

Package: rcloner (via r-universe)

May 26, 2026

Title Interface to 'rclone' Cloud Storage Utility

Version 0.0.1

Description Provides an R interface to 'rclone' <<https://rclone.org>>, a command-line program for managing files on cloud storage. 'rclone' supports over 40 cloud storage providers including 'S3'-compatible services ('Amazon S3', 'MinIO', 'Ceph'), 'Google Cloud Storage', 'Azure Blob Storage', and many others. This package downloads and manages the 'rclone' binary automatically and wraps its commands as R functions, returning results as data frames where appropriate.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports fs, jsonlite, processx, utils

Suggests covr, knitr, rmarkdown, spelling, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Language en-US

VignetteBuilder knitr

URL <https://boettiger-lab.github.io/rcloner/>,
<https://github.com/boettiger-lab/rcloner>

BugReports <https://github.com/boettiger-lab/rcloner/issues>

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://boettiger-lab.r-universe.dev>

Date/Publication 2026-03-22 20:53:41 UTC

RemoteUrl <https://github.com/boettiger-lab/rcloner>

RemoteRef HEAD

RemoteSha 2091105b186a6706bd41f1c0d707c127b67e28af

Contents

install_rclone	2
rclone	3
rclone_about	4
rclone_available	4
rclone_cat	5
rclone_check	6
rclone_config_create	6
rclone_config_delete	7
rclone_config_set	8
rclone_config_show	9
rclone_copy	9
rclone_copyurl	10
rclone_delete	11
rclone_link	11
rclone_listremotes	12
rclone_ls	12
rclone_lsd	13
rclone_mkdir	14
rclone_move	14
rclone_purge	15
rclone_rmdir	15
rclone_size	16
rclone_stat	17
rclone_sync	17
rclone_version	18
Index	19

install_rclone	<i>Install the rclone binary</i>
----------------	----------------------------------

Description

Downloads and installs the rclone binary for the current platform. The binary is stored in a user-writable directory and does not require system-level installation.

Usage

```
install_rclone(
  force = FALSE,
  dest = getOption("rcloner.dir", tools::R_user_dir("rcloner", "data"))
)
```

Arguments

force	Logical; if TRUE, re-download even if rclone is already installed. Default FALSE.
dest	Directory in which to install rclone. Defaults to the value of options("rcloner.dir"), falling back to tools::R_user_dir("rcloner", "data").

Value

Invisibly, the path to the installed rclone binary.

Examples

```
## Not run:
install_rclone()

## End(Not run)
```

rclone	<i>Run an rclone command</i>
--------	------------------------------

Description

Low-level wrapper that runs any rclone command. Higher-level functions like `rclone_copy()`, `rclone_ls()`, etc. call this internally.

Usage

```
rclone(args = character(), env = NULL, echo = FALSE, error_on_status = TRUE)
```

Arguments

args	Character vector of arguments passed to rclone (the command and its flags). Alternatively a single space-delimited string.
env	Named character vector of environment variables, or NULL.
echo	Logical; if TRUE, print stdout/stderr as it is produced. Default FALSE.
error_on_status	Logical; if TRUE (default), stop with an error when rclone exits with a non-zero status.

Value

A list with elements status, stdout, stderr, and timeout (invisibly).

Examples

```
rclone("version")  
## Not run:  
rclone(c("ls", "myremote:mybucket"))  
  
## End(Not run)
```

rclone_about	<i>Get quota information for a remote</i>
--------------	---

Description

Get quota information for a remote

Usage

```
rclone_about(path)
```

Arguments

path Remote name or path (e.g. "myremote:").

Value

A list with quota details (free, used, total bytes).

Examples

```
## Not run:  
rclone_about("myremote:")  
  
## End(Not run)
```

rclone_available	<i>Check whether rclone is available</i>
------------------	--

Description

Returns TRUE if the rclone binary can be found and executed. Useful as a condition for `\examplesIf`.

Usage

```
rclone_available()
```

Value

Logical scalar.

Examples

```
rclone_available()
```

rclone_cat	<i>Output file content to R</i>
------------	---------------------------------

Description

Reads the content of a remote file and returns it as a character string, analogous to cat on the command line.

Usage

```
rclone_cat(path)
```

Arguments

path	Remote path to the file.
------	--------------------------

Value

Character string with the file content.

Examples

```
tmp <- tempfile(fileext = ".txt")
writeLines("hello rclone", tmp)
txt <- rclone_cat(tmp)
cat(txt)
## Not run:
txt <- rclone_cat("myremote:bucket/readme.txt")
cat(txt)

## End(Not run)
```

rclone_check	<i>Check whether two remotes are in sync</i>
--------------	--

Description

Compares the files in src and dest without transferring anything. Differences are reported as messages.

Usage

```
rclone_check(src, dest, ...)
```

Arguments

src	Source path.
dest	Destination path.
...	Additional rclone flags.

Value

Invisibly, the rclone result list (check stderr for differences).

Examples

```
src <- tempfile()
dir.create(src)
dest <- tempfile()
dir.create(dest)
rclone_check(src, dest)
```

rclone_config_create	<i>Configure an rclone remote</i>
----------------------	-----------------------------------

Description

Creates or updates a named remote in the rclone config file. This is the primary way to register cloud storage credentials with rclone.

Usage

```
rclone_config_create(name, type, ...)
```

Arguments

name	Name for the remote (used as the prefix in paths, e.g. "myremote:bucket/key").
type	Remote type, e.g. "s3", "gcs", "azureblob", "drive". Run <code>rclone("help backends")</code> to see all supported types.
...	Named parameters passed to the remote configuration. For S3 remotes this typically includes <code>provider</code> , <code>access_key_id</code> , <code>secret_access_key</code> , and optionally <code>endpoint</code> (for non-AWS S3).

Value

Invisibly, the result of the `rclone` command.

Examples

```
## Not run:
# AWS S3
rclone_config_create("aws", type = "s3",
  provider = "AWS",
  access_key_id = Sys.getenv("AWS_ACCESS_KEY_ID"),
  secret_access_key = Sys.getenv("AWS_SECRET_ACCESS_KEY"),
  region = "us-east-1"
)

# MinIO / S3-compatible
rclone_config_create("minio", type = "s3",
  provider = "Minio",
  access_key_id = Sys.getenv("MINIO_ACCESS_KEY"),
  secret_access_key = Sys.getenv("MINIO_SECRET_KEY"),
  endpoint = "https://minio.example.com"
)

## End(Not run)
```

`rclone_config_delete` *Delete an rclone remote*

Description

Delete an rclone remote

Usage

```
rclone_config_delete(name)
```

Arguments

name	Remote name to delete.
------	------------------------

Value

Invisibly, the result of the rclone command.

Examples

```
## Not run:
rclone_config_delete("myremote")

## End(Not run)
```

rclone_config_set *Set individual parameters on an existing remote*

Description

Set individual parameters on an existing remote

Usage

```
rclone_config_set(name, ...)
```

Arguments

name	Remote name.
...	Named parameters to set (key = value).

Value

Invisibly, the result of the rclone command.

Examples

```
## Not run:
rclone_config_set("myremote", region = "us-west-2")

## End(Not run)
```

rclone_config_show *Show rclone configuration*

Description

Show rclone configuration

Usage

```
rclone_config_show(name = NULL)
```

Arguments

name Optional remote name. If omitted, shows all remotes.

Value

Character string with the configuration (invisibly). Also prints to the console.

Examples

```
rclone_config_show()
```

rclone_copy *Copy files between remotes or local paths*

Description

Copies files from src to dest, skipping files that are identical. Does not delete files at the destination.

Usage

```
rclone_copy(src, dest, progress = FALSE, ...)
```

Arguments

src Source path (local or remote).
 dest Destination path (local or remote).
 progress Logical; show transfer progress. Default FALSE.
 ... Additional rclone flags as character strings.

Value

Invisibly, the rclone result list.

Examples

```
src <- tempfile()
dir.create(src)
writeLines("hello", file.path(src, "file.txt"))
dest <- tempfile()
rclone_copy(src, dest)
list.files(dest)
## Not run:
rclone_copy("myremote:bucket/file.csv", "/tmp/file.csv")

## End(Not run)
```

rclone_copyurl

Copy a URL directly to a remote

Description

Downloads a URL and uploads it to dest without storing it locally first.

Usage

```
rclone_copyurl(url, dest, ...)
```

Arguments

url	URL to copy.
dest	Destination remote path.
...	Additional rclone flags.

Value

Invisibly, the rclone result list.

Examples

```
## Not run:
rclone_copyurl("https://example.com/data.csv", "myremote:bucket/data.csv")

## End(Not run)
```

rclone_delete	<i>Delete files at a remote path</i>
---------------	--------------------------------------

Description

Deletes the files (not directories) at the given path. To remove a directory and all its contents use [rclone_purge\(\)](#).

Usage

```
rclone_delete(path, ...)
```

Arguments

path	Remote path to delete.
...	Additional rclone flags.

Value

Invisibly, the rclone result list.

Examples

```
d <- tempfile()
dir.create(d)
writeLines("hi", file.path(d, "f.txt"))
rclone_delete(d)
```

rclone_link	<i>Generate a public link for a remote file</i>
-------------	---

Description

Not all remotes support this operation.

Usage

```
rclone_link(path)
```

Arguments

path	Remote path to the file.
------	--------------------------

Value

Character string with the public URL.

Examples

```
## Not run:
rclone_link("myremote:bucket/file.csv")

## End(Not run)
```

rclone_listremotes	<i>List configured remotes</i>
--------------------	--------------------------------

Description

List configured remotes

Usage

```
rclone_listremotes()
```

Value

Character vector of remote names.

Examples

```
rclone_listremotes()
```

rclone_ls	<i>List objects in a remote path</i>
-----------	--------------------------------------

Description

Returns a data frame of objects at the given path, parsed from rclone's JSON output. This is the primary listing function and maps to `rclone lsjson`.

Usage

```
rclone_ls(path, recursive = FALSE, dirs_only = FALSE, files_only = FALSE, ...)
```

Arguments

path	Remote path in rclone syntax, e.g. "myremote:bucket/prefix". For local paths use the filesystem path directly.
recursive	Logical; if TRUE, list all objects recursively. Default FALSE.
dirs_only	Logical; if TRUE, list only directories. Default FALSE.
files_only	Logical; if TRUE, list only files (no directories). Default FALSE.
...	Additional flags passed to <code>rclone lsjson</code> as strings (e.g. "--fast-list").

Value

A data frame with columns Path, Name, Size, MimeType, ModTime, IsDir, and optionally others returned by rclone.

Examples

```
rclone_ls(tempdir())
rclone_ls(tempdir(), recursive = TRUE)
## Not run:
rclone_ls("myremote:mybucket")

## End(Not run)
```

rclone_ls	<i>List directories in a remote path</i>
-----------	--

Description

Convenience wrapper around [rclone_ls\(\)](#) that returns only directories.

Usage

```
rclone_ls(path, recursive = FALSE, ...)
```

Arguments

path	Remote path in rclone syntax, e.g. "myremote:bucket/prefix". For local paths use the filesystem path directly.
recursive	Logical; if TRUE, list all objects recursively. Default FALSE.
...	Additional flags passed to rclone lsjson as strings (e.g. "--fast-list").

Value

A data frame (see [rclone_ls\(\)](#)).

Examples

```
rclone_ls(tempdir())
```

rclone_mkdir	<i>Create a directory or bucket</i>
--------------	-------------------------------------

Description

Create a directory or bucket

Usage

```
rclone_mkdir(path)
```

Arguments

path	Remote path to create.
------	------------------------

Value

Invisibly, the rclone result list.

Examples

```
d <- tempfile()
rclone_mkdir(d)
```

rclone_move	<i>Move files between remotes or local paths</i>
-------------	--

Description

Moves files from src to dest. Source files are deleted after a successful transfer.

Usage

```
rclone_move(src, dest, progress = FALSE, ...)
```

Arguments

src	Source path (local or remote).
dest	Destination path (local or remote).
progress	Logical; show transfer progress. Default FALSE.
...	Additional rclone flags as character strings.

Value

Invisibly, the rclone result list.

Examples

```
src <- tempfile()
dir.create(src)
writeLines("hello", file.path(src, "file.txt"))
dest <- tempfile()
rclone_move(src, dest)
```

rclone_purge	<i>Remove a path and all its contents</i>
--------------	---

Description

Deletes all files and directories under path. Use with caution.

Usage

```
rclone_purge(path)
```

Arguments

path Remote path to purge.

Value

Invisibly, the rclone result list.

Examples

```
d <- tempfile()
dir.create(d)
writeLines("hi", file.path(d, "f.txt"))
rclone_purge(d)
```

rclone_rmdir	<i>Remove an empty directory</i>
--------------	----------------------------------

Description

Remove an empty directory

Usage

```
rclone_rmdir(path)
```

Arguments

path Remote path to the directory.

Value

Invisibly, the rclone result list.

Examples

```
d <- tempfile()
dir.create(d)
rclone_rmdir(d)
```

rclone_size	<i>Get total size of a remote path</i>
-------------	--

Description

Get total size of a remote path

Usage

```
rclone_size(path)
```

Arguments

path Remote path.

Value

A named list with count (number of objects) and bytes (total size in bytes).

Examples

```
rclone_size(tempdir())
```

rclone_stat	<i>Get metadata for a single object</i>
-------------	---

Description

Get metadata for a single object

Usage

```
rclone_stat(path)
```

Arguments

path Remote path to a single object.

Value

A one-row data frame with object metadata, or an error if not found.

Examples

```
f <- tempfile()
writeLines("hello", f)
rclone_stat(f)
```

rclone_sync	<i>Sync a source to a destination</i>
-------------	---------------------------------------

Description

Makes dest identical to src, **deleting** files in dest that are not in src. Use [rclone_copy\(\)](#) if you do not want deletions.

Usage

```
rclone_sync(src, dest, progress = FALSE, ...)
```

Arguments

src Source path (local or remote).
dest Destination path (local or remote).
progress Logical; show transfer progress. Default FALSE.
... Additional rclone flags as character strings.

Value

Invisibly, the rclone result list.

Examples

```
src <- tempfile()
dir.create(src)
writeLines("hello", file.path(src, "file.txt"))
dest <- tempfile()
rclone_sync(src, dest)
## Not run:
rclone_sync("myremote:bucket", "/local/backup")

## End(Not run)
```

rclone_version	<i>Show rclone version information</i>
----------------	--

Description

Show rclone version information

Usage

```
rclone_version()
```

Value

Invisibly, a character string with the version output.

Examples

```
rclone_version()
```

Index

install_rclone, 2

rclone, 3

rclone_about, 4

rclone_available, 4

rclone_cat, 5

rclone_check, 6

rclone_config_create, 6

rclone_config_delete, 7

rclone_config_set, 8

rclone_config_show, 9

rclone_copy, 9

rclone_copy(), 3, 17

rclone_copyurl, 10

rclone_delete, 11

rclone_link, 11

rclone_listremotes, 12

rclone_ls, 12

rclone_ls(), 3, 13

rclone_lsd, 13

rclone_mkdir, 14

rclone_move, 14

rclone_purge, 15

rclone_purge(), 11

rclone_rmdir, 15

rclone_size, 16

rclone_stat, 17

rclone_sync, 17

rclone_version, 18