Cmake Primer

What is CMake?

- Cross-platform, open-source build system
- Family of tools for building, testing, and packaging software
- CMake is used to control the software compilation process using platform and compiler independent configuration files
- CMake generates native makefiles or solution files

- First implemented in 2000
- Current release: 3.6.3
- Website: https://cmake.org/

Using CMake

- Command line:

  `cmake [<options>] (<path-to-source> | <path-to-existing-build>)`
  configures a build system for the specified cmake project dir

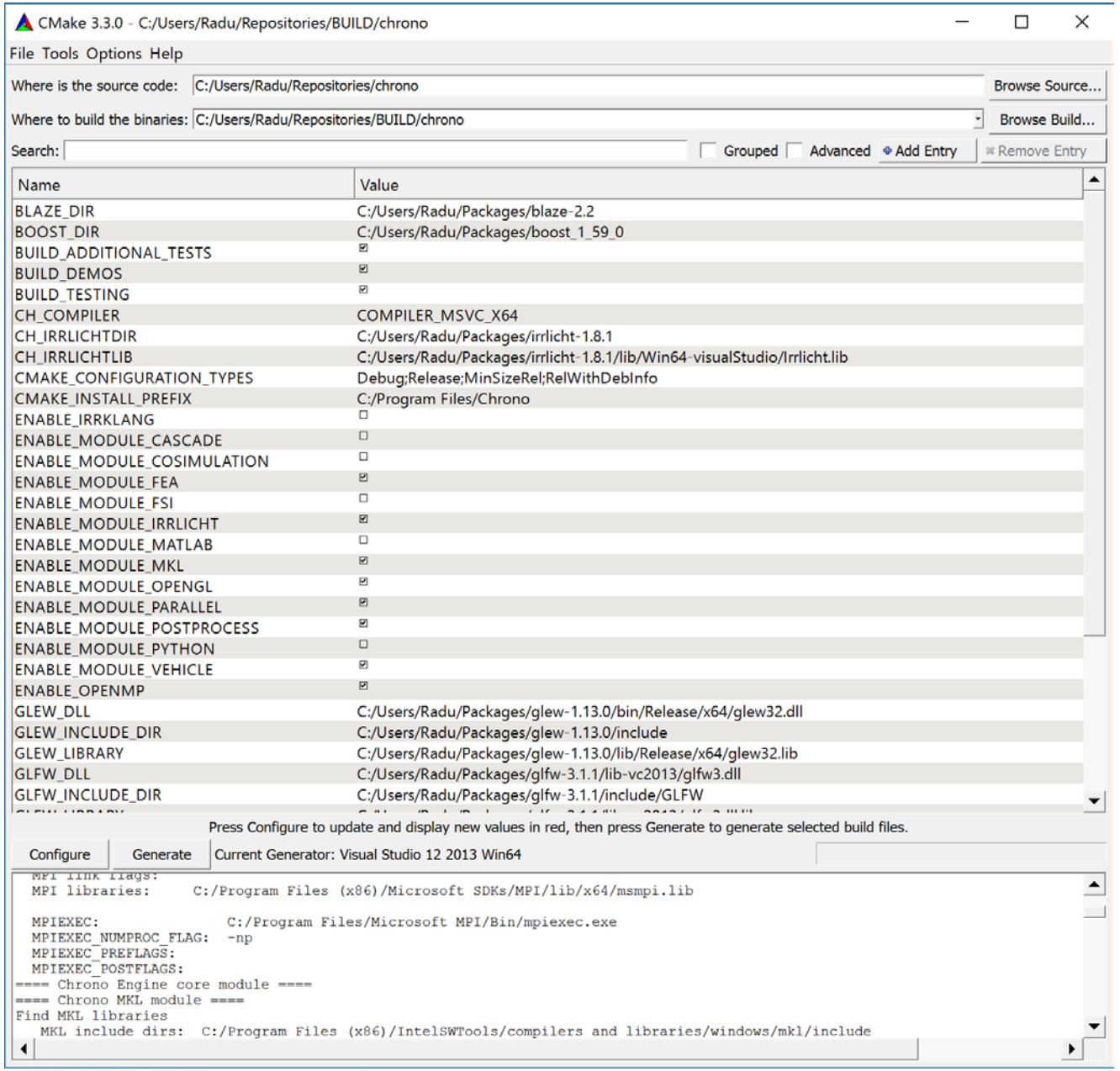
  `cmake [(-D<var>=<value>)...] -P <cmake-script-file>`
  executes a cmake script file

  `cmake --build <dir> [<options>] [-- <build-tool-options>...]`
  executes the build process through a generic interface

  `cmake -E <command> [<options>...]`
  gives you cross platform commands

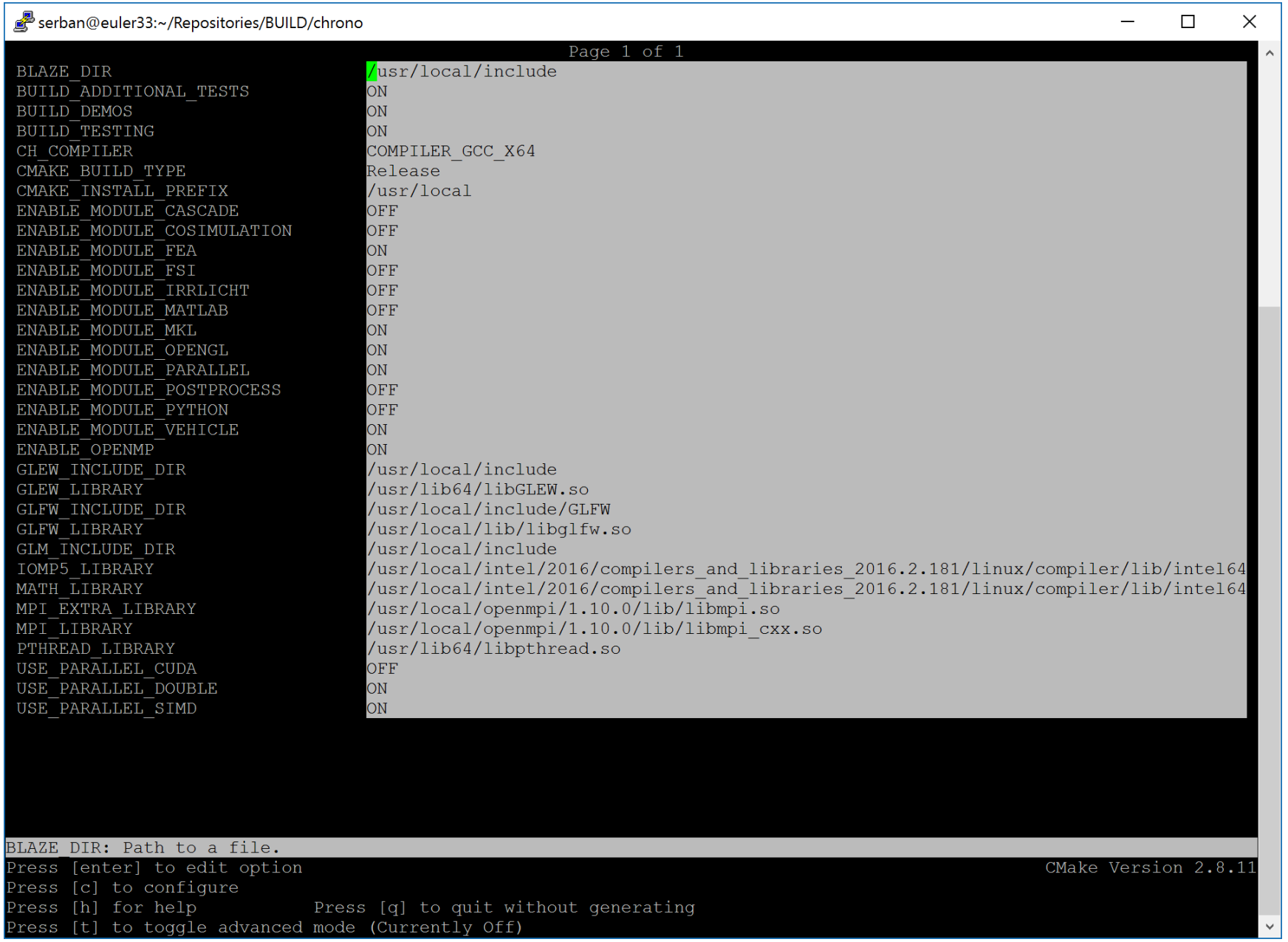
Using CMake

- cmake-gui



Using CMake

- ccmake



The CMake build process

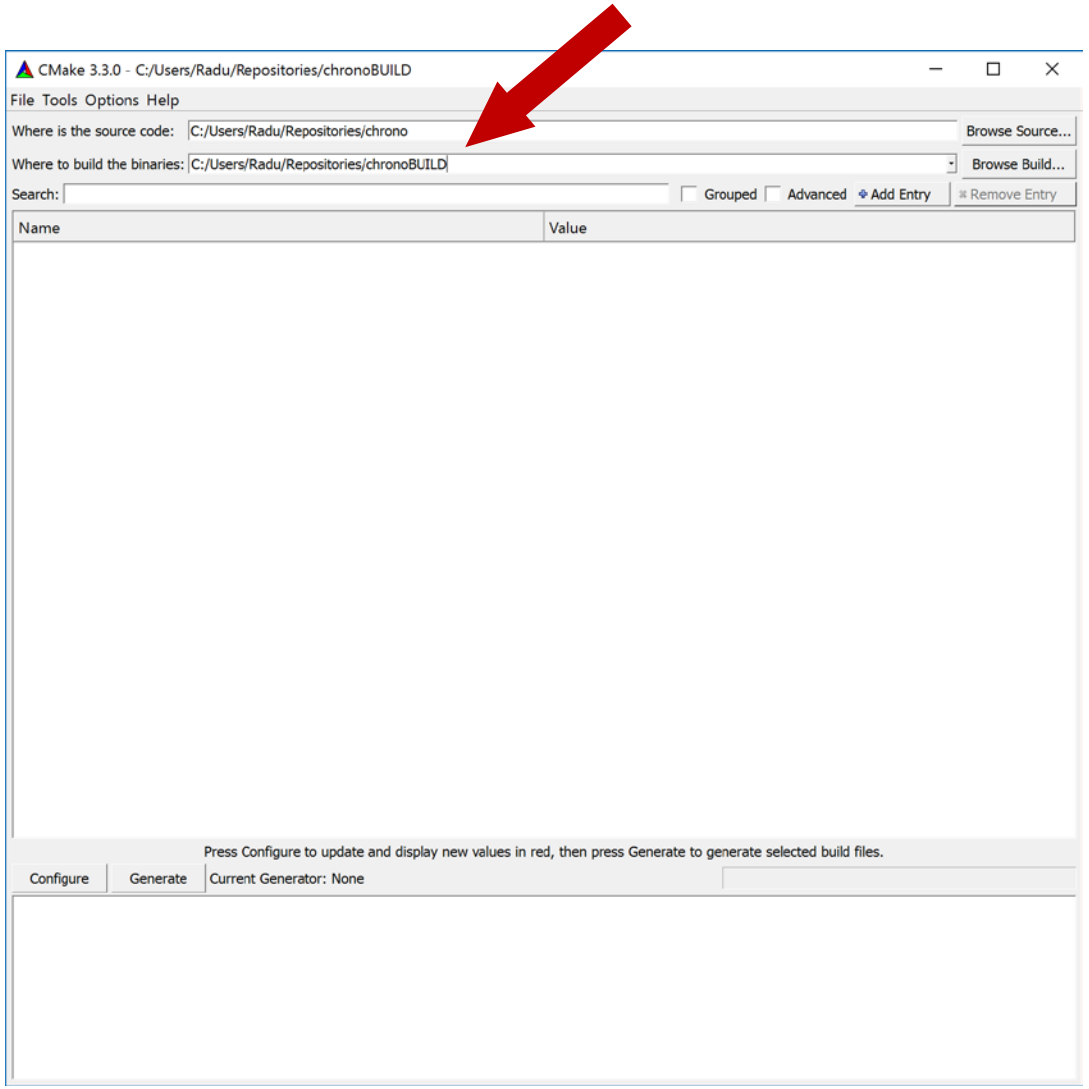
- Consists of two stages:
  1. Create standard (platform-specific) build files from the configuration files
  2. Use the platform native build tools for the actual building

- Creating the build files
  - always prefer out-of-source builds
  - if using cmake-gui, select the generator (for Chrono, make sure to use 64-bit)
  - [iterate] set options → configure
  - generate

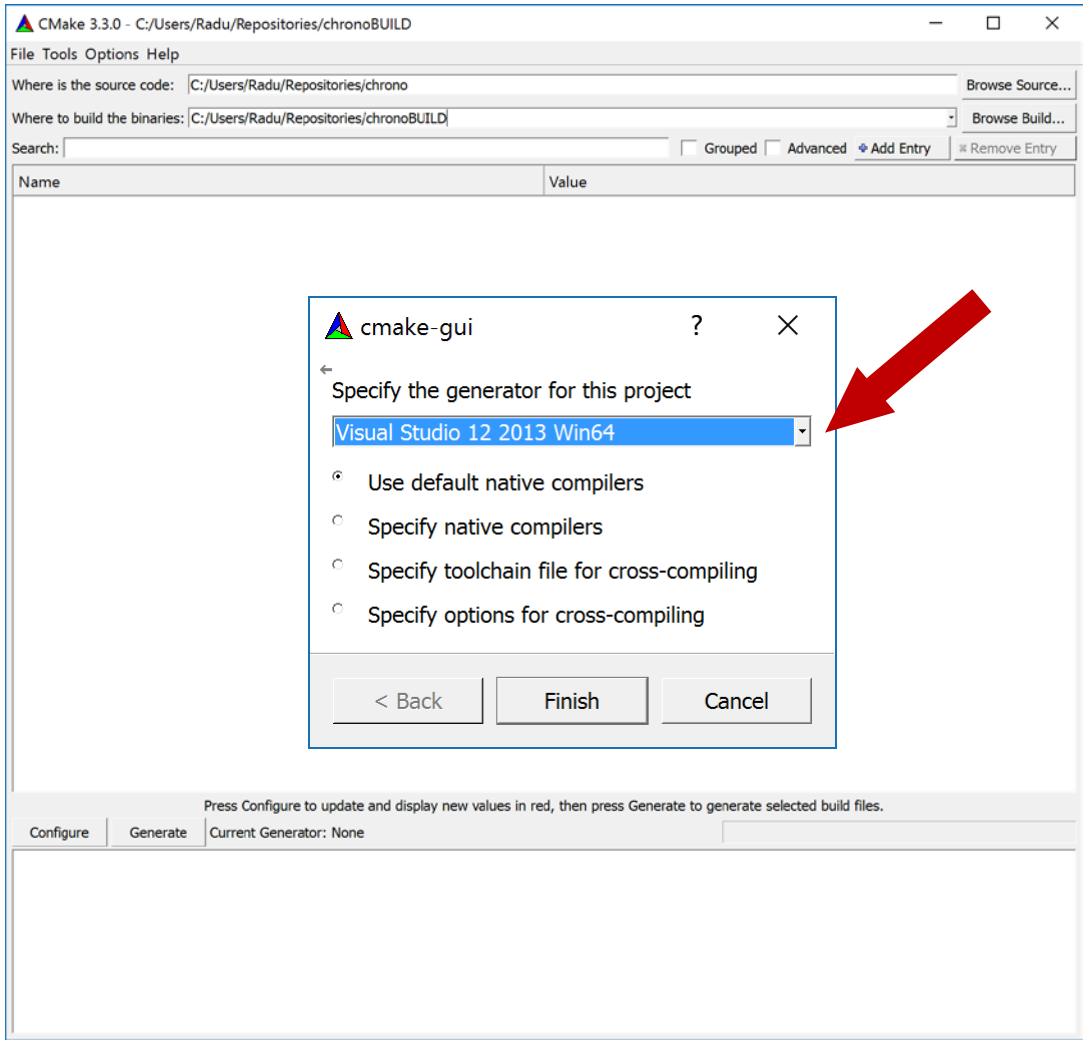
Chrono CMake configuration

Using cmake-gui on Windows

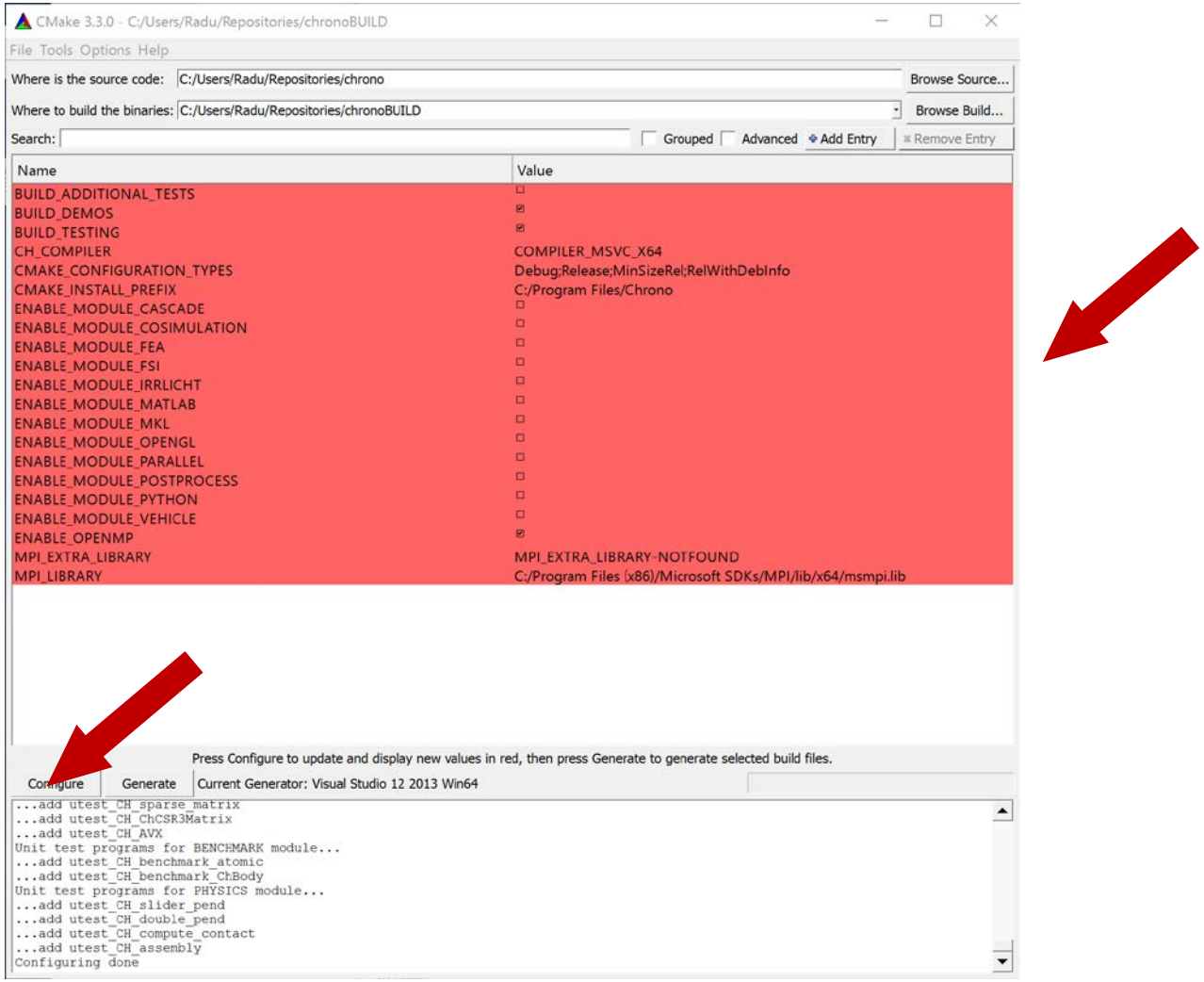
Specify source and build locations



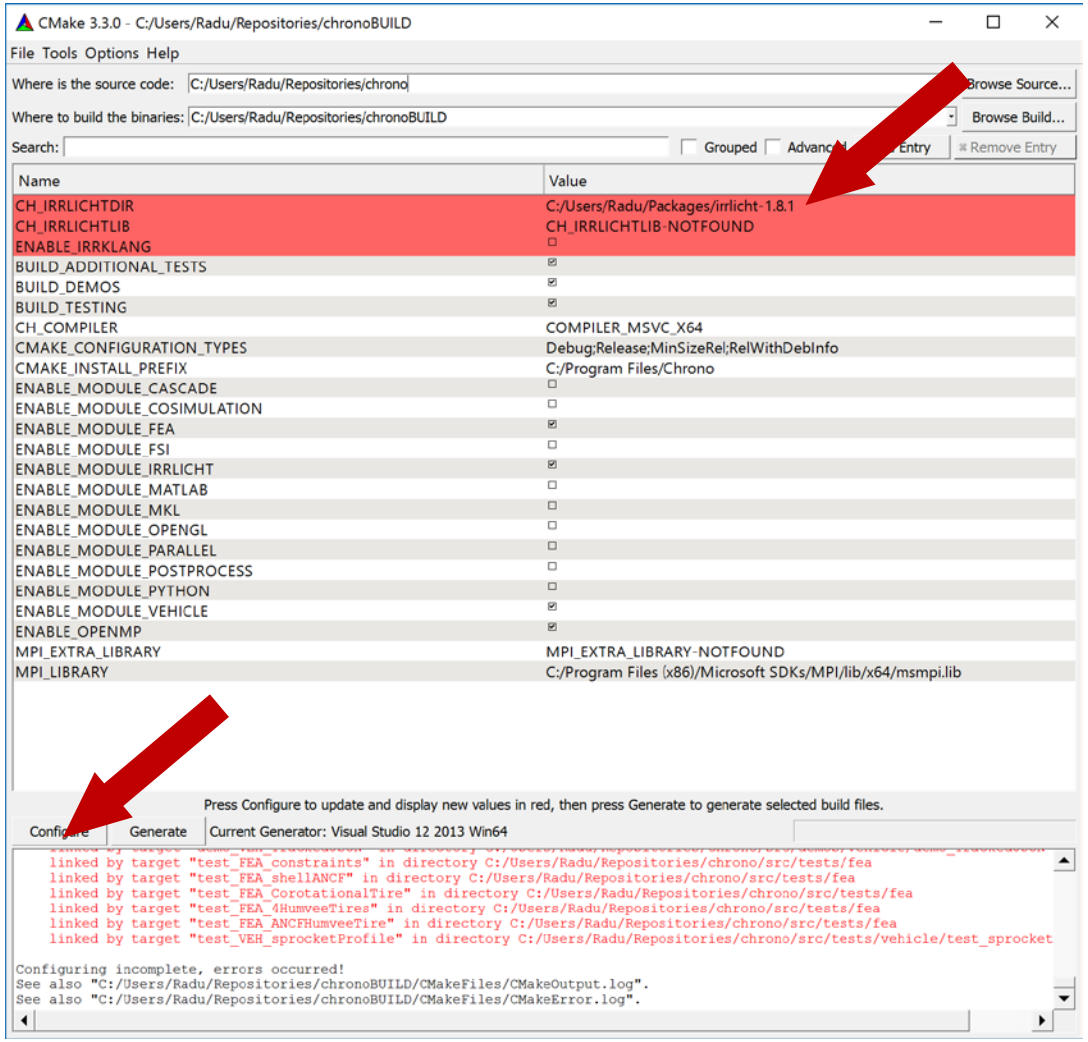
Select generator



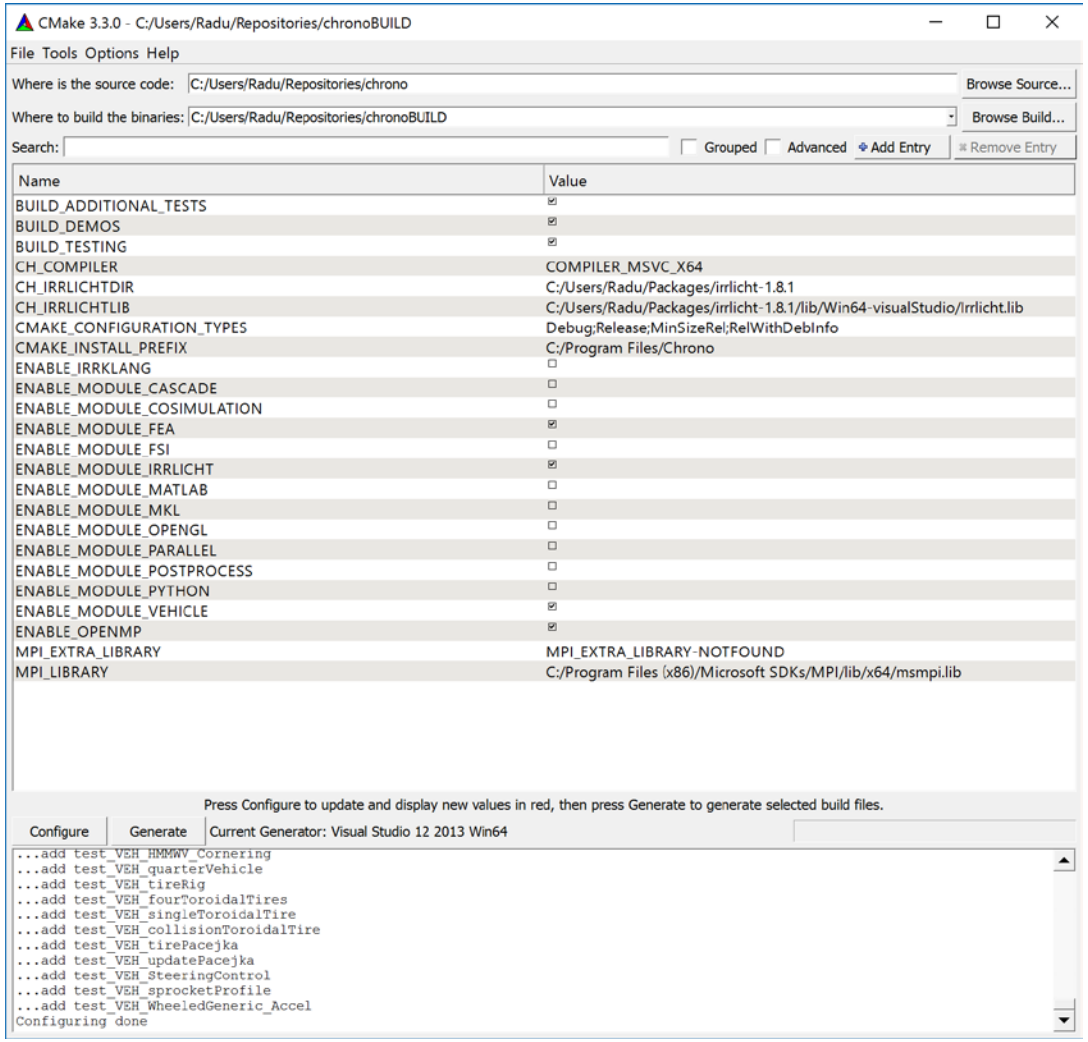
Set configuration options



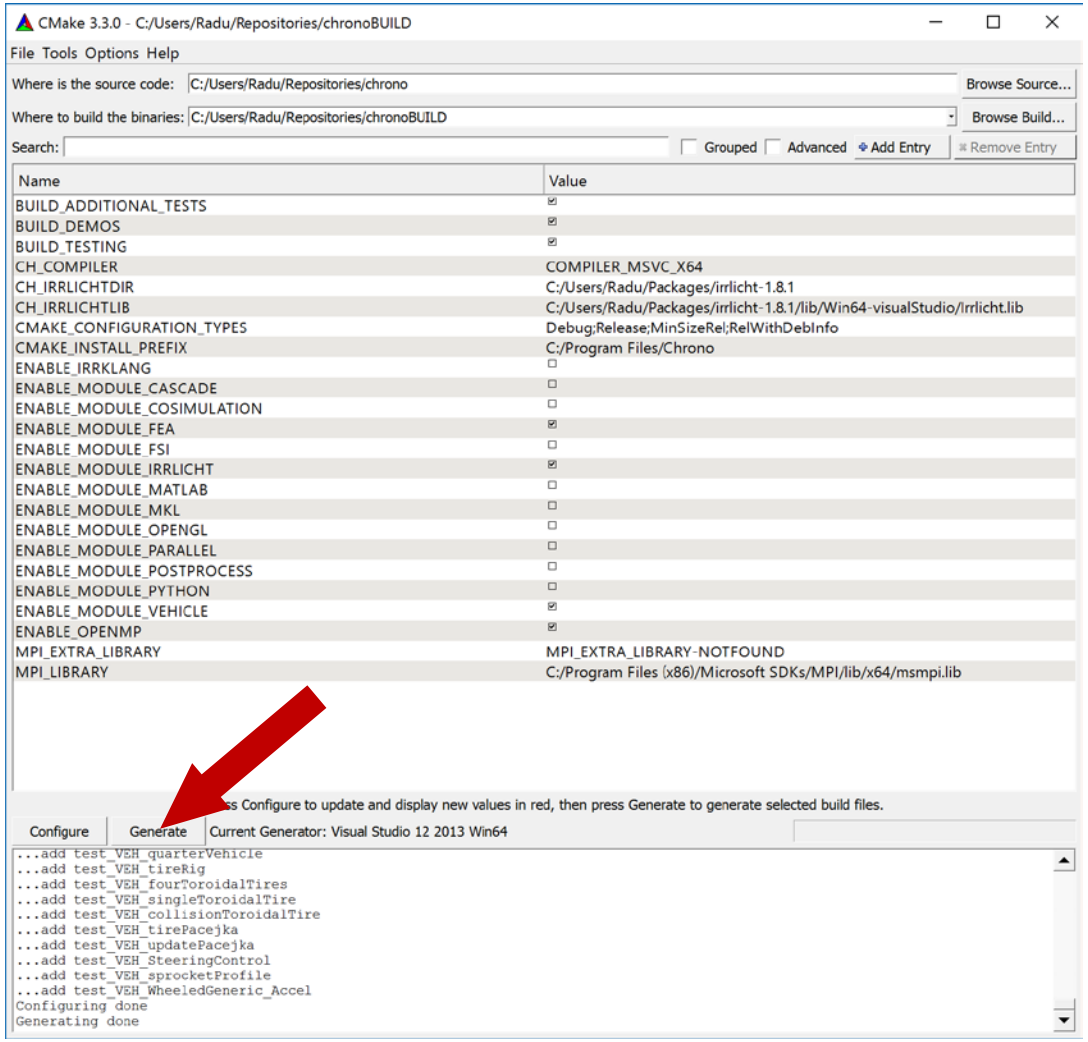
Set new configuration options



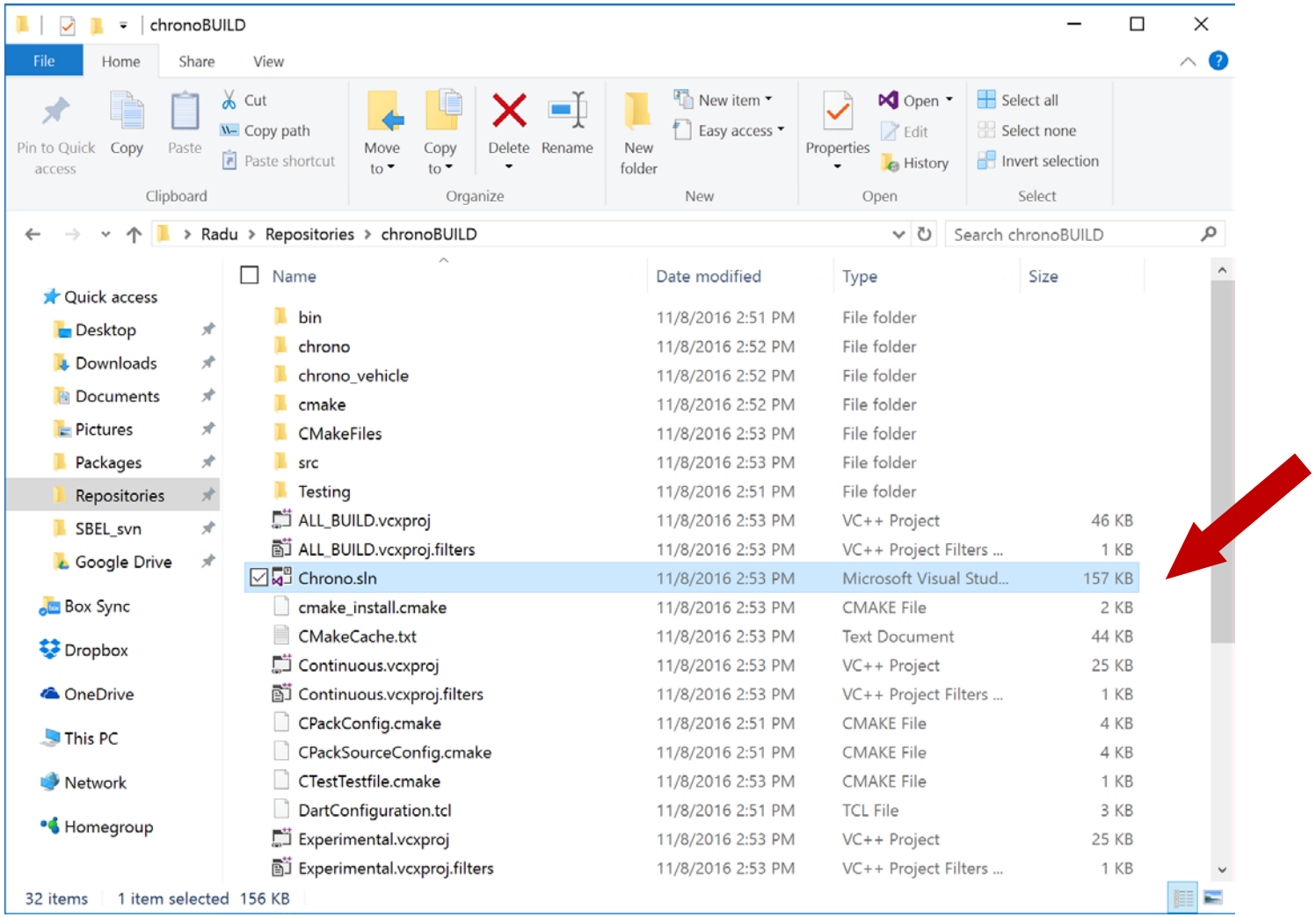
Iterate until all dependencies are resolved



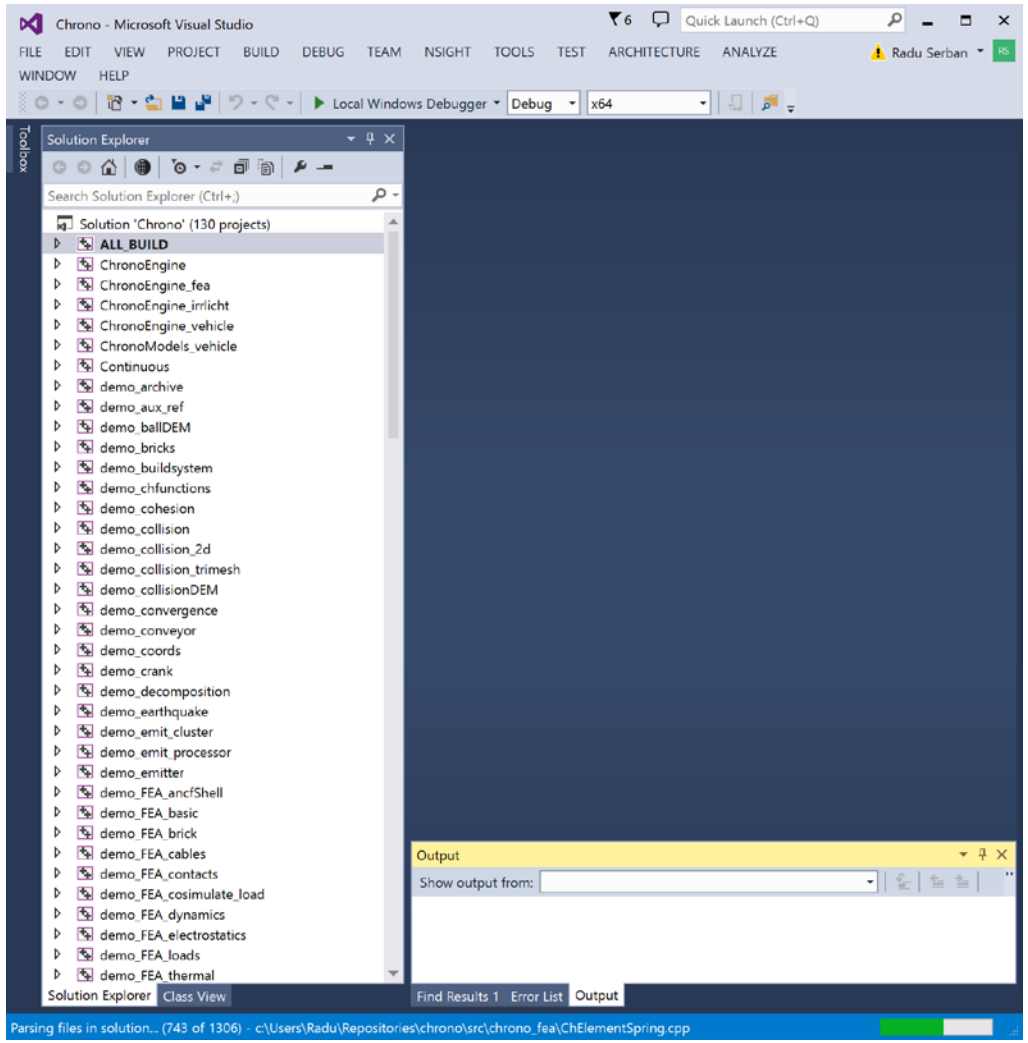
Generate native build files (VS solution file)



Generated solution file



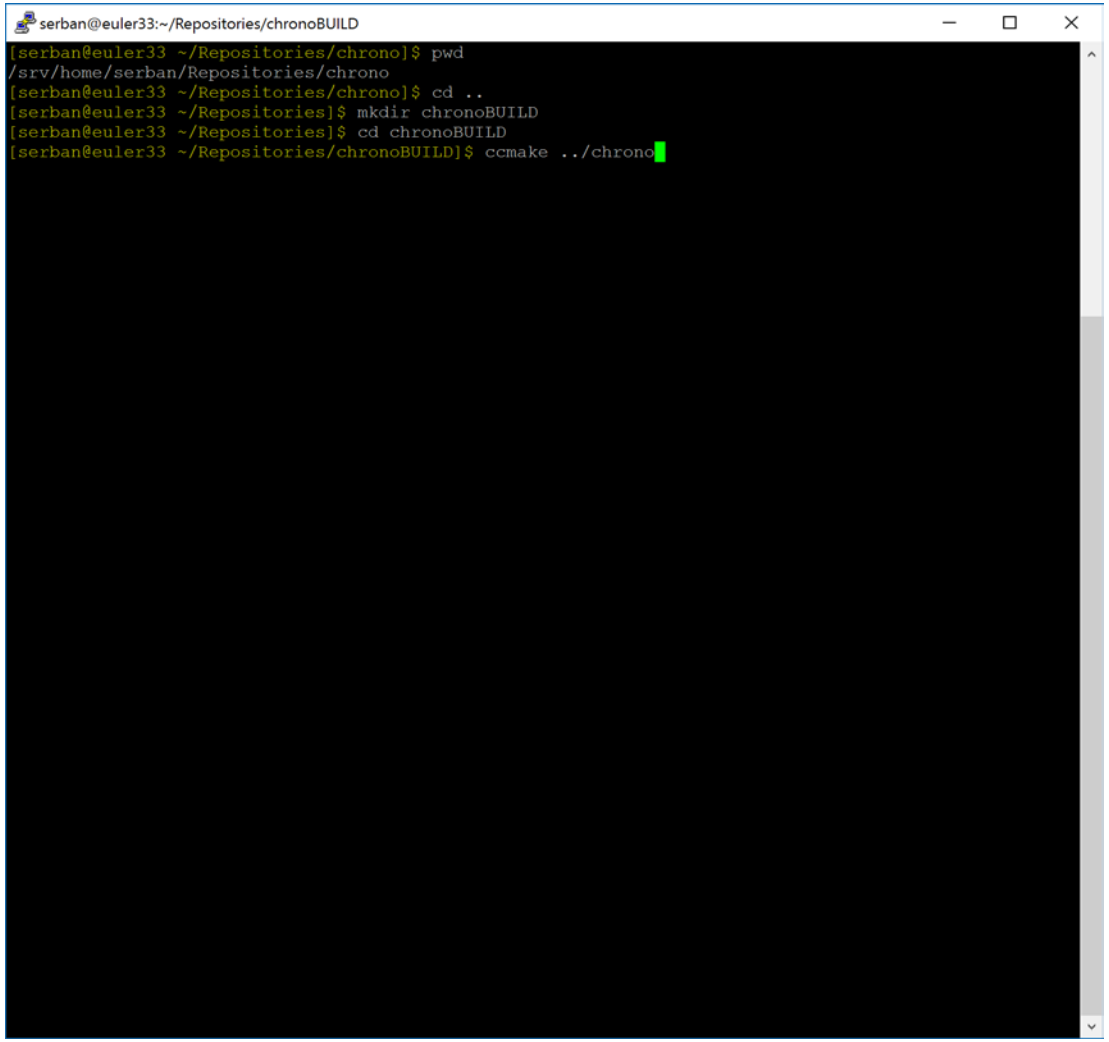
Build project



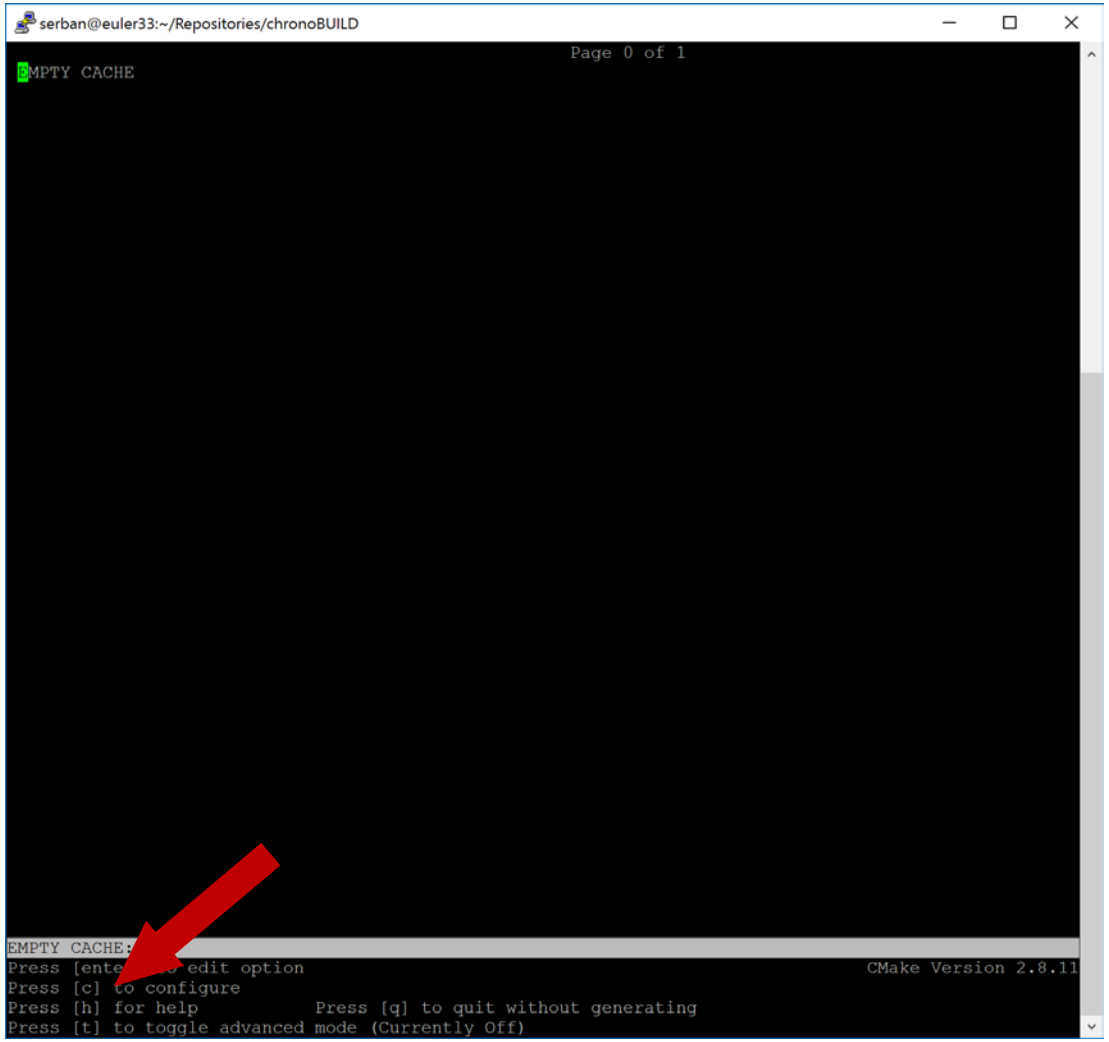
Chrono CMake configuration

Using ccmake on Linux

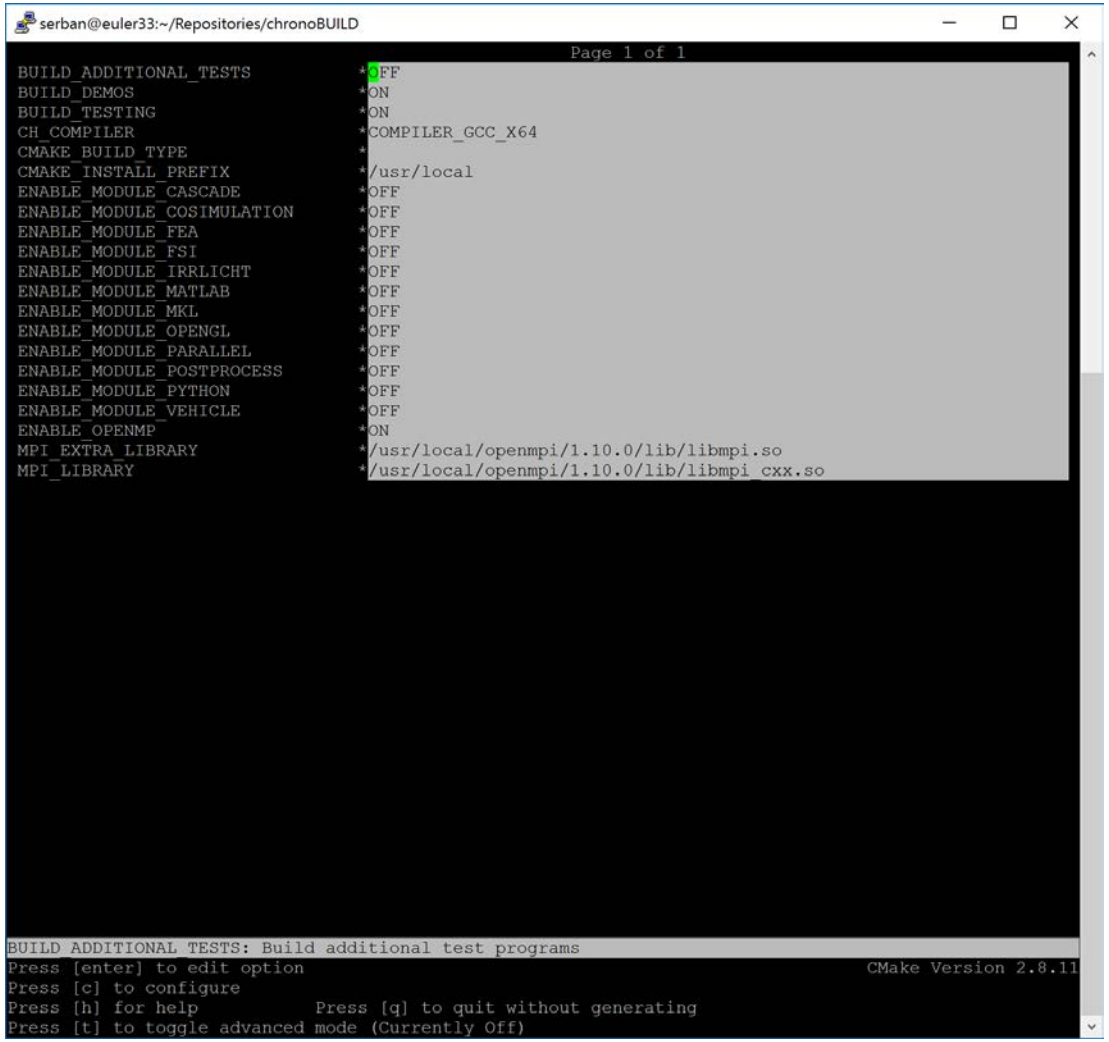
Prepare out-of-source build



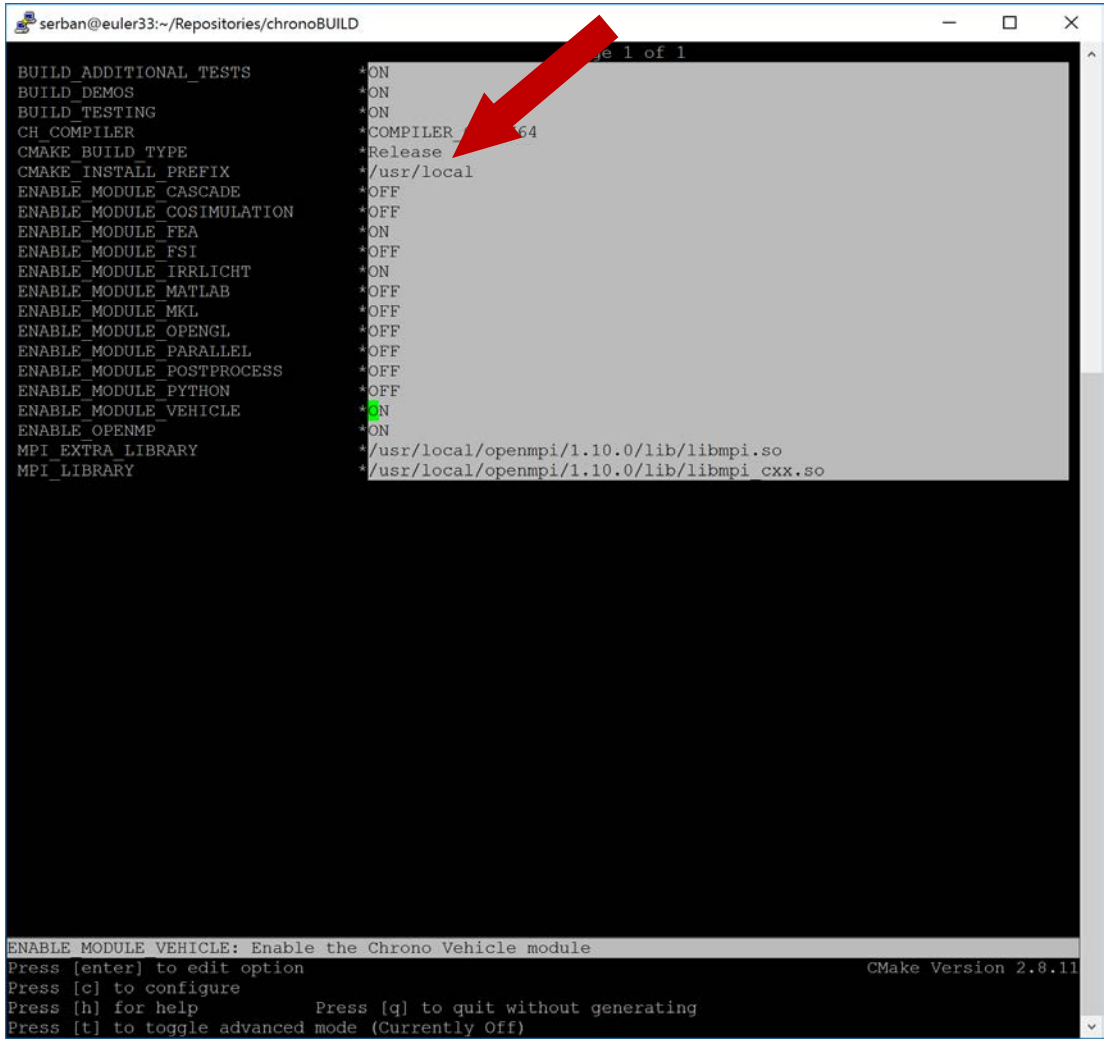
Initial configuration



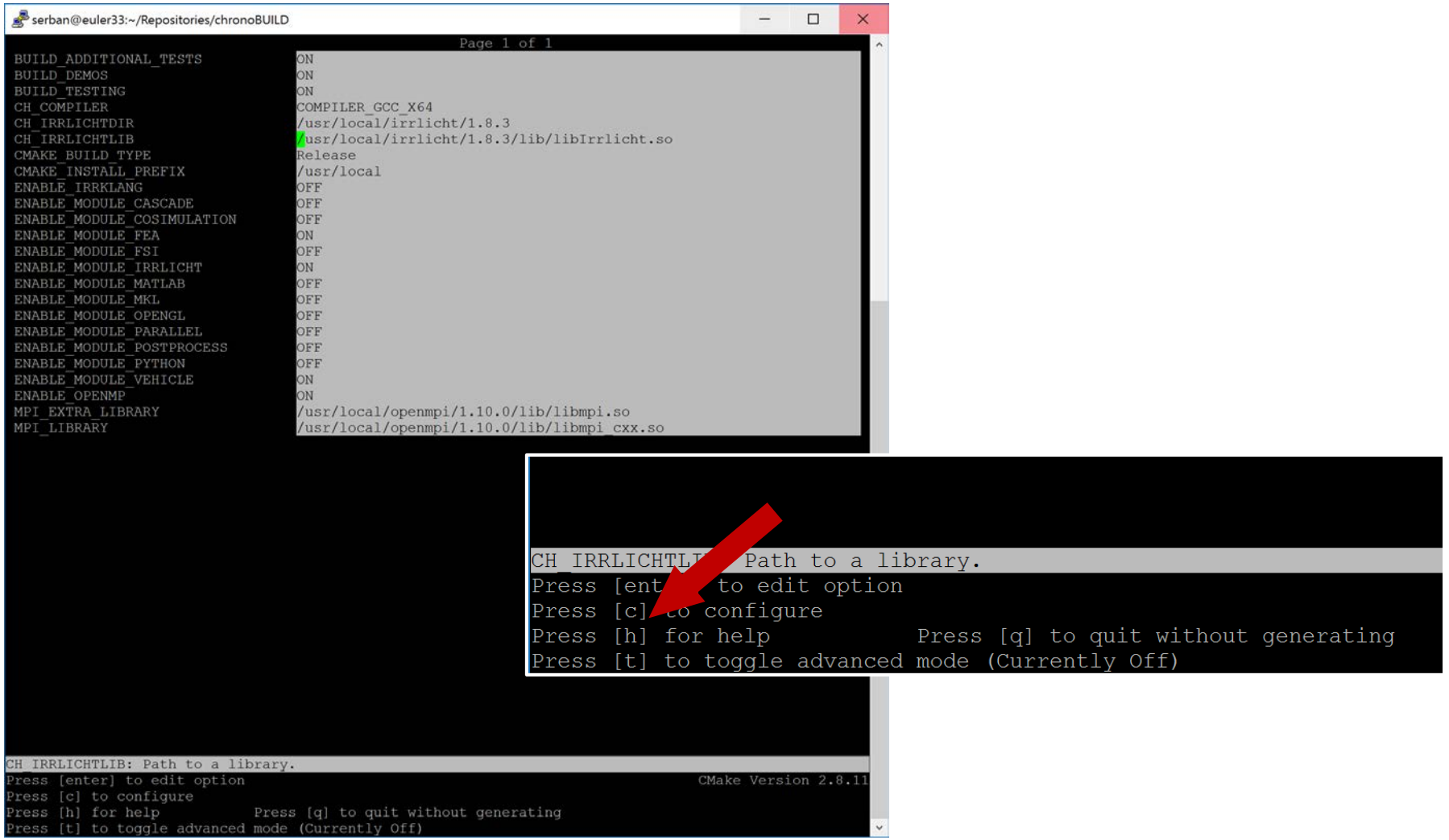
Default settings



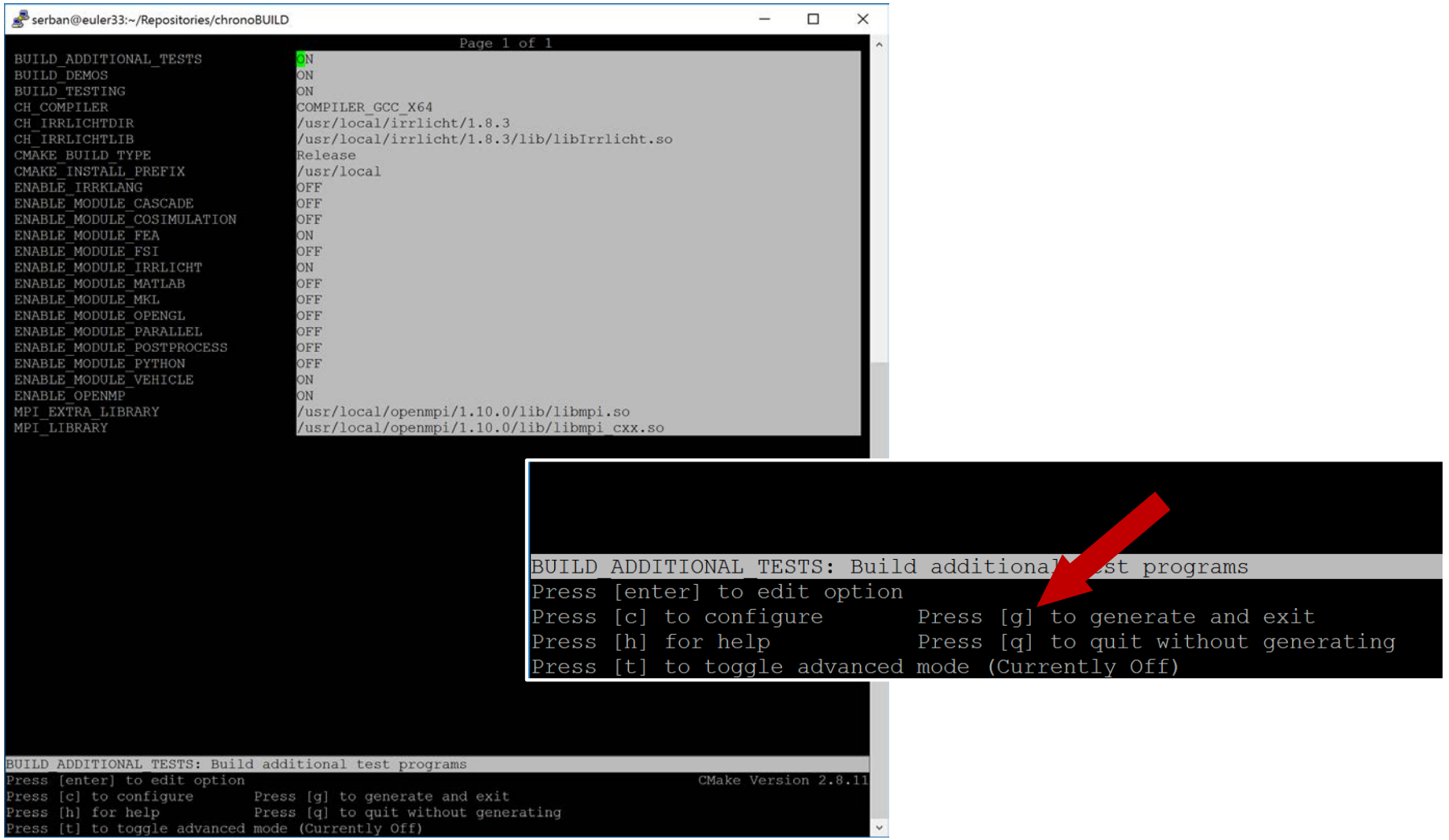
Set options


serban@euler33:~/Repositories/chronoBUILD
BUILD_ADDITIONAL_TESTS *ON
BUILD_DEMOS *ON
BUILD_TESTING *ON
CH_COMPILER *COMPILER_
CMAKE_BUILD_TYPE *Release
CMAKE_INSTALL_PREFIX */usr/local
ENABLE_MODULE_CASCADE *OFF
ENABLE_MODULE_COSIMULATION *OFF
ENABLE_MODULE_FEA *ON
ENABLE_MODULE_FSI *OFF
ENABLE_MODULE_IRRLICHT *ON
ENABLE_MODULE_MATLAB *OFF
ENABLE_MODULE_MKL *OFF
ENABLE_MODULE_OPENGL *OFF
ENABLE_MODULE_PARALLEL *OFF
ENABLE_MODULE_POSTPROCESS *OFF
ENABLE_MODULE_PYTHON *OFF
ENABLE_MODULE_VEHICLE *ON
ENABLE_OPENMP *ON
MPI_EXTRA_LIBRARY */usr/local/openmpi/1.10.0/lib/libmpi.so
MPI_LIBRARY */usr/local/openmpi/1.10.0/lib/libmpi_cxx.so
ENABLE_MODULE_VEHICLE: Enable the Chrono Vehicle module
Press [enter] to edit option
CMake Version 2.8.11
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)

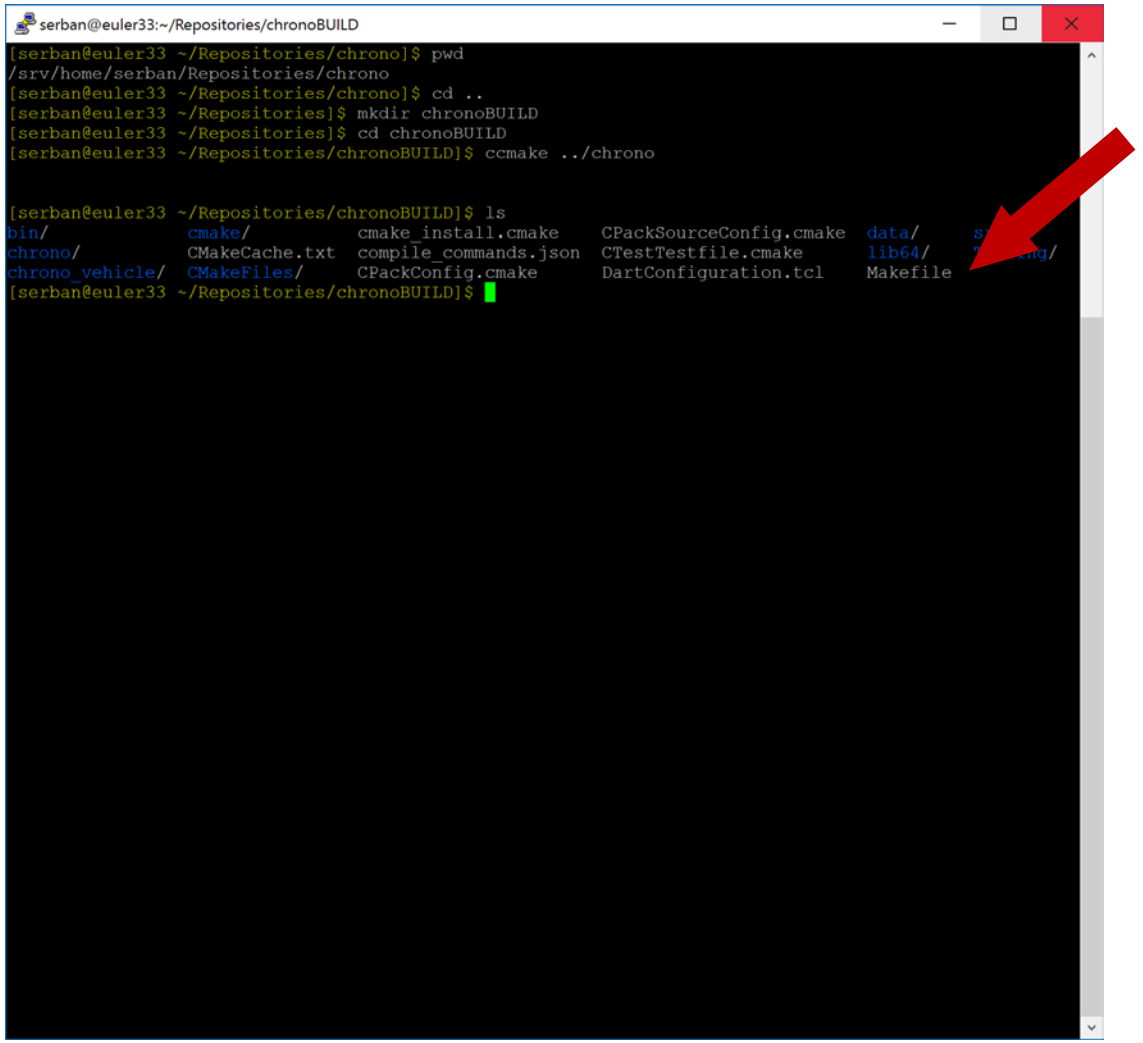
Iterate

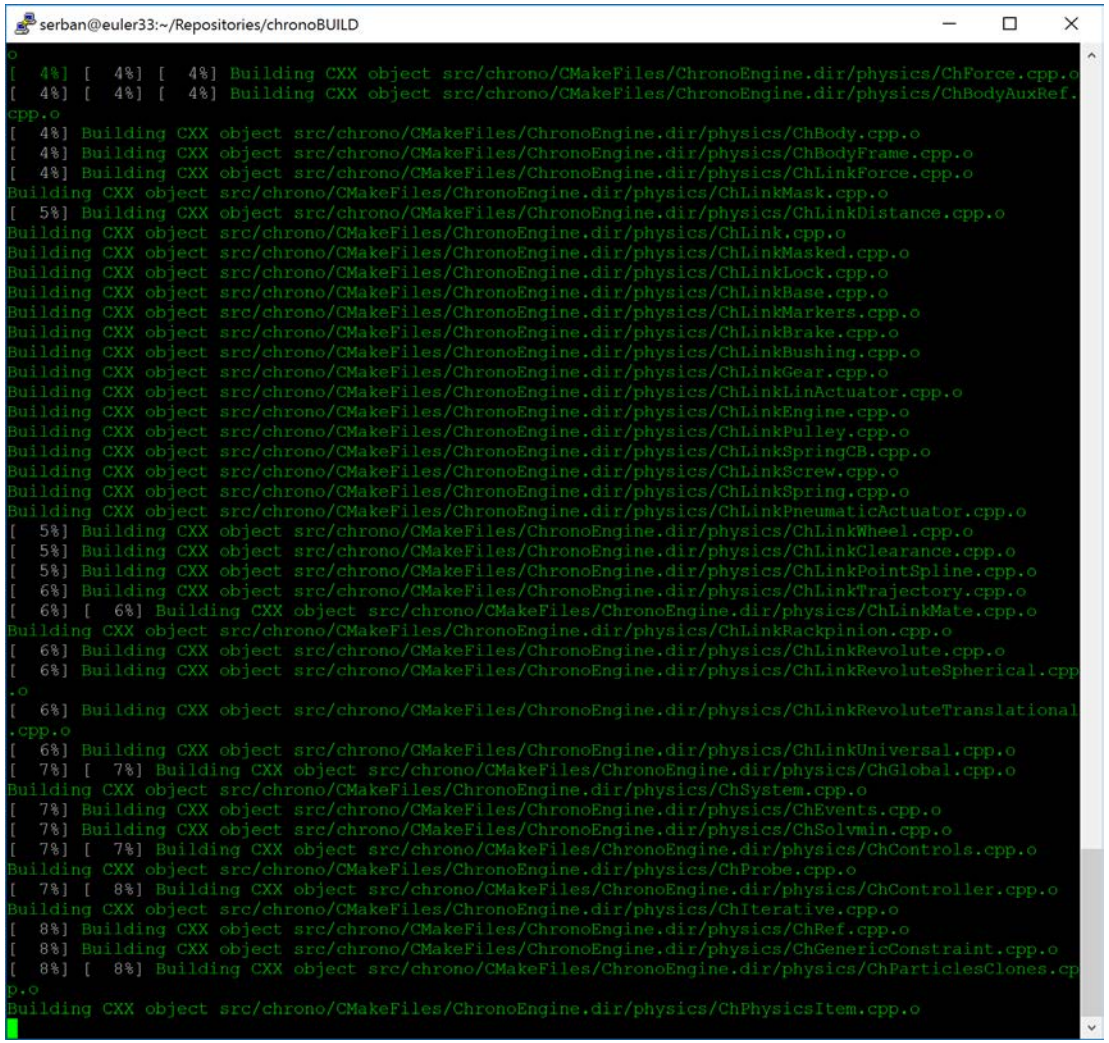

serban@euler33:~/Repositories/chronoBUILD
Page 1 of 1
BUILD_ADDITIONAL_TESTS ON
BUILD_DEMOS ON
BUILD_TESTING ON
CH_COMPILER COMPILER_GCC_X64
CH_IRRLICHTDIR /usr/local/irrlicht/1.8.3
CH_IRRLICHTLIB /usr/local/irrlicht/1.8.3/lib/libIrrlicht.so
CMAKE_BUILD_TYPE Release
CMAKE_INSTALL_PREFIX /usr/local
ENABLE_IRRKLANG OFF
ENABLE_MODULE_CASCADE OFF
ENABLE_MODULE_COSIMULATION OFF
ENABLE_MODULE_FEA ON
ENABLE_MODULE_FSI OFF
ENABLE_MODULE_IRRLICHT ON
ENABLE_MODULE_MATLAB OFF
ENABLE_MODULE_MKL OFF
ENABLE_MODULE_OPENGL OFF
ENABLE_MODULE_PARALLEL OFF
ENABLE_MODULE_POSTPROCESS OFF
ENABLE_MODULE_PYTHON OFF
ENABLE_MODULE_VEHICLE ON
ENABLE_OPENMP ON
MPI_EXTRA_LIBRARY /usr/local/openmpi/1.10.0/lib/libmpi.so
MPI_LIBRARY /usr/local/openmpi/1.10.0/lib/libmpi_cxx.so
CH_IRRLICHTLIB: Path to a library.
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
CMake Version 2.8.11
CH_IRRLICHTLIB: Path to a library.
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)

Generate native build files (Unix makefiles)



Generated makefile



Build project

% make