
NSViewAnimation Class Reference

[Cocoa](#) > [Graphics & Imaging](#)



2006-05-23



Apple Inc.
© 2006 Apple Computer, 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, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

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

NSViewAnimation Class Reference 5

Overview 5

Tasks 6

 Initializing an NSViewAnimation Object 6

 Getting and Setting View-animation Dictionaries 6

Instance Methods 6

 initWithViewAnimations: 6

 setViewAnimations: 6

 viewAnimations 7

Constants 7

 View Animation Dictionary Keys 7

 Values for NSViewAnimationEffectKey 8

Document Revision History 11

Index 13

NSViewAnimation Class Reference

Inherits from	NSAnimation : NSObject
Conforms to	NSCoding (NSAnimation) NSCopying (NSAnimation) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.4 and later.
Companion guide	Cocoa Drawing Guide
Declared in	NSAnimation.h
Related sample code	iSpend QTCoreVideo301 Reducer

Overview

The `NSViewAnimation` class, a public subclass of `NSAnimation`, offers a convenient way to animate multiple views and windows. The animation effects you can achieve are limited to changes in frame location and size, and to fade-in and fade-out effects.

An `NSViewAnimation` object takes an array of dictionaries from which it determines the objects to animate and the effects to apply to them. Each dictionary must have a target object and, optionally, properties that specify beginning and ending frame and whether to fade in or fade out. (See [“View Animation Dictionary Keys”](#) (page 7) for further information.) Animations with `NSViewAnimation` are, by default, in non-blocking mode over a duration of 0.5 seconds using the ease in-out animation curve. But you can configure the animation to have any duration, curve, frame rate, and blocking mode. You may also set progress marks, assign a delegate, and implement delegation methods in order to animate view and windows concurrent with the ones specified as targets in the view-animation dictionary.

Invoking the `NSAnimation` `stopAnimation` method on a running `NSViewAnimation` object moves the animation to the end frame.

Tasks

Initializing an NSViewAnimation Object

- [initWithViewAnimations:](#) (page 6)
Returns an `NSViewAnimation` object initialized with the supplied information.

Getting and Setting View-animation Dictionaries

- [setViewAnimations:](#) (page 6)
Sets the dictionaries defining the objects to animate.
- [viewAnimations](#) (page 7)
Returns the array of dictionaries defining the objects to animate.

Instance Methods

initWithViewAnimations:

Returns an `NSViewAnimation` object initialized with the supplied information.

```
- (id)initWithViewAnimations:(NSArray *)viewAnimations
```

Parameters

viewAnimations

An array of `NSDictionary` objects. Each dictionary specifies a view or window to animate and the effect to apply. *viewAnimations* can be `nil`, but you must later set the required array of dictionaries with [setViewAnimations:](#) (page 6) if you want to use the capabilities of the `NSViewAnimation` class. See “[View Animation Dictionary Keys](#)” (page 7) for a description of valid keys and values for dictionaries in *viewAnimations*.

Return Value

The created `NSViewAnimation` object or `nil` if there was a problem initializing the object.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

iSpend
Reducer

Declared In

`NSAnimation.h`

setViewAnimations:

Sets the dictionaries defining the objects to animate.

- (void)setViewAnimations:(NSArray *)viewAnimations

Parameters

viewAnimations

An array of `NSDictionary` objects. Each dictionary specifies a view or window to animate and the effect to apply. Pass in `nil` to remove the current list of dictionaries. See “[View Animation Dictionary Keys](#)” (page 7) for a description of valid keys and values for dictionaries in *viewAnimations*.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [viewAnimations](#) (page 7)

Declared In

`NSAnimation.h`

viewAnimations

Returns the array of dictionaries defining the objects to animate.

- (NSArray *)viewAnimations

Discussion

Each dictionary in the returned array specifies a view or window to animate and the effect to apply.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setViewAnimations:](#) (page 6)

Declared In

`NSAnimation.h`

Constants

View Animation Dictionary Keys

The following string constants are keys for the dictionaries in the array passed into [initWithViewAnimations:](#) (page 6) and [setViewAnimations:](#) (page 6).

```
NSString *NSViewAnimationTargetKey;
NSString *NSViewAnimationStartFrameKey;
NSString *NSViewAnimationEndFrameKey;
NSString *NSViewAnimationEffectKey;
```

Constants

`NSViewAnimationTargetKey`

The target of the animation.

The target can be either an `NSView` object or an `NSWindow` object. This property is required.

Available in Mac OS X v10.4 and later.

Declared in `NSAnimation.h`.

`NSViewAnimationStartFrameKey`

The size and location of the window or view at the start of the animation.

The size and location are specified by an `NSRect` structure encoded in an `NSValue` object. This property is optional. If it is not specified, `NSViewAnimation` uses the frame of the window or view at the start of the animation.

Available in Mac OS X v10.4 and later.

Declared in `NSAnimation.h`.

`NSViewAnimationEndFrameKey`

The size and location of the window or view at the end of the animation.

The size and location are specified by an `NSRect` structure encoded in an `NSValue` object. This property is optional. If it is not specified, `NSViewAnimation` uses the frame of the window or view at the start of the animation. If the target is a view and the end frame is empty, the view is hidden at the end.

Available in Mac OS X v10.4 and later.

Declared in `NSAnimation.h`.

`NSViewAnimationEffectKey`

An effect to apply to the animation.

Takes a string constant specifying fade-in or fade-out effects for the target:

`NSViewAnimationFadeInEffect` and `NSViewAnimationFadeOutEffect`. If the target is a view and the effect is to fade out, the view is hidden at the end. If the effect is to fade in an initially hidden view and the end frame is non-empty, the view is unhidden at the end. If the target is a window, the window is ordered in or out as appropriate to the effect. This property is optional.

Available in Mac OS X v10.4 and later.

Declared in `NSAnimation.h`.

Declared In

`NSAnimation.h`

Values for NSViewAnimationEffectKey

The following constants specify the animation effect to apply and are used as values for the animation effect property of the animation view. See the description of `NSViewAnimationEffectKey` for usage details.


```
NSString *NSViewAnimationFadeInEffect;  
NSString *NSViewAnimationFadeOutEffect;
```

Constants

`NSViewAnimationFadeInEffect`

Specifies a fade-in type of effect.

Available in Mac OS X v10.4 and later.

Declared in `NSAnimation.h`.

`NSViewAnimationFadeOutEffect`

Specifies a fade-out type of effect.

Available in Mac OS X v10.4 and later.

Declared in `NSAnimation.h`.

Availability

Available in Mac OS X v10.4 and later

Declared In

`NSAnimation.h`

Document Revision History

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

Date	Notes
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

I

`initWithViewAnimations:` [instance method 6](#)

N

`NSViewAnimationEffectKey` [constant 8](#)

`NSViewAnimationEndFrameKey` [constant 8](#)

`NSViewAnimationFadeInEffect` [constant 9](#)

`NSViewAnimationFadeOutEffect` [constant 9](#)

`NSViewAnimationStartFrameKey` [constant 8](#)

`NSViewAnimationTargetKey` [constant 8](#)

S

`setViewAnimations:` [instance method 6](#)

V

[Values for NSViewAnimationEffectKey 8](#)

[View Animation Dictionary Keys 7](#)

`viewAnimations` [instance method 7](#)