

CRIMSON Build instructions for Windows

Marija Marcan

November 27, 2019

1 Dependencies to install before building CRIMSON

Before building CRIMSON software itself you will need to download and install the following:

- Git - 2.19.2 or later (<https://git-scm.com/downloads>)
- CMake - 3.13.0 (<https://cmake.org/download/>)
- Qt - 5.7 (exactly) - best to install using Qt's Unified Windows Installer. From all the components under 5.7 you only need "msvc2013 64-bit", "Qt WebEngine" and "Qt Script"
- Visual Studio - VS2013

For installing the above software follow the instructions as provided by their publisher.

Please note that the CRIMSON build procedure has been tested and is supported for the versions of software mentioned above. While it might be possible to successfully build CRIMSON using other versions of Git, CMake or Visual Studio it is not necessary that the CRIMSON build steps as described below would be sufficient, so proceed at your own responsibility.

2 Building CRIMSON

2.1 Setup the project directory

Create a root folder in which the source code and build will be stored. The root folder for the project should be as short as possible (due to some limitations of the Windows command line). For example, a source folder like "C:\CRIMSON" and a build folder like "C:\CRIMSON\sb" should be good enough.

2.2 Get the source code via Git

Get the source code via git by cloning into the root folder selected above by following these steps:

1. Open the Windows Command Prompt (Windows - type "cmd" - press Enter).
2. Change the current working directory to the location of project root directory. e.g. for root directory "C:\CRIMSON" type

```
cd c/CRIMSON
```

into the command prompt and press Enter.

3. Type

```
git clone https://username@bitbucket.org/cafa/crimson.git .
```

(Notice the dot at the end of the above command, it is there intentionally!)
and press "Enter".

4. Enter your bitbucket password when prompted.

2.3 Generate project files with CMake

1. In the root folder of the CRIMSON project create a new folder for the build, e.g. "C:\CRIMSON\sb"
2. Open the CMake desktop app.
3. Under "Where is the source code" enter the path to the CRIMSON root folder (e.g. "C:\CRIMSON")
4. Under "Where to build the binaries" enter the path to the CRIMSON build folder you created just recently (e.g. "C:\CRIMSON\sb")

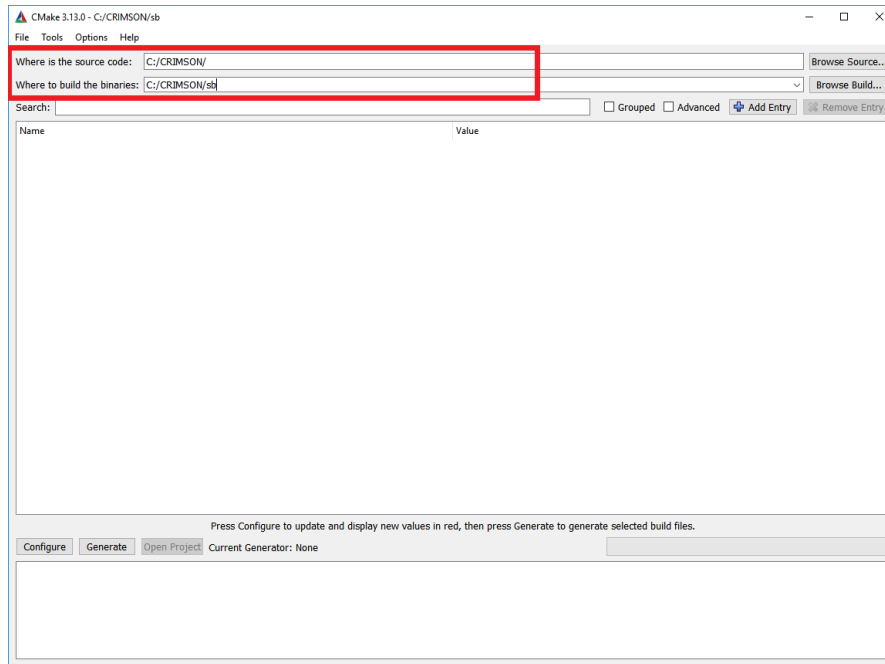


Figure 1: Specifying source and build directories

5. Press "Configure"
6. In the new window that pops up, under "Specify the generator for this project" select "Visual Studio 12 2013 Win64". Leave the remaining settings as they are and press "Finish".

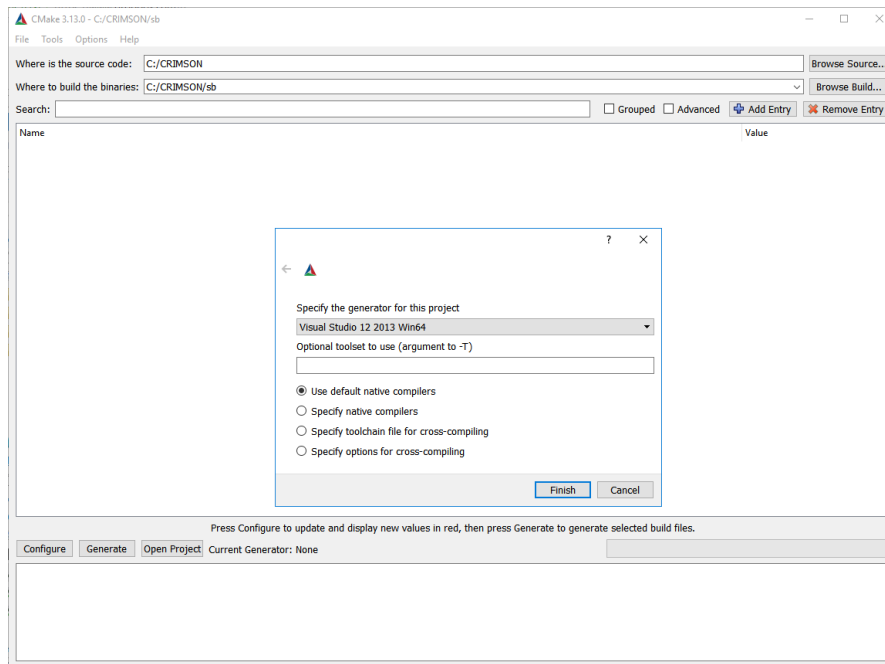


Figure 2: Specify project generator

7. An error message pops up. Press okay to continue.

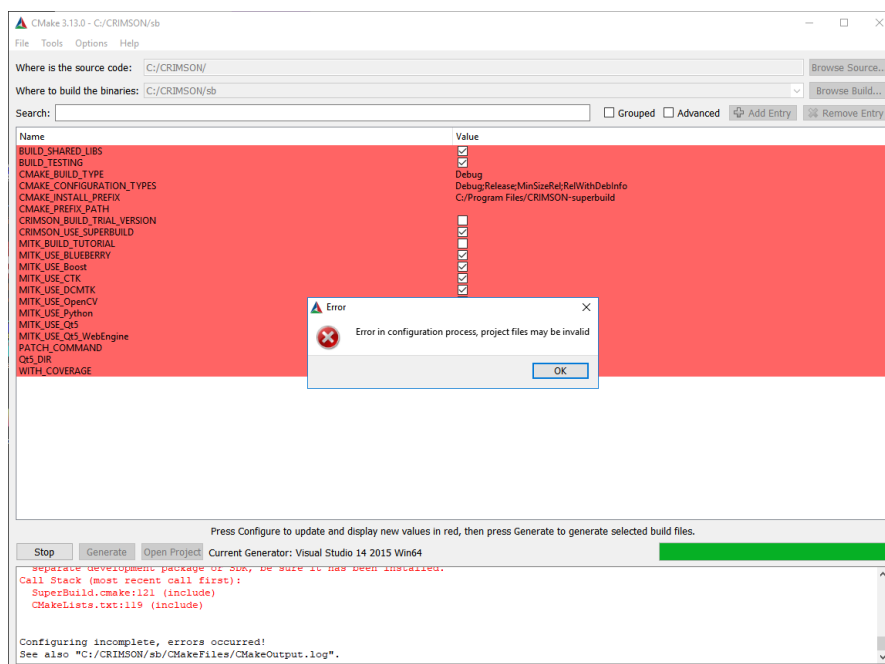


Figure 3: An expected error

8. The large window in the centre now contains pairs of variable names and values colored red, some of which need to be manually set for a successful build. Find a variable "Qt5_DIR" and set it to "C:\Qt\5.7\msvc2013_64\lib\cmake\Qt5" (assuming that you installed Qt in the default directory "C:\Qt", otherwise replace the "Qt" part by the path which you chose during Qt installation).
9. Change CMAKE_BUILD_TYPE as desired (Release, Debug or RelWithDebInfo). Leave other variables as they are.
10. Press "Configure".
11. Additional variables that can be modified appear now. You need to manually specify the locations of flowsolver and presolver:
 - For "flowsolver_folder" enter e.g. "C:\CRIMSON\sb\CMakeExternals\Source\flowsolver"
Note: This is simply a stub folder for where CRIMSON expects flowsolver files to be located. Actual flowsolver files are copied there after running the separate flowsolver installer.
 - For "presolver_executable" enter e.g. "C:\CRIMSON\sb\CMakeExternals\Source\presolver\presolver.exe". In variables above replace "C:\CRIMSON\sb" with your own custom location of CRIMSON superbuild folder as necessary.
12. Press "Configure"
13. Press "Generate"

2.4 Build your project in Visual Studio

1. Open the build folder of the project which you have previously created (in our example this is "C:\CRIMSON\sb"). The folder contains project files which were generated by CMake for our selected version of Visual Studio (2013). These Visual Studio files belong to the CRIMSON superbuild (build configuration which automatically builds CRIMSON with all of its external dependencies, one of which is MITK. Note here that the MITK itself also has its own external dependencies). Inside the overall superbuild CRIMSON project, MITK and its dependencies are being built in a superbuild project of their own (inside the folder "MITK-superbuild"). Project files for building of other external dependencies of CRIMSON are located in folder "CMakeExternals". The project files for building of CRIMSON core are in folder "CRIMSON-build".
2. In the CRIMSON superbuild folder ("C:\CRIMSON\sb") open the file named "CRIMSON-superbuild" of the type "Microsoft Visual Studio Solution". This opens the CRIMSON superbuild project in Visual Studio.
3. Make sure that the build type selected in Visual Studio matches the build type you determined in CMake (sometimes these types do not automatically match).

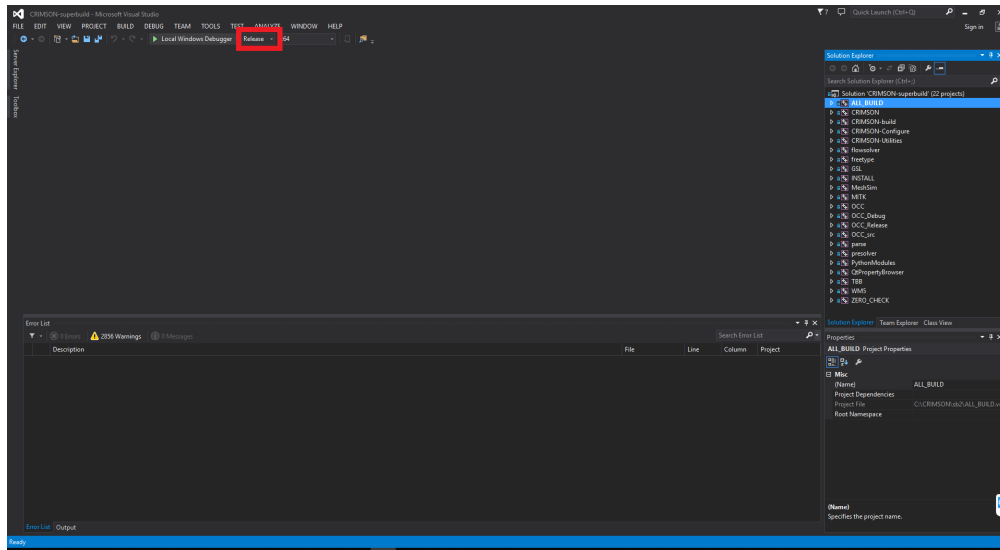


Figure 4: Build type

4. In order to avoid known collision between an "fttypes.h" header that exists under project "freetype" and in standard Windows Kit 8.1 (same name, different contents), do the following:
 - a) In the Solution Explorer pane on the right, right-click the node "freetype" and select "Properties" (last entry).
 - b) On the left side of the new window that pops up select "VC++ Directories"
 - c) Click on "Include Directories" on the right side, and then on the down arrow on the very end of that line to edit the field.
 - d) In the drop-down menu select "Edit..."

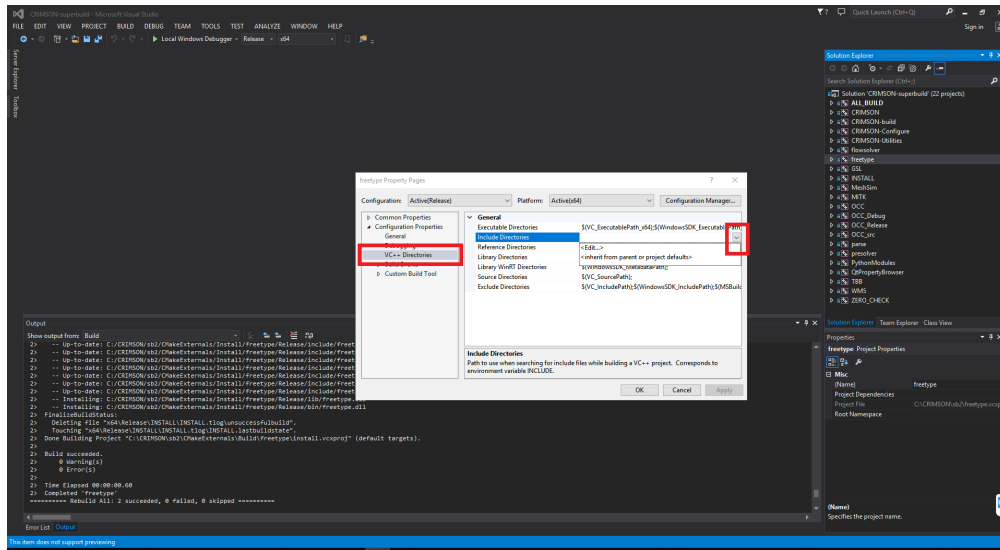


Figure 5: Project modification

- e) On the bottom of the new window, make sure the field "Inherit from parent or project defaults" is UNTICKED

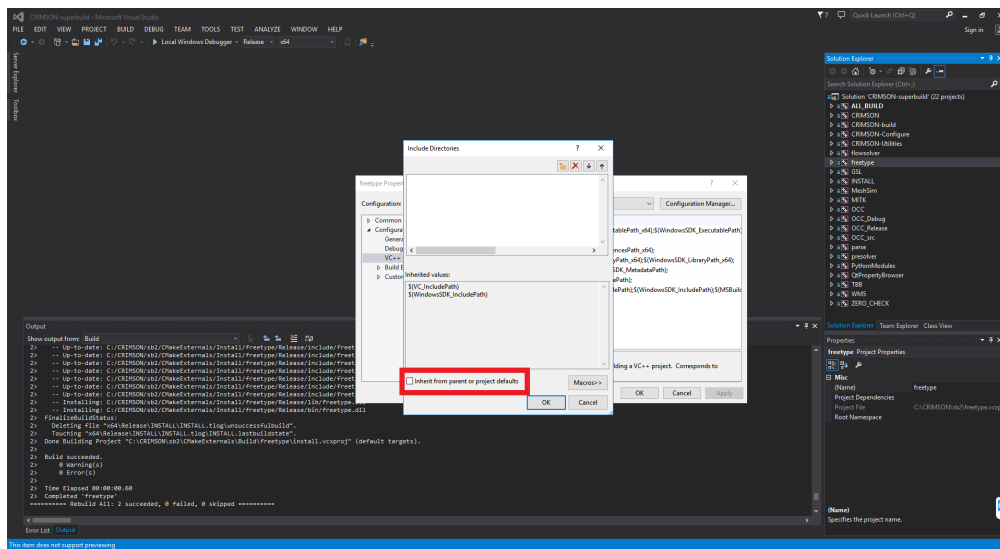


Figure 6: Project modification

5. In the Solution Explorer pane on the right, right-click the "ALL_BUILD" node and press "Build"

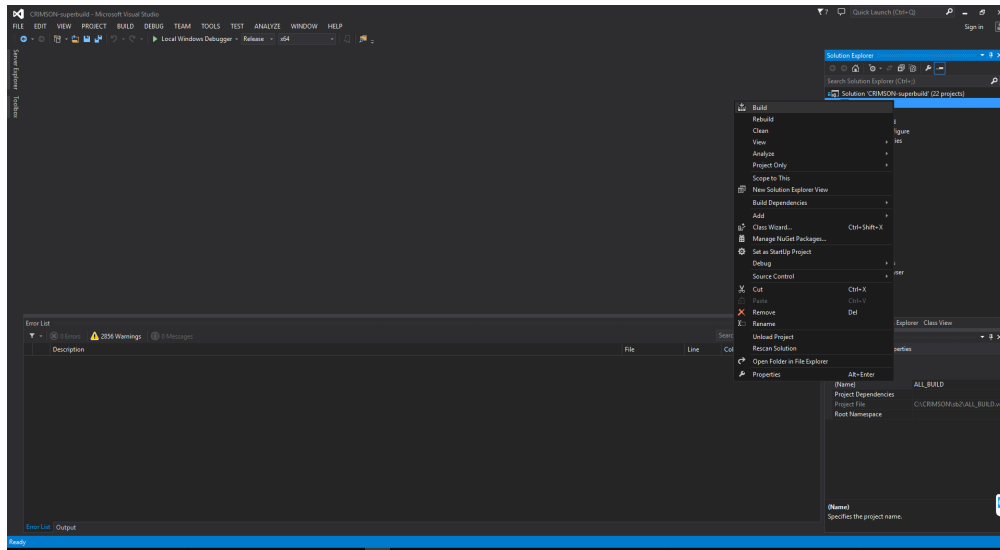


Figure 7: Building

6. Wait for all the project files to build. You can observe the progress in the "Output" window.

3 Running the built project inside Visual Studio

- In order to run CRIMSON from Visual Studio (e.g. for debugging purposes) navigate to the CRIMSON build folder ("C:\CRIMSON\sb\CRIMSON-build") and open the file named "CRIMSON" of the type "Microsoft Visual Studio Solution". This opens the CRIMSON core build project in Visual Studio.
- Right-click on the node "CRIMSON" in Solution Explorer and select "Set as StartUp project"

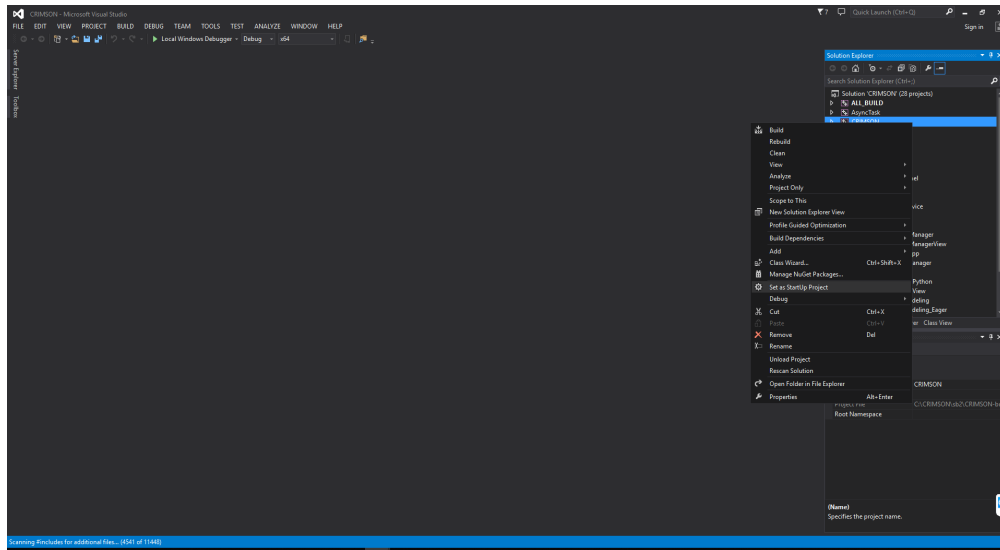


Figure 8: Set up for running

- You are now ready to run/debug CRIMSON from within Visual Studio.

4 Building CRIMSON install files (packaging)

- Open the CRIMSON core project from the CRIMSON-build folder (step 1. from the previous paragraph).
- In Solution Explorer right-click the node "PACKAGE" and select "Build".
- Wait for the project to be packaged and install files generated. You can observe the progress in the "Output" window.
- Once the packaging has successfully finished the install file can be found in the CRIMSON-build folder.

5 Adding flowsolver to your CRIMSON build

- Run the flowsolver installer.
- Make sure to specify the same folder for unpacking as the one specified under "flowsolver_folder" variable during CMake configuration.