
NSMenu Class Reference

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



2008-02-08



Apple Inc.
© 2008 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, 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

NSMenu Class Reference 7

Overview	7
Adopted Protocols	7
Tasks	8
Managing the Menu Bar	8
Creating an NSMenu Object	8
Setting Up Menu Commands	8
Finding Menu Items	8
Finding Indices of Menu Items	9
Managing Submenus	9
Enabling and Disabling Menu Items	10
Handling Keyboard Equivalents	10
Simulating Mouse Clicks	10
Managing the Title	10
Updating Menu Layout	10
Displaying Context-Sensitive Help	11
Managing Display of the State Column	11
Controlling Allocation Zones	11
Handling Highlighting	11
Managing the Delegate	11
Handling Open and Close Events	11
Handling Tracking	12
Deprecated Methods	12
Class Methods	12
menuBarVisible	12
menuZone	13
popupContextMenu:withEvent:forView:	13
popupContextMenu:withEvent:forView:withFont:	14
setMenuBarVisible:	14
setMenuZone:	14
Instance Methods	15
addItem:	15
addItemWithTitle:action:keyEquivalent:	15
attachedMenu	16
autoenablesItems	16
cancelTracking	17
contextMenuRepresentation	17
delegate	17
helpRequested:	18
highlightedItem	18
indexOfItem:	19

indexOfItemWithRepresentedObject:	19
indexOfItemWithSubmenu:	19
indexOfItemWithTag:	20
indexOfItemWithTarget:andAction:	20
indexOfItemWithTitle:	21
initWithTitle:	21
insertItem:atIndex:	22
insertItemWithTitle:action:keyEquivalent:atIndex:	23
isAttached	23
isTornOff	24
itemArray	24
itemAtIndex:	24
itemChanged:	25
itemWithTag:	25
itemWithTitle:	26
locationForSubmenu:	26
menuBarHeight	27
menuChangedMessagesEnabled	27
menuRepresentation	27
numberOfItems	28
performActionForItemAtIndex:	28
performKeyEquivalent:	29
removeItem:	29
removeItemAtIndex:	29
setAutoenablesItems:	30
setContextMenuRepresentation:	30
setDelegate:	31
setMenuChangedMessagesEnabled:	31
setMenuRepresentation:	32
setShowsStateColumn:	32
setSubmenu:forItem:	32
setSupermenu:	33
setTearOffMenuRepresentation:	33
setTitle:	33
showsStateColumn	34
sizeToFit	34
submenuAction:	34
supermenu	34
tearOffMenuRepresentation	35
title	35
update	35
Delegate Methods	36
menu:updateItem:atIndex:shouldCancel:	36
menu:willHighlightItem:	36
menuDidClose:	37
menuHasKeyEquivalent:forEvent:target:action:	37

menuNeedsUpdate: 38
menuWillOpen: 39
numberOfItemsInMenu: 39

Notifications 40

NSMenuDidAddItemNotification 40
NSMenuDidChangeItemNotification 40
NSMenuDidBeginTrackingNotification 40
NSMenuDidEndTrackingNotification 41
NSMenuDidRemoveItemNotification 41
NSMenuDidSendActionNotification 41
NSMenuWillSendActionNotification 42

Document Revision History 43

Index 45

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	DeskPictAppDockMenu MenuItemView NumberInput_IMKit_Sample PDFKitLinker2 ToolbarSample

Overview

This class defines an object that manages an application's menus.

Adopted Protocols

NSCoding

encodeWithCoder:
initWithCoder:

NSCopying

copyWithZone:

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 21)
Initializes and returns a menu having the specified title and with autoenabling of menu items turned on.

Setting Up Menu Commands

- `insertItem:atIndex:` (page 22)
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 29)
Removes a menu item from the receiver.
- `removeItemAtIndex:` (page 29)
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).

Finding Menu Items

- `itemWithTag:` (page 25)
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 24)
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 20)
Returns the index of the first menu item in the receiver that has a specified action and target.
- [indexOfItemWithRepresentedObject:](#) (page 19)
Returns the index of the first menu item in the receiver that has a given represented object.
- [indexOfItemWithSubmenu:](#) (page 19)
Returns the index of the menu item in the receiver with the given submenu.

Managing Submenus

- [setSubmenu:forItem:](#) (page 32)
Assigns a menu to be a submenu of the receiver controlled by a given menu item.
- [submenuAction:](#) (page 34)
The action method assigned to menu items that open submenus.
- [attachedMenu](#) (page 16)
Returns the menu currently attached to the receiver.
- [isAttached](#) (page 23)
Returns a Boolean value that indicates whether the receiver is currently attached to another menu.
- [locationForSubmenu:](#) (page 26)
Returns the location in screen coordinates where the given submenu is displayed when opened as a submenu of the receiver.
- [supermenu](#) (page 34)
Returns the receiver's supermenu.
- [setSupermenu:](#) (page 33)
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).

Enabling and Disabling Menu Items

- [autoenablesItems](#) (page 16)
Returns a Boolean value that indicates whether the receiver automatically enables and disables its menu items.
- [setAutoenablesItems:](#) (page 30)
Controls whether the receiver automatically enables and disables its menu items based on delegates implementing the `NSMenuValidation` informal protocol.
- [update](#) (page 35)
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.

Handling Keyboard Equivalents

- [performKeyEquivalent:](#) (page 29)
Performs the action for the menu item that corresponds to the given key equivalent.
- [menuHasKeyEquivalent:forEvent:target:action:](#) (page 37) *delegate method*
Called to allow the delegate to return the target and action for a key-down event.

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 33)
Sets the receiver's title.
- [title](#) (page 35)
Returns the receiver's title.

Updating Menu Layout

- [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 31)
Controls whether the receiver sends messages to the application's windows upon each menu change.
- [sizeToFit](#) (page 34)
Resizes the receiver to exactly fit its items.
- [menu:updateItem:atIndex:shouldCancel:](#) (page 36) *delegate method*
Called to let the delegate update a menu item before it is displayed.

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.

Managing Display of the State Column

- [setShowsStateColumn:](#) (page 32)
Sets whether the receiver displays the state column.
- [showsStateColumn](#) (page 34)
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 14)
Sets the zone from which NSMenu objects should be allocated

Handling Highlighting

- [highlightedItem](#) (page 18)
Returns the highlighted item in the receiver.
- [menu:willHighlightItem:](#) (page 36) *delegate method*
Called to indicates that a menu is about to highlight a given item.

Managing the Delegate

- [setDelegate:](#) (page 31)
Sets the receiver's delegate.
- [delegate](#) (page 17)
Returns the receiver's delegate.

Handling Open and Close Events

- [menuWillOpen:](#) (page 39) *delegate method*
Sent when a menu is about to open.

- [menuDidClose:](#) (page 37) *delegate method*
Sent after a menu closed.

Handling Tracking

- [numberOfItemsInMenu:](#) (page 39) *delegate method*
Called when a menu is about to be displayed at the start of a tracking session so the delegate can specify the number of items in the menu.
- [menuNeedsUpdate:](#) (page 38) *delegate method*
Called when a menu is about to be displayed at the start of a tracking session so the delegate can modify the menu.
- [cancelTracking](#) (page 17)
Dismisses the menu and ends all menu tracking.

Deprecated Methods

- [contextMenuRepresentation](#) (page 17)
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [setContextMenuRepresentation:](#) (page 30)
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [tearOffMenuRepresentation](#) (page 35)
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [setTearOffMenuRepresentation:](#) (page 33)
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [setMenuRepresentation:](#) (page 32)
Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)
- [menuRepresentation](#) (page 27)
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

MenuItemView

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)**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*.

Availability

Available in Mac OS X v10.3 and later.

See Also

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

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

setMenuZone:

Sets the zone from which `NSMenu` objects should be allocated

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

Parameters*zone*

The memory zone to set.

Availability

Available in Mac OS X v10.0 and later.

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 22). 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 29)
- `removeItemAtIndex:` (page 29)

Related Sample Code

MenuItemView

PDFKitLinker2

QTAudioExtractionPanel

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

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

Related Sample Code

Clock Control

UIElementInspector

Declared In

NSMenu.h

attachedMenu

Returns the menu currently attached to the receiver.

- (NSMenu *)attachedMenu

Return ValueThe menu currently attached to the receiver or *nil* if there's no such object.**Availability**

Available in Mac OS X v10.0 and later.

See Also

- [isAttached](#) (page 23)

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 30)

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.

Related Sample Code

`MenuItemView`

Declared In

`NSMenu.h`

contextMenuRepresentation

Deprecated. (**Deprecated.** 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.

Declared In

`NSMenu.h`

delegate

Returns the receiver's delegate.

- (id)delegate

Return Value

The receiver's delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setDelegate:](#) (page 31)

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.

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.

See Also

- [menu:willHighlightItem:](#) (page 36)

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 22)
- [itemAtIndex:](#) (page 24)

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 22)
- [itemAtIndex:](#) (page 24)

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 22)
- [itemAtIndex:](#) (page 24)

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 25)
- [insertItem:atIndex:](#) (page 22)
- [itemAtIndex:](#) (page 24)

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 19)
- [insertItemAtIndex:](#) (page 22)
- [itemAtIndex:](#) (page 24)

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)
- [insertItemAtIndex:](#) (page 22)
- [itemAtIndex:](#) (page 24)

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 30)**Related Sample Code**

DeskPictAppDockMenu

MenuItemView

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 40), 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 24)- [removeItem:](#) (page 29)**Related Sample Code**

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 ValueThe 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.

Declared In

NSMenu.h

isAttached

Returns a Boolean value that indicates whether the receiver is currently attached to another menu.

```
- (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.

See Also- [attachedMenu](#) (page 16)**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.

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

- [itemWithTag:](#) (page 25)
- [itemWithTitle:](#) (page 26)
- [itemAtIndex:](#) (page 24)
- [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 Value

The 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 25)
- [itemWithTitle:](#) (page 26)
- [itemArray](#) (page 24)

Related Sample Code

`MenuItemView`

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 24)
- [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 25)
- [itemAtIndex:](#) (page 24)
- [itemArray](#) (page 24)

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.

```
- (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.

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.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setMenuChangedMessagesEnabled:](#) (page 31)

Declared In

`NSMenu.h`

menuRepresentation

Deprecated. (**Deprecated.** 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.

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)

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 42) and [NSMenuDidSendActionNotification](#) (page 41).

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:](#) (page 37)

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 29)

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 29)

Related Sample Code

ObjectPath

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 16)

Related Sample Code

SearchField

Declared In

NSMenu.h

setContextMenuRepresentation:

Deprecated. (**Deprecated.** 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.

Declared In

NSMenu.h

setDelegate:

Sets the receiver's delegate.

```
- (void)setDelegate:(id)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 17)

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.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [menuChangedMessagesEnabled](#) (page 27)

Declared In

NSMenu.h

setMenuRepresentation:

Deprecated. (**Deprecated.** 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.

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 34)

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 34).

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 34)

Declared In

NSMenu.h

setTearOffMenuRepresentation:

Deprecated. (**Deprecated.** Mac OS X does not use menu representations to draw menus.)

- (void)setTearOffMenuRepresentation:(id)*menuRep*

Availability

Available in Mac OS X v10.0 and later.

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 35)

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 32)

Declared In

NSMenu.h

sizeToFit

Resizes the receiver to exactly fit its items.

- (void)sizeToFit

Availability

Available in Mac OS X v10.0 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 33)

Declared In

NSMenu.h

tearOffMenuRepresentation

Deprecated. (**Deprecated.** 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.

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 33)

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`

Delegate Methods

menu:updateItem:atIndex:shouldCancel:

Called to let the delegate update a menu item before it is displayed.

```
- (BOOL)menu:(NSMenu *)menu updateItem:(NSMenuItem *)item atIndex:(NSInteger)index
    shouldCancel:(BOOL)shouldCancel
```

Parameters

menu

The menu object that owns *item*.

item

The menu-item object that may be updated.

index

The integer index of the menu item.

shouldCancel

Set to YES if, due to some user action, the menu no longer needs to be displayed before all the menu items have been updated. You can ignore this flag, return YES, and continue; or you can save your work (to save time the next time your delegate is called) and return NO to stop the updating.

Return Value

YES to continue the process. If you return NO, your `menu:updateItem:atIndex:shouldCancel:` is not called again. In that case, it is your responsibility to trim any extra items from the menu.

Discussion

If your [numberOfItemsInMenu:](#) (page 39) delegate method returns a positive value, then your `menu:updateItem:atIndex:shouldCancel:` method is called for each item in the menu. You can then update the menu title, image, and so forth for each menu item.

Availability

Available in Mac OS X v10.3 and later.

Declared In

`NSMenu.h`

menu:willHighlightItem:

Called to indicate that a menu is about to highlight a given item.

```
- (void)menu:(NSMenu *)menu
  willHighlightItem:(NSMenuItem *)item
```

Parameters*menu*

The menu object about to highlight an item.

item

The item about to be highlighted.

Discussion

Only one item per menu can be highlighted at a time. If *item* is `nil`, it means that all items in the menu are about to be unhighlighted

Availability

Available in Mac OS X v10.5 and later.

See Also- [highlightedItem](#) (page 18)**Declared In**

NSMenu.h

menuDidClose:

Sent after a menu closed.

```
- (void)menuDidClose:(NSMenu *)menu
```

Parameters*menu*

The menu that closed.

Special Considerations

Do not modify the structure of the menu or the menu items during this method.

Availability

Available in Mac OS X v10.5 and later.

See Also- [menuWillOpen:](#) (page 39)**Declared In**

NSMenu.h

menuHasKeyEquivalent:forEvent:target:action:

Called to allow the delegate to return the target and action for a key-down event.

```
- (BOOL)menuHasKeyEquivalent:(NSMenu *)menu forEvent:(NSEvent *)event target:(id *)target action:(SEL *)action
```

Parameters*menu*

The menu object sending the delegation message.

event

An `NSEvent` object representing a key-down event.

target

Return by reference the target object for the menu item that corresponds to the event. Specify `nil` to requests the menu's target.

action

Return by reference the action selector for the menu item that corresponds to the event.

Return Value

If there is a valid and enabled menu item that corresponds to this key-down even, return `YES` after specifying the target and action. Return `NO` if there are no items with that key equivalent or if the item is disabled.

Discussion

If the delegate does not define this method, the menu is populated to find out if any items have a matching key equivalent.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [performKeyEquivalent:](#) (page 29)
- [performActionForItemAtIndex:](#) (page 28)
- [setDelegate:](#) (page 31)

Declared In

`NSMenu.h`

menuNeedsUpdate:

Called when a menu is about to be displayed at the start of a tracking session so the delegate can modify the menu.

```
- (void)menuNeedsUpdate:(NSMenu *)menu
```

Parameters*menu*

The menu object that is about to be displayed.

Discussion

You can change the menu by adding, removing or modifying menu items. Be sure to set the proper enable state for any new menu items. If populating the menu will take a long time, implement [numberOfItemsInMenu:](#) (page 39) and [menu:updateItem:atIndex:shouldCancel:](#) (page 36) instead.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setDelegate:](#) (page 31)

Declared In

NSMenu.h

menuWillOpen:

Sent when a menu is about to open.

- (void)menuWillOpen:(NSMenu *)*menu***Parameters***menu*

The menu that is about to open.

Special Considerations

Do not modify the structure of the menu or the menu items during this method.

Availability

Available in Mac OS X v10.5 and later.

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

NSMenu.h

numberOfItemsInMenu:

Called when a menu is about to be displayed at the start of a tracking session so the delegate can specify the number of items in the menu.

- (NSInteger)numberOfItemsInMenu:(NSMenu *)*menu***Parameters***menu*

The menu object about to be displayed.

Return Value

The number of menu items in the menu.

Discussion

If you return a positive value, the menu is resized by either removing or adding items. Newly created items are blank. After the menu is resized, your [menu:updateItem:atIndex:shouldCancel:](#) (page 36) method is called for each item. If you return a negative value, the number of items is left unchanged and [menu:updateItem:atIndex:shouldCancel:](#) is not called. If you can populate the menu quickly, you can implement [menuNeedsUpdate:](#) (page 38) instead of [numberOfItemsInMenu:](#) and [menu:updateItem:atIndex:shouldCancel:](#).

Availability

Available in Mac OS X v10.3 and later.

See Also- [setDelegate:](#) (page 31)

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
@ <code>"NSMenuItemIndex"</code>	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

NSMenuDidChangeItemNotification

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
@ <code>"NSMenuItemIndex"</code>	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
@ "NSMenuItemIndex"	An NSNumber 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
@ "MenuItem"	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
@MenuItem	The menu item that was chosen.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSMenu.h

Document Revision History

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

Date	Notes
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](#)
attachedMenu [instance method 16](#)
autoenablesItems [instance method 16](#)

C

cancelTracking [instance method 17](#)
contextMenuRepresentation [instance method 17](#)

D

delegate [instance method 17](#)

H

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

I

indexOfItem: [instance method 19](#)
indexOfItemWithRepresentedObject: [instance method 19](#)
indexOfItemWithSubmenu: [instance method 19](#)
indexOfItemWithTag: [instance method 20](#)
indexOfItemWithTarget:andAction: [instance method 20](#)
indexOfItemWithTitle: [instance method 21](#)
initWithTitle: [instance method 21](#)
insertItem:atIndex: [instance method 22](#)

insertItemWithTitle:action:keyEquivalent:atIndex: [instance method 23](#)
isAttached [instance method 23](#)
isTornOff [instance method 24](#)
itemArray [instance method 24](#)
itemAtIndex: [instance method 24](#)
itemChanged: [instance method 25](#)
itemWithTag: [instance method 25](#)
itemWithTitle: [instance method 26](#)

L

locationForSubmenu: [instance method 26](#)

M

menuBarHeight [instance method 27](#)
menuBarVisible [class method 12](#)
menuChangedMessagesEnabled [instance method 27](#)
menu:updateItem:atIndex:shouldCancel: [<NSObject> delegate method 36](#)
menu:willHighlightItem: [<NSObject> delegate method 36](#)
menuDidClose: [<NSObject> delegate method 37](#)
menuHasKeyEquivalent:forEvent:target:action: [<NSObject> delegate method 37](#)
menuNeedsUpdate: [<NSObject> delegate method 38](#)
menuRepresentation [instance method 27](#)
menuWillOpen: [<NSObject> delegate method 39](#)
menuZone [class method 13](#)

N

NSMenuDidAddItemNotification [notification 40](#)
NSMenuDidBeginTrackingNotification [notification 40](#)
NSMenuDidChangeItemNotification [notification 40](#)
NSMenuDidEndTrackingNotification [notification 41](#)

NSMenuDidRemoveItemNotification [notification 41](#)
 NSMenuDidSendActionNotification [notification 41](#)
 NSMenuWillSendActionNotification [notification 42](#)
 numberOfItems [instance method 28](#)
 numberOfItemsInMenu: <NSObject> [delegate method 39](#)

P

performActionForItemAtIndex: [instance method 28](#)
 performKeyEquivalent: [instance method 29](#)
 popUpContextMenu:withEvent:forView: [class method 13](#)
 popUpContextMenu:withEvent:forView:withFont: [class method 14](#)

R

removeItemAtIndex: [instance method 29](#)
 removeItem: [instance method 29](#)

S

setAutoenablesItems: [instance method 30](#)
 setContextMenuRepresentation: [instance method 30](#)
 setDelegate: [instance method 31](#)
 setMenuBarVisible: [class method 14](#)
 setMenuChangedMessagesEnabled: [instance method 31](#)
 setMenuRepresentation: [instance method 32](#)
 setMenuZone: [class method 14](#)
 setShowsStateColumn: [instance method 32](#)
 setSubmenu:forItem: [instance method 32](#)
 setSupermenu: [instance method 33](#)
 setTearOffMenuRepresentation: [instance method 33](#)
 setTitle: [instance method 33](#)
 showsStateColumn [instance method 34](#)
 sizeToFit [instance method 34](#)
 submenuAction: [instance method 34](#)
 supermenu [instance method 34](#)

T

tearOffMenuRepresentation [instance method 35](#)
 title [instance method 35](#)

U

update [instance method 35](#)