CMAKE TUTORIAL

# Basics

From [1] we found out about basics from cmake (cross-platform tool for defining and managing code builds- single definition of how the project is built).

* Generates build input for different specific platforms (unix ->makefiles, windows -> visual studio projects), case insensitive
* CMakeLists.txt is used to configure the builds, one per directory of source code.
* Generator of cmakefiles build types:
* Visual studio supports multiple configuration (Debug-not optimized, Release, RelWithDebInfo, MinSizeRel), it will generate all configurations at once and with –config swich we are deciding which one to build
* “Unix Makefiles” generator support single configuration with user defined variable *CMAKE\_BUILD\_TYPE*.
* Command **project** sets the PROJECT\_NAME variable and other (PROJECT\_SOURCE\_DIR, PROJECT\_BINARY\_DIR <https://cmake.org/cmake/help/v3.0/command/project.html#command:project>
* *cmake – E -> command line tool mode* <https://cmake.org/cmake/help/v3.7/manual/cmake.1.html#command-line-tool-mode>
* *cmake –build -> build tool mode*
* softwares found in /man -> **cmake, ccmake, cpack, ctest, cmake-gui**
* variables: ${var}, multiple arguments using set command
* cmake-buildsystem (7) is shown in cmake path (man) but don’t know how to run it <https://cmake.org/cmake/help/latest/manual/cmake-buildsystem.7.html#introduction>
* <https://cmake.org/cmake/help/latest/manual/cmake-language.7.html#organization>
* After testing cmake **–i** .. (out of source interactive mode, which is no longer supported) found out that **–D** is used to set cache values of the command line. For terminal based support we should use ccmake.

# Install mariadb on windows:

Create OUS (out of source) build with default MSVSC (visual studio compiler):

**mkdir bld**

**cd bld**

**cmake ..**

**cmake --build . –config Debug # make –j8 is not working without gcc**

Problem with windows in mariadb in order to use gcc/g++ which is too foreign and hostile for windows:

* Cannot use MinGW (minimalistic gnu for windows) generator in mariaDB (**cmake .. –G “MinGW Makefiles” –DCMAKE\_BUILD\_TYPE=Debug**) (default is RelWithDebInfo), even if mingw32-pthreads-win32 from MinGW Installation Manager (mingv-setup.exe) is supported.

**Still problem with CTRL-C in windows.**

# *# Out-of-source build: mkdir bld, cd bld, cmake .. (by default we are using MSVSC generator), cmake --build . --config Debug*

# *# Output will be visible in bld/Debug/app.exe*

# *# To remove dir in windows => rd /s /q "bld"*

# *# To echo dir in windows => echo %cd%*

# *# To show process in windows => tasklist*

# *# To kill process in windows => taskkill /F /PID #PID*

# Literature

[1] <https://riptutorial.com/cmake>

[2.1] <https://cmake.org/cmake-tutorial/>

[2.2] <https://gitlab.kitware.com/cmake/cmake/blob/master/Help/guide/tutorial/index.rst>