
NSLevelIndicatorCell Class Reference

User Experience: Controls



2009-06-03



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, 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

NSLevelIndicatorCell Class Reference 5

Overview	5
Tasks	5
Initializing NSLevelIndicatorCell Objects	5
Configuring the Range of Values	5
Managing Tick Marks	6
Setting the Level-Indicator Image	6
Instance Methods	7
criticalValue	7
initWithLevelIndicatorStyle:	7
levelIndicatorStyle	7
maxValue	8
minValue	8
numberOfMajorTickMarks	8
numberOfTickMarks	8
rectOfTickMarkAtIndex:	9
setCriticalValue:	9
setImage:	9
setLevelIndicatorStyle:	10
setMaxValue:	10
setMinValue:	10
setNumberOfMajorTickMarks:	11
setNumberOfTickMarks:	11
setTickMarkPosition:	11
setWarningValue:	12
tickMarkPosition	12
tickMarkValueAtIndex:	12
warningValue	13
Constants	13
NSLevelIndicatorStyle	13

Document Revision History 15

Index 17

NSLevelIndicatorCell Class Reference

Inherits from	NSActionCell : NSCell : NSObject
Conforms to	NSCoding (NSCell) NSCopying (NSCell) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.4 and later.
Declared in	NSLevelIndicatorCell.h

Overview

`NSLevelIndicatorCell` is a subclass of `NSActionCell` that provides several level indicator display styles including: capacity, ranking and relevancy. The capacity style provides both continuous and discrete modes.

Tasks

Initializing NSLevelIndicatorCell Objects

- [initWithLevelIndicatorStyle:](#) (page 7)
Initializes the receiver with the style specified by *levelIndicatorStyle*.

Configuring the Range of Values

- [setMinValue:](#) (page 10)
Sets the minimum value the receiver can represent to *minValue*.
- [minValue](#) (page 8)
Returns the receiver's minimum value.
- [setMaxValue:](#) (page 10)
Sets the maximum value the receiver can represent to *maxValue*.
- [maxValue](#) (page 8)
Returns the receiver's maximum value.
- [setLevelIndicatorStyle:](#) (page 10)
Sets the style of the receiver to *levelIndicatorStyle*.

- [levelIndicatorStyle](#) (page 7)
Returns the level indicator style of the receiver.
- [setWarningValue:](#) (page 12)
Sets the receiver's warning value to *warningValue*.
- [warningValue](#) (page 13)
Returns the receiver's warning value.
- [setCriticalValue:](#) (page 9)
Sets the receiver's critical value to *criticalValue*.
- [criticalValue](#) (page 7)
Returns the receiver's critical value.

Managing Tick Marks

- [setTickMarkPosition:](#) (page 11)
Sets where tick marks appear relative to the receiver.
- [tickMarkPosition](#) (page 12)
Returns how the receiver's tick marks are aligned with it.
- [setNumberOfTickMarks:](#) (page 11)
Sets the number of tick marks displayed by the receiver (which include those assigned to the minimum and maximum values) to *numberOfTickMarks*.
- [numberOfTickMarks](#) (page 8)
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 8)
Returns the number of major tick marks associated with the receiver.
- [tickMarkValueAtIndex:](#) (page 12)
Returns the receiver's value represented by the tick mark at index (the minimum-value tick mark has an index of 0).
- [rectOfTickMarkAtIndex:](#) (page 9)
Returns the bounding rectangle of the tick mark identified by *index* (the minimum-value tick mark is at index 0).

Setting the Level-Indicator Image

- [setImage:](#) (page 9)
Sets the image displayed by the receiver for the `NSRatingLevelIndicatorStyle` to *image*.

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 9)

Declared In

NSLevelIndicatorCell.h

initWithLevelIndicatorStyle:

Initializes the receiver with the style specified by *levelIndicatorStyle*.

- (id)initWithLevelIndicatorStyle:(NSLevelIndicatorStyle)levelIndicatorStyle

Discussion

The default value and minimum value are 0.0. The default maximum value is dependent on *levelIndicatorStyle*. For continuous styles, the default maximum value is 100.0. For discrete styles the default maximum value is 5.0.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSLevelIndicatorCell.h

levelIndicatorStyle

Returns the level indicator style of the receiver.

- (NSLevelIndicatorStyle)levelIndicatorStyle

Discussion

Possible return values are described in "[Constants](#)" (page 13).

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicatorCell.h

maxValue

Returns the receiver's maximum value.

- (double)maxValue

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicatorCell.h

minValue

Returns the receiver's minimum value.

- (double)minValue

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicatorCell.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

NSLevelIndicatorCell.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 11)

Declared In

NSLevelIndicatorCell.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

NSLevelIndicatorCell.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 7)

Declared In

NSLevelIndicatorCell.h

setImage:

Sets the image displayed by the receiver for the NSRatingLevelIndicatorStyle to *image*.

- (void)setImage:(NSImage *)*image*

Discussion

The image is lightened to indicate a highlighted selection and dots are drawn for empty spots. The image is not stretched and no space is added between images. Setting *image* to *nil* causes the default star image to be used.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSLevelIndicatorCell.h

setLevelIndicatorStyle:

Sets the style of the receiver to *levelIndicatorStyle*.

- (void)setLevelIndicatorStyle:(NSLevelIndicatorStyle)levelIndicatorStyle

Discussion

The available values of *levelIndicatorStyle* are described in “[Constants](#)” (page 13).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [levelIndicatorStyle](#) (page 7)

Related Sample Code

ButtonMadness

Declared In

NSLevelIndicatorCell.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

- [levelIndicatorStyle](#) (page 7)

Declared In

NSLevelIndicatorCell.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 8)

Declared In

NSLevelIndicatorCell.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.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [numberOfMajorTickMarks](#) (page 8)

Declared In

NSLevelIndicatorCell.h

setNumberOfTickMarks:

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

- (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 8)

Declared In

NSLevelIndicatorCell.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 8) returns 0).

Availability

Available in Mac OS X v10.4 and later.

See Also

- [tickMarkPosition](#) (page 12)

Declared In

NSLevelIndicatorCell.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 13)

Declared In

NSLevelIndicatorCell.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 11)

Declared In

NSLevelIndicatorCell.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

NSLevelIndicatorCell.h

warningValue

Returns the receiver's warning value.

- (double)warningValue

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSLevelIndicatorCell.h

Constants

NSLevelIndicatorStyle

The following constants specify the level indicator's style and are used by [initWithLevelIndicatorStyle:](#) (page 7), [levelIndicatorStyle](#) (page 7), and [setLevelIndicatorStyle:](#) (page 10).

```
enum {
    NSRelevancyLevelIndicatorStyle,
    NSContinuousCapacityLevelIndicatorStyle,
    NSDiscreteCapacityLevelIndicatorStyle,
    NSRatingLevelIndicatorStyle
};
typedef NSUInteger NSLevelIndicatorStyle;
```

Constants

NSRelevancyLevelIndicatorStyle

A style similar to the rank column displayed when searching in Mail.app.

Available in Mac OS X v10.4 and later.

Declared in NSLevelIndicatorCell.h.

NSRatingLevelIndicatorStyle

A style similar to the star ranking displays provided in iTunes and iPhoto.

Available in Mac OS X v10.4 and later.

Declared in NSLevelIndicatorCell.h.

NSDiscreteCapacityLevelIndicatorStyle

A style similar to audio level indicators in iTunes.

Available in Mac OS X v10.4 and later.

Declared in NSLevelIndicatorCell.h.

NSContinuousCapacityLevelIndicatorStyle

A style that is often used to indicate conditions such as how much data is on a hard disk.

Available in Mac OS X v10.4 and later.

Declared in NSLevelIndicatorCell.h.

Document Revision History

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

Date	Notes
2009-06-03	Added NSLevelIndicatorStyle typedef.
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** [7](#)

I

`initWithLevelIndicatorStyle:` **instance method** [7](#)

L

`levelIndicatorStyle` **instance method** [7](#)

M

`maxValue` **instance method** [8](#)

`minValue` **instance method** [8](#)

N

`NSContinuousCapacityLevelIndicatorStyle`
constant [14](#)

`NSDiscreteCapacityLevelIndicatorStyle` **constant**
[14](#)

`NSLevelIndicatorStyle` [13](#)

`NSRatingLevelIndicatorStyle` **constant** [13](#)

`NSRelevancyLevelIndicatorStyle` **constant** [13](#)

`numberOfMajorTickMarks` **instance method** [8](#)

`numberOfTickMarks` **instance method** [8](#)

R

`rectOfTickMarkAtIndex:` **instance method** [9](#)

S

`setCriticalValue:` **instance method** [9](#)

`setImage:` **instance method** [9](#)

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

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

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

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

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

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

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

T

`tickMarkPosition` **instance method** [12](#)

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

W

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