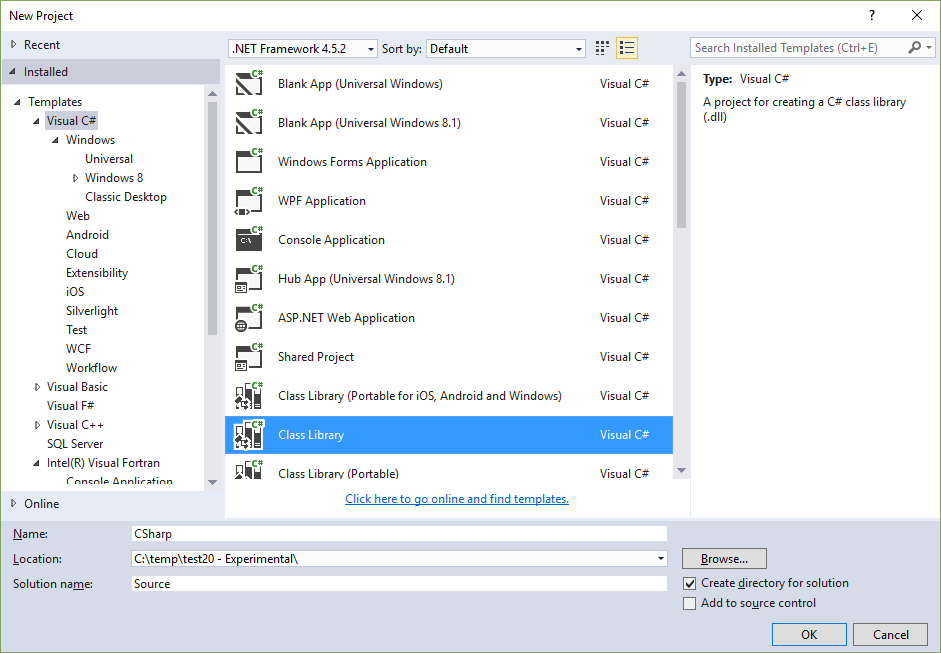
Before starting, see: <https://software.intel.com/en-us/articles/configuring-visual-studio-for-mixed-language-applications>

* “01 Experimental” provides simple routines used to test communications between FORTRAN, C++, C/CLI, and C#.
* “02 FExcel” provides routines designed to read and write data between FORTRAN and Excel.

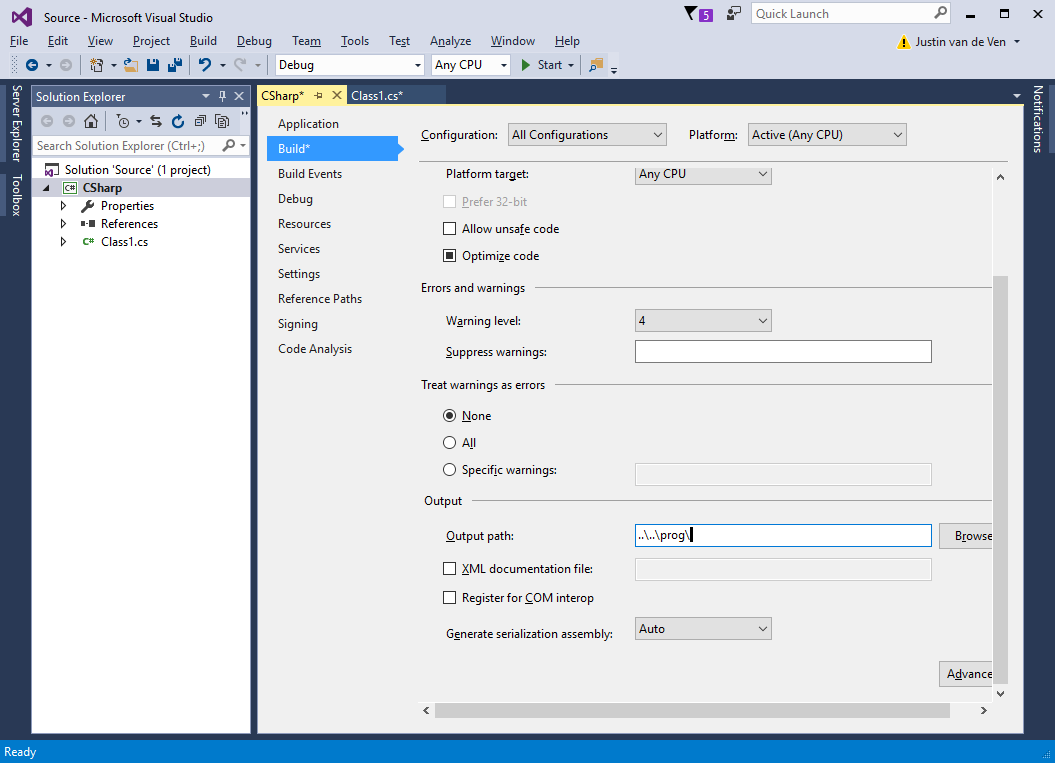
Set-up notes for Solution in 00 Experimental

IN VISUAL STUDIO:

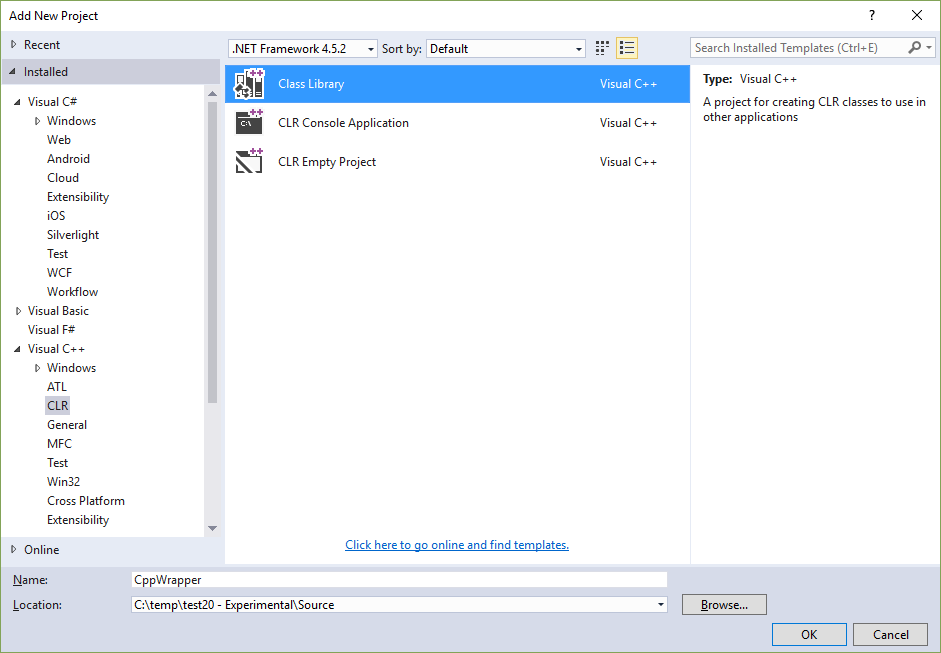
1. Add a new C# project, named CSharp, in Solution named Source



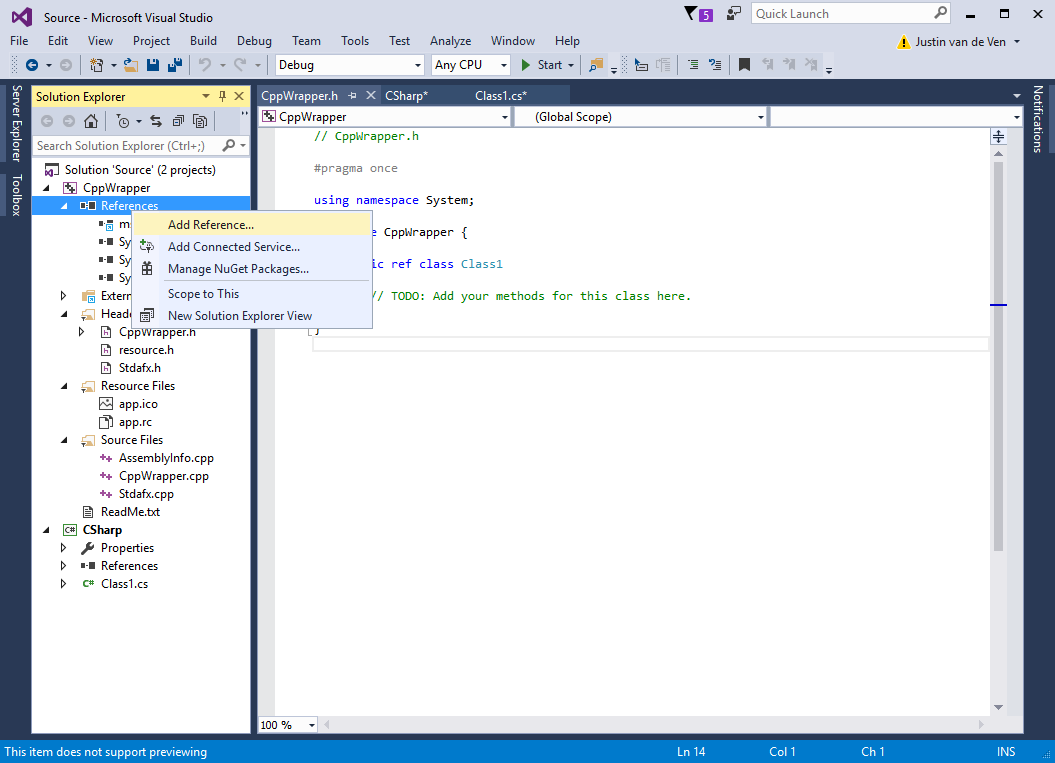
1. Copy in C# code – this code is standard for C#
2. Change output directory

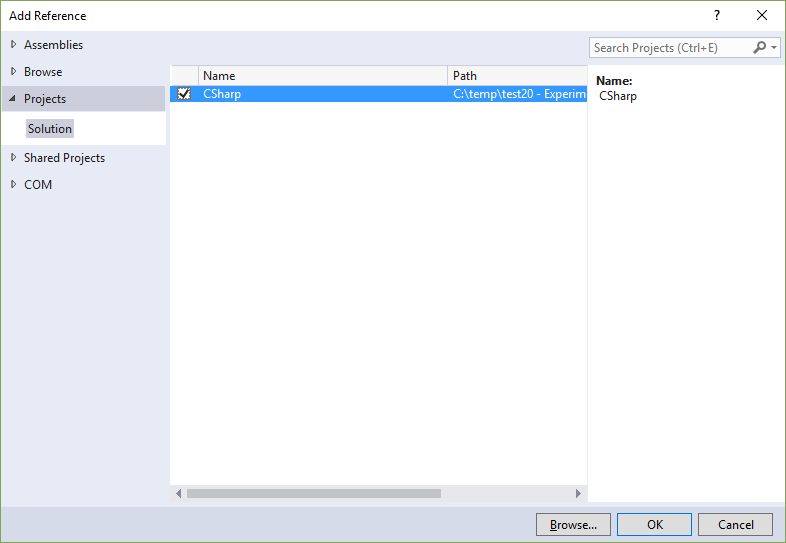


1. Check that the solution builds, and generates a file CSharp.dll in the output directory prog
2. Add a new C++ CLR project DLL, named CppWrapper

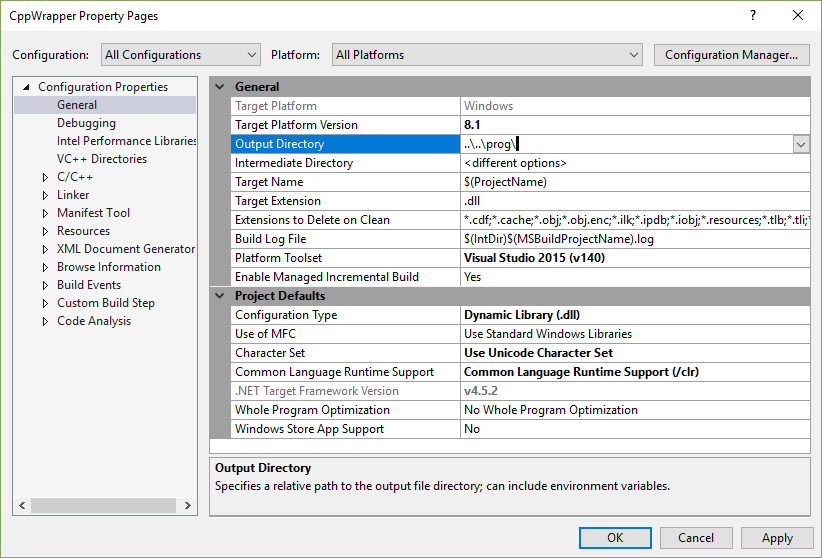


1. Add the C# project as a reference to CppWrapper

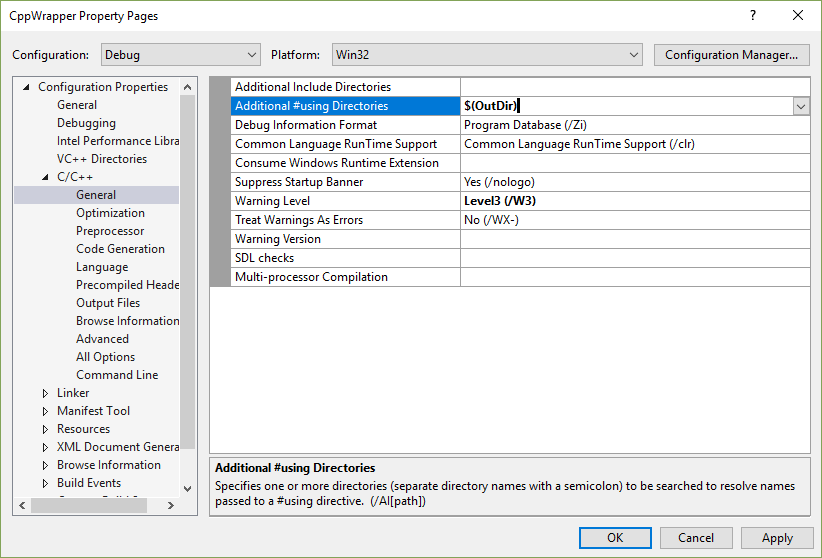




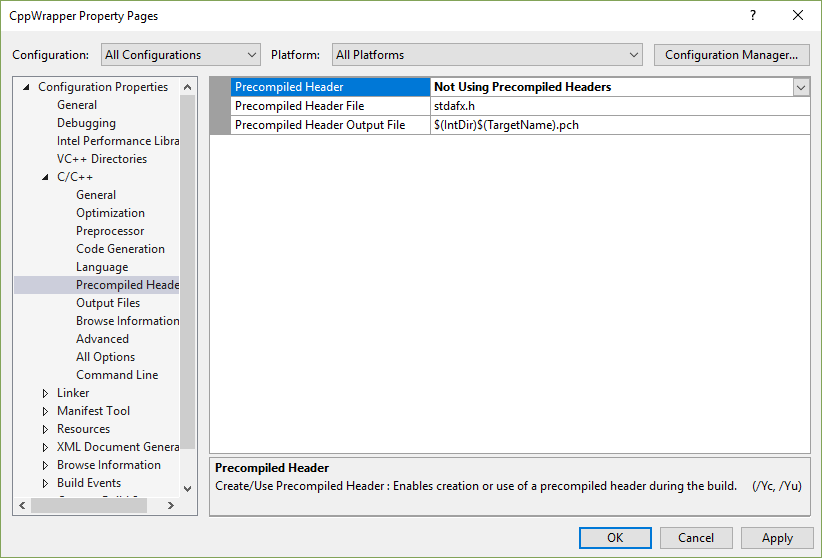
1. Change output directory



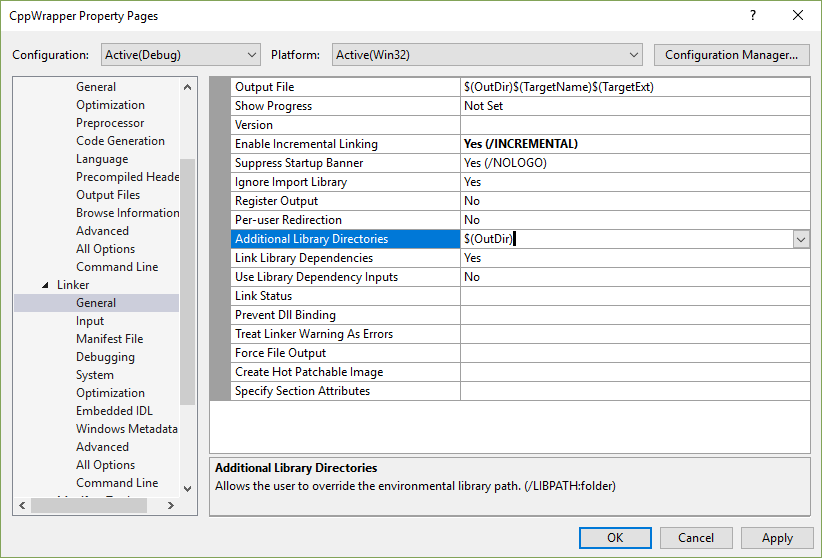
1. Add using directory



1. Suppress precompiled headers



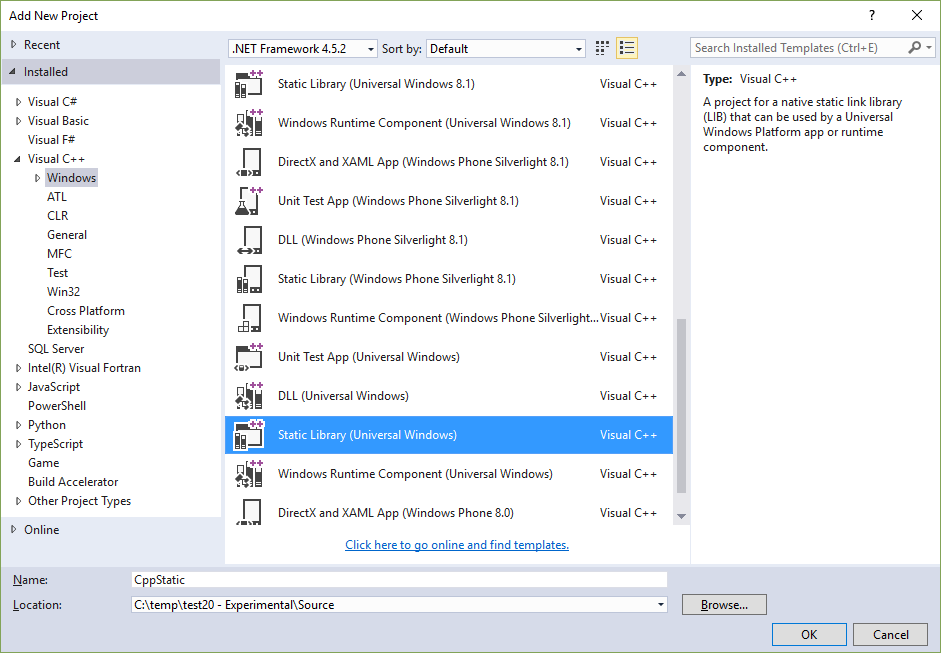
1. Add additional library directory



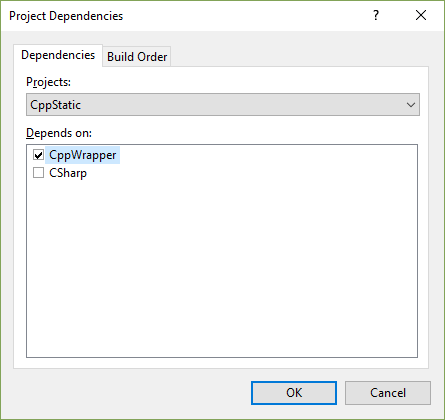
1. Add code to CppWrapper.cpp
2. Add code to CppWrapper.h
3. Rebuild solution

From here it is possible to link the C++ CLR wrapper to alternative Native C++ projects. The current write-up is to facilitate links with FORTRAN, which requires a static C++ library. See <http://pragmateek.com/using-c-from-native-c-with-the-help-of-ccli-v2/> for an example linked to a C++ DLL and then C++ console.

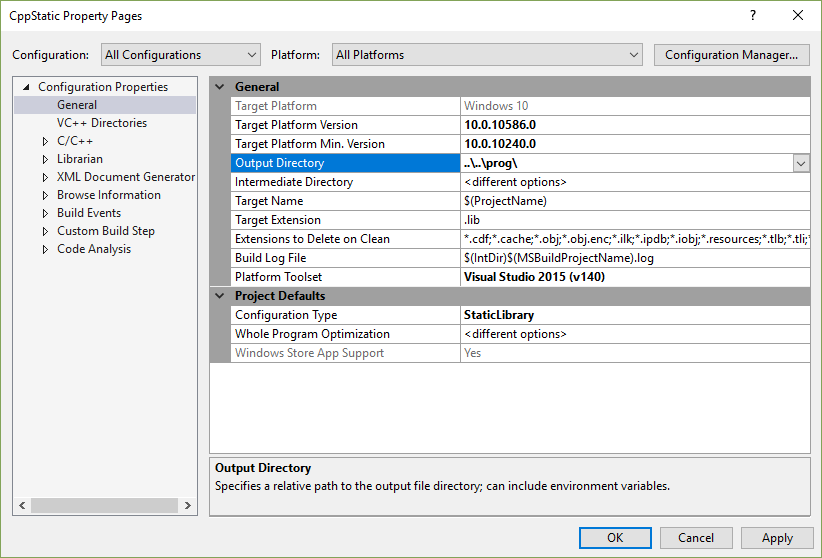
1. Add a new C++ static library project, named CppStatic



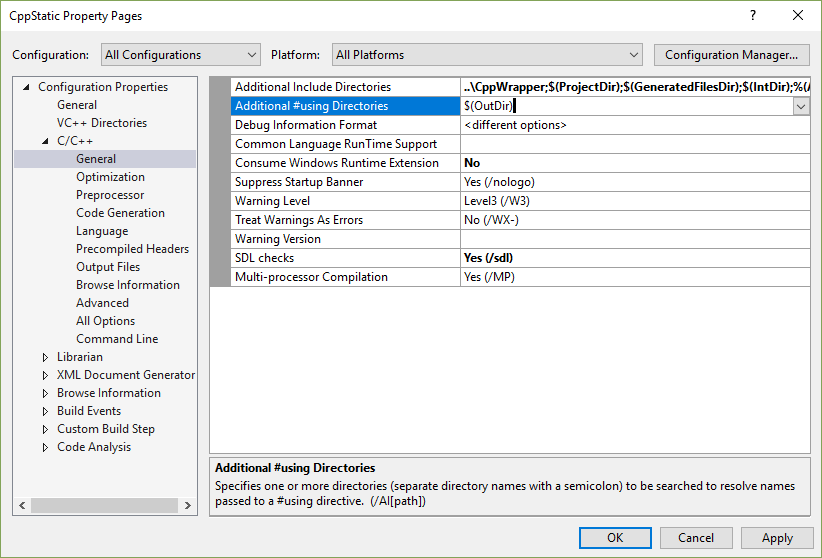
1. Make the CppStatic dependent on CppWrapper



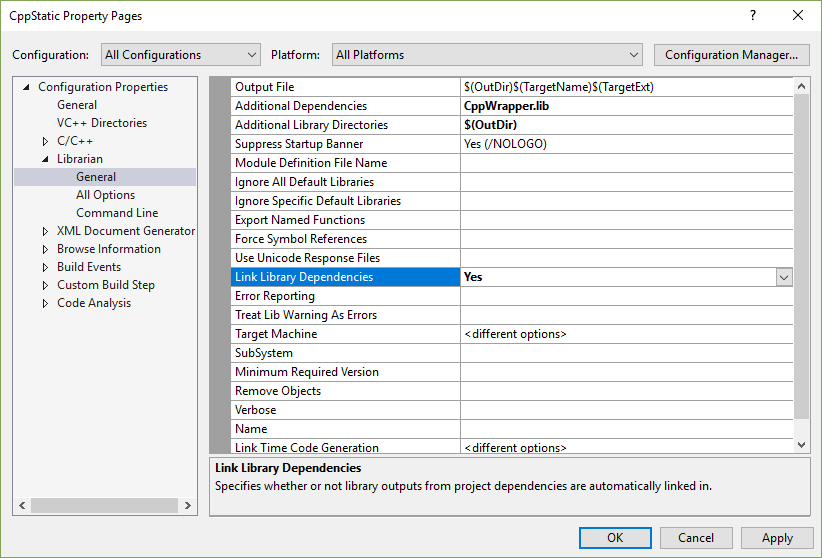
1. Change the output directory



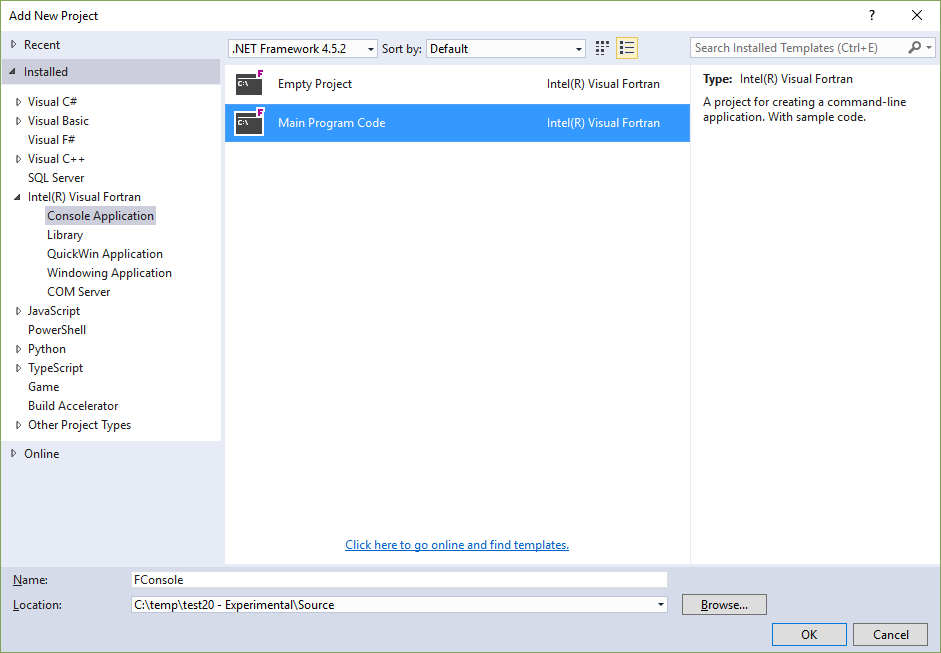
1. Add CppWrapper project directory to “Additional Include Directory” and output directory to “Additional Include directory”



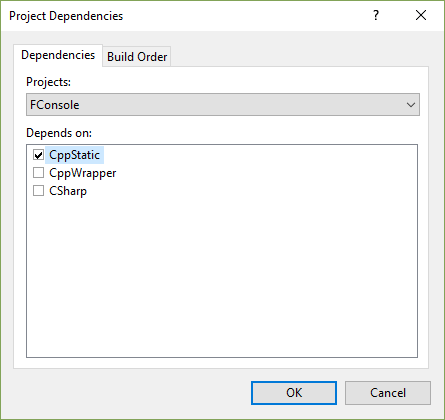
1. Add CppWrapper.lib to “Additional Library Dependencies”, and output directory to “Additional Library Directories”, and select Link Library Dependencies



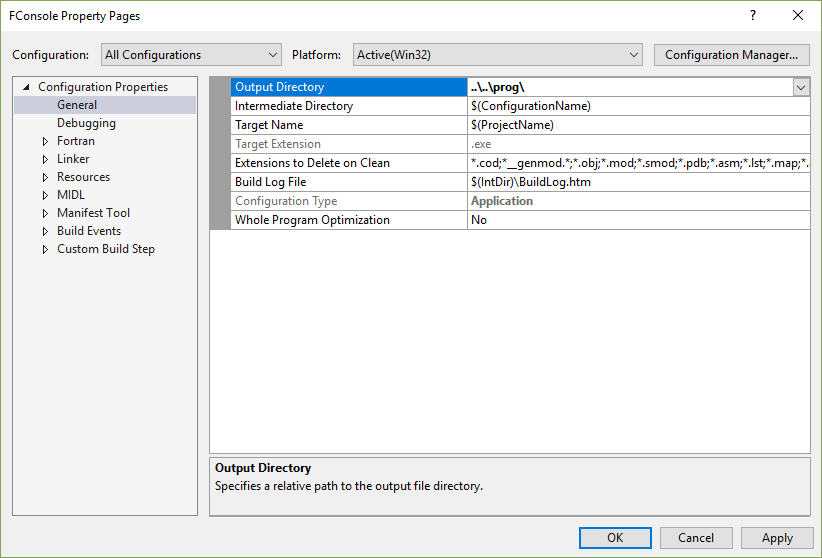
1. Add code to CppStatic.cpp
2. Add a new FORTRAN console project, named FConsole



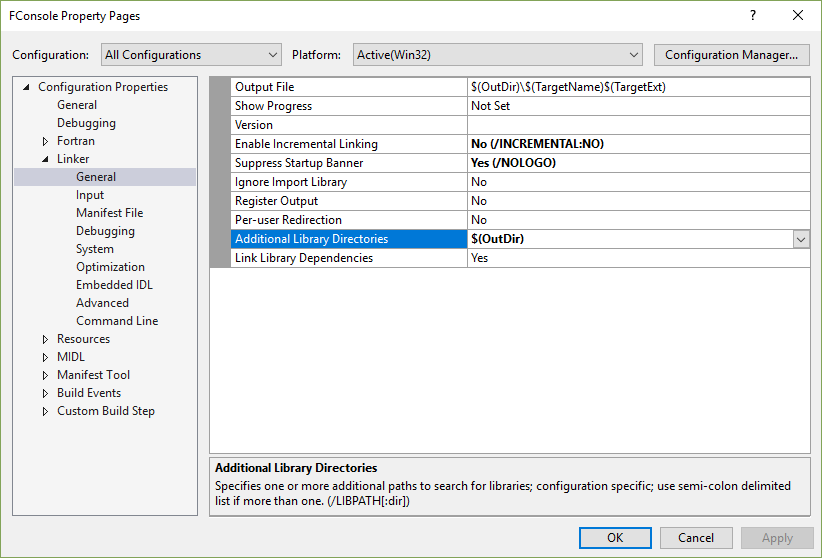
1. Make FConsole dependent on CppStatic



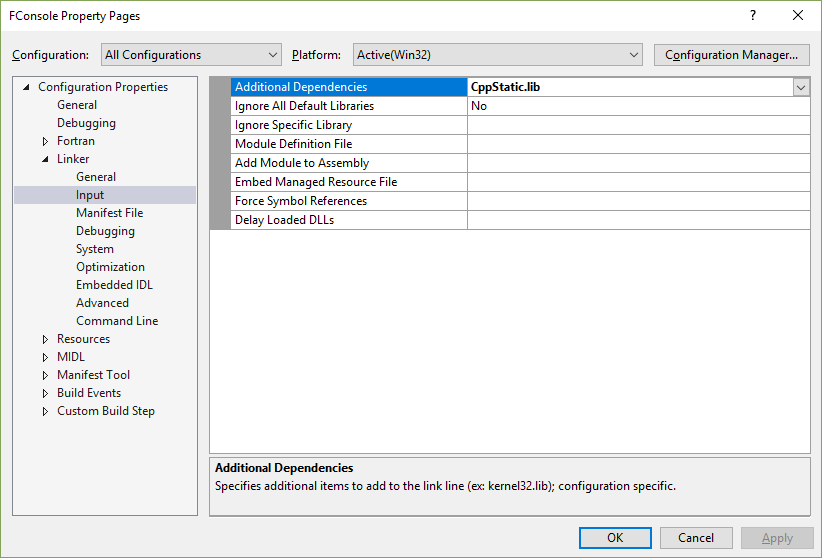
1. Change the output directory



1. Add Additional library directory

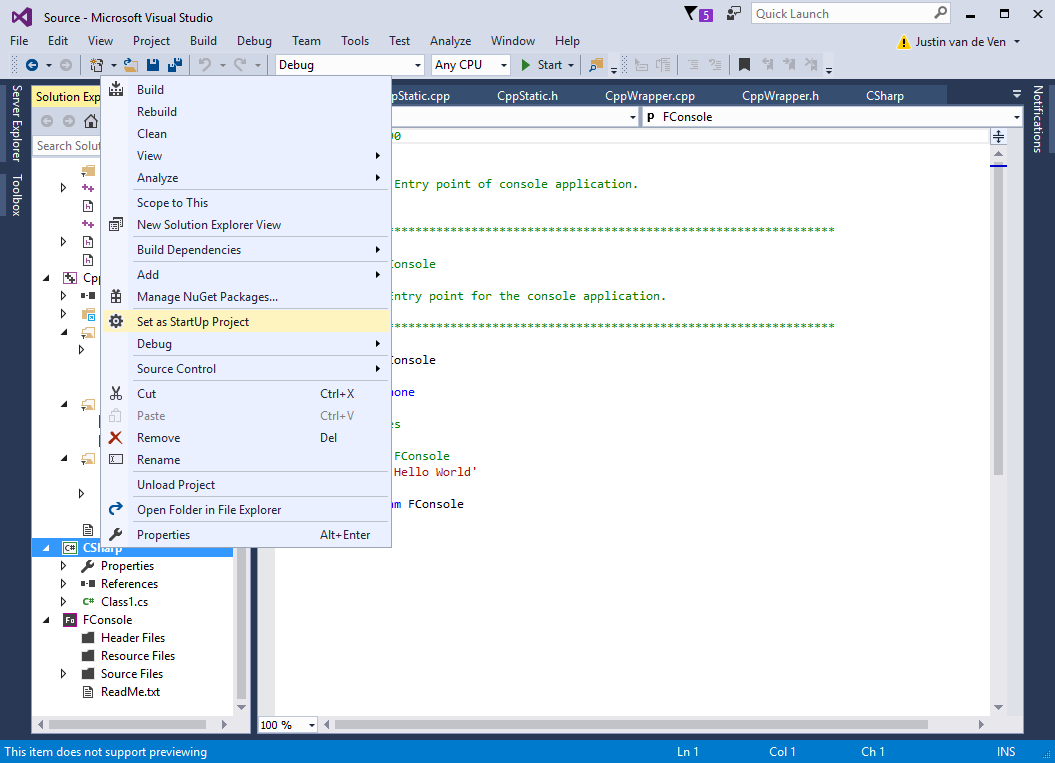


1. Add CppStatic.lib to Additional Dependencies

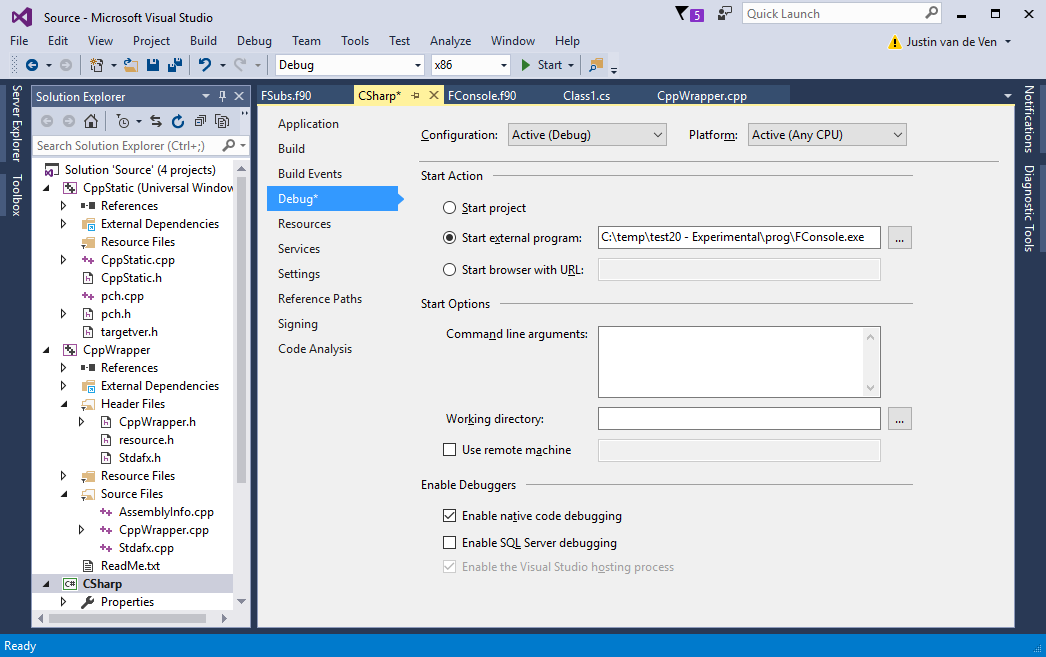


1. Copy Code to FConsole.f90
2. Add a new FSubs.F90 file, and copy code
3. Save all and build
4. Set C# to start-up file, for debugging:

<https://software.intel.com/en-us/articles/how-to-debug-a-fortran-application-calling-a-managed-dll>



1. Define FConsole.exe file as external program to start for debugging of C# project and enable native code debugging



1. It should now be possible to walk through the code with the VS debugger…