Package: rattlerthis (via r-universe)

November 13, 2026

Title The 'rattler' Backend for 'condathis'

Version 0.0.1

Description Solves, installs and activates 'Conda' environments in the R process through Rust bindings (using 'extendr') to the 'rattler' library (<https://github.com/conda/rattler>), the engine that powers 'pixi', without an external package manager binary. It is the ``rattler'' backend of 'condathis' and exports no functions: use the functions of 'condathis' with method = ``rattler''.

License MIT + file LICENSE

URL https://github.com/luciorq/rattlerthis

BugReports https://github.com/luciorq/rattlerthis/issues

Depends R (>= 4.3)

Imports cli, condathis (>= 0.2.0), fs, processx, rlang

Suggests callr, jsonlite, parallel, ps, testthat (>= 3.0.0), withr

Remotes luciorq/condathis

Config/condathis/backend rattler

Config/Needs/dev devtools, pkgload, remotes, rcmdcheck, covr, testthat, usethis, pak, knitr, quarto, pkgdown, roxygen2, roxygen2md, lintr, styler, rextendr

Config/rextendr/version 0.5.0

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

SystemRequirements GNU make, Cargo (Rust's package manager), rustc >= 1.88.0, OpenSSL on Linux (libssl-dev / openssl-devel)

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev libclang-dev

Repository https://luciorq.r-universe.dev

Date/Publication 2026-11-12 21:59:18 UTC

RemoteUrl https://github.com/luciorq/rattlerthis

RemoteRef HEAD

RemoteSha 4bee6582708305b75748d6666c7781ee29d3cd56

Contents

rattlerthis-package 2

Index **5**

`rattlerthis-package` *rattlerthis: The 'rattler' Backend for 'condathis'*

Description

rattlerthis is the backend `"rattler"` of condathis. It creates Conda environments, installs packages into them and activates them in the R process, through the Rust library 'rattler' (https://github.com/conda/rattler), without a package manager binary. It exports no functions and has no options of its own: use condathis's functions with `method = "rattler"`.

Details

```
condathis::create_env("ripgrep", env_name = "rg", method = "rattler")
condathis::run("rg", "--version", env_name = "rg")
condathis::list_envs(method = "rattler")
```

condathis loads rattlerthis when `method = "rattler"` is used, and when rattler environments exist, so `method = "auto"` finds them. Every argument, check, message, result and condition class is condathis's, as are `condathis::with_sandbox_dir()` and `condathis::clean_cache()`. The environments live in `condathis::backend_root_dir("rattler")`, the directory `rattler` inside condathis's data directory, with rattler's own package cache (`pkgs`) and channel metadata cache (`repodata_cache`). The generic Conda variables `CONDA_OVERRIDE_*` are honored, as by micromamba.

Differences from the micromamba backend

Both backends behave the same wherever possible: `condathis::run()`, `condathis::run_bin()` and `condathis::run_pipeline()` activate the environment the same way (also one given by `condathis::env_path()`), the activation is cached per environment until the environment or the session's variables change, and condathis lets only one R session at a time change an environment. What a user may notice:

- Link scripts: rattler never runs the `pre-link`, `post-link` or `pre-unlink` scripts of packages, and condathis warns with class `condathis_link_scripts_skipped`, naming the packages. micromamba runs them (and ignores a script that fails). `activate.d` scripts run with both backends. rattler does not byte-compile `noarch` Python packages at install time either.

- Output: rattler runs no command, so the level `"cmd"` prints a one-line description (`Running rattler create ...`), `"full"` prints one line per package of the transaction, and the spinner is cli's. In the results of `condathis::create_env()` and `condathis::install_packages()`, `cmd`, `stderr` and `pid` are `NA` (`stderr` is pip's after pip steps), and `stdout` holds the transaction lines (`""` when nothing changed).
- Errors have condathis's classes with both backends. Instead of micromamba's status classes, a rattler failure has the parent classes `rattlerthis_rattler_<kind>` (such as `rattlerthis_rattler_solve`) and `rattlerthis_rattler_error`. The error texts differ.
- pip: the `pip:` sections of a YAML file and the pip packages of lock files are installed as micromamba installs them, with the environment's pip after the Conda packages, and `condathis::list_packages()` lists them alike. pip's cache on Linux is in the `cache` directory of each backend's root; on macOS and Windows pip keeps it in the user's own cache directory with both.
- The environment of a command: micromamba's activation also sets `MAMBA_EXE` and puts a directory of its own (`condabin`) on `PATH`. On Linux and macOS it sets `LC_ALL` to `C.UTF-8`, where rattler keeps the session's value, and drops single quotes from the values of `conda-meta/state` and `etc/conda/env_vars.d`, which rattler passes as written. On Windows it also sets `PYTHONUTF8`, `PYTHONIOENCODING` and `_CONDA_OLD_CHCP`, and puts the environment name in `PROMPT`. `CONDA_ENVS_PATH`, `CONDA_PKGS_DIRS` and `MAMBA_PKGS_DIRS` name each backend's own directories.
- An interrupt (Ctrl-C) stops a solve or a download at once, but waits for a link that has started to finish.
- Files: the package caches are separate, so a package that both backends use is downloaded twice; after `condathis::clean_cache()`, rattler's empty `pkgs` (with its lock file) and `repodata_cache` directories stay. Every rattler environment has a `CACHEDIR.TAG` file (and, on macOS, is excluded from Time Machine backups), and its paths are 8 characters longer than micromamba's (the 260-character path limit of Windows). On Windows, a file of a running program that cannot be deleted when its environment is removed stays in condathis's trash and is deleted by a later removal; micromamba leaves it as `.mamba_trash`.
- rattler solves for the platform rattlerthis was built for, which is the platform of the micromamba binary condathis installs for that R (an x64 R on an ARM64 Windows PC solves for `win-64`); micromamba solves for the platform of the binary it runs, which differs when condathis uses a binary another R installed or one given with `CONDATHIS_MICROMAMBA_PATH`.
- In a forked child process on macOS (`parallel::mcparallel()`, `parallel::mclapply()`), creates, installs and downloads fail with an error; running commands works. On Linux forks are fully supported, and PSOCK workers work everywhere.
- Network: both backends honor the same settings, as micromamba has them when condathis runs it: the proxy variables `http_proxy`, `https_proxy`, `all_proxy` and `no_proxy` (the last three also in capitals), the system's CA certificates (or `ssl/cert.pem` in condathis's directory) but no CA variable such as `SSL_CERT_FILE`, and the credentials of `~/.mamba/auth/authentication.json`, `.netrc` and channel URLs. Channel metadata is reused as micromamba reuses it (for an hour, or as long as the server allows), but for a channel with a platform directory that the server does not have. Network errors are retried with other waits and time-outs, and on Windows rattler checks certificate revocation only against the system's cache, where micromamba asks the network.
- Channels: a local channel in a directory with a space works with rattler everywhere; micromamba reads it through another path to the same directory, which condathis cannot make on

a Windows volume without short (8.3) names, or on Linux and macOS when condathis's own directory has a space.

- Environment metadata: a # comment line in `conda-meta/pinned` is skipped by rattler, where micromamba fails every install. rattler never adds environments to `~/.conda/environments.txt`; micromamba does only when it needs the user's home (saved channel logins, `.netrc`, or an environment file that runs pip or names the home). rattler's `conda-meta/history` has no `# cmd:` and `# conda version:` lines, and can list the packages of one transaction in another order.

Author(s)

Maintainer: Lucio Queiroz <luciorqueiroz@gmail.com> [copyright holder]

Authors:

- Lucio Queiroz <luciorqueiroz@gmail.com> [copyright holder]

See Also

`condathis::create_env()`, `condathis::register_backend()`, `condathis::backend_root_dir()`

Index

condathis::backend_root_dir(), *4*
condathis::clean_cache(), *2*, *3*
condathis::create_env(), *3*, *4*
condathis::env_path(), *2*
condathis::install_packages(), *3*
condathis::list_packages(), *3*
condathis::register_backend(), *4*
condathis::run(), *2*
condathis::run_bin(), *2*
condathis::run_pipeline(), *2*
condathis::with_sandbox_dir(), *2*

rattlerthis *(*rattlerthis-package*)*, 2
rattlerthis-package, 2