gVirtualXRay – Installation Guide

Dr Franck P. Vidal

31th August 2016

Contents

List of Tables

1 Requirements

We rely on the CMake toolchain. To make life easy, we have decided to automatically download and compile most of the software libraries in the build tree. It is optional, of course, in case you want to use your system's libraries. The default behaviour is to let CMake download and install the dependencies. The list of tools and libraries required to build gVirtualXRay is summarised in Table 1.

Table 1: Tools and libraries required to build gVirtualXRay. Components marked with (*) cannot be installed automatically.

Name	Component	Platform	Status	URL
C++ compiler	Development tool	Linux, Mac OS X & Windows	Required	Platform dependant
CMake (*)	Development tool	Linux, Mac OS X & Windows	Required	`http://www.cmake.org/`
OpenGL (*)	Library	Linux, Mac OS X & Windows	Required	`http://www.opengl.org/`
Gzip	System tool	Linux, Mac OS X & Windows	Required	`http://www.gzip.org/`
GLEW	Library	Linux, Mac OS X & Windows	Required	`http://glew.sourceforge.net/`
Zlib	Library	Linux, Mac OS X & Windows	Required	`http://www.zlib.net/`
FreeType	Library	Linux, Mac OS X & Windows	Optional	`http://www.freetype.org/`
GCDM	Library	Linux, Mac OS X & Windows	Optional	`http://gdcm.sourceforge.net/`
LibTIFF	Library	Linux, Mac OS X & Windows	Optional	`http://www.libtiff.org/`
XCOM	Database	Linux, Mac OS X & Windows	Optional	`http://physics.nist.gov/PhysRefData/Xcom/Text/download.html`
GLFW	Library	Linux, Mac OS X & Windows	Optional	`http://www.glfw.org/`
GLUT	Library	Linux, Mac OS X & Windows	Optional	`http://freeglut.sourceforge.net/`
FLTK	Library	Linux, Mac OS X & Windows	Optional	`http://www.fltk.org/`
Qt4 (*)	Library	Linux, Mac OS X & Windows	Optional	`https://www.qt.io/`
Qt5 (*)	Library	Linux, Mac OS X & Windows	Optional	`https://www.qt.io/`
Doxygen (*)	Development tool	Linux, Mac OS X & Windows	Optional	`http://www.doxygen.org/`

1.1 Linux

1.2 Windows

1.3 Mac OS X

2 Requirements

The list of tools and libraries required to build gVirtualXRay is summarised in Table 1.

2.1 Main Library

To build the library, you will need:

-  [**cmake**]
  CMake is a cross-platform tool that we use to create the Unix Makefiles, XCode project files, MS VC++ project files, etc. for the development platform that you may wish to use.
- The GNU zip (Gzip) [**gzip**]
  Gzip is used to compress the data required by the program at run-time (e.g. OpenGL Shading Language (GLSL) source code).
- OpenGL [**opengl**]
  OpenGL is used to perform the simulation computations on the Graphics Processor Unit (GPU).

- The OpenGL Extension Wrangler Library (GLEW) [**glew**]
  GLEW is used to activate some of the OpenGL's extension that are needed by gVirtualXRay.

-  [**zlib**]
  The Zlib is used to decompress the data at run-time (the data compressed using Gzip).

- FreeType2 [**freetype**]
  It is used to display text.

2.2 Demos and tutorials

To build the demos and tutorials, you will need:

- An implementation of The OpenGL Utility Toolkit (GLUT) [**glut**] (only  [**fltk**] and  [**freeglut**]'s implementations have been tested).
  They are used to create the OpenGL's windows.

- GLFW [**glfw**] is also used to create windows. GLUT does not support OpenGL 3.x or above, GLFW does.

3 Options

The list of optional tools and libraries to build gVirtualXRay is summarised in Table 2.

3.1 Main Library

-  [**itk**]
  It is used to store 2D images or 3D volumes.

-  [**doxygen**]
  It is used to create the automatic documentation of the main library from the header files.

- XCOM: Photon Cross Sections Database [**XCOM**]
  It is used to generate accurate mass attenuation coefficients.

Table 2: Optional tools and libraries to build gVirtualXRay.

Name	Component	Platform	Status	URL
ITK	Library	Linux, Mac OS X & Windows	Optional	`http://www.itk.org/`
Doxygen	Development tool	Linux, Mac OS X & Windows	Optional	`http://www.doxygen.org/`
FreeType2	Library	Linux, Mac OS X & Windows	Optional	`http://www.freetype.org/`
XCOM	Database	Linux, Mac OS X & Windows	Optional	`http://physics.nist.gov/PhysRefData/Xcom/Text/download.html`
GLUT	Library	Linux, Mac OS X & Windows	Required	`http://www.opengl.org/resources/libraries/glut/`
FLTK	Library	Linux, Mac OS X & Windows	Optional	`http://www.fltk.org/`
freeglut	Library	Linux, Mac OS X & Windows	Optional	`http://freeglut.sourceforge.net/`
GLFW	Library	Linux, Mac OS X & Windows	Optional	`http://www.glfw.org/`

4 Compilation From the Source

4.1 Using SVN

Get the latest release from the Subversion (SVN) server:

```
$ svn checkout svn://svn.code.sf.net/p/gvirtualxray/code/trunk gvirtualxray-code
```

On Windows platforms, you may want to use TortoiseSVN [**tortoisesvn**].
Set up the development environment using CMake.

```
$ cd gvirtualxray-code
$ mkdir bin-release
$ cd bin-release
```

4.2 Build the Project With Unix Makefiles

On Unix-like platforms (inc. Linux and Mac OS X), to generate Makefiles, build the project, and install the library, type:

```
$ cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=RELEASE ../cmake
$ make
$ make install  # You might need super-user RW rights
```

4.3 Build the Project With XCode

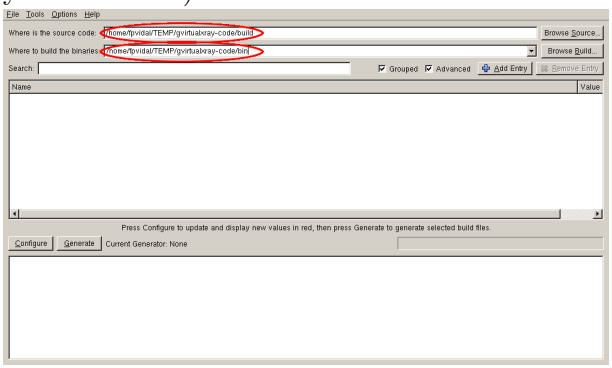
On Mac OS X platforms, to generate XCode project, type:

```
$ cmake -G "Xcode" -DCMAKE_BUILD_TYPE=RELEASE ../cmake
```

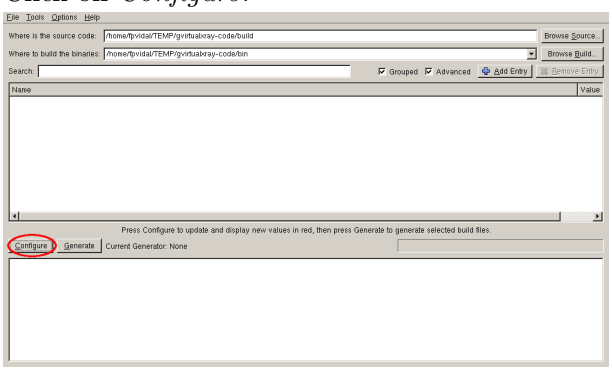
4.4 Setting up the Project Using the GUI

On all the platforms (inc. Windows), use the graphical user interface (GUI) to have more options (e.g. select different generators such as Visual C++, set variables). Then follow the instructions provided by the GUI:

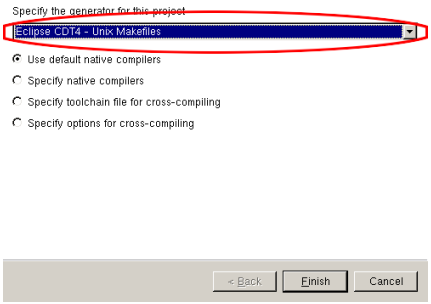
Select where the source (i.e. CMakeLists.txt) is. Select where to build the binaries. This is where the Makefiles, XCode files or Visual C++ files will be created (depending on the generator that you will select).



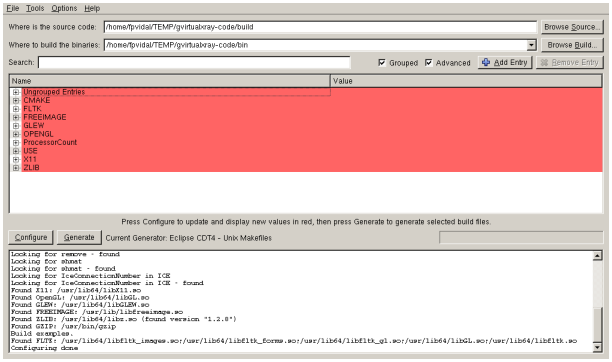
Click on *Configure*:



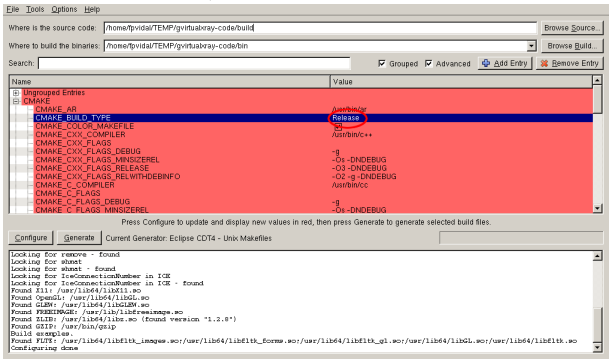
Select the generator that you wish to use:



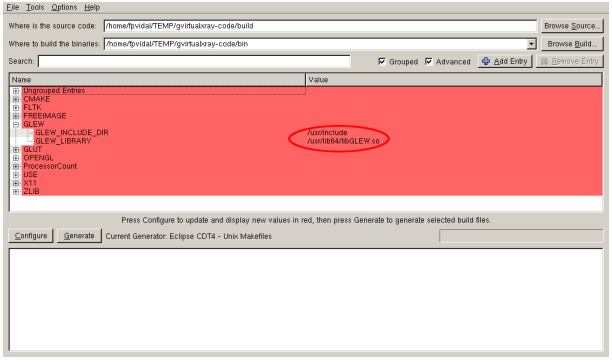
CMake will try to set up your development environment:



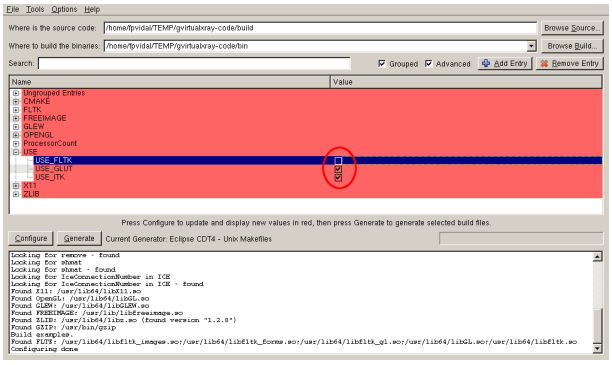
By default, the build type is Release, you can change it (with *Debug*, *MinSizeRel* or *RelWithDebInfo*):



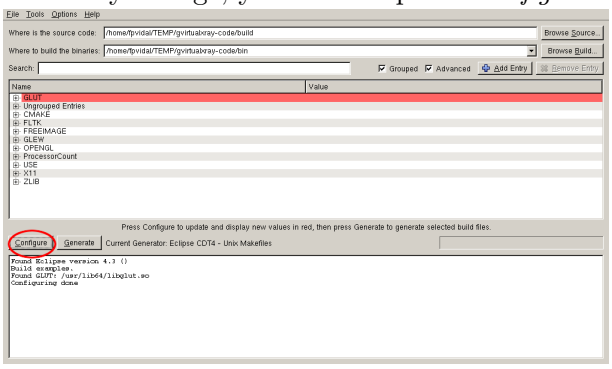
You can also manually modify or set the path to your libraries:



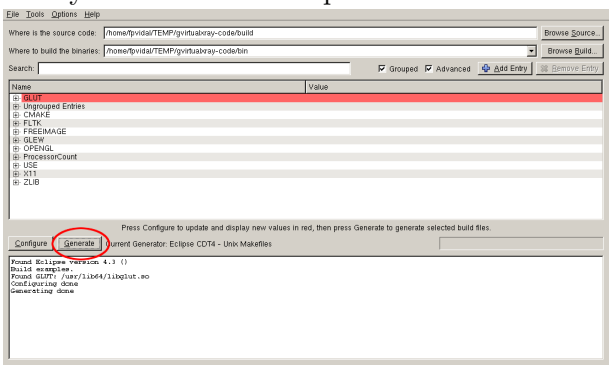
You can select different options, e.g. using Fast Light Toolkit (FLTK) or FreeGLUT (one or the other, not both), activate Insight Segmentation and Registration Toolkit (ITK):



After any change, you have to press *Configure* again so that the changes are effective:



Once you have configured the project, click on *Generate*. You are now ready to build the project with your selected development environment.



To make it a little easier on Windows platforms, it might be worth setting up environment variables to specify where you have installed some of the development tools/libraries. Here is a list of the variables that may be useful to set:

- Gzip:
  - `GZIP_TOOL`
- GLEW:
  - `GLEW_INCLUDE_DIR`
  - `GLEW_LIBRARY`
- Zlib:
  - `ZLIB_INCLUDE_DIR`
  - `ZLIB_LIBRARY`
- Boost:
  - `Boost_DIR`

- If you use FreeGLUT:
  - `GLUT_INCLUDE_DIR`
  - `GLUT_LIBRARY`
- Or if you use FLTK:
  - `FLTK_INCLUDE_DIR`
  - `FLTK_BASE_LIBRARY`
  - `FLTK_FLUID_EXECUTABLE`
  - `FLTK_FORMS_LIBRARY`
  - `FLTK_GL_LIBRARY`
  - `FLTK_IMAGES_LIBRARY`
  - `FLTK_MATH_LIBRARY`
- If you use GLFW:
  - `glfw_DIR`
- If you use ITK:
  - `ITK_DIR`
  - And maybe `VTK_DIR` if ITK was compiled with Visualization Toolkit (VTK) support
- If you use XCOM:
  - `XCOM_PATH`
- If you use FreeType:
  - `FREETYPE_INCLUDE_DIR_freetype2`
  - `FREETYPE_INCLUDE_DIR_ft2build`
  - `FREETYPE_LIBRARY`

5 Tested platforms

Successful compilation and installation have been tested on:

- Linux openSUSE 12.3 (x86_64) using the native compiler (g++ 4.7.2).
- Linux openSUSE 12.3 (x86_64) using the Windows cross compiler (mingw32-g++ 4.8.1) provided by M cross environment (MXE).
- Linux openSUSE 13.2 (x86_64) using the native compiler (g++ 4.8.3).
- Windows 8.1 (x86_64) using Visual C++ 2010 in 32 bits.
- Windows 7 (x86) using Visual C++ 2013 in 32 bits.
- Mac OS X 10.8 (x86_64) using the native compiler (LLVM 4.2) in 64 bits.
- Mac OS X 10.8 (x86_64) using the GNU compiler (g++ 4.7) in 64 bits.
- Mac OS X 10.8 (x86_64) using the Windows cross compiler (mingw32-g++ 4.8.1) provided by MXE.