

---

# CATransition Class Reference

Graphics & Animation: Animation



2008-05-27



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, iPhone, Mac, Mac OS, 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

---

## **CATransition Class Reference 5**

---

Overview	5
Tasks	5
Transition Start and End Point	5
Transition Properties	5
Custom Transition Filter	6
Properties	6
endProgress	6
filter	6
startProgress	7
subtype	7
type	7
Constants	8
Common Transition Types	8
Common Transition Subtypes	8

---

## **Document Revision History 11**

---



# CATransition 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

The `CATransition` class implements transition animations for a layer. You can specify the transition effect from a set of predefined transitions or (on Mac OS X) by providing a custom `CIFilter` instance.

## Tasks

### Transition Start and End Point

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

Indicates the start point of the receiver as a fraction of the entire transition.

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

Indicates the end point of the receiver as a fraction of the entire transition.

### Transition Properties

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

Specifies the predefined transition type.

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

Specifies an optional subtype that indicates the direction for the predefined motion-based transitions.

## Custom Transition Filter

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

An optional Core Image filter object that provides the transition.

## Properties

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

### endProgress

Indicates the end point of the receiver as a fraction of the entire transition.

```
@property float endProgress
```

#### Discussion

The value must be greater than or equal to [startProgress](#) (page 7), and not greater than 1.0. If `endProgress` is less than [startProgress](#) (page 7) the behavior is undefined. The default value is 1.0.

#### Availability

Available in iOS 2.0 and later.

#### Declared In

`CAAnimation.h`

### filter

An optional Core Image filter object that provides the transition.

```
@property(retain) CIFilter *filter
```

#### Discussion

If specified, the filter must support both `kCIInputImageKey` and `kCIInputTargetImageKey` input keys, and the `kCIOutputImageKey` output key. The filter may optionally support the `kCIInputExtentKey` input key, which is set to a rectangle describing the region in which the transition should run. If `filter` does not support the required input and output keys the behavior is undefined.

Defaults to `nil`. When a transition filter is specified the [type](#) (page 7) and [subtype](#) (page 7) properties are ignored.

#### Special Considerations

While the `CATransition` class exposes this property, Core Image is not available in iOS. Currently the filters available for this property are undefined.

#### Availability

Available in iOS 2.0 and later.

#### Declared In

`CAAnimation.h`

## startProgress

Indicates the start point of the receiver as a fraction of the entire transition.

```
@property float startProgress
```

### Discussion

Legal values are numbers between 0.0 and 1.0. For example, to start the transition half way through its progress set `startProgress` to 0.5. The default value is 0.

### Availability

Available in iOS 2.0 and later.

### Declared In

`CAAnimation.h`

## subtype

Specifies an optional subtype that indicates the direction for the predefined motion-based transitions.

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

### Discussion

The possible values are shown in “[Common Transition Subtypes](#)” (page 8). The default is `nil`.

This property is ignored if a custom transition is specified in the `filter` (page 6) property.

### Availability

Available in iOS 2.0 and later.

### Declared In

`CAAnimation.h`

## type

Specifies the predefined transition type.

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

### Discussion

The possible values are shown in “[Common Transition Types](#)” (page 8). This property is ignored if a custom transition is specified in the `filter` (page 6) property. The default is `kCATransitionFade` (page 8).

### Availability

Available in iOS 2.0 and later.

### Declared In

`CAAnimation.h`

## Constants

### Common Transition Types

These constants specify the transition types that can be used with the [type](#) (page 7) property.

```
NSString * const kCATransitionFade;
NSString * const kCATransitionMoveIn;
NSString * const kCATransitionPush;
NSString * const kCATransitionReveal;
```

#### Constants

`kCATransitionFade`

The layer's content fades as it becomes visible or hidden.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCATransitionMoveIn`

The layer's content slides into place over any existing content. The [“Common Transition Subtypes”](#) (page 8) are used with this transition.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCATransitionPush`

The layer's content pushes any existing content as it slides into place. The [“Common Transition Subtypes”](#) (page 8) are used with this transition.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCATransitionReveal`

The layer's content is revealed gradually in the direction specified by the transition subtype. The [“Common Transition Subtypes”](#) (page 8) are used with this transition.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

#### Declared In

`CATransition.h`

### Common Transition Subtypes

These constants specify the direction of motion-based transitions. They are used with the [subtype](#) (page 7) property.



```
NSString * const kCATransitionFromRight;  
NSString * const kCATransitionFromLeft;  
NSString * const kCATransitionFromTop;  
NSString * const kCATransitionFromBottom;
```

### Constants

`kCATransitionFromRight`

The transition begins at the right side of the layer.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCATransitionFromLeft`

The transition begins at the left side of the layer.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCATransitionFromTop`

The transition begins at the top of the layer.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

`kCATransitionFromBottom`

The transition begins at the bottom of the layer.

Available in iOS 2.0 and later.

Declared in `CAAnimation.h`.

### Declared In

`CATransition.h`



# Document Revision History

---

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

Date	Notes
2008-05-27	Updated for iOS 3.0.
2007-10-31	Added the default value for the transition type.
2007-07-24	New document that describes the class provides transition animations to layers.

## REVISION HISTORY

### Document Revision History