



add_executable()

- executable name
- define multiple executables by calling multiple times
- BEST PRACTICE:** Target name independent of project name

target_link_libraries()

- Always specify
- PRIVATE: A uses B internally
- PUBLIC: A uses B internally & in its interface
- INTERFACE: A doesn't require B internally, it's only used in its interface
- Can also specify:
 - full path to library file
 - plain library name
 - link flag (**DISCOURAGED!!!**)
 - Avoid using debug, optimized, and general

project()

- Results cached in...
- can be omitted
- make_minimum_required(VERSION major minor [patch] [version])
- Required
- Version 3.5 would be the absolute oldest
- Later versions give more OS distribution version

Policy Mechanisms

- Should be first line of CMakeLists.txt
- cmake_minimum_required(VERSION major minor [patch] [version])
- Specify major

semantic_versioning

- Twice about versioning the early

Commands

: Line

#[] : Block

CMakelists.txt

CMakeCache.txt

Project File Generation

Build

Test

Pack

Knit

Cache

Config

Concise

For multi-config

generators only

Multiple targets may

be specified in CMake 3.15+

Very useful for automation

Points to build directory

My App

Cmake --build /path/to/build --config Debug --target My App

Get familiar w/ these CLI tools

ctest

- Test scheduling & reporting
- Exit code 0 generally considered passing
- Supports parallel tests (-j)
- use build target

ctest

enable_testing()

- Call after project

add_test()

- Define test case

NAME: COMMAND

- Shell
- CMake executable target

cpack

• Produce binary packages

• Internally cmake --install of

- product to temp staging area

• CPack module

cpack -G ZIP;WIX [-c config Release]

• package build target

CPACK_GENERATORS

- include(CPack)

Avoid installs straight from build tree.. package then install (2 steps)

Cmake -G "Unix Makefiles" -B build

Generator Type

Build Directory

Default is host platform if omitted.

CMAKE_GENERATOR

- (CMake 3.15+)