INN570 2013 Week 8 Prac Exercises

Switching the UI culture at run time.

The objective of this practical exercise is to allow the user to switch the language of the interface without terminating the application. It is not necessary for an internationalized application to have this capacity; but learning how it is done helps you understand the way resources are used.

1. A Birthday Book with Culture Selection

Download the *Birthday Book* application as it is packaged for the exercise. This uses the *SelectCulture* form with a simple Birthday book. However, it is a culture neutral form and the *SelectCulture* form is now called from inside the *Birthday Book* form, there being an additional button "*Set Culture*" to allow this to happen.

You need to adapt this program to load strings for the appropriate culture, but without using the form's "Localizable" property, which will handle things automatically.

• *Explore the application*

Just verify for yourself that the program works by running the application. Have a look at what happens when you press the *Set Culture* button. The program calls the *SelectCulture* dialog form and then loads a resource file. This is done by two new methods in class *BBook*. Double click on the form *Birthday Book* to open its corresponding code, you will find two new methods, *cmdSelect_Click* and *LoadUIStrings*. *cmdSelect_Clic*k is used to open the Select Culture form, and *LoadUIStrings* is used to load resources files. Two resources files are provided in the package. The current resources files contain only the text strings for two components on the *Birthday Book* form, button *Exit* and label *File*. The *SelectCulture* dialog has the effect of setting this text property. You will recall that the *UICulture* is the critical setting, as this one governs resource searches. Run the application and try selecting the following UI Culture:

1. *de-LU German* (*Luxembourg*) or any *de* culture
2. *bg-BG Bulgarian* (*Bulgaria*) or any other UI culture

Check the appearance of the Birthday Book form, explain the differences you observe. It may help you to look at the contents of the resource files *UIStrings.resx* and *UIStrings.de.resx*. Just double click on one of the resource files, and it will open up a table.

2. Modify the Birthday Book with Culture Selection

• *Add new strings in resources files*

You can edit resource files from inside Visual Studio. Just double click on one of the resource files, and it will open up a table where you can enter your resources.

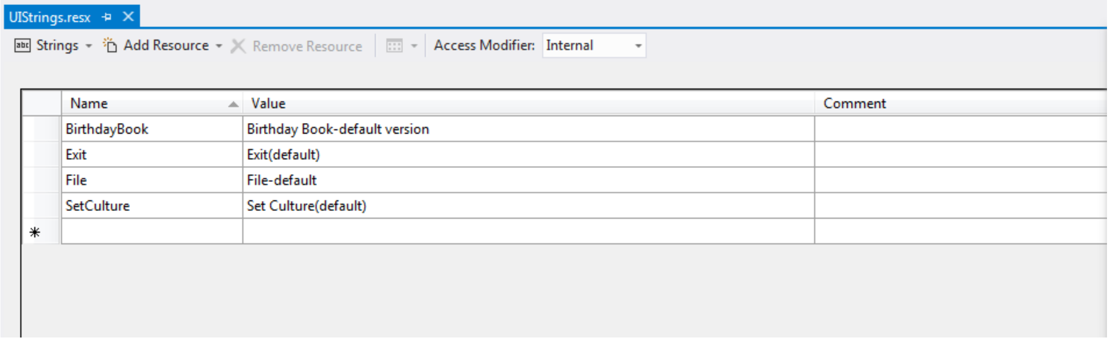
- o Add "*BirthdayBook*" and "*Birthday Book – default version*" as name and value respectively in a new row in the default resource file *UIStrings.resx*.

- Add “*BirthdayBook*” and “*Birthday Book – German version*” as name and value respectively in a new row in the German resource file *UIStrings.de.resx*.
- Add the following line in method `LoadUIStrings()`

```
this.Text = res.GetString("BirthdayBook");
```

Run the application again and observe the difference.

You can add more strings to the resources files for other components in the form *Birthday Book*, for example, the *Set Culture* button. You need to modify the method `LoadUIStrings()` accordingly (the name of the ‘*Set Culture’* button is ‘*cmdSelect*’, you need to change the ‘*Text*’ property of the button). The *UIStrings.resx* file may look like the following table after adding a new string “*SetCulture*” as Name and “*Set Culture(default)*” as Value.

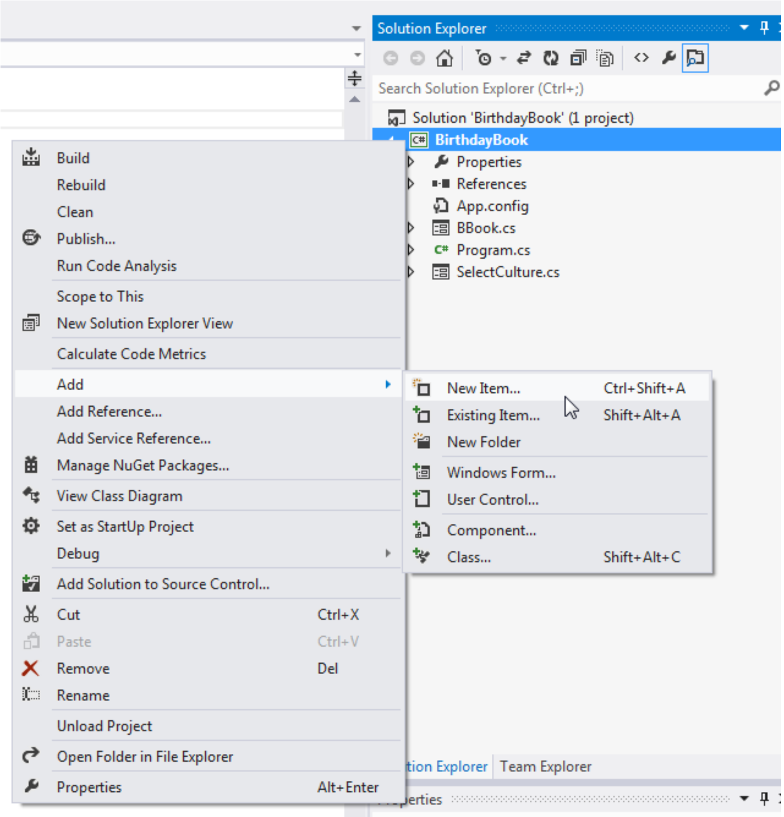


Name	Value	Comment
BirthdayBook	Birthday Book-default version	
Exit	Exit(default)	
File	File-default	
SetCulture	Set Culture(default)	

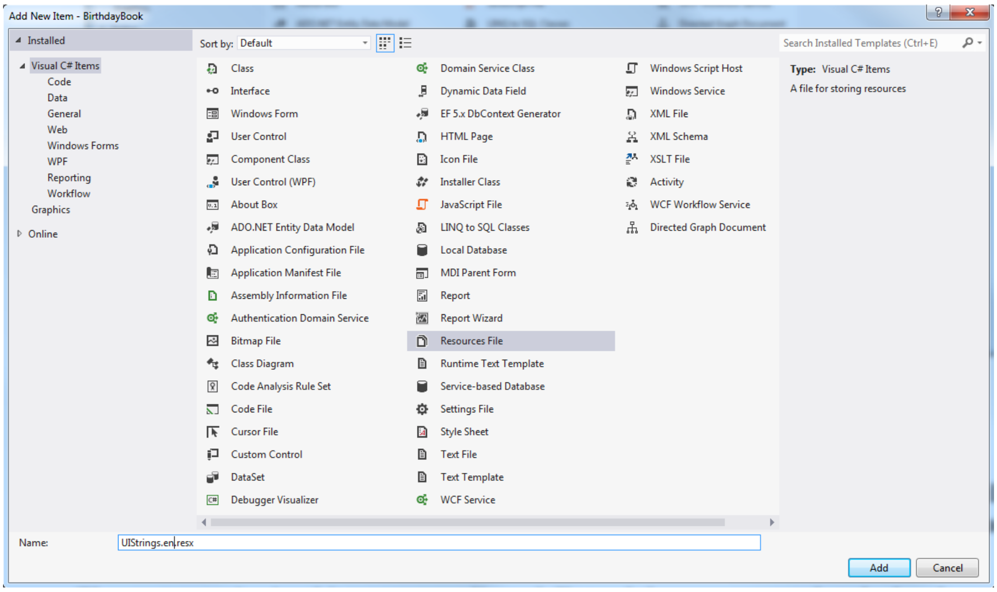
- ***Add new resources files***

You can add a new resource file simply by using "*Add New Item*" from the Project menu.

- Add a new culture for English. You need to add a resource file called *UIStrings.en.resx*. You can do this by right clicking on the *BirthdayBook* project in Solution Explorer, and adding an assembly resource file. Here is an image of the menu.



Here is how you select the new resources file from the menu:



Type: Visual C# Items

A file for storing resources

Name: UIStrings.en.resx

Name the the resource file as *UIStrings.en.resx*.

- Add the following strings in the resources file, you can add more if you wish.

Name	**Value**
File	*File -English*
Exit	*Exit(en)*
BirthdayBook	*Birthday Book-English version*

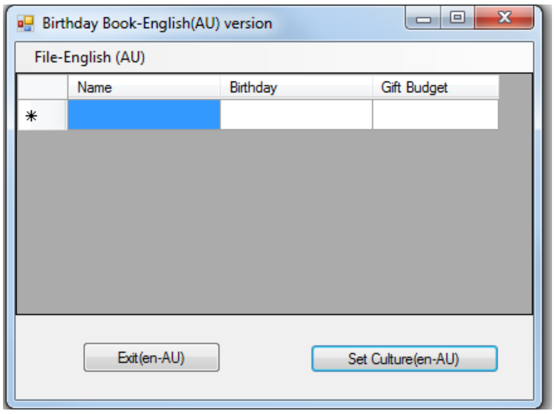
SetCulture	*Set Culture(en)*

Run the application and select culture "en-English", observe the result.

- ***Add another resource file***

Try adding a resource file for Australia (en-AU).

When the Australian culture is selected for the UI culture, you should make the *BBook* form like different from the neutral en resource. You may use a more specific string to indicate the en-AU culture, e.g., the following one.



- ***Current Culture for Formats***

This application allows you to set the current culture for date/time/currency formats (*Select Cultu*re) as well as for user interface culture (*Select UI Culture*). When you open a file, the current culture will be determined by the culture in the file. If you want a different culture for the formats, you need to re-set the current culture using *Select Culture*.

- Run the program and open file *birthdayBook-US.txt*. The date/time/currency format is English-US format.
- Select a current culture (not UI Culture) using *Select Cultu*re, for example, en-AU English, then when you click on the entries on the form, the date/time/currency format will be changed to English-AU format. When you save the form, the culture in the saved file will be en-AU.