# Compiling "LibRoadRunner" for CC3D on Mac OS X (tested on OS X 10.8 "Mountain Lion")

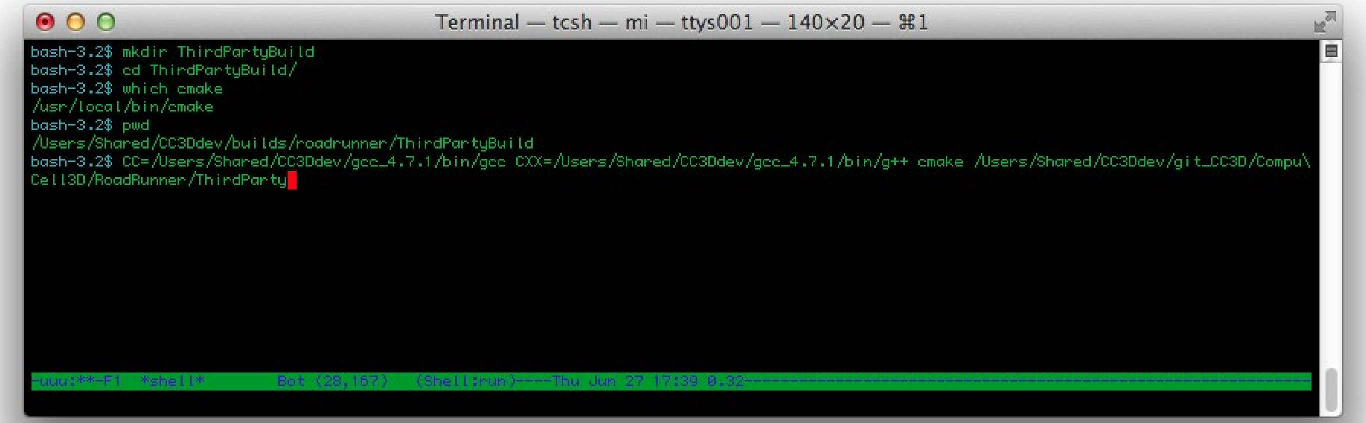
The CC3D *git repository* contains a version of the *RoadRunner* library customized for CC3D use. The source code is available at `<CC3D_Git_root>/RoadRunner`. For the purposes of this guide, the *local* CC3D *git* repository is located at `/Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/`.

The compilation of *RoadRunner* takes two separate steps. First we build the *ThirdParty* libraries, as found in the `/Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner/ThirdParty/` directory. Then we build the *RoadRunner* library from the source code in the `/Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner/` directory.

## Building the "ThirdParty" libraries

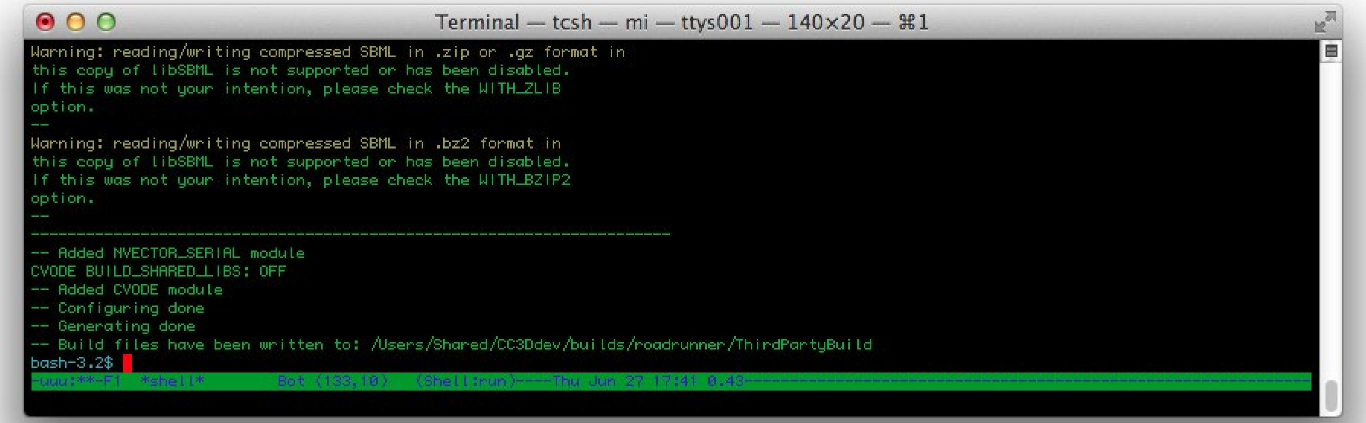
### Cmake Configuration and Makefile generation

Due to a bug in *cmake*, the C/C++ compiler's location can not be changed once cmake/ccmake is run on a specific *CMakeCache.txt* configuration. Since we want to specify the standard *gcc* to compile *RoadRunner* on Mac OS X, just as we do for CC3D, we have to explicitly define the `CC` and `CXX` shell environment variables *before* calling *cmake*, otherwise *cmake* would default to Mac OS X's default compiler Clang/LLVM. From Terminal.app using the `bash` shell, type:



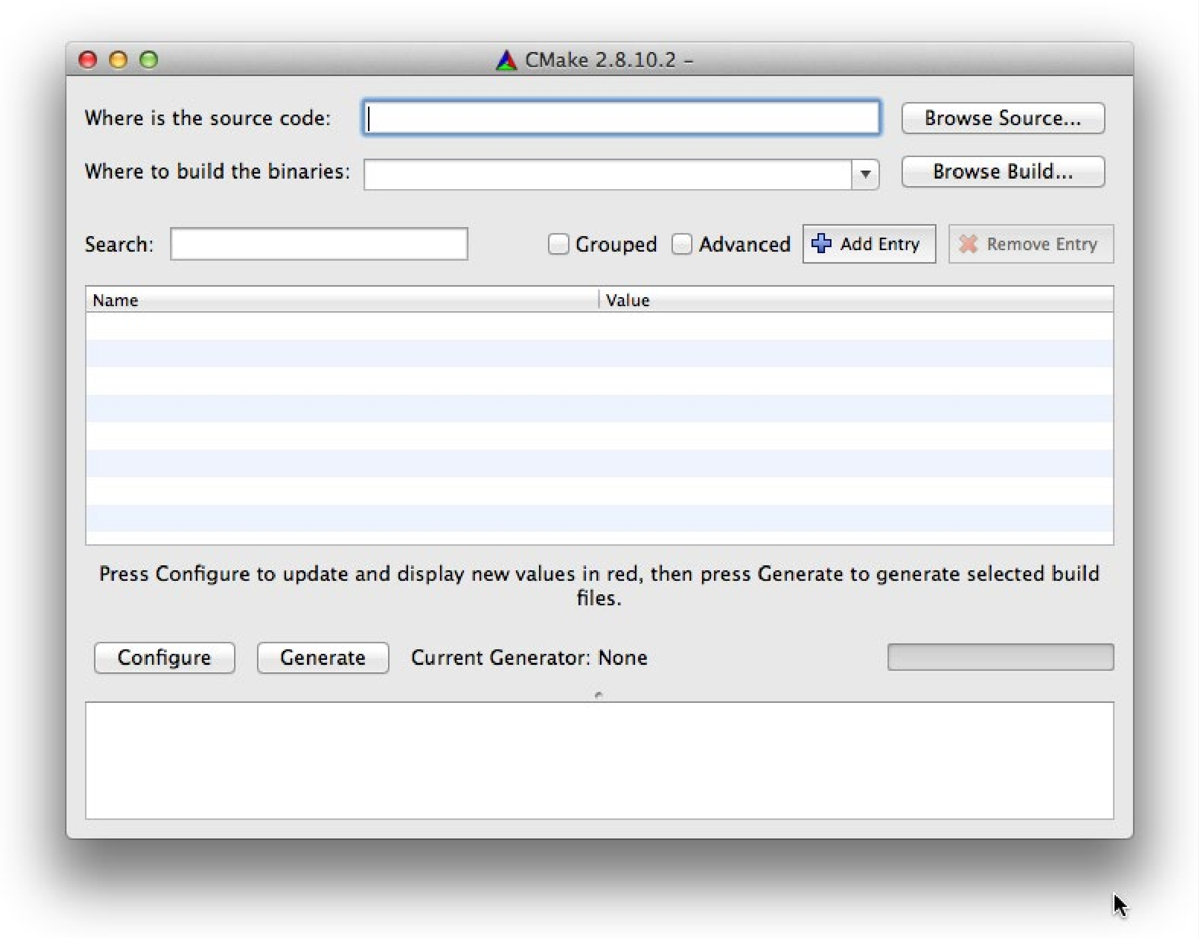
```
bash-3.2$ mkdir ThirdPartyBuild
bash-3.2$ cd ThirdPartyBuild/
bash-3.2$ which cmake
/usr/local/bin/cmake
bash-3.2$ pwd
/Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild
bash-3.2$ CC=/Users/Shared/CC3Ddev/gcc_4.7.1/bin/gcc CXX=/Users/Shared/CC3Ddev/gcc_4.7.1/bin/g++ cmake /Users/Shared/CC3Ddev/git_CC3D/Compu\
Cell3D/RoadRunner/ThirdParty









-uuu:**-F1  *shell*        Bot (28,167)   (Shell:run)----Thu Jun 27 17:39 0.32------------------------------------------------------------
```

i.e. `CC=<gcc C compiler location>  CXX=<gcc C++ compiler location>  cmake  <ThirdParty source code location>` which ensures that cmake will use the specified C and C++ compilers for the build process:



```
Warning: reading/writing compressed SBML in .zip or .gz format in
this copy of libSBML is not supported or has been disabled.
If this was not your intention, please check the WITH_ZLIB
option.
--
Warning: reading/writing compressed SBML in .bz2 format in
this copy of libSBML is not supported or has been disabled.
If this was not your intention, please check the WITH_BZIP2
option.
--
----------------------------------------------------------------------
-- Added NVECTOR_SERIAL module
CVODE BUILD_SHARED_LIBS: OFF
-- Added CVODE module
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild
bash-3.2$ 
-uuu:**-F1  *shell*        Bot (133,10)   (Shell:run)----Thu Jun 27 17:41 0.43--------------------------------------------------------------
```

Once cmake has completed its first run, we proceed with specifying further build settings in *Cmake.app*. If it's the first time *CMake.app* is run, there will be no predefined *path* values for source code and binaries:

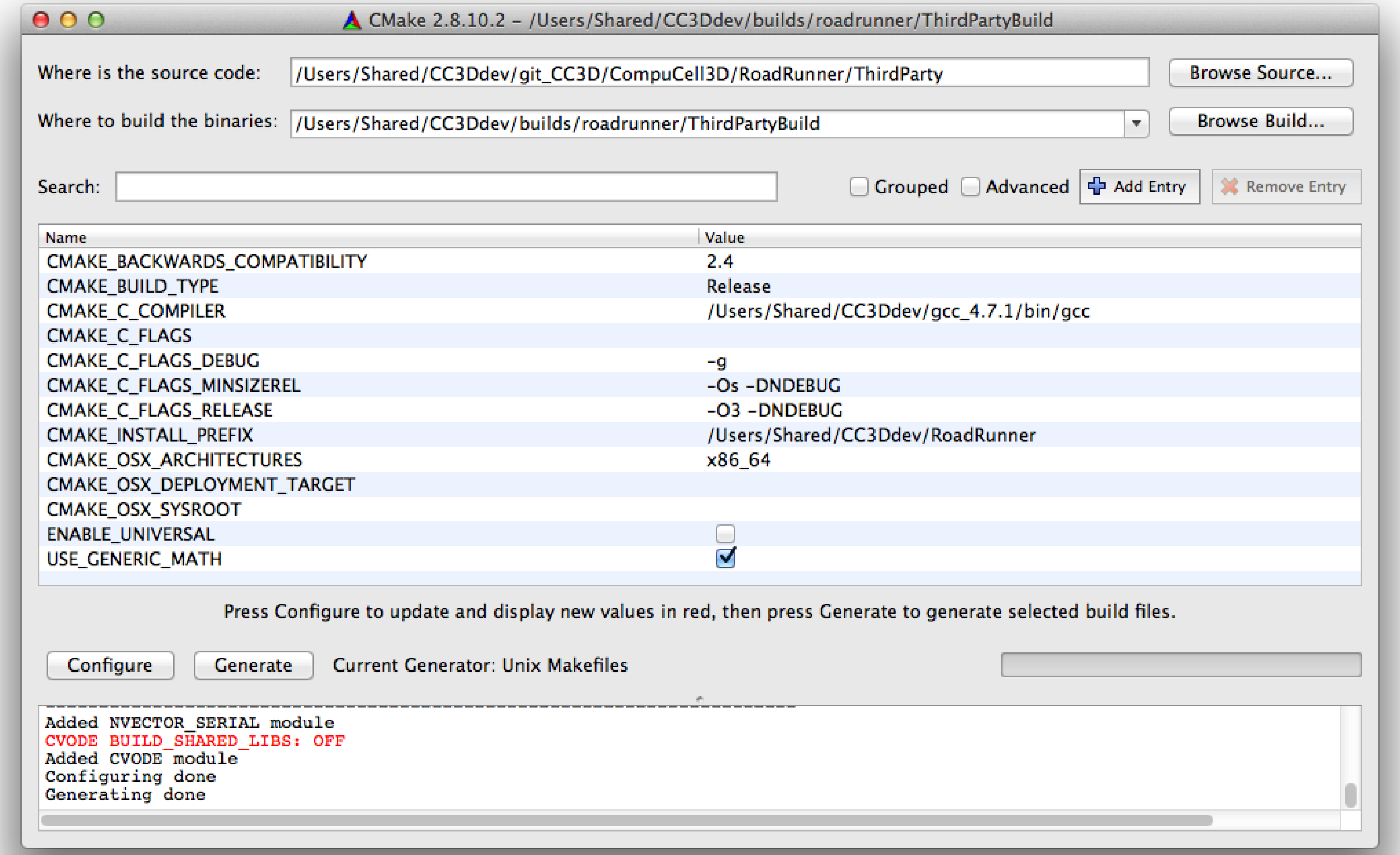


Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

Click on *Browse Source...* and locate the *ThirdParty* source directory (in our case it is `/Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner/ThirdParty`), then click on *Browse Build...* and locate the build directory (in our case it is `/Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild`). Click on *Configure*, and once CMake completes its initial configuration, change the following values in the CMake.app list of build settings:

- `CMAKE_BUILD_TYPE` to `Release`
- `CMAKE_INSTALL_PREFIX` to your install directory (in our case it is `/Users/Shared/CC3Ddev/RoadRunner`)
- `CMAKE_OSX_ARCHITECTURES` to `x86_64` (unless you're running on 32-bit hardware)
- `ENABLE_UNIVERSAL` can be left unchecked, since we're not building multiple-architecture binaries.



Where is the source code: /Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner/ThirdParty

Where to build the binaries: /Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild

| Name | Value |
|---|---|
| CMAKE_BACKWARDS_COMPATIBILITY | 2.4 |
| CMAKE_BUILD_TYPE | Release |
| CMAKE_C_COMPILER | /Users/Shared/CC3Ddev/gcc_4.7.1/bin/gcc |
| CMAKE_C_FLAGS | |
| CMAKE_C_FLAGS_DEBUG | -g |
| CMAKE_C_FLAGS_MINSIZEREL | -Os -DNDEBUG |
| CMAKE_C_FLAGS_RELEASE | -O3 -DNDEBUG |
| CMAKE_INSTALL_PREFIX | /Users/Shared/CC3Ddev/RoadRunner |
| CMAKE_OSX_ARCHITECTURES | x86_64 |
| CMAKE_OSX_DEPLOYMENT_TARGET | |
| CMAKE_OSX_SYSROOT | |
| ENABLE_UNIVERSAL | ☐ |
| USE_GENERIC_MATH | ☑ |

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

Current Generator: Unix Makefiles

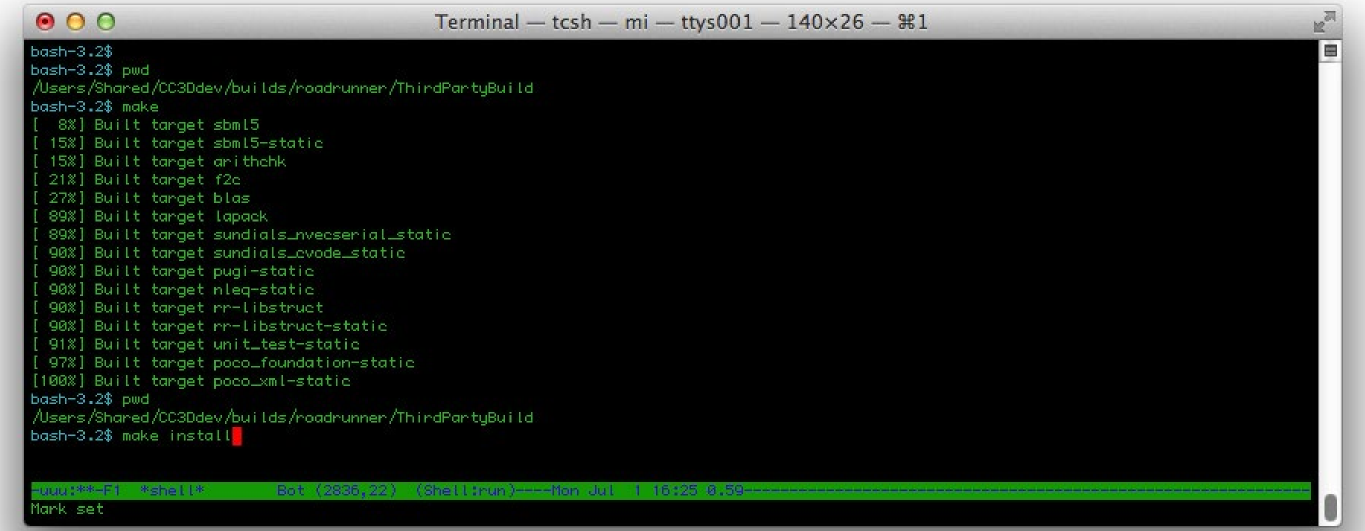
```
Added NVECTOR_SERIAL module
CVODE BUILD_SHARED_LIBS: OFF
Added CVODE module
Configuring done
Generating done
```

Then click on *Generate* in the CMake.app window to generate Makefile files for the *ThirdParty* build. Some build values may change when *Generate* is invoked: double-check that all build values are as needed, e.g. that the `CMAKE_INSTALL_PREFIX` points to our specified install directory, etc. and click on *Generate* again if necessary. Once CMake.app completes its *Generate* process, you can quit CMake.app and resume the build process in the Terminal.app window.

**"ThirdParty" build**

Return to the Terminal.app window, and `cd` to the directory where the Makefiles have been generated, in our case the `/Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild` directory. Once in that directory, run make and make install (if building on multi-processor, *make* can take advantage of parallel *gcc* compilation, by specifying the number of parallel compilation instances: e.g. `make -j4` for 4 CPU cores, etc.) :

```
cd /Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild
make
make install
```



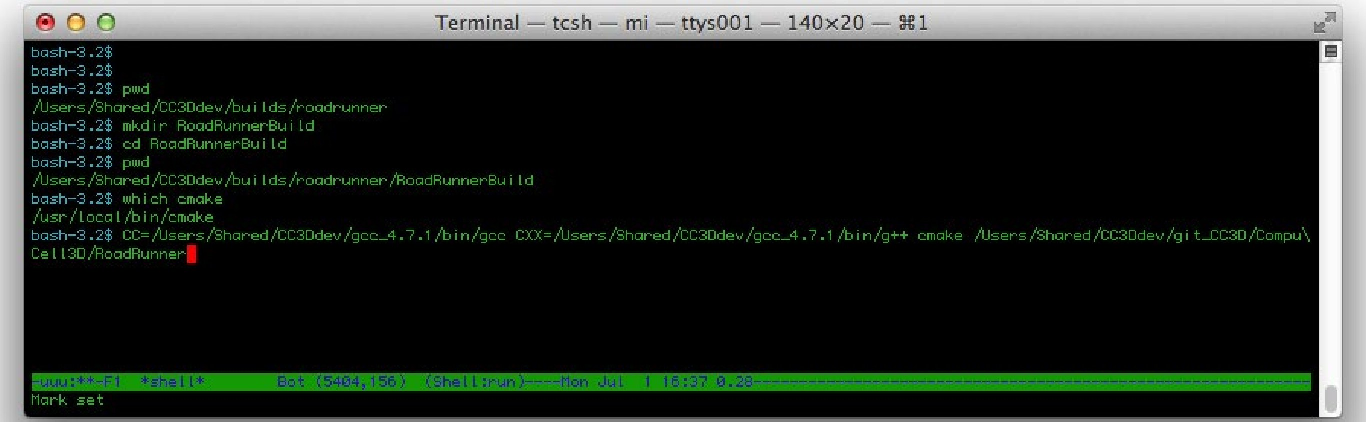
```
bash-3.2$
bash-3.2$ pwd
/Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild
bash-3.2$ make
[  8%] Built target sbml5
[ 15%] Built target sbml5-static
[ 15%] Built target arithchk
[ 21%] Built target f2c
[ 27%] Built target blas
[ 89%] Built target lapack
[ 89%] Built target sundials_nvecserial_static
[ 90%] Built target sundials_cvode_static
[ 90%] Built target pugi-static
[ 90%] Built target nleq-static
[ 90%] Built target rr-libstruct
[ 90%] Built target rr-libstruct-static
[ 91%] Built target unit_test-static
[ 97%] Built target poco_foundation-static
[100%] Built target poco_xml-static
bash-3.2$ pwd
/Users/Shared/CC3Ddev/builds/roadrunner/ThirdPartyBuild
bash-3.2$ make install


-uuu:**-F1  *shell*        Bot (2836,22)  (Shell:run)----Mon Jul  1 16:25 0.59------------------------------------------------------------
Mark set
```

This installs the required *ThirdParty* libraries in `/Users/Shared/CC3Ddev/RoadRunner`.

## Building "RoadRunner" for CC3D on Mac OS X (tested on OS X 10.8 "Mountain Lion")

### Cmake Configuration and Makefile generation

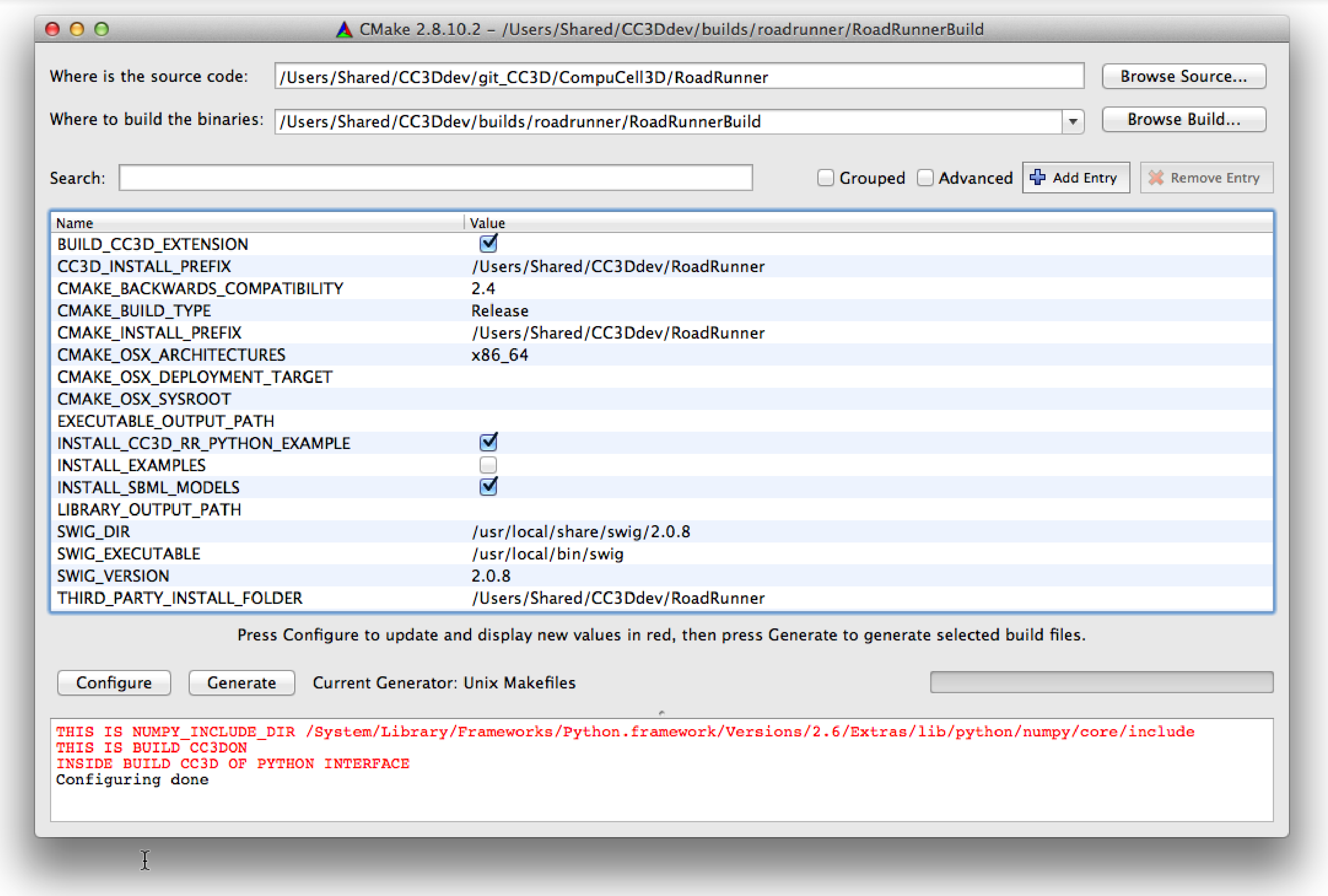
Similarly to how we configured *cmake* for building the required *ThirdParty* libraries above, to configure the *RoadRunner* build we specify the standard *gcc* for compilation on Mac OS X *before* calling *cmake*, otherwise *cmake* would default to Mac OS X's default compiler Clang/LLVM. From Terminal.app in the `bash` shell:



```
bash-3.2$
bash-3.2$
bash-3.2$ pwd
/Users/Shared/CC3Ddev/builds/roadrunner
bash-3.2$ mkdir RoadRunnerBuild
bash-3.2$ cd RoadRunnerBuild
bash-3.2$ pwd
/Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild
bash-3.2$ which cmake
/usr/local/bin/cmake
bash-3.2$ CC=/Users/Shared/CC3Ddev/gcc_4.7.1/bin/gcc CXX=/Users/Shared/CC3Ddev/gcc_4.7.1/bin/g++ cmake /Users/Shared/CC3Ddev/git_CC3D/Compu\
Cell3D/RoadRunner






-uuu:**-F1  *shell*        Bot (5404,156)  (Shell:run)----Mon Jul  1 16:37 0.28------------------------------------------------------------
Mark set
```

Once cmake has completed its first run, we proceed with specifying further build settings in *Cmake.app*. As for building the *ThirdParty* libraries, select the correct source (*/Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner*) and build (*/Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild*) directories. Click on *Configure*, then after CMake completes its initial configuration, change the following values in the CMake.app list of build settings:

- `CMAKE_BUILD_TYPE` to `Release`
- `CMAKE_INSTALL_PREFIX` to your install directory (in our case it is `/Users/Shared/CC3Ddev/RoadRunner`)
- `THIRD_PARTY_INSTALL_PREFIX` to where the *ThirdParty* libraries were installed (in our case it is `/Users/Shared/CC3Ddev/RoadRunner`)
- `CMAKE_OSX_ARCHITECTURES` to `x86_64` (unless you're running on 32-bit hardware)
- `BUILD_CC3D_EXTENSION` needs to be checked



CMake 2.8.10.2 – /Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild

Where is the source code: /Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner

Where to build the binaries: /Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild

| Name | Value |
|---|---|
| BUILD_CC3D_EXTENSION | ☑ |
| CC3D_INSTALL_PREFIX | /Users/Shared/CC3Ddev/RoadRunner |
| CMAKE_BACKWARDS_COMPATIBILITY | 2.4 |
| CMAKE_BUILD_TYPE | Release |
| CMAKE_INSTALL_PREFIX | /Users/Shared/CC3Ddev/RoadRunner |
| CMAKE_OSX_ARCHITECTURES | x86_64 |
| CMAKE_OSX_DEPLOYMENT_TARGET | |
| CMAKE_OSX_SYSROOT | |
| EXECUTABLE_OUTPUT_PATH | |
| INSTALL_CC3D_RR_PYTHON_EXAMPLE | ☑ |
| INSTALL_EXAMPLES | □ |
| INSTALL_SBML_MODELS | ☑ |
| LIBRARY_OUTPUT_PATH | |
| SWIG_DIR | /usr/local/share/swig/2.0.8 |
| SWIG_EXECUTABLE | /usr/local/bin/swig |
| SWIG_VERSION | 2.0.8 |
| THIRD_PARTY_INSTALL_FOLDER | /Users/Shared/CC3Ddev/RoadRunner |

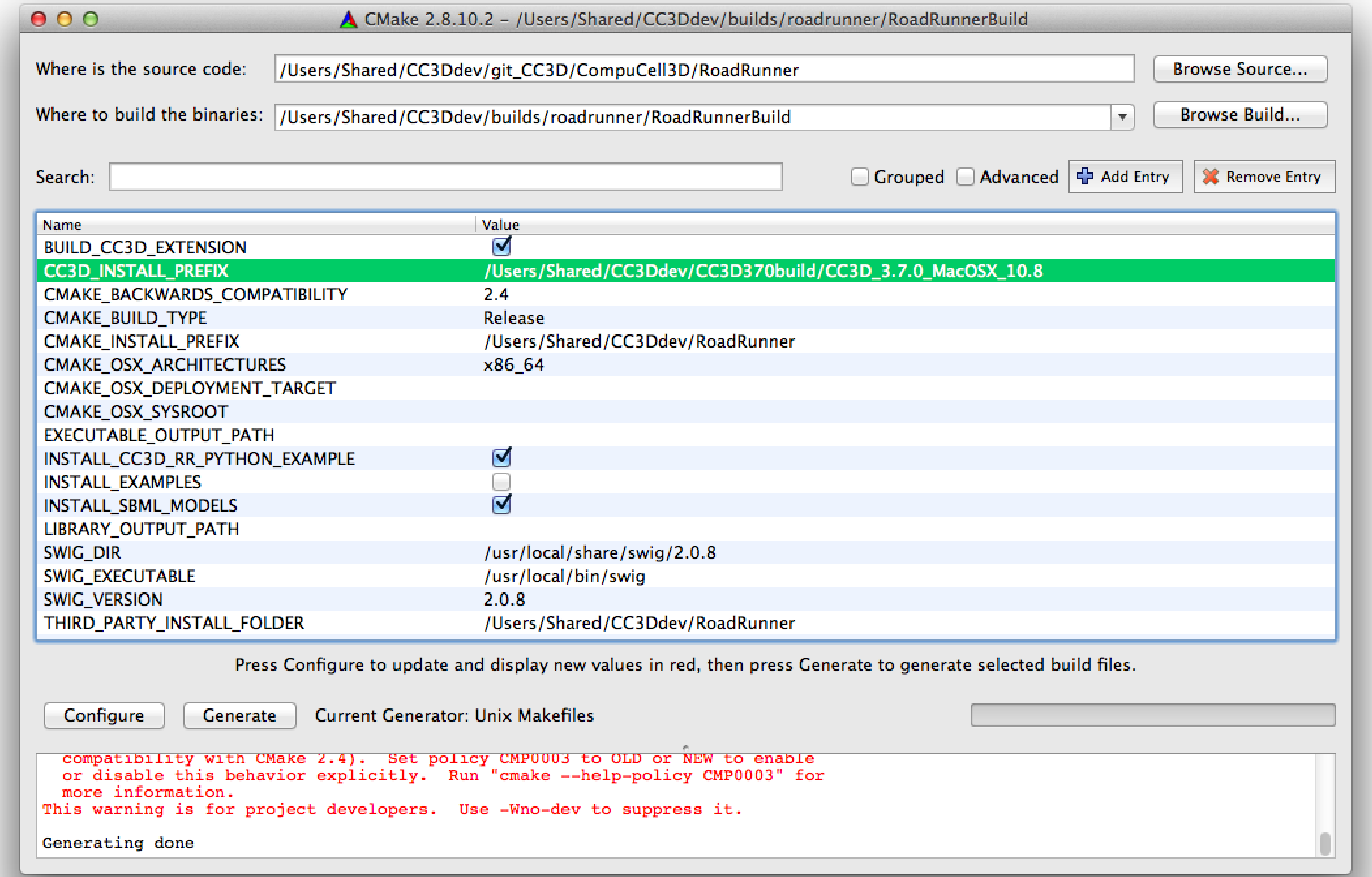
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Unix Makefiles

```
THIS IS NUMPY_INCLUDE_DIR /System/Library/Frameworks/Python.framework/Versions/2.6/Extras/lib/python/numpy/core/include
THIS IS BUILD CC3DON
INSIDE BUILD CC3D OF PYTHON INTERFACE
Configuring done
```

Click *Configure* again, then change the following values in the CMake.app list of build settings:

- `CC3D_INSTALL_PREFIX` needs to be changed to the base CC3D directory, (in our case it is `/Users/Shared/CC3Ddev/CC3D370build/CC3D_3.7.0_MacOSX_10.8`)



Where is the source code: /Users/Shared/CC3Ddev/git_CC3D/CompuCell3D/RoadRunner

Where to build the binaries: /Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild

| Name | Value |
|---|---|
| BUILD_CC3D_EXTENSION | ☑ |
| CC3D_INSTALL_PREFIX | /Users/Shared/CC3Ddev/CC3D370build/CC3D_3.7.0_MacOSX_10.8 |
| CMAKE_BACKWARDS_COMPATIBILITY | 2.4 |
| CMAKE_BUILD_TYPE | Release |
| CMAKE_INSTALL_PREFIX | /Users/Shared/CC3Ddev/RoadRunner |
| CMAKE_OSX_ARCHITECTURES | x86_64 |
| CMAKE_OSX_DEPLOYMENT_TARGET | |
| CMAKE_OSX_SYSROOT | |
| EXECUTABLE_OUTPUT_PATH | |
| INSTALL_CC3D_RR_PYTHON_EXAMPLE | ☑ |
| INSTALL_EXAMPLES | □ |
| INSTALL_SBML_MODELS | ☑ |
| LIBRARY_OUTPUT_PATH | |
| SWIG_DIR | /usr/local/share/swig/2.0.8 |
| SWIG_EXECUTABLE | /usr/local/bin/swig |
| SWIG_VERSION | 2.0.8 |
| THIRD_PARTY_INSTALL_FOLDER | /Users/Shared/CC3Ddev/RoadRunner |

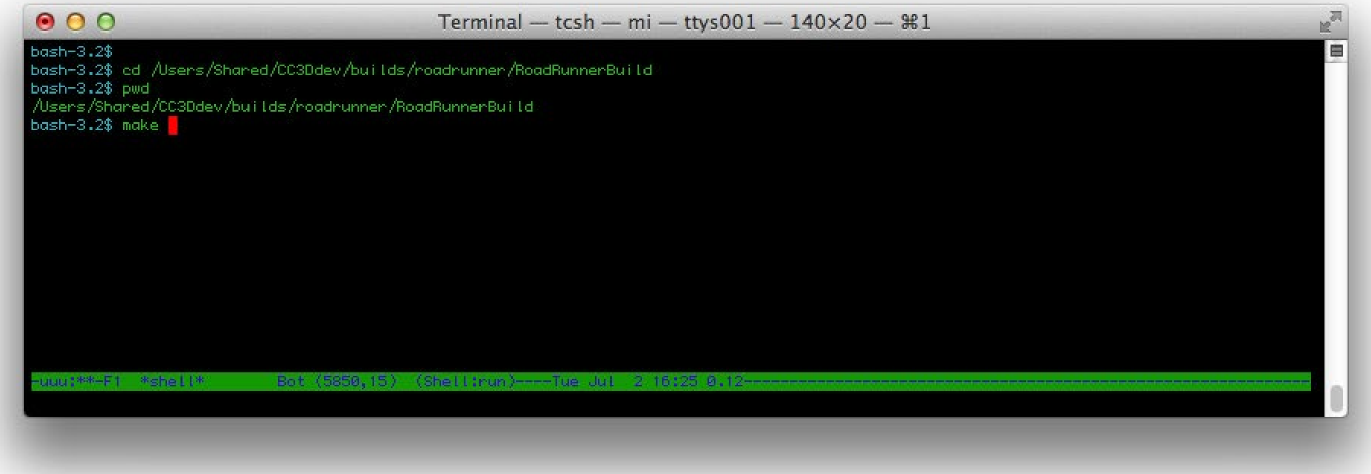
Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Unix Makefiles

```
  compatibility with CMake 2.4).  Set policy CMP0003 to OLD or NEW to enable
  or disable this behavior explicitly.  Run "cmake --help-policy CMP0003" for
  more information.
This warning is for project developers.  Use -Wno-dev to suppress it.

Generating done
```

**"RoadRunner" compilation and install:**

Compiling and installing *RoadRunner* is very similar to the *ThirdParty* libraries compilation above.



```
bash-3.2$
bash-3.2$ cd /Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild
bash-3.2$ pwd
/Users/Shared/CC3Ddev/builds/roadrunner/RoadRunnerBuild
bash-3.2$ make
```

```
-uuu:**-F1  *shell*       Bot (5850,15)  (Shell:run)----Tue Jul  2 16:25 0.12-----------------------------------------------------------------
```

1) from Terminal.app in the *bash* shell, cd to the `/Users/Shared/CC3Ddev/CC3D370build/CC3D_3.7.0_MacOSX_10.8` directory:

```
cd /home/m/CODE_TGIT_build/RoadRunnerBuild
```

2) start the build process with *make*:

```
make
```

3) finally, install *RoadRunner* in the CC3D base directory:

```
make install
```

Now your CC3D installtion should have *RoadRunner* installed and all SBMLSolver based simulations should run.