

# NSColor

**Inherits From:** NSObject  
**Conforms To:** NSCopying, NSCoding  
**Declared In:** appkit/NSColor.h

## Creating an NSColor from Component Values

- + (NSColor \*)**colorWithCalibratedHue:(float)hue** Creates and returns a new NSColor whose color space is NSCalibratedRGBColorSpace, whose opacity value is *alpha*, and whose components in HSB space would be *hue*, *saturation*, and *brightness*. All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.
  - saturation:(float)saturation**
  - brightness:(float)brightness**
  - alpha:(float)alpha**
- + (NSColor \*)**colorWithCalibratedRed:(float)red** Creates and returns a new NSColor whose color space is NSCalibratedRGBColorSpace, whose opacity value is *alpha*, and whose RGB components are *red*, *green*, and *blue*. All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.
  - green:(float)green**
  - blue:(float)blue**
  - alpha:(float)alpha**
- + (NSColor \*)**colorWithCalibratedWhite:(float)white** Creates and returns a new NSColor whose color space is NSCalibratedWhiteColorSpace, whose opacity value is *alpha*, and whose grayscale value is *white*. All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.
  - alpha:(float)alpha**
- + (NSColor \*)**colorWithCatalogName:(NSString \*)listName**

<b>colorName:(NSString *)colorName</b>	Creates and returns a new NSColor whose color space is NSNamedColorSpace, by finding the color named <i>colorName</i> in the catalog named <i>listName</i> .
+ (NSColor *) <b>colorWithDeviceCyan:(float)cyan magenta:(float)magenta yellow:(float)yellow black:(float)black alpha:(float)alpha</b>	Creates and returns a new NSColor whose color space is NSDeviceCMYKColorSpace, whose opacity value is <i>alpha</i> , and whose CMYK components are <i>cyan</i> , <i>magenta</i> , <i>yellow</i> , and <i>black</i> . All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.
+ (NSColor *) <b>colorWithDeviceHue:(float)hue saturation:(float)saturation brightness:(float)brightness alpha:(float)alpha</b>	Creates and returns a new NSColor whose color space is NSDeviceRGBColorSpace, whose opacity value is <i>alpha</i> , and whose components in HSB space would be <i>hue</i> , <i>saturation</i> , and <i>brightness</i> . All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.
+ (NSColor *) <b>colorWithDeviceRed:(float)red green:(float)green blue:(float)blue alpha:(float)alpha</b>	Creates and returns a new NSColor whose color space is NSDeviceRGBColorSpace, whose opacity value is <i>alpha</i> , and whose RGB components are <i>red</i> , <i>green</i> , and <i>blue</i> . All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.
+ (NSColor *) <b>colorWithDeviceWhite:(float)white alpha:(float)alpha</b>	Creates and returns a new NSColor whose color space is NSDeviceWhiteColorSpace, whose opacity value is <i>alpha</i> , and whose grayscale value is <i>white</i> . All values are legal, but values less than 0.0 are set to 0.0, and values greater than 1.0 are set to 1.0.

## Creating an NSColor With Preset Components

+ (NSColor *) <b>blackColor</b>	Returns an NSColor in NSCalibratedWhiteColorSpace whose grayscale value is 0.0 and whose alpha value is 1.0.
+ (NSColor *) <b>blueColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 0.0, 0.0, 1.0 and whose alpha value is 1.0.
+ (NSColor *) <b>brownColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 0.6, 0.4, 0.2 and whose alpha value is 1.0.

+ (NSColor *) <b>clearColor</b>	Returns an NSColor in NSCalibratedWhiteColorSpace whose grayscale and alpha values are both 0.0.
+ (NSColor *) <b>cyanColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 0.0, 1.0, 1.0 and whose alpha value is 1.0.
+ (NSColor *) <b>darkGrayColor</b>	Returns an NSColor in NSCalibratedWhiteColorSpace whose grayscale value is 1/3 and whose alpha value is 1.0.
+ (NSColor *) <b>grayColor</b>	Returns an NSColor in NSCalibratedWhiteColorSpace whose grayscale value is 0.5 and whose alpha value is 1.0.
+ (NSColor *) <b>greenColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 0.0, 1.0, 0.0 and whose alpha value is 1.0.
+ (NSColor *) <b>lightGrayColor</b>	Returns an NSColor in NSCalibratedWhiteColorSpace whose grayscale value is 2/3 and whose alpha value is 1.0.
+ (NSColor *) <b>magentaColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 1.0, 0.0, 1.0 and whose alpha value is 1.0.
+ (NSColor *) <b>orangeColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 1.0, 0.5, 0.0 and whose alpha value is 1.0.
+ (NSColor *) <b>purpleColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 0.5, 0.0, 0.5 and whose alpha value is 1.0.
+ (NSColor *) <b>redColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 1.0, 0.0, 0.0 and whose alpha value is 1.0.
+ (NSColor *) <b>whiteColor</b>	Returns an NSColor in NSCalibratedWhiteColorSpace whose grayscale and alpha values are both 1.0.
+ (NSColor *) <b>yellowColor</b>	Returns an NSColor in NSCalibratedRGBColorSpace whose RGB value is 1.0, 1.0, 0.0 and whose alpha value is 1.0.

## Ignoring Alpha Components

+ (BOOL) <b>ignoresAlpha</b>	Returns YES (the default) if the application hides the color panel's opacity slider and sets imported colors' alpha values to 1.0.
+ (void) <b>setIgnoresAlpha:(BOOL)<i>flag</i></b>	If <i>flag</i> is YES, no opacity slider is displayed in the color panel, and colors dragged in or pasted have their alpha values set to 1.0.

## Retrieving a Set of Components

- (void)**getCyan:**(float \*)*cyan*  
**magenta:**(float \*)*magenta*  
**yellow:**(float \*)*yellow*  
**black:**(float \*)*black*  
**alpha:**(float \*)*alpha*
- (void)**getHue:**(float \*)*hue*  
**saturation:**(float \*)*saturation*  
**brightness:**(float \*)*brightness*  
**alpha:**(float \*)*alpha*
- (void)**getRed:**(float \*)*red*  
**green:**(float \*)*green*  
**blue:**(float \*)*blue*  
**alpha:**(float \*)*alpha*
- (void)**getWhite:**(float \*)*white*  
**alpha:**(float \*)*alpha*

Returns the CMYK and alpha values in the respective arguments. If **NULL** is passed in as an argument, the method doesn't set that value. It's an error if the receiver isn't a CMYK color.

Returns the HSB and alpha values in the respective arguments. If **nil** is passed in as an argument, the method doesn't set that value. It's an error if the receiver isn't a CMYK color.

Returns the RGB and alpha values in the respective arguments. If **nil** is passed in as an argument, the method doesn't set that value. It's an error if the receiver isn't a CMYK color.

Returns the grayscale and alpha values in the respective arguments. If **nil** is passed in as an argument, the method doesn't set that value. It's an error if the receiver isn't a CMYK color.

## Retrieving Individual Components

- (float)**alphaComponent**
- (float)**blackComponent**
- (float)**blueComponent**
- (float)**brightnessComponent**
- (NSString \*)**catalogNameComponent**
- (NSString \*)**colorNameComponent**
- (float)**cyanComponent**
- (float)**greenComponent**
- (float)**hueComponent**

Returns the alpha (opacity) component (1.0 by default).

Returns the black component. It's an error if the receiver isn't a CMYK color.

Returns the blue component. It's an error if the receiver isn't an RGB color.

Returns the brightness component of the HSB color equivalent to the receiver. It's an error if the receiver isn't an RGB color.

Returns the name of the catalog containing this color, or **nil** if the receiver's color space isn't **NSNamedColorSpace**.

Returns the name of this color, or **nil** if the receiver's color space isn't **NSNamedColorSpace**.

Returns the cyan component. It's an error if the receiver isn't a CMYK color.

Returns the green component. It's an error if the receiver isn't an RGB color.

Returns the hue component of the HSB color equivalent to the receiver. It's

- (float)**magentaComponent** Returns the magenta component. It's an error if the receiver isn't a CMYK color.
- (float)**redComponent** Returns the red component. It's an error if the receiver isn't an RGB color.
- (float)**saturationComponent** Returns the saturation component of the HSB color equivalent to the receiver. It's an error if the receiver isn't an RGB color.
- (float)**whiteComponent** Returns the white component. It's an error if the receiver isn't a grayscale color.
- (float)**yellowComponent** Returns the yellow component. It's an error if the receiver isn't a CMYK color.

## Converting to Another Color Space

- (NSString \*)**colorSpaceName** Returns the name of the NSColor's color space.
- (NSColor \*)**colorUsingColorSpaceName:(NSString \*)colorSpace** Returns a newly created NSColor whose color is the same as the receiver's, except that the new NSColor is in the color space named *colorSpace*. This method calls **colorUsingColorSpaceName:device:** with the current device, indicating that the color is appropriate for the current device (the current window if drawing, or the current printer if printing).
- (NSColor \*)**colorUsingColorSpaceName:(NSString \*)colorSpace  
device:(NSDictionary \*)deviceDescription** Returns a newly created NSColor whose color is the same as the receiver's, except that the new NSColor is in the color space named *colorSpace* and is specific to the device described by *deviceDescription*.

## Changing the Color

- (NSColor \*)**blendedColorWithFraction:(float)fraction  
ofColor:(NSColor \*)aColor** Returns a newly created NSColor in NSCalibratedRGBColorSpace whose component values are a weighted sum of the receiver's and *aColor*'s. The method converts *aColor* and a copy of the receiver to RGB, and then sets each component of the returned color to *fraction* of *aColor*'s value

plus 1 - *fraction* of the receiver's. If the colors can't be converted to NSCalibratedRGBColorSpace, **nil** is returned.

- (NSColor \*)**colorWithAlphaComponent:**(float)*alpha*

Returns a newly created NSColor that has the same color space and component values as the receiver, except that its alpha component is *alpha*. If the receiver's color space doesn't include an alpha component, the receiver is returned.

## Copying and Pasting

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

Returns the NSColor currently on the pasteboard, or **nil** if the pasteboard doesn't contain color data. The returned color's alpha component is set to 1.0 if **ignoresAlpha** returns YES.

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

Writes the receiver's data to the pasteboard, unless the pasteboard doesn't support color data (in which case the method does nothing).

## Drawing

- (void)**drawSwatchInRect:**(NSRect)*rect*

Draws the current color in the rectangle *rect*. 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.

- (void)**set**

Sets the color of subsequent PostScript drawing to the color that the receiver represents. If the application is drawing to the screen rather than printing, this method also sets the current drawing context's alpha value to the value returned by **alphaComponent**.