

# Package ‘loder’

May 8, 2026

**Version** 0.2.1

**Date** 2022-12-16

**Title** Dependency-Free Access to PNG Image Files

**Suggests** testthat, covr

**Description** Read and write access to PNG image files using the LodePNG library. The package has no external dependencies.

**License** BSD\_3\_clause + file LICENCE

**Encoding** UTF-8

**URL** <https://github.com/jonclayden/loder>

**BugReports** <https://github.com/jonclayden/loder/issues>

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Author** Jon Clayden [aut, cre],  
Lode Vandevenne [aut]

**Maintainer** Jon Clayden <code@clayden.org>

**Repository** CRAN

**Date/Publication** 2022-12-16 18:20:05 UTC

## Contents

inspectPng . . . . .	2
readPng . . . . .	3
writePng . . . . .	4
<b>Index</b>	<b>5</b>

inspectPng

*Read metadata from a PNG file*

---

**Description**

Inspect a PNG file, returning parsed metadata relating to it.

**Usage**

```
inspectPng(file)

## S3 method for class 'lodermeta'
print(x, ...)
```

**Arguments**

file	A character string giving the file name to read from.
x	An object of class "lodermeta".
...	Additional arguments (which are ignored).

**Details**

The LodePNG library is used to parse the PNG file at the specified path. The result is a string like the input, but of class "lodermeta" and with several attributes set describing the file's contents. There is a print method for these objects.

**Value**

inspectPng returns a character vector of class "lodermeta". The print method is called for its side-effect.

**See Also**

readPng to read the pixel values.

**Examples**

```
path <- system.file("extdata", "pngsuite", package="loder")
inspectPng(file.path(path, "basn6a08.png"))
```

---

readPng	<i>Read a PNG file</i>
---------	------------------------

---

### Description

Read an image from a PNG file and convert the pixel data into an R array.

### Usage

```
readPng(file)

## S3 method for class 'loder'
print(x, ...)
```

### Arguments

file	A character string giving the file name to read from.
x	An object of class "loder".
...	Additional arguments (which are ignored).

### Details

The LodePNG library is used to read the PNG file at the specified path. LodePNG can handle a wide variety of subformats and bit depths, but the output of this function is currently standardised to an integer-mode array with 8-bit range, i.e. between 0 and 255. Attributes specifying the background colour, spatial resolution and/or aspect ratio are attached to the result if this information is stored with the image.

### Value

readPng returns an integer-mode array of class "loder". The print method is called for its side-effect.

### See Also

inspectPng to read only metadata from the file. In addition, the readPNG function in the venerable png package offers similar functionality to readPng, but relies on an external installation of libpng. By contrast, loder includes the LodePNG library.

### Examples

```
path <- system.file("extdata", "pngsuite", package="loder")
image <- readPng(file.path(path, "basn6a08.png"))
print(image)
attributes(image)
```

---

 writePng

*Write a PNG file*


---

**Description**

Write a numeric or logical array to a PNG file.

**Usage**

```
writePng(image, file, ..., compression = 4L, interlace = FALSE)
```

**Arguments**

image	An array containing the pixel data.
file	A character string giving the file name to write to.
...	Additional metadata elements, which override equivalently named attributes of image. See Details.
compression	Compression level, an integer value between 0 (no compression, fastest) and 6 (maximum compression, slowest).
interlace	Logical value: should the image be interlaced?

**Details**

The LodePNG library is used to write a PNG file at the specified path. The source data should be of logical, integer or numeric mode. Metadata attributes of the image will be stored where applicable, and may be overwritten using named arguments. LodePNG will choose the bit depth of the final image.

Attributes which are currently stored are as follows. In each case an argument of the appropriate name can be used to override a value stored with the image.

range A numeric 2-vector giving the extremes of the intensity window, i.e. the black and white points. Values outside this range will be clipped.

background A hexadecimal colour string giving the background colour.

dpi A numeric 2-vector giving the dots-per-inch resolution of the image in each dimension.

asp The aspect ratio of the image. Ignored if dpi is present and valid.

text A character vector (possibly named) of text strings to store in the file. Only ASCII and UTF-8 encoded strings are currently supported.

Dimensions are always taken from the image, and cannot be modified here.

**Value**

The file argument, invisibly.

**See Also**

[readPng](#) for reading images.

# Index

`inspectPng`, 2

`print.loder (readPng)`, 3

`print.lodermeta (inspectPng)`, 2

`readPng`, 3, 4

`writePng`, 4