

Package ‘nat.utils’

July 22, 2025

Title File System Utility Functions for 'NeuroAnatomy Toolbox'

Version 0.6.1

Description Utility functions that may be of general interest but are specifically required by the 'NeuroAnatomy Toolbox' ('nat'). Includes functions to provide a basic make style system to update files based on timestamp information, file locking and 'touch' utility. Convenience functions for working with file paths include 'abs2rel', 'split_path' and 'common_path'. Finally there are utility functions for working with 'zip' and 'gzip' files including integrity tests.

License GPL (>= 3)

Imports utils, checkmate

Suggests testthat (>= 0.9), roxygen2, covr

ByteCompile true

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

URL <https://github.com/natverse/nat.utils>

BugReports <https://github.com/natverse/nat.utils/issues>

NeedsCompilation no

Author Gregory Jefferis [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-0587-9355>>)

Maintainer Gregory Jefferis <jefferis@gmail.com>

Repository CRAN

Date/Publication 2023-06-07 08:30:05 UTC

Contents

abs2rel	2
common_path	3

file.swap	4
find_extdata	5
gzip.crc	6
is.gzip	6
makelock	7
make_chunks	8
ncpus	9
read_nl_from_parts	9
RunCmdForNewerInput	10
save_nl_in_parts	12
split_path	13
touch	14
zipinfo	15
zipok	15
Index	17

abs2rel	<i>Remove common part of two paths, leaving relative path</i>
---------	---

Description

Remove common part of two paths, leaving relative path

Usage

abs2rel(path, stempath = getwd(), StopIfNoCommonPath = FALSE)

Arguments

- path Paths to make relative
- stempath Root to which path will be made relative
- StopIfNoCommonPath Error if no path in common

Value

Character vector containing relative path

Author(s)

jefferis

See Also

[path.expand](#), [normalizePath](#)
Other path_utils: [common_path\(\)](#), [split_path\(\)](#)

Examples

```
path = "/Volumes/JData/JPeople/Sebastian/images"
abs2rel(path, '/Volumes/JData')
```

common_path	<i>Find common prefix of two or more (normalised) file paths</i>
-------------	--

Description

Find common prefix of two or more (normalised) file paths

Usage

```
common_path(paths, normalise = FALSE, fsep = .Platform$file.sep)
```

Arguments

paths	Character vector of file paths
normalise	Whether to normalise paths (with normalizePath , default FALSE)
fsep	Optional path separator (defaults to <code>.Platform\$file.sep</code>)

Details

Note that for absolute paths, the common prefix will be returned e.g. `common_path(c("/a", "/b"))` is `"/"`

Note that [normalizePath](#) 1) operates according to the conventions of the current runtime platform 2) is called with `winslash=.Platform$file.sep` which means that normalised paths will eventually end up separated by `"\"` by default on Windows rather than by `"/"`, which is `normalizePath`'s standard behaviour.

Value

Character vector of common prefix, `""` when there is no common prefix, or the original value of paths when fewer than 2 paths were supplied.

See Also

[normalizePath](#)

Other path_utils: [abs2rel\(\)](#), [split_path\(\)](#)

Examples

```

common_path(c("/a", "/b"))
common_path(c("/a/b/", "/a/b"))
common_path(c("/a/b/d", "/a/b/c/d"))
common_path(c("/a/b/d", "/b/c/d"))
common_path(c("a", "b"))
common_path(c("", "/a"))
common_path(c("~/", "~/"))
common_path(c("~/a/b/d", "~/a/b/c/d"), normalise = FALSE)
common_path(c("~/", "~/"), normalise = FALSE)

```

file.swap

Swap names of two files (by renaming first to a temporary file)

Description

Swap names of two files (by renaming first to a temporary file)

Usage

```
file.swap(f1, f2)
```

Arguments

f1, f2 Paths to files

Value

logical indicating success

Author(s)

jefferis

See Also

[file.rename](#)

find_extdata	<i>Construct paths to files in the extdata folder of a package</i>
--------------	--

Description

Construct paths to files in the extdata folder of a package

Usage

```
find_extdata(..., package = NULL, firstpath = NULL, Verbose = FALSE)
```

Arguments

...	components of the path (eventually appended to location of extdata)
package	The package to search
firstpath	An additional location to check before looking anywhere else
Verbose	Whether to print messages about failed paths while looking for extdata

Details

inst/extdata is the conventional place to store data that is not managed directly by the standard R package mechanisms. Unfortunately its location changes at different stages of the package build/load process, since in the final package all folders underneath inst are moved directly to the package root.

Value

A character vector containing the constructed path

See Also

Other extdata: [read_nl_from_parts\(\)](#), [save_nl_in_parts\(\)](#)

Examples

```
find_extdata(package='nat.utils')
```

`gzip.crc`*Extract the CRC (32 bit hash) of a gzip file*

Description

Reads the crc from a gzip file, assuming it is the last 4 bytes of the file. First checks for a valid gzip magic number at the start of the file.

Usage

```
gzip.crc(f)
```

Arguments

<code>f</code>	Path to a gzip file
----------------	---------------------

Details

CRC32 is not a strong hash like SHA1 or even MD5, but it does provide a basic hash of the **un-compressed contents** of the gzip file. NB CRCs are stored in little endian byte order regardless of platform.

Value

hexadecimal formatted

Examples

```
rdsfile=system.file('help/aliases.rds')
gzip.crc(rdsfile)
```

`is.gzip`*Check if a file is a gzip file*

Description

Check if a file is a gzip file

Usage

```
is.gzip(f)
```

Arguments

<code>f</code>	Path to file to test
----------------	----------------------

Value

logical indicating whether `f` is in gzip format (or NA if the file cannot be accessed)

Examples

```
notgzipfile=tempfile()
writeLines('not a gzip', notgzipfile)
is.gzip(notgzipfile)
con=gzfile(gzipfile<-tempfile(),open='wt')
writeLines('This one is gzipped', con)
close(con)
is.gzip(gzipfile)
unlink(c(notgzipfile,gzipfile))
```

`makelock`

Make and remove (NFS safe) lock files

Description

Creates a lock file on disk containing a message that should identify the current R session. Will return FALSE if someone else has already made a lockfile. In order to avoid race conditions typical on NFS mounted drives `makelock` appends a unique message to the lock file and then reads the file back in. Only if the unique message is the first line in the file will `makelock` return TRUE.

`removelock` displays a warning and returns false if lockfile cannot be removed. No error message is given if the file does not exist.

Usage

```
makelock(lockfile, lockmsg, CreateDirectories = TRUE)
```

```
removelock(lockfile)
```

Arguments

<code>lockfile</code>	Path to lockfile
<code>lockmsg</code>	Character vector with message to be written to lockfile
<code>CreateDirectories</code>	Recursively create directories implied by lockfile path

Value

logical indicating success

Author(s)

jefferis

Examples

```

makelock(lock<-tempfile())
stopifnot(!makelock(lock))
removelock(lock)

```

make_chunks

Split inputs into a number of chunks

Description

Split inputs into a number of chunks

Usage

```
make_chunks(x, size = length(x), nchunks = NULL, chunksize = NULL)
```

Arguments

x	A vector of inputs e.g. ids, neurons etc (optional, see examples)
size	The number of inputs (defaults to length(x) when x is present)
nchunks	The desired number of chunks
chunksize	The desired number of items per chunk

Details

You must specify exactly one of nchunks and chunksize.

Value

The elements of x split into a list of chunks or (when x is missing) a vector of integer indices in the range 1:nchunks specifying the chunk for each input element .

Examples

```

make_chunks(1:11, nchunks=2)
make_chunks(size=11, chunksize=2)

```

ncpus	<i>Return number of cpus (or a default on failure)</i>
-------	--

Description

Return number of cpus (or a default on failure)

Usage

```
ncpus(default = 1L)
```

Arguments

default Number of cores to assume if detectCores fails

Value

Integer number of cores

integer number of cores always ≥ 1 for default values

Author(s)

jefferis

See Also

[detectCores](#)

Examples

```
ncpus()
```

read_nl_from_parts	<i>Make a neuronlist object from two separate files</i>
--------------------	---

Description

Make a neuronlist object from two separate files

Usage

```
read_nl_from_parts(datapath, dfpath = NULL, package = NULL, ...)
```

Arguments

datapath	Path to the data object
dfpath	Path to the data.frame object (constructed from datapath when NULL, see details)
package	Character vector naming a package whose extdata directory will be sought (with find_extdata) and prepended to the two input paths.
...	Additional arguments passed to find_extdata

Details

It is expected that you will use this in an R source file within the data folder of a package. See **Examples** for more information.

If dfpath is missing, it will be inferred from datapath according to the following pattern:

- myblob.rda main data file
- myblob.df.rda metadata file

Value

a neuronlist object

See Also

Other extdata: [find_extdata\(\)](#), [save_nl_in_parts\(\)](#)

Examples

```
## Not run:
# you could use the following in a file
# data/make_data.R
delayedAssign('pns', read_nl_from_parts('pns.rds', package='testlazyneuronlist'))
# based on objects created by
save_nl_in_parts(pns)
# which would make:
# - inst/extdata/pns.rds
# - inst/extdata/pns.df.rds

## End(Not run)
```

RunCmdForNewerInput *Run a command if input files are newer than outputs*

Description

Run a command if input files are newer than outputs

Usage

```
RunCmdForNewerInput(
  cmd,
  infiles,
  outfiles,
  Verbose = FALSE,
  UseLock = FALSE,
  Force = FALSE,
  ReturnInputTimes = FALSE,
  ...
)
```

Arguments

cmd	An expression , a string or NA/NULL
infiles	Character vector of path to one or more input files
outfiles	Character vector of path to one or more output files
Verbose	Write information to consolve (Default FALSE)
UseLock	Stop other processes working on this task (Default FALSE)
Force	Ignore file modification times and always produce output if input files exist.
ReturnInputTimes	Return mtimes of input files (default FALSE)
...	additional parameters passed to system call.

Details

cmd can be an R expression, which is [evaluated](#) if necessary in the environment calling RunCmdForNewerInput, a string to be passed to [system](#) or NULL/NA in which cases the files are checked and TRUE or FALSE is returned depending on whether action is required.

When UseLock=TRUE, the lock file created is called outfiles[1].lock

When ReturnInputTimes=TRUE, the input mtimes are returned as an attribute of a logical value (if available).

Value

logical indicating if cmd was run or for an R expression, eval(cmd)

See Also

[makelock](#), [eval](#), [expression](#)

Examples

```
## Not run:
RunCmdForNewerInput(expression(myfunc("somefile")))

## End(Not run)
```

save_nl_in_parts

Save a neuronlist object into separate data and metadata parts

Description

Save a neuronlist object into separate data and metadata parts

Usage

```
save_nl_in_parts(
  x,
  datapath = NULL,
  dfpath = NULL,
  extdata = TRUE,
  format = c("rds", "rda"),
  ...
)
```

Arguments

x	A neuronlist object to save in separate parts
datapath	Optional path to new data file (constructed from name of x argument when missing)
dfpath	Optional path to new metadata file (constructed from datapath when missing)
extdata	Logical indicating whether the files should be saved into extdata folder (default TRUE, when FALSE the paths are untouched)
format	Either 'rds' (default) or 'rda'.
...	Additional arguments passed to saveRDS or save (based on the value of format).

Details

Saves a neuronlist into separate data and metadata parts. This can significantly mitigate git repository bloat since only the metadata object will change when any metadata is updated. By default the objects will be saved into the package inst/extdata folder with sensible names based on the incoming object. E.g. if x=mypns the files will be

- mypns.rds
- mypns.df.rds

Value

character vector with path to the saved files (returned invisibly)

See Also

Other extdata: [find_extdata\(\)](#), [read_nl_from_parts\(\)](#)

Examples

```
## Not run:
save_nl_in_parts(pns)
# which would make:
# - inst/extdata/pns.rds
# - inst/extdata/pns.df.rds

save_nl_in_parts(pns, format='rda')
# which would make:
# - inst/extdata/pns.rda
# - inst/extdata/pns.df.rda

save_nl_in_parts(pns, 'mypns.rda')
# which would make (NB format argument wins):
# - inst/extdata/mypns.rds
# - inst/extdata/mypns.df.rds

## End(Not run)
```

split_path	<i>Split file path into individual components (optionally including separators)</i>
------------	---

Description

Split file path into individual components (optionally including separators)

Usage

```
split_path(
  path,
  include.fseps = FALSE,
  omit.duplicate.fseps = FALSE,
  fsep = .Platform$file.sep
)
```

Arguments

path	A path with directories separated by fseps.
include.fseps	Whether to include the separators in the returned character vector (default FALSE)
omit.duplicate.fseps	Whether to omit duplicate file separators if include.fseps=TRUE (default FALSE).
fsep	The path separator (default to .Platform\$file.sep)

Value

A character vector with one element for each component in the path (including path separators if include.fseps=TRUE).

See Also[file.path](#)Other path_utils: [abs2rel\(\)](#), [common_path\(\)](#)**Examples**

```

split_path("/a/b/c")
split_path("a/b/c")
parts=split_path("/a/b/c", include.fseps=TRUE)
# join parts back up again
paste(parts, collapse = "")
split_path("a/b//c", include.fseps=TRUE, omit.duplicate.fseps=TRUE)
# Windows style
split_path("C:\\a\\b\\c", fsep="\\")

```

touch

*Use unix touch utility to change file's timestamp***Description**

If neither a time or a reference file is provided then the current time is used. If the file does not already exist, it is created unless Create=FALSE.

Usage

```

touch(
  file,
  time,
  reference,
  timestoupdate = c("access", "modification"),
  Create = TRUE
)

```

Arguments

file	Path to file to modify
time	Absolute time in POSIXct format
reference	Path to a reference file
timestoupdate	"access" or "modification" (default both)
Create	Logical indicating whether to create file (default TRUE)

Value

TRUE or FALSE according to success

Author(s)

jefferis

zipinfo	<i>Return information about a zip archive using system unzip command</i>
---------	--

Description

Return information about a zip archive using system unzip command

Usage

```
zipinfo(f)
```

Arguments

f	Path to one (or more) files
---	-----------------------------

Details

Uses system unzip command.

Value

dataframe of information

Author(s)

jefferis

See Also

[zip](#)

Other ziputils: [zipok\(\)](#)

zipok	<i>Verify integrity of one or more zip files</i>
-------	--

Description

Verify integrity of one or more zip files

Usage

```
zipok(f, Verbose = FALSE)
```

Arguments

f	Path to one (or more) files
Verbose	Whether to be Verbose (default FALSE)

Details

Uses system unzip command.

Value

TRUE when file OK, FALSE otherwise

Author(s)

jefferis

See Also

Other ziputils: [zipinfo\(\)](#)

Index

- * **extdata**
 - find_extdata, [5](#)
 - read_nl_from_parts, [9](#)
 - save_nl_in_parts, [12](#)
- * **path_utils**
 - abs2rel, [2](#)
 - common_path, [3](#)
 - split_path, [13](#)
- * **ziputils**
 - zipinfo, [15](#)
 - zipok, [15](#)
- abs2rel, [2](#), [3](#), [14](#)
- common_path, [2](#), [3](#), [14](#)
- detectCores, [9](#)
- eval, [11](#)
- expression, [11](#)
- file.path, [14](#)
- file.rename, [4](#)
- file.swap, [4](#)
- find_extdata, [5](#), [10](#), [12](#)
- gzip.crc, [6](#)
- is.gzip, [6](#)
- make_chunks, [8](#)
- makelock, [7](#), [11](#)
- ncpus, [9](#)
- normalizePath, [2](#), [3](#)
- path.expand, [2](#)
- read_nl_from_parts, [5](#), [9](#), [12](#)
- removeLock (makelock), [7](#)
- RunCmdForNewerInput, [10](#)
- save, [12](#)
- save_nl_in_parts, [5](#), [10](#), [12](#)
- saveRDS, [12](#)
- split_path, [2](#), [3](#), [13](#)
- system, [11](#)
- touch, [14](#)
- zip, [15](#)
- zipinfo, [15](#), [16](#)
- zipok, [15](#), [15](#)