Package: exiftoolr (via r-universe)

October 28, 2026

Type Package

Title ExifTool Functionality from R

Version 0.2.10

Date 2026-10-28

Maintainer Joshua O'Brien <joshmobrien@gmail.com>

Description Reads, writes, and edits EXIF and other file metadata using ExifTool <https://exiftool.org/>, returning read results as a data frame. ExifTool supports many different metadata formats including EXIF, GPS, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP and ID3, Lyrics3, as well as the maker notes of many digital cameras by Apple, Canon, Casio, DJI, FLIR, FujiFilm, GE, Google, GoPro, HP, JVC/Victor, Kodak, Leaf, Minolta/Konica-Minolta, Motorola, Nikon, Nintendo, Olympus/Epson, Panasonic/Leica, Pentax/Asahi, Phase One, Reconyx, Ricoh, Samsung, Sanyo, Sigma/Foveon and Sony.

License GPL-2

URL https://github.com/JoshOBrien/exiftoolr#readme, https://joshobrien.github.io/exiftoolr/

BugReports https://github.com/JoshOBrien/exiftoolr/issues

SystemRequirements Perl

Depends R (>= 3.0.0)

Imports backports, curl, jsonlite, zip, data.table

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Joshua O'Brien [aut, cre]

Config/pak/sysreqs perl

Repository https://cran.r-universe.dev

Date/Publication 2026-10-28 21:45:09 UTC

RemoteUrl https://github.com/cran/exiftoolr

RemoteRef HEAD

RemoteSha a32829269813b2244526faca5c17a925295abdd8

Contents

configure_exiftoolr 2
exif_call 3
exif_read 4
install_exiftool 6

Index **8**

`configure_exiftoolr` *Configure package to point to ExifTool and/or Perl*

Description

Configure package to point to ExifTool and/or Perl

Usage

```
configure_exiftoolr(
  command = NULL,
  perl_path = NULL,
  allow_win_exe = TRUE,
  quiet = FALSE
)
```

Arguments

`command`	Character string giving the exiftool command.
`perl_path`	Path to a Perl executable.
`allow_win_exe`	Logical. If running on a Windows machine, and if a standalone exiftool executable is available, should it be used?
`quiet`	Logical. Should function should be chatty?

Value

A character string giving the exiftool command, returned invisibly.

exif_call *Call ExifTool from R*

Description

Uses `system2()` to run a basic call to `exiftool`.

Usage

```
exif_call(
  args = NULL,
  path = NULL,
  stdout = TRUE,
  quiet = FALSE,
  ...,
  config_file = NULL,
  common_args = NULL
)

exif_version(quiet = TRUE)
```

Arguments

`args`	Character vector of arguments, each written in same form as you would if writing them on the command line (e.g. `"-n"` or `"-csv"`)
`path`	A character vector giving one or more file paths.
`stdout`	Where output to stdout should be sent. If `TRUE` (the default), the output is captured in a character vector. For other options, see the help file for `system2`, the function to which this argument's value gets passed along.
`quiet`	Use `FALSE` to display diagnostic information. Default value is `FALSE`.
`...`	Additional arguments to be passed to `system2()`.
`config_file`	Path to a config file of the format expected by Exiftool's command line `-config` option. (See Details for an explanation of why this one option cannot be passed directly to `args` via the `-config` argument.)
`common_args`	A character vector of arguments to be applied to all executed commands when the Exiftool `-execute` option is being used. (See Details for an explanation of why this option cannot be passed directly to `args` via `-common_args` argument.)

Details

For examples of the command-line calls to ExifTool (all of which can be reproduced by calls to `exif_call`), see https://exiftool.org/examples.html.

Under the hood, `exif_call()` writes the options in `args` to a text file and then calls Exiftool, passing that text file's contents to Exiftool via its `-@ ARGFILE` option. `-config` and `-common_args` are the two options that may not be used in such a `-@ ARGFILE`, so we handle that option separately using `exif_call()`'s `config_file` argument.

Value

The standard output as a character vector.

Examples

```
## Not run:
## Find local ExifTool version using exif_version() or exif_call()
exif_version()
exif_call(args = "-ver")

## Make temporary copies of a couple jpeg files
tmpdir <- tempdir()
src_files <- dir(system.file(package = "exiftoolr", "images"),
                 full.names = TRUE)
files <- file.path(tmpdir, basename(src_files))
file.copy(src_files, files)

## Both of the following extract the same tags:
exif_read(files, tags = c("filename", "imagesize"))
exif_call(args = c("-n", "-j", "-q", "-filename", "-imagesize"),
          path = files)

## Set value of a new "Artist" field in photo's metadata
file1 <- files[1]
exif_read(file1, tags = "artist")
exif_call(path = file1, args = "-Artist=me")
exif_read(file1, tags = "artist")

## Remove all but a few essential fields
length(exif_read(file1))
exif_call(path = file1, args = "-all=")
length(exif_read(file1))
exif_read(file1)

## Clean up
unlink(files)

## End(Not run)
```

`exif_read` *Read EXIF and other metadata from files*

Description

Reads EXIF and other metadata into a `data.frame` by calling Phil Harvey's ExifTool command-line application.

Usage

```
exif_read(
  path,
  tags = NULL,
  recursive = FALSE,
  args = NULL,
  quiet = TRUE,
  pipeline = c("json", "csv")
)
```

Arguments

path	A vector of filenames.
tags	A vector of tags to output. It is a good idea to specify this when reading large numbers of files, as it decreases the output overhead significantly. Spaces will be stripped in the output data frame. This parameter is not case-sensitive.
recursive	TRUE to pass the "-r" option to ExifTool.
args	Additional arguments.
quiet	Use FALSE to display diagnostic information. Default value is TRUE
pipeline	One of "json" (the default) or "csv". Controls whether the exiftool executable, behind the scenes, extracts metadata into a JSON data structure or a tabular csv. The JSON pipeline works well in most cases, but (as documented at https://exiftool.org/exiftool_pod.html) does not properly handle non-UTF-8 character sets. If the metadata fields include characters that are not encoded using UTF-8 and that need to be handled by setting the "-charset" option, use the "csv" pipeline as demonstrated in the second example below.

Details

From the ExifTool website: "ExifTool is a platform-independent Perl library plus a command-line application for reading, writing and editing meta information in a wide variety of files. ExifTool supports many different metadata formats including EXIF, GPS, IPTC, XMP, JFIF, GeoTIFF, ICC Profile, Photoshop IRB, FlashPix, AFCP and ID3, as well as the maker notes of many digital cameras by Canon, Casio, DJI, FLIR, FujiFilm, GE, GoPro, HP, JVC/Victor, Kodak, Leaf, Minolta/Konica-Minolta, Motorola, Nikon, Nintendo, Olympus/Epson, Panasonic/Leica, Pentax/Asahi, Phase One, Reconyx, Ricoh, Samsung, Sanyo, Sigma/Foveon and Sony."

For more information, see the ExifTool website.

Value

A data frame of class "exiftoolr" with one row per file processed. The first column, named "SourceFile" gives the name(s) of the processed files. Subsequent columns contain info from the tags read from those files.

Note that binary tags such as thumbnails are loaded as base64-encoded strings that start with "base64:". Although these are truncated in the printed representation of the data.frame returned by the function, they are left unaltered in the data.frame itself.

References

https://exiftool.org

Examples

```
## Not run:
files <- dir(system.file(package = "exiftoolr", "images"),
             pattern = "LaSals|Lizard", full.names = TRUE)
exif_read(files)
exif_read(files, tags = c("filename", "imagesize"))

## Use pipeline="csv" for images needing explicit specification
## and proper handling of a non-default character sets
img_file <- system.file(package = "exiftoolr", "images", "QS_Hongg.jpg")
args <- c("-charset", "exiftool=cp1250")
res <- exif_read(img_file, args = args, pipeline = "csv")
res[["City"]]  ## "Zurich", with an umlaut over the "u"

## End(Not run)
```

---

install_exiftool *Install ExifTool, downloading (by default) the current version*

Description

Install the current version of ExifTool

Usage

```
install_exiftool(
  install_location = NULL,
  win_exe = NULL,
  local_exiftool = NULL,
  quiet = FALSE
)
```

Arguments

install_location	Path to the directory into which ExifTool should be installed. If NULL (the default), installation will be into the directory returned by backports:::R_user_dir("exiftoolr").
win_exe	Logical, only used on Windows machines. Should we install the standalone ExifTool Windows executable or the ExifTool Perl library? (The latter relies, for its execution, on an existing installation of Perl being present on the user's machine.) If set to NULL (the default), the function installs the Windows executable on Windows machines and the Perl library on other operating systems.

`local_exiftool`	If installing ExifTool from a local "*.zip" or ".tar.gz", supply the path to that file as a character string. With default value, 'NULL', the function downloads ExifTool from `https://exiftool.org` and then installs it.
`quiet`	Logical. Should function should be chatty?

Value

Called for its side effect

Index

configure_exiftoolr, 2

exif_call, 3
exif_read, 4
exif_version *(*exif_call*)*, 3

install_exiftool, 6

system2, *3*