

Package: swatches (via r-universe)

August 14, 2024

Type Package

Title Read, Inspect, Manipulate, and Save Color Swatch Files

Version 0.6.0

Date 2022-04-02

Maintainer Bob Rudis <bob@rud.is>

Description There are numerous places to create and download color palettes. These are usually shared in Adobe swatch file formats of some kind. There is also often the need to use standard palettes developed within an organization to ensure that aesthetics are carried over into all projects and output. Now there is a way to read these swatch files in R and avoid transcribing or converting color values by hand or with other programs. This package provides functions to read and inspect Adobe Color (ACO), Adobe Swatch Exchange (ASE), GIMP Palette (GPL), OpenOffice palette (SOC) files and KDE Palette ("`colors") files. Detailed descriptions of Adobe Color and Swatch Exchange file formats as well as other swatch file formats can be found at <http://www.selapa.net/swatches/colors/fileformats.php>.

URL <https://github.com/hrbrmstr/swatches>

BugReports <https://github.com/hrbrmstr/swatches/issues>

License MIT + file LICENSE

Encoding UTF-8

NeedsCompilation yes

Imports httr, pack, stringi, xml2, tools, colorspace, grDevices, graphics, methods, stats

Suggests testthat, covr

Roxygen list(markdown = TRUE)

Depends R (>= 3.2.0)

RoxygenNote 7.1.2

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

RemoteUrl <https://github.com/hrbrmstr/swatches>

RemoteRef HEAD

RemoteSha e2598bde11fdb3415fd7190b838c6b3f9597f194

Contents

add_color	2
ase_encode	3
create_ase	4
hex_to_ase	4
read_aco	5
read_ase	6
read_gpl	7
read_kde	8
read_palette	8
read_soc	9
show_palette	10
trim_palette	10

Index **11**

add_color	<i>Add a named color to an ASE swatch object</i>
-----------	--

Description

Add a named color to an ASE swatch object

Usage

```
add_color(ase_obj, name, model, color_vals, type)
```

Arguments

ase_obj	An ASE object created with create_ase()
name	name of the color
model	one of "RGB", "Gray", "CMYK", or "LAB"
color_vals	vector of color values associated with the chosen model
type	one of "global", "spot", or "process"

Value

ase_obj (invisibly)

Examples

```
create_ase() |>
add_color(
  name = "RGB Red",
  model = "RGB",
  color_vals = as.vector(col2rgb("#FF0000")/255),
  type = "global"
) |>
add_color(
  name = "RGB Yellow",
  model = "RGB",
  color_vals = as.vector(col2rgb("#FFFF00")/255),
  type = "global"
) |>
ase_encode() |>
writeBin(tempfile(fileext = ".ase"))
```

ase_encode

Encode a built ASE object for output

Description

Encode a built ASE object for output

Usage

```
ase_encode(data)
```

Arguments

data An ASE swatch object created by [create_ase\(\)](#)

Value

raw vector

Examples

```
create_ase() |>
add_color(
  name = "RGB Red",
  model = "RGB",
  color_vals = as.vector(col2rgb("#FF0000")/255),
  type = "global"
) |>
add_color(
  name = "RGB Yellow",
  model = "RGB",
  color_vals = as.vector(col2rgb("#FFFF00")/255),
```

```
    type = "global"  
  ) |>  
  ase_encode() |>  
  writeBin(tempfile(fileext = ".ase"))
```

create_ase

Create an ASE object

Description

Create an ASE object

Usage

```
create_ase()
```

Value

a list (invisibly) classed as ase

Examples

```
create_ase() |>  
add_color(  
  name = "RGB Red",  
  model = "RGB",  
  color_vals = as.vector(col2rgb("#FF0000")/255),  
  type = "global"  
) |>  
add_color(  
  name = "RGB Yellow",  
  model = "RGB",  
  color_vals = as.vector(col2rgb("#FFFF00")/255),  
  type = "global"  
) |>  
ase_encode() |>  
writeBin(tempfile(fileext = ".ase"))
```

hex_to_ase

Convert a list of named, hexadecimal RGB colors into an ASE object

Description

Convert a list of named, hexadecimal RGB colors into an ASE object

Usage

```
hex_to_ase(colors, type)
```

Arguments

colors named vector of RGB hex colors
 type one of "global", "spot", or "process"

Value

ase object

Examples

```
ase_temp <- tempfile(fileext = ".ase")
on.exit(unlink(ase_temp))

github_url <- "https://github.com/picwellwisher12pk/en_us/raw/master/Swatches/Metal.ase"
metal <- read_ase(github_url)

hex_to_ase(metal, "global") |>
  ase_encode() |>
  writeBin(ase_temp)

read_ase(ase_temp)
```

read_aco	<i>Read colors from Adobe Color (ACO) files</i>
----------	---

Description

Given a path or URL to an .aco file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_aco(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path partial or full file path or URL to an ACO file
 use_names add color names to the vector (defaults to TRUE). See NOTE
 .verbose show extra information about ACO file processing

Note

When using named color palettes in a ggplot2 scale_ context, you must unname, set use_names to FALSE or override their names to map to your own factor levels.

Examples

```
# built-in palette
eighties <- read_aco(system.file("palettes", "tomorrow_night_eighties.aco", package="swatches"))
print(eighties)
show_palette(eighties)

# from the internet directly
## Not run:
tomorrow_night <- read_aco("https://bit.ly/tomorrow-night-aco")
print(tomorrow_night)
show_palette(tomorrow_night)

## End(Not run)
```

read_ase

Read colors from Adobe Swatch Exchange (ASE) files

Description

Given a path or URL to an .ase file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_ase(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to an ASE file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about ASE file processing

Note

When using named color palettes in a ggplot2 scale_ context, you must unname, set use_names to FALSE or override their names to map to your own factor levels. Also, Neither Lab nor greyscale colors are supported.

Examples

```
# built-in palette
keep_the_change <- read_ase(system.file("palettes", "keep_the_change.ase", package="swatches"))
print(keep_the_change)
show_palette(keep_the_change)

# from the internet directly
## Not run:
github_url <- "https://github.com/picwellwisher12pk/en_us/raw/master/Swatches/Metal.ase"
```

```
metal <- read_gpl(github_url)
print(metal)
show_palette(metal)

## End(Not run)
```

read_gpl

Read colors from GIMP Palette (GPL) files

Description

Given a path or URL to an .gpl file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_gpl(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to a GPL file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about GPL file processing

Note

When using named color palettes in a ggplot2 scale_ context, you must unname, set use_names to FALSE or override their names to map to your own factor levels. Also, Neither Lab nor greyscale colors are supported.

Examples

```
# built-in palette
gimp16 <- read_gpl(system.file("palettes", "base16.gpl", package="swatches"))
print(gimp16)
show_palette(gimp16)

# from the internet directly
## Not run:
bright <- read_gpl(URL)
print(bright)
show_palette(bright)

## End(Not run)
```

read_kde	<i>Read colors from KDE Palette (colors) files</i>
----------	--

Description

Given a path or URL to an .colors file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_kde(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to a GPL file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about GPL file processing

Note

When using named color palettes in a ggplot2 `scale_` context, you must `unnamed`, set `use_names` to `FALSE` or override their names to map to your own factor levels.

Examples

```
# built-in palette
fourty <- read_kde(system.file("palettes", "fourty.colors", package="swatches"))
print(fourty)
#show_palette(fourty)

# show_palette(bright)
```

read_palette	<i>Read colors from palette files</i>
--------------	---------------------------------------

Description

Given a path or URL to an palette file, this function will attempt to determine which palette file format to read by the file type and return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_palette(path, use_names = TRUE, .verbose = FALSE)
```


Arguments

path	partial or full file path or URL to a GPL file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about GPL file processing

Note

When using named color palettes in a ggplot2 `scale_` context, you must `unnamed`, set `use_names` to FALSE or override their names to map to your own factor levels.

read_soc	<i>Read colors from OpenOffice Palette (SOC) files</i>
----------	--

Description

Given a path or URL to an `.soc` file, this function will return a named character vector (if color names are present) of hex RGB colors.

Usage

```
read_soc(path, use_names = TRUE, .verbose = FALSE)
```

Arguments

path	partial or full file path or URL to a GPL file
use_names	add color names to the vector (defaults to TRUE). See NOTE
.verbose	show extra information about GPL file processing

Note

When using named color palettes in a ggplot2 `scale_` context, you must `unnamed`, set `use_names` to FALSE or override their names to map to your own factor levels.

Examples

```
# built-in palette
soc_file <- system.file("palettes", "ccooo.soc", package="swatches")
system(sprintf("cat %s", soc_file))
ccooo <- read_soc(soc_file)
print(ccooo)
show_palette(ccooo)

# from the internet directly
## Not run:
galaxy <- read_soc("https://www.openoffice.org/ui/VisualDesign/docs/colors/galaxy.soc")
print(galaxy)
show_palette(galaxy)

## End(Not run)
```

show_palette	<i>Display a color palette</i>
--------------	--------------------------------

Description

Given a character vector (hex RGB values), display palette in graphics window.

Usage

```
show_palette(palette)
```

Arguments

palette vector of character hex RGB values

Examples

```
# built-in palette
keep_the_change <- read_ase(system.file("palettes",
                                       "keep_the_change.ase", package="swatches"))
print(keep_the_change)
# show_palette(keep_the_change)
```

trim_palette	<i>Try to intelligently reduce a large palette down to a reasonable smaller set of colors</i>
--------------	---

Description

Given a palette and a desired number of colors to use, this function will compute CIEDE2000 and attempt to reduce the input set to a distinct smaller set of colors based on color distances.

Usage

```
trim_palette(pal, n = 5)
```

Arguments

pal input palette to reduct
n number of desired colors

Value

vector of n colors from pal

Note

internal CIEDE2000 color distance implementation by Gaurav Sharma & Maynard P Baalthazar

Index

`add_color`, 2
`ase_encode`, 3

`create_ase`, 4
`create_ase()`, 2, 3

`hex_to_ase`, 4

`read_aco`, 5
`read_ase`, 6
`read_gpl`, 7
`read_kde`, 8
`read_palette`, 8
`read_soc`, 9

`show_palette`, 10

`trim_palette`, 10