CIColor Class Reference

Cocoa > Graphics & Imaging

2006-12-05

Contents

CIColor Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	Library/Frameworks/QuartzCore.framework
Availability	Mac OS X v10.4 and later
Declared in	CIColor.h
Companion guides	Core Image Programming Guide Color Management Overview
Related sample code	CIAnnotation

Overview

The `CIColor` class contains color values and the color space for which the color values are valid. You use `CIColor` objects in conjunction with other Core Image classes, such as `CIFilter`, `CIContext`,and `CIImage`, to take advantage of the built-in Core Image filters when processing images.

A color space defines a one-, two-, three-, or four-dimensional environment whose color components represent intensity values. A color component is also referred to as a color channel. An RGB color space, for example, is a three-dimensional color space whose stimuli are the red, green, and blue intensities that make up a given color. Regardless of the color space, in Core Image, color values range from `0.0` to `1.0`, with `0.0` representing an absence of that component (0 percent) and `1.0` representing 100 percent.

Colors also have an alpha component that represents the opacity of the color, with `0.0` meaning completely transparent and `1.0` meaning completely opaque. If a color does not have an explicit alpha component, Core Image paints the color as if the alpha component equals `1.0`. You always provide unpremultiplied color components to Core Image and Core Image provides unpremultiplied color components to you. Core Image premultiplies each color component with the alpha value in order to optimize calculations. For more information on premultiplied alpha values see *Core Image Programming Guide*.

Tasks

Initializing Color Objects

- `initWithCGColor:` (page 11)
  Initializes a color object with a Quartz color.

Creating Color Objects

+ `colorWithCGColor:` (page 7)
  Creates a color object from a Quartz color.
+ `colorWithRed:green:blue:` (page 7)
  Creates a color object using the specified RGB color component values
+ `colorWithRed:green:blue:alpha:` (page 8)
  Creates a color object using the specified RGBA color component values.
+ `colorWithString:` (page 9)
  Creates a color object using the RGBA color component values specified by a string.

Getting Color Components

- `alpha` (page 9)
  Returns the alpha value of the color.
- `blue` (page 10)
  Returns the blue component of the color.
- `colorSpace` (page 10)
  Returns the Quartz 2D color space associated with the color.
- `components` (page 10)
  Returns the color components of the color.
- `green` (page 11)
  Returns the green component of the color.
- `numberOfComponents` (page 11)
  Returns the number of color components in the color.
- `red` (page 12)
  Returns the red component of the color.
- `stringRepresentation` (page 12)
  Returns a formatted string that specifies the components of the color.

Class Methods

colorWithCGColor:

Creates a color object from a Quartz color.

```
+ (CIColor *)colorWithCGColor:(CGColorRef)c
```

Parameters

c

A Quartz color (`CGColorRef` object) created using a Quartz color creation function such as `CGColorCreate`.

Return Value

A Core Image color object that represents a Quartz color.

Discussion

A `CGColorRef` object is the fundamental opaque data type used internally by Quartz to represent colors. For more information on Quartz 2D color and color spaces, see *Quartz 2D Programming Guide*.

You can pass a `CGColorRef` object that represents any color space, including CMYK, but Core Image converts all color spaces to the Core Image working color space before it passes the color space to the filter kernel. The Core Image working color space uses three color components plus alpha.

Availability

Mac OS X v10.4 and later.

See Also

`+ colorWithRed:green:blue:` (page 7)
`+ colorWithRed:green:blue:alpha:` (page 8)
`+ colorWithString:` (page 9)

Declared In

`CIColor.h`

colorWithRed:green:blue:

Creates a color object using the specified RGB color component values

```
+ (CIColor *)colorWithRed:(CGFloat)r green:(CGFloat)g blue:(CGFloat)b
```

Parameters

r

The value of the red component.

g

The value of the green component.

b

The value of the blue component.

Return Value
A Core Image color object that represents an RGB color in the color space specified by the Quartz 2D constant `kCGColorSpaceGenericRGB`.

Availability
Mac OS X v10.4 and later.

See Also
`+ colorWithCGColor:` (page 7)
`+ colorWithRed:green:blue:alpha:` (page 8)
`+ colorWithString:` (page 9)

Declared In
`CIColor.h`

colorWithRed:green:blue:alpha:

Creates a color object using the specified RGBA color component values.

```
+ (CIColor *)colorWithRed:(CGFloat)r green:(CGFloat)g blue:(CGFloat)b
    alpha:(CGFloat)a
```

Parameters

r
The value of the red component.

g
The value of the green component.

b
The value of the blue component.

a
The value of the alpha component.

Return Value
A Core Image color object that represents an RGB color in the color space specified by the Quartz 2D constant `kCGColorSpaceGenericRGB` and an alpha value.

Availability
Mac OS X v10.4 and later.

See Also
`+ colorWithCGColor:` (page 7)
`+ colorWithRed:green:blue:` (page 7)
`+ colorWithString:` (page 9)

Related Sample Code
CIAnnotation

Declared In
`CIColor.h`

colorWithString:

Creates a color object using the RGBA color component values specified by a string.

```
+ (CIColor *)colorWithString:(NSString *)representation
```

Parameters

representation

A string that is in one of the formats returned by the `stringRepresentation` method. For example, the string:

```
@"0.5 0.7 0.3 1.0"
```

indicates an RGB color whose components are 50% red, 70% green, 30% blue, and 100% opaque (alpha value of 1.0). The string representation always has four components—red, green, blue, and alpha. The default value for the alpha component is `1.0`.

Return Value

A Core Image color object that represents an RGB color in the color space specified by the Quartz 2D constant `kCGColorSpaceGenericRGB`.

Availability

Mac OS X v10.4 and later.

See Also

`+ colorWithCGColor:` (page 7)

`+ colorWithRed:green:blue:` (page 7)

`+ colorWithRed:green:blue:alpha:` (page 8)

Declared In

`CIColor.h`

Instance Methods

alpha

Returns the alpha value of the color.

```
- (CGFloat)alpha
```

Return Value

The alpha value. A color created without an explicit alpha value has an alpha of 1.0 by default.

Availability

Mac OS X v10.4 and later.

See Also

`- components` (page 10)

Declared In

`CIColor.h`

blue

Returns the blue component of the color.

```
- (CGFloat)blue
```

Return Value
The unpremultiplied blue component of the color.

Availability
Mac OS X v10.4 and later.

See Also
`- components` (page 10)

Declared In
`CIColor.h`

colorSpace

Returns the Quartz 2D color space associated with the color.

```
- (CGColorSpaceRef)colorSpace
```

Return Value
The Quartz 2D color space (`CGColorSpaceRef` object). You are responsible for disposing of this color space by calling the Quartz 2D function `CGColorSpaceRelease`.

Availability
Mac OS X v10.4 and later.

See Also
`- components` (page 10)

Declared In
`CIColor.h`

components

Returns the color components of the color.

```
- (const CGFloat *)components
```

Return Value
An array of color components, specified as floating-point values in the range of 0.0 through 1.0. This array includes an alpha component if there is one.

Availability
Mac OS X v10.4 and later.

See Also
`- numberOfComponents` (page 11)
`- stringRepresentation` (page 12)

Declared In
`CIColor.h`

green

Returns the green component of the color.

```
- (CGFloat)green
```

Return Value
The unpremultiplied green component of the color.

Availability
Mac OS X v10.4 and later.

See Also
`- components` (page 10)

Declared In
`CIColor.h`

initWithCGColor:

Initializes a color object with a Quartz color.

```
- (id)initWithCGColor:(CGColorRef)c
```

Parameters

c

A Quartz color (`CGColorRef`) created using a Quartz color creation function such as `CGColorCreate`.

Discussion
A `CGColorRef` object is the fundamental opaque data type used internally by Quartz to represent colors. For more information on Quartz 2D color and color spaces, see *Quartz 2D Programming Guide*.

You can pass a `CGColorRef` object that represents any color space, including CMYK, but Core Image converts all color spaces to the Core Image working color space before it passes the color space to the filter kernel. The Core Image working color space uses three color components plus alpha.

Availability
Mac OS X v10.4 and later.

Declared In
`CIColor.h`

numberOfComponents

Returns the number of color components in the color.

```
- (size_t)numberOfComponents
```

Return Value
The number of color components, which includes an alpha component if there is one.

Availability
Mac OS X v10.4 and later.

See Also
`- components` (page 10)

Declared In
`CIColor.h`

red

Returns the red component of the color.

```
- (CGFloat)red
```

Return Value
The unpremultiplied red component of the color.

Availability
Mac OS X v10.4 and later.

See Also
`- components` (page 10)

Declared In
`CIColor.h`

stringRepresentation

Returns a formatted string that specifies the components of the color.

```
- (NSString *)stringRepresentation
```

Return Value
The formatted string.

Discussion
The string representation always has four components—red, green, blue, and alpha. The default value for the alpha component is `1.0`.F or example, this string:

```
@"0.5 0.7 0.3 1.0"
```

indicates an RGB color whose components are 50% red, 70% green, 30% blue, and 100% opaque (alpha value of 1.0).

Availability
Mac OS X v10.4 and later.

See Also
`- components` (page 10)

Declared In
`CIColor.h`

Document Revision History

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

Date	Notes
2006-12-05	Revised class meta information.
2006-05-23	First publication of this content as a separate document.
	Added parameter descriptions and updated Class Description.

Index