Package 'ptspotter'

August 13, 2023

Title Helper Functions for Use with ``ProjectTemplate''

Version 1.0.2

Description Utility functions produced specifically for (but not limited to) working with 'ProjectTemplate' data pipelines. This package helps to quickly create and manage sequentially numbered scripts, quickly set up logging with 'log4r' and functions to help debug and monitor procedures.

License MIT + file LICENSE

URL https://github.com/r-leyshon/ptspotter

BugReports https://github.com/r-leyshon/ptspotter/issues

Depends beepr (>= 1.3), log4r (>= 0.3.2), this.path (>= 0.2.0)

Imports pryr (>= 0.1.4), stringr (>= 1.4.0), utils

Suggests covr, knitr, markdown, ProjectTemplate (>= 0.9.3), rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Rich Leyshon [aut, cph, cre], Iris Simmons [ctb]

Maintainer Rich Leyshon <leyshonrr@hotmail.co.uk>

Repository CRAN

Date/Publication 2023-08-13 14:00:02 UTC

R topics documented:

`adj_file_nos` *Adjust file numbers.*

Description

This function is used to increment / decrease sequential scripts within the specified directory, allowing efficient adjustment of script sequence for additional or removed files.

Usage

```
adj_file_nos(target, directory = NULL, action = "up", step = 1)
```

Arguments

`target`	Required. The number in the sequential scripts to begin the adjustment. Use single digits only. The adjustment will affect script with that leading digit and greater.
`directory`	The directory holding the sequential scripts.
`action`	Defaults to "up". Whether to adjust file numbers up or down.
`step`	Defaults to 1. The step by which to increment or decrement the file numbering.

Value

Renumbers filenames in the specified directory, according to the specified action. Only affects the target file and above.

Examples

```
seq_file_ops(n = 10, target_dir = "munge")

# Increase files numbered 6 and above by 1
adj_file_nos(target = 6, directory = "munge")

# Increase above target files by a further 2
adj_file_nos(target = 6, directory = "munge", step = 2)

# Use step = "down" to restore original sequence
adj_file_nos(target = 6, directory = "munge", action = "down", step = 3)

# writing books or websites:
seq_file_ops(n = 5, target_dir = "images", filetype = "png")
# adjust by decimals
adj_file_nos(target = 1, directory = "images", step = 0.1)

# tidying up environment
```

```
unlink(c("munge", "images"), recursive = TRUE)
```

---

log_enable *log_enable*

Description

Assigns the necessary global scope objects for logging with "log4r".

Usage

```
log_enable(
  logfile_loc = NULL,
  pos = 1,
  logger_nm = my_logger,
  appender_nm = file_app
)
```

Arguments

logfile_loc	The path to the logfile. Suggested use "logs/logfile.txt".
pos	The environment which to assign pipeline_message. Defaults to 1, equivalent to the .GlobalEnv.
logger_nm	What to call the logger. Provide unquoted strings with no spaces. Defaults to my_logger.
appender_nm	What to call the appender function. Provide unquoted strings with no spaces. Defaults to file_app.

Value

Creates logger and file appender.

Examples

```
# create logging infrastructure
log_file_ops(dir_path = "logs/logfile.txt")
# enable logging
log_enable(logfile_loc = "logs/logfile.txt")

# tidy up environment
unlink("logs", recursive = TRUE)
```

log_file_ops *log_file_ops*

Description

Create the necessary file infrastructure to efficiently start logging with "log4r".

Usage

```
log_file_ops(dir_path = NULL, logfile_nm = "logfile")
```

Arguments

dir_path	The name of the folder in which the logfile should be saved. Creates the folder if required.
logfile_nm	Provide a name for the logfile. Do not include suffix. Defaults to "logfile".

Value

Creates log directory and log file if required. Calls log_enable() to assign necessary logging objects in specified scope.

Examples

```
log_file_ops(dir_path = "logs")

unlink("logs", recursive = TRUE)
```

memory_report *Perform garbage collection and log allocated memory.*

Description

Used to log memory allocation at points during sequential script execution.

Usage

```
memory_report()
```

Value

Performs garbage collection then messages memory size and script name currently being executed.

Examples

```
try(memory_report())
```

`seq_file_ops` *seq_file_ops*

Description

Quickly create the required number of sequentially labelled files.

Usage

```
seq_file_ops(n, target_dir = NULL, filetype = "R", force = FALSE)
```

Arguments

`n`	The number of files to create. Also accepts numerical vector.
`target_dir`	Directory to create files. Creates the directory if file.exists(target_dir) evaluates to FALSE.
`filetype`	The suffix to append the filename. Defaults to ".R".
`force`	Defaults to FALSE. If set to TRUE, seq_file_ops will overwrite any pre-existing files that match the write filenames asked for.

Value

Write a series of sequentially numbered files within a specified directory. Creates the directory if required.

Examples

```
seq_file_ops(n = 10, target_dir = "munge")

seq_file_ops(n = c(1, 3:8, 10), target_dir = "complex_vector")

# if force == FALSE, pre-existing numbered scripts will not be overwritten
# only 02-.R and 09-.R are written below
seq_file_ops(10, target_dir = "complex_vector")

unlink("munge", recursive = TRUE)
unlink("complex_vector", recursive = TRUE)
```

wrap_up *Wrap up file execution.*

Description

Used to interrupt sequential script execution while testing or debugging. Outputs an auditory signal and breaks sequential script execution, identifying the script at which execution was interrupted. Is a Sys.time() object is passed to `start_time`, messages the elapsed time.

Usage

```
wrap_up(start_time = NULL)
```

Arguments

`start_time`	Optional POSIXct object, created by assigning Sys.time() to an object prior to executing `wrap_up()`.

Value

Interrupts sequential script execution with an auditory signal. Logs the elapsed time if start_time is used, outputs the script location.

Examples

```
# halt execution with no timing
try(wrap_up())

# create timing checkpoint
s_time <- Sys.time()
# halt execution with timing
try(wrap_up(s_time))
```

Index