

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

October 29, 2024

Type Package

Title Make and Modify File Names and Metadata

Version 0.1.0.9002

Maintainer Eli Pousson <eli.pousson@gmail.com>

Description Work with filenames and paths and read and write file metadata.

License MIT + file LICENSE

URL <https://github.com/elipousson/filenamr>,
<https://elipousson.github.io/filenamr/>

BugReports <https://github.com/elipousson/filenamr/issues>

Depends R (>= 2.10)

Imports cli, cliExtras (>= 0.1.0), glue, rappdirs, rlang (>= 1.1.0),
utils

Suggests covr, dplyr, exiftoolr, janitor, lubridate, testthat (>= 3.0.0), tibble, withr

Remotes elipousson/cliExtras, JoshOBrien/exiftoolr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

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

RemoteUrl <https://github.com/elipousson/filenamr>

RemoteRef HEAD

RemoteSha 00a9c304f8a8662cbdacf73ac6689e1adc4b8c9

Contents

check_file_overwrite	2
check_path_fileext	3
default_exif_tags	3
default_exif_xwalk	4
fmt_exif_direction	4
get_data_dir	5
get_path_fileext	6
list_pkg_data	7
make_filename	8
read_exif	10
set_file_path	12
str_affix	13

Index	15
--------------	-----------

check_file_overwrite	<i>Check if a file exists and remove file or error</i>
----------------------	--

Description

The filename or path must include a single file extension.

Usage

```
check_file_overwrite(
  filename = NULL,
  path = NULL,
  overwrite = TRUE,
  quiet = FALSE,
  ask = TRUE,
  .envir = caller_env(),
  call = caller_env()
)
```

Arguments

filename	File name, Default: NULL. Optional if path is supplied.
path	File path, Default: NULL. Optional if filename is supplied.
overwrite	If TRUE, remove a file with the same name and path
quiet	If TRUE, suppress informational messages, Default: FALSE
ask	If TRUE, overwrite is FALSE, and session is interactive, ask if user wants to overwrite the file. Default: TRUE
.envir	Ignored at present.
call	The execution environment of a currently running function, e.g. caller_env(). The function will be mentioned in error messages as the source of the error. See the call argument of abort() for more information.

check_path_fileext *Check if a file path has a file extension*

Description

`check_path_fileext()` checks if a character vector of file paths have a file extension or a specified file extension. Errors if any elements of do not pass the condition.

Usage

```
check_path_fileext(
  path,
  fileext = NULL,
  message = "{.arg {arg}} must have a file extension.",
  arg = caller_arg(path),
  call = caller_env()
)
```

Arguments

path	Character vector with file path or paths to check. Required.
fileext	Optional file extension string. If NULL, path must have a file extension. If fileext is a character string, all elements of path must have a matching file extension.
message	It is formatted via a call to <code>cli_bullets()</code> .
arg	An argument name as a string. This argument will be mentioned in error messages as the input that is at the origin of a problem.
call	The execution environment of a currently running function, e.g. <code>call = caller_env()</code> . The corresponding function call is retrieved and mentioned in error messages as the source of the error. You only need to supply call when throwing a condition from a helper function which wouldn't be relevant to mention in the message. Can also be NULL or a defused function call to respectively not display any call or hard-code a code to display. For more information about error calls, see Including function calls in error messages .

default_exif_tags *Default EXIF, XMP-dc, and IPTC tags*

Description

A vector of default EXIF, XMP-dc, and IPTC tags (including wildcard EXIF tag "GPS") used by `read_exif()`.

Usage

```
default_exif_tags
```

Format

A length 26 character vector.

```
default_exif_xwalk    EXIF data column name crosswalk
```

Description

A named vector with a crosswalk of default column names returned by `exiftoolr::exif_read()` and the replacement values used as new names.

Usage

```
default_exif_xwalk
```

Format

A length 29 named character vector.

```
fmt_exif_direction    Format a data.frame column as a cardinal direction in degrees and wind direction
```

Description

Format a data.frame column as a cardinal direction in degrees and wind direction

Usage

```
fmt_exif_direction(  
  data,  
  winds = 8,  
  .after = "img_direction",  
  call = caller_env()  
)
```

Arguments

data	A data.frame or character vector to format with <code>isstatic::as_cardinal_bearing()</code> . If data is a data.frame, the photo direction is assumed to be in a column named "img_direction" and the new cardinal bearing is added to a column named "img_cardinal_wind".
winds	Number of winds to use for results (4, 8, or 16).
.after	Column name passed to .after parameter of <code>dplyr::mutate()</code> .
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.

See Also

`isstatic::as_cardinal_bearing()`

get_data_dir

Check if data directory exists and create a new directory if needed

Description

Get the path for a package-specific cache directory with `rappdirs::user_cache_dir()`, check for the existence of a data directory, optionally create a new directory at the provided path location.

Usage

```
get_data_dir(
  path = NULL,
  cache = FALSE,
  create = TRUE,
  ask = TRUE,
  appname = NULL,
  pkg = NULL,
  allow_null = TRUE,
  quiet = FALSE,
  call = caller_env()
)

list_path_filenames(
  path,
  fileext = NULL,
  pattern = NULL,
  full.names = TRUE,
  call = caller_env(),
  ...
)
```

Arguments

path	Path to directory for use as data directory.
cache	If TRUE, and path is NULL set path to <code>rappdirs::user_cache_dir()</code> (using value of pkg as appname). If path is not NULL, the path is returned even if cache is TRUE.
create	If FALSE and path does not exist, return path with a warning. If TRUE and <code>rlang::is_interactive()</code> is TRUE, ask user if directory should be created. If the session not interactive and create is TRUE, a new directory will be created.
ask	If TRUE, create is FALSE, and session is interactive, ask to create directory if the provided directory does not exist.
appname, pkg	pkg is used if appname is NULL. Passed to <code>rappdirs::user_cache_dir()</code>
allow_null	If TRUE, path is NULL, cache is FALSE, return the NULL path value; defaults to TRUE.
quiet	If TRUE, suppress informational messages.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.
fileext	File extension. If supplied to <code>list_path_filenames()</code> and pattern is NULL, only return file names matching this extension.
pattern	an optional regular expression . Only file names which match the regular expression will be returned.
full.names	a logical value. If TRUE, the directory path is prepended to the file names to give a relative file path. If FALSE, the file names (rather than paths) are returned.
...	Additional parameters passed to <code>list.files()</code> by <code>list_path_filenames()</code> .

get_path_fileext

Get file extensions for files at a path

Description

If fileext is provided, `get_path_fileext()` will pass the file extension forward without checking it. `list_path_fileext()` is a more basic function that list file extensions at a path directory.

Usage

```
get_path_fileext(
  path,
  fileext = NULL,
  n = 1,
  quiet = FALSE,
  call = caller_env()
)
```

```
list_path_fileext(path, allow_null = FALSE, call = caller_env(), ...)
```

Arguments

path	A single directory or file path. The directory or file must exist.
fileext	If fileext is supplied, the function returns the file extension as is. If NULL (default), one or more file extensions are extracted from files at the path location.
n	Max number of unique file types to return. Returns warning and n most common file types if path has more than n unique file types.
quiet	If TRUE, suppress informational messages.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.
allow_null	If TRUE, <code>list_path_fileext()</code> returns NULL if path is NULL or if no files exist at the path location. If FALSE (default), abort if either condition is met.
...	Additional parameters passed by <code>list_path_fileext()</code> to <code>list.files()</code> .

list_pkg_data	<i>List package datasets, extdata files, and files in package cache</i>
---------------	---

Description

`list_pkg_data()` calls three helper functions and returns a data.frame. `list_pkg_datasets()` lists package datasets (using the library path as the path), `list_pkg_extdata()` lists extdata files for a package (using basename without a file extension as the item name), `list_pkg_cachedata()` lists data in the package cache folder returned by `rappdirs::user_cache_dir()`.

Usage

```
list_pkg_data(
  pkg = NULL,
  version = NULL,
  dir = "extdata",
  lib.loc = NULL,
  call = caller_env()
)

list_pkg_datasets(pkg, lib.loc = NULL)

list_pkg_extdata(pkg, dir = "extdata", full.names = TRUE, recursive = TRUE)

list_pkg_cachedata(pkg, full.names = TRUE, recursive = TRUE)
```

Arguments

pkg	The package names. Can include version requirements, e.g. "pkg (>= 1.0.0)".
version	Minimum versions for pkg. If supplied, must be the same length as pkg. NA elements stand for any versions.

dir	One or more directory names to pass as the first argument of <code>system.file()</code> . Defaults to "extdir".
lib.loc	a character vector with path names of R libraries. See ‘Details’ for the meaning of the default value of NULL.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.
full.names	a logical value. If TRUE, the directory path is prepended to the file names to give a relative file path. If FALSE, the file names (rather than paths) are returned.
recursive	logical. Should the listing recurse into directories?

make_filename	<i>Make file name and path with optional label, prefix, or postfix</i>
---------------	--

Description

A helper function to create consistent file names for plots created with data files.

Usage

```
make_filename(
  name = NULL,
  label = NULL,
  fileext = NULL,
  filename = NULL,
  path = NULL,
  prefix = NULL,
  postfix = NULL,
  pad = NULL,
  width = NULL,
  cache = FALSE,
  appname = NULL,
  pkg = NULL,
  create = TRUE,
  increment = NULL,
  call = caller_env()
)
```

Arguments

name	Name to make file name converted to snake case with <code>janitor::make_clean_names()</code> , e.g. "Residential zoning map" becomes "residential_zoning_map". If the name includes a file extension it is assumed that the filename has been provided as the name parameter.
------	---

label	Label to combine with name converted to snake case with <code>janitor::make_clean_names()</code> . The label is designed to identify the area or other shared characteristics across multiple data files, maps, or plots. label is ignored if name is NULL or if name includes a file extension.
fileext	File type or extension. Optional if filename or path include a file extension.
filename	File name; if filename is NULL and path does not include a file extension, name and file extension are both required.
path	Path to file or data directory. Optional. If path includes a file extension and filename and fileext are both NULL, the filename and extension included with path will be used instead. If multiple file extensions are provided to filename, path, or fileext, <code>make_filename()</code> will abort.
prefix	File name prefix. "date" adds a date prefix, "time" adds a date/time prefix; defaults to NULL.
postfix	File name postfix; defaults to NULL.
pad	Single padding character added to digits in string; defaults to "0"
width	Minimum width of padded strings.
cache	If TRUE, path is set to the package cache directory using <code>get_data_dir()</code> ; defaults to FALSE.
appname, pkg	pkg is used if appname is NULL. Passed to <code>rappdirs::user_cache_dir()</code>
create	If FALSE and path does not exist, return path with a warning. If TRUE and <code>rlang::is_interactive()</code> is TRUE, ask user if directory should be created. If the session not interactive and create is TRUE, a new directory will be created.
increment	If TRUE, increment digits in string by 1. If numeric, increment digits in string by value. If NULL, 0, or if no digits are present in string, return string as is.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the call argument of <code>abort()</code> for more information.

Examples

```
make_filename(
  filename = "image.jpeg"
)
```

```
make_filename(
  name = "plot",
  label = "Group a",
  fileext = "png"
)
```

```
make_filename(
  name = "plot",
  prefix = "date",
  fileext = "png"
)
```

```

make_filename(
  name = "map_1",
  increment = TRUE,
  fileext = "geojson"
)

```

read_exif	<i>Read EXIF metadata to create a simple feature object or write EXIF metadata to image files</i>
-----------	---

Description

`read_exif()` read EXIF data from folder of files. Optionally assigns a cardinal direction based on the direction metadata and recodes the orientation metadata. Note that tags must include GPS tags if you plan to create an sf object based on the resulting data.frame object.

For `write_exif()` the parameters are used to multiple tags with the same values:

- title: Title, IPTC:Headline, IPTC:ObjectName, XMP-dc:Title
- description: ImageDescription, XMP-dc:Description, and IPTC:Caption-Abstract
- keywords: Keywords, IPTC:Keywords, XMP-dc:Subject

Usage

```

read_exif(
  path = NULL,
  fileext = NULL,
  tags = NULL,
  format_exif = TRUE,
  xwalk = NULL,
  tz = NULL,
  .name_repair = "check_unique",
  ...
)

```

```

write_exif(
  path,
  fileext = NULL,
  title = NULL,
  author = NULL,
  credit = author,
  date = NULL,
  keywords = NULL,
  description = NULL,
  alt = NULL,
  metadata = NULL,
  args = NULL,
)

```

```

    overwrite = TRUE,
    append_keywords = FALSE,
    quiet = FALSE,
    call = caller_env()
)

```

Arguments

path	A path to folder or file.
fileext	The file extension or file type; defaults to NULL.
tags	List of EXIF tags to read from files. If NULL (default), set to option "filenamr.exif_tags" or default default_exif_tags.
format_exif	If TRUE (default), rename columns based on xwalk values, add cardinal directions based on bearing, and format date columns.
xwalk	If NULL, set to option "filenamr.exif_xwalk" or default default_exif_xwalk.
tz	Time zone to pass to <code>lubridate::ymd_hms()</code> if format_exif is TRUE. Typically set to <code>Sys.timezone()</code> to convert date/time columns.
.name_repair	Treatment of problematic column names: <ul style="list-style-type: none"> • "minimal": No name repair or checks, beyond basic existence, • "unique": Make sure names are unique and not empty, • "check_unique": (default value), no name repair, but check they are unique, • "universal": Make the names unique and syntactic • a function: apply custom name repair (e.g., <code>.name_repair = make.names</code> for names in the style of base R). • A purrr-style anonymous function, see <code>rlang::as_function()</code> <p>This argument is passed on as repair to <code>vctrs::vec_as_names()</code>. See there for more details on these terms and the strategies used to enforce them.</p>
...	Additional parameters to pass to <code>exiftoolr::exif_read()</code>
title	Title to add to file metadata with exiftoolr, Default: NULL.
author	Author to add to file metadata to the "Author" and "XMP-dc:creator" tags. Default: NULL.
credit	Credit to add to file metadata to the "IPTC:Credit" and "XMP-dc:Credit" tags. Defaults to the same value as author.
date	Date to add to file metadata with exiftoolr (not currently working). Defaults to NULL.
keywords	Keyword(s) added to file metadata to "IPTC:Keywords" and "XMP-dc:Subject" tags. Defaults to NULL.
description	Description added to the "ImageDescription", "IPTC:Caption-Abstract", and "XMP-dc:Description" tags.
alt	Text to pass as alt text to the "IPTC:AltTextAccessibility" and "iTXt" (PNG files only) tags. Defaults to NULL.
metadata	Any of the other metadata parameters (title, author, credit, date, keywords, description, and alt) can also be set by passing a named list or data.frame to metadata. If an argument is supplied, any conflicting value in metadata is ignored.

args	Alternate arguments passed to <code>exiftoolr::exif_call()</code> . Other tag parameters are appended to args if they are not NULL.
overwrite	If TRUE, overwrite any existing EXIF metadata present in the provided fields; defaults to TRUE
append_keywords	If TRUE, append keywords, if FALSE, replace keywords in file metadata.
quiet	If TRUE (default), suppress function messages.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

Value

A tibble of EXIF and other metadata from files located in the path directory.

set_file_path	<i>Set a file path and validate path file extension</i>
---------------	---

Description

This is a flexible wrapper for `isstatic::file_path()` that allows you to provide a file path as a filename or path parameter. If the path contains a file extension and the `fileext` parameter is provided, the function aborts if the two file extensions do not match.

Usage

```
set_file_path(
  filename = NULL,
  path = NULL,
  fileext = NULL,
  allow_null = FALSE,
  call = caller_env()
)
```

Arguments

filename	File name. Optional if path is supplied.
path	File path. Optional if filename is supplied.
fileext	File extension. If the path supplied using filename and path <i>does not</i> end with a file extension, fileext is used as the file extension for the returned path. If the path, <i>does</i> end with a file extension, and fileext is used to validate the supplied.
allow_null	If TRUE, return NULL if filename and path are NULL. If FALSE, error if filename and path are both NULL.
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.

str_affix

*Apply a prefix or postfix to a string***Description**

Prefix and postfix can include more than one value that are added in the same order provided. For [str_affix\(\)](#), the string must be a single character string.

Usage

```
str_affix(
  string = NULL,
  prefix = NULL,
  postfix = NULL,
  sep = "_",
  pad = NULL,
  width = NULL,
  use_clean_names = TRUE,
  case = "snake",
  replace = c("`" = "", "`" = "", "%`" = "_pct_", "#`" = "_num_"),
  use_make_names = TRUE,
  call = caller_env(),
  ...
)

str_prefix(
  string = NULL,
  prefix = NULL,
  sep = "_",
  is_postfix = FALSE,
  date.format = "%Y-%m-%d",
  time.format = "%Y-%m-%d_%I-%M-%S%p",
  use_clean_names = TRUE,
  case = "snake",
  replace = c("`" = "", "`" = "", "%`" = "_pct_", "#`" = "_num_"),
  use_make_names = TRUE,
  ...
)
```

Arguments

string	A single string that the attach prefix or postfix is added to.
prefix	Character string or character vector to add to string parameter as a prefix.
postfix	Character string or character vector to add to string parameter as a postfix.
sep	Separator character passed as the collapse parameter of paste() .
pad	Single padding character added to digits in string; defaults to "0"

width	Minimum width of padded strings.
use_clean_names	If TRUE, prefix, postfix, and string are all converted to snake case with <code>janitor::make_clean_names()</code> .
case	The desired target case (default is "snake") will be passed to <code>snakecase::to_any_case()</code> with the exception of "old_janitor", which exists only to support legacy code (it preserves the behavior of <code>clean_names()</code> prior to addition of the "case" argument (janitor versions $\leq 0.3.1$). "old_janitor" is not intended for new code. See to_any_case for a wide variety of supported cases, including "sentence" and "title" case.
replace	A named character vector where the name is replaced by the value.
use_make_names	Should <code>make.names()</code> be applied to ensure that the output is usable as a name without quoting? (Avoiding <code>make.names()</code> ensures that the output is locale-independent but quoting may be required.)
call	The execution environment of a currently running function, e.g. <code>caller_env()</code> . The function will be mentioned in error messages as the source of the error. See the <code>call</code> argument of <code>abort()</code> for more information.
...	Additional parameters passed to <code>janitor::make_clean_names()</code> if <code>use_clean_names</code> is TRUE.
is_postfix	If TRUE, use the prefix string as a postfix; defaults to FALSE.
date.format, time.format	Date or time format. Only used by <code>str_prefix</code> if prefix is "date" or "time" and not currently accessible when using <code>str_affix()</code> or <code>make_filename()</code> .

Details

- `str_affix()`: Add a label, prefix, and postfix to string
- `str_prefix()`: Add a prefix or a postfix to a string

Index

- * **datasets**
 - default_exif_tags, 3
 - default_exif_xwalk, 4
- * **read_write**
 - make_filename, 8
- abort(), 2, 5–9, 12, 14
- check_file_overwrite, 2
- check_path_fileext, 3
- check_path_fileext(), 3
- cli_bullets(), 3
- default_exif_tags, 3
- default_exif_xwalk, 4
- defused function call, 3
- dplyr::mutate(), 5
- exiftoolr::exif_call(), 12
- exiftoolr::exif_read(), 4, 11
- fmt_exif_direction, 4
- get_data_dir, 5
- get_data_dir(), 9
- get_path_fileext, 6
- get_path_fileext(), 6
- Including function calls in error
 - messages, 3
- isstatic::as_cardinal_bearing(), 5
- isstatic::file_path(), 12
- janitor::make_clean_names(), 8, 9, 14
- list.files(), 6, 7
- list_path_fileext (get_path_fileext), 6
- list_path_fileext(), 6, 7
- list_path_filenames (get_data_dir), 5
- list_path_filenames(), 6
- list_pkg_cachedata (list_pkg_data), 7
- list_pkg_cachedata(), 7
- list_pkg_data, 7
- list_pkg_data(), 7
- list_pkg_datasets (list_pkg_data), 7
- list_pkg_datasets(), 7
- list_pkg_extdata (list_pkg_data), 7
- list_pkg_extdata(), 7
- lubridate::ymd_hms(), 11
- make_filename, 8
- make_filename(), 14
- paste(), 13
- rappdirs::user_cache_dir(), 5–7, 9
- read_exif, 10
- read_exif(), 3, 10
- regular expression, 6
- rlang::as_function(), 11
- rlang::is_interactive(), 6, 9
- set_file_path, 12
- str_affix, 13
- str_affix(), 13, 14
- str_prefix, 14
- str_prefix (str_affix), 13
- str_prefix(), 14
- system.file(), 8
- to_any_case, 14
- vctrs::vec_as_names(), 11
- write_exif (read_exif), 10
- write_exif(), 10