

# Package ‘cronR’

March 3, 2017

**Type** Package

**Title** Schedule R Scripts and Processes with the 'cron' Job Scheduler

**Version** 0.3.0

**Date** 2017-03-03

**Author** c(person("Jan", "Wijffels", role = c("aut", "cre", "cph"), email = "jwijffels@bnosac.be"), person("BNOSAC", role = "cph"), person("Kevin", "Ushey", role = c("aut", "cre", "cph"), email = "kevinushey@gmail.com"))

**Maintainer** Jan Wijffels <jwijffels@bnosac.be>

**Description** Create, edit, and remove 'cron' jobs on your unix-alike system. The package provides a set of easy-to-use wrappers to 'crontab'. It also provides an RStudio add-in to easily launch and schedule your scripts.

**URL** <https://github.com/bnosac/cronR>

**Imports** digest

**Suggests** knitr, miniUI, shiny (>= 0.11), shinyFiles (>= 0.6.0), testthat

**License** MIT + file LICENSE

**VignetteBuilder** knitr

**OS\_type** unix

**SystemRequirements** cron

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-03-03 21:25:49

## R topics documented:

cron_add	2
cron_clear	3
cron_load	4
cron_ls	4

cron_njobs . . . . .	5
cron_rm . . . . .	5
cron_rscript . . . . .	6
cron_rstudioaddin . . . . .	7
cron_save . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

cron_add	<i>Make a simple cron job</i>
----------	-------------------------------

---

## Description

Generate a cron job, and pass it to crontab.

The goal is to be able to translate simple English statements of intent to the actual cron statement that could execute that intent. For example,

*"I want to run a job daily at 7AM."*

is simply

```
cron_add(<command>, "daily", at="7AM")
```

Another example, *"I want to run a job on the 15th of every month."*

is

```
cron_add(<command>, "monthly", days_of_month="15th")
```

## Usage

```
cron_add(command, frequency = "daily", at, days_of_month, days_of_week,
months, id, tags = "", description = "", dry_run = FALSE, user = "")
```

```
cronjob(command, frequency = "daily", at, days_of_month, days_of_week, months,
id, tags = "", description = "", dry_run = FALSE, user = "")
```

## Arguments

command	A command to execute.
frequency	A character string equal to one of "minutely", "hourly", "daily", "monthly", or "yearly".
at	The actual time of day at which to execute the command. When unspecified, we default to "3AM", when the command is to be run less frequently than "hourly".
days_of_month	Optional; the day(s) of the month on which we execute the command.
days_of_week	Optional; the day(s) of the week on which we execute the command.
months	Optional; the month(s) of the year on which we execute the command.
id	An id, or name, to give to the cronjob task, for easier revision in the future.
tags	A set of tags, used for easy listing and retrieval of cron jobs.

description	A short description of the job, and its purpose.
dry_run	Boolean; if TRUE we do not submit the cron job; instead we return the parsed text that would be submitted as a cron job.
user	The user whose cron jobs we wish to examine.

### Examples

```
## Not run:
f <- system.file(package = "cronR", "extdata", "helloworld.R")
cmd <- cron_rscript(f)
cmd

cron_add(command = cmd, frequency = 'minutely',
  id = 'test1', description = 'My process 1', tags = c('lab', 'xyz'))
cron_add(command = cmd, frequency = 'daily', at='7AM', id = 'test2')
cron_njobs()

cron_ls()
cron_clear(ask=FALSE)
cron_ls()

cmd <- cron_rscript(f, rscript_args = c("productx", "arg2", "123"))
cmd
cron_add(cmd, frequency = 'minutely', id = 'job1', description = 'Customers')
cron_add(cmd, frequency = 'hourly', id = 'job2', description = 'Weather')
cron_add(cmd, frequency = 'hourly', id = 'job3', days_of_week = c(1, 2))
cron_add(cmd, frequency = 'hourly', id = 'job4', at = '00:20', days_of_week = c(1, 2))
cron_add(cmd, frequency = 'daily', id = 'job5', at = '14:20')
cron_add(cmd, frequency = 'daily', id = 'job6', at = '14:20', days_of_week = c(0, 3, 5))
cron_add(cmd, frequency = 'daily', id = 'job7', at = '23:59', days_of_month = c(1, 30))
cron_add(cmd, frequency = 'monthly', id = 'job8', at = '10:30',
  days_of_month = 'first', days_of_week = '*')
cron_ls()
cron_clear(ask=FALSE)

## End(Not run)
```

---

cron_clear	<i>Clear all cron jobs</i>
------------	----------------------------

---

### Description

Clear all cron jobs

### Usage

```
cron_clear(ask = TRUE, user = "")
```

**Arguments**

ask            Boolean; ask before removal?  
 user           The user whose crontab we are clearing.

**Examples**

```
## Not run:
f <- system.file(package = "cronR", "extdata", "helloworld.R")
cmd <- cron_rscript(f)
cron_add(command = cmd, frequency = 'minutely', id = 'test1', description = 'My process 1')
cron_add(command = cmd, frequency = 'daily', at="7AM", id = 'test2', description = 'My process 2')
cron_njobs()

cron_ls()
cron_clear(ask=FALSE)
cron_ls()

## End(Not run)
```

---

cron\_load            *Load a crontab from file*

---

**Description**

Load a crontab from file

**Usage**

```
cron_load(file, user = "")
```

**Arguments**

file            The file location of a crontab.  
 user           The user for whom we will be loading a crontab.

---

cron\_ls            *List the contents of a crontab*

---

**Description**

We only list the contents that are handled by cronR.

**Usage**

```
cron_ls(id, tags, user = "")
```

**Arguments**

id	Return cron jobs with a certain id.
tags	Return cron jobs with a certain (set of) tags.
user	The user's crontab to display

**Examples**

```
cron_ls()
```

---

cron_njobs	<i>List the number of rCron cron jobs</i>
------------	---

---

**Description**

List the number of rCron cron jobs

**Usage**

```
cron_njobs(user = "")
```

**Arguments**

user	The user whose cron jobs we wish to examine.
------	--

**Examples**

```
cron_njobs()
```

---

cron_rm	<i>Remove a cronjob</i>
---------	-------------------------

---

**Description**

Use this command to remove a cron job added by cron\_add.

**Usage**

```
cron_rm(id, dry_run = FALSE, user = "")
```

**Arguments**

id	A set of ids, partially matched from the beginning, we wish to remove. We only remove the id if it is uniquely matched in the file.
dry_run	Boolean; if TRUE we do not submit the cron job; instead we return the parsed text that would be submitted as a cron job.
user	The user whose crontab we will be modifying.

**Examples**

```
## Not run:
f <- system.file(package = "cronR", "extdata", "helloworld.R")
cmd <- cron_rscript(f)
cron_add(command = cmd, frequency = 'minutely', id = 'test1', description = 'My process 1')
cron_njobs()
cron_ls()
cron_rm(id = "test1")
cron_njobs()
cron_ls()

## End(Not run)
```

---

cron_rscript	<i>Create a command to execute an R script which can be scheduled with cron_add</i>
--------------	---

---

**Description**

Create a command to execute an R script which can be scheduled with cron\_add where the stdin and stderr will be passed on to a log

**Usage**

```
cron_rscript(rscript, rscript_log = sprintf("%s.log",
  tools::file_path_sans_ext(rscript)), rscript_args = "",
  cmd = file.path(Sys.getenv("R_HOME"), "bin", "Rscript"),
  log_append = TRUE)
```

**Arguments**

rscript	character string with the path to an R script with .r or .R extension
rscript_log	where to put the log, defaults in the same directory and with the same filename as rscript but with extension .log.
rscript_args	a character string with extra arguments to be passed on to Rscript
cmd	path to Rscript. Defaults to R_HOME/bin/Rscript
log_append	logical, append to the log or overwrite

**Value**

a character string with a command which can e.g. be put as a cronjob for running a simple R script at specific timepoints

## Examples

```
f <- system.file(package = "cronR", "extdata", "helloworld.R")
cron_rscript(f)
cron_rscript(f, rscript_args = "more arguments passed on to the call")
cron_rscript(f, rscript_args = c("more", "arguments", "passed", "on", "to", "the", "call"))

cron_rscript(f, log_append = FALSE)
cron_rscript(f, log_append = TRUE)
```

---

cron_rstudioaddin	<i>Launch an RStudio addin which allows to schedule an Rscript interactively.</i>
-------------------	---

---

## Description

Launch an RStudio addin which allows to schedule an Rscript interactively.

## Usage

```
cron_rstudioaddin(RscriptRepository)
```

## Arguments

RscriptRepository  
path to the folder where R scripts will be copied to and launched. Defaults to the extdata folder in the cronR R library

## Value

the return of [runGadget](#)

## Examples

```
## Not run:
cron_rstudioaddin()

## End(Not run)
```

---

cron_save	<i>Save the current crontab</i>
-----------	---------------------------------

---

**Description**

Save the current crontab

**Usage**

```
cron_save(file, overwrite = FALSE, user = "")
```

**Arguments**

file	The file location at which you wish to save your crontab.
overwrite	logical; should we overwrite the file at path file if it already exists?
user	The user whose cron jobs we will be saving.

**See Also**

[file.copy](#)

**Examples**

```
## Not run:
cron_add(command = cron_rscript(system.file(package = "cronR", "extdata", "helloworld.R")),
  frequency = 'minutely', id = 'test1', description = 'My process 1')
cron_save(file="crontab_backup", overwrite=TRUE)
cron_clear()
cron_load(file="crontab_backup")

## End(Not run)
```



# Index

`cron_add`, 2  
`cron_clear`, 3  
`cron_load`, 4  
`cron_ls`, 4  
`cron_njobs`, 5  
`cron_rm`, 5  
`cron_rscript`, 6  
`cron_rstudioaddin`, 7  
`cron_save`, 8  
`cronjob (cron_add)`, 2  
  
`file.copy`, 8  
  
`runGadget`, 7