
CGShading Reference

Graphics & Animation: 2D Drawing



2006-12-22



Apple Inc.
© 2003, 2006 Apple Computer, 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, iPhone, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

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

CGShading Reference 5

Overview	5
Functions by Task	5
Creating Shading Objects	5
Retaining and Releasing Shading Objects	5
Getting the CType ID	6
Functions	6
CGShadingCreateAxial	6
CGShadingCreateRadial	7
CGShadingGetTypeID	7
CGShadingRelease	8
CGShadingRetain	8
Data Types	9
CGShadingRef	9

Document Revision History 11

CGShading Reference

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

Overview

`CGShadingRef` is an opaque type used to define linear (axial) and radial gradient fills whose color transitions are controlled by a function (`CGFunctionRef`) that you provide. Shading means to fill using a smooth transition between colors across an area. To paint with a Quartz shading, you call `CGContextDrawShading`. This function fills the current clipping path using the specified color gradient, calling your parametric function repeatedly as it draws.

An alternative to using a `CGShading` object is to use the `CGGradientRef` opaque type. For applications that run in Mac OS X v10.5 and later, `CGGradient` objects are much simpler to use. (See *CGGradient Reference*.)

Functions by Task

Creating Shading Objects

[CGShadingCreateAxial](#) (page 6)

Creates a shading object to use for axial shading.

[CGShadingCreateRadial](#) (page 7)

Creates a shading object to use for radial shading.

Retaining and Releasing Shading Objects

[CGShadingRetain](#) (page 8)

Increments the retain count of a shading object.

[CGShadingRelease](#) (page 8)

Decrements the retain count of a shading object.

Getting the CType ID

[CGShadingGetTypeID](#) (page 7)

Returns the Core Foundation type identifier for Quartz shading objects.

Functions

CGShadingCreateAxial

Creates a shading object to use for axial shading.

```
CGShadingRef CGShadingCreateAxial (
    CGColorSpaceRef colorspace,
    CGPoint start,
    CGPoint end,
    CGFunctionRef function,
    bool extendStart,
    bool extendEnd
);
```

Parameters

colorspace

The color space in which color values are expressed. Quartz retains this object; upon return, you may safely release it.

start

The starting point of the axis, in the shading's target coordinate space.

end

The ending point of the axis, in the shading's target coordinate space.

function

A `CGFunction` object created by the function `CGFunctionCreate`. This object refers to your function for creating an axial shading. Quartz retains this object; upon return, you may safely release it.

extendStart

A Boolean value that specifies whether to extend the shading beyond the starting point of the axis.

extendEnd

A Boolean value that specifies whether to extend the shading beyond the ending point of the axis.

Return Value

A new Quartz axial shading. You are responsible for releasing this object using [CGShadingRelease](#) (page 8).

Discussion

An axial shading is a color blend that varies along a linear axis between two endpoints and extends indefinitely perpendicular to that axis. When you are ready to draw the shading, call the function `CGContextDrawShading`.

Availability

Declared In

`CGShading.h`

CGShadingCreateRadial

Creates a shading object to use for radial shading.

```
CGShadingRef CGShadingCreateRadial (
    CGColorSpaceRef colorspace,
    CGPoint start,
    CGFloat startRadius,
    CGPoint end,
    CGFloat endRadius,
    CGFunctionRef function,
    bool extendStart,
    bool extendEnd
);
```

Parameters

colorspace

The color space in which color values are expressed. Quartz retains this object; upon return, you may safely release it.

start

The center of the starting circle, in the shading's target coordinate space.

startRadius

The radius of the starting circle, in the shading's target coordinate space.

end

The center of the ending circle, in the shading's target coordinate space.

endRadius

The radius of the ending circle, in the shading's target coordinate space.

function

A CGFunction object created by the function `CGFunctionCreate`. This object refers to your function for creating a radial shading. Quartz retains this object; upon return, you may safely release it.

extendStart

A Boolean value that specifies whether to extend the shading beyond the starting circle.

extendEnd

A Boolean value that specifies whether to extend the shading beyond the ending circle.

Return Value

A new Quartz radial shading. You are responsible for releasing this object using [CGShadingRelease](#) (page 8).

Discussion

A radial shading is a color blend that varies between two circles. To draw the shading, call the function `CGContextDrawShading`.

Availability

Declared In

`CGShading.h`

CGShadingGetTypeID

Returns the Core Foundation type identifier for Quartz shading objects.

```

CTypeID CGShadingGetTypeID (
    void
);

```

Return Value

The Core Foundation identifier for the opaque type [CGShadingRef](#) (page 9).

Availability**Declared In**

CGShading.h

CGShadingRelease

Decrements the retain count of a shading object.

```

void CGShadingRelease (
    CGShadingRef shading
);

```

Parameters

shading

The shading object to release.

Discussion

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

Availability**Declared In**

CGShading.h

CGShadingRetain

Increments the retain count of a shading object.

```

CGShadingRef CGShadingRetain (
    CGShadingRef shading
);

```

Parameters

shading

The shading object to retain.

Return Value

The same shading object you passed in as the *shading* parameter.

Discussion

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

Availability**Declared In**

CGShading.h

Data Types

CGShadingRef

An opaque type that represents a Quartz shading.

```
typedef struct CGShading *CGShadingRef;
```

Availability

Available in iOS 2.0 and later.

Declared In

CGShading.h

Document Revision History

This table describes the changes to *CGShading Reference*.

Date	Notes
2006-12-22	Updated the introduction to provide a contrast with the <code>CGGradient</code> opaque type.
	All instances of the <code>float</code> data type were changed to the <code>CGFloat</code> data type.
2005-11-09	Added that Quartz retains the <code>CGFunction</code> object passed to a shading function.
2005-04-29	Revised introduction and added a few sentences to two functions.
	See CGShadingCreateAxial (page 6) and CGShadingCreateRadial (page 7).
2004-02-26	First version of this document. An earlier version of this information appeared in <i>Quartz 2D Reference</i> .

REVISION HISTORY

Document Revision History