

Introduction to Modern CMake

C++ dependency tutorial

2022-06-02

Guilhem Saurel

Available at

<https://homepages.laas.fr/gsaurel/talks/modern-cmake.pdf>

Under License



<https://creativecommons.org/licenses/by-sa/4.0/>

Source

`https://gitlab.laas.fr/gsaurel/talks :
modern-cmake.md`

Discussions

`https://im.laas.fr/#/room/#modern-cmake:laas.fr`

Configure

- Find the dependencies you need
- Declare the options you want

Configure

- Find the dependencies you need
- Declare the options you want

Compile

- Generate the right arguments for your compiler
- Run it

Configure

- Find the dependencies you need
- Declare the options you want

Compile

- Generate the right arguments for your compiler
- Run it

Run

- Start your executable with its associated *.so

Example

- You develop a library called “my-lib”
- It needs a dependency from someone else called “a-dep”

- You develop a library called “my-lib”
- It needs a dependency from someone else called “a-dep”

```

└─ $A_DEP_INSTALL_PREFIX
    └─ include
        └─ a-dep
            └─ fwd.hpp
    └─ lib
        └─ liba-dep.so

```


FindA-dep.cmake

```
find_path(a_dep_INCLUDE_DIRS NAMES a-dep/fwd.hpp)  
find_library(a_dep_LIBRARIES NAMES liba-dep.so)
```

FindA-dep.cmake

```
find_path(a_dep_INCLUDE_DIRS NAMES a-dep/fwd.hpp)  
find_library(a_dep_LIBRARIES NAMES liba-dep.so)
```

CMakeLists.txt

```
find_package(a-dep REQUIRED)  
include_directories(${a_dep_INCLUDE_DIRS})  
link_libraries(${a_dep_LIBRARIES})  
add_library(my-prog main.cpp include/my-proj/fwd.hpp)
```

CMakeLists.txt

```
find_package(a-dep REQUIRED)
add_library(my-lib main.cpp include/my-prog/fwd.hpp)
target_link_libraries(my-prog PUBLIC a-dep::a-dep)
```

a-dep-config.cmake

```
add_library(a-dep::a-dep SHARED IMPORTED)
set_target_properties(a-dep::a-dep PROPERTIES
    LINK_LIBRARIES "liba-dep.so"
    INCLUDE_DIRECTORIES "${_IMPORTED_PREFIX}/include")
```

- 1 a-dep can change whatever it needs

- 1 a-dep can change whatever it needs
- 2 No more naming issues like

```
set(A_DEP_INCLUDE_DIRS ${a_dep_INCLUDE_DIRS})  
set(A_DEP_INCLUDE_DIR ${a_dep_INCLUDE_DIRS})  
set(a_dep_INCLUDE_DIR ${a_dep_INCLUDE_DIRS})
```

1 a-dep can change whatever it needs

2 No more naming issues like

```
set(A_DEP_INCLUDE_DIRS ${a_dep_INCLUDE_DIRS})  
set(A_DEP_INCLUDE_DIR ${a_dep_INCLUDE_DIRS})  
set(a_dep_INCLUDE_DIR ${a_dep_INCLUDE_DIRS})
```

3 Relocatable packages

my-lib-config.cmake from legacy a-dep

```
add_library(my-lib::my-lib SHARED IMPORTED)
set_target_properties(my-lib::my-lib PROPERTIES
  LINK_LIBRARIES "libmy-lib.so;/usr/lib/liba-dep.so"
  INCLUDE_DIRECTORIES
  "${_IMPORTED_PREFIX}/include;/usr/include")
```

my-lib-config.cmake from legacy a-dep

```
add_library(my-lib::my-lib SHARED IMPORTED)
set_target_properties(my-lib::my-lib PROPERTIES
  LINK_LIBRARIES "libmy-lib.so;/usr/lib/liba-dep.so"
  INCLUDE_DIRECTORIES
    "${_IMPORTED_PREFIX}/include;/usr/include")
```

my-lib-config.cmake from modern a-dep

```
find_dependency(a-dep REQUIRED)
add_library(my-lib::my-lib SHARED IMPORTED)
set_target_properties(my-lib::my-lib PROPERTIES
  LINK_DEPENDENT_LIBRARIES "a-dep::a-dep"
  LINK_LIBRARIES "libmy-lib.so"
  INCLUDE_DIRECTORIES "${_IMPORTED_PREFIX}/include")
```


That's all folks !

Thanks for your time :)