

Conan Cheat Sheet

Last edited by [VREDDY10](#) 6 months ago

Conan 1.x Cheat Sheet

CONAN 1.X CHEATSHEET

Conan 2.0 is now out! please head to [conan.io](#) to learn more

Show Local Client Configuration

Conan application configuration

```
$ conan config get
```

Contents of a profile (eg. default)

```
$ conan profile show default
```

Remote Repositories

```
$ conan remote list
```

Add and modify configurations

Install collection of configs

```
$ conan config install <url>
```

Change a single config value

```
$ conan config set general.revisions_enabled=1
```

Add a remote

```
$ conan remote add my_remote <url>
```

Provide credentials for remote

```
$ conan user -p <password> -r my_remote <username>
```

Display information from recipes or references

Displays attributes of conanfile.py

```
$ conan inspect <path> -a <attribute>
```

Displays content of conanfile.py for a reference

```
$ conan get <reference>
```

Display dependency graph info for a recipe

```
$ conan info <path_or_reference>
```

Search Packages

Search for packages in a remote

```
$ conan search zlib -r conancenter
```

Consume Packages

Install package using just a reference

```
$ conan install <package_reference>
```

Install list of packages from conanfile

```
$ cat conanfile.txt
[requires]
zlib/1.2.11
$ conan install <path_to_conanfile>
```

Consume packages in build system via generators

```
$ cat conanfile.txt
[requires]
zlib/1.2.11
[generators]
cmake_find_package
msbuild
make
```

Install requirements and generate files

```
$ mkdir build && cd build
$ conan install ..
```

Run your build system (one of the following)

```
$ cmake .. && cmake --build .
$ msbuild myproject.sln
$ make
```

Create a package

Create a recipe (conanfile.py) from templates

```
$ conan new <reference> -m <template>
```

Just export the recipe to local cache

```
$ conan export <path_to_conanfile>
```

Create package from recipe for one configuration
Also implicitly does install and export steps

```
$ conan create . -pr <profile>
```

Upload a Package

One or more with wildcard support, with binaries

```
$ conan upload zlib* -r remote --all
```

Copy packaged files out of Conan cache

Using the deploy generator

```
$ conan install zlib/1.2.11@ -g deploy
```

Conan Recipe Methods in Package Creation

The diagram illustrates the Conan recipe creation process. It starts with a box labeled 'package.info()' with the text 'From all dependency recipes' below it. An arrow points down to a sequence of four boxes: 'exports exports_sources source()', 'generate()', 'build()', and 'package()'. A feedback arrow points from the 'build()' box back to the 'generate()' box.

[Reference for Conan 1.x Cheat Sheet](#)

Conan 2.x Cheat Sheet

CONAN 2.0 CHEATSHEET

Search Packages

Search for packages in a remote

```
$ conan search "zlib/*" -r conancenter
```

Consume Packages

Install package using just a reference

```
$ conan install --requires zlib/1.2.13
```

Install list of packages from conanfile

```
$ cat conanfile.txt
[requires]
zlib/1.2.13
$ conan install . # path to a conanfile
```

Consume packages in build system via generators

```
$ cat conanfile.txt
[requires]
zlib/1.2.13
[generators]
CMakeToolchain
CMakeDeps
[layout]
cmake_layout
```

Install requirements and generate files

```
$ conan install . # path to a conanfile
```

Run your build system (one of the following)

```
# With CMake >= 3.23
$ cmake --preset conan-release
$ cmake --build --preset conan-release
```

Configure local client

Initial Conan application preparation

```
$ conan profile detect
```

Show possible Conan application configuration

```
$ conan config list
```

Contents of a profile (eg. default)

```
$ conan profile show -pr default
```

Install collection of configs

```
$ conan config install <url_or_path>
```

Remote repository configurations

Remote Repositories

```
$ conan remote list
```

Add a remote

```
$ conan remote add my_remote <url>
```

Provide credentials within CI pipeline for a remote

```
$ conan remote login my_remote <username> -p <password>
```

Display information from recipes or references

Displays attributes of conanfile.py

```
$ conan inspect . # path to a conanfile
```

Display dependency graph info for a reference

```
$ conan graph info --requires zlib/1.2.13
```

Display dependency graph info for a recipe

```
$ conan graph info . --format=html > graph.html #
path to a conanfile
```

Create a package

Create a recipe (conanfile.py) from templates

```
$ conan new cmake_lib --define name=hello -d
version=0.1
```

Create package from recipe for one configuration
Also implicitly does install and export steps

```
$ conan create . # path to a conanfile
```

Upload a Package

One or more with wildcard support, with binaries

```
$ conan upload "zlib/*" -r my_remote
```

Copy packaged files out of Conan cache

Using the deploy generator

```
$ conan install --requires zlib/1.2.13 --
deploy full_deploy -g CMakeDeps
```

Conan Recipe Methods in Package Creation

The diagram illustrates the Conan recipe creation process. It starts with a box labeled 'package.info()' with the text 'From all dependency recipes' below it. An arrow points down to a sequence of four boxes: 'exports exports_sources source()', 'generate()', 'build()', and 'package()'. A feedback arrow points from the 'build()' box back to the 'generate()' box.

[Reference for Conan 2.x Cheat Sheet](#)

FAQs at <https://q2a.visteon.com/tag/git/>

https://eu.git.visteon.com/devnext/devnext_v3/-/wikis/Conan-Cheat-Sheet

1/1