

Wake up from the nightmare of using external dependencies with Conan

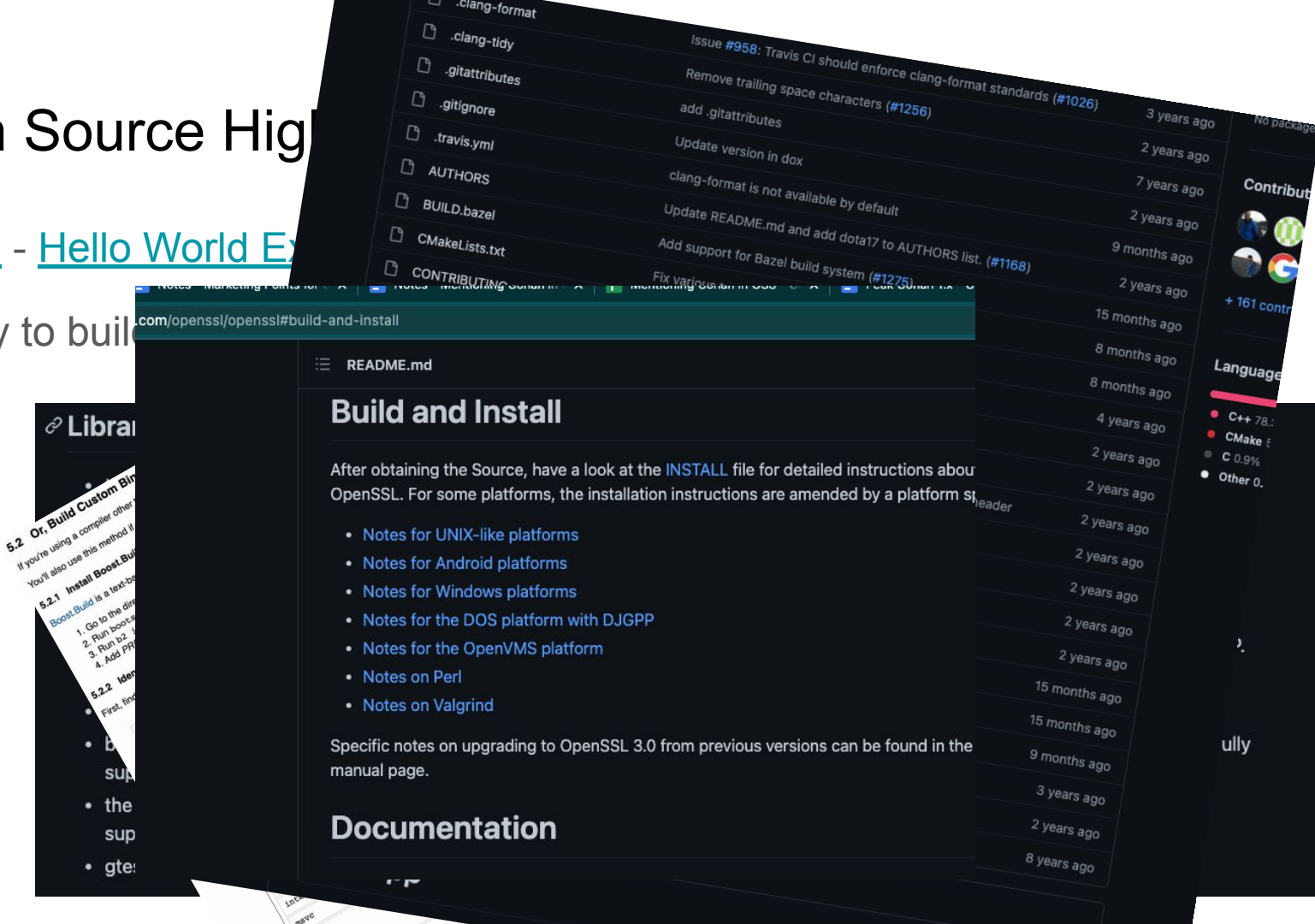
Christopher McArthur

Bay Area C++ Meetup
August 10th, 2022

Open Source High

[Drogon](#) - [Hello World Example](#)

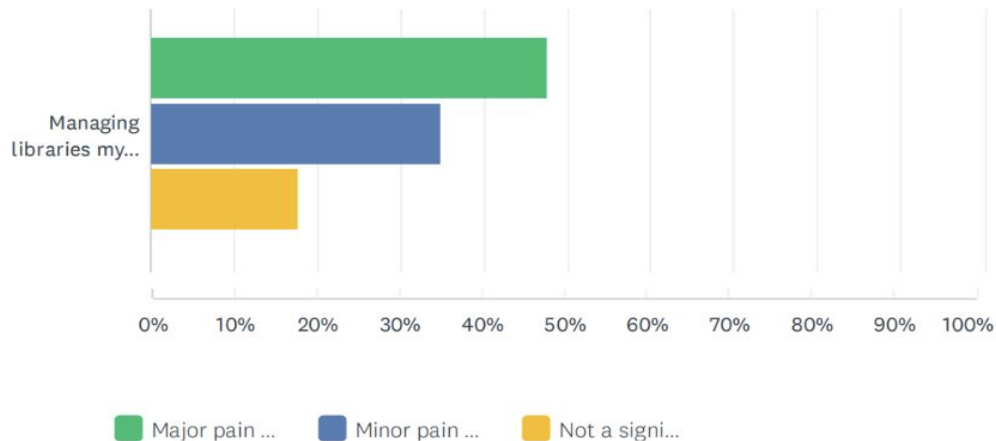
Let's try to build



A problem we are all familiar with

C++ ISO's "Lite" results - Q6 Which of these do you find frustrating about C++ development?

	MAJOR PAIN POINT	MINOR PAIN POINT	NOT A SIGNIFICANT ISSUE FOR ME	TOTAL	WEIGHTED AVERAGE
Managing libraries my application depends on	47.63% 563	34.77% 411	17.60% 208	1,182	2.30



More than 80% of us share this pain!

So how do we address
this problem?

Package Managers

Why package managers?

- Quick and easy access to the pool of quality open source.
- Zero worries about dependency graph
- Integrate with the technologies you are already using
- Better awareness about platform compatibility

There's lot of options

- System or distros (apt/brew/conda)
 - Literally the universe
- Build from source (hunter/vcpkg/CMake's Fetch Content/b2)
 - Gets the job done, time consuming, build system specific
- Binary management (Conan)
 - Endless optimizable. Quick installation. All build systems

Big takeaway - Just use any!

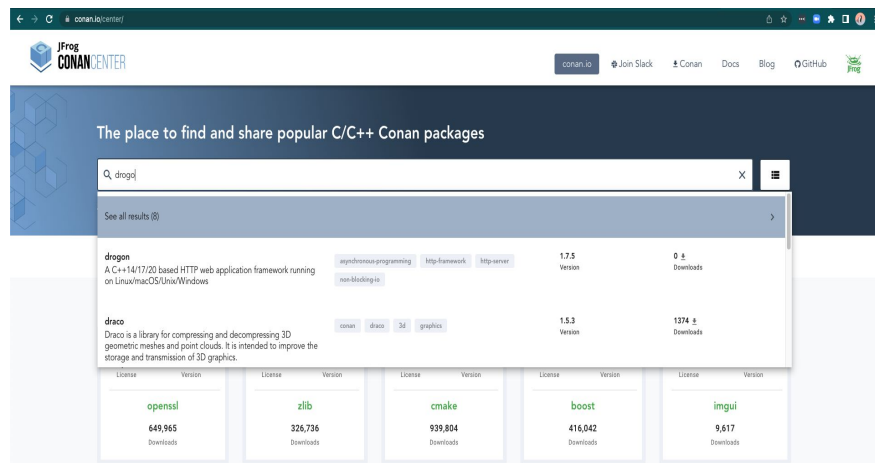
What is Conan?

The C++ Package Manager

- All platforms (it's Python)
- All build system (tons of integrations)
- Absolute **visibility on dependencies** graph (Huge improvements in Conan 2.0)
- Create and re-use binaries
- Decentralized (host your own recipes and packages)
- **Extensible, Customizable**

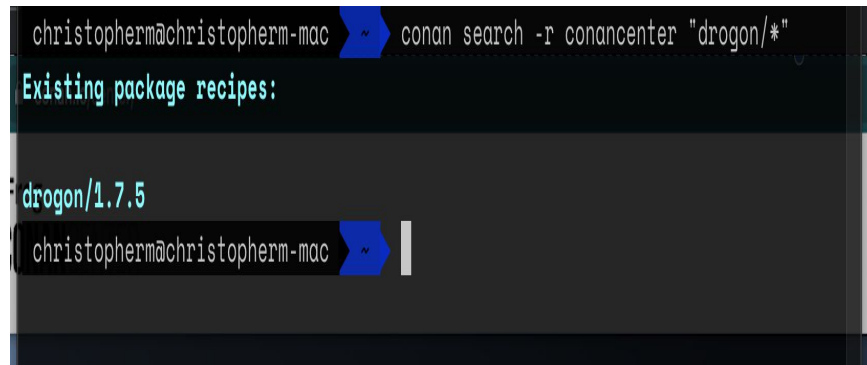
Search

- Web UI <https://conan.io/center>



- Command line

conan search -r conancenter "drogon/*"



Install

We can manage binaries for different compilers, versions, etc. Stay Tuned!

Download existing
conan install drogon/1.7.5@ -s compiler.version=12.0 -g CMakeDeps

Build form from source (specific settings)
conan install drogon/1.7.5@ -g CMakeDeps --build missing

```
Installing package: drogon/1.7.5
Requirements
boost/1.79.0 from 'conancenter' - Downloaded
bzip2/1.0.8 from 'conancenter' - Downloaded
c-ares/1.18.1 from 'conancenter' - Downloaded
drogon/1.7.5 from 'conancenter' - Downloaded
jsoncpp/1.9.5 from 'conancenter' - Downloaded
libbacktrace/ccl.20210118 from 'conancenter' - Downloaded
libiconv/1.17 from 'conancenter' - Downloaded
openssl/1.1.1o from 'conancenter' - Downloaded
trantor/1.5.5 from 'conancenter' - Downloaded
zlib/1.2.12 from 'conancenter' - Downloaded

Packages
boost/1.79.0:b21b89907b3b08decebf99595db53e7819cef80f - Download
bzip2/1.0.8:75fe66bf8e61230e9fcb361f13b6b3baa483e0a - Download
c-ares/1.18.1:49195aea8e76fd7c4eaf35bf5e4bce294641381f - Download
drogon/1.7.5:bc6348af19ba7a8bbd31b7f79ade4c9da88d585b - Download
jsoncpp/1.9.5:6a83df7f783e7ee89a83cf2fe72b5ff676758e2a6 - Download
libbacktrace/ccl.20210118:647afeb9d3b0a2d3d316e80b24d38c714cc6900 - Download
libiconv/1.17:647afeb9d3b0a2d3d316e80b24d38c714cc6900 - Download
openssl/1.1.1o:647afeb9d3b0a2d3d316e80b24d38c714cc6900 - Download
trantor/1.5.5:2a2c76b73c378d084a80384d2ac5ecb772ce102c - Download
zlib/1.2.12:647afeb9d3b0a2d3d316e80b24d38c714cc6900 - Download

Installing (downloading, building) binaries...
bzip2/1.0.8: Retrieving package 75fe66bf8e61230e9fcb361f13b6b3baa483e0a from remote 'conancenter'
Downloading conanmanifest.txt completed [0.32k]
Downloading conaninfo.txt completed [0.46k]
Downloading conan_package.tgz: 100%|#####| 87.7k/87.7k [00:00<00:00, 769kB/
Downloading conan_package.tgz completed [87.67k]
```

```
[ 75%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/ArrayParser.cc.o
[ 76%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/Criteria.cc.o
[ 78%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/DbClient.cc.o
[ 80%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/DbClientImpl.cc.o
[ 82%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/DbClientLockFree.cc.o
[ 83%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/DbConnection.cc.o
[ 85%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/Exception.cc.o
[ 87%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/Field.cc.o
[ 89%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/Result.cc.o
[ 91%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/Row.cc.o
[ 92%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/SqlBinder.cc.o
[ 94%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/TransactionImpl.cc.o
[ 96%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/arm_lib/src/RestfulController.cc.o
[ 98%] Building CXX object source_subfolder/CMakeFiles/drogon.dir/lib/src/DbClientManagerSkipped.cc.o
[100%] Linking CXX static library ../lib/libdrogon.a
[100%] Built target drogon
drogon/1.7.8: Package '6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1' built
drogon/1.7.8: Build folder /Users/christopher/.conan/data/drogon/1.7.5/_/_/build/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1
drogon/1.7.8: Generated conaninfo.txt
drogon/1.7.8: Generated conanbuildinfo.txt
drogon/1.7.8: Generating the package
drogon/1.7.8: Package folder /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1
drogon/1.7.8: Calling package()
Consolidate compiler generated dependencies of target drogon
[100%] Built target drogon
Install the project...
-- Install configuration: "Release"
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/lib/libdrogon.a
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/Attribute.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/CacheMap.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/Cookie.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/DrClassMap.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/DrObject.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/DrTemplate.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/DrTemplateBase.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/HttpBinder.h
-- Installing: /Users/christopher/.conan/data/drogon/1.7.5/_/_/package/6ece8ac241a0880e1e8e514fb3fd12f9ff2e32b1/include/drogon/HttpClient.h
```

Install

```
conan install drogon/1.7.5@ -s compiler.version=12.0 -g CMakeDeps
```

Installs a specific package
reference “name/version”

Change settings

Generate
“find_package”
CMake

```
CMakeDeps' calling 'generate()'
env generators
erm@christopherm-mac /tmp/testing23e234234$ ls -la

69 christopherm wheel 2208 Aug 10 13:15 .
20 root wheel 640 Aug 10 13:14 ..
1 christopherm wheel 3756 Aug 10 13:15 BZip2-Target-release.cmake
1 christopherm wheel 1430 Aug 10 13:15 BZip2-release-x86_64-data.cmake
1 christopherm wheel 1304 Aug 10 13:15 BZip2Config.cmake
1 christopherm wheel 574 Aug 10 13:15 BZip2ConfigVersion.cmake
1 christopherm wheel 855 Aug 10 13:15 BZip2Targets.cmake
1 christopherm wheel 134835 Aug 10 13:15 Boost-Target-release.cmake
1 christopherm wheel 56701 Aug 10 13:15 Boost-release-x86_64-data.cmake
1 christopherm wheel 1305 Aug 10 13:15 BoostConfig.cmake
1 christopherm wheel 577 Aug 10 13:15 BoostConfigVersion.cmake
1 christopherm wheel 855 Aug 10 13:15 BoostTargets.cmake
1 christopherm wheel 3878 Aug 10 13:15 Drogon-Target-release.cmake
1 christopherm wheel 1594 Aug 10 13:15 Drogon-release-x86_64-data.cmake
1 christopherm wheel 1316 Aug 10 13:15 DrogonConfig.cmake
1 christopherm wheel 574 Aug 10 13:15 DrogonConfigVersion.cmake
1 christopherm wheel 864 Aug 10 13:15 DrogonTargets.cmake
1 christopherm wheel 1604 Aug 10 13:15 FindBZip2.cmake
1 christopherm wheel 1620 Aug 10 13:15 FindIconv.cmake
1 christopherm wheel 1644 Aug 10 13:15 FindOpenSSL.cmake
1 christopherm wheel 1587 Aug 10 13:15 FindZLIB.cmake
```

```
conan install drogon/1.7.5@ -g CMakeDeps --build missing
```

Use the detected system
as default settings

Determine which
binaries are missing
(from remote and local)
then build them from
source

Conan 2.0:

```
conan install --reference drogon/1.7.5
```

Create a project

Let's create a new project for our "Hello World" example

```
conan new hello/0.1 --template=cmake_exe
```

```
File saved: CMakeLists.txt
```

```
File saved: conanfile.py
```

```
File saved: include/hello.h
```

```
File saved: src/hello.cpp
```

```
File saved: src/main.cpp
```

This gives us a basic CMake project with an executable target called "hello" with include and src directories

- This is the default "cmake_layout" that Conan provides

Conan 2.0:

```
conan new cmake_exe -d name=hello -d version=0.1
```

First look at our Conanfile.py

Basic Attributes

```
class HelloConan(ConanFile):  
    name = "hello"  
    version = "0.1"  
  
    # Optional metadata  
    license = "<Put the package license here>"  
    author = "<Put your name here> <And your email here>"  
    url = "<Package recipe repository url here, for issues about the package>"  
    description = "<Description of Hello here>"  
    topics = ("<Put some tag here>", "<here>", "<and here>")  
  
    # Binary configuration  
    settings = "os", "compiler", "build_type", "arch"
```

First look at our Conanfile.py

Integration with CMake

```
def layout(self):
    cmake_layout(self)

def generate(self):
    tc = CMakeToolchain(self)
    tc.generate()

def build(self):
    cmake = CMake(self)
    cmake.configure()
    cmake.build()

def package(self):
    cmake = CMake(self)
    cmake.install()
```

Conan install creates the files to tell CMake what we are doing (some settings) using [CMake Toolchains](#) and where to load our files

Compile with “[cmake --build](#)”

Install with “[cmake --install](#)”

Conan's CMake
integration in
transparent!

Build and run hello world

```
conan install --profile:build default conanfile.py
```

```
cmake .. --preset release
```

```
Preset CMake variables:

CMAKE_BUILD_TYPE="Release"
CMAKE_POLICY_DEFAULT_CMP0091="NEW"
CMAKE_TOOLCHAIN_FILE:FILEPATH="/private/tmp/my-self-contained-project/build/generators/conan_toolchain.cmake"

-- Using Conan toolchain: /tmp/my-self-contained-project/build/generators/conan_toolchain.cmake
-- The CXX compiler identification is AppleClang 13.1.6.13160021
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
```

```
cmake --build build/Release
```

```
[ 33%] Building CXX object CMakeFiles/helo.dir/src/helo.cpp.o
[ 66%] Building CXX object CMakeFiles/helo.dir/src/main.cpp.o
[100%] Linking CXX executable helo
[100%] Built target helo
```

```
./build/bin/hello
```

```
helo/0.1: Hello World Release!
helo/0.1: __x86_64__ defined
helo/0.1: __cplusplus199711
helo/0.1: __GNUC__4
helo/0.1: __GNUC_MINOR__2
helo/0.1: __clang_major__13
```

```
conan install -pr:b default conanfile
```

Configuration:

```
[settings]
arch=x86_64
arch_build=x86_64
build_type=Release
compiler=apple-clang
compiler.libcxx=libc++
compiler.version=13
os=Macos
os_build=Macos
[options]
[build_requires]
[env]
```

conanfile.py (helo/0.1): Installing package

Requirements

Packages

Installing (downloading, building) binaries...

Adding our dependency

Step 1 - Conanfile.py

```
24
25+     def requirements(self):
26+         self.requires("drogon/1.7.5")
27+
28     def generate(self):
29+         deps = CMakeDeps(self)
30+         deps.generate()
31
32         tc = CMakeToolchain(self)
33         tc.generate()
```

Step 2 - CMakeLists.txt

```
1  cmake_minimum_required(VERSION 3.15)
2  project(hello CXX)
3
4+ find_package(Drogon CONFIG REQUIRED)
5+
6  add_executable(hello src/hello.cpp src/main.cpp)
7  target_include_directories(hello PUBLIC include)
8+ target_compile_features(hello PUBLIC cxx_std_14)
9+ target_link_libraries(hello PUBLIC Drogon::Drogon)
10
11 install(TARGETS hello)
12
```

Step 3 - Copy Main.cpp

```
1  #include "hello.h"
2+ #include <drogon/drogon.h>
3+ using namespace drogon;
4
5  int main() {
6      hello();
7+      app().registerHandler(
```


Congratulations.
That's all.

Let's Try It

<https://github.com/drogonframework/drogon/tree/master/examples/helloworld>

conan install conanfile.py

cmake . --preset release

cmake --build build/Release

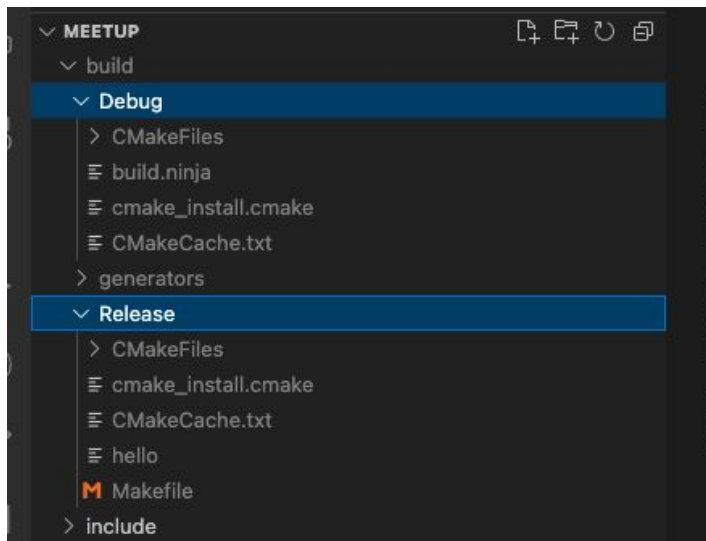
./build/Release/hello

```
hello/0.1: Hello World Release!
hello/0.1: __x86_64__ defined
hello/0.1: __cplusplus201402
hello/0.1: __GNUC__4
hello/0.1: __GNUC_MINOR__2
hello/0.1: __clang_major__13
hello/0.1: __clang_minor__1
hello/0.1: __apple_build_version__13160021
20220802 22:09:25.259347 UTC 6398398 INFO Server running on 127.0.0.1:8848 - m
in.cpp:25
^C20220802 22:09:28.095683 UTC 6398398 WARN SIGINT signal received. - HttpAppF
```

Setting and Config

Ninja Debug Build

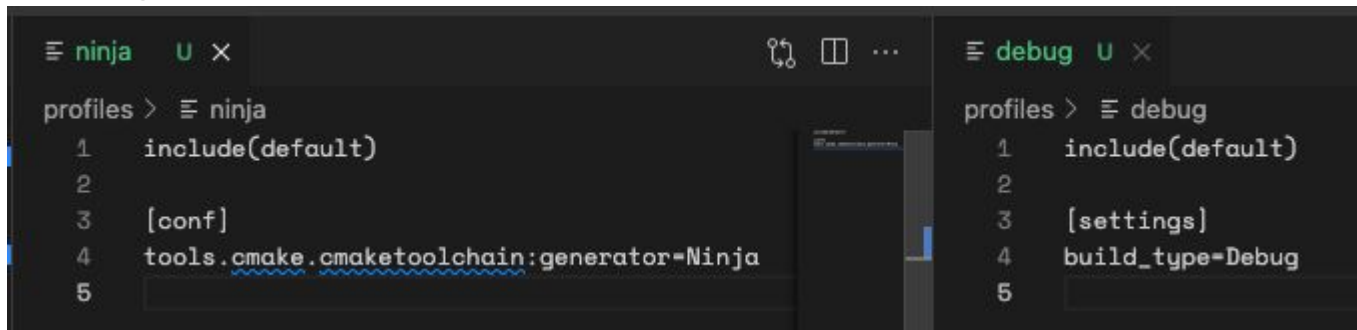
```
conan install conanfile.py -c  
tools.cmake.cmaketoolchain:generator=Ninja  
a -s build_type=Debug
```



- Settings: General attributes that affect all binaries (not specific to a project, see “Options”)
- Configuration: Properties you can choose when working with different build system or tools.

Making profiles

Let's create two profiles, install them and trying building our example



The image shows two side-by-side terminal windows. The left window is titled 'ninja' and shows the creation of a Conan profile named 'ninja'. The right window is titled 'debug' and shows the creation of a Conan profile named 'debug'. Both profiles inherit from 'default' and specify a generator.

```
profiles > ninja
1 include(default)
2
3 [conf]
4 tools.cmake.cmaketoolchain:generator=Ninja
5

profiles > debug
1 include(default)
2
3 [settings]
4 build_type=Debug
5
```

conan install -pr ninja -pr debug conanfile.py

We could have made one profile with both values

Publishing our package

conan create conanfile.py

conan upload hello/0.1

OSS Author?

Submit a PR to conan center index!



JFrog
CONANCENTER

Documentation

- [Code of conduct](#)
- [Changelog](#)
- [User documentation](#)
 - [Contributing to Conan Center Index](#)
 - [Adding Packages to ConanCenter](#)
 - [Policy about patching](#)
 - [Review Process](#)