
CARenderer Class Reference

[Graphics & Imaging](#) > Quartz



2008-04-30



Apple Inc.
© 2008 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, Objective-C, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, 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

CARenderer Class Reference 5

- Overview 5
- Tasks 5
 - Rendered Layer 5
 - Renderer Geometry 5
 - Create a New Renderer 5
 - Render a Frame 6
- Properties 6
 - bounds 6
 - layer 6
- Class Methods 7
 - rendererWithCGLContext:options: 7
- Instance Methods 7
 - addUpdateRect: 7
 - beginFrameAtTime:timeStamp: 7
 - endFrame 8
 - nextFrameTime 8
 - render 8
 - updateBounds 9

Document Revision History 11

Index 13

CARenderer Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/QuartzCore.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	CARenderer.h
Companion guides	Core Animation Programming Guide Core Animation Cookbook

Overview

`CARenderer` allows an application to render a layer tree into a CGL context. For real-time output you should use an instance of `NSView` to host the layer-tree.

Tasks

Rendered Layer

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

The root layer of the layer-tree the receiver should render.

Renderer Geometry

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

The bounds of the receiver.

Create a New Renderer

+ [rendererWithCGLContext:options:](#) (page 7)

Creates and returns a `CARenderer` instance with the render target specified by the Core OpenGL context.

Render a Frame

- [beginFrameAtTime:timeStamp:](#) (page 7)
Begin rendering a frame at the specified time.
- [updateBounds](#) (page 9)
Returns the bounds of the update region that contains all pixels that will be rendered by the current frame.
- [addUpdateRect:](#) (page 7)
Adds the rectangle to the update region of the current frame.
- [render](#) (page 8)
Render the update region of the current frame to the target context.
- [nextFrameTime](#) (page 8)
Returns the time at which the next update should happen.
- [endFrame](#) (page 8)
Release any data associated with the current frame.

Properties

For more about Objective-C properties, see “Properties” in *The Objective-C 2.0 Programming Language*.

bounds

The bounds of the receiver.

```
@property CGRect bounds
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

CARenderer.h

layer

The root layer of the layer-tree the receiver should render.

```
@property(retain) CALayer *layer
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

CARenderer.h

Class Methods

rendererWithCGLContext:options:

Creates and returns a `CAREnderer` instance with the render target specified by the Core OpenGL context.

```
+ (CAREnderer *)rendererWithCGLContext:(void *)ctx
  options:(NSDictionary *)dict
```

Parameters

ctx

A Core OpenGL render context that is used as the render target.

dict

A dictionary of optional parameters.

Return Value

A new instance of `CAREnderer` that will use *ctx* as the render target.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CAREnderer.h`

Instance Methods

addUpdateRect:

Adds the rectangle to the update region of the current frame.

```
- (void)addUpdateRect:(CGRect)aRect
```

Parameters

aRect

A rectangle defining the region to be added to the update region.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CAREnderer.h`

beginFrameAtTime:timeStamp:

Begin rendering a frame at the specified time.

```
- (void)beginFrameAtTime:(CFTimeInterval)timeInterval
  timeStamp:(CVTimeStamp *)timeStamp
```

Parameters

timeInterval

The layer time.

timeStamp

The display timestamp associated with *timeInterval*. Can be null.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CARenderer.h`

endFrame

Release any data associated with the current frame.

- (void)endFrame

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CARenderer.h`

nextFrameTime

Returns the time at which the next update should happen.

- (CFTimeInterval)nextFrameTime

Return Value

The time at which the next update should happen.

Discussion

If infinite, no update needs to be scheduled yet. If *nextFrameTime* is the current frame time, a continuous animation is running and an update should be scheduled after an appropriate delay.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CARenderer.h`

render

Render the update region of the current frame to the target context.

- (void)render

Availability

Available in Mac OS X v10.5 and later.

Declared In

CARenderer.h

updateBounds

Returns the bounds of the update region that contains all pixels that will be rendered by the current frame.

- (CGRect)updateBounds

Return Value

The bounds of the update region..

Discussion

Initially `updateBounds` will include all differences between the current frame and the previously rendered frame.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CARenderer.h

Document Revision History

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

Date	Notes
2008-04-30	Corrected description in the Overview.
2007-07-24	New document that describes the class that allows Core Animation rendering to a CGContext.

REVISION HISTORY

Document Revision History

Index

A

`addUpdateRect:` [instance method 7](#)

B

`beginFrameAtTime:timeStamp:` [instance method 7](#)
`bounds` [instance property 6](#)

E

`endFrame` [instance method 8](#)

L

`layer` [instance property 6](#)

N

`nextFrameTime` [instance method 8](#)

R

`render` [instance method 8](#)
`rendererWithCGLContext:options:` [class method 7](#)

U

`updateBounds` [instance method 9](#)