Building GPlates from source on Mac OS X without package managers

Christian Heine[*]

Revision: 1067 Date: 2009-02-26

February 26, 2009

Abstract

This article describes how to build the open-source plate tectonic software GPlates on Mac OS X without the need of package managers like MacPorts or Fink. The procedure uses available installer packages for OS X from the net and only requires to compile one single library from source. I have been able to install GPlates smoothly on a Mac Pro and oldish 12" PowerBook using the recipe below.

1 For the impatient

This is for the impatient ones which don't need lengthy explanations:

- Install XCode from the Mac Dev Center
- Install the GDAL framework and dependancies from William Kyngesbury's site
- Install CMake from the CMake project
- Install the Qt framework from Trolltech
- Download and compile the ICU libaries from the ICU Project
- Download and unarchive the Boost libaries
- Download and unarchive the GPlates source
- Open CMake.app and point it to the directory where you unarchived the GPlates source code.
- Configure GPlates build in CMake.app
- Compile GPlates in Terminal.app by typing `make` in the GPlates source directory.

[*] Email: c.heine@usyd.edu.au

2 Prerequisites

This build requires that you have basic familiarity with the command line and the Terminal on Mac OS X using Terminal.app.

In order to be able to compile some of the components, it is required to have a c++ compiler installed on your machine. This is conveniently provided by Apple in form of the XCode development tools, which can be downloaded free of charge from the Mac Dev Center in the Apple Developer Connection. A login is required in order to be able to download the disk image. Be aware that this is a hefty download, around 900 MB.

Download the XCode tools disk image, double click to attach it and run the installer by double clicking on it. This will install a new directory on the top level of your hard disk called *Developer*. Now you should have access to the necessary compilers you need for the build process. Check whether g++ is installed properly by opening a terminal window (using *Terminal.app* in Applications → Utilities) and typing:

```
which g++
```

This should result in being a path returned which most likely looks like this:

```
/usr/bin/g++
```

Now you're ready to proceed and install the rest of the required software to compile GPlates successfully.

3 Downloading and installing dependancies

GPlates is dependant on several libraries which need to be installed before you can compile GPlates on your machine. These are partly provided by generous souls on the web as precompiled binaries nicely packaged as installer files and partly only as source code. You will need:

- GDAL (Geodata abstraction library)
- CMake
- Qt
- ICU libaries from the ICU Project
- Boost libaries
- GPlates source code

Some of the libraries are packaged as Mac OS X frameworks. This the Apple way of more conveniently packaging Unix Software/Libraries. According to a set directory layout, these libraries and utilities are installed in
`/Library/Frameworks/<FrameworkName>.framework`.
If you add this to your `$PATH`, you can access the utilities as if they would have been installed in, e.g., `/usr/local/`.

Depending on your personal preference of where to install applications (and administrator rights), it might be a good idea to create a directory structure where you place the GPlates source code and the associated dependencies ICU and Boost libraries (the rest is taken care of). This could be done within the Applications folder on the top level of your hard disk structure or somewhere in you home directory, e.g. `/User/<username>/GPlatesApp`.

3.1 GDAL framework

Source code and project site: http://www.gdal.org/
OS X Framework: http://www.kyngchaos.com/software:frameworks

Download the GDAL framework disk image from William Kyngesbury's website. Make sure to download all dependancy frameworks GDAL relies upon. These are the SQlite3, GEOS, PROJ, and the UnixImageIO frameworks (as of late Feb 2009). Install all packages by mounting the disk images and running the installer. Based on the information on Kyngesbury's site, you need to figure out in which order you need to install the individual frameworks.

3.2 Qt

Company site: http://www.trolltech.com
OS X Framework: http://www.qtsoftware.com/downloads/opensource/appdev/mac-os-cpp

Download the Qt installer disk image from Trolltech/Qtsoftware which is available a little further down on the page: ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.4.3.dmg. Mount the disk image and install the Qt framework.

3.3 CMake

Source code and project site: http://www.cmake.org
OS X Application: http://www.cmake.org/cmake/resources/software.html

Next, download the CMake binary distribution (http://www.cmake.org/files/v2.6/cmake-2.6.3-Darwin-universal.dmg, only for Tiger apparently, but works on Leopard). The installation will put a GUI frontend in your Applications folder, named `CMake <version>.app`. It should also automagically put symbolic links in `/usr/local/bin` which allow you to run cmake from the command line in the Terminal.

3.4 Boost libraries

Project site: http://www.boost.org
Source code: http://www.boost.org/users/download

The Boost libraries are not specifically packaged for Mac OS X. Download the Boost libraries archive from http://sourceforge.net/project/showfiles.php?group_id=7586&package_id=8041&release_id=659602 (version 1.38.0 was fine as of Feb 2009) and place the archive in the GPlates directory you created earlier on. You can simply unarchive the code by double clicking on the file.

3.5 ICU libraries

Project site: http://site.icu-project.org
Source code: http://site.icu-project.org/download

As the Boost libraries, the ICU libraries are also not specifically packaged for Mac OS X and need to be compiled from source. Download the source code archive for Unix using this link: http://download.icu-project.org/files/icu4c/4.0.1/icu4c-4_0_1-src.tgz

Move the source code archive into your GPlates directory and unarchive it. Now, open Terminal.app and change into your the `source` subdirectory of the ICU directory. You can do this by typing `cd` and dropping the folder from the Finder into the terminal window. Once this is done, type the following commands in the Terminal window:

```
./configure
make
make install
```

Depending on your machine they might take some time to finish.

Now, you're all set to compile GPlates.

4 Compiling GPlates

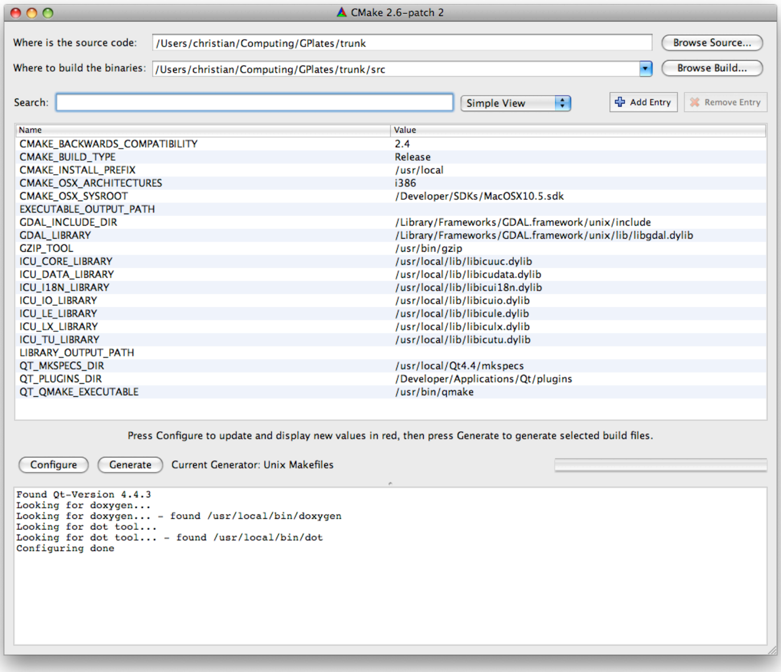
Project site: http://www.gplates.org
Source code: http://www.gplates.org/download

Download the latest GPlates source code from the GPlates site on Sourceforge: http://sourceforge.net/project/showfiles.php?group_id=74515&package_id=74881 . Unzip the compressed archive in your GPlates directory.

Open the CMake.app and point it to the source code for GPlates and where you want the binaries to be located (I have been using the same directory). Once you have set this, click the "Configure" button and use the default settings CMake is suggesting ("Unix Makefiles"). Confirm this by hitting the "OK" button and let CMake do its magic.

You should end up with a window like in Fig. 1. In case of errors, you can alter the path by clicking in the main CMake.app window and punch in the path by hand. Make sure to set the source directory to the top level GPlates directory. If CMake is happy, switch to the terminal window again and change into the `src` subdirectory in the GPlates directory. Type `make` to compile GPlates. Once the compilation has finished, you will find the excutable file in the very same directory (Fig 2).

Double-clicking the gplates executable (Fig. 2) should now open first another terminal window and the GPlates application window (Fig. 3).



CMake 2.6-patch 2

Where is the source code: /Users/christian/Computing/GPlates/trunk

Where to build the binaries: /Users/christian/Computing/GPlates/trunk/src

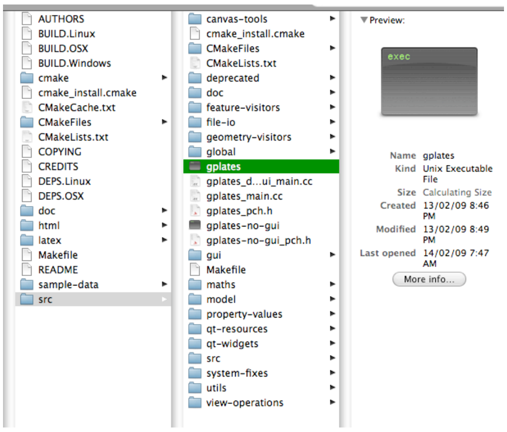
Name	Value
CMAKE_BACKWARDS_COMPATIBILITY	2.4
CMAKE_BUILD_TYPE	Release
CMAKE_INSTALL_PREFIX	/usr/local
CMAKE_OSX_ARCHITECTURES	i386
CMAKE_OSX_SYSROOT	/Developer/SDKs/MacOSX10.5.sdk
EXECUTABLE_OUTPUT_PATH	
GDAL_INCLUDE_DIR	/Library/Frameworks/GDAL.framework/unix/include
GDAL_LIBRARY	/Library/Frameworks/GDAL.framework/unix/lib/libgdal.dylib
GZIP_TOOL	/usr/bin/gzip
ICU_CORE_LIBRARY	/usr/local/lib/libicuuc.dylib
ICU_DATA_LIBRARY	/usr/local/lib/libicudata.dylib
ICU_I18N_LIBRARY	/usr/local/lib/libicui18n.dylib
ICU_IO_LIBRARY	/usr/local/lib/libicuio.dylib
ICU_LE_LIBRARY	/usr/local/lib/libicule.dylib
ICU_LX_LIBRARY	/usr/local/lib/libiculx.dylib
ICU_TU_LIBRARY	/usr/local/lib/libicutu.dylib
LIBRARY_OUTPUT_PATH	
QT_MKSPECS_DIR	/usr/local/Qt4.4/mkspecs
QT_PLUGINS_DIR	/Developer/Applications/Qt/plugins
QT_QMAKE_EXECUTABLE	/usr/bin/qmake

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

Current Generator: Unix Makefiles

```
Found Qt-Version 4.4.3
Looking for doxygen...
Looking for doxygen... - found /usr/local/bin/doxygen
Looking for dot tool...
Looking for dot tool... - found /usr/local/bin/dot
Configuring done
```

Figure 1: CMake configuration for GPlates.

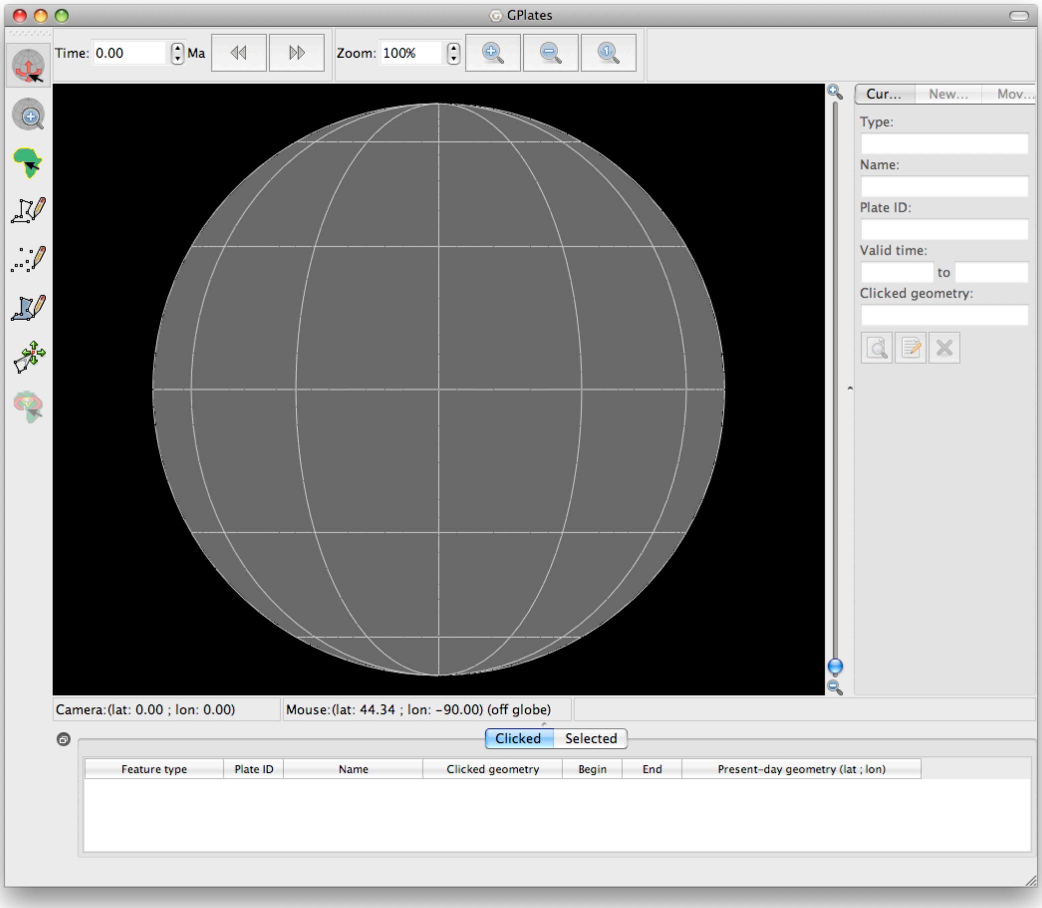


AUTHORS
BUILD.Linux
BUILD.OSX
BUILD.Windows
cmake
cmake_install.cmake
CMakeCache.txt
CMakeFiles
CMakeLists.txt
COPYING
CREDITS
DEPS.Linux
DEPS.OSX
doc
html
latex
Makefile
README
sample-data
src

canvas-tools
cmake_install.cmake
CMakeFiles
CMakeLists.txt
deprecated
doc
feature-visitors
file-io
geometry-visitors
global
gplates
gplates_d...ui_main.cc
gplates_main.cc
gplates_pch.h
gplates-no-gui
gplates-no-gui_pch.h
gui
Makefile
maths
model
property-values
qt-resources
qt-widgets
src
system-fixes
utils
view-operations

Name gplates
Kind Unix Executable File
Size Calculating Size
Created 13/02/09 8:46 PM
Modified 13/02/09 8:49 PM
Last opened 14/02/09 7:47 AM

Figure 2: GPlates executable.



Time: 0.00 Ma

Zoom: 100%

Type:

Name:

Plate ID:

Valid time: to

Clicked geometry:

Camera: (lat: 0.00 ; lon: 0.00)

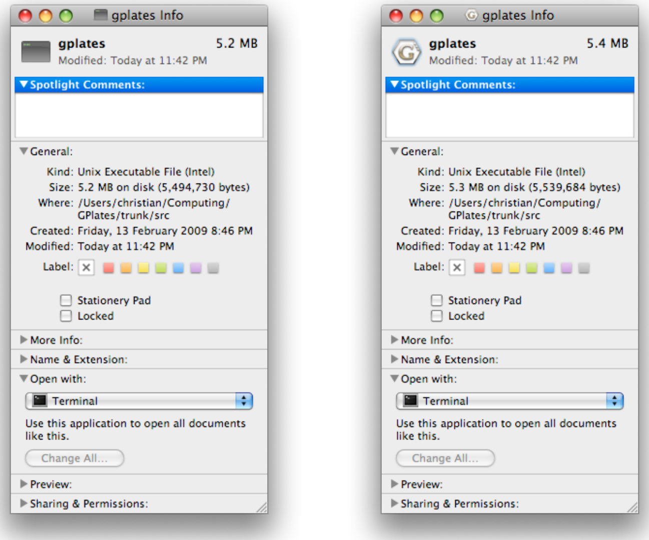
Mouse: (lat: 44.34 ; lon: -90.00) (off globe)

Feature type	Plate ID	Name	Clicked geometry	Begin	End	Present-day geometry (lat ; lon)

Figure 3: GPlates main window on Mac OS X.

4.1 Icon-tuning

You can tune up the icon for the application Mac-style. Select the GPlates executable and press command+i. This opens an info window (Fig. 4a). Now, descend into `<gplates source directory>/src/qt-resources/` and open the file `GPlates-icon-160x160-opaque.png` using Preview.app. Press command+c (copy), select the info window again, click on the little file icon in the upper left corner (should then have a blue halo) and paste the graphic there by pressing command+v.



gplates 5.2 MB
Modified: Today at 11:42 PM

Kind: Unix Executable File (Intel)
Size: 5.2 MB on disk (5,494,730 bytes)
Where: /Users/christian/Computing/GPlates/trunk/src
Created: Friday, 13 February 2009 8:46 PM
Modified: Today at 11:42 PM

Open with: Terminal
Use this application to open all documents like this.

(a) Before

gplates 5.4 MB
Modified: Today at 11:42 PM

Kind: Unix Executable File (Intel)
Size: 5.3 MB on disk (5,539,684 bytes)
Where: /Users/christian/Computing/GPlates/trunk/src
Created: Friday, 13 February 2009 8:46 PM
Modified: Today at 11:42 PM

Open with: Terminal
Use this application to open all documents like this.

(b) After

Figure 4: Tuning the GPlates icon for Mac OS X.

5 Links

- GDAL (Geodata abstraction library) `http://www.gdal.org/`
- CMake: `http://www.cmake.org/`
- Qt: `http://www.trolltech.com/`
- ICU libaries: `http://site.icu-project.org/`
- Boost libaries: `http://www.boost.org/`
- GPlates source code: `http://www.gplates.org/`