**TemperatureConverter\_SeanMcCann\_20150519**

**GitHub:**

To the reviewer(s) of this solution.

Some brief notes:

TemperatureConverter.sln comprises of 2 projects:

TemperatureConverter.csproj

TemperatureConverterDomain.csproj

TemperatureConverter.csproj essentially is the front end and contains the Views and ViewModels that comprise the GUI.

I have adhered to the MVVM pattern which seems to one that has emerged as the winner over that last few years. To this end I opted to use Caliburn Mirco which I have used successfully in the past and it purports to do 90% of what Prism does with 10% of the source code (always a code thing in my view). In terms of one of the SOLID principles (the D) I have used Autofac for Depenceny Injection. (Constructor Injection in this case).

I have added a toplevel style in the App.xaml just to briefly demonstrate that I can use them but clearly it’s a very trivial style in this example.

Where reasonable and sensible I have extracted interfaces. This can be seen in both the ITemperatureConverterCalulator and ILanguageSettingsManager interfaces. There reason for doing this of course is to potentially switch in different concrete implementations and to aid unit testing.

I downloaded Caliburn Micro and Autofac packages using NuGet. The .dlls will restore on a build.

If this is an issue I could upload the packages onto GitHub.

TemperatureConverterDomain.cs proj is effectively where the domain objects (Models) reside.

As usual these are the objects that have the main business logic built into them to try and keep a clear separation from the presentation layer.

I hope you enjoy reviewing my submission and would welcome any feedback.

Kind Regards,

Sean McCann