Package 'rsixel'

January 21, 2026

Title Encoding and Decoding Sixel Images

Version 0.0.4

Description Provides a native R implementation for encoding and decoding 'sixel' graphics (<https://vt100.net/docs/vt3xx-gp/chapter14.html>), and a dedicated 'sixel' graphics device that allows plots to be rendered directly within compatible terminal emulators.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports png

Suggests jpeg, magick

NeedsCompilation yes

Author Weifan Lv [aut, cre]

Maintainer Weifan Lv <lvweifan@pku.edu.cn>

Repository CRAN

Date/Publication 2026-01-21 19:10:02 UTC

Contents

`imgcat` *Create SIXEL escape sequence for image file*

Description

Create SIXEL escape sequence for image file. `jpeg`, `png` or `magick` packages are required to read image files. Image with alpha channel will be blended with the specified background color.

Usage

```
imgcat(
  path,
  ...,
  max.colors = 256,
  iter.max = 10,
  background = "white",
  file = ""
)
```

Arguments

`path`	character, path to a image file.
`...`	other positional arguments will be omitted.
`max.colors`	integer, max colors of the palette. The maximum is 256. This parameter will be passed to `sixelEncode`.
`iter.max`	integer, maximum number of iterations for k-means clustering. This parameter will be passed to `sixelEncode`.
`background`	character, background color to blend with for pixel with transparency. Default is "white".
`file`	A connection, or a character string naming the file to print to. This parameter will be passed to `cat`

Value

None (invisible 'NULL').

Examples

```
imgcat(system.file("img", "Rlogo.jpg", package="jpeg"))
```

readSIXEL *Read a SIXEL image*

Description

Reads an image from a SIXEL file into a raster array.

Usage

```
readSIXEL(source)
```

Arguments

source	character, name of the file to read from.

Value

A raster array with values ranging from 0 to 1. The array has dimensions (height, width, 3) where the third dimension represents the R, G, and B color channels.

Examples

```
# read a sample file
img <- readSIXEL(system.file("snake.six", package="rsixel"))
```

sixel *SIXEL graphics device*

Description

A graphics device that outputs SIXEL sequences to the console when closed. This device wraps the png() device and encodes the output as SIXEL.

Usage

```
sixel(
  file = "",
  width = 480,
  height = 480,
  max.colors = 256,
  iter.max = 10,
  background = "white",
  ...
)
```

Arguments

file	A connection, or a character string naming the file to print to. This parameter will be passed to cat. Default is "" (stdout).
width	integer, width of the output image in pixels. Default is 480.
height	integer, height of the output image in pixels. Default is 480.
max.colors	integer, max colors of the palette. The maximum is 256. This parameter will be passed to sixelEncode. Default is 256.
iter.max	integer, maximum number of iterations for k-means clustering. This parameter will be passed to sixelEncode. Default is 10.
background	character, background color to blend with for pixel with transparency. Default is "white".
...	Additional arguments passed to png().

Value

The device number (invisible).

Examples

```
sixel()
plot(c(1, 2))
dev.off()
```

---

sixelDecode *Decode SIXEL escape sequence to image data*

Description

Parse a SIXEL escape sequence and convert it to a raster array.

Usage

```
sixelDecode(data)
```

Arguments

data	character, SIXEL escape sequence.

Value

A raster array with values ranging from 0 to 1. The array has dimensions (height, width, 3) where the third dimension represents the R, G, and B color channels.

Examples

```
# read sixel sequence
sixel_file <- system.file("snake.six", package="rsixel")
sixel_data <- readChar(sixel_file, file.info(sixel_file)$size)
img <- sixelDecode(sixel_data)
```

`sixelEncode` *Create SIXEL escape sequence from image data*

Description

Create SIXEL escape sequence from image data. Quantization is done by k-means clustering.

Usage

```
sixelEncode(image, max.colors = 256, iter.max = 10)
```

Arguments

`image`	a three dimensional RGB array with values ranging from 0 to 1.
`max.colors`	integer, max colors of the palette. The maximum is 256. Default is 256.
`iter.max`	integer, maximum number of iterations for k-means clustering.

Value

SIXEL escape sequence

Examples

```
img <- png::readPNG(system.file("img", "Rlogo.png", package="png"))
cat(sixelEncode(img, 4))
```

Index