

Package: muLogR (via r-universe)

October 14, 2026

Title muLogR

Description Logging and code structuring for R. Based on code from the RnBeads package.

Date 2019-06-06

Version 0.2

Encoding UTF-8

License GPL-3

Collate 'muLogR.R' 'logger.R'

RoxygenNote 6.1.1

Repository <https://blaserlab.r-universe.dev>

Date/Publication 2019-06-06 17:46:46 UTC

RemoteUrl <https://github.com/demuellae/muLogR>

RemoteRef HEAD

RemoteSha dd5eac923ff9c9cea8205d22b58cdf719105d6cd

Contents

logger.getfiles	2
logger.isinitialized	2
logger.start	3
logger.status	4
logger.validate.file	5

Index	7
--------------	----------

logger.getfiles	<i>logger.getfiles</i>
-----------------	------------------------

Description

Gets the files currently used by the logger.

Usage

```
logger.getfiles()
```

Value

Vector storing the full names of the files that are being used by the logger. This vector contains NA as an element if the logger is (also) using the console for its output. If the logger is not initialized, this function returns NULL.

Author(s)

adapted by Fabian Mueller from RnBeads code by Yassen Assenov

See Also

[logger.isinitialized](#) to check if logging is activated; [logger.start](#) for initializing a logger or starting a section

Examples

```
if (NA %in% logger.getfiles())  
  cat("Console logger is enabled\n")
```

logger.isinitialized	<i>logger.isinitialized</i>
----------------------	-----------------------------

Description

Checks if the logger is initialized.

Usage

```
logger.isinitialized()
```

Value

TRUE if the logger was initialized and is in use; FALSE otherwise.

Author(s)

adapted by Fabian Mueller from RnBeads code by Yassen Assenov

See Also

[logger.start](#) for initializing a logger or starting a section

Examples

```
if (!logger.isinitialized())  
  logger.start(fname = NA)
```

`logger.start`*Log File Management*

Description

Functions for logger management.

Usage

```
logger.start(txt = character(0), fname = NULL)  
  
logger.completed()  
  
logger.close()
```

Arguments

txt	Description to add to the log file. The words STARTED and COMPLETED are prepended to the message upon initialization and completion of the section, respectively.
fname	Name of the log file and/or console. Note that at most one file name can be specified. The function <code>logger.start</code> normalizes the given name, that is, it converts it to an absolute name. If this parameter is NA, logger messages are printed to the console. If it is a two-element vector containing one file name and NA, the logger is (re)initialized to print messages both to the given file name and the console. A value of NULL (default) indicates the logger should continue using the previously specified file.

Value

None (invisible NULL).

Details

`logger.start` initializes the logger and/or starts a new section. `logger.completed` completes the last (innermost) open section in the log. `logger.close` deinitializes the logger. Note that after reinitialization or deinitialization, the information about the current output file, as well as any open sections, is deleted.

Author(s)

adapted by Fabian Mueller from RnBeads code by Yassen Assenov

See Also

`logger.isinitialized`

Examples

```
if (!logger.isinitialized())
  logger.start(fname = NA)
logger.start("Tests for Significance")
logger.completed()
logger.close()
```

`logger.status`

Writing text messages to the log file.

Description

Appends a single-line status message to the log text file. The message is prepended by its type, which is one of STATUS, INFO, WARNING or ERROR.

Usage

```
logger.status(txt)

logger.info(txt)

logger.warning(txt)

logger.error(txt, terminate = FALSE)
```

Arguments

<code>txt</code>	Text to add to the log file. This must be a character vector; its elements are concatenated using a single space (" ") as a separator.
<code>terminate</code>	Flag indicating if the execution is to be terminated after this error message is added to the log.

Value

None (invisible NULL).

Author(s)

adapted by Fabian Mueller from RnBeads code by Yassen Assenov

See Also

[logger.isinitialized](#) to check if logging is activated; [logger.start](#) for initializing a logger or starting a section

Examples

```
if (!logger.isinitialized())
  logger.start(fname = NA)
logger.status(c("Reached step", 2))
```

`logger.validate.file` *logger.validate.file*

Description

Validates the specified file or directory exists. Prints an error or a warning message to the log if it does not exist, it is not of the accepted type or is not accessible.

Usage

```
logger.validate.file(file, is.file = TRUE, terminate = TRUE)
```

Arguments

<code>file</code>	Name of file or directory to validate.
<code>is.file</code>	Flag indicating if the given name must denote an existing file. If this is FALSE, the given name must denote a directory. Set this to NA if both types are an acceptable scenario.
<code>terminate</code>	Flag indicating if the execution is to be terminated in case the validation fails. This parameter determines if an error message (<code>terminate</code> is TRUE) or a warning message (<code>terminate</code> is FALSE) is to be sent to the log when the specified file or directory does not exist, is not of the accepted type or is not accessible.

Value

Whether the validation succeeded or not, invisibly. Note that when `terminate` is TRUE and the validation fails, the R session is closed and thus no value is returned.

Author(s)

adapted by Fabian Mueller from RnBeads code by Yassen Assenov

Examples

```
if (!logger.isinitialized())  
  logger.start(fname = NA)  
# Validate the current working directory exists  
logger.validate.file(getwd(), FALSE)
```

Index

`logger.close(logger.start)`, 3
`logger.completed(logger.start)`, 3
`logger.error(logger.status)`, 4
`logger.getfiles`, 2
`logger.info(logger.status)`, 4
`logger.isinitialized`, 2, 2, 5
`logger.start`, 2, 3, 3, 5
`logger.status`, 4
`logger.validate.file`, 5
`logger.warning(logger.status)`, 4