



Download

- Binaries
- Source Code
- Developer Zone

Help

- CC3D User Forum
- Manuals
- Tutorials
- F.A.Q.

Demos

- Simulation Movies
- Screenshots
- Model Repository

Publications

- Publications
- Theses
- Talks and Posters

Events

- Workshops

About

- People
- Contact Us
- Mailing List

Search Site

Building CompuCell3D on Windows using Visual Studio 2015

Building CompuCell3D from source on Windows systems using freely available Visual Studio 2015 is fairly straightforward. We provide all dependencies so the only thing you need to do is to install Visual Studio 2015 (Community Edition is free and fully functional), [SWIG](#), [CMAKE](#) and [GIT for windows](#). The following will help you build and install CC3D on most Windows systems.

Prerequisites

Hardware

CompuCell3D may build and run with less capable hardware, but has been tested with the following:

- 512MB RAM
- Hardware 3D Graphics Acceleration (most modern graphics cards)

Build Tools and Dependencies

[SWIG](#), [CMAKE](#) and [GIT for windows](#)

- [SWIG](#) (version 1.3 or higher recommended) - after installing SWIG add SWIG installation directory to search path
- [CMAKE](#) (CMake-GUI is recommended)
- [GIT for windows](#)
- A directory with precompiled dependencies (that include Python and all libraries that CC3D depends on)

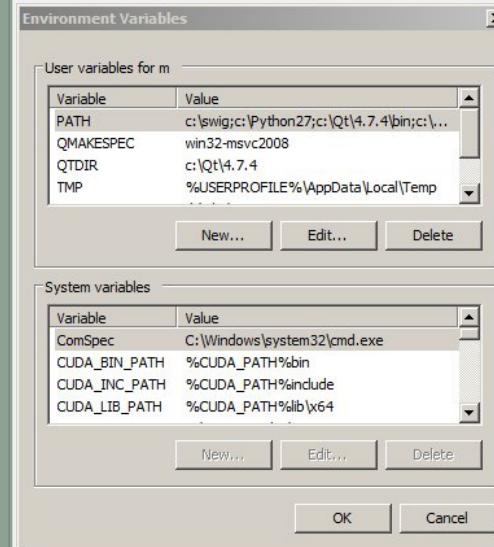
= Installing CC3D Compile Dependencies =

Go to [dependency repository](#) and download [cc3d_vs2015_dependencies_2017.zip](#)

Unpack it to your **C:/** or **D:/** drive and make sure you have either **C:/prerequisites_2017** or **D:/prerequisites_2017** on your system.

Adding SWIG to search Path

In my case swig was installed to **C:/swig**. Then I go to open Control Panel » System » Advanced » Environment Variables. They are separated into user and machine specific values. You can view and edit their values there. Their current values upon launch are made available to all programs. I will add **c:/swig** to PATH variable (I am editing user environment variables, not the global ones) as shown below:

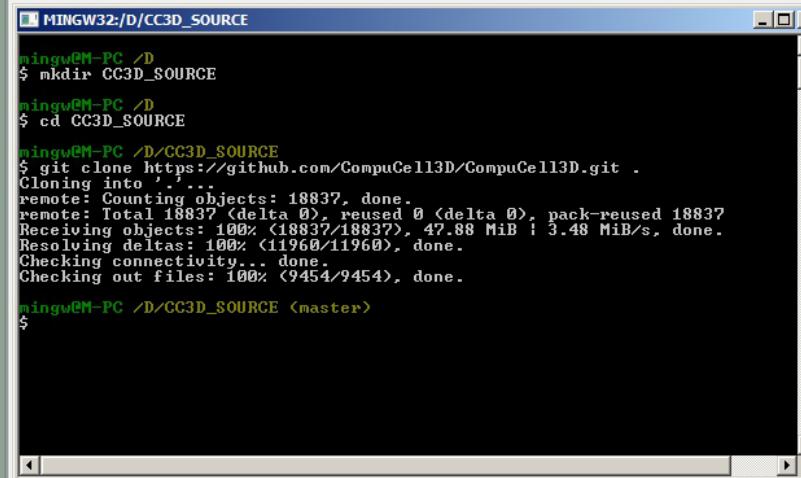


Source Code

Once the dependencies have been satisfied, obtain the source code from our GIT repository using the following command from git command-line tool:

```
git clone https://github.com/CompuCell3D/CompuCell3D.git D:/CC3D_SOURCE
```

Take a look at the screenshot to see how this would look:



```
MINGW32:D/CC3D_SOURCE
mingw@M-PC ~
$ mkdir CC3D_SOURCE
mingw@M-PC ~
$ cd CC3D_SOURCE
mingw@M-PC /D/CC3D_SOURCE
$ git clone https://github.com/CompuCell3D/CompuCell3D.git .
Cloning into '.'...
remote: Counting objects: 18837, done.
remote: Total 18837 (delta 0), reused 0 (delta 0), pack-reused 18837
Receiving objects: 100% (18837/18837), 47.88 MiB / 3.48 MiB/s, done.
Resolving deltas: 100% (11960/11960), done.
Checking connectivity... done.
Checking out files: 100% (9454/9454), done.

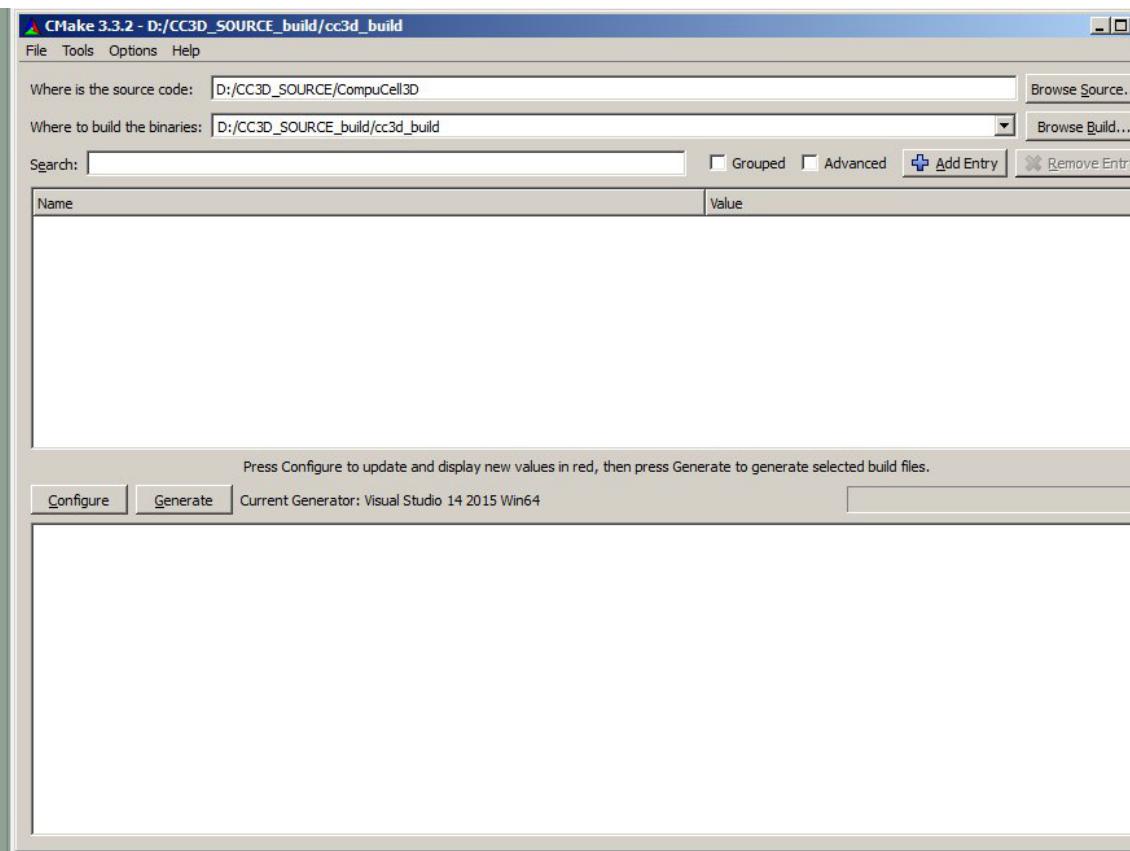
mingw@M-PC /D/CC3D_SOURCE (master)
$
```

This will create a directory d:\CC3D_SOURCE` and download the source code into it.

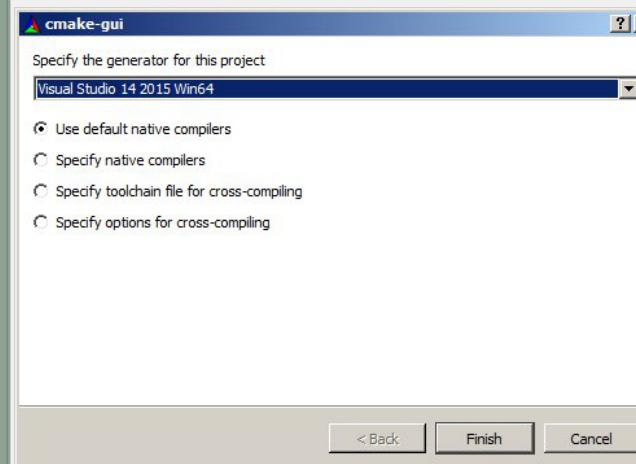
At this point we are ready to configure CC3D for building using Visual Studio 2015

Generating VS 2015 project using CMake

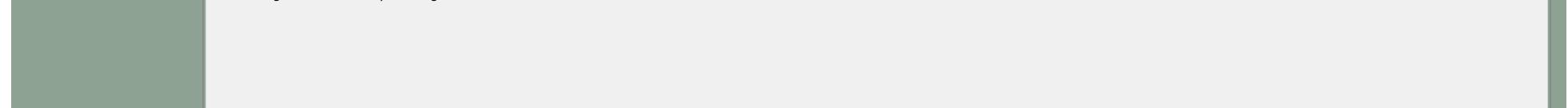
CompuCell3D is configured using the CMake build system. Double click CMake icon and set source and build directories to d:\CC3D_SOURCE\CompuCell3D and D:\CC3D_SOURCE\cc3d_build as show below:

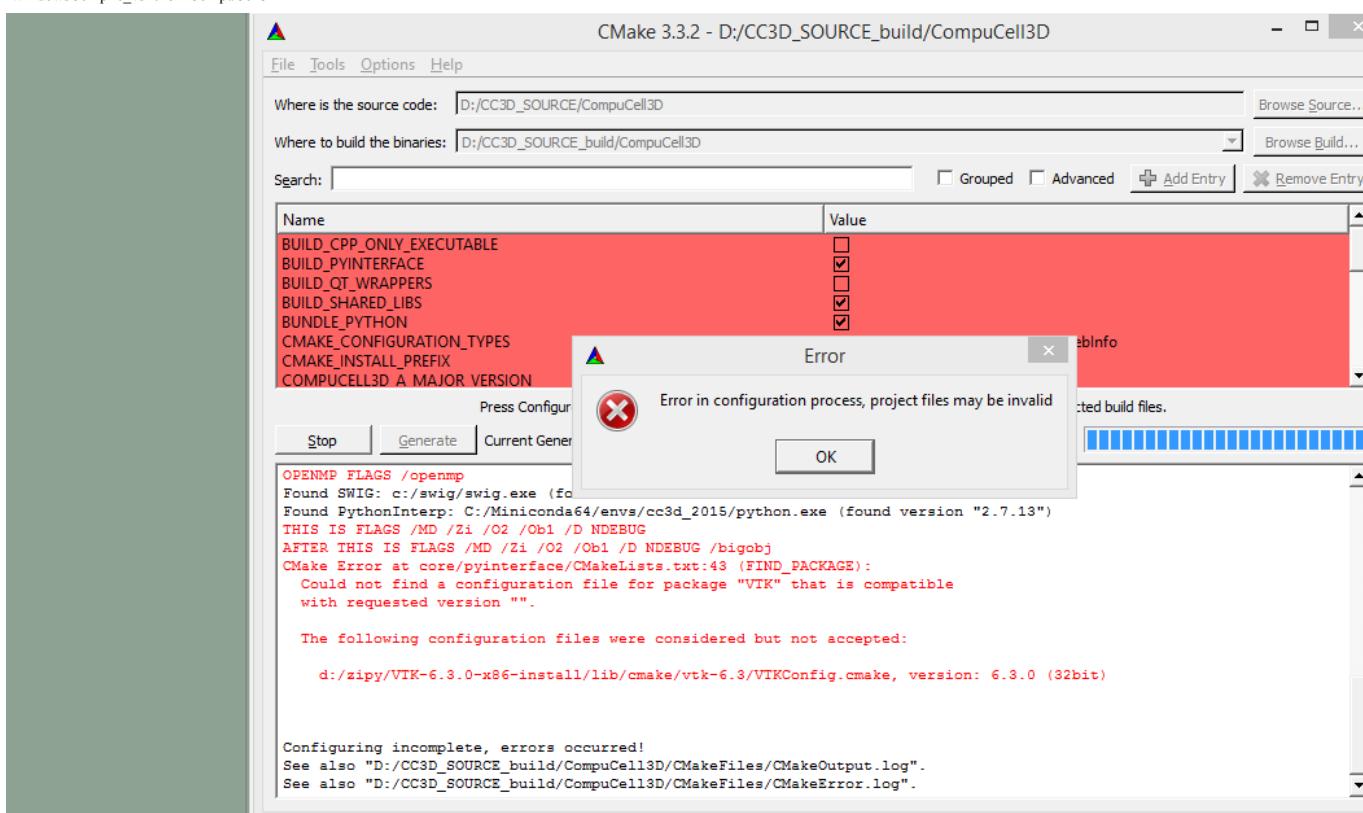


After clicking *Configure* you will get a pop up dialog where you should select Visual Studio Project 2015 as a target project generator - here we will choose 64-bit build



After configuration is done you will get a screen that looks somewhat similar to the one below:

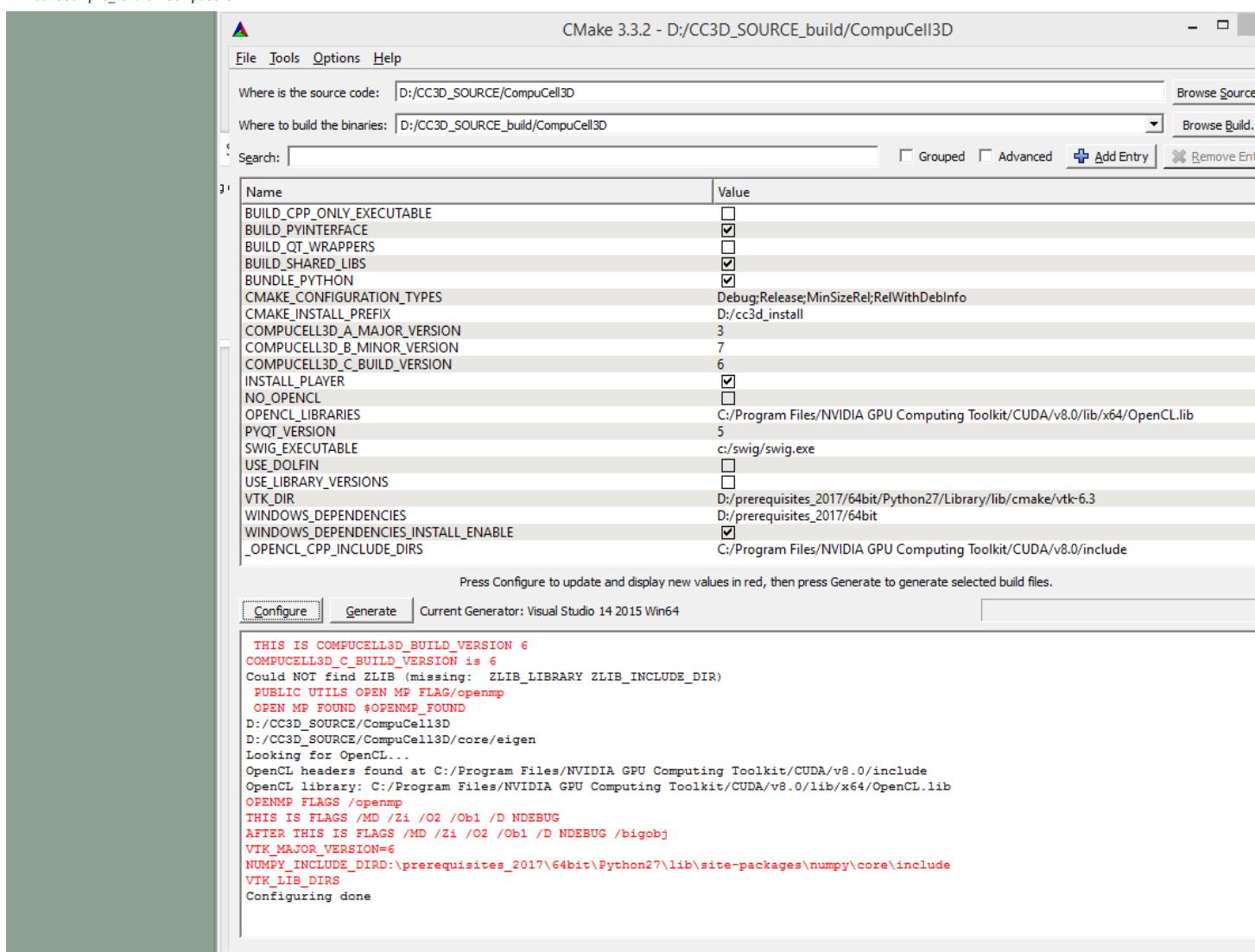




As you can see it informs you that the error has occurred. This is normal and it means we need to provide certain type of information to CMAKE:

- Directory with VTK CMake configure scripts (**VTK_DIR** = D:/prerequisites_2017/64bit/Python27/Library/lib/cmake/vtk-6.3)
- Folder into which we should install CC3D (**CMAKE_INSTALL_PREFIX** = D:/cc3d_install)
- Folder with CC3D Windows Dependencies (**WINDOWS_DEPENDENCIES** = D:/prerequisites_2017/64bit)

After we input these changes (see below) and click *Configure* again we will get the following screen:

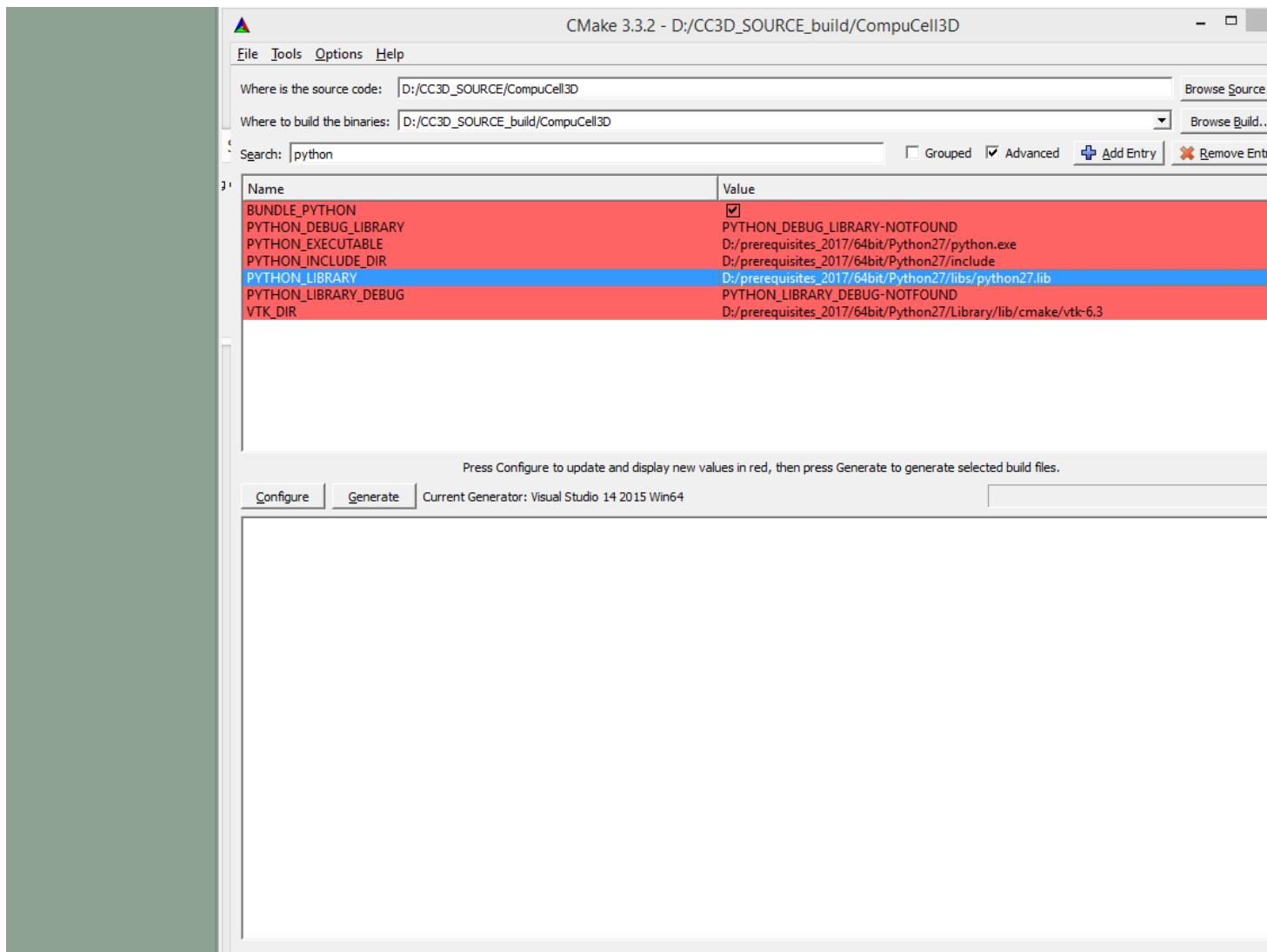


We are almost done, but now we need to make sure that Python library, Python include path and Python executable point to the Python distribution that is bundled in the *D:/prerequisites_2017/64bit* folder:

To do that click **Advanced** check-box at the top and type Python in the search bar next to the box you have just clicked. The input the following:

- **PYTHON_EXECUTABLE** = *D:/prerequisites_2017/64bit/Python27/python.exe*
- **PYTHON_INCLUDE_DIR** = *D:/prerequisites_2017/64bit/Python27/include*
- **PYTHON_LIBRARY** = *D:/prerequisites_2017/64bit/Python27/libs/python27.lib*

as shown below:

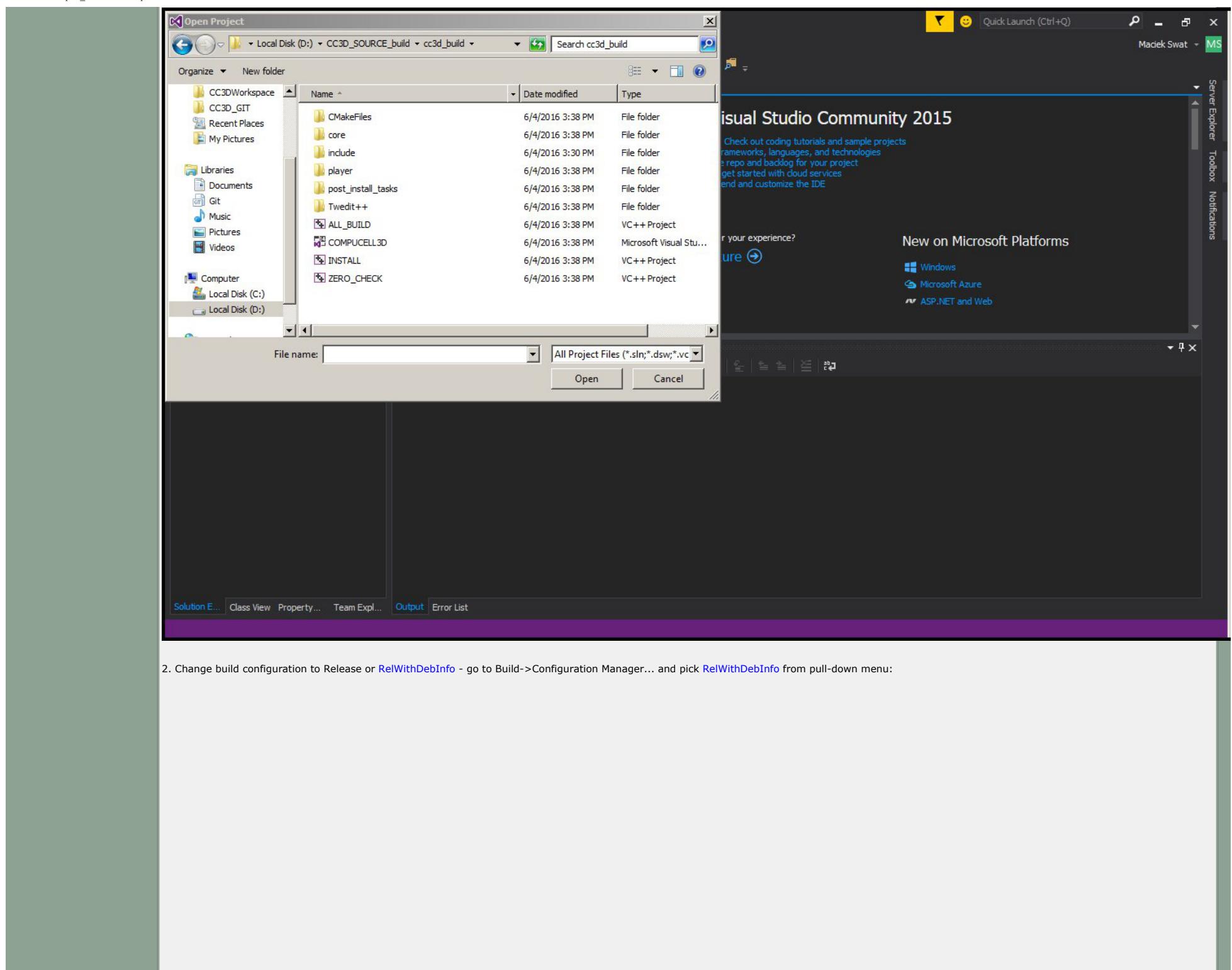


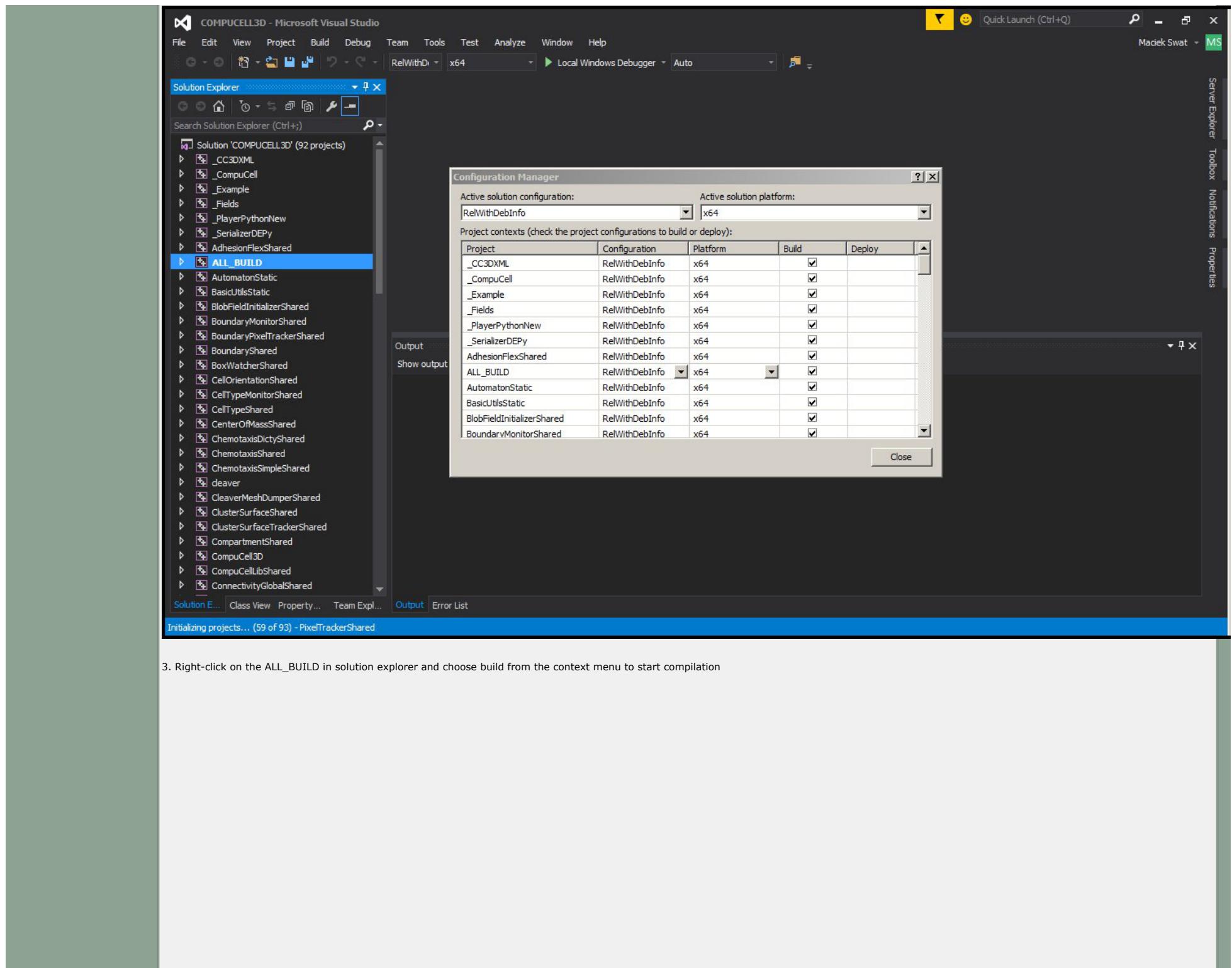
You may check **NO_OPEN_CL** option if you do not have OPEN_CL toolkit installed

Click **Configure** , followed by **Generate** buttons and then open up Visual Studio 2015.

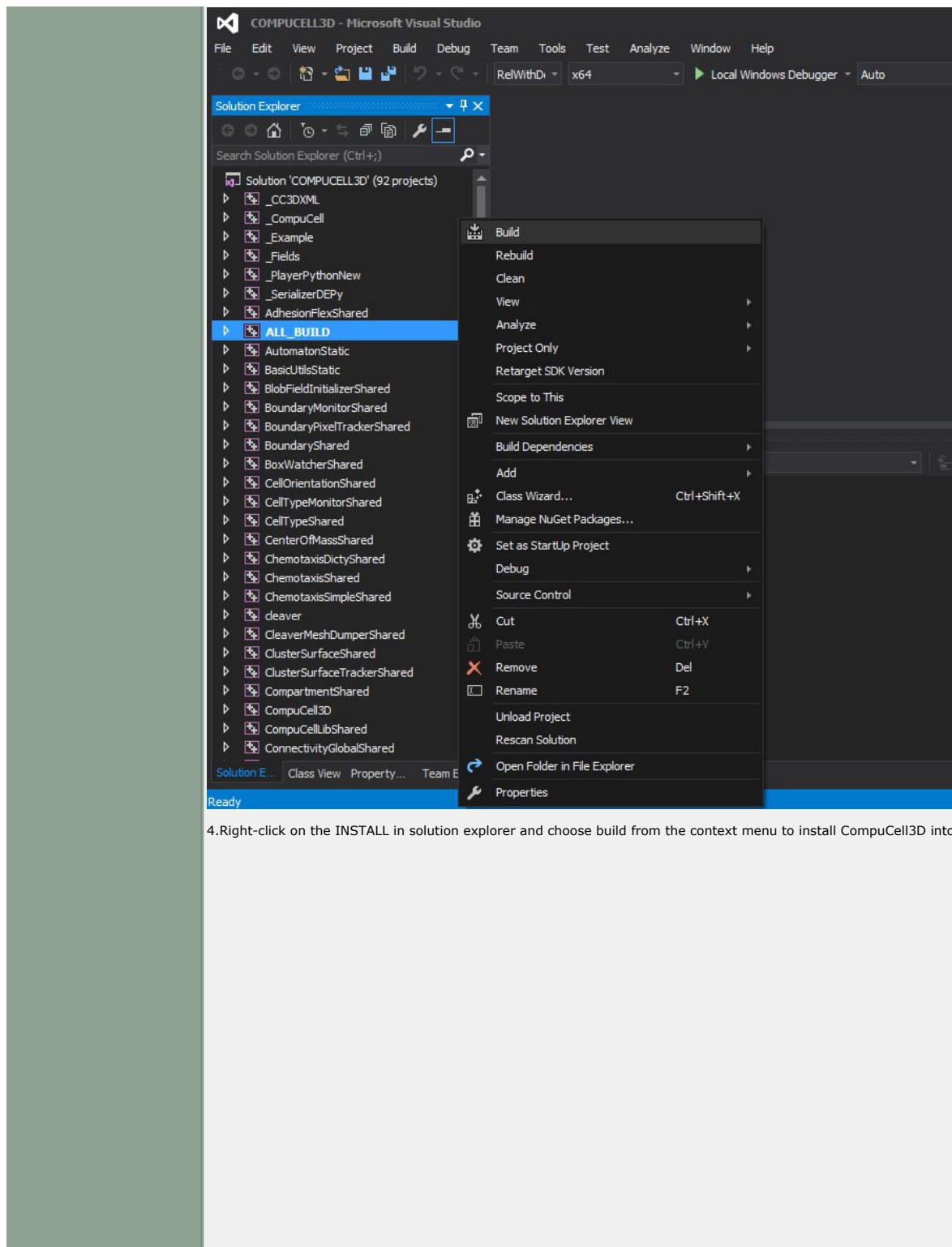
Compilation with Visual Studio

1. Open up ALL_BUILD project in **D:/CC3D_SOURCE_build/cc3d_build** - go to File->Project/Solution... and navigate to **D:/CC3D_SOURCE_build/cc3d_build** and choose ALL_BUILD:

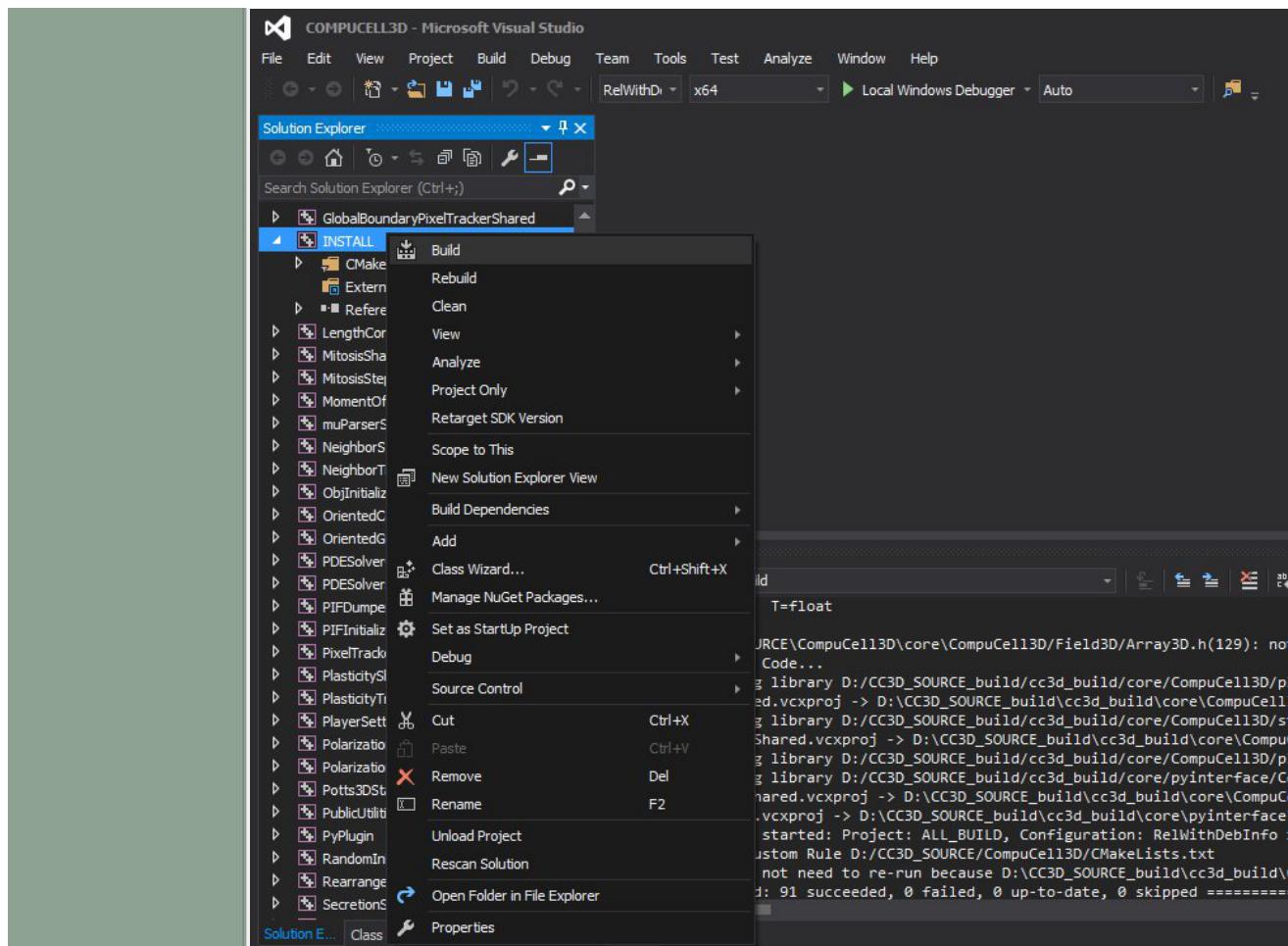




3. Right-click on the ALL_BUILD in solution explorer and choose build from the context menu to start compilation



4.Right-click on the INSTALL in solution explorer and choose build from the context menu to install CompuCell3D into d:\CompuCell3D:



It takes a while for windows to copy all the files to the installation folder but once this is done , go to the directory where you installed CC3D (in my case it was `d:/cc3d_install`) by typing at command-line prompt

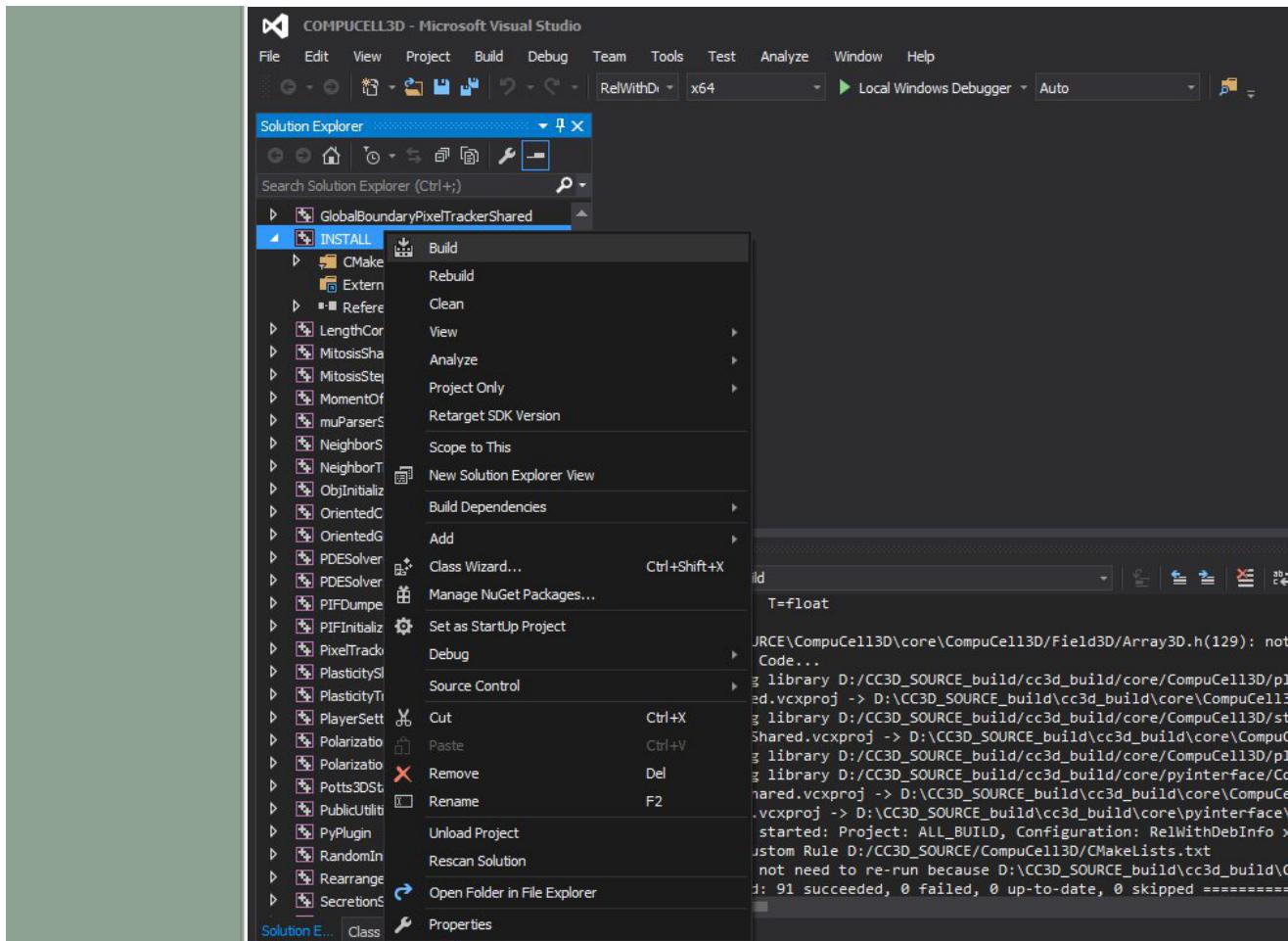
```
D:
```

followed by

```
cd cc3d_install
```

followed by `CompuCell3d` start-up script:

```
compuccell3d.bat
```



This completes manual build of CC3D on Windows using Visual Studio 2015.

Building CompuCell3D on Windows using Visual Studio 2015

Building CompuCell3D from source on Windows systems using freely available Visual Studio 2015 is fairly straightforward. We provide all dependencies so the only thing you need to do is to install Visual Studio 2015 (Community Edition is free and fully functional), [SWIG](#), [CMAKE](#) and [GIT for windows](#). The following will help you build and install CC3D on most Windows systems.

Prerequisites

Hardware

CompuCell3D may build and run with less capable hardware, but has been tested with the following:

- 512MB RAM
- Hardware 3D Graphics Acceleration (most modern graphics cards)

Build Tools and Dependencies

[SWIG](#), [CMAKE](#) and [GIT for windows](#)

- **SWIG** (version 1.3 or higher recommended) - after installing SWIG add SWIG installation directory to search path
- **CMAKE** (CMake-GUI is recommended)
- **GIT for windows**
- A directory with precompiled dependencies (that include Python and all libraries that CC3D depends on)

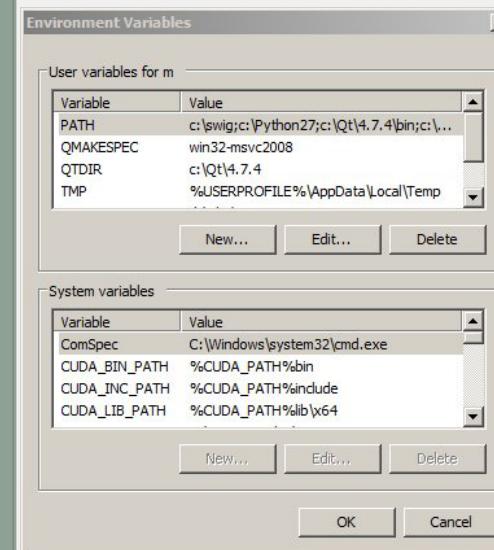
= Installing CC3D Compile Dependencies =

Go to [dependency repository](#) and download [CC3D_VS2015_dependencies.zip](#)

Unpack it to your **C:/** or **D:/** drive and make sure you have either **C:/prerequisites** or **D:/prerequisites** on your system.

Adding SWIG to search Path

In my case swig was installed to **C:/swig**. Then I go to open Control Panel » System » Advanced » Environment Variables. They are separated into user and machine specific values. You can view and edit their values there. Their current values upon launch are made available to all programs. I will add **c:\swig** to PATH variable (I am editing user environment variables, not the global ones) as shown below:

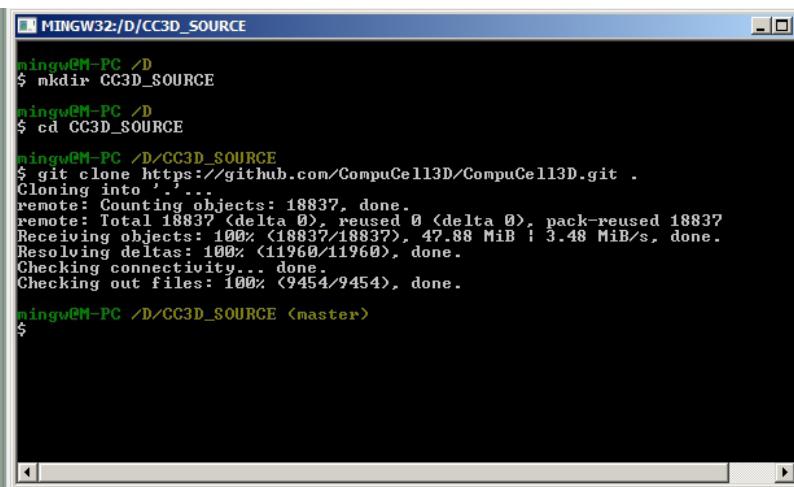


Source Code

Once the dependencies have been satisfied, obtain the source code from our GIT repository using the following command from git command-line tool:

```
git clone https://github.com/CompuCell3D/CompuCell3D.git d:\CODE_TGIT
```

Take a look at the screenshot to see how this would look:



```
mingw@M-PC ~
$ mkdir CC3D_SOURCE
mingw@M-PC ~
$ cd CC3D_SOURCE
mingw@M-PC ~/D/CC3D_SOURCE
$ git clone https://github.com/CompuCell3D/CompuCell3D.git .
Cloning into '...'...
remote: Counting objects: 18837, done.
remote: Total 18837 (delta 0), reused 0 (delta 0), pack-reused 18837
Receiving objects: 100% (18837/18837), 47.88 MiB / 3.48 MiB/s, done.
Resolving deltas: 100% (11960/11960), done.
Checking connectivity... done.
Checking out files: 100% (9454/9454), done.

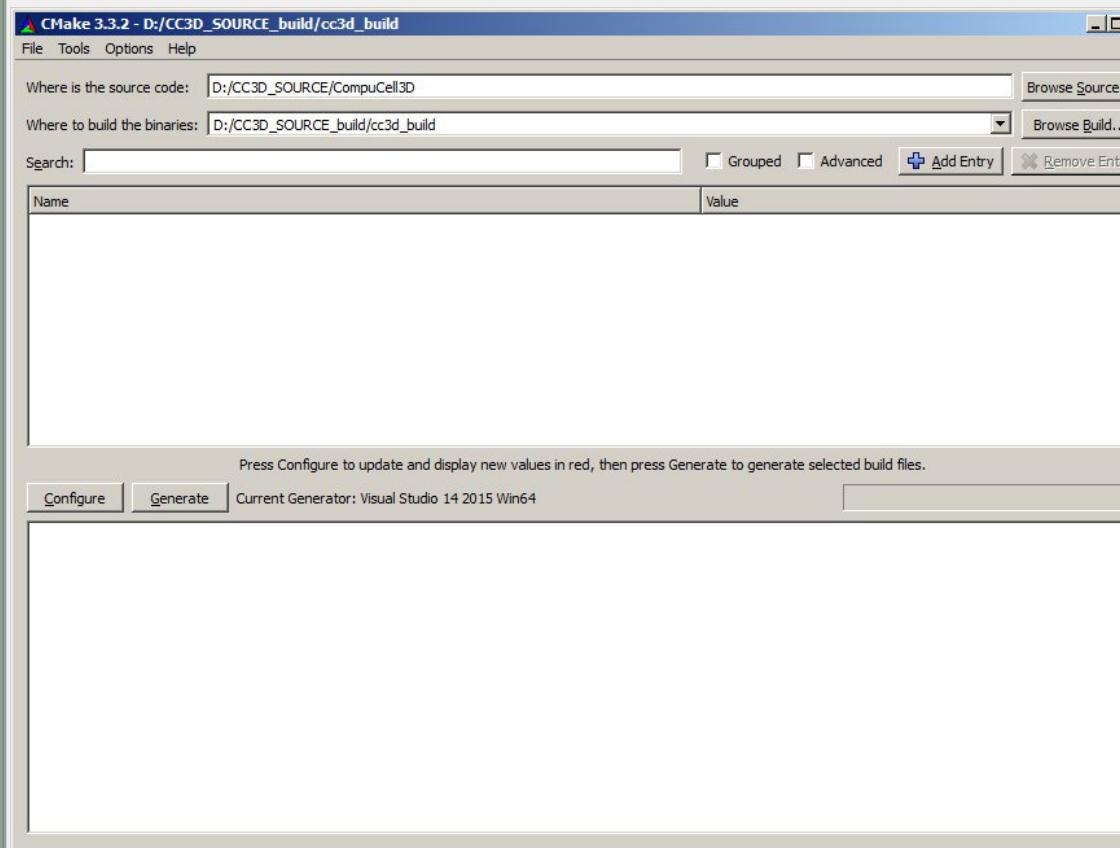
mingw@M-PC ~/D/CC3D_SOURCE (master)
$
```

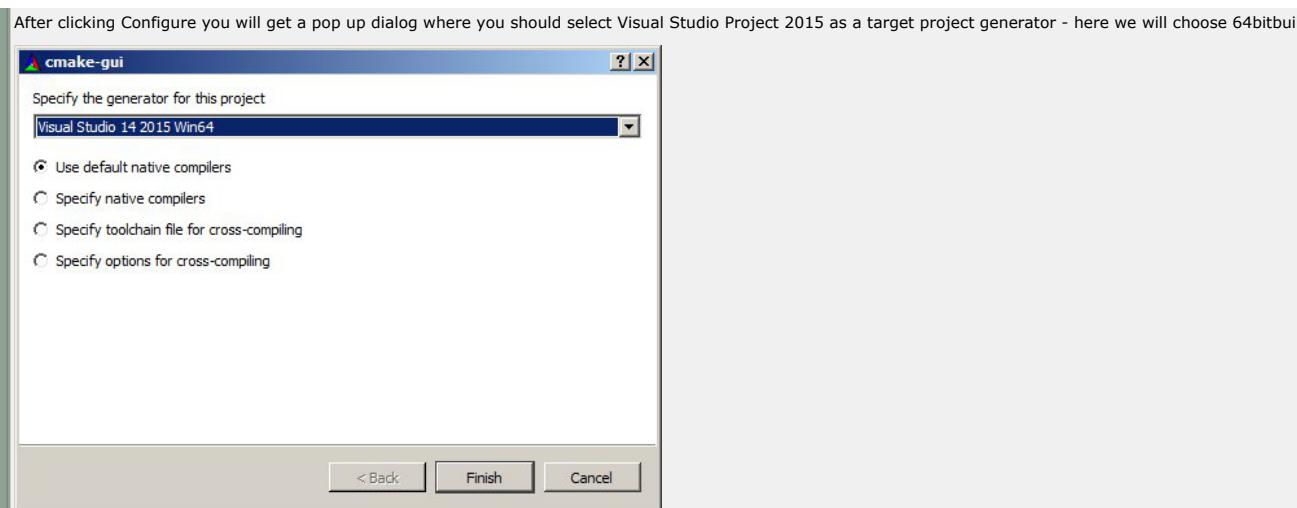
This will create a directory d:\CC3D_SOURCE` and download the source code into it.

At this point we are ready to configure CC3D for building using VS 2008

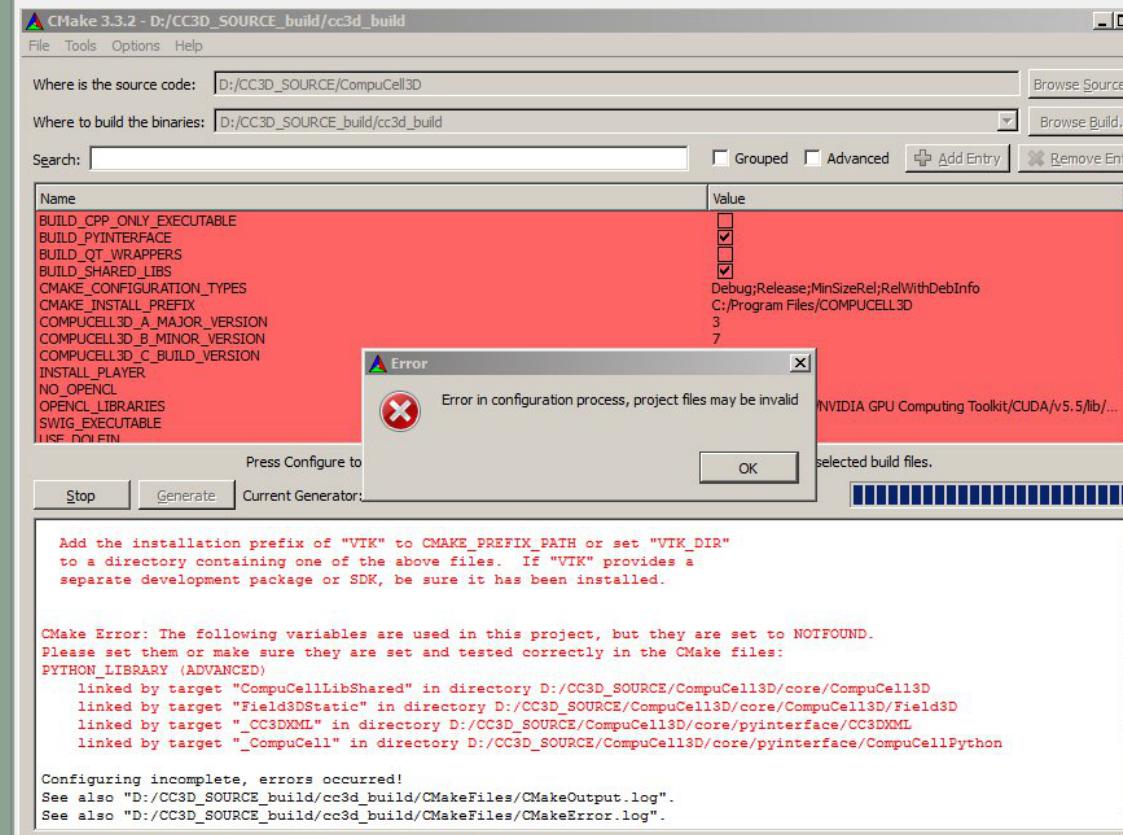
Generating VS 2015 project using CMake

CompuCell3D is configured using the CMake build system. Double click CMake icon and set source and build directories to d:\CC3D_SOURCE\CompuCell3D and D:\CC3D_SOURCE\cc3d_build as show below:

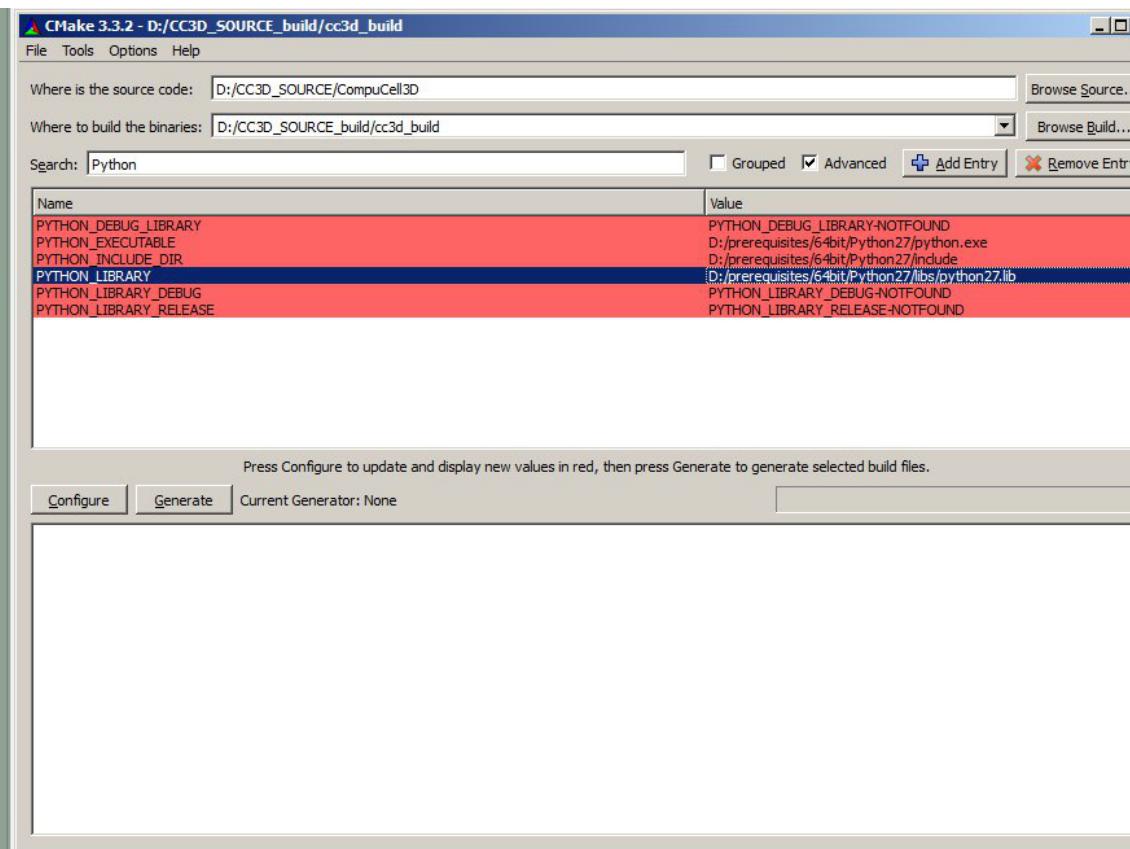




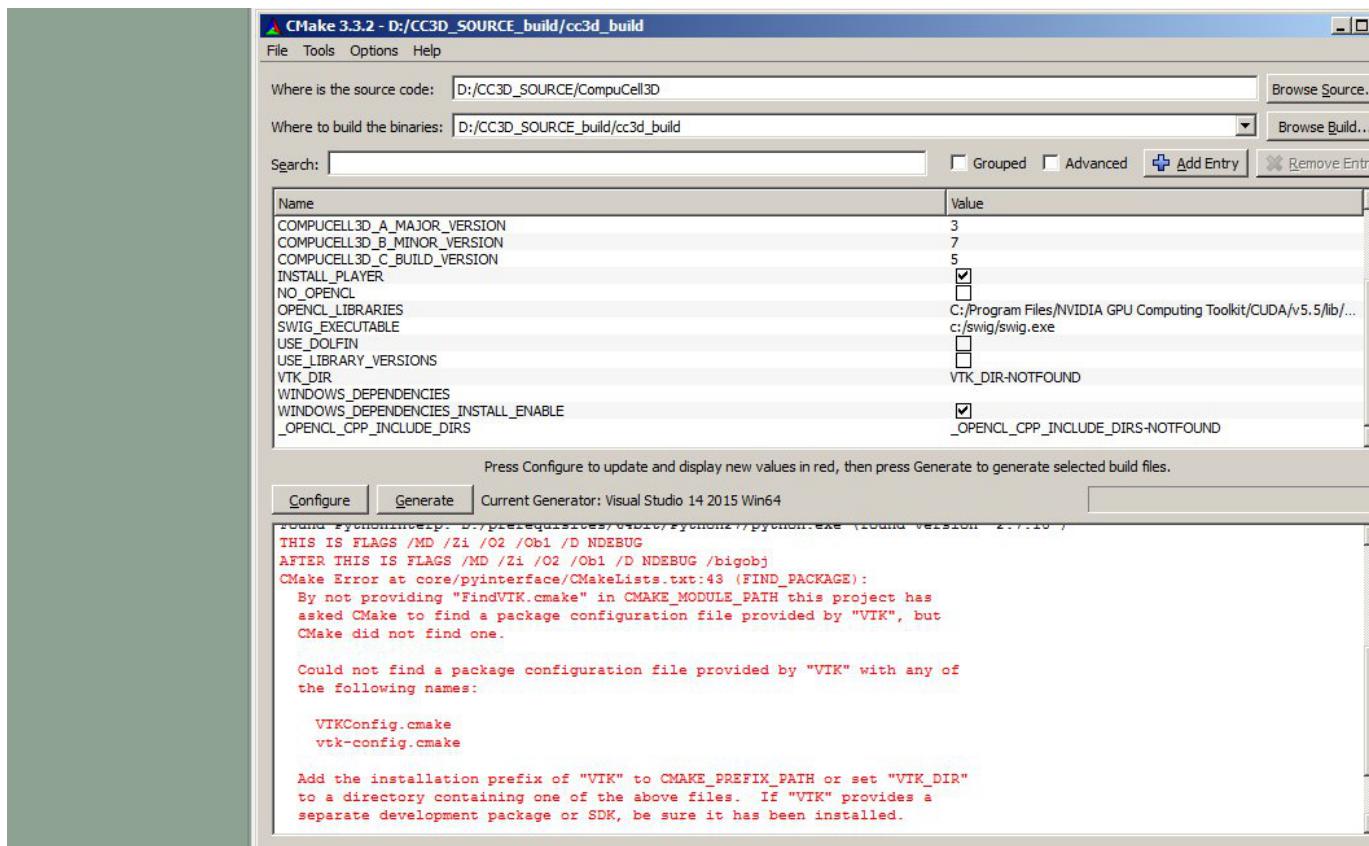
After configuration is done you will get a screen that looks somewhat similar to the one below:



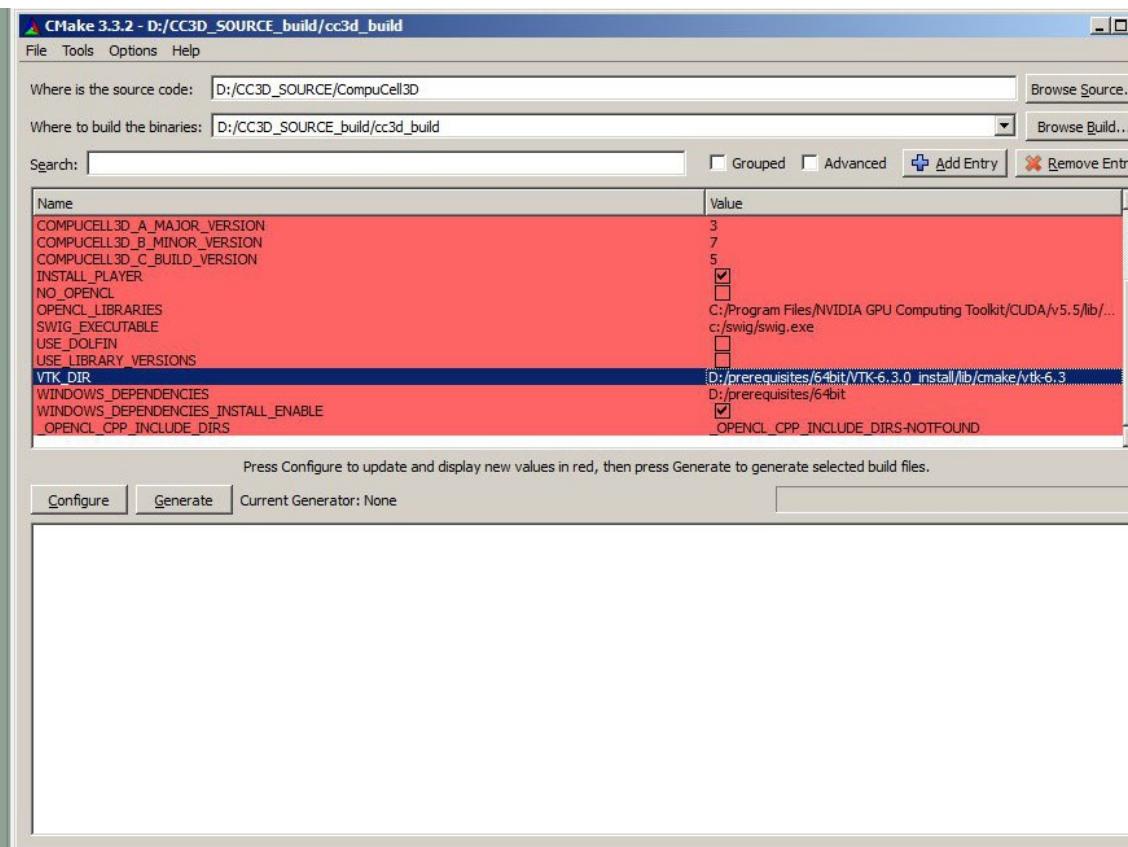
As you can see it informs you that the error has occurred. This is normal and it means we need to provide certain type of information to CMAKE. First let's set proper Python paths in the CMAKE window. Click **Advanced** check-box at the top and type Python in the search bar next to the box you have just clicked. Make sure that all Python paths are input as in the screen below. If you installed prerequisites to the **C://** drive then replace **D:/** with the **C:/** in the entries below and hit **Configure** button again



As you can see we still get errors , this time it is about missing VTK.

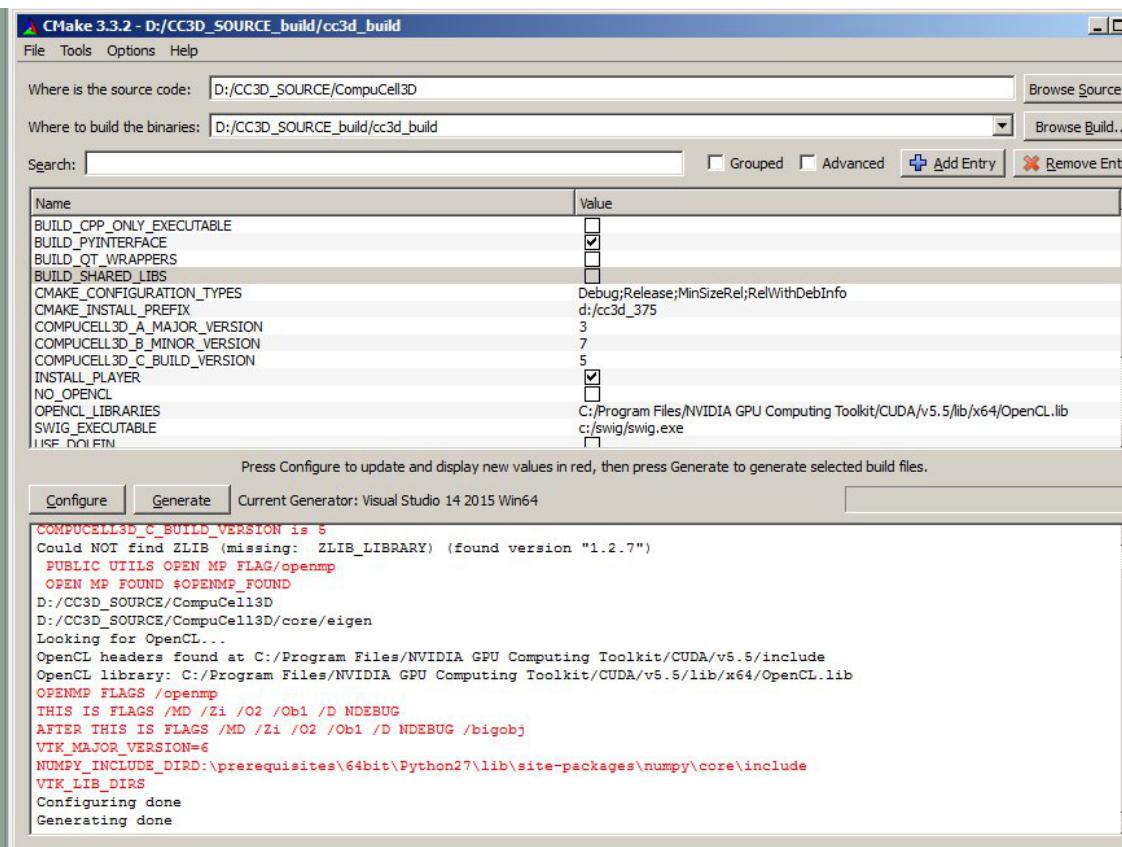


Let's fix that by pointing CMAKE to the VTK installation that is included in the **D:/prerequisites** directory. We will also change the location of Windows dependencies to point to **D:/prerequisites** so in the screen below there are two changes one in line with **VTK_DIR** and another one in the line **WINDOWS_DEPENDENCIES**:



You may check **NO_OPEN_CL** option if you do not have OPEN_CL toolkit installed

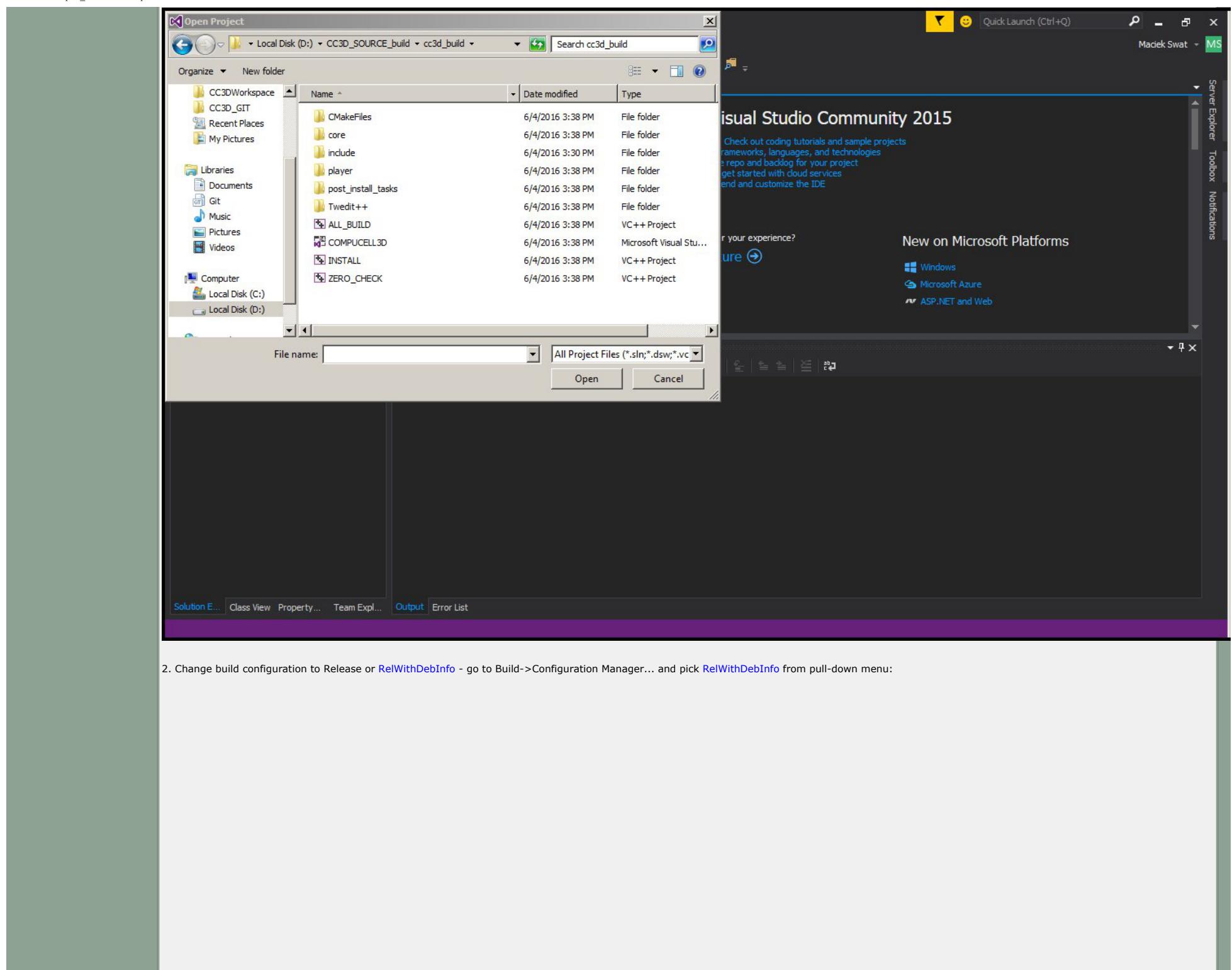
Additionally, choose the location where you want CC3D to be installed. To do this change **CMAKE_INSTALL_PREFIX** line as shown below. Here we are installing CC3D to the **D:/cc3d_375** directory. Feel free to alter this location as you see fit.



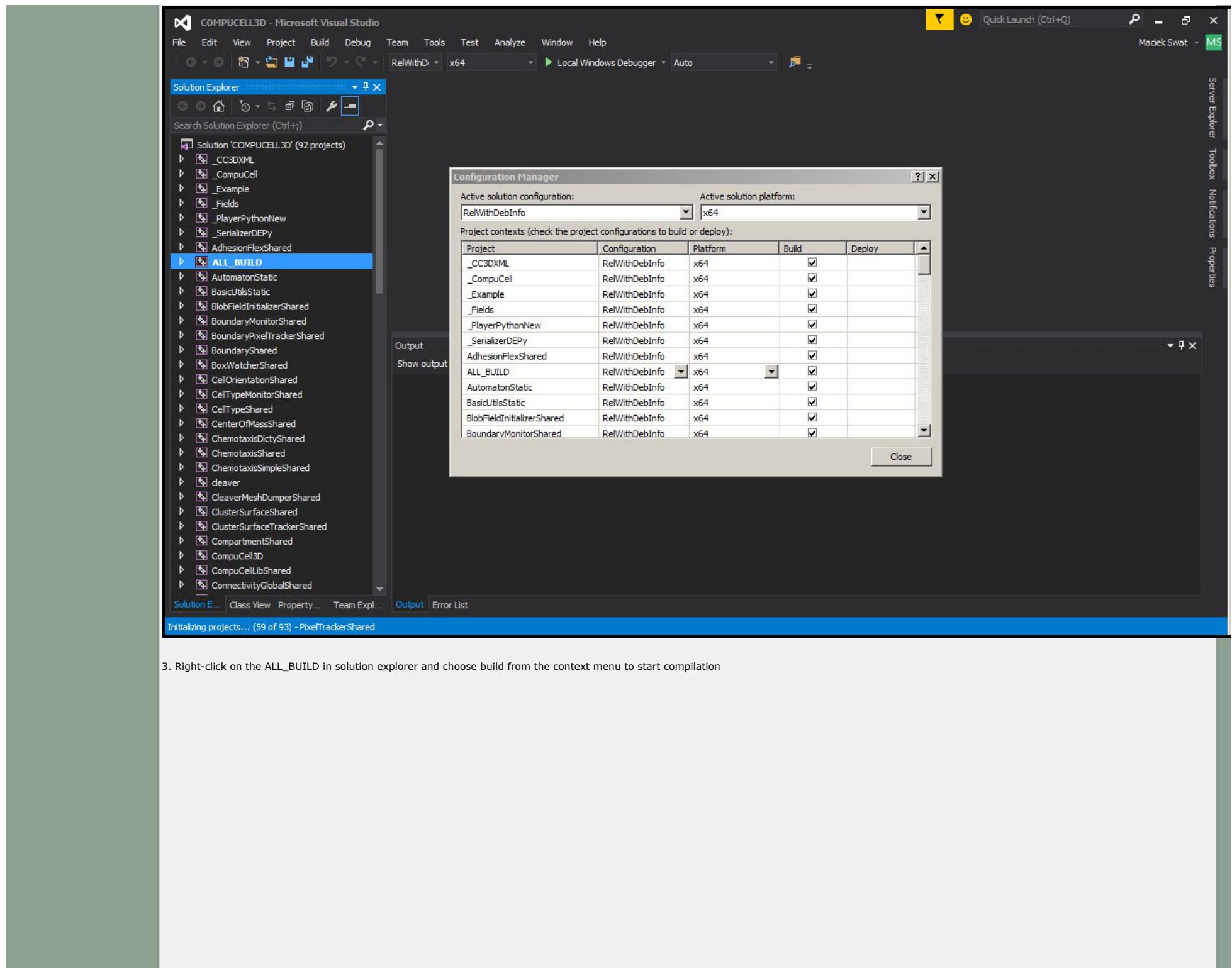
Click **Configure** , AND **Generate** buttons and then open up Visual Studio 2015.

Compilation with Visual Studio

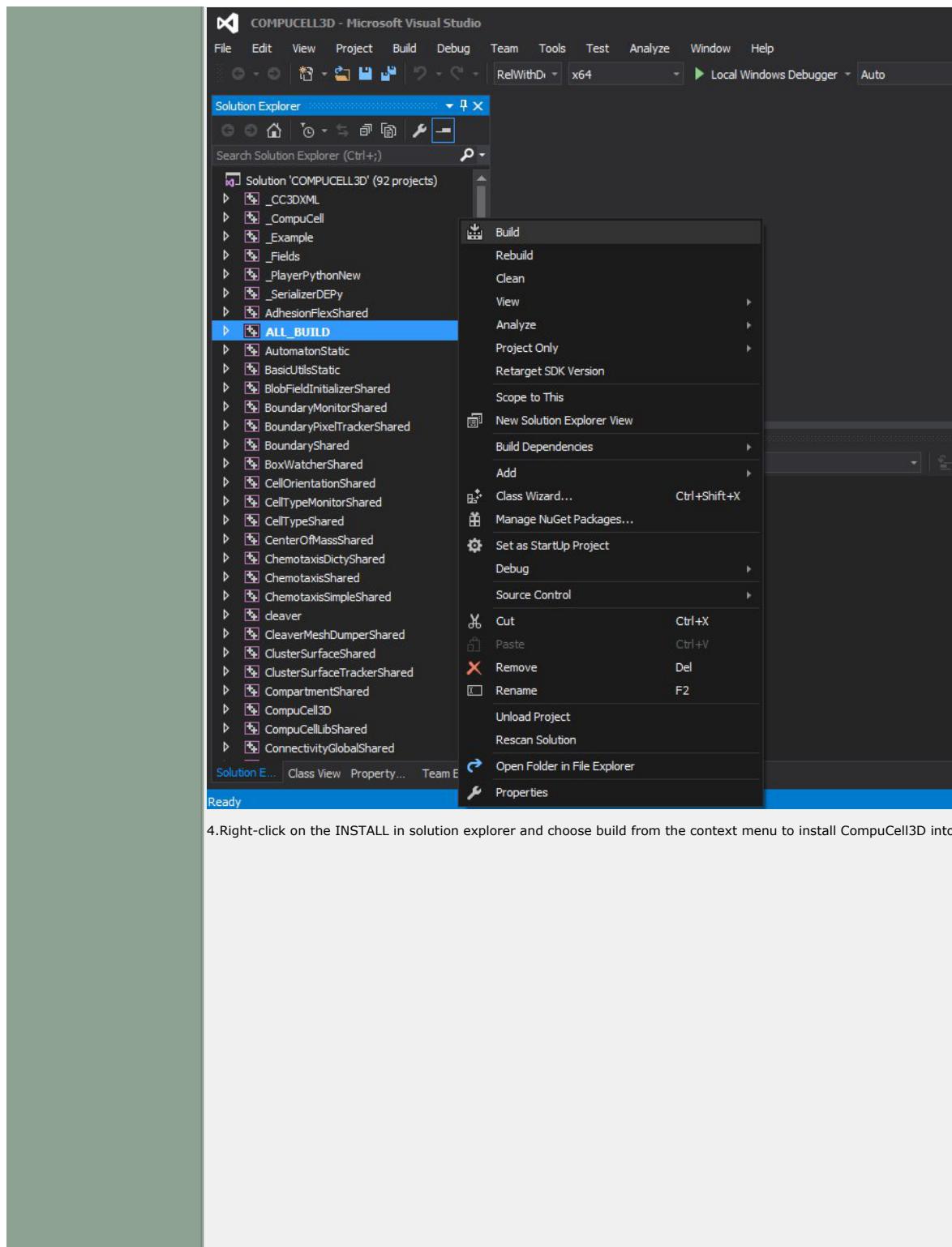
1. Open up ALL_BUILD project in **D:/CC3D_SOURCE_build/cc3d_build** - go to File->Project/Solution... and navigate to **D:/CC3D_SOURCE_build/cc3d_build** and choose ALL_BUILD:



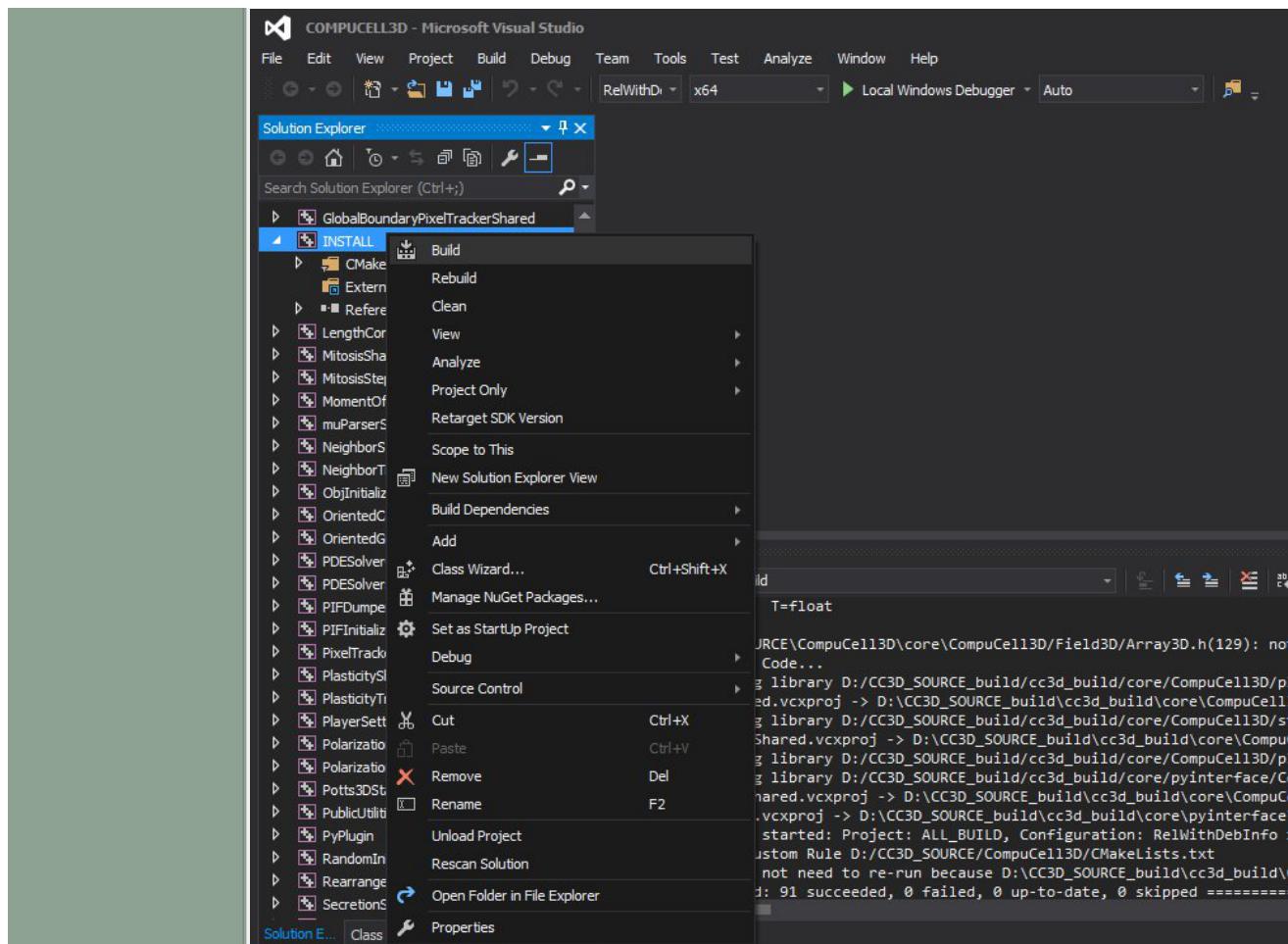
2. Change build configuration to Release or [RelWithDebInfo](#) - go to Build->Configuration Manager... and pick [RelWithDebInfo](#) from pull-down menu:



3. Right-click on the ALL_BUILD in solution explorer and choose build from the context menu to start compilation



4. Right-click on the INSTALL in solution explorer and choose build from the context menu to install CompuCell3D into d:\CompuCell3D:



It takes a while for windows to copy all the files to the installation folder but once this is done , go to the directory where you installed CC3D by typing at command-line prompt

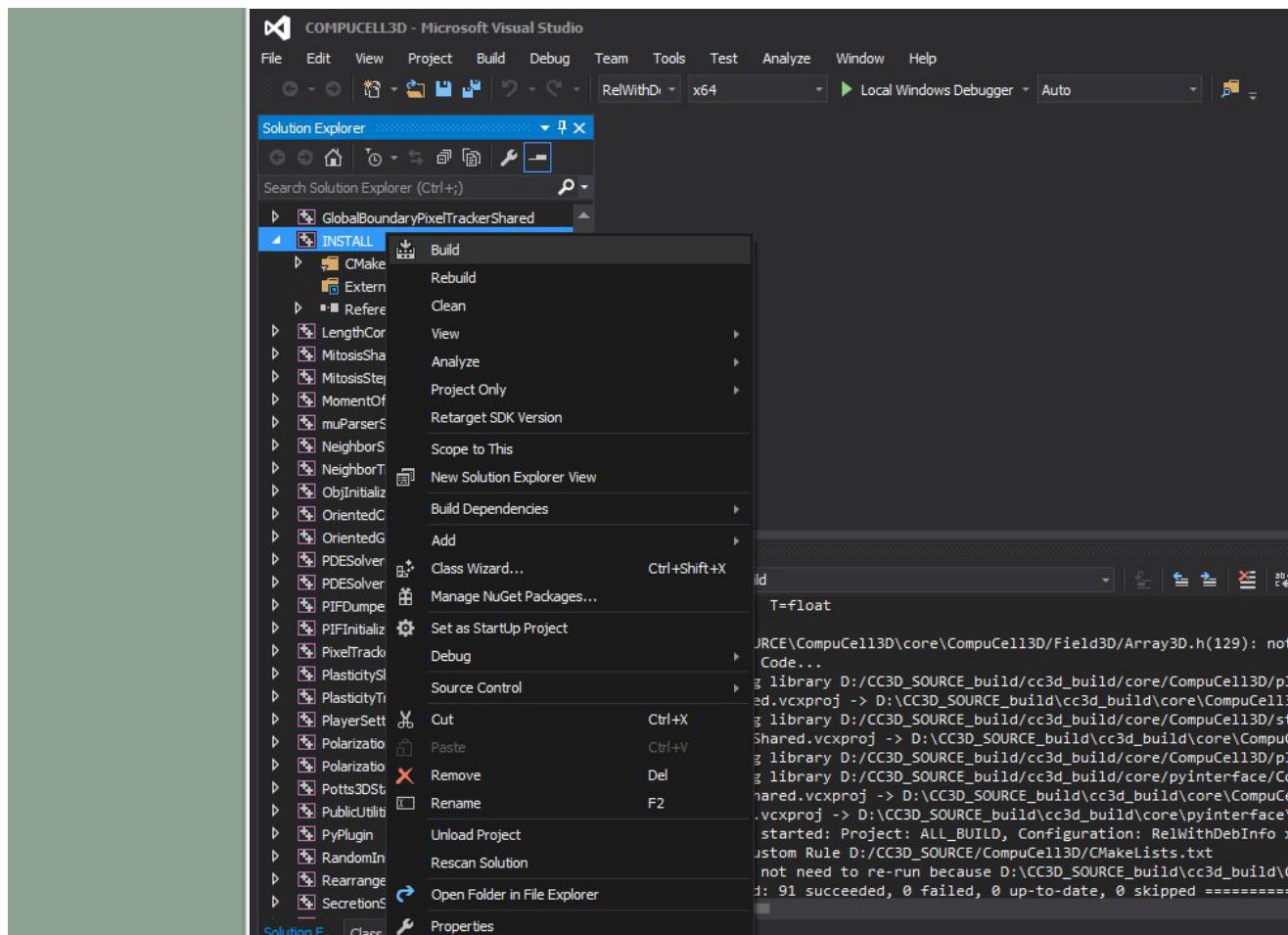
```
D:
```

followed by

```
cd CompuCell3D
```

followed by [CompuCell3d start-up script:](#)

```
compuCell3d.bat
```



This completes manual build of CC3D on Windows using Visual Studio 2015.

Maintained by [IU](#) and the [Biocomplexity Institute](#)
CC3D® and The CompuCell 3D Environment® logo are registered trademarks.