

# 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

