

---

# CGColorSpace Reference

[Graphics & Imaging](#) > Quartz



2008-09-09



Apple Inc.  
© 2003, 2008 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, ColorSync, iChat, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

Adobe, Acrobat, and PostScript are trademarks or registered trademarks of Adobe Systems Incorporated in the U.S. and/or other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, **APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE**

**ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **CGColorSpace Reference 5**

|   |    |
|---|----|
| Overview  | 5  |
| Functions by Task                                 | 6  |
| Creating Device-Independent Color Spaces          | 6  |
| Creating Generic or Device-Dependent Color Spaces | 6  |
| Creating Special Color Spaces                     | 6  |
| Getting Information About Color Spaces            | 6  |
| Retaining and Releasing Color Spaces              | 7  |
| Functions   | 7  |
| CGColorSpaceCopyICCProfile                        | 7  |
| CGColorSpaceCreateCalibratedGray                  | 7  |
| CGColorSpaceCreateCalibratedRGB                   | 8  |
| CGColorSpaceCreateDeviceCMYK                      | 9  |
| CGColorSpaceCreateDeviceGray                      | 10 |
| CGColorSpaceCreateDeviceRGB                       | 10 |
| CGColorSpaceCreateICCBased                        | 11 |
| CGColorSpaceCreateIndexed                         | 12 |
| CGColorSpaceCreateLab                             | 12 |
| CGColorSpaceCreatePattern                         | 13 |
| CGColorSpaceCreateWithName                        | 14 |
| CGColorSpaceCreateWithPlatformColorSpace          | 14 |
| CGColorSpaceGetBaseColorSpace                     | 15 |
| CGColorSpaceGetColorTable                         | 15 |
| CGColorSpaceGetColorTableCount                    | 16 |
| CGColorSpaceGetModel                              | 16 |
| CGColorSpaceGetNumberOfComponents                 | 16 |
| CGColorSpaceGetTypeID                             | 17 |
| CGColorSpaceRelease                               | 17 |
| CGColorSpaceRetain                                | 18 |
| Data Types  | 18 |
| CGColorSpaceRef                                   | 18 |
| Constants   | 19 |
| Color Space Names                                 | 19 |
| Color Space Models                                | 19 |
| Color Rendering Intents                           | 21 |
| Named Color Spaces (Deprecated)                   | 22 |

---

## **Document Revision History 23**

---

## **Index 25**

---



# CGColorSpace Reference

---

|                         |   |
|-------------------------|---|
| <b>Derived From:</b>    | <i>CType Reference</i>  |
| <b>Framework:</b>       | ApplicationServices/ApplicationServices.h                               |
| <b>Declared in</b>      | CGColorSpace.h  |
| <b>Companion guides</b> | Quartz 2D Programming Guide<br>CGColor Reference<br>CGContext Reference |

## Overview

The `CGColorSpaceRef` opaque type encapsulates color space information that is used to specify how Quartz interprets color information. A color space specifies how color values are interpreted. 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.

Quartz 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. Pattern color spaces are supported in Mac OS X version 10.1 and later.

## Functions by Task

### Creating Device-Independent Color Spaces

[CGColorSpaceCreateCalibratedGray](#) (page 7)

Creates a calibrated grayscale color space.

[CGColorSpaceCreateCalibratedRGB](#) (page 8)

Creates a calibrated RGB color space.

[CGColorSpaceCreateICCBased](#) (page 11)

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

[CGColorSpaceCreateLab](#) (page 12)

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

### Creating Generic or Device-Dependent Color Spaces

In Mac OS X v10.4 and later, the color space returned by each of these functions is no longer device-dependent and is replaced by a generic counterpart.

[CGColorSpaceCreateDeviceCMYK](#) (page 9)

Creates a device-dependent CMYK color space.

[CGColorSpaceCreateDeviceGray](#) (page 10)

Creates a device-dependent grayscale color space.

[CGColorSpaceCreateDeviceRGB](#) (page 10)

Creates a device-dependent RGB color space.

[CGColorSpaceCreateWithPlatformColorSpace](#) (page 14)

Creates a platform-specific color space.

### Creating Special Color Spaces

[CGColorSpaceCreateIndexed](#) (page 12)

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

[CGColorSpaceCreatePattern](#) (page 13)

Creates a pattern color space.

[CGColorSpaceCreateWithName](#) (page 14)

Creates a specified type of Quartz color space.

### Getting Information About Color Spaces

[CGColorSpaceCopyICCProfile](#) (page 7)

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

[CGColorSpaceGetNumberOfComponents](#) (page 16)

Returns the number of color components in a color space.

[CGColorSpaceGetTypeID](#) (page 17)

Returns the Core Foundation type identifier for Quartz color spaces.

[CGColorSpaceGetModel](#) (page 16)

Returns the color space model of the provided color space.

[CGColorSpaceGetBaseColorSpace](#) (page 15)

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

[CGColorSpaceGetColorTableCount](#) (page 16)

Returns the number of entries in the color table of an indexed color space.

[CGColorSpaceGetColorTable](#) (page 15)

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

## Retaining and Releasing Color Spaces

[CGColorSpaceRelease](#) (page 17)

Decrements the retain count of a color space.

[CGColorSpaceRetain](#) (page 18)

Increments the retain count of a color space.

## Functions

### CGColorSpaceCopyICCProfile

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

```
CFDataRef CGColorSpaceCopyICCProfile(
    CGColorSpaceRef space
);
```

#### Parameters

*space*

The color space whose ICC profile you want to obtain.

#### Return Value

The ICC profile or NULL if the color space does not have an ICC profile.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

CGColorSpace.h

### CGColorSpaceCreateCalibratedGray

Creates a calibrated grayscale color space.

```
CGColorSpaceRef CGColorSpaceCreateCalibratedGray (
    const CGFloat whitePoint[3],
    const CGFloat blackPoint[3],
    CGFloat gamma
);
```

**Parameters***whitePoint*

An array of 3 numbers specifying the tristimulus value, in the CIE 1931 XYZ-space, of the diffuse white point.

*blackPoint*

An array of 3 numbers specifying the tristimulus value, in CIE 1931 XYZ-space, of the diffuse black point.

*gamma*

The gamma value appropriate to the imaging device.

**Return Value**

A new calibrated gray color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns `NULL`.

**Discussion**

Creates a device-independent grayscale color space that represents colors relative to a reference white point. This white point is based on the whitest light that can be generated by the output device. Colors in a device-independent color space should appear the same when displayed on different devices, to the extent that the capabilities of the device allow.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CGColorSpace.h`

**CGColorSpaceCreateCalibratedRGB**

Creates a calibrated RGB color space.

```
CGColorSpaceRef CGColorSpaceCreateCalibratedRGB (
    const CGFloat whitePoint[3],
    const CGFloat blackPoint[3],
    const CGFloat gamma[3],
    const CGFloat matrix[9]
);
```

**Parameters***whitePoint*

An array of 3 numbers specifying the tristimulus value, in the CIE 1931 XYZ-space, of the diffuse white point.

*blackPoint*

An array of 3 numbers specifying the tristimulus value, in CIE 1931 XYZ-space, of the diffuse black point.

*gamma*

An array of 3 numbers specifying the gamma for the red, green, and blue components of the color space.



*matrix*

An array of 9 numbers specifying the linear interpretation of the gamma-modified RGB values of the color space with respect to the final XYZ representation.

#### Return Value

A new calibrated RGB color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

#### Discussion

Creates a device-independent RGB color space that represents colors relative to a reference white point. This white point is based on the whitest light that can be generated by the output device. Colors in a device-independent color space should appear the same when displayed on different devices, to the extent that the capabilities of the device allow.

For color spaces that require a detailed gamma, such as the piecewise transfer function used in sRGB or ITU-R BT.709, you may want to use the function [CGColorSpaceCreateICCBased](#) (page 11) instead, because it can accurately represent these gamma curves.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`CGColorSpace.h`

## CGColorSpaceCreateDeviceCMYK

Creates a device-dependent CMYK color space.

```
CGColorSpaceRef CGColorSpaceCreateDeviceCMYK (
    void
);
```

#### Return Value

A device-dependent CMYK color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

#### Discussion

In Mac OS X v10.4 and later, this color space is no longer device-dependent and is replaced by the generic counterpart—`kCGColorSpaceGenericCMYK`—described in “[Color Space Names](#)” (page 19). If you use this function in Mac OS X v10.4 and later, colors are mapped to the generic color spaces. If you want to bypass color matching, use the color space of the destination context.

Colors in a device-dependent color space are not transformed or otherwise modified when displayed on an output device—that is, there is no attempt to maintain the visual appearance of a color. As a consequence, colors in a device color space often appear different when displayed on different output devices. For this reason, device color spaces are not recommended when color preservation is important.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`CGColorSpace.h`

## CGColorSpaceCreateDeviceGray

Creates a device-dependent grayscale color space.

```
CGColorSpaceRef CGColorSpaceCreateDeviceGray (
    void
);
```

### Return Value

A device-dependent gray color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

### Discussion

In Mac OS X v10.4 and later, this color space is no longer device-dependent and is replaced by the generic counterpart—`kCGColorSpaceGenericGray`—described in “[Color Space Names](#)” (page 19). If you use this function in Mac OS X v10.4 and later, colors are mapped to the generic color spaces. If you want to bypass color matching, use the color space of the destination context.

Colors in a device-dependent color space are not transformed or otherwise modified when displayed on an output device—that is, there is no attempt to maintain the visual appearance of a color. As a consequence, colors in a device color space often appear different when displayed on different output devices. For this reason, device color spaces are not recommended when color preservation is important.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

`CGColorSpace.h`

## CGColorSpaceCreateDeviceRGB

Creates a device-dependent RGB color space.

```
CGColorSpaceRef CGColorSpaceCreateDeviceRGB (
    void
);
```

### Return Value

A device-dependent RGB color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

### Discussion

In Mac OS X v10.4 and later, this color space is no longer device-dependent and is replaced by the generic counterpart—`kCGColorSpaceGenericRGB`—described in “[Color Space Names](#)” (page 19). If you use this function in Mac OS X v10.4 and later, colors are mapped to the generic color spaces. If you want to bypass color matching, use the color space of the destination context.

Colors in a device-dependent color space are not transformed or otherwise modified when displayed on an output device—that is, there is no attempt to maintain the visual appearance of a color. As a consequence, colors in a device color space often appear different when displayed on different output devices. For this reason, device color spaces are not recommended when color preservation is important.

### Availability

Available in Mac OS X v10.0 and later.

**Declared In**

CGColorSpace.h

**CGColorSpaceCreateICCBased**

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

```
CGColorSpaceRef CGColorSpaceCreateICCBased (
    size_t nComponents,
    const CGFloat *range,
    CGDataProviderRef profile,
    CGColorSpaceRef alternate
);
```

**Parameters***nComponents*

The number of color components in the color space defined by the ICC profile data. This must match the number of components actually in the ICC profile and must equal 1, 3, or 4.

*range*

An array of numbers that specify the minimum and maximum valid values of the corresponding color components. The size of the array is two times the number of components. If  $c[k]$  is the  $k$ th color component, the valid range is  $\text{range}[2*k] \leq c[k] \leq \text{range}[2*k+1]$ .

*profile*

A data provider that supplies the ICC profile.

*alternateSpace*

An alternate color space to use in case the ICC profile is not supported. The alternate color space must have *nComponents* color components. You must supply an alternate color space. If this parameter is NULL, then the function returns NULL.

**Return Value**

A new ICC-based color space object. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

**Discussion**

This function creates an ICC-based color space from an ICC color profile, as defined by the International Color Consortium. ICC profiles define the reproducible color gamut (the range of colors supported by a device) and other characteristics of a particular output device, providing a way to accurately transform the color space of one device to the color space of another. The ICC profile is usually provided by the manufacturer of the device. Additionally, some color monitors and printers contain electronically embedded ICC profile information, as do some bitmap formats such as TIFF. Colors in a device-independent color space should appear the same when displayed on different devices, to the extent that the capabilities of the device allow.

You may want to use this function for a color space that requires a detailed gamma, such as the piecewise transfer function used in sRGB or ITU-R BT.709, because this function can accurately represent these gamma curves.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CGColorSpace.h

## CGColorSpaceCreateIndexed

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

```
CGColorSpaceRef CGColorSpaceCreateIndexed (
    CGColorSpaceRef baseSpace,
    size_t lastIndex,
    const unsigned char *colorTable
);
```

### Parameters

*baseSpace*

The color space on which the color table is based.

*lastIndex*

The maximum valid index value for the color table. The value must be less than or equal to 255.

*colorTable*

An array of  $m \times (\text{lastIndex} + 1)$  bytes, where  $m$  is the number of color components in the base color space. Each byte is an unsigned integer in the range 0 to 255 that is scaled to the range of the corresponding color component in the base color space.

### Return Value

A new indexed color space object. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

### Discussion

An indexed color space contains a color table with up to 255 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 value in an indexed color space is treated as an index into the color table of the color space. The data in the table is in meshed format. (For example, for an RGB color space RGB, RGB, RGB, and so on.)

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

CGColorSpace.h

## CGColorSpaceCreateLab

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

```
CGColorSpaceRef CGColorSpaceCreateLab (
    const CGFloat whitePoint[3],
    const CGFloat blackPoint[3],
    const CGFloat range[4]
);
```

### Parameters

*whitePoint*

An array of 3 numbers that specify the tristimulus value, in the CIE 1931 XYZ-space, of the diffuse white point.

*blackPoint*

An array of 3 numbers that specify the tristimulus value, in CIE 1931 XYZ-space, of the diffuse black point.

*range*

An array of 4 numbers that specify the range of valid values for the a\* and b\* components of the color space. The a\* component represents values running from green to red, and the b\* component represents values running from blue to yellow.

**Return Value**

A new L\*a\*b\* color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

**Discussion**

The CIE L\*a\*b\* space is a nonlinear transformation of the Munsell color notation system (a system which specifies colors by hue, value, and saturation—or “chroma”—values), designed to match perceived color difference with quantitative distance in color space. The L\* component represents the lightness value, the a\* component represents values running from green to red, and the b\* component represents values running from blue to yellow. This roughly corresponds to the way the human brain is thought to decode colors. Colors in a device-independent color space should appear the same when displayed on different devices, to the extent that the capabilities of the device allow.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CGColorSpace.h

**CGColorSpaceCreatePattern**

Creates a pattern color space.

```
CGColorSpaceRef CGColorSpaceCreatePattern (
    CGColorSpaceRef baseSpace
);
```

**Parameters***baseSpace*

For masking patterns, the underlying color space that specifies the colors to be painted through the mask. For colored patterns, you should pass NULL.

**Return Value**

A new pattern color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

**Discussion**

For information on creating and using patterns, see *Quartz 2D Programming Guide* and *CGPattern Reference*. Quartz retains the color space you pass in. Upon return, you may safely release it by calling [CGColorSpaceRelease](#) (page 17).

**Availability**

Available in Mac OS X v10.1 and later.

**Declared In**

CGColorSpace.h

## CGColorSpaceCreateWithName

Creates a specified type of Quartz color space.

```
CGColorSpaceRef CGColorSpaceCreateWithName (
    CFStringRef name
);
```

### Parameters

*name*

A color space name. See “Color Space Names” (page 19) for a list of the valid Quartz-defined names.

### Return Value

A new generic color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

### Discussion

You can use this function to create a generic color space. For more information, see “Color Space Names” (page 19).

Prior to Mac OS X v10.4, you could pass this function one of the constants defined in “Named Color Spaces (Deprecated)” (page 22). As of Mac OS X v10.4, this function returns a generic color space even if you pass is one of the deprecated named color spaces.

### Availability

Available in Mac OS X v10.2 and later.

### Declared In

CGColorSpace.h

## CGColorSpaceCreateWithPlatformColorSpace

Creates a platform-specific color space.

```
CGColorSpaceRef CGColorSpaceCreateWithPlatformColorSpace (
    void *platformColorSpaceReference
);
```

### Parameters

*platformColorSpace*

A generic pointer to a platform-specific color space. In Mac OS X, pass a `CMProfileRef`—a ColorSync profile. Quartz uses this pointer (and the underlying information) only during the function call.

### Return Value

A new color space. You are responsible for releasing this object by calling [CGColorSpaceRelease](#) (page 17). If unsuccessful, returns NULL.

### Discussion

Colors in a device-dependent color space are not transformed or otherwise modified when displayed on an output device—that is, there is no attempt to maintain the visual appearance of a color. As a consequence, colors in a device color space often appear different when displayed on different output devices. For this reason, device color spaces are not recommended when color preservation is important.

### Availability

Available in Mac OS X v10.1 and later.

**Related Sample Code**

CarbonSketch

**Declared In**

CGColorSpace.h

**CGColorSpaceGetBaseColorSpace**

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

```
CGColorSpace CGColorSpaceGetBaseColorSpace(
    CGColorSpaceRef space
);
```

**Parameters***space*

A color space object for a pattern or indexed color space.

**Return Value**The base color space if the *space* parameter is a pattern or indexed color space; otherwise, NULL.**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CGColorSpace.h

**CGColorSpaceGetColorTable**

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

```
void CGColorSpaceGetColorTable(
    CGColorSpaceRef space,
    unsigned char *table);
);
```

**Parameters***space*

A color space object for an indexed color space.

*table*

The array pointed to by *table* should be at least as large as the number of entries in the color table. On output, the array contains the table data in the same format as that passed to [CGColorSpaceCreateIndexed](#) (page 12).

**Discussion**

This function does nothing if the color space is not an indexed color space. To determine whether a color space is an indexed color space, call the function [CGColorSpaceGetModel](#) (page 16).

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**[CGColorSpaceGetColorTableCount](#) (page 16)

**Declared In**

CGColorSpace.h

**CGColorSpaceGetColorTableCount**

Returns the number of entries in the color table of an indexed color space.

```
size_t CGColorSpaceGetColorTableCount(
    CGColorSpaceRef space
);
```

**Parameters***space*

A color space object for an indexed color space.

**Return Value**The number of entries in the color table of the *space* parameter if the color space is an indexed color space; otherwise, returns 0.**Availability**

Available in Mac OS X v10.5 and later.

**See Also**[CGColorSpaceGetColorTable](#) (page 15)**Declared In**

CGColorSpace.h

**CGColorSpaceGetModel**

Returns the color space model of the provided color space.

```
CGColorSpaceModel CGColorSpaceGetModel(
    CGColorSpaceRef space
);
```

**Parameters***space*

A color space object.

**Return Value**One of the constants described in [“Color Space Models”](#) (page 19).**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CGColorSpace.h

**CGColorSpaceGetNumberOfComponents**

Returns the number of color components in a color space.



```
size_t CGColorSpaceGetNumberOfComponents (
    CGColorSpaceRef cs
);
```

**Parameters***cs*

The Quartz color space to examine.

**Return Value**

The number of color components in the specified color space, not including the alpha value. For example, for an RGB color space, `CGColorSpaceGetNumberOfComponents` returns a value of 3.

**Discussion**

A color space defines an n-dimensional space whose dimensions (or components) represent intensity values. For example, you specify colors in RGB space as three intensity values: red, green, and blue. You can use the `CGColorSpaceGetNumberOfComponents` function to obtain the number of components in a given color space.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

`CGColorSpace.h`

**CGColorSpaceGetTypeID**

Returns the Core Foundation type identifier for Quartz color spaces.

```
CFTypeID CGColorSpaceGetTypeID (
    void
);
```

**Return Value**

The identifier for the opaque type [CGColorSpaceRef](#) (page 18).

**Availability**

Available in Mac OS X v10.2 and later.

**Declared In**

`CGColorSpace.h`

**CGColorSpaceRelease**

Decrements the retain count of a color space.

```
void CGColorSpaceRelease (
    CGColorSpaceRef cs
);
```

**Parameters***cs*

The Quartz color space to release.

**Discussion**

This function is equivalent to `CFRelease`, except that it does not cause an error if the `cs` parameter is `NULL`.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CGColorSpace.h

**CGColorSpaceRetain**

Increments the retain count of a color space.

```
CGColorSpaceRef CGColorSpaceRetain (  
    CGColorSpaceRef cs  
);
```

**Parameters**

*cs*

The Quartz color space to retain.

**Return Value**

The same color space you passed in as the *cs* parameter.

**Discussion**

This function is equivalent to `CFRetain`, except that it does not cause an error if the *cs* parameter is `NULL`.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CGColorSpace.h

## Data Types

**CGColorSpaceRef**

An opaque type that encapsulates color space information.

```
typedef struct CGColorSpace *CGColorSpaceRef;
```

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

CGColorSpace.h

## Constants

### Color Space Names

Convenience constants for commonly used color spaces.

```
CFStringRef kCGColorSpaceGenericGray
CFStringRef kCGColorSpaceGenericRGB
CFStringRef kCGColorSpaceGenericCMYK
CFStringRef kCGColorSpaceGenericRGBLinear
CFStringRef kCGColorSpaceAdobeRGB1998
CFStringRef kCGColorSpaceSRGB
```

#### Constants

```
kCGColorSpaceGenericGray
```

The name of the generic gray color space.

```
kCGColorSpaceGenericRGB
```

The name of the generic RGB color space.

```
kCGColorSpaceGenericCMYK
```

The name of the generic CMYK color space.

```
kCGColorSpaceGenericRGBLinear
```

The name of the generic linear RGB color space. This is the same as `kCGColorSpaceGenericRGB` (page 19), but with a gamma equal to 1.0.

```
kCGColorSpaceAdobeRGB1998
```

The name of the Adobe RGB (1998) color space. For more information, see "Adobe RGB (1998) Color Image Encoding", Version 2005-05, Adobe Systems Inc. (<http://www.adobe.com>).

```
kCGColorSpaceSRGB
```

The name of the sRGB color space.

#### Discussion

A color space name constant can be passed as a parameter to the function `CGColorSpaceCreateWithName` (page 14). These color spaces replace “Named Color Spaces (Deprecated)” (page 22), which are deprecated in Mac OS X v10.4.

#### Declared In

```
CGColorSpace.h
```

### Color Space Models

Models for color spaces.

```
enum CGColorSpaceModel {
    kCGColorSpaceModelUnknown = -1,
    kCGColorSpaceModelMonochrome,
    kCGColorSpaceModelRGB,
    kCGColorSpaceModelCMYK,
    kCGColorSpaceModelLab,
    kCGColorSpaceModelDeviceN,
    kCGColorSpaceModelIndexed,
    kCGColorSpaceModelPattern
};
typedef int32_t CGColorSpaceModel;
```

**Constants**

`kCGColorSpaceModelUnknown`

**An unknown color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelMonochrome`

**A monochrome color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelRGB`

**An RGB color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelCMYK`

**A CMYK color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelLab`

**A Lab color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelDeviceN`

**A DeviceN color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelIndexed`

**An indexed color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

`kCGColorSpaceModelPattern`

**A pattern color space model.**

Available in Mac OS X v10.5 and later.

Declared in `CGColorSpace.h`.

**Declared In**

CGColorSpace.h

**Color Rendering Intents**

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

```
enum CGColorRenderingIntent {
    kCGRenderingIntentDefault,
    kCGRenderingIntentAbsoluteColorimetric,
    kCGRenderingIntentRelativeColorimetric,
    kCGRenderingIntentPerceptual,
    kCGRenderingIntentSaturation
};
typedef enum CGColorRenderingIntent CGColorRenderingIntent;
```

**Constants**

kCGRenderingIntentDefault

The default rendering intent for the graphics context.

Available in Mac OS X v10.0 and later.

Declared in CGColorSpace.h.

kCGRenderingIntentAbsoluteColorimetric

Map colors outside of the gamut of the output device to the closest possible match inside the gamut of the output device. This can produce a clipping effect, where two different color values in the gamut of the graphics context are mapped to the same color value in the output device's gamut. Unlike the relative colorimetric, absolute colorimetric does not modify colors inside the gamut of the output device.

Available in Mac OS X v10.0 and later.

Declared in CGColorSpace.h.

kCGRenderingIntentRelativeColorimetric

Map colors outside of the gamut of the output device to the closest possible match inside the gamut of the output device. This can produce a clipping effect, where two different color values in the gamut of the graphics context are mapped to the same color value in the output device's gamut. The relative colorimetric shifts all colors (including those within the gamut) to account for the difference between the white point of the graphics context and the white point of the output device.

Available in Mac OS X v10.0 and later.

Declared in CGColorSpace.h.

kCGRenderingIntentPerceptual

Preserve the visual relationship between colors by compressing the gamut of the graphics context to fit inside the gamut of the output device. Perceptual intent is good for photographs and other complex, detailed images.

Available in Mac OS X v10.0 and later.

Declared in CGColorSpace.h.

kCGRenderingIntentSaturation

Preserve the relative saturation value of the colors when converting into the gamut of the output device. The result is an image with bright, saturated colors. Saturation intent is good for reproducing images with low detail, such as presentation charts and graphs.

Available in Mac OS X v10.0 and later.

Declared in CGColorSpace.h.

**Discussion**

The rendering intent specifies how Quartz should handle colors that are not located within the gamut of the destination color space of a graphics context. It determines the exact method used to map colors from one color space to another. If you do not explicitly set the rendering intent by calling the function `CGContextSetRenderingIntent`, the graphics context uses the relative colorimetric rendering intent, except when drawing sampled images.

**Declared In**

`CGColorSpace.h`

**Named Color Spaces (Deprecated)**

Color spaces used in the Preferences application.

```
#define kCGColorSpaceUserCMYK CFSTR("kCGColorSpaceUserCMYK")
#define kCGColorSpaceUserGray CFSTR("kCGColorSpaceUserGray")
#define kCGColorSpaceUserRGB CFSTR("kCGColorSpaceUserRGB")
```

**Constants**

`kCGColorSpaceUserCMYK`  
A user-defined CMYK color space.

`kCGColorSpaceUserGray`  
A user-defined gray color space.

`kCGColorSpaceUserRGB`  
A user-defined RGB color space.

**Discussion**

These constants are deprecated in Mac OS X v10.4. Instead use [“Color Space Names”](#) (page 19).

The named color spaces are user-configurable in the “Default Profiles for Documents” pane, located in Mac OS 10.2 in the ColorSync preference panel, and in Mac OS 10.3 in the Displays Color Preference panel. See also [`CGColorSpaceCreateWithName`](#) (page 14).

**Availability**

Available in Mac OS X v10.2 and later but deprecated in Mac OS X v10.4.

**Declared In**

`CGColorSpace.h`

# Document Revision History

This table describes the changes to *CGColorSpace Reference*.

| Date       | Notes  |
|------------|--|
| 2008-09-09 | Added information on color spaces and detailed gamma functions.  |
|            | See the Discussion for <a href="#">CGColorSpaceCreateICCBased</a> (page 11) and <a href="#">CGColorSpaceCreateCalibratedRGB</a> (page 8).  |
|            | Removed the constant <code>kCGColorSpaceGenericRGBHDR</code> .   |
| 2007-07-18 | Updated for Mac OS X v10.5.  |
|            | All instances of the <code>float</code> data type were changed to the <code>CGFloat</code> data type.  |
|            | Added <a href="#">CGColorSpaceCopyICCProfile</a> (page 7), <a href="#">CGColorSpaceGetModel</a> (page 16), <a href="#">CGColorSpaceGetBaseColorSpace</a> (page 15), <a href="#">CGColorSpaceGetColorTableCount</a> (page 16), and <a href="#">CGColorSpaceGetColorTable</a> (page 15). |
|            | Added the constants <code>kCGColorSpaceGenericRGBHDR</code> , <code>kCGColorSpaceAdobeRGB1998</code> , <a href="#">kCGColorSpaceGenericRGB</a> (page 19), and <a href="#">kCGColorSpaceSRGB</a> (page 19).   |
|            | Renamed “Generic Color Spaces” to “Color Space Names” (page 19).   |
|            | Added “Color Space Models” (page 19).  |
| 2005-04-29 | Updated for Mac OS X v10.4.  |
|            | Deprecated “Named Color Spaces (Deprecated)” (page 22) and added “Color Space Names” (page 19). Fixed wrong link in the function <a href="#">CGColorSpaceCreateWithName</a> (page 14).   |
| 2004-08-31 | Added introductory material.   |
| 2004-02-26 | First version of this document. An earlier version of this information appeared in <i>Quartz 2D Reference</i> .  |

## REVISION HISTORY

### Document Revision History



# Index

---

## C

---

CGColorSpaceCopyICCPProfile [function 7](#)  
CGColorSpaceCreateCalibratedGray [function 7](#)  
CGColorSpaceCreateCalibratedRGB [function 8](#)  
CGColorSpaceCreateDeviceCMYK [function 9](#)  
CGColorSpaceCreateDeviceGray [function 10](#)  
CGColorSpaceCreateDeviceRGB [function 10](#)  
CGColorSpaceCreateICCBased [function 11](#)  
CGColorSpaceCreateIndexed [function 12](#)  
CGColorSpaceCreateLab [function 12](#)  
CGColorSpaceCreatePattern [function 13](#)  
CGColorSpaceCreateWithName [function 14](#)  
CGColorSpaceCreateWithPlatformColorSpace [function 14](#)  
CGColorSpaceGetBaseColorSpace [function 15](#)  
CGColorSpaceGetColorTable [function 15](#)  
CGColorSpaceGetColorTableCount [function 16](#)  
CGColorSpaceGetModel [function 16](#)  
CGColorSpaceGetNumberOfComponents [function 16](#)  
CGColorSpaceGetTypeID [function 17](#)  
CGColorSpaceRef [data type 18](#)  
CGColorSpaceRelease [function 17](#)  
CGColorSpaceRetain [function 18](#)  
[Color Rendering Intents 21](#)  
[Color Space Models 19](#)  
[Color Space Names 19](#)

## K

---

kCGColorSpaceAdobeRGB1998 [constant 19](#)  
kCGColorSpaceGenericCMYK [constant 19](#)  
kCGColorSpaceGenericGray [constant 19](#)  
kCGColorSpaceGenericRGB [constant 19](#)  
kCGColorSpaceGenericRGBLinear [constant 19](#)  
kCGColorSpaceModelCMYK [constant 20](#)  
kCGColorSpaceModelDeviceN [constant 20](#)  
kCGColorSpaceModelIndexed [constant 20](#)  
kCGColorSpaceModelLab [constant 20](#)  
kCGColorSpaceModelMonochrome [constant 20](#)

kCGColorSpaceModelPattern [constant 20](#)  
kCGColorSpaceModelRGB [constant 20](#)  
kCGColorSpaceModelUnknown [constant 20](#)  
kCGColorSpaceSRGB [constant 19](#)  
kCGColorSpaceUserCMYK [constant 22](#)  
kCGColorSpaceUserGray [constant 22](#)  
kCGColorSpaceUserRGB [constant 22](#)  
kCGRenderingIntentAbsoluteColorimetric [constant 21](#)  
kCGRenderingIntentDefault [constant 21](#)  
kCGRenderingIntentPerceptual [constant 21](#)  
kCGRenderingIntentRelativeColorimetric [constant 21](#)  
kCGRenderingIntentSaturation [constant 21](#)

## N

---

[Named Color Spaces \(Deprecated\) 22](#)