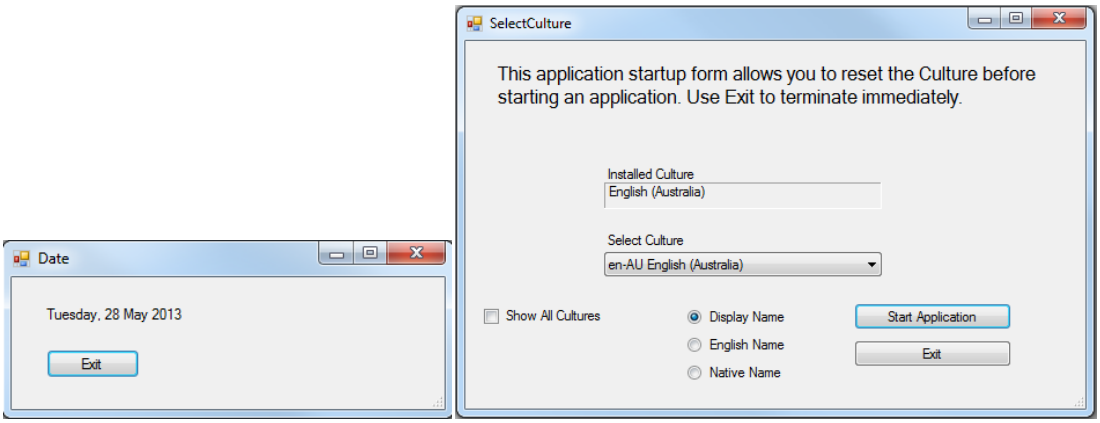
INN570 2013 Week 6 Prac Exercises

Locale Awareness in .NET

The objective of this practical exercise is to show how an application can be made locale aware. You will work from two application programs, *Date* and *BirthdayBook*, placed on Blackboard.

1. A Date Application with Culture Selection

Download the *Date* application as it is packaged for the exercise. The supplied application includes two forms as shown below. A form with title *Date* is used to show dates, the other form with title *SelectCulture* allows a user to select cultures for a thread.



The application runs a loop, repeatedly showing the culture selection form and then the date form when the user clicks on the Start Application button. The loop is terminated when the user clicks on Exit button from the culture selection form. This allows you to run a form repeatedly with different cultures set for the thread, to see how it behaves.

2. Explore the Culture selection dialog

The culture selection dialog features drop down lists of cultures which can be selected.

There is a check box *Show All Cultures*, which determines whether these drop down lists include all known cultures, or just those which are installed in your system.

There are option boxes, which let the drop down lists be shown in English in the default culture of the system, or in the native tongue of each culture individually.

Observe the list of all cultures, displayed in their native name.

Run the application, select some different cultures, observe the format of the displayed date, for example, observe the difference in date format when choosing Australia English(en-AU) and United States English (en-US).

3. A Birthday Book Application with Culture Selection

Download the other application *BirthdayBook* as it is packaged for the exercise. The supplied application also includes two forms. One form with title *Birthday Book* has

been used in previous workshop classes, the other form is the *SelectCulture* form as shown above.

Run the application and select English (US) as the Culture, in the Birthday book, add a person called "Mai" who was born on May 7, 1990, and another person called "Jules" who was born on July 5, 1998, save the birthday book somewhere with a file name as *birthdayBook_US.txt*, then exit from the birthday form. Since the selected culture is English (US), the dates are in the format of month/day/year.

Run the application again with the Culture set to English (AU), and open the birthday book *birthdayBook_US.txt* saved in the first step. You will find that the format of the date is still month/day/year even English(AU) was chosen.

Add another person, John born on December 25, 2001 into the birthday book, the date will be displayed as 25/12/2001.

Why are the dates incorrect?

If the birthday book files are intended to be shared between users from different cultures, how could you fix this problem?

Birthday book information should be saved in a culture independent binary form. However, this would mean that the files are no longer usable by other text based applications such as Notebook.

Extend the Birthday Book so that it saves extra information in the text file to help resolve the problem.

One approach is to store a culture used to write the file. This must be used when you read the file; but note that you must display the information in the culture selected by the user, not the culture of the file.

Open the code *BBook.cs* by double clicking on the form.

- In method *saveToolStripMenuItem_Click()*, add the following two lines of code after the declaration statement of *StreamWriter MyStream*. These two lines of code will write the culture name at the beginning of the birthday book file saved by this application.

```
String cultureName = Thread.CurrentThread.CurrentCulture.Name;
MyStream.WriteLine(cultureName);
```

- In method *openToolStripMenuItem_Click()*, add the following two lines of code after the declaration statement of *StreamReader MyStream*, to create a CultureInfo object.

```
String cultureName = MyStream.ReadLine();
CultureInfo format = new CultureInfo(cultureName,true);
```

- Still in method *openToolStripMenuItem_Click()*, use the following line to replace `fields[1].Trim()` in the while-loop, to parse all date strings using the parse method of the DateTime type.

```
DateTime.Parse(fields[1].Trim(), format, DateTimeStyles.NoCurrentDateDefault)
```

Repeat the experiment by choosing English (US) as the Culture and re-entering 'Mai' and 'May 7 1990', 'Jules' and 'July 5 1998', then save to a new file *NewbirthdayBook-US*. Open the file *newbirthdayBook-US* using Notepad, you will see the culture name 'en-US' at the top of the file.

Run the application again with the interface set to English (AU), and open the file *newbirthdayBook-US.txt*. You will find that the format of the date becomes day/month/year. Again, add another person, John born on December 25, 2001 into the birthday book, the date will be displayed as 25/12/2001. Save the file to *birthdayBook-AUS*.

Then, select English (US) as the culture, open file *birthdayBook-AU*, you will find that the dates are displayed in the format of month/day/year.

4. How much will you budget to spend on their gift?

Now we want to add one more column in the birthday book table to display your budget for birthday gift. Open the code *BBook.cs* by double clicking on the form.

- At the beginning of class *BBook*, change the value of constant variable *NumColumns* from 2 to 3, then after the variable declaration '*private DataTable m_BBookTable;*' , add the following declaration to declare a constant variable *BudgetCol*:

```
const string BudgetCol = "Cost";
```

- In method *BBook_load*, add the following code at the end of the method to create a new column in the table:

```
this.m_BBookTable.Columns.Add(new DataColumn(BudgetCol,
                              Type.GetType("System.Decimal")));
DataGridViewCellStyle cs = new DataGridViewCellStyle();
cs.Format = "C2";

BBookGrid.Columns[BudgetCol].DefaultCellStyle = cs;
```

- In method *openToolStripMenuItem_Click*, in the nested *if* statement inside the *while* loop, add the following lines at the end of the inner *if* statement:

```
NumberStyles style = NumberStyles.Number | NumberStyles.AllowCurrencySymbol;

m_BBookTable.Rows[m_BBookTable.Rows.Count-1][BudgetCol] =
Decimal.Parse(fields[2].Trim(), style,
                              CultureInfo.CreateSpecificCulture(cultureName));

DataGridViewCellStyle cs = new DataGridViewCellStyle();
cs.Format = "C2";
BBookGrid.Columns[BudgetCol].DefaultCellStyle = cs;
```

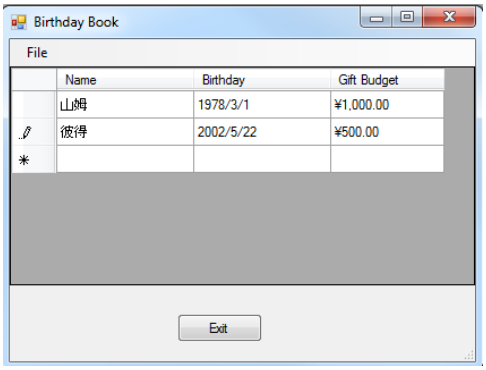
- In method *saveToolStripMenuItem_Click*, in the *foreach* statement, replace

```
MyStream.WriteLine(MyRow[NameCol].ToString()+Separator+
MyRow[DateCol].ToString());
```

with the following statement:

```
MyStream.WriteLine(MyRow[NameCol].ToString()+ Separator + MyRow[DateCol].ToString()
+     Separator     +     Decimal.Parse(MyRow[BudgetCol].ToString()).ToString("C2",
CultureInfo.CreateSpecificCulture(cultureName)));
```

Run the application again with the Culture set to Chinese (Simplified, PRC), you will see that the birthday table contains three columns. Choose the simplified Chinese IME, enter： 山(shan)姆(mu), March 1 1978, 1000; 彼(bi)得(de), May 22 2002, 500. The birthday book interface will look like below:



Save it as *birthdayBookBudget-zh*. You can find the currency symbol in *BirthdayBookbudget-zh.txt* and the date format are both correct. Then, run the application again with the Culture set to Chinese (Simplified, PRC), open the file *BirthdayBookbudget-zh.txt*, you can find that the dates and currency are displayed correctly.

Now, run the program with the Culture set to English (Australia) and open the file *BirthdayBookbudget-zh.txt*, you will find that the date format and the currency symbol are all correct, but the names are still in Chinese. This is because this application does not translate the names from one language to another. It simply reads from the input file and displays the text onto the form Birthday Book. We will come back to the translation issue in a later prac.