

# Package ‘rcmdcheck’

May 7, 2019

**Title** Run 'R CMD check' from 'R' and Capture Results

**Version** 1.3.3

**Author** Gábor Csárdi

**Maintainer** Gábor Csárdi <csardi.gabor@gmail.com>

**Description** Run 'R CMD check' from 'R' and capture the results of the individual checks. Supports running checks in the background, timeouts, pretty printing and comparing check results.

**License** MIT + file LICENSE

**LazyData** true

**URL** <https://github.com/r-Lib/rcmdcheck#readme>

**BugReports** <https://github.com/r-Lib/rcmdcheck/issues>

**RoxygenNote** 6.1.1

**Suggests** covr, knitr, mockery, rmarkdown, testthat

**Imports** callr (>= 3.1.1.9000), cli (>= 1.1.0), crayon, desc (>= 1.2.0), digest, pkgbuild, prettyunits, R6, rprojroot, sessioninfo (>= 1.1.1), utils, withr, xopen

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-05-07 12:00:02 UTC

## R topics documented:

check_details . . . . .	2
compare_checks . . . . .	3
compare_to_cran . . . . .	3
cran_check_flavours . . . . .	4
cran_check_results . . . . .	5
parse_check . . . . .	5
parse_check_url . . . . .	6

print.rcmdcheck . . . . .	6
print.rcmdcheck_comparison . . . . .	7
rcmdcheck . . . . .	7
rcmdcheck_process . . . . .	8
xopen.rcmdcheck . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

check_details	<i>Query R CMD check results and parameters</i>
---------------	---

---

## Description

Query R CMD check results and parameters

## Usage

```
check_details(check)
```

## Arguments

check            A check result.

## Value

A named list with elements:

- package: package name.
- version: package version.
- rversion: R version.
- notes: character vector of check NOTES, each NOTE is an element.
- warnings: character vector of check WARNINGS, each WARNING is an element.
- errors: character vector of check ERRORS, each ERROR is an element. A check timeout adds an ERROR to this vector.
- platform: check platform
- checkdir: check directory.
- install\_out: the output of the installation, contents of the `00install.out` file. A single string.
- description: the contents of the DESCRIPTION file of the package. A single string.
- session\_info: the output of `sessioninfo::session_info()`, from the R session performing the checks.
- checkdir: the path to the check directory, if it hasn't been cleaned up yet, or NA. The check directory is automatically cleaned up, when the check object is deleted (garbage collected).
- cran: whether it is a CRAN packaged package.
- bioc: whether it is a BioConductor package.

---

compare_checks	<i>Compare a set of check results to another check result</i>
----------------	---

---

**Description**

Compare a set of check results to another check result

**Usage**

```
compare_checks(old, new)
```

**Arguments**

old	A check result, or a list of check results.
new	A check result.

**Value**

An rcmdcheck\_comparison object with fields:

- package: the name of the package, string,
- versions: package versions, length two character,
- status: comparison status, see below,
- old: list of rcmdcheck objects the old check(s),
- new: rcmdcheck object, the new check,
- cmp:

**See Also**

Other check comparisons: [compare\\_to\\_cran](#)

---

compare_to_cran	<i>Compare a check result to CRAN check results</i>
-----------------	---

---

**Description**

Compare a check result to CRAN check results

**Usage**

```
compare_to_cran(check, flavours = cran_check_flavours(check$package))
```

**Arguments**

check            A check result.  
flavours        CRAN check flavour(s) to use. By default all platforms are used.

**Value**

An rmdcheck\_comparison object.

**See Also**

Other check comparisons: [compare\\_checks](#)

---

cran\_check\_flavours    *Download and show all CRAN check flavour platforms*

---

**Description**

If the package argument is NULL, then all current platforms are downloaded. If the package argument is specified, then all flavours used for the latests package checks for that package, are downloaded and returned.

**Usage**

```
cran_check_flavours(package = NULL)
```

**Arguments**

package        CRAN package name or NULL.

**Value**

Character vector of platform ids.

**Examples**

```
## Not run:  
cran_check_flavours()  
cran_check_flavours("simplegraph")  
  
## End(Not run)
```

---

cran_check_results	<i>Download and parse R CMD check results from CRAN</i>
--------------------	---

---

**Description**

Download and parse R CMD check results from CRAN

**Usage**

```
cran_check_results(package, flavours = cran_check_flavours(package),  
  quiet = TRUE)
```

**Arguments**

package	Name of a single package to download the checks for.
flavours	CRAN check flavours to use. Defaults to all flavours that were used to check the package.
quiet	Whether to omit the download progress bars.

**Value**

A list of rcmdcheck objects.

---

parse_check	<i>Parse R CMD check results from a file or string</i>
-------------	--

---

**Description**

At most one of `file` or `text` can be given. If both are `NULL`, then the current working directory is checked for a `00check.log` file.

**Usage**

```
parse_check(file = NULL, text = NULL, ...)
```

**Arguments**

file	The <code>00check.log</code> file, or a directory that contains that file. It can also be a connection object.
text	The contentst of a <code>00check.log</code> file.
...	Other arguments passed onto the constructor. Used for testing.

**Value**

An rcmdcheck object, the check results.

**See Also**[parse\\_check\\_url](#)

---

parse_check_url	<i>Shorthand to parse R CMD check results from a URL</i>
-----------------	--

---

**Description**

Shorthand to parse R CMD check results from a URL

**Usage**

```
parse_check_url(url, quiet = TRUE)
```

**Arguments**

url	URL to parse the results from. Note that it should not contain HTML markup, just the text output.
quiet	Passed to <code>download.file</code> .

**Value**

An `rcmdcheck` object, the check results.

**See Also**[parse\\_check](#)

---

print.rcmdcheck	<i>Print R CMD check results</i>
-----------------	----------------------------------

---

**Description**

Print R CMD check results

**Usage**

```
## S3 method for class 'rcmdcheck'  
print(x, header = TRUE, ...)
```

**Arguments**

x	Check result object to print.
header	Whether to print a header.
...	Additional arguments, currently ignored.

---

```
print.rcmdcheck_comparison
      Print R CMD check result comparisons
```

---

**Description**

See [compare\\_checks\(\)](#) and [compare\\_to\\_cran\(\)](#).

**Usage**

```
## S3 method for class 'rcmdcheck_comparison'
print(x, header = TRUE, ...)
```

**Arguments**

x	R CMD check result comparison object.
header	Whether to print the header. You can suppress the header if you want to use the printout as part of another object's printout.
...	Additional arguments, currently ignored.

---

```
rcmdcheck      Run R CMD check from R and Capture Results
```

---

**Description**

Run R CMD check from R programmatically, and capture the results of the individual checks.

**Usage**

```
rcmdcheck(path = ".", quiet = FALSE, args = character(),
  build_args = character(), check_dir = NULL, libpath = .libPaths(),
  repos = getOption("repos"), timeout = Inf, error_on = c("never",
  "error", "warning", "note"))
```

**Arguments**

path	Path to a package tarball or a directory.
quiet	Whether to print check output during checking.
args	Character vector of arguments to pass to R CMD check. (Note that instead of the <code>--output</code> option you should use the <code>check_dir</code> argument, because <code>--output</code> cannot deal with spaces and other special characters on Windows.)
build_args	Character vector of arguments to pass to R CMD build

check_dir	Path to a directory where the check is performed. If this is not NULL, then the a temporary directory is used, that is cleaned up when the returned object is garbage collected.
libpath	The library path to set for the check. The default uses the current library path.
repos	The repos option to set for the check. This is needed for cyclic dependency checks if you use the <code>--as-cran</code> argument. The default uses the current value.
timeout	Timeout for the check, in seconds, or as a <code>base::difftime</code> object. If it is not finished before this, it will be killed. <code>Inf</code> means no timeout. If the check is timed out, that is added as an extra error to the result object.
error_on	Whether to throw an error on R CMD check failures. Note that the check is always completed (unless a timeout happens), and the error is only thrown after completion. If "never", then no errors are thrown. If "error", then only ERROR failures generate errors. If "warning", then WARNING failures generate errors as well. If "note", then any check failure generated an error.

### Value

An S3 object (list) with fields `errors`, `warnings` and `notes`. These are all character vectors containing the output for the failed check.

---

rcmdcheck_process	<i>Run an R CMD check process in the background</i>
-------------------	---

---

### Description

`rcmdcheck_process` is an R6 class, that extends the `callr::rcmd_process` class (which in turn extends `processx::process`).

### Usage

```
cp <- rcmdcheck_process$new(path = ".", args = character(),
  build_args = character(), check_dir = NULL,
  libpath = .libPaths(), repos = getOption("repos"))
```

```
cp$parse_results()
```

Other methods are inherited from `callr::rcmd_process` and `processx::process`.

Note that you calling the `get_output_connection` and `get_error_connection` method on this is not a good idea, because then the `stdout` and/or `stderr` of the process will not be collected for `parse_results()`.

You can still use the `read_output_lines()` and `read_error_lines()` methods to read the standard output and error, `parse_results()` is not affected by that.



**Arguments**

- `cp`: A new `rcmdcheck_process` object.
- `path`: Path to a package tree or a package archive file. This is the package to check.
- `args`: Command line arguments to R CMD check.
- `build_args`: Command line arguments to R CMD build.
- `check_dir`: Directory for the results.
- `libpath`: The library path to set for the check.
- `repos`: The `repos` option to set for the check. This is needed for cyclic dependency checks if you use the `--as-cran` argument. The default uses the current value.

**Details**

Most methods are inherited from `call::rcmd_process` and `processx::process`.

`cp$parse_results()` parses the results, and returns an S3 object with fields `errors`, `warnings` and `notes`, just like `rcmdcheck()`. It is an error to call it before the process has finished. Use the `wait()` method to wait for the check to finish, or the `is_alive()` method to check if it is still running.

---

<code>xopen.rcmdcheck</code>	<i>Open the check directory in a file browser window</i>
------------------------------	--

---

**Description**

Open the check directory in a file browser window

**Usage**

```
## S3 method for class 'rcmdcheck'
xopen(target, app = NULL, quiet = FALSE, ...)
```

**Arguments**

<code>target</code>	<code>rcmdcheck()</code> result.
<code>app</code>	Specify the app to open <code>target</code> with, and its arguments, in a character vector. Note that app names are platform dependent.
<code>quiet</code>	Whether to echo the command to the screen, before running it.
<code>...</code>	Additional arguments, not used currently.

# Index

`base::difftime`, 8

`callr::rcmd_process`, 8, 9

`check_details`, 2

`compare_checks`, 3, 4

`compare_checks()`, 7

`compare_to_cran`, 3, 3

`compare_to_cran()`, 7

`cran_check_flavours`, 4

`cran_check_results`, 5

`parse_check`, 5, 6

`parse_check_url`, 6, 6

`print.rcmdcheck`, 6

`print.rcmdcheck_comparison`, 7

`processx::process`, 8, 9

`rcmdcheck`, 7

`rcmdcheck()`, 9

`rcmdcheck-package (rcmdcheck)`, 7

`rcmdcheck_process`, 8

`sessioninfo::session_info()`, 2

`xopen.rcmdcheck`, 9