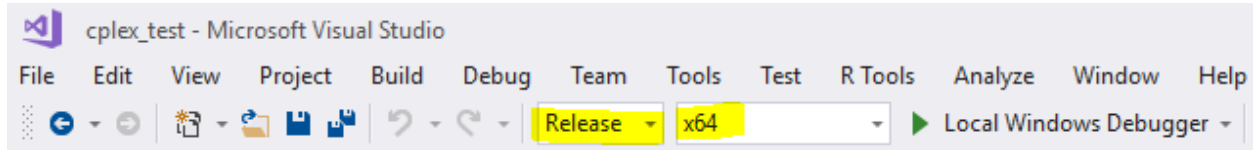


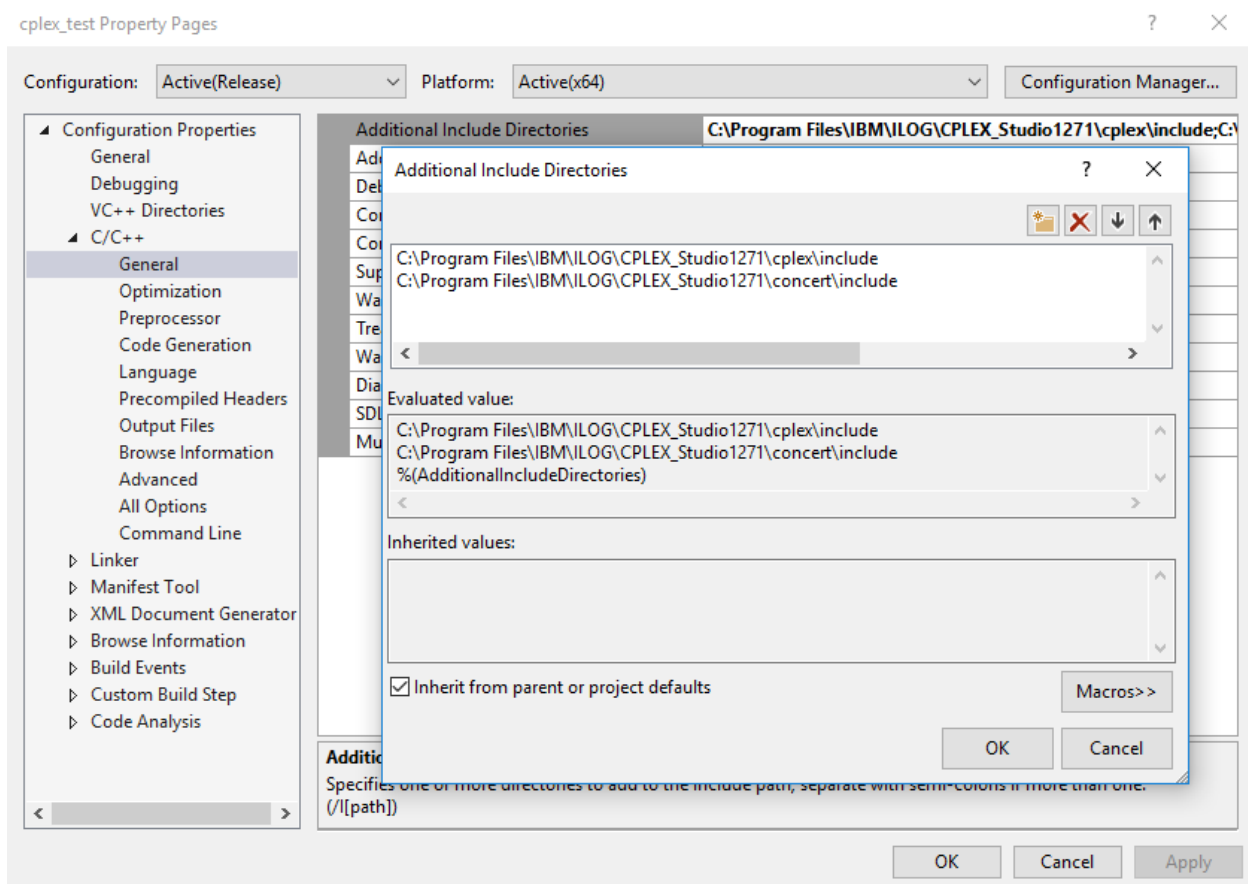
1. Install CPLEX by running the installer downloaded from IBM or the included installer:
'cos_installer-12.7.1.0.R1-win-x86-64.exe'
 - a. This installer is for installing CPLEX version 12.7.1. If installing a different version of CPLEX, keep in mind the version number, as it will be important for some of the following steps for establishing the correct paths and settings when creating a new CPLEX project. Examples will be provided using version 12.7.1 (*1271 for paths), but use your version during these instances.
 - b. Note: this not only installs the appropriate CPLEX libraries, but also the standalone programming environment. If necessary, you can run CPLEX code within this environment, but the syntax is *not* C++.
2. Ensure the computer is connected to the internet and install Visual Studio 2017 by running either the installer downloaded directly from Microsoft or the included installer:
'vs_2017_community.exe'.
 - a. While running the installer, install the appropriate C++ modules for development, as well as any additional modules desired.

When complete, open visual studio and create an empty Visual C++ Project. This will be your new CPLEX project. The following steps will add the appropriate settings to the single new CPLEX project; unless the modifications are added to the default project settings, the following steps will need to be added to each new CPLEX project. These instructions are borrowed from the following link and modified for troubleshooting: <https://www.leandro-coelho.com/how-to-configure-ms-visual-studio-to-use-ibm-cplex-concert/>

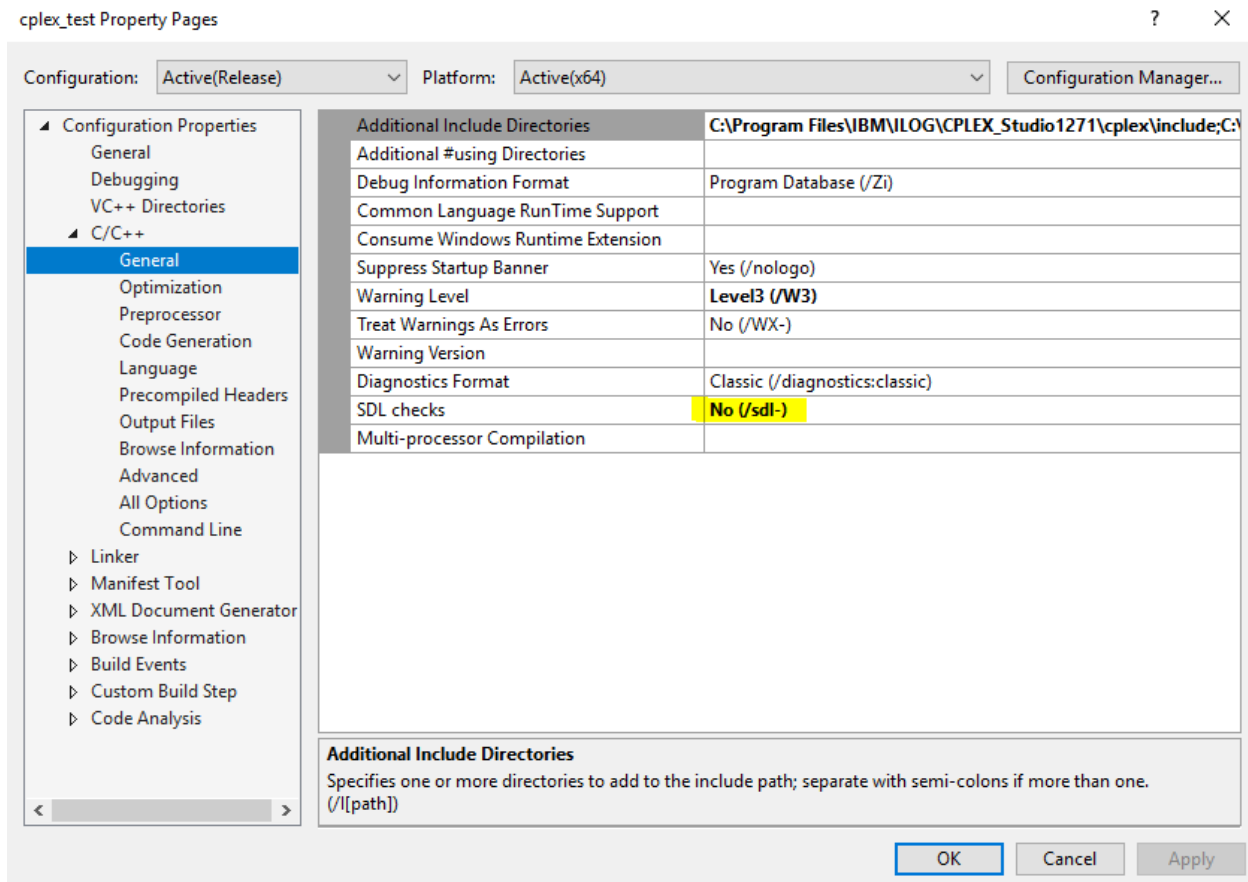
3. Set the project to be in 'Release' and 'x64' mode at the top of the Visual Studio menu. It should look like this:



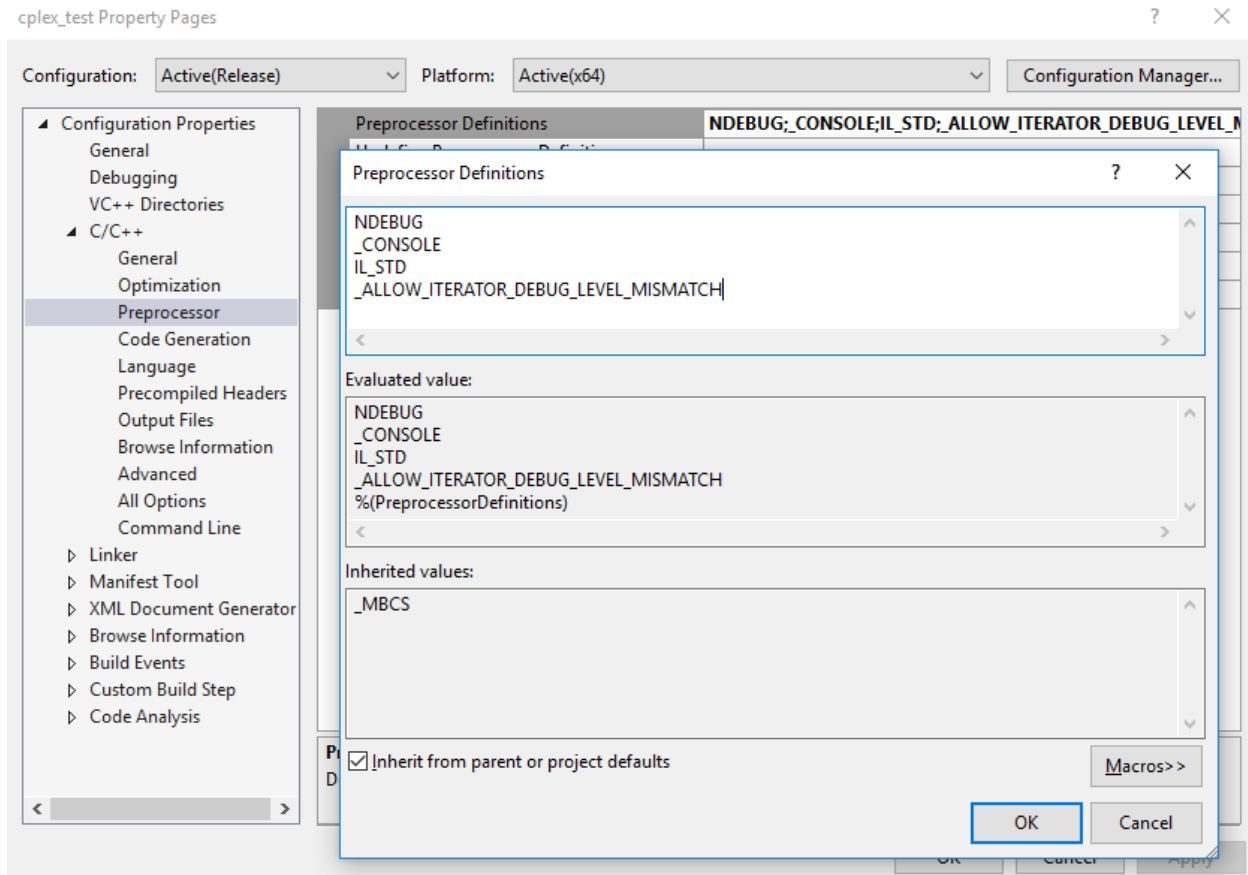
4. In the new CPLEX project, right click on the project in the solution explorer and select 'Properties'.
 - a. In Configuration Properties > C/C++ > General, open and edit the property: 'Additional Include Directories'
 - i. Here you will need to add the full path to the include directories of CPLEX and CONCERT. This is determined by the install directory and version of CPLEX installed. Mine looks like:



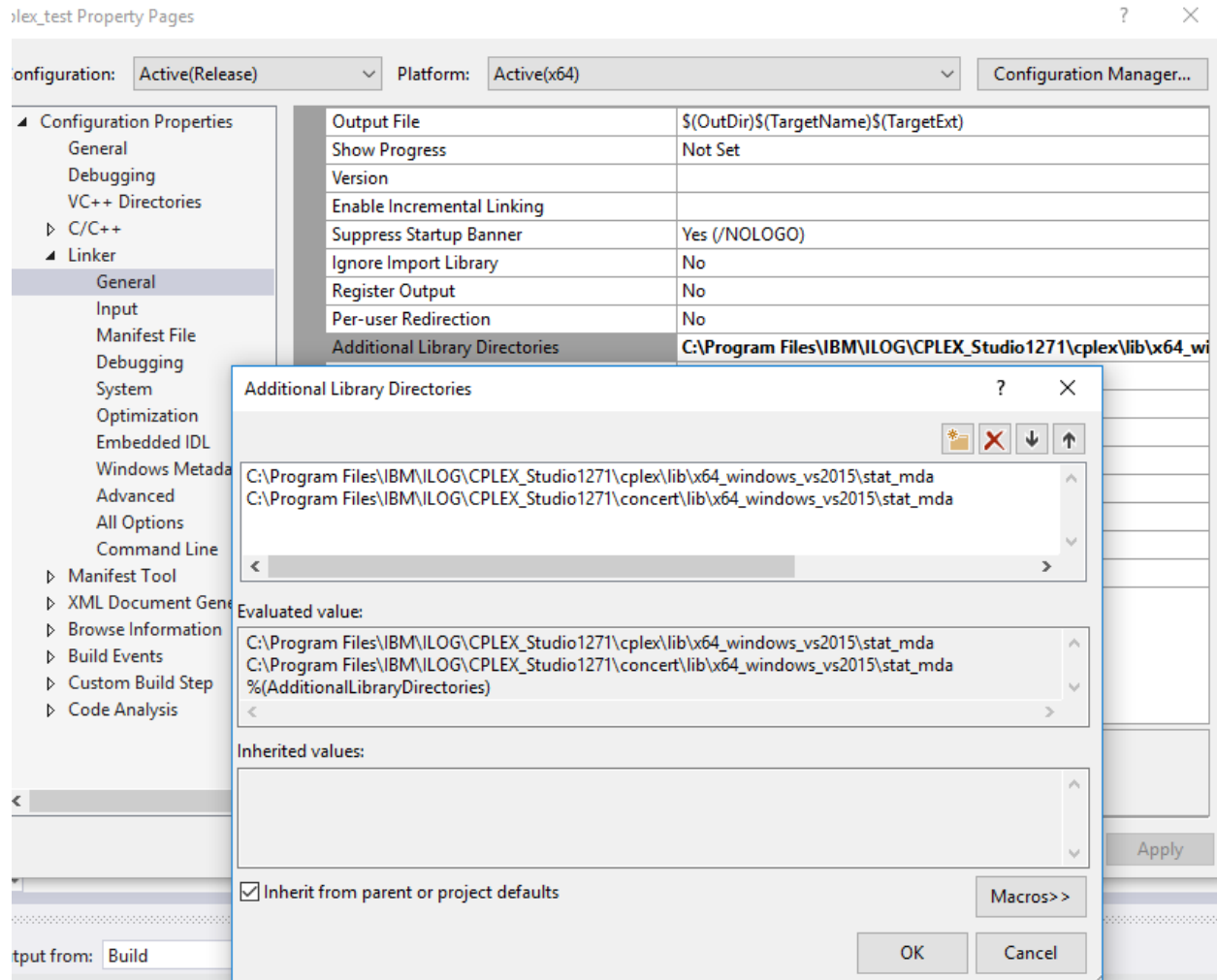
- ii. On the same properties page, make sure the property 'SDL checks' is set to 'No'. One of the IloCplex files appears to be depreciated, and will stop compilation if this is set to 'Yes'.



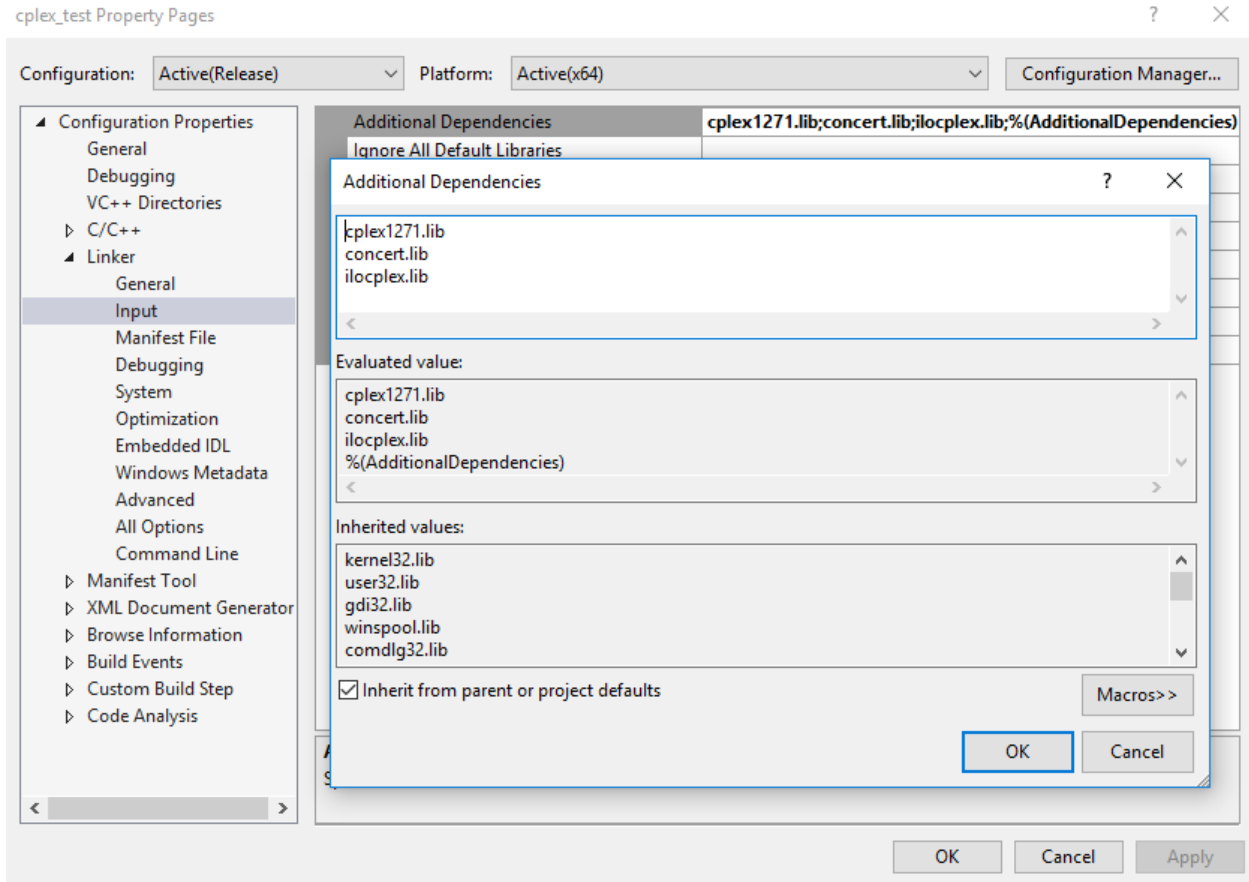
- b. In Configuration Properties > C/C++ > Preprocessor, open and edit the property: 'Preprocessor Definitions'
- i. Add 'NDEBUG', '_CONSOLE', 'IL_STD', and '_ALLOW_ITERATOR_DEBUG_LEVEL_MISMATCH'. It should look something like this:



- c. In Configuration Properties > Linker > General, open and edit the property: 'Additional Library Directories'
- i. Add the folders corresponding to your system and visual studio platform, which should under the ..\lib directory near to the ..\include directories used for step 3a; vs2015 files are sufficient for vs2017. Mine looks like:



- d. In Configuration Properties > Linker > Input, open and edit the property: 'Additional Dependencies'
 - i. Add the necessary dependencies: 'cplex*.lib', 'concert.lib', 'ilocplex.lib', where * is the version number; for me, the file is 'cplex1271.lib'. Mine looks like:



After you have followed these instructions, you should be able to create a CPLEX model/file and compile it as necessary. I have included a test project, *cplex_test*, which should be able to open and build in your version of CPLEX. The modifications have been made to the project as per the computer it was built on, but so long as the CPLEX installation path is the same, it should still build and compile. If the paths are different than they are indicated here in this document, then, if you update the paths as necessary, you should be able to compile this project as proof that CPLEX is installed correctly.