

Course 2957B Module 2 Lab – additional notes

This lab provides you with a code segment to use to populate a combo box with a sorted list of culture names. It uses *Array.Sort* to sort the array of *CultureInfo* objects returned by *CultureInfo.GetCultures*.

```
private void PopulateCultureList()
{
    // Get the list of available cultures.
    CultureInfo[] cultures = CultureInfo.GetCultures
        (CultureTypes.SpecificCultures);

    // The list needs to be sorted to be usable.
    Array.Sort<CultureInfo>(cultures, new Comparison<CultureInfo>
        (delegate(CultureInfo x, CultureInfo y)
        {
            return string.Compare(x.DisplayName, y.DisplayName,
                StringComparison.CurrentCulture);
        }));

    // Populate the combobox with all the available cultures.
    this.CulturesComboBox.Items.AddRange(cultures);

    // Set to the current UI culture.
    this.CulturesComboBox.SelectedItem =
        Thread.CurrentThread.CurrentUICulture;
}
```

The second parameter in *Array.Sort* is an interesting example of the use on an **anonymous delegate**. This overload of *Array.Sort* takes two parameters – the array to sort and a *Comparison<CultureInfo>* instance. *Comparison<T>* is a delegate, as shown in the MSDN documentation:

```
generic<typename T>  
public delegate int Comparison( T x, T y)
```

A *Comparison* delegate can reference a method with the signature

```
int method(T, T)
```

So a *Comparison<CultureInfo>* delegate can reference a method with the signature

```
int method(CultureInfo, CultureInfo)
```

To illustrate how the delegate is working, the code above could be written to use a named delegate, the *CompareCultures* method, which has the appropriate signature, in the following version of the code.

```
private void PopulateCultureList()
{
    // Get the list of available cultures.
    CultureInfo[] cultures = CultureInfo.GetCultures
```

```

        (CultureTypes.SpecificCultures);

// The list needs to be sorted to be usable.
Array.Sort<CultureInfo>(cultures, new
    Comparison<CultureInfo>(CompareCultures));

// Populate the combobox with all the available cultures.
this.CulturesComboBox.Items.AddRange(cultures);

// Set to the current UI culture.
this.CulturesComboBox.SelectedItem =
    Thread.CurrentThread.CurrentUICulture;
}

private int CompareCultures(CultureInfo x, CultureInfo y)
{
    return string.Compare(x.DisplayName, y.DisplayName,
        StringComparison.CurrentCulture);
}

```

Note that the shorter version of the syntax for instantiating a delegate could be used instead:

```

// The list needs to be sorted to be usable.
Array.Sort<CultureInfo>(cultures, CompareCultures);

```

The original version has the code for the comparison method written **inline** where the delegate is instantiated. The delegate method does not need to be named as it does not need to refer to a method defined elsewhere, so it is an **anonymous delegate**, indicated by the *delegate* key word.

Sadly, Visual Basic does not allow anonymous delegates, so the VB version of the code uses a named delegate.