
NSLevelIndicator Class Reference

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



2007-03-01



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, iPhoto, iTunes, 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

NSLevelIndicator Class Reference 7

Overview	7
Tasks	8
Configuring the Range of Values	8
Managing Tick Marks	8
Instance Methods	9
criticalValue	9
maxValue	9
minValue	9
numberOfMajorTickMarks	9
numberOfTickMarks	10
rectOfTickMarkAtIndex:	10
setCriticalValue:	10
setMaxValue:	11
setMinValue:	11
setNumberOfMajorTickMarks:	11
setNumberOfTickMarks:	12
setTickMarkPosition:	12
setWarningValue:	13
tickMarkPosition	13
tickMarkValueAtIndex:	13
warningValue	14

Document Revision History 15

Index 17

Figures

NSLevelIndicator Class Reference 7

Figure 1 Major and minor tick marks in a level indicator 12

NSLevelIndicator Class Reference

Inherits from	NSControl : NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	NSLevelIndicator.h
Related sample code	QTRecorder

Overview

NSLevelIndicator is a subclass of NSControl that displays a value on a linear scale. Level indicators provide a visual representation of a level or amount of something, using discrete values. While similar to NSSlider, it provides a more customized visual feedback to the user. Level indicators do not have a “knob” indicating the current setting or allowing the user to adjust settings. The supported indicator styles include:

- A capacity style level indicator. The continuous mode for this style is often used to indicate conditions such as how much data is on hard disk. The discrete mode is similar to audio level indicators in audio playback applications. You can specify both a warning value and a critical value that provides additional visual feedback to the user.
- A ranking style level indicator. This is similar to the star ranking displays provided in iTunes and iPhoto. You can also specify your own ranking image.
- A relevancy style level indicator. This style is used to display the relevancy of a search result, for example in Mail.

NSLevelIndicator uses an NSLevelIndicatorCell to implement much of the control's functionality. NSLevelIndicator provides cover methods for most of NSLevelIndicatorCell's methods, which invoke the corresponding cell method.

Tasks

Configuring the Range of Values

- `setMinValue:` (page 11)
Sets the minimum value the receiver can represent to *minValue*.
- `minValue` (page 9)
Returns the receiver's minimum value.
- `setMaxValue:` (page 11)
Sets the maximum value the receiver can represent to *maxValue*.
- `maxValue` (page 9)
Returns the receiver's maximum value.
- `setWarningValue:` (page 13)
Sets the receiver's warning value to *warningValue*.
- `warningValue` (page 14)
Returns the receiver's warning value.
- `setCriticalValue:` (page 10)
Sets the receiver's critical value to *criticalValue*.
- `criticalValue` (page 9)
Returns the receiver's critical value.

Managing Tick Marks

- `setTickMarkPosition:` (page 12)
Sets where tick marks appear relative to the receiver.
- `tickMarkPosition` (page 13)
Returns how the receiver's tick marks are aligned with it.
- `setNumberOfTickMarks:` (page 12)
Sets the number of tick marks displayed by the receiver (which include those assigned to the minimum and maximum values) to *count*.
- `numberOfTickMarks` (page 10)
Returns the number of tick marks associated with the receiver.
- `setNumberOfMajorTickMarks:` (page 11)
Sets the number of major tick marks displayed by the receiver.
- `numberOfMajorTickMarks` (page 9)
Returns the number of major tick marks associated with the receiver.
- `tickMarkValueAtIndex:` (page 13)
Returns the receiver's value represented by the tick mark at index (the minimum-value tick mark has an index of 0).
- `rectOfTickMarkAtIndex:` (page 10)
Returns the bounding rectangle of the tick mark identified by *index* (the minimum-value tick mark is at index 0).

Instance Methods

criticalValue

Returns the receiver's critical value.

- (double)criticalValue

Availability

Available in Mac OS X v10.4 and later.

See Also

- [setCriticalValue:](#) (page 10)

Declared In

NSLevelIndicator.h

maxValue

Returns the receiver's maximum value.

- (double)maxValue

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicator.h

minValue

Returns the receiver's minimum value.

- (double)minValue

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicator.h

numberOfMajorTickMarks

Returns the number of major tick marks associated with the receiver.

- (NSInteger)numberOfMajorTickMarks

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicator.h

numberOfTickMarks

Returns the number of tick marks associated with the receiver.

- (NSInteger)numberOfTickMarks

Discussion

The tick marks assigned to the minimum and maximum values are included.

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicator.h

rectOfTickMarkAtIndex:

Returns the bounding rectangle of the tick mark identified by *index* (the minimum-value tick mark is at index 0).

- (NSRect)rectOfTickMarkAtIndex:(NSInteger) *index*

Discussion

If no tick mark is associated with *index*, the method raises a `NSRangeException`.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSLevelIndicator.h

setCriticalValue:

Sets the receiver's critical value to *criticalValue*.

- (void)setCriticalValue:(double) *criticalValue*

Availability

Available in Mac OS X v10.4 and later.

See Also

- [criticalValue](#) (page 9)

Declared In

NSLevelIndicator.h

setMaxValue:

Sets the maximum value the receiver can represent to *maxValue*.

```
- (void)setMaxVaLue:(double)maxVaLue
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [maxVaLue](#) (page 9)

Declared In

NSLevelIndicator.h

setMinValue:

Sets the minimum value the receiver can represent to *minValue*.

```
- (void)setMinVaLue:(double)minVaLue
```

Availability

Available in Mac OS X v10.4 and later.

See Also

- [minVaLue](#) (page 9)

Declared In

NSLevelIndicator.h

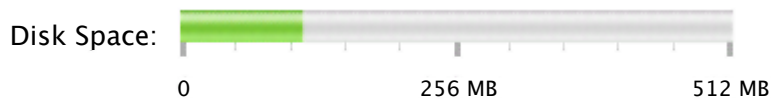
setNumberOfMajorTickMarks:

Sets the number of major tick marks displayed by the receiver.

```
- (void)setNumberOfMajorTickMarks:(NSInteger)count
```

Discussion

The *count* must be less than or equal to the number of tick marks returned by [numberOfTickMarks](#) (page 10). For example, if the number of tick marks is 11 and you specify 3 major tick marks, the resulting level indicator will display 3 major tickmarks alternating with 8 minor tick marks, as in the example shown in Figure 1.

Figure 1 Major and minor tick marks in a level indicator**Availability**

Available in Mac OS X v10.4 and later.

See Also

- [numberOfMajorTickMarks](#) (page 9)

Declared In

NSLevelIndicator.h

setNumberOfTickMarks:

Sets the number of tick marks displayed by the receiver (which include those assigned to the minimum and maximum values) to *count*.

```
- (void)setNumberOfTickMarks:(NSInteger)count
```

Discussion

By default, this value is 0, and no tick marks appear. The number of tick marks assigned to a slider, along with the slider's minimum and maximum values, determines the values associated with the tick marks.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [numberOfTickMarks](#) (page 10)

Declared In

NSLevelIndicator.h

setTickMarkPosition:

Sets where tick marks appear relative to the receiver.

```
- (void)setTickMarkPosition:(NSTickMarkPosition)position
```

Discussion

This method has no effect if no tick marks have been assigned (that is, [numberOfTickMarks](#) (page 10) returns 0).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [tickMarkPosition](#) (page 13)

Declared In

NSLevelIndicator.h

setWarningValue:Sets the receiver's warning value to *warningValue*.

- (void)setWarningValue:(double)warningValue

Availability

Available in Mac OS X v10.4 and later.

See Also- [warningValue](#) (page 14)**Declared In**

NSLevelIndicator.h

tickMarkPosition

Returns how the receiver's tick marks are aligned with it.

- (NSTickMarkPosition)tickMarkPosition

Discussion

The default alignments are NSTickMarkBelow and NSTickMarkLeft.

Availability

Available in Mac OS X v10.4 and later.

See Also- [setTickMarkPosition:](#) (page 12)**Declared In**

NSLevelIndicator.h

tickMarkValueAtIndex:

Returns the receiver's value represented by the tick mark at index (the minimum-value tick mark has an index of 0).

- (double)tickMarkValueAtIndex:(NSInteger)index

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSLevelIndicator.h

warningValue

Returns the receiver's warning value.

- (double)warningValue

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicator.h

Document Revision History

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

Date	Notes
2007-03-01	Revised task headings.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

C

`criticalValue` **instance method** [9](#)

M

`maxValue` **instance method** [9](#)

`minValue` **instance method** [9](#)

N

`numberOfMajorTickMarks` **instance method** [9](#)

`numberOfTickMarks` **instance method** [10](#)

R

`rectOfTickMarkAtIndex:` **instance method** [10](#)

S

`setCriticalValue:` **instance method** [10](#)

`setMaxValue:` **instance method** [11](#)

`setMinValue:` **instance method** [11](#)

`setNumberOfMajorTickMarks:` **instance method** [11](#)

`setNumberOfTickMarks:` **instance method** [12](#)

`setTickMarkPosition:` **instance method** [12](#)

`setWarningValue:` **instance method** [13](#)

T

`tickMarkPosition` **instance method** [13](#)

`tickMarkValueAtIndex:` **instance method** [13](#)

W

`warningValue` **instance method** [14](#)