**Lab 0831**

The concordance data you built previously will be more useful if it is readily accessible. To make it accessible you are to build a simple Windows Presentation Foundation application that starts the concordance process in the same Windows process and leaves the data accessible after it has been assembled.

You will use the same input files as before, but there will be no file output.

The WPF application is to be a single screen with controls that will display a list of words for selection (single selection), provide for a choice of either word count and usage count ranking or a usage location list. The output must be displayed in the screen, in a legible form, with recognition that some location lists will be lengthy.

You should

1. Add a new WPF App (.Net Framework) project to the existing Solution.
2. Copy the three paths from the debug properties of the Concordance project into your new WPF project.
3. Set the WPF project as the startup project.
4. Change the namespace of the Concordance project and its code to TLG.Concordance.
5. Change the output type of the Concordance project to Class Library.
6. Change the Project Dependencies of the solution so Concordance builds first.
7. Add an event handler to the top-level WPF application (App.xaml) to catch the startup event. You must change both App.xaml and App.xaml.cs.  
     
    public partial class App : Application  
    {  
    private void Application\_Startup(object sender, StartupEventArgs e)  
    {  
    )  
    }  
     
   <Application x:Class="ConcordanceDisplay.App"  
    xmlns=<http://schemas.microsoft.com/winfx/2006/xaml/presentation>  
    xmlns:x=<http://schemas.microsoft.com/winfx/2006/xaml>  
    xmlns:local="clr-namespace:ConcordanceDisplay"  
    StartupUri="MainWindow.xaml"  
    Startup="Application\_Startup">  
    <Application.Resources>  
    </Application.Resources>  
   </Application>
8. Transfer the variable definitions and the methods of the top-level application in Concordance to the appropriate spots in the Application\_Startup event handler. Remember to account for the difference between Main() and an event handler.
9. Convert the error messaging in the transferred code to throw appropriate exceptions rather than using Console.WriteLine(). There is no console associated with the WPF application.
10. Add some code to the application startup event handler  
      
     anlz = new Analyzer();  
     this.Properties["Analyzer"] = anlz;
11. Add some code to MainWindow.xaml.cs  
      
     public partial class MainWindow : Window  
     {  
     internal TLG.Concordance.Analyzer analyzer;  
      
     public MainWindow()  
     {  
     analyzer = (TLG.Concordance.Analyzer)  
     Application.Current.Properties["Analyzer"];  
     InitializeComponent();  
     }  
     }  
    }