

Discover Packages > Standard library > image > gif 

**gif**



package

standard library

Version: **go1.20.1**

Latest

| Published: Feb 14, 2023

| License: **BSD-3-Clause**| Imports: **10**Imported by: **3,988****Details** Valid **go.mod** file  Redistributable license  Tagged version  Stable version [Learn more](#)**Repository**[cs.opensource.google/go/go](https://cs.opensource.google/go/go)**Links** [Report a Vulnerability](#) Documentation <> **Documentation****Overview**

Package gif implements a GIF image decoder and encoder.

The GIF specification is at <https://www.w3.org/Graphics/GIF/spec-gif89a.txt>.

**Index**

## Constants

`func Decode(r io.Reader) (image.Image, error)``func DecodeConfig(r io.Reader) (image.Config, error)``func Encode(w io.Writer, m image.Image, o *Options) error``func EncodeAll(w io.Writer, g *GIF) error``type GIF``func DecodeAll(r io.Reader) (*GIF, error)``type Options`**Constants**[View Source](#)

```
const (  
    DisposalNone          = 0x01  
    DisposalBackground    = 0x02  
    DisposalPrevious      = 0x03  
)
```

Disposal Methods.

## Variables

This section is empty.

## Functions

### func Decode

```
func Decode(r io.Reader) (image.Image, error)
```

Decode reads a GIF image from `r` and returns the first embedded image as an `image.Image`.

### func DecodeConfig

```
func DecodeConfig(r io.Reader) (image.Config, error)
```

DecodeConfig returns the global color model and dimensions of a GIF image without decoding the entire image.

### func Encode

added in go1.2

```
func Encode(w io.Writer, m image.Image, o *Options) error
```

Encode writes the Image `m` to `w` in GIF format.

### func EncodeAll

added in go1.2

```
func EncodeAll(w io.Writer, g *GIF) error
```

EncodeAll writes the images in `g` to `w` in GIF format with the given loop count and delay between frames.

## Types

### type GIF

```
type GIF struct {
    Image []*image.Paletterd // The successive images.
    Delay []int               // The successive delay times, one per frame, in 100ths of a second.
    // LoopCount controls the number of times an animation will be
    // restarted during display.
    // A LoopCount of 0 means to loop forever.
    // A LoopCount of -1 means to show each frame only once.
    // Otherwise, the animation is looped LoopCount+1 times.
    LoopCount int
    // Disposal is the successive disposal methods, one per frame. For
    // backwards compatibility, a nil Disposal is valid to pass to EncodeAll,
    // and implies that each frame's disposal method is 0 (no disposal)
```

```
// specified).
Disposal []byte
// Config is the global color table (palette), width and height. A nil or
// empty-color.Palette Config.ColorModel means that each frame has its own
// color table and there is no global color table. Each frame's bounds must
// be within the rectangle defined by the two points (0, 0) and
// (Config.Width, Config.Height).
//
// For backwards compatibility, a zero-valued Config is valid to pass to
// EncodeAll, and implies that the overall GIF's width and height equals
// the first frame's bounds' Rectangle.Max point.
Config image.Config
// BackgroundIndex is the background index in the global color table, for
// use with the DisposalBackground disposal method.
BackgroundIndex byte
}
```

GIF represents the possibly multiple images stored in a GIF file.

## func DecodeAll

```
func DecodeAll(r io.Reader) (*GIF, error)
```

DecodeAll reads a GIF image from r and returns the sequential frames and timing information.

## type Options

added in go1.2

```
type Options struct {
    // NumColors is the maximum number of colors used in the image.
    // It ranges from 1 to 256.
    NumColors int

    // Quantizer is used to produce a palette with size NumColors.
    // palette.Plan9 is used in place of a nil Quantizer.
    Quantizer draw.Quantizer

    // Drawer is used to convert the source image to the desired palette.
    // draw.FloydSteinberg is used in place of a nil Drawer.
    Drawer draw.Drawer
}
```

Options are the encoding parameters.

## Source Files

[View all](#) 

[reader.go](#)

[writer.go](#)

## Why Go

[Use Cases](#)

[Case Studies](#)

## Get Started

[Playground](#)

[Tour](#)

[Stack Overflow](#)

[Help](#)

## Packages

[Standard Library](#)

[About Go Packages](#)

## About

[Download](#)

[Blog](#)

[Issue Tracker](#)

[Release Notes](#)

[Brand Guidelines](#)

[Code of Conduct](#)

## Connect

[Twitter](#)

[GitHub](#)

[Slack](#)

[r/golang](#)

[Meetup](#)

[Golang Weekly](#)

---

[Copyright](#)

[Terms of Service](#)

[Privacy Policy](#)

[Report an Issue](#)



Google