

---

# NSScrollView 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, 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

## **NSScrollView Class Reference 5**

---

Class at a Glance	5
Overview	6
Tasks	6
Calculating Layout	6
Determining Component Sizes	6
Managing Graphics Attributes	6
Managing the Scrolled Views	7
Managing Scrollers	7
Managing Rulers	8
Setting Scrolling Behavior	9
Updating Display After Scrolling	10
Arranging Components	10
Class Methods	10
contentSizeForFrameSize:hasHorizontalScroller:hasVerticalScroller:borderType:	10
frameSizeForContentSize:hasHorizontalScroller:hasVerticalScroller:borderType:	10
rulerViewClass	11
setRulerViewClass:	11
Instance Methods	12
autohidesScrollers	12
backgroundColor	12
borderType	12
contentSize	13
contentView	13
documentCursor	13
documentView	14
documentVisibleRect	14
drawsBackground	14
hasHorizontalRuler	15
hasHorizontalScroller	15
hasVerticalRuler	15
hasVerticalScroller	16
horizontalLineScroll	16
horizontalPageScroll	17
horizontalRulerView	17
horizontalScroller	18
lineScroll	18
pageScroll	18
reflectScrolledClipView:	19
rulersVisible	20
scrollsDynamically	20

- scrollWheel: 20
- setAutohidesScrollers: 21
- setBackgroundColor: 21
- setBorderType: 21
- setContentView: 22
- setDocumentCursor: 22
- setDocumentView: 23
- setDrawsBackground: 23
- setHasHorizontalRuler: 24
- setHasHorizontalScroller: 24
- setHasVerticalRuler: 25
- setHasVerticalScroller: 25
- setHorizontalLineScroll: 26
- setHorizontalPageScroll: 26
- setHorizontalRulerView: 26
- setHorizontalScroller: 27
- setLineScroll: 27
- setPageScroll: 28
- setRulersVisible: 28
- setScrollsDynamically: 29
- setVerticalLineScroll: 29
- setVerticalPageScroll: 30
- setVerticalRulerView: 30
- setVerticalScroller: 31
- tile 31
- verticalLineScroll 31
- verticalPageScroll 32
- verticalRulerView 32
- verticalScroller 33

---

**Document Revision History 35**

---

**Index 37**

---

# NSScrollView Class Reference

---

<b>Inherits from</b>	NSView : NSResponder : NSObject
<b>Conforms to</b>	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Scroll View Programming Guide for Cocoa
<b>Declared in</b>	NSScrollView.h
<b>Related sample code</b>	ColorMatching Quartz Composer WWDC 2005 TextEdit Sketch-112 TextEditPlus WhackedTV

## Class at a Glance

An `NSScrollView` allows the user to scroll a document view that's too large to display in its entirety. In addition to the document view, it displays horizontal and vertical scrollers and rulers (depending on which it's configured to have).

## Principal Attributes

---

- Configurable scrollers
- Configurable rulers
- Small and large increment scrolling
- Dynamic (continuous) scrolling
- Display of a special cursor over its document view

### Interface Builder

Drag a scroll view to a window.

## Commonly Used Methods

---

[scrollView:setDocumentView:](#) (page 23)

Sets the scroll view's document view.

[scrollView:setLineScroll:](#) (page 27)

Sets the amount by which the document view moves during scrolling.

[scrollView:setRulersVisible:](#) (page 28)

Displays or hides rulers.

## Overview

The `NSScrollView` class is the central coordinator for the Application Kit's scrolling machinery, composed of this class, `NSClipView`, and `NSScroller`. An `NSScrollView` displays a portion of a document view that's too large to be displayed whole and provides `NSScroller` scroll bars that allow the user to move the document view within the `NSScrollView`. Note that, when using an `NSClipView` within an `NSScrollView` (the usual configuration), you should issue messages that control background drawing state to the `NSScrollView`, rather than messaging the `NSClipView` directly.

## Tasks

### Calculating Layout

+ [scrollView:contentSizeForFrameSize:hasHorizontalScroller:hasVerticalScroller:borderType:](#) (page 10)

Returns the size of a content view for an `NSScrollView` whose frame size is *frameSize*.

+ [scrollView:frameSizeForContentSize:hasHorizontalScroller:hasVerticalScroller:borderType:](#) (page 10)

Returns the frame size of an `NSScrollView` that contains a content view whose size is *contentType*.

### Determining Component Sizes

- [scrollView:contentType:](#) (page 13)

Returns the size of the receiver's content view.

- [scrollView:documentVisibleRect:](#) (page 14)

Returns the portion of the document view, in its own coordinate system, visible through the receiver's content view.

### Managing Graphics Attributes

- [scrollView:setBackgroundColor:](#) (page 21)

Sets the color of the content view's background to *aColor*.

- [backgroundColor](#) (page 12)  
Returns the content view's background color.
- [drawsBackground](#) (page 14)  
Returns YES if the receiver cell fills the background with its background color; otherwise, NO.
- [setDrawsBackground:](#) (page 23)  
Sets whether the receiver draws its background.
- [setBorderType:](#) (page 21)  
Sets the border type of the receiver to *borderType*.
- [borderType](#) (page 12)  
Returns a value that represents the type of border surrounding the receiver; see the description of [setBorderType:](#) (page 21) for a list of possible values.
- [setDocumentCursor:](#) (page 22)  
Sets the cursor used when the cursor is over the content view to *aCursor*, by sending [setDocumentCursor:](#) (page 22) to the content view.
- [documentCursor](#) (page 13)  
Returns the content view's document cursor.

## Managing the Scrolled Views

- [setContentView:](#) (page 22)  
Sets the receiver's content view, the view that clips the document view, to *aView*.
- [contentView](#) (page 13)  
Returns the receiver's content view, the view that clips the document view.
- [setDocumentView:](#) (page 23)  
Sets the receiver's document view to *aView*.
- [documentView](#) (page 14)  
Returns the view the receiver scrolls within its content view.

## Managing Scrollers

- [setHorizontalScroller:](#) (page 27)  
Sets the receiver's horizontal scroller to *aScroller*, establishing the appropriate target-action relationships between them.
- [horizontalScroller](#) (page 18)  
Returns the receiver's horizontal scroller, regardless of whether the receiver is currently displaying it, or nil if the receiver has none.
- [setHasHorizontalScroller:](#) (page 24)  
Determines whether the receiver keeps a horizontal scroller.
- [hasHorizontalScroller](#) (page 15)  
Returns YES if the receiver displays a horizontal scroller, NO if it doesn't.
- [setVerticalScroller:](#) (page 31)  
Sets the receiver's vertical scroller to *aScroller*, establishing the appropriate target-action relationships between them.

- [verticalScroller](#) (page 33)  
Returns the receiver's vertical scroller, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.
- [setHasVerticalScroller:](#) (page 25)  
Determines whether the receiver keeps a vertical scroller.
- [hasVerticalScroller](#) (page 16)  
Returns YES if the receiver displays a vertical scroller, NO if it doesn't.
- [setAutohidesScrollers:](#) (page 21)  
Determines whether the receiver automatically hides its scroll bars when they are not needed.
- [autohidesScrollers](#) (page 12)  
Returns YES when autohiding is set for scroll bars in the receiver.

## Managing Rulers

- + [setRulerViewClass:](#) (page 11)  
Sets the default class to be used for ruler objects in NSScrollViews to *aClass*.
- + [rulerViewClass](#) (page 11)  
Returns the default class to be used for ruler objects in NSScrollViews.
- [setHasHorizontalRuler:](#) (page 24)  
Determines whether the receiver keeps a horizontal ruler object.
- [hasHorizontalRuler](#) (page 15)  
Returns YES if the receiver maintains a horizontal ruler view, NO if it doesn't.
- [setHorizontalRulerView:](#) (page 26)  
Sets the receiver's horizontal ruler view to *aRulerView*.
- [horizontalRulerView](#) (page 17)  
Returns the receiver's horizontal ruler view, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.
- [setHasVerticalRuler:](#) (page 25)  
Determines whether the receiver keeps a vertical ruler object.
- [hasVerticalRuler](#) (page 15)  
Returns YES if the receiver maintains a vertical ruler view, NO if it doesn't.
- [setVerticalRulerView:](#) (page 30)  
Sets the receiver's vertical ruler view to *aRulerView*.
- [verticalRulerView](#) (page 32)  
Returns the receiver's vertical ruler view, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.
- [setRulersVisible:](#) (page 28)  
Determines whether the receiver displays its rulers.
- [rulersVisible](#) (page 20)  
Returns YES if the receiver was set to show rulers using [setRulersVisible:](#) (page 28) (whether or not it has rulers at all), NO if it was set to hide them.



## Setting Scrolling Behavior

- [setLineScroll:](#) (page 27)  
Sets the horizontal and vertical line scroll amounts to *aFloat*.
- [lineScroll](#) (page 18)  
Returns the vertical line scroll amount: the amount by which the receiver scrolls itself vertically when scrolling line by line, expressed in the content view's coordinate system.
- [setHorizontalLineScroll:](#) (page 26)  
Sets the amount by which the receiver scrolls itself horizontally when scrolling line by line to *aFloat*, expressed in the content view's coordinate system.
- [horizontalLineScroll](#) (page 16)  
Returns the amount by which the receiver scrolls itself horizontally when scrolling line by line, expressed in the content view's coordinate system.
- [setVerticalLineScroll:](#) (page 29)  
Sets the amount by which the receiver scrolls itself vertically when scrolling line by line to *aFloat*, expressed in the content view's coordinate system.
- [verticalLineScroll](#) (page 31)  
Returns the amount by which the receiver scrolls itself vertically when scrolling line by line, expressed in the content view's coordinate system.
- [setPageScroll:](#) (page 28)  
Sets the horizontal and vertical page scroll amounts to *aFloat*.
- [pageScroll](#) (page 18)  
Returns the vertical page scroll amount: the amount of the document view kept visible when scrolling vertically page by page, expressed in the content view's coordinate system.
- [setHorizontalPageScroll:](#) (page 26)  
Sets the amount of the document view kept visible when scrolling horizontally page by page to *aFloat*, expressed in the content view's coordinate system.
- [horizontalPageScroll](#) (page 17)  
Returns the amount of the document view kept visible when scrolling horizontally page by page, expressed in the content view's coordinate system.
- [setVerticalPageScroll:](#) (page 30)  
Sets the amount of the document view kept visible when scrolling vertically page by page to *aFloat*, expressed in the content view's coordinate system.
- [verticalPageScroll](#) (page 32)  
Returns the amount of the document view kept visible when scrolling vertically page by page, expressed in the content view's coordinate system.
- [setScrollsDynamically:](#) (page 29)  
Determines whether the receiver redraws its document view while scrolling continuously.
- [scrollsDynamically](#) (page 20)  
Returns YES if the receiver redraws its document view while tracking the knob, NO if it redraws only when the scroller knob is released.
- [scrollWheel:](#) (page 20)  
Scrolls the receiver up or down, in response to the user moving the mouse's scroll wheel specified by *theEvent*.

## Updating Display After Scrolling

- [reflectScrolledClipView:](#) (page 19)  
Adjusts the receiver's scrollers to reflect the size and positioning of its content view.

## Arranging Components

- [tile](#) (page 31)  
Lays out the components of the receiver: the content view, the scrollers, and the ruler views.

## Class Methods

### **contentSizeForFrameSize:hasHorizontalScroller:hasVerticalScroller:borderType:**

Returns the size of a content view for an NSScrollView whose frame size is *frameSize*.

```
+ (NSSize)contentSizeForFrameSize:(NSSize)frameSize hasHorizontalScroller:(BOOL)hFlag
  hasVerticalScroller:(BOOL)vFlag borderType:(NSBorderType)borderType
```

#### Discussion

*hFlag* and *vFlag* indicate whether a horizontal or vertical scroller, respectively, is present. If either flag is YES then the content size is reduced in the appropriate dimension by the width of an NSScroller. The *borderType* argument indicates the appearance of the NSScrollView's edge, which also affects the content size; see the description of [setBorderType:](#) (page 21) for a list of possible values.

For an existing NSScrollView, you can simply use the [contentSize](#) (page 13) method.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- + [frameSizeForContentSize:hasHorizontalScroller:hasVerticalScroller:borderType:](#) (page 10)
- + [scrollerWidth](#) (NSScroller)

#### Related Sample Code

Quartz Composer WWDC 2005 TextEdit  
TextEditPlus

#### Declared In

NSScrollView.h

### **frameSizeForContentSize:hasHorizontalScroller:hasVerticalScroller:borderType:**

Returns the frame size of an NSScrollView that contains a content view whose size is *contentSize*.

```
+ (NSSize)frameSizeForContentSize:(NSSize) contentSize
   hasHorizontalScroller:(BOOL) hFlag hasVerticalScroller:(BOOL) vFlag
   borderType:(NSBorderType) borderType
```

**Discussion**

The *hFlag* and *vFlag* arguments indicate whether a horizontal or vertical scroller, respectively, is present. If either flag is YES then the frame size is increased in the appropriate dimension by the width of an NSScroller. *borderType* indicates the appearance of the NSScrollView's edge, which also affects the frame size; see the description of [setBorderType:](#) (page 21) for a list of possible values.

For an existing NSScrollView, you can simply use the `frame` method and extract its size.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [contentSizeForFrameSize:hasHorizontalScroller:hasVerticalScroller:borderType:](#) (page 10)

+ `scrollerWidth` (NSScroller)

**Related Sample Code**

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

**Declared In**

NSScrollView.h

## rulerViewClass

Returns the default class to be used for ruler objects in NSScrollViews.

```
+ (Class)rulerViewClass
```

**Discussion**

This class is normally NSRulerView.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [setRulerViewClass:](#) (page 11)

**Declared In**

NSScrollView.h

## setRulerViewClass:

Sets the default class to be used for ruler objects in NSScrollViews to *aClass*.

```
+ (void)setRulerViewClass:(Class) aClass
```

**Discussion**

This class is normally NSRulerView, but you can use this method to set it to a custom subclass of NSRulerView.

This method simply sets a global variable private to NSScrollView. Subclasses of NSScrollView should override both this method and [rulerViewClass](#) (page 11) to store their ruler view classes in private variables.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [rulerViewClass](#) (page 11)

**Declared In**

NSScrollView.h

## Instance Methods

### autohidesScrollers

Returns YES when autohiding is set for scroll bars in the receiver.

- (BOOL)autohidesScrollers

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setAutohidesScrollers:](#) (page 21)

**Declared In**

NSScrollView.h

### backgroundColor

Returns the content view's background color.

- (NSColor \*)backgroundColor

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setBackgroundColor:](#) (page 21)

- [backgroundColor](#) (NSClipView)

**Declared In**

NSScrollView.h

### borderType

Returns a value that represents the type of border surrounding the receiver; see the description of [setBorderType:](#) (page 21) for a list of possible values.

- (NSBorderType)borderType

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSScrollView.h

## contentSize

Returns the size of the receiver's content view.

- (NSSize)contentSize

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

+ [contentSizeForFrameSize:hasHorizontalScroller:hasVerticalScroller:borderType:](#) (page 10)

**Related Sample Code**

CustomSave

**Declared In**

NSScrollView.h

## contentView

Returns the receiver's content view, the view that clips the document view.

- (NSClipView \*)contentView

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setContentView:](#) (page 22)

- [documentView](#) (page 14)

**Related Sample Code**

WhackedTV

**Declared In**

NSScrollView.h

## documentCursor

Returns the content view's document cursor.

- (NSCursor \*)documentCursor

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setDocumentCursor:](#) (page 22)
- `documentCursor` (NSClipView)

**Declared In**

NSScrollView.h

## **documentView**

Returns the view the receiver scrolls within its content view.

- (id)documentView

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setDocumentView:](#) (page 23)
- `documentView` (NSClipView)

**Declared In**

NSScrollView.h

## **documentVisibleRect**

Returns the portion of the document view, in its own coordinate system, visible through the receiver's content view.

- (NSRect)documentVisibleRect

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- `documentVisibleRect` (NSClipView)
- `visibleRect` (NSView)

**Declared In**

NSScrollView.h

## **drawsBackground**

Returns YES if the receiver cell fills the background with its background color; otherwise, NO.

- (BOOL)drawsBackground

**Availability**

Available in Mac OS X v10.0 and later.

### Declared In

NSScrollView.h

## hasHorizontalRuler

Returns YES if the receiver maintains a horizontal ruler view, NO if it doesn't.

- (BOOL)hasHorizontalRuler

### Discussion

Display of rulers is controlled using the [setRulersVisible:](#) (page 28) method.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [horizontalRulerView](#) (page 17)
- [setHasHorizontalRuler:](#) (page 24)
- [hasVerticalRuler](#) (page 15)
- + [rulerViewClass](#) (page 11)

### Declared In

NSScrollView.h

## hasHorizontalScroller

Returns YES if the receiver displays a horizontal scroller, NO if it doesn't.

- (BOOL)hasHorizontalScroller

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [horizontalScroller](#) (page 18)
- [setHasHorizontalScroller:](#) (page 24)
- [hasVerticalScroller](#) (page 16)

### Related Sample Code

Quartz Composer WWDC 2005 TextEdit  
TextEditPlus

### Declared In

NSScrollView.h

## hasVerticalRuler

Returns YES if the receiver maintains a vertical ruler view, NO if it doesn't.

- (BOOL)hasVerticalRuler

**Discussion**

Display of rulers is controlled using the [setRulersVisible:](#) (page 28) method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [verticalRulerView](#) (page 32)
- [setHasVerticalRuler:](#) (page 25)
- [hasHorizontalRuler](#) (page 15)
- + [rulerViewClass](#) (page 11)

**Declared In**

NSScrollView.h

**hasVerticalScroller**

Returns YES if the receiver displays a vertical scroller, NO if it doesn't.

- (BOOL)hasVerticalScroller

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [verticalScroller](#) (page 33)
- [setHasVerticalScroller:](#) (page 25)
- [hasHorizontalScroller](#) (page 15)

**Declared In**

NSScrollView.h

**horizontalLineScroll**

Returns the amount by which the receiver scrolls itself horizontally when scrolling line by line, expressed in the content view's coordinate system.

- (CGFloat)horizontalLineScroll

**Discussion**

This amount is used when the user clicks the scroll arrows on the horizontal scroll bar without holding down a modifier key.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setHorizontalLineScroll:](#) (page 26)
- [verticalLineScroll](#) (page 31)
- [setLineScroll:](#) (page 27)
- [horizontalPageScroll](#) (page 17)



**Declared In**

NSScrollView.h

## horizontalPageScroll

Returns the amount of the document view kept visible when scrolling horizontally page by page, expressed in the content view's coordinate system.

```
- (CGFloat)horizontalPageScroll
```

**Discussion**

This amount is used when the user clicks the scroll arrows on the horizontal scroll bar while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setHorizontalPageScroll:](#) (page 26)
- [verticalPageScroll](#) (page 32)
- [setPageScroll:](#) (page 28)
- [horizontalLineScroll](#) (page 16)

**Declared In**

NSScrollView.h

## horizontalRulerView

Returns the receiver's horizontal ruler view, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.

```
- (NSRulerView *)horizontalRulerView
```

**Discussion**

If the receiver is set to display a horizontal ruler view and doesn't yet have one, this method creates an instance of the ruler view class set using the class method [setRulerViewClass:](#) (page 11). Display of rulers is controlled using the [setRulersVisible:](#) (page 28) method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [hasHorizontalRuler](#) (page 15)
- [verticalRulerView](#) (page 32)

**Related Sample Code**

Sketch-112

**Declared In**

NSScrollView.h

## horizontalScroller

Returns the receiver's horizontal scroller, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.

- (NSScroller \*)horizontalScroller

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSScrollView.h

## lineScroll

Returns the vertical line scroll amount: the amount by which the receiver scrolls itself vertically when scrolling line by line, expressed in the content view's coordinate system.

- (CGFloat)lineScroll

**Discussion**

This amount is used when the user clicks the scroll arrows on the vertical scroll bar without holding down a modifier key. As part of its implementation, this method calls [verticalLineScroll](#) (page 31).

Note that a scroll view can have two different line scroll amounts: [verticalLineScroll](#) (page 31) and [horizontalLineScroll](#) (page 16). Use this method only if you can be sure they're both the same; for example, you always use [setLineScroll:](#) (page 27), which sets both amounts to the same value.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setLineScroll:](#) (page 27)
- [verticalPageScroll](#) (page 32)
- [horizontalPageScroll](#) (page 17)

**Declared In**

NSScrollView.h

## pageScroll

Returns the vertical page scroll amount: the amount of the document view kept visible when scrolling vertically page by page, expressed in the content view's coordinate system.

- (CGFloat)pageScroll

**Discussion**

This amount is used when the user clicks the scroll arrows on the vertical scroll bar while holding down the Option key. As part of its implementation, this method calls [verticalPageScroll](#) (page 32).

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page.

Note that a scroll view can have two different page scroll amounts: [verticalPageScroll](#) (page 32) and [horizontalPageScroll](#) (page 17). Use this method only if you can be sure they're both the same; for example, you always use [setPageScroll:](#) (page 28), which sets both amounts to the same value.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [setPageScroll:](#) (page 28)
- [verticalLineScroll](#) (page 31)
- [horizontalLineScroll](#) (page 16)

#### Declared In

NSScrollView.h

## reflectScrolledClipView:

Adjusts the receiver's scrollers to reflect the size and positioning of its content view.

```
- (void)reflectScrolledClipView:(NSClipView *)aClipView
```

#### Parameters

*aClipView*

The clip view being adjusted to. If *aClipView* is any view object other than the receiver's content view, the method does nothing.

#### Discussion

This method is invoked automatically during scrolling and when an `NSClipView` object's relationship to its document view changes; you should rarely need to invoke it yourself, but may wish to override it for custom updating or other behavior. If you override this method, be sure to call the superclass implementation. If you do not, other controls (such as the current scrollers) may not be updated properly.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [contentView](#) (page 13)
- [documentView](#) (page 14)

#### Related Sample Code

WhackedTV

#### Declared In

NSScrollView.h

## rulersVisible

Returns YES if the receiver was set to show rulers using [setRulersVisible:](#) (page 28) (whether or not it has rulers at all), NO if it was set to hide them.

- (BOOL)rulersVisible

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [hasHorizontalRuler](#) (page 15)

- [hasVerticalRuler](#) (page 15)

### Related Sample Code

Sketch-112

### Declared In

NSScrollView.h

## scrollsDynamically

Returns YES if the receiver redraws its document view while tracking the knob, NO if it redraws only when the scroller knob is released.

- (BOOL)scrollsDynamically

### Discussion

NSScrollView scrolls dynamically by default.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setScrollsDynamically:](#) (page 29)

### Declared In

NSScrollView.h

## scrollWheel:

Scrolls the receiver up or down, in response to the user moving the mouse's scroll wheel specified by *theEvent*.

- (void)scrollWheel:(NSEvent \*)*theEvent*

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

NSScrollView.h

## setAutohidesScrollers:

Determines whether the receiver automatically hides its scroll bars when they are not needed.

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

### Discussion

The horizontal and vertical scroll bars are hidden independently of each other. When autohiding is on and the content of the receiver doesn't extend beyond the size of the clip view on a given axis, the scroller on that axis is removed to leave more room for the content.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [autohidesScrollers](#) (page 12)

### Related Sample Code

WhackedTV

### Declared In

NSScrollView.h

## setBackground-color:

Sets the color of the content view's background to *aColor*.

```
- (void)setBackgroundColor:(NSColor *)aColor
```

### Discussion

This color is used to paint areas inside the content view that aren't covered by the document view.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [backgroundColor](#) (page 12)  
- [setBackgroundColor:](#) (NSClipView)

### Declared In

NSScrollView.h

## setBorderType:

Sets the border type of the receiver to *borderType*.

```
- (void)setBorderType:(NSBorderType)borderType
```

### Discussion

*borderType* may be one of:

```
NSNoBorder  
NSLineBorder
```

NSBezelBorder  
NSGrooveBorder

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [borderType](#) (page 12)

**Related Sample Code**

CustomSave  
Sketch-112

**Declared In**

NSScrollView.h

**setContentView:**

Sets the receiver's content view, the view that clips the document view, to *aView*.

```
- (void)setContentView:(NSClipView *)aView
```

**Discussion**

If *aView* has a document view, this method also sets the receiver's document view to be the document view of *aView*. The original content view retains its document view.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [contentView](#) (page 13)  
- [setDocumentView:](#) (page 23)

**Declared In**

NSScrollView.h

**setDocumentCursor:**

Sets the cursor used when the cursor is over the content view to *aCursor*, by sending [setDocumentCursor:](#) (page 22) to the content view.

```
- (void)setDocumentCursor:(NSCursor *)aCursor
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [documentCursor](#) (page 13)

**Declared In**

NSScrollView.h

**setDocumentView:**

Sets the receiver's document view to *aView*.

- (void)setDocumentView:(NSView \*)*aView*

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [documentView](#) (page 14)
- [setDocumentView: \(NSClipView\)](#)

**Related Sample Code**

CustomSave  
 Quartz Composer QCTV  
 VideoViewer  
 WhackedTV

**Declared In**

NSScrollView.h

**setDrawsBackground:**

Sets whether the receiver draws its background.

- (void)setDrawsBackground:(BOOL)*flag*

**Discussion**

If *flag* is NO, copy-on-scroll is automatically disabled.

If your NSScrollView encloses an NSClipView sending a `setDrawsBackground:` message with a parameter of NO to the NSScrollView has the added effect of sending the NSClipView a `setCopiesOnScroll:` message with a parameter of NO. The side effect of sending the `setDrawsBackground:` message directly to the NSClipView instead would be the appearance of “trails” (vestiges of previous drawing) in the document view as it is scrolled.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [drawsBackground](#) (page 14)
- [copiesOnScroll \(NSClipView\)](#)
- [setDrawsBackground: \(NSClipView\)](#)

**Related Sample Code**

Quartz Composer QCTV

**Declared In**

NSScrollView.h

**setHasHorizontalRuler:**

Determines whether the receiver keeps a horizontal ruler object.

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

**Discussion**

If *flag* is YES, the receiver allocates a horizontal ruler the first time it's needed. Display of rulers is handled independently with the [setRulersVisible:](#) (page 28) method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [hasHorizontalRuler](#) (page 15)
- [horizontalRulerView](#) (page 17)
- [setHasVerticalRuler:](#) (page 25)

**Related Sample Code**

Sketch-112

**Declared In**

NSScrollView.h

**setHasHorizontalScroller:**

Determines whether the receiver keeps a horizontal scroller.

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

**Discussion**

If *flag* is YES, the receiver allocates and displays a horizontal scroller as needed. An NSScrollView by default has neither a horizontal nor a vertical scroller.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [hasHorizontalScroller](#) (page 15)
- [horizontalScroller](#) (page 18)
- [setHasVerticalScroller:](#) (page 25)

**Related Sample Code**

CustomSave

Quartz Composer QCTV

VideoViewer

WhackedTV

**Declared In**

NSScrollView.h



## setHasVerticalRuler:

Determines whether the receiver keeps a vertical ruler object.

- (void)setHasVerticalRuler:(BOOL)flag

### Discussion

If *flag* is YES, the receiver allocates a vertical ruler the first time it's needed. Display of rulers is handled independently with the [setRulersVisible:](#) (page 28) method.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [hasVerticalRuler](#) (page 15)
- [verticalRulerView](#) (page 32)
- [setHasHorizontalRuler:](#) (page 24)
- [setRulersVisible:](#) (page 28)

### Related Sample Code

Sketch-112

### Declared In

NSScrollView.h

## setHasVerticalScroller:

Determines whether the receiver keeps a vertical scroller.

- (void)setHasVerticalScroller:(BOOL)flag

### Discussion

If *flag* is YES, the receiver allocates and displays a vertical scroller as needed. An NSScrollView by default has neither a vertical nor a horizontal scroller.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [hasVerticalScroller](#) (page 16)
- [verticalScroller](#) (page 33)
- [setHasHorizontalScroller:](#) (page 24)

### Related Sample Code

CustomSave

Quartz Composer QCTV

VideoViewer

WhackedTV

### Declared In

NSScrollView.h

## setHorizontalLineScroll:

Sets the amount by which the receiver scrolls itself horizontally when scrolling line by line to *aFloat*, expressed in the content view's coordinate system.

```
- (void)setHorizontalLineScroll:(CGFloat)aFloat
```

### Discussion

This amount is the amount used when the user clicks the scroll arrows on the horizontal scroll bar without holding down a modifier key. When displaying text in an NSScrollView, for example, you might set this amount to the height of a single line of text in the default font.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [lineScroll](#) (page 18)
- [setPageScroll:](#) (page 28)

### Declared In

NSScrollView.h

## setHorizontalPageScroll:

Sets the amount of the document view kept visible when scrolling horizontally page by page to *aFloat*, expressed in the content view's coordinate system.

```
- (void)setHorizontalPageScroll:(CGFloat)aFloat
```

### Discussion

This amount is used when the user clicks the scroll arrows on the horizontal scroll bar while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page. Thus, setting the page scroll amount to 0.0 implies that the entire visible portion of the document view is replaced when a page scroll occurs.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [pageScroll](#) (page 18)
- [setLineScroll:](#) (page 27)

### Declared In

NSScrollView.h

## setHorizontalRulerView:

Sets the receiver's horizontal ruler view to *aRulerView*.

```
- (void)setHorizontalRulerView:(NSRulerView *)aRulerView
```

**Discussion**

You can use this method to override the default ruler class set using the class method [setRulerViewClass:](#) (page 11). Display of rulers is controlled using the [setRulersVisible:](#) (page 28) method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [horizontalRulerView](#) (page 17)
- [setHasHorizontalRuler:](#) (page 24)
- [setVerticalRulerView:](#) (page 30)
- [setRulersVisible:](#) (page 28)

**Declared In**

NSScrollView.h

**setHorizontalScroller:**

Sets the receiver's horizontal scroller to *aScroller*, establishing the appropriate target-action relationships between them.

```
- (void)setHorizontalScroller:(NSScroller *)aScroller
```

**Discussion**

To make sure the scroller is visible, invoke the [setHasHorizontalScroller:](#) (page 24) method with an argument of YES.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [horizontalScroller](#) (page 18)
- [setVerticalScroller:](#) (page 31)

**Declared In**

NSScrollView.h

**setLineScroll:**

Sets the horizontal and vertical line scroll amounts to *aFloat*.

```
- (void)setLineScroll:(CGFloat)aFloat
```

**Discussion**

The line scroll is the amount by which the receiver scrolls itself when scrolling line by line, expressed in the content view's coordinate system. It's used when the user clicks the scroll arrows without holding down a modifier key. When displaying text in an NSScrollView, for example, you might set this value to the height of a single line of text in the default font.

As part of its implementation, this method calls [setVerticalLineScroll:](#) (page 29) and [setHorizontalLineScroll:](#) (page 26).

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [verticalLineScroll](#) (page 31)
- [horizontalLineScroll](#) (page 16)

#### Declared In

NSScrollView.h

## setPageScroll:

Sets the horizontal and vertical page scroll amounts to *aFloat*.

```
- (void)setPageScroll:(CGFloat)aFloat
```

#### Discussion

The page scroll is the amount of the document view kept visible when scrolling page by page to *aFloat*, expressed in the content view's coordinate system. It's used when the user clicks the scroll arrows while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page. Thus, setting the page scroll amount to 0.0 implies that the entire visible portion of the document view is replaced when a page scroll occurs.

As part of its implementation, this method calls [setVerticalPageScroll:](#) (page 30) and [setHorizontalPageScroll:](#) (page 26).

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [verticalPageScroll](#) (page 32)
- [verticalLineScroll](#) (page 31)

#### Declared In

NSScrollView.h

## setRulersVisible:

Determines whether the receiver displays its rulers.

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

#### Discussion

If *flag* is YES, the receiver displays its rulers (creating them if needed). If *flag* is NO, the receiver doesn't display its rulers.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [rulersVisible](#) (page 20)
- [hasHorizontalRuler](#) (page 15)
- [hasVerticalRuler](#) (page 15)

**Related Sample Code**

Sketch-112

**Declared In**

NSScrollView.h

**setScrollsDynamically:**

Determines whether the receiver redraws its document view while scrolling continuously.

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

**Discussion**

If *flag* is YES it does; if *flag* is NO it redraws only when the scroller knob is released. NSScrollView scrolls dynamically by default.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [scrollsDynamically](#) (page 20)

**Related Sample Code**

WhackedTV

**Declared In**

NSScrollView.h

**setVerticalLineScroll:**

Sets the amount by which the receiver scrolls itself vertically when scrolling line by line to *aFloat*, expressed in the content view's coordinate system.

```
- (void)setVerticalLineScroll:(CGFloat)aFloat
```

**Discussion**

This value is the amount used when the user clicks the scroll arrows on the vertical scroll bar without holding down a modifier key. When displaying text in an NSScrollView, for example, you might set this value to the height of a single line of text in the default font.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [verticalLineScroll](#) (page 31)

- [setHorizontalLineScroll:](#) (page 26)
- [lineScroll](#) (page 18)
- [setVerticalPageScroll:](#) (page 30)

**Declared In**

NSScrollView.h

**setVerticalPageScroll:**

Sets the amount of the document view kept visible when scrolling vertically page by page to *aFloat*, expressed in the content view's coordinate system.

```
- (void)setVerticalPageScroll:(CGFloat)aFloat
```

**Discussion**

This amount is used when the user clicks the scroll arrows on the vertical scroll bar while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page. Thus, setting the page scroll amount to 0.0 implies that the entire visible portion of the document view is replaced when a page scroll occurs.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [verticalPageScroll](#) (page 32)
- [setHorizontalPageScroll:](#) (page 26)
- [pageScroll](#) (page 18)
- [setVerticalLineScroll:](#) (page 29)

**Declared In**

NSScrollView.h

**setVerticalRulerView:**

Sets the receiver's vertical ruler view to *aRulerView*.

```
- (void)setVerticalRulerView:(NSRulerView *)aRulerView
```

**Discussion**

You can use this method to override the default ruler class set using the class method [setRulerViewClass:](#) (page 11). Display of rulers is controlled using the [setRulersVisible:](#) (page 28) method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [verticalRulerView](#) (page 32)

- [setHasVerticalRuler:](#) (page 25)
- [setHorizontalRulerView:](#) (page 26)
- [setRulersVisible:](#) (page 28)

**Declared In**

NSScrollView.h

**setVerticalScroller:**

Sets the receiver's vertical scroller to *aScroller*, establishing the appropriate target-action relationships between them.

```
- (void)setVerticalScroller:(NSScroller *)aScroller
```

**Discussion**

To make sure the scroller is visible, invoke the [setHasVerticalScroller:](#) (page 25) method with an argument of YES.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [verticalScroller](#) (page 33)
- [setHorizontalScroller:](#) (page 27)

**Declared In**

NSScrollView.h

**tile**

Lays out the components of the receiver: the content view, the scrollers, and the ruler views.

```
- (void)tile
```

**Discussion**

You rarely need to invoke this method, but subclasses may override it to manage additional components.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

Quartz Composer WWDC 2005 TextEdit  
TextEditPlus

**Declared In**

NSScrollView.h

**verticalLineScroll**

Returns the amount by which the receiver scrolls itself vertically when scrolling line by line, expressed in the content view's coordinate system.

- (CGFloat)verticalLineScroll

#### Discussion

This amount is used when the user clicks the scroll arrows on the vertical scroll bar without holding down a modifier key.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [setVerticalLineScroll:](#) (page 29)
- [horizontalLineScroll](#) (page 16)
- [setLineScroll:](#) (page 27)
- [verticalPageScroll](#) (page 32)

#### Declared In

NSScrollView.h

## verticalPageScroll

Returns the amount of the document view kept visible when scrolling vertically page by page, expressed in the content view's coordinate system.

- (CGFloat)verticalPageScroll

#### Discussion

This amount is used when the user clicks the scroll arrows on the vertical scroll bar while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [setVerticalPageScroll:](#) (page 30)
- [horizontalPageScroll](#) (page 17)
- [setPageScroll:](#) (page 28)
- [verticalLineScroll](#) (page 31)

#### Declared In

NSScrollView.h

## verticalRulerView

Returns the receiver's vertical ruler view, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.

- (NSRulerView \*)verticalRulerView



**Discussion**

If the receiver is set to display a vertical ruler view and doesn't yet have one, this method creates an instance of the ruler view class set using the class method [setRulerViewClass:](#) (page 11). Display of rulers is controlled using the [setRulersVisible:](#) (page 28) method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [hasVerticalRuler](#) (page 15)
- [horizontalRulerView](#) (page 17)

**Related Sample Code**

Sketch-112

**Declared In**

NSScrollView.h

## verticalScroller

Returns the receiver's vertical scroller, regardless of whether the receiver is currently displaying it, or `nil` if the receiver has none.

```
- (NSScroller *)verticalScroller
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [hasVerticalScroller](#) (page 16)
- [setVerticalScroller:](#) (page 31)
- [horizontalScroller](#) (page 18)

**Related Sample Code**

Quartz Composer QCTV

WhackedTV

**Declared In**

NSScrollView.h



# Document Revision History

---

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

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

**REVISION HISTORY**

Document Revision History

# Index

---

## A

---

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

## B

---

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

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

## C

---

contentSize **instance method** [13](#)

contentSizeForFrameSize:hasHorizontalScroller:  
hasVerticalScroller:borderType: **class**  
**method** [10](#)

contentView **instance method** [13](#)

## D

---

documentCursor **instance method** [13](#)

documentView **instance method** [14](#)

documentVisibleRect **instance method** [14](#)

drawsBackground **instance method** [14](#)

## F

---

frameSizeForContentSize:hasHorizontalScroller:  
hasVerticalScroller:borderType: **class**  
**method** [10](#)

## H

---

hasHorizontalRuler **instance method** [15](#)

hasHorizontalScroller **instance method** [15](#)

hasVerticalRuler **instance method** [15](#)

hasVerticalScroller **instance method** [16](#)

horizontalLineScroll **instance method** [16](#)

horizontalPageScroll **instance method** [17](#)

horizontalRulerView **instance method** [17](#)

horizontalScroller **instance method** [18](#)

## L

---

lineScroll **instance method** [18](#)

## P

---

pageScroll **instance method** [18](#)

## R

---

reflectScrolledClipView: **instance method** [19](#)

rulersVisible **instance method** [20](#)

rulerViewClass **class method** [11](#)

## S

---

scrollsDynamically **instance method** [20](#)

scrollWheel: **instance method** [20](#)

setAutohidesScrollers: **instance method** [21](#)

setBackgroundColor: **instance method** [21](#)

setBorderType: **instance method** [21](#)

setContentView: **instance method** [22](#)

setDocumentCursor: **instance method** [22](#)

setDocumentView: **instance method** [23](#)

setDrawsBackground: **instance method** [23](#)

setHasHorizontalRuler: **instance method** [24](#)

setHasHorizontalScroller: **instance method** [24](#)

setHasVerticalRuler: **instance method** [25](#)

setHasVerticalScroller: instance method [25](#)  
setHorizontalLineScroll: instance method [26](#)  
setHorizontalPageScroll: instance method [26](#)  
setHorizontalRulerView: instance method [26](#)  
setHorizontalScroller: instance method [27](#)  
setLineScroll: instance method [27](#)  
setPageScroll: instance method [28](#)  
setRulersVisible: instance method [28](#)  
setRulerViewClass: class method [11](#)  
setScrollsDynamically: instance method [29](#)  
setVerticalLineScroll: instance method [29](#)  
setVerticalPageScroll: instance method [30](#)  
setVerticalRulerView: instance method [30](#)  
setVerticalScroller: instance method [31](#)

## T

---

tile instance method [31](#)

## V

---

verticalLineScroll instance method [31](#)  
verticalPageScroll instance method [32](#)  
verticalRulerView instance method [32](#)  
verticalScroller instance method [33](#)