

Package ‘screenshot’

August 22, 2026

Type Package

Title Take Screenshots (Screen Capture) from R Command

Version 0.9.3

Description Take screenshots from R command and locate an image position.

URL <https://github.com/matutosi/screenshot>,
<https://matutosi.github.io/screenshot/>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, fs, imager, magick, purrr, rlang, stringr, tibble,
utils

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

Depends R (>= 4.1.0)

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Toshikazu Matsumura [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-2306-6355>>)

Maintainer Toshikazu Matsumura <matutosi@gmail.com>

Repository CRAN

Date/Publication 2026-08-22 09:20:02 UTC

Contents

bitmap2png	2
clipboard2bitmap	3
clipboard_sample	3
compare_table	4

count_val_freq	4
create_header	5
crop_image	5
display_corner	6
display_size	7
get_clipboard_image	7
get_os	8
hex2little_endian	8
image2gray_matrix	9
index2xy	9
install_screenshot	10
is_all_same	11
locate_image	11
locate_ndl_in_hay	13
reset_transparent	14
save_bmp	14
save_clipboard_image	15
screenshot	16
screenshot_exists	17
xy_pos	17

Index	19
--------------	-----------

bitmap2png	<i>Converts a bitmap image to PNG using</i>
------------	---

Description

Converts a bitmap image to PNG using

Usage

bitmap2png(path)

Arguments

path Path to the bitmap image.

Value

The result of the system call.

Examples

```
## Not run:
bitmap2png("path/to/image.bmp")

## End(Not run)
```

clipboard2bitmap	<i>Save clipboard image to temporary BMP file</i>
------------------	---

Description

This function works only on windows. This function saves the image currently in the clipboard to a temporary BMP file.

Usage

```
clipboard2bitmap()
```

Value

Path to the temporary BMP file.

Examples

```
## Not run:  
clipboard2bitmap()  
  
## End(Not run)
```

clipboard_sample	<i>Sample data of clipboard bitmap array</i>
------------------	--

Description

Sample data of clipboard bitmap array

Usage

```
clipboard_sample
```

Format

raw data of bitmap array with 246292 length

Examples

```
data(clipboard_sample)
```

compare_table	<i>Compare values within tow arrays or matrices. Helper function for locate_ndl_in_hay().</i>
---------------	---

Description

Compare values within tow arrays or matrices. Helper function for locate_ndl_in_hay().

Usage

```
compare_table(ndl_mt, hay_mt)
```

Arguments

ndl_mt, hay_mt A matrix.

Value

A tibble.

Examples

```
val <- seq(from = 0, to = 1, by = 0.1)
mt_1 <- matrix(sample(val, 20, replace = TRUE))
mt_2 <- matrix(sample(val, 100, replace = TRUE))
compare_table(mt_1, mt_2)
```

count_val_freq	<i>Helper function for compare_table().</i>
----------------	---

Description

Helper function for compare_table().

Usage

```
count_val_freq(mt, colname)
```

Arguments

mt A numeric matrix or array.
colname A string of name for count.

Value

A dataframe.

Examples

```
mt <- sample(1:10, 30, replace = TRUE)
count_val_freq(mt, "freq")
```

create_header	Create a BMP header
---------------	---------------------

Description

Create a BMP header

Usage

```
create_header(clipboard)
```

Arguments

clipboard A raw vector of the clipboard contents.

Value

A raw vector of the BMP header.

Examples

```
data(clipboard_sample)
create_header(clipboard_sample)
```

crop_image	Cut off a part of image from a whole image.
------------	---

Description

Cut off a part of image from a whole image.

Usage

```
crop_image(image, pos_x, pos_y, w = 50, h = 20)
```

```
hay2needle(image, pos_x, pos_y, w = 50, h = 20)
```

Arguments

image	An image of cimg.
pos_x, pos_y	A numeric to indicate the top left corner of cutting image. When NULL, position will be randomly sampled.
w, h	A numeric for width or height of the cutting image.

Value

An image of cimg object.

Examples

```
image <- imager::load.example("parrots")
cropped_image <- crop_image(image, 200, 250, 100, 50)
layout(c(1:2))
plot(image)
plot(cropped_image)
```

display_corner	<i>Get display corner of screen</i>
----------------	-------------------------------------

Description

This function returns the coordinates of the specified corner of the display. This function works only on windows.

Usage

```
display_corner(size, corner = "bottom_left", width = 600, height = 600)
```

Arguments

size	Integers to specify width or height of display size.
corner	A string to specify a corner of the display. "top_left", "top_right", "bottom_left", or "bottom_right".
width, height	A integer to specify width or height of the corner.

Value

A numeric vector of length 4 representing the coordinates of the specified corner.

Examples

```
## Not run:
size <- display_size()
display_corner(size, "top_left", 800, 800)

## End(Not run)
```

display_size	<i>Get the size of the display.</i>
--------------	-------------------------------------

Description

This function works only on windows.

Usage

```
display_size()
```

Value

A list with two elements, width and height, which are the width and height of the display.

Examples

```
## Not run:
display_size()

## End(Not run)
```

get_clipboard_image	<i>Retrieves the image from the clipboard</i>
---------------------	---

Description

This function works only on windows.

Usage

```
get_clipboard_image()
```

Value

A raw vector containing the image data.

Examples

```
## Not run:
get_clipboard_image()

## End(Not run)

data(clipboard_sample)
head(clipboard_sample, 100)
header <- create_header(clipboard_sample)
image_data <- c(header, clipboard_sample)
path <- fs::path_temp(ext = "bmp")
save_bmp(image_data, path)
# shell.exec(path)
fs::file_delete(path)
```

get_os	<i>Get OS name</i>
--------	--------------------

Description

Get OS name

Usage

```
get_os()
```

Value

A string of OS name

Examples

```
get_os()
```

hex2little_endian	<i>Convert hexadecimal string to little-endian</i>
-------------------	--

Description

Convert hexadecimal string to little-endian

Usage

```
hex2little_endian(x)
```

Arguments

x Hexadecimal string

Value

Little-endian hexadecimal string

Examples

```
hex2little_endian("01234567")
hex2little_endian("012345")
```

image2gray_matrix	<i>Convert cimg class into grayscale xy matrix. Helper function for locate_image(). Use grayscale to Speed up and to simplify code.</i>
-------------------	---

Description

Convert cimg class into grayscale xy matrix. Helper function for locate_image(). Use grayscale to Speed up and to simplify code.

Usage

```
image2gray_matrix(img)
```

Arguments

img A cimg object.

Value

An xy dimensional matrix.

index2xy	<i>Convert array index into xy location in matrix. Helper function for locate_ndl_in_hay().</i>
----------	---

Description

Convert array index into xy location in matrix. Helper function for locate_ndl_in_hay().

Usage

```
index2xy(index, nrow)
```

Arguments

index, nrow A numeric.

Value

A numeric pair of xy location.

Examples

```
nrow <- 4
matrix(1:12, nrow = nrow)
purrr::map(1:12, index2xy, nrow = nrow)
```

install_screenshot	<i>Install command line screenshot for Windows.</i>
--------------------	---

Description

Codes are from URL shown below. <https://superuser.com/questions/75614/take-a-screen-shot-from-command-line-in-windows#answer-1751844> On Mac screencapture is usually available. On Linux GNOME desktop use gnome-screenshot. If not installed, run `sudo apt install gnome-screenshot`.

Usage

```
install_screenshot(bin_dir = "")
```

Arguments

bin_dir A string of directory to be installed.

Value

A string of installed directory.

Examples

```
if(interactive()){
  # need only on Win
  if(get_os() == "win"){
    bin_dir <- fs::path_package("screenshot")
    # if you want to install another directory
    # bin_dir <- "SET_YOUR_DIRECTORY"
    install_screenshot(bin_dir)
  }
}
```

is_all_same	<i>Helper function for locate_ndl_in_hay().</i>
-------------	---

Description

Helper function for locate_ndl_in_hay().

Usage

```
is_all_same(ndl_mt, hay_mt, base_xy, tol_val = 0)
```

Arguments

ndl_mt, hay_mt	A matrix
base_xy	A numeric pair of xy location.
tol_val	A numeric of the tolerance of the comparison.

Value

A logical.

locate_image	<i>Locate needle image position on a screenshot image.</i>
--------------	--

Description

Locate needle image position on a screenshot image.

Usage

```
locate_image(
  needle_image,
  center = TRUE,
  exact = TRUE,
  timeout = 5,
  corner = NULL,
  width = 600,
  height = 300,
  size = NULL,
  scale = NULL,
  bin_dir = "",
  tol = 0
)
```

Arguments

needle_image	A string of image file path or a cimg class object of imager library.
center	A logical. TRUE returns center position of needle_image.
exact	A logical. Check matching exactly or not. FALSE compares sampled pixels only.
timeout	A numeric for timeout seconds.
corner	A string to specify a corner of the display. "top_left", "top_right", "bottom_left", or "bottom_right".
width, height	A integer to specify width or height of the corner.
size	Integers to specify width or height of display size.
scale	A numeric to specify display scale.
bin_dir	A string for directory name of screenshot.exe on Win.
tol	A numeric for the tolerance of the comparison, in steps of 255 grayscale levels. 0 needs an exact match, which is right for a screenshot saved as PNG. Use a positive value for an image that has been through a lossy format such as JPEG.

Value

A numeric pair of xy location.

Examples

```
## Not run:
sc <- screenshot()
if(sc != ""){
  sc_image <- imager::load.image(sc)
  w <- 100
  h <- 80
  pos_x <- 1
  pos_y <- imager::height(sc_image) - h
  needle <- crop_image(sc_image, pos_x, pos_y, w, h)
  (locate_image(needle)) # center location
  pos <- locate_image(needle, center = FALSE)
  found <- crop_image(sc_image, pos[1], pos[2], w, h)
  layout(c(1:3))
  plot(sc_image)
  plot(needle)
  plot(found)
  # usse `coner` to limit searching field
  # `coner` can be used in Windows
  pos <- locate_image(needle, corner = "bottom_left", center = FALSE)
}

## End(Not run)
```

locate_ndl_in_hay	<i>Locate needle image matrix position in a haystack_image matrix. Helper function for locate_image().</i>
-------------------	--

Description

Searches the value that the needle and the haystack have in common and that appears fewest times in the haystack, and uses one of its positions as an anchor. Every candidate position of the needle is then a difference between a position of the value in the haystack and the anchor, so only those candidates have to be compared.

Usage

```
locate_ndl_in_hay(ndl_mt, hay_mt, exact = TRUE, timeout = 5, tol = 0)
```

Arguments

ndl_mt, hay_mt	A matrix
exact	A logical. Check matching exactly or not. FALSE compares sampled pixels only.
timeout	A numeric for timeout seconds.
tol	A numeric for the tolerance of the comparison, in steps of 255 grayscale levels. 0 needs an exact match. Use a positive value for an image that has been through a lossy format such as JPEG.

Value

A numeric pair of xy location for needle image.

Examples

```
haystack_image <- imager::load.example("parrots")
w <- 100
h <- 50
needle_image <- crop_image(haystack_image, 129, 257, w, h)
hay_mt <- image2gray_matrix(haystack_image)
ndl_mt <- image2gray_matrix(needle_image)
(pos <- locate_ndl_in_hay(ndl_mt, hay_mt))

found <- crop_image(haystack_image, pos[1], pos[2], w, h)
layout(c(1:3))
plot(haystack_image)
plot(needle_image)
plot(found)
```

reset_transparent	<i>Reset transparent background of an image</i>
-------------------	---

Description

Reset transparent background of an image

Usage

```
reset_transparent(path)
```

Arguments

path	A character string specifying the file path to the image to be processed.
------	---

Value

The file path to the processed image.

Examples

```
## Not run:
path <- "directory/image.png"
reset_transparent(path)

## End(Not run)
```

save_bmp	<i>Save an image as a BMP file</i>
----------	------------------------------------

Description

Save an image as a BMP file

Usage

```
save_bmp(image_data, path)
```

Arguments

image_data	A raster image data object, such as an array of pixel values or an R object representing an image.
path	The path to the file to be saved.

Value

Saves the image as a BMP file at the specified path.

Examples

```
## Not run:  
# Create an image data object  
image_data <- matrix(rnorm(100), ncol = 10)  
# Save the image as a BMP file  
save_bmp(image_data, "image.bmp")  
  
## End(Not run)
```

save_clipboard_image	<i>Saves an image from the clipboard to a file</i>
----------------------	--

Description

This function works only on windows.

Usage

```
save_clipboard_image(path = "", reset_transparent = TRUE)
```

Arguments

path	Optional path to save the image to. If not specified, a temporary file will be created.
reset_transparent	A logical. If true, the image of transparent color will be removed.

Value

The path to the saved image file.

Examples

```
## Not run:  
# Save the image from the clipboard to a file  
save_clipboard_image("clipboard_image.png")  
  
## End(Not run)
```

screenshot	<i>Take a screenshot.</i>
------------	---------------------------

Description

Need to install screenshot.exe on Win by install_screenshot().

Usage

```
screenshot(file = "", bin_dir = "", quote = FALSE)
```

Arguments

file	A string for file name of screenshot.
bin_dir	A string for directory name of screenshot.exe on Win.
quote	Deprecated and ignored. The command is always quoted.

Details

The command is always quoted, so that a path with a space (such as "C:/Program Files/R/R-4.5.1/library/screenshot") works.

Value

A file name of screenshot. When "", screenshot will be saved in a tempral directory.

See Also

```
install_screenshot()
```

Examples

```
if(interactive()){  
  
  sc <- screenshot()  
  if(sc != ""){  
    sc_image <- imager::load.image(sc)  
    plot(sc_image)  
  }  
  
}
```

screenshot_exists	<i>Find screenshot exec file.</i>
-------------------	-----------------------------------

Description

Find screenshot exec file.

Usage

```
screenshot_exists(bin_dir = "")
```

Arguments

bin_dir	A string for directory name screenshot.exe exec file. No need on Mac and Linux.
---------	---

Value

A logical.

Examples

```
screenshot_exists()
```

xy_pos	<i>Get xy position of a value in a matrix Helper function for locate_ndl_in_hay().</i>
--------	--

Description

Get xy position of a value in a matrix Helper function for locate_ndl_in_hay().

Usage

```
xy_pos(mt, val)
```

Arguments

mt	A matrix
val	A matrix

Value

A numeric pairs of xy location.

Examples

```
nrow <- 4  
mt <- matrix(1:12, nrow = nrow)  
xy_pos(mt, 5)
```

Index

* datasets

clipboard_sample, 3

bitmap2png, 2

clipboard2bitmap, 3

clipboard_sample, 3

compare_table, 4

count_val_freq, 4

create_header, 5

crop_image, 5

display_corner, 6

display_size, 7

get_clipboard_image, 7

get_os, 8

hay2needle (crop_image), 5

hex2little_endian, 8

image2gray_matrix, 9

index2xy, 9

install_screenshot, 10

is_all_same, 11

locate_image, 11

locate_ndl_in_hay, 13

reset_transparent, 14

save_bmp, 14

save_clipboard_image, 15

screenshot, 16

screenshot_exists, 17

xy_pos, 17