
NSScroller Class Reference

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



2009-04-08



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, Mac, Mac OS, and Quartz 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

NSScroller Class Reference 5

Class at a Glance	5
Overview	6
Tasks	6
Determining NSScroller Size	6
Laying out an NSScroller	6
Setting the Knob Position	6
Calculating Layout	7
Drawing the Parts	7
Event Handling	7
Setting Control Tint	8
Class Methods	8
scrollerWidth	8
scrollerWidthForControlSize:	8
Instance Methods	8
arrowsPosition	8
checkSpaceForParts	9
controlSize	9
controlTint	9
drawArrow:highlight:	10
drawKnob	10
drawKnobSlotInRect:highlight:	11
drawParts	11
highlight:	11
hitPart	12
knobProportion	12
rectForPart:	12
setArrowsPosition:	13
setControlSize:	13
setControlTint:	14
setKnobProportion:	14
testPart:	14
trackKnob:	15
trackScrollButtons:	15
usableParts	15
Constants	16
NSScrollerPart	16
NSScrollerArrow	17
NSScrollArrowPosition	17
NSUsableScrollerParts	18

Appendix A **Deprecated NSScroller Methods** 21

Deprecated in Mac OS X v10.5 21

 setFloatValue:knobProportion: 21

Document Revision History 23

Index 25

NSScroller 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 v10.0 and later.
Companion guide	Cocoa Drawing Guide
Declared in	NSScroller.h
Related sample code	PDFKitLinker2 Quartz Composer WWDC 2005 TextEdit TextEditPlus

Class at a Glance

An NSScroller object is a user control for scrolling a document view within a container view. You normally don't need to program with NSScrollers, as Interface Builder allows you to fully configure them with an NSScrollView.

Principal Attributes

- Scrolling by small and large increments
- Proportional knob showing visible amount of document

Commonly Used Methods

[hitPart](#) (page 12)

Indicates where the user clicked the NSScroller.

[floatValue](#) (NSControl)

Returns the position of the NSScroller's knob.

[setFloatValue:knobProportion:](#) (page 21)

Sets the position and size of the NSScroller's knob.

Overview

An `NSScroller` object controls scrolling of a document view within an `NSScrollView`'s clip view (or potentially another kind of container view). It typically displays a pair of buttons that the user can click to scroll by a small amount (called a line increment or decrement) and Alt-click to scroll by a large amount (called a page increment or decrement), plus a slot containing a knob that the user can drag directly to the desired location. The knob indicates both the position within the document view and, by varying in size within the slot, the amount visible relative to the size of the document view. You can configure whether an `NSScroller` object uses scroll buttons, but it always draws the knob when there's room for it.

Don't use an `NSScroller` when an `NSSlider` would be better. A slider represents a range of values for something in the application and lets the user choose a setting. A scroller represents the relative position of the visible portion of a view and lets the user choose which portion to view.

Tasks

Determining NSScroller Size

- + `scrollerWidth` (page 8)
Returns the width of "normal-sized" instances.
- + `scrollerWidthForControlSize:` (page 8)
Returns the width of the scroller based on `controlSize`.
- `setControlSize:` (page 13)
Sets the size of the receiver.
- `controlSize` (page 9)
Returns the size of the receiver.

Laying out an NSScroller

- `setArrowsPosition:` (page 13)
Sets the location of the scroll buttons within the receiver to `location`, or inhibits their display.
- `arrowsPosition` (page 8)
Returns the location of the scroll buttons within the receiver, as described in `NSScrollArrowPosition` (page 17).

Setting the Knob Position

- `setKnobProportion:` (page 14)
Sets the proportion of the knob slot the knob should fill.
- `knobProportion` (page 12)
Returns the portion of the knob slot the knob should fill, as a floating-point value from 0.0 (minimal size) to 1.0 (fills the slot).

- [setFloatValue:knobProportion:](#) (page 21) **Deprecated in Mac OS X v10.5**
Sets the position of the knob to *aFloat*, which is a value from 0.0 (indicating the top or left end) to 1.0 (the bottom or right end). (**Deprecated**. Code that targets Mac OS X 10.5 and later should use [setKnobProportion:](#) (page 14) and [setDoubleValue:](#).)

Calculating Layout

- [rectForPart:](#) (page 12)
Returns the rectangle occupied by *aPart*, which for this method is interpreted literally rather than as an indicator of scrolling direction.
- [testPart:](#) (page 14)
Returns the part that would be hit by a mouse-down event at *aPoint* (expressed in the window's coordinate system).
- [checkSpaceForParts](#) (page 9)
Checks to see if there is enough room in the receiver to display the knob and buttons.
- [usableParts](#) (page 15)
Returns a value indicating which parts of the receiver are displayed and usable.

Drawing the Parts

- [drawArrow:highlight:](#) (page 10)
Draws the scroll button indicated by *arrow*, which is either `NSScrollerIncrementArrow` (the down or right scroll button) or `NSScrollerDecrementArrow` (up or left).
- [drawKnobSlotInRect:highlight:](#) (page 11)
Draws the portion of the scroller's track, possibly including the line increment and decrement arrow buttons, that falls in the given *slotRect*.
- [drawKnob](#) (page 10)
Draws the knob.
- [drawParts](#) (page 11)
Caches images for the scroll buttons and knob.
- [highlight:](#) (page 11)
Highlights or unhighlights the scroll button the user clicked.

Event Handling

- [hitPart](#) (page 12)
Returns a part code indicating the manner in which the scrolling should be performed.
- [trackKnob:](#) (page 15)
Tracks the knob and sends action messages to the receiver's target.
- [trackScrollButtons:](#) (page 15)
Tracks the scroll buttons and sends action messages to the receiver's target.

Setting Control Tint

- [controlTint](#) (page 9)
Returns the receiver's control tint.
- [setControlTint:](#) (page 14)
Sets the receiver's control tint.

Class Methods

scrollerWidth

Returns the width of “normal-sized” instances.

```
+ (CGFloat)scrollerWidth
```

Discussion

NSScrollView uses this value to lay out its components. Subclasses that use a different width should override this method.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

scrollerWidthForControlSize:

Returns the width of the scroller based on *controlSize*.

```
+ (CGFloat)scrollerWidthForControlSize:(NSControlSize)controlSize
```

Discussion

Valid values for *controlSize* are described in NSCell's NSControlSize.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

Instance Methods

arrowsPosition

Returns the location of the scroll buttons within the receiver, as described in [NSScrollArrowPosition](#) (page 17).

- (NSScrollArrowPosition)arrowsPosition

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

NSScroller.h

checkSpaceForParts

Checks to see if there is enough room in the receiver to display the knob and buttons.

- (void)checkSpaceForParts

Discussion

[usableParts](#) (page 15) returns the state calculated by this method. You should never need to invoke this method; it's invoked automatically whenever the NSScroller's size changes.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

controlSize

Returns the size of the receiver.

- (NSControlSize)controlSize

Discussion

Valid return values are described in NSControlSize.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

NSScroller.h

controlTint

Returns the receiver's control tint.

- (NSControlTint)controlTint

Discussion

Valid return values are described in `NSControlTint`.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

`NSScroller.h`

drawArrow:highlight:

Draws the scroll button indicated by *arrow*, which is either `NSScrollerIncrementArrow` (the down or right scroll button) or `NSScrollerDecrementArrow` (up or left).

```
- (void)drawArrow:(NSScrollerArrow)arrow highlight:(BOOL)flag
```

Discussion

If *flag* is YES, the button is drawn highlighted; otherwise it's drawn normally. You should never need to invoke this method directly, but may wish to override it to customize the appearance of scroll buttons.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [drawKnob](#) (page 10)
- [rectForPart:](#) (page 12)

Declared In

`NSScroller.h`

drawKnob

Draws the knob.

```
- (void)drawKnob
```

Discussion

You should never need to invoke this method directly, but may wish to override it to customize the appearance of the knob.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [drawArrow:highlight:](#) (page 10)
- [rectForPart:](#) (page 12)

Declared In

`NSScroller.h`

drawKnobSlotInRect:highlight:

Draws the portion of the scroller's track, possibly including the line increment and decrement arrow buttons, that falls in the given `slotRect`.

```
- (void)drawKnobSlotInRect:(NSRect)slotRecthighlight:(BOOL)flag
```

Parameters

slotRect

The rectangle in which to draw the knob slot.

flag

If `flag` is YES, any scroll arrow button that falls within `slotRect` is drawn highlighted; otherwise it's drawn normally.

Discussion

Only one arrow button will be shown highlighted at a time, so you can expect this method to sometimes be invoked with a `slotRect` that encompasses only one arrow.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSScroller.h`

drawParts

Caches images for the scroll buttons and knob.

```
- (void)drawParts
```

Discussion

It's invoked only once when the NSScroller is created. You may want to override this method if you alter the look of the NSScroller, but you should never invoke it directly.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScroller.h`

highlight:

Highlights or unhighlights the scroll button the user clicked.

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

Discussion

The receiver invokes this method while tracking the mouse; you should not invoke it directly. If `flag` is YES, the appropriate part is drawn highlighted; otherwise it's drawn normally.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [drawArrow:highlight:](#) (page 10)
- [rectForPart:](#) (page 12)

Declared In

NSScroller.h

hitPart

Returns a part code indicating the manner in which the scrolling should be performed.

```
- (NSScrollerPart)hitPart
```

Discussion

See [NSScrollerPart](#) (page 16) for a list of part codes.

This method is typically invoked by an NSScrollView to determine how to scroll its document view when it receives an action message from the NSScroller.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

knobProportion

Returns the portion of the knob slot the knob should fill, as a floating-point value from 0.0 (minimal size) to 1.0 (fills the slot).

```
- (CGFloat)knobProportion
```

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

rectForPart:

Returns the rectangle occupied by *aPart*, which for this method is interpreted literally rather than as an indicator of scrolling direction.

```
- (NSRect)rectForPart:(NSScrollerPart)aPart
```

Discussion

See [NSScrollerPart](#) (page 16) for a list of possible values for *aPart*.

Note the interpretations of `NSScrollerDecrementPage` and `NSScrollerIncrementPage`. The actual part of an NSScroller that causes page-by-page scrolling varies, so as a convenience these part codes refer to useful parts different from the scroll buttons.

Returns `NSZeroRect` if the part requested isn't present on the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [hitPart](#) (page 12)
- [testPart:](#) (page 14)
- [usableParts](#) (page 15)

Declared In

`NSScroller.h`

setArrowsPosition:

Sets the location of the scroll buttons within the receiver to *location*, or inhibits their display.

```
- (void)setArrowsPosition:(NSScrollArrowPosition)location
```

Discussion

See [NSScrollArrowPosition](#) (page 17) for a list of possible values for *location*.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [arrowsPosition](#) (page 8)

Declared In

`NSScroller.h`

setControlSize:

Sets the size of the receiver.

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

Discussion

Valid values for *controlSize* are described in `NSControlSize`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [controlSize](#) (page 9)

Related Sample Code

Quartz Composer QCTV
WhackedTV

Declared In

`NSScroller.h`

setControlTint:

Sets the receiver's control tint.

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

Discussion

Valid values for *controlTint* are described in NSControlTint.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [controlTint](#) (page 9)

Declared In

NSScroller.h

setKnobProportion:

Sets the proportion of the knob slot the knob should fill.

- (void)setKnobProportion:(CGFloat)*proportion*

Parameters

proportion

A floating point value between 0.0 (minimal size) and 1.0 (fills the entire slot).

Discussion

Code that targets Mac OS X 10.5 and later should use [-setKnobProportion:](#) (page 14) and [setDoubleValue::](#) in preference to the deprecated method that they replace, [setFloatValue:knobProportion:](#) (page 21). These methods provide more uniform, Key Value Coding-compatible access to the two values, and allow for a double-precision scroll position.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSScroller.h

testPart:

Returns the part that would be hit by a mouse-down event at *aPoint* (expressed in the window's coordinate system).

- (NSScrollerPart)testPart:(NSPoint)*aPoint*

Discussion

See [NSScrollerPart](#) (page 16) for a list of possible return values.

Note the interpretations of [NSScrollerDecrementPage](#) and [NSScrollerIncrementPage](#). The actual part of an NSScroller that causes page-by-page scrolling varies, so as a convenience these part codes refer to useful parts different from the scroll buttons.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [hitPart](#) (page 12)
- [rectForPart:](#) (page 12)

Declared In

NSScroller.h

trackKnob:

Tracks the knob and sends action messages to the receiver's target.

```
- (void)trackKnob:(NSEvent *)theEvent
```

Discussion

This method is invoked automatically when the receiver receives *theEvent* mouse-down event in the knob; you should not invoke it directly.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

trackScrollButtons:

Tracks the scroll buttons and sends action messages to the receiver's target.

```
- (void)trackScrollButtons:(NSEvent *)theEvent
```

Discussion

This method is invoked automatically when the receiver receives *theEvent* mouse-down event in a scroll button; you should not invoke this method directly.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

usableParts

Returns a value indicating which parts of the receiver are displayed and usable.

```
- (NSUsableScrollerParts)usableParts
```

Discussion

See [NSUsableScrollerParts](#) (page 18) for a list of possible values.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [checkSpaceForParts](#) (page 9)
- [arrowsPosition](#) (page 8)

Declared In

NSScroller.h

Constants

NSScrollerPart

These constants specify the different parts of the scroller:

```
typedef enum _NSScrollerPart {
    NSScrollerNoPart = 0,
    NSScrollerDecrementPage = 1,
    NSScrollerKnob = 2,
    NSScrollerIncrementPage = 3,
    NSScrollerDecrementLine = 4,
    NSScrollerIncrementLine = 5,
    NSScrollerKnobSlot = 6
} NSScrollerPart;
```

Constants

NSScrollerKnob

Directly to the NSScroller's value, as given by floatValue.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerKnobSlot

Directly to the NSScroller's value, as given by floatValue.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerDecrementLine

Up or left by a small amount.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerDecrementPage

Up or left by a large amount.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerIncrementLine

Down or right by a small amount.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerIncrementPage

Down or right by a large amount.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerNoPart

Don't scroll at all.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

NSScrollerArrow

These constants describe the two scroller buttons and are used by [drawArrow:highlight:](#) (page 10):

```
typedef enum _NSScrollerArrow {  
    NSScrollerIncrementArrow = 0,  
    NSScrollerDecrementArrow = 1  
} NSScrollerArrow;
```

Constants

NSScrollerIncrementArrow

The down or right scroll button.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerDecrementArrow

The up or left scroll button.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

NSScrollArrowPosition

These constants specify where the scroller's buttons appear and are used by [arrowsPosition](#) (page 8) and [setArrowsPosition:](#) (page 13).

```
typedef enum _NSScrollArrowPosition {
    NSScrollerArrowsMaxEnd      = 0,
    NSScrollerArrowsMinEnd     = 1,
    NSScrollerArrowsDefaultSetting = 0,
    NSScrollerArrowsNone       = 2
} NSScrollArrowPosition;
```

Constants

NSScrollerArrowsMaxEnd

Buttons at bottom or right. This constant has been deprecated.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerArrowsMinEnd

Buttons at top or left. This has been deprecated.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

NSScrollerArrowsDefaultSetting

Buttons are displayed according to the system-wide appearance preferences.

Available in Mac OS X v10.1 and later.

Declared in NSScroller.h.

NSScrollerArrowsNone

No buttons.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScroller.h

NSUsableScrollerParts

These constants specify which parts of the scroller are visible.

```
typedef enum _NSUsableScrollerParts {
    NSNoScrollerParts = 0,
    NSOnlyScrollerArrows = 1,
    NSAllScrollerParts = 2
} NSUsableScrollerParts;
```

Constants

NSNoScrollerParts

Scroller has neither a knob nor scroll buttons, only the knob slot.

Available in Mac OS X v10.0 and later.

Declared in NSScroller.h.

`NSOnlyScrollerArrows`

Scroller has only scroll buttons, no knob.

Available in Mac OS X v10.0 and later.

Declared in `NSScroller.h`.

`NSAllScrollerParts`

Scroller has at least a knob, possibly also scroll buttons.

Available in Mac OS X v10.0 and later.

Declared in `NSScroller.h`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScroller.h`

Deprecated NSScroller Methods

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

Deprecated in Mac OS X v10.5

setFloatValue:knobProportion:

Sets the position of the knob to *aFloat*, which is a value from 0.0 (indicating the top or left end) to 1.0 (the bottom or right end). (Deprecated in Mac OS X v10.5. Code that targets Mac OS X 10.5 and later should use [setKnobProportion:](#) (page 14) and `setDoubleValue:.`)

```
- (void)setFloatValue:(float)aFloat knobProportion:(CGFloat)knobProp
```

Discussion

Also sets the proportion of the knob slot filled by the knob to *knobProp*, also a value from 0.0 (minimal size) to 1.0 (fills the slot).

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

See Also

- `floatValue` (NSControl)
- [knobProportion](#) (page 12)

Declared In

NSScroller.h

Document Revision History

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

Date	Notes
2009-04-08	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.
	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

arrowsPosition **instance method** [8](#)

C

checkSpaceForParts **instance method** [9](#)
controlSize **instance method** [9](#)
controlTint **instance method** [9](#)

D

drawArrow:highlight: **instance method** [10](#)
drawKnob **instance method** [10](#)
drawKnobSlotInRect:highlight: **instance method**
[11](#)
drawParts **instance method** [11](#)

H

highlight: **instance method** [11](#)
hitPart **instance method** [12](#)

K

knobProportion **instance method** [12](#)

N

NSAllScrollerParts **constant** [19](#)
NSNoScrollerParts **constant** [18](#)
NSOnlyScrollerArrows **constant** [19](#)
NSScrollArrowPosition **data type** [17](#)

NSScrollerArrow **data type** [17](#)
NSScrollerArrowsDefaultSetting **constant** [18](#)
NSScrollerArrowsMaxEnd **constant** [18](#)
NSScrollerArrowsMinEnd **constant** [18](#)
NSScrollerArrowsNone **constant** [18](#)
NSScrollerDecrementArrow **constant** [17](#)
NSScrollerDecrementLine **constant** [16](#)
NSScrollerDecrementPage **constant** [16](#)
NSScrollerIncrementArrow **constant** [17](#)
NSScrollerIncrementLine **constant** [16](#)
NSScrollerIncrementPage **constant** [17](#)
NSScrollerKnob **constant** [16](#)
NSScrollerKnobSlot **constant** [16](#)
NSScrollerNoPart **constant** [17](#)
NSScrollerPart **data type** [16](#)
NSUsableScrollerParts **data type** [18](#)

R

rectForPart: **instance method** [12](#)

S

scrollerWidth **class method** [8](#)
scrollerWidthForControlSize: **class method** [8](#)
setArrowsPosition: **instance method** [13](#)
setControlSize: **instance method** [13](#)
setControlTint: **instance method** [14](#)
setFloatValue:knobProportion: **instance method**
[21](#)
setKnobProportion: **instance method** [14](#)

T

testPart: **instance method** [14](#)
trackKnob: **instance method** [15](#)
trackScrollButtons: **instance method** [15](#)

U

usableParts instance method [15](#)