
NSProgressIndicator Class Reference

[Cocoa](#) > [User Experience](#)



2008-03-11



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, Aqua, 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

NSProgressIndicator Class Reference 5

Overview	5
Tasks	5
Animating the Progress Indicator	5
Advancing the Progress Bar	6
Setting the Appearance	6
Instance Methods	7
controlSize	7
controlTint	7
doubleValue	8
incrementBy:	8
isBezeled	9
isDisplayedWhenStopped	9
isIndeterminate	9
maxValue	10
minValue	10
setBezeled:	11
setControlSize:	11
setControlTint:	11
setDisplayedWhenStopped:	12
setDoubleValue:	12
setIndeterminate:	13
setMaxValue:	13
setMinValue:	13
setStyle:	14
setUsesThreadedAnimation:	14
sizeToFit	15
startAnimation:	15
stopAnimation:	15
style	16
usesThreadedAnimation	16
Constants	17
NSProgressIndicatorThickness	17
NSProgressIndicatorStyle	17

Appendix A Deprecated NSProgressIndicator Methods 19

Deprecated in Mac OS X v10.5 and later	19
animate:	19
animationDelay	19
setAnimationDelay:	20

Document Revision History 21

Index 23

NSProgressIndicator Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Progress Indicators
Declared in	NSProgressIndicator.h
Related sample code	BackgroundExporter CIVideoDemoGL MyPhoto StickiesExample URL CacheInfo

Overview

The `NSProgressIndicator` class lets an application display a progress indicator to show that a lengthy task is under way. Some progress indicators are indeterminate and do nothing more than spin to show that the application is busy. Others are determinate and show the percentage of the task that has been completed.

Tasks

Animating the Progress Indicator

- [setUsesThreadedAnimation:](#) (page 14)
Sets a hint as to whether the receiver should implement animation of the progress indicator in a separate thread.
- [startAnimation:](#) (page 15)
Starts the animation of an indeterminate progress indicator.
- [stopAnimation:](#) (page 15)
Stops the animation of an indeterminate progress indicator.

- [usesThreadedAnimation](#) (page 16)
Returns whether the receiver implements the animation of the progress indicator in a separate thread.
- [animate:](#) (page 19) **Deprecated in Mac OS X v10.5 and later**
This action method advances the progress animation of an indeterminate progress animator by one step. (**Deprecated.** `NSProgressIndicator` no longer supports incrementing the animation using this method. Use the [startAnimation:](#) (page 15) and [stopAnimation:](#) (page 15) methods to perform animation of the progress indicator.)
- [animationDelay](#) (page 19) **Deprecated in Mac OS X v10.5 and later**
Returns the delay, in seconds, between animation steps for an indeterminate progress indicator. (**Deprecated.** Progress indicators no longer allow the animation delay to be set.)
- [setAnimationDelay:](#) (page 20) **Deprecated in Mac OS X v10.5 and later**
Sets the delay, in seconds, between animation steps for an indeterminate progress indicator. (**Deprecated.** Progress indicators no longer allow the animation delay to be set.)

Advancing the Progress Bar

- [incrementBy:](#) (page 8)
Advances the progress bar of a determinate progress indicator by the specified amount.
- [setDoubleValue:](#) (page 12)
Sets the value that indicates the current extent of the receiver.
- [doubleValue](#) (page 8)
Returns a value that indicates the current extent of the progress bar of a determinate progress indicator.
- [setMinValue:](#) (page 13)
Sets the minimum value for the receiver.
- [minValue](#) (page 10)
Returns the minimum value for the progress bar of a determinate progress indicator.
- [setMaxValue:](#) (page 13)
Sets the maximum value for the receiver.
- [maxValue](#) (page 10)
Returns the maximum value for the progress bar of a determinate progress indicator.

Setting the Appearance

- [setControlSize:](#) (page 11)
Sets the size of the receiver.
- [controlSize](#) (page 7)
Returns the size of the receiver.
- [setControlTint:](#) (page 11)
Sets the receiver's control tint.
- [controlTint](#) (page 7)
Returns the receiver's control tint.
- [setBezeled:](#) (page 11)
Sets whether the receiver's frame has a three-dimensional bezel.

- [isBezeled](#) (page 9)
Returns a Boolean value indicating whether the receiver's frame has a bezel.
- [setIndeterminate:](#) (page 13)
Sets whether the receiver is indeterminate.
- [isIndeterminate](#) (page 9)
Returns a Boolean value indicating whether the receiver is indeterminate.
- [setStyle:](#) (page 14)
Sets the style of the progress indicator (bar or spinning).
- [style](#) (page 16)
Returns the style of the progress indicator (bar or spinning).
- [sizeToFit](#) (page 15)
This action method resizes the receiver to an appropriate size depending on what [style](#) (page 16) returns.
- [setDisplayWhenStopped:](#) (page 12)
Sets whether the receiver hides itself when it isn't animating.
- [isDisplayedWhenStopped](#) (page 9)
Returns a Boolean value indicating whether the receiver shows itself even when it's not animating.

Instance Methods

controlSize

Returns the size of the receiver.

- (NSControlSize)controlSize

Return Value

A constant indicating the size of the control. Valid return values are described in NSCell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setControlSize:](#) (page 11)

Declared In

NSProgressIndicator.h

controlTint

Returns the receiver's control tint.

- (NSControlTint)controlTint

Return Value

A constant indicating the current control tint. Valid return values are described in NSCell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setControlTint:](#) (page 11)

Declared In

NSProgressIndicator.h

doubleValue

Returns a value that indicates the current extent of the progress bar of a determinate progress indicator.

- (double)doubleValue

Return Value

The value representing the current extent of a determinate progress bar. For example, a determinate progress indicator goes from 0.0 to 100.0 by default. If the progress bar has advanced halfway across the view, the value returned by `doubleValue` would be 50.0. An indeterminate progress indicator does not use this value.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [incrementBy:](#) (page 8)

- [setDoubleValue:](#) (page 12)

Declared In

NSProgressIndicator.h

incrementBy:

Advances the progress bar of a determinate progress indicator by the specified amount.

- (void)incrementBy:(double)*delta*

Parameters

delta

The amount by which to increment the progress bar. For example, if you want to advance a progress bar from 0.0 to 100.0 in 20 steps, you would invoke [incrementBy:](#) (page 8) 20 times with a delta value of 5.0.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [doubleValue](#) (page 8)

Declared In

NSProgressIndicator.h

isBezeled

Returns a Boolean value indicating whether the receiver's frame has a bezel.

- (BOOL)isBezeled

Return Value

YES if the receiver's frame has a three-dimensional bezel; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setBezeled:](#) (page 11)

Declared In

NSProgressIndicator.h

isDisplayedWhenStopped

Returns a Boolean value indicating whether the receiver shows itself even when it's not animating.

- (BOOL)isDisplayedWhenStopped

Return Value

YES if the progress indicator shows itself even when it's not animating. By default, this returns YES if [style](#) (page 16) is `NSProgressIndicatorBarStyle` and NO if [style](#) (page 16) is `NSProgressIndicatorSpinningStyle`.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [setDisplayedWhenStopped:](#) (page 12)

Declared In

NSProgressIndicator.h

isIndeterminate

Returns a Boolean value indicating whether the receiver is indeterminate.

- (BOOL)isIndeterminate

Return Value

YES if the progress bar is indeterminate; otherwise NO. This applies only if [style](#) (page 16) returns `NSProgressIndicatorBarStyle`.

Discussion

If [style](#) (page 16) returns `NSProgressIndicatorSpinningStyle`, the indicator is always indeterminate, regardless of what this method returns.

A determinate indicator displays how much of the task has been completed. An indeterminate indicator shows simply that the application is busy.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setIndeterminate:](#) (page 13)

Declared In

NSProgressIndicator.h

maxValue

Returns the maximum value for the progress bar of a determinate progress indicator.

- (double)maxValue

Return Value

The maximum value of the progress indicator. By default, a determinate progress indicator goes from 0.0 to 100.0, so the value returned would be 100.0. An indeterminate progress indicator does not use this value.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [minValue](#) (page 10)

- [setMaxValue:](#) (page 13)

Declared In

NSProgressIndicator.h

minValue

Returns the minimum value for the progress bar of a determinate progress indicator.

- (double)minValue

Return Value

The minimum value of the progress indicator. By default, a determinate progress indicator goes from 0.0 to 100.0, so the value returned would be 0.0. An indeterminate progress indicator does not use this value.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [maxValue](#) (page 10)

- [setMinValue:](#) (page 13)

Declared In

NSProgressIndicator.h

setBezeled:

Sets whether the receiver's frame has a three-dimensional bezel.

```
- (void)setBezeled:(BOOL)flag
```

Parameters

flag

YES if the progress indicator is bezeled; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isBezeled](#) (page 9)

Declared In

NSProgressIndicator.h

setControlSize:

Sets the size of the receiver.

```
- (void)setControlSize:(NSControlSize)size
```

Parameters

size

A constant indicating the size of the control. Valid values for *size* are described in NSCell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [controlSize](#) (page 7)

Declared In

NSProgressIndicator.h

setControlTint:

Sets the receiver's control tint.

```
- (void)setControlTint:(NSControlTint)controlTint
```

Parameters

controlTint

A constant indicating the desired control tint. Valid values for *controlTint* are described in NSCell.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [controlTint](#) (page 7)

Declared In

NSProgressIndicator.h

setDisplayWhenStopped:

Sets whether the receiver hides itself when it isn't animating.

- (void)setDisplayWhenStopped:(BOOL)isDisplayed

Parameters*isDisplayed*

YES to hide the progress indicator when it isn't animating; otherwise NO. By default, this is YES if [style](#) (page 16) is NSProgressIndicatorBarStyle, and NO if [style](#) (page 16) is NSProgressIndicatorSpinningStyle.

Availability

Available in Mac OS X v10.2 and later.

See Also- [isDisplayedWhenStopped](#) (page 9)**Declared In**

NSProgressIndicator.h

setDoubleValue:

Sets the value that indicates the current extent of the receiver.

- (void)setDoubleValue:(double)doubleValue

Parameters*doubleValue*

The current extent of a determinate progress indicator.

Discussion

An indeterminate progress indicator does not use this value.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [doubleValue](#) (page 8)
- [incrementBy:](#) (page 8)
- [setMaxValue:](#) (page 13)
- [setMinValue:](#) (page 13)

Related Sample Code

CIVideoDemoGL

Declared In

NSProgressIndicator.h

setIndeterminate:

Sets whether the receiver is indeterminate.

- (void)setIndeterminate:(BOOL)flag

Parameters

flag

YES if the progress indicator should be indeterminate; otherwise NO.

Discussion

This method only has an effect if [style](#) (page 16) returns `NSProgressIndicatorBarStyle`. If [style](#) (page 16) returns `NSProgressIndicatorSpinningStyle`, the indicator is always indeterminate, regardless of what you pass to this method.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isIndeterminate](#) (page 9)

Declared In

`NSProgressIndicator.h`

setMaxValue:

Sets the maximum value for the receiver.

- (void)setMaxVaLue:(double)newMaximum

Parameters

newMaximum

The maximum value of the progress indicator. An indeterminate progress indicator does not use this value.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [maxVaLue](#) (page 10)

Declared In

`NSProgressIndicator.h`

setMinValue:

Sets the minimum value for the receiver.

- (void)setMinVaLue:(double)newMinimum

Parameters

newMinimum

The minimum value of the progress indicator. An indeterminate progress indicator does not use this value.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [minValue](#) (page 10)

Declared In

NSProgressIndicator.h

setStyle:

Sets the style of the progress indicator (bar or spinning).

```
- (void)setStyle:(NSProgressIndicatorStyle)style
```

Parameters

style

A constant indicating the style of the progress indicator. Possible values for *style* are described in [NSProgressIndicatorStyle](#) (page 17).

Availability

Available in Mac OS X v10.2 and later.

See Also

- [style](#) (page 16)

Declared In

NSProgressIndicator.h

setUsesThreadedAnimation:

Sets a hint as to whether the receiver should implement animation of the progress indicator in a separate thread.

```
- (void)setUsesThreadedAnimation:(BOOL)flag
```

Parameters

flag

YES to indicate that animation of the progress indicator should occur in a separate thread; otherwise, NO. This value is only a hint and may be ignored.

Discussion

If the application becomes multithreaded as a result of an invocation of this method, the application's performance could become noticeably slower.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [usesThreadedAnimation](#) (page 16)

Related Sample Code

StickiesExample

Declared In

NSProgressIndicator.h

sizeToFit

This action method resizes the receiver to an appropriate size depending on what [style](#) (page 16) returns.

- (void)sizeToFit

Discussion

Use this after you use [setStyle:](#) (page 14) to re-size the receiver.

Availability

Available in Mac OS X v10.2 and later.

Declared In

NSProgressIndicator.h

startAnimation:

Starts the animation of an indeterminate progress indicator.

- (void)startAnimation:(id)sender

Parameters

sender

The object sending the message.

Discussion

Does nothing for a determinate progress indicator.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [animationDelay](#) (page 19)

- [stopAnimation:](#) (page 15)

Related Sample Code

BackgroundExporter

MenuItemView

SampleScannerApp

StickiesExample

Declared In

NSProgressIndicator.h

stopAnimation:

Stops the animation of an indeterminate progress indicator.

- (void)stopAnimation:(id)sender

Parameters

sender

The object sending the message.

Discussion

Does nothing for a determinate progress indicator.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [animationDelay](#) (page 19)
- [startAnimation:](#) (page 15)

Related Sample Code

BackgroundExporter
SampleScannerApp
StickiesExample

Declared In

NSProgressIndicator.h

style

Returns the style of the progress indicator (bar or spinning).

- (NSProgressIndicatorStyle)style

Return Value

A constant indicating the style of the progress indicator. Possible return values are described in [NSProgressIndicatorStyle](#) (page 17).

Availability

Available in Mac OS X v10.2 and later.

See Also

- [setStyle:](#) (page 14)

Declared In

NSProgressIndicator.h

usesThreadedAnimation

Returns whether the receiver implements the animation of the progress indicator in a separate thread.

- (BOOL)usesThreadedAnimation

Return Value

YES if animation of the progress indicator occurs in a separate thread; otherwise, NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setUsesThreadedAnimation:](#) (page 14)

Declared In

NSProgressIndicator.h

Constants

NSProgressIndicatorThickness

Specify the height of a progress indicator.

```
typedef enum _NSProgressIndicatorThickness {
    NSProgressIndicatorPreferredThickness      = 14,
    NSProgressIndicatorPreferredSmallThickness = 10,
    NSProgressIndicatorPreferredLargeThickness = 18,
    NSProgressIndicatorPreferredAquaThickness  = 12
} NSProgressIndicatorThickness;
```

Constants

NSProgressIndicatorPreferredThickness
14

Available in Mac OS X v10.0 and later.

Declared in NSProgressIndicator.h.

NSProgressIndicatorPreferredSmallThickness
10

Available in Mac OS X v10.0 and later.

Declared in NSProgressIndicator.h.

NSProgressIndicatorPreferredLargeThickness
18

Available in Mac OS X v10.0 and later.

Declared in NSProgressIndicator.h.

NSProgressIndicatorPreferredAquaThickness
12

Available in Mac OS X v10.0 and later.

Declared in NSProgressIndicator.h.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSProgressIndicator.h

NSProgressIndicatorStyle

Specify the progress indicator's style and are used by [style](#) (page 16) and [setStyle:](#) (page 14).

```
typedef enum _NSProgressIndicatorStyle {  
    NSProgressIndicatorBarStyle = 0,  
    NSProgressIndicatorSpinningStyle = 1  
} NSProgressIndicatorStyle;
```

Constants

NSProgressIndicatorBarStyle

A rectangular indicator that can be determinate or indeterminate.

Available in Mac OS X v10.2 and later.

Declared in NSProgressIndicator.h.

NSProgressIndicatorSpinningStyle

A small square indicator that can be indeterminate only .

Available in Mac OS X v10.2 and later.

Declared in NSProgressIndicator.h.

Availability

Available in Mac OS X v10.2 and later.

Declared In

NSProgressIndicator.h

Deprecated NSProgressIndicator Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

Deprecated in Mac OS X v10.5 and later

animate:

This action method advances the progress animation of an indeterminate progress animator by one step. (Deprecated in Mac OS X v10.5 and later. NSProgressIndicator no longer supports incrementing the animation using this method. Use the [startAnimation:](#) (page 15) and [stopAnimation:](#) (page 15) methods to perform animation of the progress indicator.)

- (void)animate:(id)sender

Parameters

sender

The object sending the message.

Availability

Deprecated in Mac OS X v10.5 and later.

See Also

- [startAnimation:](#) (page 15)
- [stopAnimation:](#) (page 15)

Related Sample Code

MenuItemView

Declared In

NSProgressIndicator.h

animationDelay

Returns the delay, in seconds, between animation steps for an indeterminate progress indicator. (Deprecated in Mac OS X v10.5 and later. Progress indicators no longer allow the animation delay to be set.)

- (NSTimeInterval)animationDelay

Return Value

The delay between animation steps. By default, the animation delay is set to 1/12 of a second (5.0/60.0). A determinate progress indicator does not use the animation delay value.

Availability

Deprecated in Mac OS X v10.5 and later.

Declared In

NSProgressIndicator.h

setAnimationDelay:

Sets the delay, in seconds, between animation steps for an indeterminate progress indicator. (Deprecated in Mac OS X v10.5 and later. Progress indicators no longer allow the animation delay to be set.)

```
- (void)setAnimationDelay:(NSTimeInterval)delay
```

Parameters*delay*

The number of seconds between animation steps. By default, the animation delay is set to 1/12 of a second (5.0/60.0). Setting the delay to a `double` value larger than 5.0/60.0 slows the animation, while setting the delay to a smaller value speeds it up. A determinate progress indicator does not use the animation delay value.

Availability

Deprecated in Mac OS X v10.5 and later.

Declared In

NSProgressIndicator.h

Document Revision History

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

Date	Notes
2008-03-11	Corrected typos.
2007-06-21	Deprecated <code>animate:</code> , <code>animationDelay</code> , and <code>setAnimationDelay</code> .
2006-06-28	Updated the description of the <code>setUsesThreadedAnimation:</code> method.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

animate: [instance method 19](#)
animationDelay [instance method 19](#)

C

controlSize [instance method 7](#)
controlTint [instance method 7](#)

D

doubleValue [instance method 8](#)

I

incrementBy: [instance method 8](#)
isBezeled [instance method 9](#)
isDisplayedWhenStopped [instance method 9](#)
isIndeterminate [instance method 9](#)

M

maxValue [instance method 10](#)
minValue [instance method 10](#)

N

NSProgressIndicatorBarStyle [constant 18](#)
NSProgressIndicatorPreferredAquaThickness
[constant 17](#)
NSProgressIndicatorPreferredLargeThickness
[constant 17](#)

NSProgressIndicatorPreferredSmallThickness
[constant 17](#)
NSProgressIndicatorPreferredThickness [constant 17](#)
NSProgressIndicatorSpinningStyle [constant 18](#)
NSProgressIndicatorStyle [data type 17](#)
NSProgressIndicatorThickness [data type 17](#)

S

setAnimationDelay: [instance method 20](#)
setBezeled: [instance method 11](#)
setControlSize: [instance method 11](#)
setControlTint: [instance method 11](#)
setDisplayedWhenStopped: [instance method 12](#)
setDoubleValue: [instance method 12](#)
setIndeterminate: [instance method 13](#)
setMaxValue: [instance method 13](#)
setMinValue: [instance method 13](#)
setStyle: [instance method 14](#)
setUsesThreadedAnimation: [instance method 14](#)
sizeToFit [instance method 15](#)
startAnimation: [instance method 15](#)
stopAnimation: [instance method 15](#)
style [instance method 16](#)

U

usesThreadedAnimation [instance method 16](#)