Package 'foghorn'

July 16, 2026

Title Summarize CRAN Check Results in the Terminal

Version 1.6.2

Description The CRAN check results and where your package stands in the CRAN submission queue in your R terminal.

License MIT + file LICENSE

URL https://fmichonneau.github.io/foghorn/, https://github.com/fmichonneau/foghorn

BugReports https://github.com/fmichonneau/foghorn/issues

Depends R (>= 3.1.0)

Imports cli (>= 3.6.1), curl (>= 2.2), httr2 (>= 1.0.0), rlang (>= 0.4.3), rvest (>= 0.3.2), tibble (>= 1.2), xml2 (>= 1.0.0)

Suggests covr, dplyr, knitr, progress, rmarkdown, testthat, withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Francois Michonneau [aut, cre], Ben Bolker [ctb]

Maintainer Francois Michonneau <francois.michonneau@gmail.com>

Repository CRAN

Date/Publication 2026-07-16 14:10:16 UTC

Contents

check_cran_results 2
cran_details 2
cran_incoming 4

cran_results 6
n_cran_flavors 8
summary_cran_results 9
visit_cran_check 10
winbuilder_queue 10

Index **12**

`check_cran_results` *Deprecated functions*

Description

Deprecated functions provided for back compatibility.

Usage

```
check_cran_results(...)
```

Arguments

`...`	see documentation for `cran_results` and `summary_cran_details`

`cran_details` *Get details about the CRAN check results for packages*

Description

Given the names of packages published on CRAN, return the output of checks that return notes, warnings or errors.

Usage

```
cran_details(pkg, src = c("website", "crandb"), ...)

## S3 method for class 'cran_details'
summary(object, show_log = TRUE, print_ok = TRUE, ...)

summary_cran_details(
  pkg,
  src = c("website", "crandb"),
  show_log = TRUE,
  print_ok = TRUE,
  ...
)
```

Arguments

pkg	character vector of the names for the packages on CRAN
src	if "website" the data is scrapped from the CRAN website, if "crandb" the data is downloaded from a RDS file hosted on the CRAN servers (which is used to generate the information found on the CRAN website).
...	additional arguments to control where the data from the check results are coming from and how they are downloaded from the CRAN servers (see Details section).
object	an object created by cran_details
show_log	Should the messages of the "Check Details" be printed? (logical)
print_ok	if TRUE the summary method will print a "all clear" message for package(s) that have an OK status for all CRAN checks.

Details

Where does the data come from?

The data comes from the CRAN servers. They generate RDS files that contains information regarding the results of the checks for all the packages, and all the flavors. This data is then used to generate the web pages.

foghorn provides access to either of these data sources. If you choose src = "website" the data is scrapped from the CRAN website. If you only need to check a few packages, this is a good option. If you choose src = "crandb" the RDS files (about 20Mb) are downloaded first from the CRAN servers.

The option max_requests can be used to limit how many pages will be scrapped from the CRAN website. The default is set to 50, use Inf to ignore this check. Consider using src = "crandb" if you need to get data from many packages or maintainers. Note that this an approximation of the number of requests that will be performed and there will be situations where more requests than this limit will be performed to retrieve the results.

The deadline column contains the date set by CRAN to fix issues with the CRAN checks before the package gets archived. If the existence of a deadline has been checked but no date has been set by the CRAN Maintainers, the deadline column for the package will be set to "". If there is a deadline, the content will be a date stored as character.

The value of the argument show can be set using the local option foghorn_columns.

When choosing src = "crandb" you can also specify the following options:

- dest: a folder where to store the RDS files (tempdir() by default).
- protocol: either https or http.
- overwrite: when FALSE (default), if the file exists in dest then it will not be downloaded again. When TRUE the file gets downloaded every time it's needed.

Value

a tibble listing the names of the packages that have non- OK check results, the nature of the result (WARN, ERROR, FAIL, NOTE, or other issues).

See Also

Note that the `tools` package contains unexported functions that can be used to extract summary information from the check results. Specifically `tools:::sumarize_CRAN_check_status` is similar to `show_cran_results`.

`cran_incoming` *List packages in CRAN incoming queue.*

Description

Check where your package stands in the CRAN incoming queue.

Usage

```
cran_incoming(
  pkg = NULL,
  folders = cran_incoming_folders(),
  sort_by_date = TRUE
)

cran_incoming_folders(include_archive = FALSE)
```

Arguments

`pkg`	Optionally provide a vector of package names to limit the results to these packages.
`folders`	Which folders of the CRAN queue do you want to inspect? Default: all the non-human folders.
`sort_by_date`	when `TRUE` (default), the output is sorted in decreasing order according to the submission time.
`include_archive`	when `TRUE`, the function `cran_incoming_folders()` also returns the `archive` folder.

Details

When submitting a package to CRAN, it undergoes a series of checks before it is published and publicly available. `cran_incoming()` allows you to check the packages that are currently in the queue, and the folder where they are located. This information could help you track your package submission. Only the following folders are considered (approximately in order of the CRAN queue sequence): `newbies`, `inspect`, `pretest`, `recheck`, `pending`, `waiting`, `publish`, `special`. The folder `archive` is not inspected by default. The folders named after the initials of the CRAN volunteers are not inspected.

Value

`cran_incoming()` returns `tibble` with the following columns:

package package name

version package version

cran_folder folder where the package was found

time date/time package was entered in the folder

size the size of the package tarball

`cran_incoming_folders()` returns a character vector of the names of the folders used as part of the CRAN submission process, `archive` being included optionally.

Note that if the package version is not provided, it will appear as `NA` in the `tibble`.

Disclaimer

The information provided here is only to give you an indication of where your package stands in the submission process. It can be useful to confirm that your package has been correctly uploaded to CRAN. Please consult the CRAN Repository Policy if you have any questions.

Note

The meaning of the package folders is as follows (see Hornik, Ligges and Zeileis https://journal.r-project.org/news/RJ-2018-1-cran/RJ-2018-1-cran.pdf and Uwe Ligges mailing list comment https://stat.ethz.ch/pipermail/r-package-devel/2019q1/003631.html):

newbies for first time submission; package will be manually inspected.

inspect package is awaiting manual inspection; always happens for first time submissions and for packages with problems that are likely to be false positives

pretest a human has triggered a new auto-check of the package

recheck package has passed checks and is waiting for reverse dependency checking

pending a CRAN team member has to do a closer inspection and needs more time

waiting CRAN's decision is waiting for a response from the package maintainer, e.g. when issues are present that CRAN cannot check for in the incoming checks

publish package is awaiting publication

special package is undergoing an additional, non-standard check (e.g. `noLD`, `valgrind`, `gcc-san`); unlike the other folders, `special` itself contains one sub-folder per check, so packages found here are reported with a `cran_folder` of the form `special/<check>`, e.g. `special/noLD`

archive package rejected: it does not pass the checks cleanly and the problems are unlikely to be false positives

References

- Hornik, Ligges and Zeileis. "Changes on CRAN: 2017-12-01 to 2018-06-30", R Journal 10(1), July 2018. https://journal.r-project.org/news/RJ-2018-1-cran/RJ-2018-1-cran.pdf
- Maëlle Salmon, Locke Data, Stephanie Locke, Mitchell O'Hara-Wild, Hugo Gruson. "CRAN incoming dashboard", https://lockedata.github.io/cransays/articles/dashboard.html

See Also

cran_winbuilder

Examples

```
## Not run:
  ## all the packages in the CRAN incoming queue
  cran_incoming()
  ## to include all the folders including `archive`
  cran_incoming(folders = cran_incoming_folders(include_archive = TRUE)
  ## to only include a folder, e.g., `inspect`
  cran_incoming(folders = "inspect")
  ## if the package `foo` is in the queue, it will appear below
  cran_incoming(pkg = "foo")

## End(Not run)
```

---

cran_results *Table of the CRAN check results*

Description

Make a table that summarizes the results of the CRAN checks for a set of packages specified by a maintainer or by names.

Usage

```
cran_results(
  email = NULL,
  pkg = NULL,
  show = getOption("foghorn_columns", c("error", "fail", "warn", "note", "ok",
    "deadline")),
  src = c("website", "crandb"),
  max_requests = 50,
  ...
)
```

Arguments

email	email address for package maintainers (character vector)
pkg	package names (character vector)
show	columns of the data frame to show (all are shown by default). See 'Details' for more information.
src	if "website" the data is scrapped from the CRAN website, if "crandb" the data is downloaded from a RDS file hosted on the CRAN servers (which is used to generate the information found on the CRAN website).

max_requests	maximum number of requests allowed to be performed, ignored when using src = "crandb". Use Inf to skip this check. (See Details).
...	additional arguments to control where the data from the check results are coming from and how they are downloaded from the CRAN servers (see Details section).

Details

Given the email address of a package maintainer, and/or a vector of package names, returns a tibble that allows you to detect potential issues with your packages on CRAN.

Where does the data come from?

The data comes from the CRAN servers. They generate RDS files that contains information regarding the results of the checks for all the packages, and all the flavors. This data is then used to generate the web pages.

foghorn provides access to either of these data sources. If you choose src = "website" the data is scrapped from the CRAN website. If you only need to check a few packages, this is a good option. If you choose src = "crandb" the RDS files (about 20Mb) are downloaded first from the CRAN servers.

The option max_requests can be used to limit how many pages will be scrapped from the CRAN website. The default is set to 50, use Inf to ignore this check. Consider using src = "crandb" if you need to get data from many packages or maintainers. Note that this an approximation of the number of requests that will be performed and there will be situations where more requests than this limit will be performed to retrieve the results.

The deadline column contains the date set by CRAN to fix issues with the CRAN checks before the package gets archived. If the existence of a deadline has been checked but no date has been set by the CRAN Maintainers, the deadline column for the package will be set to "". If there is a deadline, the content will be a date stored as character.

The value of the argument show can be set using the local option foghorn_columns.

When choosing src = "crandb" you can also specify the following options:

- dest: a folder where to store the RDS files (tempdir() by default).
- protocol: either https or http.
- overwrite: when FALSE (default), if the file exists in dest then it will not be downloaded again. When TRUE the file gets downloaded every time it's needed.

Value

a data frame that tabulates the number of CRAN flavors that return errors, warnings, notes, or OK for the packages. See 'Details' section for more information about the Deadline column.

See Also

Note that the tools package contains unexported functions that can be used to extract summary information from the check results. Specifically tools:::sumarize_CRAN_check_status is similar to show_cran_results.

Examples

```
## Not run:
  cran_results(pkg="MASS")

## End(Not run)
```

`n_cran_flavors` *The number of CRAN flavors*

Description

The CRAN flavors, the systems on which CRAN tests all packages regularly, are listed https://cran.r-project.org/web/checks/check_flavors.html. To get the correct results, foghorn needs to know how many flavors CRAN uses. This function reads the number of flavors that CRAN currently uses, and caches it (per session, in the `tempdir()` folder). Arguments control caching, fall back, and default values.

Usage

```
n_cran_flavors(
  use_cache = getOption("foghorn.use_cache", TRUE),
  force_default = getOption("foghorn.force_default", FALSE),
  n_flavors = getOption("foghorn.n_flavors", 12L)
)
```

Arguments

`use_cache`	Should the value for the number of flavors be read to/ written from the cache? (default: `TRUE`)
`force_default`	Should the default value be used? (default: `FALSE`). When `TRUE`, the number of flavors is read from the Internet.
`n_flavors`	What is the default number of flavors? (default: `12L`)

Details

The default values for the arguments are read from options. Given that `n_cran_flavors` function is relied on internally to provide accurate information to the user, using options allows you to control how the function behaves directly. In general, the default values should not be changed. They are provided in case you have issues connecting to the web page listing the number of flavors, or you do not want to use caching.

The options can be set:

- by session, using, for instance,`options("foghorn.use_cache" = FALSE)`.
- permanently, by adding `options("foghorn.use_cache" = FALSE)` in your `.Rprofile`.
- for a specific call, using the `withr` package: `withr::with_options(foghorn.use_cache = FALSE, ...)`.

Value

The number of CRAN check flavors (as an integer).

`summary_cran_results` *Summary of the CRAN check results*

Description

Given the email address of a package maintainer, and/or a vector of package names, it displays at the console a summary of the check results run on the CRAN flavors. This function is designed to be included in your .Rprofile to be run (periodically) at start up.

Usage

```
summary_cran_results(
  email = NULL,
  pkg = NULL,
  compact = FALSE,
  print_ok = TRUE,
  ...
)

## S3 method for class 'cran_results'
summary(object, compact = FALSE, print_ok = TRUE, ...)

show_cran_results(...)
```

Arguments

`email`	email address for package maintainers (character vector)
`pkg`	package names (character vector)
`compact`	if `TRUE`, all the packages with non-OK results are listed in a single line, otherwise they are listed on multiple lines.
`print_ok`	if `TRUE` the summary method will print a "all clear" message for package(s) that have an OK status for all CRAN checks.
`...`	additional arguments to control where the data from the check results are coming from and how they are downloaded from the CRAN servers (see Details section).
`object`	an object created by `cran_results`

Value

Prints the packages that return errors, warnings, and notes on the CRAN flavors. The number in parenthesis after the name of the packages indicates the number of CRAN flavors that produce these results.

Examples

```
## Not run:
   summary_cran_results(email = c("user1@company1.com", "user2@company2.com"))
   summary_cran_results(email = "user1@company1.com",
                        pkg = c("pkg1", "pkg2"))

## End(Not run)
```

`visit_cran_check` *Visit the CRAN check results page*

Description

Visit the page in your web browser for a given package or a maintainer's email address

Usage

```
visit_cran_check(pkg = NULL, email = NULL)
```

Arguments

`pkg`	name of the package to check the results for
`email`	email address of the package maintainer

Value

The URL from the CRAN check results page invisibly

`winbuilder_queue` *Show the win-builder queue*

Description

Check whether your package is in the win-builder queue.

Usage

```
winbuilder_queue(
  pkg = NULL,
  folders = c("R-release", "R-devel", "R-oldrelease")
)
```

Arguments

pkg	Optionally provide a vector of package names to limits the results to these packages.
folders	Which folders of the win-builder queue do you want to inspect? Default: the 3 architectures win-builder provides.

Details

To check whether your package has successfully been submitted to win-builder, or to check whether there is unusual delay in processing packages submitted to win-builder, `winbuilder_queue` allows you to inspect the packages that are in the queue to be processed by the win-builder service.

Value

A `tibble` with the following columns:

package package name

version package version

folder the folder indicating the R version that will be used to perform the checks

time the date and time at which the package tarball was uploaded on win-builder

size the size of the package tarball

References

- Maëlle Salmon, 2020. "Everything you should know about WinBuilder" https://blog.r-hub.io/2020/04/01/win-builder/
- Uwe Ligges. Building and checking R source packages for Windows. https://win-builder.r-project.org/

Examples

```
## Not run:
  ## Get all the packages in the win-builder queue
  winbuilder_queue()
  ## Check if the 'dplyr' package is in the win-builder queue
  winbuilder_queue(pkg = "dplyr")
  ## Check which packages are in the R-devel queue
  winbuilder_queue(folders = "R-devel")

## End(Not run)
```

Index

check_cran_results, 2
cran_details, 2
cran_incoming, 4
cran_incoming_folders *(*cran_incoming*)*, 4
cran_results, 6

n_cran_flavors, 8

show_cran_results
        *(*summary_cran_results*)*, 9
summary.cran_details *(*cran_details*)*, 2
summary.cran_results
        *(*summary_cran_results*)*, 9
summary_cran_details *(*cran_details*)*, 2
summary_cran_results, 9

visit_cran_check, 10

winbuilder_queue, 10