NSMenuView Class Reference

Cocoa > User Experience

2009-02-04

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

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

NSMenuView Class Reference

Inherits from	NSView : NSResponder : NSObject
Conforms to	NSAnimatablePropertyContainer (NSView) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later. Not available to 64-bit applications.
Companion guide	Application Menu and Pop-up List Programming Topics for Cocoa
Declared in	NSMenuView.h

Overview

The NSMenuView class handles the display of menus on the user's screen. A menu view displays its menu either horizontally or vertically and allows the user to interact with the items of that menu, either to navigate through hierarchical menus or to select a particular item.

Note: `NSMenuView` is deprecated and is no longer used to draw menus. Calling its methods will not affect the appearance of your menus. Instead, use `NSView` customizations on an `NSMenuItem`.

Tasks

Initializing a Menu View

`- initAsTearOff` (page 12)
Deprecated. Tear-off menus are not supported in Mac OS X.

`- initWithFrame:` (page 13)
Initialized a newly allocated menu view with a specified frame rectangle.

Managing Menu View Attributes

+ `menuBarHeight` (page 8)
    Returns the height of the menu bar.

– `setMenu:` (page 20)
    Sets the menu to be displayed in the receiver

– `menu` (page 17)
    Returns the menu object associated with this menu view.

– `setHorizontal:` (page 19)
    Sets the orientation of the menu.

– `isHorizontal` (page 14)
    Returns `YES` if the menu is displayed horizontally; such as for a menu bar, otherwise returns `NO`.

– `setFont:` (page 19)
    Sets the default font to use when drawing the menu text.

– `font` (page 10)
    Returns the default font used to draw the menu text.

– `setHighlightedItemIndex:` (page 19)
    Highlights the menu item at a specific location.

– `highlightedItemIndex` (page 10)
    Returns the index of the currently highlighted menu item, or –1 if no menu item in the menu is highlighted.

– `setMenuItemCell:forItemAtIndex:` (page 21)
    Replaces the menu item cell at a specific location.

– `menuItemCellForItemAtIndex:` (page 17)
    Returns the menu item cell at the specified location.

– `attachedMenuView` (page 9)
    Returns the receiver's attached menu view.

– `attachedMenu` (page 8)
    Returns the menu object associated with this object's attached menu view.

– `isAttached` (page 13)
    Returns `YES` if this menu is currently attached to its parent menu, `NO` otherwise.

– `isTornOff` (page 14)
    Deprecated. Tear-off menus are not supported in Mac OS X.

– `horizontalEdgePadding` (page 11)
    Returns the amount of horizontal space used for padding menu item components.

– `setHorizontalEdgePadding:` (page 20)
    Sets the horizontal padding for menu item components.

Responding to Notifications

– `itemChanged:` (page 15)
    Marks the menu view as needing to be resized so changes in size resulting from a change in the menu will be tracked.

- `itemAdded:` (page 14)
  Creates a new menu item cell for the newly created item and marks the menu view as needing to be resized.
- `itemRemoved:` (page 15)
  Removes the removed item's menu item cell and marks the menu view as needing to be resized.

Working With Submenus

- `detachSubmenu` (page 10)
  Detaches the window associated with the currently visible submenu and removes any menu item highlights.
- `attachSubmenuForItemAtIndex:` (page 9)
  Attaches the submenu associated with the menu item at *`index`*.

Calculating Menu Geometry

- `update` (page 24)
  Asks the associated menu object to update itself.
- `setNeedsSizing:` (page 21)
  Sets a flag that indicates whether the layout is invalid and needs resizing.
- `needsSizing` (page 17)
  Returns `YES` if the menu view needs to be resized due to changes in the menu object, `NO` otherwise.
- `sizeToFit` (page 22)
  Used internally by the menu view to cache information about the menu item geometry.
- `stateImageOffset` (page 23)
  Returns the offset to the space reserved for state images of this menu.
- `stateImageWidth` (page 23)
  Returns the maximum width of the state images used by this menu.
- `imageAndTitleOffset` (page 11)
  Returns the offset to the starting point of a menu item's image and title section.
- `imageAndTitleWidth` (page 11)
  Returns the maximum width of a menu item's image and title section.
- `keyEquivalentOffset` (page 15)
  Returns the beginning position of the menu's key equivalent text.
- `keyEquivalentWidth` (page 16)
  Returns the width of the menu's key equivalent text.
- `innerRect` (page 13)
  Returns the drawing rectangle for the menu contents.
- `rectOfItemAtIndex:` (page 18)
  Returns the drawing rectangle of the specified menu item.
- `indexOfItemAtPoint:` (page 12)
  Returns the index of the menu item underneath the specified or –1 if no menu item is underneath that point.

`- setNeedsDisplayForItemAtIndex:` (page 21)
: Adds the region occupied by the menu item at a specific location to the menu view's invalid region.

`- locationForSubmenu:` (page 16)
: Returns the origin of the submenu view's window.

`- setWindowFrameForAttachingToRect:onScreen:preferredEdge:popUpSelectedItem:` (page 22)
: Causes the menu view to resize its window so its frame is the appropriate size to attach to a specified rectangle within the screen.

Handling Events

`- performActionWithHighlightingForItemAtIndex:` (page 18)
: Uses the associated menu object to perform the action associated with the specified item when a key equivalent is pressed.

`- trackWithEvent:` (page 24)
: Handles events sent to this menu view.

Class Methods

menuBarHeight

Returns the height of the menu bar.

```
+ (CGFloat)menuBarHeight
```

Discussion
This method is superseded in Mac OS X v10.4 by the NSMenu `menuBarHeight` instance method.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In
`NSMenuView.h`

Instance Methods

attachedMenu

Returns the menu object associated with this object's attached menu view.

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

Discussion
The attached menu view is the one associated with the currently visible submenu, if any.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- attachedMenuView` (page 9)
`- isAttached` (page 13)

Declared In
`NSMenuView.h`

attachedMenuView

Returns the receiver's attached menu view.

```
- (NSMenuView *)attachedMenuView
```

Discussion
The attached menu view is the one associated with the currently visible submenu, if any.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- attachedMenu` (page 8)
`- detachSubmenu` (page 10)
`- isAttached` (page 13)

Declared In
`NSMenuView.h`

attachSubmenuForItemAtIndex:

Attaches the submenu associated with the menu item at *index*.

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

Discussion
This method prepares the submenu for display by positioning its window and ordering it to the front.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setWindowFrameForAttachingToRect:onScreen:preferredEdge:popUpSelectedItem:` (page 22)
`- orderFront:` (NSWindow)

Declared In
`NSMenuView.h`

detachSubmenu

Detaches the window associated with the currently visible submenu and removes any menu item highlights.

```
- (void)detachSubmenu
```

Discussion

If the submenu itself displays further submenus, this method detaches the windows associated with those submenus as well.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

- attachSubmenuForItemAtIndex: (page 9)
- setHighlightedItemIndex: (page 19)
- orderOut: (NSWindow)

Declared In

`NSMenuView.h`

font

Returns the default font used to draw the menu text.

```
- (NSFont *)font
```

Discussion

New items use this font by default, although the item’s menu item cell can use a different font.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

- setFont: (page 19)

Declared In

`NSMenuView.h`

highlightedItemIndex

Returns the index of the currently highlighted menu item, or –1 if no menu item in the menu is highlighted.

```
- (NSInteger)highlightedItemIndex
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

- setHighlightedItemIndex: (page 19)

Declared In

`NSMenuView.h`

horizontalEdgePadding

Returns the amount of horizontal space used for padding menu item components.

```
- (CGFloat)horizontalEdgePadding
```

Discussion

The edge padding is added to the sides of each menu item component. This space is used to provide a visual separation between components of the menu item.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- setHorizontalEdgePadding:` (page 20)

Declared In

`NSMenuView.h`

imageAndTitleOffset

Returns the offset to the starting point of a menu item's image and title section.

```
- (CGFloat)imageAndTitleOffset
```

Discussion

The image and title section of a menu item displays an image, a title, or possibly both as a way to identify the purpose of the menu item. The value returned by this method is used for all menu items of the menu.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- imageAndTitleWidth` (page 11)

`- stateImageOffset` (page 23)

`- keyEquivalentOffset` (page 15)

Declared In

`NSMenuView.h`

imageAndTitleWidth

Returns the maximum width of a menu item's image and title section.

```
- (CGFloat)imageAndTitleWidth
```

Discussion

The image and title section of a menu item displays an image, a title, or possibly both as a way to identify the purpose of the menu item. The value returned by this method is used for all menu items of the menu.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- imageAndTitleOffset` (page 11)

`- stateImageWidth` (page 23)

`- keyEquivalentWidth` (page 16)

Declared In

`NSMenuView.h`

indexOfItemAtPoint:

Returns the index of the menu item underneath the specified or –1 if no menu item is underneath that point.

```
- (NSInteger)indexOfItemAtPoint:(NSPoint)point
```

Discussion

This method considers the menu borders as part of the item when calculating whether *point* is in the menu item rectangle. This method invokes the `rectOfItemAtIndex:` (page 18) method to obtain the basic rectangle for each menu item but may adjust that rectangle before testing.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- rectOfItemAtIndex:` (page 18)

Declared In

`NSMenuView.h`

initAsTearOff

Deprecated. Tear-off menus are not supported in Mac OS X.

```
- (id)initAsTearOff
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In
`NSMenuView.h`

initWithFrame:

Initialized a newly allocated menu view with a specified frame rectangle.

```
- (id)initWithFrame:(NSRect)frame
```

Discussion
This method is the designated initialization method for NSMenuView.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In
`NSMenuView.h`

innerRect

Returns the drawing rectangle for the menu contents.

```
- (NSRect)innerRect
```

Discussion
This rectangle is different (typically smaller) from the view bounds in that it does not include the space used to draw the menu borders.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- bounds` (NSView)

Declared In
`NSMenuView.h`

isAttached

Returns `YES` if this menu is currently attached to its parent menu, `NO` otherwise.

```
- (BOOL)isAttached
```

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- attachedMenu` (page 8)

`- attachedMenuView` (page 9)

Declared In
`NSMenuView.h`

isHorizontal

Returns `YES` if the menu is displayed horizontally; such as for a menu bar, otherwise returns `NO`.

```
- (BOOL)isHorizontal
```

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setHorizontal:` (page 19)

Declared In
`NSMenuView.h`

isTornOff

Deprecated. Tear-off menus are not supported in Mac OS X.

```
- (BOOL)isTornOff
```

Discussion
Returns `YES` if this menu view's window is disassociated from its parent menu.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In
`NSMenuView.h`

itemAdded:

Creates a new menu item cell for the newly created item and marks the menu view as needing to be resized.

```
- (void)itemAdded:(NSNotification *)notification
```

Discussion
This method is registered with the menu view's associated NSMenu object for notifications of the type `NSMenuDidAddItemNotification`. The *`notification`* parameter contains the notification data.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setNeedsSizing:` (page 21)

Declared In
`NSMenuView.h`

itemChanged:

Marks the menu view as needing to be resized so changes in size resulting from a change in the menu will be tracked.

```
- (void)itemChanged:(NSNotification *)notification
```

Discussion
This method is registered with the menu view's associated NSMenu object for notifications of the type `NSMenuDidChangeItemNotification`. The *notification* parameter contains the notification data.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setNeedsSizing:` (page 21)

Declared In
`NSMenuView.h`

itemRemoved:

Removes the removed item's menu item cell and marks the menu view as needing to be resized.

```
- (void)itemRemoved:(NSNotification *)notification
```

Discussion
This method is registered with the menu view's associated NSMenu object for notifications of the type `NSMenuDidRemoveItemNotification`. The *notification* parameter contains the notification data.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setNeedsSizing:` (page 21)

Declared In
`NSMenuView.h`

keyEquivalentOffset

Returns the beginning position of the menu's key equivalent text.

```
- (CGFloat)keyEquivalentOffset
```

Discussion

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`– keyEquivalentWidth` (page 16)

`– stateImageOffset` (page 23)

`– imageAndTitleOffset` (page 11)

Declared In

`NSMenuView.h`

keyEquivalentWidth

Returns the width of the menu's key equivalent text.

```
- (CGFloat)keyEquivalentWidth
```

Discussion

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`– keyEquivalentOffset` (page 15)

`– stateImageWidth` (page 23)

`– imageAndTitleWidth` (page 11)

Declared In

`NSMenuView.h`

locationForSubmenu:

Returns the origin of the submenu view's window.

```
- (NSPoint)locationForSubmenu:(NSMenu *)aSubmenu
```

Discussion

The *aSubmenu* parameter specifies the submenu being positioned and must belong to a menu item of this menu view. This method positions the submenu adjacent to its menu item as well as possible given the type of menu and the space constraints of the user's screen.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- setWindowFrameForAttachingToRect:onScreen:preferredEdge:popUpSelectedItem:` (page 22)

`- sizeToFit` (page 22)

Declared In

`NSMenuView.h`

menu

Returns the menu object associated with this menu view.

```
- (NSMenu *)menu
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- setMenu:` (page 20)

Declared In

`NSMenuView.h`

menuItemCellForItemAtIndex:

Returns the menu item cell at the specified location.

```
- (NSMenuItemCell *)menuItemCellForItemAtIndex:(NSInteger)index
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- setMenuItemCell:forItemAtIndex:` (page 21)

`- sizeToFit` (page 22)

Declared In

`NSMenuView.h`

needsSizing

Returns `YES` if the menu view needs to be resized due to changes in the menu object, `NO` otherwise.

```
- (BOOL)needsSizing
```

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setNeedsSizing:` (page 21)

Declared In
`NSMenuView.h`

performActionWithHighlightingForItemAtIndex:

Uses the associated menu object to perform the action associated with the specified item when a key equivalent is pressed.

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

Discussion
Because the menu item at index might not currently be visible, this method provides visual feedback by highlighting the nearest visible parent menu item before performing the action. After the action has been sent, this method removes the highlighting for the menu item.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- performActionForItemAtIndex:` (NSMenu)

Declared In
`NSMenuView.h`

rectOfItemAtIndex:

Returns the drawing rectangle of the specified menu item.

```
- (NSRect)rectOfItemAtIndex:(NSInteger)index
```

Discussion
The drawing rectangle may not be the same width or height as the actual menu and in fact is typically smaller to account for borders drawn by the menu view.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- innerRect` (page 13)
`- needsSizing` (page 17)

`- sizeToFit` (page 22)

Declared In
`NSMenuView.h`

setFont:

Sets the default font to use when drawing the menu text.

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

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- font` (page 10)

Declared In
`NSMenuView.h`

setHighlightedItemIndex:

Highlights the menu item at a specific location.

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

Discussion
Specify –1 for *`index`* to remove all highlighting from the menu.

The rectangle of the menu item is marked as invalid and is redrawn the next time the event loop comes around. If another menu item was previously highlighted, that menu item is redrawn without highlights when the event loop comes around again.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- setNeedsDisplayForItemAtIndex:` (page 21)

`- highlightedItemIndex` (page 10)

Declared In
`NSMenuView.h`

setHorizontal:

Sets the orientation of the menu.

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

Discussion

If *`flag`* is `YES`, the menu's items are displayed horizontally; otherwise the menu's items are displayed vertically.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`– isHorizontal` (page 14)

Declared In

`NSMenuView.h`

setHorizontalEdgePadding:

Sets the horizontal padding for menu item components.

```
- (void)setHorizontalEdgePadding:(CGFloat)pad
```

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`– horizontalEdgePadding` (page 11)

Declared In

`NSMenuView.h`

setMenu:

Sets the menu to be displayed in the receiver

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

Discussion

This method invokes the `setNeedsSizing:` (page 21) method to force the menu view's layout to be recalculated before drawing.

This method adds the menu view to the new NSMenu object's list of observers. The notifications this method establishes notify this menu view when menu items in the NSMenu object are added, removed, or changed. This method removes the menu view from its previous NSMenu object's list of observers.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`– setNeedsSizing:` (page 21)

`– itemAdded:` (page 14)

`– itemRemoved:` (page 15)

`– itemChanged:` (page 15)

Declared In

`NSMenuView.h`

setMenuItemCell:forItemAtIndex:

Replaces the menu item cell at a specific location.

```
- (void)setMenuItemCell:(NSMenuItemCell *)cell forItemAtIndex:(NSInteger)index
```

Discussion

This method does not change the contents of the menu itself; it changes only the cell used to display the menu item at *index*. The old cell is released, and both the new cell and the menu view are marked as needing resizing.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- menuItemCellForItemAtIndex:` (page 17)

`- setNeedsSizing:` (page 21)

Declared In

`NSMenuView.h`

setNeedsDisplayForItemAtIndex:

Adds the region occupied by the menu item at a specific location to the menu view's invalid region.

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

Discussion

The region to be redrawn includes the space occupied by the menu borders. This invalid region is redrawn the next time the event loop comes around.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- rectOfItemAtIndex:` (page 18)

`- setNeedsDisplayInRect:` (NSView)

Declared In

`NSMenuView.h`

setNeedsSizing:

Sets a flag that indicates whether the layout is invalid and needs resizing.

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

Discussion

If `flag` is `YES`, the menu contents have changed or the menu appearance has changed. This method is used internally; you should not need to invoke it directly unless you are implementing a subclass that can cause the layout to become invalid.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- sizeToFit` (page 22)

Declared In

`NSMenuView.h`

setWindowFrameForAttachingToRect:onScreen:preferredEdge:popUpSelectedItem:

Causes the menu view to resize its window so its frame is the appropriate size to attach to a specified rectangle within the screen.

```
- (void)setWindowFrameForAttachingToRect:(NSRect)screenRect onScreen:(NSScreen
    *)screen preferredEdge:(NSRectEdge)edge
    popUpSelectedItem:(NSInteger)selectedItemIndex
```

Discussion

If `selectedItemIndex` contains a value other than –1, this method attempts to position the menu such that the item at `selectedItemIndex` appears on top of `screenRect`.

The `selectedItemIndex` parameter specifies the amount by which the selected item's rectangle overlaps `screenRect`.

If the preferred edge, `edge`, cannot be honored, because there is not enough room, the opposite edge is used. If the rectangle does not completely fit either edge, this method uses the edge where there is more room.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability

Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also

`- sizeToFit` (page 22)

Declared In

`NSMenuView.h`

sizeToFit

Used internally by the menu view to cache information about the menu item geometry.

```
- (void)sizeToFit
```

Discussion
This cache is updated as necessary when menu items are added, removed, or changed.

The geometry of each menu item is determined by asking its corresponding menu item cell. The menu item cell is obtained from the `menuItemCellForItemAtIndex:` (page 17) method.

Availability
Available in Mac OS X v10.0 and later.
Not available to 64-bit applications.

See Also
- `setNeedsSizing:` (page 21)
- `menuItemCellForItemAtIndex:` (page 17)

Declared In
`NSMenuView.h`

stateImageOffset

Returns the offset to the space reserved for state images of this menu.

```
- (CGFloat)stateImageOffset
```

Discussion
The offset is used for all menu items of the menu.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability
Available in Mac OS X v10.0 and later.
Not available to 64-bit applications.

See Also
- `horizontalEdgePadding` (page 11)
- `setHorizontalEdgePadding:` (page 20)
- `sizeToFit` (page 22)

Declared In
`NSMenuView.h`

stateImageWidth

Returns the maximum width of the state images used by this menu.

```
- (CGFloat)stateImageWidth
```

Discussion
The width is used for all menu items of the menu.

If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view information.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- sizeToFit` (page 22)

Declared In
`NSMenuView.h`

trackWithEvent:

Handles events sent to this menu view.

```
- (BOOL)trackWithEvent:(NSEvent *)event
```

Discussion
If *event* is a mouse event, this method tracks the cursor position in the menu and displays the menus as appropriate. This method also handles mouse clicks that result in the selection of a menu item, in which case the menu item's action is performed.

You should not need to use this method directly.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

Declared In
`NSMenuView.h`

update

Asks the associated menu object to update itself.

```
- (void)update
```

Discussion
If any changes have been made to the menu's contents, this method invokes `sizeToFit` (page 22) to update the menu view's layout.

Availability
Available in Mac OS X v10.0 and later.

Not available to 64-bit applications.

See Also
`- sizeToFit` (page 22)
`- setNeedsSizing:` (page 21)
`- update` (NSMenu)

Declared In
`NSMenuView.h`

Document Revision History

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

Date	Notes
2009-02-04	Added information regarding what to use in place of NSMenuView.
2007-03-02	Revised task headings.
2006-05-23	First publication of this content as a separate document.

Index