CMakeEssential

Table of Contents

Copyright

{copyright}

About

Essential CMake snippets for software development with modern CMake

Documentation

TODO!

Usage

git subtree

CMake FetchContent

git submodule

Copy & Paste

The methods above are preferred for sure. But as a last resort, you may of course use the

CMakeEssential snippets by storing them into your project’s file structure.

If your project has a Directory Layout similar to this project’s, the right place should be `external/cmake-essential`.

Contribution

Issues

This project uses the /-/issues[lightweight issue tracker] provided by GitLab. There is no corresponding Mantis bug-tracker project.

Directory Layout

CMakeLists.txt and its includes try to follow ideas from The optimal CMake project structure by Stanislav Arnaudov.

More importantly, this project aims to adhere to The Pitchfork Layout (PFL) by Colby Pike. This is also the source for the following directory description (though they were adapted to this specific project):

PFL prescribes several directories that should appear at the root of the project tree. Not all of the directories are required, but they have an assigned purpose, and no other directory in the filesystem may assume the role of one of these directories. That is, these directories must be the ones used if their purpose is required.

Other directories should not appear at the root.

build/

A special directory that should not be considered part of the source of the project. Used for storing ephemeral build results. Must not be checked into source control. If using source control, must be ignored using source control ignore-lists. — So .gitignore has an entry for it.

src/

Main compilable source location. Must be present for projects with compiled components that do not use submodules. In the presence of include/, also contains private headers. — empty

include/

Directory for public headers. May be present. May be omitted for projects that do not distinguish between private/public headers. May be omitted for projects that use submodules. — Contains the "essential" *.cmake files required by projects that consume CMakeEssential.

tests/

Directory for tests. — empty

examples/

Directory for samples and examples. — empty

external/

Directory for packages/projects to be used by the project, but not edited as part of the project. — empty

extras/

Directory containing extra/optional submodules for the project. — empty

data/

Directory containing non-source code aspects of the project. This might include graphics and markup files. — Contains the logo.

tools/

Directory containing development utilities, such as build and refactoring scripts. — empty

docs/

Directory for project documentation. — Contains README.in.adoc and LICENSE.in.adoc.

libs/

Directory for main project submodules. The libs/ directory must not be used unless the project wishes to subdivide itself into submodules. Its presence excludes the src/ and include/ directories. — Unused!

File Name Conventions

Who has ever tried to find the underscore on a foreign keyboard? I (Max) have, and it is cumbersome.

So, here are the most important rules for filenames:

1. Filename rules apply to names of folders.
2. No whitespace in filenames.
3. No "foreign" characters in filenames. Stick to `[a-zA-Z0-9_.-+]`.
4. Avoid unnecessary abbreviations.
5. Separate words preferably with a single dash `"-"`.
6. No underscores in examples.
7. No underscores in executables. — Neither in scripts, which are executables for that matter.
8. No uppercase in executables. — Not even `"IVEN2"` :-(
9. Avoid uppercase letters, whenever it's not (really) helping.
    1. Exception: `README.adoc`, `LICENSE.adoc` — these must stick out.
    2. Exception: `CMakeFile` and friends — it's conventional for those.
    3. Exception: QML-files — Qt requires it.
10. No extension for (bash) scripts — Because Commandname Extensions [are] Considered Harmful.

11. TODO: coding guidelines for module / class names and their filenames.

Conventional Commits

Developers are encouraged to adhere to the Conventional Commits "[…] specification for adding human and machine readable meaning to commit messages".