Package 'taskscheduleR'

October 24, 2018

Type Package

Title Schedule R Scripts and Processes with the Windows Task Scheduler

Description Schedule R scripts/processes with the Windows task scheduler. This
allows R users to automate R processes on specific time points from R itself.

Maintainer Jan Wijffels <jwijffels@bnosac.be>

License AGPL-3

Version 1.4

URL https://github.com/bnosac/taskscheduleR

OS_type windows

Date 2018-10-24

Imports data.table, tools, utils

VignetteBuilder knitr

Suggests knitr, miniUI, shiny, testthat

RoxygenNote 6.0.1

NeedsCompilation no

Author Jan Wijffels [aut, cre, cph],
BNOSAC [cph],
Oliver Belmans [cph, aut]

Repository CRAN

Date/Publication 2018-10-24 18:50:03 UTC

R topics documented:

---

taskscheduleR-package   *Schedule R Scripts/Processes with the Windows Task Scheduler*

Description

Schedule R scripts/processes with the Windows task scheduler. This allows R users to automate R processes on specific timepoints from R itself. The package is basically a wrapper around the Schtasks.exe functionality. More information about schtasks can be found at https://msdn.microsoft.com/en-us/library/windows/desktop/bb736357 or at the doc folder inside this package.

---

taskschedulerAddin   *Launch an RStudio addin which allows to schedule an Rscript inter-actively.*

Description

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

Usage

```
taskschedulerAddin(RscriptRepository, debug = TRUE)
```

Arguments

RscriptRepository

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

debug   passed on to [taskscheduler_create](#)

Value

the return of [runGadget](#)

Examples

```
## Not run:
taskschedulerAddin()

## End(Not run)
```

---

taskscheduler_create *Schedule an R script with the Windows task scheduler.*

Description

Schedule an R script with the Windows task scheduler. E.g. daily, weekly, once, at startup, ... More information about the scheduling format can be found in the docs/schtasks.pdf file inside this package. The rscript file will be scheduled with Rscript.exe and the log of the run will be put in the .log file which can be found in the same directory as the location of the rscript

Usage

```
taskscheduler_create(taskname = basename(rscript), rscript,
  schedule = c("ONCE", "MONTHLY", "WEEKLY", "DAILY", "HOURLY", "MINUTE",
  "ONLOGON", "ONIDLE"), starttime = format(Sys.time() + 62, "%H:%M"),
  startdate = format(Sys.Date(), "%d/%m/%Y"), days = c("*", "MON", "TUE",
  "WED", "THU", "FRI", "SAT", "SUN", 1:31), months = c("*", "JAN", "FEB",
  "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP", "OCT", "NOV", "DEC"),
  modifier, idletime = 60L, Rexe = file.path(Sys.getenv("R_HOME"), "bin",
  "Rscript.exe"), rscript_args = "", schtasks_extra = "", debug = FALSE)
```

Arguments

taskname	a character string with the name of the task. Defaults to the filename. Should not contain any spaces.
rscript	the full path to the .R script with the R code to execute. Should not contain any spaces.
schedule	when to schedule the rscript. Either one of 'ONCE', 'MONTHLY', 'WEEKLY', 'DAILY', 'HOURLY', 'MINUTE', 'ONLOGON', 'ONIDLE'.
starttime	a timepoint in HH:mm format indicating when to run the script. Defaults to within 62 seconds.
startdate	a date that specifies the first date on which to run the task. Only applicable if schedule is of type 'MONTHLY', 'WEEKLY', 'DAILY', 'HOURLY', 'MINUTE'. Defaults to today in '%d/%m/%Y' format. Change to your locale format if needed.
days	character string with days on which to run the script if schedule is 'WEEKLY' or 'MONTHLY'. Possible values are * (all days). For weekly: 'MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT', 'SUN' or a vector of these in your locale. For monthly: 1:31 or a vector of these.
months	character string with months on which to run the script if schedule is 'MONTHLY'. Possible values are * (all months) or 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC' or a vector of these in your locale.
modifier	a modifier to apply. See the docs/schtasks.pdf

idletime	integer containing a value that specifies the amount of idle time to wait before running a scheduled ONIDLE task. The valid range is 1 - 999 minutes.
Rexe	path to Rscript.exe which will be used to run the script. Defaults to Rscript at the bin folder of R_HOME.
rscript_args	character string with further arguments passed on to Rscript. See args in Rscript.
schtasks_extra	character string with further schtasks arguments. See the inst/docs/schtasks.pdf
debug	logical to print the system call to screen

Value

the system call to schtasks /Create

Examples

```
myscript <- system.file("extdata", "helloworld.R", package = "taskscheduleR")
cat(readLines(myscript), sep = "\n")

## Not run:
## Run script once at a specific timepoint (within 62 seconds)
runon <- format(Sys.time() + 62, "%H:%M")
taskscheduler_create(taskname = "myfancyscript", rscript = myscript,
 schedule = "ONCE", starttime = runon)

## Run every day at the same time on 09:10, starting from tomorrow on
## Mark: change the format of startdate to your locale if needed (e.g. US: %m/%d/%Y)
taskscheduler_create(taskname = "myfancyscriptdaily", rscript = myscript,
 schedule = "DAILY", starttime = "09:10", startdate = format(Sys.Date()+1, "%d/%m/%Y"))

## Run every week on Sunday at 09:10
taskscheduler_create(taskname = "myfancyscript_sun", rscript = myscript,
  schedule = "WEEKLY", starttime = "09:10", days = 'SUN')

## Run every 5 minutes, starting from 10:40
taskscheduler_create(taskname = "myfancyscript_5min", rscript = myscript,
  schedule = "MINUTE", starttime = "10:40", modifier = 5)

## Run every minute, giving some command line arguments which can be used in the script itself
taskscheduler_create(taskname = "myfancyscript_withargs_a", rscript = myscript,
  schedule = "MINUTE", rscript_args = "productxyz 20160101")
taskscheduler_create(taskname = "myfancyscript_withargs_b", rscript = myscript,
  schedule = "MINUTE", rscript_args = c("productabc", "20150101"))

alltasks <- taskscheduler_ls()
subset(alltasks, TaskName %in% c("myfancyscript", "myfancyscriptdaily"))
# The field TaskName might have been different on Windows with non-english language locale

taskscheduler_delete(taskname = "myfancyscript")
taskscheduler_delete(taskname = "myfancyscriptdaily")
taskscheduler_delete(taskname = "myfancyscript_sun")
taskscheduler_delete(taskname = "myfancyscript_5min")
taskscheduler_delete(taskname = "myfancyscript_withargs_a")
```

```
taskscheduler_delete(taskname = "myfancyscript_withargs_b")

## Have a look at the log
mylog <- system.file("extdata", "helloworld.log", package = "taskscheduleR")
cat(readLines(mylog), sep = "\n")

## End(Not run)
```

---

taskscheduler_delete	*Delete a specific task which was scheduled in the Windows task scheduler.*

Description

Delete a specific task which was scheduled in the Windows task scheduler.

Usage

```
taskscheduler_delete(taskname)
```

Arguments

taskname   the name of the task to delete. See the example.

Value

the system call to schtasks /Delete

Examples

```
## Not run:
x <- taskscheduler_ls()
x
# The field TaskName might have been different on Windows with non-english language locale
task <- x$TaskName[1]
taskscheduler_delete(taskname = task)

## End(Not run)
```

---

taskscheduler_ls	*Get all the tasks which are currently scheduled at the Windows task scheduler.*

Description

Get all the tasks which are currently scheduled at the Windows task scheduler.

Usage

```
taskscheduler_ls(encoding = "UTF-8", ...)
```

Arguments

encoding	encoding of the CSV which schtasks.exe generates. Defaults to UTF-8.
...	optional arguments passed on to fread in order to read in the CSV file which schtasks generates

Value

a data.frame with scheduled tasks as returned by schtasks /Query for which the Taskname or second column in the dataset the preceding \ is removed

Examples

```
x <- taskscheduler_ls()
x
```

---

taskscheduler_runnow	*Immediately run a specific task available in the Windows task scheduler.*

Description

Immediately run a specific task available in the Windows task scheduler.

Usage

```
taskscheduler_runnow(taskname)
```

Arguments

taskname	the name of the task to run. See the example.

Value

the system call to schtasks /Run

Examples

```
## Not run:
myscript <- system.file("extdata", "helloworld.R", package = "taskscheduleR")
taskscheduler_create(taskname = "myfancyscript", rscript = myscript,
 schedule = "ONCE", starttime = format(Sys.time() + 10*60, "%H:%M"))

taskscheduler_runnow("myfancyscript")
Sys.sleep(5)
taskscheduler_stop("myfancyscript")


taskscheduler_delete(taskname = "myfancyscript")

## End(Not run)
```

---

taskscheduler_stop	*Stop the run of a specific task which is running in the Windows task scheduler.*

Description

Stop the run of a specific task which is running in the Windows task scheduler.

Usage

```
taskscheduler_stop(taskname)
```

Arguments

taskname   the name of the task to stop. See the example.

Value

the system call to schtasks /End

Examples

```
## Not run:
myscript <- system.file("extdata", "helloworld.R", package = "taskscheduleR")
taskscheduler_create(taskname = "myfancyscript", rscript = myscript,
 schedule = "ONCE", starttime = format(Sys.time() + 10*60, "%H:%M"))

taskscheduler_runnow("myfancyscript")
Sys.sleep(5)
taskscheduler_stop("myfancyscript")


taskscheduler_delete(taskname = "myfancyscript")

## End(Not run)
```

Index