

Hunter

Hunter

a CMake driven cross-platform package manager
for C/C++ projects

Hunter is one of many package/dependancy -manager

- * Conan
- * vcpkg
- * Homebrew (OSX)
- * Meso
- * Spack
- * Buckaroo
- * cget
- * CocoaPods
- * mxxn
- * ... many more?

- * CMake – Based
- * No external other dependencies
- * Easy to use
- * Easy to extend
- * No binaries shipped
- * OpenSource
- * Created and maintained by Ruslan Baratov

Who should care about Hunter?

- * If you already using CMake
- * If you want a new/better way for getting/managing dependencies
- * Just want to try new stuff :-)

How to compile a Hunter based Project?

Like you would compile a CMake project, NO need to install any new software.

```
:> cd <project-folder>  
:> mkdir build  
:> cd build  
:> cmake ...  
:> make
```

How to use Hunter in my Project?

- * add the HunterGate.cmake to you project
- * inside CMakeLists.txt

```
cmake_minimum_required(VERSION 3.9)

### Hunter „initial implementation“ ###
include("cmake/HunterGate.cmake")

### Hunter snapshot that will be used ###
HunterGate(
  URL "https://github.com/ruslo/hunter/archive/v0.23.33.tar.gz"
  SHA1 "acbf4b9b77b5a0160adc1da1cdda35f64b14196a"
)

project(HunterSimple)
```

How to use Hunter in my Project?

- * declare what you need
- * and use dependency as know

```
#### Download dependencies ####  
hunter_add_package(GTest)  
  
#### Find dependencies ####  
find_package(GTest CONFIG REQUIRED) # GTest::main  
  
#### Targets ####  
add_executable(simple simple.cpp)  
target_link_libraries(simple PUBLIC GTest::main)
```


Facts about hunter

- * approx. 400 and more packages
 - * boost, qt, Gtest, spdlog, folly, fmt, sqlite,
- * package-list updated often(every 2/3 days)
- * Many more features
 - * Easy to make your own package-list
 - * Support for a cache-server
 - * Awesome Documentation/Examples/Tests

Online Links:

Main Project:

<https://hunter.sh>

Good Examples:

<https://github.com/forexample/hunter-simple>

Jason Turner weekly Hunter:

https://www.youtube.com/watch?v=O2_N8OzPGWQ