Brat 4.2.0 – Command Line Build - Instructions for Debian 8, 64 bit

Contents

Introduction

Building BRAT and all its dependencies, or at least those that are not available as pre-compiled binaries for some intended build configuration, is a complex and time consuming task. In spite of BRAT support for all major desktop operating systems and most common processor architectures, providing installers for all of them as detailed above, the user may still want to try to build BRAT for some unsupported operating system or unusual configuration.

The INSTALL document, distributed with the BRAT source package, tries to cover the major guidelines and issues about this task, for all supported operating systems and major build tools, and to suggest some hints for the unsupported ones. But, given the big number of different compilation options, both for BRAT and for the packages that it depends on, it is not possible to cover all possible combinations, which are still increased by the possibly different operating system configurations, distributions, build tools, and so on. Because of this, the document is merely indicative in many respects, leaving some options for the user to decide. It is not an instruction set that, if accurately followed step by step, ensures a successful build of the whole BRAT package. However, it is possible to have such an instruction set for a specific environment and build configuration. This sort of document can be useful either as a reference for similar contexts or to try to infer the necessary adjustments for different ones, if at all possible. This file consists precisely in a set of guidelines and detailed steps to build a specific BRAT configuration, in a freshly installed Debian 8, 64 bit.

General Notes

Tools and terminal commands

Besides the command line toolchain for C++ builds, in particular g++ (Debian 4.9.2-10) 4.9.2, CMake version >= 3.2.3 must also be installed and referenced in the PATH environment variable.

Other necessary tools or libraries will be listed in the section for the dependency that first requires them. The characters "$>", followed by a space, are used to represent the command line prompt. As an example, the following line is an instruction to run the chrpath tool with some options:

```
$> chrpath -r '$ORIGIN/../../.' ./*/*.so
```

In this case, only the text beginning with "chrpath", that is, after "$> " should be entered in the command line.

Build type

For all packages, including BRAT, an out-of-source build type is used. This means that the files generated by the build procedures go to a directory outside the source directory, typically called "build", and to its sub-directories.

The build process generates intermediate files, only required by the build tools themselves, and final files, which are the targets of the build, necessary for client software to be able to use the package. After building, another step is required, the installation, which separates the final from the intermediate files and places those under a specified directory, typically called "bin".

Build tree structure

When decompressing the source packages, the source directory names are usually different in the different packages. Except where explicitly noted, it is always assumed that they were renamed "source". It is recommended that, for each package, a directory is created, named after the package, and that the sources be uncompressed there. When there are several versions in the same system, each version can typically have its own sub-directory. For instance, in the case of Qt, the build tree will consist in the root directory "Qt", the top sub-directory named after the version, "5.7.0", and the following subdirectories:

```
.../Qt/5.7.0/source
.../Qt/5.7.0/build
```

.../Qt/5.7.0/bin

Here, "..." means some absolute path under the user home directory. Note that if you decompress the Qt 5.7.0 package in .../Qt/5.7.0, a sub-directory named “qt-everywhere-opensource-src-5.7.0” will be created. By renaming it to "source", you get the .../Qt/5.7.0/source directory, which is where the build procedure will look for the source files.

In the example build described in this document, most packages are located under directories with absolute paths beginning with prefixes like, for instance:

/home/commonuser/s3-altb/lib/GUI/<package>

or

/home/commonuser/s3-altb/lib/GUI/Qt/<package>

or

/home/commonuser/s3-altb/lib/Graphics/<package>

Please adjust the paths shown in this document to your own by changing these prefixes to those that match the directories you choose to build BRAT and its dependencies in your system.

Cleaning intermediate files

In general, the build procedures for all packages will include the “make” and often the “make install” steps (sometimes the make step also installs). A “make clean” step could also be executed after each successful installation, but it is not included in the instructions below. This is an optional step, that can free a lot of disk space, but that implies rebuilding the entire package if an error occurs and/or the user decides to correct the build settings. It is recommended that the clean steps are executed only after all packages and BRAT were successfully built.

Building BRAT Dependencies

Qt 5.7.0

Tools and dependencies

Install the tool chrpath, using the command:

```
$> sudo apt-get install chrpath
```

To install the Qt5 dependencies, use the following commands:

```
$> sudo apt-get install build-dep qt5-default
$> sudo apt-get install libxcb-xinerama0-dev
$> sudo apt-get install build-essential perl python git
$> sudo apt-get install "^libxcb.*" libx11-xcb-dev libglu1-mesa-dev libxrender-dev libxi-dev
```

The packages Qt WebEngine and Qt Multimedia are not used or supported by this version of BRAT. Nor are they included in the build instructions in this document. Should the user need to compile them as dependencies of other software, these commands should also be run:

```
$> sudo apt-get install libssl-dev libxcursor-dev libxcomposite-dev libxdamage-dev libxrandr-dev libfontconfig1-dev libcap-dev libxtst-dev libpulse-dev libudev-dev libpci-dev libnss3-dev libasound2-dev libxss-dev libegl1-mesa-dev gperf bison
$> sudo apt-get install libasound2-dev libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
```

Build

Decompress the Qt 5.7.0 package qt-everywhere-opensource-src-5.7.0.tar as explained in the general noes.

Now, supposing you have the sources in

```
/home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/source
```

create the directory

/home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/build/x86_64/Debug

Open the terminal in this directory or change (cd) to it and enter the command

```
$> /home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/source/configure -v -opensource -confirm-license -prefix
/home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/bin/x86_64/Debug -platform linux-g++-64 -debug -qt-xcb -
no-gtk -R ./ -L/usr/local/opt/openssl/lib -I/usr/local/opt/openssl/include -openssl -opengl -nomake
examples -nomake tests -no-compile-examples -skip wayland -skip webengine
```

When it finishes, enter the following commands:

```
$> make
```

```
$> make install
```

Note that the make command can take some hours to finish.

Post-build step

Change to the directory

/home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/plugins

and run the command

```
$> chrpath -r '$ORIGIN/../../.' ./*/*.so
```

The results of the build can be found in

/home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/bin/x86_64/Debug

Note about packages dependent of Qt5 and built with CMake

Several BRAT dependencies, and the BRAT executables themselves, are built with CMake and depend of Qt5. It is necessary to instruct CMake to find the path for the Qt5 installation we just build. There are several possibilities to pass command line arguments to CMake to achieve this. The simplest one is to define CMAKE_PREFIX_PATH to point to the Qt5 installation directory mentioned above, but this may not work for all packages or in all systems, and then it is necessary to be more explicit and define specific variables for the several Qt5 modules.

When the instructions in this document use specific variables for the each Qt5 module, it may happen that the respective package does not need all Qt5 CMake variables used in the commands, and in those cases the list can be shortened. However, if in doubt, use the full list and you can safely disregard CMake warnings such as

"Manually-specified variables were not used by the project: (...)"

OpenSceneGraph (OSG)

Decompress the package OpenSceneGraph-3.4.0.zip in a dedicated directory as explained in the general notes. In our example, after renaming "OpenSceneGraph-3.4.0" to "source", the source directory is

/home/commonuser/s3-altb/lib/Graphics/OSG/3-4-0_14/source

Create the directory

/home/commonuser/s3-altb/lib/Graphics/OSG/3-4-0_14/build/x86_64/Debug

Change to that directory (command cd), and enter the command:

```
$> cmake -DCMAKE_INSTALL_PREFIX:PATH=../../../bin/x86_64 -DBUILD_OSG_EXAMPLES:BOOL=OFF -
DBUILD_TESTING:BOOL=OFF -DCMAKE_CXX_FLAGS:STRING= -m64 -std=c++11 -DCMAKE_C_FLAGS:STRING=
-m64 -DCMAKE_EXE_LINKER_FLAGS:STRING= -m64 -DCMAKE_BUILD_TYPE:STRING=Debug -
DDYNAMIC_OPENSCENEGRAPH:BOOL=ON -DBUILD_SHARED_LIBS:BOOL=ON -
DDYNAMIC_OPENTHREADS:BOOL=ON -DDYNAMIC_OPENSCENEGRAPH:BOOL=ON -
DCMAKE_USE_RELATIVE_PATHS=TRUE -DINCLUDE_INSTALL_DIR=../../../bin/x86_64/include -
DBUILD_OPENTHREADS_WITH_QT:BOOL=OFF -
DQt5Widgets_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Widgets
Qt5WebKitWidgets_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Web
KitWidgets -DQT_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/include -
```

DQT_QMAKE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake -DOSG_USE_QT:BOOL=ON -DDESIRED_QT_VERSION=5 ../../../source

When it finishes, enter the following commands:

$> make

$> make install

Note that the make command can take some hours to execute.

Post-build step

You should also enter the command:

$> ln -sf /home/commonuser/s3-altb/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib64 /home/commonuser/s3-altb/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib

This link is used in the build instructions that follow

The results of the build can be find in

/home/commonuser/s3-altb/lib/Graphics/OSG/3-4-0_14/bin/x86_64

QWT

In the case of QWT, use the package qwt-5.2.3-vs2015-qt5-patched.zip, which has some corrections and adjustments to BRAT build requirements. It is available in GitHub:

https://github.com/BRATDEV/main.

Decompress it in a dedicated directory, as explained in the general notes. In our example, the source directory is

/home/commonuser/s3-altb/lib/GUI/Qt/qwt/5-2-3_14/source

Create the directory

/home/commonuser/s3-altb/lib/GUI/Qt/qwt/5-2-3_14/build/x86_64/Debug

Change to that directory (cd), and enter:

```
$> /home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake ../../../source/src/src.pro -
r -spec linux-g++-64 CONFIG+=debug CONFIG+=plugin $ADDITIONAL_SETTINGS
INSTALLBASE=../../bin/x86_64/Debug DESTDIR=./
```

Then, enter the commands:

```
$> make
```

```
$> make install
```

If, after the make command, you see lines like the following in the terminal, you can safely disregard them:

```
mv: ‘libqwt.so’ and ‘./libqwt.so’ are the same file
Makefile:517: recipe for target 'libqwt.so' failed
make: [libqwt.so] Error 1 (ignored)
```

The results of the build can be found in

/home/commonuser/s3-altb/lib/GUI/Qt/qwt/5-2-3_14/bin/x86_64/Debug

QwtPlot3d

As with QWT, and for the same reasons, with qwtplot3d you should also use the package provided by BRAT in GitHub, qwtplot3d-0.2.7-qt5-vs2015-patched.tar.gz:

https://github.com/BRATDEV/main

Decompress it in a dedicated directory, as explained in the general notes. In our example, the source directory is

```
/home/commonuser/s3-altb/lib/GUI/Qt/qwtplot3d/0-2-7_14/source
```

Create the directory

```
/home/commonuser/s3-altb/lib/GUI/Qt/qwtplot3d/0-2-7_14/build/x86_64/Debug
```

Change to that directory (cd), and enter:

```
$> /home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake
../../../source/qwtplot3d.pro -r -spec linux-g++-64 CONFIG+=debug CONFIG+=plugin QMAKE_CXXFLAGS+=-
fvisibility=default DESTDIR=../../../bin/x86_64/Debug/lib OBJECTS_DIR=./
```

Then, enter the command:

```
$> make
```

The result of the build is libqwtplot3d.so and it can be found in

```
/home/commonuser/s3-altb/lib/GUI/Qt/qwtplot3d/0-2-7_14/bin/x86_64/Debug/lib
```

QScintilla

Build QScintilla using the package QScintilla-gpl-2.9-patched.zip, provided by BRAT in GitHub:

https://github.com/BRATDEV/main

Decompress it in a dedicated directory as explained in the general notes. In our example, the source directory is

```
/home/commonuser/s3-altb/lib/GUI/Qt/QScintilla/2-9_14/source
```

Create the directory

```
/home/commonuser/s3-altb/lib/GUI/Qt/QScintilla/2-9_14/build/x86_64/Debug
```

Change to that directory (cd), and enter:

```
$> /home/commonuser/s3-altb/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake
../../../source/Qt4Qt5/qscintilla.pro -r -spec linux-g++-64 QMAKE_CXXFLAGS+=-fvisibility=default
CONFIG+=staticlib CONFIG-=dll QMAKE_LFLAGS+=-stdlib=libc++ CONFIG+=debug
DESTDIR=../../../bin/x86_64/Debug
```

Then, enter the command:

```
$> make
```

The result of the build is libqscintilla2.a, which can be found in

```
/home/commonuser/s3-altb/lib/GUI/Qt/QScintilla/2-9_14/bin/x86_64/Debug
```

QCA

Decompress qca-2.1.0.tar in a dedicated directory, as explained in the general notes.

In our example, after renaming the sub-directory "qca-2.1.0" to "source", the full path of the source directory is

```
/home/commonuser/s3-altb/lib/GUI/Qt/qca/2-1-0/source
```

Original sources correction

The original source files need a minor correction. Please open the file

```
/home/commonuser/s3-altb/lib/GUI/Qt/qca/2-1-0/source/include/QtCrypto/qca_basic.h
```

in a text editor and add the line

```
#include <QIODevice>
```

before line 36, which is:

```
#include "qca_core.h"
```

Save and close the file.

Build

Create the directory

```
/home/commonuser/s3-altb/lib/GUI/Qt/qca/2-1-0/build/x86_64/Debug
```

Change to that directory (cd), and execute the following command:

```
$> cmake -DCMAKE_INSTALL_PREFIX:PATH=/home/commonuser/dev/lib/GUI/Qt/qca/2-1-
0/bin/x86_64/Debug -DCMAKE_CXX_FLAGS:STRING="-m64 -std=c++11" -DCMAKE_C_FLAGS:STRING="-
m64" -DCMAKE_EXE_LINKER_FLAGS:STRING="-m64 -std=c++11" -
DCMAKE_CONFIGURATION_TYPES:STRING="Debug;Release" -DCMAKE_BUILD_TYPE:STRING=Debug -
DOSX_FRAMEWORK:BOOL=OFF -DWITH_ossl_PLUGIN=yes -
DQt5Core_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Core -
DQt5Gui_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Gui -
DQt5Widgets_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Widgets -
DQt5Network_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Network -
DQt5Xml_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Xml -
DQt5XmlPatterns_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5XmlPa
tterns -DQt5Svg_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Svg -
```

DQt5Concurrent_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Concurrent -DQt5PrintSupport_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5PrintSupport -DQt5UiTools_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5UiTools -DQt5Script_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Script -DQt5Sql_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Sql -DQt5OpenGL_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5OpenGL -DQt5Positioning_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Positioning -DQt5Test_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Test -DENABLE_QT5:BOOL=ON -DWITH_QSPATIALITE:BOOL=OFF -DQT_LRELEASE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/lrelease -DWITH_QTWEBKIT=OFF -DQt5Core_QMAKE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake -DQT_QMAKE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake -DQT4_BUILD:BOOL=OFF ../../../source

Then, enter the commands:

```
$> make
```

```
$> make install
```

The results of the build can be found in

/home/commonuser/s3-altb/lib/GUI/Qt/qca/2-1-0/bin/x86_64/Debug

GEOS

Decompress geos-3.4.2.tar.gz in a dedicated directory as explained in the general notes.

In our example, the full path of the source directory is

/home/commonuser/s3-altb/lib/Graphics/QGIS/geos/3-4-2

Then, create the directory:

/home/commonuser/s3-altb/lib/Graphics/QGIS/geos/3-4-2/build/x86_64/Debug

Open a terminal there and execute the following command:

```
$> cmake -DCMAKE_INSTALL_PREFIX:PATH=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-
2/bin/x86_64/Debug -DCMAKE_CXX_FLAGS:STRING="-m64 -std=c++11 -fPIC" -
DCMAKE_C_FLAGS:STRING="-m64 -fPIC" -DCMAKE_EXE_LINKER_FLAGS:STRING="-m64  -std=c++11" -
DCMAKE_CONFIGURATION_TYPES:STRING="Debug;Release" -DCMAKE_BUILD_TYPE:STRING=Debug -
DCMAKE_USE_RELATIVE_PATHS=TRUE ../../../source
```

Then, enter the commands:

```
$> make
```

```
$> make install
```

The installed build outputs the can be found in

/home/commonuser/s3-altb/lib/Graphics/QGIS/geos/3-4-2/bin/x86_64/Debug

OsgEarth

Decompress gwaldron-osgearth-osgearth-2.7-0-g25ce0e1.tar.gz in a dedicated directory as explained in the general notes. In our example, after renaming the sub-directory "gwaldron-osgearth-25ce0e1" to "source", the full path of the source directory is

/home/commonuser/s3-altb/lib/Graphics/OSG/osgEarth/2-7_14/source

Then, create the directory:

/home/commonuser/s3-altb/lib/Graphics/OSG/osgEarth/2-7_14/build/x86_64/Debug

Open a command line in this directory.

Additional package dependencies

The package osgEarth depends of 2 additional system packages that you need to install. Execute the following commands in the console, as root:

```
$> apt-get install libcurl4-gnutls-dev

$> apt-get install libgdal-dev
```

Build

To make the OSG libraries, previously built, available to the CMake configuration process, enter also the following command:

```
$> export LD_LIBRARY_PATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib:$LD_LIBRARY_PATH
```

Then, execute:

```
$> cmake -DCMAKE_INSTALL_PREFIX:PATH=../../../bin/x86_64 -
DGEOS_INCLUDE_DIR=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-2/bin/x86_64/Debug/include
-DGEOS_LIBRARY=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-
2/bin/x86_64/Debug/lib/libgeos.so -DGEOS_CONFIG=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-
4-2/bin/x86_64/Debug/bin/geos-config -DCMAKE_CXX_FLAGS:STRING="-m64 -std=c++11" -
D__STDC_LIMIT_MACROS=1 -DCMAKE_C_FLAGS:STRING="-m64" -D__STDC_LIMIT_MACROS=1 -
DCMAKE_EXE_LINKER_FLAGS:STRING="-m64 -std=c++11" -D__STDC_LIMIT_MACROS=1 -
DCMAKE_BUILD_TYPE:STRING=Debug -DCMAKE_CONFIGURATION_TYPES:STRING="Debug;Release" -
DBUILD_SHARED_LIBS:BOOL=ON -DDYNAMIC_OSGEARTH:BOOL=ON -
DOSG_DIR=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64 -
DOSG_INCLUDE_DIR:PATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/include -
DOSG_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib/libosg.so -
DOSG_LIBRARY_DEBUG:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgd.so -
DCMAKE_PREFIX_PATH=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug -
DQT_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/include -
DQT_QMAKE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake -
DOSG_USE_QT:BOOL=ON -DDESIRED_QT_VERSION=5 -DOSGEARTH_USE_QT:BOOL=ON ../../../source
```

After CMake successfully runs, enter:

```
$> make
```

As with some of the previous libraries, osgEarth can take a few hours to compile.

Finally, execute

```
$> make install
```

The installed build outputs the can be found in

```
/home/commonuser/s3-altb/lib/Graphics/OSG/osgEarth/2-7_14/bin/x86_64
```

Post-build step

You should also enter the command:

```
$> ln -sf /home/commonuser/s3-altb/lib/Graphics/OSG/osgEarth/2-7_14/bin/x86_64/lib64
/home/commonuser/s3-altb/lib/Graphics/OSG/osgEarth/2-7_14/bin/x86_64/lib
```

This link is used in the build instructions that follow.

QGIS

Decompress qgis-latest.tar.bz2 in a dedicated directory as explained in the general notes. In our example, after renaming the sub-directory "qgis-2.16.1" to "source", the full path of the source directory is this

```
/home/commonuser/s3-altb/lib/Graphics/QGIS/2-16-1/source
```

Original sources correction

The original source files in the following list need corrections. Open each file in a text editor and save it after doing the respective correction:

1. Add the following lines to the file /home/commonuser/s3-altb/lib/Graphics/QGIS/2-16-1/source/src/plugins/globe/CMakeLists.txt after line 58:

```
IF (QT5_BUILD)
  FIND_PACKAGE(Qt5OpenGL REQUIRED)
  INCLUDE_DIRECTORIES(${Qt5OpenGL_INCLUDE_DIRS})
  TARGET_LINK_LIBRARIES(globeplugin ${Qt5OpenGL_LIBRARIES})
ENDIF(QT5_BUILD)
```

2. Add the following line to the file /home/commonuser/s3-altb/lib/Graphics/QGIS/2-16-1/source/src/plugins/globe/qgsglobevectorlayerproperties.h after line 33 (inside class declaration, bu before any other declarations):

```
Q_OBJECT
```

Additional package dependencies

In the case of QGIS you will also need to install some packages not installed by default with the operating system. So, execute the following commands, as root:

```
$> apt-get install flex
$> apt-get install grass-dev
$> apt-get install libqjson-dev
$> apt-get install libspatialindex-dev
```

Build

After the corrections are saved and additional dependencies installed, create the directory:

/home/commonuser/s3-altb/lib/Graphics/QGIS/2-16-1/build/x86_64/Debug

and open there the console.

Then, execute:

```
$> cmake -DCMAKE_INSTALL_PREFIX:PATH=/home/commonuser/dev/lib/Graphics/QGIS/2-16-
1/bin/x86_64/Debug -DCMAKE_BUILD_TYPE:STRING=Debug -DCMAKE_CXX_FLAGS:STRING="-m64 -
std=c++11" -DCMAKE_C_FLAGS:STRING="-m64" -DCMAKE_EXE_LINKER_FLAGS:STRING="-m64 -Wl,-
rpath=./" -DGEOS_INCLUDE_DIR=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-
2/bin/x86_64/Debug/include -DGEOS_LIBRARY=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-
2/bin/x86_64/Debug/lib/libgeos_c.so -
DGEOS_CONFIG=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-2/bin/x86_64/Debug/bin/geos-
config -DENABLE_TESTS:BOOL=OFF -DUSE_CXX_11:BOOL=TRUE -DWITH_BINDINGS:BOOL=OFF -
DOSGEARTH_ELEVATION_QUERY:BOOL=ON -
DQWT_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/qwt/5-2-3_14/bin/x86_64/Debug/include -
DQWT_LIBRARY=/home/commonuser/dev/lib/GUI/Qt/qwt/5-2-3_14/bin/x86_64/Debug/lib/libqwt.so -
DOPENTHREADS_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libOpenThreadsd.so -
DOSG_INCLUDE_DIR:PATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/include -
DOSG_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib/libosgd.so
-DOSGDB_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgDBd.so -
DOSGGA_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgGAd.so -
DOSGQT_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgQtd.so -
DOSGUTIL_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgUtild.so -
DOSGVIEWER_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgViewerd.so -
DOSGEARTH_INCLUDE_DIR=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/include -DOSGEARTH_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib64/libosgEarthd.so -
DOSGEARTHFEATURES_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib64/libosgEarthFeaturesd.so -
DOSGEARTHUTIL_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib64/libosgEarthUtild.so -
DOSGEARTHANNOTATION_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib64/libosgEarthAnnotationd.so -
DOSGEARTHQT_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib64/libosgEarthQtd.so -
DOSGEARTHSYMBOLOGY_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib64/libosgEarthSymbologyd.so -
DQSCINTILLA_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/QScintilla/2-9_14/source/Qt4Qt5 -
```

```
DQSCINTILLA_LIBRARY=/home/commonuser/dev/lib/GUI/Qt/QScintilla/2-
9_14/bin/x86_64/Debug/libqscintilla2.a -DCMAKE_USE_RELATIVE_PATHS=TRUE -
DQt5Core_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Core -
DQt5Gui_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Gui -
DQt5Widgets_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Widgets -
DQt5Network_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Network -
DQt5Xml_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Xml -
DQt5XmlPatterns_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5XmlPa
tterns -DQt5Svg_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Svg -
DQt5Concurrent_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Concur
rent -
DQt5PrintSupport_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5PrintS
upport -
DQt5UiTools_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5UiTools -
DQt5Script_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Script -
DQt5Sql_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Sql -
DQt5OpenGL_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5OpenGL -
DQt5Positioning_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Positio
ning -DQt5Test_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/lib/cmake/Qt5Test -
DENABLE_QT5:BOOL=ON -DWITH_QSPATIALITE:BOOL=OFF -
DQT_LRELEASE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/lrelease -
DWITH_QTWEBKIT=OFF -
DQT_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/include -
DQT_QMAKE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake -
DOSG_USE_QT:BOOL=ON -DDESIRED_QT_VERSION=5 -
DQCA_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/qca/2-1-0/bin/x86_64/Debug/include/QtCrypto
-DWITH_GLOBE:BOOL=ON -DQCA_LIBRARY=/home/commonuser/dev/lib/GUI/Qt/qca/2-1-
0/bin/x86_64/Debug/lib/libqca.so -DWITH_GRASS:BOOL=TRUE ../../../source
```

After CMake successfully configure and generate, execute the commands, as usual:

```
$> make
```

```
$> make install
```

QGIS can also take a few hours to build, depending on your system.

The installed build outputs the can be found in

```
/home/commonuser/s3-altb/lib/Graphics/QGIS/2-16-1/bin/x86_64/Debug
```

Xerces

Decompress xerces-c-3.1.1.tar.gz in a dedicated directory as explained in the general notes. In our example, after renaming xerces-c-3.1.1 t to source, the full path of the source directory is this

```
/home/commonuser/s3-altb/lib/XML/XSD/xerces-c-3.1.1/source
```

Open a terminal in this (source) directory, and not in a dedicated build directory, as with most other dependencies. Then, execute the sequence of commands:

```
./configure prefix=/home/commonuser/dev/lib/XML/XSD/xerces-c-3.1.1/bin/x86_64
```

```
$> make
```

```
$> make install
```

The installed build outputs the can be found in

```
/home/commonuser/s3-altb/lib/XML/XSD/xerces-c-3.1.1/bin/x86_64
```

Building BRAT 4.2.0

Extract brat-4.2.0.tar.gz in a dedicated directory as explained in the general notes. In our example, after renaming brat-4.2.0 to source, the full path of the source directory is this

```
/home/commonuser/s3-altb/project/archive/brat-4.2.0/source
```

After this, create the directory:

```
/home/commonuser/s3-altb/project/archive/brat-4.2.0/build/x86_64/Debug
```

and open there the console.

Additional package dependencies

You will also need to install some packages not installed by default with the operating system. So, execute the following commands, as root:

```
$> apt-get install python3.4-dev
$> apt-get install python3.4-dbg
$> apt-get install libidn11-dev
$> apt-get install libssh2-1-dev
$> apt-get install libsasl2-dev
$> apt-get install libldap2-dev
$> apt-get install xsdcxx
```

Build

After the installation of the additional packages finishes, execute:

```
$> cmake -DCMAKE_PREFIX_PATH=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug -
DBRAT_TARGET_PROCESSOR=x86_64 -DCMAKE_INSTALL_PREFIX:PATH=/home/commonuser/s3-
```

```
altb/project/archive/brat-4.2.0/bin/x86_64/Debug -DCMAKE_BUILD_TYPE:STRING=Debug -
DQWT_BIN_DIR=/home/commonuser/dev/lib/GUI/Qt/qwt/5-2-3_14/bin/x86_64/Debug/lib -
DQWT3D_BIN_DIR=/home/commonuser/dev/lib/GUI/Qt/qwtplot3d/0-2-7_14/bin/x86_64/Debug/lib -
DQGIS_BIN_DIR=/home/commonuser/dev/lib/Graphics/QGIS/2-16-1/bin/x86_64/Debug/lib -
DXercesC_INCLUDE_DIR=/home/commonuser/dev/lib/XML/XSD/xerces-c-3.1.1/bin/x86_64/include -
DXercesC_LIBRARY=/home/commonuser/dev/lib/XML/XSD/xerces-c-3.1.1/bin/x86_64/lib/libxerces-c.so -
DXSD_INCLUDE_DIR=/home/commonuser/dev/lib/XML/XSD/xerces-c-3.1.1/bin/x86_64/include -
DOSG_EARTH_BIN_DIR=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-7_14/bin/x86_64/lib -
DPYTHON_INCLUDE_DIR=/usr/include/python3.4 -DPYTHON_LIBRARY=/usr/lib/python3.4/config-3.4dm-
x86_64-linux-gnu/libpython3.4dm.a -DPYTHON_BIN_DIR=/usr/lib/python3.4 -
DOSG_BIN_DIR=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib -
DQWT_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/qwt/5-2-3_14/bin/x86_64/Debug/include -
DQWT_LIBRARY=/home/commonuser/dev/lib/GUI/Qt/qwt/5-2-3_14/bin/x86_64/Debug/lib/libqwt.so -
DQWTPLOT3D_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/qwtplot3d/0-2-7_14/source/include -
DQWTPLOT3D_LIBRARY=/home/commonuser/dev/lib/GUI/Qt/qwtplot3d/0-2-
7_14/bin/x86_64/Debug/lib/libqwtplot3d.so -
DOPENTHREADS_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libOpenThreadsd.so -
DOSG_INCLUDE_DIR:PATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/include -
DOSG_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib/libosgd.so
-DOSGDB_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgDBd.so -
DOSGGA_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgGAd.so -
DOSGQT_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgQtd.so -
DOSGUTIL_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgUtild.so -
DOSGVIEWER_LIBRARY:FILEPATH=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgViewerd.so -
DOSGSHADOW_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgShadowd.so -
DOSGTEXT_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-0_14/bin/x86_64/lib/libosgTextd.so -
DOSGWIDGET_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgWidgetd.so -DOSGSIM_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-
4-0_14/bin/x86_64/lib/libosgSimd.so -
DOSGTERRAIN_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgTerraind.so -DOSGFX_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgFXd.so -
DOSGMANIPULATOR_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/3-4-
0_14/bin/x86_64/lib/libosgManipulatord.so -
DOSGEARTH_INCLUDE_DIR=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/include -DOSGEARTH_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib/libosgEarthd.so -
DOSGEARTHFEATURES_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib/libosgEarthFeaturesd.so -
DOSGEARTHUTIL_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib/libosgEarthUtild.so -
DOSGEARTHQT_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib/libosgEarthQtd.so -
```

```
DOSGEARTHSYMBOLOGY_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib/libosgEarthSymbologyd.so -
DOSGEARTHANNOTATION_LIBRARY=/home/commonuser/dev/lib/Graphics/OSG/osgEarth/2-
7_14/bin/x86_64/lib/libosgEarthAnnotationd.so -DBUILD_TESTING:BOOL=OFF -
DCMAKE_CXX_FLAGS:STRING="-m64 -std=c++11" -DCMAKE_C_FLAGS:STRING="-m64" -
DCMAKE_EXE_LINKER_FLAGS:STRING="-fno-lto" -DBRAT_BUILD_GUI=ON -DBRATHL_BUILD_FORTRAN=OFF
-DQCA_BIN_DIR=/home/commonuser/dev/lib/GUI/Qt/qca/2-1-0/bin/x86_64/Debug/lib -
DQGIS_INCLUDE_DIR=/home/commonuser/dev/lib/Graphics/QGIS/2-16-1/bin/x86_64/Debug/include/qgis
-DQGIS_PLUGINS_DIR=/home/commonuser/dev/lib/Graphics/QGIS/2-16-
1/bin/x86_64/Debug/lib/qgis/plugins -
DQGIS_CORE_LIBRARY=/home/commonuser/dev/lib/Graphics/QGIS/2-16-
1/bin/x86_64/Debug/lib/libqgis_core.so -
DQGIS_GUI_LIBRARY=/home/commonuser/dev/lib/Graphics/QGIS/2-16-
1/bin/x86_64/Debug/lib/libqgis_gui.so -
DQGIS_ANALYSIS_LIBRARY=/home/commonuser/dev/lib/Graphics/QGIS/2-16-
1/bin/x86_64/Debug/lib/libqgis_analysis.so -DGDAL_INCLUDE_DIR=/usr/include/gdal -
DGEOS_LIBRARY=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-
2/bin/x86_64/Debug/lib/libgeos_c.so -
DGEOS_INCLUDE_DIR=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-2/bin/x86_64/Debug/include
-DGEOS_BIN_DIR=/home/commonuser/dev/lib/Graphics/QGIS/geos/3-4-2/bin/x86_64/Debug/lib -
DQCA_LIBRARY=/home/commonuser/dev/lib/GUI/Qt/qca/2-1-0/bin/x86_64/Debug/lib/libqca.so -
DQCA_INCLUDE_DIR=/home/commonuser/dev/lib/GUI/Qt/qca/2-1-0/bin/x86_64/Debug/include/QtCrypto
-DQT_QMAKE_EXECUTABLE=/home/commonuser/dev/lib/GUI/Qt/5.7.0/bin/x86_64/Debug/bin/qmake -
DENABLE_QT5=yes -DRSYNC_BIN_DIR=/usr/bin -DCMAKE_USE_RELATIVE_PATHS=TRUE -
DHDF5_BUILD_FORTRAN:BOOL=OFF -DBUILD_TESTING:BOOL=OFF -DHDF5_BUILD_EXAMPLES:BOOL=OFF -
DENABLE_TESTS:BOOL=OFF -DENABLE_DAP:BOOL=ON -DIS_CENTOS_SYSTEM:BOOL=OFF
/home/commonuser/s3-altb/project/archive/brat-4.2.0/source
```

After CMake successfully configured and generated the build files, execute the commands:

```
$> make
```

BRAT can take some time to finish building. Then, execute

```
$> make install
```

The installed build outputs will be found in

```
/home/commonuser/s3-altb/project/archive/brat-4.2.0/bin/x86_64/Debug
```

Testing the build

Change to the directory

```
/home/commonuser/s3-altb/project/archive/brat-4.2.0/bin/x86_64/Debug/bin
```

and execute:

```
$> ./brat
```

to run the main BRAT GUI application.

The SCHEDULER, as well as the BRAT command line tools (whose names start with "Brat") can be started in the same way, from the same directory.