
NSMenu Class Reference

User Experience: Menus



2009-06-24



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

.Mac is a registered service mark of Apple Inc.

Apple, the Apple logo, 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

NSMenu Class Reference 7

Overview 7

Tasks 7

Managing the Menu Bar 7

Creating an NSMenu Object 8

Adding and Removing Menu Items 8

Finding Menu Items 8

Finding Indices of Menu Items 8

Managing Submenus 9

Enabling and Disabling Menu Items 9

Getting and Setting the Menu Font 10

Handling Keyboard Equivalents 10

Simulating Mouse Clicks 10

Managing the Title 10

Configuring Menu Size 10

Getting Menu Properties 10

Managing Menu Change Notifications 11

Displaying Contextual Menus 11

Displaying Context-Sensitive Help 11

Managing Display of the State Column 11

Controlling Allocation Zones 11

Handling Highlighting 12

Managing the Delegate 12

Handling Tracking 12

Deprecated Methods 12

Class Methods 12

menuBarVisible 12

menuZone 13

popUpContextMenu:withEvent:forView: 13

popUpContextMenu:withEvent:forView:withFont: 14

setMenuBarVisible: 14

Instance Methods 15

addItem: 15

addItemWithTitle:action:keyEquivalent: 15

allowsContextMenuPlugIns 16

autoenablesItems 17

cancelTracking 17

cancelTrackingWithoutAnimation 17

delegate 18

font 18

helpRequested: 18

highlightedItem	19
indexOfItem:	19
indexOfItemWithRepresentedObject:	20
indexOfItemWithSubmenu:	20
indexOfItemWithTag:	20
indexOfItemWithTarget:andAction:	21
indexOfItemWithTitle:	21
initWithTitle:	22
insertItem:atIndex:	23
insertItemWithTitle:action:keyEquivalent:atIndex:	23
isTornOff	24
itemArray	24
itemAtIndex:	25
itemChanged:	25
itemWithTag:	26
itemWithTitle:	26
menuBarHeight	27
menuChangedMessagesEnabled	27
minimumWidth	28
numberOfItems	28
performActionForItemAtIndex:	28
performKeyEquivalent:	29
popupMenuPositioningItem:atLocation:inView:	29
propertiesToUpdate	30
removeAllItems	31
removeItem:	31
removeItemAtIndex:	32
setAllowsContextMenuPlugins:	32
setAutoenablesItems:	33
setDelegate:	33
setFont:	33
setMenuChangedMessagesEnabled:	34
setMinimumWidth:	34
setShowsStateColumn:	35
setSubmenu:forItem:	35
setSupermenu:	36
setTitle:	36
showsStateColumn	36
size	37
submenuAction:	37
supermenu	37
title	38
update	38
Constants	38
NSMenuProperties	38
Notifications	39

- NSMenuDidAddItemNotification 39
- NSMenuDidChangeItemNotification 40
- NSMenuDidBeginTrackingNotification 40
- NSMenuDidEndTrackingNotification 40
- NSMenuDidRemoveItemNotification 41
- NSMenuDidSendActionNotification 41
- NSMenuWillSendActionNotification 41

Appendix A Deprecated NSMenu Methods 43

- Deprecated in Mac OS X v10.2 43
 - setMenuZone: 43
 - attachedMenu 43
 - contextMenuRepresentation 44
 - isAttached 44
 - locationForSubmenu: 44
 - menuRepresentation 45
 - setContextMenuRepresentation: 45
 - setMenuRepresentation: 45
 - setTearOffMenuRepresentation: 45
 - sizeToFit 46
 - tearOffMenuRepresentation 46

Document Revision History 47

Index 49

NSMenu Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Application Menu and Pop-up List Programming Topics for Cocoa
Declared in	NSMenu.h
Related sample code	GLUT MenuItemView MenuMadness PDFKitLinker2 QTAudioContextInsert

Overview

The `NSMenu` class defines an object that manages an application's menus.

Tasks

Managing the Menu Bar

- + `menuBarVisible` (page 12)
Returns a Boolean value that indicates whether the menu bar is visible.
- + `setMenuBarVisible:` (page 14)
Sets whether the menu bar is visible and selectable by the user.
- `menuBarHeight` (page 27)
Returns the menu bar height for the current application's main menu.

Creating an NSMenu Object

- `initWithTitle:` (page 22)
Initializes and returns a menu having the specified title and with autoenabling of menu items turned on.

Adding and Removing Menu Items

- `insertItemAtIndex:` (page 23)
Inserts a menu item into the receiver at a specific location.
- `insertItemWithTitle:action:keyEquivalent:atIndex:` (page 23)
Creates and adds a menu item at a specified location in the receiver.
- `addItem:` (page 15)
Adds a menu item to the end of the receiver.
- `addItemWithTitle:action:keyEquivalent:` (page 15)
Creates a new menu item and adds it to the end of the receiver.
- `removeItem:` (page 31)
Removes a menu item from the receiver.
- `removeItemAtIndex:` (page 32)
Removes the menu item at a specified location in the receiver.
- `itemChanged:` (page 25)
Invoked when a menu item is modified visually (for example, its title changes).
- `removeAllItems` (page 31)
Removes all the menu items in the receiver.

Finding Menu Items

- `itemWithTag:` (page 26)
Returns the first menu item in the receiver with the specified tag.
- `itemWithTitle:` (page 26)
Returns the first menu item in the receiver with a specified title.
- `itemAtIndex:` (page 25)
Returns the menu item at a specific location of the receiver.
- `numberOfItems` (page 28)
Returns the number of menu items in the receiver, including separator items.
- `itemArray` (page 24)
Returns an array containing the receiver's menu items.

Finding Indices of Menu Items

- `indexOfItem:` (page 19)
Returns the index identifying the location of a specified menu item in the receiver.

- [indexOfItemWithTitle:](#) (page 21)
Returns the index of the first menu item in the receiver that has a specified title.
- [indexOfItemWithTag:](#) (page 20)
Returns the index of the first menu item in the receiver identified by a tag.
- [indexOfItemWithTarget:andAction:](#) (page 21)
Returns the index of the first menu item in the receiver that has a specified action and target.
- [indexOfItemWithRepresentedObject:](#) (page 20)
Returns the index of the first menu item in the receiver that has a given represented object.
- [indexOfItemWithSubmenu:](#) (page 20)
Returns the index of the menu item in the receiver with the given submenu.

Managing Submenus

- [setSubmenu:forItem:](#) (page 35)
Assigns a menu to be a submenu of the receiver controlled by a given menu item.
- [submenuAction:](#) (page 37)
The action method assigned to menu items that open submenus.
- [supermenu](#) (page 37)
Returns the receiver's supermenu.
- [setSupermenu:](#) (page 36)
Sets the receiver's supermenu.
- [isTornOff](#) (page 24)
Returns a Boolean value that indicates whether the receiver is offscreen or attached to another menu (or if it's the main menu).
- [attachedMenu](#) (page 43) **Deprecated in Mac OS X v10.2**
Returns the menu currently attached to the receiver.
- [isAttached](#) (page 44) **Deprecated in Mac OS X v10.2**
Returns a Boolean value that indicates whether the receiver is currently attached to another menu.
- [locationForSubmenu:](#) (page 44) **Deprecated in Mac OS X v10.2**
Returns the location in screen coordinates where the given submenu is displayed when opened as a submenu of the receiver.

Enabling and Disabling Menu Items

- [autoenablesItems](#) (page 17)
Returns a Boolean value that indicates whether the receiver automatically enables and disables its menu items.
- [setAutoenablesItems:](#) (page 33)
Controls whether the receiver automatically enables and disables its menu items based on delegates implementing the `NSMenuValidation` informal protocol.
- [update](#) (page 38)
Enables or disables the receiver's menu items based on the `NSMenuValidation` informal protocol and sizes the menu to fit its current menu items if necessary.

Getting and Setting the Menu Font

- [font](#) (page 18)
Returns the font used to display the menu and its submenus.
- [setFont:](#) (page 33)
Sets the font used to display the menu and its submenus.

Handling Keyboard Equivalents

- [performKeyEquivalent:](#) (page 29)
Performs the action for the menu item that corresponds to the given key equivalent.

Simulating Mouse Clicks

- [performActionForItemAtIndex:](#) (page 28)
Causes the application to send the action message of a specified menu item to its target.

Managing the Title

- [setTitle:](#) (page 36)
Sets the receiver's title.
- [title](#) (page 38)
Returns the receiver's title.

Configuring Menu Size

- [minimumWidth](#) (page 28)
Returns the minimum width of the menu.
- [setMinimumWidth:](#) (page 34)
Set the minimum width of the menu.
- [size](#) (page 37)
Returns the size of the menu.
- [sizeToFit](#) (page 46) **Deprecated in Mac OS X v10.2**
Resizes the receiver to exactly fit its items.

Getting Menu Properties

- [propertiesToUpdate](#) (page 30)
Returns the available properties for the menu.

Managing Menu Change Notifications

- [menuChangedMessagesEnabled](#) (page 27)
Returns a Boolean value that indicates whether messages are sent to the application's windows upon each change to the receiver.
- [setMenuChangedMessagesEnabled:](#) (page 34)
Controls whether the receiver sends messages to the application's windows upon each menu change.

Displaying Contextual Menus

- [allowsContextMenuPlugIns](#) (page 16)
Returns whether the popup menu allows appending of contextual menu plugin items.
- [setAllowsContextMenuPlugIns:](#) (page 32)
Sets whether the popup menu allows appending of contextual menu plugin items.

Displaying Context-Sensitive Help

- + [popupContextMenu:withEvent:forView:](#) (page 13)
Displays a contextual menu over a view for an event.
- + [popupContextMenu:withEvent:forView:withFont:](#) (page 14)
Displays a contextual menu over a view for an event using a specified font.
- [helpRequested:](#) (page 18)
Overridden by subclasses to implement specialized context-sensitive help behavior.
- [popupMenuItem:atLocation:inView:](#) (page 29)
Pops up the menu at the specified location.

Managing Display of the State Column

- [setShowsStateColumn:](#) (page 35)
Sets whether the receiver displays the state column.
- [showsStateColumn](#) (page 36)
Returns a Boolean value that indicates whether the receiver displays the state column.

Controlling Allocation Zones

- + [menuZone](#) (page 13)
Returns the zone from which NSMenu objects should be allocated.
- + [setMenuZone:](#) (page 43) **Deprecated in Mac OS X v10.2**
Sets the zone from which NSMenu objects should be allocated

Handling Highlighting

- [highlightedItem](#) (page 19)
Returns the highlighted item in the receiver.

Managing the Delegate

- [setDelegate:](#) (page 33)
Sets the receiver's delegate.
- [delegate](#) (page 18)
Returns the receiver's delegate.

Handling Tracking

- [cancelTracking](#) (page 17)
Dismisses the menu and ends all menu tracking.
- [cancelTrackingWithoutAnimation](#) (page 17)
Dismisses the menu and ends all menu tracking without displaying the associated animation.

Deprecated Methods

- [contextMenuRepresentation](#) (page 44) **Deprecated in Mac OS X v10.2**
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [menuRepresentation](#) (page 45) **Deprecated in Mac OS X v10.2**
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [setContextMenuRepresentation:](#) (page 45) **Deprecated in Mac OS X v10.2**
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [setMenuRepresentation:](#) (page 45) **Deprecated in Mac OS X v10.2**
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [setTearOffMenuRepresentation:](#) (page 45) **Deprecated in Mac OS X v10.2**
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [tearOffMenuRepresentation](#) (page 46) **Deprecated in Mac OS X v10.2**
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)

Class Methods

menuBarVisible

Returns a Boolean value that indicates whether the menu bar is visible.

+ (BOOL)menuBarVisible

Return Value

YES if the menu bar is visible, otherwise NO.

Availability

Available in Mac OS X v10.2 and later.

See Also

+ [setMenuBarVisible:](#) (page 14)

Declared In

NSMenu.h

menuZone

Returns the zone from which NSMenu objects should be allocated.

```
+ (NSZone *)menuZone
```

Return Value

The zone from which NSMenu objects should be allocated.

Discussion

The zone is created if necessary.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

DesktopImage

GLUT

MenuItemView

MenuMadness

ObjectPath

Declared In

NSMenu.h

popUpContextMenu:withEvent:forView:

Displays a contextual menu over a view for an event.

```
+ (void)popUpContextMenu:(NSMenu *)menu withEvent:(NSEvent *)event forView:(NSView *)view
```

Parameters

menu

The menu object to use for the contextual menu.

event

An NSEvent object representing the event.

view

The view object over which to display the contextual menu.

Availability

Available in Mac OS X v10.0 and later.

See Also

+ [popUpContextMenu:withEvent:forView:withFont:](#) (page 14)

Related Sample Code

GLUT

Declared In

NSMenu.h

popUpContextMenu:withEvent:forView:withFont:

Displays a contextual menu over a view for an event using a specified font.

```
+ (void)popUpContextMenu:(NSMenu *)menu withEvent:(NSEvent *)event forView:(NSView *)view withFont:(NSFont *)font
```

Parameters

menu

The menu object to use for the contextual menu.

event

An `NSEvent` object representing the event.

view

The view object over which to display the contextual menu.

font

An `NSFont` object representing the font for the contextual menu. If you pass in `nil` for the font, the method uses the default font for *menu*.

Discussion

Specifying a font using the font parameter is discouraged. Instead set the menu's font using the [setFont:](#) (page 33) method and pass `nil` for the *font* parameter.

Availability

Available in Mac OS X v10.3 and later.

See Also

+ [popUpContextMenu:withEvent:forView:](#) (page 13)

- [popUpMenuItem:atLocation:inView:](#) (page 29)

Declared In

NSMenu.h

setMenuBarVisible:

Sets whether the menu bar is visible and selectable by the user.

```
+ (void)setMenuBarVisible:(BOOL)visible
```

Parameters*visible*

YES if menu bar is to be visible, otherwise NO.

Availability

Available in Mac OS X v10.2 and later.

See Also+ [menuBarVisible](#) (page 12)**Declared In**

NSMenu.h

Instance Methods

addItem:

Adds a menu item to the end of the receiver.

- (void)addItem:(NSMenuItem *)*newItem***Parameters***newItem*The menu item (an object conforming to the `NSMenuItem` protocol) to add to the menu.**Discussion**This method invokes [insertItemAtIndex:](#) (page 23). Thus, the receiver does not accept the menu item if it already belongs to another menu. After adding the menu item, the receiver updates itself.**Availability**

Available in Mac OS X v10.0 and later.

See Also

- [addItemWithTitle:action:keyEquivalent:](#) (page 15)
- [removeItem:](#) (page 31)
- [removeItemAtIndex:](#) (page 32)

Related Sample Code

MenuItemView

MenuMadness

PDFKitLinker2

QTAudioContextInsert

QTAudioExtractionPanel

Declared In

NSMenu.h

addItemWithTitle:action:keyEquivalent:

Creates a new menu item and adds it to the end of the receiver.

```
- (NSMenuItem *)addItemWithTitle:(NSString *)aString action:(SEL)aSelector
    keyEquivalent:(NSString *)keyEquiv
```

Parameters*aString*

A string to be made the title of the menu item.

aSelector

The action-message selector to assign to the menu item.

*keyEquiv*A string identifying the key to use as a key equivalent for the menu item. If you do not want the menu item to have a key equivalent, *keyEquiv* should be an empty string (@" ") and not *nil*.**Return Value**The created menu item (an object conforming to the *NSMenuItem* protocol) or *nil* if the object couldn't be created.**Availability**

Available in Mac OS X v10.0 and later.

See Also

- [addItem:](#) (page 15)
- [removeItem:](#) (page 31)
- [removeItemAtIndex:](#) (page 32)

Related Sample Code

Clock Control

ClockControl

UIElementInspector

Declared In

NSMenu.h

allowsContextMenuPlugins

Returns whether the popup menu allows appending of contextual menu plugin items.

```
- (BOOL)allowsContextMenuPlugins
```

Return Value

YES if the popup menu allows appending of contextual menu plugin items, otherwise NO.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [setAllowsContextMenuPlugins:](#) (page 32)

Declared In

NSMenu.h

autoenablesItems

Returns a Boolean value that indicates whether the receiver automatically enables and disables its menu items.

- (BOOL)autoenablesItems

Return Value

YES if the receiver automatically enables and disables its menu items (based on the `NSMenuValidation` informal protocol), otherwise NO.

Discussion

By default, `NSMenu` objects autoenable their menu items. See the protocol specification for more information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAutoenablesItems:](#) (page 33)

Declared In

`NSMenu.h`

cancelTracking

Dismisses the menu and ends all menu tracking.

- (void)cancelTracking

Availability

Available in Mac OS X v10.5 and later.

See Also

- [cancelTrackingWithoutAnimation](#) (page 17)

Related Sample Code

`MenuItemView`

Declared In

`NSMenu.h`

cancelTrackingWithoutAnimation

Dismisses the menu and ends all menu tracking without displaying the associated animation.

- (void)cancelTrackingWithoutAnimation

Availability

Available in Mac OS X v10.6 and later.

See Also

- [cancelTracking](#) (page 17)

Declared In

NSMenu.h

delegate

Returns the receiver's delegate.

- (id < NSMenuDelegate >)delegate

Return Value

The receiver's delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also- [setDelegate:](#) (page 33)**Declared In**

NSMenu.h

font

Returns the font used to display the menu and its submenus.

- (NSFont *)font

Return Value

The font object used for displaying the menu.

Availability

Available in Mac OS X v10.6 and later.

See Also- [setFont:](#) (page 33)**Declared In**

NSMenu.h

helpRequested:

Overridden by subclasses to implement specialized context-sensitive help behavior.

- (void)helpRequested:(NSEvent *)event

Parameters*event*An `NSEvent` object representing the event associated with the help request.**Discussion**Subclasses in their implementation of this method should cause the Help Manager (`NSHelpManager`) to display the help associated with the receiver. Never invoke this method directly.

Special Considerations

On Mac OS X v10.6 and later this method has no effect. This method may be deprecated in a future release.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [showContextHelpForObject:locationHint:](#) (NSHelpManager)

Declared In

NSMenu.h

highlightedItem

Returns the highlighted item in the receiver.

- (NSMenuItem *)highlightedItem

Return Value

Returns the highlighted item in the receiver, or `nil` if no item in the menu is highlighted.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSMenu.h

indexOfItem:

Returns the index identifying the location of a specified menu item in the receiver.

- (NSInteger)indexOfItem:(NSMenuItem *)anObject

Parameters

anObject

A menu item—that is an object conforming to the `NSMenuItem` protocol.

Return Value

The integer index of the menu item or, if no such menu item is in the menu, `-1`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [insertItem:atIndex:](#) (page 23)

- [itemAtIndex:](#) (page 25)

Declared In

NSMenu.h

indexOfItemWithRepresentedObject:

Returns the index of the first menu item in the receiver that has a given represented object.

```
- (NSInteger)indexOfItemWithRepresentedObject:(id)anObject
```

Parameters

anObject

A represented object of the receiver.

Return Value

The integer index of the menu item or, if no such menu item is in the menu, -1.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItemWithTag:](#) (page 20)
- [indexOfItemWithTitle:](#) (page 21)
- [insertItem:atIndex:](#) (page 23)
- [itemAtIndex:](#) (page 25)

Declared In

NSMenu.h

indexOfItemWithSubmenu:

Returns the index of the menu item in the receiver with the given submenu.

```
- (NSInteger)indexOfItemWithSubmenu:(NSMenu *)anObject
```

Parameters

anObject

A menu object that is a menu item of the receiver (that is, a submenu).

Return Value

The integer index of the menu item or, if no such menu item is in the menu, -1.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [insertItem:atIndex:](#) (page 23)
- [itemAtIndex:](#) (page 25)

Declared In

NSMenu.h

indexOfItemWithTag:

Returns the index of the first menu item in the receiver identified by a tag.

```
- (NSInteger)indexOfItemWithTag:(NSInteger)aTag
```

Parameters*aTag*

An integer tag associated with the menu item of the receiver.

Return Value

The integer index of the menu item or, if no such menu item is in the menu, -1.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [itemWithTag:](#) (page 26)
- [insertItem:atIndex:](#) (page 23)
- [itemAtIndex:](#) (page 25)

Declared In

NSMenu.h

indexOfItemWithTarget:andAction:

Returns the index of the first menu item in the receiver that has a specified action and target.

```
- (NSInteger)indexOfItemWithTarget:(id)anObject andAction:(SEL)actionSelector
```

Parameters*anObject*

An object that is set as the target of a menu item of the receiver.

actionSelector

A selector identifying an action method. If *actionSelector* is NULL, the first menu item in the receiver that has target *anObject* is returned

Return Value

The integer index of the menu item or, if no such menu item is in the menu, -1.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItemWithTag:](#) (page 20)
- [indexOfItemWithTitle:](#) (page 21)
- [indexOfItemWithRepresentedObject:](#) (page 20)
- [insertItem:atIndex:](#) (page 23)
- [itemAtIndex:](#) (page 25)

Declared In

NSMenu.h

indexOfItemWithTitle:

Returns the index of the first menu item in the receiver that has a specified title.

```
- (NSInteger)indexOfItemWithTitle:(NSString *)aTitle
```

Parameters*aTitle*

The title of a menu item in the receiver.

Return Value

The integer index of the menu item or, if no such menu item is in the menu, -1.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [itemWithTitle:](#) (page 26)
- [insertItem:atIndex:](#) (page 23)
- [itemAtIndex:](#) (page 25)

Declared In

NSMenu.h

initWithTitle:

Initializes and returns a menu having the specified title and with autoenabling of menu items turned on.

- (id)initWithTitle:(NSString *)*aTitle***Parameters***aTitle*

The title to assign to the receiver.

Return ValueThe initialized NSMenu object or *nil* if the object could not be initialized.**Special Considerations**

This method is the designated initializer for the class.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAutoenablesItems:](#) (page 33)

Related Sample Code

DeskPictAppDockMenu

GLUT

MenuItemView

MenuMadness

SearchField

Declared In

NSMenu.h

insertItem:atIndex:

Inserts a menu item into the receiver at a specific location.

```
- (void)insertItem:(NSMenuItem *)newItem atIndex:(NSInteger)index
```

Parameters

newItem

An object conforming to the `NSMenuItem` protocol that represents a menu item.

index

An integer index identifying the location of the menu item in the menu.

Discussion

This method posts an `NSMenuDidAddItemNotification` (page 39), allowing interested observers to update as appropriate. This method is a primitive method. All item-addition methods end up calling this method, so this is where you should implement custom behavior on adding new items to a menu in a custom subclass. If the menu item already exists in another menu, it is not inserted and the method raises an exception of type `NSInternalInconsistencyException`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addItem:](#) (page 15)
- [itemAtIndex:](#) (page 25)
- [removeItem:](#) (page 31)

Related Sample Code

ButtonMadness
SearchField

Declared In

`NSMenu.h`

insertItemWithTitle:action:keyEquivalent:atIndex:

Creates and adds a menu item at a specified location in the receiver.

```
- (NSMenuItem *)insertItemWithTitle:(NSString *)aString action:(SEL)aSelector  
keyEquivalent:(NSString *)keyEquiv atIndex:(NSInteger)index
```

Parameters

aString

A string to be made the title of the menu item.

aSelector

The action-message selector to assign to the menu item.

keyEquiv

A string identifying the key to use as a key equivalent for the menu item. If you do not want the menu item to have a key equivalent, *keyEquiv* should be an empty string (@" ") and not `nil`.

index

An integer index identifying the location of the menu item in the menu.

Return Value

The new menu item (an object conforming to the `NSMenuItem` protocol) or `nil` if the item could not be created

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

ButtonMadness

Declared In

`NSMenu.h`

isTornOff

Returns a Boolean value that indicates whether the receiver is offscreen or attached to another menu (or if it's the main menu).

- (BOOL)isTornOff

Return Value

NO if the receiver is offscreen or attached to another menu (or if it's the main menu), otherwise YES.

Special Considerations

On Mac OS X v10.6 and later this method has no effect.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSMenu.h`

itemArray

Returns an array containing the receiver's menu items.

- (NSArray *)itemArray

Return Value

An array containing the receiver's menu items.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [initWithTag:](#) (page 26)
- [initWithTitle:](#) (page 26)
- [itemAtIndex:](#) (page 25)
- [numberOfItems](#) (page 28)

Related Sample Code

EnhancedAudioBurn

WhackedTV

Declared In

NSMenu.h

itemAtIndex:

Returns the menu item at a specific location of the receiver.

- (NSMenuItem *)itemAtIndex:(NSInteger) *index***Parameters***index*

An integer index locating a menu item in a menu.

Return ValueThe found menu item (an object conforming to the `NSMenuItem` protocol) or `nil` if the object couldn't be found.**Discussion**This method raises an exception if *index* is out of bounds.**Availability**

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItem:](#) (page 19)
- [itemWithTag:](#) (page 26)
- [itemWithTitle:](#) (page 26)
- [itemArray](#) (page 24)

Related Sample Code

DeskPictAppDockMenu
 DragNDropOutlineView
 MenuItemView
 OutputBinsPDE
 QTCoreVideo101

Declared In

NSMenu.h

itemChanged:

Invoked when a menu item is modified visually (for example, its title changes).

- (void)itemChanged:(NSMenuItem *) *anObject***Parameters***anObject*

The menu item that has visually changed.

Discussion

This method is not called for changes involving the menu item's action, target, represented object, or tag. Posts an [NSMenuDidChangeItemNotification](#) (page 40).

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

itemWithTag:

Returns the first menu item in the receiver with the specified tag.

```
- (NSMenuItem *)itemWithTag:(NSInteger)aTag
```

Parameters

aTag

A numeric tag associated with a menu item.

Return Value

The found menu item (an object conforming to the `NSMenuItem` protocol) or `nil` if the object couldn't be found.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItemWithTag:](#) (page 20)
- [itemWithTitle:](#) (page 26)
- [itemAtIndex:](#) (page 25)
- [itemArray](#) (page 24)

Declared In

NSMenu.h

itemWithTitle:

Returns the first menu item in the receiver with a specified title.

```
- (NSMenuItem *)itemWithTitle:(NSString *)aString
```

Parameters

aString

The title of a menu item.

Return Value

The found menu item (an object conforming to the `NSMenuItem` protocol) or `nil` if the object couldn't be found.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItemWithTitle:](#) (page 21)
- [itemWithTag:](#) (page 26)
- [itemAtIndex:](#) (page 25)
- [itemArray](#) (page 24)

Declared In

NSMenu.h

menuBarHeight

Returns the menu bar height for the current application's main menu.

- (CGFloat)menuBarHeight

Return Value

The receiver's main menu bar height or 0.0 if the receiver is some other menu.

Discussion

This method supersedes the `menuBarHeight` class method of the `NSMenuView` class.

Availability

Available in Mac OS X v10.4 and later.

Declared In

NSMenu.h

menuChangedMessagesEnabled

Returns a Boolean value that indicates whether messages are sent to the application's windows upon each change to the receiver.

- (BOOL)menuChangedMessagesEnabled

Return Value

YES if messages are sent to the application's windows upon each change to the receiver, otherwise NO.

Special Considerations

On Mac OS X v10.6 and later this method has no effect. This method may be deprecated in a future release.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setMenuChangedMessagesEnabled:](#) (page 34)

Declared In

NSMenu.h

minimumWidth

Returns the minimum width of the menu.

- (CGFloat)minimumWidth

Return Value

The minimum width of the menu in screen coordinates.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSMenu.h

numberOfItems

Returns the number of menu items in the receiver, including separator items.

- (NSInteger)numberOfItems

Return Value

The number of menu items in the receiver, including separator items.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [itemArray](#) (page 24)

Related Sample Code

DeskPictAppDockMenu

QTCoreVideo101

Declared In

NSMenu.h

performActionForItemAtIndex:

Causes the application to send the action message of a specified menu item to its target.

- (void)performActionForItemAtIndex:(NSInteger) *index*

Parameters

index

The integer index of a menu item.

Discussion

If a target is not specified, the message is sent to the first responder. As a side effect, this method posts [NSMenuWillSendActionNotification](#) (page 41) and [NSMenuDidSendActionNotification](#) (page 41).

In Mac OS X v10.6 and later the `performActionForItemAtIndex:` no longer triggers menu validation. This is because validation is typically done during menu tracking or key equivalent matching, so the subsequent `performActionForItemAtIndex:` validation was redundant. To trigger validation explicitly, use invoke the [update](#) (page 38) method.

In Mac OS X v10.6 `performActionForItemAtIndex:`, when called, now triggers highlighting in the menu bar. It also sends out appropriate accessibility notifications indicating the item was selected.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

NSMenu.h

performKeyEquivalent:

Performs the action for the menu item that corresponds to the given key equivalent.

```
- (BOOL)performKeyEquivalent:(NSEvent *)theEvent
```

Parameters

theEvent

An `NSEvent` object that represents a key-equivalent event.

Return Value

YES if *theEvent* is a key equivalent that the receiver handled, NO if it is not a key equivalent that it should handle.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [performActionForItemAtIndex:](#) (page 28)
- `menuHasKeyEquivalent:forEvent:target:action:` (`NSMenuDelegate`)

Declared In

NSMenu.h

popUpMenuPositioningItem:atLocation:inView:

Pops up the menu at the specified location.

```
- (BOOL)popUpMenuPositioningItem:(NSMenuItem *)item atLocation:(NSPoint)location  
inView:(NSView *)view
```

Parameters

item

The menu item to be positioned at the specified location in the view.

location

The location in the *view* coordinate system to display the menu item.

view

The view to display the menu item over.

Return Value

YES if menu tracking ended because an item was selected, and NO if menu tracking was cancelled for any reason.

Discussion

Pops up the receiver as a popup menu. The top left corner of the specified item (if specified, *item* must be present in the receiver) is positioned at the specified location in the specified view, interpreted in the view's own coordinate system.

If *item* is *nil*, the menu is positioned such that the top left of the menu content frame is at the given location.

If *view* is *nil*, the location is interpreted in the screen coordinate system. This allows you to pop up a menu disconnected from any window.

Availability

Available in Mac OS X v10.6 and later.

See Also

+ [popupContextMenu:withEvent:forView:](#) (page 13)

+ [popupContextMenu:withEvent:forView:withFont:](#) (page 14)

Declared In

NSMenu.h

propertiesToUpdate

Returns the available properties for the menu.

- (NSMenuProperties)propertiesToUpdate

Return Value

A bitwise-C or of the values in the “[NSMenuProperties](#)” (page 38) that are applicable to this menu.

Discussion

The `propertiesToUpdate` method may be called on the menu from specific callbacks to determine which properties the have been defined, and whether or not they are relevant to the changes you need to make. It is intended to allow more efficient updating of the menu in certain circumstances.

For example, if [NSMenuPropertyItemImage](#) (page 39) is set, your delegate does not need to update the images of the menu items, because the images are not needed (for example, during key equivalent matching). If the [NSMenuPropertyItemImage](#) (page 39) bit is 0, you can avoid updating the menu's images, which may improve performance if computing the images is expensive.

You only have to update a property if it may have changed since you last set it, even if the corresponding bit is 1. For example, if the title of a menu item never changes, you only have to set it once.

You may call this from the menu delegate methods `menuNeedsUpdate:`, or the menu validation methods `-validateMenuItem:` or `validateUserInterfaceItem:`. Calling this at other times will raise an exception.

If a menu property does not have a corresponding bit, you should ensure it is always set properly after the callback returns.

Calling this is optional; it is always acceptable to fully update all properties of the menu.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSMenu.h

removeAllItems

Removes all the menu items in the receiver.

- (void)removeAllItems

Discussion

This method is more efficient than removing menu items individually.

Unlike the other remove methods, this method does not post [NSMenuDidChangeItemNotification](#) (page 40) notifications.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [removeItem](#): (page 31)
- [removeItemAtIndex](#): (page 32)

Declared In

NSMenu.h

removeItem:

Removes a menu item from the receiver.

- (void)removeItem:(NSMenuItem *)*anItem*

Parameters

anItem

The menu item to remove.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addItem](#): (page 15)
- [addItemWithTitle:action:keyEquivalent:](#) (page 15)
- [removeItemAtIndex](#): (page 32)

Declared In

NSMenu.h

removeItemAtIndex:

Removes the menu item at a specified location in the receiver.

```
- (void)removeItemAtIndex:(NSInteger) index
```

Parameters

index

An integer index identifying the menu item.

Discussion

After it removes the menu item, this method posts an [NSMenuDidRemoveItemNotification](#) (page 41).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addItem:](#) (page 15)
- [addItemWithTitle:action:keyEquivalent:](#) (page 15)
- [removeItem:](#) (page 31)

Related Sample Code

ButtonMadness
 DeskPictAppDockMenu
 ObjectPath
 QTAudioContextInsert
 QTAudioExtractionPanel

Declared In

NSMenu.h

setAllowsContextMenuPlugins:

Sets whether the popup menu allows appending of contextual menu plugin items.

```
- (void)setAllowsContextMenuPlugins:(BOOL) allows
```

Parameters

allows

YES if the popup menu should allow context menu plugin items to be appending, otherwise NO.

Discussion

Contextual menu plugins are system-wide services provided by other applications. For example, a contextual menu plugin might provide an “Open URL...” service. By enabling context menu plugins your application’s contextual menu will display the appropriate items for the currently selected data type.

See *Services Implementation Guide* for more information on contextual menu plugins.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSMenu.h

setAutoenablesItems:

Controls whether the receiver automatically enables and disables its menu items based on delegates implementing the `NSMenuValidation` informal protocol.

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

Parameters

flag

If *flag* is YES, menu items are automatically enabled and disabled. If *flag* is NO, menu items are not automatically enabled or disabled.

Discussion

See the `NSMenuValidation` protocol specification for more information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [autoenablesItems](#) (page 17)

Related Sample Code

GLUT

SearchField

Declared In

NSMenu.h

setDelegate:

Sets the receiver's delegate.

```
- (void)setDelegate:(id < NSMenuDelegate >)anObject
```

Parameters

anObject

The object to set as delegate.

Discussion

You can use the delegate to populate a menu just before it is going to be drawn and to check for key equivalents without creating a menu item.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [delegate](#) (page 18)

Declared In

NSMenu.h

setFont:

Sets the font used to display the menu and its submenus.

- (void)setFont:(NSFont *)*font*

Parameters

font

The font object to use.

Discussion

This font will be used to display the menu and any submenus that have not had their font set explicitly.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [font](#) (page 18)

Declared In

NSMenu.h

setMenuChangedMessagesEnabled:

Controls whether the receiver sends messages to the application's windows upon each menu change.

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

Parameters

flag

YES if the receiver should send a message at each menu change, NO otherwise.

Discussion

To avoid the “flickering” effect of many successive menu changes, invoke this method with *flag* set to NO, make changes to the menu, and invoke the method again with *flag* set to YES. This approach has the effect of batching changes and applying them all at once.

Special Considerations

On Mac OS X v10.6 and later this method has no effect. This method may be deprecated in a future release.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [menuChangedMessagesEnabled](#) (page 27)

Declared In

NSMenu.h

setMinimumWidth:

Set the minimum width of the menu.

- (void)setMinimumWidth:(CGFloat)*width*

Parameters*width*

The minimum width of the menu in screen coordinates.

Discussion

The menu will not draw smaller than its minimum width, but may draw larger if it needs more space. The default value is 0.

Availability

Available in Mac OS X v10.6 and later.

See Also

- [minimumWidth](#) (page 28)

Declared In

NSMenu.h

setShowsStateColumn:

Sets whether the receiver displays the state column.

```
- (void)setShowsStateColumn:(BOOL)showsState
```

Parameters*showsState*

YES to display the state column, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [showsStateColumn](#) (page 36)

Declared In

NSMenu.h

setSubmenu:forItem:

Assigns a menu to be a submenu of the receiver controlled by a given menu item.

```
- (void)setSubmenu:(NSMenu *)aMenu forItem:(NSMenuItem *)anItem
```

Parameters*aMenu*

A menu object that is to be a submenu of the receiver.

anItem

A menu item (that is, an object conforming to the `NSMenuItem` protocol) that controls *aMenu*. The method sets the action of *anItem* to [submenuAction:](#) (page 37).

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

setSupermenu:

Sets the receiver's supermenu.

- (void)setSupermenu:(NSMenu *)*supermenu***Parameters***supermenu*

A menu object to set as the supermenu of the receiver.

Discussion

You should never invoke this method directly; it is public so subclassers can add behavior to the default implementation. Subclassers should call the superclass's method as part of their implementation.

Availability

Available in Mac OS X v10.0 and later.

See Also- [supermenu](#) (page 37)**Declared In**

NSMenu.h

setTitle:

Sets the receiver's title.

- (void)setTitle:(NSString *)*aString***Parameters***aString*

A string to assign as the new title of the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also- [title](#) (page 38)**Declared In**

NSMenu.h

showsStateColumn

Returns a Boolean value that indicates whether the receiver displays the state column.

- (BOOL)showsStateColumn

Return Value

YES if the receiver displays the state column, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setShowsStateColumn:](#) (page 35)

Declared In

NSMenu.h

size

Returns the size of the menu.

- (NSSize)size

Return Value

The size of the menu in screen coordinates.

Discussion

The menu may draw at a smaller size when shown, depending on its positioning and display configuration.

Availability

Available in Mac OS X v10.6 and later.

Declared In

NSMenu.h

submenuAction:

The action method assigned to menu items that open submenus.

- (void)submenuAction:(id)sender

Discussion

You may override this method to implement different behavior. Never invoke this method directly.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

supermenu

Returns the receiver's supermenu.

- (NSMenu *)supermenu

Return Value

The receiver's supermenu or `nil` if it has none.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setSupermenu:](#) (page 36)

Declared In

NSMenu.h

title

Returns the receiver's title.

- (NSString *)title

Return Value

The receiver's title.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTitle:](#) (page 36)

Related Sample Code

ToolbarSample

Declared In

NSMenu.h

update

Enables or disables the receiver's menu items based on the `NSMenuValidation` informal protocol and sizes the menu to fit its current menu items if necessary.

- (void)update

Discussion

See the `NSMenuValidation` protocol specification for more information.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

Constants

NSMenuProperties

These constants are used as a bitmask for specifying a set of menu or menu item properties, used in the [propertiesToUpdate](#) (page 30).

```
enum {
    NSMenuItemTitle = 1 << 0,
    NSMenuItemAttributedTitle = 1 << 1,
    NSMenuItemKeyEquivalent = 1 << 2,
    NSMenuItemImage = 1 << 3,
    NSMenuItemEnabled = 1 << 4,
    NSMenuItemAccessibilityDescription = 1 << 5
};
typedef NSUInteger NSMenuItemProperties;
```

Constants

NSMenuItemTitle

The menu item's title.

Available in Mac OS X v10.6 and later.

Declared in NSMenu.h.

NSMenuItemAttributedTitle

The menu item's attributed string title.

Available in Mac OS X v10.6 and later.

Declared in NSMenu.h.

NSMenuItemKeyEquivalent

The menu item's key equivalent.

Available in Mac OS X v10.6 and later.

Declared in NSMenu.h.

NSMenuItemImage

The menu image.

Available in Mac OS X v10.6 and later.

Declared in NSMenu.h.

NSMenuItemEnabled

Whether the menu item is enabled or disabled.

Available in Mac OS X v10.6 and later.

Declared in NSMenu.h.

NSMenuItemAccessibilityDescription

The menu item's accessibility description.

Available in Mac OS X v10.6 and later.

Declared in NSMenu.h.

Notifications

NSMenuDidAddItemNotification

Posted after a menu item is added to the menu. The notification object is the instance of `NSMenu` that just added the new menu item. The `userInfo` dictionary contains the following information:

Key	Value
@"NSMenuItemIndex"	An <code>NSNumber</code> object containing the integer index of the menu item that was added.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

NSMenuDidChangeltemNotification

Posted after a menu item in the menu changes appearance. Changes include enabling/disabling, changes in state, and changes to title. The notification object is the instance of `NSMenu` with the menu item that changed. The `userInfo` dictionary contains the following information:

Key	Value
@"NSMenuItemIndex"	An <code>NSNumber</code> object containing the integer index of the menu item that changed.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

NSMenuDidBeginTrackingNotification

Posted when menu tracking begins. The notification object is the main menu bar (`[NSApp mainMenu]`) or the root menu of a popup button. This notification does not contain a `userInfo` dictionary.

Note: This notification is available in versions 10.3 and 10.4 of Mac OS X, however it is not publicly declared so you must declare the name constant as an `extern`, for example:

```
extern NSString *NSMenuDidBeginTrackingNotification;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSMenu.h

NSMenuDidEndTrackingNotification

Posted when menu tracking ends, even if no action is sent. The notification object is the main menu bar (`[NSApp mainMenu]`) or the root menu of a popup button. This notification does not contain a `userInfo` dictionary.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSMenu.h

NSMenuDidRemoveItemNotification

Posted after a menu item is removed from the menu. The notification object is the instance of `NSMenu` that just removed the menu item. The `userInfo` dictionary contains the following information:

Key	Value
@ <code>"NSMenuItemIndex"</code>	An <code>NSNumber</code> object containing the integer index of the menu item that was removed. Note that this index may no longer be valid and in any event no longer points to the menu item that was removed.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

NSMenuDidSendActionNotification

Posted just after the application dispatches a menu item's action method to the menu item's target. The notification object is the instance of `NSMenu` containing the chosen menu item. The `userInfo` dictionary contains the following information:

Key	Value
@ <code>"MenuItem"</code>	The menu item that was chosen.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

NSMenuWillSendActionNotification

Posted just before the application dispatches a menu item's action method to the menu item's target. The notification object is the instance of `NSMenu` containing the chosen menu item. The `userInfo` dictionary contains the following information:

Key	Value
@ <code>"MenuItem"</code>	The menu item that was chosen.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

Deprecated NSMenu Methods

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

Deprecated in Mac OS X v10.2

setMenuZone:

Sets the zone from which NSMenu objects should be allocated (**Deprecated in Mac OS X v10.2.**)

```
+ (void)setMenuZone:(NSZone *)zone
```

Parameters

zone

The memory zone to set.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

attachedMenu

Returns the menu currently attached to the receiver. (**Deprecated in Mac OS X v10.2.**)

```
- (NSMenu *)attachedMenu
```

Return Value

The menu currently attached to the receiver or `nil` if there's no such object.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

See Also

- [isAttached](#) (page 44)

Declared In

NSMenu.h

contextMenuRepresentation

Deprecated. (Deprecated in Mac OS X v10.2. Mac OS X does not use menu representations to draw menus.)

- (id)contextMenuRepresentation

Return Value

nil.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

isAttached

Returns a Boolean value that indicates whether the receiver is currently attached to another menu. (Deprecated in Mac OS X v10.2.)

- (BOOL)isAttached

Return Value

YES if the receiver is currently attached to another menu, otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

See Also

- [attachedMenu](#) (page 43)

Declared In

NSMenu.h

locationForSubmenu:

Returns the location in screen coordinates where the given submenu is displayed when opened as a submenu of the receiver. (Deprecated in Mac OS X v10.2.)

- (NSPoint)locationForSubmenu:(NSMenu *)aSubmenu

Parameters

aSubmenu

A menu object that is a submenu of the receiver.

Return Value

An `NSPoint` structure describing the location or (0,0, 0,0) if the submenu does not exist in the receiver.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Deprecated NSMenu Methods

Declared In

NSMenu.h

menuRepresentation

Deprecated. (Deprecated in Mac OS X v10.2. Mac OS X does not use menu representations to draw menus.)

- (id)menuRepresentation

Return Value

nil.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

setContextMenuRepresentation:

Deprecated. (Deprecated in Mac OS X v10.2. Mac OS X does not use menu representations to draw menus.)

- (void)setContextMenuRepresentation:(id)menuRep

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

setMenuRepresentation:

Deprecated. (Deprecated in Mac OS X v10.2. Mac OS X does not use menu representations to draw menus.)

- (void)setMenuRepresentation:(id)menuRep

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

setTearOffMenuRepresentation:

Deprecated. (Deprecated in Mac OS X v10.2. Mac OS X does not use menu representations to draw menus.)

- (void)setTearOffMenuRepresentation:(id)menuRep

Deprecated NSMenu Methods

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

sizeToFit

Resizes the receiver to exactly fit its items. (Deprecated in Mac OS X v10.2.)

- (void)sizeToFit

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

tearOffMenuRepresentation

Deprecated. (Deprecated in Mac OS X v10.2. Mac OS X does not use menu representations to draw menus.)

- (id)tearOffMenuRepresentation

Return Value

nil.

Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.2.

Declared In

NSMenu.h

Document Revision History

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

Date	Notes
2009-06-24	Added additional Mac OS X v10.6 API. Added <code>propertiesToUpdate</code> method.
2009-04-17	Updated for Mac OS X v10.6. Delegate methods moved to <i>NSMenuDelegate Protocol Reference</i> .
2008-02-08	Corrected parameter descriptions for <code>popupContextMenu:withEvent:forView:withFont:.</code>
2007-10-31	Corrected the method declaration for the <code>menuDidClose:</code> method.
2007-02-21	Updated to include API introduced in Mac OS X v10.5.
2006-06-28	Made minor changes to conform to reference consistency guidelines.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

`addItem`: [instance method 15](#)
`addItemWithTitle:action:keyEquivalent:` [instance method 15](#)
`allowsContextMenuPlugIns` [instance method 16](#)
`attachedMenu` [instance method 43](#)
`autoenablesItems` [instance method 17](#)

C

`cancelTracking` [instance method 17](#)
`cancelTrackingWithoutAnimation` [instance method 17](#)
`contextMenuRepresentation` [instance method 44](#)

D

`delegate` [instance method 18](#)

F

`font` [instance method 18](#)

H

`helpRequested`: [instance method 18](#)
`highlightedItem` [instance method 19](#)

I

`indexOfItem`: [instance method 19](#)

`indexOfItemWithRepresentedObject`: [instance method 20](#)
`indexOfItemWithSubMenu`: [instance method 20](#)
`indexOfItemWithTag`: [instance method 20](#)
`indexOfItemWithTarget:andAction:` [instance method 21](#)
`indexOfItemWithTitle`: [instance method 21](#)
`initWithTitle`: [instance method 22](#)
`insertItem:atIndex`: [instance method 23](#)
`insertItemWithTitle:action:keyEquivalent:atIndex:` [instance method 23](#)
`isAttached` [instance method 44](#)
`isTornOff` [instance method 24](#)
`itemArray` [instance method 24](#)
`itemAtIndex`: [instance method 25](#)
`itemChanged`: [instance method 25](#)
`itemWithTag`: [instance method 26](#)
`itemWithTitle`: [instance method 26](#)

L

`locationForSubMenu`: [instance method 44](#)

M

`menuBarHeight` [instance method 27](#)
`menuBarVisible` [class method 12](#)
`menuChangedMessagesEnabled` [instance method 27](#)
`menuRepresentation` [instance method 45](#)
`menuZone` [class method 13](#)
`minimumWidth` [instance method 28](#)

N

`NSMenuItemDidAddItemNotification` [notification 39](#)
`NSMenuItemDidBeginTrackingNotification` [notification 40](#)
`NSMenuItemDidChangeItemNotification` [notification 40](#)

NSMenuDidEndTrackingNotification **notification** 40
 NSMenuDidRemoveItemNotification **notification** 41
 NSMenuDidSendActionNotification **notification** 41
NSMenuProperties 38
 NSMenuItemAccessibilityDescription
 constant 39
 NSMenuItemAttributedTitle **constant** 39
 NSMenuItemEnabled **constant** 39
 NSMenuItemImage **constant** 39
 NSMenuItemKeyEquivalent **constant** 39
 NSMenuItemTitle **constant** 39
 NSMenuWillSendActionNotification **notification** 41
 numberOfItems **instance method** 28

P

performActionForItemAtIndex: **instance method** 28
 performKeyEquivalent: **instance method** 29
 popUpContextMenu:withEvent:forView: **class**
 method 13
 popUpContextMenu:withEvent:forView:withFont:
 class method 14
 popUpMenuPositioningItem:atLocation:inView:
 instance method 29
 propertiesToUpdate **instance method** 30

R

removeAllItems **instance method** 31
 removeItemAtIndex: **instance method** 32
 removeItem: **instance method** 31

S

setAllowsContextMenuPlugIns: **instance method** 32
 setAutoenablesItems: **instance method** 33
 setContextMenuRepresentation: **instance method**
 45
 setDelegate: **instance method** 33
 setFont: **instance method** 33
 setMenuBarVisible: **class method** 14
 setMenuChangedMessagesEnabled: **instance method**
 34
 setMenuRepresentation: **instance method** 45
 setMenuZone: **class method** 43
 setMinimumWidth: **instance method** 34
 setShowsStateColumn: **instance method** 35
 setSubmenu:forItem: **instance method** 35
 setSupermenu: **instance method** 36

setTearOffMenuRepresentation: **instance method**
 45
 setTitle: **instance method** 36
 showsStateColumn **instance method** 36
 size **instance method** 37
 sizeToFit **instance method** 46
 submenuAction: **instance method** 37
 supermenu **instance method** 37

T

tearOffMenuRepresentation **instance method** 46
 title **instance method** 38

U

update **instance method** 38