Conan C/C++ package manager

Recipe conanfile.py → `conan create` → **Local cache** ~/.conan2 → `conan upload` → **Remote** conancenter, Artifactory

CONCEPTS

Conan 2 reads a conanfile.txt or conanfile.py; a profile picks compiler and settings, generators feed your build system.

Settings, options and conf given on the command line override the profile; use -pr:b for the build machine and -pr:h for the target.

Binaries are keyed by a package ID hashed from settings and options; any difference means a different binary.

PROJECT SETUP

`conan profile detect`
Generate the default profile

`conan new cmake_lib -d name=<n> -d version=<v>`
Scaffold a CMake library recipe

`conan new -h`
List the built-in templates

`conan version`
Print the Conan version

`conan config home`
Print the Conan home folder

INSTALL & BUILD

`conan install .`
Install dependencies of this project

`conan install . --build=missing`
Build binaries that are not available

`conan build .`
Install deps and run the recipe's build()

`conan install . -of <dir>`
Write generated files to another folder

`conan install . -g CMakeDeps`
Add a generator

`conan install --requires=<pkg>/<ver>`
Install a package without a recipe

`conan install . -s build_type=Debug`
Override a setting

`conan install . -o <pkg>/*:shared=True`
Set a package option

`conan install . -pr:b <profile> -pr:h <profile>`
Cross-build with two profiles

CREATE PACKAGES

`conan create .`
Build the recipe and test it

`conan create . --build=missing`
Create, building missing deps

`conan create . -tf=""`
Create without running test_package

`conan export .`
Put only the recipe in the cache

`conan test test_package <pkg>/<ver>`
Test an existing package

`conan editable list`
Show editable packages

PROFILES

`conan profile list`
List profiles in the cache

`conan profile show`
Show the resolved default profile

`conan profile show -pr <profile> -s build_type=Debug`
Preview a profile plus overrides

`conan profile path <profile>`
Print a profile's file path

PROFILES (cont.)

`conan profile detect --name <profile>`
Detect into a named profile

REMOTES

`conan remote list`
List remotes

`conan remote add <name> <url>`
Add a remote

`conan remote login <remote> <user>`
Log in to a remote

SEARCH & LIST

`conan search "<pkg>*"`
Search recipes in all remotes

`conan search "<pkg>*" -r <remote>`
Search one remote

`conan list "<pkg>/*"`
List cached versions

`conan list "<pkg>/<ver>:*"`
List cached binaries

`conan list "<pkg>/*" -r <remote>`
List versions on a remote

`conan graph info .`
Show the dependency graph

UPLOAD & CACHE

`conan upload "<pkg>/<ver>" -r <remote>`
Upload recipe and binaries

`conan upload "*" -r <remote> -c`
Upload everything, no prompt

`conan upload "<pkg>/<ver>" -r <remote> --only-recipe`
Upload only the recipe

`conan download "<pkg>/<ver>" -r <remote>`
Download without installing

`conan remove "<pkg>/<ver>"`
Delete from the local cache

`conan remove "<pkg>/*" -c`
Remove all versions, no prompt

`conan cache path <pkg>/<ver>`
Print a recipe's cache folder

`conan cache clean`
Drop source and build folders

LOCKFILES

`conan lock create .`
Freeze dependency versions