
CGGradient Reference

[Graphics & Imaging > Quartz](#)



2007-05-01



Apple Inc.
© 2007 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, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple 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

CGGradient Reference 5

- Overview 5
- Functions by Task 5
 - Creating a CGGradient Object 5
 - Retaining and Releasing a CGGradient Object 5
 - Getting the Type ID for a CGGradient Object 6
- Functions 6
 - CGGradientCreateWithColorComponents 6
 - CGGradientCreateWithColors 7
 - CGGradientGetTypeID 7
 - CGGradientRelease 8
 - CGGradientRetain 8
- Data Types 9
 - CGGradientRef 9
- Constants 9
 - Gradient Drawing Options 9

Document Revision History 11

Index 13

CGGradient Reference

Derived From:	CType
Framework:	ApplicationServices/ApplicationServices.h
Companion guide	Quartz 2D Programming Guide
Declared in	CGGradient.h

Overview

A gradient defines a smooth transition between colors across an area. The `CGGradientRef` opaque type, and the functions that operate on it, make creating and using radial and axial gradient fills an easy task. A `CGGradient` object has a color space, two or more colors, and a location for each color. The color space cannot be a pattern or indexed color space, otherwise it can be any Quartz color space (`CGColorSpaceRef`).

Colors can be provided as component values (such as red, green, blue) or as Quartz color objects (`CGColorRef`). In Quartz, component can vary from 0.0 to 1.0, designating the proportion of the component present in the color.

A location is a normalized value. When it comes time to paint the gradient, Quartz maps the normalized location values to the points in coordinate space that you provide.

If you want more precise control over gradients, or if your application runs in versions of Mac OS X that are earlier than v10.5, see *CGShading Reference*.

Functions by Task

Creating a CGGradient Object

[CGGradientCreateWithColorComponents](#) (page 6)

Creates a `CGGradient` object from a color space and the provided color components and locations.

[CGGradientCreateWithColors](#) (page 7)

Creates a `CGGradient` object from a color space and the provided color objects and locations.

Retaining and Releasing a CGGradient Object

[CGGradientRelease](#) (page 8)

Decrements the retain count of a `CGGradient` object.

[CGGradientRetain](#) (page 8)

Increments the retain count of a CGGradient object.

Getting the Type ID for a CGGradient Object

[CGGradientGetTypeID](#) (page 7)

Returns the Core Foundation type identifier for CGGradient objects.

Functions

CGGradientCreateWithColorComponents

Creates a CGGradient object from a color space and the provided color components and locations.

```
CGGradientRef CGGradientCreateWithColorComponents(
    CGColorSpaceRef space,
    const CGFloat components[],
    const CGFloat locations[],
    size_t count
);
```

Parameters

space

The color space to use for the gradient. You cannot use a pattern or indexed color space.

components

The color components for each color that defines the gradient. The components should be in the color space specified by *space*. If you are unsure of the number of components, you can call the function `CGColorSpaceGetNumberOfComponents`.

The number of items in this array should be the product of *count* and the number of components in the color space. For example, if the color space is an RGBA color space and you want to use two colors in the gradient (one for a starting location and another for an ending location), then you need to provide 8 values in *components*—red, green, blue, and alpha values for the first color, followed by red, green, blue, and alpha values for the second color.

locations

The location for each color provided in *components*. Each location must be a `CGFloat` value in the range of 0 to 1, inclusive. If 0 and 1 are not in the *locations* array, Quartz uses the colors provided that are closest to 0 and 1 for those locations.

If *locations* is `NULL`, the first color in *components* is assigned to location 0, the last color in *components* is assigned to location 1, and intervening colors are assigned locations that are at equal intervals in between.

count

The number of locations provided in the *locations* parameters.

Return Value

A CGGradient object.

Availability

Available in Mac OS X v10.5 and later.

See Also

CGContextDrawLinearGradient

CGContextDrawRadialGradient

Declared In

CGGradient.h

CGGradientCreateWithColors

Creates a CGGradient object from a color space and the provided color objects and locations.

```
CGGradientRef CGGradientCreateWithColors(
    CGColorSpaceRef space,
    CFArrayRef colors,
    const CGFloat locations[]
);
```

Parameters*space*

The color space to use for the gradient. You cannot use a pattern or indexed color space.

colors

A non-empty array of CGColor objects that should be in the color space specified by *space*. If *space* is not NULL, each color will be converted (if necessary) to that color space and the gradient will drawn in that color space. Otherwise, each color will be converted to and drawn in the GenericRGB color space.

locations

The location for each color provided in *colors*; each location must be a CGFloat value in the range of 0 to 1, inclusive. If 0 and 1 are not in the *locations* array, Quartz uses the colors provided that are closest to 0 and 1 for those locations.

If *locations* is NULL, the first color in *colors* is assigned to location 0, the last color in *colors* is assigned to location 1, and intervening colors are assigned locations that are at equal intervals in between.

The *locations* array should contain the same number of items as the *colors* array.

Return Value

A CGGradient object.

Availability

Available in Mac OS X v10.5 and later.

See Also

CGContextDrawLinearGradient

CGContextDrawRadialGradient

Declared In

CGGradient.h

CGGradientGetTypeID

Returns the Core Foundation type identifier for CGGradient objects.

```

CTypeID CGGradientGetTypeID (
    void
);

```

Return Value

The Core Foundation identifier for the opaque type `CGGradientRef`.

Availability

Available in Mac OS X version 10.5 and later.

Declared In

`CGGradient.h`

CGGradientRelease

Decrements the retain count of a `CGGradient` object.

```

void CGGradientRelease (
    CGGradientRef gradient
);

```

Parameters

gradient

The gradient object to release.

Discussion

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

Availability

Available in Mac OS X version 10.2 and later.

Declared In

`CGGradient.h`

CGGradientRetain

Increments the retain count of a `CGGradient` object.

```

CGGradientRef CGGradientRetain(
    CGGradientRef gradient
);

```

Parameters

gradient

The gradient object to retain.

Return Value

The same gradient object that you passed in as the *gradient* parameter.

Discussion

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

Availability

Available in Mac OS X version 10.5 and later.

Declared In

CGGradient.h

Data Types

CGGradientRef

An opaque type that represents a Quartz gradient.

```
typedef struct CGGradient *CGGradientRef;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

CGGradient.h

Constants

Gradient Drawing Options

Drawing locations for gradients.

```
enum {
    kCGGradientDrawsBeforeStartLocation = (1 << 0),
    kCGGradientDrawsAfterEndLocation = (1 << 1)
};
typedef enum CGGradientDrawingOptions CGGradientDrawingOptions;
```

Constants

`kCGGradientDrawsBeforeStartLocation`

The fill should extend beyond the starting location. The color that extends beyond the starting point is the solid color defined by the CGGradient object to be at location 0.

Available in Mac OS X v10.5 and later.

Declared in CGGradient.h.

`kCGGradientDrawsAfterEndLocation`

The fill should extend beyond the ending location. The color that extends beyond the ending point is the solid color defined by the CGGradient object to be at location 1.

Available in Mac OS X v10.5 and later.

Declared in CGGradient.h.

Declared In

CGGradient.h

Document Revision History

This table describes the changes to *CGGradient Reference*.

Date	Notes
2007-05-01	New document that describes the functions that create and operate on Quartz gradient objects.

REVISION HISTORY

Document Revision History

Index

C

CGGradientCreateWithColorComponents **function** [6](#)
CGGradientCreateWithColors **function** [7](#)
CGGradientGetTypeID **function** [7](#)
CGGradientRef **data type** [9](#)
CGGradientRelease **function** [8](#)
CGGradientRetain **function** [8](#)

G

Gradient Drawing Options [9](#)

K

kCGGradientDrawsAfterEndLocation **constant** [9](#)
kCGGradientDrawsBeforeStartLocation **constant**
[9](#)