

CMake file example for building a C++ project

CMake is cross-platform free and open-source software for build automation, testing, packaging and installation of software by using a compiler-independent method. CMake is not a build system itself; it generates another system's build files.

Below is a simple example for C++ project:

```
cmake_minimum_required(VERSION 3.10)
project(ExampleProject)

# Add the source files
set(SRCS main.cpp)

# Add the executable
add_executable(ExampleProject ${SRCS})
```

The project name is `ExampleProject` and the source file for the project is `main.cpp`. The executable is added using the `add_executable` command, with the project name and source files as arguments.

Below is an advanced example:

```
cmake_minimum_required(VERSION 3.10)
project(ExampleProject CXX)

# Set the C++ standard to use
set(CMAKE_CXX_STANDARD 17)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

# Set the build type to release by default
if(NOT CMAKE_BUILD_TYPE)
    set(CMAKE_BUILD_TYPE Release)
endif()

# Add the source files
file(GLOB_RECURSE SRCS "src/*.cpp")

# Add the include directories
include_directories(include)

# Add the executable
add_executable(ExampleProject ${SRCS})

# Link libraries if needed
# target_link_libraries(ExampleProject [library1 library2 ...])

# Enable testing
enable_testing()

# Add tests
# add_test(NAME ExampleTest COMMAND ExampleProject [arguments])

# Install the executable
install(TARGETS ExampleProject DESTINATION bin)
```

For above, the project name is `ExampleProject` and is built as an executable. The C++ standard used is C++17. The build type is set to `Release` by default, but can be changed by specifying a different type when running CMake. The source files are located in the `src` directory and are added using the `file(GLOB_RECURSE ...)` command. The include directories are added using the `include_directories` command. The executable is added using the `add_executable` command. If the project requires any libraries, they can be linked using the `target_link_libraries` command. The project can be installed using the `install` command. The `enable_testing` command enables testing and individual tests can be added using the `add_test` command.