Annex 1

ifc2indoorgml- Installation Guide

Dependencies: (the indicated versions are the tested ones, but lower versions may still work)

- CMake (>= 3.1)
- Qt5 (>=5.10)
- CGAL (>=5.3)
- OpenSceneGraph (>= 3.6)
- IFC++

For all these tools, it is critical to make sure that their system version used (32bits or 64bits) is consistent. This means that if your system is 64bits, whether you download them already compiled (binaries) or you compile them from their sources.

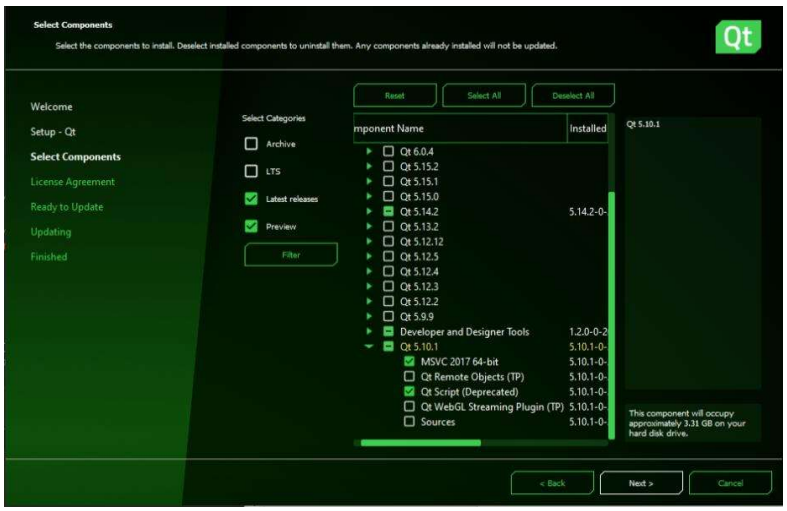
Step 1: Install CMake

https://cmake.org/download/

It is common to use cmake in the terminal, but it may be handy to have the UI, mainly for an easier configuration of the projects to build.

Step 2: Install Qt5 (<= 5.10) binaries

Go to https://www.qt.io/download-qt-installer and download the Qt download assistant. Select a version depending on your system's preferred compiler (e.g. in my case, on a Windows 10 64-bit, I use MSVC 2017 64-bit). Also, make sure to select **QtScript** among the listed components (even though it is deprecated). **Versions that do not start with 5.X.X are not part of Qt5.**



Step 3: Install CGAL (>=5.3)

https://www.cgal.org/download.html

CGAL is a header-only library, which means that you don't have to compile/build any resource to use it. Therefore, I would recommend downloading the sources on GitHub (https://github.com/CGAL/cgal/releases) – e.g. CGAL-5.3-library.zip. CGAL has some dependencies on other libraries (Qt5, GMP and MPFR). While it should automatically detect the Qt5 installed, binaries of GMP and MPFR libraries for Windows (64bits) are provided in the same GitHub page that provides the sources. More guidance on installing CGAL can be found here: https://doc.cgal.org/latest/Manual/general_intro.html.

Step 4: Install OpenSceneGraph

http://www.openscenegraph.org/index.php/download-section/stable-releases

After installing OSG, it is important to set its relevant folders in the system **environment PATHS**. Make sure you have these following properly set:

1. the **include** folder in the OSG installation folder (e.g. C:\Tools\OpenSceneGraph-3.6.0\include)
2. the folder containing the OSG **release** libraries (e.g., osg.lib). Optionally, you could add the directory with the **debug** libraries too (e.g., osgd.lib).

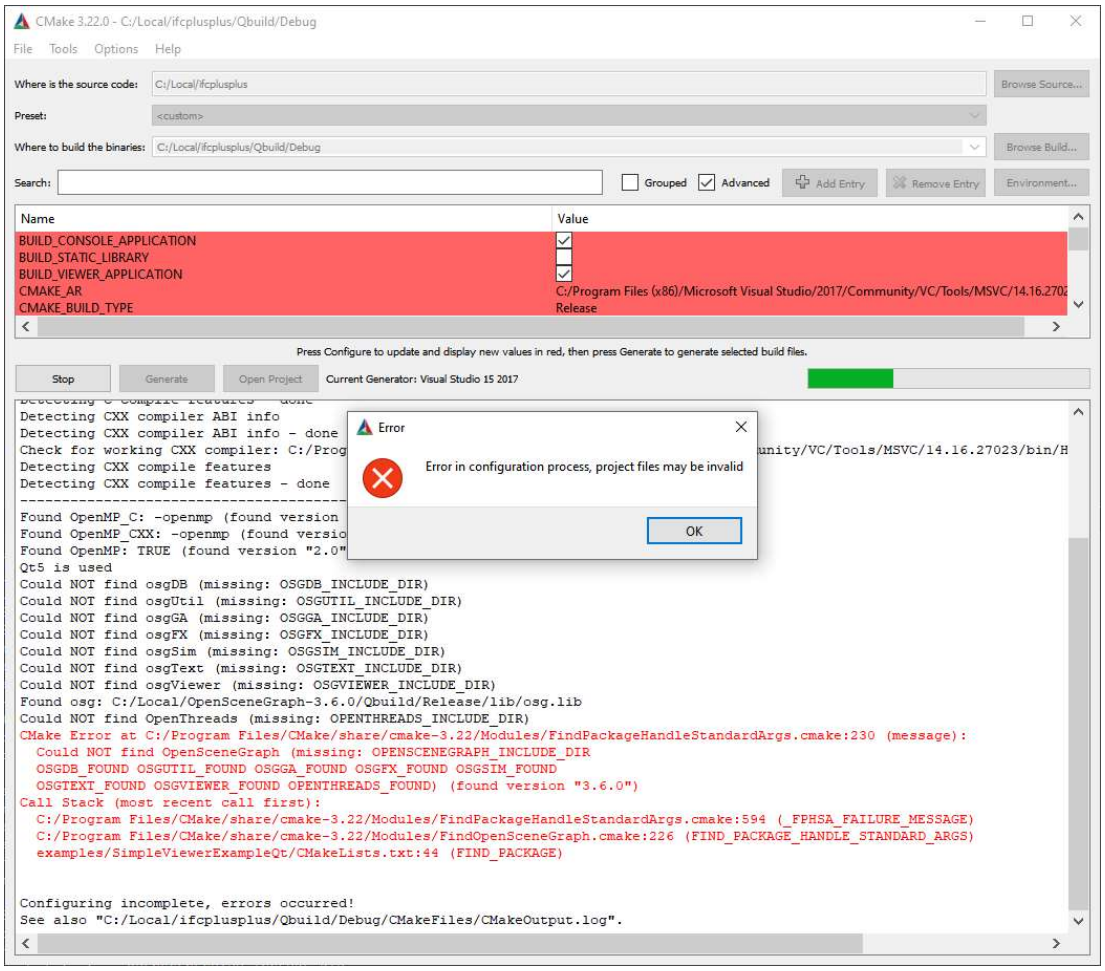
On Windows, you can simply add the corresponding directories to the "path" environment variable without having to set a specific variable name for each of them. This has not been tested for Linux and Mac, but the same behaviour is expected. Furthermore, chances are that those variables will be handled automatically if you use a package manager of your OS (e.g. apt-get install, homebrew, etc.).

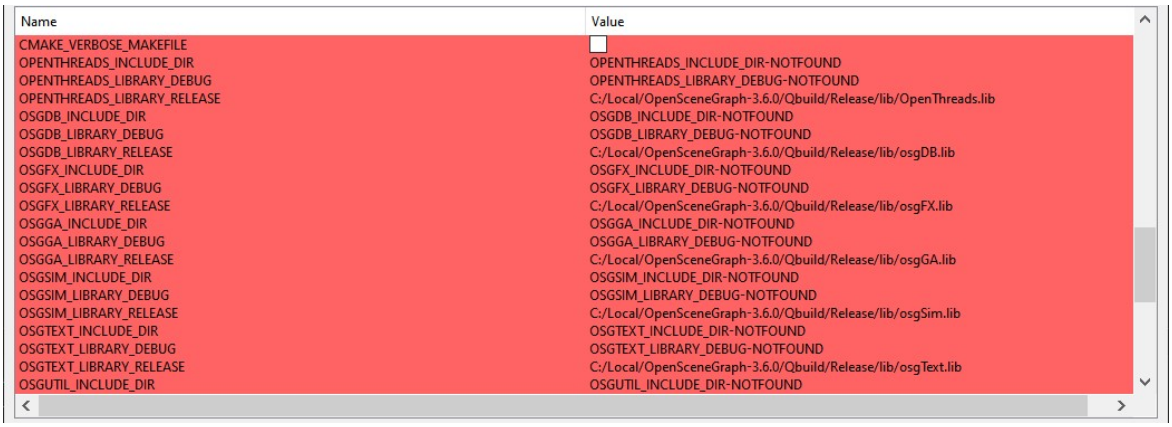
Step 5: Install IFC++

https://github.com/ifcquery/ifcplusplus

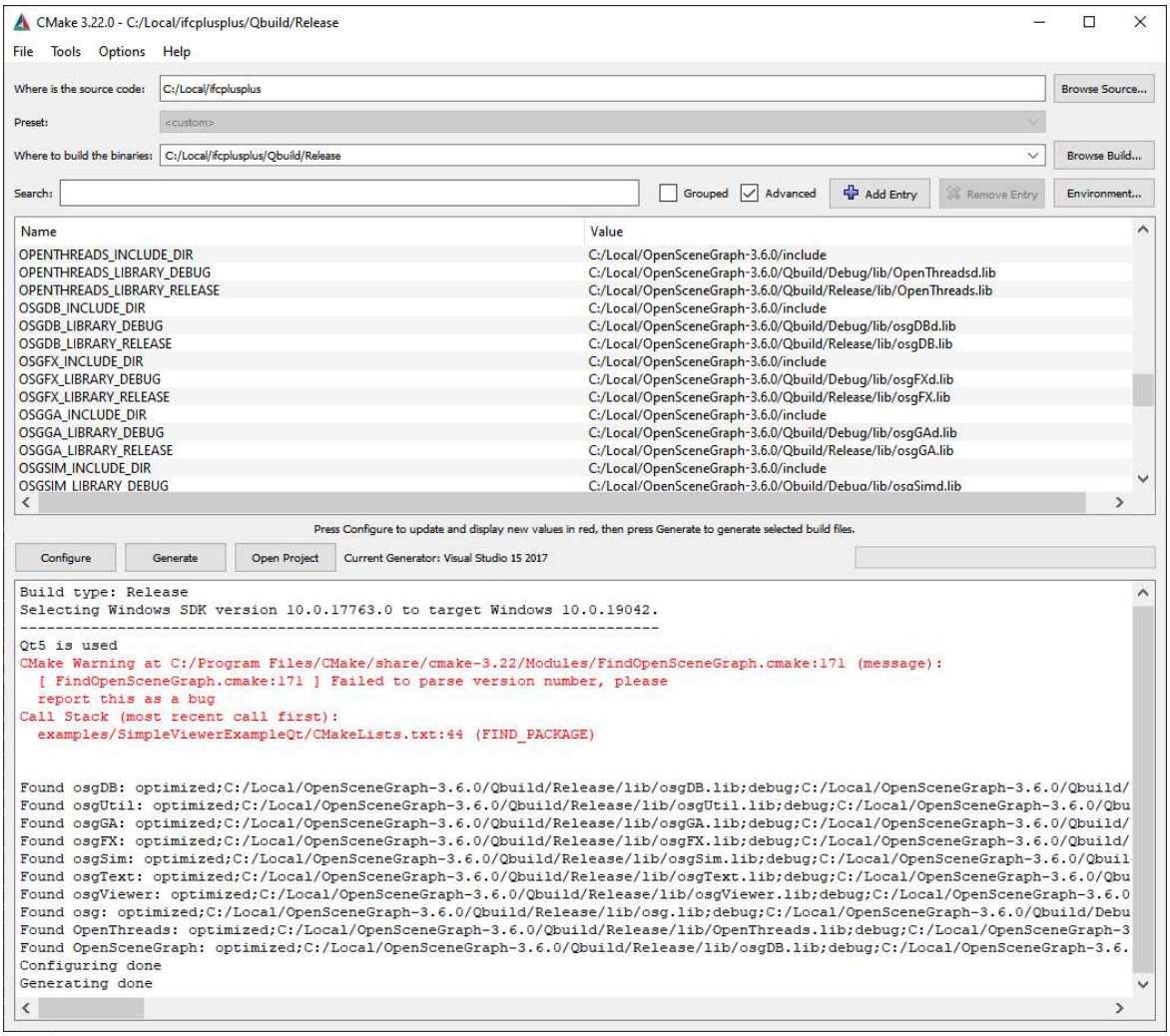
This library is not much documented unfortunately. A building/installation guide is provided (see **Build IFC++.pdf** document), but I could not successfully replicate it. Instead, after installing Qt5 and OSG (IFC++ depend on them), I could compile the two necessary components of the library: **IfcPlusPlus** and **Carve**. The CMakeLists of both components can be found in the main folder (in /IfcPlusPlus and external/Carve). If you intend to contribute to the code of the project, it may be handy to build both the Debug and Release versions of the IFC++ library, by setting up the CMAKE_BUILD_TYPE variable accordingly.

- **IfcPlusPlus**: run the CMakeLists.txt wich CMake. You may see a similar error message when some dependency components are not found:





In the cases above, Qt5 was successfully found, but not the OSG components (osgDB, osgUtil, etc.). Simply change their corresponding values to point to the right files and folders as indicated in the image below. Although Debug and Release versions of OSG are provided, only the release version would be enough, as the debugging of ifc2indoorgml would not depend on it.



Notes: The message in red in the above image is simply a warning. It will not affect the rest of the process and can be ignored. Also,

Once the configuration and generation are done with CMake, you should be able to build the IFC++ libraries and obtain the following files (on Windows): **IfcPlusPlus.dll, IfcPlusPlus.exp, IfcPlusPlus.lib.** For other OS, you may just have the **.lib** file, or **.a**. Also the files will have a 'd' at the end of their names for their Debug versions (e.g. IfcPlusPlus**d**.lib).

- Same process as above should be followed with the CMakeLists.txt file in the external/Carve folder, to obtain the single file **carve.lib** (or **carved.lib** for the Debug version).

Step 6: Install ifc2indoorgml

https://github.com/grid-unsw/ifc2indoorgml

If all the above dependencies are properly installed, it should be fairly easy to build ifc2indoorgml. A simple run of CMake should be enough to generate the sources and build them. If it fails, it is probably

because the libraries above are not visible to CMake. In that case, you can either ensure that all the folders containing the sources and the above libraries are included in the environment PATH of your system, or you could simply add them manually to CMake, just like we did in Step 5. Below is a snapshot of the folders that needed to be visible in my case for CMake to automatically configure and generate the build files:

- **Qt5**

Qt5_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5
Qt5Core_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5Core
Qt5Gui_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5Gui
Qt5OpenGL_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5OpenGL
Qt5Script_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5Script
Qt5Svg_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5Svg
Qt5Widgets_DIR	C:/Local/Qt/5.10.1/msvc2017_64/lib/cmake/Qt5Widgets

- **CGAL** (and related dependencies: Boost, GMP and MPFR; GMPXX can be ignored)

⌄ Boost	
Boost_DEBUG	☐
Boost_INCLUDE_DIR	C:/Local/boost_1_66_0
⌄ CGAL	
CGAL_Boost_USE_STATIC_LIBS	☐
CGAL_CTEST_DISPLAY_MEM_AND_TIME	☐
CGAL_DEV_MODE	☐
CGAL_DIR	C:/Local/CGAL/CGAL-5.3
CGAL_TEST_DRAW_FUNCTIONS	☐
CGAL_WITH_GMPXX	☐
› CMAKE	
⌄ GMP	
GMP_INCLUDE_DIR	C:/Local/CGAL/CGAL-5.3/auxiliary/gmp/include
GMP_LIBRARY_DEBUG	C:/Local/CGAL/CGAL-5.3/auxiliary/gmp/lib/libgmp-10.lib
GMP_LIBRARY_RELEASE	C:/Local/CGAL/CGAL-5.3/auxiliary/gmp/lib/libgmp-10.lib
› GMPXX	
⌄ MPFR	
MPFR_INCLUDE_DIR	C:/Local/CGAL/CGAL-5.3/auxiliary/gmp/include
MPFR_LIBRARIES	C:/Local/CGAL/CGAL-5.3/auxiliary/gmp/lib/libmpfr-4.lib
MPFR_LIBRARIES_DIR	C:/Local/CGAL/CGAL-5.3/auxiliary/gmp/lib

- **OSG**

▾ OPENTHREADS	
OPENTHREADS_INCLUDE_DIR	C:/Local/OpenSceneGraph-3.6.0/include
OPENTHREADS_LIBRARY_DEBUG	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Debug/lib/OpenThreadsd.lib
OPENTHREADS_LIBRARY_RELEASE	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Release/lib/OpenThreads.lib
▾ OSG	
OSG_INCLUDE_DIR	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Release/include
OSG_LIBRARY_DEBUG	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Debug/lib/osgd.lib
OSG_LIBRARY_RELEASE	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Release/lib/osg.lib
▾ OSGTEXT	
OSGTEXT_INCLUDE_DIR	C:/Local/OpenSceneGraph-3.6.0/include
OSGTEXT_LIBRARY_DEBUG	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Debug/lib/osgTextd.lib
OSGTEXT_LIBRARY_RELEASE	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Release/lib/osgText.lib
▾ OSGUTIL	
OSGUTIL_INCLUDE_DIR	C:/Local/OpenSceneGraph-3.6.0/include
OSGUTIL_LIBRARY_DEBUG	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Debug/lib/osgUtild.lib
OSGUTIL_LIBRARY_RELEASE	C:/Local/OpenSceneGraph-3.6.0/Qbuild/Release/lib/osgUtil.lib

- **IFC++**

Carve_LIBRARIES	C:/Local/ifcplusplus/Qbuild/x64/Debug/external/Carve/Debug/carved.lib
IFCPP_LIBRARIES	C:/Local/ifcplusplus/Qbuild/x64/Debug/IfcPlusPlus/Debug/IfcPlusPlusd.lib

Annex 2

ifc2indoorgml – Keyboard shortcuts

Key(s)	Description
+	Increase size of edges
-	Decrease size of edges
C	Switch clipping plane display mode
E	Toggles edges display
M	Toggles mono color
N	Inverse direction of normals
O	Toggles 2D mode only
R	Toggles random face colors
S	Switch between flat/Gouraud shading display
T	Toggles text display
U	Move camera direction upside down
V	Toggles vertices display
W	Toggles faces display
PgUp	Decrease light (all colors, use shift/alt/ctrl for one rgb component)
PgDown	Increase light (all colors, use shift/alt/ctrl for one rgb component)
Ctrl++	Increase size of vertices
Ctrl+-	Decrease size of vertices
Alt+C	Toggle clipping plane rendering on/off
Standard viewer keys	
Space	Changes camera mode (observe or fly)
A	Toggles the display of the world axis
F	Toggles the display of the FPS
G	Toggles the display of the XY grid
H	Opens this help window
Return	Starts/stops the animation
Left	Moves camera left
Up	Moves camera up
Right	Moves camera right
Down	Moves camera down
Shift+?	Toggles the display of the text
Ctrl+Q	Exits program
Alt+Return	Toggles full screen display
Camera paths are controlled using the F1..F12 keys (noted *Fx* below):	
Fx	Plays path (or resets saved position)
Alt+*Fx*	Adds a key frame to path (or defines a position)
Alt+*Fx*+*Fx*	Deletes path (or saved position)