

---

# NSColor Class Reference

Graphics & Animation: 2D Drawing



2009-01-06



Apple Inc.  
© 2009 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, Cocoa, iChat, Mac, Mac OS, Quartz, and Spaces are trademarks of Apple Inc., registered in the United States and other countries.

Aperture is a trademark of Apple Inc.

Java and all Java-based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

OpenGL is a registered trademark of Silicon Graphics, Inc.

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

## NSColor Class Reference 7

---

Class at a Glance	7
Overview	8
Adopted Protocols	8
Tasks	8
Creating an NSColor Object from Component Values	8
Creating an NSColor with Preset Components	9
Working with Pattern Images	10
Creating a System Color—an NSColor Whose Value Is Specified by User Preferences	10
Ignoring Alpha Components	12
Copying and Pasting	12
Retrieving a Set of Components	12
Retrieving Individual Components	12
Working with the Color Space	13
Changing the Color	14
Drawing	14
Class Methods	14
alternateSelectedControlColor	14
alternateSelectedControlTextColor	15
blackColor	15
blueColor	16
brownColor	16
clearColor	16
colorForControlTint:	17
colorFromPasteboard:	17
colorWithCalibratedHue:saturation:brightness:alpha:	18
colorWithCalibratedRed:green:blue:alpha:	18
colorWithCalibratedWhite:alpha:	19
colorWithCatalogName:colorName:	20
colorWithCGColor:	20
colorWithColorSpace:components:count:	21
colorWithDeviceCyan:magenta:yellow:black:alpha:	22
colorWithDeviceHue:saturation:brightness:alpha:	22
colorWithDeviceRed:green:blue:alpha:	23
colorWithDeviceWhite:alpha:	24
colorWithPatternImage:	25
controlAlternatingRowBackgroundColors	25
controlBackgroundColor	25
controlColor	26
controlDarkShadowColor	26
controlHighlightColor	27

controlLightHighlightColor	27
controlShadowColor	27
controlTextColor	28
currentControlTint	28
cyanColor	29
darkGrayColor	29
disabledControlTextColor	29
grayColor	30
greenColor	30
gridColor	31
headerColor	31
headerTextColor	32
highlightColor	32
ignoresAlpha	32
keyboardFocusIndicatorColor	33
knobColor	33
lightGrayColor	34
magentaColor	34
orangeColor	34
purpleColor	35
redColor	35
scrollBarColor	36
secondarySelectedControlColor	36
selectedControlColor	36
selectedControlTextColor	37
selectedKnobColor	37
selectedMenuItemColor	38
selectedMenuItemTextColor	38
selectedTextBackgroundColor	38
selectedTextColor	39
setIgnoresAlpha:	39
shadowColor	40
textBackgroundColor	40
textColor	41
whiteColor	41
windowBackgroundColor	42
windowFrameColor	42
windowFrameTextColor	42
yellowColor	43
Instance Methods	43
alphaComponent	43
blackComponent	44
blendedColorWithFraction:ofColor:	44
blueComponent	45
brightnessComponent	45
catalogNameComponent	46

- colorNameComponent 46
- colorSpace 47
- colorSpaceName 47
- colorUsingColorSpace: 48
- colorUsingColorSpaceName: 48
- colorUsingColorSpaceName:device: 49
- colorWithAlphaComponent: 49
- cyanComponent 50
- drawSwatchInRect: 50
- getComponents: 51
- getCyan:magenta:yellow:black:alpha: 51
- getHue:saturation:brightness:alpha: 52
- getRed:green:blue:alpha: 53
- getWhite:alpha: 54
- greenComponent 54
- highlightWithLevel: 55
- hueComponent 55
- localizedCatalogNameComponent 56
- localizedColorNameComponent 56
- magentaComponent 56
- numberOfComponents 57
- patternImage 57
- redComponent 58
- saturationComponent 58
- set 59
- setFill 59
- setStroke 59
- shadowWithLevel: 60
- whiteComponent 60
- writeToPasteboard: 61
- yellowComponent 61
- Constants 62
  - AppKit Versions for NSColor Bug Fixes 62
- Notifications 62
  - NSSystemColorsDidChangeNotification 62

---

**Document Revision History 63**

---

**Index 65**

---



# NSColor Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding NSCopying NSPasteboardWriting NSPasteboardReading NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Color Programming Topics for Cocoa
<b>Declared in</b>	NSColor.h
<b>Related sample code</b>	Cocoa OpenGL From A View to A Movie From A View to A Picture Quartz Composer WWDC 2005 TextEdit Sketch-112

## Class at a Glance

An `NSColor` object represents a color, which is defined in a color space, each point of which has a set of components (such as red, green, and blue) that uniquely define a color.

## Principal Attributes

---

- Color space
- Color components

Various `colorWith...` and `colorUsing...` methods.

Preset colors:  `blackColor` (page 15),  `blueColor` (page 16), and so on.

## Commonly Used Methods

---

[colorUsingColorSpaceName:](#) (page 48)

Creates an `NSColor` instance in the specified color space.

[set](#) (page 59)

Sets the drawing color.

## Overview

An `NSColor` object represents color and sometimes opacity (alpha). By sending a [set](#) (page 59) message to an `NSColor` instance, you set the color for the current drawing context. Setting the color causes subsequently drawn graphics to have the color represented by the `NSColor` instance.

It is invalid to use an accessor method related to components of a particular color space on an `NSColor` object that is not in that color space. For example, methods such as [redComponent](#) (page 58) and [getRed:green:blue:alpha:](#) (page 53) work on color objects in the calibrated and device RGB color spaces. If you send such a message to an `NSColor` object in the CMYK color space, an exception is raised. Further, the methods [getComponents:](#) (page 51) and [numberOfComponents](#) (page 57) work in color spaces that have individual components. Thus they return the components of `NSColor` objects as individual floating-point values regardless of whether they're based on `NSColorSpace` objects or named color spaces. However, older component-fetching methods such as `redComponent` and `getRed:green:blue:alpha:` are only effective on `NSColor` objects based on named color spaces.

If you have an `NSColor` object in an unknown color space and you want to extract its components, you should first convert the color object to a known color space before using the component accessor methods of that color space.

## Adopted Protocols

### NSCoding

`encodeWithCoder:`

`initWithCoder:`

### NSCopying

`copyWithZone:`

## Tasks

### Creating an `NSColor` Object from Component Values

+ [colorWithCalibratedHue:saturation:brightness:alpha:](#) (page 18)

Creates and returns an `NSColor` object using the given opacity and HSB color space components.

- + [colorWithCalibratedRed:green:blue:alpha:](#) (page 18)  
Creates and returns an NSColor object using the given opacity and RGB components.
- + [colorWithCalibratedWhite:alpha:](#) (page 19)  
Creates and returns an NSColor object using the given opacity and grayscale value.
- + [colorWithCatalogName:colorName:](#) (page 20)  
Creates and returns an NSColor object by finding the color with the specified name in the given catalog.
- + [colorWithDeviceCyan:magenta:yellow:black:alpha:](#) (page 22)  
Creates and returns an NSColor object using the given opacity value and CMYK components.
- + [colorWithDeviceHue:saturation:brightness:alpha:](#) (page 22)  
Creates and returns an NSColor object using the given opacity value and HSB color space components.
- + [colorWithDeviceRed:green:blue:alpha:](#) (page 23)  
Creates and returns an NSColor object using the given opacity value and RGB components.
- + [colorWithDeviceWhite:alpha:](#) (page 24)  
Creates and returns an NSColor object using the given opacity and grayscale values.
- + [colorWithCGColor:](#) (page 20)  
Converts a Core Image color object to its NSColor equivalent.
- + [colorWithColorSpace:components:count:](#) (page 21)  
Returns an NSColor object created from the specified components of the given color space.

## Creating an NSColor with Preset Components

- + [blackColor](#) (page 15)  
Returns an NSColor object whose grayscale value is 0.0 and whose alpha value is 1.0.
- + [blueColor](#) (page 16)  
Returns an NSColor object whose RGB value is 0.0, 0.0, 1.0 and whose alpha value is 1.0.
- + [brownColor](#) (page 16)  
Returns an NSColor object whose RGB value is 0.6, 0.4, 0.2 and whose alpha value is 1.0.
- + [clearColor](#) (page 16)  
Returns an NSColor object whose grayscale and alpha values are both 0.0.
- + [cyanColor](#) (page 29)  
Returns an NSColor object whose RGB value is 0.0, 1.0, 1.0 and whose alpha value is 1.0.
- + [darkGrayColor](#) (page 29)  
Returns an NSColor object whose grayscale value is 1/3 and whose alpha value is 1.0.
- + [grayColor](#) (page 30)  
Returns an NSColor object whose grayscale value is 0.5 and whose alpha value is 1.0.
- + [greenColor](#) (page 30)  
Returns an NSColor object whose RGB value is 0.0, 1.0, 0.0 and whose alpha value is 1.0.
- + [lightGrayColor](#) (page 34)  
Returns an NSColor object whose grayscale value is 2/3 and whose alpha value is 1.0.
- + [magentaColor](#) (page 34)  
Returns an NSColor object whose RGB value is 1.0, 0.0, 1.0 and whose alpha value is 1.0.
- + [orangeColor](#) (page 34)  
Returns an NSColor object whose RGB value is 1.0, 0.5, 0.0 and whose alpha value is 1.0.

- + [purpleColor](#) (page 35)  
Returns an `NSColor` object whose RGB value is 0.5, 0.0, 0.5 and whose alpha value is 1.0.
- + [redColor](#) (page 35)  
Returns an `NSColor` object whose RGB value is 1.0, 0.0, 0.0 and whose alpha value is 1.0.
- + [whiteColor](#) (page 41)  
Returns an `NSColor` object whose grayscale and alpha values are both 1.0.
- + [yellowColor](#) (page 43)  
Returns an `NSColor` object whose RGB value is 1.0, 1.0, 0.0 and whose alpha value is 1.0.

## Working with Pattern Images

- + [colorWithPatternImage:](#) (page 25)  
Creates and returns an `NSColor` object that uses the specified image pattern.
- [patternImage](#) (page 57)  
Returns the image that the receiver is using as a pattern.

## Creating a System Color—an NSColor Whose Value Is Specified by User Preferences

- + [alternateSelectedControlColor](#) (page 14)  
Returns the system color used for the face of a selected control.
- + [alternateSelectedControlTextColor](#) (page 15)  
Returns the system color used for text in a selected control.
- + [colorForControlTint:](#) (page 17)  
Returns the `NSColor` object specified by the given control tint.
- + [controlBackgroundColor](#) (page 25)  
Returns the system color used for the background of large controls.
- + [controlColor](#) (page 26)  
Returns the system color used for the flat surfaces of a control.
- + [controlAlternatingRowBackgroundColors](#) (page 25)  
Returns an array containing the system specified background colors for alternating rows in tables and lists.
- + [controlHighlightColor](#) (page 27)  
Returns the system color used for the highlighted bezels of controls.
- + [controlLightHighlightColor](#) (page 27)  
Returns the system color used for light highlights in controls.
- + [controlShadowColor](#) (page 27)  
Returns the system color used for the shadows dropped from controls.
- + [controlDarkShadowColor](#) (page 26)  
Returns the system color used for the dark edge of the shadow dropped from controls.
- + [controlTextColor](#) (page 28)  
Returns the system color used for text on controls that aren't disabled.

- + [currentControlTint](#) (page 28)  
Returns the current system control tint.
- + [disabledControlTextColor](#) (page 29)  
Returns the system color used for text on disabled controls.
- + [gridColor](#) (page 31)  
Returns the system color used for the optional gridlines in, for example, a table view.
- + [headerColor](#) (page 31)  
Returns the system color used as the background color for header cells in table views and outline views.
- + [headerTextColor](#) (page 32)  
Returns the system color used for text in header cells in table views and outline views.
- + [highlightColor](#) (page 32)  
Returns the system color that represents the virtual light source on the screen.
- + [keyboardFocusIndicatorColor](#) (page 33)  
Returns the system color that represents the keyboard focus ring around controls.
- + [knobColor](#) (page 33)  
Returns the system color used for the flat surface of a slider knob that hasn't been selected.
- + [scrollBarColor](#) (page 36)  
Returns the system color used for scroll "bars"—that is, for the groove in which a scroller's knob moves.
- + [secondarySelectedControlColor](#) (page 36)  
Returns the system color used in non-key views.
- + [selectedControlColor](#) (page 36)  
Returns the system color used for the face of a selected control.
- + [selectedControlTextColor](#) (page 37)  
Returns the system color used for text in a selected control—a control being clicked or dragged.
- + [selectedMenuItemColor](#) (page 38)  
Returns the system color used for the face of selected menu items.
- + [selectedMenuItemTextColor](#) (page 38)  
Returns the system color used for the text in menu items.
- + [selectedTextBackgroundColor](#) (page 38)  
Returns the system color used for the background of selected text.
- + [selectedTextColor](#) (page 39)  
Returns the system color used for selected text.
- + [selectedKnobColor](#) (page 37)  
Returns the system color used for the slider knob when it is selected.
- + [shadowColor](#) (page 40)  
Returns the system color that represents the virtual shadows cast by raised objects on the screen.
- + [textBackgroundColor](#) (page 40)  
Returns the system color used for the text background.
- + [textColor](#) (page 41)  
Returns the system color used for text.
- + [windowBackgroundColor](#) (page 42)  
Returns a pattern color that will draw the ruled lines for the window background.

- + [windowFrameColor](#) (page 42)  
Returns the system color used for window frames, except for their text.
- + [windowFrameTextColor](#) (page 42)  
Returns the system color used for the text in window frames.

## Ignoring Alpha Components

- + [ignoresAlpha](#) (page 32)  
Returns a Boolean value indicating whether the application supports alpha.
- + [setIgnoresAlpha:](#) (page 39)  
Specifies whether an application supports alpha.

## Copying and Pasting

- + [colorFromPasteboard:](#) (page 17)  
Returns the `NSColor` currently on the given pasteboard.
- [writeToPasteboard:](#) (page 61)  
Writes the receiver's data to the specified pasteboard.

## Retrieving a Set of Components

- [getCyan:magenta:yellow:black:alpha:](#) (page 51)  
Returns the receiver's CMYK and opacity values.
- [getHue:saturation:brightness:alpha:](#) (page 52)  
Returns the receiver's HSB component and opacity values in the respective arguments.
- [getRed:green:blue:alpha:](#) (page 53)  
Returns the receiver's RGB component and opacity values in the respective arguments.
- [getWhite:alpha:](#) (page 54)  
Returns the receiver's grayscale value and alpha values.
- [getComponents:](#) (page 51)  
Returns the components of the receiver as an array.
- [numberOfComponents](#) (page 57)  
Returns the number of components in the receiver.

## Retrieving Individual Components

- [alphaComponent](#) (page 43)  
Returns the receiver's alpha (opacity) component.
- [blackComponent](#) (page 44)  
Returns the receiver's black component.
- [blueComponent](#) (page 45)  
Returns the receiver's blue component.

- [brightnessComponent](#) (page 45)  
Returns the brightness component of the HSB color equivalent to the receiver.
- [catalogNameComponent](#) (page 46)  
Returns the name of the catalog containing the receiver's name.
- [colorNameComponent](#) (page 46)  
Returns the receiver's name.
- [cyanComponent](#) (page 50)  
Returns the receiver's cyan component.
- [greenComponent](#) (page 54)  
Returns the receiver's green component.
- [hueComponent](#) (page 55)  
Returns the hue component of the HSB color equivalent to the receiver.
- [localizedCatalogNameComponent](#) (page 56)  
Returns the name of the catalog containing the receiver's name as a localized string.
- [localizedColorNameComponent](#) (page 56)  
Returns the name of the receiver as a localized string.
- [magentaComponent](#) (page 56)  
Returns the receiver's magenta component.
- [redComponent](#) (page 58)  
Returns the receiver's red component.
- [saturationComponent](#) (page 58)  
Returns the saturation component of the HSB color equivalent to the receiver.
- [whiteComponent](#) (page 60)  
Returns the receiver's white component.
- [yellowComponent](#) (page 61)  
Returns the receiver's yellow component.

## Working with the Color Space

- [colorSpaceName](#) (page 47)  
Returns the name of the receiver's color space.
- [colorUsingColorSpaceName:](#) (page 48)  
Creates and returns an `NSColor` whose color is the same as the receiver's, except that the new `NSColor` is in the specified color space.
- [colorUsingColorSpaceName:device:](#) (page 49)  
Creates and returns an `NSColor` object whose color is the same as the receiver's, except that the new `NSColor` is in the given color space and is specific to the given device.
- [colorSpace](#) (page 47)  
Returns an object representing the color space of the receiver.
- [colorUsingColorSpace:](#) (page 48)  
Returns a new color object representing the color of the receiver in the specified color space.

## Changing the Color

- [blendedColorWithFraction:ofColor:](#) (page 44)  
Creates and returns an `NSColor` object whose component values are a weighted sum of the receiver's and the specified color object's.
- [colorWithAlphaComponent:](#) (page 49)  
Creates and returns an `NSColor` object that has the same color space and component values as the receiver, but the specified alpha component.
- [highlightWithLevel:](#) (page 55)  
Returns an `NSColor` object that represents a blend between the receiver and the highlight color returned by [highlightColor](#) (page 32).
- [shadowWithLevel:](#) (page 60)  
Returns an `NSColor` object that represents a blend between the receiver and the shadow color returned by [shadowColor](#) (page 40).

## Drawing

- [drawSwatchInRect:](#) (page 50)  
Draws the current color in the given rectangle.
- [set](#) (page 59)  
Sets the color of subsequent drawing to the color that the receiver represents.
- [setFill](#) (page 59)  
Sets the fill color of subsequent drawing to the receiver's color.
- [setStroke](#) (page 59)  
Sets the stroke color of subsequent drawing to the receiver's color.

## Class Methods

### alternateSelectedControlColor

Returns the system color used for the face of a selected control.

```
+ (NSColor *)alternateSelectedControlColor
```

#### Return Value

The system color used for the face of a selected control—a control being clicked or dragged. This color can be used where iApp-like highlighting is desired. For general information about system colors, see [Accessing System Colors](#).

#### Availability

Available in Mac OS X v10.2 and later.

#### See Also

- + [alternateSelectedControlTextColor](#) (page 15)
- + [selectedControlColor](#) (page 36)

**Declared In**

NSColor.h

**alternateSelectedControlTextColor**

Returns the system color used for text in a selected control.

+ (NSColor \*)alternateSelectedControlTextColor

**Return Value**

The system color used for text in a selected control—a control being clicked or dragged. This color can be used where iApp-like highlighting is desired. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.2 and later.

**See Also**+ [alternateSelectedControlColor](#) (page 14)+ [selectedControlTextColor](#) (page 37)**Related Sample Code**

PhotoSearch

**Declared In**

NSColor.h

**blackColor**

Returns an NSColor object whose grayscale value is 0.0 and whose alpha value is 1.0.

+ (NSColor \*)blackColor

**Return Value**

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [blackComponent](#) (page 44)**Related Sample Code**

DockTile

PDF Annotation Editor

SpeedometerView

WebKitPluginStarter

WebKitPluginWithJavaScript

**Declared In**

NSColor.h

## blueColor

Returns an `NSColor` object whose RGB value is 0.0, 0.0, 1.0 and whose alpha value is 1.0.

```
+ (NSColor *)blueColor
```

### Return Value

The `NSColor` object.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [blueComponent](#) (page 45)

### Related Sample Code

`CocoaVideoFrameToGWorld`

`FunkyOverlayWindow`

`Grady`

`Rulers`

`VertexPerformanceTest`

### Declared In

`NSColor.h`

## brownColor

Returns an `NSColor` object whose RGB value is 0.6, 0.4, 0.2 and whose alpha value is 1.0.

```
+ (NSColor *)brownColor
```

### Return Value

The `NSColor` object.

### Availability

Available in Mac OS X v10.0 and later.

### Related Sample Code

`StickiesWithCoreData`

### Declared In

`NSColor.h`

## clearColor

Returns an `NSColor` object whose grayscale and alpha values are both 0.0.

```
+ (NSColor *)clearColor
```

### Return Value

The `NSColor` object.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

[AnimatedTableView](#)

[CompositeLab](#)

[FunkyOverlayWindow](#)

[Sketch-112](#)

[TrackBall](#)

**Declared In**

NSColor.h

**colorForControlTint:**

Returns the NSColor object specified by the given control tint.

```
+ (NSColor *)colorForControlTint:(NSControlTint)controlTint
```

**Parameters**

*controlTint*

The control tint for which to return an NSColor object. This is one of the tint settings. For more on control tints, see [Using the System Control Tint](#).

**Return Value**

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [currentControlTint](#) (page 28)

**Declared In**

NSColor.h

**colorFromPasteboard:**

Returns the NSColor currently on the given pasteboard.

```
+ (NSColor *)colorFromPasteboard:(NSPasteboard *)pasteBoard
```

**Parameters**

*pasteBoard*

The pasteboard from which to return the color.

**Return Value**

The color currently on the pasteboard or *nil* if *pasteBoard* doesn't contain color data. The returned color's alpha component is set to 1.0 if [ignoresAlpha](#) (page 32) returns YES.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [writeToPasteboard:](#) (page 61)

**Related Sample Code**

CompositeLab

Sketch+Accessibility

Sketch-112

**Declared In**

NSColor.h

**colorWithCalibratedHue:saturation:brightness:alpha:**

Creates and returns an `NSColor` object using the given opacity and HSB color space components.

```
+ (NSColor *) colorWithCalibratedHue:(CGFloat)hue saturation:(CGFloat)saturation
  brightness:(CGFloat)brightness alpha:(CGFloat)alpha
```

**Parameters**

*hue*

The hue component of the color object in the HSB color space.

*saturation*

The saturation component of the color object in the HSB color space.

*brightness*

The brightness (or value) component of the color object in the HSB color space.

*alpha*

The opacity value of the color object,

**Return Value**

The color object.

**Discussion**

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [colorWithCalibratedRed:green:blue:alpha:](#) (page 18)

+ [colorWithDeviceHue:saturation:brightness:alpha:](#) (page 22)

- [getHue:saturation:brightness:alpha:](#) (page 52)

**Related Sample Code**

NewsReader

**Declared In**

NSColor.h

**colorWithCalibratedRed:green:blue:alpha:**

Creates and returns an `NSColor` object using the given opacity and RGB components.

```
+ (NSColor *)colorWithCalibratedRed:(CGFloat)red green:(CGFloat)green
  blue:(CGFloat)blue alpha:(CGFloat)alpha
```

**Parameters***red*

The red component of the color object.

*green*

The green component of the color object.

*blue*

The blue component of the color object.

*alpha*

The opacity value of the color object.

**Return Value**

The color object.

**Discussion**

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**+ [colorWithCalibratedHue:saturation:brightness:alpha:](#) (page 18)+ [colorWithDeviceRed:green:blue:alpha:](#) (page 23)- [getRed:green:blue:alpha:](#) (page 53)**Related Sample Code**

Color Sampler

GLChildWindowDemo

ImageApp

ImageKitDemo

WhackedTV

**Declared In**

NSColor.h

**colorWithCalibratedWhite:alpha:**

Creates and returns an NSColor object using the given opacity and grayscale value.

```
+ (NSColor *)colorWithCalibratedWhite:(CGFloat)white alpha:(CGFloat)alpha
```

**Parameters***white*

The grayscale value of the color object.

*alpha*

The opacity value of the color object.

**Return Value**

The color object.

**Discussion**

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + [colorWithDeviceWhite:alpha:](#) (page 24)
- [getWhite:alpha:](#) (page 54)

**Related Sample Code**

Clock Control  
CocoaSlides  
MatrixMixerTest  
SampleRaster  
WhackedTV

**Declared In**

NSColor.h

**colorWithCatalogName:colorName:**

Creates and returns an NSColor object by finding the color with the specified name in the given catalog.

```
+ (NSColor *) colorWithCatalogName:(NSString *)listName colorName:(NSString *)colorName
```

**Parameters**

*listName*

The name of the catalog in which to find the specified color; this may be a standard catalog.

*colorName*

The name of the color. Note that the color must be defined in the named color space to retrieve it with this method.

**Return Value**

The color object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [catalogNameComponent](#) (page 46)
- [colorNameComponent](#) (page 46)
- [localizedCatalogNameComponent](#) (page 56)

**Declared In**

NSColor.h

**colorWithCGColor:**

Converts a Core Image color object to its NSColor equivalent.

```
+ (NSColor *)colorWithCIColor:(CIColor *)color
```

**Parameters**

*color*

The Core Image color to convert.

**Return Value**

The `NSColor` object corresponding to the specified Core Image color.

**Discussion**

The method raises if the color space and components associated with *color* are `nil` or invalid.

**Availability**

Available in Mac OS X v10.4 and later.

**Related Sample Code**

CIColorTracking

FunHouse

**Declared In**

NSColor.h

**colorWithColorSpace:components:count:**

Returns an `NSColor` object created from the specified components of the given color space.

```
+ (NSColor *)colorWithColorSpace:(NSColorSpace *)space components:(const CGFloat *)components count:(NSInteger)numberOfComponents
```

**Parameters**

*space*

An `NSColorSpace` object representing a color space. The method raises if this is `nil`.

*components*

An array of the components in the specified color space to use to create the `NSColor` object. The order of these components is determined by the color-space profile, with the alpha component always last. (If you want the created color to be opaque, specify 1.0 for the alpha component.)

*numberOfComponents*

The number of components in the `components` array. This should match the number dictated by the specified color space plus one for alpha. This method raises an exception if they do not match.

**Return Value**

The color object. If *space* represents a color space that cannot be used with `NSColor` objects—for example, a “pattern” color space—the method returns `nil`.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

– [colorUsingColorSpace:](#) (page 48)

**Related Sample Code**

AnimatedTableView

**Declared In**  
NSColor.h

## colorWithDeviceCyan:magenta:yellow:black:alpha:

Creates and returns an NSColor object using the given opacity value and CMYK components.

```
+ (NSColor *)colorWithDeviceCyan:(CGFloat)cyan magenta:(CGFloat)magenta
  yellow:(CGFloat)yellow black:(CGFloat)black alpha:(CGFloat)alpha
```

### Parameters

*cyan*

The cyan component of the color object.

*magenta*

The magenta component of the color object.

*yellow*

The yellow component of the color object.

*black*

The black component of the color object.

*alpha*

The opacity value of the color object.

### Return Value

The color object.

### Discussion

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0. In PostScript, this color space corresponds directly to the device-dependent operator `setcmykcolor`.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [getCyan:magenta:yellow:black:alpha:](#) (page 51)

**Declared In**  
NSColor.h

## colorWithDeviceHue:saturation:brightness:alpha:

Creates and returns an NSColor object using the given opacity value and HSB color space components.

```
+ (NSColor *)colorWithDeviceHue:(CGFloat)hue saturation:(CGFloat)saturation
  brightness:(CGFloat)brightness alpha:(CGFloat)alpha
```

### Parameters

*hue*

The hue component of the color object.

*saturation*

The saturation component of the color object.

*brightness*

The brightness component of the color object.

*alpha*

The opacity value of the color object.

#### Return Value

The color object.

#### Discussion

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0. In PostScript, this color space corresponds directly to the device-dependent operator `setrgbcolor`.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- + [colorWithCalibratedHue:saturation:brightness:alpha:](#) (page 18)
- + [colorWithDeviceRed:green:blue:alpha:](#) (page 23)
- [getHue:saturation:brightness:alpha:](#) (page 52)

#### Related Sample Code

MenuItemView

#### Declared In

NSColor.h

## colorWithDeviceRed:green:blue:alpha:

Creates and returns an `NSColor` object using the given opacity value and RGB components.

```
+ (NSColor *) colorWithDeviceRed:(CGFloat)red green:(CGFloat)green blue:(CGFloat)blue
  alpha:(CGFloat)alpha
```

#### Parameters

*red*

The red component of the color object.

*green*

The green component of the color object.

*blue*

The blue component of the color object.

*alpha*

The opacity value of the color object.

#### Return Value

The color object.

#### Discussion

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0. In PostScript, this color space corresponds directly to the device-dependent operator `setrgbcolor`.

#### Availability

Available in Mac OS X v10.0 and later.

**See Also**

- + [colorWithCalibratedRed:green:blue:alpha:](#) (page 18)
- + [colorWithDeviceHue:saturation:brightness:alpha:](#) (page 22)
- [getRed:green:blue:alpha:](#) (page 53)

**Related Sample Code**

Cocoa OpenGL  
 From A View to A Movie  
 From A View to A Picture  
 IconCollection  
 OpenGL Screensaver

**Declared In**

NSColor.h

**colorWithDeviceWhite:alpha:**

Creates and returns an NSColor object using the given opacity and grayscale values.

```
+ (NSColor *) colorWithDeviceWhite:(CGFloat)white alpha:(CGFloat)alpha
```

**Parameters**

*white*

The grayscale value of the color object.

*alpha*

The opacity value of the color object.

**Return Value**

The color object.

**Discussion**

Values below 0.0 are interpreted as 0.0, and values above 1.0 are interpreted as 1.0. In PostScript, this color space corresponds directly to the device-dependent operator `setgray`.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + [colorWithCalibratedWhite:alpha:](#) (page 19)
- [getWhite:alpha:](#) (page 54)

**Related Sample Code**

FilterDemo  
 Link Snoop  
 PDFKitLinker2

**Declared In**

NSColor.h

## colorWithPatternImage:

Creates and returns an `NSColor` object that uses the specified image pattern.

```
+ (NSColor *) colorWithPatternImage:(NSImage *) image
```

### Parameters

*image*

The image to use as the pattern for the color object. The image is tiled starting at the bottom of the window. The image is not scaled.

### Return Value

The `NSColor` object. This color object is autoreleased.

### Availability

Available in Mac OS X v10.0 and later.

### Related Sample Code

Dicey

FilterDemo

Worm

### Declared In

`NSColor.h`

## controlAlternatingRowBackgroundColors

Returns an array containing the system specified background colors for alternating rows in tables and lists.

```
+ (NSArray *) controlAlternatingRowBackgroundColors
```

### Return Value

An array of `NSColor` objects specifying the system colors used for rows in tables and lists. You should not assume the array will contain only two colors. For general information on system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

`NSColor.h`

## controlBackgroundColor

Returns the system color used for the background of large controls.

```
+ (NSColor *) controlBackgroundColor
```

### Return Value

The system color used for the background of large controls such as browsers, table views, and clip views. For general information on system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

**Declared In**

NSColor.h

**controlColor**

Returns the system color used for the flat surfaces of a control.

```
+ (NSColor *)controlColor
```

**Return Value**

The system color used for the flat surfaces of a control. By default, the control color is a pattern color that will draw the ruled lines for the window background, which is the same as returned by [windowBackgroundColor](#) (page 42).

If you use `controlColor` assuming that it is a solid, you may have an incorrect appearance. You should use [lightGrayColor](#) (page 34) in its place.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

Quartz Composer WWDC 2005 TextEdit

**Declared In**

NSColor.h

**controlDarkShadowColor**

Returns the system color used for the dark edge of the shadow dropped from controls.

```
+ (NSColor *)controlDarkShadowColor
```

**Return Value**

Of the two dark borders that run along the bottom and right of controls, representing shadows, the color of the outer, darker border. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [controlShadowColor](#) (page 27)

**Related Sample Code**

QuickLookSketch

Sketch+Accessibility

Sketch-112

**Declared In**

NSColor.h

## controlHighlightColor

Returns the system color used for the highlighted bezels of controls.

```
+ (NSColor *)controlHighlightColor
```

### Return Value

Of the two light borders that run along the top and left of controls, representing reflections from a light source in the upper left, the color of the inner, duller border. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [controlLightHighlightColor](#) (page 27)

### Related Sample Code

DictionaryController

### Declared In

NSColor.h

## controlLightHighlightColor

Returns the system color used for light highlights in controls.

```
+ (NSColor *)controlLightHighlightColor
```

### Return Value

Of the two light borders that run along the top and left of controls, representing reflections from a light source in the upper left, the color of the outer, brighter border. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [controlHighlightColor](#) (page 27)

### Declared In

NSColor.h

## controlShadowColor

Returns the system color used for the shadows dropped from controls.

```
+ (NSColor *)controlShadowColor
```

### Return Value

Of the two dark borders that run along the bottom and right of controls, representing shadows, the color of the inner, lighter border. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [controlDarkShadowColor](#) (page 26)

**Declared In**

NSColor.h

**controlTextColor**

Returns the system color used for text on controls that aren't disabled.

```
+ (NSColor *)controlTextColor
```

**Return Value**

The color used for text on enabled controls. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [disabledControlTextColor](#) (page 29)

**Related Sample Code**

NewsReader

TextSizingExample

**Declared In**

NSColor.h

**currentControlTint**

Returns the current system control tint.

```
+ (NSControlTint)currentControlTint
```

**Return Value**

The current system control tint.

**Discussion**

An application can register for the `NSControlTintDidChangeNotification` notification to be notified of changes to the system control tint.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

+ [colorForControlTint:](#) (page 17)

**Declared In**

NSColor.h

## **cyanColor**

Returns an `NSColor` object whose RGB value is 0.0, 1.0, 1.0 and whose alpha value is 1.0.

```
+ (NSColor *)cyanColor
```

### **Return Value**

The `NSColor` object.

### **Availability**

Available in Mac OS X v10.0 and later.

### **See Also**

- [cyanComponent](#) (page 50)

### **Related Sample Code**

LayoutManagerDemo

RGB Image

### **Declared In**

`NSColor.h`

## **darkGrayColor**

Returns an `NSColor` object whose grayscale value is 1/3 and whose alpha value is 1.0.

```
+ (NSColor *)darkGrayColor
```

### **Return Value**

The `NSColor` object.

### **Availability**

Available in Mac OS X v10.0 and later.

### **See Also**

+ [lightGrayColor](#) (page 34)

+ [grayColor](#) (page 30)

### **Related Sample Code**

iChatTheater

QuickLookDownloader

TextLinks

TrackIt

### **Declared In**

`NSColor.h`

## **disabledControlTextColor**

Returns the system color used for text on disabled controls.

```
+ (NSColor *)disabledControlTextColor
```

### Return Value

The color used for text on disabled controls. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [controlTextColor](#) (page 28)

### Related Sample Code

NewsReader

### Declared In

NSColor.h

## grayColor

Returns an NSColor object whose grayscale value is 0.5 and whose alpha value is 1.0.

```
+ (NSColor *)grayColor
```

### Return Value

The NSColor object.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [lightGrayColor](#) (page 34)

+ [darkGrayColor](#) (page 29)

### Related Sample Code

ImageMap

ImageMapExample

MenuItemView

PhotoSearch

URL CacheInfo

### Declared In

NSColor.h

## greenColor

Returns an NSColor object whose RGB value is 0.0, 1.0, 0.0 and whose alpha value is 1.0.

```
+ (NSColor *)greenColor
```

### Return Value

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [greenComponent](#) (page 54)

**Related Sample Code**

MatrixMixerTest

TargetGallery

WebKitPluginStarter

WebKitPluginWithJavaScript

WhackedTV

**Declared In**

NSColor.h

## gridColor

Returns the system color used for the optional gridlines in, for example, a table view.

```
+ (NSColor *)gridColor
```

**Return Value**

The system color used for gridlines. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

IconCollection

**Declared In**

NSColor.h

## headerColor

Returns the system color used as the background color for header cells in table views and outline views.

```
+ (NSColor *)headerColor
```

**Return Value**

The system color used as the background for header cells in table and outline views. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSColor.h

## headerTextColor

Returns the system color used for text in header cells in table views and outline views.

```
+ (NSColor *)headerTextColor
```

### Return Value

The system color used for text in header cells in table and outline views. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

NSColor.h

## highlightColor

Returns the system color that represents the virtual light source on the screen.

```
+ (NSColor *)highlightColor
```

### Return Value

The system color for the virtual light source on the screen.

### Discussion

This method is invoked by the [highlightWithLevel:](#) (page 55) method. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [highlightWithLevel:](#) (page 55)

### Declared In

NSColor.h

## ignoresAlpha

Returns a Boolean value indicating whether the application supports alpha.

```
+ (BOOL)ignoresAlpha
```

### Return Value

YES if the application doesn't support alpha; otherwise NO. This value is consulted when an application imports alpha (through color dragging, for instance). The value determines whether the color panel has an opacity slider.

This value is YES by default, indicating that the opacity components of imported colors will be set to 1.0. If an application wants alpha, it can invoke the [setIgnoresAlpha:](#) (page 39) method with a parameter of NO.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [setIgnoresAlpha:](#) (page 39)

- [alphaComponent](#) (page 43)

**Declared In**

NSColor.h

**keyboardFocusIndicatorColor**

Returns the system color that represents the keyboard focus ring around controls.

```
+ (NSColor *)keyboardFocusIndicatorColor
```

**Return Value**

The system color representing the focus ring.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSColor.h

**knobColor**

Returns the system color used for the flat surface of a slider knob that hasn't been selected.

```
+ (NSColor *)knobColor
```

**Return Value**

The system color used for an unselected slider knob.

**Discussion**

The knob's beveled edges, which set it in relief, are drawn in highlighted and shadowed versions of the face color. When a knob is selected, its color changes to [selectedKnobColor](#) (page 37). For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

QuickLookSketch

Sketch+Accessibility

Sketch-112

**Declared In**

NSColor.h

## lightGrayColor

Returns an `NSColor` object whose grayscale value is 2/3 and whose alpha value is 1.0.

```
+ (NSColor *)lightGrayColor
```

### Return Value

The `NSColor` object.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [grayColor](#) (page 30)

+ [darkGrayColor](#) (page 29)

### Related Sample Code

IconCollection

IdentitySample

Quartz Composer WWDC 2005 TextEdit

Sketch-112

TrackIt

### Declared In

`NSColor.h`

## magentaColor

Returns an `NSColor` object whose RGB value is 1.0, 0.0, 1.0 and whose alpha value is 1.0.

```
+ (NSColor *)magentaColor
```

### Return Value

The `NSColor` object.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [magentaComponent](#) (page 56)

### Related Sample Code

CocoaVideoFrameToGWorld

LayoutManagerDemo

RGB Image

### Declared In

`NSColor.h`

## orangeColor

Returns an `NSColor` object whose RGB value is 1.0, 0.5, 0.0 and whose alpha value is 1.0.

```
+ (NSColor *)orangeColor
```

**Return Value**

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

DispatchFractal

Grady

MenuItemView

**Declared In**

NSColor.h

## purpleColor

Returns an NSColor object whose RGB value is 0.5, 0.0, 0.5 and whose alpha value is 1.0.

```
+ (NSColor *)purpleColor
```

**Return Value**

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

MenuItemView

**Declared In**

NSColor.h

## redColor

Returns an NSColor object whose RGB value is 1.0, 0.0, 0.0 and whose alpha value is 1.0.

```
+ (NSColor *)redColor
```

**Return Value**

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [redComponent](#) (page 58)

**Related Sample Code**

MatrixMixerTest

UIElementInspector

WebKitPluginStarter

WebKitPluginWithJavaScript  
WhackedTV

**Declared In**  
NSColor.h

## scrollBarColor

Returns the system color used for scroll “bars”—that is, for the groove in which a scroller’s knob moves

+ (NSColor \*)scrollBarColor

### Return Value

The system color used for scroll bars. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

**Declared In**  
NSColor.h

## secondarySelectedControlColor

Returns the system color used in non-key views.

+ (NSColor \*)secondarySelectedControlColor

### Return Value

The system color used in non-key views. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.1 and later.

### See Also

+ [selectedControlColor](#) (page 36)

**Declared In**  
NSColor.h

## selectedControlColor

Returns the system color used for the face of a selected control.

+ (NSColor \*)selectedControlColor

### Return Value

The system color used for the face of a selected control—a control being dragged or clicked. For general information about system colors, see [Accessing System Colors](#)

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + [selectedControlTextColor](#) (page 37)
- + [secondarySelectedControlColor](#) (page 36)
- + [alternateSelectedControlColor](#) (page 14)

**Related Sample Code**

CocoaAUHost

DictionaryController

**Declared In**

NSColor.h

**selectedControlTextColor**

Returns the system color used for text in a selected control—a control being clicked or dragged.

```
+ (NSColor *)selectedControlTextColor
```

**Return Value**

The system color used for text in a selected control—a control being clicked or dragged. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + [selectedControlColor](#) (page 36)
- + [alternateSelectedControlTextColor](#) (page 15)

**Related Sample Code**

TextSizingExample

**Declared In**

NSColor.h

**selectedKnobColor**

Returns the system color used for the slider knob when it is selected.

```
+ (NSColor *)selectedKnobColor
```

**Return Value**

The system color used for a slider knob that is selected—that is, dragged. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**+ [knobColor](#) (page 33)**Declared In**

NSColor.h

**selectedMenuItemColor**

Returns the system color used for the face of selected menu items.

+ (NSColor \*)selectedMenuItemColor

**Return Value**The system color used for selected menu items. For general information about system colors, see [Accessing System Colors](#).**Availability**

Available in Mac OS X v10.0 and later.

**See Also**+ [selectedMenuItemTextColor](#) (page 38)**Declared In**

NSColor.h

**selectedMenuItemTextColor**

Returns the system color used for the text in menu items.

+ (NSColor \*)selectedMenuItemTextColor

**Return Value**The system color used for text in selected menu items. For general information about system colors, see [Accessing System Colors](#).**Availability**

Available in Mac OS X v10.0 and later.

**See Also**+ [selectedMenuItemColor](#) (page 38)**Declared In**

NSColor.h

**selectedTextBackgroundColor**

Returns the system color used for the background of selected text.

+ (NSColor \*)selectedTextBackgroundColor

**Return Value**

The system color used for the background of selected text. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [selectedTextColor](#) (page 39)

**Related Sample Code**

[TextSizingExample](#)

**Declared In**

NSColor.h

**selectedTextColor**

Returns the system color used for selected text.

```
+ (NSColor *)selectedTextColor
```

**Return Value**

The system color used for selected text. For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [selectedTextBackgroundColor](#) (page 38)

**Declared In**

NSColor.h

**setIgnoresAlpha:**

Specifies whether an application supports alpha.

```
+ (void)setIgnoresAlpha:(BOOL)flag
```

**Parameters**

*flag*

YES to indicate that the application won't support alpha. By default, applications ignore alpha.

**Discussion**

If the application doesn't support alpha, no opacity slider is displayed in the color panel, and colors dragged in or pasted have their alpha values set to 1.0. Applications that need to import alpha can invoke this method with *flag* set to NO and explicitly make colors opaque in cases where it matters to them. Note that calling this with a value of YES overrides any value set with the NSColorPanel method [setShowsAlpha:](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + [ignoresAlpha](#) (page 32)
- [alphaComponent](#) (page 43)

**Related Sample Code**

Aperture Edit Plugin - Borders & Titles  
 CompositeLab  
 Quartz Composer QCTV  
 Tinted Image

**Declared In**

NSColor.h

## shadowColor

Returns the system color that represents the virtual shadows cast by raised objects on the screen.

```
+ (NSColor *)shadowColor
```

**Return Value**

The system color for the virtual shadows case by raised objects on the screen.

**Discussion**

This method is invoked by [shadowWithLevel:](#) (page 60). For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [shadowWithLevel:](#) (page 60)

**Declared In**

NSColor.h

## textBackgroundColor

Returns the system color used for the text background.

```
+ (NSColor *)textBackgroundColor
```

**Return Value**

The system color used for the background of text. When text is selected, its background color changes to the return value of [selectedTextBackgroundColor](#) (page 38). For general information about system colors, see [Accessing System Colors](#).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + [textColor](#) (page 41)

### Related Sample Code

[TextSizingExample](#)

### Declared In

NSColor.h

## textColor

Returns the system color used for text.

```
+ (NSColor *)textColor
```

### Return Value

The system color used for text. When text is selected, its color changes to the return value of [selectedTextColor](#) (page 39). For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [textBackgroundColor](#) (page 40)

### Declared In

NSColor.h

## whiteColor

Returns an NSColor object whose grayscale and alpha values are both 1.0.

```
+ (NSColor *)whiteColor
```

### Return Value

The NSColor object.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [whiteComponent](#) (page 60)

### Related Sample Code

[Cocoa OpenGL](#)

[DragItemAround](#)

[OpenGL Screensaver](#)

[Quartz Composer WWDC 2005 TextEdit](#)

[Sketch-112](#)

### Declared In

NSColor.h

## windowBackgroundColor

Returns a pattern color that will draw the ruled lines for the window background.

```
+ (NSColor *)windowBackgroundColor
```

### Return Value

The pattern color used for the background of a window. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### Related Sample Code

[AnimatedTableView](#)

### Declared In

NSColor.h

## windowFrameColor

Returns the system color used for window frames, except for their text.

```
+ (NSColor *)windowFrameColor
```

### Return Value

The system color used for window frames. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [windowFrameTextColor](#) (page 42)

### Declared In

NSColor.h

## windowFrameTextColor

Returns the system color used for the text in window frames.

```
+ (NSColor *)windowFrameTextColor
```

### Return Value

The system color used for text in window frames. For general information about system colors, see [Accessing System Colors](#).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [windowFrameColor](#) (page 42)

**Declared In**

NSColor.h

**yellowColor**

Returns an NSColor object whose RGB value is 1.0, 1.0, 0.0 and whose alpha value is 1.0.

```
+ (NSColor *)yellowColor
```

**Return Value**

The NSColor object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [yellowComponent](#) (page 61)

**Related Sample Code**

LayoutManagerDemo

RGB Image

SimpleStickies

StickiesWithCoreData

WebKitPluginStarter

**Declared In**

NSColor.h

## Instance Methods

**alphaComponent**

Returns the receiver's alpha (opacity) component.

```
- (CGFloat)alphaComponent
```

**Return Value**

The alpha component of the color object. If the receiver has no alpha component, this is 1.0 (opaque).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getCyan:magenta:yellow:black:alpha:](#) (page 51)

- [getHue:saturation:brightness:alpha:](#) (page 52)

- [getRed:green:blue:alpha:](#) (page 53)

- [getWhite:alpha:](#) (page 54)

**Related Sample Code**

DragItemAround

Monochrome Image

#### Declared In

NSColor.h

## blackComponent

Returns the receiver's black component.

- (CGFloat)blackComponent

#### Return Value

The color object's black component.

#### Discussion

This method works only with objects representing colors in the `NSDeviceCMYKColorSpace` color space. Sending it to other objects raises an exception.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [getCyan:magenta:yellow:black:alpha:](#) (page 51)

#### Declared In

NSColor.h

## blendedColorWithFraction:ofColor:

Creates and returns an `NSColor` object whose component values are a weighted sum of the receiver's and the specified color object's.

- (NSColor \*)blendedColorWithFraction:(CGFloat)fraction ofColor:(NSColor \*)color

#### Parameters

*fraction*

The amount of the color to blend with the receiver's color. The method converts *color* and a copy of the receiver to RGB, and then sets each component of the returned color to *fraction* of *color*'s value plus  $1 - \textit{fraction}$  of the receiver's.

*color*

The color to blend with the receiver's color.

#### Return Value

The resulting color object or `nil` if the colors can't be converted.

#### Availability

Available in Mac OS X v10.0 and later.

#### Related Sample Code

DockTile

SpeedometerView

WebKitPluginStarter

WebKitPluginWithJavaScript

**Declared In**

NSColor.h

## blueComponent

Returns the receiver's blue component.

- (CGFloat)blueComponent

**Return Value**

The color object's blue component.

**Discussion**

This method works only with objects representing colors in the `NSCalibratedRGBColorSpace` or `NSDeviceRGBColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getRed:green:blue:alpha:](#) (page 53)

**Related Sample Code**

CIColorTracking

Color Sampler

QTCoreVideo301

Sketch+Accessibility

**Declared In**

NSColor.h

## brightnessComponent

Returns the brightness component of the HSB color equivalent to the receiver.

- (CGFloat)brightnessComponent

**Return Value**

The color object's brightness component.

**Discussion**

This method works only with objects representing colors in the `NSCalibratedRGBColorSpace` or `NSDeviceRGBColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getHue:saturation:brightness:alpha:](#) (page 52)

**Declared In**

NSColor.h

**catalogNameComponent**

Returns the name of the catalog containing the receiver's name.

- (NSString \*)catalogNameComponent

**Return Value**

The name of the catalog containing the color object.

**Discussion**

This method raises an exception if the receiver's color space isn't NSNamedColorSpace.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**[+ colorWithCatalogName:colorName:](#) (page 20)[- colorNameComponent](#) (page 46)[- localizedCatalogNameComponent](#) (page 56)**Declared In**

NSColor.h

**colorNameComponent**

Returns the receiver's name.

- (NSString \*)colorNameComponent

**Return Value**

The name of the color object.

**Discussion**

This method raises an exception if the receiver's color space isn't NSNamedColorSpace.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**[+ colorWithCatalogName:colorName:](#) (page 20)[- catalogNameComponent](#) (page 46)[- localizedCatalogNameComponent](#) (page 56)**Declared In**

NSColor.h

## colorSpace

Returns an object representing the color space of the receiver.

```
- (NSColorSpace *)colorSpace
```

### Return Value

An object representing a color space. The returned `NSColorSpace` object may represent a custom color space.

### Discussion

Calling this method raises an exception if the receiver is not based on a color space represented by an `NSColorSpace` object—specifically, colors designated by `NSNamedColorSpace` and `NSPatternColorSpace`. If you are unsure about a color object, convert it to an equivalent `NSColorSpace`-based object before calling this method. Color objects created with color-space names `NSCalibratedWhiteColorSpace`, `NSCalibratedBlackColorSpace`, `NSCalibratedRGBColorSpace`, `NSDeviceWhiteColorSpace`, `NSDeviceBlackColorSpace`, `NSDeviceRGBColorSpace`, `NSDeviceCMYKColorSpace`, or `NSCustomColorSpace`—or with the `NSColorSpace` class methods corresponding to these names—are safe to use with this method. See “About Color Spaces” in *Color Programming Topics for Cocoa* for a list of these corresponding methods.

### Availability

Available in Mac OS X v10.4 and later.

### See Also

- [getComponents:](#) (page 51)
- [numberOfComponents](#) (page 57)

### Declared In

`NSColor.h`

## colorSpaceName

Returns the name of the receiver’s color space.

```
- (NSString *)colorSpaceName
```

### Return Value

The name of the color space.

### Discussion

This method should be implemented in subclasses of `NSColor`.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [colorUsingColorSpaceName:](#) (page 48)
- [colorUsingColorSpaceName:device:](#) (page 49)

### Related Sample Code

`MyCustomColorPicker`

**Declared In**  
NSColor.h

## colorUsingColorSpace:

Returns a new `NSColor` object representing the color of the receiver in the specified color space.

```
- (NSColor *)colorUsingColorSpace:(NSColorSpace *)space
```

### Parameters

*space*

The color space of the new `NSColor` object.

### Return Value

The new `NSColor` object. This method converts the receiver's color to an equivalent one in the new color space. Although the new color might have different component values, it looks the same as the original. Returns `nil` if conversion is not possible.

If the receiver's color space is the same as that specified in *space*, this method returns the same `NSColor` object.

### Availability

Available in Mac OS X v10.4 and later.

### See Also

+ [colorWithColorSpace:components:count:](#) (page 21)

### Related Sample Code

Quartz 2D Shadings

Quartz Composer QCTV

**Declared In**  
NSColor.h

## colorUsingColorSpaceName:

Creates and returns an `NSColor` whose color is the same as the receiver's, except that the new `NSColor` is in the specified color space.

```
- (NSColor *)colorUsingColorSpaceName:(NSString *)colorSpace
```

### Parameters

*colorSpace*

The name of the color space containing the new `NSColor` object. If *colorSpace* is `nil`, the most appropriate color space is used.

### Return Value

The new `NSColor` object or `nil` if the specified conversion cannot be done.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [colorSpaceName](#) (page 47)

**Related Sample Code**

CWCocoaComponent

DispatchFractal

GLChildWindowDemo

MenuItemView

Reminders

**Declared In**

NSColor.h

**colorUsingColorSpaceName:device:**

Creates and returns an `NSColor` object whose color is the same as the receiver's, except that the new `NSColor` is in the given color space and is specific to the given device.

```
- (NSColor *)colorUsingColorSpaceName:(NSString *)colorSpace device:(NSDictionary *)deviceDescription
```

**Parameters***colorSpace*

The name of the color space containing the new `NSColor` object. If *colorSpace* is `nil`, the most appropriate color space is used.

*deviceDescription*

The device description. Device descriptions can be obtained from windows, screens, and printers with the `deviceDescription` method.

If *deviceDescription* is `nil`, the current device (as obtained from the currently lockFocus'ed view's window or, if printing, the current printer) is used.

**Return Value**

The new `NSColor` object or `nil` if the specified conversion cannot be done.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [colorSpaceName](#) (page 47)
- [colorUsingColorSpaceName:](#) (page 48)

**Declared In**

NSColor.h

**colorWithAlphaComponent:**

Creates and returns an `NSColor` object that has the same color space and component values as the receiver, but the specified alpha component.

```
- (NSColor *)colorWithAlphaComponent:(CGFloat)alpha
```

**Parameters***alpha*

The opacity value of the new `NSColor` object.

**Return Value**

The new `NSColor` object. If the receiver's color space doesn't include an alpha component, the receiver is returned.

**Discussion**

A subclass with explicit opacity components should override this method to return a color with the specified alpha.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [alphaComponent](#) (page 43)
- [blendedColorWithFraction:ofColor:](#) (page 44)

**Related Sample Code**

FunkyOverlayWindow  
ImageMap  
ImageMapExample  
Sketch-112  
WebKitPluginStarter

**Declared In**

`NSColor.h`

**cyanComponent**

Returns the receiver's cyan component.

- (CGFloat)cyanComponent

**Return Value**

The color object's cyan component.

**Discussion**

This method works only with objects representing colors in the `NSDeviceCMYKColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getCyan:magenta:yellow:black:alpha:](#) (page 51)

**Declared In**

`NSColor.h`

**drawSwatchInRect:**

Draws the current color in the given rectangle.

- (void)drawSwatchInRect:(NSRect)rect

**Parameters***rect*

The rectangle in which to draw the color.

**Discussion**

Subclasses adorn the rectangle in some manner to indicate the type of color. This method is invoked by color wells, swatches, and other user interface objects that need to display colors.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSColor.h

**getComponents:**

Returns the components of the receiver as an array.

```
- (void)getComponents:(CGFloat *)components
```

**Parameters***components*

An array containing the components of the color object as `float` values.

**Discussion**

You can invoke this method on `NSColor` objects created from custom color spaces to get the individual floating point components, including alpha. Raises an exception if the receiver doesn't have floating-point components. To find out how many components are in the *components* array, send the receiver a [numberOfComponents](#) (page 57) message.

**Availability**

Available in Mac OS X v10.4 and later.

**See Also**

- [colorSpace](#) (page 47)

**Related Sample Code**

AnimatedTableView

DispatchFractal

Quartz Composer QCTV

**Declared In**

NSColor.h

**getCyan:magenta:yellow:black:alpha:**

Returns the receiver's CMYK and opacity values.

```
- (void)getCyan:(CGFloat *)cyan magenta:(CGFloat *)magenta yellow:(CGFloat *)yellow
  black:(CGFloat *)black alpha:(CGFloat *)alpha
```

**Parameters***cyan*

Upon return, contains the cyan component of the color object.

*magenta*

Upon return, contains the magenta component of the color object.

*yellow*

Upon return, contains the yellow component of the color object.

*black*

Upon return, contains the black component of the color object.

*alpha*

Upon return, contains opacity value of the color object.

**Discussion**

If `NULL` is passed in as an argument, the method doesn't set that value. This method works only with objects representing colors in the `NSDeviceCMYKColorSpace`. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [alphaComponent](#) (page 43)
- [blackComponent](#) (page 44)
- [cyanComponent](#) (page 50)
- [magentaComponent](#) (page 56)
- [yellowComponent](#) (page 61)

**Declared In**

NSColor.h

**getHue:saturation:brightness:alpha:**

Returns the receiver's HSB component and opacity values in the respective arguments.

```
- (void)getHue:(CGFloat *)hue saturation:(CGFloat *)saturation brightness:(CGFloat *)brightness alpha:(CGFloat *)alpha
```

**Parameters***hue*

Upon return, contains the hue component of the color object.

*saturation*

Upon return, contains the saturation component of the color object.

*brightness*

Upon return, contains the brightness component of the color object.

*alpha*

Upon return, contains the opacity value of the color object.

**Discussion**

If `NULL` is passed in as an argument, the method doesn't set that value. This method works only with objects representing colors in the `NSCalibratedRGBColorSpace` or `NSDeviceRGBColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [alphaComponent](#) (page 43)
- [brightnessComponent](#) (page 45)
- [hueComponent](#) (page 55)
- [saturationComponent](#) (page 58)

**Related Sample Code**

MenuItemView

**Declared In**

NSColor.h

**getRed:green:blue:alpha:**

Returns the receiver's RGB component and opacity values in the respective arguments.

```
- (void)getRed:(CGFloat *)red green:(CGFloat *)green blue:(CGFloat *)blue
  alpha:(CGFloat *)alpha
```

**Parameters**

*red*

Upon return, contains the red component of the color object.

*green*

Upon return, contains the green component of the color object.

*blue*

Upon return, contains the blue component of the color object.

*alpha*

Upon return, contains the opacity value of the color object.

**Discussion**

If NULL is passed in as an argument, the method doesn't set that value. This method works only with objects representing colors in the NSCalibratedRGBColorSpace or NSDeviceRGBColorSpace color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [alphaComponent](#) (page 43)
- [blueComponent](#) (page 45)
- [greenComponent](#) (page 54)
- [redComponent](#) (page 58)

**Related Sample Code**

CWCocoaComponent

FunHouse

GLChildWindowDemo

Quartz 2D Shadings

## Reminders

**Declared In**

NSColor.h

**getWhite:alpha:**

Returns the receiver's grayscale value and alpha values.

```
- (void)getWhite:(CGFloat *)white alpha:(CGFloat *)alpha
```

**Parameters***white*

Upon return, contains the grayscale value of the color object.

*alpha*

Upon return, contains the opacity value of the color object.

**Discussion**

If `NULL` is passed in as an argument, the method doesn't set that value. This method works only with objects representing colors in the `NSCalibratedWhiteColorSpace`, `NSCalibratedBlackColorSpace`, `NSDeviceBlackColorSpace`, or `NSDeviceWhiteColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [alphaComponent](#) (page 43)
- [whiteComponent](#) (page 60)

**Declared In**

NSColor.h

**greenComponent**

Returns the receiver's green component.

```
- (CGFloat)greenComponent
```

**Return Value**

The color object's green component.

**Discussion**

This method works only with objects representing colors in the `NSCalibratedRGBColorSpace` or `NSDeviceRGBColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getRed:green:blue:alpha:](#) (page 53)

**Related Sample Code**

CIColorTracking

Color Sampler

QTCoreVideo301

Sketch+Accessibility

**Declared In**

NSColor.h

**highlightWithLevel:**

Returns an `NSColor` object that represents a blend between the receiver and the highlight color returned by `highlightColor` (page 32).

- (`NSColor *`)`highlightWithLevel:(CGFloat)highlightLevel`

**Parameters**

*highlightLevel*

The amount of the highlight color that is blended with the receiver's color. This should be a number from 0.0 through 1.0. A *highlightLevel* below 0.0 is interpreted as 0.0; a *highlightLevel* above 1.0 is interpreted as 1.0.

**Return Value**

The new `NSColor` object. Returns `nil` if the colors can't be converted.

**Discussion**

Invoke this method when you want to brighten the receiving `NSColor` for use in highlights.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- `shadowWithLevel:` (page 60)

**Declared In**

NSColor.h

**hueComponent**

Returns the hue component of the HSB color equivalent to the receiver.

- (`CGFloat`)`hueComponent`

**Return Value**

The color object's hue component.

**Discussion**

This method works only with objects representing colors in the `NSCalibratedRGBColorSpace` or `NSDeviceRGBColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getHue:saturation:brightness:alpha:](#) (page 52)

**Declared In**

NSColor.h

## localizedCatalogNameComponent

Returns the name of the catalog containing the receiver's name as a localized string.

- (NSString \*)localizedCatalogNameComponent

**Return Value**

The name of catalog containing the color object's name as a localized string. This string may be displayed in user interface items like color pickers.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [colorWithCatalogName:colorName:](#) (page 20)

- [colorNameComponent](#) (page 46)

**Declared In**

NSColor.h

## localizedColorNameComponent

Returns the name of the receiver as a localized string.

- (NSString \*)localizedColorNameComponent

**Return Value**

The name of color object as a localized string. This string may be displayed in user interface items like color pickers.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [colorWithCatalogName:colorName:](#) (page 20)

- [catalogNameComponent](#) (page 46)

- [colorNameComponent](#) (page 46)

- [localizedCatalogNameComponent](#) (page 56)

**Declared In**

NSColor.h

## magentaComponent

Returns the receiver's magenta component.

- (CGFloat)magentaComponent

#### Return Value

The color object's magenta component.

#### Discussion

This method works only with objects representing colors in the `NSDeviceCMYKColorSpace` color space. Sending it to other objects raises an exception.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [getColorSpace](#):yellow:black:alpha: (page 51)

#### Declared In

NSColor.h

## numberOfComponents

Returns the number of components in the receiver.

- (NSInteger)numberOfComponents

#### Return Value

The number of components in the color object. The floating-point components counted include alpha. This method raises an exception if the receiver doesn't have floating-point components.

#### Availability

Available in Mac OS X v10.4 and later.

#### See Also

- [colorSpace](#) (page 47)  
- [getComponents:](#) (page 51)

#### Related Sample Code

AnimatedTableView

#### Declared In

NSColor.h

## patternImage

Returns the image that the receiver is using as a pattern.

- (NSImage \*)patternImage

#### Return Value

The image used by the color object. If the receiver doesn't have an image, this method raises an exception.

#### Availability

Available in Mac OS X v10.0 and later.

**Declared In**  
NSColor.h

## redComponent

Returns the receiver's red component.

- (CGFloat)redComponent

**Return Value**  
The color object's red component.

**Discussion**  
This method works only with objects representing colors in the NSCalibratedRGBColorSpace or NSDeviceRGBColorSpace color space. Sending it to other objects raises an exception.

**Availability**  
Available in Mac OS X v10.0 and later.

**See Also**  
- [getRed:green:blue:alpha:](#) (page 53)

**Related Sample Code**  
CIColorTracking  
Color Sampler  
QTCoreVideo301  
Sketch+Accessibility

**Declared In**  
NSColor.h

## saturationComponent

Returns the saturation component of the HSB color equivalent to the receiver.

- (CGFloat)saturationComponent

**Return Value**  
The color object's saturation component.

**Discussion**  
This method works only with objects representing colors in the NSCalibratedRGBColorSpace or NSDeviceRGBColorSpace color space. Sending it to other objects raises an exception.

**Availability**  
Available in Mac OS X v10.0 and later.

**See Also**  
- [getHue:saturation:brightness:alpha:](#) (page 52)

**Declared In**  
NSColor.h

## set

Sets the color of subsequent drawing to the color that the receiver represents.

- (void)set

### Discussion

This method should be implemented in subclasses.

### Availability

Available in Mac OS X v10.0 and later.

### Related Sample Code

FilterDemo

MatrixMixerTest

Sketch-112

WebKitPluginStarter

WhackedTV

### Declared In

NSColor.h

## setFill

Sets the fill color of subsequent drawing to the receiver's color.

- (void)setFill

### Discussion

This method should be implemented in subclasses.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [setStroke](#) (page 59)

### Related Sample Code

AnimatedTableView

JSPong

SampleRaster

### Declared In

NSColor.h

## setStroke

Sets the stroke color of subsequent drawing to the receiver's color.

- (void)setStroke

### Discussion

This method should be implemented in subclasses.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setFill](#) (page 59)

**Related Sample Code**

SampleRaster

**Declared In**

NSColor.h

**shadowWithLevel:**

Returns an `NSColor` object that represents a blend between the receiver and the shadow color returned by [shadowColor](#) (page 40).

```
- (NSColor *)shadowWithLevel:(CGFloat)shadowLevel
```

**Parameters**

*shadowLevel*

The amount of the shadow color used for the blend. This should be a number from 0.0 through 1.0. A *shadowLevel* below 0.0 is interpreted as 0.0; a *shadowLevel* above 1.0 is interpreted as 1.0.

**Return Value**

The new `NSColor` object. Returns `nil` if the colors can't be converted.

**Discussion**

Invoke this method when you want to darken the receiving `NSColor` for use in shadows.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [highlightWithLevel:](#) (page 55)

**Declared In**

NSColor.h

**whiteComponent**

Returns the receiver's white component.

```
- (CGFloat)whiteComponent
```

**Return Value**

The color object's white component.

**Discussion**

This method works only with objects representing colors in the `NSCalibratedWhiteColorSpace`, `NSCalibratedBlackColorSpace`, `NSDeviceBlackColorSpace`, or `NSDeviceWhiteColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getWhite:alpha:](#) (page 54)

**Declared In**

NSColor.h

**writeToPasteboard:**

Writes the receiver's data to the specified pasteboard.

```
- (void)writeToPasteboard:(NSPasteboard *)pasteBoard
```

**Parameters**

*pasteBoard*

The pasteboard to which to write the receiver's color data. If this pasteboard doesn't support color data, the method does nothing.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [colorFromPasteboard:](#) (page 17)

**Declared In**

NSColor.h

**yellowComponent**

Returns the receiver's yellow component.

```
- (CGFloat)yellowComponent
```

**Return Value**

The color object's yellow component.

**Discussion**

This method works only with objects representing colors in the `NSDeviceCMYKColorSpace` color space. Sending it to other objects raises an exception.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [getCyan:magenta:yellow:black:alpha:](#) (page 51)

**Declared In**

NSColor.h

## Constants

For definitions of `NSColor` constants, as well as a discussion of their usage, see “About Color Spaces” in *Color Programming Topics for Cocoa*.

### AppKit Versions for NSColor Bug Fixes

The version of the AppKit framework containing a specific bug fix.

```
#define NSAppKitVersionNumberWithPatternColorLeakFix 641.0
```

#### Constants

`NSAppKitVersionNumberWithPatternColorLeakFix`

The specific version of the AppKit framework that introduced the fix for correctly autoreleasing objects returned by the `colorWithPatternImage:` (page 25) method. Developers should not need to use this constant unless they are writing applications for Mac OS X v10.1 and earlier.

Available in Mac OS X v10.2 and later.

Declared in `NSColor.h`.

## Notifications

### NSSystemColorsDidChangeNotification

Sent when the system colors have been changed (such as through a system control panel interface).

This notification contains no notification object and no *userInfo* dictionary.

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`NSColor.h`

# Document Revision History

---

This table describes the changes to *NSColor Class Reference*.

Date	Notes
2009-01-06	Added definition for <code>NSAppKitVersionNumberWithPatternColorLeakFix</code> macro.
2008-10-15	Clarified applicability of certain methods to <code>NSColorSpace</code> -based color objects.
2006-05-23	First publication of this content as a separate document.
	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

alphaComponent **instance method** [43](#)  
alternateSelectedControlColor **class method** [14](#)  
alternateSelectedControlTextColor **class method** [15](#)  
AppKit Versions for NSColor Bug Fixes [62](#)

## B

---

blackColor **class method** [15](#)  
blackComponent **instance method** [44](#)  
blendedColorWithFraction:ofColor: **instance method** [44](#)  
blueColor **class method** [16](#)  
blueComponent **instance method** [45](#)  
brightnessComponent **instance method** [45](#)  
brownColor **class method** [16](#)

## C

---

catalogNameComponent **instance method** [46](#)  
clearColor **class method** [16](#)  
colorForControlTint: **class method** [17](#)  
colorFromPasteboard: **class method** [17](#)  
colorNameComponent **instance method** [46](#)  
colorSpace **instance method** [47](#)  
colorSpaceName **instance method** [47](#)  
colorUsingColorSpace: **instance method** [48](#)  
colorUsingColorSpaceName: **instance method** [48](#)  
colorUsingColorSpaceName:device: **instance method** [49](#)  
colorWithAlphaComponent: **instance method** [49](#)  
colorWithCalibratedHue:saturation:brightness:alpha: **class method** [18](#)  
colorWithCalibratedRed:green:blue:alpha: **class method** [18](#)

colorWithCalibratedWhite:alpha: **class method** [19](#)  
colorWithCatalogName:colorName: **class method** [20](#)  
colorWithCIColor: **class method** [20](#)  
colorWithColorSpace:components:count: **class method** [21](#)  
colorWithDeviceCyan:magenta:yellow:black:alpha: **class method** [22](#)  
colorWithDeviceHue:saturation:brightness:alpha: **class method** [22](#)  
colorWithDeviceRed:green:blue:alpha: **class method** [23](#)  
colorWithDeviceWhite:alpha: **class method** [24](#)  
colorWithPatternImage: **class method** [25](#)  
controlAlternatingRowBackgroundColors **class method** [25](#)  
controlBackgroundColor **class method** [25](#)  
controlColor **class method** [26](#)  
controlDarkShadowColor **class method** [26](#)  
controlHighlightColor **class method** [27](#)  
controlLightHighlightColor **class method** [27](#)  
controlShadowColor **class method** [27](#)  
controlTextColor **class method** [28](#)  
currentControlTint **class method** [28](#)  
cyanColor **class method** [29](#)  
cyanComponent **instance method** [50](#)

## D

---

darkGrayColor **class method** [29](#)  
disabledControlTextColor **class method** [29](#)  
drawSwatchInRect: **instance method** [50](#)

## G

---

getComponents: **instance method** [51](#)  
getCyan:magenta:yellow:black:alpha: **instance method** [51](#)

getHue:saturation:brightness:alpha: **instance method 52**  
 getRed:green:blue:alpha: **instance method 53**  
 getWhite:alpha: **instance method 54**  
 grayColor **class method 30**  
 greenColor **class method 30**  
 greenComponent **instance method 54**  
 gridColor **class method 31**

## H

---

headerColor **class method 31**  
 headerTextColor **class method 32**  
 highlightColor **class method 32**  
 highlightWithLevel: **instance method 55**  
 hueComponent **instance method 55**

## I

---

ignoresAlpha **class method 32**

## K

---

keyboardFocusIndicatorColor **class method 33**  
 knobColor **class method 33**

## L

---

lightGrayColor **class method 34**  
 localizedCatalogNameComponent **instance method 56**  
 localizedColorNameComponent **instance method 56**

## M

---

magentaColor **class method 34**  
 magentaComponent **instance method 56**

## N

---

NSAppKitVersionNumberWithPatternColorLeakFix **constant 62**  
 NSSystemColorsDidChangeNotification **notification 62**

numberOfComponents **instance method 57**

## O

---

orangeColor **class method 34**

## P

---

patternImage **instance method 57**  
 purpleColor **class method 35**

## R

---

redColor **class method 35**  
 redComponent **instance method 58**

## S

---

saturationComponent **instance method 58**  
 scrollbarColor **class method 36**  
 secondarySelectedControlColor **class method 36**  
 selectedControlColor **class method 36**  
 selectedControlTextColor **class method 37**  
 selectedKnobColor **class method 37**  
 selectedMenuItemColor **class method 38**  
 selectedMenuItemTextColor **class method 38**  
 selectedTextBackgroundColor **class method 38**  
 selectedTextColor **class method 39**  
 set **instance method 59**  
 setFill **instance method 59**  
 setIgnoresAlpha: **class method 39**  
 setStroke **instance method 59**  
 shadowColor **class method 40**  
 shadowWithLevel: **instance method 60**

## T

---

textBackgroundColor **class method 40**  
 textColor **class method 41**

## W

---

whiteColor **class method 41**  
 whiteComponent **instance method 60**

windowBackgroundColor **class method** [42](#)  
windowFrameColor **class method** [42](#)  
windowFrameTextColor **class method** [42](#)  
writeToPasteboard: **instance method** [61](#)

## Y

---

yellowColor **class method** [43](#)  
yellowComponent **instance method** [61](#)