

---

# CAPropertyAnimation Class Reference

Graphics & Animation: Animation



2009-03-09



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

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

## **CAPropertyAnimation Class Reference 5**

---

- Overview 5
- Tasks 5
  - Animated Key Path 5
  - Property Value Calculation Behavior 5
  - Creating an Animation 6
- Properties 6
  - additive 6
  - cumulative 6
  - keyPath 7
  - valueFunction 7
- Class Methods 7
  - animationWithKeyPath: 7

## **Document Revision History 9**

---



# CAPROPERTYANIMATION Class Reference

---

<b>Inherits from</b>	CAAnimation : NSObject
<b>Conforms to</b>	NSCoding (CAAnimation) NSCopying (CAAnimation) CAAction (CAAnimation) CAMediaTiming (CAAnimation) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/QuartzCore.framework
<b>Availability</b>	Available in iOS 2.0 and later.
<b>Declared in</b>	CAAnimation.h
<b>Companion guides</b>	Core Animation Programming Guide Core Animation Cookbook

## Overview

`CAPROPERTYANIMATION` is an abstract subclass of `CAAnimation` for creating animations that manipulate the value of layer properties. The property is specified using a key path that is relative to the layer using the animation.

## Tasks

### Animated Key Path

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

Specifies the key path the receiver animates.

### Property Value Calculation Behavior

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

Determines if the value of the property is the value at the end of the previous repeat cycle, plus the value of the current repeat cycle.

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

Determines if the value specified by the animation is added to the current render tree value to produce the new render tree value.

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

An optional value function that is applied to interpolated values.

## Creating an Animation

+ [animationWithKeyPath:](#) (page 7)

Creates and returns an `CAPropertyAnimation` instance for the specified key path.

## Properties

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

### additive

Determines if the value specified by the animation is added to the current render tree value to produce the new render tree value.

```
@property(getter=isAdditive) BOOL additive
```

#### Discussion

If YES, the value specified by the animation will be added to the current render tree value of the property to produce the new render tree value. The addition function is type-dependent, e.g. for affine transforms the two matrices are concatenated. The default is NO.

#### Availability

Available in iOS 2.0 and later.

#### Declared In

`CAAnimation.h`

### cumulative

Determines if the value of the property is the value at the end of the previous repeat cycle, plus the value of the current repeat cycle.

```
@property(getter=isCumulative) BOOL cumulative
```

#### Discussion

If YES, then the value of the property is the value at the end of the previous repeat cycle, plus the value of the current repeat cycle. If NO, the value of the property is simply the value calculated for the current repeat cycle. The default is NO.

#### Availability

Available in iOS 2.0 and later.

**Declared In**

CAAnimation.h

**keyPath**

Specifies the key path the receiver animates.

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

**Discussion**

The key path is relative to the layer the receiver is attached to.

**Availability**

Available in iOS 2.0 and later.

**Declared In**

CAAnimation.h

**valueFunction**

An optional value function that is applied to interpolated values.

```
@property(retain) CAValueFunction *valueFunction
```

**Discussion**

If the `valueFunction` property is not `nil`, the function is applied to the values interpolated by the animation as they are applied to the presentation layer. Defaults to `nil`.

**Availability**

Available in iOS 3.0 and later.

**Declared In**

CAAnimation.h

## Class Methods

**animationWithKeyPath:**

Creates and returns an `CAPropertyAnimation` instance for the specified key path.

```
+ (id)animationWithKeyPath:(NSString *)keyPath
```

**Parameters**

*keyPath*

The key path of the property to be animated.

**Return Value**

A new instance of `CAPropertyAnimation` with the key path set to *keyPath*.

**Availability**

Available in iOS 2.0 and later.

**Declared In**

CAAnimation.h



# Document Revision History

---

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

Date	Notes
2009-03-09	Updated for iOS 3.0. Added new valueFunction property.
2007-07-24	New document that describes the abstract class for interpolating the value of a layer property during animation.

## REVISION HISTORY

### Document Revision History