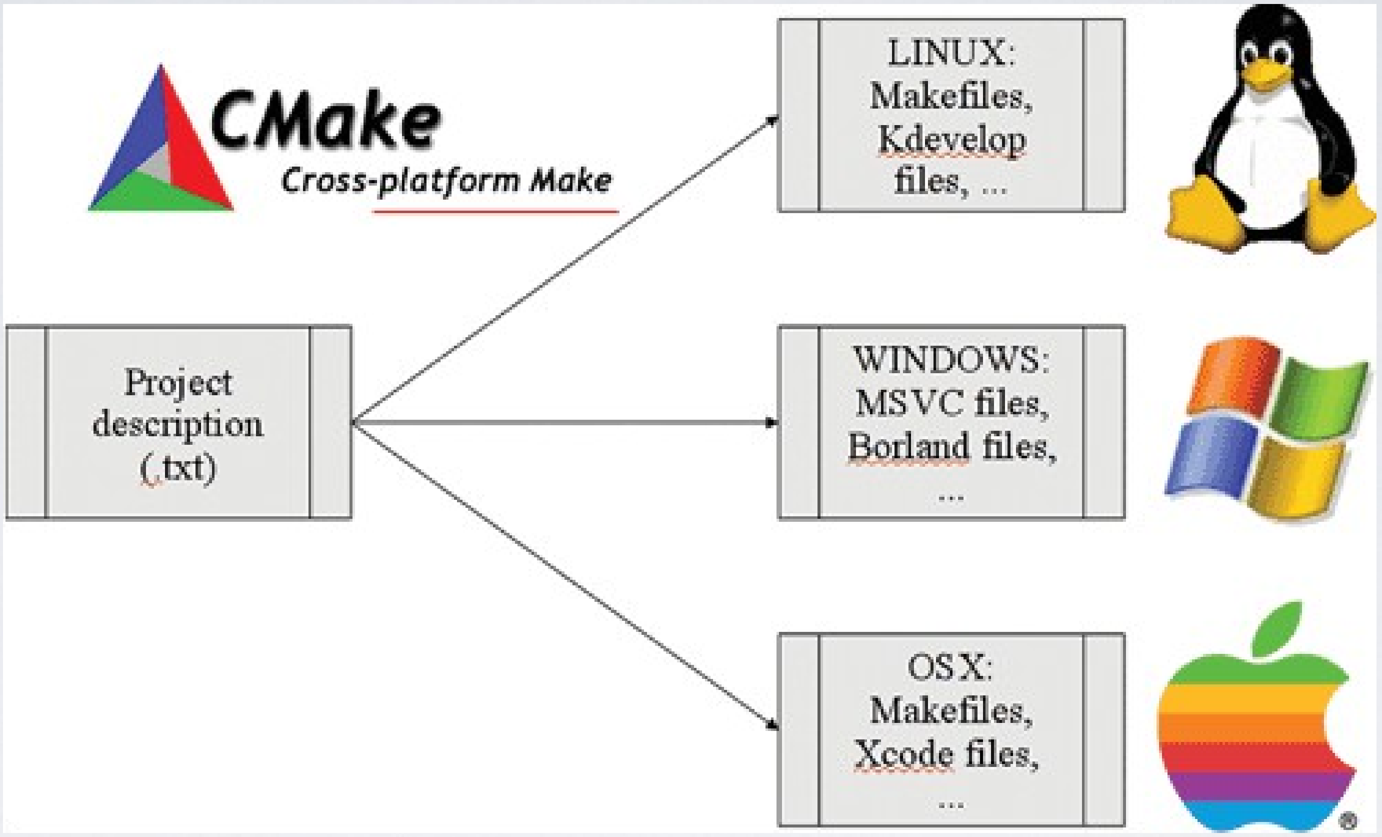
CMake

Cross-platform Make

new build system for Sofa

CMake : a replacement for qmake


CMake
Cross-platform Make
Project description (.txt)
LINUX: Makefiles, Kdevelop files, ...
WINDOWS: MSVC files, Borland files, ...
OSX: Makefiles, Xcode files, ...

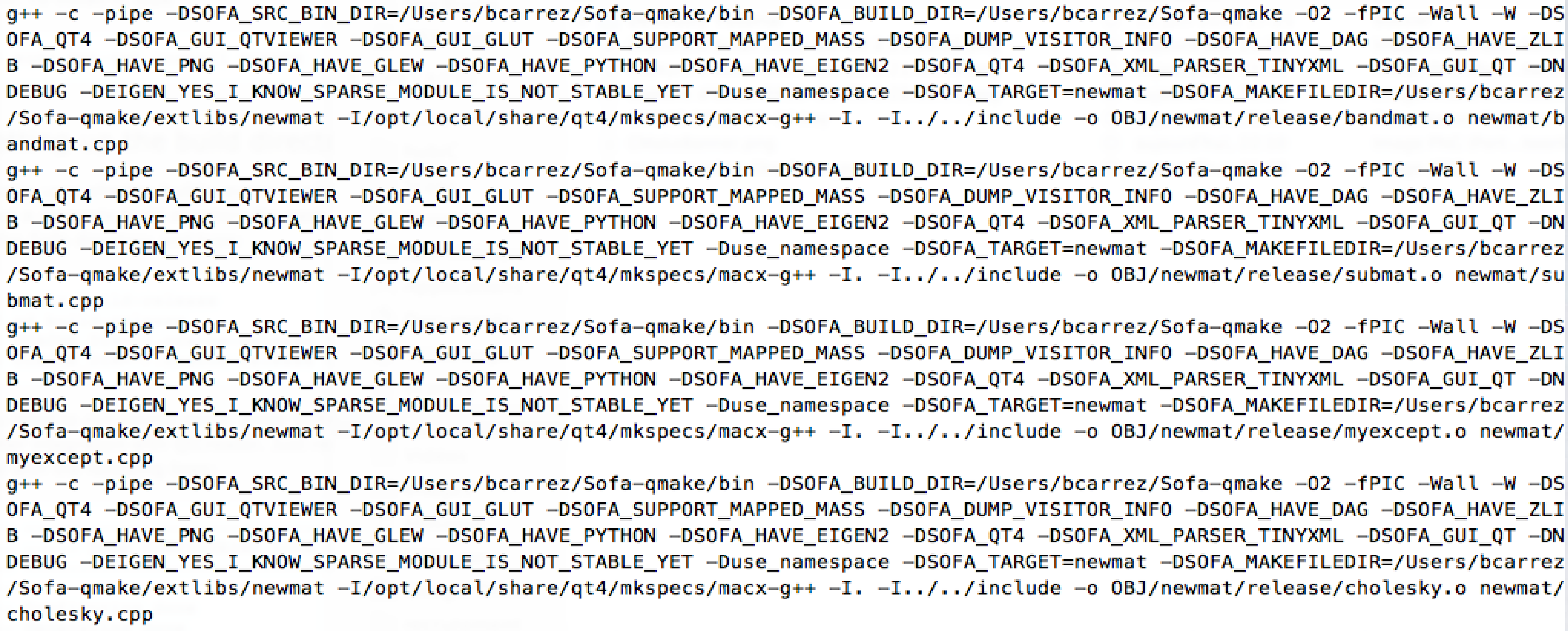
Basic build

cf. https://wiki.sofa-framework.org/wiki/CMakeBuild

checkout source
create a build directory wherever you want
run cmake
customize your setup (optional)
re-run cmake
make
done!

readability

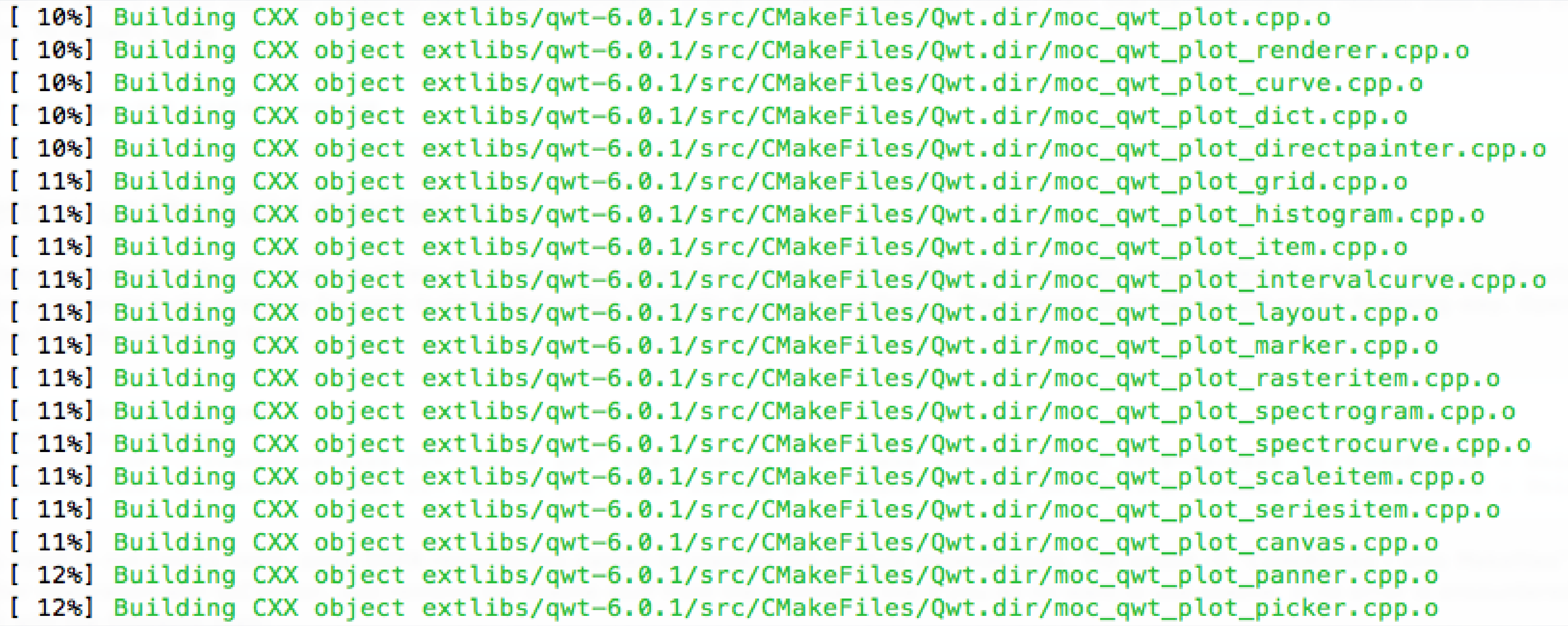
Before...



```
g++ -c -pipe -DSOFA_SRC_BIN_DIR=/Users/bcarrez/Sofa-qmake/bin -DSOFA_BUILD_DIR=/Users/bcarrez/Sofa-qmake -O2 -fPIC -Wall -W -DS
OFA_QT4 -DSOFA_GUI_QTVIEWER -DSOFA_GUI_GLUT -DSOFA_SUPPORT_MAPPED_MASS -DSOFA_DUMP_VISITOR_INFO -DSOFA_HAVE_DAG -DSOFA_HAVE_ZLI
B -DSOFA_HAVE_PNG -DSOFA_HAVE_GLEW -DSOFA_HAVE_PYTHON -DSOFA_HAVE_EIGEN2 -DSOFA_QT4 -DSOFA_XML_PARSER_TINYXML -DSOFA_GUI_QT -DN
DEBUG -DEIGEN_YES_I_KNOW_SPARSE_MODULE_IS_NOT_STABLE_YET -Duse_namespace -DSOFA_TARGET=newmat -DSOFA_MAKEFILEDIR=/Users/bcarrez
/Sofa-qmake/extlibs/newmat -I/opt/local/share/qt4/mkspecs/macx-g++ -I. -I../../include -o OBJ/newmat/release/bandmat.o newmat/b
andmat.cpp
g++ -c -pipe -DSOFA_SRC_BIN_DIR=/Users/bcarrez/Sofa-qmake/bin -DSOFA_BUILD_DIR=/Users/bcarrez/Sofa-qmake -O2 -fPIC -Wall -W -DS
OFA_QT4 -DSOFA_GUI_QTVIEWER -DSOFA_GUI_GLUT -DSOFA_SUPPORT_MAPPED_MASS -DSOFA_DUMP_VISITOR_INFO -DSOFA_HAVE_DAG -DSOFA_HAVE_ZLI
B -DSOFA_HAVE_PNG -DSOFA_HAVE_GLEW -DSOFA_HAVE_PYTHON -DSOFA_HAVE_EIGEN2 -DSOFA_QT4 -DSOFA_XML_PARSER_TINYXML -DSOFA_GUI_QT -DN
DEBUG -DEIGEN_YES_I_KNOW_SPARSE_MODULE_IS_NOT_STABLE_YET -Duse_namespace -DSOFA_TARGET=newmat -DSOFA_MAKEFILEDIR=/Users/bcarrez
/Sofa-qmake/extlibs/newmat -I/opt/local/share/qt4/mkspecs/macx-g++ -I. -I../../include -o OBJ/newmat/release/submat.o newmat/su
bmat.cpp
g++ -c -pipe -DSOFA_SRC_BIN_DIR=/Users/bcarrez/Sofa-qmake/bin -DSOFA_BUILD_DIR=/Users/bcarrez/Sofa-qmake -O2 -fPIC -Wall -W -DS
OFA_QT4 -DSOFA_GUI_QTVIEWER -DSOFA_GUI_GLUT -DSOFA_SUPPORT_MAPPED_MASS -DSOFA_DUMP_VISITOR_INFO -DSOFA_HAVE_DAG -DSOFA_HAVE_ZLI
B -DSOFA_HAVE_PNG -DSOFA_HAVE_GLEW -DSOFA_HAVE_PYTHON -DSOFA_HAVE_EIGEN2 -DSOFA_QT4 -DSOFA_XML_PARSER_TINYXML -DSOFA_GUI_QT -DN
DEBUG -DEIGEN_YES_I_KNOW_SPARSE_MODULE_IS_NOT_STABLE_YET -Duse_namespace -DSOFA_TARGET=newmat -DSOFA_MAKEFILEDIR=/Users/bcarrez
/Sofa-qmake/extlibs/newmat -I/opt/local/share/qt4/mkspecs/macx-g++ -I. -I../../include -o OBJ/newmat/release/myexcept.o newmat/
myexcept.cpp
g++ -c -pipe -DSOFA_SRC_BIN_DIR=/Users/bcarrez/Sofa-qmake/bin -DSOFA_BUILD_DIR=/Users/bcarrez/Sofa-qmake -O2 -fPIC -Wall -W -DS
OFA_QT4 -DSOFA_GUI_QTVIEWER -DSOFA_GUI_GLUT -DSOFA_SUPPORT_MAPPED_MASS -DSOFA_DUMP_VISITOR_INFO -DSOFA_HAVE_DAG -DSOFA_HAVE_ZLI
B -DSOFA_HAVE_PNG -DSOFA_HAVE_GLEW -DSOFA_HAVE_PYTHON -DSOFA_HAVE_EIGEN2 -DSOFA_QT4 -DSOFA_XML_PARSER_TINYXML -DSOFA_GUI_QT -DN
DEBUG -DEIGEN_YES_I_KNOW_SPARSE_MODULE_IS_NOT_STABLE_YET -Duse_namespace -DSOFA_TARGET=newmat -DSOFA_MAKEFILEDIR=/Users/bcarrez
/Sofa-qmake/extlibs/newmat -I/opt/local/share/qt4/mkspecs/macx-g++ -I. -I../../include -o OBJ/newmat/release/cholesky.o newmat/
cholesky.cpp
```

readability

…and now :



```
[ 10%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot.cpp.o
[ 10%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_renderer.cpp.o
[ 10%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_curve.cpp.o
[ 10%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_dict.cpp.o
[ 10%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_directpainter.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_grid.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_histogram.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_item.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_intervalcurve.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_layout.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_marker.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_rasteritem.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_spectrogram.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_spectrocurve.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_scaleitem.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_seriesitem.cpp.o
[ 11%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_canvas.cpp.o
[ 12%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_panner.cpp.o
[ 12%] Building CXX object extlibs/qwt-6.0.1/src/CMakeFiles/Qwt.dir/moc_qwt_plot_picker.cpp.o
```

SHADOW builds

- ✓ all the objet files, libs, binaries... are generated in a separate tree
- ✓ Keeps the source tree clean !
- ✓ Multiple build configuration on the same machine, using the same source (32/64 bits, float/double, basic setup/lot of plugins, gcc/llvm, etc)
- ✓ keep a working compiled version left

Configuration

- ✓ No more sofa-local.cfg stuff…
- ✓ All options saved in the CMakeCache.txt file at the build root
- ✓ A different configuration for each shadow build directory

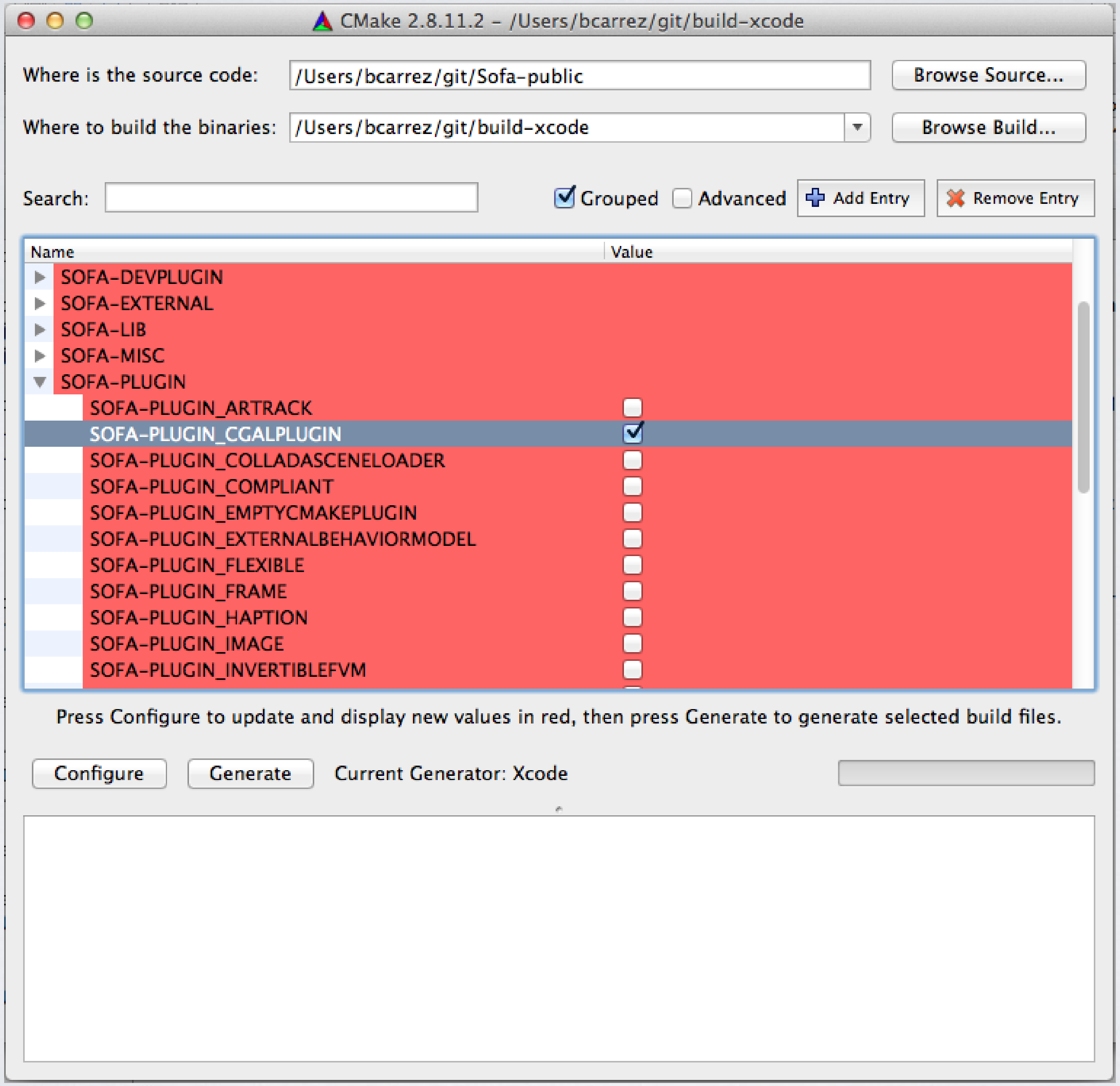
OPTIONS

- ✓ paths (ex: SOFA-EXTERNAL_BOOST_PATH)
- ✓ on/off (ex: plugins, projects, etc..)
- ✓ commands (ex: CMAKE_CXX_COMPILER)
- ✓ flags (ex: CMAKE_CXX_FLAGS_DEBUG)

... most of them are for advanced use, only a few usually needed

Configuration

User-friendly : cmake-gui



CMake 2.8.11.2 – /Users/bcarrez/git/build–xcode

Where is the source code: /Users/bcarrez/git/Sofa–public

Where to build the binaries: /Users/bcarrez/git/build–xcode

Search:

☑ Grouped ☐ Advanced

Name	Value
▶ SOFA–DEVPLUGIN	
▶ SOFA–EXTERNAL	
▶ SOFA–LIB	
▶ SOFA–MISC	
▼ SOFA–PLUGIN	
SOFA–PLUGIN_ARTRACK	☐
SOFA–PLUGIN_CGALPLUGIN	☑
SOFA–PLUGIN_COLLADASCENELOADER	☐
SOFA–PLUGIN_COMPLIANT	☐
SOFA–PLUGIN_EMPTYCMAKEPLUGIN	☐
SOFA–PLUGIN_EXTERNALBEHAVIORMODEL	☐
SOFA–PLUGIN_FLEXIBLE	☐
SOFA–PLUGIN_FRAME	☐
SOFA–PLUGIN_HAPTION	☐
SOFA–PLUGIN_IMAGE	☐
SOFA–PLUGIN_INVERTIBLEFVM	☐

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Xcode

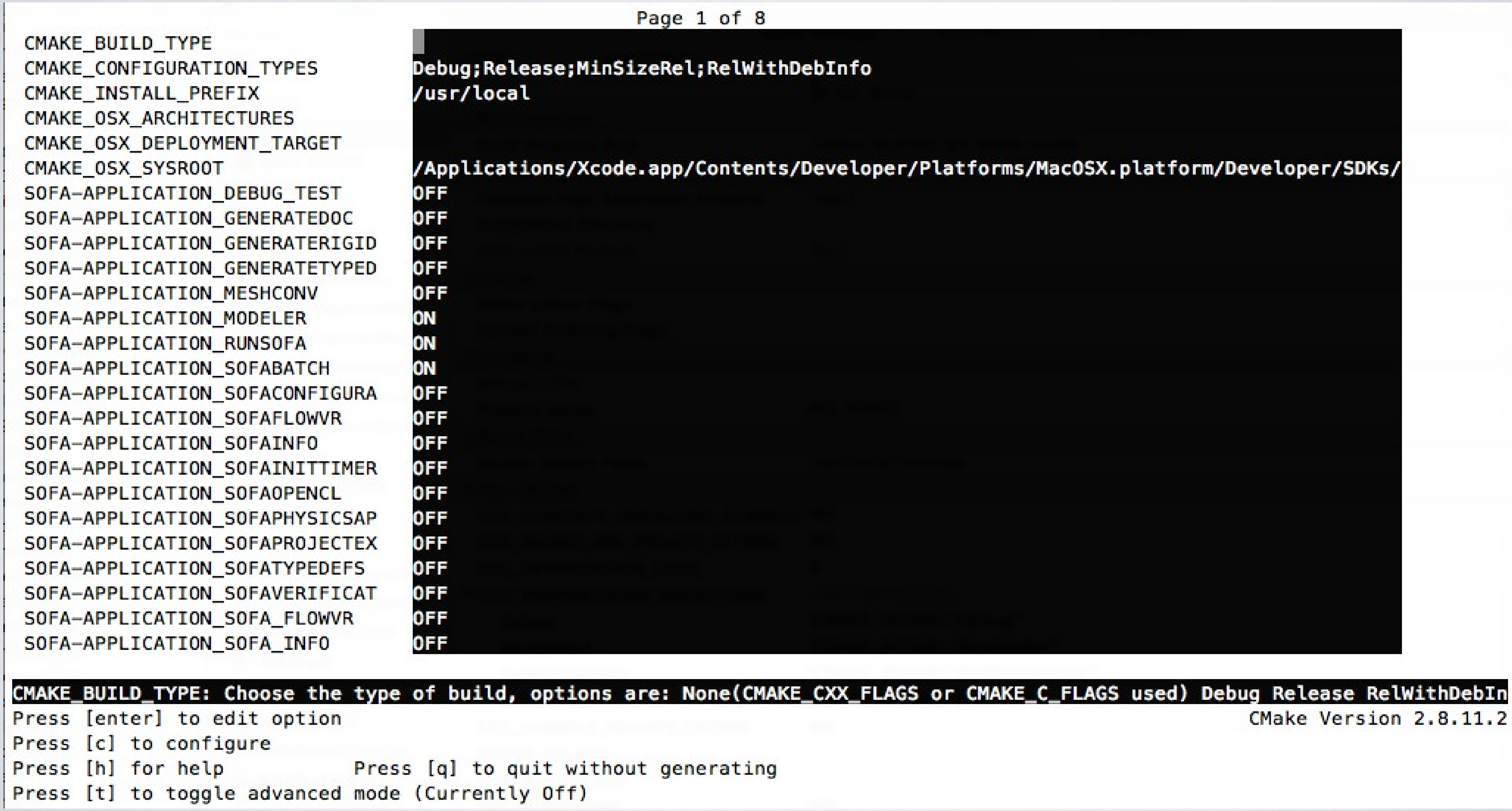
Configuration

User-friendly : cmake-gui

Demo

Configuration

Less user-friendly : ccmake



```
                                                  Page 1 of 8
 CMAKE_BUILD_TYPE
 CMAKE_CONFIGURATION_TYPES        Debug;Release;MinSizeRel;RelWithDebInfo
 CMAKE_INSTALL_PREFIX             /usr/local
 CMAKE_OSX_ARCHITECTURES
 CMAKE_OSX_DEPLOYMENT_TARGET
 CMAKE_OSX_SYSROOT                /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/
 SOFA-APPLICATION_DEBUG_TEST      OFF
 SOFA-APPLICATION_GENERATEDOC     OFF
 SOFA-APPLICATION_GENERATERIGID   OFF
 SOFA-APPLICATION_GENERATETYPED   OFF
 SOFA-APPLICATION_MESHCONV        OFF
 SOFA-APPLICATION_MODELER         ON
 SOFA-APPLICATION_RUNSOFA         ON
 SOFA-APPLICATION_SOFABATCH       ON
 SOFA-APPLICATION_SOFACONFIGURA   OFF
 SOFA-APPLICATION_SOFAFLOWVR      OFF
 SOFA-APPLICATION_SOFAINFO        OFF
 SOFA-APPLICATION_SOFAINITTIMER   OFF
 SOFA-APPLICATION_SOFAOPENCL      OFF
 SOFA-APPLICATION_SOFAPHYSICSAP   OFF
 SOFA-APPLICATION_SOFAPROJECTEX   OFF
 SOFA-APPLICATION_SOFATYPEDEFS    OFF
 SOFA-APPLICATION_SOFAVERIFICAT   OFF
 SOFA-APPLICATION_SOFA_FLOWVR     OFF
 SOFA-APPLICATION_SOFA_INFO       OFF

CMAKE_BUILD_TYPE: Choose the type of build, options are: None(CMAKE_CXX_FLAGS or CMAKE_C_FLAGS used) Debug Release RelWithDebIn
Press [enter] to edit option                                                                             CMake Version 2.8.11.2
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

Configuration

HARDCORE MODE : command-line

```
cmake -DSOFA-PLUGIN_SOFAPYTHON=ON
-DSOFA-PLUGIN_CGALPLUGIN=ON
-DSOFA-PLUGIN_COLLADASCENELOADER=ON
-DSOFA-PLUGIN_COMPLIANT=ON
-DSOFA-PLUGIN_FLEXIBLE=ON
-DSOFA-PLUGIN_FRAME=ON
-DSOFA-PLUGIN_IMAGE=ON
-DSOFA-PLUGIN_INVERTIBLEFVM=ON
-DSOFA-PLUGIN_MANIFOLDTOPOLOGIES=ON
-DSOFA-PLUGIN_MULTITHREADING=ON
-DSOFA-PLUGIN_SENSABLEEMULATION=ON
-DSOFA-LIB_COMPONENT_SPARSE_SOLVER=ON
-DSOFA-PLUGIN_REGISTRATION=ON
-DSOFA-PLUGIN_SOFACUDA=ON
-DSOFA-MISC_TESTS=ON
```

IDE

- ✓ QtCreator : loads CMakeLists.txt directly
- ✓ XCode : «cmake -G Xcode ..»
- ✓ Eclipse, Kdevelop, … «cmake –help»
- ✓ Visual Studio … use cmake-gui

PLUGINS

Automatic folders scan :
good bye sofa-dependencies.prf !

Multi-projects PLUGINS

the system scans *recursively* for CMakeLists.txt in a plugin folder

all projects are put under the same plugin option

project named with the pattern *_test will be compiled only if you enabled unit tests

UNDER THE HOOD

Sofa/cmake/ folder contains all the CMake utility scripts

DON’T TOUCH IT

(unless you REALLY know what you’re doing)

UNDER THE HOOD

useful functions

include_directories
add_library / add_executable
AddLinkerDependencies
AddCompilerDefinitions

UNDER THE HOOD

useful variables

PROJECT_NAME
SOFA_LIB_DIR
SOFA_EXTLIBS_DIR
SOFA_SHARE_DIR
....

UNDER THE HOOD

Use of 3rd-party SDKs

Quick and dirty way:

```
include_directories(/usr/local/truc/include)
AddLinkerDependencies(/usr/local/lib/libtruc.so)
```

… not very platform-independant :-/

UNDER THE HOOD

Use of 3rd-party SDKs

find_package(*libname* REQUIRED)

usually creates these variables:
*libname*_FOUND
*libname*_INCLUDE_DIRS
*libname*_LIBRARIES

include_directories(${*libname*_INCLUDE_DIRS})
AddLinkerDependencies(${*libname*_LIBRARIES})

Can also create some functions and/or variables
ex: cuda_include_directories

UNDER THE HOOD

Advanced topic : Custom Find*MYLIB*.cmake

script located in Sofa/cmake/ folder
useful with find_package for non-standard SDKs…
…or to override standard CMake scripts

examples:
FindHAPI.cmake (non-standard)
FindOPENGL.cmake (override)

CONCLUSION

✓ Transition period

✓ We are here to help you

✓ Feedback welcome !