
CGPattern Reference

[Graphics & Imaging](#) > Quartz



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, 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

CGPattern Reference 5

- Overview 5
- Functions by Task 5
 - Creating a Pattern 5
 - Getting the CType ID 5
 - Retaining and Releasing a Pattern 5
- Functions 6
 - CGPatternCreate 6
 - CGPatternGetTypeID 7
 - CGPatternRelease 7
 - CGPatternRetain 8
- Callbacks 8
 - CGPatternDrawPatternCallback 8
 - CGPatternReleaseInfoCallback 9
- Data Types 10
 - CGPatternRef 10
 - CGPatternCallbacks 10
- Constants 11
 - Tiling Patterns 11

Document Revision History 13

Index 15

CGPattern Reference

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

Overview

The `CGPatternRef` opaque type represents a pattern that you can use to stroke along or fill in a graphics path. Quartz tiles the pattern cell for you, based on parameters you specify when you call [CGPatternCreate](#) (page 6).

To create a dashed line, see `CGContextSetLineDash` in *CGContext Reference*.

Functions by Task

Creating a Pattern

[CGPatternCreate](#) (page 6)
Creates a pattern object.

Getting the CType ID

[CGPatternGetTypeID](#) (page 7)
Returns the type identifier for Quartz patterns.

Retaining and Releasing a Pattern

[CGPatternRetain](#) (page 8)
Increments the retain count of a Quartz pattern.

[CGPatternRelease](#) (page 7)
Decrements the retain count of a Quartz pattern.

Functions

CGPatternCreate

Creates a pattern object.

```
CGPatternRef CGPatternCreate (
    void *info,
    CGRect bounds,
    CGAffineTransform matrix,
    CGFloat xStep,
    CGFloat yStep,
    CGPatternTiling tiling,
    bool isColored,
    const CGPatternCallbacks *callbacks
);
```

Parameters

info

A pointer to private storage used by your pattern drawing function, or `NULL`. For more information, see the discussion below.

bounds

The bounding box of the pattern cell, specified in pattern space. (Pattern space is an abstract space that maps to the default user space by the transformation matrix you specify with the `matrix` parameter.) The drawing done in your pattern drawing function is clipped to this rectangle.

matrix

A matrix that represents a transform from pattern space to the default user space of the context in which the pattern is used. If no transform is needed, pass the identity matrix.

xStep

The horizontal displacement between cells, specified in pattern space. For no additional horizontal space between cells (so that each pattern cell abuts the previous pattern cell in the horizontal direction), pass the width of the pattern cell.

yStep

The vertical displacement between cells, specified in pattern space. For no additional vertical space between cells (so that each pattern cell abuts the previous pattern cell in the vertical direction), pass the height of the pattern cell.

tiling

A `CGPatternTiling` constant that specifies the desired tiling method. For more information about tiling methods, see “[Tiling Patterns](#)” (page 11).

isColored

If you want to draw your pattern using its own intrinsic color, pass `true`. If you want to draw an uncolored (or masking) pattern that uses the fill or stroke color in the graphics state, pass `false`.

callbacks

A pointer to a pattern callback function table—your pattern drawing function is an entry in this table. See [CGPatternCallbacks](#) (page 10) for more information about callback function tables for patterns.

Return Value

A new Quartz pattern. You are responsible for releasing this object using [CGPatternRelease](#) (page 7).

Discussion

Quartz calls your drawing function at the appropriate time to draw the pattern cell. A pattern cell must be invariant—that is, the pattern cell should be drawn exactly the same way each time the drawing function is called.

The appearance of a pattern cell is unaffected by changes in the graphics state of the context in which the pattern is used.

See [CGPatternDrawPatternCallback](#) (page 8) for more information about pattern drawing functions.

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CGPattern.h

CGPatternGetTypeID

Returns the type identifier for Quartz patterns.

```
CTypeID CGPatternGetTypeID (
    void
);
```

Return Value

The identifier for the opaque type [CGPatternRef](#) (page 10).

Availability

Available in Mac OS X version 10.2 and later.

Declared In

CGPattern.h

CGPatternRelease

Decrements the retain count of a Quartz pattern.

```
void CGPatternRelease (
    CGPatternRef pattern
);
```

Parameters

pattern

The pattern to release.

Discussion

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

Availability

Available in Mac OS X version 10.1 and later.

Declared In

CGPattern.h

CGPatternRetain

Increments the retain count of a Quartz pattern.

```
CGPatternRef CGPatternRetain (
    CGPatternRef pattern
);
```

Parameters

pattern

The pattern to retain.

Return Value

The same pattern you passed in as the *pattern* parameter.

Discussion

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

Availability

Available in Mac OS X version 10.1 and later.

Declared In

`CGPattern.h`

Callbacks

CGPatternDrawPatternCallback

Draws a pattern cell.

```
typedef void (*CGPatternDrawPatternCallback) (
    void * info,
    CGContextRef context
);
```

If you name your function `MyDrawPattern`, you would declare it like this:

```
void MyDrawPattern (
    void * info,
    CGContextRef context
);
```

Parameters

info

A generic pointer to private data associated with the pattern. This is the same pointer you supplied to [CGPatternCreate](#) (page 6).

context

The graphics context for drawing the pattern cell.

Discussion

When a pattern is used to stroke or fill a graphics path, Quartz calls your custom drawing function at the appropriate time to draw the pattern cell. The cell should be drawn exactly the same way each time the drawing function is called.

In a drawing function associated with an uncolored pattern, you should not attempt to set a stroke or fill color or color space—if you do so, the result is undefined.

To learn how to associate your drawing function with a Quartz pattern, see [CGPatternCreate](#) (page 6) and [CGPatternCallbacks](#) (page 10).

Availability

Available in Mac OS X v10.2 and later.

Declared In

CGPattern.h

CGPatternReleaseInfoCallback

Release private data or resources associated with the pattern.

```
typedef void (*CGPatternReleaseInfoCallback) (
    void * info
);
```

If you name your function `MyCGPatternReleaseInfo`, you would declare it like this:

```
void MyCGPatternReleaseInfo (
    void * info
);
```

Parameters

info

A generic pointer to private data shared among your callback functions. This is the same pointer you supplied to [CGPatternCreate](#) (page 6).

Discussion

Quartz calls your release function when it frees your pattern object.

To learn how to associate your release function with a Quartz pattern, see [CGPatternCreate](#) (page 6) and [CGPatternCallbacks](#) (page 10).

Availability

Available in Mac OS X v10.2 and later.

Declared In

CGPattern.h

Data Types

CGPatternRef

An opaque type that represents a pattern.

```
typedef struct CGPattern * CGPatternRef;
```

Availability

Available in Mac OS X v10.1 and later.

Declared In

CGPattern.h

CGPatternCallbacks

A structure that holds a version and two callback functions for drawing a custom pattern.

```
struct CGPatternCallbacks {
    unsigned int version;
    CGPatternDrawPatternCallback drawPattern;
    CGPatternReleaseInfoCallback releaseInfo;
};
typedef struct CGPatternCallbacks CGPatternCallbacks;
```

Fields

version

The version of the structure passed in as a parameter to the [CGPatternCreate](#) (page 6). For this version of the structure, you should set this value to zero.

drawPattern

A pointer to a custom function that draws the pattern. For information about this callback function, see [CGPatternDrawPatternCallback](#) (page 8).

releaseInfo

An optional pointer to a custom function that's invoked when the pattern is released. [CGPatternReleaseInfoCallback](#) (page 9).

Discussion

You supply a `CGPatternCallbacks` structure to the function [CGPatternCreate](#) (page 6) to create a data provider for direct access. The functions specified by the `CGPatternCallbacks` structure are responsible for drawing the pattern and for handling the pattern's memory management.

Availability

Available in Mac OS X v10.1 and later.

Declared In

CGPattern.h

Constants

Tiling Patterns

Different methods for rendering a tiled pattern.

```
enum CGPatternTiling {
    kCGPatternTilingNoDistortion,
    kCGPatternTilingConstantSpacingMinimalDistortion,
    kCGPatternTilingConstantSpacing
};
typedef enum CGPatternTiling CGPatternTiling;
```

Constants

`kCGPatternTilingNoDistortion`

The pattern cell is not distorted when painted. The spacing between pattern cells may vary by as much as 1 device pixel.

Available in Mac OS X v10.1 and later.

Declared in `CGPattern.h`.

`kCGPatternTilingConstantSpacingMinimalDistortion`

Pattern cells are spaced consistently. The pattern cell may be distorted by as much as 1 device pixel when the pattern is painted.

Available in Mac OS X v10.1 and later.

Declared in `CGPattern.h`.

`kCGPatternTilingConstantSpacing`

Pattern cells are spaced consistently, as with `kCGPatternTilingConstantSpacingMinimalDistortion`. The pattern cell may be distorted additionally to permit a more efficient implementation.

Available in Mac OS X v10.1 and later.

Declared in `CGPattern.h`.

Declared In

`CGPattern.h`

Document Revision History

This table describes the changes to *CGPattern Reference*.

| Date | Notes |
|------------|---|
| 2006-12-22 | Updated for Mac OS X v10.5. |
| | All instances of the <code>float</code> data type were changed to the <code>CGFloat</code> data type. |
| 2005-04-29 | Updated for Mac OS X v10.4. |
| 2004-08-31 | Added introductory material. |
| 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

Index

C

CGPatternCallbacks **structure** [10](#)
CGPatternCreate **function** [6](#)
CGPatternDrawPatternCallback **callback** [8](#)
CGPatternGetTypeID **function** [7](#)
CGPatternRef **data type** [10](#)
CGPatternRelease **function** [7](#)
CGPatternReleaseInfoCallback **callback** [9](#)
CGPatternRetain **function** [8](#)

K

kCGPatternTilingConstantSpacing **constant** [11](#)
kCGPatternTilingConstantSpacingMinimalDistortion
constant [11](#)
kCGPatternTilingNoDistortion **constant** [11](#)

T

Tiling Patterns [11](#)