
CATextLayer Class Reference

Graphics & Animation



2010-05-24



Apple Inc.
© 2010 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, iPhone, Mac, Mac OS, Objective-C, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

Helvetica is a registered trademark of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

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

CATextLayer Class Reference 5

Overview	5
Tasks	5
Getting and Setting the Text	5
Text Visual Properties	6
Text Alignment and Truncation	6
Properties	6
alignmentMode	6
font	6
fontSize	7
foregroundColor	7
string	8
truncationMode	8
wrapped	8
Constants	9
Truncation modes	9
Horizontal alignment modes	9

Document Revision History 11

CATextLayer Class Reference

Inherits from	CALayer : NSObject
Conforms to	NSCoding (CALayer) CAMediaTiming (CALayer) NSObject (NSObject)
Framework	/System/Library/Frameworks/QuartzCore.framework
Availability	Available in iOS 3.2 and later.
Declared in	CATextLayer.h
Companion guides	Core Animation Programming Guide Core Animation Cookbook

Overview

The `CATextLayer` provides simple text layout and rendering of plain or attributed strings. The first line is aligned to the top of the layer.

Note: `CATextLayer` disables sub-pixel antialiasing when rendering text. Text can only be drawn using sub-pixel antialiasing when it is composited into an existing opaque background at the same time that it's rasterized. There is no way to draw subpixel-antialiased text by itself, whether into an image or a layer, separately in advance of having the background pixels to weave the text pixels into. Setting the `opacity` property of the layer to `YES` does not change the rendering mode.

Note: In Mac OS X, when a `CATextLayer` instance is positioned using the `CAConstraintLayoutManager` class the bounds of the layer is resized to fit the text content.

Tasks

Getting and Setting the Text

`string` (page 8) *property*

The text to be rendered by the receiver.

Text Visual Properties

[font](#) (page 6) *property*

The font used to render the receiver's text.

[fontSize](#) (page 7) *property*

The font size used to render the receiver's text. Animatable.

[foregroundColor](#) (page 7) *property*

The color used to render the receiver's text. Animatable.

Text Alignment and Truncation

[wrapped](#) (page 8) *property*

Determines whether the text is wrapped to fit within the receiver's bounds.

[alignmentMode](#) (page 6) *property*

Determines how individual lines of text are horizontally aligned within the receiver's bounds.

[truncationMode](#) (page 8) *property*

Determines how the text is truncated to fit within the receiver's bounds.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C Programming Language*.

alignmentMode

Determines how individual lines of text are horizontally aligned within the receiver's bounds.

```
@property(copy) NSString *alignmentMode
```

Discussion

The possible values are described in “[Horizontal alignment modes](#)” (page 9). Defaults to `kCAAlignmentNatural` (page 10).

Availability

Available in iOS 3.2 and later.

Declared In

`CATextLayer.h`

font

The font used to render the receiver's text.

```
@property CTypeRef font
```

Discussion

May be either a `CTFontRef`, a `CGFontRef`, an instance of `NSFont` (Mac OS X only), or a string naming the font. (In iOS, you cannot assign a `UIFont` object to this property.) Defaults to Helvetica.

The `font` property is only used when the `string` (page 8) property is not an `NSAttributedString`.

Note: If the `font` property specifies a font size (if it is a `CTFontRef`, a `CGFontRef`, an instance of `NSFont`) the font size is ignored.

Availability

Available in iOS 3.2 and later.

Declared In

`CATextLayer.h`

fontSize

The font size used to render the receiver's text. Animatable.

```
@property CGFloat fontSize
```

Discussion

Defaults to 36.0.

The `font` property is only used when the `string` (page 8) property is not an `NSAttributedString`.

Note: Implicit animation of this property is only enabled in applications compiled for Mac OS X v10.6 and later.

Availability

Available in iOS 3.2 and later.

Declared In

`CATextLayer.h`

foregroundColor

The color used to render the receiver's text. Animatable.

```
@property CGColorRef foregroundColor
```

Discussion

Defaults to opaque white.

The `foregroundColor` property is only used when the `string` (page 8) property is not an `NSAttributedString`.

Note: Implicit animation of this property is only enabled in applications compiled for Mac OS X v10.6 and later.

Availability

Available in iOS 3.2 and later.

Declared In

CATextLayer.h

string

The text to be rendered by the receiver.

```
@property(copy) id string
```

Discussion

The text must be an instance of `NSString` or `NSAttributedString`. Defaults to `nil`.

Availability

Available in iOS 3.2 and later.

Declared In

CATextLayer.h

truncationMode

Determines how the text is truncated to fit within the receiver's bounds.

```
@property(copy) NSString *truncationMode
```

Discussion

The possible values are described in [“Truncation modes”](#) (page 9). Defaults to `kCATruncationNone` (page 9).

Availability

Available in iOS 3.2 and later.

Declared In

CATextLayer.h

wrapped

Determines whether the text is wrapped to fit within the receiver's bounds.

```
@property(getter=isWrapped) BOOL wrapped
```

Discussion

Defaults to `NO`.

Availability

Available in iOS 3.2 and later.

Declared In

CATextLayer.h

Constants

Truncation modes

These constants are used by the [truncationMode](#) (page 8) property.

```
NSString * const kCATruncationNone;
NSString * const kCATruncationStart;
NSString * const kCATruncationEnd;
NSString * const kCATruncationMiddle;
```

Constants**kCATruncationNone**

If the [wrapped](#) (page 8) property is YES, the text is wrapped to the receiver's bounds, otherwise the text is clipped to the receiver's bounds.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCATruncationStart

Each line is displayed so that the end fits in the container and the missing text is indicated by some kind of ellipsis glyph.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCATruncationEnd

Each line is displayed so that the beginning fits in the container and the missing text is indicated by some kind of ellipsis glyph.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCATruncationMiddle

Each line is displayed so that the beginning and end fit in the container and the missing text is indicated by some kind of ellipsis glyph in the middle.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

Declared In

CATextLayer.h

Horizontal alignment modes

These constants are used by the [alignmentMode](#) (page 6) property.

```
NSString * const kCAAlignmentNatural;  
NSString * const kCAAlignmentLeft;  
NSString * const kCAAlignmentRight;  
NSString * const kCAAlignmentCenter;  
NSString * const kCAAlignmentJustified;
```

Constants

kCAAlignmentNatural

Use the natural alignment of the text's script.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCAAlignmentLeft

Text is visually left aligned.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCAAlignmentRight

Text is visually right aligned.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCAAlignmentCenter

Text is visually center aligned.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

kCAAlignmentJustified

Text is justified.

Available in iOS 3.2 and later.

Declared in CATextLayer.h.

Declared In

CATextLayer.h

Document Revision History

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

Date	Notes
2010-05-24	Corrected typo.
2010-03-02	Added this class to the Quartz Core framework for iOS 3.2.
2009-03-07	Updated for Mac OS X v10.6.
2007-12-11	Added description of the sub-pixel antialiasing limitations of CATextLayer rendering.
2007-10-31	Added note about the font property font size being ignored.
2007-07-24	New document that describes the class that simplifies rendering text in a layer.

REVISION HISTORY

Document Revision History