

---

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

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

---

## **NSGradient Class Reference 7**

---

Overview 7

Tasks 8

    Initialization 8

    Primitive Drawing Methods 8

    Drawing Linear Gradients 8

    Drawing Radial Gradients 8

    Getting Gradient Properties 8

Instance Methods 9

    colorSpace 9

    drawFromCenter:radius:toCenter:radius:options: 9

    drawFromPoint:toPoint:options: 10

    drawInBezierPath:angle: 10

    drawInBezierPath:relativeCenterPosition: 11

    drawInRect:angle: 12

    drawInRect:relativeCenterPosition: 13

    getColor:location:atIndex: 13

    initWithColors: 14

    initWithColors:atLocations:colorSpace: 15

    initWithColorsAndLocations: 15

    initWithStartingColor:endingColor: 16

    interpolatedColorAtLocation: 17

    numberOfColorStops 17

Constants 17

    NSGradientDrawingOptions 17

    Gradient Drawing Options 18

---

## **Document Revision History 19**

---

## **Index 21**

---



# Tables

## NSGradient Class Reference 7

---

Table 1      Linear gradient starting points. 12



# NSGradient Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding NSCopying NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Companion guide</b>	Cocoa Drawing Guide
<b>Declared in</b>	NSGradient.h
<b>Related sample code</b>	CocoaSlides Grady IconCollection MenuItemView Squiggles

## Overview

The `NSGradient` class provides support for drawing gradient fill colors, also known as shadings in Quartz. This class provides convenience methods for drawing radial or linear (axial) gradients for rectangles and `NSBezierPath` objects. It also supports primitive methods that let you customize the shape of the gradient fill.

A gradient consists of two or more color changes over the range of the gradient shape. When creating a gradient object, you specify the colors and their locations relative to the start and end of the gradient. This combination of color and location is known as a **color stop**. During drawing, the `NSGradient` object uses the color stop information to compute color changes for you and passes that information to the Quartz shading functions.

Because the `NSGradient` class uses Quartz shadings, drawing is handled by computing the colors at a given point mathematically. This technique results in smooth gradients regardless of the resolution of the target device.

For more information about gradients and their appearance, see *Gradients in Quartz 2D Programming Guide*.

## Tasks

### Initialization

- [initWithStartingColor:endingColor:](#) (page 16)  
Initializes a newly allocated gradient object with two colors.
- [initWithColors:](#) (page 14)  
Initializes a newly allocated gradient object with an array of colors.
- [initWithColorsAndLocations:](#) (page 15)  
Initializes a newly allocated gradient object with a comma-separated list of arguments.
- [initWithColors:atLocations:colorSpace:](#) (page 15)  
Initializes a newly allocated gradient object with the specified colors, color locations, and color space.

### Primitive Drawing Methods

- [drawFromPoint:toPoint:options:](#) (page 10)  
Draws a linear gradient between the specified start and end points.
- [drawFromCenter:radius:toCenter:radius:options:](#) (page 9)  
Draws a radial gradient between the specified circles.

### Drawing Linear Gradients

- [drawInRect:angle:](#) (page 12)  
Fills the specified rectangle with a linear gradient.
- [drawInBezierPath:angle:](#) (page 10)  
Fills the specified path with a linear gradient.

### Drawing Radial Gradients

- [drawInRect:relativeCenterPosition:](#) (page 13)  
Draws a radial gradient starting at the center of the specified rectangle.
- [drawInBezierPath:relativeCenterPosition:](#) (page 11)  
Draws a radial gradient starting at the center point of the specified path.

### Getting Gradient Properties

- [colorSpace](#) (page 9)  
Returns the color space of the colors associated with the receiver.
- [numberOfColorStops](#) (page 17)  
Returns the number of color stops associated with the receiver.

- [getColor:location:atIndex:](#) (page 13)  
Returns information about the color stop at the specified index in the receiver's color array.
- [interpolatedColorAtLocation:](#) (page 17)  
Returns the color of the rendered gradient at the specified relative location.

## Instance Methods

### colorSpace

Returns the color space of the colors associated with the receiver.

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

#### Return Value

The color space object used by the receiver's colors.

#### Discussion

When the receiver is initialized, colors that do not conform to the receiver's color space are converted automatically.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

NSGradient.h

### drawFromCenter:radius:toCenter:radius:options:

Draws a radial gradient between the specified circles.

```
- (void)drawFromCenter:(NSPoint)startCenter radius:(CGFloat)startRadius  
toCenter:(NSPoint)endCenter radius:(CGFloat)endRadius  
options:(NSGradientDrawingOptions)options
```

#### Parameters

*startCenter*

The center point of the circle that represents the beginning of the gradient.

*startRadius*

The radius of the circle that represents the beginning of the gradient.

*endCenter*

The center point of the circle that represents the end of the gradient.

*endRadius*

The radius of the circle that represents the end of the gradient.

*options*

The gradient options, if any. You can use these options to extend the gradient size beyond the start and end circles. For more information, see [“Gradient Drawing Options”](#) (page 18).

**Discussion**

This method draws a radial gradient pattern starting at the first circle and ending at the second circle. The gradient color transitions occur in circular bands emanating from the starting circle and ending at the second circle.

This is a primitive method used by the `NSGradient` class to draw radial gradients. Because this method does not perform any clipping of the gradient fill pattern, you must ensure that the clipping region is configured properly if you intend to invoke this method directly. By default, the clipping region is set to the current view or window in which drawing occurs.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`NSGradient.h`

**drawFromPoint:toPoint:options:**

Draws a linear gradient between the specified start and end points.

```
- (void)drawFromPoint:(NSPoint)startingPoint toPoint:(NSPoint)endingPoint
options:(NSGradientDrawingOptions)options
```

**Parameters**

*startingPoint*

The starting point for the gradient, in the local coordinate system. The gradient's first color is drawn at this point.

*endingPoint*

The end point for the gradient, in the local coordinate system. The gradient's last color is drawn at this point.

*options*

The gradient options, if any. You can use these options to extend the gradient size beyond the start and end points. For more information, see [“Gradient Drawing Options”](#) (page 18).

**Discussion**

This method draws the gradient color changes along the line formed by the two points. The gradient fill extends perpendicularly outward from line until it reaches the edges of the current clipping region.

This is a primitive method used by the `NSGradient` class to draw linear gradients. Because this method does not perform any clipping of the gradient fill pattern, you must ensure that the clipping region is configured properly if you intend to invoke this method directly. By default, the clipping region is set to the current view or window in which drawing occurs.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`NSGradient.h`

**drawInBezierPath:angle:**

Fills the specified path with a linear gradient.

```
- (void)drawInBezierPath:(NSBezierPath *)path angle:(CGFloat)angle
```

### Parameters

*path*

The path object to fill.

*angle*

The angle of the linear gradient, specified in degrees. Positive values indicate rotation in the counter-clockwise direction relative to the horizontal axis.

### Discussion

This convenience method behaves in a similar way to the `drawInRect:angle:` method, with the path object replacing the rectangle as the clipping region. Like the other method, the start and end colors are guaranteed to be visible at the farthest ends of the path.

The gradient formed by this method is clipped to *path*.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [drawInRect:angle:](#) (page 12)
- [drawFromPoint:toPoint:options:](#) (page 10)

### Related Sample Code

CocoaSlides

### Declared In

NSGradient.h

## drawInBezierPath:relativeCenterPosition:

Draws a radial gradient starting at the center point of the specified path.

```
- (void)drawInBezierPath:(NSBezierPath *)path
    relativeCenterPosition:(NSPoint)relativeCenterPosition
```

### Parameters

*path*

The path to fill.

*relativeCenterPosition*

The relative location within the bounding rectangle of *path* to use as the center point of the gradient's end circle. Each coordinate must contain a value between -1.0 and 1.0. A coordinate value of 0 represents the center of the path's bounding rectangle along the given axis. In the default coordinate system, a value of -1.0 corresponds to the bottom or left edge of the bounding rectangle and a value of 1.0 corresponds to the top or right edge.

### Discussion

The center point of the starting circle is the same as the center point of *path*. The radius of the starting circle is 0, resulting in the starting circle being just a point.

The center point of the end circle starts at the center point of *path* and is modified by the value in the *relativeCenterPosition* parameter. For example, if *relativeCenterPosition* contains the point (1.0, 1.0), the center of the end circle is located in the top-right corner of the path's bounding rectangle. The radius of the end circle is set to the smallest value that ensures *rect* is covered by the end circle.

The gradient formed by this method is clipped to *path*.

#### Availability

Available in Mac OS X v10.5 and later.

#### See Also

- [drawInRect:relativeCenterPosition:](#) (page 13)
- [drawFromCenter:radius:toCenter:radius:options:](#) (page 9)

#### Declared In

NSGradient.h

## drawInRect:angle:

Fills the specified rectangle with a linear gradient.

```
- (void)drawInRect:(NSRect)rect angle:(CGFloat)angle
```

#### Parameters

*rect*

The rectangle to fill.

*angle*

The angle of the linear gradient, specified in degrees. Positive values indicate rotation in the counter-clockwise direction relative to the horizontal axis.

#### Discussion

This convenience method draws a linear gradient inside the specified rectangle. The gradient is drawn so that the start and end colors are guaranteed to be visible in opposite corners of the rectangle. The angle of rotation determines which corner contains the start color; see Table 1.

**Table 1** Linear gradient starting points.

Rotation angle	Start corner
0-89 degrees	Lower-left
90-179 degrees	Lower-right
180-269 degrees	Upper-right
270-359 degrees	Upper-left

The gradient's color transitions occur along the line formed by the angle of rotation. For example, a rotation of 0 degrees results in colors changing from left-to-right across the rectangle, while a rotation of 90 degrees results in colors changing from bottom to top.

The gradient drawn by this method is clipped to *rect*.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [drawInBezierPath:angle:](#) (page 10)
- [drawFromPoint:toPoint:options:](#) (page 10)

**Declared In**

NSGradient.h

**drawInRect:relativeCenterPosition:**

Draws a radial gradient starting at the center of the specified rectangle.

```
- (void)drawInRect:(NSRect)rect
    relativeCenterPosition:(NSPoint)relativeCenterPosition
```

**Parameters**

*rect*

The rectangle to fill.

*relativeCenterPosition*

The relative location within the rectangle to use as the center point of the gradient's end circle. Each coordinate must contain a value between -1.0 and 1.0. A coordinate value of 0 represents the center of *rect* along the given axis. In the default coordinate system, a value of -1.0 corresponds to the bottom or left edge of the rectangle and a value of 1.0 corresponds to the top or right edge.

**Discussion**

The center point of the starting circle is the same as the center point of *rect*. The radius of the starting circle is 0, resulting in the starting circle being just a point.

The center point of the end circle starts at the center point of *rect* and is modified by the value in the *relativeCenterPosition* parameter. For example, if *relativeCenterPosition* contains the point (1.0, 1.0), the center of the end circle is located in the top-right corner of *rect*. The radius of the end circle is set to the smallest value that ensures *rect* is covered by the end circle.

The gradient formed by this method is clipped to *rect*.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [drawInRect:relativeCenterPosition:](#) (page 13)
- [drawFromCenter:radius:toCenter:radius:options:](#) (page 9)

**Declared In**

NSGradient.h

**getColor:location:atIndex:**

Returns information about the color stop at the specified index in the receiver's color array.

```
- (void)getColor:(NSColor **)color location:(CGFloat *)location
  atIndex:(NSInteger)index
```

**Parameters***color*

On input, a pointer to a color object. On output, the color at the specified index in the receiver's color array. You may specify `nil` if you are not interested in this parameter.

*location*

On input, a pointer to a floating point number. On output, contains the location value associated with the color. This value is between 0.0 and 1.0. It is used to determine the position of the color relative to the start and end points of the gradient. You may specify `NULL` if you are not interested in this parameter.

*index*

The index of the color you want.

**Discussion**

This method returns the color stop information that was used to create the receiver. It does not return the interpolated color values at any point along the gradient. The location of the gradient's first color is typically 0.0 and the location of the last color is typically 1.0, although the locations can vary depending on how the receiver was created.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [numberOfColorStops](#) (page 17)
- [interpolatedColorAtLocation:](#) (page 17)

**Declared In**

NSGradient.h

**initWithColors:**

Initializes a newly allocated gradient object with an array of colors.

```
- (id)initWithColors:(NSArray *)colorArray
```

**Parameters***colorArray*

An array of `NSColor` objects representing the colors to use to initialize the gradient. There must be at least two colors in the array. The first color is placed at location 0.0 and the last at location 1.0. If there are more than two colors, the additional colors are placed at evenly spaced intervals between the first and last colors.

**Return Value**

The initialized `NSGradient` object.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initWithColors:atLocations:colorSpace:](#) (page 15)

**Related Sample Code**

DispatchFractal

**Declared In**

NSGradient.h

**initWithColors:atLocations:colorSpace:**

Initializes a newly allocated gradient object with the specified colors, color locations, and color space.

```
- (id)initWithColors:(NSArray *)colorArray atLocations:(const CGFloat *)locations
    colorSpace:(NSColorSpace *)colorSpace
```

**Parameters***colorArray*

An array of `NSColor` objects representing the colors in the gradient.

*locations*

An array of `CGFloat` values containing the location for each color in the gradient. Each value must be in the range 0.0 to 1.0. There must be the same number of locations as are colors in the *colorArray* parameter.

*colorSpace*

The color space to use for the gradient.

**Return Value**

The initialized `NSGradient` object.

**Discussion**

This method is the designated initializer of `NSGradient`. The colors in the *colorArray* parameter are converted to the specified color space if they are not already in that color space.

Typically, at least one color should have a location of 0.0 and one should have a location of 1.0. If these locations are not specified, the color at the closest color stop is used to fill the gap.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSGradient.h

**initWithColorsAndLocations:**

Initializes a newly allocated gradient object with a comma-separated list of arguments.

```
- (id)initWithColorsAndLocations:(NSColor *)firstColor, ...
```

**Parameters***firstColor*

The first color in the gradient.

...

A comma-separated list of alternating `NSColor` objects and location arguments (specified as `CGFloat` values). The first value after *firstColor* must be a location. Each location value must be between 0.0 and 1.0. The list must be `nil`-terminated.

**Return Value**

The initialized `NSGradient` object.

**Discussion**

Typically, at least one color should have a location of 0.0 and one should have a location of 1.0. If these locations are not specified, the color at the closest color stop is used to fill the gap.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initWithColors:atLocations:colorSpace:](#) (page 15)

**Related Sample Code**

Squiggles

**Declared In**

`NSGradient.h`

**initWithStartingColor:endingColor:**

Initializes a newly allocated gradient object with two colors.

```
- (id)initWithStartingColor:(NSColor *)startingColor endingColor:(NSColor *)endingColor
```

**Parameters**

*startingColor*

The starting color of the gradient. The location of this color is fixed at 0.0.

*endingColor*

The ending color of the gradient. The location of this color is fixed at 1.0.

**Return Value**

The initialized `NSGradient` object.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [initWithColors:atLocations:colorSpace:](#) (page 15)

**Related Sample Code**

CocoaSlides

Grady

IconCollection

MenuItemView

**Declared In**

`NSGradient.h`

## interpolatedColorAtLocation:

Returns the color of the rendered gradient at the specified relative location.

```
- (NSColor *)interpolatedColorAtLocation:(CGFloat)location
```

### Parameters

*location*

The location value for the color you want. This value must be between 0.0 and 1.0. This value need not correspond to the location of one of the color objects used to create the gradient.

### Discussion

This method does not simply return the color values used to initialize the receiver. Instead, it computes the value that would be drawn at the specified location.

The start color of the gradient is always located at 0.0 and the end color is always at 1.0.

### Availability

Available in Mac OS X v10.5 and later.

### Related Sample Code

DispatchFractal

### Declared In

NSGradient.h

## numberOfColorStops

Returns the number of color stops associated with the receiver.

```
- (NSInteger)numberOfColorStops
```

### Return Value

The number of colors in the receiver's color array.

### Discussion

Gradients must have at least two color stops: one defining the location of the start color and one defining the location of the end color. Gradients may have additional color stops located at different transition points in between the start and end stops.

### Availability

Available in Mac OS X v10.5 and later.

### Declared In

NSGradient.h

## Constants

### NSGradientDrawingOptions

Specifies gradient drawing options.

```
typedef NSUInteger NSGradientDrawingOptions;
```

**Discussion**

The constant values associated with this type are listed in [“Gradient Drawing Options”](#) (page 18).

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSGradient.h

## Gradient Drawing Options

These constants are used by the primitive drawing methods to determine if drawing occurs outside of the gradient start and end locations.

```
enum {
    NSGradientDrawsBeforeStartingLocation = (1 << 0),
    NSGradientDrawsAfterEndingLocation = (1 << 1),
};
```

**Constants**

NSGradientDrawsBeforeStartingLocation

Drawing extends before the gradient starting point.

Available in Mac OS X v10.5 and later.

Declared in NSGradient.h.

NSGradientDrawsAfterEndingLocation

Drawing extends beyond the gradient end point.

Available in Mac OS X v10.5 and later.

Declared in NSGradient.h.

**Declared In**

AppKit/NSGradient.h

# Document Revision History

---

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

Date	Notes
2009-01-06	Updated the method descriptions that involve drawing radial gradients with relative center positions.
2007-04-12	New document describing methods for creating gradient fill patterns.

## REVISION HISTORY

### Document Revision History

# Index

---

## C

---

colorSpace **instance method** [9](#)

## D

---

drawFromCenter:radius:toCenter:radius:options:  
**instance method** [9](#)

drawFromPoint:toPoint:options: **instance method**  
[10](#)

drawInBezierPath:angle: **instance method** [10](#)

drawInBezierPath:relativeCenterPosition:  
**instance method** [11](#)

drawInRect:angle: **instance method** [12](#)

drawInRect:relativeCenterPosition: **instance**  
**method** [13](#)

## G

---

getColor:location:atIndex: **instance method** [13](#)

Gradient Drawing Options [18](#)

## I

---

initWithColorsAndLocations: **instance method** [15](#)

initWithColors: **instance method** [14](#)

initWithColors:atLocations:colorSpace: **instance**  
**method** [15](#)

initWithStartingColor:endingColor: **instance**  
**method** [16](#)

interpolatedColorAtLocation: **instance method** [17](#)

## N

---

NSGradientDrawingOptions **data type** [17](#)

NSGradientDrawsAfterEndingLocation **constant** [18](#)

NSGradientDrawsBeforeStartingLocation **constant**  
[18](#)

numberOfColorStops **instance method** [17](#)