CONAN 1.X CHEATSHEET

JFrog | **CONAN** C/C++ Package Manager

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

package_info()

From all dependency recipes

exports
exports_sources
source() → generate() → build() → package()