Hunter Documentation

Release 0.23

Ruslan Baratov

Jul 31, 2023

Contents

Welcome to the Hunter package manager documentation!

Hunter is a CMake driven cross-platform package manager for C/C++[1] projects. With the help of Hunter you can organize builds for **Linux**, **Windows**, **macOS**, **iOS**, **Android**, **Raspberry Pi** and other platforms. Third-party external projects are highly customizable, effectively allowing you to have myriad variants of directories with them based on combinations of version to build, static/shared, CMake `-D` options, Release/Debug, etc.

Separate root directories will be created for each variant, so they all can be used simultaneously on one machine without conflicts (just like virtualenv but automatically). Going further: each such root directory can be shared between several local projects if configuration of externals matches. So when you are starting another project from scratch and use the same external packages, there will be no additional copy or build triggered; the only overhead is checking the existence of a `DONE` stamp file for each package. In case your local environment is similar enough to the continuous integration environment of Travis/AppVeyor service, then build will not be triggered at all - cached binaries will be downloaded from GitHub server instead.

Mainly Hunter is designed to manage packages with CMake build system under the hood and existing CMake packages can be quite easily integrated into system, but non-CMake packages are also supported too using custom templates (build schemes) with `ExternalProject_Add` command(s).

The Hunter client is a collection of CMake-only modules (i.e. it's **not a binary** like `apt-get` or script like `brew`) so it supports out-of-the-box all platforms/generators/IDEs which CMake can handle, like Visual Studio, Xcode, *Android Studio*, QtCreator, NMake, Ninja, Cygwin or MinGW. It works fine with CMake-GUI too.

The prime directive used for adding package to the current root is `hunter_add_package` which companioning CMake's `find_package`. For example:

```
hunter_add_package(Boost COMPONENTS system filesystem iostreams)
find_package(Boost CONFIG REQUIRED system filesystem iostreams)
```

Sounds interesting? Keep reading!

[1] C++ is the main goal, works for other types as well. See *Manage anything*.

CHAPTER 1

Brief overview

This is a brief overview of big picture. It takes about 5 minutes of reading but will show you the main features/aspects of using Hunter. Please **don't make any assumptions** about how Hunter works without reading this part. Also avoid running real code for now, it will be covered in next *Quick start* section.

1.1 What is it?

Every Hunter release (Atom feed) archive is a meta-package with build instructions and URLs of real packages. The archive will be downloaded and unpacked automatically by the `HunterGate` CMake module. You only need to set the `URL` and `SHA1`:

```
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)
```

Here is the content of the archive in simplified form:

```
Hunter (0.23.297) = {
    Boost (1.65.1, 1.65.0, 1.66.0, 1.66.0-p0, 1.67, ...),
    GTest (1.7.0, ...),
    OpenCV (4.1.1-p0, 4.0.0-p3, 3.4.6-p0, ...),
    OpenSSL (1.1.1, 1.1.1a, 1.1.1b, 1.1.1c, 1.1.1d, 1.1.1g, 1.1.1g-p0, ...),
    ...
}
```

If you download and unpack it, you can view some internals. Let's look at the OpenSSL package properties:

```
wget https://github.com/cpp-pm/hunter/archive/v0.14.5.tar.gz
tar xf v0.14.5.tar.gz
```

hunter.cmake holds the list of available versions:

```
grep -A3 -m3 VERSION hunter-0.14.5/cmake/projects/OpenSSL/hunter.cmake
```

```
    VERSION
    "1.1.1"
    URL
    "https://github.com/openssl/openssl/archive/OpenSSL_1_1_1.tar.gz"
--
    VERSION
    "1.1.1a"
    URL
    "https://github.com/openssl/openssl/archive/OpenSSL_1_1_1a.tar.gz"
--
    VERSION
    "1.1.1b"
    URL
    "https://github.com/openssl/openssl/archive/OpenSSL_1_1_1b.tar.gz"
```

Install instructions from build scheme url_sha1_openssl.cmake.in:

```
grep -A1 INSTALL_COMMAND hunter-0.14.5/cmake/projects/OpenSSL/schemes/url_sha1_
↪openssl.cmake.in
```

```
INSTALL_COMMAND
make install_sw
```

Default version from default.cmake (is customizable, see *Config-ID*):

```
grep '\<OpenSSL\>' -m1 hunter-0.14.5/cmake/configs/default.cmake
```

```
hunter_default_version(OpenSSL VERSION 1.1.1)
```

See also:

- *Detailed sources layout*
- *Creating version on the fly from Git submodule*

1.2 Automatic builds

1.2.1 No dependencies in README

Build instructions from the Hunter archive are triggered automatically when the `hunter_add_package` function called, hence there is no need to specify dependencies in a raw `README` file like:

```
For OSX please do:
> brew install foo boo

For Ubuntu please do:
> sudo apt-get install foo boo

Then run build:
> cmake -H. -B_builds
> cmake --build _builds
```

Now it's simply:

```
Just run build:
> cmake -H. -B_builds # dependencies installed automatically
> cmake --build _builds
```

1.2.2 Optional dependencies

Optional dependency? No problem, optional dependencies are expressed in a straightforward way:

```
# required dependencies
hunter_add_package(foo)
hunter_add_package(boo)
if(BUILD_WITH_BAR)
  hunter_add_package(bar)
endif()
```

Now instead of:

```
Additionally if you want bar support please run:
> brew install bar # OSX
> sudo apt-get install bar # Ubuntu

Then run build:
> cmake -H. -B_builds -DBUILD_WITH_BAR=YES
```

It's simply:

```
> cmake -H. -B_builds -DBUILD_WITH_BAR=YES # + install bar
```

1.2.3 Compared to a 'requirements.txt' style approach

Note that Hunter's approach differs from a requirements.txt-like approach (i.e. when external packages are specified in a separate file). This allows Hunter to avoid duplication of logic in many cases, even if the `requirements.txt` style approach also automatically downloads dependencies too.

Imagine that we have to specify dependencies in some kind of `requirements.cmake` file and there is a user option `BUILD_WITH_BAR`:

```
# requirements.cmake

if(WIN32 AND BUILD_WITH_BAR)
  command_to_install(Bar)
endif()
```

Or, in the case that it isn't CMake code, this might by something fancy like `requirements.json`:

```
{
  "dependencies":
  {
    "package": "Bar",
    "platform": "windows",
    "cmake option": "BUILD_WITH_BAR"
  }
}
```

You would have to repeat the same condition in the `CMakeLists.txt` file:

```
# requirements.cmake

if(WIN32 AND BUILD_WITH_BAR)
  command_to_install(Bar)
endif()
```

```
# CMakeLists.txt

if(WIN32 AND BUILD_WITH_BAR)
  find_package(Bar CONFIG REQUIRED)
  target_compile_definitions(... PUBLIC "WITH_BAR")
endif()
```

Later, when you need to change this dependency in `CMakeLists.txt`, you'd better not forget to also modify `requirements.cmake` accordingly. Remember real world libraries can have nontrivial chain of conditions, e.g. OpenCV components.

Stackoverflow

- Pip: optional dependencies

1.3 Don't Repeat Yourself

If you are already familiar with ExternalProject_Add and have written some super-build projects before, you are probably already aware that writing a complete solution with toolchains, build types, build options, parallel jobs, forwarding of compiler flags, and making it work correctly for all generators is not a trivial task.

Hunter stores ExternalProject_Add recipes as a set of templates. Once written, formula (build scheme) can be reused by other projects without copying the collection of super-build files (DRY principle). When a new package with a new scheme is introduced, all you need to do is just update the `SHA1`/`URL` of `HunterGate` command.

Stackoverflow

- How to use CMake ExternalProject_Add or alternatives in a cross platform way?

1.4 Customization

You have full control of how packages will be built. You can create your own mapping of `version` -> `URL`, add globals like compiler and flags, or add new build options to external packages.

1.4.1 Hunter-ID

First level of customization. Hunter archive.

`Hunter-ID` is the first 7 digits of `SHA1` of Hunter archive. This level defines list of available packages and mapping `version -> URL/SHA1`. Several `Hunter-ID` can coexist in the same `HUNTER_ROOT` directory. `HunterGate` command will control your choice:

Hunter-ID				
1eae623	Hunter version	0.8.3		
	SHA1 of archive	1eae623cb5ce9da39c8c3e1b0f6e452f244ddc17		
	Working directory	${HUNTER_ROOT}/_Base/1eae623/...		
	Packages	Foo[1]	1.0.0	mysite.xyz/Foo-1.0.0.tar.gz
		Boo	2.0.0	mysite.xyz/Boo-2.0.0.tar.gz
			2.1.0	mysite.xyz/Boo-2.1.0.tar.gz
		Roo	1.2.3	mysite.xyz/Roo-1.2.3.tar.gz
			1.2.4	mysite.xyz/Roo-1.2.4.tar.gz
e07a124	Hunter version	0.8.4		
	SHA1 of archive	e07a124192b0a47b0b60ade40fa873a42ec27822		
	Working directory	${HUNTER_ROOT}/_Base/e07a124/...		
	Packages	Awesome	1.0.0	example.com/Awesome-1.0.0.tar.gz
		Best	2.0.0	example.com/Best-2.0.0.tar.gz
			2.0.1	example.com/Best-2.0.1.tar.gz
		Foo[1]	1.0.0	example.com/Foo-1.0.0-patch-1.tar.gz

Message in logs:

```
-- [hunter] [ Hunter-ID: 1eae623 | Toolchain-ID: ... | Config-ID: ... ]
-- [hunter] [ Hunter-ID: e07a124 | Toolchain-ID: ... | Config-ID: ... ]
```

Hunter

- Releases
- Atom feed

1.4.2 Toolchain-ID

Second level of customization. Compiler and flags.

Each build can be run with different toolchains. In general the result is a completely different root directory (containing (`lib`/`include`). For example on Windows you can simultaneously build Visual Studio (32/64), NMake, Cygwin and MinGW projects, on Linux GCC/Clang, on Mac Xcode, Makefile, iOS. Or choose various clang tools like static analyzer/sanitizers and other platforms like Android/Raspberry Pi. Each toolchain file will be forwarded to external project. This means, if you create a toolchain with compiler `g++` and flag `-std=c++11` all dependent projects will be built by `g++ -std=c++11`. Information about toolchain has some internal representation (`toolchain.info`). As identifier (ID) the first 7 digits of the `SHA1` hash of this file are used.

[1] Yep, same version but different URL/SHA1. No conflicts.

The toolchain file is the only way to apply global settings for 3rd party projects in Hunter.

Only CMAKE_TOOLCHAIN_FILE will be forwarded for all packages, neither standard CMAKE_CXX_COMPILER/CMAKE_CXX_FLAGS nor custom variables like ANDROID_FOO=boo will be applied globally. First reason is the simplicity of forwarding logic, second reason is about distinguishing local and global settings. E.g. if a user wants to set `-Wall` only for the local project he can use CMAKE_CXX_FLAGS. If user wants to set `-Wall` globally then he can use CMAKE_TOOLCHAIN_FILE.

Hunter-ID 1eae623

Toolchain-ID			
d46ea0b	Working directory	${HUNTER_ROOT}/_Base/1eae623/d46ea0b/...	
		Compiler	Flags
		gcc	
c018e63	Working directory	${HUNTER_ROOT}/_Base/1eae623/c018e63/...	
		Compiler	Flags
		clang	
c39da39	Working directory	${HUNTER_ROOT}/_Base/1eae623/c39da39/...	
		Compiler	Flags
		clang	-std=c++11
7450099	Working directory	${HUNTER_ROOT}/_Base/1eae623/7450099/...	
		Compiler	Flags
		arm-linux-androideabi-g++	-march=armv7-a
2d935ea	Working directory	${HUNTER_ROOT}/_Base/1eae623/2d935ea/...	
		Compiler	Flags
		clang	-fsanitize=thread

Message in logs:

```
-- [hunter] [ Hunter-ID: 1eae623 | Toolchain-ID: d46ea0b | Config-ID: ... ]
-- [hunter] [ Hunter-ID: 1eae623 | Toolchain-ID: c018e63 | Config-ID: ... ]
-- [hunter] [ Hunter-ID: 1eae623 | Toolchain-ID: c39da39 | Config-ID: ... ]
```

Examples on GitHub

- Android example

CGold

- Platforms

Polly

- Collection of toolchains

Simple toolchains

Building with `-fPIC`:

```
# toolchain.cmake
set(CMAKE_POSITION_INDEPENDENT_CODE TRUE)
```

Building with `-std=c++11`:

```
# toolchain.cmake
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_CXX_STANDARD_REQUIRED YES)
```

CGold

- C++11 toolchain
- Set default toolchain

1.4.3 Config-ID

Third level of customization. Set version of package to build and its build options.

`Config-ID` is the first 7 digits of `SHA1` of the file with `hunter_config` commands (internal unified representation). This level can be customized with HunterGate options: `GLOBAL`, `LOCAL` and `FILEPATH`. Packages from `Hunter-ID 1eae623` can be built using different versions and different CMake arguments:

Hunter-ID 1eae623	Toolchain-ID d46ea0b

Config-ID				
`0fa873a`	Working directory	`${HUNTER_ROOT}/_Base/1eae623/d46ea0b/0fa873a/...`		
	Packages	Name	Version	Options
		`Foo`	`1.0.0`	
		`Boo`	`2.0.0`	`BOO_WITH_SOMETHING=YES`
`e9da39c`	Working directory	`${HUNTER_ROOT}/_Base/1eae623/d46ea0b/e9da39c/...`		
	Packages	Name	Version	Options
		`Foo`	`2.1.0`	`FOO_SUPER_MODE=YES`
		`Boo`	`3.0.0`	`BUILD_SHARED_LIBS=ON`

Message in logs:

```
-- [hunter] [ Hunter-ID: 1eae623 | Toolchain-ID: d46ea0b | Config-ID: 0fa873a ]
-- [hunter] [ Hunter-ID: 1eae623 | Toolchain-ID: d46ea0b | Config-ID: e9da39c ]
```

See also:

- Example
- *HUNTER_BUILD_SHARED_LIBS*
- *HUNTER_CONFIGURATION_TYPES*

1.4.4 Build types

- Build types like Release/Debug

1.5 Shareable

A root directory with installed packages can be shared between several local projects. If one local project triggers installation of a new third party package, then the root directory will be locked against modifications until the install has either finished or interrupted (i.e. Ctrl-C). Other local projects that try to run `hunter_add_package` for the same root at the same time will automatically wait for the root to be unlocked. Note that different root directories have independent locks, e.g. triggering a build of `OpenSSL` for `iOS` will not delay building of `GTest` for `Android`. Internally this is done using the `file(LOCK ...)` CMake command (available since `3.2`).

Similar synchronization is done when initializing Hunter using the `HunterGate` command. This is a very handy feature for CI systems where an environment is shared between several jobs, e.g. Jenkins.

1.6 Binaries from server

Hunter has an internal mechanism that saves the binaries of installed packages along with meta-data about the toolchain, build options, and dependencies. This allows Hunter to avoid triggering the same build when a new root directory is created. For example, when a user changes the version of `OpenSSL` from `1.0.1` to `1.0.2` it will affect `Config-ID`, so a new root will be created. However, it will not affect how `GTest` builds (if it's not a dependency), so the `GTest` archive can be unpacked from the local cache. The cache can be kept local or uploaded to a Hunter cache server.

See also:

- *Uploading to server*

1.6.1 Details

The default server with cached binaries is cpp-pm/hunter-cache. Archives are saved as GitHub release assets and each is associated with a git tag. Available packages can be queried using an `upload.*` HTTP query from the GitHub branches URL:

- cpp-pm/hunter@upload.

Note that some toolchains may not work for specific packages. Check the status in the Travis CI job details. For example, Qt is broken for the iOS armv7s architecture, so we have to use the `ios-*-wo-armv7s` toolchains:

- https://travis-ci.org/cpp-pm/hunter/builds/140158080

Binaries/headers are stored in archives and archives are the result of packing the `CMAKE_INSTALL_PREFIX` directory produced by the `cmake --build _builds --target install` command. The idea is similar to CPack functionality but is extended for non-CMake packages too.

```
> mkdir temp-dir
> cd temp-dir
[temp-dir]> wget https://github.com/cpp-pm/hunter-cache/releases/download/cache/
↪aaee852f00aa3a2a884281e8920315a77fb14465.tar.bz2
[temp-dir]> tar xf aaee852f00aa3a2a884281e8920315a77fb14465.tar.bz2
[temp-dir]> ls include/gtest/gtest.h
include/gtest/gtest.h
```

```
[temp-dir]> ls lib/libgtest.a
lib/libgtest.a
```

1.7 Manage anything

You can manage anything that can be downloaded by `URL` and checked with an `SHA1` hash:

- C/C++ packages

```
hunter_add_package(Boost)
find_package(Boost CONFIG REQUIRED)

hunter_add_package(OpenSSL)
find_package(OpenSSL REQUIRED)
```

- CMake modules

```
hunter_add_package(sugar)
find_package(sugar CONFIG REQUIRED)

sugar_files(...)
```

- Additional sources (OpenCV example):

```
set(OPENCV_EXTRA_MODULES_PATH "" CACHE PATH "Where to look for additional OpenCV␣
↪modules")
if(OPENCV_WITH_EXTRA_MODULES)
  hunter_add_package(OpenCV-Extra)
  set(OPENCV_EXTRA_MODULES_PATH "${OPENCV-EXTRA_ROOT}/modules")
endif()
```

- Resources (pictures, data for testing, . . .)

```
hunter_add_package(MyData)

add_test(NAME FooTest1 COMMAND foo --use-data "${MYDATA_ROOT}/case-1.png")
add_test(NAME FooTest2 COMMAND foo --use-data "${MYDATA_ROOT}/case-2.png")
# ...
```

Note:

- See *alternative approach*

1.8 Backward compatibility

Turn Hunter off by adding one option HUNTER_ENABLED=NO to use your old settings. For example:

```
  add_executable(foo openssl-example.cpp)

  hunter_add_package(OpenSSL)
  find_package(OpenSSL REQUIRED)
  target_link_libraries(foo PUBLIC OpenSSL::SSL OpenSSL::Crypto)
```

by default this code will trigger download and build of OpenSSL:

```
> rm -rf _builds
> cmake -H. -B_builds -DCMAKE_VERBOSE_MAKEFILE=YES
> cmake --build _builds

/usr/bin/c++
    CMakeFiles/foo.dir/openssl-example.cpp.o
    -o foo
    -rdynamic
    /.../_Base/a9bd96a/e8394c3/dd69ac4/Install/lib/libssl.a
    /.../_Base/a9bd96a/e8394c3/dd69ac4/Install/lib/libcrypto.a
    -ldl
```

but adding `HUNTER_ENABLED=NO` make it **skip all Hunter instructions** and system library will be used instead:

```
> rm -rf _builds
> cmake -H. -B_builds -DCMAKE_VERBOSE_MAKEFILE=YES -DHUNTER_ENABLED=NO
> cmake --build _builds

/usr/bin/c++
    CMakeFiles/foo.dir/openssl-example.cpp.o
    -o foo
    -rdynamic
    /usr/lib/x86_64-linux-gnu/libssl.so
    /usr/lib/x86_64-linux-gnu/libcrypto.so
```

Note: As you can see `hunter_add_package` has no effect when `HUNTER_ENABLED` is `OFF` hence such code is redundant:

```
if(HUNTER_ENABLED)
  hunter_add_package(foo)
endif()
```

It will behave in the same way as just `hunter_add_package(foo)` alone.

`HUNTER_ENABLED=NO` can be set by default using CMake `option`:

```
# before HunterGate
option(HUNTER_ENABLED "Enable Hunter package manager" NO)
HunterGate(URL ... SHA1 ...)
```

So this makes it easy to use Hunter as experimental feature. All information that users need to know about new commands is that `hunter_add_package` and `HunterGate` will do nothing as long as `HUNTER_ENABLED` is `NO`.

Note that Hunter itself add `HUNTER_ENABLED=YES` while building third party package. It means that if package released with `HUNTER_ENABLED=NO` by default it still can be used in Hunter, no extra modifications needed.

1.8.1 Helper modules

Not all packages have the same CMake usage API. E.g. for CURL in Hunter there is imported target `CURL::libcurl` but there are only `CURL_INCLUDE_DIRS` and `CURL_LIBRARIES` defined in standard `FindCURL` module.

To mimic Hunter API disabled-mode modules can be used.

`HunterGate` will load them automatically when `HUNTER_ENABLED=OFF` and they are located in `${CMAKE_CURRENT_LIST_DIR}/cmake/Hunter/disabled-mode`:

```
> cmake -H. -B_builds -DHUNTER_ENABLED=NO -DHUNTER_STATUS_DEBUG=ON
-- [hunter *** DEBUG *** ...] Adding "disabled-mode" modules: /.../cmake/Hunter/
↪disabled-mode
```

Module `CURLConfig` from "disabled-mode" modules will be added to CMake search path, loaded, call standard `FindCURL` and create imported target `CURL::libcurl`. This will allow to use same API with and without Hunter:

```
hunter_add_package(CURL)
find_package(CURL CONFIG REQUIRED)
target_link_libraries(foo PUBLIC CURL::libcurl)
```

Examples on GitHub

- Disabled mode example

1.9 CMake only

No other dependencies - **just CMake** and your environment/IDE (no need for Git or Python or anything).

1.10 Works everywhere!

Hunter works everywhere: CMake-GUI, Qt Creator, Visual Studio, Xcode, Cygwin, MinGW, Jenkins, Travis etc.

See also:

- *F.A.Q.: How to use Hunter in Android Studio?*

CHAPTER 2

Quick start

Short description of main commands. First part is about choosing CMake version to use, then example of commands for downloading Boost components. At the end take a look at GitHub repository with tiny project that use GTest and try it yourself.

Examples on GitHub

- HunterGate
- Weather

2.1 Notes about version of CMake

- 3.2.0 **Minimum required**
  - New `continue` command
  - New synchronization command `file(LOCK ...)`
- 3.4.1
  - **Buggy**, see issue #405
- 3.5.0 **Minimum for iOS projects**
  - New variable CMAKE_IOS_INSTALL_COMBINED
  - iOS toolchains
- 3.7.0
  - Minimum version for packages with *protected sources*
  - `USERPWD` sub-option for `file(DOWNLOAD|UPLOAD ...)`
  - `HTTP_{USERNAME|PASSWORD}` sub-options for `ExternalProject_Add`

  - List of URLs can be passed to `ExternalProject_Add`. Used by *HUNTER_DOWNLOAD_SERVER*.
- 3.7.1 **Minimum for Android projects**
  - CMake now supports Cross Compiling for Android with simple toolchain files
  - Polly Android toolchains
- 3.9.2 **Minimum for Android NDK r16+**

Tip:

- CMake milestones (old version)

Note: If you're building CMake from sources please make sure that *HTTPS support is enabled in CURL*.

Note: In theory CMake 3.0 can be used with Hunter versions before v0.22 but in practice you have to work with v0.14.3 because `continue` added to v0.14.4 code.

Note: Latest Hunter release with support of old Android toolchains (before CMake 3.7.1) is v0.16.36

2.2 First step

Set HUNTER_ROOT environment variable to **an empty directory**. This directory will be used by `HunterGate` module for storing packages and utility files. Using environment variable is recommended but not mandatory, see other options.

Set minimum CMake version:

```
cmake_minimum_required(VERSION 3.2)
```

Copy HunterGate module to your project and include it:

```
> mkdir cmake
> wget https://raw.githubusercontent.com/cpp-pm/gate/master/cmake/HunterGate.cmake -O␣
↪cmake/HunterGate.cmake
```

```
include("cmake/HunterGate.cmake")
```

This module will download archive automatically from `URL` that you provide to the `HUNTER_ROOT` directory (it means that there is **no need to clone this repository** in general, see notes):

```
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)
```

Now project can be started:

```
project(Foo)
```

Let's download and install `boost.{regex,system,filesystem}`:

```
hunter_add_package(Boost COMPONENTS regex system filesystem)
```

Hunter part is done, now well known CMake-style kung fu (see *Boost*):

```
find_package(Boost CONFIG REQUIRED regex system filesystem)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC Boost::regex Boost::system Boost::filesystem)
```

Summarize:

```
cmake_minimum_required(VERSION 3.2)

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)

project(Foo)

hunter_add_package(Boost COMPONENTS regex system filesystem)
find_package(Boost CONFIG REQUIRED regex system filesystem)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC Boost::regex Boost::system Boost::filesystem)
```

Build it:

```
> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON -DCMAKE_BUILD_TYPE=Release
> cmake --build _builds --config Release
```

2.3 Simple example

Examples on GitHub

- Tiny project with GTest

2.4 Uninstall

All directories inside `${HUNTER_ROOT}/_Base` are reconstructible. You can remove all temps (downloads, unpacked directories, installed directories etc.) by command:

```
rm -rf "${HUNTER_ROOT}/_Base"
```

Or remove particular `Hunter-ID` by command:

```
rm -rf "${HUNTER_ROOT}/_Base/62422b8" # remove installed libraries
rm -rf "${HUNTER_ROOT}/_Base/Download/Hunter/0.8.3/62422b8" # remove Hunter itself
```

CHAPTER 3

User guides

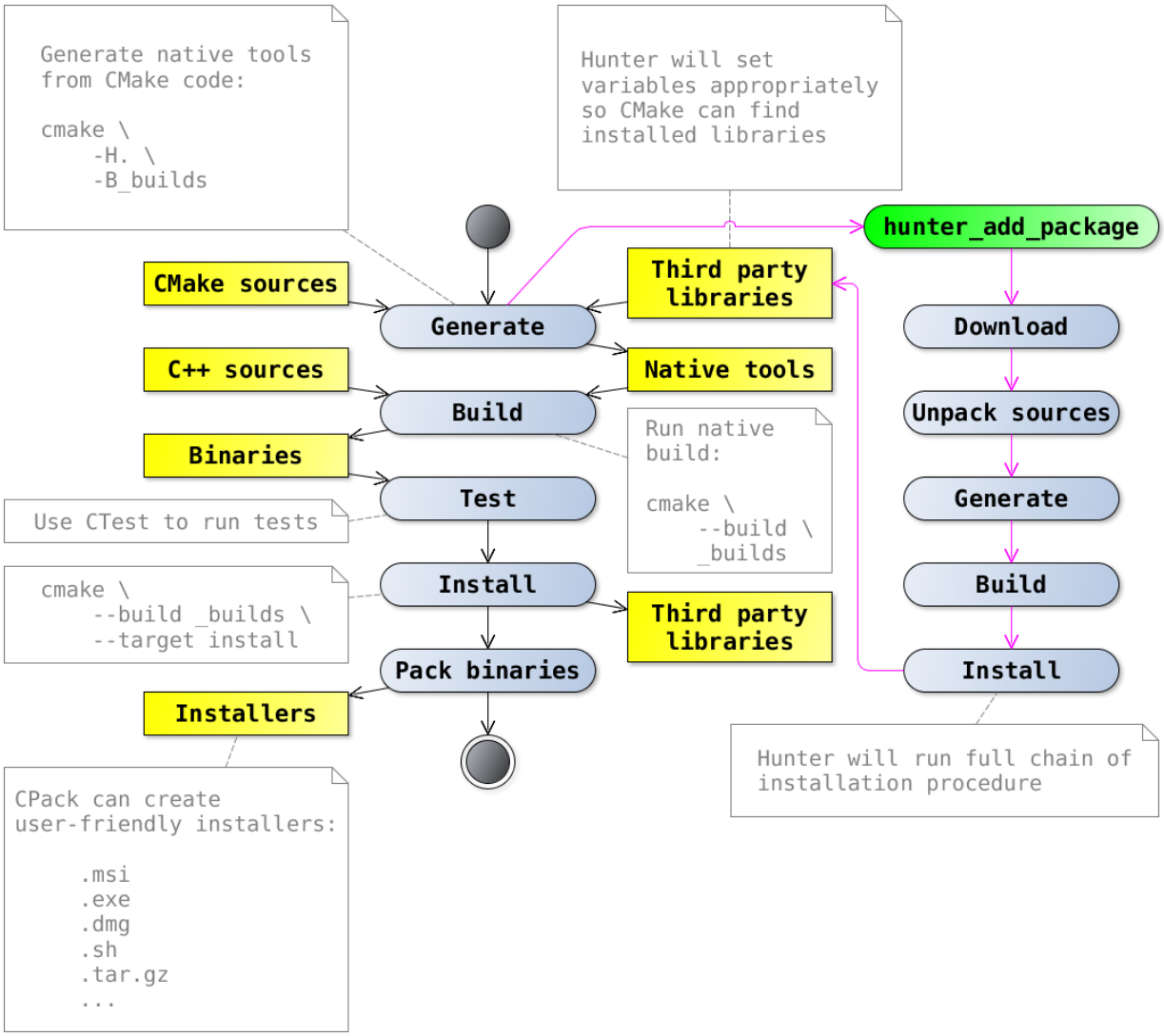
Guides split into sections by role of developer in project. Sections should be read sequentially one by one.

3.1 Regular user

How does Hunter affect end users which do usually run `*.msi` or `*.dmg` installers? The answer - it does not at all. Hunter in fact do quite the same stuff that developer do with packages: download, build, install, reuse in other projects. There will be no new functionality introduced by package - it will be installed in the same way as system package manager or custom build script do. But giving you much more control and allow you experimenting seamlessly. From some point of view Hunter is like adding unit-testing to your project. It's the tool that will not extend final behavior of application directly. However just like with unit-testing users will probably notice result effect such as stability/quality in overall.

3.1.1 Hunter in CMake environment

Here is an activity diagram showing the location of Hunter in regular CMake tools environment:



3.2 CMake user

This kind of developer can **read** CMake code that was written by more experienced CMake developers. They understand some simple features such as adding an executable with the `add_executable` command, and that this command contains a list of source files associated with the executable. They will probably have difficulty understanding why, in some cases, the `include_directories` command is called, but in others, `target_include_directories` is called instead. The main target of modifications is C++ code.

Such developers can:

- Add more targets to projects
- Add more sources to targets
- Add C++ flags that don't break compatibility (e.g. warnings/optimization)

Such developers can't:

- Add more external dependencies to project

- Adding flags that can break compatibility (e.g. `-std=c++11` or `/MT`)

See also:

3.2.1 Protected sources

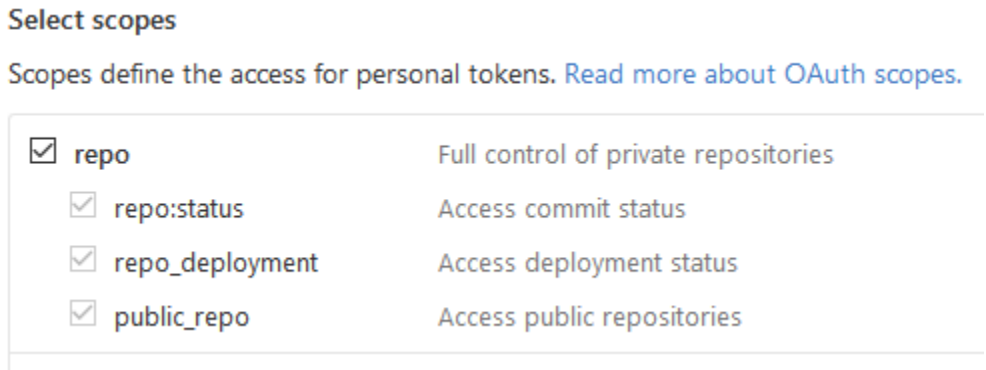
Hunter can manage access to the package with sources protected by HTTP user-name/password credentials. Such packages should be *marked as protected* in corresponding `hunter.cmake` file. Passwords should be set in *Hunter passwords file* using *hunter_http_password* function.

Hint for GitHub users

You don't have to store your personal password in `passwords.cmake` file. Instead you can generate personal access token and use it as `PASSWORD`:

```
hunter_http_password(Foo USERNAME "myname" PASSWORD "123...abc")
```

Note: Since token used to access private repositories you have to set `repo` scope ("Full control of private repositories"):



3.2.2 Private data download

Using *hunter_private_data* module user can download files that are private for the current project, i.e. some data that **will not be shared** with other projects. Unlike regular packages such data is not injectable, i.e. user will not be able to add his own code just by changing version of private data (well there even no such essence as private data version). This feature is quite orthogonal to main Hunter functionality and just use Hunter root directory and tools like stamps and locks.

As an example you can download file for testing:

```
hunter_private_data(
    URL "https://example.com/myfile.txt"
    SHA1 "abcxxxxxx123"
    FILE "myfile.txt"
    LOCATION myfile_path
)

add_test(NAME foo COMMAND foo --text-file ${myfile_path})
```

File `myfile.txt` will be downloaded once to outside `HUNTER_ROOT` directory. When you create two build directories:

```
> rm -rf _builds
> cmake -H. -B_builds/Debug -DCMAKE_BUILD_TYPE=Debug
> cmake -H. -B_builds/Release -DCMAKE_BUILD_TYPE=Release
```

They both will share same `myfile.txt` file. If for example you switch to different Git branch with different version of `myfile.txt` file Hunter will download this version and create separate directory basing on new hash. Same variable `myfile_path` will point to new location.

You can use *hunter_private_data_password* module to specify credentials for downloading password protected data.

See also:

- *F.A.Q.: How to download private GitHub asset*

3.2.3 Using license files

After package was installed Hunter will search for the license file(s) in sources. Next priority is used (see script):

- Licenses specified explicitly by HUNTER_INSTALL_LICENSE_FILES
- Default names (only first found used):
  - `LICENSE`
  - `LICENSE.txt`
  - `COPYING`
  - `COPYING.txt`
  - `license`
  - `license.txt`
  - `copying`
  - `copying.txt`
- Files found by command `file(GLOB ... "${package_source_dir}/LICENSE*")`

In case search was successful variable `<PACKAGE>_LICENSES` can be used to obtain full paths to the licenses (example):

```
hunter_add_package(Boost)

set(project_license "${CMAKE_CURRENT_BINARY_DIR}/LICENSE.txt")
file(
    WRITE
    "${project_license}"
    "Some info about this project license.\n\n"
)

string(COMPARE EQUAL "${Boost_LICENSES}" "" is_empty)
if(is_empty)
  message(FATAL_ERROR "No licenses")
endif()

file(APPEND "${project_license}" "== 3rd party licenses ==\n\n")
```

```
foreach(x ${Boost_LICENSES})
  file(READ "${x}" content)
  get_filename_component(license_name "${x}" NAME)
  file(APPEND "${project_license}" "== Boost (${license_name}) ==\n\n")
  file(APPEND "${project_license}" "${content}")
endforeach()
```

Warning: For Hunter version < `v0.18.0`:

- `HUNTER_INSTALL_LICENSE_FILES` not used
- The `*_LICENSE` variable should be used instead of `*_LICENSES`
- `*_LICENSE` contains only of one file (it's not a list)

3.3 Hunter user

3.3.1 Use version from Git submodule

Hunter allows the creation of an archive with sources on the fly by getting it from a Git submodule.

Example:

```
> git clone https://github.com/hunter-test-cases/git-submodule-integration
> cd git-submodule-integration
[git-submodule-integration]> git submodule update --init .
```

To instruct Hunter to use the contents of the submodule, add a local config file and set the `GIT_SUBMODULE` flag:

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
    LOCAL # <----- load cmake/Hunter/config.cmake
)
```

```
# cmake/Hunter/config.cmake
hunter_config(fruits GIT_SUBMODULE "3rdParty/fruits")
```

The path set by the `GIT_SUBMODULE` flag is the same as in the `.gitmodules` file:

```
[git-submodule-integration]> cat .gitmodules
[submodule "3rdParty/fruits"]
  path = 3rdParty/fruits
  url = https://github.com/cgold-examples/fruits
```

At the configure step Hunter will run the command `git archive` to pack sources:

```
[git-submodule-integration]> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
...
```

```
-- [hunter *** DEBUG *** ...] Creating archive '/.../git-submodule-integration/_
↪builds/_3rdParty/Hunter/git-archives/fruits.tar'
...
```

Let's build the project and run tests:

```
[git-submodule-integration]> cmake --build _builds
[git-submodule-integration]> (cd _builds && ctest -VV)
...
1: Quick meal:
1:   plum x 2
1:   pear x 1
...
```

If you want to make changes to the dependent project (the one added as submodule) and test them, you have to **commit** patches first:

```
[git-submodule-integration]> cd 3rdParty/fruits
[fruits]> grep return lib/fruits/rosaceae/Plum.cpp
return "plum";
[fruits]> vim lib/fruits/rosaceae/Plum.cpp
[fruits]> grep return lib/fruits/rosaceae/Plum.cpp
return "plum-v2";
[fruits]> git add lib/fruits/rosaceae/Plum.cpp
[fruits]> git commit -m 'Update'
```

Go back to the parent directory and run build. There is no need to run configure again, corresponding Git files are watched by CMake hence the configure step will start automatically when the build step is invoked:

```
[fruits]> cd ../..
[git-submodule-integration]> cmake --build _builds
```

Run tests to see changes:

```
[git-submodule-integration]> (cd _builds && ctest -VV)
1: Quick meal:
1:   plum-v2 x 2
1:   pear x 1
```

Possible problems with GIT_SUBMODULE

When using a package via the `GIT_SUBMODULE` option, the Hunter defined CMake variable `HUNTER_<package>_VERSION` is set to the commit hash of the Git sub-module. If the `hunter.cmake` file of the package contains logic that depends on the value of the `HUNTER_<package>_VERSION` variable, using the `GIT_SUBMODULE` option may break the package build. If that is the case you can add explicit `VERSION` value to *hunter_config*.

Use subdirectory of submodule

To instruct hunter to archive a subdirectory of the Git submodule add the keyword `HUNTER_SUBMODULE_SOURCE_SUBDIR` to the CMake arguments:

```
# cmake/Hunter/config.cmake
hunter_config(fruits GIT_SUBMODULE "3rdParty/fruits"
  CMAKE_ARGS "HUNTER_SUBMODULE_SOURCE_SUBDIR=app")
```

The created archive will contain just the subfolder `app` of the submodule.

GIT_SUBMODULE vs add_subdirectory

Note that we can achieve the same by adding sources with `add_subdirectory`:

```
# top level CMakeLists.txt
# ...

add_subdirectory(3rdParty/fruits)
```

The only pros of `add_subdirectory` approach is that build artifacts of the `fruits` will live in our `_builds` directory. `GIT_SUBMODULE` will add new package in the same way as regular release-based packages added, meaning that after installation all build artifacts will be removed. Every new version start build from scratch.

Next cons of using `add_subdirectory`:

- Dependent project `fruits` is not installed, hence CMake API usage may be different. If package has target `fruits_rosaceae` internally then after installation it can be `fruits::fruits_rosaceae`
- For the same reason C++ API may be different, e.g. `#include` directives
- It's not two separate projects now - it's one big project. Hence they will share same cache which may lead to options conflicts, targets name conflicts, targets from both projects will be installed, tests from both projects will be run
- Correctness. Note that `add_subdirectory` can be used only for dependencies which is not used by other packages in Hunter. If current project use package `zoo` which depends on `fruits` we can't do `add_subdirectory(fruits)` since `hunter_add_package(zoo)` will build and use `fruits` from Hunter. See next chapter for details

Injection

`GIT_SUBMODULE` allow you to correctly inject new version of package into existent hierarchy of packages.

For example let's take a look at the project which use TIFF, TIFF depends on ZLIB:

```
> git clone https://github.com/hunter-test-cases/git-submodule-integration-deps
> cd git-submodule-integration-deps
[git-submodule-integration-deps]> git submodule update --init .
```

First let's remove `LOCAL` config and build standard TIFF with standard ZLIB:

```
# CMakeLists.txt
cmake_minimum_required(VERSION 3.2)

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)

project(foo)

hunter_add_package(TIFF)
find_package(TIFF CONFIG REQUIRED)
```

Config-ID is `f743b0b`:

```
[git-submodule-integration-deps]> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
...
-- Downloading...
   dst='~/.hunter/_Base/Download/ZLIB/1.2.8-p3/573dc28/v1.2.8-p3.tar.gz'
   timeout='none'
-- Using src='https://github.com/hunter-packages/zlib/archive/v1.2.8-p3.tar.gz'
...
/usr/bin/cc ... -isystem ~/.hunter/_Base/3b39eff/e1266bb/f743b0b/Install/include ... /
↪.../tif_zip.c
```

Now let's add `LOCAL` back and run build again:

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
    LOCAL
)

project(foo)

hunter_add_package(TIFF)
find_package(TIFF CONFIG REQUIRED)
```

```
# cmake/Hunter/config.cmake
hunter_config(ZLIB GIT_SUBMODULE "3rdparty/zlib")
```

```
[git-submodule-integration-deps]> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
```

Now we are getting sources from locally created `ZLIB.tar` archive:

```
...
-- verifying file...
     file='/.../_builds/_3rdParty/Hunter/git-archives/ZLIB.tar'
...
```

And **rebuilding** TIFF with newly installed ZLIB, Config-ID changed from `f743b0b` to `817c9cb`:

```
/usr/bin/cc ... -isystem ~/.hunter/_Base/3b39eff/e1266bb/817c9cb/Install/include ... /
↪.../tif_zip.c
```

To achieve the same with `add_subdirectory` you have to clone TIFF package too. Then you have to be sure that TIFF supports external ZLIB targets configuration, call `add_subdirectory(3rdparty/zlib)` first, then `add_subdirectory(3rdparty/TIFF)`. Note that if you **don't know** that TIFF depends on ZLIB and you just call `add_subdirectory(3rdparty/zlib)` you will end up with incorrect configuration!

3.3.2 Injecting current Git repository

It is possible to pack current Git repository and use created archive as a package. Such scenario is common for the projects with usage example code.

For instance we have project `fruits`:

```
> git clone https://github.com/cgold-examples/fruits
> cd fruits
[fruits]>
```

There is top level `CMakeLists.txt`:

```
[fruits]> grep '^project' CMakeLists.txt
project(fruits VERSION 1.0.0)
```

And subdirectory `example` that can be used as a stand-alone project:

```
[fruits]> grep 'add_subdirectory(example)' CMakeLists.txt
  add_subdirectory(example)

[fruits]> grep '^project' example/CMakeLists.txt
project(fruits-example)
```

If you start building from top you can build `fruits` and `fruits-example` as a one big monolithic project:

```
[fruits]> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
[fruits]> cmake --build _builds
Scanning dependencies of target fruits_rosaceae
...
Scanning dependencies of target fruits_quick_meal
[ 95%] Building CXX object example/fruits/quick_meal/CMakeFiles/fruits_quick_meal.dir/
↪main.cpp.o
[100%] Linking CXX executable fruits_quick_meal
[100%] Built target fruits_quick_meal
```

However you can build `fruits-example` as a stand-alone project. In this case `fruits` will be packed on the fly and installed as a Hunter package:

```
[fruits]> rm -rf _builds
[fruits]> cd example
[fruits/example]> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
...
-- [hunter *** DEBUG *** ...] Creating archive '/.../fruits/example/_builds/_3rdParty/
↪Hunter/git-archives/fruits.tar'
...
-- [hunter] Building fruits
...
Install the project...
/.../bin/cmake -P cmake_install.cmake
-- Install configuration: "Release"
-- Installing: /.../.hunter/_Base/19e4a2f/489ecc6/e734c3e/Build/fruits/Install/
↪include/fruits/fruits.hpp
-- Installing: /.../.hunter/_Base/19e4a2f/489ecc6/e734c3e/Build/fruits/Install/
↪include/fruits/rosaceae/Plum.hpp
-- Installing: /.../.hunter/_Base/19e4a2f/489ecc6/e734c3e/Build/fruits/Install/lib/
↪cmake/fruits/fruitsConfig.cmake
-- Installing: /.../.hunter/_Base/19e4a2f/489ecc6/e734c3e/Build/fruits/Install/lib/
↪libfruits_rosaceae.a
...
```

There is no `fruits` libraries while building project:

```
[fruits/example]> cmake --build _builds
Scanning dependencies of target fruits_vegan_party
```

```
[ 25%] Building CXX object fruits/vegan_party/CMakeFiles/fruits_vegan_party.dir/main.
↪cpp.o
[ 50%] Linking CXX executable fruits_vegan_party
[ 50%] Built target fruits_vegan_party
Scanning dependencies of target fruits_quick_meal
[ 75%] Building CXX object fruits/quick_meal/CMakeFiles/fruits_quick_meal.dir/main.
↪cpp.o
[100%] Linking CXX executable fruits_quick_meal
[100%] Built target fruits_quick_meal
```

Local `config.cmake` file:

```
[fruits/example]> cat cmake/Hunter/config.cmake
hunter_config(fruits GIT_SELF)
```

Hint: It can be useful for testing `find_package(fruits ...)` functionality and that generated `fruitsConfig.cmake` file is correct.

Note: Under the hood `git archive` command is used to pack the project, hence if you want to test modifications you have to commit them. This is similar to `GIT_SUBMODULE` feature. But unlike `GIT_SUBMODULE` feature not all the dirty files will be checked. While using `GIT_SELF` the dirty files inside `fruits/example` directory will be ignored (check log messages). Also if you want to ignore **any** untracked file, you can use the *HUNTER_GIT_SELF_IGNORE_UNTRACKED* option.

3.3.3 Uploading binaries

It's possible to upload local `Cache` directory with binaries to server for future reuse.

Variables and modules related to uploading:

- List of servers that will be used for **downloading binaries** can be set in *HUNTER_CACHE_SERVERS* variable
- If you want to check that there is no third party **builds triggered** by CMake and all packages downloaded from server you can use *HUNTER_DISABLE_BUILDS* variable
- Variable *HUNTER_USE_CACHE_SERVERS* can be used to specify **downloading policy**
- **Uploading parameters** can be set using *hunter_upload_password* module in *Hunter passwords file*
- Use *HUNTER_RUN_UPLOAD=YES* option to **start upload** procedure

Warning: **All entries** from `Cache` directory will be uploaded, not only cache for the current build. Take this information into account while doing upload!

Using GitHub repository as binary cache server

It is possible to upload Hunter binary cache to the server. Next shown an example of using GitHub as a hosting. All big raw `*.tar.bz2` archives uploaded as assets to release with names `cache-*` (directory layout does not matter) and all small text files with meta information uploaded directly to branch `master` (directory layout matters) (see hunter-cache as example).

Note: If you have shared folder in your network there is no need to use any scripts, you can just set `HUNTER_ROOT` variable to location of this directory.

Note: Currently upload procedure is implemented using Python script with `requests` and `gitpython` modules, check that you have Python installed in your system. This limitation will be removed in future. Downloading from server done by `file(DOWNLOAD ...)` CMake commands, so client is still CMake-only based. Module `gitpython` expects Git executable installed in system. You can use environment variable *HUNTER_GIT_EXECUTABLE* to specify custom path.

Example

Next example will show how to setup GitHub binary cache server:

- https://github.com/forexample/hunter-cache

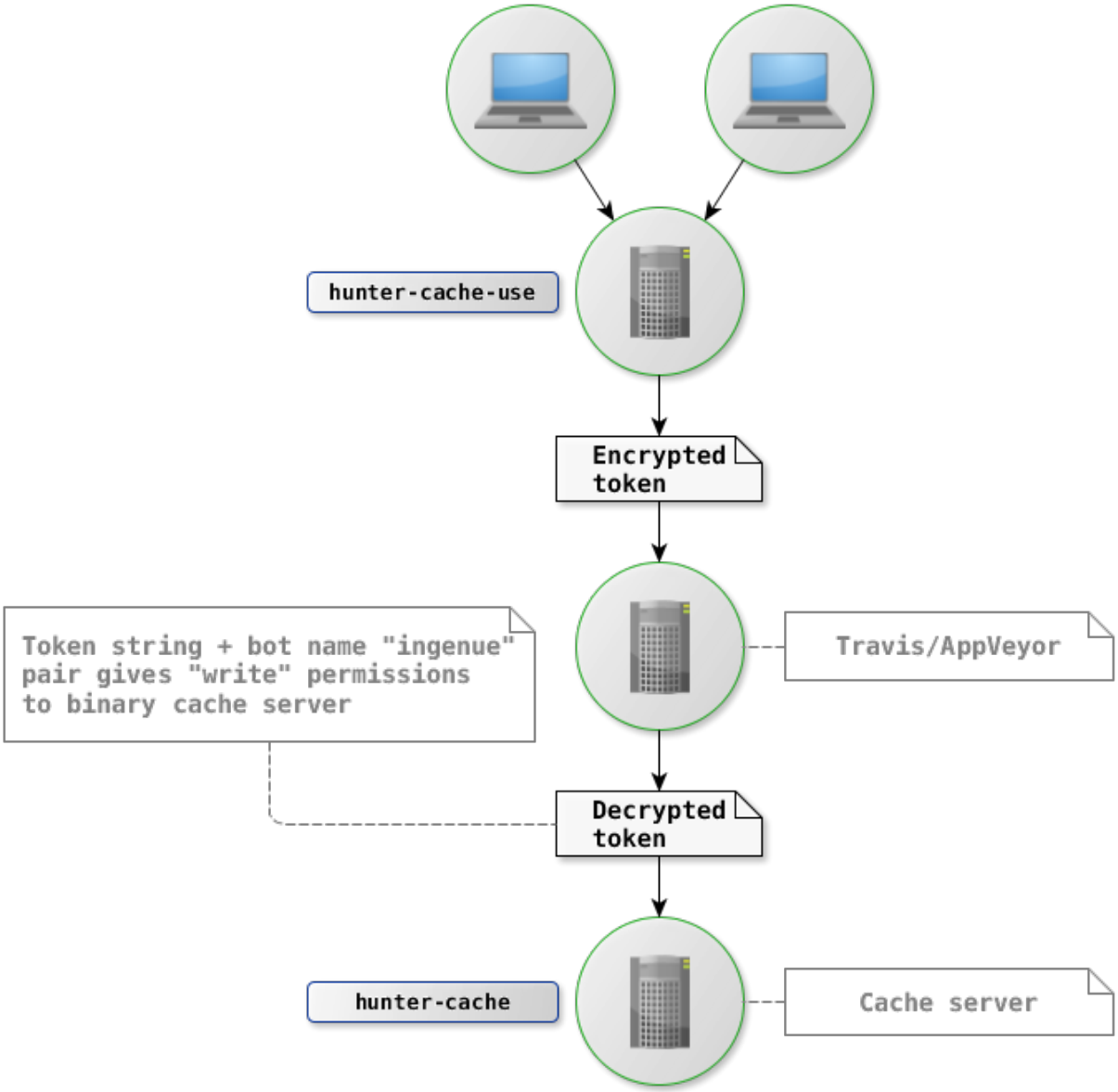
Which will be managed by bot account:

- https://github.com/ingenue

Cache will be uploaded for CI jobs in repository:

- https://github.com/forexample/hunter-cache-use

Diagram:



Workflow:

- Users push code to `hunter-cache-use` repository
- `hunter-cache-use` CI configs hold encrypted token
- When encrypted token reach CI, CI knows how to decrypt it
- Using decrypted token CI can act on bot behalf and upload binaries
- Binaries can be reused by anybody who have added `hunter-cache` to the `HUNTER_CACHE_SERVERS`

Setup

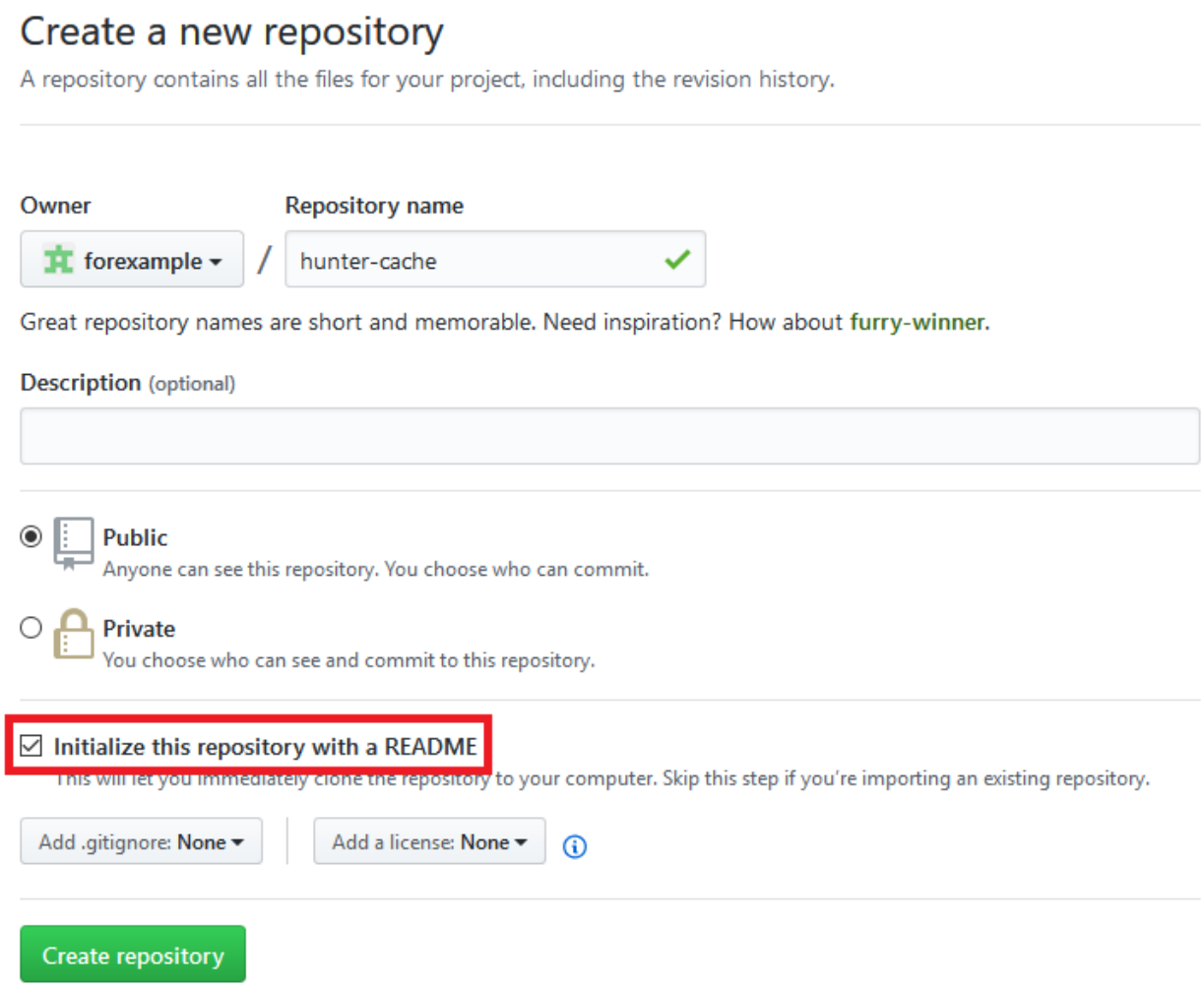
Direction of setup procedure is inversed:

- Create cache server

- Create bot account
- Create token
- Give bot write access to cache server
- Encrypt token
- Save token in CI configs

Create cache server

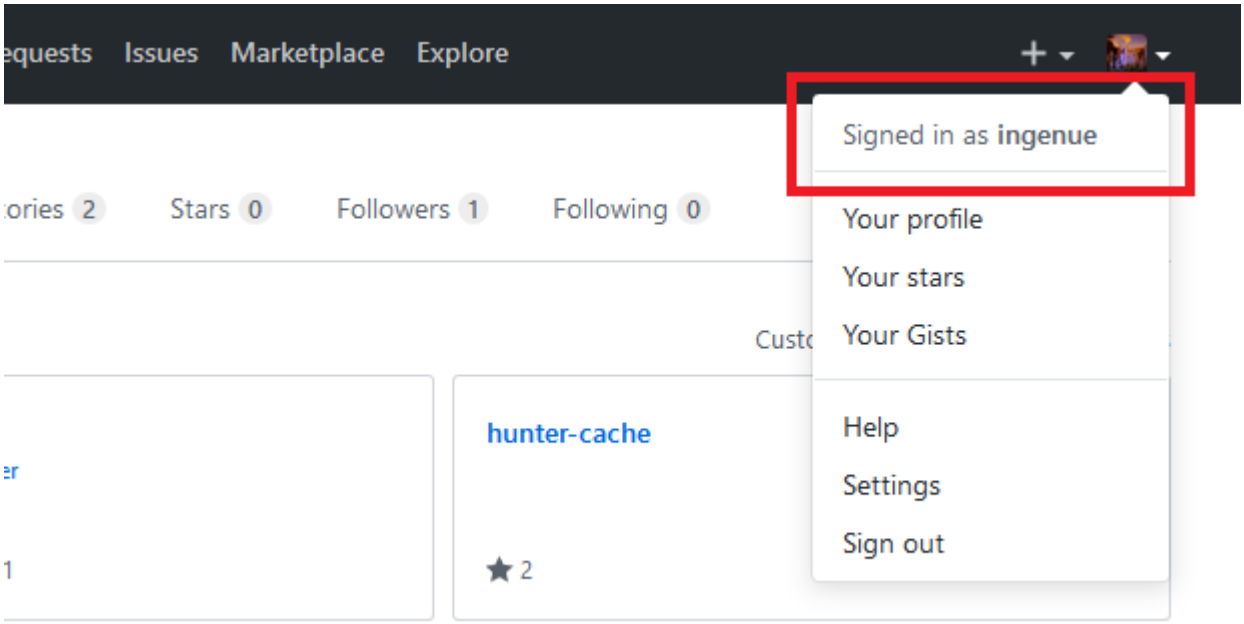
Create repository with at least one file:



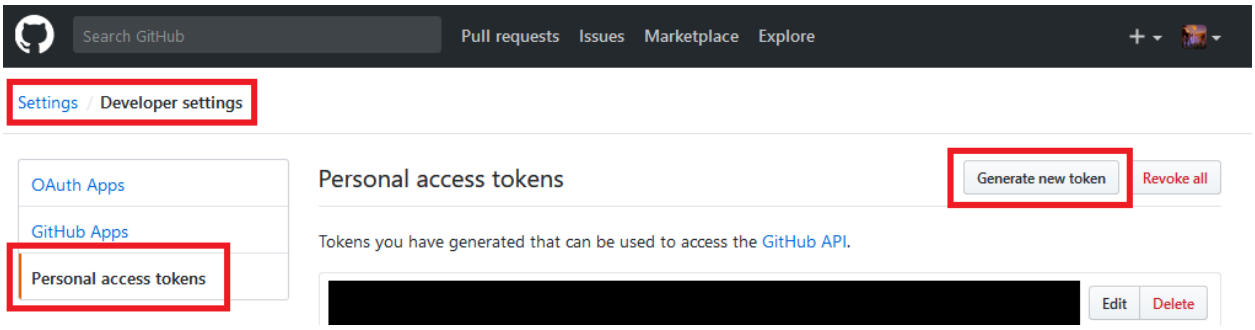
Note that if repository will be empty it will not be possible to create tags for assets.

Create bot token

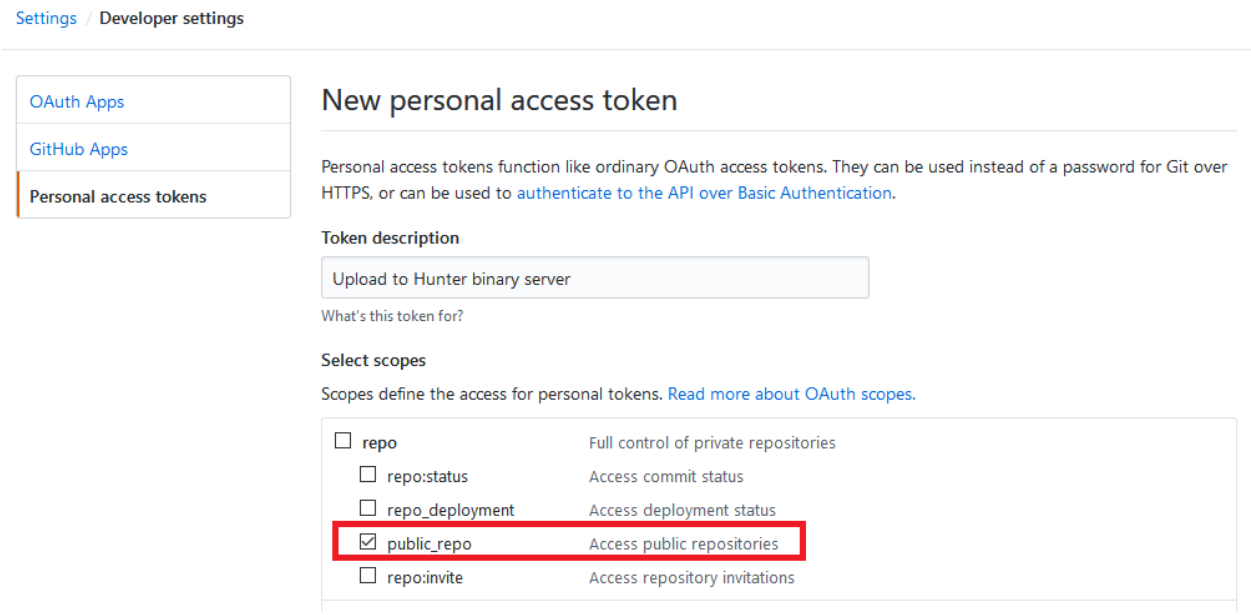
Login to GitHub with the **bot** account, in our case it's `ingenue`:



Settings → Developer settings → Personal access tokens → Generate new token:



Set `public_repo` check-box and create token:



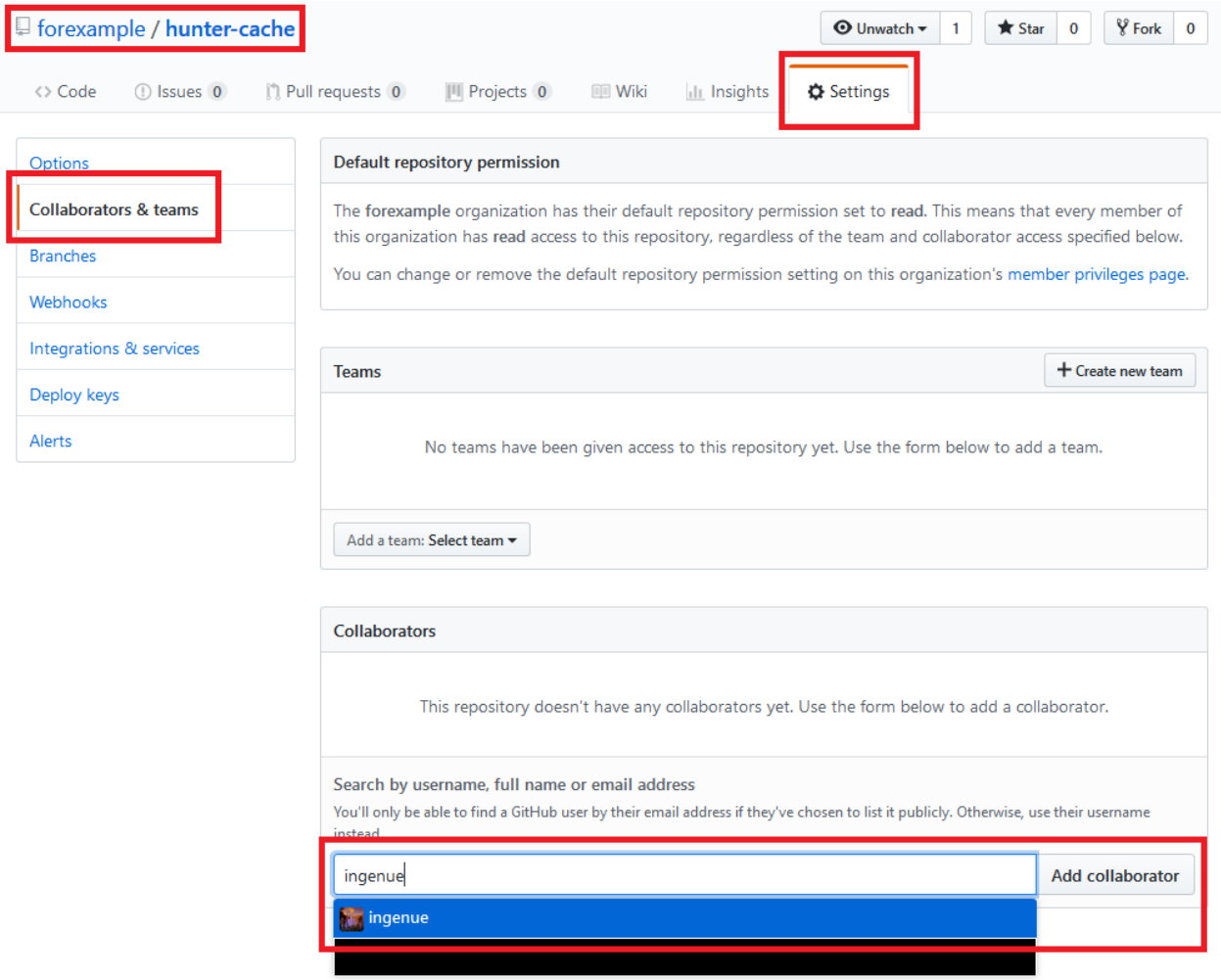
Note: Keep you token private! It's the same as your password!

See also:

- GitHub: creating token

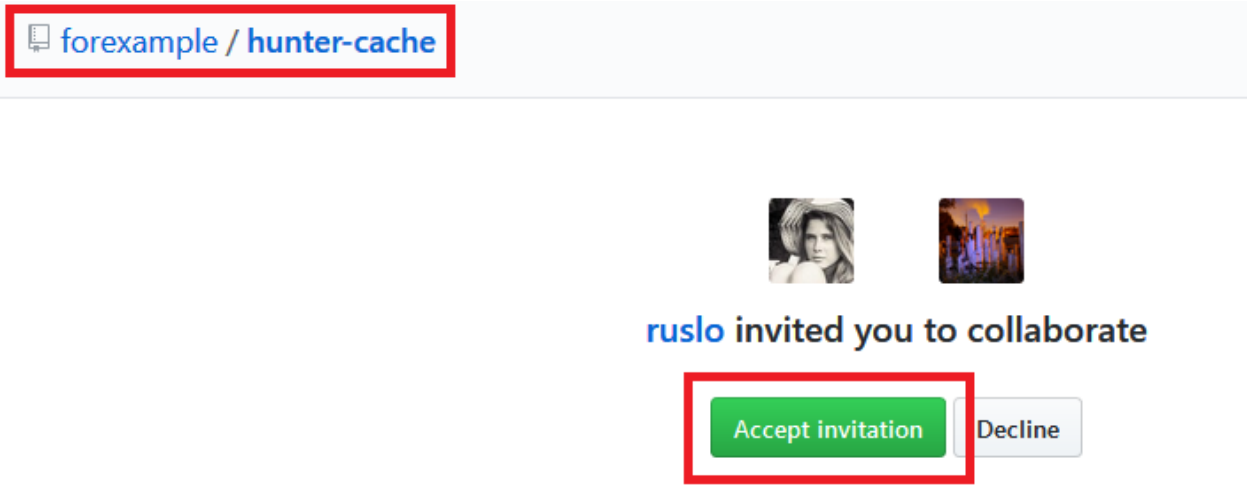
Access

Add `ingenue` bot as a collaborator to `hunter-cache`:



Note: Bot doesn't interact with `hunter-cache-use` so there is no need to set any permissions there.

You should receive email about invitation. Login as **bot** and accept it:



Travis CI

Now we will save token as a secured environment variable `GITHUB_USER_PASSWORD` in Travis and AppVeyor.

Note: Visit https://travis-ci.org and register `hunter-cache-use` repository there.

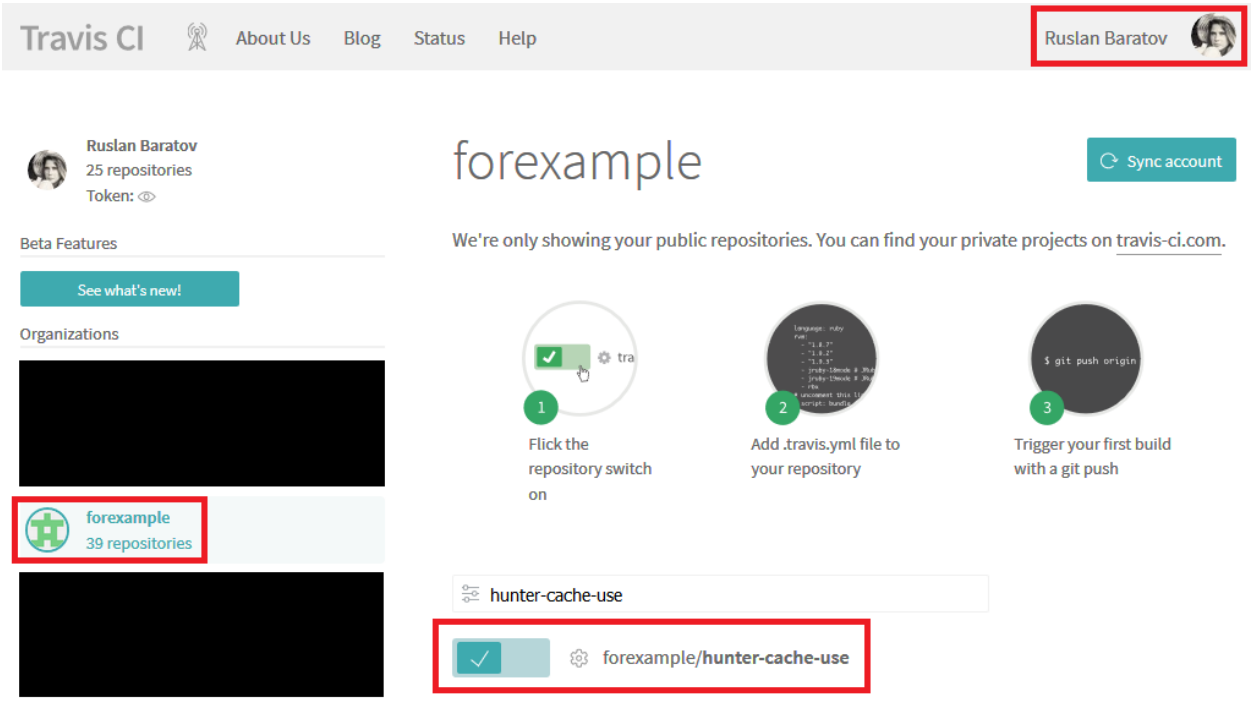
Excerpts from documentation (1 and 2) for OS X (see also this repo):

```
> gem install travis # for Ubuntu it will be 'sudo gem install travis'
```

If you have problems with installing `travis` try to install ruby from brew:

```
> brew install ruby
```

Login with account with which you have registered repository for CI. In my case it's my personal account `ruslo`:



Login with `ruslo` (add `--pro` if repository is private):

```
> travis login
We need your GitHub login to identify you.
This information will not be sent to Travis CI, only to api.github.com.
The password will not be displayed.

Try running with --github-token or --auto if you don't want to enter your password
↪anyway.

Username: ruslo
Password for ruslo: xxxxxx
Two-factor authentication code for ruslo: xxxxxx
Successfully logged in as ruslo!
```

```
> travis whoami
You are ruslo (Ruslan Baratov)
```

Encrypt token:

```
> travis encrypt -r forexample/hunter-cache-use GITHUB_USER_PASSWORD=62xxxxxx2e
Please add the following to your .travis.yml file:

  secure: "EWdxxxxxxfkk="

Pro Tip: You can add it automatically by running with --add.
```

And add it to `.travis.yml`:

```
env:
  global:
    - secure: "EWdxxxxxxfkk="
```

See also:

- .travis.yml example

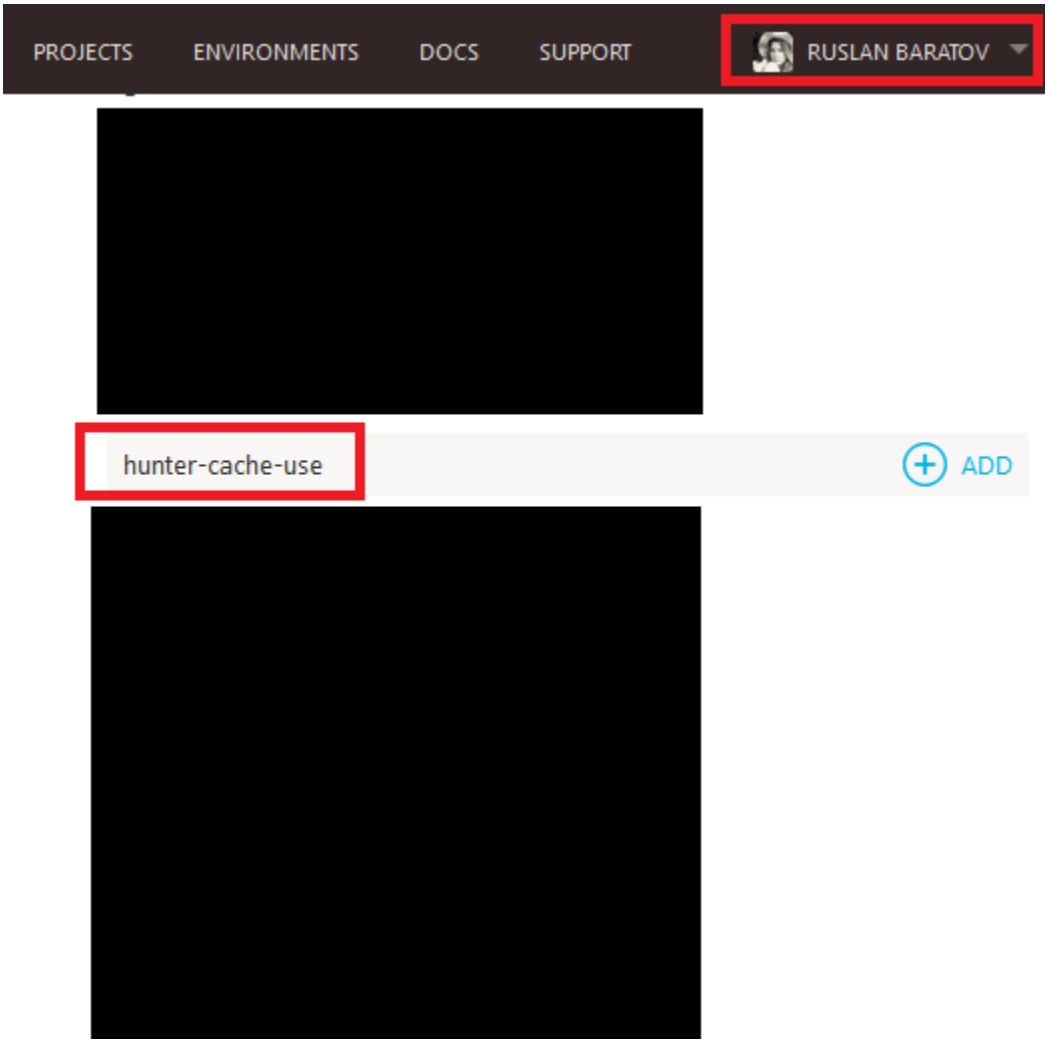
AppVeyor

Note: Visit https://appveyor.com and register `hunter-cache-use` repository there.

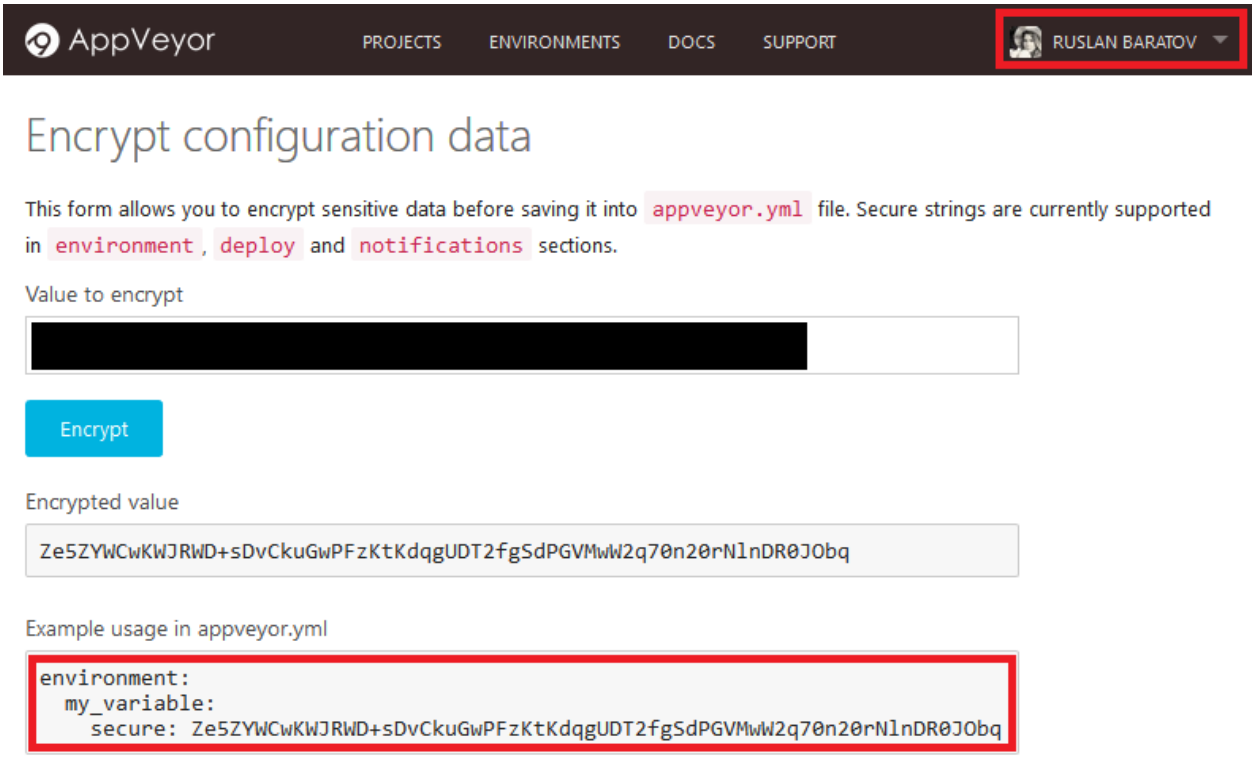
Note: You may want to turn on feature:

- `Enable secure variables in Pull Requests from the same repository only`

Login with account with which you have registered repository for CI. In my case it's my personal account `ruslo`:



While being logged in with the same account use this form to encrypt bot token:



Add it to the `appveyor.yml`:

```
environment:
  global:
    GITHUB_USER_PASSWORD:
      secure: Ze5xxxxxxObq
```

See also:

- appveyor.yml example

CMake code

CI systems are ready, now let's do CMake code.

Note: CMake variables for Hunter should be set to cache before HunterGate, see

- *Hunter: User variables*

HUNTER_CACHE_SERVERS is a list of servers we will use to **download** binaries. We need only one server `https://github.com/forexample/hunter-cache`:

```
set(
    HUNTER_CACHE_SERVERS
    "https://github.com/forexample/hunter-cache"
    CACHE
```

```
    STRING
    "Default cache server"
)
```

We want *HUNTER_RUN_UPLOAD* to be set to `ON` by default only when it's a CI server and secured variable `GITHUB_USER_PASSWORD` is defined. In practice it means:

- Upload will be triggered when new commit pushed to branch
- Upload will be triggered when pull request opened basing on branch **of the same repository**
- Upload will not be available when pull request opened basing on branch from **another repository**
- If `GITHUB_USER_PASSWORD` environment variable defined on local machine there will be no upload by default
- If `GITHUB_USER_PASSWORD` environment variable defined on local machine and `HUNTER_RUN_UPLOAD=ON` **added by user** upload will happen

```
string(COMPARE EQUAL "$ENV{TRAVIS}" "true" is_travis)
string(COMPARE EQUAL "$ENV{APPVEYOR}" "True" is_appveyor)
string(COMPARE EQUAL "$ENV{GITHUB_USER_PASSWORD}" "" password_is_empty)

if((is_travis OR is_appveyor) AND NOT password_is_empty)
  option(HUNTER_RUN_UPLOAD "Upload cache binaries" ON)
endif()
```

File with passwords:

```
set(
    HUNTER_PASSWORDS_PATH
    "${CMAKE_CURRENT_LIST_DIR}/cmake/Hunter/passwords.cmake"
    CACHE
    FILEPATH
    "Hunter passwords"
)
```

There will be no real passwords there, only configuration for repositories and instruction to read password from environment variable `GITHUB_USER_PASSWORD`:

```
# cmake/Hunter/passwords.cmake

hunter_upload_password(
    # REPO_OWNER + REPO = https://github.com/forexample/hunter-cache
    REPO_OWNER "forexample"
    REPO "hunter-cache"

    # USERNAME = https://github.com/ingenue
    USERNAME "ingenue"

    # PASSWORD = GitHub token saved as a secure environment variable
    PASSWORD "$ENV{GITHUB_USER_PASSWORD}"
)
```

Full project available here:

- https://github.com/forexample/hunter-cache-use

Using Nexus Repository manager as binary cache server

It is possible to use `Nexus Repository Manager` as a binary cache server instead of GitHub.

Nexus installation

In order to install and configure Nexus Repository Manager, please follow official documentation. There is also possibility do download `docker images` where `Nexus Repository Manager` is already installed:

- Nexus Repository Manager 2
- Nexus Repository Manager 3

Nexus adding, configuring and managing repositories

To create new or manage existing repository follow this links:

- Adding a new repository
- Managing repositories
- Configuring repositories

Uploading cache binaries to Nexus

The simplest way to upload local cache binaries to Nexus server is by using `cURL`:

```
$ cd hunter/_Base/Cache/meta
$ CACHE_REPOSITORY_URL="http://my.nexus.server.com/content/repositories/hunter/cache"
$ find ./ -type f -exec curl -u nexuser:nexpwd --upload-file "{}" "$CACHE_REPOSITORY_
↪URL/meta/{}"
$ cd ../raw
$ find ./ -type f -exec curl -u nexuser:nexpwd --upload-file "{}" "$CACHE_REPOSITORY_
↪URL/raw/{}"
```

Configuring Hunter to use Nexus

Set *HUNTER_CACHE_SERVERS* variable before `HunterGate` to configure `Hunter` to use `Nexus` server:

```
set(
    HUNTER_CACHE_SERVERS
    "http://my.nexus.server.com/content/repositories/hunter/cache"
    CACHE
    STRING
    "Hunter cache servers"
)

HunterGate(URL "..." SHA1 "...")
```

Using Artifactory as binary cache server

It is possible to use `Artifactory` as a binary cache server.

- https://jfrog.com/artifactory/

Start and prepare

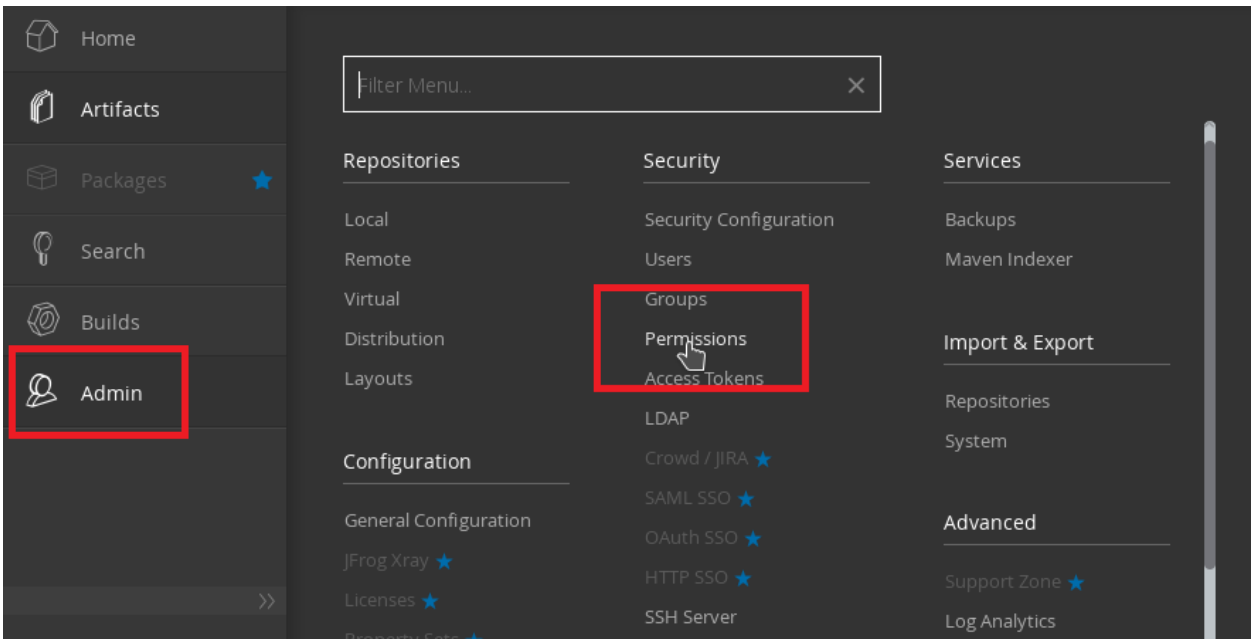
As an example it will be shown how to start Artifactory on local machine from Docker.

Pull and start docker image, forward port 8081:

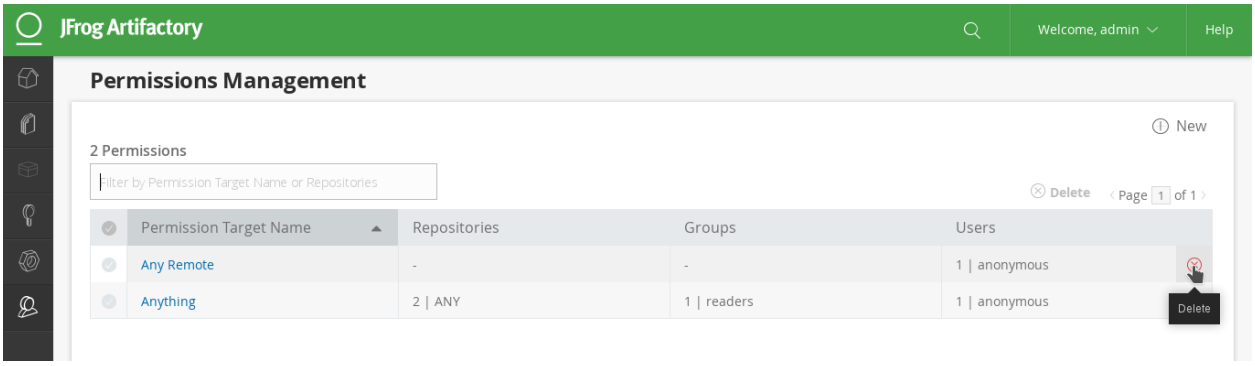
```
> docker run -it -p 8081:8081 docker.bintray.io/jfrog/artifactory-oss bash
```

Open URL http://localhost:8081 in browser and use default login `admin`/`password` to enter.

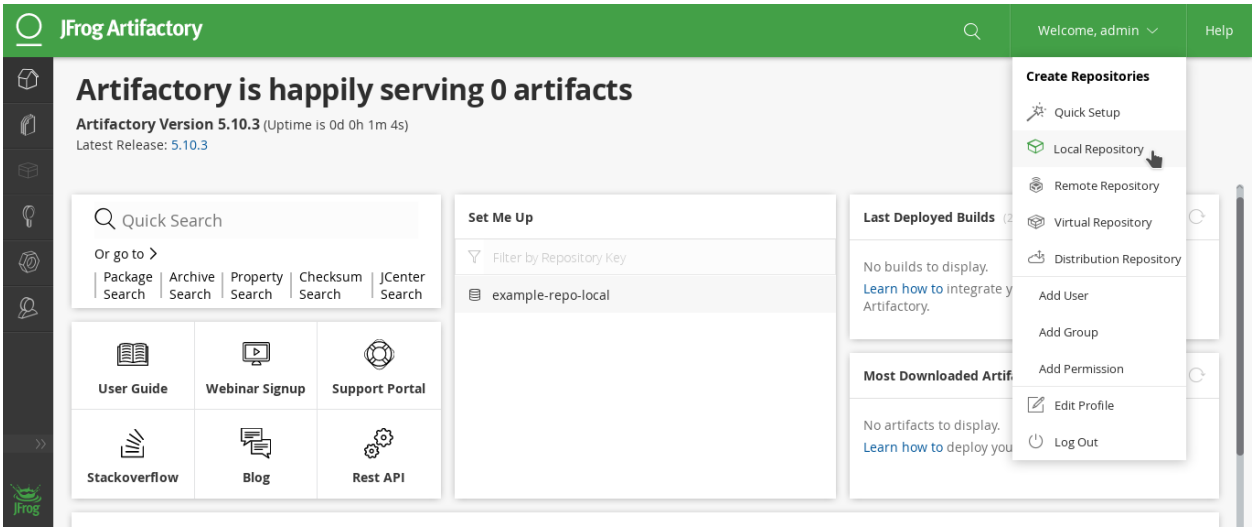
We will use key-based access to binaries without any anonymous reads so let's remove default permission. Go to `Admin` -> `Permissions`:



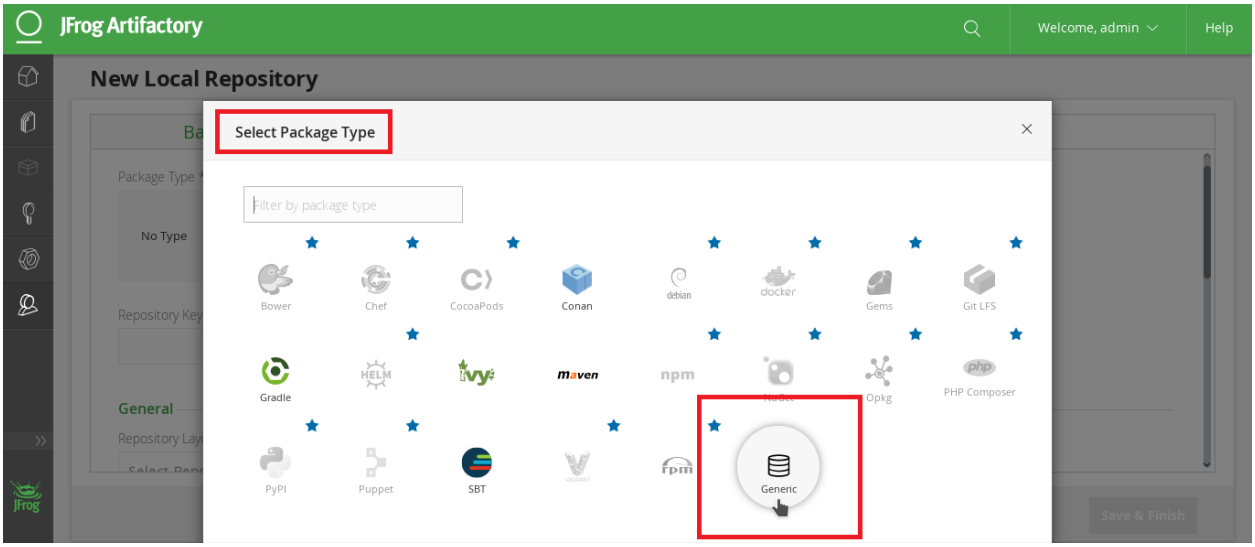
And remove everything:



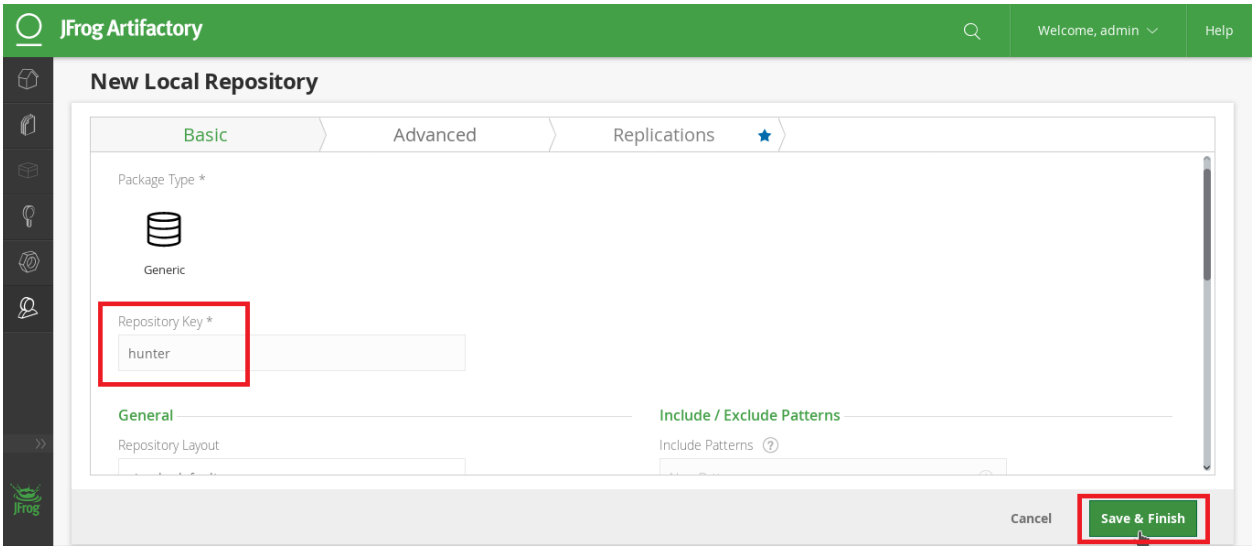
Create `Local repository`:



And choose type `Generic`:

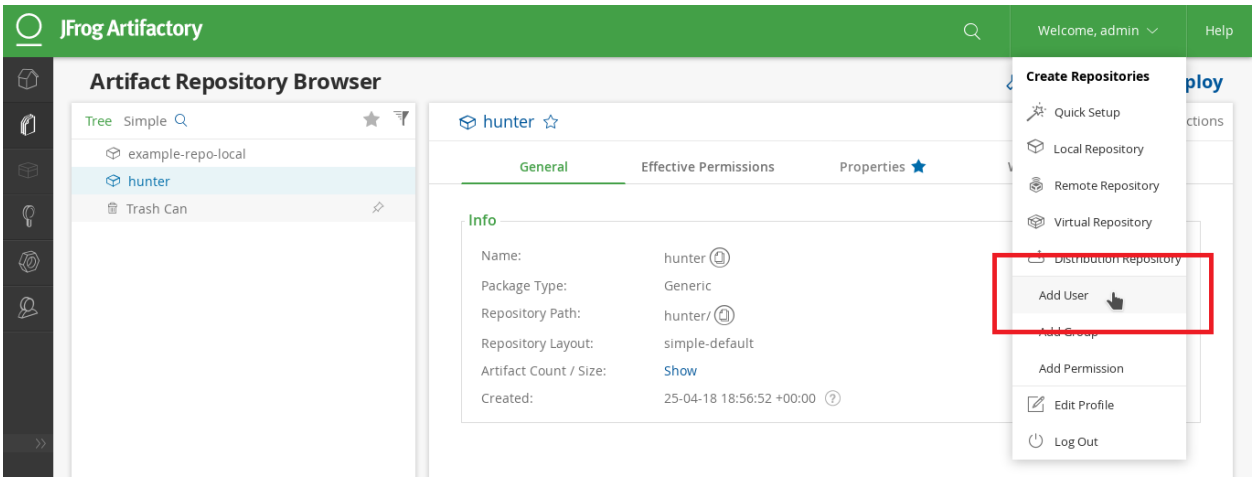


Name it `hunter` and click `Save & Finish`:

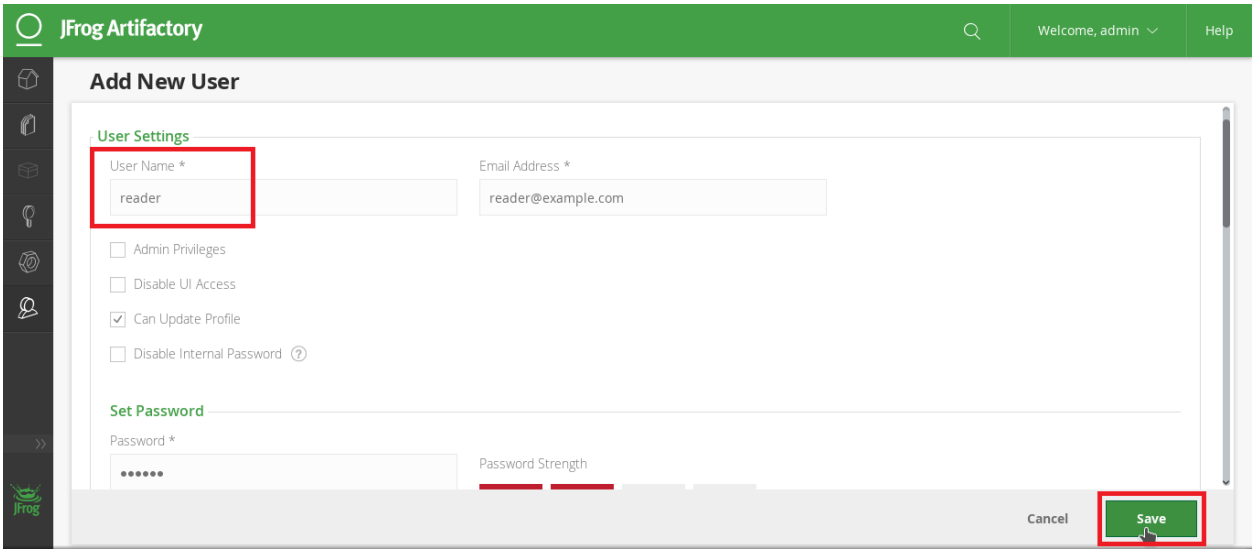


Next let's create user `reader` who will have `Read` access and user `writer` who will have `Deploy/Cache` access.

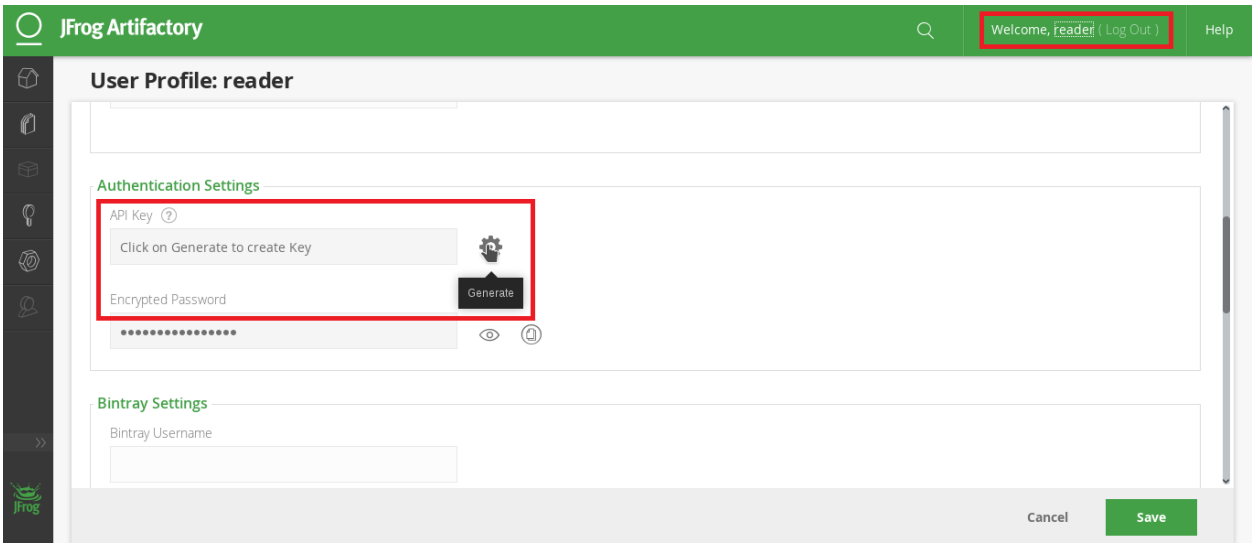
Click `Add User`:



Enter `reader` name and password, click `Save` :



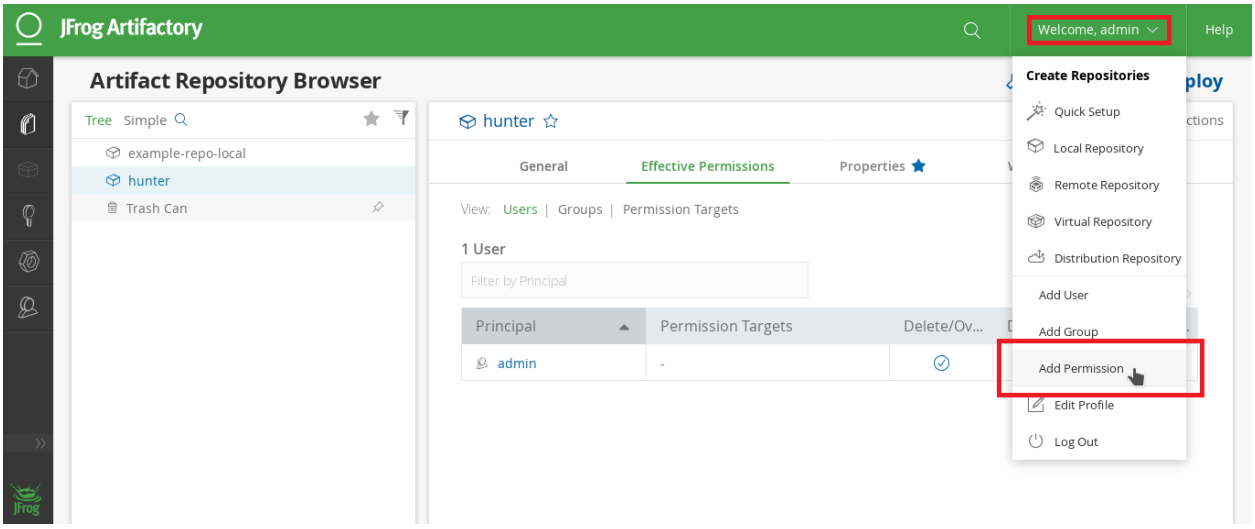
Login as `reader`, go to `Profile` and generate API Key:



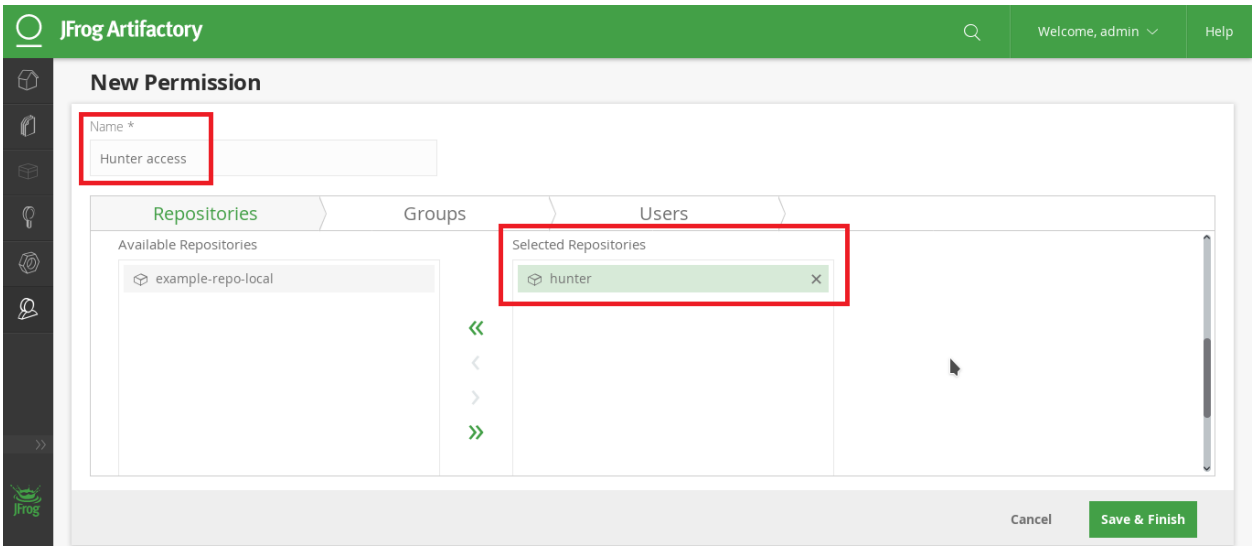
Save this key, further it will be referenced as `artifactory_reader_key`.

Do the same for `writer` user, writer's key will be referenced as `artifactory_writer_key`.

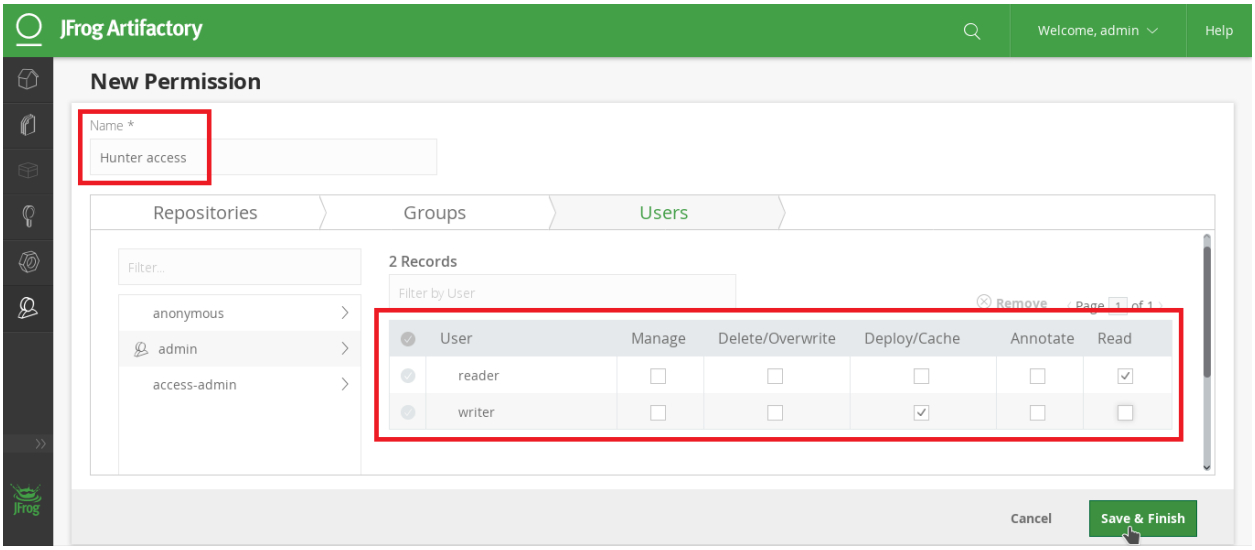
Login as `admin` to give permissions for users:



Name it `Hunter access` and add `hunter` to repositories:



Go to `Users` tab and add `reader/writer`. Give `reader` access of type `Read`. Give `writer` user access or type `Deploy/Cache`:



Note: In real example you will create account with upload access that can do both `Read` and `Deploy/Cache`.

CMake code

Set *HUNTER_CACHE_SERVERS* variable before `HunterGate` to configure `Hunter` to use `Artifactory` server:

```
cmake_minimum_required(VERSION 3.2)

set(
    HUNTER_CACHE_SERVERS
    "http://localhost:8081/artifactory/hunter"
    CACHE
    STRING
    "Default cache server"
)

option(HUNTER_RUN_UPLOAD "Upload cache binaries" ON)

set(
    HUNTER_PASSWORDS_PATH
    "${CMAKE_CURRENT_LIST_DIR}/cmake/Hunter/passwords.cmake"
    CACHE
    FILEPATH
    "Hunter passwords"
)

include(cmake/HunterGate.cmake)
HunterGate(URL "..." SHA1 "...")

project(foo)

hunter_add_package(PNG)
```

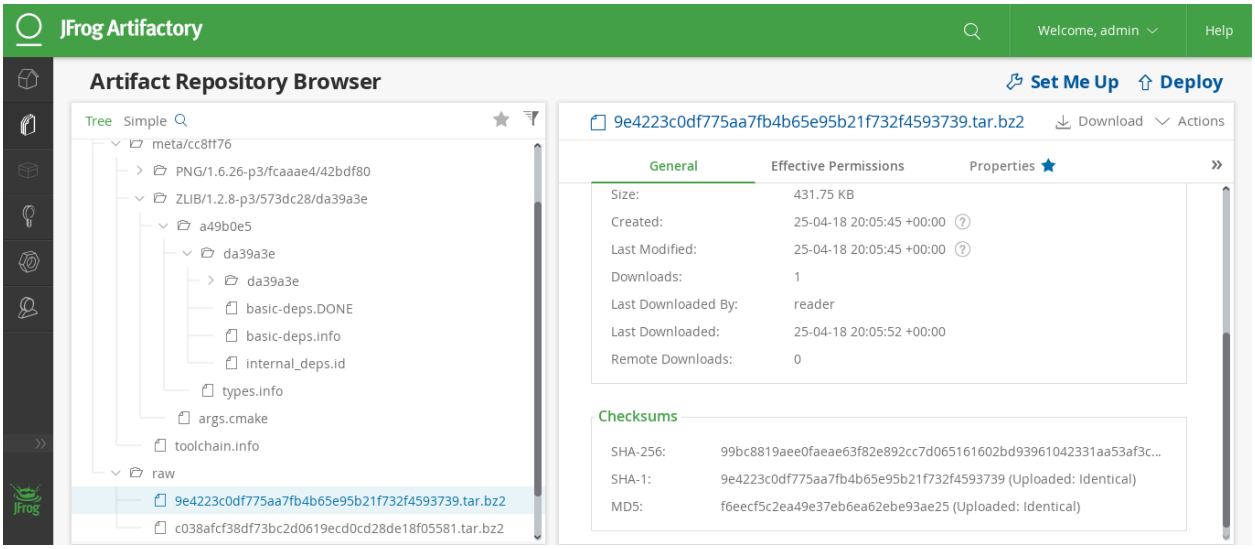
Artifactory keys can be set by `HTTPHEADER` in *Hunter passwords file*:

```
set(artifactory_reader_key "...")
set(artifactory_writer_key "...")

set(server "http://localhost:8081/artifactory/hunter")

hunter_cache_server_password(
    SERVER "${server}"
    HTTPHEADER "X-JFrog-Art-Api: ${artifactory_reader_key}"
    SUB_SHA1_SUFFIX
)

hunter_upload_password(
    SERVER "${server}"
    HTTPHEADER "X-JFrog-Art-Api: ${artifactory_writer_key}"
    SUB_SHA1_SUFFIX
)
```

Note: Artifactory treats URLs like `https://my.server.com/.../file.txt.sha1` as a special URL to get SHA1 hash of file `https://my.server.com/.../file.txt`. Use `SUB_SHA1_SUFFIX` to download all internal Hunter cache meta files of form `abc.sha1` by using `abc_sha1` URL.

Note: `http://localhost:8081` repeated 3 times, that may looks redundant but in general `HUNTER_CACHE_SERVERS` is a list, hence there may be several different servers used. For each server there may be one `hunter_cache_server_password(SERVER ...)` command. Server from `hunter_upload_password` may not be in `HUNTER_CACHE_SERVERS` list at all, though it might not make a lot of sense.

Run CMake build. After build finished update page and check the state of `hunter` repository:



Effectively it's the same as local *Cache layout*.

See also:

- *F.A.Q.: Why binaries from server not used?*

3.3.4 TODO

- add more find_packages
- add toolchain-id flags
- add hunter_add_package
- custom configs
- add package
- -> CGold

3.4 Hunter developer

3.4.1 CMake launch

Parameters that should be specified while launching new CMake instance and `ExternalProject_Add`.

CMAKE_TOOLCHAIN_FILE

- hunter_download
- url_sha1_cmake
- HunterGate

CMAKE_MAKE_PROGRAM

- hunter_download
- url_sha1_cmake
- HunterGate

Note: Use case: Ninja generator without adding Ninja executable to PATH

Note: Test: Check NMake slash/backslash on Windows:

- hunter_finalize

CMAKE_GENERATOR

- hunter_download
- hunter_url_sha1
- HunterGate

CMAKE_GENERATOR_TOOLSET

- hunter_download

Note: Not needed in `ExternalProject_Add` because it will be set by CMake:

- ExternalProject

CMAKE_GENERATOR_PLATFORM

- hunter_download

Note: Not needed in `ExternalProject_Add` because it will be set by CMake:

- ExternalProject

3.4.2 Binary formula

```
SHA1(toolchain.info)
+ SHA1(archive with sources)
+ SHA1(args.cmake)
+ SHA1(types.info)
+ SHA1(internal_deps.id)
+ SHA1(deps.info)
= cache.sha1
```

In terms of *cache*:

```
Toolchain-ID
+ Archive-ID
+ Args-ID
+ Types-ID
+ Internal-Deps-ID
+ Deps-ID
= SHA1 of binaries
```

Note: `*-ID` checked for collision, see *Layout common*.

Hunter-ID

See also:

- *Hunter-ID*

Version -> SHA1 mapping: `Config-ID` can have only `VERSION`, SHA1 of sources will be taken from `hunter.cmake`. I.e. `Hunter-ID` + `Config-ID` -> `Archive-ID`.

`hunter.cmake` can contain default CMake arguments for a package. Resulting arguments will be created by merging default arguments from `hunter.cmake` (low priority) and user arguments from `Config-ID` (high priority). I.e. `Hunter-ID` + `Config-ID` -> `Args-ID`.

`hunter.cmake` can contain default configuration types (`Release`/`Debug`/etc.) for a package. Resulting configuration types will be created by analyzing `HUNTER_CONFIGURATION_TYPES` (low priority), default configuration types from `hunter.cmake` (high priority) and user configuration types from `Config-ID` (highest priority). I.e. `Hunter-ID` + `Toolchain-ID` + `Config-ID` -> `Types-ID`.

See also:

- hunter_get_configuration_types

`hunter.cmake` can contain *PACKAGE_INTERNAL_DEPS_ID*. This variable used only for custom non-CMake build schemes: `Hunter-ID` -> `Internal-Deps-ID`.

Toolchain-ID

See also:

- *Toolchain-ID*

Global settings for all packages, no package specific information saved here. Created by analyzing an output of compilation of C++ file ShowPredefined.cpp (created from list). We get unified information about compiler, compiler version, compiler flags, etc. (everything from user's `CMAKE_TOOLCHAIN_FILE`).

Additionally next global variables saved there too:

- `IPHONEOS_ARCHS` (Polly toolchains)
- `IPHONESIMULATOR_ARCHS` (Polly toolchains)
- `CMAKE_GENERATOR`
- `HUNTER_CONFIGURATION_TYPES`
- `HUNTER_TOOLCHAIN_UNDETECTABLE_ID`
- `HUNTER_BUILD_SHARED_LIBS`
- `OSX_SDK_VERSION` (Polly toolchains)

Config-ID

Package specific information saved here. Created by merging file with hunter_default_version commands and user's `config.cmake` with `hunter_config` commands (if present). Result is automatically generated `config.cmake` file with `hunter_final_config` command. First 7 digits of SHA1 of `config.cmake` forms `Config-ID`. Used while calculating `Archive-ID`, `Args-ID`, `Types-ID` (see above).

See also:

- *Layouts*
- *Internal modules*

CHAPTER 4

Packages

List of packages and usage instructions for each package.

All packages

- *Alphabetically*
- Index

4.1 All packages

4.1.1 ARM_NEON_2_x86_SSE

- Official
- Hunterized
- Example

```
hunter_add_package(ARM_NEON_2_x86_SSE)
find_package(ARM_NEON_2_x86_SSE CONFIG REQUIRED)
target_link_libraries(... ARM_NEON_2_x86_SSE::ARM_NEON_2_x86_SSE)
```

4.1.2 AllTheFlopsThreads

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official

- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.3 Android-ARM-EABI-v7a-System-Image

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.4 Android-ARM64-v8a-System-Image

- Example

```
hunter_add_package(Android-ARM64-v8a-System-Image)
```

4.1.5 Android-Apk

- Official
- Hunterized
- Usage examples

```
hunter_add_package(Android-Apk)
list(APPEND CMAKE_MODULE_PATH "${ANDROID-APK_ROOT}")
include(AndroidApk)
add_library(simple ...)
android_create_apk(NAME simple DIRECTORY "${CMAKE_CURRENT_BINARY_DIR}/apk")
```

4.1.6 Android-Build-Tools

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.7 Android-Google-APIs

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.8 Android-Google-APIs-Intel-x86-Atom-System-Image

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.9 Android-Google-Repository

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.10 Android-Intel-x86-Atom-System-Image

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.11 Android-MIPS-System-Image

- Example

```
hunter_add_package(Android-MIPS-System-Image)
```

4.1.12 Android-Modules

- Official
- Hunterized

```
hunter_add_package(Android-Modules)

list(APPEND CMAKE_MODULE_PATH "${ANDROID-MODULES_ROOT}")
```

```
include(AndroidNdkGdb)
include(AndroidNdkModules)
```

4.1.13 Android-SDK

This module helps to create Android SDK directory:

- https://github.com/hunter-packages/android-sdk

```
hunter_add_package(Android-SDK)
message("Path to `android`: ${ANDROID-SDK_ROOT}/android-sdk/tools/android")
message("Path to `emulator`: ${ANDROID-SDK_ROOT}/android-sdk/tools/emulator")
message("Path to `adb`: ${ANDROID-SDK_ROOT}/android-sdk/platform-tools/adb")
```

- Examples

4.1.14 Android-SDK-Platform

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.15 Android-SDK-Platform-tools

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.16 Android-SDK-Tools

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.17 Android-Support-Repository

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.18 AngelScript

- Official
- Hunterized
- Example

```
hunter_add_package(AngelScript)
find_package(AngelScript CONFIG REQUIRED)
target_link_libraries(boo PUBLIC AngelScript::AngelScript)
```

4.1.19 ArrayFire

- Official
- Hunterized

- Example
- Testing
- Available since

```
hunter_add_package(ArrayFire)
find_package(ArrayFire CONFIG REQUIRED)
target_link_libraries(... ArrayFire::af)
target_link_libraries(... ArrayFire::afcpu)
```

4.1.20 Assimp

- http://assimp.org/
- Official GitHub repo
- Hunterized
- Example

```
hunter_add_package(Assimp)
find_package(Assimp CONFIG REQUIRED)
target_link_libraries(... Assimp::assimp)
```

4.1.21 Async++

- Official
- Hunterized
- Available since
- Added by Andrei Laphin (pr-268)

```
hunter_add_package(Async++)
find_package(Async++ CONFIG REQUIRED)

target_link_libraries(... Async++::Async++)
```

4.1.22 Avahi

- Official
- Example
- Available since
- Added by Damien Buhl (pr-237)

```
hunter_add_package(Avahi)
find_package(Avahi REQUIRED)
target_link_libraries(... Avahi::common Avahi::client Avahi::compat_libdns_sd)
```

4.1.23 BZip2

- Official
- Hunterized
- Example
- Available since

```
hunter_add_package(BZip2)

find_package(BZip2 CONFIG REQUIRED)
target_link_libraries(... BZip2::bz2)
```

4.1.24 Beast

- Official
- Hunterized
- Available since

```
hunter_add_package(Beast)
find_package(Beast CONFIG REQUIRED)
target_link_libraries(... Beast::Beast)
```

4.1.25 Boost

```
# Header-only libraries
hunter_add_package(Boost)
find_package(Boost CONFIG REQUIRED)
target_link_libraries(... Boost::boost)
```

- Example

Since boost 1.70 you should use for header only libraries as target:

```
target_link_libraries(... Boost::headers)
```

```
# Boost components (see list below)
hunter_add_package(Boost COMPONENTS system filesystem)
find_package(Boost CONFIG REQUIRED system filesystem)
target_link_libraries(... Boost::system Boost::filesystem)
```

Examples:

- Boost-uuid
- Boost-system
- Boost-iostreams
- Boost-filesystem
- Boost-math
- Boost-contract

- Boost-stacktrace

List of components and availability (other libraries are header-only):

```
  hunter_append_component(${common_args} COMPONENT atomic          SINCE 1.53.0)
  hunter_append_component(${common_args} COMPONENT chrono          SINCE 1.47.0)
  hunter_append_component(${common_args} COMPONENT container       SINCE 1.56.0)
  hunter_append_component(${common_args} COMPONENT context         SINCE 1.51.0)
  hunter_append_component(${common_args} COMPONENT contract        SINCE 1.67.0)
  hunter_append_component(${common_args} COMPONENT coroutine       SINCE 1.53.0)
  hunter_append_component(${common_args} COMPONENT coroutine2      SINCE 1.60.0 ␣
↪UNTIL 1.65.0)
  hunter_append_component(${common_args} COMPONENT date_time       SINCE 1.29.0)
  hunter_append_component(${common_args} COMPONENT exception       SINCE 1.36.0)
  hunter_append_component(${common_args} COMPONENT fiber           SINCE 1.62.0)
  hunter_append_component(${common_args} COMPONENT filesystem      SINCE 1.30.0)
  hunter_append_component(${common_args} COMPONENT graph           SINCE 1.18.0)
  hunter_append_component(${common_args} COMPONENT graph_parallel  SINCE 1.18.0)
  hunter_append_component(${common_args} COMPONENT iostreams       SINCE 1.33.0)
  hunter_append_component(${common_args} COMPONENT json            SINCE 1.75.0)
  hunter_append_component(${common_args} COMPONENT locale          SINCE 1.48.0)
  hunter_append_component(${common_args} COMPONENT log             SINCE 1.54.0)
  hunter_append_component(${common_args} COMPONENT math            SINCE 1.23.0)
  hunter_append_component(${common_args} COMPONENT metaparse       SINCE 1.61.0 UNTIL␣
↪1.66.0)
  hunter_append_component(${common_args} COMPONENT mpi             SINCE 1.35.0)
  hunter_append_component(${common_args} COMPONENT nowide          SINCE 1.74.0)
  hunter_append_component(${common_args} COMPONENT program_options SINCE 1.32.0)
  hunter_append_component(${common_args} COMPONENT python          SINCE 1.19.0)
  hunter_append_component(${common_args} COMPONENT random          SINCE 1.15.0)
  hunter_append_component(${common_args} COMPONENT regex           SINCE 1.18.0)
  hunter_append_component(${common_args} COMPONENT serialization   SINCE 1.32.0)
  hunter_append_component(${common_args} COMPONENT signals         SINCE 1.29.0 UNTIL␣
↪1.69.0)
  hunter_append_component(${common_args} COMPONENT stacktrace      SINCE 1.65.0)
  hunter_append_component(${common_args} COMPONENT system          SINCE 1.35.0)
  hunter_append_component(${common_args} COMPONENT test            SINCE 1.21.0)
  hunter_append_component(${common_args} COMPONENT thread          SINCE 1.25.0)
  hunter_append_component(${common_args} COMPONENT timer           SINCE 1.9.0)
  hunter_append_component(${common_args} COMPONENT type_erasure    SINCE 1.60.0)
  hunter_append_component(${common_args} COMPONENT url             SINCE 1.81.0)
  hunter_append_component(${common_args} COMPONENT wave            SINCE 1.33.0)
```

CMake options

You can use `CMAKE_ARGS` feature (see customization) to pass options to boost build or to append config macros in the default boost user config file (`boost/config/user.hpp`):

- Options of special form `<COMPONENT-UPPERCASE>_<OPTION>=<VALUE>` will be added to `b2` as `-s <OPTION>=<VALUE>` while building component . For example:

```
# Add 'NO_BZIP2=1' to the b2 build of iostreams library,
# i.e. `b2 -s NO_BZIP2=1`
hunter_config(
    Boost
    VERSION ${HUNTER_Boost_VERSION}
    CMAKE_ARGS IOSTREAMS_NO_BZIP2=1
)
```

- boost.iostreams options
- Options CONFIG_MACRO_<ID>=<VALUE> will append #define <ID> <VALUE> to the default boost user config header file. And options CONFIG_MACRO=<ID_1>;<ID_2>;...;<ID_n> will append #define <ID_1>, #define <ID_2>, ..., #define <ID_n>. Example:

```
hunter_config(
    Boost
    VERSION ${HUNTER_Boost_VERSION}
    CMAKE_ARGS
    CONFIG_MACRO=BOOST_REGEX_MATCH_EXTRA;BOOST_MPL_CFG_NO_PREPROCESSED_HEADERS
    CONFIG_MACRO_BOOST_MPL_LIMIT_LIST_SIZE=3
)
```

  Will append the next lines to boost/config/user.hpp:

```
#define BOOST_REGEX_MATCH_EXTRA
#define BOOST_MPL_CFG_NO_PREPROCESSED_HEADERS
#define BOOST_MPL_LIMIT_LIST_SIZE 3
```

- Option USE_CONFIG_FROM_BOOST=ON use the package configuration file provided by the boost project.

  Since boost version 1.70.0, the boost project provide a well maintained package configuration file for use with find_package's config mode. As minimum required CMake version you need 3.3.

  See the difference between following example:

  - Boost-log
  - Boost-log-useBoostConfig

- Option BOOST_USE_WINAPI_VERSION=<API_VERSION> use on Windows in order to set the Windows API version used for building the boost libraries.

  Since Boost 1.78.0 Boost.Log exports additional symbols when building for Windows 8 or newer. So it is recommended to set the CMake variable BOOST_USE_WINAPI_VERSION in the CMake-toolchain file (or the CMAKE_ARGS) to the same value as the defines _WIN32_WINNT and WINVER.

  - Boost.WinAPI documentation

  The version passed must match the hexadecimal integer values used for _WIN32_WINNT and WINVER. The version numbers are described in Windows Headers documentation.

  API_VERSION is passed as a hexadecimal integer e.g. BOOST_USE_WINAPI_VERSION=0x0603 sets the Windows API version to Windows 8.1.

Python

To require Boost Python to be built against a specific version of Python installed on the system, option PYTHON_VERSION=<VALUE> may be used. In this case, if the required components of Python are located, user_config.jam will be appended with the following line:

```
using python  : <requested_version_number> : <path to Python executable> :
<path to Python include directory> : <path to directory containing the Python library>
↪ ;
```

Example for Python 2:

```
# config.cmake
hunter_config(
  Boost
  VERSION ${HUNTER_Boost_VERSION}
  CMAKE_ARGS
  PYTHON_VERSION=2.7.15
)
```

```
# CMakeLists.txt
hunter_add_package(Boost COMPONENTS python)
if(Boost_VERSION VERSION_LESS 106700)
  find_package(Boost CONFIG REQUIRED python)
else()
  find_package(Boost CONFIG REQUIRED python27)
endif()
```

Note: Python<x> component arguments to `find_package(Boost ...)` after Boost version 1.67 require a specific version suffix, e.g. python37.

Example for Python 3:

```
# config.cmake
hunter_config(
  Boost
  VERSION ${HUNTER_Boost_VERSION}
  CMAKE_ARGS
  PYTHON_VERSION=3.6.7
)
```

```
# CMakeLists.txt
hunter_add_package(Boost COMPONENTS python)
if(Boost_VERSION VERSION_LESS 106700)
  find_package(Boost CONFIG REQUIRED python3)
else()
  find_package(Boost CONFIG REQUIRED python36)
endif()
```

Python NumPy

To build the NumPy plugin for Boost Python use option `HUNTER_ENABLE_BOOST_PYTHON_NUMPY=True`. This will require `pip_numpy` and therefore `hunter_venv`, see their docs for details and requirements.

Example:

```
# config.cmake
hunter_config(
  Boost
  VERSION ${HUNTER_Boost_VERSION}
  CMAKE_ARGS
  PYTHON_VERSION=${PYTHON_VERSION}
  HUNTER_ENABLE_BOOST_PYTHON_NUMPY=True
)
```

Math

When using Boost Math you will need to link in the libraries, however these are not named `math` but rather are individual based on what you need to link it, the easiest of which is to link in all parts:

```
hunter_add_package(Boost COMPONENTS math)
find_package(Boost CONFIG REQUIRED math_c99 math_c99f math_c99l math_tr1 math_tr1f␣
↪math_tr1l)
target_link_libraries(...
  Boost::math_c99
  Boost::math_c99f
  Boost::math_c99l
  Boost::math_tr1
  Boost::math_tr1f
  Boost::math_tr1l
)
```

If you are using only the header-only parts of Boost::Math then the libraries can be ignored:

```
hunter_add_package(Boost COMPONENTS math)
find_package(Boost CONFIG REQUIRED)
```

4.1.26 BoostCompute

- Official
- Hunterized
- Example
- Available since

```
hunter_add_package(BoostCompute)
find_package(BoostCompute CONFIG REQUIRED)
target_link_libraries(... BoostCompute::boost_compute)
```

4.1.27 BoostProcess

- Official
- Hunterized
- Example
- Added by Alexander Lamaison (pr-330)
- Available since

```
hunter_add_package(BoostProcess)
find_package(BoostProcess CONFIG REQUIRED)
target_link_libraries(... BoostProcess::boost_process)
```

4.1.28 BoringSSL

Warning:

- This library implements *OpenSSL* API. Usage of this package can lead to conflicts. Please read this issue and make sure you're understand what you're doing.

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1186)

```
hunter_add_package(BoringSSL)
find_package(BoringSSL CONFIG REQUIRED)
target_link_libraries(boo BoringSSL::ssl BoringSSL::crypto)
```

4.1.29 Box2D

- http://box2d.org
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(Box2D)
find_package(Box2D CONFIG REQUIRED)
target_link_libraries(boo PUBLIC Box2D::Box2D)
```

4.1.30 CLAPACK

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.31 CLI11

- Official
- Example
- Added by Paweł Bylica (pr-1446)

```
hunter_add_package(CLI11)
find_package(CLI11 CONFIG REQUIRED)

add_executable(CLI11-example main.cpp)
target_link_libraries(CLI11-example CLI11::CLI11)
```

4.1.32 CURL

- Official
- Hunterized
- Example

```
hunter_add_package(CURL)
find_package(CURL CONFIG REQUIRED)
target_link_libraries(... CURL::libcurl)
```

4.1.33 CapnProto

- Official
- Example
- Available since

```
hunter_add_package(CapnProto)

find_package(CapnProto CONFIG REQUIRED)
target_link_libraries(... CapnProto::capnp)
```

4.1.34 Catch

- Official
- Hunterized
- Available since

```
find_package(Catch2 CONFIG REQUIRED)

set(SOURCES main.cpp
            foo_test.cpp
            foo.cpp)
set(HEADERS foo.hpp)

add_executable(foo_test ${SOURCES} ${HEADERS})
target_link_libraries(foo_test PUBLIC Catch2::Catch2)
```

4.1.35 Clang

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.36 ClangToolsExtra

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.37 Comet

- Official

```
hunter_add_package(Comet)
find_package(Comet CONFIG REQUIRED)
target_link_libraries(... Comet::comet)
```

4.1.38 CppNetlib

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official

- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.39 CppNetlibUri

- __FIXME__ Hunterized

```
hunter_add_package(CppNetlibUri)
find_package(CppNetlibUri CONFIG REQUIRED)

target_link_libraries(foo network-uri)
```

4.1.40 CreateLaunchers

- Official GitHub
- Example

```
hunter_add_package(CreateLaunchers)
find_package(CreateLaunchers CONFIG REQUIRED)

add_executable(CreateLaunchers_test main.cpp)

include(CreateLaunchers)

create_target_launcher(CreateLaunchers_test
    ARGS "-a"
    RUNTIME_LIBRARY_DIRS "./"
    WORKING_DIRECTORY ${CMAKE_CURRENT_LIST_DIR}
)
```

4.1.41 CsvParserCPlusPlus

- Official
- Available since

```
hunter_add_package(CsvParserCPlusPlus)
find_package(CsvParserCPlusPlus CONFIG REQUIRED)
target_link_libraries(... CsvParserCPlusPlus::csv_parser_cplusplus)
```

4.1.42 EGL-Registry

- Official

- Hunterized
- Example
- Added by Rahul Sheth (pr-423)

```
hunter_add_package(EGL-Registry)
find_package(EGL-Registry CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC EGL::Registry-Headers)
```

4.1.43 Eigen

- http://eigen.tuxfamily.org
- Official
- Official Git mirror on GitHub
- Hunterized
- Maintainer: https://github.com/NeroBurner

```
hunter_add_package(Eigen)
find_package(Eigen3 CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo Eigen3::Eigen)
```

For Hunter < v0.17.15

```
hunter_add_package(Eigen)
find_package(Eigen REQUIRED)
target_link_libraries(... Eigen::eigen)
```

4.1.44 Expat

- Official
- Example
- Added by Alexander Lamaison (pr-59)

```
hunter_add_package(Expat)
find_package(EXPAT REQUIRED)

target_link_libraries(... ${EXPAT_LIBRARIES})
target_include_directories(... ${EXPAT_INCLUDE_DIRS})
```

CI

- https://github.com/cpp-pm/hunter-testing/tree/pkg.expat

4.1.45 FLAC

- Official
- Hunterized
- Example
- Added by drodin (pr-N)

```
hunter_add_package(FLAC)
find_package(FLAC CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC FLAC::FLAC++)
```

4.1.46 FP16

- Official
- Hunterized
- Example
- Added by xsacha (pr-1787)

```
hunter_add_package(FP16)
find_package(FP16 REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC FP16::fp16)
```

4.1.47 FakeIt

- Official
- Hunterized
- Available since

```
hunter_add_package(FakeIt)
find_package(FakeIt CONFIG REQUIRED)
target_link_libraries(... FakeIt::FakeIt)
```

4.1.48 Fruit

- Official
- Hunterized
- Versions
- Example
- Added by Alexey Shevchenko (pr-1527)

```
hunter_add_package(Fruit)
find_package(Fruit CONFIG REQUIRED)

add_executable(foo foo.cpp)

target_link_libraries(foo PRIVATE Fruit::fruit)
```

Note: Boost disabled by default since it has issues on some platforms.

Use `FRUIT_USES_BOOST=ON` to enable it.

```
# config.cmake
hunter_config(Fruit
    VERSION ${HUNTER_Fruit_VERSION}
    CMAKE_ARGS
        FRUIT_USES_BOOST=ON
)
```

4.1.49 FunctionalPlus

- Official
- Example

```
hunter_add_package(FunctionalPlus)
find_package(FunctionalPlus CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo FunctionalPlus::fplus)
```

4.1.50 GPUImage

- Official
- Hunterized
- Example

```
hunter_add_package(GPUImage)
find_package(GPUImage CONFIG REQUIRED)

add_executable(foo foo.mm)
target_link_libraries(foo PUBLIC GPUImage::gpuimage)
```

4.1.51 GSL

- GitHub mirror
- Hunterized
- Example

```
hunter_add_package(GSL)
find_package(GSL CONFIG REQUIRED)
target_link_libraries(... GSL::gsl)
```

4.1.52 GTest

- Official
- Hunterized (old repo)
- Dev branch
- Versions
- Example
- Added by Knitschi (pr-306)

```
hunter_add_package(GTest)
find_package(GTest CONFIG REQUIRED)

target_link_libraries(foo GTest::gtest_main) # GTest::gtest will be linked␣
↪automatically
target_link_libraries(boo GTest::gtest)
```

Bugs

- Cygwin GCC build failed with `c++11` flag

4.1.53 GMock

- Available since

For package versions 1.8.0-hunter-p1 and higher the package also includes GMock. When finding the GMock package GTest is automatically included. Note that package version 1.8.0-hunter-p1 does **not** support the **MinGW** and **Visual Studio 2005** toolchains, so GMock is not available in these cases.

```
hunter_add_package(GTest)
find_package(GTest CONFIG REQUIRED)

# GMock::gmock and GTest::gtest will be linked automatically
target_link_libraries(foo GTest::gmock_main)
```

4.1.54 HalideIR

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(HalideIR)
find_package(HalideIR CONFIG REQUIRED)
```

```
add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC HalideIR::HalideIR)
```

4.1.55 HastyNoise

- Official
- Example
- Added by Casey (pr-1832)

HastyNoise SIMD open source noise generation library with a large collection of different noise algorithms.

```
hunter_add_package(HastyNoise)
find_package(HastyNoise CONFIG REQUIRED)

add_executable(hastynoise_test main.cpp)
target_link_libraries(hastynoise_test HastyNoise::hastyNoise)
```

4.1.56 ICU

- http://site.icu-project.org/
- Hunterized
- Example
- Stand-alone example with advanced testing

```
hunter_add_package(ICU)
find_package(ICU CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC ICU::uc)
```

Available targets:

- `ICU::data`
- `ICU::i18n`
- `ICU::io` (only if `ICU_BUILD_IO=YES`)
- `ICU::le`
- `ICU::lx` (available in *icu-lx* package)
- `ICU::tu` (only if `ICU_BUILD_TOOLS=YES`)
- `ICU::uc`
- `ICU::pkgdata` (only if `ICU_BUILD_TOOLS=YES`)
- `ICU::icupkg` (only if `ICU_BUILD_TOOLS=YES`)

If `ICU_BUILD_TOOLS` is set to `YES` also next variables available:

- `ICU_PKGDATA_EXECUTABLE`
- `ICU_ICUPKG_EXECUTABLE`

If `ICU_DATA_ARCHIVE_MODE` is set to `YES` also next variables available:

- `ICU_DATA_FILE`

Options:

- `ICU_DATA_ARCHIVE_MODE=ON` (equals to `--with-data-packaging=archive`)

4.1.57 IF97

- Official
- Hunterized
- Example
- Added by Jorrit Wronski (pr-1201)

```
hunter_add_package(IF97)
find_package(IF97 CONFIG REQUIRED)
target_link_libraries(IF97 IF97::IF97)
```

4.1.58 Igloo

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.59 IlmBase

- Official
- Hunterized
- Example
- Added by Harry Mallon (pr-138)

```
hunter_add_package(IlmBase)
find_package(IlmBase CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC IlmBase::Half IlmBase::Imath)
```

4.1.60 Imath

- Official
- Example
- Added by Harry Mallon (pr-391)

```
hunter_add_package(Imath)
find_package(Imath CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Imath::Imath)
```

4.1.61 Immer

- Official
- Example
- Added by Joerg-Christian Boehme (pr-104)

```
hunter_add_package(Immer)
find_package(Immer CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC immer)
```

4.1.62 Jpeg

- Official
- Hunterized
- Example

```
hunter_add_package(Jpeg)
find_package(JPEG CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC JPEG::jpeg)
```

4.1.63 JsonSpirit

- Official
- Hunterized

```
hunter_add_package(JsonSpirit)
find_package(JsonSpirit CONFIG REQUIRED)

target_link_libraries(foo json)
```

4.1.64 KTX-Software

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-435)

```
hunter_add_package(KTX-Software)
find_package(KTX-Software CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC KTX-Software::ktx)
```

4.1.65 KhronosDataFormat

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-280)

```
hunter_add_package(KhronosDataFormat)
find_package(KhronosDataFormat CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Khronos::KhronosDataFormat)
```

4.1.66 LAPACK

- Official
- Hunterized
- Example
- Available since
- Added by NeroBurner (pr-860)

```
hunter_add_package(LAPACK)
find_package(LAPACK CONFIG REQUIRED)
target_link_libraries(foo blas lapack)
```

4.1.67 LLVM

- Official
- Hunterized
- Example

```
hunter_add_package(LLVM)
find_package(LLVM CONFIG REQUIRED)

include_directories(${LLVM_INCLUDE_DIRS})
add_definitions(${LLVM_DEFINITIONS})
llvm_map_components_to_libnames(llvm_libs support core)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC ${llvm_libs})
```

See also:

- http://llvm.org/docs/CMake.html#embedding-llvm-in-your-project

Usage issues

- Exceptions are not available, need to build with _HAS_EXCEPTION=0 (Visual Studio)

4.1.68 LLVMCompilerRT

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.69 Lager

- Official
- Hunterized
- Example
- Added by Joerg-Christian Boehme (pr-118)

```
hunter_add_package(Lager)
find_package(Lager CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC lager)
```

4.1.70 Leathers

- Official

```
hunter_add_package(Leathers)
find_package(Leathers CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Leathers::leathers)
```

4.1.71 Leptonica

- Official
- GitHub
- Hunterized
- Example
- Available since
- Added by Sacha Refshauge (pr-815)

```
hunter_add_package(Leptonica)
find_package(Leptonica CONFIG REQUIRED)
add_executable(example example.c)

target_link_libraries(example Leptonica::leptonica)
```

4.1.72 LibCDS

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1194)

```
hunter_add_package(LibCDS)
find_package(LibCDS CONFIG REQUIRED)
target_link_libraries(... LibCDS::cds)  # Use cds-s for static library
```

4.1.73 Libcxx

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z

- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.74 Libcxxabi

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.75 Libevent

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1019)

```
hunter_add_package(Libevent)
find_package(Libevent CONFIG REQUIRED)
target_link_libraries(... Libevent::event_core
                          Libevent::event_extra)
```

4.1.76 Libssh2

- Official
- Hunterized
- Added by Alexander Lamaison (pr-48)

4.1.77 LodePNG

- Official
- Hunterized
- Example

- Added by Brad Kotsopoulos (pr-1636)

```
hunter_add_package(LodePNG)
find_package(LodePNG CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC LodePNG::lodepng)
```

4.1.78 Lua

- Official
- Hunterized
- Example
- Available since
- Added by Denis Kerzhemanov (pr-283)

```
hunter_add_package(Lua)
find_package(Lua CONFIG REQUIRED)
```

```
# Imported target can be used as-is
# in "build time" commands like 'add_custom_target'
add_custom_target(
    show_lua_version
    Lua::lua -v
    COMMENT "Show version of Lua executable"
)
```

```
# Full path to executable 'LUA_EXECUTABLE' should be used
# for "generate time" commands like 'execute_process'
execute_process(
    COMMAND ${LUA_EXECUTABLE} -v
    RESULT_VARIABLE result
    OUTPUT_VARIABLE output
    ERROR_VARIABLE error
    OUTPUT_STRIP_TRAILING_WHITESPACE
    ERROR_STRIP_TRAILING_WHITESPACE
)
```

```
# Library usage
target_link_libraries(boo PUBLIC Lua::lua_lib)
```

4.1.79 MathFu

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1687)

```
hunter_add_package(MathFu)
find_package(MathFu CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC MathFu::mathfu)
```

4.1.80 Microsoft.GSL

- Official
- Hunterized
- Example
- Added by Stefan Reinhold (pr-1499)

```
hunter_add_package(Microsoft.GSL)
find_package(Microsoft.GSL CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Microsoft.GSL::GSL)
```

4.1.81 MySQL-client

- Official
- Example
- Available since

```
hunter_add_package(MySQL-client)
find_package(MySQL-client REQUIRED)
target_link_libraries(... "MySQL::libmysql")
```

before Hunter v0.19.58

```
hunter_add_package(MySQL-client)
find_package(MySQL-client REQUIRED)
target_link_libraries(... "MySQL::client")
```

4.1.82 NASM

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.83 NLopt

- Official
- Hunterized
- Example
- Added by t0p4 (pr-1617)

```
hunter_add_package(NLopt)
find_package(NLopt CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC NLopt::nlopt)
```

4.1.84 ONNX

- Official
- Hunterized
- Example
- Added by xsacha (pr-1785)

```
hunter_add_package(ONNX)
find_package(ONNX CONFIG REQUIRED)

add_executable(test test.cpp)
target_link_libraries(test PUBLIC onnx::onnx onnx::onnxifi)
```

4.1.85 OpenAL

- Official
- Example
- Added by Isaac Hier (pr-1128)

```
hunter_add_package(OpenAL)
find_package(OpenAL CONFIG REQUIRED)
target_link_libraries(... OpenAL::OpenAL)
```

4.1.86 OpenBLAS

- Official
- Hunterized
- Available since

- Example

```
hunter_add_package(OpenBLAS)
find_package(OpenBLAS CONFIG REQUIRED)
target_link_libraries(... OpenBLAS::OpenBLAS)
```

Starting with OpenBLAS v0.3.21 LAPACK support is enabled by default in Hunter. This is due to upstream adding a f2c-converted copy of LAPACK 3.9.0 as fallback if no Fortran compiler is available.

4.1.87 OpenCL

- Official
- Hunterized
- Example
- Available since

Adds OpenCL headers and ICD (Installable Client Driver) https://github.com/KhronosGroup/OpenCL-ICD-Loader

- Platforms: Windows VS12+/MSYS, Linux
- Version: currently OpenCL 2.1+

Usage

```
hunter_add_package(OpenCL)
find_package(OpenCL CONFIG REQUIRED)

target_link_libraries(... PRIVATE OpenCL::OpenCL)
```

Pitfalls

- Requirements for Ubuntu (see issue 853):

```
sudo apt-get install mesa-common-dev
```

4.1.88 OpenCL-Headers

- Official
- Example
- Added by Rahul Sheth (pr-534)

```
hunter_add_package(OpenCL-Headers)
find_package(OpenCLHeaders CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC OpenCL::Headers)
```

4.1.89 OpenCL-cpp

- Official
- Hunterized
- Example

Adds c++ wrappers for OpenCL, http://github.khronos.org/OpenCL-CLHPP/

- Platforms: Windows VS12+/MSYS, Linux (limited by OpenCL package)

Usage

```
hunter_add_package(OpenCL-cpp)
find_package(OpenCL-cpp CONFIG REQUIRED)

target_link_libraries(... PRIVATE OpenCL-cpp::OpenCL-cpp)
```

4.1.90 OpenCV

- Official
- Hunterized
- Example

```
hunter_add_package(OpenCV)
find_package(OpenCV REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PRIVATE ${OpenCV_LIBS})
```

Qt back-end

Qt back-end support for desktop can be enabled by creating a custom config with:

```
hunter_config(
    OpenCV
    VERSION ${HUNTER_OpenCV_VERSION}
    CMAKE_ARGS WITH_QT=YES
)
```

FFmpeg support

```
hunter_config(
    OpenCV
    VERSION ${HUNTER_OpenCV_VERSION}
    CMAKE_ARGS
    WITH_FFMPEG=ON
    OPENCV_FFMPEG_USE_FIND_PACKAGE=YES
)
```

Known issues

- since hunter v0.18.44 OpenCV can't find system libraries to link against on Linux. If you need those dependencies (FFMPEG, GTK, GStreamer, V4L2, etc.) you need to fork hunter and revert commit f6f0965 in your fork. Try this fork for some already patched releases.

4.1.91 OpenCV-Extra

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.92 OpenEXR

- Official
- Example
- Added by Harry Mallon (pr-164)

```
hunter_add_package(OpenEXR)
find_package(OpenEXR CONFIG REQUIRED)

add_executable(boo boo.cpp)

# For OpenEXR < 3.0
#  - see https://github.com/AcademySoftwareFoundation/Imath/blob/master/docs/
↪PortingGuide2-3.md
# target_link_libraries(boo PUBLIC OpenEXR::IlmImf)
target_link_libraries(boo PUBLIC OpenEXR::OpenEXR)
```

4.1.93 OpenGL-Registry

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-424)

```
hunter_add_package(OpenGL-Registry)
find_package(OpenGL-Registry CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC OpenGL::Registry-Headers)
```

4.1.94 OpenNMTTokenizer

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(OpenNMTTokenizer)
find_package(OpenNMTTokenizer CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo OpenNMTTokenizer::OpenNMTTokenizer)
```

4.1.95 OpenSSL

- Official
- Example

```
hunter_add_package(OpenSSL)
find_package(OpenSSL REQUIRED)
target_link_libraries(foo PUBLIC OpenSSL::SSL OpenSSL::Crypto)
```

Fixed/workaround

- space in path, related
- Can't be build with `make -jN`: https://github.com/ruslo/hunter/issues/87

Using ASM optimization on Windows

To be able to use ASM optimization on Windows, you need to set the `ASM_SUPPORT=ON` option.

4.1.96 OpenSceneGraph

- Official
- Hunterized
- Example
- Added by t0p4 (pr-1689)

```
hunter_add_package(OpenSceneGraph)
find_package(OpenSceneGraph CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC OpenSceneGraph::osg)
```

4.1.97 Opus

- Official
- Example
- Added by drodin (pr-245)

```
hunter_add_package(Opus)
find_package(Opus CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC Opus::opus)
```

4.1.98 PNG

- Official
- Hunterized
- Example

```
hunter_add_package(PNG)
find_package(PNG CONFIG REQUIRED)

add_executable(boo main.cpp)
target_link_libraries(boo PRIVATE PNG::png)
```

For compatibility with the FindPNG module, the `ALIAS` target `PNG::PNG` can be used too (requires CMake version >= 3.11!):

```
add_executable(baz main.cpp)
target_link_libraries(baz PRIVATE PNG::PNG)
```

4.1.99 PROJ4

- Official
- Example

```
hunter_add_package(PROJ4)
find_package(PROJ4 CONFIG REQUIRED)

add_executable(hello-proj4 hello-proj4.cpp)
target_link_libraries(hello-proj4 PUBLIC proj)
```

4.1.100 PhysUnits

- Official
- Example
- Added by Stefan Reinhold (pr-1503)

```
hunter_add_package(PhysUnits)
find_package(PhysUnits CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC PhysUnits::PhysUnits)
```

4.1.101 PocoCpp

POCO C++ Libraries

- Official
- Official Repository
- Hunterized Repository

```
hunter_add_package(PocoCpp)
find_package(Poco REQUIRED Foundation CONFIG)
target_link_libraries(... Poco::Foundation)
```

`find_package` command requires at least one of the following components, else CMake will fail.

Components are as follows:

- Foundation (`Poco::Foundation`)
- JSON (`Poco::JSON`)
- Net (`Poco::Net`)
- Util (`Poco::Util`)
- XML (`Poco::XML`)
- Zip (`Poco::Zip`)

Note: Components can be enabled or disabled by using following:

```
hunter_config(PocoCpp VERSION 1.10.0
  CMAKE_ARGS
    ENABLE_DATA=OFF
    ....
)
```

The map between `CMAKE_ARGS` and PocoCpp components can be found here.

4.1.102 PostgreSQL

- Official
- Hunterized
- Example

- Available since
- Added by Alexandre Pretyman (pr-301)

```
hunter_add_package(PostgreSQL)
find_package(PostgreSQL REQUIRED)
target_link_libraries(... PostgreSQL::libpq)
```

Ubuntu workarounds for missing dependencies (See pr-301):

```
sudo apt-get install libreadline-dev
sudo apt-get install bison
sudo apt-get install flex
```

4.1.103 Protobuf

- Official
- Hunterized
- Example
- Available since
- Added by Antal Tátrai (pr-340)

```
hunter_add_package(Protobuf)
find_package(Protobuf CONFIG REQUIRED)
target_link_libraries(... protobuf::libprotobuf)
```

Mixing toolchains

Example of mixing host and target toolchains in **one** CMake step (e.g. build `protoc` executable for OSX host and use it to build `libprotobuf` for Android target):

- https://github.com/forexample/protobuf-toolchains-mix

4.1.104 Qt

- Official
- Source archives

See also:

- Example: Qt Widgets
- iOS examples
- Android example

Usage

Qt is split into components. Each component installs its corresponding `Qt5*Config.cmake` and libraries.

Examples:

```
hunter_add_package(Qt)
# same as: hunter_add_package(Qt COMPONENTS qtbase)

find_package(Qt5Concurrent REQUIRED)
find_package(Qt5Core REQUIRED)
find_package(Qt5Gui REQUIRED)
find_package(Qt5Network REQUIRED)
find_package(Qt5OpenGL REQUIRED)
find_package(Qt5OpenGLExtensions REQUIRED)
find_package(Qt5PrintSupport REQUIRED)
find_package(Qt5Sql REQUIRED)
find_package(Qt5Test REQUIRED)
find_package(Qt5Widgets REQUIRED)
find_package(Qt5Xml REQUIRED)
find_package(Qt5DBus REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtsvg)
find_package(Qt5Svg REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtxmlpatterns)
find_package(Qt5XmlPatterns REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtlocation)
find_package(Qt5Positioning REQUIRED)
find_package(Qt5Location REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtdeclarative)
find_package(Qt5Qml REQUIRED)
find_package(Qt5Quick REQUIRED)
find_package(Qt5QuickTest REQUIRED)
find_package(Qt5QuickWidgets REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtmultimedia)
find_package(Qt5Multimedia REQUIRED)
find_package(Qt5MultimediaWidgets REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtsensors)
find_package(Qt5Sensors REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtwebsockets)
find_package(Qt5WebSockets REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtwebchannel)
find_package(Qt5WebChannel REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qttools)
find_package(Qt5Designer REQUIRED)
find_package(Qt5Help REQUIRED)
find_package(Qt5LinguistTools REQUIRED)
find_package(Qt5UiPlugin REQUIRED)
find_package(Qt5UiTools REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtscript)
find_package(Qt5Script REQUIRED)
```

```
find_package(Qt5ScriptTools REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtquick1)
find_package(Qt5Declarative REQUIRED)
```

```
hunter_add_package(Qt COMPONENTS qtimageformats)
find_package(Qt5Gui) # load plugins
# targets available:
#   * Qt5::QDDSPlugin
#   * Qt5::QICNSPlugin
#   * Qt5::QJp2Plugin
#   * Qt5::QMngPlugin
#   * Qt5::QTgaPlugin
#   * Qt5::QTiffPlugin
#   * Qt5::QWbmpPlugin
#   * Qt5::QWebpPlugin
```

```
hunter_add_package(Qt COMPONENTS qtquickcontrols)
# no *.cmake modules installed
```

Customization

- `QT_WITH_GSTREAMER`
  - Build with *gstreamer*
  - You will need this when building Qt application with camera support on Linux
  - Adds `-gstreamer 1.0`
  - Only configuration with shared libraries tested. Also you have to set runtime paths with `LD_LIBRARY_PATH`/`GST_PLUGIN_PATH`, see example.
  - To test GStreamer camera you can run `gst-launch -v -m camerabin`
- `QT_OPENGL_DESKTOP`
  - Use OpenGL installed on Windows
  - Visual Studio
  - Adds `-opengl desktop`
  - Qt Configure Options

Windows "path too long"

Using *HUNTER_BINARY_DIR* is not helping with *path too long* errors. The only way to build Qt is to use short path for `HUNTER_ROOT` directory.

See also:

- https://bugreports.qt.io/browse/QTBUG-66652
- https://bugreports.qt.io/browse/QTBUG-64298

Pitfalls

- Python is required to be in `PATH` if you're building the `qtdeclarative` component
- Conflicts with system Qt: bug with workaround
- iOS (Qt < 5.9): you must use `qtmn` instead of `main`:

```
#include <QtGlobal> // Q_OS_IOS

#if defined(Q_OS_IOS)
extern "C" int qtmn(int argc, char** argv) {
#else
int main(int argc, char **argv) {
#endif
```

you will see next error without this fix applied:

```
Error: You are creating QApplication before calling UIApplicationMain.
If you are writing a native iOS application, and only want to use Qt for
parts of the application, a good place to create QApplication is from
within 'applicationDidFinishLaunching' inside your UIApplication
delegate.
```

Stackoverflow

- Run-time error for Qt application on iOS built via CMake

- `QtQuick2Plugin` conflict.

  Both `plugins/qmltooling/libqmldbg_qtquick2.a` and `qml/QtQuick.2/libqtquick2plugin.a` implement this plugin:

```
[Install]> nm -gU plugins/qmltooling/libqmldbg_qtquick2.a | grep static_plugin
00000000000000b0 T __Z31qt_static_plugin_QtQuick2Pluginv
```

```
[Install]> nm -gU qml/QtQuick.2/libqtquick2plugin.a | grep static_plugin
0000000000000080 T __Z31qt_static_plugin_QtQuick2Pluginv
```

Linking of `libqmldbg_qtquick2.a` may lead to the next runtime error:

```
module "QtQuick" plugin "qtquick2plugin" not found
```

if you see this error try to remove usage of target `Qt5::QtQuick2Plugin` and variable `Qt5Qml_PLUGINS`.

- Static QML plugins loading issue and workaround: https://bugreports.qt.io/browse/QTBUG-35754
- iOS with `armv7s` architecture is broken: https://bugreports.qt.io/browse/QTBUG-48805
- **Errors when compiling on Linux Debian** without manually installing some Qt dependencies first. See Qt Issue 2. The Problem can be fixed by installing the necessary libraries before calling CMake with the command:

```
> sudo apt-get install libfontconfig1-dev libfreetype6-dev libx11-dev libxext-dev␣
↪libxfixes-dev libxi-dev libxrender-dev libxcb1-dev libx11-xcb-dev libxcb-glx0-
↪dev
```

- Requirements for Ubuntu for Hunter v0.14.14+ (need `GL,EGL`: `/usr/include/GL/gl.h`, `usr/include/EGL/egl.h`):

```
> sudo apt-get install libegl1-mesa-dev libgl1-mesa-dev libegl1-mesa-drivers
```

- Extra libraries for Android tools on Ubuntu needed (see this answer)

Hints

- Set `QT_DEBUG_PLUGINS=1` environment variable to obtain some diagnostic info: http://doc.qt.io/qt-5.5/deployment-plugins.html

4.1.105 QtAndroidCMake

- Official
- Hunterized
- For usage, see README at https://github.com/hunter-packages/qt-android-cmake
- HelloGL2 example

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.106 QtCMakeExtra

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.107 QtPropertyEditor

- Official
- Hunterized
- Example
- Added by t0p4 (pr-1670)

```
hunter_add_package(QtPropertyEditor)
find_package(QtPropertyEditor CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC QtPropertyEditor::PropertyEditor)
```

4.1.108 QtQmlManager

- Official
- Example

```
hunter_add_package(QtQmlManager)
list(APPEND CMAKE_MODULE_PATH "${QTQMLMANAGER_ROOT}/cmake")
include(QtCopyQmlTo)

QtCopyQmlTo(${qml_dir})
```

4.1.109 Qwt

- Official
- Hunterized
- Example
- Added by t0p4 (pr-1626)

```
hunter_add_package(Qwt)
find_package(Qwt CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Qwt::qwt)
```

4.1.110 RapidJSON

- http://rapidjson.org/
- GitHub official
- Hunterized
- Example

```
hunter_add_package(RapidJSON)
find_package(RapidJSON CONFIG REQUIRED)
target_link_libraries(... RapidJSON::rapidjson)
```

Macros

RapidJSON defines a few macros to configure the library. If different libraries use different settings, undefined behavior can occur. We set up the defines to be exported when set, and we set the `RAPIDJSON_HAS_STDSTRING=1` and `RAPIDJSON_NOMEMBERITERATORCLASS` by default. These can be overridden with a custom config

```
hunter_config(
    RapidJSON
    VERSION ${HUNTER_RapidJSON_VERSION}
    CMAKE_ARGS
      RAPIDJSON_HAS_STDSTRING=OFF
      RAPIDJSON_NOMEMBERITERATORCLASS=OFF
)
```

4.1.111 RapidXML

- Official
- Original Fork of Marcin Kalicinski's RapidXml library
- Hunterized
- Example

```
hunter_add_package(RapidXML)
find_package(RapidXML REQUIRED CONFIG)
target_link_libraries(foo RapidXML::RapidXML)
```

4.1.112 RedisClient

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(RedisClient)
find_package(RedisClient CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC RedisClient::RedisClient)
```

4.1.113 SDL2

- Official
- Hunterized
- Example
- Available since v0.14.29
- Added by Cyberunner23 (pr-451)

```
hunter_add_package(SDL2)
find_package(SDL2 CONFIG REQUIRED)
#...
target_link_libraries(... SDL2::SDL2)
```

Available targets: `SDL2::SDL2`, `SDL2::SDL2main`

4.1.114 SDL_image

- Official
- Hunterized
- Example
- Available since v0.19.86
- Added by wheybags (pr-989)

```
hunter_add_package(SDL_image)
find_package(SDL_image CONFIG REQUIRED)
#...
target_link_libraries(main
    SDL_image::SDL_image)
```

4.1.115 SDL_mixer

- Official
- Hunterized
- Example
- Available since v0.19.56
- Added by wheybags (pr-924)

```
hunter_add_package(SDL_mixer)
find_package(SDL_mixer CONFIG REQUIRED)
#...
target_link_libraries(foo SDL_mixer::SDL_mixer)
```

4.1.116 SDL_net

- Official
- Hunterized
- Example
- Added by drodin (pr-452)

```
hunter_add_package(SDL_net)
find_package(SDL_net CONFIG REQUIRED)
#...
target_link_libraries(foo SDL_net::SDL_net)
```

4.1.117 SDL_ttf

- Official
- Hunterized
- Example

- Added by Dennis Biber (pr-1251)

```
hunter_add_package(SDL_ttf)
find_package(SDL_ttf CONFIG REQUIRED)
target_link_libraries(... SDL_ttf::SDL_ttf)
```

4.1.118 SFML

- Official
- Hunterized
- Example
- Added by drodin (pr-N)

```
hunter_add_package(SFML)
find_package(SFML COMPONENTS graphics CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC sfml-graphics)
```

4.1.119 SPIRV-Headers

- Official
- Example
- Added by Mathieu-Andre Chiasson (pr-13)

```
hunter_add_package(SPIRV-Headers)
find_package(SPIRV-Headers CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC SPIRV-Headers::SPIRV-Headers)
```

4.1.120 SPIRV-Tools

- Official
- Hunterized
- Example
- Added by Mathieu-Andre Chiasson (pr-23)

```
hunter_add_package(SPIRV-Tools)
find_package(SPIRV-Tools CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC SPIRV-Tools::SPIRV-Tools)
```

Available targets: `SPIRV-Tools::SPIRV-Tools-opt`, `SPIRV-Tools::SPIRV-Tools-reduce`, `SPIRV-Tools::SPIRV-Tools-link`, `SPIRV-Tools::SPIRV-Tools`

4.1.121 SimpleSignal

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1685)

```
hunter_add_package(SimpleSignal)
find_package(SimpleSignal CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC SimpleSignal::SimpleSignal)
```

4.1.122 Snappy

- Official
- Example
- Available since v0.19.68
- Added by Paweł Bylica (pr-949)

```
hunter_add_package(sleef)
find_package(sleef CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main sleef::sleef)
```

4.1.123 Sober

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.124 Sources-for-Android-SDK

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.125 Sqlpp11

- Official
- Hunterized
- Example
- Added by xsacha (pr-1786)

```
hunter_add_package(Sqlpp11)
find_package(Sqlpp11 CONFIG REQUIRED)

add_executable(example example.cpp)
target_link_libraries(example sqlpp11)
```

4.1.126 SuiteSparse

- Official
- Hunterized
- Example
- Available since
- Added by Neroburner (pr-861)
- Dependencies:
- *LAPACK*

```
hunter_add_package(SuiteSparse)
find_package(SuiteSparse CONFIG REQUIRED)
target_link_libraries(foo SuiteSparse::cholmod)
```

The following targets are available:

- `SuiteSparse::suitesparseconfig`
- `SuiteSparse::amd`
- `SuiteSparse::btf`
- `SuiteSparse::camd`
- `SuiteSparse::ccolamd`

- `SuiteSparse::colamd`
- `SuiteSparse::cholmod`
- `SuiteSparse::cxsparse`
- `SuiteSparse::klu`
- `SuiteSparse::ldl`
- `SuiteSparse::umfpack`
- `SuiteSparse::spqr`

4.1.127 TCLAP

- Official
- Hunterized
- Example
- Added by cyberunner23 (pr-1419)

```
hunter_add_package(TCLAP)
find_package(TCLAP CONFIG REQUIRED)

add_executable(foo main.cpp)
target_link_libraries(foo TCLAP::TCLAP)
```

4.1.128 TIFF

- Official
- Hunterized
- Example

```
hunter_add_package(TIFF)
find_package(TIFF CONFIG REQUIRED)
target_link_libraries(... TIFF::libtiff)
```

4.1.129 Tesseract

- Official
- Hunterized
- Example
- Available since
- Added by Sacha Refshauge (pr-830)

```
hunter_add_package(Tesseract)
find_package(Tesseract CONFIG REQUIRED)

add_executable(example example.cpp)
target_link_libraries(example Tesseract::libtesseract)
```

4.1.130 Urho3D

- https://urho3d.github.io
- Official GitHub
- Hunterized
- Example
- Stand-alone example (includes configuration for iOS and Android)

You have to explicitly switch to these versions of dependencies:

```
# config.cmake

hunter_config(Lua VERSION 5.1.5-p3)
hunter_config(SDL2 VERSION 2.0.4-urho-p4)
```

Because Urho3D is using custom version of *SDL2* which is not fully compatible with upstream official API and *toluapp* is not working with default *Lua* version.

```
hunter_add_package(Urho3D)
find_package(Urho3D CONFIG REQUIRED)
target_link_libraries(boo PUBLIC Urho3D::Urho3D)
```

Customization

- Option `URHO3D_DATABASE_ODBC` is `OFF` by default. Package `ODBC` is not implemented in Hunter. `URHO3D_DATABASE_SQLITE=ON` will enable database support using *sqlite3* (this is a default).

4.1.131 Vulkan-Headers

- Official
- Example
- Added by Rahul Sheth (pr-67)

```
hunter_add_package(Vulkan-Headers)
find_package(VulkanHeaders CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Vulkan::Headers)
```

4.1.132 VulkanMemoryAllocator

- Official
- Hunterized
- Example
- Added by Jon Spencer (pr-1509)

To use this package the vulkan headers (not part of Hunter at the time of writing) must be installed. On debian variants use "apt-get install libvulkan-dev". On Mac, Windows, and iOS download and install the Vulkan SDK from here. Recent versions of the android NDK supports Vulkan out of the box.

```
hunter_add_package(VulkanMemoryAllocator)
find_package(VulkanMemoryAllocator CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC VulkanMemoryAllocator::VulkanMemoryAllocator)
```

4.1.133 WDC

- Official
- Hunterized
- Available since WDC v1.0.9

```
hunter_add_package(WDC)
find_package(WDC CONFIG REQUIRED)
target_link_libraries(... WDC::libwdc)
```

4.1.134 WTL

- Official
- Example
- Available since
- Added by Alexander Lamaison (pr-329)

```
hunter_add_package(WTL)
find_package(WTL CONFIG REQUIRED)
target_link_libraries(... WTL::WTL)
```

4.1.135 Washer

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.136 WebKit

- https://webkit.org/
- Unofficial mirror of the WebKit SVN repository
- Hunterized
- Example

```
hunter_add_package(WebKit)
find_package(WebKit CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(
    boo
    PUBLIC
    WebKit::JavaScriptCore
    WebKit::WTF
)
```

4.1.137 WebP

WebP codec: library to encode and decode images in WebP format. This package contains the library that can be used in other programs to add WebP support, as well as the command line tools `cwebp` and `dwebp`.

- Official
- Hunterized
- Example
- Added by Mathieu-Andre Chiasson (pr-1371)

```
hunter_add_package(WebP)
find_package(WebP CONFIG REQUIRED)
add_executable(main main.cpp)
target_link_libraries(main WebP::webp)
```

4.1.138 WinSparkle

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.139 YAJL

- Official
- Hunterized
- Example
- Added by Fredrik Appelros (pr-1837)

```
hunter_add_package(YAJL)
find_package(YAJL CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC YAJL::yajl)
```

4.1.140 ZLIB

- Official
- Hunterized
- Example

```
hunter_add_package(ZLIB)
find_package(ZLIB CONFIG REQUIRED)

add_executable(boo main.c)
target_link_libraries(boo PRIVATE ZLIB::zlib)
```

For compatibility with FindZLIB module, the `ALIAS` target `ZLIB::ZLIB` can be used too:

```
add_executable(baz main.c)
target_link_libraries(baz PRIVATE ZLIB::ZLIB)
```

4.1.141 ZMQPP

- Official
- Official GitHub
- Hunterized
- Example
- Available since
- Added by Antal Tátrai (pr-343)

```
# This will failed if C++11 is not enabled or not supported.
hunter_add_package(ZMQPP)

find_package(ZMQPP CONFIG REQUIRED)

target_link_libraries(... ZMQPP::zmqpp)
```

Note

This library requires C++11.

Bugs

- Currently can be used only on Linux

4.1.142 ZeroMQ

- Official
- Example
- Available since
- Added by Antal Tátrai (pr-334)
- Testing branch

```
hunter_add_package(ZeroMQ)

find_package(ZeroMQ CONFIG REQUIRED)

# or ZeroMQ::libzmq-static
target_link_libraries(... ZeroMQ::libzmq)
```

Bugs

- Tests does not work properly on Windows and OSX

4.1.143 Zug

- Official
- Example
- Added by Joerg-Christian Boehme (pr-107)

```
hunter_add_package(Boost)
find_package(Boost CONFIG REQUIRED)

hunter_add_package(Zug)
find_package(Zug CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC zug Boost::boost)
```

4.1.144 abseil

- Official
- Example

- Added by Rahul Sheth (pr-242)

```
hunter_add_package(abseil)
find_package(absl CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC absl::strings)
```

4.1.145 accelerate

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(accelerate REQUIRED)
target_link_libraries(... accelerate::accelerate)
```

Same as

```
target_link_libraries(... "-framework Accelerate")
```

- https://developer.apple.com/documentation/accelerate?language=objc

4.1.146 acf

- Official
- Example
- Added by David Hirvonen (pr-1176)

```
hunter_add_package(acf)
find_package(acf CONFIG REQUIRED)
target_link_libraries(acf acf::acf)
```

4.1.147 actionlib

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1931)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(actionlib)
find_package(catkin CONFIG REQUIRED COMPONENTS actionlib)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.148 aes

- Official
- Hunterized
- Example

```
hunter_add_package(aes)
find_package(aes CONFIG REQUIRED)
target_link_libraries(... aes::aes)
```

4.1.149 aglet

- Official
- Example

```
hunter_add_package(aglet)
find_package(aglet CONFIG REQUIRED)
target_link_libraries(... aglet::aglet)
```

4.1.150 android

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(android REQUIRED)
target_link_libraries(... android::android)
```

4.1.151 android_arm64_v8a_system_image_packer

- Official
- Example

```
hunter_add_package(android_arm64_v8a_system_image_packer)
```

4.1.152 android_arm_eabi_v7a_system_image_packer

- Official
- Example

```
hunter_add_package(android_arm_eabi_v7a_system_image_packer)
```

4.1.153 android_build_tools_packer

- Official
- Example

```
hunter_add_package(android_build_tools_packer)
```

4.1.154 android_google_apis_intel_x86_atom_system_image_packer

- Official
- Example

```
hunter_add_package(android_google_apis_intel_x86_atom_system_image_packer)
```

4.1.155 android_google_apis_packer

- Official
- Example

```
hunter_add_package(android_google_apis_packer)
```

4.1.156 android_google_repository_packer

- Official
- Example

```
hunter_add_package(android_google_repository_packer)
```

4.1.157 android_intel_x86_atom_system_image_packer

- Official
- Example

```
hunter_add_package(android_intel_x86_atom_system_image_packer)
```

4.1.158 android_log

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(android_log REQUIRED)
target_link_libraries(... android_log::android_log)
```

4.1.159 android_mips_system_image_packer

- Official
- Example

```
hunter_add_package(android_mips_system_image_packer)
```

4.1.160 android_sdk_packer

- Official
- Example

```
hunter_add_package(android_sdk_packer)
find_package(android_sdk_packer CONFIG REQUIRED)
```

4.1.161 android_sdk_platform_packer

- Official
- Example

```
hunter_add_package(android_sdk_platform_packer)
```

4.1.162 android_sdk_platform_tools_packer

- Official
- Example

```
hunter_add_package(android_sdk_platform_tools_packer)
```

4.1.163 android_sdk_tools_packer

- Official
- Example

```
hunter_add_package(android_sdk_tools_packer)
```

4.1.164 android_support_repository_packer

- Official
- Example

```
hunter_add_package(android_support_repository_packer)
```

4.1.165 angles

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1928)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(angles)
find_package(catkin CONFIG REQUIRED COMPONENTS angles)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.166 apg

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-268)

```
hunter_add_package(apg)
find_package(apg CONFIG REQUIRED)

add_executable(boo boo.cpp)
#Available:
# apg::apg
# apg::unicode
# apg::bmp
# apg::wav
# apg::pixfont
# apg::maths
# apg::interp
# apg::tga
# apg::console

target_link_libraries(boo PUBLIC apg::console apg::bmp)
```

4.1.167 appkit

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(appkit REQUIRED)
target_link_libraries(... appkit::appkit)
```

Same as

```
target_link_libraries(... "-framework AppKit")
```

- https://developer.apple.com/documentation/appkit?language=objc

4.1.168 applicationservices

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(applicationservices REQUIRED)
target_link_libraries(... applicationservices::applicationservices)
```

Same as

```
target_link_libraries(... "-framework ApplicationServices")
```

- https://developer.apple.com/documentation/applicationservices?language=objc

4.1.169 arabica

- Official
- Hunterized
- Example
- Added by Fredrik Appelros (pr-1838)

```
hunter_add_package(arabica)
find_package(arabica CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC arabica::arabica)
```

4.1.170 asio

- Official
- Hunterized
- Example

```
hunter_add_package(asio)
find_package(asio CONFIG REQUIRED)
target_link_libraries(... asio::asio_static)
```

asio::asio_shared and *asio::asio_headeronly* targets are also available.

4.1.171 asio-grpc

- Official
- Example
- Added by Tradias (pr-554)

```
hunter_add_package(asio-grpc)
find_package(asio-grpc CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC asio-grpc::asio-grpc)
```

CMake options

The `CMAKE_ARGS` feature (see customization) can be used to customize `asio-grpc`:

- To use standalone `Asio` instead of `Boost.Asio`:

```
hunter_config(
    asio-grpc
    VERSION ${HUNTER_asio-grpc_VERSION}
    CMAKE_ARGS
      ASIO_GRPC_HUNTER_BACKEND_BOOST_ASIO=OFF
      ASIO_GRPC_HUNTER_BACKEND_STANDALONE_ASIO=ON
)
```

- To use Boost.Container instead of `<memory_resource>`:

```
hunter_config(
    asio-grpc
    VERSION ${HUNTER_asio-grpc_VERSION}
    CMAKE_ARGS
      ASIO_GRPC_USE_BOOST_CONTAINER=ON
)
```

For more options see asio-grpc repository.

4.1.172 assetslibrary

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(assetslibrary REQUIRED)
target_link_libraries(... assetslibrary::assetslibrary)
```

Same as

```
target_link_libraries(... "-framework AssetsLibrary")
```

- https://developer.apple.com/documentation/assetslibrary?language=objc

4.1.173 astc-encoder

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-286)

```
hunter_add_package(astc-encoder)

# find_package(astc-encoder CONFIG REQUIRED) # for v1.0-2.0
find_package(astcencoder CONFIG REQUIRED) # for v3.0+

add_executable(boo boo.cpp)
# target_link_libraries(boo PUBLIC astc-encoder::astcenc) # for v1.0-v2.0
target_link_libraries(boo PUBLIC astcencoder::astcenc-static) # for v3.0+
```

4.1.174 audiotoolbox

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(audiotoolbox REQUIRED)
target_link_libraries(... audiotoolbox::audiotoolbox)
```

Same as

```
target_link_libraries(... "-framework AudioToolbox")
```

- https://developer.apple.com/documentation/audiotoolbox?language=objc

4.1.175 audiounit

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(audiounit REQUIRED)
target_link_libraries(... audiounit::audiounit)
```

Same as

```
target_link_libraries(... "-framework AudioUnit")
```

- https://developer.apple.com/documentation/audiounit?language=objc

4.1.176 autobahn-cpp

- Official Autobahn

- Official Autobahn-cpp
- Hunterized
- Examples

This is a C++11 library. On Windows only VS14 is supported.

```
set (CMAKE_CXX_STANDARD11)
hunter_add_package(autobahn-cpp)
find_package(autobahn-cpp CONFIG REQUIRED)
target_link_libraries(... autobahn-cpp::autobahn-cpp)
```

4.1.177 autoutils

- Project
- Example
- Added by isaachier (pr-1273)

```
# download autoutils
hunter_add_package(autoutils)
find_package(autoutils CONFIG REQUIRED)

# include modules
include(AutoutilsCheckHeader)

autoutils_check_header("stdio.h")
if(NOT HAVE_STDIO_H)
  message(FATAL_ERROR "Cannot find stdio.h")
endif()
```

4.1.178 avfoundation

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(avfoundation REQUIRED)
target_link_libraries(... avfoundation::avfoundation)
```

Same as

```
target_link_libraries(... "-framework AVFoundation")
```

- https://developer.apple.com/documentation/avfoundation?language=objc

4.1.179 aws-c-common

- Official
- Hunterized
- Example

- Added by insufficientchocolate (pr-1694)

```
hunter_add_package(aws-c-common)

find_package(aws-c-common CONFIG REQUIRED)

# To use exported modules
get_filename_component(AWS_CMAKE_MODULE_PATH "${aws-c-common_DIR}/../../cmake"␣
↪ABSOLUTE)
list(APPEND CMAKE_MODULE_PATH "${AWS_CMAKE_MODULE_PATH}")
include(AwsSIMD)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC AWS::aws-c-common)
```

4.1.180 aws-sdk-cpp

- Official
- Hunterized
- Example
- Added by Harry Mallon (pr-474)

```
# You need to set which parts of the SDK you want to build (default is s3)
#
# hunter_config(aws-sdk-cpp
#   VERSION 1.9.94
#   CMAKE_ARGS
#     BUILD_ONLY=s3
# )
#

hunter_add_package(aws-sdk-cpp)
find_package(AWSSDK REQUIRED COMPONENTS s3)

if (NOT WIN32)
  find_package(CURL CONFIG REQUIRED)
endif()

add_executable(boo boo.cpp)
message("${AWSSDK_LINK_LIBRARIES}")
target_link_libraries(boo PUBLIC ${AWSSDK_LINK_LIBRARIES})
```

4.1.181 aws_lambda_cpp

- Official
- Hunterized
- Example
- Added by https://github.com/kevinkjt2000 (pr-429)

```
hunter_add_package(aws_lambda_cpp)
find_package(aws-lambda-runtime CONFIG REQUIRED)
```

```
add_executable(${PROJECT_NAME} ./handler.cpp)
target_link_libraries(${PROJECT_NAME} PRIVATE AWS::aws-lambda-runtime)
aws_lambda_package_target(${PROJECT_NAME})
```

4.1.182 basis_universal

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-273)

```
hunter_add_package(basis_universal)
find_package(basis_universal CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC basis_universal::basisu_encoder)
```

4.1.183 benchmark

- Official
- Example
- Added by Isaac Hier (pr-1088)

```
hunter_add_package(benchmark)
find_package(benchmark CONFIG REQUIRED)
target_link_libraries(... benchmark::benchmark)
```

4.1.184 bento4

- Official
- Hunterized
- Example
- Added by Brad Kotsopoulos (pr-1797)

```
hunter_add_package(bento4)
find_package(bento4 CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC bento4::ap4)
```

4.1.185 Binaryen

- Official
- Hunterized
- Example

- Added by Warchant (pr-1751)

```
hunter_add_package(binaryen)
find_package(binaryen CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC binaryen::binaryen)
```

4.1.186 bison

- Official
- Example
- Added by Isaac Hier (pr-1036)

```
hunter_add_package(bison)
find_package(BISON REQUIRED)
BISON_TARGET(MyParser parser.y ${CMAKE_CURRENT_BINARY_DIR}/parser.cpp)
add_executable(bison main.cpp ${BISON_MyParser_OUTPUTS})
```

4.1.187 boost-pba

- Hunterized
- Example

```
hunter_add_package(boost-pba)
find_package(boost-pba CONFIG REQUIRED)
target_link_libraries(... boost-pba::boost-pba)
```

4.1.188 botan

- Official
- Example
- Added by Jörg-Christian Böhme (pr-1922)

```
hunter_add_package(botan)
find_package(botan-2 CONFIG REQUIRED)

add_executable(bte boo.cpp)
target_link_libraries(bte PUBLIC PkgConfig::botan-2)
```

4.1.189 breakpad

- Official
- Hunterized
- Example
- Added by t0p4 (pr-1631)

```
hunter_add_package(breakpad)
find_package(breakpad CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC breakpad::libbreakpad)
```

4.1.190 bullet

- http://bulletphysics.org/wordpress/
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(bullet)
find_package(bullet CONFIG REQUIRED)
target_link_libraries(
    boo
    PUBLIC
    bullet::Bullet3Common
    bullet::BulletCollision
    bullet::BulletDynamics
    bullet::BulletInverseDynamics
    bullet::BulletSoftBody
    bullet::LinearMath
)
```

4.1.191 byte-lite

- Official
- Example

```
hunter_add_package(byte-lite)
find_package(byte-lite CONFIG REQUIRED)
target_link_libraries(... nonstd::byte-lite)
```

4.1.192 c-ares

- Official
- Example
- Added by Isaac Hier (pr-1087)

```
hunter_add_package(c-ares)
find_package(c-ares CONFIG REQUIRED)
target_link_libraries(... c-ares::cares)
```

4.1.193 caffe

- Official
- Hunterized
- Available since
- Example

```
hunter_add_package(caffe)
find_package(Caffe CONFIG REQUIRED)
target_link_libraries(... caffe)
```

Notes

- Works only on Linux with minimal set of dependencies (e.g. no CUDA)
- Android port: https://github.com/sh1r0/caffe-android-lib

4.1.194 carbon

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(carbon REQUIRED)
target_link_libraries(... carbon::carbon)
```

Same as

```
target_link_libraries(... "-framework Carbon")
```

4.1.195 catkin

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1407)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(catkin)
find_package(catkin CONFIG REQUIRED)

catkin_package()
```

4.1.196 cctz

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1370)

```
hunter_add_package(cctz)
find_package(cctz CONFIG REQUIRED)
add_executable(main main.cpp)
target_link_libraries(main cctz::cctz)
```

4.1.197 ccv

- Official
- Hunterized
- Example

```
hunter_add_package(ccv)
find_package(ccv REQUIRED CONFIG)

add_executable(foo test.c)
target_link_libraries(foo PRIVATE ccv::ccv)
```

4.1.198 cereal

- Official
- Hunterized

```
hunter_add_package(cereal)
find_package(cereal CONFIG REQUIRED)
target_link_libraries(... cereal::cereal)
```

4.1.199 ceres-solver

- Official
- Hunterized
- Examples:
- Basic
- with SuiteSparse
- with SuiteSparse dyn LAPACK
- Available since v0.18.30
- Added by NeroBurner (pr-648)
- with SuiteSparse support

- Available since v0.19.40
- Added by NeroBurner (pr-898)
- Maintainer: https://github.com/NeroBurner

Ceres Solver is an open source C++ library for modeling and solving large, complicated optimization problems. It can be used to solve Non-linear Least Squares problems with bounds constraints and general unconstrained optimization problems. It is a mature, feature rich, and performant library that has been used in production at Google since 2010. For more, see official website.

Usage

```
hunter_add_package(ceres-solver)
find_package(Ceres CONFIG REQUIRED)

target_link_libraries(... PRIVATE ceres)
```

with SuiteSparse

To get `ceres-solver` with `SuiteSparse` and static `LAPACK` add a local `cmake/Hunter/config.cmake` file with the following contents:

```
hunter_config(ceres-solver
  VERSION ${HUNTER_ceres-solver_VERSION} CMAKE_ARGS
    LAPACK=ON
    SUITESPARSE=ON
    CXSPARSE=ON # since 1.14.0-p1
)
```

Don't forget to add `enable_language(Fortran)` in your projects `CMakeLists.txt`.

with SuiteSparse and dynamic LAPACK

To get `ceres-solver` with `SuiteSparse` and dynamic `LAPACK` add a local `cmake/Hunter/config.cmake` file with the following contents:

```
hunter_config(ceres-solver
  VERSION ${HUNTER_ceres-solver_VERSION} CMAKE_ARGS
    LAPACK=ON
    SUITESPARSE=ON
    CXSPARSE=ON # since 1.14.0-p1
)
hunter_config(LAPACK
  VERSION ${HUNTER_LAPACK_VERSION}
  CMAKE_ARGS BUILD_SHARED_LIBS=ON
)
```

With a dynamic `LAPACK` library the `enable_language(Fortran)` is not needed. But when shipping your project one must also ship the shared `LAPACK` library.

with OpenBLAS as alternative to LAPACK

Since `v0.3.21` `OpenBLAS` provides a f2c-converted copy of `LAPACK` `v3.9.0`. This copy is used when building without a Fortran compiler. Using this in `ceres-solver` and `SuiteSparse` enables us to build a pure C++ library. Which means the resulting library can be static with no Fortran runtime dependencies.

Since Hunter v0.24.9 `SuiteSparse` per default is built against `OpenBLAS`, which in Hunter per default compiles without Fortran and with `LAPACK` enabled.

```
hunter_config(ceres-solver
  VERSION ${HUNTER_ceres-solver_VERSION} CMAKE_ARGS
    LAPACK=ON
    WITH_OPENBLAS=ON # since 2.1.0-p0
    SUITESPARSE=ON
    CXSPARSE=ON # since 1.14.0-p1
)
```

4.1.200 cgltf

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-275)

```
hunter_add_package(cgltf)
find_package(cgltf CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC cgltf::cgltf)
```

4.1.201 check_ci_tag

- Official GitHub
- Example

```
project(foo VERSION 1.0.0)

hunter_add_package(check_ci_tag)
find_package(check_ci_tag CONFIG REQUIRED)

# Read environment variables like TRAVIS_TAG/APPVEYOR_REPO_TAG_NAME
# and check they match PROJECT_VERSION
check_ci_tag()
```

4.1.202 chromium_zlib

- https://chromium.googlesource.com/chromium/src/third_party/+/master/zlib
- https://github.com/hunter-packages/chromium_zlib
- Example

```
hunter_add_package(chromium_zlib)
find_package(ZLIB CONFIG REQUIRED)

add_executable(boo main.c)
target_link_libraries(boo PUBLIC ZLIB::ZLIB)
```

4.1.203 civetweb

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(civetweb)
find_package(civetweb CONFIG REQUIRED)
target_link_libraries(boo PUBLIC civetweb::c-library)
```

4.1.204 clBLAS

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.205 class_loader

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1899)**
  - Contribution partially as part of work at SeeByte Ltd.

By default `BUILD_SHARED_LIBS=ON` used for `class_loader` because if linking against `class_loader` statically, plugins would not be loaded correctly at runtime, because both the plugin and the library loading it would use their own copy of `class_loader`.

```
hunter_add_package(class_loader)
find_package(catkin CONFIG REQUIRED COMPONENTS class_loader)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.206 cmcstl2

- Official
- Example
- Added by dvirtz (pr-1801)

```
hunter_add_package(cmcstl2)
find_package(cmcstl2 CONFIG REQUIRED)

add_executable(simple simple.cpp)
target_link_libraries(simple stl2)
```

4.1.207 complex_bessel

- Official
- Example
- Added by craffael (pr-472)

```
hunter_add_package(complex_bessel)
find_package(complex_bessel CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC complex_bessel::complex_bessel)
```

4.1.208 convertutf

- Hunterized
- Example

```
hunter_add_package(convertutf)
find_package(convertutf CONFIG REQUIRED)
target_link_libraries(... convertutf::convertutf)
```

4.1.209 coreaudio

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(coreaudio REQUIRED)
target_link_libraries(... coreaudio::coreaudio)
```

Same as

```
target_link_libraries(... "-framework CoreAudio")
```

- https://developer.apple.com/documentation/coreaudio?language=objc

4.1.210 coredata

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(coredata REQUIRED)
target_link_libraries(... coredata::coredata)
```

Same as

```
target_link_libraries(... "-framework CoreData")
```

- https://developer.apple.com/documentation/coredata?language=objc

4.1.211 corefoundation

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(corefoundation REQUIRED)
target_link_libraries(... corefoundation::corefoundation)
```

Same as

```
target_link_libraries(... "-framework CoreFoundation")
```

- https://developer.apple.com/documentation/corefoundation?language=objc

4.1.212 coregraphics

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(coregraphics REQUIRED)
target_link_libraries(... coregraphics::coregraphics)
```

Same as

```
target_link_libraries(... "-framework CoreGraphics")
```

- https://developer.apple.com/documentation/coregraphics?language=objc

4.1.213 corelocation

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(corelocation REQUIRED)
target_link_libraries(... corelocation::corelocation)
```

Same as

```
target_link_libraries(... "-framework CoreLocation")
```

- https://developer.apple.com/documentation/corelocation?language=objc

4.1.214 coremedia

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(coremedia REQUIRED)
target_link_libraries(... coremedia::coremedia)
```

Same as

```
target_link_libraries(... "-framework CoreMedia")
```

- https://developer.apple.com/documentation/coremedia?language=objc

4.1.215 coremotion

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(coremotion REQUIRED)
target_link_libraries(... coremotion::coremotion)
```

Same as

```
target_link_libraries(... "-framework CoreMotion")
```

- https://developer.apple.com/documentation/coremotion?language=objc

4.1.216 coretext

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(coretext REQUIRED)
target_link_libraries(... coretext::coretext)
```

Same as

```
target_link_libraries(... "-framework CoreText")
```

- https://developer.apple.com/documentation/coretext?language=objc

4.1.217 corevideo

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(corevideo REQUIRED)
target_link_libraries(... corevideo::corevideo)
```

Same as

```
target_link_libraries(... "-framework CoreVideo")
```

- https://developer.apple.com/documentation/corevideo?language=objc

4.1.218 corrade

- Official
- Example
- Developed by Vladimír Vondruš
- Added by Pascal Thomet (pr-1646)

usage

```
set(components Containers Interconnect PluginManager TestSuite Utility)

foreach(comp ${components})
  list(APPEND components_with_prefix Corrade::${comp})
endforeach()

hunter_add_package(corrade)
find_package(Corrade CONFIG REQUIRED COMPONENTS ${components})

add_executable(foo foo.cpp)
target_link_libraries(foo PRIVATE ${components_with_prefix})
```

About

corrade is a C++11/C++14 multiplatform utility library.

Known issues

Cross compilation to iOS and Android might fail since the build require to find `corrade-rc` (native executable) in your path. In order to build an iOS or Android package, first compile `corrade-rc` natively, and add it to your path.

4.1.219 cpp-statsd-client

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-291)

```
hunter_add_package(cpp-statsd-client)
find_package(cpp-statsd-client CONFIG REQUIRED)

add_executable(boo boo.cpp)
#For mingw/msys
if(WIN32)
  target_compile_definitions(boo PRIVATE _WIN32_WINNT=0x601)
endif()
target_link_libraries(boo PUBLIC cpp-statsd-client::cpp-statsd-client)
```

4.1.220 cpp_redis

- Official
- Official github fork
- Hunterized
- Example
- Available since

```
hunter_add_package(cpp_redis)
find_package(cpp_redis CONFIG REQUIRED)
target_link_libraries(... cpp_redis::cpp_redis)
```

4.1.221 cppast

- Official
- *Hunterized <https://github.com/cpp-pm/cppast>*
- Example
- Added by Joerg-Christian Boehme (pr-110)

```
hunter_add_package(cppast)
find_package(cppast CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC cppast::cppast)
```

4.1.222 cppcodec

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1676)

```
hunter_add_package(cppcodec)
find_package(cppcodec CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC cppcodec::cppcodec)
```

4.1.223 cppfs

- Official
- Example
- Added by Joerg-Christian Boehme (pr-92)

```
hunter_add_package(cppfs)
find_package(cppfs CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PRIVATE cppfs::cppfs)
```

4.1.224 cpr

- Official
- Hunterized
- Example
- Added by dvirtz (pr-1212)

```
hunter_add_package(cpr)
find_package(cpr CONFIG REQUIRED)

add_executable(cpr_example ...)
target_link_libraries(cpr_example cpr::cpr)
```

4.1.225 cpuinfo

- Official
- Hunterized
- Example
- Added by xsacha (pr-1789)

```
hunter_add_package(cpuinfo)
find_package(cpuinfo CONFIG REQUIRED)

add_executable(test example.cpp)
target_link_libraries(test cpuinfo::cpuinfo)
```

4.1.226 crashpad

- Official
- Hunterized
- Example
- Available since

  ```
  hunter_add_package(crashpad)
  find_package(crashpad CONFIG REQUIRED)
  target_link_libraries(... crashpad::crashpad_client)
  ```

Use this code in case you want to copy `crashpad_handler` to the directory with `foo` executable:

```
add_custom_command(
    TARGET foo
    PRE_BUILD
    COMMAND
    "${CMAKE_COMMAND}" -E copy
    "$<TARGET_FILE:crashpad::crashpad_handler>"
    "$<TARGET_FILE_DIR:foo>"
)
```

4.1.227 crashup

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.228 crc32c

- Official
- Example
- Added by Isaac Hier (pr-1243)

```
hunter_add_package(crc32c)
find_package(Crc32c CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main Crc32c::crc32c)
```

4.1.229 cryptopp

- Official
- Official GitHub
- Hunterized
- Example
- Added by Paweł Bylica (pr-1041)

```
hunter_add_package(cryptopp)
find_package(cryptopp CONFIG REQUIRED)

add_executable(cryptopp-test main.cpp)
target_link_libraries(cryptopp-test PRIVATE cryptopp-static)
```

4.1.230 ctti

- Official
- Hunterized
- Example
- Added by Casey (pr-1518)

Compile Time Type Information for the C++ programming language.

- Compilers: VS15, Clang >= 3.6.2
- Does not support GCC with optimizations (https://github.com/Manu343726/ctti/issues/19)
- Does not support VS17, bug in compiler

```
hunter_add_package(ctti)
find_package(ctti CONFIG REQUIRED)
```

```
add_executable(ctti_test main.cpp)
target_link_libraries(ctti_test ctti::ctti)
```

4.1.231 cub

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1162)

```
hunter_add_package(cub)
find_package(cub CONFIG REQUIRED)
target_link_libraries(foo cub::cub)
```

4.1.232 cvmatio

- Official
- Hunterized

```
hunter_add_package(cvmatio)
find_package(cvmatio CONFIG REQUIRED)
target_link_libraries(... cvmatio::cvmatio)
```

4.1.233 cvsteer

- Official
- Example

```
hunter_add_package(cvsteer)
find_package(cvsteer CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC cvsteer::cvsteer)
```

4.1.234 cxxopts

- Official
- Example

```
hunter_add_package(cxxopts)
find_package(cxxopts CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC cxxopts::cxxopts)
```

4.1.235 czmq

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.236 damageproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.237 date

- Official
- Example

```
hunter_add_package(date)
find_package(date CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC date::date)
```

The target to link against changed upstream across versions:

- 2.4.1 and before, it is *date_interface*.
- 2.4.1-e12095f and after, it is *date::date*.

4.1.238 dbus

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.239 debug_assert

- Official
- Example
- Added by dvirtz (pr-1127)

```
hunter_add_package(debug_assert)
find_package(debug_assert CONFIG REQUIRED)
target_link_libraries(debug_assert_example debug_assert)
```

4.1.240 dest

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(dest)
find_package(dest CONFIG REQUIRED)
target_link_libraries(... dest::dest)
```

4.1.241 dfdutils

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-416)

```
hunter_add_package(dfdutils)
find_package(dfdutils CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC KhronosGroup::dfdutils)
```

4.1.242 dlib

- Official
- Official GitHub
- Hunterized
- Example
- Available since

```
hunter_add_package(dlib)
find_package(dlib CONFIG REQUIRED)

add_executable(bayes_net_ex bayes_net_ex.cpp)
target_link_libraries(bayes_net_ex PUBLIC dlib::dlib)
```

4.1.243 dlpack

- Official GitHub
- Example

```
hunter_add_package(dlpack)
find_package(dlpack CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC dlpack::dlpack)
```

4.1.244 dmlc-core

- Official
- Hunterized
- Example

```
hunter_add_package(dmlc-core)
find_package(dmlc CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC dmlc::dmlc)
```

4.1.245 doctest

- Official

- Available since

```
hunter_add_package(doctest)

find_package(doctest CONFIG REQUIRED)

add_executable(doctest_test ${SOURCES} ${HEADERS})
target_link_libraries(doctest_test PUBLIC doctest::doctest)
```

4.1.246 double-conversion

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1017)

```
hunter_add_package(double-conversion)
find_package(double-conversion CONFIG REQUIRED)
target_link_libraries(... double-conversion::double-conversion)
```

4.1.247 draco

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1688)

```
hunter_add_package(draco)
find_package(draco CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC draco::draco)
```

4.1.248 dri2proto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.249 dri3proto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.250 drishti

- Official
- Example

```
hunter_add_package(drishti)
find_package(drishti CONFIG REQUIRED)

add_executable(drishti foo.cpp)
target_link_libraries(drishti PUBLIC drishti::drishti)
```

Next custom dependencies should be set in local `config.cmake` file explicitly:

```
hunter_config(
    xgboost
    VERSION 0.40-p10
    CMAKE_ARGS XGBOOST_USE_HALF=ON XGBOOST_USE_CEREAL=ON XGBOOST_DO_LEAN=ON
)

hunter_config(
    acf
    VERSION ${HUNTER_acf_VERSION}
    CMAKE_ARGS
    ACF_BUILD_OGLES_GPGPU=ON
)

if(ANDROID)
  # https://travis-ci.org/ingenue/hunter/jobs/287844545
  # Will be fixed in Android NDK 17
  set(drishti_dlib_version 19.2-p2)
  # error: 'struct lconv' has no member named 'decimal_point' -/-
```

```
  hunter_config(nlohmann_json VERSION 2.1.1-p1)
else()
  set(drishti_dlib_version 19.6-p2)
endif()

hunter_config(
    dlib
    VERSION ${drishti_dlib_version}
    CMAKE_ARGS
    DLIB_USE_BLAS=OFF
    DLIB_USE_LAPACK=OFF
    DLIB_USE_MKL_FFT=OFF
)
```

4.1.251 drishti_assets

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.252 drishti_faces

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.253 drm

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.254 duktape

- http://duktape.org/
- Hunterized
- Example

```
hunter_add_package(duktape)
find_package(duktape CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC duktape::duktape)
```

4.1.255 dynalo

- Official
- Example
- Added by Yassine Maddouri (pr-1350)

```
hunter_add_package(dynalo)
find_package(dynalo CONFIG REQUIRED)

add_executable(dynalo-example-loader dynalo-example-loader.cpp)
target_link_libraries(dynalo-example-loader dynalo)
```

4.1.256 egl

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(egl REQUIRED)
target_link_libraries(... egl::egl)
```

4.1.257 eigen3-nnls

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.258 enet

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.259 entityx

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1697)

```
hunter_add_package(entityx)
find_package(entityx CONFIG REQUIRED)
```

```
add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC entityx::entityx)
```

4.1.260 eos

- Official GitHub
- Hunterized
- Example
- Available since

```
hunter_add_package(eos)
find_package(eos CONFIG REQUIRED)
target_link_libraries(... eos::eos)
```

4.1.261 etc2comp

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-284)

```
hunter_add_package(etc2comp)
find_package(etc2comp CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC etc2comp::EtcLib)
```

4.1.262 ethash

- Official
- Example
- Added by Paweł Bylica (pr-1430)

```
hunter_add_package(ethash)
find_package(ethash CONFIG REQUIRED)

add_executable(use_ethash main.cpp)
target_link_libraries(use_ethash ethash::ethash)
```

4.1.263 eventpp

- Official
- Example
- Added by bazfp

```
hunter_add_package(eventpp)
find_package(eventpp CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main eventpp::eventpp)
include_directories(${EVENTPP_INCLUDE_DIR})
```

4.1.264 farmhash

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1150)

```
hunter_add_package(farmhash)
find_package(farmhash CONFIG REQUIRED)
target_link_libraries(farmhash farmhash::farmhash)
```

4.1.265 fast_obj

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-274)

```
hunter_add_package(fast_obj)
find_package(fast_obj CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC fast_obj::fast_obj)
```

4.1.266 ffmpeg

- https://ffmpeg.org/
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(ffmpeg)
find_package(ffmpeg CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(
    boo
    PUBLIC
    ffmpeg::avcodec
    ffmpeg::avformat
```

```
    ffmpeg::avutil
    ffmpeg::swresample
    ffmpeg::swscale
)
```

4.1.267 fft2d

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1161)

```
hunter_add_package(fft2d)
find_package(fft2d CONFIG REQUIRED)
target_link_libraries(fft2d fft2d::fft2d)
```

4.1.268 filament

- https://google.github.io/filament/Filament.html
- Official GitHub repo
- Hunterized
- Example

```
hunter_add_package(filament)
find_package(filament CONFIG REQUIRED)
target_link_libraries(... filament::filament)
```

4.1.269 fixesproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.270 flatbuffers

- Official GitHub
- Example

```
hunter_add_package(flatbuffers)
find_package(Flatbuffers CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC flatbuffers::flatbuffers)
```

Old version

When using flatbuffers version 1.3.0-p3, `find_package` argument must be in lowercase `flatbuffers`:

```
hunter_add_package(flatbuffers)
find_package(flatbuffers CONFIG REQUIRED)
target_link_libraries(... flatbuffers::flatbuffers)
```

- Hunterized

4.1.271 flex

- Official
- Example
- Added by Isaac Hier (pr-1039)

Simple flex example (no bison).

```
hunter_add_package(flex)
find_package(FLEX REQUIRED)

FLEX_TARGET(MyScanner numbers.lex ${CMAKE_CURRENT_BINARY_DIR}/numbers.cpp)
add_executable(main ${FLEX_MyScanner_OUTPUTS})
target_include_directories(main PUBLIC ${FLEX_INCLUDE_DIRS})
target_link_libraries(main ${FLEX_LIBRARIES})
```

More complex example involving flex and bison. Based on FindFLEX.

```
find_package(BISON REQUIRED)
hunter_add_package(flex)
find_package(FLEX REQUIRED)
BISON_TARGET(MyParser parser.y ${CMAKE_CURRENT_BINARY_DIR}/parser.cpp)
FLEX_TARGET(MyScanner lexer.l  ${CMAKE_CURRENT_BINARY_DIR}/lexer.cpp)
ADD_FLEX_BISON_DEPENDENCY(MyScanner MyParser)  # MyParser defines tokens for MyScanner

add_executable(main main.cpp ${BISON_MyParser_OUTPUTS} ${FLEX_MyScanner_OUTPUTS})
target_include_directories(main
    $<BUILD_INTERFACE:${CMAKE_CURRENT_BINARY_DIR}>
    ${BISON_INCLUDE_DIRS}
    ${FLEX_INCLUDE_DIRS})
target_link_libraries(main ${BISON_LIBRARIES} ${FLEX_LIBRARIES})
```

4.1.272 fmt

- Official
- Example
- Available since
- Added by Dmitry Panteleev (pr-413)
- Testing branch

```
hunter_add_package(fmt)

find_package(fmt CONFIG REQUIRED)

# or fmt-header-only
target_link_libraries(... fmt)
```

Bugs

- Looks like on Android the `<clocale>` API is not implemented, so `{:n} formatter <https://github.com/fmtlib/fmt/issues/305>'__ does not work, see https://github.com/fmtlib/fmt/issues/327

4.1.273 folly

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(folly)
find_package(folly CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Folly::folly)
```

4.1.274 forcefeedback

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(forcefeedback REQUIRED)
target_link_libraries(... forcefeedback::forcefeedback)
```

Same as

```
target_link_libraries(... "-framework ForceFeedback")
```

- https://developer.apple.com/documentation/forcefeedback?language=objc

4.1.275 foundation

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

Note: Framework `Cocoa` is a pseudo framework which in fact is just a combination of:

- `Foundation`
- `AppKit`
- `CoreData`

```
find_package(foundation REQUIRED)
target_link_libraries(... foundation::foundation)
```

Same as

```
target_link_libraries(... "-framework Foundation")
```

- https://developer.apple.com/documentation/foundation?language=objc

4.1.276 freetype

- Official
- Hunterized
- Example
- Added by Denis A. Kerzhemanov (pr-284)

```
hunter_add_package(freetype)
find_package(freetype CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PRIVATE freetype::freetype)
```

For compatibility with FindFreetype module, the `ALIAS` target `Freetype::Freetype` can be used too:

```
add_executable(baz boo.cpp)
target_link_libraries(baz PRIVATE Freetype::Freetype)
```

4.1.277 freetype-gl

- Official
- Hunterized
- Example (pr-249)

```
hunter_add_package(freetype-gl)
find_package(freetype-gl CONFIG REQUIRED)
```

```
add_executable(freetype-gl-example main.cpp)
target_link_libraries(freetype-gl-example PRIVATE freetype-gl::freetype-gl)
```

4.1.278 frugally-deep

- Official
- Example

```
hunter_add_package(frugally-deep)
find_package(frugally-deep CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo frugally-deep::fdeep)
```

4.1.279 gRPC

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1089)

```
hunter_add_package(gRPC)
find_package(gRPC CONFIG REQUIRED)
target_link_libraries(... gRPC::grpc)
```

4.1.280 gamecontroller

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(gamecontroller REQUIRED)
target_link_libraries(... gamecontroller::gamecontroller)
```

Same as

```
target_link_libraries(... "-framework GameController")
```

- https://developer.apple.com/documentation/gamecontroller?language=objc

4.1.281 gauze

- Official
- Example

```
hunter_add_package(gauze)
find_package(gauze CONFIG REQUIRED)
gauze_add_test(NAME foo COMMAND foo)
```

4.1.282 gemmlowp

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1149)

```
hunter_add_package(gemmlowp)
find_package(gemmlowp CONFIG REQUIRED)
target_link_libraries(gemmlowp gemmlowp::gemmlowp)
```

4.1.283 geos

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.284 getopt

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.285 gflags

- Official
- Hunterized
- Example
- Available since

```
hunter_add_package(gflags)
find_package(gflags CONFIG REQUIRED)
target_link_libraries(... gflags)
```

4.1.286 giflib

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1152)

```
hunter_add_package(giflib)
find_package(giflib CONFIG REQUIRED)
target_link_libraries(giflib giflib::giflib)
```

4.1.287 gl4es

- Official
- Hunterized
- Example
- Added by drodin (pr-143)

```
hunter_add_package(gl4es)
find_package(gl4es CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC gl4es::GL)
```

4.1.288 glapi

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(glapi REQUIRED)
target_link_libraries(... glapi::glapi)
```

Ubuntu:

```
> sudo apt-get install -y libgl1-mesa-dev
```

Travis:

```
addons:
  apt:
    packages:
      - libgl1-mesa-dev
```

4.1.289 glbinding

- Official
- Hunterized
- Example
- Added by NeroBurner (pr-1073)

```
hunter_add_package(glbinding)
find_package(glbinding CONFIG REQUIRED)
target_link_libraries(glbinding glbinding::glbinding)
```

4.1.290 gles2

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(gles2 REQUIRED)
target_link_libraries(... gles2::gles2)
```

4.1.291 gles3

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(gles3 REQUIRED)
target_link_libraries(... gles3::gles3)
```

4.1.292 glew

- http://glew.sourceforge.net
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(glew)
find_package(glew CONFIG REQUIRED)
target_link_libraries(boo PUBLIC glew::glew)
```

4.1.293 glfw

- Official
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(glfw)
find_package(glfw3 REQUIRED)
target_link_libraries(... glfw)
```

4.1.294 glib

- Official
- Example

```
hunter_add_package(glib)
find_package(glib-2.0 CONFIG REQUIRED)
target_link_libraries(... PkgConfig::glib-2.0)
```

4.1.295 glkit

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(glkit REQUIRED)
target_link_libraries(... glkit::glkit)
```

Same as

```
target_link_libraries(... "-framework GLKit")
```

- https://developer.apple.com/documentation/glkit?language=objc

4.1.296 glm

- Official
- Hunterized
- Example

```
hunter_add_package(glm)
find_package(glm REQUIRED)
target_link_libraries(... PRIVATE glm)
```

4.1.297 globjects

- Official
- Hunterized
- Example
- Added by NeroBurner (pr-1075)

Required customization:

```
hunter_config(glbinding VERSION "2.1.3-p0")
```

Usage:

```
hunter_add_package(globjects)
find_package(globjects CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PRIVATE globjects::globjects)
```

4.1.298 glog

- Official
- Hunterized
- Maintainer: https://github.com/NeroBurner

```
hunter_add_package(glog)
find_package(glog CONFIG REQUIRED)
target_link_libraries(... glog::glog)
```

For Hunter <= v0.17.15:

```
hunter_add_package(glog)
find_package(glog CONFIG REQUIRED)
target_link_libraries(... glog)
```

Warning: Does not work on Android:

- https://github.com/google/glog/issues/59

4.1.299 glproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.300 glslang

- Official
- Hunterized
- Example
- Added by Jon Spencer (pr-1475)

```
hunter_add_package(glslang)
find_package(glslang CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC glslang::glslang)
```

4.1.301 glu

- Official
- Hunterized
- Example
- Added by drodin (pr-N)

```
hunter_add_package(glu)
find_package(glu CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC glu::GLU)
```

4.1.302 gsl-lite

- Official
- Example

```
hunter_add_package(gsl-lite)
find_package(gsl-lite CONFIG REQUIRED)
target_link_libraries(... gsl::gsl-lite)
```

4.1.303 gst_plugins_bad

- Official
- Example

```
hunter_add_package(gst_plugins_bad)
find_package(gstreamer-bad-video-1.0 CONFIG REQUIRED)
target_link_libraries(... PkgConfig::gstreamer-bad-video-1.0)
```

4.1.304 gst_plugins_base

- Official
- Example

```
hunter_add_package(gst_plugins_base)
find_package(gstreamer-video-1.0 CONFIG REQUIRED)
target_link_libraries(... PkgConfig::gstreamer-video-1.0)
```

4.1.305 gst_plugins_good

- Official
- Example

```
hunter_add_package(gst_plugins_good)
# ???
```

4.1.306 gst_plugins_ugly

- Official
- Example

```
hunter_add_package(gst_plugins_ugly)
# ???
```

4.1.307 gstreamer

- Official
- Example

```
hunter_add_package(gstreamer)
find_package(gstreamer-1.0 CONFIG REQUIRED)
target_link_libraries(... PkgConfig::gstreamer-1.0)
```

Warning:

- Only Linux tested

4.1.308 gumbo

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1062)

```
hunter_add_package(gumbo)
find_package(gumbo CONFIG REQUIRED)
target_link_libraries(... gumbo::gumbo)
```

4.1.309 h3

- Official
- Example
- Added by Isaac Hier (pr-1408)

```
find_package(h3 CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main h3::h3)
```

4.1.310 half

- Official
- Hunterized
- Example
- Available since
- Added by David Hirvonen (pr-286)

```
hunter_add_package(half)
find_package(half CONFIG REQUIRED)
target_link_libraries(... half::half)
```

4.1.311 harfbuzz

- Official
- Hunterized
- Example
- Added by Jon Spencer (pr-1440)

```
hunter_add_package(harfbuzz)
find_package(harfbuzz CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC harfbuzz::harfbuzz)
```

4.1.312 hdf5

- Hunterized
- Example

```
hunter_add_package(hdf5)

find_package(ZLIB CONFIG REQUIRED)
find_package(szip CONFIG REQUIRED)
find_package(hdf5 CONFIG REQUIRED)

target_link_libraries(... hdf5)
```

4.1.313 highwayhash

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1151)

```
hunter_add_package(highwayhash)
find_package(highwayhash CONFIG REQUIRED)
target_link_libraries(highwayhash highwayhash::highwayhash)
```

4.1.314 http-parser

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1375)

```
hunter_add_package(http-parser)
find_package(http-parser CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main http-parser::http_parser)
```

4.1.315 hunter_venv

- https://github.com/hunter-packages/hunter_venv
- Example

This package is used to create an isolated Python environment inside Hunter and is a workaround for missing Python package. It is designed to be used with FindPython module. CMake 3.13 is a minimum required (see details below).

```
hunter_add_package(hunter_venv)
find_package(hunter_venv CONFIG REQUIRED)

find_package(Python REQUIRED)

add_custom_target(python_version ALL Python::Interpreter --version)

execute_process(COMMAND ${Python_EXECUTABLE} --version RESULT_VARIABLE result)
if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

Python version that will be used to create environment can be set by `HUNTER_VENV_PYTHON_VERSION` variable:

```
# local config.cmake

hunter_config(
    hunter_venv
    VERSION ${HUNTER_hunter_venv_VERSION}
    CMAKE_ARGS HUNTER_VENV_PYTHON_VERSION=3.6.7
)
```

Requested Python version and `virtualenv` should be installed in a system.

Default values for `HUNTER_VENV_PYTHON_VERSION` will match testing CI environment of Travis/AppVeyor machines:

```
if(APPLE)
  set(__hunter_venv_default_python "3.7.5")
elseif(MSYS)
  set(__hunter_venv_default_python "3.7.5")
elseif(WIN32)
  set(__hunter_venv_default_python "3.6.8")
else()
  set(__hunter_venv_default_python "3.5.2")
endif()

hunter_cmake_args(
    hunter_venv
    CMAKE_ARGS
    HUNTER_VENV_PYTHON_VERSION=${__hunter_venv_default_python}
)
```

At this moment the procedure of making a relocatable Python environment is not robust (see virtualenv issue #1169). Because of that `activate` and `deactivate` scripts removed from the created environment and for other scripts shebangs set to general `#!/usr/bin/env python` value. It means that before running a Python script, you will have to set the `PATH` environment variable accordingly. As a more convenient and less error-prone approach, you can use the `Python_EXECUTABLE` variable:

```
execute_process(
    COMMAND ${Python_EXECUTABLE} -c "import sys"
    RESULT_VARIABLE result
)
```

```
execute_process(
    COMMAND ${Python_EXECUTABLE} -c "print ('Hello Hunter!')"
    RESULT_VARIABLE result
)
```

```
execute_process(
    COMMAND ${Python_EXECUTABLE} ${CMAKE_CURRENT_LIST_DIR}/script.py
    RESULT_VARIABLE result
)
```

While calling `find_package(hunter_venv CONFIG REQUIRED)` variables `Python*_FIND_REGISTRY` and `CMAKE_FIND_FRAMEWORK` will be set to `NEVER`. Otherwise, `find_package(Python REQUIRED)` will return Python executable from the system instead of Python from created virtual environment:

- https://cmake.org/cmake/help/v3.13/module/FindPython.html#hints
- https://cmake.org/cmake/help/v3.13/module/FindPython2.html#hints
- https://cmake.org/cmake/help/v3.13/module/FindPython3.html#hints

4.1.316 hypre

- Official
- Example
- Added by craffael (pr-420)

```
hunter_add_package(hypre)
find_package(HYPRE CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC HYPRE::HYPRE)
```

Note: MPI is disabled by default.

Use `HYPRE_WITH_MPI=ON` to enable it (and make sure MPI is installed on your system).

```
# config.cmake
hunter_config(hypre
    VERSION ${HUNTER_hypre_VERSION}
    CMAKE_ARGS
        HYPRE_WITH_MPI=ON
)
```

4.1.317 ice

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized

- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.318 icu-le-hb

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(icu-le-hb)
find_package(icu-le-hb CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC icu-le-hb::icu-le-hb)
```

4.1.319 icu-lx

- http://site.icu-project.org
- Hunterized
- Example

This library is part of the ICU project and separated from the main *ICU* package to break a circular dependency (see documentation):

- *ICU (with lx)* -> *icu-le-hb* -> *harfbuzz* -> *ICU (without lx)*

```
hunter_add_package(icu-lx)
find_package(icu-lx CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC ICU::lx)
```

4.1.320 imageio

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(imageio REQUIRED)
target_link_libraries(... imageio::imageio)
```

Same as

```
target_link_libraries(... "-framework ImageIO")
```

- https://developer.apple.com/documentation/imageio?language=objc

4.1.321 imagequant

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1672)

```
hunter_add_package(imagequant)
find_package(imagequant CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC imagequant::imagequant)
```

4.1.322 imgui

- Official
- Hunterized
- Example
- Added by Casey (pr-1521)

Immediate-mode, bloat-free graphical user interface library for C++

```
hunter_add_package(imgui)
find_package(imgui CONFIG REQUIRED)

add_executable(imgui_test main.cpp)
target_link_libraries(imgui_test imgui::imgui)
```

4.1.323 imshow

- Official
- Hunterized
- Example

```
hunter_add_package(imshow)
find_package(imshow CONFIG REQUIRED)
target_link_libraries(... imshow::imshow)
```

4.1.324 inja

- Official
- Example

- Added by Jorrit Wronski (pr-1207)

```
hunter_add_package(inja)
find_package(inja CONFIG REQUIRED)
target_link_libraries(inja inja::inja)
```

4.1.325 inputproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.326 intltool

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.327 intsizeof

- Official
- Example

```
hunter_add_package(intsizeof)
find_package(intsizeof CONFIG REQUIRED)
target_link_libraries(... PUBLIC intsizeof::intsizeof)
```

4.1.328 intx

- Official
- Example
- Added by Paweł Bylica (pr-1846)

```
hunter_add_package(intx)
find_package(intx CONFIG REQUIRED)

add_executable(use_intx main.cpp)
target_link_libraries(use_intx intx::intx)
```

4.1.329 iokit

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(iokit REQUIRED)
target_link_libraries(... iokit::iokit)
```

Same as

```
target_link_libraries(... "-framework IOKit")
```

- https://developer.apple.com/documentation/iokit?language=objc

4.1.330 ios_sim

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.331 ippicv

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.332 Iroha-ed25519

- Official
- Example
- Added by Warchant (pr-1740)

```
hunter_add_package(iroha-ed25519)
find_package(ed25519 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC iroha::ed25519)
```

4.1.333 irrXML

- Official
- Hunterized

```
hunter_add_package(irrXML)
find_package(irrXML CONFIG REQUIRED)
target_link_libraries(... irrXML::irrXML)
```

4.1.334 ittapi

- Official
- Hunterized
- Example
- Added by Raffael Casagrande (pr-483)

```
hunter_add_package(ittapi)
find_package(ittapi CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC ittapi::ittnotify)
```

4.1.335 jaegertracing

- Official
- Example
- Added by Isaac Hier (pr-1453)

```
find_package(jaegertracing CONFIG REQUIRED)
add_executable(main main.cpp)
target_link_libraries(main jaegertracing::jaegertracing-static)
```

4.1.336 jansson

- Official
- Hunterized
- Example
- Added by Isaac Hier (*pr-1372 <https://github.com/ruslo/hunter/pull/1372>*)

```
hunter_add_package(jansson)
find_package(jansson CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main jansson::jansson)
```

4.1.337 jasper

- http://www.ece.uvic.ca/~frodo/jasper/
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(jasper)
find_package(jasper CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC jasper::libjasper)
```

4.1.338 javascriptcore

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(javascriptcore REQUIRED)
target_link_libraries(... javascriptcore::javascriptcore)
```

Same as

```
target_link_libraries(... "-framework JavaScriptCore")
```

- https://developer.apple.com/documentation/javascriptcore?language=objc

4.1.339 jo_jpeg

- http://www.jonolick.com/code.html
- Hunterized
- Example

```
hunter_add_package(jo_jpeg)
find_package(jo_jpeg CONFIG REQUIRED)
target_link_libraries(foo jo_jpeg::jo_jpeg)
```

4.1.340 jpeg-compressor

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-269)

```
hunter_add_package(jpeg-compressor)
find_package(jpeg-compressor CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC jpeg-compressor::jpgd)
```

4.1.341 jsmn

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-279)

```
hunter_add_package(jsmn)
find_package(jsmn CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC jsmn::jsmn)
```

4.1.342 jsoncpp

- Official
- Example
- Available since

```
hunter_add_package(jsoncpp)

find_package(jsoncpp CONFIG REQUIRED)
target_link_libraries(... jsoncpp_lib_static)
```

4.1.343 jwt-cpp

- Official
- Example
- Added by Thalhammer (pr-467)

```
hunter_add_package(jwt-cpp)
find_package(jwt-cpp CONFIG REQUIRED)

add_executable(sample main.cpp)
target_link_libraries(sample PUBLIC jwt-cpp::jwt-cpp)
```

4.1.344 kNet

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(kNet)
find_package(kNet CONFIG REQUIRED)
target_link_libraries(boo PUBLIC kNet::kNet)
```

4.1.345 kbproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.346 lcms

- http://www.littlecms.com/
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(lcms)
find_package(lcms CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC lcms::lcms)
```

4.1.347 lehrfempp

- Official
- Example
- Added by craffael (pr-1629)

```
hunter_add_package(lehrfempp)
find_package(lehrfempp CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC LF::lf.base)
```

4.1.348 leveldb

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1246)

```
hunter_add_package(leveldb)
find_package(leveldb CONFIG REQUIRED)
target_link_libraries(... leveldb::leveldb)
```

4.1.349 libarchive

- Official
- Example
- Added by Timothy Stack (pr-293)

```
hunter_add_package(libarchive)
find_package(libarchive CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC PkgConfig::libarchive)
```

4.1.350 libbacktrace

- Official
- Example
- Added by Joerg-Christian Boehme (pr-174)

```
hunter_add_package(libbacktrace)
find_package(libbacktrace REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC libbacktrace::libbacktrace)
```

4.1.351 libcpuid

- http://libcpuid.sourceforge.net/
- Hunterized
- Example

```
hunter_add_package(libcpuid)
find_package(libcpuid CONFIG REQUIRED)
target_link_libraries(boo PUBLIC libcpuid::libcpuid)
```

4.1.352 libdaemon

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.353 libdill

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1069)

```
hunter_add_package(libdill)
find_package(libdill CONFIG REQUIRED)
target_link_libraries(libdill libdill::dill)
```

4.1.354 libevhtp

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1392)

```
hunter_add_package(libevhtp)
find_package(libevhtp CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main libevhtp::evhtp)
```

4.1.355 libffi

- Official
- Example

```
hunter_add_package(libffi)
find_package(libffi CONFIG REQUIRED)
target_link_libraries(... PkgConfig::libffi)
```

4.1.356 libigl

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1876)

```
hunter_add_package(libigl)
find_package(libigl CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC igl::common igl::core)
```

4.1.357 libjpeg-turbo

- https://libjpeg-turbo.org
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(libjpeg-turbo)
find_package(libjpeg-turbo CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC libjpeg-turbo::jpeg-static)
```

4.1.358 libjson-rpc-cpp

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.359 libmill

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1063)

```
hunter_add_package(libmill)
find_package(libmill CONFIG REQUIRED)
# `mill_s` is static library, `mill` is shared library
target_link_libraries(libmill libmill::mill_s)
```

4.1.360 libogg

- Official
- Hunterized
- Example
- Added by Meralis40 (pr-1451)

```
hunter_add_package(libogg)
find_package(libogg CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo libogg::ogg)
```

4.1.361 libpcre

- Official
- Example

```
hunter_add_package(libpcre)
find_package(libpcre CONFIG REQUIRED)
target_link_libraries(... PkgConfig::libpcre)
```

4.1.362 librtmp

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.363 libscrypt

- Official
- Hunterized
- Example

```
hunter_add_package(libscrypt)
find_package(libscrypt CONFIG REQUIRED)

add_executable(libscrypt_test main.cpp)
target_link_libraries(libscrypt_test libscrypt::scrypt)
```

4.1.364 libsodium

- Official Repository

```
hunter_add_package(libsodium)
find_package(libsodium CONFIG REQUIRED)
#...
target_link_libraries(... libsodium::libsodium)
```

4.1.365 libunibreak

- Official
- Example
- Added by Jon Spencer (pr-1443)

```
hunter_add_package(libunibreak)
find_package(libunibreak CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC PkgConfig::libunibreak)
```

4.1.366 libusb

- Official
- Official GitHub
- Example
- Added by Sebastien Collier (pr-1830)

```
hunter_add_package(libusb)
find_package(libusb-1.0 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC PkgConfig::libusb-1.0)
```

Pitfalls

- Requirements for Ubuntu:

```
sudo apt-get install libudev-dev
```

- Or configure without udev support

```
# Hunter configuration file, for example 'cmake/Hunter/config.cmake'
hunter_config(libusb VERSION [version] CMAKE_ARGS EXTRA_FLAGS=--disable-udev)
```

4.1.367 libuv

- Official
- Testing branch
- Example CMakeLists.txt
- Example C file

```
hunter_add_package(libuv)
find_package(libuv CONFIG REQUIRED)
target_link_libraries(... libuv::uv)
```

uv_ssize_t

Since libuv 1.14.0-p1 type `uv_ssize_t` should be used in API instead of `ssize_t`. This is not a part of official `1.x` API but will be the part of next official release. See for details:

- https://github.com/libuv/libuv/pull/1519

4.1.368 libxdg-basedir

- Official
- Example
- Added by tastytea (pr-1924)

```
hunter_add_package(libxdg-basedir)
find_package(libxdg-basedir CONFIG REQUIRED)

add_executable(example example.cpp)
target_link_libraries(example PUBLIC PkgConfig::libxdg-basedir)
```

4.1.369 libxml2

- http://xmlsoft.org/
- Example

```
hunter_add_package(libxml2)
find_package(libxml2 CONFIG REQUIRED)

add_executable(boo main.c)
target_link_libraries(boo PRIVATE libxml2::libxml2)
```

4.1.370 libyuv

- Official
- Documentation
- Hunterized
- Example

```
hunter_add_package(libyuv)
find_package(libyuv CONFIG REQUIRED)
target_link_libraries(... PUBLIC libyuv::yuv)
```

4.1.371 libzip

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1877)

```
hunter_add_package(libzip)
find_package(libzip CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC libzip::zip)
```

4.1.372 lmdb

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1172)

```
hunter_add_package(lmdb)
find_package(liblmdb CONFIG REQUIRED)
target_link_libraries(lmdb liblmdb::lmdb)
```

4.1.373 lmdbxx

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1250)

```
hunter_add_package(lmdbxx)
find_package(lmdbxx CONFIG REQUIRED)
target_link_libraries(... lmdbxx::lmdbxx)
```

4.1.374 log4cplus

- Official
- Hunterized
- Example
- Available since

```
hunter_add_package(log4cplus)
find_package(log4cplus CONFIG REQUIRED)
target_link_libraries(... log4cplus::log4cplus)
```

4.1.375 lss

- Official
- Hunterized
- Example

```
hunter_add_package(lss)
find_package(lss CONFIG REQUIRED)
```

4.1.376 lz4

- http://www.lz4.org
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(lz4)
find_package(lz4 CONFIG REQUIRED)
target_link_libraries(boo PUBLIC lz4::lz4)
```

4.1.377 lzma

- Official
- Hunterized
- Example
- Available since

```
hunter_add_package(lzma)
find_package(lzma CONFIG REQUIRED)
target_link_libraries(... lzma::lzma)
```

4.1.378 magnum

magnum is a lightweight and modular C++11/C++14 graphics middleware for games and data visualization

- Official
- Main Site
- Example
- Developed by Vladimír Vondruš
- Added by Pascal Thomet (pr-1731)

```
hunter_add_package(magnum)
find_package(Magnum CONFIG REQUIRED
    GL
    MeshTools
    Primitives
    Shaders
    Sdl2Application)

add_executable(magnum-primitives PrimitivesExample.cpp)
target_link_libraries(magnum-primitives PRIVATE
    Magnum::Application
    Magnum::GL
```

```
    Magnum::Magnum
    Magnum::MeshTools
    Magnum::Primitives
    Magnum::Shaders)
```

4.1.379 md5

- Hunterized
- Example

```
hunter_add_package(md5)
find_package(md5 CONFIG REQUIRED)
target_link_libraries(boo PUBLIC md5::md5)
```

4.1.380 meshoptimizer

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-283)

```
hunter_add_package(meshoptimizer)
find_package(meshoptimizer CONFIG REQUIRED)

# Available:
#  Libraries:
#   meshoptimizer::meshoptimizer
#   meshoptimizer::libgltfpack (if MESHOPT_BUILD_GLTFPACK=ON)
#  Binaries:
#   meshoptimizer::gltfpack (if MESHOPT_BUILD_GLTFPACK=ON)
#   meshoptimizer::demo (if MESHOPT_BUILD_DEMO=ON)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC meshoptimizer::meshoptimizer)
```

4.1.381 metal

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(metal REQUIRED)
target_link_libraries(... metal::metal)
```

Same as

```
target_link_libraries(... "-framework Metal")
```

- https://developer.apple.com/documentation/metal?language=objc

4.1.382 mini_chromium

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.383 miniz

- Official
- Example
- Added by Rahul Sheth (pr-271)

```
hunter_add_package(miniz)
find_package(miniz CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC miniz::miniz)
```

4.1.384 minizip

- Official
- Hunterized
- Example

```
hunter_add_package(minizip)
find_package(minizip CONFIG REQUIRED)
target_link_libraries(... minizip::minizip)
```

4.1.385 mkl

- Official
- Example

```
hunter_add_package(mkl)

include_directories("${MKL_ROOT}/include")

add_executable(boo boo.cpp)
```

4.1.386 mkldnn

- Official
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(mkldnn)
find_package(mkldnn CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC MKLDNN::mkldnn)
```

4.1.387 mng

- https://sourceforge.net/projects/libmng/
- Hunterized
- Example

```
hunter_add_package(mng)
find_package(mng CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC mng::mng)
```

4.1.388 mobilecoreservices

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(mobilecoreservices REQUIRED)
target_link_libraries(... mobilecoreservices::mobilecoreservices)
```

Same as

```
target_link_libraries(... "-framework MobileCoreServices")
```

- https://developer.apple.com/documentation/mobilecoreservices?language=objc

4.1.389 mojoshader

- https://www.icculus.org/mojoshader/
- Hunterized
- Example

```
hunter_add_package(mojoshader)
find_package(mojoshader CONFIG REQUIRED)
target_link_libraries(boo PUBLIC mojoshader::mojoshader)
```

4.1.390 mongoose

- Official
- Hunterized
- Example
- Added by dvirtz (pr-1195)

```
hunter_add_package(mongoose)
find_package(mongoose CONFIG REQUIRED)

add_executable(mongoose ...)
target_link_libraries(mongoose mongoose::mongoose)
```

4.1.391 mpark_variant

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.392 msgpack

- Hunterized
- Example
- Added by Antal Tátrai (pr-406)
- Available since

```
hunter_add_package(msgpack)
find_package(msgpack CONFIG REQUIRED)
target_link_libraries(... msgpack::msgpack)
```

4.1.393 mshadow

- Official GitHub
- Hunterized
- Example

```
enable_language(CUDA)
include_directories(${CMAKE_CUDA_TOOLKIT_INCLUDE_DIRECTORIES})

hunter_add_package(mshadow)
find_package(mshadow CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC mshadow::mshadow)
```

4.1.394 mtplz

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.395 mxnet

- https://mxnet.apache.org/
- Official GitHub
- Hunterized
- Example

`mxnet` is not compatible with OpenCV `4.0`, you have to explicitly switch to OpenCV `3.4`:

```
# config.cmake

hunter_config(OpenCV VERSION 3.4.3-p4)
```

Please check *TVM documentation* for additional requirements.

Note:

- Package was tested only on Linux and macOS

- Library type is forced to be `SHARED` hence all dependencies should be shared libraries (use *HUNTER_BUILD_SHARED_LIBS=ON*) (**not tested!**) or build with *toolchain with PIC*.

Note: It's highly recommended to use `export OMP_NUM_THREADS=1` while running code and compiling MXNet. Not using this variable can leads to random runtime errors and build freezes.

- https://github.com/apache/incubator-mxnet/issues/10856

```
hunter_add_package(mxnet)
find_package(mxnet CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC mxnet::mxnet)
```

4.1.396 nanoflann

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.397 nanosvg

- Official
- Hunterized
- Example
- Added by Brad Kotsopoulos (pr-1658)

```
hunter_add_package(nanosvg)
find_package(nanosvg CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC nanosvg::nanosvg)
```

4.1.398 ncnn

- Official

- Hunterized
- Example

```
hunter_add_package(ncnn)
find_package(ncnn CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC ncnn::ncnn)
```

4.1.399 ncurses

- Official
- Example
- Added by Tim Stack (pr-N)

```
hunter_add_package(ncursesw)
find_package(ncursesw CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC PkgConfig::ncursesw)
```

4.1.400 nlohmann_fifo_map

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1673)

```
hunter_add_package(nlohmann_fifo_map)
find_package(nlohmann_fifo_map CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC nlohmann_fifo_map::fifo_map)
```

4.1.401 nlohmann_json

- Official GitHub
- Example

Note: C++ 11 is required, you can find the supported compiler versions in the official README.

Usage

```
hunter_add_package(nlohmann_json)
find_package(nlohmann_json CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC nlohmann_json::nlohmann_json)
```

Old version

CMake API that should be used for versions older than `nlohmann_json` v3.2.0:

```
hunter_add_package(nlohmann_json)
find_package(nlohmann_json CONFIG REQUIRED)
target_link_libraries(... nlohmann_json)
```

Very old version

CMake API that should be used for `nlohmann_json` v1.0.0:

```
hunter_add_package(nlohmann-json)
find_package(nlohmann-json REQUIRED)
target_link_libraries(... nlohmann-json::nlohmann-json)
```

Related Hunter releases:

- v1.0.0 Available from v0.12.13 to v0.19.52
- v2.1.1+ Available since v0.19.52

migration from v1.0.0 to v2.1.1+

- replace all `nlohmann-json` with `nlohmann_json`
- add `CONFIG` to `find_package(nlohmann_json CONFIG REQUIRED)`
- shorten `target_link_libraries()` to `target_link_libraries(... nlohmann_json)` **no** `nlohmann_json::nlohmann_json`
- change `#include <json.hpp>` to `#include <nlohmann/json.hpp>`

4.1.402 nng

- Official
- Example
- Added by tnixeu (pr-45)

```
hunter_add_package(nng)
find_package(nng CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC nng::nng)
```

4.1.403 nsync

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1169)

```
hunter_add_package(nsync)
find_package(nsync CONFIG REQUIRED)
target_link_libraries(foo nsync::nsync)
```

4.1.404 occt

OpenCascade Community Technology

- Official
- Hunterized
- Example
- Added by craffael (pr-295)

```
hunter_add_package(occt)
find_package(OpenCASCADE CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo occt::TKFillet  occt::TKOffset occt::TKBool  occt::TKBO␣
↪occt::TKShHealing  occt::TKPrim occt::TKTopAlgo occt::TKBRep occt::TKGeomAlgo␣
↪occt::TKGeomBase occt::TKG3d occt::TKG2d occt::TKMath occt::TKernel)
if(APPLE OR (UNIX AND NOT ANDROID))
  target_link_libraries(boo pthread)
elseif(WIN32)
  target_link_libraries(boo ws2_32)
endif()
```

Note:

- OpenCascade consists of a number of modules. This Hunterized version supports all modules except the Draw Test Harness.
- To build shared versions of occt (recommended), please use *HUNTER_BUILD_SHARED_LIBS=ON* or build with *toolchain with PIC*.
- On Ubuntu, make sure that you have installed the following system packages: `mesa-common-dev`, `libgl1-mesa-dev`, `libxmu-dev`, `libxi-dev`

4.1.405 odb

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official

- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.406 odb-boost

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.407 odb-compiler

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.408 odb-mysql

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.409 odb-pgsql

- Official
- Example
- Available since
- Added by Alexandre Pretyman (pr-307)

```
hunter_add_package(odb-pgsql)
find_package(odb COMPONENTS pgsql)

target_link_libraries(... odb::pgsql)
```

4.1.410 odb-sqlite

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.411 ogles_gpgpu

- GitHub Official
- Hunterized
- Example
- Maintainer: https://github.com/ruslo

```
hunter_add_package(ogles_gpgpu)
find_package(ogles_gpgpu CONFIG REQUIRED)
target_link_libraries(... ogles_gpgpu::ogles_gpgpu)
```

4.1.412 oneTBB

- Official
- Example
- Added by craffael (pr-600)

```
hunter_add_package(oneTBB)
find_package(TBB CONFIG REQUIRED)
find_package(Threads REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC TBB::tbb)
```

4.1.413 oniguruma

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1391)

```
hunter_add_package(oniguruma)
find_package(oniguruma CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main oniguruma::onig)
```

4.1.414 onmt

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z

- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.415 openddlparser

- Official
- Hunterized

```
hunter_add_package(openddlparser)
find_package(openddlparser CONFIG REQUIRED)
target_link_libraries(... openddlparser::openddl_parser)
```

4.1.416 opengles

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(opengles REQUIRED)
target_link_libraries(... opengles::opengles)
```

Same as

```
target_link_libraries(... "-framework OpenGLES")
```

- https://developer.apple.com/documentation/opengles?language=objc

4.1.417 opentracing-cpp

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1070)

```
hunter_add_package(opentracing-cpp)
find_package(OpenTracing CONFIG REQUIRED)
# Shared library
target_link_libraries(... OpenTracing::opentracing)
# Static library
target_link_libraries(... OpenTracing::opentracing-static)
```

4.1.418 opusfile

- Official
- Hunterized

- Example
- Added by drodin (pr-246)

```
hunter_add_package(opusfile)
find_package(opusfile CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC opusfile::opusfile)
```

4.1.419 osmesa

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(osmesa REQUIRED)
target_link_libraries(... osmesa::osmesa)
```

Ubuntu:

```
> sudo apt-get install -y libosmesa6-dev
```

Travis:

```
addons:
  apt:
    packages:
      - libosmesa6-dev
```

4.1.420 pcg

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1377)

```
find_package(pcg CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main pcg::pcg_random)
```

4.1.421 pciaccess

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized

- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.422 pcre2

- https://www.pcre.org/
- Hunterized
- Example

```
hunter_add_package(pcre2)
find_package(PCRE2 CONFIG REQUIRED)

add_executable(boo boo.c)
target_link_libraries(boo PUBLIC PCRE2::pcre2-8)
```

4.1.423 pegtl

- Official
- Example
- Added by Jörg-Christian Böhme (pr-1905)

```
hunter_add_package(pegtl)
find_package(pegtl CONFIG REQUIRED)

add_executable(pex boo.cpp)
target_link_libraries(pex PUBLIC taocpp::pegtl)
```

4.1.424 pip_GitPython

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_GitPython)
find_package(pip_GitPython CONFIG REQUIRED)

set(test_command "import git")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
```

```
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.425 pip_astroid

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_astroid)
find_package(pip_astroid CONFIG REQUIRED)

set(test_command "import astroid")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.426 pip_boto3

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_boto3)
find_package(pip_boto3 CONFIG REQUIRED)

set(test_command "import boto3")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
```

```
  message(FATAL_ERROR "Failed")
endif()
```

4.1.427 pip_botocore

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_botocore)
find_package(pip_botocore CONFIG REQUIRED)

set(test_command "import botocore")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.428 pip_certifi

- PyPI
- Example

```
hunter_add_package(pip_certifi)
find_package(pip_certifi CONFIG REQUIRED)

set(test_command "import certifi")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.429 pip_chardet

- Official

- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_chardet)
find_package(pip_chardet CONFIG REQUIRED)

set(test_command "import chardet")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.430 pip_cpplint

- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_cpplint)
find_package(pip_cpplint CONFIG REQUIRED)

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -m cpplint --help
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.431 pip_decorator

- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_decorator)
find_package(pip_decorator CONFIG REQUIRED)

set(test_command "import decorator")
```

```
execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.432 pip_gitdb

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_gitdb)
find_package(pip_gitdb CONFIG REQUIRED)

set(test_command "import gitdb")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.433 pip_idna

- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_idna)
find_package(pip_idna CONFIG REQUIRED)

set(test_command "import idna")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)
```

```
if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.434 pip_jmespath

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_jmespath)
find_package(pip_jmespath CONFIG REQUIRED)

set(test_command "import jmespath")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.435 pip_lazy-object-proxy

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_lazy-object-proxy)
find_package(pip_lazy-object-proxy CONFIG REQUIRED)

set(test_command "import lazy_object_proxy")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.436 pip_nose

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_nose)
find_package(pip_nose CONFIG REQUIRED)

set(test_command "import nose")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.437 pip_nose-timer

- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_nose-timer)
find_package(pip_nose-timer CONFIG REQUIRED)

set(test_command "import nosetimer")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.438 pip_numpy

- Official
- Official GitHub
- PyPI

- Example

```
hunter_add_package(pip_numpy)
find_package(pip_numpy CONFIG REQUIRED)

set(test_command "import numpy")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.439 pip_pylint

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_pylint)
find_package(pip_pylint CONFIG REQUIRED)

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -m pylint --help
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.440 pip_python-dateutil

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_python-dateutil)
find_package(pip_python-dateutil CONFIG REQUIRED)

set(test_command "import dateutil")
```

```
execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.441 pip_requests

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_requests)
find_package(pip_requests CONFIG REQUIRED)

set(test_command "import requests")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.442 pip_six

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_six)
find_package(pip_six CONFIG REQUIRED)

set(test_command "import six")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
```

```
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.443 pip_smmap

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_smmap)
find_package(pip_smmap CONFIG REQUIRED)

set(test_command "import smmap")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.444 pip_urllib3

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_urllib3)
find_package(pip_urllib3 CONFIG REQUIRED)

set(test_command "import urllib3")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.445 pip_wrapt

- Official
- Official GitHub
- PyPI
- Example

```
hunter_add_package(pip_wrapt)
find_package(pip_wrapt CONFIG REQUIRED)

set(test_command "import wrapt")

execute_process(
    COMMAND
    ${Python_EXECUTABLE} -c ${test_command}
    RESULT_VARIABLE
    result
)

if(NOT result EQUAL "0")
  message(FATAL_ERROR "Failed")
endif()
```

4.1.446 pluginlib

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1926)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(pluginlib)
find_package(catkin CONFIG REQUIRED COMPONENTS pluginlib)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.447 poly2tri

- Official
- Hunterized
- Example

```
hunter_add_package(poly2tri)
find_package(poly2tri CONFIG REQUIRED)
target_link_libraries(... poly2tri::poly2tri)
```

4.1.448 polyclipping

- Official
- Hunterized
- Example

```
hunter_add_package(polyclipping)
find_package(polyclipping CONFIG REQUIRED)
target_link_libraries(... polyclipping::polyclipping)
```

4.1.449 presentproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.450 prometheus-cpp

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1680)

```
hunter_add_package(prometheus-cpp)
find_package(prometheus-cpp CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC prometheus-cpp::pull)
```

4.1.451 protobuf-c

- Official
- Hunterized
- Example
- Added by isaachier (pr-1382)

```
hunter_add_package(Protobuf)
find_package(Protobuf CONFIG REQUIRED)
hunter_add_package(protobuf-c)
find_package(protobuf-c CONFIG REQUIRED)
add_custom_command(
  OUTPUT "${CMAKE_CURRENT_BINARY_DIR}/person.pb-c.c"
         "${CMAKE_CURRENT_BINARY_DIR}/person.pb-c.h"
  COMMAND protobuf::protoc
  ARGS --plugin=$<TARGET_FILE:protobuf-c::protoc-gen-c>
       --c_out=${CMAKE_CURRENT_BINARY_DIR}
       person.proto
  DEPENDS person.proto protobuf::protoc protobuf-c::protoc-gen-c
  WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR})
add_executable(main main.c
  "${CMAKE_CURRENT_BINARY_DIR}/person.pb-c.c"
  "${CMAKE_CURRENT_BINARY_DIR}/person.pb-c.h")
target_include_directories(main PUBLIC
  $<BUILD_INTERFACE:${CMAKE_CURRENT_BINARY_DIR}>)
target_link_libraries(main protobuf-c::protobuf-c)
```

4.1.452 pthread-stubs

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.453 pthreads-win32

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-449)

```
hunter_add_package(pthreads-win32)
find_package(pthreads-win32 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC pthreads-win32::pthreads)
```

4.1.454 pugixml

- Official
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(pugixml)
find_package(pugixml CONFIG REQUIRED)
target_link_libraries(boo PUBLIC pugixml)
```

4.1.455 pybind11

- Official
- Example
- Added by Isaac Hier (pr-1140)

```
hunter_add_package(pybind11)
find_package(pybind11 CONFIG REQUIRED)
target_link_libraries(... pybind11::pybind11 pybind11::embed pybind11::module)
```

4.1.456 qhull

- Official
- Hunterized
- Example
- Added by qhull_developer (pr-1596)

```
hunter_add_package(qhull)
find_package(qhull CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo qhull::libqhull)
```

4.1.457 quartzcore

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(quartzcore REQUIRED)
target_link_libraries(... quartzcore::quartzcore)
```

Same as

```
target_link_libraries(... "-framework QuartzCore")
```

- https://developer.apple.com/documentation/quartzcore?language=objc

4.1.458 quickjs

- https://bellard.org/quickjs/
- Hunterized
- Example

```
hunter_add_package(quickjs)
find_package(quickjs CONFIG REQUIRED)

add_executable(run-test262 run-test262.c)
target_link_libraries(run-test262 PRIVATE quickjs::quickjs)
```

4.1.459 rabbitmq-c

- Official
- Hunterized
- Example

```
hunter_add_package(rabbitmq-c)
find_package(rabbitmq-c REQUIRED)
target_link_libraries(... rabbitmq-c::rabbitmq-static)
```

4.1.460 rabit

- Official
- Hunterized
- Example

```
hunter_add_package(rabit)
find_package(rabit CONFIG REQUIRED)

add_executable(foo foo.cpp)
target_link_libraries(foo PUBLIC rabit::rabit)
```

4.1.461 randrproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z

- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.462 rang

- https://agauniyal.github.io/rang/
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(rang)
find_package(rang CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC rang::rang)
```

4.1.463 range-v3

- Official
- Hunterized
- Example
- Added by dvirtz (pr-1282)

```
hunter_add_package(range-v3)
find_package(range-v3 CONFIG REQUIRED)

add_executable(comprehensions comprehensions.cpp)
target_link_libraries(comprehensions PUBLIC range-v3)
```

4.1.464 re2

- Official
- Hunterized
- Example
- Added by David Hirvonen (pr-1171)

```
hunter_add_package(re2)
find_package(RE2 CONFIG REQUIRED)
target_link_libraries(foo RE2::re2)
```

4.1.465 readline

- Official
- Example
- Added by Tim Stack (pr-1912)

```
hunter_add_package(readline)
find_package(readline REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC readline::readline)
```

4.1.466 recastnavigation

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(recastnavigation)
find_package(recastnavigation CONFIG REQUIRED)
target_link_libraries(
    boo
    PUBLIC
    recastnavigation::detour
    recastnavigation::detour_crowd
    recastnavigation::detour_tile_cache
    recastnavigation::recast
)
```

4.1.467 renderproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.468 rocksdb

- Official

- Official GitHub
- Example
- Started by Paweł Bylica chfast (pr-991)
- Completed by Isaac Hier isaachier (pr-1231)

```
hunter_add_package(rocksdb)
find_package(RocksDB CONFIG REQUIRED)

add_executable(rocksdb-test test.cpp)
target_link_libraries(rocksdb-test PUBLIC RocksDB::rocksdb)
```

4.1.469 ros

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1461)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros)
find_package(catkin CONFIG REQUIRED COMPONENTS roslib)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.470 ros_comm

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1930)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_comm)
find_package(catkin CONFIG REQUIRED COMPONENTS roscpp
    rosbag rosbag_storage topic_tools message_filters roslz4 xmlrpcpp)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.471 ros_comm_msgs

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1461)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_comm_msgs)
find_package(catkin CONFIG REQUIRED COMPONENTS rosgraph_msgs std_srvs)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
target_include_directories(main PRIVATE ${catkin_INCLUDE_DIRS})
```

4.1.472 ros_common_msgs

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1461)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_common_msgs)
find_package(catkin CONFIG REQUIRED
    COMPONENTS std_msgs actionlib_msgs geometry_msgs diagnostic_msgs nav_msgs
    sensor_msgs shape_msgs stereo_msgs trajectory_msgs visualization_msgs)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.473 ros_console_bridge

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1403)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_console_bridge)
find_package(console_bridge CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC ros::console_bridge)
```

4.1.474 ros_environment

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1450)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_environment)
find_package(catkin CONFIG REQUIRED COMPONENTS ros_environment)

catkin_package()
```

4.1.475 ros_gencpp

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1416)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_gencpp)
find_package(catkin CONFIG REQUIRED COMPONENTS gencpp)

add_message_files(FILES dummy.msg)
generate_messages()

catkin_package()
```

4.1.476 ros_geneus

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1416)

  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_geneus)
find_package(catkin CONFIG REQUIRED COMPONENTS geneus)

add_message_files(FILES dummy.msg)
generate_messages()

catkin_package()
```

4.1.477 ros_genlisp

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1416)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_genlisp)
find_package(catkin CONFIG REQUIRED COMPONENTS genlisp)

add_message_files(FILES dummy.msg)
generate_messages()

catkin_package()
```

4.1.478 ros_genmsg

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1410)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_genmsg)
find_package(catkin CONFIG REQUIRED COMPONENTS genmsg)

add_message_files(FILES dummy.msg)
generate_messages()

catkin_package()
```

4.1.479 ros_gennodejs

- Official

- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1416)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_gennodejs)
find_package(catkin CONFIG REQUIRED COMPONENTS gennodejs)

add_message_files(FILES dummy.msg)
generate_messages()

catkin_package()
```

4.1.480 ros_genpy

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1416)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_genpy)
find_package(catkin CONFIG REQUIRED COMPONENTS genpy)

add_message_files(FILES dummy.msg)
generate_messages()

catkin_package()
```

4.1.481 ros_message_generation

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1435)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_message_generation)
find_package(catkin CONFIG REQUIRED COMPONENTS message_generation)

add_message_files(FILES dummy.msg)
generate_messages()
```

```
catkin_package()
```

4.1.482 ros_message_runtime

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1439)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_message_runtime)
find_package(catkin CONFIG REQUIRED COMPONENTS message_runtime)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ros::rostime)
```

4.1.483 ros_std_msgs

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1450)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(ros_std_msgs)
find_package(catkin CONFIG REQUIRED COMPONENTS std_msgs)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.484 rosconsole

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1907)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(rosconsole)
find_package(catkin CONFIG REQUIRED COMPONENTS rosconsole)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.485 roscpp_core

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1412)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(roscpp_core)
find_package(catkin CONFIG REQUIRED COMPONENTS cpp_common rostime
    roscpp_serialization roscpp_traits)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.486 rospack

- Official
- Hunterized
- Example
- **Added by**
  - Lukas Solanka (pr-1435)
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(rospack)
find_package(catkin CONFIG REQUIRED COMPONENTS rospack)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.487 s3

- https://github.com/bji/libs3

- Hunterized
- Example

```
hunter_add_package(s3)
find_package(s3 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC s3::s3)
```

4.1.488 scelta

- Official
- Hunterized
- Example
- Added by Joerg-Christian Boehme (pr-142)

```
hunter_add_package(scelta)
find_package(scelta CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC scelta::headers)
```

4.1.489 sds

- Official
- Hunterized
- Example
- Added by Isaac Hier (pr-1254)

```
hunter_add_package(sds)
find_package(sds CONFIG REQUIRED)
target_link_libraries(... sds::sds)
```

4.1.490 sentencepiece

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(sentencepiece)
find_package(sentencepiece CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC sentencepiece::sentencepiece)
```

4.1.491 sentry

- Official
- Hunterized
- Example

```
hunter_add_package(sentry)
find_package(sentry CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC sentry::sentry)
```

4.1.492 shaderc

- Official
- Hunterized
- Example
- Added by Mathieu-Andre Chiasson (pr-N)

```
hunter_add_package(shaderc)
find_package(shaderc CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC shaderc::shaderc)
```

4.1.493 shaka_player_embedded

- Official GitHub
- Hunterized
- Example
- Stand-alone example

Next customization should be applied for dependencies:

```
hunter_config(
    CURL
    VERSION ${HUNTER_CURL_VERSION}
    CMAKE_ARGS CMAKE_USE_BORINGSSL=ON
)

hunter_config(
    v8
    VERSION 3.29.86-90da229-p0
)
```

Usage:

```
hunter_add_package(shaka_player_embedded)
find_package(shaka_player_embedded CONFIG REQUIRED)
```

```
add_executable(boo ${sources})
target_link_libraries(boo PUBLIC shaka_player_embedded::shaka_player_embedded)
```

4.1.494 sleef

- Official
- Hunterized
- Example
- Added by xsacha (pr-1780)

```
hunter_add_package(sleef)
find_package(sleef CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main sleef::sleef)
```

4.1.495 sm

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.496 smol-v

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-281)

```
hunter_add_package(smol-v)
find_package(smol-v CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC smol-v::smol-v)
```

4.1.497 soil

- https://www.lonesock.net/soil.html
- GitHub
- Hunterized
- Example
- Added by frequem (pr-1703)

```
hunter_add_package(soil)
find_package(soil CONFIG REQUIRED)
add_executable(main main.c)
target_link_libraries(main soil::soil)
```

4.1.498 sources_for_android_sdk_packer

- Official
- Example

```
hunter_add_package(sources_for_android_sdk_packer)
```

4.1.499 sparsehash

- Official
- Hunter

Example:

```
hunter_add_package(sparsehash)
set(SPARSEHASH_INCLUDE_DIRS ${SPARSEHASH_ROOT}/include)

target_include_directories(... ${SPARSEHASH_INCLUDE_DIRS})
```

4.1.500 spdlog

- Official
- Hunterized
- Example

```
hunter_add_package(spdlog)

find_package(spdlog CONFIG REQUIRED)

target_link_libraries(... spdlog::spdlog)
```

4.1.501 spirv-cross

- Official
- Example
- Added by Jon Spencer (pr-1748)

```
hunter_add_package(spirv-cross)
find_package(spirv_cross_core CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC spirv-cross-core)
```

4.1.502 sqlite3

- https://www.sqlite.org
- Hunterized
- Example

```
hunter_add_package(sqlite3)
find_package(sqlite3 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC sqlite3::sqlite3)
```

4.1.503 sse2neon

- Hunterized
- Example

```
hunter_add_package(sse2neon)
find_package(sse2neon CONFIG REQUIRED)
target_link_libraries(... sse2neon::sse2neon)
```

4.1.504 stanhull

- Hunterized
- Example

```
hunter_add_package(stanhull)
find_package(stanhull CONFIG REQUIRED)
target_link_libraries(boo PUBLIC stanhull::stanhull)
```

4.1.505 state_machine

- Official
- Example
- Added by NukeBird (pr-1163)

```
hunter_add_package(state_machine)
find_package(state_machine CONFIG REQUIRED)
target_link_libraries(sm state_machine)
```

4.1.506 stb

- Official
- Hunterized
- Example

```
hunter_add_package(stb)
find_package(stb CONFIG REQUIRED)
target_link_libraries(boo PUBLIC stb::stb)
```

Notes

Since v0.0.0-80c8f6a-p0, most stb libraries in Hunter (excluding stb_textedit and stb_tilemap_editor) have implementations built inside the Hunter package and downstream consumers use header files that do NOT respect the stb _IMPLEMENTATION flags. This is to ensure a single version of the stb libraries is built.

4.1.507 stdext-path

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(stdext-path)
find_package(stdext-path CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC stdext-path::stdext-path)
```

4.1.508 stormlib

- Official
- Hunterized
- Example
- Available since v0.19.83
- Added by wheybags (pr-877)

```
hunter_add_package(stormlib)
find_package(stormlib CONFIG REQUIRED)
#...
target_link_libraries(foo stormlib::stormlib)
```

4.1.509 sugar

- Official GitHub
- Example

```
hunter_add_package(sugar)
find_package(sugar CONFIG REQUIRED)

sugar_include(boo)
```

4.1.510 szip

- Hunterized
- Example

```
hunter_add_package(szip)
find_package(szip REQUIRED)

target_link_libraries(... szip::szip)
```

4.1.511 tacopie

- Official
- Official github fork
- Hunterized
- Example
- Available since

```
hunter_add_package(tacopie)
find_package(tacopie CONFIG REQUIRED)
target_link_libraries(... tacopie::tacopie)
```

4.1.512 taocpp-json

- Official
- Example
- Added by Jörg-Christian Böhme (pr-1906)

```
hunter_add_package(taocpp-json)
find_package(taocpp-json REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC taocpp::json)
```

4.1.513 taskflow

- Official
- Example
- Added by Raffael Casagrande (pr-371)

```
hunter_add_package(taskflow)
find_package(Taskflow CONFIG REQUIRED NO_CMAKE_PACKAGE_REGISTRY)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC Taskflow::Taskflow)
```

4.1.514 tcl

- Official
- Example
- Added by drodin (pr-1515)

```
hunter_add_package(tcl)
find_package(tcl REQUIRED)

add_executable(tcl_test tcl_test.c)
target_link_libraries(tcl_test PUBLIC tcl::tcl)
```

4.1.515 termcolor

Termcolor is a header-only C++ library for printing colored messages to the terminal. Written just for fun with a help of the Force.

- Official
- Example

```
hunter_add_package(termcolor)
find_package(termcolor CONFIG REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC termcolor::termcolor)
```

4.1.516 tf

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1933)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(tf)
find_package(catkin CONFIG REQUIRED COMPONENTS tf)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.517 tf2

- Official
- Hunterized
- Example
- **Added by Krasimir Georgiev (pr-1932)**
  - Contribution partially as part of work at SeeByte Ltd.

```
hunter_add_package(tf2)
find_package(catkin CONFIG REQUIRED COMPONENTS tf2_msgs
    tf2 tf2_ros tf2_eigen tf2_bullet tf2_sensor_msgs)

catkin_package()

add_executable(main main.cpp)
target_link_libraries(main ${catkin_LIBRARIES})
```

4.1.518 theora

- Official
- Hunterized
- Example
- Added by drodin (pr-239)

```
hunter_add_package(theora)
find_package(theora REQUIRED)

add_executable(main main.cpp)
target_link_libraries(main PUBLIC theora::theora)
```

4.1.519 thread-pool-cpp

- GitHub official
- Hunterized
- Example

```
hunter_add_package(thread-pool-cpp)
find_package(thread-pool-cpp CONFIG REQUIRED)
target_link_libraries(... thread-pool-cpp::thread-pool-cpp)
```

4.1.520 thrift

- Official
- Hunterized
- Example
- Added by isaachier (pr-1064)

This package **does not** compile the Thrift compiler by default. Nor does it compile the `thriftz` and `thrifnb` libraries. It just builds the basic `thrift` library, without SSL support. To compile the Thrift compiler, you must pass in custom CMake arguments in your toolchain, namely `BUILD_COMPILER=ON`. Similarly, to build `thriftz`, pass `WITH_ZLIB=ON`. To build `thriftnb`, pass `WITH_LIBEVENT=ON`. To compile with SSL support, pass `WITH_OPENSSL=ON`.

```
hunter_add_package(thrift)
find_package(thrift CONFIG REQUIRED)
target_link_libraries(foo PUBLIC
    thrift::thrift            # Main thrift library, thrift_static for static library
    thrift::thriftz           # thrift ZLIB support
    thrift::thriftnb)         # thrift Libevent non-blocking support
```

4.1.521 tiny-process-library

- Official
- Hunterized
- Example
- Added by Joerg-Christian Boehme (pr-102)

```
hunter_add_package(tiny-process-library)
find_package(tiny-process-library CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tiny-process-library::tiny-process-library)
```

4.1.522 tinydir

- GitHub Official
- Hunterized
- Example

```
hunter_add_package(tinydir)
find_package(tinydir CONFIG REQUIRED)
target_link_libraries(... tinydir::tinydir)
```

4.1.523 tinyexr

- Official
- Hunterized

- Example
- Added by Rahul Sheth (pr-278)

```
hunter_add_package(tinyexr)
find_package(tinyexr CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tinyexr::tinyexr)
```

4.1.524 tinygltf

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-308)

```
hunter_add_package(tinygltf)
find_package(tinygltf CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tinygltf::tinygltf)
```

4.1.525 tinyobjloader

- Official
- Example
- Added by Rahul Sheth (pr-226)

```
hunter_add_package(tinyobjloader)
find_package(tinyobjloader CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tinyobjloader::tinyobjloader)
```

4.1.526 tinyrefl

- Official
- Hunterized
- Example
- Added by Joerg-Christian Boehme (pr-137)

```
hunter_add_package(tinyrefl)
find_package(tinyrefl CONFIG REQUIRED)
find_package(tinyrefl_tool CONFIG REQUIRED)

add_executable(boo boo.cpp)
tinyrefl_tool(TARGET boo HEADERS example.hpp)
target_link_libraries(boo PUBLIC tinyrefl::tinyrefl)
```

4.1.527 TinyXML2

- Official
- Hunterized
- Example
- Added by Lukas Solanka (pr-1426)

```
hunter_add_package(tinyxml2)
find_package(tinyxml2 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tinyxml2::tinyxml2)
```

Old version

When using *tinyxml2* versions before *8.1.0*, *tinyxml2* target is not name-spaced

```
hunter_add_package(tinyxml2)
find_package(tinyxml2 CONFIG REQUIRED)
target_link_libraries(... tinyxml2)
```

4.1.528 tmxparser

- Official
- Hunterized
- Example
- Added by Sebastien Collier (pr-1829)

```
hunter_add_package(tmxparser)
find_package(tmxparser CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tmxparser)
```

4.1.529 toluapp

Warning: This package is not compatible with default *Lua* version. If you want to use this package you have to explicitly set `5.1.*` in your local config:

```
# config.cmake
hunter_config(Lua VERSION 5.1.5-p3)
```

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(toluapp)
find_package(toluapp CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC toluapp::toluapp)
```

4.1.530 tomcrypt

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.531 tommath

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.532 tsl_hat_trie

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-276)

```
hunter_add_package(tsl_hat_trie)
find_package(tsl_hat_trie CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tsl::hat_trie)
```

4.1.533 tsl_robin_map

- Official
- Example
- Added by Rahul Sheth (pr-277)

```
hunter_add_package(tsl_robin_map)
find_package(tsl-robin-map CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tsl::robin_map)
```

4.1.534 tvm

- Official GitHub
- Hunterized
- Example

Note:

- Library type is forced to be `SHARED` hence all dependencies should be shared libraries (use *HUNTER_BUILD_SHARED_LIBS=ON*) (**not tested!**) or build with *toolchain with PIC*.

Because of the LLVM + Xcode build issue, the next workaround should be applied:

```
# config.cmake

if(APPLE AND XCODE)
  hunter_config(
      LLVM
      VERSION
      ${HUNTER_LLVM_VERSION}
      CMAKE_ARGS
      LLVM_BUILD_EXTERNAL_COMPILER_RT=ON
  )
endif()
```

By default CUDA used on Linux. Example of the `travis.yml` configuration:

- https://github.com/cpp-pm/hunter-testing/blob/81c936a1e04df8f46b84c7eb22b931da5dcf4d7c/.travis.yml#L155-L163

On Android, iOS and Windows only `tvm::tvm_runtime` will be built.

```
hunter_add_package(tvm)
find_package(tvm CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC tvm::tvm_runtime)
```

4.1.535 type_safe

- Official
- Hunterized
- Example
- Added by dvirtz (pr-1143)

```
hunter_add_package(type_safe)
find_package(type_safe CONFIG REQUIRED)
target_link_libraries(type_safe_example type_safe)
```

4.1.536 uikit

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(uikit REQUIRED)
target_link_libraries(... uikit::uikit)
```

Same as

```
target_link_libraries(... "-framework UIKit")
```

- https://developer.apple.com/documentation/uikit?language=objc

4.1.537 Units

- Official
- Example
- Added by achary (pr-1602)

```
hunter_add_package(units)
find_package(units CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC units::units)
```

4.1.538 uriparser

- Official

- Example
- Added by Harry Mallon (pr-384)

```
hunter_add_package(uriparser)
find_package(uriparser CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC uriparser::uriparser)
```

4.1.539 utf8

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1711)

```
hunter_add_package(utf8)
find_package(utf8cpp CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC utf8cpp)
```

4.1.540 util_linux

- Official
- Example

```
hunter_add_package(util_linux)

find_package(blkid CONFIG REQUIRED)
find_package(fdisk CONFIG REQUIRED)
find_package(mount CONFIG REQUIRED)
find_package(smartcols CONFIG REQUIRED)
find_package(uuid CONFIG REQUIRED)

target_link_libraries(
    ...
    PkgConfig::blkid
    PkgConfig::fdisk
    PkgConfig::mount
    PkgConfig::smartcols
    PkgConfig::uuid
)
```

4.1.541 uuid

- Official
- Example
- Added by Joerg-Christian Boehme (pr-193)

```
hunter_add_package(uuid)
find_package(uuid CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC PkgConfig::uuid)
```

4.1.542 v8

- https://v8.dev/
- Official GitHub
- Hunterized
- Example

```
hunter_add_package(v8)
find_package(v8 CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(
    boo
    PUBLIC
    v8::v8_libplatform
    v8::v8_libbase
    v8::v8_base
    v8::v8_nosnapshot
    v8::v8_init
    v8::v8_initializers
    v8::v8_libsampler
)
```

4.1.543 vectorial

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-1683)

```
hunter_add_package(vectorial)
find_package(vectorial CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC vectorial::vectorial)
```

4.1.544 videotoolbox

Note: This is a helper package. There is no corresponding package in Hunter to be included by `hunter_add_package(...)`

```
find_package(videotoolbox REQUIRED)
target_link_libraries(... videotoolbox::videotoolbox)
```

Same as

```
target_link_libraries(... "-framework VideoToolbox")
```

- https://developer.apple.com/documentation/videotoolbox?language=objc

4.1.545 vorbis

- Official
- Hunterized
- Example
- Added by Jon Spencer (pr-1455)

```
hunter_add_package(vorbis)
find_package(vorbis CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC vorbis::vorbis)
```

4.1.546 vurtun-lib

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-431)

```
hunter_add_package(vurtun-lib)
find_package(vurtun CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC vurtun::lib)
```

4.1.547 websocketpp

- Hunterized
- Example
- Added by Antal Tátrai (pr-400)

```
hunter_add_package(websocketpp)
find_package(websocketpp CONFIG REQUIRED)
target_link_libraries(... websocketpp::websocketpp)
```

4.1.548 wt

- Official
- Hunterized
- Example
- Added by Casey (pr-1655)

Wt is a web GUI library in modern C++.

```
hunter_add_package(wt)
find_package(wt CONFIG REQUIRED)

add_executable(wt_test main.cpp)
target_link_libraries(wt_test Wt::Wt Wt::HTTP)
```

4.1.549 wxWidgets

- Official
- Hunterized
- Example

```
hunter_add_package(wxWidgets)

find_package(wxWidgets REQUIRED core base)
include(${wxWidgets_USE_FILE})
target_link_libraries(... ${wxWidgets_LIBRARIES})
```

Issues

- Add GTK
- Add OpenGL
- Does it work on OS X? CMakeified wxWidgets hasn't had its OSX Bakefiles ported

4.1.550 wyrm

- Official
- Example
- Added by Arnaud Sevin (pr-1790)

```
hunter_add_package(wyrm)
find_package(wyrm CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC wyrm::wyrm)
```

4.1.551 x11

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.552 x264

- https://www.videolan.org/developers/x264.html
- Example

```
hunter_add_package(x264)
find_package(x264 CONFIG REQUIRED)

add_executable(boo example.c)
target_link_libraries(boo PRIVATE PkgConfig::x264)
```

4.1.553 xatlas

- Official
- Hunterized
- Example
- Added by Rahul Sheth (pr-233)

```
hunter_add_package(xatlas)
find_package(xatlas CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC xatlas::xatlas)
```

4.1.554 xau

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official

- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.555 xcb

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.556 xcb-proto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.557 xcursor

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.558 xdamage

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.559 xext

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.560 xextproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.561 xf86vidmodeproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.562 xfixes

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.563 xgboost

- Official GitHub
- Hunterized
- Example

```
hunter_add_package(xgboost)
find_package(xgboost CONFIG REQUIRED)
target_link_libraries(... xgboost::xgboost)
```

4.1.564 xi

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.565 xinerama

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.566 xineramaproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.567 xorg-macros

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.568 xproto

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.569 xrandr

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.570 xrender

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.571 xshmfence

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.572 xtrans

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.573 xxf86vm

Warning: This page is a template and contains no real information. Please send pull request with real description.

- __FIXME__ Official
- __FIXME__ Hunterized
- __FIXME__ Example
- Available since __FIXME__ vX.Y.Z
- Added by __FIXME__ (__FIXME__ pr-N)

```
hunter_add_package(__FIXME__)
find_package(__FIXME__ CONFIG REQUIRED)
target_link_libraries(foo __FIXME__::__FIXME__)
```

4.1.574 xxhash

- Official
- Hunterized
- Example
- Added by Warchant (pr-1738)

```
hunter_add_package(xxhash)
find_package(xxhash CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC xxhash)
```

4.1.575 yaml-cpp

- Hunterized
- Example
- Added by Antal Tátrai (pr-598)
- Available since

```
hunter_add_package(yaml-cpp)
find_package(yaml-cpp CONFIG REQUIRED)
target_link_libraries(... yaml-cpp::yaml-cpp)
```

NOTE: This is the `boost` based last `c++-98` version (0.5.3).

4.1.576 zip

- Official
- Example
- Added by Rahul Sheth (pr-1878)

```
hunter_add_package(zip)
find_package(zip CONFIG REQUIRED)

add_executable(boo boo.cpp)
target_link_libraries(boo PUBLIC zip::zip)
```

4.1.577 zlog

- Official
- Example
- Added by je-vv (pr-N)

```
hunter_add_package(zlog)
find_package(zlog CONFIG REQUIRED)

add_executable(boo boo.c)
target_link_libraries(boo PUBLIC zlog::zlog)
```

4.1.578 zookeeper

- Official
- Hunterized
- Example
- Available since v0.18.5
- Added by https://github.com/wheybags (pr-639)

```
hunter_add_package(zookeeper)
find_package(zookeeper CONFIG REQUIRED)
#...
target_link_libraries(foo zookeeper::zookeeper_mt)
#target_link_libraries(foo zookeeper::zookeeper_st) # if you want the single-threaded␣
↪lib instead
```

4.1.579 zstd

- Official
- Example

```
find_package(Threads REQUIRED)
hunter_add_package(zstd)
find_package(zstd CONFIG REQUIRED)
target_link_libraries(... zstd::libzstd_static Threads::Threads)
```

zstd::libzstd_shared target is also available. It will already be linked with *Threads::Threads*.

4.2 CMake Modules

- *autoutils* - CMake utilities to imitate autotools functions
- *CreateLaunchers* - CMake module to create command line and debug launchers, including MSVC ".user" file.
- *sugar* - CMake tools and examples

4.3 Concurrency

- *ArrayFire* - general-purpose library that simplifies the process of developing software that targets parallel and massively-parallel architectures including CPUs, GPUs, and other hardware acceleration devices.
- *Async++* - concurrency framework for C++11
- *BoostCompute*
- *GPUImage* - open source iOS framework for GPU-based image and video processing
- *LibCDS* - C++ library of Concurrent Data Structures
- *libdill* - C library that makes writing structured concurrent programs easy
- *libmill* - Go-style concurrency in C
- *ogles_gpgpu* - GPGPU for mobile devices and embedded systems using OpenGL ES 2.0
- *OpenCL* - OpenCL headers and Installable Client Driver
- *OpenCL-cpp* - header only OpenCL c++ wrappers
- *thread-pool-cpp* - High performance C++14 thread pool

4.4 Containers

- *sparsehash* - C++ associative containers
- *sds* - Simple Dynamic Strings library for C

4.5 Commandline Tools

- *gflags* - contains a C++ library that implements commandline flags processing
- *cxxopts* - Lightweight C++ command line option parser
- *CLI11* - command line parser for C++11 and beyond that provides a rich feature set with a simple and intuitive interface
- *readline* - command line editor

4.6 Compiler

- *ctti* - Compile Time Type Information for the C++ programming language.
- *bison* - general-purpose parser generator.
- *flex* - a tool for generating scanners.
- *LLVM* - collection of modular and reusable compiler and toolchain technologies.

4.7 Computer Vision

- *acf* - Aggregated Channel Feature object detection in C++ and OpenGL ES 2.0.
- *ccv* - A Modern Computer Vision Library
- *cvmatio* - Matlab Mat file read and write C++ class with OpenCV bindings.
- *cvsteer* - A concise implementation of separable steerable filters via Freeman and Adelson, including second derivative of Gaussian and its Hilbert transform, implemented with the OpenCV C++ API
- *dest* - high performance 2D shape tracking leveraging machine learning methods.
- *dlib* - modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems.
- *drishti* - Real time eye tracking for embedded and mobile devices.
- *eos* - A lightweight 3D Morphable Face Model fitting library in modern C++11/14
- *Leptonica* - Open source library containing software that is broadly useful for image processing and image analysis applications
- *OpenCV* - Open Source Computer Vision Library
- *Tesseract* - Open Source OCR Engine

4.8 Compression

- *BZip2* - high-quality data compressor.
- *lz4* - Extremely Fast Compression algorithm
- *lzma* - A compression library with an API similar to that of zlib.
- *minizip* - enables to extract files from a .zip archive file.
- *szip*
- *ZLIB* - A massively spiffy yet delicately unobtrusive compression library.
- *zstd* - Very flexible very fast compression and decompression.

4.9 Crypto

- *crc32c* - CRC32C implementation with support for CPU-specific acceleration instructions
- *OpenSSL* - open source project that provides a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS>'_ and Secure Sockets Layer (SSL>'_ protocols.

4.10 Database

- *leveldb* - a fast key-value storage library
- *lmdb* - Lightning Memory-Mapped Database Manager
- *lmdbxx* - C++11 wrapper for the LMDB embedded B+ tree database library
- *sqlite3* - Popular C lib implementing a small, fast & self-contained SQL database engine
- *MySQL-client*
- *odb-mysql*
- *odb-pgsql*
- *odb-sqlite*
- *PostgreSQL*
- *rocksdb* - an embeddable persistent key-value store for fast storage

4.11 Datetime

- *cctz* - library for translating between absolute and civil times using the rules of a time zone.
- *date* - The future C++ standard <*date*> library, but for earlier C++ compilers.

4.12 Graphics 2D/3D

- *aglet* - Tiny cross platform (headless) OpenGL context creation
- *Assimp* - portable Open Source library to import various well-known 3D model formats in a uniform manner
- *freetype* - render freetype fonts
- *freetype-gl* - render freetype fonts in opengl
- *glew* - The OpenGL Extension Wrangler Library
- *imgui* - Immediate-mode, bloat-free graphical user interface library for C++
- *mojoshader* - MojoShader is a library to work with Direct3D shaders on alternate 3D APIs and non-Windows platforms
- *ogles_gpgpu* - GPGPU for mobile devices and embedded systems using OpenGL ES 2.0
- *SDL2* - A cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D.
- *SDL_ttf* - Sample library which allows to use TrueType fonts in SDL applications
- *Urho3D* - Cross-platform 2D and 3D game engine

4.13 Testing

- *benchmark* - A library to support the benchmarking of functions, similar to unit-tests
- *Catch* - A modern, C++-native, header-only, framework for unit-tests, TDD and BDD C++ Automated Test Cases in Headers
- *crashpad* - crash-reporting system.
- *FakeIt* - C++ mocking made easy. A simple yet very expressive, header-only library for C++ mocking.
- *GMock* - extension to Google Test for writing and using C++ mock classes.
- *GTest* - Google's C++ test framework!
- *Igloo* - A framework for unit testing in C++

4.14 Logging

- *fmt* - Small, safe and fast formatting library
- *glog* - C++ implementation of the Google logging module
- *log4cplus* - simple to use C++ logging API providing thread-safe, flexible, and arbitrarily granular control over log management and configuration.
- *spdlog* - Super fast C++ logging library.

4.15 Frameworks

- *Boost* - peer-reviewed portable C++ source libraries.
- *BoostProcess*
- *jaegertracing* - Jaeger C++ tracing implementation
- *opentracing-cpp* - OpenTracing API for C++
- *Qt*
- *QtQmlManager*
- *wt* - Wt is a web GUI library in modern C++.
- *wxWidgets* - Cross-Platform GUI Library

4.16 Filesystem

- *hdf5* - data model, library, and file format for storing and managing data.
- *tinydir* - Lightweight, portable and easy to integrate C directory and file reader

4.17 Machine Learning

- *caffe* - fast open framework for deep learning.
- *dlib* - modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems.
- *xgboost* - Scalable, Portable and Distributed Gradient Boosting (GBDT, GBRT or GBM>‘_ Library
- *frugally-deep* - Header-only library for using Keras models in C++

4.18 IPC/Messaging

- *CapnProto* - Cap’n Proto serialization/RPC system - core tools and C++ library
- *Comet* - Modern (idiomatic>‘_ binding between COM and C++
- *rabbitmq-c* - C-language AMQP client library for use with v2.0+ of the RabbitMQ broker.
- *ZeroMQ* - provide an abstraction of asynchronous message queues, multiple messaging patterns, message filtering (subscriptions>‘_, seamless access to multiple transport protocols and more.
- *ZMQPP* - “high-level” C++ binding for ZeroMQ/0mq/zmq

4.19 Math

- *CLAPACK*
- *Eigen* - C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms.
- *GSL* - GNU Scientific Library

- *HastyNoise* - SIMD open source noise generation library with a large collection of different noise algorithms.
- *OpenBLAS* - OpenBLAS is an optimized BLAS library based on GotoBLAS2 1.13 BSD version
- *double-conversion* - provides binary-decimal and decimal-binary routines for IEEE doubles.
- *gemmlowp* - Low-precision matrix multiplication.
- *glm* - header only C++ mathematics library for graphics software based on the OpenGL Shading Language (GLSL) specifications.
- *half* - Half-precision floating point library
- *h3* - Hexagonal hierarchical geospatial indexing system
- *poly2tri* - 2D constrained Delaunay triangulation library
- *polyclipping* - Polygon and line clipping and offsetting library

4.20 Media

- *Jpeg* - library for JPEG image compression.
- *OpenAL* - software implementation of the OpenAL 3D audio API.
- *PNG* - library for use in applications that read, create, and manipulate PNG (Portable Network Graphics) raster image files.
- *SDL_mixer* - A sample multi-channel audio mixer library for SDL.
- *TIFF*
- *giflib* - library for reading and writing gif images.
- *libyuv* - YUV scaling and conversion functionality.
- *WebP* - library to encode and decode images in WebP format.

4.21 Networking

- *asio* - C++ 11 compatible implementation of the future <*networking*> standard library.
- *autobahn-cpp* - open-source implementations of the The WebSocket Protocol and The Web Application Messaging Protocol (WAMP>'_ network protocols.
- *Avahi* - Service Discovery for Linux using mDNS/DNS-SD – compatible with Bonjour
- *Beast* - HTTP and WebSocket built on Boost.Asio in C++11
- *c-ares* - A C library for asynchronous DNS requests
- *CppNetlibUri* - C++ Network URI
- *civetweb* - Embedded C/C++ web server
- *cpr* - C++ Requests: Curl for People, a spiritual port of Python Requests
- *CURL* - A command line tool and library for transferring data with URL syntax
- *gRPC* - A high performance, open-source universal RPC framework
- *http-parser* - HTTP request/response parser for C
- *Libevent* - An event notification library for developing scalable network servers.

- *libevhtp* - Extremely-fast and secure embedded HTTP server library
- *kNet* - Low-level networking protocol library.
- *mongoose* - Embedded Web Server Library.
- *Libssh2*
- *PocoCpp* - Cross-platform C++ libraries with a network/internet focus.
- *websocketpp* - C++ websocket client/server library

4.22 Random

- *pcg* - PCG Random Number Generation

4.23 Regex

- *libpcre* - Perl-compatible regular expression library
- *oniguruma* - modern and flexible regular expression library

4.24 Robotics

4.24.1 ROS

- *actionlib* - Provides a standardized interface for interfacing with preempt-able tasks.
- *angles* - Provides a set of simple math utilities to work with angles.
- *catkin* - ROS catkin build system
- *class_loader* - ROS independent library for dynamic class (i.e. plugin) introspection and loading from runtime libraries
- *pluginlib* - Library for loading/unloading plugins in ROS packages during runtime
- *ros* - Core ROS packages
- *ros_comm* - ROS communications-related packages
- *ros_comm_msgs* - ROS *ros_comm_msgs* package
- *ros_common_msgs* - ROS *common_msgs* package - commonly used messages in ROS
- *ros_console_bridge* - ROS console bridge package (logging, . . .)
- *ros_environment* - ROS *ros_environment* package
- *ros_gencpp* - ROS gencpp package - C++ message and service data structure generation
- *ros_geneus* - ROS geneus package - EusLisp ROS message and service generators
- *ros_genlisp* - ROS genlisp package - Lisp message generation for ROS
- *ros_genmsg* - ROS genmsg package - message and service data structure generation
- *ros_gennodejs* - ROS gennodejs package - ROS JavaScript message definition and serialization generators
- *ros_genpy* - ROS genpy package - Python ROS message and service generator

- *ros_message_generation* - ROS *message_generation* package
- *ros_message_runtime* - ROS *message_runtime* package
- *ros_std_msgs* - ROS *std_msgs* package - Contains minimal messages of primitive data types and multi-arrays
- *rosconsole* - ROS package that supports console output and logging
- *roscpp_core* - ROS C++ core package
- *rospack* - ROS rospack package - a command-line tool for retrieving information about ROS packages available on the filesystem
- *tf* - Packages for common geometric calculations including the ROS transform library, "tf"
- *tf2* - A set of ROS packages for keeping track of coordinate transforms.

4.25 Scripting

- *Lua* - powerful, efficient, lightweight, embeddable scripting language.
- *pybind11* - a lightweight header-only library that exposes C++ types in Python and vice versa.

4.26 Serialize

- *cereal* - A C++11 library for serialization
- *CsvParserCPlusPlus* - C++ library for parsing text files.
- *Expat* - XML parser library in C.
- *flatbuffers* - Memory Efficient Serialization Library
- *gumbo* - An HTML5 parsing library in pure C99
- *irrXML* - simple and fast open source xml parser for C++
- *jansson* - C library for encoding, decoding and manipulating JSON data
- *JsonSpirit* - C++ JSON Library including both a json-data-structure and parser (based on Boost.Spirit>'.
- *msgpack* - efficient binary serialization format.
- *nlohmann_json* - JSON for Modern C++
- *openddlparser* - A simple and fast OpenDDL Parser. OpenDDL is the shortcut for Open Data Description Language.
- *Protobuf* - Protocol Buffers - Google's data interchange format
- *protobuf-c* - Protocol Buffers implementation in C
- *RapidJSON* - A fast JSON parser/generator for C++ with both SAX/DOM style API
- *RapidXML* - attempt to create the fastest XML parser possible, while retaining usability, portability and reasonable W3C compatibility.
- *thrift* - software framework for scalable cross-language services development
- *TinyXML2* - TinyXML2 is a simple, small, efficient, C++ XML parser that can be easily integrated into other programs.
- *yaml-cpp* - human friendly data serialization standard for all programming languages.

- *jsoncpp* - A library that allows manipulating JSON values, including serialization and deserialization to and from strings.

4.27 Terminal

- *ncurses* - text UI library
- *rang* - A Minimal, Header only Modern c++ library for terminal goodies
- *readline* - command-line editor

4.28 OS

- *Android-Apk*
- *Android-Modules*
- *Android-SDK*
- *ios_sim*
- *QtAndroidCMake*
- *Washer* - Lightweight, header-only, C++ wrapper around the Windows API
- *WTL* - Windows Template Library (WTL) is a C++ library for developing Windows applications and UI components.

Note:

- Don't see packages you need? Feel free to leave a package request.

CHAPTER 5

Creating new package

5.1 Create package

This is a guide for adding new package to Hunter. We start with the simple one (CMake based, no dependencies), then cover "hunterization" (CMake based, depends on other packages). Final is a most complex one (non-CMake packages, creating custom build scheme).

5.1.1 CMake (no dependencies)

If your CMake code is correctly written and has no dependencies then release with sources can be used **as is** in Hunter. There is no need to have `HunterGate`/`hunter_add_package` calls and no need to have a maintenance fork.

Examples of such packages:

- *flatbuffers*
  - https://github.com/google/flatbuffers
  - See flatbuffers/hunter.cmake
  - Testing table: AppVeyor, Travis
- *rocksdb*
  - https://github.com/facebook/rocksdb
  - See rocksdb/hunter.cmake
  - Testing table: Travis
- *nlohmann_json*
  - https://github.com/nlohmann/json
  - See nlohmann_json/hunter.cmake
  - Testing table: AppVeyor, Travis

Default behavior

Please check that your package respect (i.e. does not rewrite) such CMake variables like:

- CMAKE_INSTALL_PREFIX (**critical**)
- CMAKE_{C,CXX}_FLAGS + variations (**critical**)
- CMAKE_{C,CXX}_COMPILER + friends (**critical**)
- CMAKE_BUILD_TYPE (not critical, but recommended)
- CMAKE_CONFIGURATION_TYPES (not critical, but recommended)
- BUILD_SHARED_LIBS (not critical, but may result some errors)

Environment

Configuration of the package should be predictable.

For example it should not depend on the fact that some package already installed or not:

```
find_package(OpenSSL)
if(OPENSSL_FOUND)
  target_compile_definitions(... PUBLIC FOO_WITH_OPENSSL=1)
endif()
```

If package is optional then control behavior explicitly:

```
option(FOO_WITH_OPENSSL "Build with OpenSSL" ON)

if(FOO_WITH_OPENSSL)
  find_package(OpenSSL REQUIRED) # fatal error if not found!
  target_compile_definitions(... PUBLIC FOO_WITH_OPENSSL=1)
endif()
```

Same with the programs:

```
find_program(PYTHON_EXE python) # Use 'find_package(PythonInterp)' in real code
if(PYTHON_EXE)
  # generate some extra code
endif()
```

Use this code instead:

```
option(FOO_WITH_PYTHON "Build with Python" ON)

if(FOO_WITH_PYTHON)
  find_program(PYTHON_EXE python)
  if(NOT PYTHON_EXE)
    message(FATAL_ERROR "Python not found")
  endif()
endif()
```

Environment variable example:

```
if(EXISTS "$ENV{FOO_EXTRA_CODE}")
  # add some code
endif()
```

Solution:

```
option(FOO_WITH_EXTRA_CODE "Use extra code" ON)

if(FOO_WITH_EXTRA_CODE)
  if(NOT EXISTS "$ENV{FOO_EXTRA_CODE}")
    message(FATAL_ERROR "...")
  endif()
endif()
```

Note that this is kind of a natural limitation because otherwise Hunter have to save the whole outside environment like default paths, environment variables, etc. This is not doable on practice.

Exception is the variables related to compiler/toolchain like compiler version, compiler id, platforms, generators, architectures: `WIN32`, `IOS`, `ANDROID`, etc. Number of such traits is limited and forms *toolchain-id*.

CGold

- Depending on environment variable

Install XXXConfig.cmake

The easiest way to integrate installed libraries into other project is to use find_package command. Project should generate and install `*Config.cmake` files instead of using `Find*.cmake` modules. It's the one of the painless ways to support relocation - imported targets can be cached and downloaded as prebuilt binary archive from build servers. Plus only imported targets works nicely with non standard build types like `MinSizeRel` or `RelWithDebInfo`.

To check this feature you can try to install files to local directory. If result of installation looks like this:

```
Install the project...
/.../cmake -P cmake_install.cmake
-- Install configuration: "Release"
-- Installing: /.../lib/libhunter_box_1.a
-- Installing: /.../include/hunter_box_1.hpp
```

It means that this feature is missing and you need to *patch CMake code* to introduce it. Details can be found here.

Installation after fix:

```
Install the project...
/.../cmake -P cmake_install.cmake
-- Install configuration: "Release"
-- Installing: /.../lib/libhunter_box_1.a
-- Installing: /.../include/hunter_box_1.hpp
-- Installing: /.../lib/cmake/hunter_box_1/hunter_box_1Config.cmake
-- Installing: /.../lib/cmake/hunter_box_1/hunter_box_1ConfigVersion.cmake
-- Installing: /.../lib/cmake/hunter_box_1/hunter_box_1Targets.cmake
-- Installing: /.../lib/cmake/hunter_box_1/hunter_box_1Targets-release.cmake
```

CGold

- Rejected: FindXXX.cmake
- Install layout

CMake documentation

- find_package
- cmake-packages

Add package to Hunter

Next let's assume user hunterbox is trying to add hunter_box_1 project to Hunter.

Examples on GitHub

- Example: hunterbox/hunter_box_1

Recommended name for the package is lowercase separated with underscore.

C++:

```
#include <hunter_box_1/hunter_box_1.hpp>

int main() {
  hunter_box_1::foo();
}
```

```
// file hunter_box_1.hpp

namespace hunter_box_1 {
} // namespace hunter_box_1
```

CMake with Hunter:

```
hunter_add_package(hunter_box_1)
find_package(hunter_box_1 CONFIG REQUIRED)
target_link_libraries(... hunter_box_1::hunter_box_1)
```

In Hunter sources:

- `cmake/projects/hunter_box_1/hunter.cmake` file with versions
- `examples/hunter_box_1` directory with example for testing
- `docs/packages/pkg/hunter_box_1.rst` documentation for package

Fork Hunter

Hunter hosted on GitHub service where common way to add code is to fork project and create pull request.

Fork cpp-pm/hunter, clone your fork and initialize all submodules:

```
> git clone https://github.com/hunterbox/hunter
> cd hunter
[hunter]> git submodule update --init --recursive .
```

Create branch to work on new package:

```
[hunter]> git checkout -b pr.hunter_box_1
```

Add versions

Add one or several versions of `hunter_box_1` package to corresponding `hunter.cmake` file.

Copy template and substitute all strings `foo` to `hunter_box_1`:

```
[hunter]> cp -r cmake/projects/foo cmake/projects/hunter_box_1
[hunter]> sed -i 's,foo,hunter_box_1,g' cmake/projects/hunter_box_1/hunter.cmake
```

Download release archive and calculate `SHA1`:

```
> wget https://github.com/hunterbox/hunter_box_1/archive/v1.0.0.tar.gz
> openssl sha1 v1.0.0.tar.gz
SHA1(v1.0.0.tar.gz)= c724e0f8a4ebc95cf7ba628b89b998b3b3c2697d
```

Add this information to `cmake/projects/hunter_box_1/hunter.cmake` file:

```
# !!! DO NOT PLACE HEADER GUARDS HERE !!!

include(hunter_add_version)
include(hunter_cacheable)
include(hunter_download)
include(hunter_pick_scheme)

hunter_add_version(
    PACKAGE_NAME
    hunter_box_1
    VERSION
    1.0.0
    URL
    "https://github.com/hunterbox/hunter_box_1/archive/v1.0.0.tar.gz"
    SHA1
    c724e0f8a4ebc95cf7ba628b89b998b3b3c2697d
)

hunter_pick_scheme(DEFAULT url_sha1_cmake)
hunter_cacheable(hunter_box_1)
hunter_download(PACKAGE_NAME hunter_box_1)
```

Consistency

Please keep Git tag and `VERSION` in consistent state. For example if `URL` is:

```
hunter_add_version(
    # ...
    URL
    "https://github.com/hunterbox/hunter_box_1/archive/v1.3.15-da39a3e-p6.tar.gz"
    # ...
)
```

Then `VERSION` should be:

```
hunter_add_version(
    # ...
    VERSION
    1.3.15-da39a3e-p6
    URL
    "https://github.com/hunterbox/hunter_box_1/archive/v1.3.15-da39a3e-p6.tar.gz"
    # ...
)
```

CMake options

Note that it does not make sense to build and install stuff like examples, tests or documentation. Please check that your package has CMake options to disable those. If such an option is not disabled by default use `hunter_cmake_args`:

```
include(hunter_cmake_args)

# ...

# bottom of cmake/projects/foo/hunter.cmake
hunter_cmake_args(
    foo
    CMAKE_ARGS
        FOO_BUILD_EXAMPLES=OFF
        FOO_BUILD_TESTS=OFF
        FOO_BUILD_DOCUMENTATION=OFF
)

hunter_pick_scheme(DEFAULT url_sha1_cmake)
hunter_download(PACKAGE_NAME foo)
```

Options set by `hunter_cmake_args` have lower precedence than options set by `hunter_config(... CMAKE_ARGS ...)` (see order).

Build types

> **Warning:** Usually there is no need to set a build type explicitly. If the package does not work with default `Debug` + `Release` it means something is wrong with the package itself.

Default build type(s) can be set by `hunter_configuration_types`:

```
hunter_configuration_types(foo CONFIGURATION_TYPES Release)
hunter_download(PACKAGE_NAME foo)
```

User can overwrite this default by using custom `hunter_config` parameters.

Set default version

Add `hunter_default_version` directive with default version to `cmake/configs/default.cmake`:

```
hunter_default_version(hunter_box_1 VERSION 1.0.0)
```

Create example

To test the integration of the package into another project a simple example will be used. Copy the template example and substitute all strings `foo` with `hunter_box_1`:

```
[hunter]> cp -r examples/foo examples/hunter_box_1
[hunter]> sed -i 's,foo,hunter_box_1,g' examples/hunter_box_1/*
```

Tweak all files in `examples/hunter_box_1` directory to fit headers and names of imported targets.

Add documentation

Each package should have a *page with information and usage example*.

To create such a page copy the template file and substitute all strings `foo` with the project name (for example `hunter_box_1`):

```
[hunter]> cp docs/packages/pkg/foo.rst docs/packages/pkg/hunter_box_1.rst
[hunter]> sed -i 's,foo,hunter_box_1,g' docs/packages/pkg/hunter_box_1.rst
```

Open file `docs/packages/pkg/hunter_box_1.rst` and tweak all entries.

Substitute `unsorted` with some tag in directive `single: unsorted ; foo`. This tag will be used on this page.

If you want to have two tags add another line with `single`:

```
.. index::
  single: category_1 ; foo
  single: category_2 ; foo
```

See also:

- Gentoo packages
- Ubuntu packages

Note: Since you don't know the pull request number a priori leave it as `N` for now. You can update it later.

Commit

Now save all changes by doing a commit:

```
[hunter]> git branch
  master
* pr.hunter_box_1

[hunter]> git add cmake/configs/default.cmake
[hunter]> git add cmake/projects/hunter_box_1/
[hunter]> git add docs/packages/pkg/hunter_box_1.rst
[hunter]> git add examples/hunter_box_1/

[hunter]> git commit -m "Add 'hunter_box_1' package"
```

Test package

Hunter uses GitHub Actions for *continuous integration testing*. You can also test *package building* and *documentation* locally, however this is optional.

Testing will be performed automatically on pull request. To perform testing on push to your Hunter fork, ensure that GitHub Actions are enabled for your repository - Managing GitHub Actions.

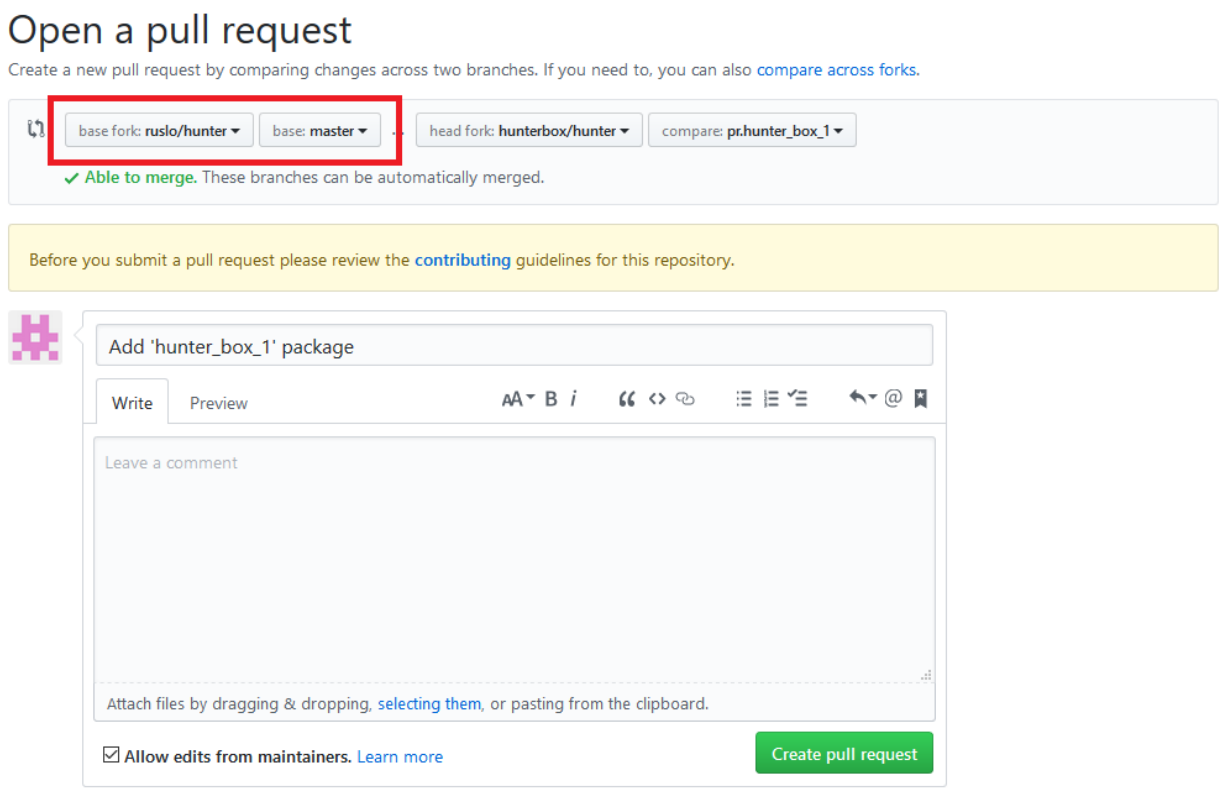
Package build testing will be performed for multiple platforms (different toolchains). If some toolchains are working and some toolchains failed it means the project has platform-specific problems. Note that you don't have to have all toolchains working and there is **no need to fix all issues you see**. If **at least documentation test is passing** and *some toolchain tests are working* you can make a pull request and you or somebody else can apply fixes later.

If you're sure that testing is failing due to system specific requirements and NOT due to package dependencies or platform specific code errors, or your package contains components and needs to perform some special tests with different examples - you can *modify default build matrix and scripts*.

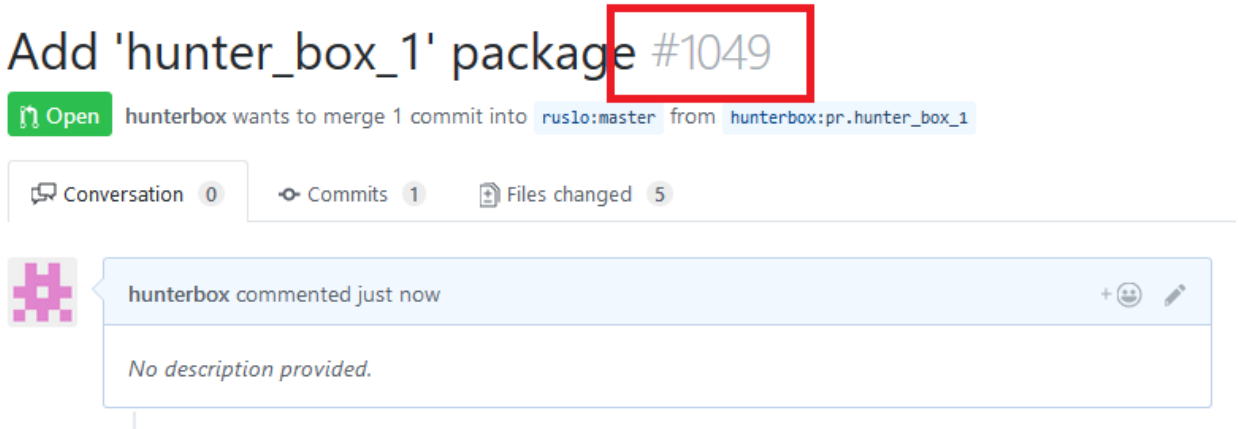
Pull requests

After CI testing is done you can open a pull request with package:

```
[hunter]> git checkout pr.hunter_box_1
[hunter]> git push -u origin pr.hunter_box_1
```



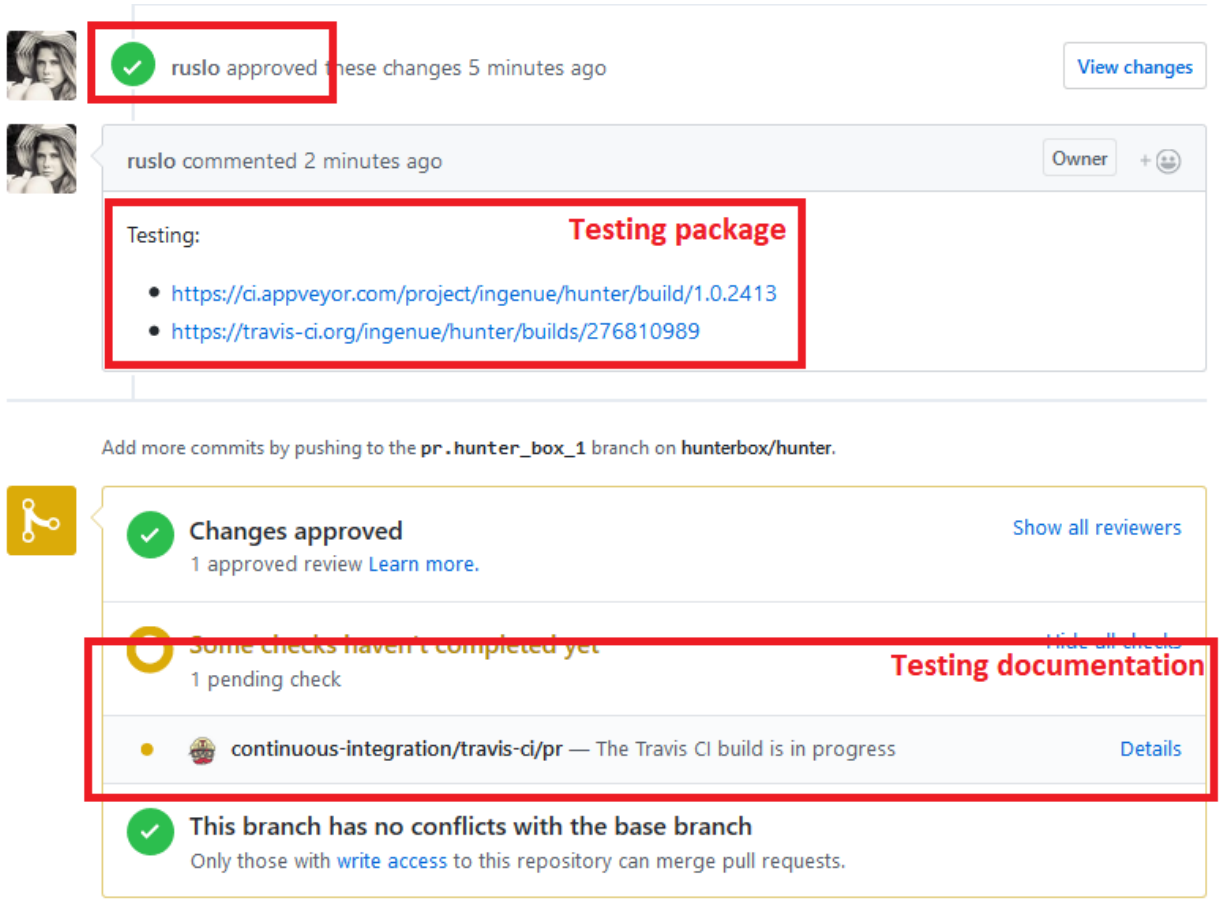
At this moment you know the pull request number:



Add it to documentation:

```
[hunter]> git checkout pr.hunter_box_1
[hunter]> vim docs/packages/pkg/hunter_box_1.rst
[hunter]> git add docs/packages/pkg/hunter_box_1.rst
[hunter]> git commit -m 'Pull request number'
[hunter]> git push
```

Pull request will be approved and tests run on CI, documentation will be tested automatically:



Release

After all tests pass the pull request will be merged. New release will be created:

You can use new `URL`/`SHA1`:

Clean

At this moment working branch can be removed:

```
[hunter]> git checkout master
[hunter]> git push origin :pr.hunter_box_1
[hunter]> git branch -D pr.hunter_box_1
```

Badge

Badge in `README.rst` can signal that package `hunter_box_1` is available via Hunter:

```
|hunter|

.. |hunter| image:: https://img.shields.io/badge/hunter-hunter_box_1-blue.svg
  :target: https://hunter.readthedocs.io/en/latest/packages/pkg/hunter_box_1.html
  :alt: Hunter
```

Example:

- https://github.com/hunter-packages/gauze/blob/master/README.rst

5.1.2 CMake (with dependencies)

If your project uses external packages (i.e. has command `find_package(Foo)`) you need to patch it first so these packages can be found in the Hunter root directory instead of the standard one:

```
hunter_add_package(Foo)
find_package(Foo)
```

Note:

- *Patching sources (hunter-packages)*

Conflict

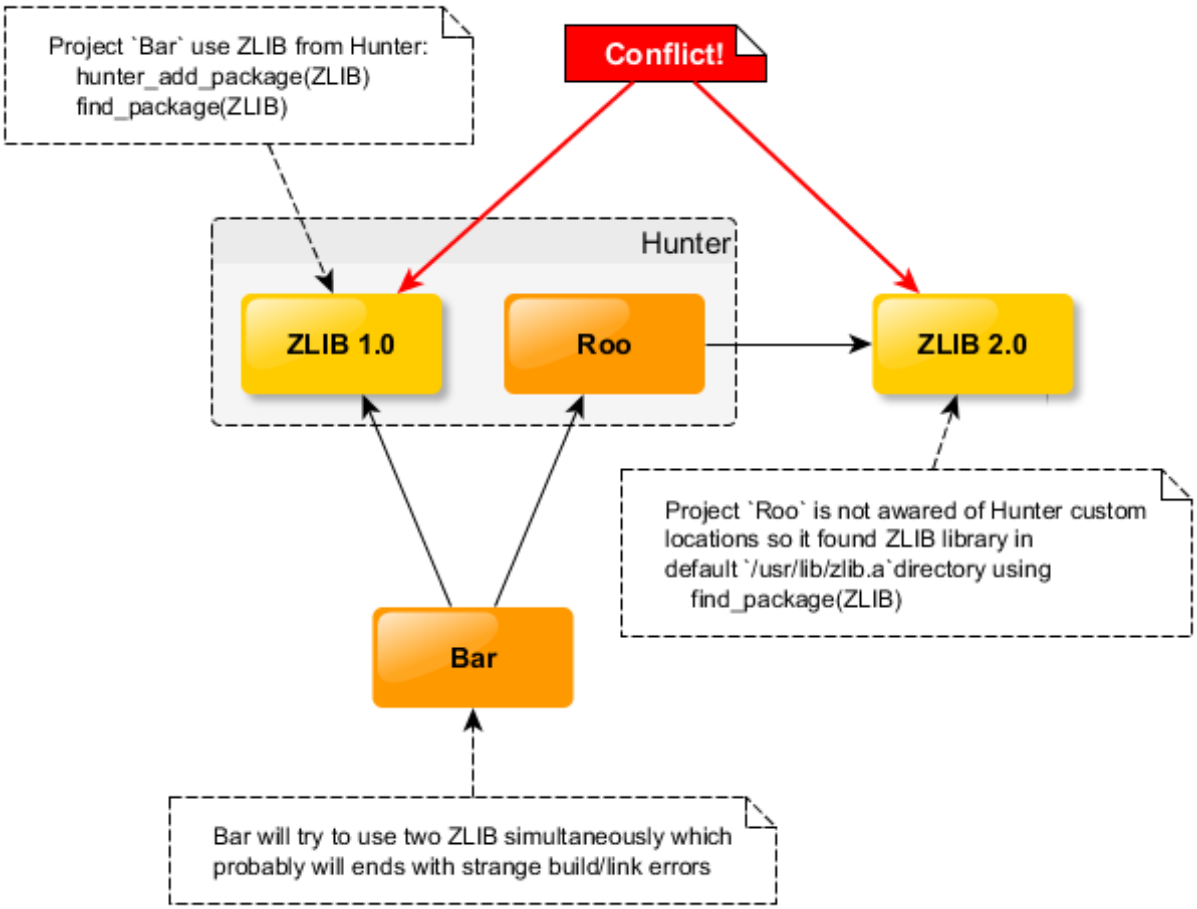
Without the `hunter_add_package(Foo)` call one package will be found in the standard location and another one in the Hunter root directory. The found packages may conflict with each other.

Consider the next example: Project `Roo` is not aware about Hunter custom locations. It's just using regular `find_package`:

```
# Roo/CMakeLists.txt

find_package(ZLIB)
```

Project `Bar` depends on `ZLIB` and `Roo`. Both packages are downloaded by `hunter_add_package` commands:

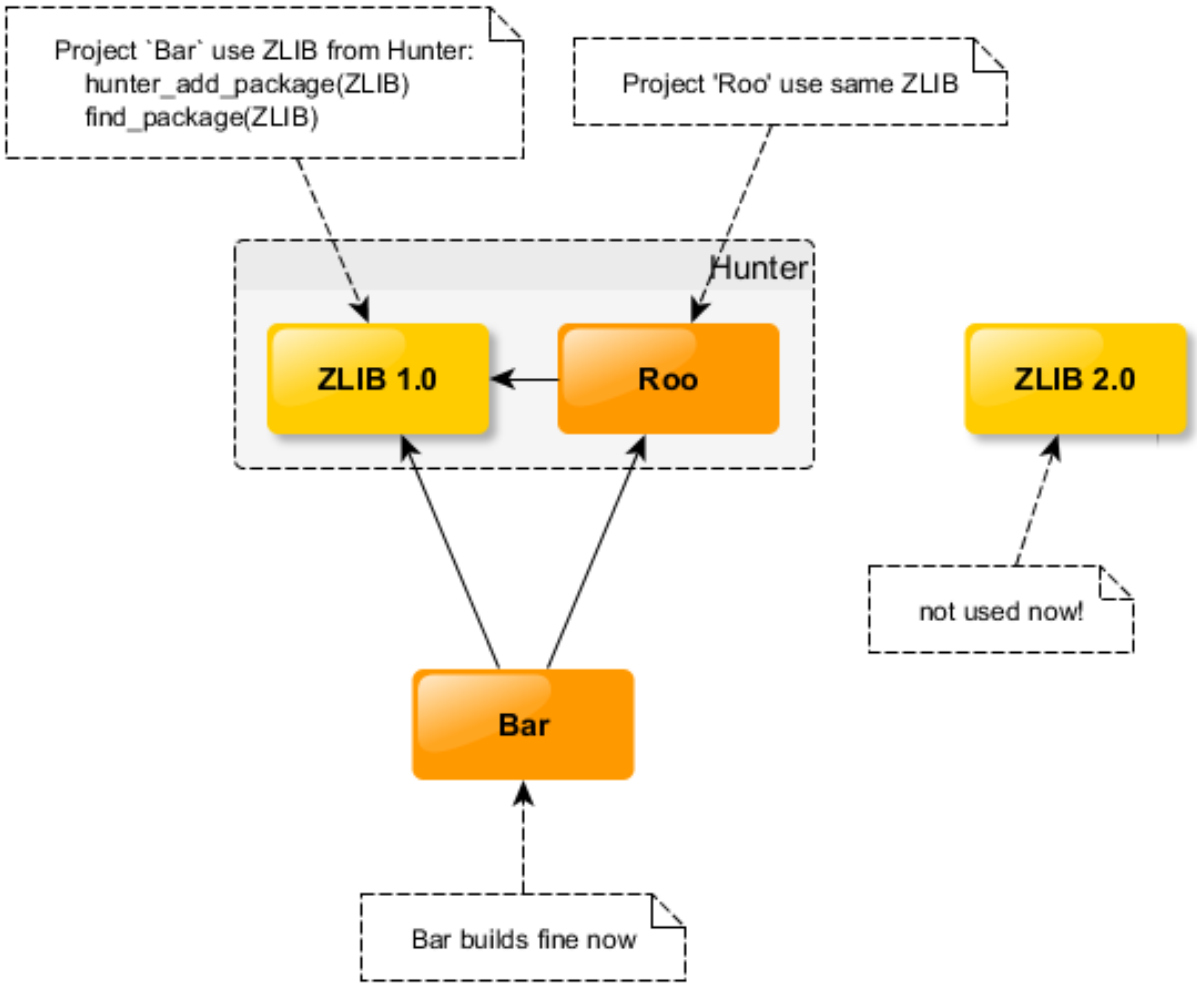
```
# Bar/CMakeLists.txt

hunter_add_package(Roo)
find_package(Roo)

hunter_add_package(ZLIB)
find_package(ZLIB)
```



Fix

To fix this issue you need to patch project `Roo` so it will use `ZLIB` from Hunter. In terms of CMake code it means adding `HunterGate` and `hunter_add_package` (see *First Step*):

```
# Roo/CMakeLists.txt

include("cmake/HunterGate.cmake")
HunterGate(...)
hunter_add_package(ZLIB)
find_package(ZLIB CONFIG REQUIRED)
```



Note that now the main project `Bar` and the hunter dependency `Roo` contain a `HunterGate` command. The URL and SHA1 of the `HunterGate` command my not match. In this case the URL and SHA1 of the main project `Bar` are used for both `HunterGate` commands. The user does not need to manage them manually. The same is true for sub-projects added by `add_subdirectory` calls.

5.1.3 Autotools

Very often, you will come across a package that uses autotools as its build system and does not support CMake builds. Although Hunter prefers CMake builds when possible, it does support autotools projects when no CMake build is available. Here is how to do it.

```
# !!! DO NOT PLACE HEADER GUARDS HERE !!!

include(hunter_add_version)
include(hunter_configuration_types)
include(hunter_pick_scheme)
include(hunter_download)
include(hunter_cacheable)
include(hunter_cmake_args)

hunter_add_version(
```

```
    PACKAGE_NAME
    foo
    VERSION
    "1.2.3"
    URL
    "https://example.com/foo-1.2.3.tar.gz"
    SHA1
    da39a3ee5e6b4b0d3255bfef95601890afd80709
)

# More versions...

# Optional platform customization.
if (ANDROID OR IOS)
  hunter_cmake_args(
      foo
      CMAKE_ARGS
          EXTRA_FLAGS=--enable-x
  )
endif()

hunter_configuration_types(foo CONFIGURATION_TYPES Release)
hunter_pick_scheme(DEFAULT url_sha1_autotools)
hunter_cacheable(foo)
hunter_download(PACKAGE_NAME foo)
```

Note that the build may not be cacheable if autotools generation expands absolute paths. Try using `hunter_cacheable` and see if it works.

Many autotools projects generate pkg-config files. These can be used to generate a CMake config. For example, consider using the following in your package's `hunter.cmake` file:

```
hunter_cmake_args(
    foo
    CMAKE_ARGS
        PKGCONFIG_EXPORT_TARGETS=foo
)
```

In the above example, package `foo` generates a file `foo.pc` in the autotools build. Hunter then uses `foo.pc` to generate a CMake config file `fooConfig.cmake`. Now, our dependent project `Bar` has a much simpler `CMakeLists.txt`:

```
hunter_add_package(foo)
find_package(foo CONFIG REQUIRED)
add_executable(bar ${BAR_SOURCES})
target_link_libraries(bar PUBLIC PkgConfig::foo)
```

When following this pkg-config practice and attempting to keep `foo` cacheable, you must add this piece of code to your package's `hunter.cmake`:

```
hunter_download(PACKAGE_NAME foo
    PACKAGE_INTERNAL_DEPS_ID "1"  # Increment for each new pull request
    PACKAGE_UNRELOCATABLE_TEXT_FILES
    lib/pkgconfig/foo.pc)
```

The pkg-config files will probably need to be patched so that they do not point to the directory they are initially installed into. `PACKAGE_UNRELOCATABLE_TEXT_FILES` identifies these files for Hunter to patch.

If the autotools build does not produce a pkg-config output file, you must add `Findfoo.cmake` place it in the `cmake/find` directory so Hunter can find the package. This script should also provide import targets for dependent builds, such that linking against `foo::foo` pulls in the foo includes and libraries. In this case, dependent projects will use code similar to the following:

```
hunter_add_package(foo)
find_package(foo REQUIRED)
add_executable(bar ${BAR_SOURCES})
target_link_libraries(bar PUBLIC foo::foo)
```

Extra flags for configure

It is possible to add extra flags for `./configure` step both globally in `cmake/projects/<package>/hunter.cmake`:

```
hunter_cmake_args(
    foo
    CMAKE_ARGS
        EXTRA_FLAGS=--enable-x
)
```

and locally in `cmake/Hunter/config.cmake`:

```
hunter_config(
    foo
    VERSION
        ${HUNTER_foo_VERSION}
    CMAKE_ARGS
        EXTRA_FLAGS=--enable-y
)
```

If you use local approach then any flags from global configuration will be ignored, i.e. if you want to have both global `--enable-x` and local `--enable-y` then you have to set them explicitly:

```
hunter_config(
    foo
    VERSION
        ${HUNTER_foo_VERSION}
    CMAKE_ARGS
        EXTRA_FLAGS=--enable-x --enable-y
)
```

5.1.4 Non-CMake: custom scheme

Non-CMake projects can be added too. But sometimes it's not a trivial task (for example there are a 3 custom schemes for OpenSSL. In general it's better to apply a patch to an existing CMake build and use *CMake (no dependencies)* add instruction. Anyway here is a guide how to add a project with custom build:

Test it manually

```
> wget https://github.com/phonegap/ios-sim/archive/1.8.2.tar.gz
> openssl sha1 1.8.2.tar.gz
SHA1(1.8.2.tar.gz)= 4328b3c8e6b455631d52b7ce5968170c9769eb1e
```

```
> tar xf 1.8.2.tar.gz
> cd ios-sim-1.8.2/
> xcodebuild -target ios-sim -configuration Release
> ls build/Release/ios-sim
build/Release/ios-sim
```

Test it using `ExternalProject_Add`

```
> cat CMakeLists.txt
cmake_minimum_required(VERSION 3.2)

include(ExternalProject) # ExternalProject_Add

ExternalProject_Add(
    ios_sim
    URL
    "https://github.com/phonegap/ios-sim/archive/1.8.2.tar.gz"
    URL_HASH
    SHA1=4328b3c8e6b455631d52b7ce5968170c9769eb1e
    CONFIGURE_COMMAND
    ""
    BUILD_COMMAND
    xcodebuild -target ios-sim -configuration Release
    BUILD_IN_SOURCE
    1
    INSTALL_COMMAND
    "${CMAKE_COMMAND}" -E make_directory "${CMAKE_INSTALL_PREFIX}"
    COMMAND
    "${CMAKE_COMMAND}" -E copy build/Release/ios-sim "${CMAKE_INSTALL_PREFIX}"
)
> cmake -H. -B_builds -DCMAKE_INSTALL_PREFIX=`pwd`/_install
> cmake --build _builds/
> ls _install/
ios-sim
```

Add new package

First, custom build scheme need to be added to `cmake/schemes` directory:

```
> cd ${HUNTER_ROOT}
> cat cmake/schemes/url_sha1_ios_sim.cmake.in
# This is configuration file, variable @SOME_VARIABLE_NAME@ will be substituted␣
↪during configure_file command
cmake_minimum_required(VERSION 3.2)

# If such variables like `CMAKE_CXX_FLAGS` or `CMAKE_CXX_COMPILER` not used by scheme
# setting `LANGUAGES` to `NONE` will speed-up build a little bit. If you have any␣
↪problems/glitches
# use regular `project(Hunter)` command
project(Hunter LANGUAGES NONE)

include(ExternalProject) # ExternalProject_Add

# some Hunter modules will be used
list(APPEND CMAKE_MODULE_PATH "@HUNTER_SELF@/cmake/modules")
```

```
include(hunter_status_debug)
include(hunter_assert_not_empty_string)

# print this message if HUNTER_STATUS_DEBUG option is ON
hunter_status_debug("Scheme: url_sha1_ios_sim")

# Check variables is not empty
hunter_assert_not_empty_string("@HUNTER_SELF@")
hunter_assert_not_empty_string("@HUNTER_EP_NAME@")
hunter_assert_not_empty_string("@HUNTER_PACKAGE_URL@")
hunter_assert_not_empty_string("@HUNTER_PACKAGE_SHA1@")
hunter_assert_not_empty_string("@HUNTER_PACKAGE_DOWNLOAD_DIR@")
hunter_assert_not_empty_string("@HUNTER_PACKAGE_SOURCE_DIR@")
hunter_assert_not_empty_string("@HUNTER_INSTALL_PREFIX@")

ExternalProject_Add(
    @HUNTER_EP_NAME@ # Name of the external project. Actually not used set for␣
↪beautify logging messages
    URL
    @HUNTER_PACKAGE_URL@ # URL of the package to download
    URL_HASH
    SHA1=@HUNTER_PACKAGE_SHA1@ # SHA1 hash
    DOWNLOAD_DIR
    "@HUNTER_PACKAGE_DOWNLOAD_DIR@" # Archive destination location
    SOURCE_DIR
    "@HUNTER_PACKAGE_SOURCE_DIR@" # Unpack directory
    INSTALL_DIR
    "@HUNTER_INSTALL_PREFIX@" # not used actually (see install command)
    CONFIGURE_COMMAND
    ""
    BUILD_COMMAND
    xcodebuild -target ios-sim -configuration Release
    BUILD_IN_SOURCE
    1
    INSTALL_COMMAND
    "@CMAKE_COMMAND@" -E copy build/Release/ios-sim "@HUNTER_INSTALL_PREFIX@"
)
```

Next steps are similar to *CMake (no dependencies)*.

```
> cat cmake/projects/ios_sim/hunter.cmake

# !!! DO NOT PLACE HEADER GUARDS HERE !!!

include(hunter_add_version)
include(hunter_download)
include(hunter_pick_scheme)

hunter_add_version(
    PACKAGE_NAME
    ios_sim
    VERSION
    "1.8.2"
    URL
    "https://github.com/phonegap/ios-sim/archive/1.8.2.tar.gz"
    SHA1
    4328b3c8e6b455631d52b7ce5968170c9769eb1e
```

```
)

hunter_pick_scheme(DEFAULT url_sha1_ios_sim) # Use new custom scheme
hunter_download(PACKAGE_NAME ios_sim)
```

```
> grep ios_sim cmake/config/default.cmake
hunter_default_version(ios_sim VERSION 1.8.2)
```

Using

Now package ready to be used:

```
> cat CMakeLists.txt
cmake_minimum_required(VERSION 3.2)

include("cmake/HunterGate.cmake")

HunterGate(
    URL "https://url/to/your/hunter-archive.tar.gz"
    SHA1 "put-archive-sha1-here"
)

hunter_add_package(ios_sim)

find_program(IOS_SIM_EXECUTABLE ios-sim ${IOS_SIM_ROOT})
message("ios_sim: ${IOS_SIM_EXECUTABLE}")
> cmake -H. -B_builds
-- [hunter] HUNTER_ROOT: /.../Hunter
-- [hunter] [ Hunter-ID: 7912489 | Config-ID: 9ec2ff8 | Toolchain-ID: c018e63 ]
-- [hunter] IOS_SIM_ROOT: /.../Hunter/_Base/7912489/9ec2ff8/c018e63/Install (ver.: 1.
↪8.2)
...
-- downloading...
     src='https://github.com/phonegap/ios-sim/archive/1.8.2.tar.gz'
-- [download 100% complete]
ios_sim: /.../Hunter/_Base/7912489/9ec2ff8/c018e63/Install/ios-sim
```

Default behavior

Note that such CMake variables like:

- CMAKE_{C,CXX}_FLAGS
- CMAKE_{C,CXX}_COMPILER
- CMAKE_CONFIGURATION_TYPES
- BUILD_SHARED_LIBS

must be checked manually for each custom build scheme (see *CMake (no dependencies)*).

5.2 Update package

Note: If package lives in https://github.com/hunter-packages, it should be released there first. Check *Patch sources* section.

Create branch for working on package update:

```
[hunter]> git checkout master
[hunter]> git checkout -b pr.hunter_box_1
```

Calculate SHA1 of release:

```
> wget https://github.com/hunterbox/hunter_box_1/archive/v1.0.1.tar.gz
> openssl sha1 v1.0.1.tar.gz
SHA1(v1.0.1.tar.gz)= 10d046eec6c8b0aabd28bd3d1b99faf6beeb226b
```

Add URL and SHA1 to corresponding `hunter.cmake`:

```
--- /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/
↪creating-new/hunter.cmake
+++ /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/
↪creating-new/hunter-NEW.cmake
@@ -19,6 +19,17 @@
     4fa7fe75629f148a61cedc6ba0bce74f177a6747
 )

+hunter_add_version(
+    PACKAGE_NAME
+    hunter_box_1
+    VERSION
+    1.0.1
+    URL
+    "https://github.com/hunterbox/hunter_box_1/archive/v1.0.1.tar.gz"
+    SHA1
+    10d046eec6c8b0aabd28bd3d1b99faf6beeb226b
+)
+
 hunter_pick_scheme(DEFAULT url_sha1_cmake)
 hunter_cacheable(hunter_box_1)
 hunter_download(PACKAGE_NAME hunter_box_1)
```

Hint: Put new `hunter_add_version` at the bottom of file, diff will look prettier in this case.

Update default version in `cmake/configs/default.cmake`:

```
--- /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/
↪creating-new/default.cmake
+++ /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/
↪creating-new/default-NEW.cmake
@@ -181,7 +181,7 @@
 hunter_default_version(glproto VERSION 1.4.17)
 hunter_default_version(half VERSION 1.1.0-p1)
 hunter_default_version(hdf5 VERSION 1.8.15-p1)
-hunter_default_version(hunter_box_1 VERSION 1.0.0)
+hunter_default_version(hunter_box_1 VERSION 1.0.1)
 hunter_default_version(ice VERSION 1.0.8)
```

```
hunter_default_version(imshow VERSION 1.0.0-p0)
hunter_default_version(inputproto VERSION 2.2)
```

Commit changes:

```
[hunter]> git add cmake/projects/hunter_box_1/hunter.cmake
[hunter]> git add cmake/configs/default.cmake
[hunter]> git commit -m "Update 'hunter_box_1' to v1.0.1"
```

5.2.1 Test package

Hunter uses GitHub Actions for *continuous integration testing*. You can also test *package building* and *documentation* locally, however this is optional.

Testing will be performed automatically on pull request. To perform testing on push to your Hunter fork, ensure that GitHub Actions are enabled for your repository - Managing GitHub Actions.

Package build testing will be performed for multiple platforms (different toolchains). If some toolchains are working and some toolchains failed it means the project has platform-specific problems. Note that you don't have to have all toolchains working and there is **no need to fix all issues you see**. If **at least documentation test is passing** and *some toolchain tests are working* you can make a pull request and you or somebody else can apply fixes later.

If you're sure that testing is failing due to system specific requirements and NOT due to package dependencies or platform specific code errors, or your package contains components and needs to perform some special tests with different examples - you can *modify default build matrix and scripts*.

5.2.2 Submit the pull request to Hunter

Once tests are passing, the package update in `pr.hunter_box_1` should be pushed to your Hunter fork:

```
[hunter]> git push -u origin pr.hunter_box_1
```

Finally, a pull request should be opened to send the package update to the main Hunter repository, as illustrated in the previous section pull request screen shot (see example).

5.3 Test package

Hunter uses GitHub Actions for *continuous integration testing*. You can also test *package building* and *documentation* locally, however this is optional.

Testing will be performed automatically on pull request. To perform testing on push to your Hunter fork, ensure that GitHub Actions are enabled for your repository - Managing GitHub Actions.

Package build testing will be performed for multiple platforms (different toolchains). If some toolchains are working and some toolchains failed it means the project has platform-specific problems. Note that you don't have to have all toolchains working and there is **no need to fix all issues you see**. If **at least documentation test is passing** and *some toolchain tests are working* you can make a pull request and you or somebody else can apply fixes later.

If you're sure that testing is failing due to system specific requirements and NOT due to package dependencies or platform specific code errors, or your package contains components and needs to perform some special tests with different examples - you can *modify default build matrix and scripts*.

5.3.1 CI testing

Refer to GitHub Actions Documentation for better understanding of Hunter CI testing.

Two types of tests are performed, and appropriate workflows run:

1. **Documentation testing.**

- This workflow will run if any file in `docs` or `examples` directory has been changed.
- This is done to ensure that documentation is building correctly.

2. **Package build testing with various toolchains.**

- This workflow will run if any file in `cmake/projects` directory has been changed.
- Default toolchains for tests are:
  - Windows: Visual Studio, NMake, Ninja, MinGW, MSYS
  - Linux: GCC, Clang, Android, Clang Analyzer, Sanitize Address, Sanitize Leak
  - macOS: Clang + Makefile, Xcode, iOS

Override default tests

GitHub Actions workflow files are located in `.github/workflows`:

> **Warning: Please don't modify these files.** Review them to understand test steps.

- `ci-docs.yml` - workflow file for testing documentation
  - Checks if files in `docs` or `examples` directories have been changed
  - If that's the case, runs *documentation testing* on GitHub Ubuntu runner
- `ci.yml` - workflow file for package build testing
  - Checks which files in `cmake/projects` directory have been changed
  - Sets up *build matrix* accordingly
  - Runs builds on GitHub-hosted runners
  - Uploads jobs status to GitHub Pages - Pakages status
- `set_matrix.py` - script to perform build strategy matrix manipulation
  - Checks if package has overridden build matrix
  - If not, substitutes **example** property of the default matrix with project's name
  - Checks if package has a build script override and sets build script accordingly
  - Merges build matrices if multiple projects are tested
- `set_status.py` - script to perform manipulations with job's status .json
  - Splits job's .json if multiple projects were tested in one workflow run
  - Sorts by toolchain alphabetically

Default package build strategy matrix and scripts are located in `.github/workflows/ci`:

> **Warning: Please don't modify these files.** Instead create a `ci` subdirectory in your package directory and copy files, that you need to change, there.

- `matrix.json` - *include* part of GitHub Actions build strategy matrix
- `build.sh` - build script for *nix systems
- `build.cmd` - build script for Windows

Build matrix

> **Warning:** Don't copy and modify the default matrix if your package doesn't have components and you only need to change build scrips. This will lead to you project testing toolchains diverge from default ones in the future.

```
[
{ "example": "foo", "toolchain": "clang-cxx17",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "gcc-7-cxx17",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "gcc-8-cxx17-fpic",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "gcc-9-cxx17-fpic",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "android-ndk-r17-api-24-arm64-v8a-clang-libcxx14",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "analyze-cxx17",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "sanitize-address-cxx17",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "sanitize-leak-cxx17",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "sanitize-thread-cxx17",
↪"os": "ubuntu-20.04", "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "osx-10-15-make-cxx14",
↪"os": "macos-10.15",  "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "osx-12-3-arch-universal2-cxx17",
↪"os": "macos-12",     "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "ios-nocodesign-15-5-arm64-cxx17",
↪"os": "macos-12",     "python": "3.8", "script": "build.sh"  },
{ "example": "foo", "toolchain": "ninja-vs-16-2019-win64-cxx17",
↪"os": "windows-2019", "python": "3.8", "script": "build.cmd" },
{ "example": "foo", "toolchain": "nmake-vs-16-2019-win64-cxx17",
↪"os": "windows-2019", "python": "3.8", "script": "build.cmd" },
{ "example": "foo", "toolchain": "vs-16-2019-win64-sdk-10-0-18362-0-cxx17",
↪"os": "windows-2019", "python": "3.8", "script": "build.cmd" },
{ "example": "foo", "toolchain": "vs-16-2019-win64-store-10-cxx17",
↪"os": "windows-2019", "python": "3.8", "script": "build.cmd" },
{ "example": "foo", "toolchain": "mingw-cxx17",
↪"os": "windows-2019", "python": "3.8", "script": "build.cmd" },
{ "example": "foo", "toolchain": "msys-cxx17",
↪"os": "windows-2019", "python": "3.8", "script": "build.cmd" }
]
```

Each line defines parameters for a job that will run on GitHub-hosted runner:

- **example** - subdirectory name in the `examples` directory. You need to change the default value `foo` to your project's (or project component's) example directory name
- **toolchain** - Polly toolchain
- **os** - Supported GitHub-hosted runner. Set this according to toolchain.
- **python** - Python version installed on the runner. Change this if your project uses Python scripts (that require specific Python version) for testing.
- **script** - Build script executed on the runner. The path of the script is relative to directory where `matrix.json` is located. If the script with defined name was not found in this path, default path `.github/workflows/ci` will be used.

Example matrix override:

A part of `cmake/projects/Boost/ci/matrix.json`:

```
[
{ "example": "Boost",               "toolchain": "clang-cxx17",              "os":
↪"ubuntu-16.04", "python": "3.8", "script": "build.sh" },
{ "example": "Boost-python",        "toolchain": "gcc-7-cxx17",              "os":
↪"ubuntu-16.04", "python": "3.5", "script": "build.sh" },
{ "example": "Boost-python-numpy", "toolchain": "gcc-7-cxx17",              "os":
↪"ubuntu-16.04", "python": "3.5", "script": "build-add-virtualenv.sh" },
{ "example": "Boost",               "toolchain": "vs-15-2017-win64-cxx17",   "os":
↪"windows-2016", "python": "3.8", "script": "build.cmd" },
{ "example": "Boost-python",        "toolchain": "vs-14-2015-win64-sdk-8-1", "os":
↪"windows-2016", "python": "3.5", "script": "build.cmd" },
{ "example": "Boost-python-numpy", "toolchain": "vs-14-2015-win64-sdk-8-1", "os":
↪"windows-2016", "python": "3.5", "script": "build-add-virtualenv.cmd" }
]
```

Build scripts

Scripts are executed in the Hunter's root directory.

Software installed on GitHub-hosted runners

Environment variables:

- **TOOLCHAIN** - build matrix's **toolchain** parameter
- **PROJECT_DIR** - **example** parameter

Default build script for Ubuntu and macOS runners is `.github/workflows/ci/build.sh` (bash script)

```
# Set the correct Python PATH
export PATH="${HUNTER_PYTHON_LOCATION}:${PATH}"

# Install Python package 'requests'
python -m pip install --upgrade pip
python -m pip install requests gitpython

# Install latest Polly toolchains and scripts
wget https://github.com/cpp-pm/polly/archive/master.zip
unzip master.zip
POLLY_ROOT="`pwd`/polly-master"
export PATH="${POLLY_ROOT}/bin:${PATH}"
```

```
# Install dependencies (CMake, Android NDK)
install-ci-dependencies.py --prune-archives

# Tune locations
export PATH="`pwd`/_ci/cmake/bin:${PATH}"

# Installed if toolchain is Android (otherwise directory doesn't exist)
export ANDROID_NDK_r10e="`pwd`/_ci/android-ndk-r10e"
export ANDROID_NDK_r11c="`pwd`/_ci/android-ndk-r11c"
export ANDROID_NDK_r15c="`pwd`/_ci/android-ndk-r15c"
export ANDROID_NDK_r16b="`pwd`/_ci/android-ndk-r16b"
export ANDROID_NDK_r17="`pwd`/_ci/android-ndk-r17"

# Use Xcode 13.4 for macOS 12.x and iOS 15.x toolchains
if [[ "$TOOLCHAIN" =~ "osx-12" || "$TOOLCHAIN" =~ "ios-nocodesign-15" ]]; then
    export DEVELOPER_DIR="/Applications/Xcode_13.4.app/Contents/Developer"
fi

# Run build script
if [[ "$BRANCH_NAME" == "master" && ! -z "$GITHUB_USER_PASSWORD" ]]; then
    python jenkins.py --upload
else
    python jenkins.py
fi
```

Default build script for Windows runner - `.github/workflows/ci/build.cmd` (batch file) is similar.

- installs Polly and all necessary dependencies
- defines default environment variables
- runs `jenkins.py` script to *test building of a project*.

Warning: If you don't need to alter Polly installation or predefined environment variables, don't copy and modify default script. Instead call the default script from your custom script, see example.

Examples of override build scripts:

Note: Set `matrix.json` job parameter according to the script name - f.e. **"script": "build-ubuntu.sh"**

for Ubuntu runner `cmake/projects/<PACKAGE_NAME>/ci/build-ubuntu.sh`:

```
export HUNTER_JOBS_NUMBER=1
pip install virtualenv
sudo apt-get install libgl1-mesa-dev

bash .github/workflows/ci/build.sh
```

for macOS `cmake/projects/<PACKAGE_NAME>/ci/build-macos.sh`

```
export HUNTER_JOBS_NUMBER=1
pip install virtualenv

bash .github/workflows/ci/build.sh
```

for Windows `cmake/projects/<PACKAGE_NAME>/ci/build.cmd`:

```
set HUNTER_JOBS_NUMBER=1
pip install virtualenv

.github/workflows/ci/build.cmd
```

5.3.2 Testing locally

This step is optional since we will run tests on the CI server. However it's the fastest way to check that everything is ready and working correctly.

Script `jenkins.py` will package a temporary Hunter archive based on current state and build the specified example. This script uses Polly toolchains.

Check you have Python 3 installed, clone Polly, add its `bin` folder to `PATH` environment variable, go back to Hunter repository and run test.

On Linux:

```
> which python3
/usr/bin/python3

> git clone https://github.com/cpp-pm/polly
> cd polly
[polly]> export PATH="`pwd`/bin:$PATH"

> cd hunter
[hunter]> which polly.py
/.../bin/polly.py

[hunter]> polly.py --help
Python version: 3.5
usage: polly.py [-h]
...

[hunter]> TOOLCHAIN=gcc PROJECT_DIR=examples/hunter_box_1 ./jenkins.py
```

On Windows:

```
> git clone https://github.com/cpp-pm/polly
> cd polly
[polly]> set PATH=%CD%\bin;%PATH%

> cd hunter
[hunter]> where polly.py
C:\...\bin\polly.py

[hunter]> polly.py --help
Python version: 3.5
usage: polly.py [-h]
...

[hunter]> set TOOLCHAIN=vs-12-2013
[hunter]> set PROJECT_DIR=examples\hunter_box_1
[hunter]> .\jenkins.py
```

Stackoverflow

- How to execute Python scripts in Windows?

Testing documentation locally

To locally check if the documentation is still building you can run:

```
[hunter]> cd docs
[hunter/docs]> source ./jenkins.sh
(_venv) [hunter/docs]> ./make.sh
```

If the documentation contains spelling errors or unrecognized names, the documentation test build will fail and report the unrecognized strings. Fix any spelling errors and test the build again. Any remaining errors can be fixed by adding all correct but unrecognized names, string, or terms to the `spelling` header at the top of the document entry `docs/packages/pkg/bar-baz.rst`. In this example, `bar-baz` would be a package name that is not in the dictionary.

```
.. spelling::

  bar
  baz

.. index::
  single: unsorted ; bar-baz

.. _pkg.bar-baz:
```

Add entries for each term until the test build completes successfully.

Common mistake

Please do not forget to substitute `===`.

Good:

```
hunter_box_1
============
```

Bad:

```
hunter_box_1
===
```

5.4 Patch sources

You may need to patch sources to apply CMake best practices or hunterize package with dependencies.

In practice patching requires to have a fork of a project. In general it does not matter where the fork is located. But it matters that there is a central place for the patched packages:

- https://github.com/cpp-pm

If you want to create new fork let me know about it in a corresponding issue with "new package" label, I will create a new team and add you so you can push changes.

Please follow next rules:

- Instead of pushing changes directly to branch, open **pull request** so other developers can review your changes.
- **Rebase** your changes, check that history has not merge commits. In this case it will be easier to do review and reuse your work in future.
- Start working on patches from latest stable upstream tag. I.e. if latest release is `v1.3.15` then create branch `hunter-1.3.15` and add patches there. If you want to have version that is not released yet, say `da39a3e`, then create branch `hunter-1.3.15-da39a3e`. If there are not tags in upstream then start with dummy `v0.0.0` version to avoid conflict with future releases, i.e. create branch `hunter-0.0.0-da39a3e`.
- Keep other branches in a **clean state** so we can always do `git merge --ff-only` from upstream.
- Please do push commits **only related to hunterization**. Do not push general fixes and improvements, do push them **upstream** instead. Perfect hunterization should contain only:
  - Adding `HunterGate` module (example)
  - Including it with some URL/SHA1 (example)
  - Adding `hunter_add_package` commands (example)
- **Test** your changes. Add temporary release to Hunter system and check that `hunter_add_package(foo)` is actually working. Do it at least for one toolchain *locally* but of course it will be better if you test all of them *remotely*.

Note that I'm not willing and can't maintain all packages in practice. Therefore I do add all developers to the team **if they ask to**. If you want to be a **maintainer**, keep eye on changes, pull requests, be responsible for review and releases - let me know.

Also note that Hunter is designed to have **zero maintenance** for such tasks, since you can add `HUNTER_ENABLED=OFF` option at the top of the project to skip all package management stuff (see *Backward compatibility*). It means you can push branch `hunter` to upstream without affecting functionality of the project. As a summary it may sounds strange, but the final goal of this organization is to have no forks of packages at all.

CHAPTER 6

FAQ

6.1 How to use Hunter in Android Studio?

CMake can be used as a build tool for native C/C++ libraries in Android Studio. If CMake project has third party dependencies these dependencies can be managed by Hunter.

6.1.1 Example

As an example let's take a look at a simple project with one tiny *md5 package* dependency. The project is a slight modification of the HelloJni sample.

Examples on GitHub

- Android Studio with Hunter

Note: The code was tested with Android Studio: 3.3, 3.4.1, 3.5 beta 2

Check you have at least CMake 3.9.2. Such a requirement needed to work with Android NDK r16+:

```
> cmake --version

cmake version 3.9.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

Check you have Ninja build tool installed:

```
> which ninja
/usr/bin/ninja
```

You can use your system package manager (e.g., on Ubuntu do `sudo apt-get install ninja-build`) or download it from GitHub releases, unpack and add to `PATH`:

- https://github.com/ninja-build/ninja/releases

Get the sources:

```
> git clone https://github.com/forexample/android-studio-with-hunter
> cd android-studio-with-hunter
[android-studio-with-hunter]>
```

Android Studio project configuration files reside in the `android-studio` directory but before opening it you have to create the `local.properties` file and add the `cmake.dir` entry there.

See also:

- Android Studio: Use CMake 3.7 or higher

You may want to add the paths to Android NDK/SDK as well (if `ndk.dir` and `sdk.dir` not present in `local.properties` then they will be set by Android Studio to default locations):

```
[android-studio-with-hunter]> cd android-studio
[android-studio-with-hunter/android-studio]> cat local.properties

ndk.dir=/home/your/path/to/android-sdk/ndk-bundle
sdk.dir=/home/your/path/to/android-sdk
cmake.dir=/home/your/path/to/cmake
```

Hint: Since `local.properties` contains information about a local machine you should add it to .gitignore.

Warning: Android NDK r19+ is *not supported*. You **have to** switch to a lower version explicitly, e.g. to NDK r18b.

Please check that `cmake.dir` has such value that `<cmake.dir>/bin/cmake` executable exists.

At this moment you can launch Android Studio and open your project but note that Gradle will start configuring, it will trigger CMake configuration which will trigger Hunter builds for 3 architectures:

```
[android-studio-with-hunter/android-studio]> cat app/build.gradle

android {
    ...
    defaultConfig {
        ...
        abi {
            enable true

            reset()
            include 'x86_64', 'armeabi-v7a', 'arm64-v8a'

            universalApk false
        }
    }
    ...
}
```

As an alternative, you are able to build one architecture at a time using `-Parch=`:

```
[android-studio-with-hunter/android-studio]> ./gradlew asDebug -Parch=arm64-v8a

> Task :app:externalNativeBuildDebug
Build hello-jni arm64-v8a
[1/2] Building CXX object CMakeFiles/hello-jni.dir/hello-jni.cpp.o
[2/2] Linking CXX shared library ../../../../build/intermediates/cmake/debug/obj/
↪arm64-v8a/libhello-jni.so

BUILD SUCCESSFUL in 4s
30 actionable tasks: 2 executed, 28 up-to-date
```

CMake binary directory will be set to `app/.externalNativeBuild/cmake/debug/arm64-v8a/`, you can find CMake logs there:

```
[android-studio-with-hunter/android-studio]> grep 'Hunter-ID' app/.
↪externalNativeBuild/cmake/debug/arm64-v8a/cmake_build_output.txt

[hunter] [ Hunter-ID: 4959eb9 | Toolchain-ID: 8e0b164 | Config-ID: 48b836e ]
```

Or even start CMake build without using Gradle:

```
[android-studio-with-hunter/android-studio]> touch ../CMakeLists.txt
[android-studio-with-hunter/android-studio]> cmake --build app/.externalNativeBuild/
↪cmake/debug/arm64-v8a
[1/1] Re-running CMake...
-- [hunter *** DEBUG *** 2018-07-25T19:52:14] HUNTER_ROOT set using HOME environment␣
↪variable
...
-- [hunter] [ Hunter-ID: 4959eb9 | Toolchain-ID: 8e0b164 | Config-ID: 48b836e ]
...
-- Configuring done
-- Generating done
-- Build files have been written to: /.../android-studio-with-hunter/android-studio/
↪app/.externalNativeBuild/cmake/debug/arm64-v8a
[1/1] Linking CXX shared library ../../../../build/intermediates/cmake/debug/obj/
↪arm64-v8a/libhello-jni.so
```

6.1.2 Issues

Detached CMake

If Gradle build fails the underlying CMake process will **keep running**.

```
> ./gradlew assembleDebug -Parch=armeabi-v7a
...

    * What went wrong:
    Execution failed for task ':app:generateJsonModelDebug'.
    > Format specifier '%s'
```

CMake is active:

```
> ps aux | grep cmake
```

```
... cmake -E server --experimental --debug
... cmake --build /.../__HUNTER/_Base/87420eb/2e091e5/84f821a/Build/OpenCV/Build
... cmake -E touch /.../__HUNTER/_Base/87420eb/2e091e5/84f821a/Build/OpenCV/Build/
↪OpenCV-Release-prefix/src/OpenCV-Release-stamp/OpenCV-Release-download
... cmake -P /.../__HUNTER/_Base/87420eb/2e091e5/84f821a/Build/OpenCV/Build/OpenCV-
↪Release-prefix/src/OpenCV-Release-stamp/download-OpenCV-Release.cmake
```

Internal files locked:

```
> lslocks | grep cmake.lock

cmake ... /.../__HUNTER/_Base/Download/OpenCV/4.0.0-p0/90680ea/cmake.lock
cmake ... /.../__HUNTER/_Base/87420eb/2e091e5/84f821a/cmake.lock
```

You **should not** run Gradle build again, wait for CMake job to finish or force it to stop (e.g., `kill -9`).

See issues:

- https://issuetracker.google.com/issues/123895238
- https://issuetracker.google.com/issues/75268076

No CMake files

Not all CMake files necessary for the build will be created if the initial configure step will fail. In this case, you can add `return()` command right **after the first hunter_add_package** call (this is where initialization is happening and all `*-ID` calculated) to mimic successful CMake configure step:

```
# ...
hunter_add_package(md5)
return() # Early exit
```

Run Gradle again:

```
[android-studio-with-hunter/android-studio]> ./gradlew asDebug -Parch=arm64-v8a
```

Remove `return()` from CMake code, now you will be able to run CMake:

```
[android-studio-with-hunter/android-studio]> cmake --build app/.externalNativeBuild/
↪cmake/debug/arm64-v8a
```

Example of how it can be done in a continuous integration build:

- CMakeLists.txt
- Testing script

Android NDK r19+

Android NDK r19 is not supported by built-in CMake modules (which is a requirement). The workaround is to download and use Android NDK r18 or lower:

- https://developer.android.com/ndk/downloads/older_releases.html

and add path to NDK to `local.properties`:

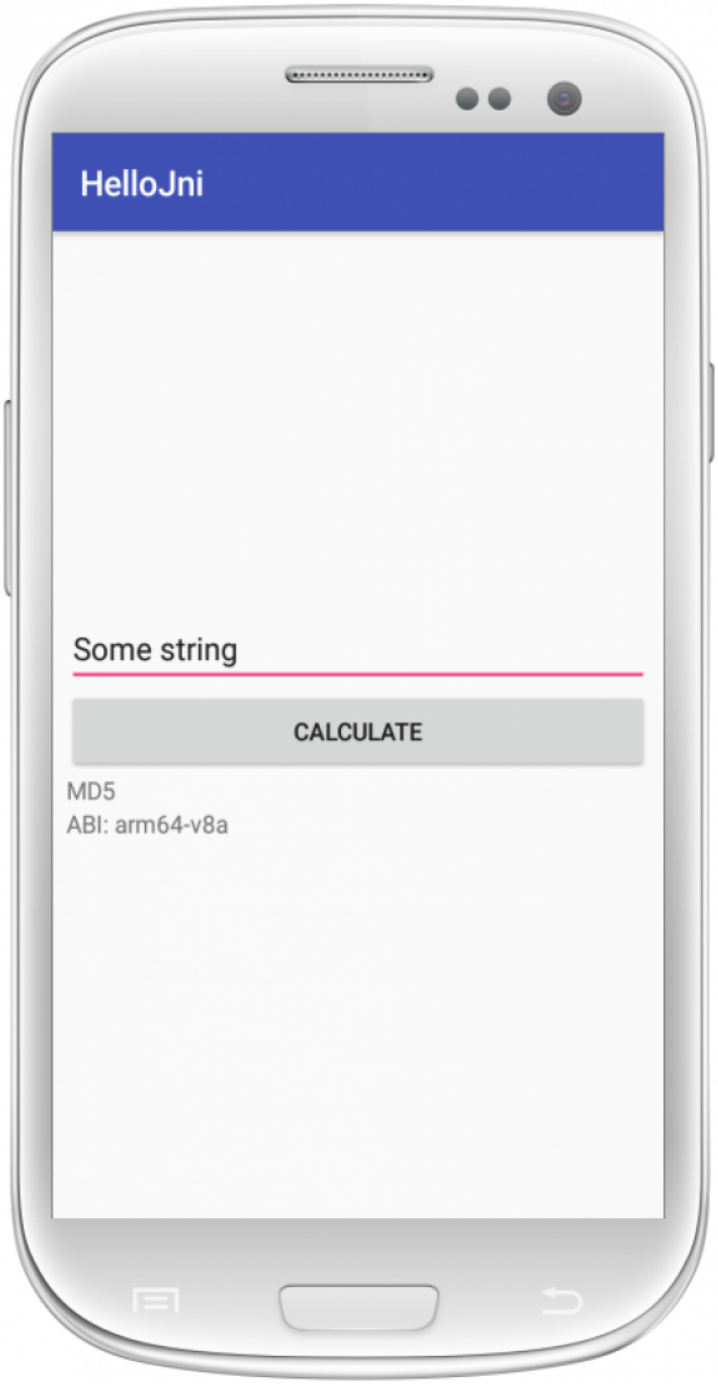
```
ndk.dir=/home/your/path/to/android-ndk-r18
sdk.dir=/home/your/path/to/android-sdk
cmake.dir=/home/your/path/to/cmake
```

See also:

- https://gitlab.kitware.com/cmake/cmake/issues/18739
- https://gitlab.kitware.com/cmake/cmake/issues/18787

6.1.3 Project

Open Android Studio project, connect your device and click `Run 'app' (Shift + F10)`. You should see `HelloJni` based application started:

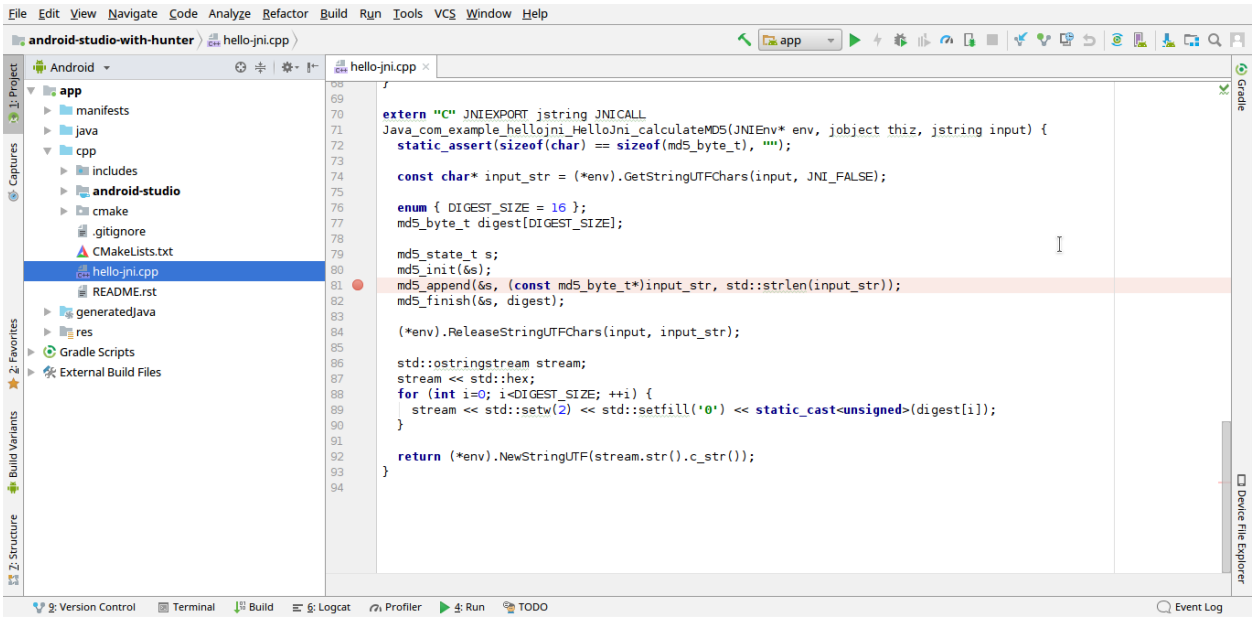


If you take a look at `CMakeLists.txt` of the project you will find the option for keeping third party sources:

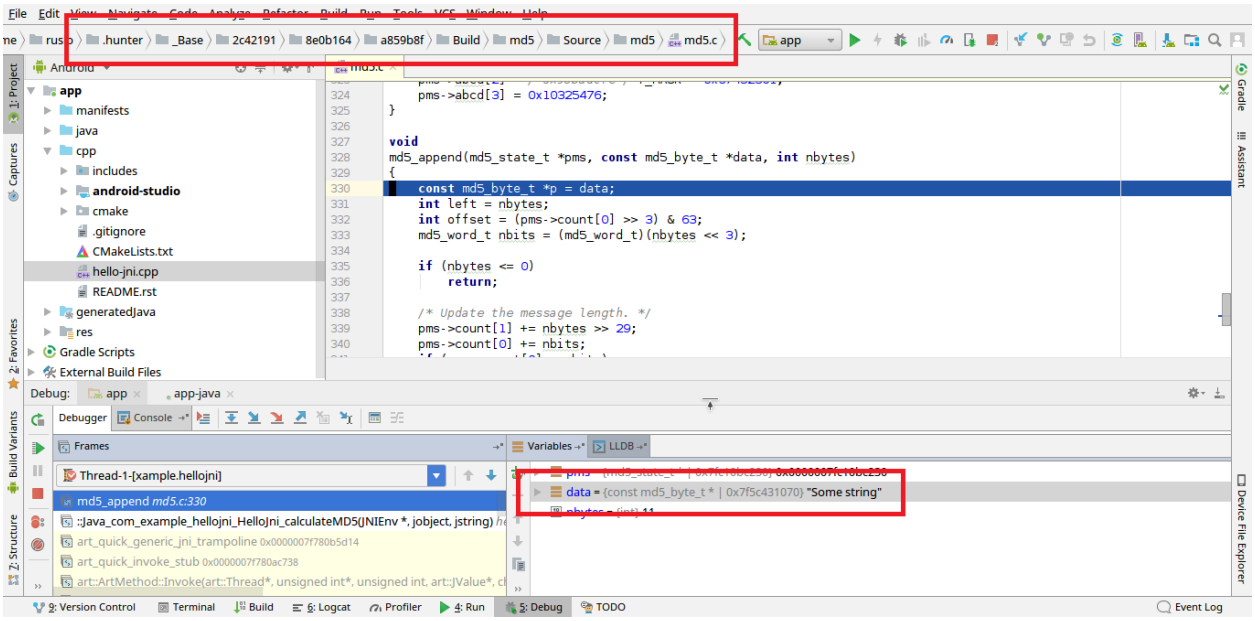
```
option(HUNTER_KEEP_PACKAGE_SOURCES "Keep third party sources" ON)
```

> **Warning:** Please make sure to read documentation about *HUNTER_KEEP_PACKAGE_SOURCES* before adding it to your project.

It means that debugger can be used to step into md5 package source code. Open `hello-jni.cpp` file and set the breakpoint to `md5_append` call:



Click `Debug 'app' (Shift + F9)` to run an application in Debug mode. After the application started click `CALCULATE` button on the device. When debugger will reach `md5_append` call click `Step Into (F7)`. As you can see debugger stepped into the `md5.c` source code of third party md5 package and "data" with value "Some string" passed to "md5_append" function:



6.1.4 Integration

Here is a description of the integration approach.

CMake toolchain file used to customize third party packages builds in Hunter. And since Android Studio provides it's own toolchain for a build such action do introduce a little quirk. Some of the variables like `ANDROID_ABI` was read from a command line and is not part of the toolchain, hence Hunter will not forward them to third parties. A user also may want to add extra settings to the toolchain. And one more problem is that variables provided by Android Studio toolchain little bit differ from ones expected by a project that relies on `CMAKE_ANDROID_*` conventions (introduced in CMake 3.7).

As a workaround for all the issues above, we can inject our own toolchain with `FORCE`.

Add extra CMake argument to `build.gradle` configuration:

```
externalNativeBuild {
    cmake {
        arguments '-DANDROID_STL=c++_static',
            // Extra custom variable to
            // trigger workaround code.
            '-DHELLOJNI_ANDROID_STUDIO=1'
    }
}
```

Note: Please name this variable next to your project to avoid clashes with other projects that can be added by `add_subdirectory`.

Use this variable for triggering CMake workaround code, note that toolchain should be set **before** first `project` command:

```
if(HELLOJNI_ANDROID_STUDIO)
  set(gen_toolchain "${CMAKE_CURRENT_BINARY_DIR}/generated/toolchain.cmake")
  configure_file(
      "${CMAKE_CURRENT_LIST_DIR}/cmake/template/toolchain.cmake.in"
      "${gen_toolchain}"
      @ONLY
  )
  set(CMAKE_TOOLCHAIN_FILE "${gen_toolchain}" CACHE PATH "" FORCE)
endif()

# ...

project(...)
```

The content of the latest `toolchain.cmake.in` template can be found here:

- https://github.com/forexample/android-studio-with-hunter/blob/master/cmake/template/toolchain.cmake.in

6.2 How to download private GitHub asset?

If you want to download private GitHub asset you have to use GitHub API. First you have to find out URL with asset id. For example get info about tag `v3.2.1` using `curl` command:

```
> curl -s -u \
    ${username}:${token} \
    https://api.github.com/repos/${orgname}/${reponame}/releases/tags/v3.2.1
```

Name, id and URL of asset:

```
> curl -s -u \
    ${username}:${token} \
    https://api.github.com/repos/${orgname}/${reponame}/releases/tags/v3.2.1 \
    | grep -A3 '"url":.*assets'

    "url": "https://api.github.com/repos/.../.../releases/assets/7654321",
    "id": 7654321,
    "name": "hello.txt",
    "label": null,
```

Use asset URL in *hunter_private_data* and add extra `Accept:application/octet-stream` header:

```
# CMakeLists.txt

hunter_private_data(
    URL "https://api.github.com/repos/${orgname}/${reponame}/releases/assets/7654321"
    SHA1 "..."
    CREDENTIALS "github"
    HTTPHEADER "Accept:application/octet-stream"
    FILE hello.txt
    LOCATION myfile
)
```

Add GitHub credentials using *hunter_private_data_password*:

```
# ~/.config/Hunter/passwords.cmake

hunter_private_data_password(
    CREDENTIALS "github"
    USERNAME "${username}"
    PASSWORD "${github_token}"
)
```

See also:

- GitHub API: Get release by tag name
- GitHub API: Get a single release asset

6.3 How to fix download error?

6.3.1 Unsupported protocol

Most sources downloaded by HTTPS protocol so CMake should be build with CURL with enabled OpenSSL. Without HTTPS support you will see this error:

```
error: downloading
'https://...' failed

  status_code: 1
```

```
  status_string: "Unsupported protocol"
  log: Protocol "https" not supported or disabled in libcurl

Closing connection -1
```

Note:

- Example of building CMake with CURL + OpenSSL

You can check that everything is fine by invoking this script:

```
# script.cmake

cmake_minimum_required(VERSION 3.2)

file(
    DOWNLOAD
    "https://github.com/cpp-pm/hunter/archive/v0.23.13.tar.gz"
    "${CMAKE_CURRENT_LIST_DIR}/hunter-archive.tar.gz"
    EXPECTED_HASH SHA1=ef7d6ac5a4ba88307b2bea3e6ed7206c69f542e8
    SHOW_PROGRESS
    TLS_VERIFY ON
)
```

```
> cmake -P script.cmake
```

6.3.2 TLS issues

TODO

Real fix instructions here

If you have any problems with TLS verification you can suppress TLS checks by setting *HUNTER_TLS_VERIFY* to `OFF`.

6.4 How to fix hash mismatch error?

6.4.1 da39a3ee5e6b4b0d3255bfef95601890afd80709

If you see error like this:

```
does not match expected value
expected: '...'
actual: 'da39a3ee5e6b4b0d3255bfef95601890afd80709'
```

It means you're experiencing some *download error*.

`da39a3ee5e6b4b0d3255bfef95601890afd80709` is a hash of an empty file:

```
> echo -n "" | openssl sha1
(stdin)= da39a3ee5e6b4b0d3255bfef95601890afd80709
```

6.4.2 Other

GitHub creates release archives on-the-fly and they are not guaranteed to be stable. This fact was discovered after few years of relying on the assumption of stability of such archives :)

In most cases the problem can be solved just by updating Hunter to latest version and using latest packages version in case you have saved non-default versions in `LOCAL`.

There will be no automatic update of hashes introduced since it affects binary cache, hence all the packages should be re-uploaded. And upload procedure is not automatic yet. Instead update will be introduced on demand.

The best solution is to find archive with old SHA1 in local *Download directory*. Then upload it as asset attached to the same release tag and add new URL to Hunter. In this case it will fix new builds and keep old cache binaries, feel free to open new issue and provide the link to old archive.

See also:

- https://github.com/ruslo/hunter/issues/1032

Note: It's not a Hunter specific issue. All tools that rely on the stability of GitHub archives was affected, as an example:

- https://github.com/Homebrew/homebrew-core/issues/18044
- https://github.com/libgit2/libgit2/issues/4343

6.5 How does Hunter interact with other package managers?

Mixing package managers in general is a bad idea (see for example Macports and Homebrew). Such approach may lead to *conflicts*, violates *automatic downloads principle* and making impossible the usage of central server with binaries because it's hard to obtain an information about dependency. Also it does break reproducibility feature of Hunter since version of system package may differs for different OS, e.g. default version of Boost on `Ubuntu 14.04` is `1.54.0` and for `Ubuntu 16.04` it's `1.58.0`. Secondly if current version of installed system package fit your needs it may not be true after you perform an upgrade. Hunter has a "static" nature: for good or for bad nothing will change until you substitute arguments of `HunterGate` module.

Note that Hunter can install packages in the same way as regular package managers do: you can set all build types to `Release`, `BUILD_SHARED_LIBS=ON` and upload binaries to public. So the question `How to make Hunter interact with system libraries?` should really be `How to improve Hunter so it behaves like system package manager?`, e.g. add new packages that are currently missing.

6.6 Why binaries from server not used?

See also:

- *Using GitHub cache server*

If settings and environment of your local project does not match environment of Travis/AppVeyor services (this is where binaries usually uploaded from) you will see `Cache miss` message and package will be build locally:

```
-- [hunter *** DEBUG *** ...] Downloading file (try #1 of 10):
-- [hunter *** DEBUG *** ...]   https://raw.githubusercontent.com/cpp-pm/hunter-cache/
↪master/aa85dd8/GTest/1.8.0-hunter-p2/93148cb/da39a3e/a49b0e5/356a192/basic-deps.DONE
-- [hunter *** DEBUG *** ...]   -> /.../_Base/Cache/meta/aa85dd8/GTest/1.8.0-hunter-
↪p2/93148cb/da39a3e/a49b0e5/356a192/basic-deps.DONE
-- [hunter *** DEBUG *** ...] File not found
-- [hunter *** DEBUG *** ...] Cache miss (no basic dependencies info found: /.../_
↪Base/Cache/meta/aa85dd8/GTest/1.8.0-hunter-p2/93148cb/da39a3e/a49b0e5/356a192/basic-
↪deps.DONE)
```

6.6.1 Reproduce environment

Next information will help you to set your environment.

- Xcode 6.1 used by default on Travis CI:
- The OS X Build Environment
- iOS 8.1 SDK is default for Xcode 6.1
- Xcode 6.1 direct download link
- Xcode 7.3.1 used for `osx_image: xcode7.3` on Travis CI:
- Xcode 7.3.1 image
- iOS 9.3 SDK is default for Xcode 7.3
- Xcode 7.3.1 direct download link
- Visual Studio versions on AppVeyor:
- https://www.appveyor.com/docs/installed-software/#visual-studio
- Docker can be used for reproducing Travis CI Linux environment:
- Install Docker on Ubuntu: https://docs.docker.com/engine/installation/linux/ubuntulinux/
- Pull image and run container by (see details):

```
> docker pull quay.io/ruslo/hunter-travis-trusty # pull/update image
> docker run -it quay.io/ruslo/hunter-travis-trusty bash
travis@...:~$ (cd polly && git pull) # fetch last changes
travis@...:~$ (cd hunter && git pull) # - // -
travis@...:~$  cd hunter && TOOLCHAIN=gcc PROJECT_DIR=examples/GTest ./jenkins.py --
↪verbose --clear-except
```

Starting GUI:

```
> xhost +
> docker run -it -e DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix quay.io/ruslo/hunter-
↪travis-trusty bash
travis@...:~$ firefox
```

6.6.2 Information from logs

When `HUNTER_STATUS_DEBUG` is `ON` you can find information about servers and cache state.

List of servers used (HUNTER_CACHE_SERVERS):

```
-- [hunter *** DEBUG *** ...] List of cache servers:
-- [hunter *** DEBUG *** ...]   * https://github.com/cpp-pm/hunter-cache
```

Meta information not found on server (cache miss):

```
-- [hunter *** DEBUG *** ...] Try to download file (try #0 of 3):
-- [hunter *** DEBUG *** ...]   https://raw.githubusercontent.com/cpp-pm/hunter-cache/
↪master/2695528/GTest/1.8.0-hunter-p2/93148cb/da39a3e/a49b0e5/356a192/basic-deps.info
-- [hunter *** DEBUG *** ...]   -> /.../_Base/Cache/meta/2695528/GTest/1.8.0-hunter-
↪p2/93148cb/da39a3e/a49b0e5/356a192/basic-deps.info
-- [hunter *** DEBUG *** ...] File not found
-- [hunter *** DEBUG *** ...] Cache miss (no basic dependencies info found: /.../_
↪Base/Cache/meta/2695528/GTest/1.8.0-hunter-p2/93148cb/da39a3e/a49b0e5/356a192/basic-
↪deps.DONE)
```

Meta information found on server (cache hit):

```
-- [hunter *** DEBUG *** ...] Try to download file (try #0 of 3):
-- [hunter *** DEBUG *** ...]   https://raw.githubusercontent.com/cpp-pm/hunter-cache/
↪master/2695528/GTest/1.8.0-hunter-p2/93148cb/da39a3e/a49b0e5/356a192/da39a3e/cache.
↪sha1
-- [hunter *** DEBUG *** ...]   -> /.../_Base/Cache/meta/2695528/GTest/1.8.0-hunter-
↪p2/93148cb/da39a3e/a49b0e5/356a192/da39a3e/cache.sha1
-- [hunter] Cache HIT: GTest
-- [hunter] Cache info: /.../_Base/Cache/meta/2695528/GTest/1.8.0-hunter-p2/93148cb/
↪da39a3e/a49b0e5/356a192/da39a3e/cache.sha1
```

Downloading archive with binaries:

```
-- [hunter *** DEBUG *** ...] Try to download file (try #0 of 3):
-- [hunter *** DEBUG *** ...]   https://github.com/cpp-pm/hunter-cache/releases/
↪download/cache/da62fc35901e07d30db7a1c19b7358855978e11f.tar.bz2
-- [hunter *** DEBUG *** ...]   -> /.../_Base/Cache/raw/
↪da62fc35901e07d30db7a1c19b7358855978e11f.tar.bz2
-- [hunter *** DEBUG *** ...] Unpacking:
-- [hunter *** DEBUG *** ...]   /.../_Base/Cache/raw/
↪da62fc35901e07d30db7a1c19b7358855978e11f.tar.bz2
-- [hunter *** DEBUG *** ...]   -> /.../_Base/3f0dbc9/6104b67/2695528/Install
```

See also:

- Example of log

6.6.3 Debugging mismatches

If environment looks the same and you're expecting everything to work fine but still see `Cache miss` message you can download meta directory and do investigate problem:

```
> git clone https://github.com/cpp-pm/hunter-cache
```

Information about missing cache entry:

```
-- [hunter *** DEBUG *** ...] Downloading file (try #1 of 10):
-- [hunter *** DEBUG *** ...]   https://raw.githubusercontent.com/cpp-pm/hunter-cache/
↪master/aa85dd8/GTest/1.8.0-hunter-p2/93148cb/da39a3e/a49b0e5/356a192/basic-deps.DONE
```

First `aa85dd8` id is about toolchain. You can find the path to toolchain info in logs:

```
-- [hunter *** DEBUG *** ...] HUNTER_TOOLCHAIN_ID_PATH: /.../_Base/86b1bc9/aa85dd8
```

```
> openssl sha1 /.../_Base/86b1bc9/aa85dd8/toolchain.info
SHA1(toolchain.info)= aa85dd86f2feefe76397d7b624ccb6c09d971fe5
```

You can see that there is no `aa85dd8` entry in cache:

```
> ls hunter-cache/aa85dd8
ls: cannot access 'hunter-cache/aa85dd8': No such file or directory
```

However in Travis build log `toolchain-id` is `8928885`:

```
> ls hunter-cache/8928885/toolchain.info
hunter-cache/8928885/toolchain.info
```

Compare both files to figure out what's wrong:

```
> diff hunter-cache/8928885/toolchain.info /.../_Base/86b1bc9/aa85dd8/toolchain.info
...
< #define __GNUC_MINOR__ 8
< #define __GNUC_PATCHLEVEL__ 1
---
> #define __GNUC_MINOR__ 4
> #define __GNUC_PATCHLEVEL__ 0
111,112c115,116
< #define __GNUC__ 4
< #define __GNUG__ 4
---
> #define __GNUC__ 5
> #define __GNUG__ 5
```

It means that local GCC version is `5.4.0` and server version is `4.8.1`.

6.7 Why do we need forks?

Forks put the burden of pushing new branches/releases from upstream, merging and resolving conflicts by maintainers and at the first view look like a bad, aggressively intrusive choice. But in practice it's the clearest, robust and universal solution for all the issues related to integration of package into Hunter.

Note: Forks are not requirement. Hunterization changes can be pushed upstream without affecting main functionality, see *compatibility* for details. And if package has no dependencies it **can be used as is** in Hunter, see *examples*.

Note: As already noted *here* all the issues that are not related to hunterization should be pushed upstream. Including most of the issues described in this section.

6.7.1 Solution for bundled sources

Take a look at this example:

- https://github.com/dmlc/rabit/tree/0759d5ed2bfa1ecfc8f00ab955d8618db474f280/include

Here package `rabit` has bundled dependencies `dmlc`. In fact `dmlc` folder is a separated project and lives here:

- https://github.com/dmlc/dmlc-core/tree/c0871823b518093a0d04d6cba0a3291bc7b31401/include

Assuming that we can't change the order of include paths (local includes should have higher priority than external because different version of same package itself can be installed in system) there is no "soft" solution here and the only way to integrate package is **to remove** `dmlc` folder from sources. In practice it means forking the project and applying "remove folder" patch. Note that it really doesn't depend on the package manager, build system or C++ compiler. All C++ compilers works similar to

```
> c++ -I/path/to/local -I/path/to/external ...
```

Meaning `#include <dmlc/timer.h>` will always fall to the choice of picking local files.

6.7.2 Set of patch files

Another way to avoid forks is to keep needed `*.patch` files in Hunter and apply them to upstream releases before running build instructions. Such approach used by Homebrew and Gentoo for example. In practice such set of patches can be quite big, e.g. 19 commits for package `OpenCV` (add `HunterGate` module, lock version in `HunterGate`, adding `hunter_add_package` calls, applying `ANDROID_*` variables introduced by new CMake version and general improvements):

- https://github.com/hunter-packages/opencv/pull/21/commits

Note that Hunter keep all available `OpenCV` versions in `cmake/projects/OpenCV/hunter.cmake` file:

- https://github.com/cpp-pm/hunter/blob/e412a3a1e9d58056efb56cb75440aead83f2e9e5/cmake/projects/OpenCV/hunter.cmake

At this moment there are 29 versions of `OpenCV` available for users, hence it will be 19 x 29 = 551 `*.patch` files to maintain. Some of them can be shared between versions, some of them can be fused with each other, etc. If such approach will be chosen we will end up with system for maintaining patches, but there is no need to reinvent the wheel, such system already exist and called `Git`. Assuming the fact that Hunter project hosted on GitHub and GitHub offer free unlimited repositories for public projects there are no real reasons to choose `*.patch` approach over forks. The use of the forks allow us to rebase, merge, cherry-pick, discuss and review the patches easily.

6.7.3 High cohesion

High cohesion means that you should keep parts of a code base that are related to each other in a single place[1]. The fact that version `v1.0` of package `Foo` works fine with Hunter archive `v0.10` is perfectly expressed by adding child commit `Add Hunter v0.10` to parent commit `Foo v1.0`. Change of dependencies from version to version is another example.

`Foo` version `v1.0`:

```
if(WIN32)
  find_package(boo CONFIG REQUIRED)
endif()

find_package(bar CONFIG REQUIRED)
```

`Foo` version `v2.0`:

[1] http://enterprisecraftsmanship.com/2015/09/02/cohesion-coupling-difference/

```
if(FOO_WITH_BAZ)
  find_package(baz CONFIG REQUIRED)
endif()

find_package(bar CONFIG REQUIRED)
```

It's hard to make a mistake in both cases:

```
--- /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/faq/
↪foo-v1.0.cmake
+++ /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/faq/
↪foo-v1.0-hunter.cmake
@@ -1,5 +1,7 @@
 if(WIN32)
+  hunter_add_package(boo)
   find_package(boo CONFIG REQUIRED)
 endif()

+hunter_add_package(bar)
 find_package(bar CONFIG REQUIRED)
```

```
--- /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/faq/
↪foo-v2.0.cmake
+++ /home/docs/checkouts/readthedocs.org/user_builds/hunter/checkouts/latest/docs/faq/
↪foo-v2.0-hunter.cmake
@@ -1,5 +1,7 @@
 if(FOO_WITH_BAZ)
+  hunter_add_package(baz)
   find_package(baz CONFIG REQUIRED)
 endif()

+hunter_add_package(bar)
 find_package(bar CONFIG REQUIRED)
```

It will be much easier to miss something while trying to support any fork-free approach:

```
if(FOO_VERSION VERSION_EQUAL 1.0 AND WIN32)
  magic_download(boo)
endif()

if(FOO_VERSION VERSION_EQUAL 2.0 AND FOO_WITH_BAZ)
  magic_download(baz)
endif()

magic_download(bar)
```

Any non-CMake custom build scheme suffers from this problem since build instructions have to know everything about all versions available, e.g. see Boost components .

6.7.4 Rejected/pending CMake patches

Having CMake build instructions in package is the easiest way to integrate package into Hunter (but not the only one) however not all developers of the upstream projects are ready to accept CMake code because it may put burden on maintaining another build system (if CMake added as extra build system), learning new build system (if you want to substitute existing system with CMake) or increase CMake minimum version to introduce new code. https://github.com/hunter-packages is a central place where CMake friendly code can leave and shared with others.

6.7.5 Removing usage of FindXXX.cmake

Overwhelming majority of projects use `FindXXX.cmake` (or even something like `find_library`) instead of recommended `XXXConfig.cmake` approach, effectively making project non-relocatable. It's not a problem for the package managers that are using single-root directory (e.g. `/usr/lib` for `apt-get` on Ubuntu and `/usr/local/lib` for `brew` on OSX) but since Hunter allow to have *multiple custom configurations* it will not work.

See also:

- *Creating new package: Install XXXConfig.cmake*

CGold

- Rejected: FindXXX.cmake

6.7.6 Lock URL/SHA1 in HunterGate

Even if all the issues will be resolved and *'hunter_add_package' will be called by hook inside 'find_package'* it's still will be convenient to save latest successful 3rd parties configuration for debugging purposes. In terms of Hunter it means attaching URL/SHA1 arguments of `HunterGate` to some commit.

6.8 Why do we need hunter_add_package?

Usually `hunter_add_package(foo)` call placed right before similar `find_package(foo ...)` call, hence it raise the question: "If most of the information is inside `find_package` already will it be possible to get rid of the `hunter_add_package`?".

TL;DR It is possible but implementation will be tricky and usefulness of such feature in practice is quite questionable.

- Not all type of packages does found by `find_package`. For example extra sources or data for testing use `*_ROOT` variables which added to the current scope by `hunter_add_package`:

```
hunter_add_package(foo_extra)
add_subdirectory(${FOO_EXTRA_ROOT})

hunter_add_package(foo_data)
add_test(NAME foo_test COMMAND foo --use-data "${FOO_DATA_ROOT}/pic.png")
```

  Meaning that `hunter_add_package` will still exist and user will have to remember that sometimes magical download hook is inside `find_package` and sometimes `hunter_add_package` have to be called explicitly.

- Mixing unrelated functionality. `hunter_add_package(foo)` will download and install `foo` package while `find_package(foo)` should only look up for files in read-only mode. When user see the code like this:

```
hunter_add_package(foo)
find_package(foo 1.22 CONFIG REQUIRED)
```

  It's clear here that version `1.22` will not be used while downloading package since it goes **after** `hunter_add_package` call and result of `hunter_add_package` call is an installed package. If package will be installed by hook in `find_package`:

```
find_package(foo 1.22 CONFIG REQUIRED)
```

User might got a feeling that version `1.22` is installed, which is not guaranteed - version of the package locked before, after first `HunterGate` call (see *Config-ID*).

- The change of `find_package` behavior will have fragile implementation. As an example: you can introduce custom macro `find_package` and call standard CMake instructions by using `_find_package`. It's undocumented CMake feature of saving previous function under underscore starting name. Overwriting standard CMake calls simply look like a hack in my opinion. I think if you have an idea that can be solved this way, then it make sense to either design and implement it in CMake itself or don't touch original code and wrap your extra functionality in separate function. As you understand Hunter pick the latter. Also this approach will not work if user will introduce his own custom `find_package` hook, or will use `include(FindXXX)/find_library`. All this are errors that should be fixed anyway but it just show an example that you will have to patch the original code effectively nullifying the benefits of this hook.
- As showed in *F.A.Q.: Why do we need forks?* the adding of `hunter_add_package` is a relatively quite small amount of code comparing to the rest of the patch needed in practice. Such optimization is only look useful at the first glance. If you try to do few practical examples you will see that it's not a big deal and not worth optimization at all, at least for now.

So the choice here is between a clean, simple and universal solution introduced by `hunter_add_package` and tricky, undocumented, surprising for user implementation that will still not cover all the scenarios.

Some important notes:

- Adding the `hunter_add_package` to your project **do not force you** to always use only Hunter for dependency management. See *Backward compatibility* for details.
- Package with CMake and without dependencies can be used **as is** in Hunter. See *examples*.
- If you want to support `DEB` packaging in your project you have to modify CMake code:
  - https://github.com/opencv/opencv/blob/4.0.1/cmake/OpenCVPackaging.cmake#L58-L116

6.9 Why Hunter is slow?

tl;dr It's not.

6.9.1 General notes

If package is already installed then overhead for each `hunter_add_package` call is a check of one `DONE` stamp file, e.g. is equal to `if(EXISTS "/path/to/DONE")`. Penalty for such action is hardly measurable.

If package is not installed then an archive with binaries can be *downloaded from server*. There will be no source download step performed for package or dependencies, and of course there will be no configure/build/install steps.

If binaries for the package are not available on server then we have to build package from sources. Archive with sources will be downloaded once, configure/build/install steps will be performed once too. Results will be *shared between several local projects*.

By default *HUNTER_STATUS_DEBUG* option is `OFF` and you may not see some Hunter activity. If you think that Hunter hangs at some point then this option is probably need to be set to `ON`.

Variable *HUNTER_CONFIGURATION_TYPES* by default set to `Release` + `Debug`, meaning that build time is longer about twice compared to a single `Release` build. Final size of the package usually will be more than twice.

6.9.2 Use latest

Prefer latest Hunter version and default package version. E.g. Windows platform is known to have performance issues while unpacking big archives. On Windows with default anti-virus turned on Boost 1.66.0 archive took more than 4 minutes to unpack, if anti-virus turned off it's about 30 (!) seconds (on Linux machine with the weaker hardware it took less than 10 seconds). Hunter by default used 1.66.0-p0 patched archive with removed examples, tests and documentation. This reduce the size from 70.7 MB to 17.7 MB, the unpack time dropped to 8 seconds. As usual downloading from cache is the best option, e.g. Boost.system Release + Debug archive has 154 KB size:

- https://github.com/cpp-pm/hunter-cache/releases/download/cache-234d975/234d9755a85b09bcd2f266d2620707ccd514020e.tar.bz2

6.9.3 *-ID calculation

Each CMake configure step will trigger calculation of `Hunter-ID`/`Config-ID`/`Toolchain-ID` triple. While `Hunter-ID` and `Config-ID` have small overhead, the calculation of `Toolchain-ID` for some generators can be noticeable. To calculate `Toolchain-ID` Hunter will create isolated project with one C++ file, and for example with Xcode generator it may take much longer then with Makefile generator:

```
> rm -rf _builds
```

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)
project(foo)
```

```
> time cmake -H. -B_builds -GXcode
-- The C compiler identification is AppleClang ...
-- The CXX compiler identification is AppleClang ...
...
-- Configuring done
-- Generating done
-- Build files have been written to: /.../_builds
cmake -H. -B_builds -GXcode ... 18.305 total
```

Same test but Makefile generator:

```
> time cmake -H. -B_builds
-- The C compiler identification is AppleClang ...
-- The CXX compiler identification is AppleClang ...
...
-- Configuring done
-- Generating done
-- Build files have been written to: /.../_builds
cmake -H. -B_builds ... 2.400 total
```

To skip `Toolchain-ID` calculation each time CMake code changed user can add *HUNTER_NO_TOOLCHAIN_ID_RECALCULATION=ON* option:

```
> rm -rf _builds
```

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)
```

```
option(
    HUNTER_NO_TOOLCHAIN_ID_RECALCULATION
    "No Toolchain-ID recalculation"
    ON
)

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)
project(foo)

hunter_add_package(md5)
```

Initial `Toolchain-ID`:

```
> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
...
-- [hunter] Calculating Toolchain-SHA1
...
```

Reuse:

```
> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
...
-- [hunter *** DEBUG *** ...] Toolchain-ID recalculation will be skipped
...
```

When this option is `ON` user is responsible for correctness of `Toolchain-ID` value on updates of compiler and compiler flags. E.g. you have to set this option to `OFF` explicitly for every local project when you do change `CXX` environment variable, upgrade compiler, update or switch Xcode version, when you change `CMAKE_TOOLCHAIN_FILE` path or content of CMake toolchain. Violation of this rule can lead to invalid unrecoverable cache state. Because of this it's highly recommended not to use this option on machine which can be used to build and upload binaries. Note that Hunter will upload all archives from `Cache` directory, not only packages build by current local project.

As an example here are actions that can lead to incorrect cache state:

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)

option(
    HUNTER_NO_TOOLCHAIN_ID_RECALCULATION
    "No Toolchain-ID recalculation"
    ON
)

set(
    CMAKE_TOOLCHAIN_FILE
    "${CMAKE_CURRENT_LIST_DIR}/toolchain.cmake"
    CACHE
    FILEPATH
    "Default toolchain"
)

include("cmake/HunterGate.cmake")
HunterGate(
```

```
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)
project(foo)

hunter_add_package(gflags)
```

```
# toolchain.cmake

set(CMAKE_CXX_STANDARD 11)
```

Run configure stage to build `gflags`:

```
> cmake -H. -B_builds -DHUNTER_STATUS_DEBUG=ON
...
-- [hunter] [ Hunter-ID: 83f7dd1 | Toolchain-ID: 385a6e9 | Config-ID: 2b427be ]
...
/usr/bin/g++-7 ... -std=gnu++11 -c /.../gflags_completions.cc
```

Toolchain with C++11 standard will have ID `385a6e9`.

Now set standard to 14:

```
# toolchain.cmake

set(CMAKE_CXX_STANDARD 14)
```

And add "GTest" to CMakeLists.txt:

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)

option(
    HUNTER_NO_TOOLCHAIN_ID_RECALCULATION
    "No Toolchain-ID recalculation"
    ON
)

set(
    CMAKE_TOOLCHAIN_FILE
    "${CMAKE_CURRENT_LIST_DIR}/toolchain.cmake"
    CACHE
    FILEPATH
    "Default toolchain"
)

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)
project(foo)

hunter_add_package(gflags)
hunter_add_package(GTest)
```

Run build:

```
> cmake --build _builds
...
-- [hunter *** DEBUG *** ...] Toolchain-ID recalculation will be skipped
...
-- [hunter] [ Hunter-ID: 83f7dd1 | Toolchain-ID: 385a6e9 | Config-ID: 2b427be ]
...
/usr/bin/g++-7 ... -std=gnu++14 -c /.../gtest-all.cc
...
```

As you can see C++14 flag used while building new package however `Toolchain-ID` remains the same, archive with invalid information saved in cache now!

The real `Toolchain-ID` for C++14 flag is `b92ba0d`:

```
> cmake -H. -B_builds -DHUNTER_NO_TOOLCHAIN_ID_RECALCULATION=OFF
...
-- [hunter] Calculating Toolchain-SHA1
...
-- [hunter] [ Hunter-ID: 83f7dd1 | Toolchain-ID: b92ba0d | Config-ID: 2b427be ]
...
```

Option can be limited only for problematic generators, e.g. apply it to Xcode generator only:

```
cmake_minimum_required(VERSION 3.2)

if(CMAKE_GENERATOR STREQUAL "Xcode")
  option(
      HUNTER_NO_TOOLCHAIN_ID_RECALCULATION
      "No Toolchain-ID recalculation"
      ON
  )
endif()

include("cmake/HunterGate.cmake")
HunterGate(
    URL "https://github.com/cpp-pm/hunter/archive/v0.23.297.tar.gz"
    SHA1 "3319fe6a3b08090df7df98dee75134d68e2ef5a3"
)
project(foo)
```

CHAPTER 7

Contributing

There are many ways to contribute to Hunter:

- Documentation
- There is a newer version of an existing package? Notify us or send a pull request with an updated version.
- Missing a package in Hunter? *Add a new package*
- Resolve Issues
  - Can you provide an answer to an open question?
  - Can you reproduce the error?
  - Is the issue still relevant? Maybe the issue can be closed.

When contributing please follow the style guides:

- Git
- CMake

Note: The minimum version of CMake for using Hunter is `3.2`. Please check that you're not using commands from newer versions (see documentation for 3.2).

Note: Before adding or updating a package in Hunter, the package is tested. Tests are done to check if the source can be downloaded, built and linked. Head over to our repository for per package CI testing contribution to see more.

Note: If you're planning to introduce nontrivial feature it's better to discuss design first, it will save a lot of time for both you and developers.

7.1 Reporting bugs

Hunter is CMake-based package manager so it's assumed that CMake is installed and working correctly. Before reporting bugs please check:

- Appropriate version of CMake is installed. See *CMake version for Hunter*.
- Verify CMake, build tools and C/C++ compilers you're planning to use. E.g. try to build simple CMake project (check this document in case you have troubles):

```
# CMakeLists.txt

cmake_minimum_required(VERSION 3.2)
project(foo)

add_executable(foo foo.cpp)
```

```
// foo.cpp

#include <iostream>

int main() {
  std::cout << "Hello world!" << std::endl;
}
```

- If you are experiencing some download error please check F.A.Q.: *How to fix download error?*

If everything seems OK:

- Run build again with *HUNTER_STATUS_DEBUG=ON*
- Make sure you're not using *HUNTER_NO_TOOLCHAIN_ID_RECALCULATION*
- Take a look at **first** error reported by Hunter. If Hunter reports chain of errors the last error you see is **not relevant**!
- Update to latest Hunter URL/SHA1 and check that issue you have hit is not already fixed/reported
- Check this document if the first error you see is `external.build.failed`:
  - https://hunter.readthedocs.io/en/latest/reference/errors/error.external.build.failed.html
- Remove irrelevant code from your example and report one problem at a time. Try to construct SSCCE. If you need more files than just `CMakeLists.txt` it's better to create separate GitHub repository for easy copying of your example. It will be nice if you can reproduce the issue with the CI system like AppVeyor/Travis.
- **Do not remove** `~/.hunter` repository to try to fix the issue! Hunter designed to be correct and reproducible, there should be no stale/rotten artifacts inside that can affect his work. If the `rm -rf ~/.hunter` step fix the issue for you it means that either you are using Hunter wrongly or there is a bug somewhere. If you want to figure out what is the origin of the problem please do keep `~/.hunter` directory.
- Open an issue and provide next info:
  - CMake version you're using `cmake --version`. CMake build from source?
  - OS (Linux, OSX, Windows)
  - Command line you're using on generate step, e.g.

    ```
    cmake -H. -B_builds "-GVisual Studio 14 2015"
    ```

  - Are you using toolchain?

- Add log until **first error** reported by Hunter

CHAPTER 8

Contacts

8.1 Public

- Feel free to open a new issue if you want to ask a question
- Public chat room on Gitter: https://gitter.im/cpp-pm/community

8.2 Private

- Private chat room on Gitter: https://gitter.im/rbsheth

8.3 Please don't

- Please use private channels only if you have sensible or private information to share, do use public channels otherwise.
- Please do not cross-post. Do not send the same messages to all channels.
- Please avoid adding messages to closed issues unless it's absolutely necessary. If the issue is closed then it means that problem was resolved for original author or author lost interest in it. Please report a new issue with your details and fresh setup instead.
- Please don't spam channels with uninformative messages such as "+1", "me too!", "any updates?!", "please please fix it!", etc. Please use GitHub reactions instead.

CHAPTER 9

Reference

9.1 Terminology

9.1.1 Hunter passwords file

Hunter passwords file is a CMake module where user can set passwords for accessing *protected sources* using *hunter_http_password* function and for specifying *uploading parameters* using *hunter_upload_password* function.

Warning: This feature is available only if version of *CMake is 3.7+*

Hunter will look for next paths (sorted by priority):

- Path specified by *HUNTER_PASSWORDS_PATH* CMake variable
- Path specified by *HUNTER_PASSWORDS_PATH* environment variable
- Path based on `HOME` environment variable: `$ENV{HOME}/.config/Hunter/passwords.cmake` (including Windows hosts)
- Path based on `USERPROFILE` environment variable: `$ENV{USERPROFILE}/.config/Hunter/passwords.cmake` (Windows only hosts)

9.1.2 Hunterize

The process of teaching CMake project to use packages from Hunter root instead of default one.

See also:

- *New package: CMake with dependencies*

9.2 Errors

9.2.1 error.abi.detection.failure

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] ABI not detected for C compiler`

Explanation

CMake failed to detect compiler ABI info:

```
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - failed
```

This indicates some compiler problems and may lead to incorrect builds, see issue #121. Also such error occurs when your compiler is forced.

What to do

- If you enabled a language for your project, like declaring a C++ project with `project(proj CXX)`, try removing the *CXX*. See issue #579.
- Else there may be other problems with your toolchain

9.2.2 error.boost.mpi.on.windows

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] MPI is required`

What to do

- Current boost.mpi version on windows tested with MS-MPI[1], please install it (`msmpisdk.msi`), check that the command `find_package(MPI REQUIRED)` successfully works then re-run build of CMake project which use hunter

9.2.3 error.boost.toolset

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] TODO: set toolset for boost`

[1] http://msdn.microsoft.com/en-us/library/bb524831%28v=vs.85%29.aspx

What to do

This is unimplemented part of boost download scheme, there are several alternatives:

- (**The best one**) Convert boost-bjam to CMake and integrate it into hunter. Here is a good start: *boost-cmake*[1]
- If you know how to build this toolset you can try to patch the `url_sha1_boost_library.cmake.in` scheme
- Otherwise at least *file a bug*[2] please

9.2.4 error.broken.package

What happens

- CMake fatal error with a message `[hunter ** FATAL ERROR **] Broken package ...`

What to do

- It seems that the package is broken for current configuration (version, toolchain, option, ...)
- Use `HUNTER_IGNORE_BROKEN_PACKAGE_ERROR` to ignore the message and try to build a package anyway
- If you know how to fix this build error, please submit a patch

9.2.5 error.build.disabled

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] Building package from source is disabled ...`

Explanation

- It means that `HUNTER_DISABLE_BUILDS` variable is set. When this variables is set expected that no builds will be triggered, i.e. all dependencies will be loaded from cache

What to do

- Package is missing in cache => set `HUNTER_DISABLE_BUILDS=NO` to build and save package to cache (note that package and all dependencies should be cacheable)

9.2.6 error.cache.file.not.found

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] Cache not found: ...`

[1] https://github.com/boost-cmake/boost-cmake
[2] https://github.com/cpp-pm/hunter/issues/new

What to do

- This is a error while caching package. Seems that one of the package dependency is not cacheable. Check that package is relocatable and can be cached and add `hunter_cacheable` to the corresponding `hunter.cmake` file

9.2.7 error.detect.hunter.root

What happens

- CMake fatal error with message:

```
[hunter ** FATAL ERROR **] Can't detect HUNTER_ROOT
```

Explanation

- Hunter need to have directory where all the packages will be installed. Next locations tried (by priority):
  - CMake variable `HUNTER_ROOT` (*high priority*; not recommended, since you need to set it for every build)
  - Environment variable `HUNTER_ROOT` (recommended, full control over directory location shared between all projects)
  - Environment variable `HOME` (*low priority*)
  - Environment variable `SYSTEMDRIVE` (Windows only)
  - Environment variable `USERPROFILE` (Windows only)

What to do

- Fix your environment. `HOME` usually defined on Unix-like platforms and `SYSTEMDRIVE` / `USERPROFILE` on Windows
- Set `HUNTER_ROOT` environment variable (restart all CMake GUI's or terminals to apply changes)

9.2.8 error.external.build.failed

What happens

- CMake fatal error with one of the messages:
  - `[hunter ** FATAL ERROR **] Configure step failed (dir: ...)`
  - `[hunter ** FATAL ERROR **] Build step failed (dir: ...)`

Explanation

- Build of some external package failed for some reason

What to do

- Find a reason of failure. Set *HUNTER_STATUS_DEBUG=ON* to see a lot of info about build
- Take a look at pkg.NAME CI testing table. If similar toolchain is excluded (or not present at all) then the problem is known, hence **there is no need to report bug if you're not planning to fix it yourself**. For example if you check the OpenSSL testing:
  - https://github.com/cpp-pm/hunter-testing/tree/pkg.openssl

  You can see that toolchain `nmake-vs-12-2013-win64` is excluded already:
  - https://github.com/cpp-pm/hunter-testing/blob/2bdb775aa312e1634c545aa772f09730f61e2e7b/appveyor.yml#L17-L20

  So there is no need to report "OpenSSL is not working with NMake" issue.
- If you want to try to fix the error and want to ask for advice, then prefer reporting it to hunterized repository (if it exist for package). For example report Boost problems to https://github.com/hunter-packages/boost . Stale bugs with label "Broken package" will be closed if there will be no activity there **even if problem may not be fixed**.

Fixable errors

Windows

- "Path too long" error with message:

```
The specified path, file name, or both are too long.  The fully qualified
file name must be less than 260 characters, and the directory name must be
less than 248 characters.
```

Can be fixed by setting *HUNTER_ROOT* environment variable to some short path, like `C:\_hunter`. Alternatively you can set *HUNTER_BINARY_DIR* environment variable. In this case all installed packages will be kept in `HUNTER_ROOT` but all builds will be triggered in `HUNTER_BINARY_DIR`. Note that if several `Hunter-ID` will be send to `HUNTER_BINARY_DIR` they will block each other and will be build sequentially (builds in `HUNTER_ROOT` lock different directories so different `Hunter-ID` instances work in parallel). Note that the problem is about native build tool (like MSBuild) and not in CMake. CMake is already using `\\?\C:\` extended-length path format (see source code).

Mac OS X

- Runtime application error:

```
Unknown class * in Interface Builder file at path *.nib
```

  Check you have next flags linked to your target:

```
target_link_libraries(... "-all_load" "-ObjC")
```

- Stackoverflow question

Reproduce and diagnose package building errors manually

> **Warning:** This may not work always since Hunter can load extra environment variables in internal scripts.

Once you enabled `HUNTER_STATUS_DEBUG`, read the error output in order to find how to build the package manually and to reproduce the error. Read the output of CMake near the error:

```
[hunter ** FATAL ERROR **] Build step failed (dir: ~/.hunter/_Base/21f5129/d74d0a3/
↪11f31d2/Build/PocoCpp
[hunter ** FATAL ERROR **] [Directory:~/.hunter/_Base/Download/Hunter/0.19.90/21f5129/
↪Unpacked/cmake/projects/PocoCpp]

------------------------------ WIKI -------------------------------
    https://github.com/ruslo/hunter/wiki/error.external.build.failed
-------------------------------------------------------------------

CMake Error at ~/.hunter/_Base/Download/Hunter/0.19.90/21f5129/Unpacked/cmake/modules/
↪hunter_wiki.cmake:12 (message):
Call Stack (most recent call first):
  ~/.hunter/_Base/Download/Hunter/0.19.90/21f5129/Unpacked/cmake/modules/hunter_fatal_
↪error.cmake:20 (hunter_wiki)
  ...
```

Carefully note the directory that is given near the message "build step failed", and build it, like shown below

```
# this is the directory given by the error message
cd ~/.hunter/_Base/21f5129/d74d0a3/11f31d2/Build/PocoCpp
cmake --build Build/
```

Then, you can diagnose more easily the cause of the error, using you standard build tools.

9.2.9 error.hunteraddpackage.after.project

What happens

- CMake fatal error with message:

```
[hunter ** FATAL ERROR **] Please set hunter_add_package *after* project command
```

Explanation

- First call to `hunter_add_package` command will trigger calculation of internal variables like `config-id` / `toolchain-id`. This must be done after toolchain is loaded (i.e. after first `project` command) because variables from toolchain like `APPLE` or `ANDROID` can be used in configuration file with `hunter_config`.

What to do

- In general sequence must looks like this (see also error.huntergate.before.project):

```
# Check CMake version before any commands
cmake_minimum_required(VERSION 3.2)

# Load HunterGate module
```

```
include("cmake/HunterGate.cmake")

# Use HunterGate module before first `project` command
HunterGate(
    URL ...
    SHA1 ...
)

# Your project
project(Foo)

# Use hunter_add_package after project command
hunter_add_package(Boo)
```

9.2.10 error.huntergate.before.project

What happens

- CMake fatal error with message:

```
[hunter ** FATAL ERROR **] Please set HunterGate *before* project command
```

Explanation

- Hunter designed to set some internal stuff like `CMAKE_CXX_COMPILER`. Such variables must be modified before `project` command to work correctly

What to do

- In general sequence must looks like this (see also error.hunteraddpackage.after.project):

```
# Check CMake version before any commands
cmake_minimum_required(VERSION 3.2)

# Load HunterGate module
include("cmake/HunterGate.cmake")

# Use HunterGate module before first `project` command
HunterGate(
    URL ...
    SHA1 ...
)

# Your project (must exist, see note below)
project(Foo)

# Use hunter_add_package after project command
hunter_add_package(Boo)
```

- Note that if there is no `project` command in `CMakeLists.txt` then CMake will set `PROJECT_NAME` to `Project` which is same side-effect as calling `project(Project)` **before** `HunterGate`. It means there **must** be at least one `project` call in `CMakeLists.txt` (which usually quite normal requirement). Related:

https://github.com/ruslo/hunter/issues/285. Quite the same will happen if `project` command is in subdirectory so next code will not work too:

```
# CMakeLists.txt
cmake_minimum_required(VERSION 3.2)
include("cmake/HunterGate.cmake")
HunterGate(URL ... SHA1 ...)
add_subdirectory(subdir1)
```

```
# subdir1/CMakeLists.txt
project(Foo)
```

Fix is to place `project` in top `CMakeLists.txt` before `add_subdirectory`:

```
# CMakeLists.txt
cmake_minimum_required(VERSION 3.2)
include("cmake/HunterGate.cmake")
HunterGate(URL ... SHA1 ...)
project(Foo) # <--------------- before add_subdirectory
add_subdirectory(subdir1)
```

9.2.11 error.incorrect.input.data

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] <error-specific-message>`

What to do

- This kind of errors signal problems with data provided by user. Please see the particular error message and fix it. Feel free to file a bug if error message is opaque or wrong.

9.2.12 error.internal

What happens

- Any CMake fatal error with message `[hunter ** INTERNAL **] ...`

What to do

Follow this guide to check for known issues and what information needed while filing bug report: - https://hunter.readthedocs.io/en/latest/contributing.html#reporting-bugs

9.2.13 error.no.toolchain.info

What happens

- CMake fatal error with message: `[hunter ** FATAL ERROR **] No toolchain info generated`

Explanation

- Hunter use try_compile to calculate toolchain-id. Internally `*.cpp` file with `#pragma message(...)` directives used to print different system macros. This error occurs when no information printed on compilation step.

What to do

- Error may occurs for old version of compilers without `#pragma message` support. E.g. minimal version of GCC is 4.4.7. In this case you need to update your compiler.
- This approach is not working with compiler used in Homebrew, reason is unclear (see this question). As a workaround you can force standard `clang++` compiler (see toolchain and CMakeLists.txt) or disable Hunter by `HUNTER_ENABLED=OFF`.
- This may happens because of wrongly created/unsupported toolchain. Please open new issue with information about toolchain you're using.

9.2.14 error.openssl.perl.not.found

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] OpenSSL: perl not found`. Perl is *mandatory*[1] for installation on Windows.

What to do

- Install perl
- Probably restart will needed

9.2.15 error.run.install

What happens

- CMake fatal error with message:

```
[hunter ** FATAL ERROR **] Hunter not found in '...'
[hunter ** FATAL ERROR **] Set HUNTER_RUN_INSTALL=ON to auto-install it from '...'
```

Explanation

- To avoid unintended buildings installation must be triggered only by user.

[1] https://github.com/openssl/openssl/blob/master/INSTALL.W32

What to do

- Check that Hunter root directory and archive version in error message is correct and set `HUNTER_RUN_INSTALL=ON`
- Also if you're sure about the project you can set default value by `option` command:

```
option(HUNTER_RUN_INSTALL "Run Hunter autoinstall" ON)
HunterGate(
    URL "..."
    SHA1 "..."
)
```

9.2.16 error.spaces.in.hunter.root

What happens

- CMake fatal error with message:

```
[hunter ** FATAL ERROR **] HUNTER_ROOT (...) contains spaces
```

Explanation

- Though CMake works fine with spaces (hence Hunter) some packages doesn't work with paths which contain space symbols. Examples:
  - OpenSSL
  - Qt (see: `The install path must not contain any spaces`)
  - Boost MINGW

What to do

- Set another `HUNTER_ROOT` location (**recommended**)
- Set `HUNTER_ALLOW_SPACES_IN_PATH=ON` CMake variable to suppress this warning (**not** recommended)

9.2.17 error.unexpected.hunter_config

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] Unexpected 'hunter_config' usage ...`

Explanation

- `hunter_config` designed to be used in a special file `config.cmake`, which is loaded and analyzed internally by Hunter. Users must not call this function explicitly in `CMakeLists.txt`. The only correct way to use this function is to create separate `config.cmake` file and load it by `HunterGate`

What to do

- Please take a look at example

Developer notes

- Error can be suppressed by `set(HUNTER_ALLOW_CONFIG_LOADING YES)` while using in unit tests

9.2.18 error.vs.devenv

What happens

- CMake fatal error:
  - *version <= v0.23.95*: with message `[hunter ** FATAL ERROR **] Incorrect CMAKE_VS_DEVENV_COMMAND ...`
  - *version >= v0.23.96*: with message `[hunter ** FATAL ERROR **] Incorrect MSVC setup ...`

Explanation

These paths can be used to determine tools which needed to build non-CMake projects (like Boost). - CMAKE_VS_DEVENV_COMMAND contains full path to `devenv.com` tool when using full versions of Visual Studio. - CMAKE_VS_MSBUILD_COMMAND contains full path to `MSBuild.exe` tool.

What to do

- For some reason the CMake version that came with Visual Studio doesn't define this variable (see issue 751). As a workaround, use standard CMake version from https://cmake.org/download/
- If you are using Microsoft Build Tools, or a similar non-full Visual Studio installation, and Hunter version <= v0.23.95 then `CMAKE_VS_DEVENV_COMMAND` may not be available. Please try updating Hunter, which will also check `CMAKE_VS_MSBUILD_COMMAND`.
- `CMAKE_VS_DEVENV_COMMAND` will be used only if there is no `VS*COMNTOOLS` environment added after Visual Studio installation (this "feature" introduced since Visual Studio 15 2017). You can add such variable manually, e.g. for Visual Studio 15 2017 set environment variable `VS150COMNTOOLS` to value `C:\...\ide\msvc\2017\Common7\Tools`

9.2.19 error.xcrun.clang

What happens

- CMake fatal error with message `[hunter ** FATAL ERROR **] 'xcrun -f clang++' failed`

What to do

- `xcrun`[1] is a part of `Xcode` environment. Probably xcode is broken or not installed at all. May be command line tools need to be installed also. After that, simply check that `xcrun -f clang++` returns the full path to clang compiler. See also `xcode-select`[2].

9.3 User variables

9.3.1 CMake

Note: All Hunter options should be set **to cache** and **before HunterGate** so user will be able to set his own values. Also if package will be used as a third party project managed by Hunter, then Hunter should be able to forward all values from by parent to child projects. So **do not** set this variables with `FORCE` or as `INTERNAL`, and don't set them as a regular variables:

```
set(HUNTER_ENABLED ON) # BAD!
```

```
set(HUNTER_STATUS_PRINT OFF CACHE BOOL "..." FORCE) # BAD!
```

```
set(HUNTER_STATUS_DEBUG ON CACHE INTERNAL "...") # BAD!
```

```
option(HUNTER_STATUS_DEBUG "Print a lot of info" ON) # Good

# Good
set(
    HUNTER_CACHE_SERVERS
    "https://github.com/elucideye/hunter-cache"
    CACHE
    STRING
    "Hunter cache servers"
)

# Good
set(HUNTER_JOBS_NUMBER 6 CACHE STRING "Hunter jobs number")

# All user options before HunterGate
HunterGate(URL "..." SHA1 "...")
```

HUNTER_ENABLED

Turn on/off Hunter package manager. If this variable is `OFF` `HunterGate` module will do nothing, `hunter_add_package` command will have no effects. You can add this variable as `option` while experimenting to keep backward compatibility with old package manager:

```
option(HUNTER_ENABLED "Enable Hunter package manager" OFF)
HunterGate(...) # ignored
if(WITH_FOO)
  hunter_add_package(Foo) # ignored
```

[1] https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/xcrun.1.html
[2] https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/xcode-select.1.html

```
  find_package(Foo) # found in standard location
endif()
```

- Default: `ON`

HUNTER_ROOT

- Path to Hunter root directory. Set this variable if you don't want to install Hunter packages to default location
- You can set *environment variable* with the same name to avoid specifying it for every project
- See HunterGate

> **Warning:** If you manually cloned Hunter repository for development purposes the values of `URL` and `SHA1` of `HunterGate` will be ignored, see notes and testing hints.

HUNTER_STATUS_PRINT

- Print current build status
- Default: `ON`

HUNTER_STATUS_DEBUG

- Print a lot of info
- Set this variable to `ON` before submitting bugs
- Default: `OFF`

HUNTER_PACKAGE_LOG_BUILD

- Apply `LOG_BUILD 1` (see ExternalProject)

HUNTER_PACKAGE_LOG_INSTALL

- Apply `LOG_INSTALL 1` (see ExternalProject)

HUNTER_CONFIGURATION_TYPES

- Build type of the 3rd party packages
- See example
- Default: `Release`, `Debug`

HUNTER_BUILD_SHARED_LIBS

- Value for BUILD_SHARED_LIBS for 3rd party packages

HUNTER_JOBS_NUMBER

- Number of parallel builds that will be used in such native tools like `make -jN` or `xcodebuild -jobs N`
- For Visual Studio >= 12 2013 flag `/maxcpucount:N` will be added to `MSBuild`
- Set variable to `0` to disable adding any flags: `HUNTER_JOBS_NUMBER=0`
- Defaults to maximum of two:
  - NUMBER_OF_LOGICAL_CORES
  - NUMBER_OF_PHYSICAL_CORES

HUNTER_RUN_INSTALL

Set this variable to `ON` to run auto-install procedure if it's disabled by *HUNTER_DISABLE_AUTOINSTALL* environment variable.

HUNTER_RUN_UPLOAD

Set this variable to `YES` to start *uploading procedure*.

- Default: `NO`

Note: Upload will start only after any real build triggered by Hunter.

HUNTER_DISABLE_BUILDS

- Set this variable to `YES` to disable building packages from sources. This may be useful in case you want to check that package can be loaded fully from local/server cache
- Default: `NO`

HUNTER_CACHE_SERVERS

- Variable contains list of servers with cache binaries
- Variable should be modified before `HunterGate` command:

```
set(
    HUNTER_CACHE_SERVERS
    "https://github.com/elucideye/hunter-cache"
    CACHE
    STRING
    "Hunter cache servers"
)
HunterGate(URL "..." SHA1 "...")
```

Using two servers:

```
set(
    HUNTER_CACHE_SERVERS
    "https://github.com/elucideye/hunter-cache;https://github.com/cpp-pm/hunter-cache"
    CACHE
```

```
    STRING
    "Hunter cache servers"
)
HunterGate(URL "..." SHA1 "...")
```

- Default: https://github.com/cpp-pm/hunter-cache

See also:

- *Why binaries from server not used?*
- *Using Nexus Repository*

HUNTER_USE_CACHE_SERVERS

- Policy to control downloading cache from server. Possible values:
- **NO** - never download cache from server, use local cache or build from sources
- **ONLY** - never build from sources, use server/local cache
- **YES** - try to download from server, build from sources if not found

	HUNTER_USE_CACHE_SERVERS		
	NO	ONLY	YES
Build from sources	yes	no	yes
Download from server	no	yes	yes

- Default is empty string. Effectively equivalent to **YES**.

HUNTER_PASSWORDS_PATH

Path to *Hunter passwords file*.

HUNTER_KEEP_PACKAGE_SOURCES

If this variable is set to `YES` then Hunter will keep package sources after finishing installation. It may be useful for navigation in code while using debug version of libraries.

This is a workaround for issue #359 and have some usage peculiarities:

- It does not work well with Hunter cache mechanism. If package binaries will be found on server, then there will be no build stage triggered, hence there will be no sources kept. Use *HUNTER_USE_CACHE_SERVERS=NO* for always building packages on local machine from sources.
- Sources will be kept inside *Hunter-ID* directory. Hence even if all the packages will be using another *Hunter-ID*, the old *Hunter-ID* directory should not be removed.
- Some packages use in-source build (non-CMake packages) and keep all build artifacts along with sources. Hunter will just keep directory and will not track what files was the original sources/what is temporary files for build. Combining with previous peculiarity it's expected that much more disk space will be used than usually.
- If package is already installed before `HUNTER_KEEP_PACKAGE_SOURCES` set to `ON` there will be no build triggered, hence there will be no sources kept. To re-trigger the build you can add some dummy parameter to `CMAKE_ARGS`, for example:

```
hunter_config(foo VERSION ${HUNTER_foo_VERSION} CMAKE_ARGS DUMMY=1)
```

See also:

- *hunter_config(... KEEP_PACKAGE_SOURCES)*

HUNTER_DOWNLOAD_SERVER

Define a list of servers to download from.

We define the following packages for the examples:

- Package 1 name: `foo`
- Package 1 SHA1: `49dee30c5fedd8613a144f9bf6551fb46bb69e92`
- Package 1 URL: `https://foo.com/downloads/foo-1.0.tar.gz`
- Package 2 name: `boo`
- Package 2 SHA1: `b1ec7331baf4c9996497851bfa2c847a73cd6085`
- Package 2 URL: `https://server-2.com/downloads/boo-3.0.tar.gz`

If `HUNTER_DOWNLOAD_SERVER` is empty nothing changes and the following URLs are used to download the sources:

- `foo`: `https://foo.com/downloads/foo-1.0.tar.gz`
- `boo`: `https://server-2.com/downloads/boo-3.0.tar.gz`

If `HUNTER_DOWNLOAD_SERVER` is a list of servers like `https://server-1.com;https://server-2.com;https://server-3.com` then the original package URL is analyzed. If the original URL matches one of the defined servers we leave it untouched and set as a server with high priority.

For package `foo` the following URLs are passed to `ExternalProject_Add` (the original URL is not used):

- `https://server-1.com/foo/1.0/SHASUM/foo-1.0.tar.gz`
- `https://server-2.com/foo/1.0/SHASUM/foo-1.0.tar.gz`
- `https://server-3.com/foo/1.0/SHASUM/foo-1.0.tar.gz`

For package `boo` the following URLs are passed to `ExternalProject_Add` (the original URL has the highest priority):

- `https://server-2.com/downloads/boo-3.0.tar.gz` (take priority, original URL used)
- `https://server-1.com/boo/3.0/SHASUM/boo-3.0.tar.gz`
- `https://server-3.com/boo/3.0/SHASUM/boo-3.0.tar.gz`

Note: Multiple URLs are supported only with CMake 3.7+. For earlier versions the first listed URL is passed to `ExternalProject_Add`.

The retry logic is implemented in the CMake function `ExternalProject_Add`.

To create new URLs the following template is used:

```
${HUNTER_DOWNLOAD_SERVER}/${PACKAGE_NAME}/${PACKAGE_VERSION}/
${ARCHIVE_ID}/${filename}
```

- The characters `!@#$%^&*?` occurring in `${filename}` are replaced with _.

- ${ARCHIVE_ID} is the first 7 characters of the package archive SHA1 sum.

Note: This is the same structure as Hunter uses for its own *Download* directory.

Note: HUNTER_DOWNLOAD_SERVER will be applied only to packages enabled with the standard VERSION variant of *hunter_config* entries, which is the case for all default Hunter package definitions. Custom package definitions introduced with a URL/SHA1 variant on *hunter_config* in a project's local configuration, such as those included through FILEPATH or LOCAL arguments to HunterGate(), will be unaffected by this variable. The git variants of *hunter_config*, namely GIT_SUBMODULE and GIT_SELF, have no transformable URL and are also unaffected by HUNTER_DOWNLOAD_SERVER.

HUNTER_TLS_VERIFY

Define if ExternalProject_Add and file(DOWNLOAD) should verify the server certificate for https:// URLs.

Default: ON

Warning: Value OFF will disable certificate verification. It means that the only protection is SHA1 hash of sources which is weak. And if you're using binary servers (it's *default*) meta cache files like cache.sha1 will not be checked at all!

HUNTER_GIT_SELF_IGNORE_UNTRACKED

Set this option to ON if you want to ignore untracked files while using *GIT_SELF feature*.

Default: OFF

HUNTER_NO_TOOLCHAIN_ID_RECALCULATION

If set to ON Hunter will skip calculation of Toolchain-ID if value is already present in CMake cache.

Default: OFF

Note: Do not use this option while making a bug report.

Warning: This option is for the **advanced** users only. Incorrect usage of this option may lead to invalid unrecoverable cache state. Please read carefully *this page* before using this option.

9.3.2 Environment

HUNTER_ROOT

- Same as CMake's *HUNTER_ROOT* variable. If both environment and CMake variables are set then CMake has a higher priority

HUNTER_BINARY_DIR

- Use external directory `HUNTER_BINARY_DIR` for building external projects. This variable can be used to fix "path too long" error on windows

HUNTER_DISABLE_AUTOINSTALL

Set this environment variable to non-empty value (e.g. `HUNTER_DISABLE_AUTOINSTALL=ON`) to disable automatic initialization of Hunter root directory by `HunterGate` module. This will give you more control in some advanced usage situations, see examples. Set *HUNTER_RUN_INSTALL=ON* CMake variable each time you want to run auto-install procedure. Note that there is no need to set any variables if Hunter root is already installed.

HUNTER_PASSWORDS_PATH

Environment variable with functionality similar to CMake variable with *the same name*.

HUNTER_GIT_EXECUTABLE

Path to Git executable

HUNTER_JOBS_NUMBER

See *HUNTER_JOBS_NUMBER* CMake variable

9.4 User modules

9.4.1 hunter_cacheable

This command will give permission to package so it can be saved in cache. Usually each root `<hunter-id>/<toolchain-id>/<config-id>` directory can be shared between unlimited number of projects but need to build from scratch every time. Binary cache allow to save builds in cache directory and share this cache between several `<hunter-id>/<toolchain-id>/<config-id>` roots. Note that all dependencies of this package and the package itself must be relocatable.

Example of `hunter.cmake` file:

```
# cmake/project/TIFF/hunter.cmake

include(hunter_add_version)
include(hunter_cacheable)
include(hunter_download)

hunter_add_version(...)

hunter_cacheable(TIFF)
hunter_download(PACKAGE_NAME TIFF)
```

Messages in logs:

- successful build of cacheable package:

```
-- [hunter] Cache saved: /.../.hunter/_Base/Cache/raw/
↪752c8b96f5613ee865c0cda5f3306d67e463a977.tar.bz2
```

- successful cache look-up (reuse/unpacked from cache):

```
-- [hunter] Cache HIT: TIFF
```

9.4.2 hunter_check_toolchain_definition

This module can help users to get access to the C++ definitions in `toolchain.info` file (which is used for `toolchain-id` calculation).

For example if you need to check that current Windows toolchain has 64-bit architecture:

```
# cmake/project/Foo/hunter.cmake

include(hunter_check_toolchain_definition)

if(WIN32)
  # Windows platform

  # Check 64-bit or 32-bit
  hunter_check_toolchain_definition(
      NAME _WIN64 DEFINED _defined
  )

  if(_defined)
    # 64-bit
    hunter_add_version(...)
  else()
    # 32-bit
    hunter_add_version(...)
  endif()
endif()
```

Value of definition can be checked too:

```
# cmake/project/Foo/hunter.cmake

include(hunter_check_toolchain_definition)

if(WIN32)
  hunter_check_toolchain_definition(
      NAME "_WIN32_WINNT"
      DEFINED _defined
      VALUE _value
  )

  # check '_defined'

  if("${_value}" STREQUAL "0x0603")
    # Windows 8.1
  endif()
endif()
```

9.4.3 hunter_config

- Usage example

This command will choose which version of package to build exactly:

```
hunter_config(
    ${package}

    # Version from "project/${package}/hunter.cmake"
    VERSION 1.2.8-hunter

    # Arguments that will be forwarded to CMake build command (optional)
    CMAKE_ARGS OPTION1=OFF OPTION2=ON
)
```

`OPTION1=OFF` and `OPTION2=ON` will be used to build your third-party package. This is similar to ExternalProject_Add command sub-option `CMAKE_ARGS`. In the case above Hunter-engine will build this package something like this:

```
> cmake -H. -B_builds -DOPTION1=OFF -DOPTION2=ON
> cmake --build _builds --target install
```

Instead of using `VERSION` you can create source archive by packing *Git submodule*:

```
hunter_config(${package} GIT_SUBMODULE "3rdparty/${package}")
```

Or packing *Current Git repository itself*:

```
hunter_config(${package} GIT_SELF)
```

Or specifying `URL`/`SHA1` of package explicitly:

```
hunter_config(${package} URL "..." SHA1 "...")
```

All variants support specifying extra:

- `VERSION` (e.g. `VERSION 4.5.6`)
- `CMAKE_ARGS` (e.g. `CMAKE_ARGS A=4 "B=5;6"`)
- `CONFIGURATION_TYPES` (e.g. `CONFIGURATION_TYPES Release MinSizeRel`)
- `KEEP_PACKAGE_SOURCES` (see *HUNTER_KEEP_PACKAGE_SOURCES*)

9.4.4 hunter_download

- Source
- Usage examples:
- PACKAGE_NAME
- PACKAGE_COMPONENT
- PACKAGE_INTERNAL_DEPS_ID

Final stage of adding package to the project. This command will read all package related variables and start the real download/build (or cache unpack) instructions. Name of package and component set by `PACKAGE_NAME` and `PACKAGE_COMPONENT`.

Option `PACKAGE_INTERNAL_DEPS_ID` is an identifier of internal files that build the package (like build scheme or additional scripts). This variable used by cache system to detect the necessity of update the binary cache of package when non of the following changed: package sources, build types, CMake arguments, toolchain-id or dependencies. The rule of thumb is to increase the value of `PACKAGE_INTERNAL_DEPS_ID` each time you're applying change of build scheme and making it public (e.g. sending pull request). This should be done for every affected package. If you want to understand the meaning of it better and why such approach used, you can read:

- Issue #74 (Binaries cache)
- Issue #194 (Correct reusing of cache directory)

Note: This variable used only by non-CMake packages since CMake-based packages build in a standard way by `url_sha1_cmake` scheme.

9.4.5 hunter_http_password

This module helps to set user-name/password for packages with *protected sources*.

If package `Foo` has protected sources and can be accessed by setting HTTP user-name to `myname` and HTTP password to `mypassword`, this can be expressed by code:

```
# ~/.config/Hunter/passwords.cmake

hunter_http_password(Foo USERNAME "myname" PASSWORD "mypassword")
```

Note that module is file with CMake code, so all regular commands available:

```
# ~/.config/Hunter/passwords.cmake

foreach(package Foo Boo Bar)
  hunter_http_password("${package}" USERNAME "myname" PASSWORD "mypassword")
endforeach()
```

```
# ~/.config/Hunter/passwords.cmake

set(user "myname")

set(pass1 "mypassword1")
set(pass2 "mypassword2")

foreach(package Foo1 Boo1 Bar1)
  hunter_http_password("${package}" USERNAME "${user}" PASSWORD "${pass1}")
endforeach()

foreach(package Foo2 Boo2 Bar2)
  hunter_http_password("${package}" USERNAME "${user}" PASSWORD "${pass2}")
endforeach()
```

9.4.6 hunter_pick_scheme

- Source
- Example

This command used to pick a build scheme for current project and called before `hunter_download` in `project/<ProjectName>/hunter.cmake` module:

```
hunter_pick_scheme(
    DEFAULT default_scheme_name # this scheme will be used by default
    IPHONEOS ios_scheme_name # this scheme will be used to build for iOS platform
    WINDOWS windows_scheme # this scheme will be used on windows
)
```

Examples:

```
# This is regular cmake project
hunter_pick_scheme(DEFAULT url_sha1_cmake)
```

```
# This is no-install (unpack only) project
hunter_pick_scheme(DEFAULT url_sha1_unpack)
```

```
# Boost bjam
hunter_pick_scheme(
    DEFAULT url_sha1_boost_library
    IPHONEOS url_sha1_boost_ios_library
)
```

```
# OpenSSL
hunter_pick_scheme(
    DEFAULT url_sha1_openssl
    IPHONEOS url_sha1_openssl_ios
    WINDOWS url_sha1_openssl_windows
)
```

9.4.7 hunter_private_data

This module helps to download user's *private data*.

Private file that is available without specifying password:

```
# CMakeLists.txt

hunter_private_data(
    URL "https://example.com/myfile.txt"
    SHA1 "abcxxxxxx123"
    FILE "myfile.txt"
    LOCATION myfile_path
)
```

Warning: Changing name specified in `FILE` or enabling/disabling `FILE` is not allowed after download done.

Variable `myfile_path` can be used now, for example in test:

```
add_test(NAME foo COMMAND foo --text-file ${myfile_path})
```

If `FILE` is not specified then archive is assumed. Hunter will unpack it and return path to unpacked directory in `LOCATION` variable:

```
# CMakeLists.txt

hunter_private_data(
    URL "https://example.com/archive.tar.gz"
    SHA1 "abcxxxxxx123"
    LOCATION mydata_dir
)

add_test(
    NAME foo
    COMMAND
    foo
    --text-file ${mydata_dir}/poem.txt
    --image-file ${mydata_dir}/cat.png
)
```

If you need to download file protected with password you have to add `CREDENTIALS`:

```
hunter_private_data(
    URL "https://example.com/archive.tar.gz"
    SHA1 "abcxxxxxx123"
    CREDENTIALS "creds"
    LOCATION mydata_dir
)
```

And add corresponding entry in *Hunter passwords file* using *hunter_private_data_password* module:

```
# ~/.config/Hunter/passwords.cmake

hunter_private_data_password(
    CREDENTIALS "creds"
    USERNAME "..."
    PASSWORD "..."
)
```

See also:

- *F.A.Q.: How to download private GitHub asset*

9.4.8 hunter_private_data_password

This module helps to set credentials for downloading *private data*.

For each *hunter_private_data* with `CREDENTIALS`:

```
# CMakeLists.txt

hunter_private_data(
    URL "..."
    SHA1 "..."
    CREDENTIALS "creds_A"
    LOCATION my_data_A
)

hunter_private_data(
    URL "..."
    SHA1 "..."
```

```
    CREDENTIALS "creds_B"
    LOCATION my_data_B
)
```

You have to define corresponding entry with `USERNAME` and `PASSWORD`:

```
# ~/.config/Hunter/passwords.cmake

hunter_private_data_password(
    CREDENTIALS "creds_A"
    USERNAME "..."
    PASSWORD "..."
)

hunter_private_data_password(
    CREDENTIALS "creds_B"
    USERNAME "..."
    PASSWORD "..."
)
```

Same `CREDENTIALS` can be used in several entries, e.g. you can download all private GitHub data using your account name and token:

```
# CMakeLists.txt

hunter_private_data(
    URL "https://api.github.com/repos/${repo}/${project}/releases/assets/${asset_id_1}
↪"
    SHA1 "${asset_id_1_sha1}"
    CREDENTIALS "github"
    HTTPHEADER "Accept:application/octet-stream"
    LOCATION asset_1
)

hunter_private_data(
    URL "https://api.github.com/repos/${repo}/${project}/releases/assets/${asset_id_2}
↪"
    SHA1 "${asset_id_2_sha1}"
    CREDENTIALS "github"
    HTTPHEADER "Accept:application/octet-stream"
    LOCATION asset_2
)
```

```
# ~/.config/Hunter/passwords.cmake

hunter_private_data_password(
    CREDENTIALS "github"
    USERNAME "${username}"
    PASSWORD "${username_github_token}"
)
```

See also:

- *F.A.Q.: How to download private GitHub asset*

9.4.9 hunter_protected_sources

Warning: This feature implemented only for build schemes:

- `url_sha1_cmake`
- `url_sha1_unpack`

If package has sources protected by password you should use `hunter_protected_sources` to mark it so:

```
# cmake/projects/Foo/hunter.cmake

include(hunter_protected_sources)

# ...

hunter_protected_sources(Foo)
hunter_download(PACKAGE_NAME Foo)
```

Note:

- The word "sources" is important here since binaries from cache can be shared or private. E.g. if you upload the binaries produced by **private package** to the **public binaries server** then users can download **binaries** without specifying any credentials.

See also:

- *Protected sources*

9.4.10 hunter_source_subdir

- Source
- Example

For projects where the `CMakeLists.txt` is not in the root of the project tree this command can be used to specify the sub-folder the CMake project file is in.

The value is used to set the ExternalProject_Add command sub-option `SOURCE_SUBDIR`.

It is meant to be used in the project definition at `project/<ProjectName>/hunter.cmake`:

```
hunter_source_subdir(
    ${package}
    # SOURCE_SUBDIR will be forwarded to ExternalProject_Add command
    SOURCE_SUBDIR "cpp"
)
```

9.4.11 hunter_upload_password

This module helps to set uploading parameters for *binary cache server*.

If you want to use GitHub repository `https://github.com/forexample/hunter-cache` as a cache server and do uploads using bot `cpp-pm-bot` this can be expressed by code:

```
# ~/.config/Hunter/passwords.cmake

hunter_upload_password(
    REPO_OWNER "forexample"
    REPO "hunter-cache"
    USERNAME "cpp-pm-bot"
    PASSWORD "very-secured-github-token-here"
)
```

Note that module is file with CMake code, so all regular commands available. E.g. you can read password from environment variable:

```
# ~/.config/Hunter/passwords.cmake

hunter_upload_password(
    REPO_OWNER "forexample"
    REPO "hunter-cache"
    USERNAME "cpp-pm-bot"
    PASSWORD "$ENV{GITHUB_USER_PASSWORD}"
)
```

9.5 Internal variables

9.5.1 HUNTER_PACKAGE_SCHEME_<TYPE>

Type of the currently used scheme. Only one type should be set to `1`, other types should have empty values. Next table describe the difference between them:

	name	<pkg>_ROOT	EP?[1]	cache[2]
HUNTER_PACKAGE_SCHEME_DOWNLOAD	url_sha1_download	source directory	no	yes
HUNTER_PACKAGE_SCHEME_UNPACK	url_sha1_unpack	source directory	yes	yes
HUNTER_PACKAGE_SCHEME_UNPACK_INSTALL	url_sha1_unpack_install or url_sha1_unpack_bin_install	install directory	yes	yes
HUNTER_PACKAGE_SCHEME_INSTALL	*other*	install directory	yes	no

9.6 Internal modules

9.6.1 hunter_fatal_error

Wrapper for the `message(FATAL_ERROR ...)` command. Message marked as one from Hunter by adding prefix `[hunter ** FATAL ERROR **]`. Additionally current module directory printed. This command expects `ERROR_PAGE` argument which will link to the page with detailed description of the problem. For example:

[1] Does scheme use ExternalProject_Add? (information used while doing look up for stamps)
[2] Is package cacheable by default? Yes - always cacheable, No - depends on the package (see `hunter_cacheable`)

```
hunter_fatal_error(
    "Please set hunter_add_package *after* project command"
    ERROR_PAGE "error.hunteraddpackage.after.project"
)
```

will convert

- `error.hunteraddpackage.after.project`

to

- https://hunter.readthedocs.io/en/latest/reference/errors/error.hunteraddpackage.after.project.html

9.6.2 hunter_internal_error

Wrapper for the `message(FATAL_ERROR ...)`. Some internal unrecoverable error.

9.7 Layouts

9.7.1 Sources

This is a detailed sources layout:

```
.
├── docs/                              # readthedocs.io RST documentation
├── examples/<name>                    # examples for testing
├── gate/                              # HunterGate module for testing
├── maintenance/                       # scripts for generating files for Hunter
├── scripts/                           # global scripts
├── tests/<name>                       # unit-testing
└── cmake/
    ├── Hunter                         # master file
    ├── templates/                     # global CMake templates
    ├── schemes/                       # global schemes
    ├── configs/
    │   └── default.cmake              # default config
    ├── find/                          # Hunterized CMake Find*.cmake modules
    ├── modules/                       # CMake modules
    └── projects/
        └── <name>/
            ├── hunter.cmake
            ├── scripts/               # package specific scripts
            ├── templates/             # package specific CMake templates
            ├── schemes/               # package specific schemes
            │   ├── default.cmake.in
            │   ├── iphoneos.cmake.in
            │   └── windows.cmake.in
            ├── ep-stages/
            │   ├── configure.cmake.in
            │   ├── build.cmake.in
            │   └── install.cmake.in
            └── <component>
                └──  hunter.cmake
```

Note:

- all top directories except `cmake` and `scripts` can be removed from final release since none of them used by user (TODO: move `scripts` to `cmake/scripts`)
- `maintenance` holds scripts for generating files that will be saved in git. Generation done by developers and never runs by Hunter so it can be any kind of scripts, e.g. Python. This directory can be removed from release too
- for name of package specific `schemes` see hunter_pick_scheme

9.7.2 Deployed

Common

There is a common pattern for creating shareable directories (directories that can be created by several CMake instances running simultaneously):

```
<...-ID>/
├── cmake.lock
├── SHA1
└── DONE
```

- `cmake.lock` synchronization file, see file(LOCK ...) command
- `SHA1` file with SHA1 value, first 7 digits of SHA1 is `<...-ID>`
- `DONE` stamp that shows that directory created

For example we have file `toolchain.info` and we want to save it in `${HUNTER_ROOT}` directory in non-conflicting fashion. We can do:

- Create `toolchain.info` locally (out of `${HUNTER_ROOT}` directory, e.g. somewhere in `${CMAKE_BINARY_DIR}`)
- Calculate SHA1 of `toolchain.info`
- Calculate Toolchain-ID by getting first 7 digits of SHA1
- Check `<Toolchain-ID>/DONE` file exists
- If file exists check that `<Toolchain-ID>/SHA1` do matches our SHA1. Assumed that probability of collision of Toolchain-ID is very low, in case collision happen we should extend short Toolchain-ID to 8 digits
- If `<Toolchain-ID>/DONE` not exists then lock `<Toolchain-ID>/cmake.lock`
- Save `toolchain.info` to `<Toolchain-ID>/toolchain.info`
- Save SHA1 to `<Toolchain-ID>/SHA1`
- Save empty stamp file `<Toolchain-ID>/DONE`
- Unlock `<Toolchain-ID>/cmake.lock`

Base

Layout of deployed files (layout of `${HUNTER_ROOT}` directory). Layout starts with the root directory `_Base`. The purpose of this directory is to allow Hunter to be deployed inside *sources*, i.e. when repository is cloned for development.

```
_Base/
├── <Hunter-ID>/
│   ├── cmake.lock
│   ├── SHA1                                # SHA1 of Hunter archive
│   ├── DONE
│   └── <Toolchain-ID>/
│       ├── cmake.lock
│       ├── SHA1                            # SHA1 of 'toolchain.info'
│       ├── DONE
│       ├── toolchain.info
│       └── <Config-ID>/
│           ├── cmake.lock
│           ├── SHA1                        # SHA1 of 'config.cmake'
│           ├── DONE
│           ├── config.cmake                # CMake file with unified hunter_final_
↪config commands
│           ├── cache.cmake
│           ├── Install/                    # Main directory with installed packages␣
↪(global)
│           │   ├── include/
│           │   ├── lib/
│           │   └── ...
│           └── Build/                      # Directory for building packages
│               └── <Package>/
│                   ├── DONE                # Stamp - package already built and␣
↪installed
│                   ├── CMakeLists.txt      # File with ExternalProject_Add command␣
↪configured from build scheme
│                   ├── toolchain.cmake
│                   ├── args.cmake
│                   ├── cache.sha1
│                   ├── types.info
│                   ├── internal_deps.id
│                   ├── Source/             # Unpacked sources of package
│                   ├── Build/              # Build directory of package
│                   ├── Install/            # Directory for local installation (for␣
↪cacheable packages)
│                   ├── Dependencies/       # Directory for saving information about␣
↪depedencies
│                   │   └── <Package>/
│                   │       ├── __dep       # Empty stamp file, means this package␣
↪depends on <Package>
│                   │       └── <Component>/
│                   │           └── __dep
│                   └── __<Component>/      # Directory for building component, same␣
↪layout
│                       ├── DONE
│                       ├── CMakeLists.txt
│                       ├── toolchain.cmake
│                       ├── args.cmake
│                       ├── cache.sha1
│                       ├── types.info
│                       ├── internal_deps.id
│                       ├── Source/
│                       ├── Build/
│                       ├── Install/
│                       └── Dependencies/
├── Download/                               # see below
```

```
├── Cellar/                                  # see below
└── Cache/                                   # see below
```

Download

Directory for storing archives with sources. Sources will be unpacked to `<Hunter-ID>/<Toolchain-ID>/<Config-ID>/Build/<Package>/Source` directory. One exception is archives with Hunter itself since we have no information about `<Toolchain-ID>/<Config-ID>` part (we have to calculate them using Hunter code).

```
Download/
├── <Package>/
│   └── <version>/
│       └── <Archive-ID>/
│           ├── cmake.lock
│           ├── SHA1                     # SHA1 of <Package>.tar.bz2
│           ├── DONE
│           └── <Package>.tar.bz2        # archive with sources
└── Hunter/
    └── <version>/
        └── <Hunter-ID>/                 # created by HunterGate module
            ├── cmake.lock
            ├── SHA1                     # SHA1 of Hunter archive
            ├── DONE
            ├── CMakeLists.txt
            ├── <Package>.tar.bz2        # Hunter archive
            ├── Build/
            └── Unpacked/                # Unpacked Hunter archive (HUNTER_SELF)
```

Cache

Cache directory can be used by several Hunter-ID directories and consists of `raw` directory with `*.tar.bz2` files (packed installed binaries) and `meta` directory with information about such binaries (SHA1 of sources, arguments, dependencies, etc.).

```
Cache/
├─ raw/
│  └─ <cache>.tar.bz2
└─ meta/
   └─ <Toolchain-ID>/
      ├─ cmake.lock
      ├─ SHA1                                # SHA1 of toolchain.info
      ├─ DONE
      ├─ toolchain.info                      # see above
      └─ <Package>/
         └─ <__Component>/                   # (optional, if any)
            └─ <version>/
               └─ <Archive-ID>/
                  ├─ cmake.lock
                  ├─ SHA1                    # SHA1 of archive with sources
                  ├─ DONE
                  └─ <Args-ID>/
                     ├─ cmake.lock
                     ├─ SHA1                 # SHA1 of args.cmake
                     ├─ DONE
                     ├─ args.cmake           # arguments used to build this package
```

```
                        └─ <Types-ID>/
                           ├─ cmake.lock
                           ├─ SHA1                  # SHA1 of types.info
                           ├─ DONE
                           ├─ types.info            # build types (Release, Debug)
                           └─ <Internal-Deps-ID>/
                              ├─ cmake.lock
                              ├─ SHA1               # SHA1 of internal_deps.id
                              ├─ DONE
                              ├─ internal_deps.id   # PACKAGE_INTERNAL_DEPS_ID (empty for␣
↪CMake-based packages)
                              ├─ basic-deps.info    # list of explicit dependencies of␣
↪package
                              ├─ basic-deps.DONE    # stamp: basic-deps.info created
                              └─ <Deps-ID>/
                                 ├─ cmake.lock
                                 ├─ SHA1            # SHA1 of deps.info
                                 ├─ DONE
                                 ├─ cache.sha1      # file with SHA1, this SHA1 means␣
↪that binary can be
                                 │                  # unpacked from '${HUNTER_ROOT}/_Base/
↪Cache/raw/<SHA1>.tar.bz2'
                                 ├─ deps.info       # list of all dependencies and␣
↪corresponding SHA1 of cache archive
                                 ├─ CACHE.DONE      # stamp: deps.info and cache.sha1␣
↪created and ready to be used
                                 └─ from.server     # info downloaded from server, no␣
↪need to upload this entry
```

Cellar

Cellar directory consists of unpacked raw cache archives and source archives of `url_sha1_unpack` packages:

```
Cellar/
└─ <sha1>/                                  # SHA1 of unpacked archive
   └─ <id>/                                 # first 7 digits of SHA1
      ├─ cmake.lock
      ├─ SHA1
      ├─ DONE
      ├─ unpack.DONE                        # stamp: unpack operation finished
      ├─ directories.list                   # list of unpacked directories
      ├─ files.list                         # list of unpacked files
      ├─ link-all.sh                        # link script
      ├─ licenses/
      └─ raw/                               # directory with unpacked files
```

PrivateData

Directory with *downloaded private data*.

If `FILE` specified (download only):

```
PrivateData/
└─ <sha1>/                 # SHA1 of downloaded file
   └─ <id>/                 # first 7 digits of SHA1
```

```
      ├─ cmake.lock
      ├─ SHA1
      ├─ DONE
      ├─ unpack.DONE        # stamp: download operation finished
      ├─ param.file         # value specified in `FILE`
      └─ raw/<filename>     # downloaded file
```

If `FILE` not specified (download archive and unpack):

```
PrivateData/
└─ <sha1>/                 # SHA1 of downloaded file
   └─ <id>/                # first 7 digits of SHA1
      ├─ cmake.lock
      ├─ SHA1
      ├─ DONE
      ├─ unpack.DONE        # stamp: download and unpack operation finished
      ├─ archive.file       # downloaded archive
      └─ raw/*              # unpacked directory
```

9.8 Release notes

9.8.1 v0.X.Y

v0.21.X

- New *HUNTER_GIT_EXECUTABLE* environment variable
- *Private data download*
- *Download private GitHub asset*
- Requirements for uploading to GitHub updated, check *documentation* for details
- *Uploading to Artifactory server*

New packages and updates

`git diff v0.20.0..v0.21.0 -- cmake/configs/default.cmake`:

```
diff --git a/cmake/configs/default.cmake b/cmake/configs/default.cmake
index 42093d2..cff73a3 100644
--- a/cmake/configs/default.cmake
+++ b/cmake/configs/default.cmake
@@ -27,52 +27,63 @@ include(hunter_user_error)
 hunter_config(ARM_NEON_2_x86_SSE VERSION 1.0.0-p0)
 hunter_config(AllTheFlopsThreads VERSION 0.1-p0)
 hunter_config(Android-Apk VERSION 1.1.14)
-hunter_config(Android-Build-Tools VERSION 22.0.1)
-hunter_config(Android-Google-Repository VERSION 47)
+hunter_config(Android-Build-Tools VERSION 27.0.3)
+hunter_config(Android-Google-Repository VERSION 58)
 hunter_config(Android-Modules VERSION 1.0.0)
-hunter_config(Android-SDK VERSION 0.0.5)
-hunter_config(Android-SDK-Platform-tools VERSION r25.0.5)
```

```
+hunter_config(Android-SDK VERSION 0.0.6)
+hunter_config(Android-SDK-Platform-tools VERSION r27.0.1)
 hunter_config(Android-SDK-Tools VERSION 25.2.5)
 hunter_config(Android-Support-Repository VERSION 47)
 hunter_config(AngelScript VERSION 2.30-p0)
 hunter_config(ArrayFire VERSION 3.3.1-p0)
-hunter_config(Assimp VERSION 3.2-p1)
+hunter_config(Assimp VERSION 3.2-p2)
 hunter_config(Async++ VERSION 0.0.3-hunter)
 hunter_config(Avahi VERSION 0.6.31)
 hunter_config(Beast VERSION 1.0.0-b84-hunter-0)
 hunter_config(BZip2 VERSION 1.0.6-p3)
-if(MINGW)
-  # FIXME: https://ci.appveyor.com/project/ingenue/hunter/build/1.0.2229
+
+if(MSVC)
+  # https://github.com/boostorg/build/issues/299
+  hunter_config(Boost VERSION 1.66.0-p0)
+elseif(MINGW)
+  # https://github.com/boostorg/build/issues/301
   hunter_config(Boost VERSION 1.64.0)
 else()
-  hunter_config(Boost VERSION 1.66.0)
+  hunter_config(Boost VERSION 1.67.0-p1)
 endif()
+
 hunter_config(BoostCompute VERSION 0.5-p0)
 hunter_config(BoostProcess VERSION 0.5)
 hunter_config(BoringSSL VERSION 1.0.0)
 hunter_config(Box2D VERSION 2.3.1-p0)
 hunter_config(CapnProto VERSION 0.6.1)
+hunter_config(catkin VERSION 0.7.11-p1)
+hunter_config(cctz VERSION 2.2.0)
 hunter_config(CLAPACK VERSION 3.2.1)
-hunter_config(CURL VERSION 7.49.1-DEV-v9)
+hunter_config(CURL VERSION 7.59.0-p1)
 hunter_config(Clang VERSION 4.0.1-p0)
 hunter_config(ClangToolsExtra VERSION 4.0.1) # Clang
 hunter_config(Comet VERSION 4.0.2)
 hunter_config(cpr VERSION 1.3.0)
 hunter_config(CppNetlib VERSION 0.10.1-hunter-3)
-hunter_config(CppNetlibUri VERSION 1.0.4-hunter)
+hunter_config(CppNetlibUri VERSION 1.0.5-hunter)
 hunter_config(crc32c VERSION 1.0.5)
 hunter_config(CsvParserCPlusPlus VERSION 1.0.1)
 hunter_config(Eigen VERSION 3.3.4-p1)
 hunter_config(state_machine VERSION 1.1)
 hunter_config(enet VERSION 1.3.13-p1)
+hunter_config(ethash VERSION 0.1.0)
 hunter_config(Expat VERSION 2.1.1)
 if(MSVC)
   hunter_config(getopt VERSION 1.0.0-p0)
 endif()
 hunter_config(GPUImage VERSION 0.1.6-p9)
 hunter_config(GSL VERSION 2.1.0-p2)
+hunter_config(ICU VERSION 55.1-p3)
 hunter_config(Igloo VERSION 1.1.1-hunter)
-hunter_config(intsizeof VERSION 2.0.1)
```

```
+hunter_config(intsizeof VERSION 2.0.2)
+hunter_config(jansson VERSION 2.11.0)
+hunter_config(jasper VERSION 2.0.14-p2)
 hunter_config(Jpeg VERSION 9b-p3)
 hunter_config(JsonSpirit VERSION 0.0.4-hunter)
 if(MSVC_VERSION LESS 1600)
@@ -84,12 +95,14 @@ endif()
 hunter_config(LAPACK VERSION 3.7.1)
 hunter_config(LLVM VERSION 4.0.1-p0) # Clang
 hunter_config(LLVMCompilerRT VERSION 4.0.1-patched) # Clang
-hunter_config(Leathers VERSION 0.1.6)
+hunter_config(Leathers VERSION 0.1.8)
 hunter_config(Leptonica VERSION 1.74.2-p4)
 hunter_config(LibCDS VERSION 2.3.1)
 hunter_config(Libcxx VERSION 3.6.2) # Clang
 hunter_config(Libcxxabi VERSION 3.6.2) # Clang
-hunter_config(Libevent VERSION 2.1.8-p3)
+hunter_config(Libevent VERSION 2.1.8-p4)
+hunter_config(lcms VERSION 2.9-p0)
+hunter_config(libevhtp VERSION 1.2.16-p4)
 hunter_config(libffi VERSION 3.2.1)
 hunter_config(librtmp VERSION 2.4.0-p0)
 hunter_config(Libssh2 VERSION 1.7.0)
@@ -100,26 +113,29 @@ hunter_config(OpenAL VERSION 1.18.2)
 hunter_config(OpenBLAS VERSION 0.2.20-p0)
 hunter_config(OpenCL VERSION 2.1-p3)
 hunter_config(OpenCL-cpp VERSION 2.0.10-p0)
-hunter_config(OpenCV VERSION 3.4.0-p0)
-hunter_config(OpenCV-Extra VERSION 3.4.0)
+hunter_config(OpenCV VERSION 3.4.1-p1)
+hunter_config(OpenCV-Extra VERSION 3.4.1)
 hunter_config(OpenNMTTokenizer VERSION 0.2.0-p1)
-hunter_config(OpenSSL VERSION 1.1.0g)
-hunter_config(PNG VERSION 1.6.26-p1)
+hunter_config(OpenSSL VERSION 1.1.0h)
+hunter_config(PNG VERSION 1.6.26-p3)
 hunter_config(PocoCpp VERSION 1.7.9-p1)
 hunter_config(PostgreSQL VERSION 10.0.0)
-hunter_config(Protobuf VERSION 3.3.0)
+hunter_config(PROJ4 VERSION 5.0.0)
+hunter_config(Protobuf VERSION 3.5.2-p0)

 string(COMPARE EQUAL "${CMAKE_SYSTEM_NAME}" "Linux" _is_linux)
 if(_is_linux OR MINGW)
   # qt-qml example is broken on Linux
   # qt-core example is broken on MinGW
   hunter_config(Qt VERSION 5.5.1-cvpixelbuffer-2-p9)
-else()
+elseif(IOS OR ANDROID)
   hunter_config(Qt VERSION 5.9.1-p0)
+else()
+  hunter_config(Qt VERSION 5.10.1)
 endif()

 hunter_config(QtAndroidCMake VERSION 1.0.9)
-hunter_config(QtCMakeExtra VERSION 1.0.28)
+hunter_config(QtCMakeExtra VERSION 1.0.30)
 hunter_config(QtQmlManager VERSION 1.0.0)
```

```diff
 hunter_config(RapidJSON VERSION 1.1.0)
 hunter_config(RapidXML VERSION 1.13)
@@ -131,15 +147,17 @@ hunter_config(SDL_ttf VERSION 2.0.14-p0)
 hunter_config(sds VERSION 2.0.0)
 hunter_config(sqlite3 VERSION 3.21.0-p2)
 hunter_config(Sober VERSION 0.1.3)
+hunter_config(sources_for_android_sdk_packer VERSION 1.0.0)
 hunter_config(stdext-path VERSION 0.0.1-p0)
 hunter_config(stormlib VERSION 9.21-p1)
 hunter_config(sugar VERSION 1.3.0)
 hunter_config(SuiteSparse VERSION 4.5.1-p1)
-hunter_config(TIFF VERSION 4.0.2-p3)
+hunter_config(TCLAP VERSION 1.2.2-p1)
+hunter_config(TIFF VERSION 4.0.2-p5)
 hunter_config(toluapp VERSION 1.0.93-p1)
 hunter_config(tomcrypt VERSION 1.17-p3)
 hunter_config(tommath VERSION 1.0-p2)
-hunter_config(Urho3D VERSION 1.7-p13)
+hunter_config(Urho3D VERSION 1.7-p15)
 hunter_config(WTL VERSION 9.1.5321)
 hunter_config(WDC VERSION 1.1.1)
 hunter_config(Washer VERSION 0.1.2)
@@ -148,13 +166,26 @@ hunter_config(ZLIB VERSION 1.2.8-p3)
 hunter_config(ZMQPP VERSION 4.1.2)
 hunter_config(ZeroMQ VERSION 4.2.3-p1)
 hunter_config(caffe VERSION rc3-p2)
-hunter_config(acf VERSION 0.0.2)
-hunter_config(Catch VERSION 2.0.1)
+hunter_config(acf VERSION 0.1.3)
+hunter_config(Catch VERSION 2.2.1)
 hunter_config(aes VERSION 0.0.1-p1)
-hunter_config(aglet VERSION 1.2.0)
+hunter_config(aglet VERSION 1.2.2)
+hunter_config(android_arm64_v8a_system_image_packer VERSION 1.0.0)
+hunter_config(android_arm_eabi_v7a_system_image_packer VERSION 1.0)
+hunter_config(android_build_tools_packer VERSION 1.0.0)
+hunter_config(android_google_apis_intel_x86_atom_system_image_packer VERSION 1.0.0)
+hunter_config(android_google_apis_packer VERSION 1.0.0)
+hunter_config(android_google_repository_packer VERSION 1.0.0)
+hunter_config(android_intel_x86_atom_system_image_packer VERSION 1.0.0)
+hunter_config(android_mips_system_image_packer VERSION 1.0.0)
+hunter_config(android_sdk_packer VERSION 1.0.0)
+hunter_config(android_sdk_platform_packer VERSION 1.0.0)
+hunter_config(android_sdk_platform_tools_packer VERSION 1.0.0)
+hunter_config(android_sdk_tools_packer VERSION 1.0.2)
+hunter_config(android_support_repository_packer VERSION 1.0.0)
 hunter_config(autobahn-cpp VERSION 0.2.0)
 hunter_config(autoutils VERSION 0.3.0)
-hunter_config(benchmark VERSION 1.3.0)
+hunter_config(benchmark VERSION 1.4.0)
 hunter_config(bison VERSION 3.0.4-p0)
 hunter_config(boost-pba VERSION 1.0.0-p0)
 hunter_config(bullet VERSION 2.87-p0)
@@ -176,9 +207,9 @@ hunter_config(cxxopts VERSION 1.0.0-p0)
 hunter_config(czmq VERSION 4.0.2-p1)
 hunter_config(damageproto VERSION 1.2.1)
 hunter_config(dbus VERSION 1.10.0-hunter-4)
-hunter_config(debug_assert VERSION 1.3)
```

```
+hunter_config(debug_assert VERSION 1.3.2)
 hunter_config(dest VERSION 0.8.0-p4)
-hunter_config(dlib VERSION 19.8-p1)
+hunter_config(dlib VERSION 19.10-p2)
 hunter_config(dmlc-core VERSION 0.0.0-p1)
 hunter_config(doctest VERSION 1.2.0)
 hunter_config(double-conversion VERSION 3.0.0)
@@ -188,18 +219,23 @@ hunter_config(drishti VERSION 0.8.9)
 hunter_config(drishti_assets VERSION 1.8)
 hunter_config(drishti_faces VERSION 1.2)
 hunter_config(drm VERSION 2.4.67)
+hunter_config(duktape VERSION 1.5.2-p0)
+hunter_config(dynalo VERSION 1.0.3)
 hunter_config(eigen3-nnls VERSION 1.0.1)
 hunter_config(eos VERSION 0.12.1)
+hunter_config(EnumGroup VERSION 0.0.1)
 hunter_config(FakeIt VERSION 2.0.3)
 hunter_config(FunctionalPlus VERSION 0.2-p0)
-hunter_config(fft2d VERSION 1.0.0-p0)
 hunter_config(farmhash VERSION 1.1)
+hunter_config(fft2d VERSION 1.0.0-p0)
 hunter_config(fixesproto VERSION 5.0)
 hunter_config(flatbuffers VERSION 1.8.0-p1)
 hunter_config(flex VERSION 2.6.4)
-hunter_config(fmt VERSION 4.0.0)
+hunter_config(fmt VERSION 4.1.0)
+hunter_config(folly VERSION 2017.11.13.00-p0)
 hunter_config(freetype VERSION 2.6.2)
-hunter_config(gauze VERSION 0.3.2)
+hunter_config(frugally-deep VERSION 0.2.2-p0)
+hunter_config(gauze VERSION 0.4.0)
 hunter_config(gemmlowp VERSION 1.0.0)
 hunter_config(geos VERSION 3.4.2)
 hunter_config(giflib VERSION 5.1.4-p0)
@@ -210,9 +246,9 @@ hunter_config(glfw VERSION 3.3.0-p4)
 hunter_config(glib VERSION 2.54.0)
 hunter_config(glm VERSION 0.9.8.5)
 hunter_config(globjects VERSION 1.1.0-p0)
-hunter_config(glog VERSION 0.3.5-p1)
+hunter_config(glog VERSION 0.3.5-p2)
 hunter_config(glproto VERSION 1.4.17)
-hunter_config(gRPC VERSION 1.8.1)
+hunter_config(gRPC VERSION 1.9.1-p0)
 hunter_config(gst_plugins_bad VERSION 1.10.4)
 hunter_config(gst_plugins_base VERSION 1.10.4)
 hunter_config(gst_plugins_good VERSION 1.10.4)
@@ -222,6 +258,8 @@ hunter_config(gumbo VERSION 0.10.1)
 hunter_config(half VERSION 1.1.0-p1)
 hunter_config(hdf5 VERSION 1.8.15-p1)
 hunter_config(highwayhash VERSION 0.0.0)
+hunter_config(http-parser VERSION 2.8.0)
+hunter_config(h3 VERSION 3.0.4)
 hunter_config(ice VERSION 1.0.8)
 hunter_config(imshow VERSION 1.0.0-p0)
 hunter_config(inja VERSION 0.1.1)
@@ -240,7 +278,7 @@ hunter_config(libdill VERSION 1.6)
 hunter_config(libjson-rpc-cpp VERSION 0.7.0-p3)
 hunter_config(libmill VERSION 1.18)
```

```diff
 hunter_config(libogg VERSION 1.3.2-cmake3)
-hunter_config(libscrypt VERSION 1.21-p0)
+hunter_config(libscrypt VERSION 1.21-p1)
 hunter_config(libsodium VERSION 1.0.10)
 hunter_config(libuv VERSION 1.14.0-p1)
 hunter_config(libxml2 VERSION 2.9.7)
@@ -253,13 +291,15 @@ hunter_config(lzma VERSION 5.2.3-p4)
 hunter_config(md5 VERSION 1.6)
 hunter_config(mini_chromium VERSION 0.0.1-p2)
 hunter_config(minizip VERSION 1.0.1-p3)
+hunter_config(mng VERSION 2.0.3-p2)
 hunter_config(mojoshader VERSION 0.0.1)
 hunter_config(mongoose VERSION 6.10)
 hunter_config(mpark_variant VERSION 1.0.0)
 hunter_config(msgpack VERSION 1.4.1-p2)
 hunter_config(mtplz VERSION 0.1-p3)
 hunter_config(nanoflann VERSION 1.2.3-p0)
-hunter_config(nlohmann_json VERSION 3.1.0)
+hunter_config(ncnn VERSION 20180314-p2)
+hunter_config(nlohmann_json VERSION 3.1.2)
 hunter_config(nsync VERSION 1.14-p1)
 hunter_config(odb VERSION 2.4.0)
 hunter_config(odb-boost VERSION 2.4.0)
@@ -267,15 +307,18 @@ hunter_config(odb-compiler VERSION 2.4.0)
 hunter_config(odb-mysql VERSION 2.4.0)
 hunter_config(odb-pgsql VERSION 2.4.0)
 hunter_config(odb-sqlite VERSION 2.4.0)
-hunter_config(ogles_gpgpu VERSION 0.2.8)
+hunter_config(ogles_gpgpu VERSION 0.2.10)
+hunter_config(oniguruma VERSION 6.8.1-p0)
 hunter_config(onmt VERSION 0.4.1-p2)
 hunter_config(openddlparser VERSION 0.1.0-p2)
-hunter_config(opentracing-cpp VERSION 1.1.0)
+hunter_config(opentracing-cpp VERSION 1.4.0)
+hunter_config(pcg VERSION 0.0.0-p1)
 hunter_config(pciaccess VERSION 0.13.4)
 hunter_config(libpcre VERSION 8.41)
 hunter_config(poly2tri VERSION 1.0.0)
 hunter_config(polyclipping VERSION 4.8.8-p0) # for Assimp
 hunter_config(presentproto VERSION 1.0)
+hunter_config(protobuf-c VERSION 1.3.0-p1)
 hunter_config(pthread-stubs VERSION 0.3)
 hunter_config(pugixml VERSION 1.8.1)
 hunter_config(pybind11 VERSION 2.2.1)
@@ -290,6 +333,16 @@ endif()
 hunter_config(re2 VERSION 2017.11.01-p0)
 hunter_config(recastnavigation VERSION 1.4-p0)
 hunter_config(renderproto VERSION 0.11.1)
+hunter_config(ros_console_bridge VERSION 0.4.0-p0)
+hunter_config(ros_gencpp VERSION 0.6.0-p0)
+hunter_config(ros_geneus VERSION 2.2.6-p0)
+hunter_config(ros_genlisp VERSION 0.4.16-p0)
+hunter_config(ros_genmsg VERSION 0.5.10-p0)
+hunter_config(ros_gennodejs VERSION 2.0.1-p0)
+hunter_config(ros_genpy VERSION 0.6.7-p0)
+hunter_config(ros_message_generation VERSION 0.4.0-p0)
+hunter_config(roscpp_core VERSION 0.6.9-p0)
+hunter_config(rospack VERSION 2.5.0-p0)
```

```
 hunter_config(sm VERSION 1.2.1)
 hunter_config(Snappy VERSION 1.1.6-p0)
 hunter_config(sse2neon VERSION 1.0.0-p0)
@@ -299,17 +352,20 @@ if(MSVC_VERSION LESS 1800)
     # for VS12 - version without support C++11
   hunter_config(spdlog VERSION 1.0.0-p0)
 else()
-  hunter_config(spdlog VERSION 0.13.0-p1)
+  hunter_config(spdlog VERSION 0.16.3-p1)
 endif()
 hunter_config(stb VERSION 0.0.1)
 hunter_config(szip VERSION 2.1.0-p1)
 hunter_config(Tesseract VERSION 3.05.01-hunter-3)
 hunter_config(thread-pool-cpp VERSION 1.1.0)
-hunter_config(thrift VERSION 0.10.0-p2)
+hunter_config(thrift VERSION 0.11.0-p0)
 hunter_config(tinydir VERSION 1.2-p0)
-hunter_config(type_safe VERSION 0.1)
+hunter_config(tinyxml2 VERSION 6.2.0-p1)
+hunter_config(type_safe VERSION 0.2)
 hunter_config(util_linux VERSION 2.30.1)
-hunter_config(websocketpp VERSION 0.7.0-p2)
+hunter_config(WebKit VERSION 0.0.2-p0)
+hunter_config(WebP VERSION 0.6.1-p3)
+hunter_config(websocketpp VERSION 0.7.0-p3)
 hunter_config(wxWidgets VERSION 3.0.2)
 hunter_config(x11 VERSION 1.5.0)
 hunter_config(x264 VERSION snapshot-20170420-2245)
@@ -355,7 +411,7 @@ if(ANDROID)
   if(_is_api_21)
     hunter_config(Android-Google-APIs VERSION 21_r01)
     hunter_config(Android-Google-APIs-Intel-x86-Atom-System-Image VERSION 21_r10)
-    hunter_config(Android-Intel-x86-Atom-System-Image VERSION 21)
+    hunter_config(Android-Intel-x86-Atom-System-Image VERSION 21_r05)
     hunter_config(Android-SDK-Platform VERSION 21_r02)
     hunter_config(Sources-for-Android-SDK VERSION 21)
     hunter_config(Android-ARM-EABI-v7a-System-Image VERSION 21_r04)
@@ -371,6 +427,7 @@ if(ANDROID)
     hunter_config(Android-SDK-Platform VERSION 16_r05)
     hunter_config(Sources-for-Android-SDK VERSION 16)
     hunter_config(Android-ARM-EABI-v7a-System-Image VERSION 16_r04)
+    hunter_config(Android-MIPS-System-Image VERSION 16_r04)
   elseif(_is_api_24)
     hunter_config(Android-Google-APIs VERSION 24_r1)
     hunter_config(Android-Google-APIs-Intel-x86-Atom-System-Image VERSION 24_r20)
@@ -378,6 +435,7 @@ if(ANDROID)
     hunter_config(Android-SDK-Platform VERSION 24_r02)
     hunter_config(Sources-for-Android-SDK VERSION 24)
     hunter_config(Android-ARM-EABI-v7a-System-Image VERSION 24_r07)
+    hunter_config(Android-ARM64-v8a-System-Image VERSION 24_r07)
   else()
     hunter_user_error(
         "Android API (CMAKE_SYSTEM_VERSION)"
```

v0.22.X

- CMake 3.2 *required*

- `hunter_config`: `VERSION` can be combined with `GIT_SELF`, `GIT_SUBMODULE` and `URL/SHA1`. E.g. to avoid issues like described *here*.

Internal changes

- `hunter_config` functionality split:
  - `hunter_default_version` used in `cmake/configs/default.cmake`
  - `hunter_config` used by user in custom `config.cmake`
  - `hunter_final_config` used by Hunter internally in unification module (Config-ID)
- `hunter_default_version` verify alphabetical order in `cmake/configs/default.cmake`

New packages and updates

`git diff v0.21.0..v0.22.0 -- cmake/configs/default.cmake`:

```
diff --git a/cmake/configs/default.cmake b/cmake/configs/default.cmake
index cff73a3..1cd7acd 100644
--- a/cmake/configs/default.cmake
+++ b/cmake/configs/default.cmake
@@ -57,9 +57,10 @@ hunter_config(BoostProcess VERSION 0.5)
 hunter_config(BoringSSL VERSION 1.0.0)
 hunter_config(Box2D VERSION 2.3.1-p0)
 hunter_config(CapnProto VERSION 0.6.1)
-hunter_config(catkin VERSION 0.7.11-p1)
+hunter_config(catkin VERSION 0.7.11-p2)
 hunter_config(cctz VERSION 2.2.0)
 hunter_config(CLAPACK VERSION 3.2.1)
+hunter_config(CLI11 VERSION 1.5.3)
 hunter_config(CURL VERSION 7.59.0-p1)
 hunter_config(Clang VERSION 4.0.1-p0)
 hunter_config(ClangToolsExtra VERSION 4.0.1) # Clang
@@ -72,7 +73,7 @@ hunter_config(CsvParserCPlusPlus VERSION 1.0.1)
 hunter_config(Eigen VERSION 3.3.4-p1)
 hunter_config(state_machine VERSION 1.1)
 hunter_config(enet VERSION 1.3.13-p1)
-hunter_config(ethash VERSION 0.1.0)
+hunter_config(ethash VERSION 0.3.0)
 hunter_config(Expat VERSION 2.1.1)
 if(MSVC)
   hunter_config(getopt VERSION 1.0.0-p0)
@@ -83,6 +84,7 @@ hunter_config(ICU VERSION 55.1-p3)
 hunter_config(Igloo VERSION 1.1.1-hunter)
 hunter_config(intsizeof VERSION 2.0.2)
 hunter_config(jansson VERSION 2.11.0)
+hunter_config(jaegertracing VERSION 0.4.1)
 hunter_config(jasper VERSION 2.0.14-p2)
 hunter_config(Jpeg VERSION 9b-p3)
 hunter_config(JsonSpirit VERSION 0.0.4-hunter)
@@ -167,7 +169,7 @@ hunter_config(ZMQPP VERSION 4.1.2)
 hunter_config(ZeroMQ VERSION 4.2.3-p1)
 hunter_config(caffe VERSION rc3-p2)
 hunter_config(acf VERSION 0.1.3)
-hunter_config(Catch VERSION 2.2.1)
```

```
+hunter_config(Catch VERSION 2.2.2)
 hunter_config(aes VERSION 0.0.1-p1)
 hunter_config(aglet VERSION 1.2.2)
 hunter_config(android_arm64_v8a_system_image_packer VERSION 1.0.0)
@@ -209,7 +211,7 @@ hunter_config(damageproto VERSION 1.2.1)
 hunter_config(dbus VERSION 1.10.0-hunter-4)
 hunter_config(debug_assert VERSION 1.3.2)
 hunter_config(dest VERSION 0.8.0-p4)
-hunter_config(dlib VERSION 19.10-p2)
+hunter_config(dlib VERSION 19.12-p0)
 hunter_config(dmlc-core VERSION 0.0.0-p1)
 hunter_config(doctest VERSION 1.2.0)
 hunter_config(double-conversion VERSION 3.0.0)
@@ -256,6 +258,7 @@ hunter_config(gst_plugins_ugly VERSION 1.10.4)
 hunter_config(gstreamer VERSION 1.10.4)
 hunter_config(gumbo VERSION 0.10.1)
 hunter_config(half VERSION 1.1.0-p1)
+hunter_config(harfbuzz VERSION 1.7.6-p0)
 hunter_config(hdf5 VERSION 1.8.15-p1)
 hunter_config(highwayhash VERSION 0.0.0)
 hunter_config(http-parser VERSION 2.8.0)
@@ -277,9 +280,10 @@ hunter_config(libdaemon VERSION 0.14)
 hunter_config(libdill VERSION 1.6)
 hunter_config(libjson-rpc-cpp VERSION 0.7.0-p3)
 hunter_config(libmill VERSION 1.18)
-hunter_config(libogg VERSION 1.3.2-cmake3)
+hunter_config(libogg VERSION 1.3.3-p0)
 hunter_config(libscrypt VERSION 1.21-p1)
 hunter_config(libsodium VERSION 1.0.10)
+hunter_config(libunibreak VERSION 4.0)
 hunter_config(libuv VERSION 1.14.0-p1)
 hunter_config(libxml2 VERSION 2.9.7)
 hunter_config(libyuv VERSION 1514-p3)
@@ -334,6 +338,7 @@ hunter_config(re2 VERSION 2017.11.01-p0)
 hunter_config(recastnavigation VERSION 1.4-p0)
 hunter_config(renderproto VERSION 0.11.1)
 hunter_config(ros_console_bridge VERSION 0.4.0-p0)
+hunter_config(ros_environment VERSION 1.2.0-p0)
 hunter_config(ros_gencpp VERSION 0.6.0-p0)
 hunter_config(ros_geneus VERSION 2.2.6-p0)
 hunter_config(ros_genlisp VERSION 0.4.16-p0)
@@ -341,6 +346,8 @@ hunter_config(ros_genmsg VERSION 0.5.10-p0)
 hunter_config(ros_gennodejs VERSION 2.0.1-p0)
 hunter_config(ros_genpy VERSION 0.6.7-p0)
 hunter_config(ros_message_generation VERSION 0.4.0-p0)
+hunter_config(ros_message_runtime VERSION 0.4.12-p0)
+hunter_config(ros_std_msgs VERSION 0.5.11-p1)
 hunter_config(roscpp_core VERSION 0.6.9-p0)
 hunter_config(rospack VERSION 2.5.0-p0)
 hunter_config(sm VERSION 1.2.1)
@@ -363,6 +370,7 @@ hunter_config(tinydir VERSION 1.2-p0)
 hunter_config(tinyxml2 VERSION 6.2.0-p1)
 hunter_config(type_safe VERSION 0.2)
 hunter_config(util_linux VERSION 2.30.1)
+hunter_config(vorbis VERSION 1.3.6-p1)
 hunter_config(WebKit VERSION 0.0.2-p0)
 hunter_config(WebP VERSION 0.6.1-p3)
 hunter_config(websocketpp VERSION 0.7.0-p3)
```

```
@@ -446,6 +454,6 @@ if(ANDROID)
 endif()

 hunter_config(zookeeper VERSION 3.4.9-p2)
-hunter_config(tacopie VERSION 2.4.0-h1)
+hunter_config(tacopie VERSION 3.2.0-h1)
 hunter_config(cpp_redis VERSION 3.5.0-h1)
 hunter_config(IF97 VERSION 2.1.2)
```

v0.23.X

- Ninja generator learn to use *HUNTER_JOBS_NUMBER*
- New variable *HUNTER_NO_TOOLCHAIN_ID_RECALCULATION*

Internal changes

- Internal cache files forced to use `LF` line ending. Effectively it means that cache files built on Windows now will use `LF` instead of `CRLF` and **all cache from Windows became invalid**. Unified `LF` line ending allow to share cache between macOS/Linux and Windows platforms.

New packages and updates

`git diff v0.22.0..v0.23.0 -- cmake/configs/default.cmake`:

```
diff --git a/cmake/configs/default.cmake b/cmake/configs/default.cmake
index f3637ebb..ac552974 100644
--- a/cmake/configs/default.cmake
+++ b/cmake/configs/default.cmake
@@ -22,7 +22,7 @@ hunter_default_version(Android-Build-Tools VERSION 27.0.3)
 hunter_default_version(Android-Google-Repository VERSION 58)
 hunter_default_version(Android-Modules VERSION 1.0.0)
 hunter_default_version(Android-SDK VERSION 0.0.6)
-hunter_default_version(Android-SDK-Platform-tools VERSION r27.0.1)
+hunter_default_version(Android-SDK-Platform-tools VERSION r28.0.0)
 hunter_default_version(Android-SDK-Tools VERSION 25.2.5)
 hunter_default_version(Android-Support-Repository VERSION 47)
 hunter_default_version(AngelScript VERSION 2.30-p0)
@@ -49,7 +49,7 @@ hunter_default_version(BoringSSL VERSION 1.0.0)
 hunter_default_version(Box2D VERSION 2.3.1-p0)
 hunter_default_version(CLAPACK VERSION 3.2.1)
 hunter_default_version(CLI11 VERSION 1.5.3)
-hunter_default_version(CURL VERSION 7.59.0-p1)
+hunter_default_version(CURL VERSION 7.60.0-p0)
 hunter_default_version(CapnProto VERSION 0.6.1)
 hunter_default_version(Catch VERSION 2.2.2)
 hunter_default_version(Clang VERSION 4.0.1-p0)
@@ -88,10 +88,11 @@ hunter_default_version(Libcxxabi VERSION 3.6.2) # Clang
 hunter_default_version(Libevent VERSION 2.1.8-p4)
 hunter_default_version(Libssh2 VERSION 1.7.0)
 hunter_default_version(Lua VERSION 5.3.2-p2)
+hunter_default_version(Microsoft.GSL VERSION 1.0.0-p0)
 hunter_default_version(MySQL-client VERSION 6.1.9-p0)
 hunter_default_version(NASM VERSION 2.12.02)
```

```
 hunter_default_version(OpenAL VERSION 1.18.2)
-hunter_default_version(OpenBLAS VERSION 0.2.20-p0)
+hunter_default_version(OpenBLAS VERSION 0.3.1-p0)
 hunter_default_version(OpenCL VERSION 2.1-p3)
 hunter_default_version(OpenCL-cpp VERSION 2.0.10-p0)
 hunter_default_version(OpenCV VERSION 3.4.1-p1)
@@ -112,11 +113,15 @@ if(_is_linux OR MINGW)
 elseif(IOS OR ANDROID)
   hunter_default_version(Qt VERSION 5.9.1-p0)
 else()
-  hunter_default_version(Qt VERSION 5.10.1)
+  if(MSVC)
+    hunter_default_version(Qt VERSION 5.10.1)
+  else()
+    hunter_default_version(Qt VERSION 5.11.1)
+  endif()
 endif()

 hunter_default_version(QtAndroidCMake VERSION 1.0.9)
-hunter_default_version(QtCMakeExtra VERSION 1.0.30)
+hunter_default_version(QtCMakeExtra VERSION 1.0.32)
 hunter_default_version(QtQmlManager VERSION 1.0.0)
 hunter_default_version(RapidJSON VERSION 1.1.0)
 hunter_default_version(RapidXML VERSION 1.13)
@@ -135,13 +140,13 @@ hunter_default_version(WDC VERSION 1.1.1)
 hunter_default_version(WTL VERSION 9.1.5321)
 hunter_default_version(Washer VERSION 0.1.2)
 hunter_default_version(WebKit VERSION 0.0.2-p0)
-hunter_default_version(WebP VERSION 0.6.1-p3)
+hunter_default_version(WebP VERSION 0.6.1-p4)
 hunter_default_version(WinSparkle VERSION 0.4.0)
 hunter_default_version(ZLIB VERSION 1.2.8-p3)
 hunter_default_version(ZMQPP VERSION 4.1.2)
 hunter_default_version(ZeroMQ VERSION 4.2.3-p1)

-hunter_default_version(acf VERSION 0.1.3)
+hunter_default_version(acf VERSION 0.1.14)
 hunter_default_version(aes VERSION 0.0.1-p1)
 hunter_default_version(aglet VERSION 1.2.2)
 hunter_default_version(android_arm64_v8a_system_image_packer VERSION 1.0.0)
@@ -163,7 +168,7 @@ hunter_default_version(benchmark VERSION 1.4.0)
 hunter_default_version(bison VERSION 3.0.4-p0)
 hunter_default_version(boost-pba VERSION 1.0.0-p0)
 hunter_default_version(bullet VERSION 2.87-p0)
-hunter_default_version(c-ares VERSION 1.13.0)
+hunter_default_version(c-ares VERSION 1.14.0-p0)
 hunter_default_version(caffe VERSION rc3-p2)
 hunter_default_version(catkin VERSION 0.7.11-p2)
 hunter_default_version(cctz VERSION 2.2.0)
@@ -183,13 +188,14 @@ hunter_default_version(cryptopp VERSION 5.6.5-p0)
 hunter_default_version(cub VERSION 1.7.4-p0)
 hunter_default_version(cvmatio VERSION 1.0.28)
 hunter_default_version(cvsteer VERSION 0.1.2)
-hunter_default_version(cxxopts VERSION 1.0.0-p0)
+hunter_default_version(cxxopts VERSION 2.1.1-pre)
 hunter_default_version(czmq VERSION 4.0.2-p1)
 hunter_default_version(damageproto VERSION 1.2.1)
+hunter_default_version(date VERSION 2.4.1)
```

```diff
 hunter_default_version(dbus VERSION 1.10.0-hunter-4)
 hunter_default_version(debug_assert VERSION 1.3.2)
 hunter_default_version(dest VERSION 0.8.0-p4)
-hunter_default_version(dlib VERSION 19.12-p0)
+hunter_default_version(dlib VERSION 19.14-p0)
 hunter_default_version(dmlc-core VERSION 0.0.0-p1)
 hunter_default_version(doctest VERSION 1.2.0)
 hunter_default_version(double-conversion VERSION 3.0.0)
@@ -199,7 +205,7 @@ hunter_default_version(drishti VERSION 0.8.9)
 hunter_default_version(drishti_assets VERSION 1.8)
 hunter_default_version(drishti_faces VERSION 1.2)
 hunter_default_version(drm VERSION 2.4.67)
-hunter_default_version(duktape VERSION 1.5.2-p0)
+hunter_default_version(duktape VERSION 2.2.1-p0)
 hunter_default_version(dynalo VERSION 1.0.3)
 hunter_default_version(eigen3-nnls VERSION 1.0.1)
 hunter_default_version(enet VERSION 1.3.13-p1)
@@ -214,28 +220,29 @@ hunter_default_version(fmt VERSION 4.1.0)
 hunter_default_version(folly VERSION 2017.11.13.00-p0)
 hunter_default_version(freetype VERSION 2.6.2)
 hunter_default_version(frugally-deep VERSION 0.2.2-p0)
-hunter_default_version(gRPC VERSION 1.9.1-p0)
-hunter_default_version(gauze VERSION 0.4.0)
+hunter_default_version(gRPC VERSION 1.12.1-p0)
+hunter_default_version(gauze VERSION 0.5.0)
 hunter_default_version(gemmlowp VERSION 1.0.0)
 hunter_default_version(geos VERSION 3.4.2)
 hunter_default_version(getopt VERSION 1.0.0-p0)
 hunter_default_version(gflags VERSION 2.2.1)
-hunter_default_version(giflib VERSION 5.1.4-p0)
+hunter_default_version(giflib VERSION 5.1.4-p1)
 hunter_default_version(glbinding VERSION 2.1.3-p0)
 hunter_default_version(glew VERSION 2.0.0-p1)
 hunter_default_version(glfw VERSION 3.3.0-p4)
 hunter_default_version(glib VERSION 2.54.0)
-hunter_default_version(glm VERSION 0.9.8.5)
+hunter_default_version(glm VERSION 0.9.9.0)
 hunter_default_version(globjects VERSION 1.1.0-p0)
 hunter_default_version(glog VERSION 0.3.5-p2)
 hunter_default_version(glproto VERSION 1.4.17)
+hunter_default_version(glslang VERSION 7.7.2767-p0)
 hunter_default_version(gst_plugins_bad VERSION 1.10.4)
 hunter_default_version(gst_plugins_base VERSION 1.10.4)
 hunter_default_version(gst_plugins_good VERSION 1.10.4)
 hunter_default_version(gst_plugins_ugly VERSION 1.10.4)
 hunter_default_version(gstreamer VERSION 1.10.4)
 hunter_default_version(gumbo VERSION 0.10.1)
-hunter_default_version(h3 VERSION 3.0.4)
+hunter_default_version(h3 VERSION 3.0.7)
 hunter_default_version(half VERSION 1.1.0-p1)
 hunter_default_version(harfbuzz VERSION 1.7.6-p0)
 hunter_default_version(hdf5 VERSION 1.8.15-p1)
@@ -252,7 +259,7 @@ hunter_default_version(ippicv VERSION 20151201)
 hunter_default_version(irrXML VERSION 1.2)
 hunter_default_version(jaegertracing VERSION 0.4.1)
 hunter_default_version(jansson VERSION 2.11.0)
-hunter_default_version(jasper VERSION 2.0.14-p2)
+hunter_default_version(jasper VERSION 2.0.14-p3)
```

```
 hunter_default_version(jo_jpeg VERSION 0.0.1)

 if(MSVC_VERSION LESS 1600)
@@ -277,9 +284,9 @@ hunter_default_version(libogg VERSION 1.3.3-p0)
 hunter_default_version(libpcre VERSION 8.41)
 hunter_default_version(librtmp VERSION 2.4.0-p0)
 hunter_default_version(libscrypt VERSION 1.21-p1)
-hunter_default_version(libsodium VERSION 1.0.10)
+hunter_default_version(libsodium VERSION 1.0.16)
 hunter_default_version(libunibreak VERSION 4.0)
-hunter_default_version(libuv VERSION 1.14.0-p1)
+hunter_default_version(libuv VERSION 1.21.0-p0)
 hunter_default_version(libxml2 VERSION 2.9.7)
 hunter_default_version(libyuv VERSION 1514-p3)
 hunter_default_version(lmdb VERSION 0.9.21-p2)
@@ -310,7 +317,7 @@ hunter_default_version(ogles_gpgpu VERSION 0.2.10)
 hunter_default_version(oniguruma VERSION 6.8.1-p0)
 hunter_default_version(onmt VERSION 0.4.1-p2)
 hunter_default_version(openddlparser VERSION 0.1.0-p2)
-hunter_default_version(opentracing-cpp VERSION 1.4.0)
+hunter_default_version(opentracing-cpp VERSION 1.5.0)
 hunter_default_version(pcg VERSION 0.0.0-p1)
 hunter_default_version(pciaccess VERSION 0.13.4)
 hunter_default_version(poly2tri VERSION 1.0.0)
@@ -333,7 +340,10 @@ endif()
 hunter_default_version(re2 VERSION 2017.11.01-p0)
 hunter_default_version(recastnavigation VERSION 1.4-p0)
 hunter_default_version(renderproto VERSION 0.11.1)
-hunter_default_version(rocksdb VERSION 5.8.6)
+hunter_default_version(rocksdb VERSION 5.14.2)
+hunter_default_version(ros VERSION 1.14.4-p0)
+hunter_default_version(ros_comm_msgs VERSION 1.11.2-p0)
+hunter_default_version(ros_common_msgs VERSION 1.12.6-p0)
 hunter_default_version(ros_console_bridge VERSION 0.4.0-p0)
 hunter_default_version(ros_environment VERSION 1.2.0-p0)
 hunter_default_version(ros_gencpp VERSION 0.6.0-p0)
@@ -359,7 +369,7 @@ else()
   hunter_default_version(spdlog VERSION 0.16.3-p1)
 endif()

-hunter_default_version(sqlite3 VERSION 3.21.0-p2)
+hunter_default_version(sqlite3 VERSION 3.24.0-p0)
 hunter_default_version(sse2neon VERSION 1.0.0-p0)
 hunter_default_version(stanhull VERSION 0.0.1)
 hunter_default_version(state_machine VERSION 1.1)
```

Index

D

F

G

I

J

L

M

N

O

P

Q

R

W

X

Y