Core Graphics / CGImage

Class

CGImage

A bitmap image or image mask.

iOS | iPadOS | Mac Catalyst | macOS | tvOS | visionOS | watchOS

```
class CGImage
```

Overview

A bitmap image is a rectangular array of pixels, each of which represents a single sample or data point from a source image.

Topics

Creating images

```
init?(width: Int, height: Int, bitsPerComponent: Int, bitsPerPixel: Int, bytesPerRow: Int, space: CGColorSpace, bitmapInfo: CGBitmapInfo, provider: CGDataProvider, decode: UnsafePointer<CGFloat>?, shouldInterpolate: Bool, intent: CGColorRenderingIntent)
```

Creates a bitmap image from data supplied by a data provider.

```
init?(jpegDataProviderSource: CGDataProvider, decode: UnsafePointer<CGFloat>?, shouldInterpolate: Bool, intent: CGColorRenderingIntent)
```

Creates a bitmap image using JPEG-encoded data supplied by a data provider.

```
init?(pngDataProviderSource: CGDataProvider, decode: UnsafePointer<
CGFloat>?, shouldInterpolate: Bool, intent: CGColorRenderingIntent)
```

Creates a bitmap image using PNG-encoded data supplied by a data provider.

```
init?(headroom: Float, width: Int, height: Int, bitsPerComponent: Int,
bitsPerPixel: Int, bytesPerRow: Int, space: CGColorSpace, bitmapInfo:
CGBitmapInfo, provider: CGDataProvider, decode: UnsafePointer<CGFloat
>?, shouldInterpolate: Bool, intent: CGColorRenderingIntent)
```

Examining an image

```
var isMask: Bool
```

Returns whether a bitmap image is an image mask.

```
var width: Int
```

Returns the width of a bitmap image, in pixels.

```
var height: Int
```

Returns the height of a bitmap image.

```
var bitsPerComponent: Int
```

Returns the number of bits allocated for a single color component of a bitmap image.

```
var bitsPerPixel: Int
```

Returns the number of bits allocated for a single pixel in a bitmap image.

```
var bytesPerRow: Int
```

Returns the number of bytes allocated for a single row of a bitmap image.

```
var colorSpace: CGColorSpace?
```

Return the color space for a bitmap image.

```
var alphaInfo: CGImageAlphaInfo
```

Returns the alpha channel information for a bitmap image.

```
enum CGImageAlphaInfo
```

Storage options for alpha component data.

```
var dataProvider: CGDataProvider?
```

Returns the data provider for a bitmap image or image mask.

```
var decode: UnsafePointer<CGFloat>?
```

Returns the decode array for a bitmap image.

```
var shouldInterpolate: Bool
```

Returns the interpolation setting for a bitmap image.

```
var renderingIntent: CGColorRenderingIntent
```

Returns the rendering intent setting for a bitmap image.

```
var bitmapInfo: CGBitmapInfo
```

Returns the bitmap information for a bitmap image.

```
struct CGBitmapInfo
```

Component information for a bitmap image.

```
var utType: CFString?
```

The Universal Type Identifier for the image.

Copying an image

```
func copy() -> CGImage?
```

Creates a copy of a bitmap image.

```
func copy(colorSpace: CGColorSpace) -> CGImage?
```

Creates a copy of a bitmap image, replacing its colorspace.

Creating images by modifying an image

```
func cropping(to: CGRect) -> CGImage?
```

Creates a bitmap image using the data contained within a subregion of an existing bitmap image.

```
func masking(CGImage) -> CGImage?
```

Creates a bitmap image from an existing image and an image mask.

```
func copy(maskingColorComponents: [CGFloat]) -> CGImage?
```

Creating image masks

```
init?(maskWidth: Int, height: Int, bitsPerComponent: Int, bitsPerPixel:
Int, bytesPerRow: Int, provider: CGDataProvider, decode: UnsafePointer<
CGFloat>?, shouldInterpolate: Bool)
```

Creates a bitmap image mask from data supplied by a data provider.

Adopting high dynamic range (HDR)

Enhancing high dynamic range image rendering

Improve your app’s High Dynamic Range (HDR) image support with metadata.

```
var contentHeadroom: Float
```

```
var calculatedContentHeadroom: Float
```

```
var contentAverageLightLevel: Float
```

```
var calculatedContentAverageLightLevel: Float
```

```
func copy(contentAverageLightLevel: Float) -> CGImage?
```

```
func copyWithCalculatedHDRStats() -> CGImage?
```

Constants

```
enum CGImageAlphaInfo
```

Storage options for alpha component data.

```
struct CGBitmapInfo
```

Component information for a bitmap image.

Host Endian Bitmap Formats

Bit-depth constants for image bitmaps in host-endian byte order.

Working with Core Foundation types

```
class var typeID: CFTypeID
```

Returns the type identifier for CGImage objects.

Instance properties

```
var byteOrderInfo: CGImageByteOrderInfo
```

```
var containsImageSpecificToneMappingMetadata: Bool
```

```
var contentHeadroom: Float
```

```
var pixelFormatInfo: CGImagePixelFormatInfo
var shouldToneMap: Bool
```

Relationships

Conforms To

```
Equatable
Hashable
Sendable
SendableMetatype
```

See Also

Related Documentation

Quartz 2D Programming Guide

2D Drawing

```
class CGContext
```

A Quartz 2D drawing environment.

```
class CGPath
```

An immutable graphics path: a mathematical description of shapes or lines to be drawn in a graphics context.

```
class CGMutablePath
```

A mutable graphics path: a mathematical description of shapes or lines to be drawn in a graphics context.

```
class CGLayer
```

An offscreen context for reusing content drawn with Core Graphics.