Using (The GSL) Libraries with Visual Studio

Get GSL

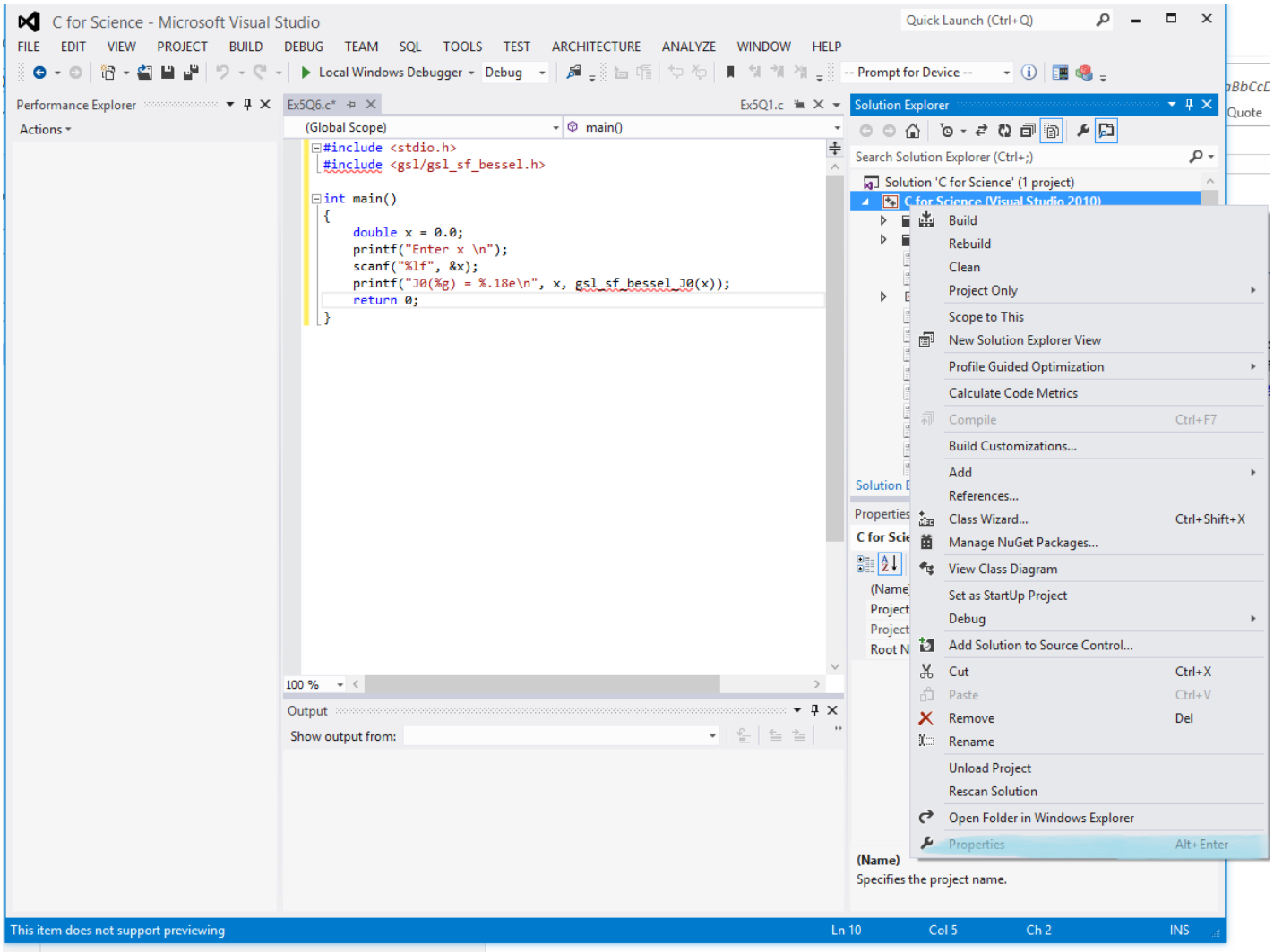
GSL (GNU Scientific Library) is designed to work with gcc. The source code is just ANSI C and can be ported and compiled for Windows with minor modifications. Many are available for download, or you can compile it yourself, here is a version I compiled to work with Visual Studio 2010, based on a previous port to Visual Studio 2008 by David Geldreich, this is available through the GNU General Public License:

http://www2.imperial.ac.uk/~shb104/c/files/other/gsl-1.13-vs2010.zip

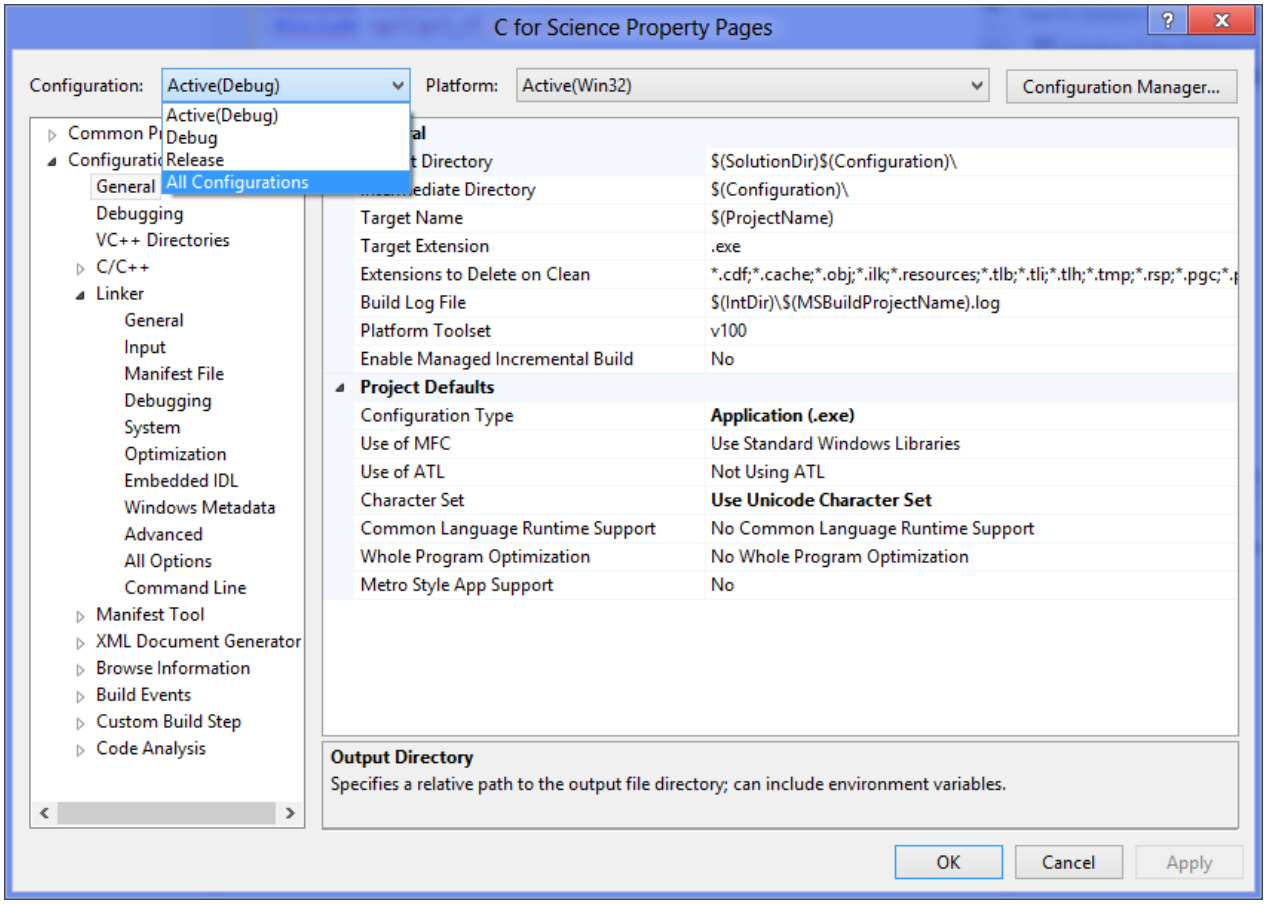
Unzip the contents of the download to somewhere such as “C:\” or “H:\”.

Setting up a Project

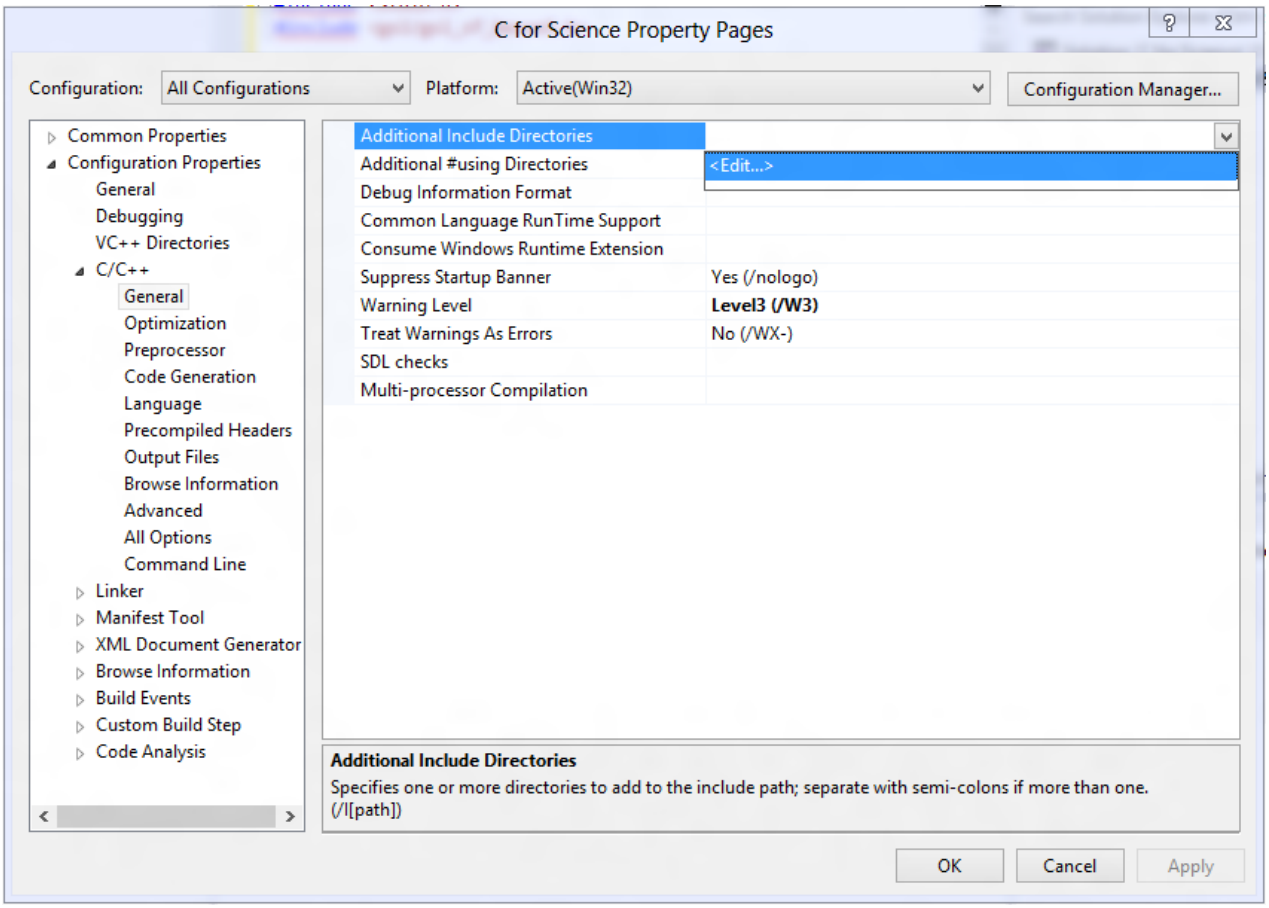
Right click on your Project and select “Properties”.



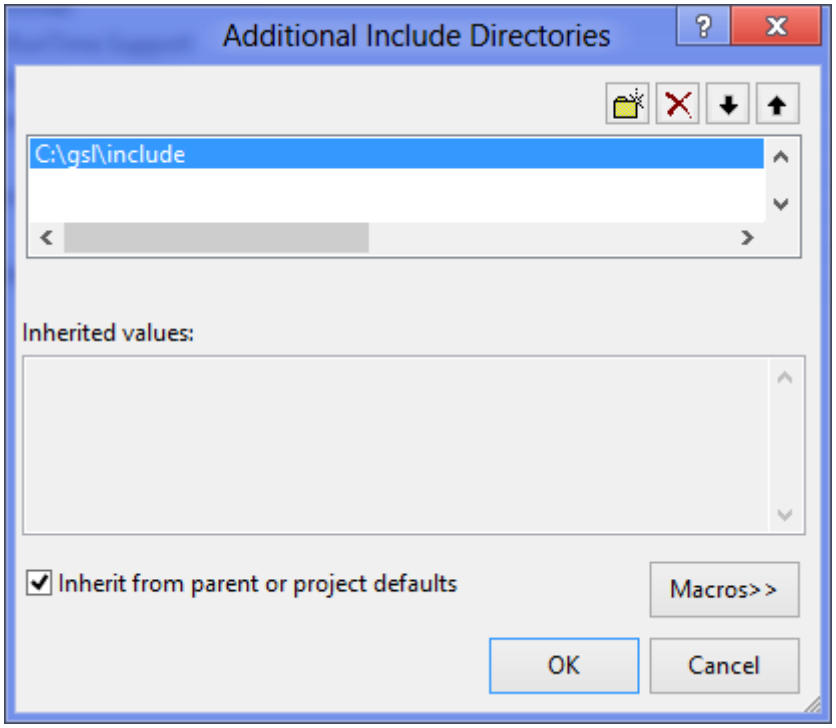
First we set up generic settings: Change the build configuration to “All Configurations”



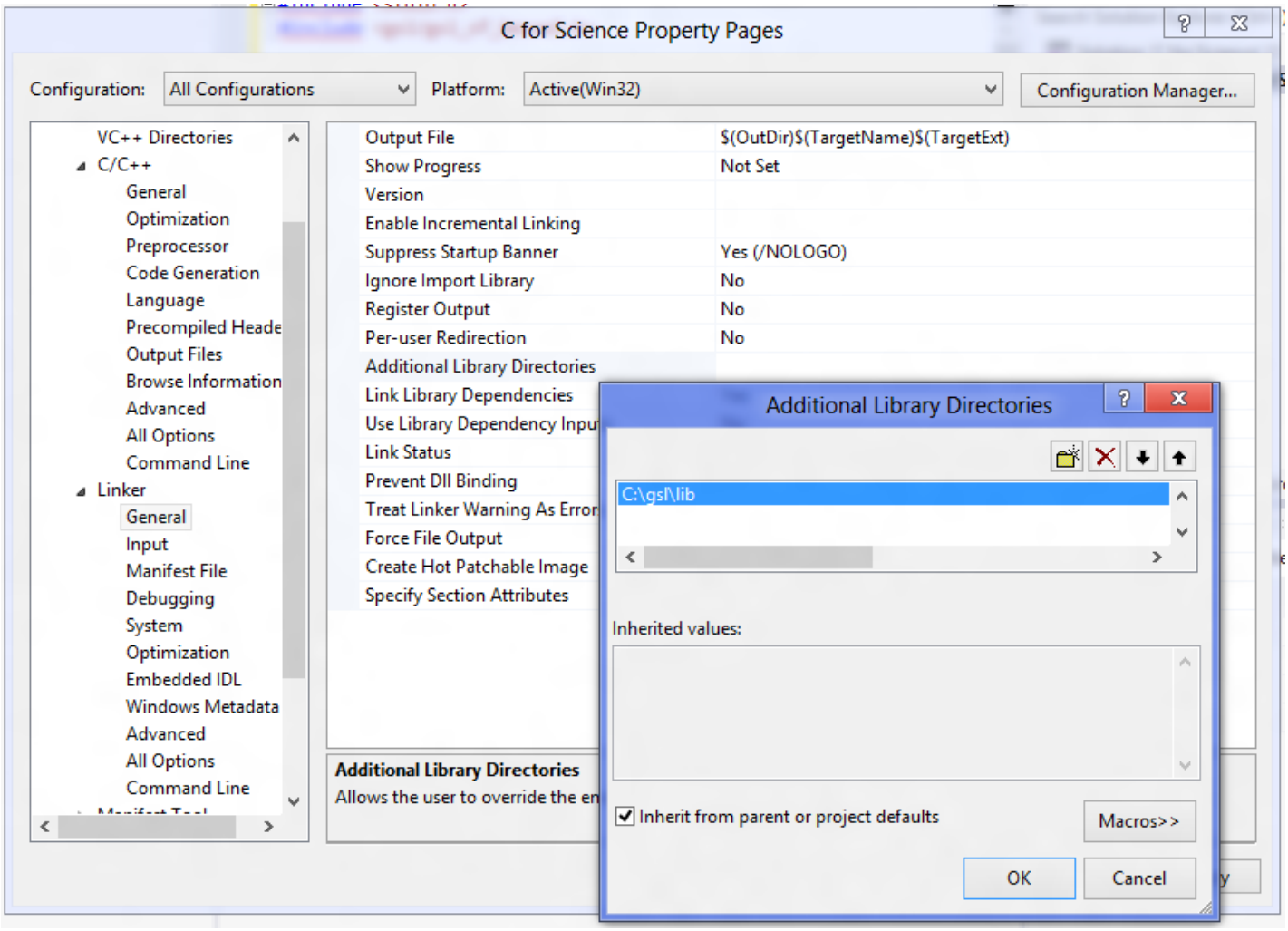
On the left hand Column, navigate to “Configuration Properties” -> “C/C++” -> “General”, then select and edit the “Additional Include Directories” line:



Add an entry for the “include” directory in your unzipped GSL.

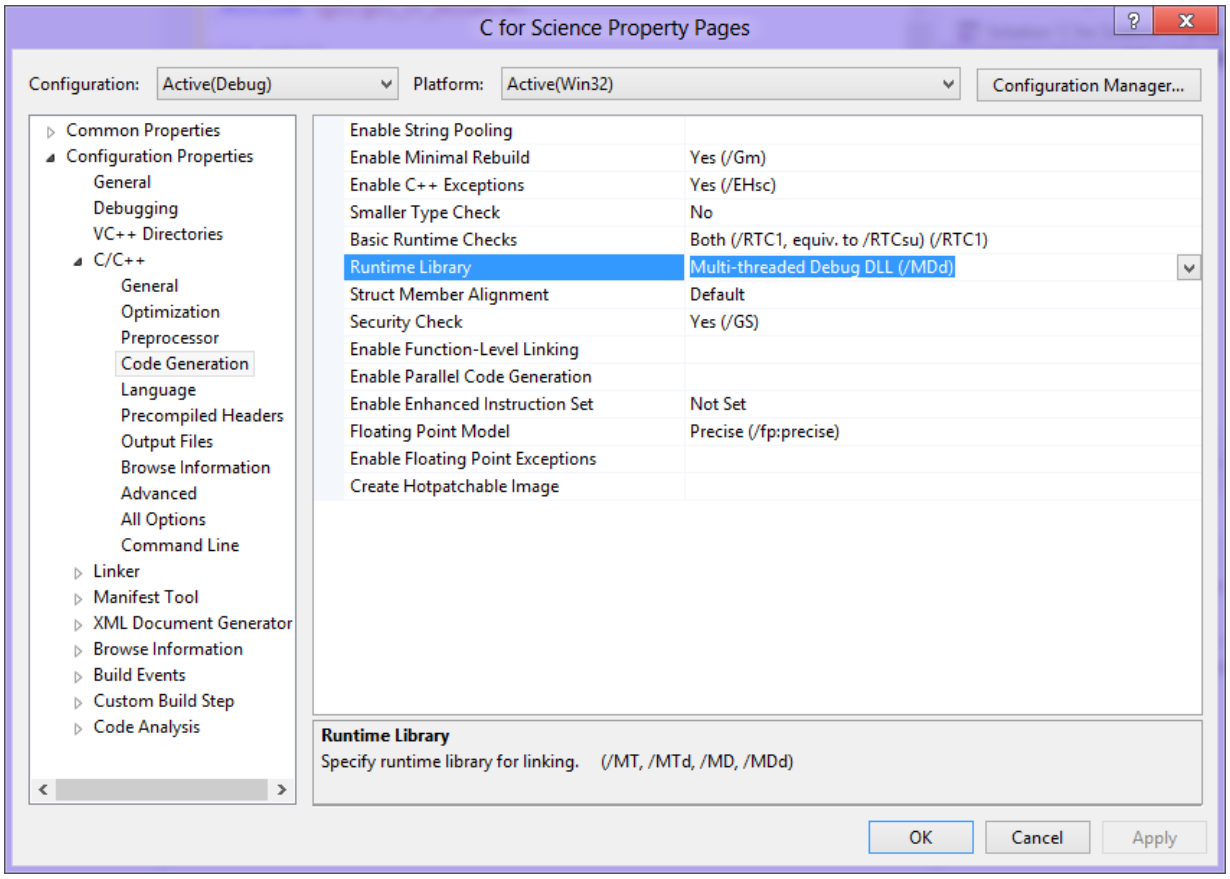


Also, in “Configuration Properties” -> “Linker” -> “General”, add the “lib” directory (it is in the same parent folder as “include”) to “Additional Library Directories”.

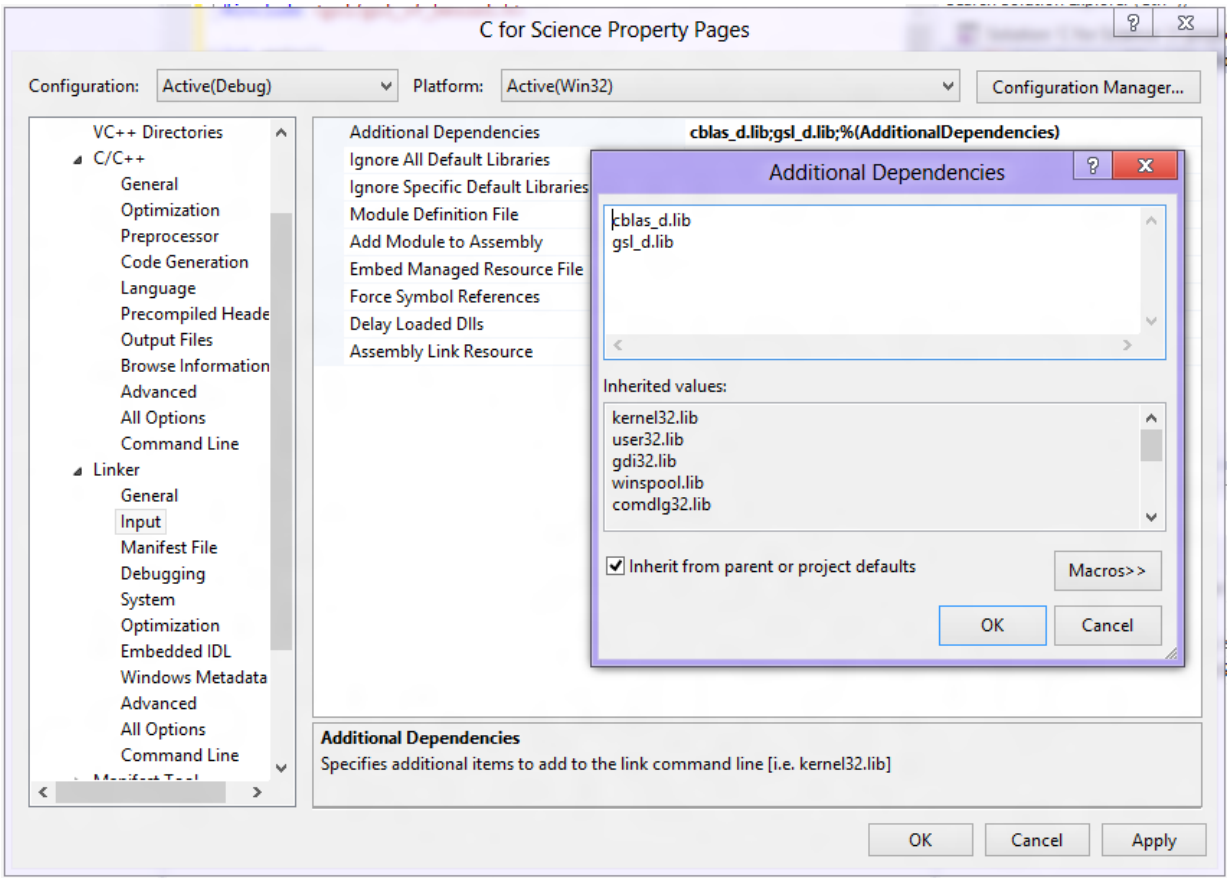


Now set up Debug specific settings: Click “Apply” and change the Project Configuration back to “Debug”.

Goto “Configuration Properities” -> “C/C++” -> “Code Generation” and ensure the Runtime Library is “/MDd”



Goto “Configuration Properties” -> “Linker” -> “Input” and add “cblas_d.lib” and “gsl_d.lib” to “Additional Dependencies”.



Now we set up Release specifics in a very similar way to Debug: Click “Apply” and change the Project Configuration back to “Debug”.

Goto “Configuration Properities” -> “C/C++” -> “Code Generation” and ensure the Runtime Library is “/MD”

Goto “Configuration Properties” -> “Linker” -> “Input” and add “cblas.lib” and “gsl.lib” to “Additional Dependencies”.

Click “OK”.