
NSCursor Class Reference

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



2009-01-06



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, Carbon, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

NSCursor Class Reference 5

Overview	5
Cursor Rectangles	6
Adopted Protocols	7
Tasks	7
Initializing a New Cursor	7
Setting Cursor Attributes	7
Controlling Which Cursor Is Current	7
Retrieving Cursor Instances	8
Class Methods	9
arrowCursor	9
closedHandCursor	9
crosshairCursor	10
currentCursor	10
disappearingItemCursor	10
hide	11
IBeamCursor	11
openHandCursor	11
pointingHandCursor	12
pop	12
resizeDownCursor	13
resizeLeftCursor	13
resizeLeftRightCursor	13
resizeRightCursor	13
resizeUpCursor	14
resizeUpDownCursor	14
setHiddenUntilMouseMoves:	14
unhide	15
Instance Methods	15
hotSpot	15
image	16
initWithImage:foregroundColorHint:backgroundColorHint:hotSpot:	16
initWithImage:hotSpot:	17
isSetOnMouseEntered	18
isSetOnMouseExited	18
mouseEntered:	18
mouseExited:	19
pop	20
push	20
set	20
setOnMouseEntered:	21

setOnMouseExited: 21
Constants 22
AppKit Versions for NSCursor Bug Fixes 22

Document Revision History 23

Index 25





NSCursor Class Reference










Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Cursor Management
Declared in	NSCursor.h
Related sample code	CarbonCocoa_PictureCursor DragItemAround PDFView Sketch-112 TextLinks

Overview

Instances of the `NSCursor` class manage the appearance of the cursor.

The following table shows and describes the system cursors, and indicates the class method for obtaining them:

Cursor	Description
	The arrow cursor (arrowCursor (page 9))
	The I-beam cursor for indicating insertion points (IBeamCursor (page 11))
	The cross-hair cursor (crosshairCursor (page 10))
	The closed-hand cursor (closedHandCursor (page 9))

Cursor	Description
	The open-hand cursor (openHandCursor (page 11))
	The pointing-hand cursor (pointingHandCursor (page 12))
	The resize-left cursor (resizeLeftCursor (page 13))
	The resize-right cursor (resizeRightCursor (page 13))
	The resize-left-and-right cursor (resizeLeftRightCursor (page 13))
	The resize-up cursor (resizeUpCursor (page 14))
	The resize-down cursor (resizeDownCursor (page 13))
	The resize-up-and-down cursor (resizeUpDownCursor (page 14))
	The disappearing item cursor (disappearingItemCursor (page 10))

In Mac OS X version 10.3 and later, cursor size is no longer limited to 16 by 16 pixels.

Cursor Rectangles

In Cocoa, you can change the currently displayed cursor based on the position of the mouse over one of your views. You might use this technique to provide visual feedback about what actions the user can take with the mouse. For example, you might display one of the resize cursors whenever the mouse moves over a portion of your view that acts as a custom resizing handle. To set this up, you associate a cursor object with one or more cursor rectangles in the view.

Cursor rectangles are a specialized type of tracking rectangles, which are used to monitor the mouse location in a view. Views implement cursor rectangles using tracking rectangles but provide methods for setting and refreshing cursor rectangles that are distinct from the generic tracking rectangle interface. For information on how to set up cursor rectangles, see “Handling Tracking-Rectangle and Cursor-Update Events in Views”.

Adopted Protocols

NSCoding

`initWithCoder:`
`encodeWithCoder:`

Tasks

Initializing a New Cursor

- [initWithImage:hotSpot:](#) (page 17)
Initializes a cursor with the given image and hot spot.
- [initWithImage:foregroundColorHint:backgroundColorHint:hotSpot:](#) (page 16)
Initializes the cursor with the specified image and hot spot.

Setting Cursor Attributes

- [image](#) (page 16)
Returns the receiver's image.
- [hotSpot](#) (page 15)
Returns the position of the cursor's hot spot.
- + [hide](#) (page 11)
Makes the current cursor invisible.
- + [unhide](#) (page 15)
Negates an earlier call to [hide](#) (page 11) by showing the current cursor.
- + [setHiddenUntilMouseMoves:](#) (page 14)
Sets whether the cursor is hidden until the mouse moves.

Controlling Which Cursor Is Current

- + [pop](#) (page 12)
Pops the current cursor off the top of the stack.
- [pop](#) (page 20)
Sends a [pop](#) (page 12) message to the receiver's class.
- [push](#) (page 20)
Puts the receiver on top of the cursor stack and makes it the current cursor.
- [set](#) (page 20)
Makes the receiver the current cursor.
- [mouseEntered:](#) (page 18)
Automatically sent to the receiver when the cursor enters a cursor rectangle owned by the receiver.

- [setOnMouseEntered:](#) (page 21)
Specifies whether the receiver accepts [mouseEntered:](#) (page 18) events.
- [isSetOnMouseEntered](#) (page 18)
Returns a Boolean value indicating whether the receiver becomes current on receiving a [mouseEntered:](#) (page 18) message.
- [mouseExited:](#) (page 19)
Automatically sent to the receiver when the cursor exits a cursor rectangle owned by the receiver.
- [setOnMouseExited:](#) (page 21)
Sets whether the receiver accepts [mouseExited:](#) (page 19) events.
- [isSetOnMouseExited](#) (page 18)
Returns a Boolean value indicating whether the receiver becomes current when it receives a [mouseExited:](#) (page 19) message.

Retrieving Cursor Instances

- + [currentCursor](#) (page 10)
Returns the application's current cursor.
- + [arrowCursor](#) (page 9)
Returns the default cursor, the arrow cursor.
- + [closedHandCursor](#) (page 9)
Returns the closed-hand system cursor.
- + [crosshairCursor](#) (page 10)
Returns the cross-hair system cursor.
- + [disappearingItemCursor](#) (page 10)
Returns a cursor indicating that the current operation will result in a disappearing item.
- + [IBeamCursor](#) (page 11)
Returns a cursor that looks like a capital I with a tiny crossbeam at its middle.
- + [openHandCursor](#) (page 11)
Returns the open-hand system cursor.
- + [pointingHandCursor](#) (page 12)
Returns the pointing-hand system cursor.
- + [resizeDownCursor](#) (page 13)
Returns the resize-down system cursor.
- + [resizeLeftCursor](#) (page 13)
Returns the resize-left system cursor.
- + [resizeLeftRightCursor](#) (page 13)
Returns the resize-left-and-right system cursor.
- + [resizeRightCursor](#) (page 13)
Returns the resize-right system cursor.
- + [resizeUpCursor](#) (page 14)
Returns the resize-up system cursor.
- + [resizeUpDownCursor](#) (page 14)
Returns the resize-up-and-down system cursor.

Class Methods

arrowCursor

Returns the default cursor, the arrow cursor.

```
+ (NSCursor *)arrowCursor
```

Return Value

The default cursor, a slanted arrow with its hot spot at the tip. The arrow cursor is the one you're used to seeing over buttons, scrollers, and many other objects in the window system.

Availability

Available in Mac OS X v10.0 and later.

See Also

- + [IBeamCursor](#) (page 11)
- + [currentCursor](#) (page 10)
- [hotSpot](#) (page 15)

Related Sample Code

CocoaDVDPlayer
PDFKitLinker2
PDFView
Sketch-112
TextLinks

Declared In

NSCursor.h

closedHandCursor

Returns the closed-hand system cursor.

```
+ (NSCursor *)closedHandCursor
```

Return Value

The closed-hand cursor.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

DragItemAround

Declared In

NSCursor.h

crosshairCursor

Returns the cross-hair system cursor.

```
+ (NSCursor *)crosshairCursor
```

Return Value

The cross-hair cursor. This cursor is used for situations when precise location is required (where the lines cross is the hot spot).

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

TrackIt

Declared In

NSCursor.h

currentCursor

Returns the application's current cursor.

```
+ (NSCursor *)currentCursor
```

Return Value

The top cursor on the application's cursor stack. This cursor may not be the visible cursor on the screen if a different application is currently active.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [set](#) (page 20)
- [push](#) (page 20)
- [pop](#) (page 20)
- [mouseEntered:](#) (page 18)
- [mouseExited:](#) (page 19)

Declared In

NSCursor.h

disappearingItemCursor

Returns a cursor indicating that the current operation will result in a disappearing item.

```
+ (NSCursor *)disappearingItemCursor
```

Return Value

The system cursor that indicates that the current operation will result in a disappearing item (for example, when dragging an item from the dock or a toolbar).

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

hide

Makes the current cursor invisible.

```
+ (void)hide
```

Discussion

If another cursor becomes current, that cursor will be invisible, too. It will remain invisible until you invoke the [unhide](#) (page 15) method.

[hide](#) (page 11) overrides [setHiddenUntilMouseMoves:](#) (page 14).

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CarbonCocoa_PictureCursor

Declared In

NSCursor.h

IBeamCursor

Returns a cursor that looks like a capital I with a tiny crossbeam at its middle.

```
+ (NSCursor *)IBeamCursor
```

Return Value

The I-beam cursor. This is the cursor that you're used to seeing over editable or selectable text. The I-beam cursor's default hot spot is where the crossbeam intersects the I.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [arrowCursor](#) (page 9)

+ [currentCursor](#) (page 10)

Declared In

NSCursor.h

openHandCursor

Returns the open-hand system cursor.

```
+ (NSCursor *)openHandCursor
```

Return Value

The open-hand cursor.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

DragItemAround

Declared In

NSCursor.h

pointingHandCursor

Returns the pointing-hand system cursor.

```
+ (NSCursor *)pointingHandCursor
```

Return Value

The pointing-hand cursor. The tip of the pointing finger is the hot spot.

Availability

Available in Mac OS X v10.3 and later.

Related Sample Code

CocoaDVDPlayer

Declared In

NSCursor.h

pop

Pops the current cursor off the top of the stack.

```
+ (void)pop
```

Discussion

The new object on the top of the stack becomes the current cursor. If the current cursor is the only cursor on the stack, this method does nothing.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [push](#) (page 20)

Related Sample Code

DragItemAround

Declared In

NSCursor.h

resizeDownCursor

Returns the resize-down system cursor.

```
+ (NSCursor *)resizeDownCursor
```

Return Value

The resize-down cursor. This cursor is used when moving or resizing an object to indicate that the user can move only in the indicated direction.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

resizeLeftCursor

Returns the resize-left system cursor.

```
+ (NSCursor *)resizeLeftCursor
```

Return Value

The resize-left cursor. This cursor is used when moving or resizing an object to indicate that the user can move only in the indicated direction.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

resizeLeftRightCursor

Returns the resize-left-and-right system cursor.

```
+ (NSCursor *)resizeLeftRightCursor
```

Return Value

The resize-left-and-right cursor. This cursor is used when moving or resizing an object and the object can be moved left or right.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

resizeRightCursor

Returns the resize-right system cursor.

```
+ (NSCursor *)resizeRightCursor
```

Return Value

The resize-right cursor. This cursor is used when moving or resizing an object to indicate that the user can move only in the indicated direction.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

resizeUpCursor

Returns the resize-up system cursor.

```
+ (NSCursor *)resizeUpCursor
```

Return Value

The resize-up cursor. This cursor is used when moving or resizing an object to indicate that the user can move only in the indicated direction.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

resizeUpDownCursor

Returns the resize-up-and-down system cursor.

```
+ (NSCursor *)resizeUpDownCursor
```

Return Value

The resize-up-and-down cursor. This cursor is used when moving or resizing an object and the object can be moved up or down.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSCursor.h

setHiddenUntilMouseMoves:

Sets whether the cursor is hidden until the mouse moves.

```
+ (void)setHiddenUntilMouseMoves:(BOOL)flag
```

Parameters*flag*

YES to hide the cursor until one of the following occurs:

- The mouse moves.
- You invoke the method again, with *flag* set to NO.

DiscussionDo not try to counter this method by invoking [unhide](#) (page 15). The results are undefined.**Availability**

Available in Mac OS X v10.0 and later.

See Also+ [hide](#) (page 11)**Declared In**

NSCursor.h

unhideNegates an earlier call to [hide](#) (page 11) by showing the current cursor.

+ (void)unhide

Availability

Available in Mac OS X v10.0 and later.

See Also+ [setHiddenUntilMouseMoves:](#) (page 14)+ [hide](#) (page 11)**Related Sample Code**

CarbonCocoa_PictureCursor

Declared In

NSCursor.h

Instance Methods

hotSpot

Returns the position of the cursor's hot spot.

- (NSPoint)hotSpot

Return Value

The point describing the position of the hot spot, specified according to the cursor's flipped coordinate system.

Discussion

For a more complete explanation, see the class description.

Note that an `NSCursor` object is immutable: you cannot change its hot spot after it's created. Instead, use [initWithImage:hotSpot:](#) (page 17) to create a new cursor with the new settings.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [initWithImage:hotSpot:](#) (page 17)

Declared In

NSCursor.h

image

Returns the receiver's image.

```
- (NSImage *)image
```

Return Value

The cursor image or `nil` if none exists

Discussion

Note that an `NSCursor` object is immutable: you cannot change its image after it's created. Instead, use [initWithImage:hotSpot:](#) (page 17) to create a new cursor with the new settings.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [initWithImage:hotSpot:](#) (page 17)

Declared In

NSCursor.h

initWithImage:foregroundColorHint:backgroundColorHint:hotSpot:

Initializes the cursor with the specified image and hot spot.

```
- (id)initWithImage:(NSImage *)newImage foregroundColorHint:(NSColor *)fg
    backgroundColorHint:(NSColor *)bg hotSpot:(NSPoint)hotSpot
```

Parameters

newImage

The image to assign to the cursor.

fg

The foreground color. This is currently ignored.

bg

The background color. This is currently ignored.

hotSpot

The point to assign as the cursor's hot spot.

Return Value

The initialized cursor object.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [initWithImage:hotSpot:](#) (page 17)

Declared In

NSCursor.h

initWithImage:hotSpot:

Initializes a cursor with the given image and hot spot.

```
- (id)initWithImage:(NSImage *)newImage hotSpot:(NSPoint)aPoint
```

Parameters

newImage

The image to assign to the cursor.

aPoint

The point to set as the cursor's hot spot.

Return Value

An initialized cursor object.

Discussion

This method is the designated initializer for the class.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [hotSpot](#) (page 15)

- [image](#) (page 16)

- [initWithImage:foregroundColorHint:backgroundColorHint:hotSpot:](#) (page 16)

Related Sample Code

PDFView

Sketch-112

TextLinks

Declared In

NSCursor.h

isSetOnMouseEntered

Returns a Boolean value indicating whether the receiver becomes current on receiving a [mouseEntered:](#) (page 18) message.

- (BOOL)isSetOnMouseEntered

Return Value

YES if the receiver will become current when it receives a [mouseEntered:](#) (page 18) message; otherwise, NO.

Discussion

To receive such a message, the receiver must first be assigned a cursor rectangle. This assignment can be made using the `NSView` method `addCursorRect:cursor:`. For a more complete explanation, see the class description.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setOnMouseEntered:](#) (page 21)
- [isSetOnMouseExited](#) (page 18)

Declared In

NSCursor.h

isSetOnMouseExited

Returns a Boolean value indicating whether the receiver becomes current when it receives a [mouseExited:](#) (page 19) message.

- (BOOL)isSetOnMouseExited

Return Value

YES if the receiver becomes current when it receives a [mouseExited:](#) (page 19) message; otherwise, NO.

Discussion

To receive such a message, the receiver must first be assigned a cursor rectangle. This assignment can be made using the `NSView` method `addCursorRect:cursor:`. For a more complete explanation, see the class description.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

NSCursor.h

mouseEntered:

Automatically sent to the receiver when the cursor enters a cursor rectangle owned by the receiver.

- (void)mouseEntered:(NSEvent *)*anEvent*

Parameters

anEvent

The event generated when the cursor enters the cursor rectangle.

Discussion

If used after [setOnMouseEntered:](#) (page 21) has been called with an argument of YES, `mouseEntered:` can make the receiver the current cursor.

In your programs, you won't invoke `mouseEntered:` explicitly. It's only included in the class interface so you can override it.

For a more complete explanation, see "Handling Tracking-Rectangle and Cursor-Update Events in Views" and the `NSView` method `addTrackingRect:owner:userData:assumeInside:`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isSetOnMouseEntered](#) (page 18)
- [mouseExited:](#) (page 19)

Declared In

`NSCursor.h`

mouseExited:

Automatically sent to the receiver when the cursor exits a cursor rectangle owned by the receiver.

- (void)mouseExited:(NSEvent *)*anEvent*

Parameters

anEvent

The event generated when the cursor exits the cursor rectangle.

Discussion

Like [mouseEntered:](#) (page 18), this message is part of the class interface only so you can override it.

For a more complete explanation, see "Handling Tracking-Rectangle and Cursor-Update Events in Views" and the `NSView` method `addTrackingRect:owner:userData:assumeInside:`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setOnMouseExited:](#) (page 21)
- [isSetOnMouseExited](#) (page 18)

Declared In

`NSCursor.h`

pop

Sends a [pop](#) (page 12) message to the receiver's class.

- (void)pop

Availability

Available in Mac OS X v10.0 and later.

See Also

- [push](#) (page 20)
- [pop](#) (page 20)

Declared In

NSCursor.h

push

Puts the receiver on top of the cursor stack and makes it the current cursor.

- (void)push

Availability

Available in Mac OS X v10.0 and later.

See Also

- [pop](#) (page 20)
- [pop](#) (page 20)

Related Sample Code

DragItemAround

Declared In

NSCursor.h

set

Makes the receiver the current cursor.

- (void)set

Availability

Available in Mac OS X v10.0 and later.

See Also

- + [currentCursor](#) (page 10)

Related Sample Code

CocoaDVDPlayer

PDFKitLinker2

TrackIt

Declared In
NSCursor.h

setOnMouseEntered:

Specifies whether the receiver accepts [mouseEntered:](#) (page 18) events.

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

Parameters

flag

YES if the receiver accepts future [mouseEntered:](#) (page 18) event messages; otherwise it ignores them.

Discussion

Accepting [mouseEntered:](#) (page 18) event messages allows the cursor to be made the current cursor when the cursor enters a view's cursor rectangle.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [mouseEntered:](#) (page 18)

Declared In
NSCursor.h

setOnMouseExited:

Sets whether the receiver accepts [mouseExited:](#) (page 19) events.

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

Parameters

flag

YES if the receiver accepts future [mouseExited:](#) (page 19) event messages; otherwise it ignores them.

Discussion

Accepting [mouseExited:](#) (page 19) event messages allows the cursor to be made the current cursor when the cursor exits a view's cursor rectangle.

Availability

Available in Mac OS X v10.0 and later.

Declared In
NSCursor.h

Constants

AppKit Versions for NSCursor Bug Fixes

The version of the AppKit framework containing a specific bug fix.

```
#define NSAppKitVersionNumberWithCursorSizeSupport 682.0
```

Constants

`NSAppKitVersionNumberWithCursorSizeSupport`

The specific version of the AppKit framework that introduced support for cursors larger than 16 x 16 pixels in size. Developers should not need to use this constant unless they are writing applications for Mac OS X v10.2 and earlier.

Available in Mac OS X v10.3 and later.

Declared in `NSCursor.h`.

Document Revision History

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

Date	Notes
2009-01-06	Added a definition for the <code>NSAppKitVersionNumberWithCursorSizeSupport</code> macro.
2008-02-08	Clarified the function of the <code>currentCursor</code> method.
2007-01-19	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

AppKit Versions for NSCursor Bug Fixes [22](#)
arrowCursor class method [9](#)

C

closedHandCursor class method [9](#)
crosshairCursor class method [10](#)
currentCursor class method [10](#)

D

disappearingItemCursor class method [10](#)

H

hide class method [11](#)
hotSpot instance method [15](#)

I

IBeamCursor class method [11](#)
image instance method [16](#)
initWithImage:foregroundColorHint:
 backgroundColorHint:hotSpot: instance method
 [16](#)
initWithImage:hotSpot: instance method [17](#)
isSetOnMouseEntered instance method [18](#)
isSetOnMouseExited instance method [18](#)

M

mouseEntered: instance method [18](#)
mouseExited: instance method [19](#)

N

NSAppKitVersionNumberWithCursorSizeSupport
 constant [22](#)

O

openHandCursor class method [11](#)

P

pointingHandCursor class method [12](#)
pop class method [12](#)
pop instance method [20](#)
push instance method [20](#)

R

resizeDownCursor class method [13](#)
resizeLeftCursor class method [13](#)
resizeLeftRightCursor class method [13](#)
resizeRightCursor class method [13](#)
resizeUpCursor class method [14](#)
resizeUpDownCursor class method [14](#)

S

set instance method [20](#)
setHiddenUntilMouseMoves: class method [14](#)

setOnMouseEntered: [instance method 21](#)
setOnMouseExited: [instance method 21](#)

U

unhide [class method 15](#)