the properties window, and examine the "Load" event. Double-clicking on the event name will yield the event handling code in the code view, similarly to the exit code.

Now, in the method *`BBook_Load (object sender, EventArgs e),`* add the following code:

```
private void BBook_Load(object sender, EventArgs e)
{
            //Initialize everything on startup
            //Create a data table with two columns
            this.m_BBookTable = new DataTable(TableName);
            this.m_BBookTable.Columns.Add(new DataColumn(NameCol,
Type.GetType("System.String")));
            this.m_BBookTable.Columns.Add(new DataColumn(DateCol,
Type.GetType("System.DateTime")));
            BBookGrid.DataSource = m_BBookTable;

            //Set up a better looking table
            BBookGrid.Columns[NameCol].Width = (BBookGrid.Width -
BBookGrid.RowHeadersWidth - 2 * GridLineWidth) / NumColumns
                  - GridLineWidth;
            BBookGrid.Columns[DateCol].Width =
BBookGrid.Columns[NameCol].Width;

}
```

You should now be able to run the application, and enter names and birthdays. You can use the help system to gain some clues as to the meaning of the code.

Further work

In the following weeks, we will work further on this application to insert handlers for the Open and Save Dialogs. See what difference it makes to run this application in different culture settings of the system. There are many ways in which this simple application could be improved or extended. If you have time, try to think of some, and see if you can implement them.