Package 'nat.utils'

August 29, 2016

Maintainer Gregory Jefferis <jefferis@gmail.com>

Author Gregory Jefferis

Version 0.5.1

License GPL-3

Title File System Utility Functions for 'NeuroAnatomy Toolbox'

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.

Suggests testthat (>= 0.9), roxygen2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-07-04 15:24:37

R topics documented:

`abs2rel` *Remove common part of two paths, leaving relative path*

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

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 .Platform$file.sep)

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

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

f	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 **uncompressed 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

f	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 is 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

`lockfile`	Path to lockfile
`lockmsg`	Character vector with message to be written to lockfile
`CreateDirectories`	Recursively create directories implied by lockfile path

Value

logical indicating success

Author(s)

jefferis

Examples

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

`nat.utils` *nat.utils: File System Utility Functions for NeuroAnatomy Toolbox*

Description

Utility functions to support the NeuroAnatomy Toolbox (nat). Includes functions to provide a basic make style system to update files based on timestamp information, file locking and other convenience functions for working with the filesystem

See Also

nat

`nat.utils-defunct` *nat.utils: Defunct functions*

Description

These functions have been retired from `nat.utils`

Usage

```
file.hardlink(from, to)
```

Arguments

`from`	Source file
`to`	(New) target hardlink file to create

See Also

`file.link`

ncpus *Return number of cpus (or a default on failure)*

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()
```

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 consolse (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)
```

---

split_path *Split file path into individual components (optionally including separators)*

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 *Return information about a zip archive using system unzip command*

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` *Verify integrity of one or more zip files*

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