

Package: `filenamer` (via `r-universe`)

August 31, 2024

Type Package

Title Easy Management of File Names

Version 0.2.4

Author David J. H. Shih

Maintainer David J. H. Shih <djh.shih@gmail.com>

Description Create descriptive file names with ease. New file names are automatically (but optionally) time stamped and placed in date stamped directories. Streamline your analysis pipeline with input and output file names that have informative tags and proper file extensions.

URL <https://bitbucket.org/djhshih/filenamer>

BugReports <https://bitbucket.org/djhshih/filenamer/issues>

License GPL (>= 3)

Depends methods

Suggests io, testthat

Encoding UTF-8

RoxygenNote 7.2.3

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

RemoteUrl <https://bitbucket.org/djhshih/filenamer>

RemoteRef HEAD

RemoteSha 940890150579a61ef83f34b40e55d14efdba437a

Contents

<code>as.character.filename</code>	2
<code>as.filename</code>	3
<code>filename</code>	3
<code>insert</code>	5
<code>is.filename</code>	6
<code>make_path</code>	7

set_fdate	8
set_fext	8
set_fpath	9
setftime	9
tag	10
trim_ext	11

Index	12
--------------	-----------

as.character.filename *Coerce a character to a filename*

Description

This function coerces a filename into a character.

Usage

```
## S3 method for class 'filename'
as.character(x, tag.char = NULL, simplify = FALSE, ...)
```

Arguments

x	a filename object
tag.char	character to delimit tags, defaults to '_'
simplify	if TRUE, all timestamps are omitted
...	unused arguments

Value

a character vector

Examples

```
x <- "data_post_2011-01-02.txt"
fn <- as.filename(x)
print(as.character(fn))
```

as.filename	<i>Coerce to a filename</i>
-------------	-----------------------------

Description

This function coerces an object into a filename, if possible.

Usage

```
as.filename(x, ...)  
  
## S3 method for class 'filename'  
as.filename(x, ...)  
  
## S3 method for class 'character'  
as.filename(x, tag.char = NULL, ...)
```

Arguments

x	a character or a filename
...	other arguments
tag.char	character to delimit tags, defaults to '_'

Value

a filename object

Examples

```
fn <- as.filename("data_raw_2011-01-01.txt")  
str(fn)
```

filename	<i>Create a filename.</i>
----------	---------------------------

Description

This function creates a filename object with a file path, tags, extensions, date stamp or date-time stamp.

Usage

```
filename(
  x,
  path = NULL,
  tag = NULL,
  ext = NULL,
  date = NULL,
  time = NULL,
  subdir = TRUE
)
```

Arguments

x	file name stem
path	path to the file
tag	tags for the file name
ext	file extension
date	date stamp (character or Date)
time	time stamp (character or POSIXct)
subdir	whether to append a date/time stamped subdirectory to path

Details

The date and time parameters can be specified as character vectors or date/time objects. If time is given as a POSIXct, it will override date. If these parameters are both NULL, automated date and time stamping may be done and is controlled by `getOption("filenamer.timestamp")`. If this option is NULL, 0, or less, no date or time stamping will be done; if it is 1, only date stamping will be done; if it is 2 or greater, date-time stamping will be done (default). Set date or time to NA to suppress date or time stamping for a particular filename. Stamps are shown in ISO 8601 date format ((platforms and are thus omitted; hyphens are omitted from date-time stamps for brevity.

By default, a date stamped subdirectory is appended to the file path. To disable this behaviour, set `subdir` to FALSE or disable path stamping globally by `options(filenamer.path.timestamp = 0)`. This option is similar to `filenamer.timestamp` above.

Value

a filename object

Examples

```
# file name is date-time stamped and put in subdirectory by default
fn <- filename("data", tag="qc", ext="txt")
print(as.character(fn))

# disable date-time stamping and subdirectory insertion
fn2 <- filename("data", tag="qc", date=NA, time=NA, subdir=FALSE)
print(as.character(fn2))
```

```
# creating a new file name from an existing one yields a new time stamp
fn3 <- filename(fn)
print(as.character(fn3))
```

insert

Insert tag or extension into a file name

Description

This function inserts a tag or extension into a file name. It can also replace an element of a file name.

Usage

```
insert(x, ...)

## S3 method for class 'filename'
insert(
  x,
  tag = NULL,
  tag.pos = NULL,
  ext = NULL,
  ext.pos = NULL,
  replace = FALSE,
  ...
)

## S3 method for class 'character'
insert(x, ...)
```

Arguments

x	file name (character or filename)
...	unused arguments
tag	one or more file name tags to insert
tag.pos	position at which to insert tag (NULL: append at the end or replace tag)
ext	one or more file extension tags to insert
ext.pos	position at which to insert extension (NULL: insert at penultimate position)
replace	if TRUE, tag or extension is replaced (default: replace last tag)

Details

By default, tags are inserted at the ultimate position and extensions at the penultimate position, if possible. (That is, the final file extension will not change, unless the insertion position is specified otherwise or the original file name had no extension.) If `replace` is TRUE, the tag at the indicated position is replaced by the new tag instead.

Value

modified object of the original type

Examples

```
f <- as.filename("data_expr_2014-05-01.tsv")

# new file name with inserted tags for saving normalized data
g <- insert(f, tag=c("mod", "norm"))
print(as.character(g))

# new file name with inserted extension for saving sorted data
h <- insert(f, ext="sorted")
print(as.character(h))

# new file name with different extension for saving in different format
i <- insert(f, ext="csv", replace=TRUE)
print(as.character(i))

# insert another tag
j <- insert(g, tag="qc", tag.pos=2)
print(as.character(j))
```

is.filename

Type checking for filename

Description

This function returns TRUE if its argument is a filename and FALSE otherwise.

Usage

```
is.filename(x)
```

Arguments

x object to check

Value

a logical value

make_path	<i>Create directory structure for a file path</i>
-----------	---

Description

This function creates directories recursively (as necessary) to the specified file.

Usage

```
make_path(x, ...)  
  
## S3 method for class 'filename'  
make_path(x, showWarnings = FALSE, recursive = TRUE, ...)  
  
## S3 method for class 'character'  
make_path(x, ...)
```

Arguments

x	file name (character or filename)
...	other arguments passed to dir.create
showWarnings	whether to show warnings
recursive	whether to recursively create all parent directories

Examples

```
## Not run:  
# CRAN policy forbids package example to write to current directory,  
# even inside \dontrun because the user may copy-and-paste and  
# pollute his/her current directory;  
# in real-world setting, the `tempdir` path prefix is unnecessary  
x <- file.path(tempdir(), "path/to/file.txt")  
  
fn <- as.filename(x)  
make_path(fn)  
  
## End(Not run)
```

set_fdate *Set date stamp in a file name*

Description

This function sets the date stamp in a file name.

Usage

```
set_fdate(x, date)
```

Arguments

x	a character or a filename
date	new date stamp (character or Date)

Value

modified object of the original type

Examples

```
x <- "data_norm_2011-01-03.txt"  
print(set_fdate(x, "2011-01-05"))
```

set_fext *Set file extension*

Description

This function sets the extension in a file name.

Usage

```
set_fext(x, ext, all)
```

Arguments

x	a character or a filename
ext	new file extension
all	replace the entire extension

Value

modified object of the original type

Examples

```
x <- "data_norm_2011-01-03.txt"
print(set_fext(x, "csv"))
```

set_fpath	<i>Set path in a file name</i>
-----------	--------------------------------

Description

This function sets the path in a file name.

Usage

```
set_fpath(x, path)
```

Arguments

x	a character or a filename
path	new path to file

Value

modified object of the original type

Examples

```
x <- "path/data_norm.txt"
print(set_fpath(x, "new_path"))
```

setftime	<i>Set time stamp in a file name</i>
----------	--------------------------------------

Description

This function sets the time stamp in a file name.

Usage

```
setftime(x, time)
```

Arguments

x	a character or a filename
time	new time stamp (character or POSIXct)

Value

modified object of the original type

Examples

```
x <- "data_norm_20110103T093015.txt"
# change the time to 30 seconds past 2:45 p.m.
print(set_ftime(x, "144530"))
# to change the date, time must be specified as well
print(set_ftime(x, "20110505T101500"))
```

tag

Insert tag or extension and coerce to character

Description

This function inserts a tag or extension into a file name and returns a character vector.

Usage

```
tag(x, ...)
```

Arguments

x a filename or character
... arguments passed to [insert](#)

Value

a character vector

Examples

```
x <- "data.txt"
y <- tag(x, "qc")
print(y)
f <- as.filename(x)
g <- tag(f, "qc")
print(g)
```

trim_ext	<i>Trim extensions from a file name</i>
----------	---

Description

This function trims extensions from a file name.

Usage

```
trim_ext(x, n)
```

Arguments

x	a character or a filename
n	number of extensions to trim off the end

Value

modified object of the original type

Examples

```
x <- "path/data.txt.gz"  
print(trim_ext(x))
```

Index

`as.character.filename`, 2

`as.filename`, 3

`dir.create`, 7

`filename`, 3

`insert`, 5, 10

`is.filename`, 6

`make_path`, 7

`set_fdate`, 8

`set_fext`, 8

`set_fpath`, 9

`setftime`, 9

`tag`, 10

`trim_ext`, 11