Class

CGColorSpace

A profile that specifies how to interpret a color value for display.

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

```
class CGColorSpace
```

Overview

A color space is multi-dimensional, and each dimension represents a specific color component. For example, the colors in an RGB color space have three dimensions or components—red, green, and blue. The intensity of each component is represented by floating point values—their range and meaning depends on the color space in question.

Different types of devices (scanners, monitors, printers) operate within different color spaces (RGB, CMYK, grayscale). Additionally, two devices of the same type (for example, color displays from different manufacturers) may operate within the same kind of color space, yet still produce a different range of colors, or gamut. Color spaces that are correctly specified ensure that an image has a consistent appearance regardless of the output device.

Core Graphics supports several kinds of color spaces:

- Calibrated color spaces ensure that colors appear the same when displayed on different devices. The visual appearance of the color is preserved, as far as the capabilities of the device allow.
- Device-dependent color spaces are tied to the system of color representation of a particular device. Device color spaces are not recommended when high-fidelity color preservation is important.
- Special color spaces—indexed and pattern. An indexed color space contains a color table with up to 256 entries and a base color space to which the color table entries are mapped. Each

entry in the color table specifies one color in the base color space. A pattern color space is used when stroking or filling with a pattern.

Topics

Creating Color Spaces

`init?(calibratedGrayWhitePoint: UnsafePointer<CGFloat>, blackPoint: UnsafePointer<CGFloat>?, gamma: CGFloat)`

Creates a calibrated grayscale color space.

`init?(calibratedRGBWhitePoint: UnsafePointer<CGFloat>, blackPoint: UnsafePointer<CGFloat>?, gamma: UnsafePointer<CGFloat>?, matrix: UnsafePointer<CGFloat>?)`

Creates a calibrated RGB color space.

`init?(iccBasedNComponents: Int, range: UnsafePointer<CGFloat>?, profile: CGDataProvider, alternate: CGColorSpace?)`

Creates a device-independent color space that is defined according to the ICC color profile specification.

`init?(indexedBaseSpace: CGColorSpace, last: Int, colorTable: UnsafePointer<UInt8>)`

Creates an indexed color space, consisting of colors specified by a color lookup table.

`init?(labWhitePoint: UnsafePointer<CGFloat>, blackPoint: UnsafePointer<CGFloat>?, range: UnsafePointer<CGFloat>?)`

Creates a device-independent color space that is relative to human color perception, according to the CIE L*a*b* standard.

`init?(patternBaseSpace: CGColorSpace?)`

Creates a pattern color space.

`init?(name: CFString)`

Creates a specified type of Quartz color space.

~~`init?(platformColorSpaceRef: UnsafeRawPointer)`~~

Creates a platform-specific color space.

Deprecated

`init?(iccData: CFTypeRef)`

Creates an ICC-based color space using the ICC profile contained in the specified data.

`init?(propertyListPlist: CFPropertyList)`

Creates a color space from a property list.

`func CGColorSpaceCreateDeviceRGB() -> CGColorSpace`

Creates a device-dependent RGB color space.

`func CGColorSpaceCreateDeviceCMYK() -> CGColorSpace`

Creates a device-dependent CMYK color space.

`func CGColorSpaceCreateDeviceGray() -> CGColorSpace`

Creates a device-dependent grayscale color space.

~~`init?(iccProfileData: CFData)`~~

Creates an ICC-based color space using the ICC profile contained in the specified data.

Deprecated

Examining a Color Space

`var baseColorSpace: CGColorSpace?`

Returns the base color space of a pattern or indexed color space.

`var numberOfComponents: Int`

Returns the number of color components in a color space.

`var model: CGColorSpaceModel`

Returns the color space model of the provided color space.

`enum CGColorSpaceModel`

Models for color spaces.

`var colorTable: [UInt8]?`

The entries in the color table of an indexed color space.

`func copyICCData() -> CFData?`

Returns a copy of the ICC profile data of the provided color space.

`func copyPropertyList() -> CFPropertyList?`

Returns a copy of the color space's properties.

~~`var iccData: CFData?`~~

Returns a copy of the ICC profile of the provided color space.

Deprecated

`var name: CFString?`

Returns the name used to create the specified color space.

`var supportsOutput: Bool`

Returns a Boolean indicating whether the color space can be used as a destination color space.

`var isWideGamutRGB: Bool`

Returns whether the RGB color space covers a significant portion of the NTSC color gamut.

Accessing System-Defined Color Spaces

`class let displayP3: CFString`

The Display P3 color space, created by Apple.

`class let displayP3_HLG: CFString`

The Display P3 color space, using the HLG transfer function.

~~`class let displayP3_PQ_EOTF: CFString`~~

The Display P3 color space, using the PQ transfer function.

Deprecated

`class let extendedLinearDisplayP3: CFString`

The Display P3 color space with a linear transfer function and extended-range values.

`class let sRGB: CFString`

The standard Red Green Blue (sRGB) color space.

`class let linearSRGB: CFString`

The sRGB color space with a linear transfer function.

`class let extendedSRGB: CFString`

The extended sRGB color space.

`class let extendedLinearSRGB: CFString`

The sRGB color space with a linear transfer function and extended-range values.

`class let genericGrayGamma2_2: CFString`

The generic gray color space that has an exponential transfer function with a power of 2.2.

```
class let extendedGray: CFString
```

The extended gray color space.

```
class let linearGray: CFString
```

The gray color space using a linear transfer function.

```
class let extendedLinearGray: CFString
```

The extended gray color space with a linear transfer function.

```
class let genericCMYK: CFString
```

The generic CMYK color space.

```
class let genericRGBLinear: CFString
```

The generic RGB color space with a linear transfer function.

```
class let genericXYZ: CFString
```

The XYZ color space, as defined by the CIE 1931 standard.

```
class let genericLab: CFString
```

The generic LAB color space.

```
class let acescgLinear: CFString
```

The ACEScg color space.

```
class let adobeRGB1998: CFString
```

The Adobe RGB (1998) color space.

```
class let dcip3: CFString
```

The DCI P3 color space, which is the digital cinema standard.

```
class let itur_709: CFString
```

The recommendation of the International Telecommunication Union (ITU) Radiocommunication sector for the BT.709 color space.

```
class let rommrgb: CFString
```

The Reference Output Medium Metric (ROMM) RGB color space.

```
class let itur_2020: CFString
```

The recommendation of the International Telecommunication Union (ITU) Radiocommunication sector for the BT.2020 color space.

~~`class let itur_2020_HLG: CFString`~~

The recommendation of the International Telecommunication Union (ITU) Radiocommunication sector for the BT.2020 color space, with the HLG transfer function.

Deprecated

~~`class let itur_2020_PQ_EOTF: CFString`~~

The recommendation of the International Telecommunication Union (ITU) Radiocommunication sector for the BT.2020 color space, with the PQ transfer function.

Deprecated

`class let extendedLinearITUR_2020: CFString`

The recommendation of the International Telecommunication Union (ITU) Radiocommunication sector for the BT.2020 color space, with a linear transfer function and extended range values.

`class let coreMedia709: CFString`

`class let displayP3_PQ: CFString`

`class let extendedDisplayP3: CFString`

`class let extendedITUR_2020: CFString`

~~`class let itur_2020_PQ: CFString`~~ Deprecated

`class let itur_2020_sRGBGamma: CFString`

`class let itur_2100_HLG: CFString`

`class let itur_2100_PQ: CFString`

`class let itur_709_HLG: CFString`

`class let itur_709_PQ: CFString`

`class let linearDisplayP3: CFString`

`class let linearITUR_2020: CFString`

Working with Core Foundation Types

`class var typeID: CFTypeID`

Returns the Core Foundation type identifier for Quartz color spaces.

Data Types

`enum CGColorRenderingIntent`

Handling options for colors that are not located within the destination color space of a graphics context.

Instance Methods

```
func isHDR() -> Bool
```

Relationships

Conforms To

```
Equatable
Hashable
Sendable
SendableMetatype
```

See Also

Related Documentation

Quartz 2D Programming Guide

Colors and Fonts

`class CGColor`

A set of components that define a color, with a color space specifying how to interpret them.

`class CGColorConversionInfo`

An object that describes how to convert between color spaces for use by other system services.

`class CGFont`

A set of character glyphs and layout information for drawing text.