Package: conan2 (via r-universe)

June 25, 2024

Title Conan the Librarian

Version 1.9.101

Description Create libraries. For us, there is no spring. Just the wind that smells fresh before the storm.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL https://github.com/mrc-ide/conan2, https://mrc-ide.github.io/conan2

BugReports https://github.com/mrc-ide/conan2/issues

Imports cli, fs, glue, prettyunits, rlang

Suggests callr, mockery, openssl, pkgdepends, remotes, renv, testthat, withr

Language en-GB

Repository https://mrc-ide.r-universe.dev

RemoteUrl https://github.com/mrc-ide/conan2

RemoteRef main

RemoteSha a2994edc1404d6a50f9929b84d6041748402cb3d

Contents

---

conan_compare   *Compare conan installations*

Description

Compare conan installations.

Usage

```
conan_compare(path_lib, curr = 0, prev = -1)
```

Arguments

path_lib	Path to the library to compare
curr	The previous installation to compare against. Can be a name (see conan_list to get names), a negative number where -n indicates "n installations ago" or a positive number where n indicates "the nth installation". The default value of 0 corresponds to the current installation.
prev	The previous installation to compare against. Can be a name (see conan_list to get names), a negative number where -n indicates "n installations ago" or a positive number where n indicates "the nth installation". The default of -1 indicates the previous installation. Must refer to an installation before curr. Use NULL or -Inf if you want to compare against the empty installation.

Value

An object of class conan_compare, which can be printed nicely.

---

conan_configure   *Configuration for conan*

Description

Configuration for running conan. Some common options and some specific to different provisioning methods.

Usage

```
conan_configure(
  method,
  ...,
  path_lib,
  path_bootstrap,
  cran = NULL,
  delete_first = FALSE,
```

```
    path = ".",
    envvars = NULL
)
```

Arguments

method	The method to use; currently `script`, `pkgdepends`, `auto` and `renv` are supported.
...	Additional arguments, method specific. See Details.
path_lib	The library to install into. Could be an absolute or a relative path.
path_bootstrap	The path to a bootstrap library to use. This needs to contain all the packages required for the method you are using. For `script` this is just `remotes`, but for `pkgdepends` it must contain the full recursive dependencies of `pkgdepends`.
cran	URL for use as the CRAN repo. If not given we will use the RStudio CRAN mirror. This option has no effect when using renv, as the URLs in your lock file determine the locations that packages are fetched from. The intended use of this option is for where a CRAN repo is misbehaving (e.g., returning 500 errors, or has an invalid/incomplete/out of date index). The most likely alternative version to use is `cran = "https://cran.r-project.org"`
delete_first	Should we delete the library before installing into it?
path	Path to the root where you would run conan from; typically this is the same path is the root of the project, often as the working directory.
envvars	Environment variables to set before running the installation. See Details for format.

Details

Different methods support different additional arguments:

- method `script` supports the argument `script`, which is the name of the script to run, defaults to "provision.R"

- method `pkgdepends` supports the arguments `refs`, which can be a character vector of references (rather than reading from the file `pkgdepends.txt`) and `policy` which is passed through to [pkgdepends::new_pkg_installation_proposal()](#).

- method `auto` takes an argument `environment` which contains a list of packages to install and source files to scan for dependencies.

- method `renv` takes no arguments.

Setting environment variables while running the installation comes uses the `envvars` argument; this system is designed to play well with hipercow, though it does not require it. We expect a `data.frame` with columns `name`, `value` and (optionally) `secret`. If `secret` is given, it must be a logical value indicating that `value` has been encrypted with an rsa public key. If any `secret` is `TRUE`, then `envvars` must also have an *attribute* key that contains the path to private rsa key to decrypt the secrets (i.e., `attr(envvars, "key")`). If you use secret environment variables, then the `openssl` package must be present in conan's bootstrap.

Value

A list with class `conan_config`. Do not modify this object.

---

conan_describe   *Describe a library*

Description

Describe a library. This creates a summary of version information from a library. This may be slightly slow on network filesystems with large libraries.

Usage

```
conan_describe(path_lib)
```

Arguments

path_lib   Path to the library

Value

A list with class `conan_describe`. We'll write some tooling to work with these soon!

---

conan_list   *Test if a conan installation is current*

Description

List conan installations, and optionally test if they are current.

Usage

```
conan_list(path_lib, hash = NULL)
```

Arguments

path_lib   Path to the library to compare

hash   A hash to compare; if given (not NULL) then we highlight installations that match
                  this hash.

Value

A [data.frame](#) with columns:

- name: the name of the installation. This might be useful with conan_compare

- time: the time the installation was started

- hash: the installation hash

- method: the method used for the installation

- args: the arguments to the installation (as a list column)

- current: Matches the hash passed in the argument hash

This object also has class conan_list so that it prints nicely, but you can drop this with as.data.frame.

---

conan_run   *Run a conan installation*

Description

Run a conan installation, in another process, blocking from this process.

Usage

```
conan_run(config, show_log = TRUE)
```

Arguments

config   Conan config, from [conan_configure()](#)

show_log   Show output from running the script (passed through to [callr::rscript](#) as show)

Value

Nothing

---

conan_write   *Write conan installation script*

Description

Write a conan installation script

Usage

```
conan_write(config, path)
```

Arguments

config	Conan config, from conan_configure()
path	The path to write to

Value

Nothing

Index

7