

---

# CAAnimation Class Reference

[Graphics & Imaging](#) > Quartz



2007-10-31



Apple Inc.  
© 2007 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, Objective-C, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple 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

---

## **CAAnimation Class Reference 5**

---

Overview	5
Tasks	5
Archiving Properties	5
Providing Default Values for Properties	5
Creating an Animation	6
Animation Attributes	6
Getting and Setting the Delegate	6
Animation Progress	6
Properties	6
delegate	6
removedOnCompletion	7
timingFunction	7
Class Methods	7
animation	7
defaultValueForKey:	8
Instance Methods	8
isRemovedOnCompletion	8
shouldArchiveValueForKey:	9
Delegate Methods	9
animationDidStart:	9
animationDidStop:finished:	9

---

## **Document Revision History 11**

---

## **Index 13**

---



# CAAnimation Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSCoding NSCopying CAAction CAMediaTiming NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/QuartzCore.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Declared in</b>	CAAnimation.h
<b>Companion guides</b>	Core Animation Programming Guide Core Animation Cookbook

## Overview

`CAAnimation` is an abstract animation class. It provides the basic support for the `CAMediaTiming` and `CAAction` protocols.

## Tasks

### Archiving Properties

- [shouldArchiveValueForKey:](#) (page 9)  
Specifies whether the value of the property for a given key is archived.

### Providing Default Values for Properties

- + [defaultValueForKey:](#) (page 8)  
Specifies the default value of the property with the specified key.

## Creating an Animation

- + `animation` (page 7)  
Creates and returns a new `CAAnimation` instance.

## Animation Attributes

- `removedOnCompletion` (page 7) *property*  
Determines if the animation is removed from the target layer's animations upon completion.
- `isRemovedOnCompletion` (page 8)  
A synthesized accessor for the `removedOnCompletion` (page 7) property.
- `timingFunction` (page 7) *property*  
An optional timing function defining the pacing of the animation.

## Getting and Setting the Delegate

- `delegate` (page 6) *property*  
Specifies the receiver's delegate object.

## Animation Progress

- `animationDidStart:` (page 9) *delegate method*  
Called when the animation begins its active duration.
- `animationDidStop:finished:` (page 9) *delegate method*  
Called when the animation completes its active duration or is removed from the object it is attached to.

## Properties

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

### **delegate**

Specifies the receiver's delegate object.

```
@property(retain) id delegate
```

#### **Discussion**

Defaults to `nil`.

**Important:** The `delegate` object is retained by the receiver. This is a rare exception to the memory management rules described in *Memory Management Programming Guide for Cocoa*.

An instance of `CAAnimation` should not be set as a delegate of itself. Doing so (outside of a garbage-collected environment) will cause retain cycles.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

**removedOnCompletion**

Determines if the animation is removed from the target layer's animations upon completion.

@property BOOL removedOnCompletion

**Discussion**

When YES, the animation is removed from the target layer's animations once its active duration has passed. Defaults to YES.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

**timingFunction**

An optional timing function defining the pacing of the animation.

@property(retain) CAMediaTimingFunction \*timingFunction

**Discussion**

Defaults to `nil`, indicating linear pacing.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

## Class Methods

**animation**

Creates and returns a new `CAAnimation` instance.

```
+ (id)animation
```

**Return Value**

An `CAAnimation` object whose input values are initialized.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

**defaultValueForKey:**

Specifies the default value of the property with the specified key.

```
+ (id)defaultValueForKey:(NSString *)key
```

**Parameters**

*key*

The name of one of the receiver’s properties.

**Return Value**

The default value for the named property. Returns `nil` if no default value has been set.

**Discussion**

If this method returns `nil` a suitable “zero” default value for the property is provided, based on the declared type of the *key*. For example, if *key* is a `CGSize` object, a size of (0,0,0) is returned. For a `CGRect` an empty rectangle is returned. For `CGAffineTransform` and `CATransform3D`, the appropriate identity matrix is returned.

**Special Considerations**

If *key* is not a known for property of the class, the result of the method is undefined.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

## Instance Methods

**isRemovedOnCompletion**

A synthesized accessor for the [removedOnCompletion](#) (page 7) property.

```
- (BOOL)isRemovedOnCompletion
```

**See Also**

[@property removedOnCompletion](#) (page 7)



**shouldArchiveValueForKey:**

Specifies whether the value of the property for a given key is archived.

```
- (BOOL)shouldArchiveValueForKey:(NSString *)key
```

**Parameters**

*key*

The name of one of the receiver's properties.

**Return Value**

YES if the specified property should be archived, otherwise NO.

**Discussion**

Called by the object's implementation of `encodeWithCoder:`. The object must implement keyed archiving.

The default implementation returns YES.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

## Delegate Methods

**animationDidStart:**

Called when the animation begins its active duration.

```
- (void)animationDidStart:(CAAnimation *)theAnimation
```

**Parameters**

*theAnimation*

The `CAAnimation` instance that started animating.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`CAAnimation.h`

**animationDidStop:finished:**

Called when the animation completes its active duration or is removed from the object it is attached to.

```
- (void)animationDidStop:(CAAnimation *)theAnimation
    finished:(BOOL)flag
```

**Parameters**

*theAnimation*

The `CAAnimation` instance that stopped animating.

*flag*

If YES, the animation reached the end of its active duration without being removed.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

CAAnimation.h

# Document Revision History

---

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

Date	Notes
2007-10-31	Added a warning to the <code>setDelegate:</code> method about retain cycles.
2007-07-24	New document that describes the abstract class that manages the timing and pacing of an animation.

**REVISION HISTORY**

Document Revision History

# Index

---

## A

---

animation **class method** [7](#)  
animationDidStart: <NSObject> delegate **method** [9](#)  
animationDidStop:finished: <NSObject> delegate **method** [9](#)

## D

---

defaultValueForKey: **class method** [8](#)  
delegate **instance property** [6](#)

## I

---

isRemovedOnCompletion **instance method** [8](#)

## R

---

removedOnCompletion **instance property** [7](#)

## S

---

shouldArchiveValueForKey: **instance method** [9](#)

## T

---

timingFunction **instance property** [7](#)