Using Solid Framework in an Excel Spreadsheet

Understanding the Power of Solid Framework

VBA can connect to external DLLS via COM

- But the .Net SolidFramework DLL is not COM enabled

- Therefore need to create a COM enabled wrapper and expose the methods that are required

This is documentation of a proof of concept

- The sample demonstrates how to expose the single method that gets the language used within a string

- Solid Framework uses this method internally during OCR and document reconstruction, but it is publicly exposed, so is available for use elsewhere

Create a new DLL that wraps the methods that need to be exposed

```csharp
[ClassInterface(ClassInterfaceType.AutoDual)]
 public class SolidFrameworkComWrapper
    {
        public void Initialise()
        {
            SolidFramework.License.Import(@"[path to license]\license.xml");
        }


        public string GetLanguage(string textValue)
        {
            Initialise();
            var language = new SolidFramework.Language.Language();
            var twoLetterLanguageCode = language.Detect(textValue);
            return twoLetterLanguageCode;
        }
    }
```

Key Points

A license is required so this should be imported when the helper method is called
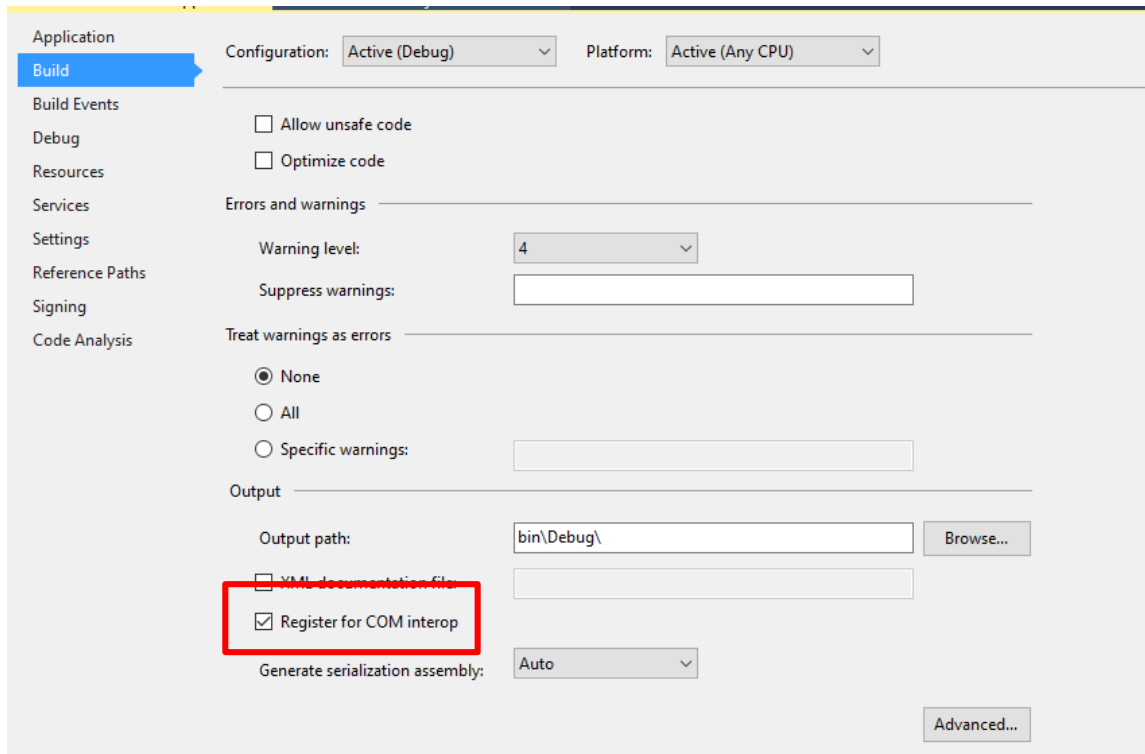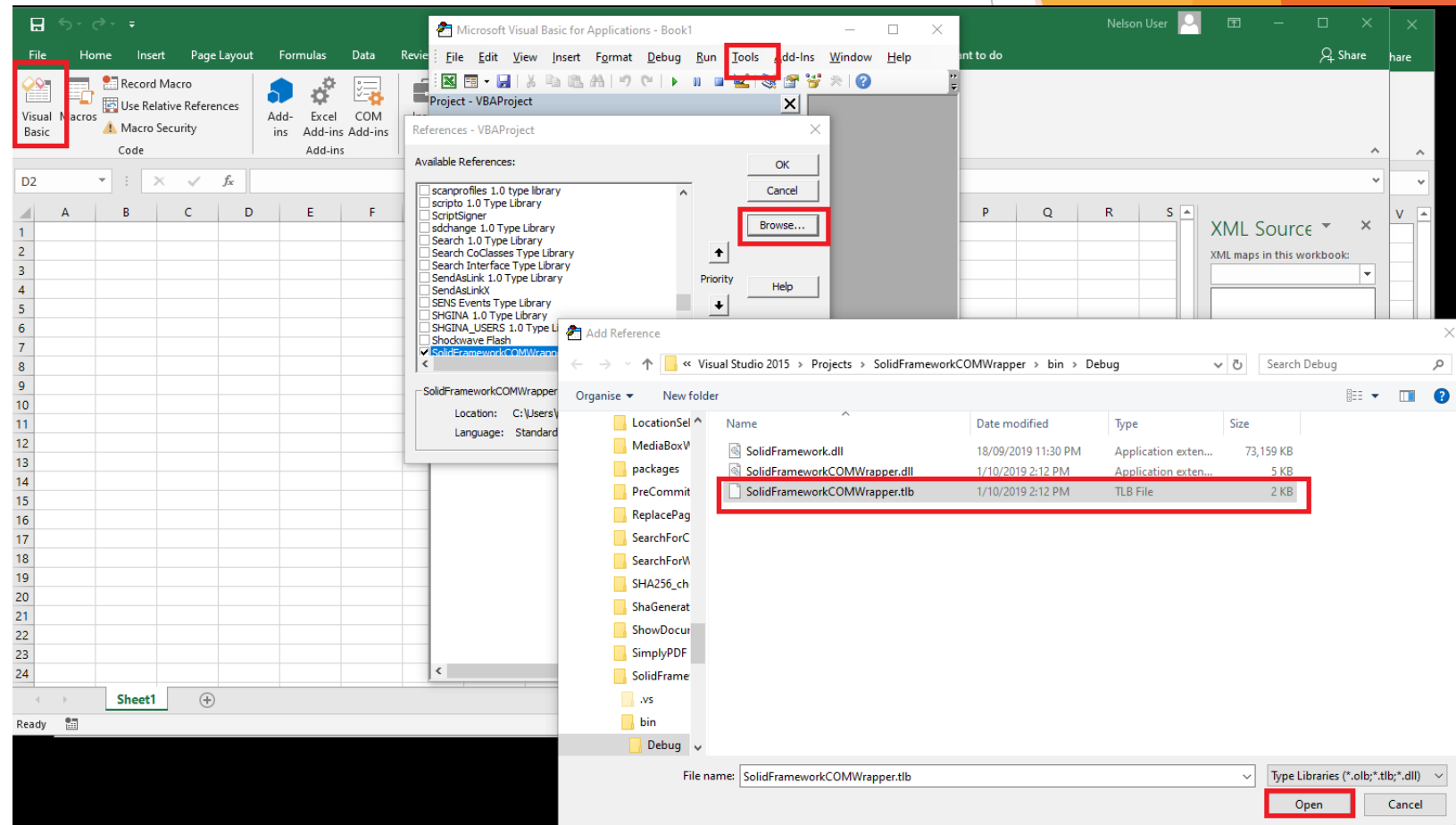

The DLL must be registered for COM Interop

Registering the generated DLL requires elevated privileges.
The easiest way to do this is to build it in Visual Studio which was started under an Administrator account
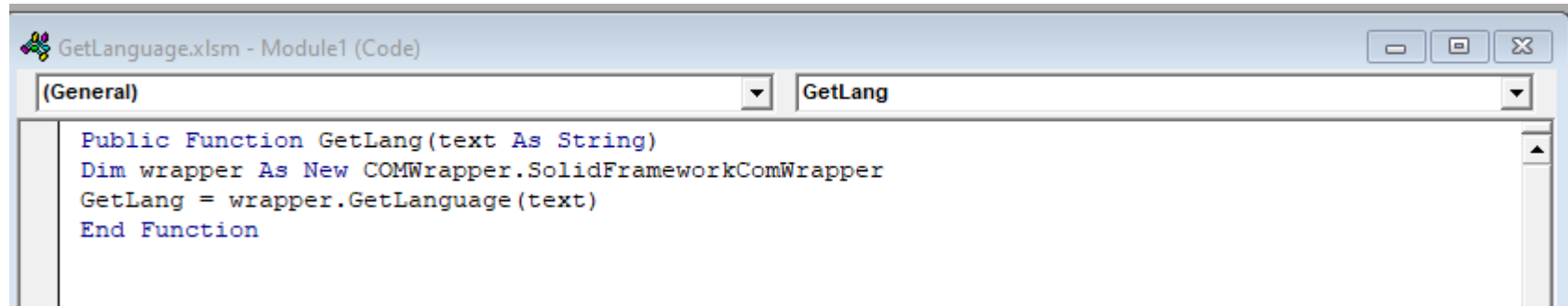
Working in Excel

▶ In the Developer tab open the Visual Basic Editor. Add a reference to the COM .tlb file via Tools
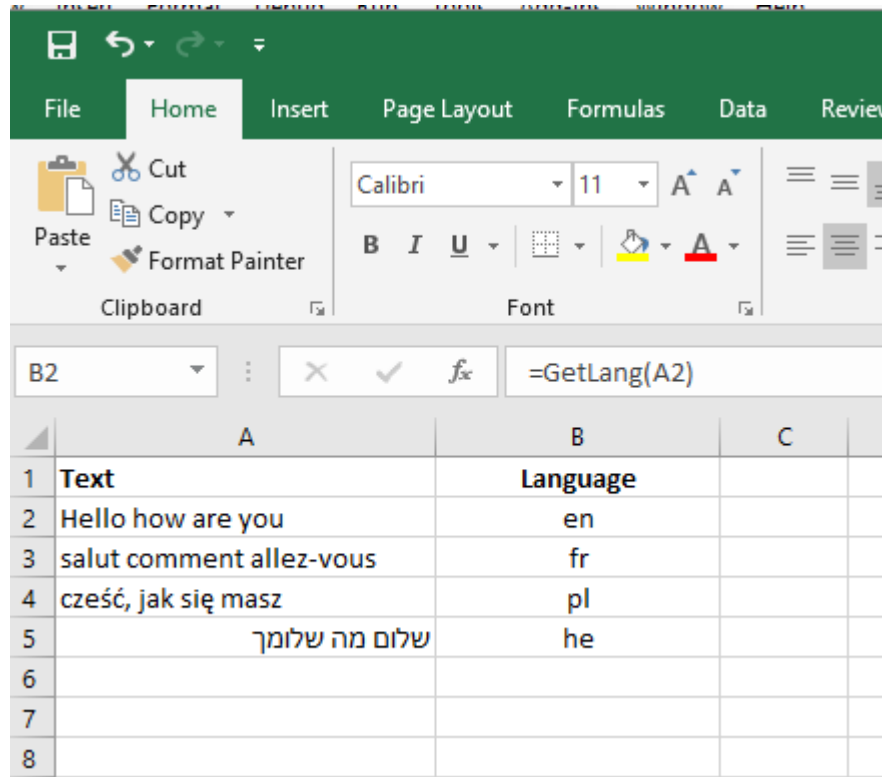
Create a new function in Excel

The function needs to create an instance of our wrapper object, then return the value from the method.


```
GetLanguage.xlsm - Module1 (Code)

(General)                                          GetLang

Public Function GetLang(text As String)
Dim wrapper As New COMWrapper.SolidFrameworkComWrapper
GetLang = wrapper.GetLanguage(text)
End Function
```

Note that Intellisense may not show unless the .Net class has the [ClassInterface(ClassInterfaceType.AutoDual)] attribute

This method can then be used in a worksheet

In this case the language that is detected within each cell is shown in the adjacent cell