
NSPopUpButtonCell Class Reference

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



2008-10-15



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

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

NSPopupButtonCell Class Reference 5

Overview	5
Tasks	5
Initialization	5
Getting and Setting Attributes	6
Adding and Removing Items	6
Accessing the Items	7
Dealing with Selection	7
Title Conveniences	8
Setting the Image	8
Handling Events and Action Messages	8
Instance Methods	8
addItemWithTitle:	8
addItemWithTitle:	9
alterStateOfSelectedItem	10
arrowPosition	10
attachPopupWithFrame:inView:	11
autoenablesItems	11
dismissPopup	12
indexOfItem:	12
indexOfItemWithRepresentedObject:	12
indexOfItemWithTag:	13
indexOfItemWithTarget:andAction:	14
indexOfItemWithTitle:	14
indexOfSelectedItem	15
initWithCell:pullsDown:	15
insertItemWithTitle:atIndex:	16
itemArray	17
itemAtIndex:	17
itemTitleAtIndex:	17
itemTitles	18
itemWithTitle:	18
lastItem	19
menu	19
numberOfItems	19
objectValue	20
performClickWithFrame:inView:	20
preferredEdge	20
pullsDown	21
removeAllItems	21
removeItemAtIndex:	22

- removeItemWithTitle: 22
- selectedItem 22
- selectItem: 23
- selectItemAtIndex: 23
- selectItemWithTag: 24
- selectItemWithTitle: 25
- setAltersStateOfSelectedItem: 25
- setArrowPosition: 26
- setAutoenablesItems: 26
- setImage: 26
- setMenu: 27
- setObjectValue: 27
- setPreferredEdge: 28
- setPullsDown: 28
- setTitle: 29
- setUsesItemFromMenu: 29
- synchronizeTitleAndSelectedItem 30
- titleOfSelectedItem 31
- usesItemFromMenu 31
- Constants 31
 - NSPopUpArrowPosition 31
- Notifications 32
 - NSPopUpButtonCellWillPopUpNotification 32

Document Revision History 33

Index 35

NSPopUpButtonCell Class Reference

Inherits from	NSMenuItemCell : NSButtonCell : NSActionCell : NSCell : NSObject
Conforms to	NSCoding (NSCell) NSCopying (NSCell) 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	NSPopUpButtonCell.h
Related sample code	MenuItemView QTAudioExtractionPanel Quartz Composer WWDC 2005 TextEdit TextEditPlus

Overview

The `NSPopUpButtonCell` class defines the visual appearance of pop-up buttons that display pop-up or pull-down menus. Pop-up menus present the user with a set of choices, much the way radio buttons do, but using much less space. Pull-down menus also provide a set of choices but present the information in a slightly different way, usually to provide a set of commands from which the user can choose.

The `NSPopUpButtonCell` class implements the user interface for the `NSPopUpButton` class.

Note that while a menu is tracking, adding, removing, or changing items on the menu is not reflected.

Tasks

Initialization

- [initWithCell:pullsDown:](#) (page 15)

Returns an `NSPopUpButtonCell` object initialized with the specified title.

Getting and Setting Attributes

- [setMenu:](#) (page 27)
Sets the pop-up button's associated menu.
- [menu](#) (page 19)
Returns the pop-up button's associated menu.
- [setPullsDown:](#) (page 28)
Sets whether the receiver behaves as a pull-down or pop-up menu.
- [pullsDown](#) (page 21)
Returns a Boolean value indicating the behavior of the control's menu.
- [setAutoenablesItems:](#) (page 26)
Sets whether the receiver automatically enables and disables its items every time a user event occurs.
- [autoenablesItems](#) (page 11)
Returns whether the receiver automatically enables and disables its items every time a user event occurs.
- [setPreferredEdge:](#) (page 28)
Sets the edge of the receiver next to which the pop-up menu should appear under restrictive screen conditions.
- [preferredEdge](#) (page 20)
Returns the edge of the receiver next to which the pop-up menu is displayed under restrictive screen conditions.
- [setUsesItemFromMenu:](#) (page 29)
Sets whether the pop-up button uses an item from the menu for its own title.
- [usesItemFromMenu](#) (page 31)
Returns a Boolean value indicating whether the pop-up button uses an item from the menu for its own title.
- [setAltersStateOfSelectedItem:](#) (page 25)
Sets whether the receiver links the state of the menu items to the current selection.
- [altersStateOfSelectedItem](#) (page 10)
Returns a Boolean value indicating whether the receiver links the state of the selected menu item to the current selection.
- [setArrowPosition:](#) (page 26)
Sets the position of the arrow displayed on the receiver.
- [arrowPosition](#) (page 10)
Returns the position of the arrow displayed on the receiver.

Adding and Removing Items

- [addItemWithTitle:](#) (page 9)
Adds an item with the specified title to the end of the menu.
- [addItemWithTitles:](#) (page 8)
Adds multiple items to the end of the menu.
- [insertItemWithTitle:atIndex:](#) (page 16)
Inserts an item at the specified position in the menu.

- [removeItemWithTitle:](#) (page 22)
Removes the item with the specified title from the menu.
- [removeItemAtIndex:](#) (page 22)
Removes the item at the specified index.
- [removeAllItems](#) (page 21)
Removes all items in the receiver's item menu.

Accessing the Items

- [itemArray](#) (page 17)
Returns the items in the menu.
- [numberOfItems](#) (page 19)
Returns the number of items in the menu.
- [indexOfItem:](#) (page 12)
Returns the index of the specified menu item.
- [indexOfItemWithTitle:](#) (page 14)
Returns the index of the item with the specified title.
- [indexOfItemWithTag:](#) (page 13)
Returns the index of the menu item with the specified tag.
- [indexOfItemWithRepresentedObject:](#) (page 12)
Returns the index of the menu item that holds the specified represented object.
- [indexOfItemWithTarget:andAction:](#) (page 14)
Returns the index of the menu item with the specified target and action.
- [itemAtIndex:](#) (page 17)
Returns the menu item at the specified index.
- [itemWithTitle:](#) (page 18)
Returns the menu item with the specified title.
- [lastItem](#) (page 19)
Returns the last item in the menu.
- [setObjectValue:](#) (page 27)
Selects the item at a specific index using an object value.
- [objectValue](#) (page 20)
Returns the index of the selected item.

Dealing with Selection

- [selectItem:](#) (page 23)
Selects the specified menu item.
- [selectItemAtIndex:](#) (page 23)
Selects the item in the menu at the specified index.
- [selectItemWithTag:](#) (page 24)
Selects the menu item with the specified tag.

- [selectItemWithTitle:](#) (page 25)
Selects the item with the specified title.
- [setTitle:](#) (page 29)
Sets the string displayed in the receiver when the user isn't pressing the mouse button.
- [selectedItem](#) (page 22)
Returns the menu item last selected by the user.
- [indexOfSelectedItem](#) (page 15)
Returns the index of the item last selected by the user.
- [synchronizeTitleAndSelectedItem](#) (page 30)
Synchronizes the the pop-up button's displayed item with the currently selected menu item.

Title Conveniences

- [itemTitleAtIndex:](#) (page 17)
Returns the title of the item at the specified index.
- [itemTitles](#) (page 18)
Returns the titles of all of the items in the menu.
- [titleOfSelectedItem](#) (page 31)
Returns the title of the item last selected by the user.

Setting the Image

- [setImage:](#) (page 26)
This method has no effect.

Handling Events and Action Messages

- [attachPopUpWithFrame:inView:](#) (page 11)
Sets up the receiver to display a menu.
- [dismissPopUp](#) (page 12)
Dismisses the pop-up button's menu by ordering its window out.
- [performClickWithFrame:inView:](#) (page 20)
Displays the receiver's menu and track mouse events in it.

Instance Methods

addItemWithTitles:

Adds multiple items to the end of the menu.

- (void)addItemWithTitles:(NSArray *)*itemTitles*

Parameters*itemTitles*

An array of `NSString` objects containing the titles of the items you want to add. Each string in the array should be unique. If an item with the same title already exists in the menu, the existing item is removed and the new one is added.

Discussion

The new menu items use the pop-up button's default action and target, but you can change these using the `setAction:` and `setTarget:` methods of the corresponding `NSMenuItem` object.

If you want to move an item, it's better to invoke `removeItemWithTitle:` (page 22) explicitly and then send this method. After adding the items, this method uses the `synchronizeTitleAndSelectedItem` (page 30) method to make sure the item being displayed matches the currently selected item.

Since this method searches for duplicate items, it should not be used if you are adding items to an already populated menu with more than a few hundred items. Add items directly to the receiver's menu instead.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addItemWithTitle:](#) (page 9)
- `setAction:` (`NSMenuItem`)
- `setTarget:` (`NSMenuItem`)

Declared In

`NSPopUpButtonCell.h`

addItemWithTitle:

Adds an item with the specified title to the end of the menu.

```
-(void)addItemWithTitle:(NSString *)title
```

Parameters*title*

The title of the new menu item. If an item with the same title already exists in the menu, the existing item is removed and the new one is added.

Discussion

The menu item uses the pop-up button's default action and target, but you can change these using the `setAction:` and `setTarget:` methods of the corresponding `NSMenuItem` object.

Since this method searches for duplicate items, it should not be used if you are adding an item to an already populated menu with more than a few hundred items. Add items directly to the button's menu instead.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [addItemWithTitle:](#) (page 8)
- `setAction:` (`NSMenuItem`)
- `setTarget:` (`NSMenuItem`)

Declared In

NSPopUpButtonCell.h

altersStateOfSelectedItem

Returns a Boolean value indicating whether the receiver links the state of the selected menu item to the current selection.

- (BOOL)altersStateOfSelectedItem

Return Value

YES if the selected menu item has its state set to `NSOnState` automatically; otherwise, NO if the state of menu items is independent of the current selection.

Discussion

This option is usually used only by pop-up menus. You typically do not alter the state of menu items in a pull-down menu.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [selectItemAtIndex:](#) (page 23)
- [selectItemWithTitle:](#) (page 25)

Declared In

NSPopUpButtonCell.h

arrowPosition

Returns the position of the arrow displayed on the receiver.

- (NSPopUpArrowPosition)arrowPosition

Return Value

The arrow position.

Discussion

`NSPopUpNoArrow` means no arrow is displayed. `NSPopUpArrowAtCenter` means the arrow is vertically centered, pointing to the right, vertically centered. `NSPopUpArrowAtBottom` means the arrow is at the bottom, pointing downward.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setArrowPosition:](#) (page 26)

Declared In

NSPopUpButtonCell.h

attachPopUpWithFrame:inView:

Sets up the receiver to display a menu.

```
- (void)attachPopUpWithFrame:(NSRect)cellFrame inView:(NSView *)controlView
```

Parameters

cellFrame

The cell's rectangle, specified in points in the coordinate system of the view in the *controlView* parameter. The menu is attached to this rectangle.

controlView

The view in which to display the pop-up button's menu.

Discussion

This call sets up the popup button cell to display a menu, which occurs in [performClickWithFrame:inView:](#) (page 20). This method sets the cell's control view and then highlights and redraws the cell. It does not show the menu.

This method also posts an [NSPopUpButtonCellWillPopUpNotification](#) (page 32). (The [NSPopUpButton](#) object sends a corresponding [NSPopUpButtonWillPopUpNotification](#).)

You normally do not call this method explicitly. It is called by the Application Kit automatically when the menu for the pop-up button is to be displayed.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [dismissPopUp](#) (page 12)

Declared In

[NSPopUpButtonCell.h](#)

autoenablesItems

Returns whether the receiver automatically enables and disables its items every time a user event occurs.

```
- (BOOL)autoenablesItems
```

Return Value

YES if the receiver automatically enables and disables items; otherwise, NO. The default value is YES.

Discussion

For more information on enabling and disabling menu items, see the [NSMenuValidation](#) .

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAutoenablesItems:](#) (page 26)

Declared In

[NSPopUpButtonCell.h](#)

dismissPopUp

Dismisses the pop-up button's menu by ordering its window out.

- (void)dismissPopUp

Discussion

If the pop-up button was not displaying its menu, this method does nothing.

You normally do not call this method explicitly. It is called by the Application Kit automatically to dismiss the menu for the pop-up button.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [attachPopUpWithFrame:inView:](#) (page 11)
- [orderOut:](#) (NSWindow)

Declared In

NSPopUpButtonCell.h

indexOfItem:

Returns the index of the specified menu item.

- (NSInteger)indexOfItem:(NSMenuItem *)*item*

Parameters

item

The menu item whose index you want.

Return Value

The index of the item or -1 if no such item was found.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItemWithRepresentedObject:](#) (page 12)
- [indexOfItemWithTag:](#) (page 13)
- [indexOfItemWithTarget:andAction:](#) (page 14)
- [indexOfItemWithTitle:](#) (page 14)
- [indexOfSelectedItem](#) (page 15)

Declared In

NSPopUpButtonCell.h

indexOfItemWithRepresentedObject:

Returns the index of the menu item that holds the specified represented object.

- (NSInteger)indexOfItemWithRepresentedObject:(id)*obj*

Parameters*obj*

The represented object associated with a menu item.

Return Value

The index of the menu item that owns the specified object, or -1 if no such menu item was found.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItem:](#) (page 12)
- [indexOfItemWithTag:](#) (page 13)
- [indexOfItemWithTarget:andAction:](#) (page 14)
- [indexOfItemWithTitle:](#) (page 14)
- [indexOfSelectedItem](#) (page 15)
- `representedObject` (NSMenuItem)
- `setRepresentedObject:` (NSMenuItem)

Declared In

NSPopUpButtonCell.h

indexOfItemWithTag:

Returns the index of the menu item with the specified tag.

```
- (NSInteger)indexOfItemWithTag:(NSInteger)tag
```

Parameters*tag*

The tag of the menu item you want.

Return Value

The index of the item or -1 if no item with the specified tag was found.

Discussion

Tags are values your application assigns to an object to identify it. You can assign tags to menu items using the `setTag:` method of `NSMenuItem`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItem:](#) (page 12)
- [indexOfItemWithRepresentedObject:](#) (page 12)
- [indexOfItemWithTarget:andAction:](#) (page 14)
- [indexOfItemWithTitle:](#) (page 14)
- [indexOfSelectedItem](#) (page 15)
- `setTag:` (NSMenuItem)

Declared In

NSPopUpButtonCell.h

indexOfItemWithTarget:andAction:

Returns the index of the menu item with the specified target and action.

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

Parameters

target

The target object associated with the menu item.

actionSelector

The action method associated with the menu item.

Return Value

The index of the menu item, or -1 if no menu item contains the specified target and action.

Discussion

If you specify `NULL` for the *actionSelector* parameter, the index of the first menu item with the specified target is returned.

The `NSPopUpButtonCell` class assigns a default action and target to each menu item, but you can change these values using the `setAction:` and `setTarget:` methods of `NSMenuItem`.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItem:](#) (page 12)
- [indexOfItemWithRepresentedObject:](#) (page 12)
- [indexOfItemWithTag:](#) (page 13)
- [indexOfItemWithTarget:andAction:](#) (page 14)
- [indexOfItemWithTitle:](#) (page 14)
- [indexOfSelectedItem](#) (page 15)
- `setAction:` (`NSMenuItem`)
- `setTarget:` (`NSMenuItem`)

Declared In

`NSPopUpButtonCell.h`

indexOfItemWithTitle:

Returns the index of the item with the specified title.

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

Parameters

title

The title of the item you want. You must not pass `nil` for this parameter.

Return Value

The index of the item or -1 if no item with the specified title was found.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItem:](#) (page 12)
- [indexOfItemWithRepresentedObject:](#) (page 12)
- [indexOfItemWithTag:](#) (page 13)
- [indexOfItemWithTarget:andAction:](#) (page 14)
- [indexOfItemWithTitle:](#) (page 14)
- [indexOfSelectedItem](#) (page 15)

Declared In

NSPopUpButtonCell.h

indexOfSelectedItem

Returns the index of the item last selected by the user.

- (NSInteger)indexOfSelectedItem

Return Value

The index of the selected item, or -1 if no item is selected.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [indexOfItem:](#) (page 12)
- [indexOfItemWithRepresentedObject:](#) (page 12)
- [indexOfItemWithTag:](#) (page 13)
- [indexOfItemWithTarget:andAction:](#) (page 14)
- [indexOfItemWithTitle:](#) (page 14)

Declared In

NSPopUpButtonCell.h

initWithCell:pullsDown:

Returns an NSPopUpButtonCell object initialized with the specified title.

- (id)initWithCell:(NSString *)stringValue pullsDown:(BOOL)pullsDown

Parameters*stringValue*

The title of the first menu. You may specify an empty string if you do not want to add an initial menu item.

pullsDown

YES if you want the receiver to display a pull-down menu; otherwise, NO if you want it to display a pop-up menu.

Return Value

An initialized NSPopUpButtonCell object, or nil if the object could not be initialized.

Discussion

This menu item is assigned the default pop-up button action that displays the menu. To set the action and target, use the `setAction:` and `setTarget:` methods of the item's corresponding `NSMenuItem` object.

This method is the designated initializer of the class.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `setAction:` (`NSMenuItem`)
- `setTarget:` (`NSMenuItem`)

Related Sample Code

QTAudioExtractionPanel

Declared In

NSPopUpButtonCell.h

insertItemWithTitle:atIndex:

Inserts an item at the specified position in the menu.

```
- (void)insertItemWithTitle:(NSString *)title atIndex:(NSInteger)index
```

Parameters

title

The title of the new item. If an item with the same title already exists in the menu, the existing item is removed and the new one is added

index

The zero-based index at which to insert the item. Specifying 0 inserts the item at the top of the menu.

Discussion

The value in *index* must represent a valid position in the array. The menu item at *index* and all those that follow it are shifted down one slot to make room for the new menu item.

This method assigns the pop-up button's default action and target to the new menu item. Use the menu item's `setAction:` and `setTarget:` methods to assign a new action and target.

Since this method searches for duplicate items, it should not be used if you are adding an item to an already populated menu with more than a few hundred items. Add items directly to the button's menu instead.

Availability

Available in Mac OS X v10.0 and later.

See Also

- ```
insertObject:atIndex: (NSMutableArray)
```
- `setAction:` (`NSMenuItem`)
  - `setTarget:` (`NSMenuItem`)

**Declared In**

NSPopUpButtonCell.h



## itemArray

Returns the items in the menu.

- (NSArray \*)itemArray

### Return Value

An array of `NSMenuItem` objects representing the items in the menu.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- `itemArray` (NSMenu)

### Declared In

`NSPopUpButtonCell.h`

## itemAtIndex:

Returns the menu item at the specified index.

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

### Parameters

*index*

The index of the item you want. The specified index must refer to an existing menu item.

### Return Value

The menu item, or `nil` if no item exists at the specified index.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [itemTitleAtIndex:](#) (page 17)

- `itemAtIndex:` (NSMenu)

### Declared In

`NSPopUpButtonCell.h`

## itemTitleAtIndex:

Returns the title of the item at the specified index.

- (NSString \*)itemTitleAtIndex:(NSInteger) *index*

### Parameters

*index*

The index of the item you want.

### Return Value

The title of the item, or an empty string if no item exists at the specified index.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [itemAtIndex:](#) (page 17)

**Declared In**

NSPopUpButtonCell.h

**itemTitles**

Returns the titles of all of the items in the menu.

- (NSArray \*)itemTitles

**Return Value**

An array of NSString objects containing the titles of every item in the menu. The titles appear in the order in which the items appear in the menu.

**Discussion**

If the menu contains separator items, the array contains an empty string (@"") for each separator item.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [itemTitleAtIndex:](#) (page 17)

**Declared In**

NSPopUpButtonCell.h

**itemWithTitle:**

Returns the menu item with the specified title.

- (NSMenuItem \*)itemWithTitle:(NSString \*)title

**Parameters**

*title*

The title of the menu item you want.

**Return Value**

The menu item, or nil if no item with the specified title exists in the menu.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [itemTitleAtIndex:](#) (page 17)

- [itemAtIndex:](#) (page 17)

**Declared In**

NSPopUpButtonCell.h

## lastItem

Returns the last item in the menu.

- (NSMenuItem \*)lastItem

### Return Value

The last menu item.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

NSPopUpButtonCell.h

## menu

Returns the pop-up button's associated menu.

- (NSMenu \*)menu

### Return Value

The menu for the pop-up button.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setMenu:](#) (page 27)

### Declared In

NSPopUpButtonCell.h

## numberOfItems

Returns the number of items in the menu.

- (NSInteger)numberOfItems

### Return Value

The number of items in the menu.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

[count \(NSArray\)](#)

### Related Sample Code

Quartz Composer WWDC 2005 TextEdit  
TextEditPlus

### Declared In

NSPopUpButtonCell.h

## objectValue

Returns the index of the selected item.

- (id)objectValue

### Return Value

An object (typically an `NSNumber` object) that responds to the `intValue` message and contains the index of the selected item.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [setObjectValue:](#) (page 27)

### Declared In

`NSPopUpButtonCell.h`

## performClickWithFrame:inView:

Displays the receiver's menu and track mouse events in it.

- (void)performClickWithFrame:(NSRect)frame inView:(NSView \*)controlView

### Parameters

*frame*

The cell's rectangle, specified in points in the coordinate system of the view in the *controlView* parameter.

*controlView*

The view in which to display the pop-up button's menu.

### Discussion

You normally do not call this method explicitly. It is called by the Application Kit automatically to handle events in the pop-up button.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [attachPopUpWithFrame:inView:](#) (page 11)

### Declared In

`NSPopUpButtonCell.h`

## preferredEdge

Returns the edge of the receiver next to which the pop-up menu is displayed under restrictive screen conditions.

- (NSRectEdge)preferredEdge

### Return Value

Possible values include `NSMinXEdge`, `NSMinYEdge`, `NSMaxXEdge`, or `NSMaxYEdge`. If no preferred edge was explicitly set, the default value is the bottom edge, which is `NSMaxYEdge` for flipped views or `NSMinYEdge` for unflipped views.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setPreferredEdge:](#) (page 28)

### Declared In

`NSPopUpButtonCell.h`

## pullsDown

Returns a Boolean value indicating the behavior of the control's menu.

- (BOOL)pullsDown

### Return Value

YES if the menu behaves like a pull-down menu; otherwise, NO if it behaves like a pop-up menu.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setPullsDown:](#) (page 28)

### Declared In

`NSPopUpButtonCell.h`

## removeAllItems

Removes all items in the receiver's item menu.

- (void)removeAllItems

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [removeItemAtIndex:](#) (page 22)
- [removeItemWithTitle:](#) (page 22)
- [insertItemWithTitle:atIndex:](#) (page 16)

### Declared In

`NSPopUpButtonCell.h`

## removeItemAtIndex:

Removes the item at the specified index.

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

### Parameters

*index*

The zero-based index indicating which item to remove. Specifying 0 removes the item at the top of the menu. The index must be valid and non-negative.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [removeAllItems](#) (page 21)
- [removeItemWithTitle:](#) (page 22)
- [insertItemWithTitle:atIndex:](#) (page 16)

### Declared In

NSPopUpButtonCell.h

## removeItemWithTitle:

Removes the item with the specified title from the menu.

```
- (void)removeItemWithTitle:(NSString *) title
```

### Parameters

*title*

The title of the item you want to remove. If no menu item exists with the specified title, this method triggers an assertion.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [removeAllItems](#) (page 21)
- [removeItemAtIndex:](#) (page 22)
- [insertItemWithTitle:atIndex:](#) (page 16)

### Declared In

NSPopUpButtonCell.h

## selectedItem

Returns the menu item last selected by the user.

```
- (NSMenuItem *)selectedItem
```

### Return Value

The menu item that is currently selected, or `nil` if no item is selected.

**Discussion**

The last selected menu item is the one that was highlighted when the user released the mouse button. It is possible for a pull-down menu's selected item to be its first item.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [selectItem:](#) (page 23)
- [selectItemAtIndex:](#) (page 23)
- [selectItemWithTitle:](#) (page 25)

**Declared In**

NSPopUpButtonCell.h

**selectItem:**

Selects the specified menu item.

```
- (void)selectItem:(NSMenuItem *)item
```

**Parameters**

*item*

The menu item to select, or `nil` if you want to deselect all menu items.

**Discussion**

By default, selecting or deselecting a menu item from a pop-up menu changes its state. Selecting a menu item from a pull-down menu does not automatically alter the state of the item. Use the [setAltersStateOfSelectedItem:](#) (page 25) method, passing it a value of `NO`, to disassociate the current selection from the state of menu items.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [selectedItem](#) (page 22)
- [selectItemAtIndex:](#) (page 23)
- [selectItemWithTitle:](#) (page 25)
- [setAltersStateOfSelectedItem:](#) (page 25)
- [setState:](#) (NSMenuItem)

**Declared In**

NSPopUpButtonCell.h

**selectItemAtIndex:**

Selects the item in the menu at the specified index.

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

**Parameters***index*

The index of the item you want to select, or `-1` you want to deselect all menu items.

**Discussion**

By default, selecting or deselecting a menu item from a pop-up menu changes its state. Selecting a menu item from a pull-down menu does not automatically alter the state of the item. Use the [setAltersStateOfSelectedItem:](#) (page 25) method, passing it a value of `NO`, to disassociate the current selection from the state of menu items.

Subclassers can override this method to catch all select calls.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [selectedItem](#) (page 22)
- [selectItem:](#) (page 23)
- [selectItemWithTitle:](#) (page 25)
- [setAltersStateOfSelectedItem:](#) (page 25)
- [setState:](#) (NSMenuItem)

**Related Sample Code**

Quartz Composer WWDC 2005 TextEdit  
TextEditPlus

**Declared In**

NSPopUpButtonCell.h

**selectItemWithTag:**

Selects the menu item with the specified tag.

```
- (BOOL)selectItemWithTag:(NSInteger)tag
```

**Parameters***tag*

The tag of the item you want to select.

**Return Value**

YES if the item was successfully selected; otherwise, NO.

**Discussion**

If no item with the specified tag is found, this method returns `NO` and leaves the menu state unchanged.

You typically assign tags to menu items from Interface Builder, but you can also assign them programmatically using the [setTag:](#) method of `NSMenuItem`.

**Availability**

Available in Mac OS X v10.4 and later.

**Declared In**

NSPopUpButtonCell.h



## selectItemWithTitle:

Selects the item with the specified title.

```
- (void)selectItemWithTitle:(NSString *)title
```

### Parameters

*title*

The title of the item to select. If you specify `nil`, an empty string, or a string that does not match the title of a menu item, this method deselects the currently selected item.

### Discussion

By default, selecting or deselecting a menu item changes its state. Use the

[setAltersStateOfSelectedItem:](#) (page 25) method, passing it a value of `NO`, to disassociate the current selection from the state of menu items.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [selectedItem](#) (page 22)
- [selectItem:](#) (page 23)
- [selectItemAtIndex:](#) (page 23)
- [setAltersStateOfSelectedItem:](#) (page 25)
- `setState:` (NSMenuItem)

### Declared In

NSPopUpButtonCell.h

## setAltersStateOfSelectedItem:

Sets whether the receiver links the state of the menu items to the current selection.

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

### Parameters

*flag*

YES if the selected menu item has its state set to `NSOnState` automatically; otherwise, NO if the state of menu items is independent of the current selection.

### Discussion

You use this method to control whether the selected menu item is linked to the state of that item. When you specify `NO` for the *flag* parameter, this method sets the state of the currently selected item to `NSOffState`.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [altersStateOfSelectedItem](#) (page 10)
- [selectedItem](#) (page 22)
- [selectItem:](#) (page 23)
- [selectItemAtIndex:](#) (page 23)
- `setState:` (NSMenuItem)

**Declared In**

NSPopUpButtonCell.h

**setArrowPosition:**

Sets the position of the arrow displayed on the receiver.

- (void)setArrowPosition:(NSPopUpArrowPosition)*position***Parameters***position*

The position of the arrow.

**Discussion**

If you specify `NSPopUpNoArrow`, the receiver displays no arrow. `NSPopUpArrowAtCenter` displays the arrow centered horizontally within the cell. `NSPopUpArrowAtBottom` displays the arrow at the edge of the cell.

This method works with `setPreferredEdge:` to determine the exact location and orientation of the arrow. For more information, see [setPreferredEdge:](#) (page 28).

This method is ignored unless the receiver is a pull-down list with a beveled border.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [arrowPosition](#) (page 10)**Declared In**

NSPopUpButtonCell.h

**setAutoenablesItems:**

Sets whether the receiver automatically enables and disables its items every time a user event occurs.

- (void)setAutoenablesItems:(BOOL)*flag***Parameters***flag*

YES if you want the receiver to automatically enable and disable items; otherwise, NO.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [autoenablesItems](#) (page 11)**Declared In**

NSPopUpButtonCell.h

**setImage:**

This method has no effect.

```
- (void)setImage:(NSImage *)anImage
```

**Parameters**

*anImage*

The image to display.

**Discussion**

The image displayed in a pop up button is taken from the selected menu item (in the case of a pop up menu) or from the first menu item (in the case of a pull-down menu).

**setMenu:**

Sets the pop-up button's associated menu.

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

**Parameters**

*menu*

The menu to associate with the pop-up button.

**Discussion**

If another menu was already associated with the pop-up button, this method releases the old menu. If you want to explicitly save the old menu, you should retain it before invoking this method.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [menu](#) (page 19)

**Related Sample Code**

QTAudioExtractionPanel

**Declared In**

NSPopUpButtonCell.h

**setObjectValue:**

Selects the item at a specific index using an object value.

```
- (void)setObjectValue:(id)object
```

**Parameters**

*object*

An `NSNumber` object containing the index (an integer) of the item you want to select. Specify the index -1 to deselect all items. You can also use an object other than an `NSNumber` object. In that case, the object must respond to the `intValue` message and return an appropriate index value.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [objectValue](#) (page 20)

**Declared In**

NSPopUpButtonCell.h

**setPreferredEdge:**

Sets the edge of the receiver next to which the pop-up menu should appear under restrictive screen conditions.

- (void)setPreferredEdge:(NSRectEdge)*edge***Parameters***edge*

The preferred edge. Possible values include NSMinXEdge, NSMinYEdge, NSMaxXEdge, or NSMaxYEdge.

**Discussion**

At display time, if attaching the menu to the preferred edge would cause part of the menu to be obscured, the pop-up button may use a different edge. If no preferred edge is set, the pop-up button uses the bottom edge by default.

This method works with [setArrowPosition:](#) (page 26) to determine the exact location of the arrow:

- If the arrow position is NSPopUpArrowAtCenter, the arrow stays in the center of the button and this method determines which edge the arrow points to. NSMinXEdge points to the left, NSMaxYEdge points to the top, NSMaxXEdge points to the right, and NSMinYEdge points to the bottom.
- If the arrow position is NSPopUpArrowAtBottom, this method determines which edge the arrow is at. NSMinXEdge places the arrow at the center of the left side, pointing to the left. NSMinYEdge places the arrow at bottom right corner, pointing up. NSMaxXEdge places the arrow at the center of the right side, pointing to the right. NSMaxYEdge places the arrow at the bottom right corner, pointing down.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [preferredEdge](#) (page 20)**Declared In**

NSPopUpButtonCell.h

**setPullsDown:**

Sets whether the receiver behaves as a pull-down or pop-up menu.

- (void)setPullsDown:(BOOL)*flag***Parameters***flag*

YES if you want the receiver to operate as a pull-down menu; otherwise, NO if you want it to operate as a pop-up menu.

**Discussion**

This method does not change the contents of the menu; it changes only the style of the menu.

When changing the menu type to a pull-down menu, if the menu was a pop-up menu and the cell alters the state of its selected items, this method sets the state of the currently selected item to `NSOffState` before changing the menu type.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [pullsDown](#) (page 21)
- [synchronizeTitleAndSelectedItem](#) (page 30)

#### Declared In

NSPopUpButtonCell.h

### setTitle:

Sets the string displayed in the receiver when the user isn't pressing the mouse button.

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

#### Parameters

*aString*

The string to display.

#### Discussion

For pull-down menus that get their titles from a menu item, this method simply sets the pop-up button cell's menu item to the first item in the menu. For pop-up menus, if a menu item whose title matches *aString* exists, this method makes that menu item the current selection; otherwise, it creates a new menu item with the title *aString*, adds it to the pop-up menu, and selects it.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [initWithCell:pullsDown:](#) (page 15)

#### Declared In

NSPopUpButtonCell.h

### setUsesItemFromMenu:

Sets whether the pop-up button uses an item from the menu for its own title.

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

#### Parameters

*flag*

YES if the button should use the first menu item as its own title; otherwise, NO. YES is the default value.

**Discussion**

For pop-up menus, the pop-up button uses the title of the currently selected menu item; if no menu item is selected, the pop-up button displays no item and is drawn empty. You can set the title or image of the pop-up button to something permanent by first calling this method (with a parameter of NO) and then calling `setMenuItem:`, as shown in the following example:

```
- (void)awakeFromNib {
 NSString *buttonImagePath = [[NSBundle mainBundle] pathForResource:@"plane"
ofType:@"png"];
 NSImage *buttonImage = [[NSImage alloc]
initWithContentsOfFile:buttonImagePath];
 NSMenuItem *menuItem = [[NSMenuItem alloc] init];
 [menuItem setImage:buttonImage];
 [menuItem setTitle:@"City"];
 [[myPopUpButton cell] setUsesItemFromMenu:NO];
 [[myPopUpButton cell] setMenuItem:menuItem];
 [buttonImage release];
 [menuItem release];
}
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [usesItemFromMenu](#) (page 31)

**Declared In**

NSPopUpButtonCell.h

**synchronizeTitleAndSelectedItem**

Synchronizes the the pop-up button's displayed item with the currently selected menu item.

```
- (void)synchronizeTitleAndSelectedItem
```

**Discussion**

If no item is currently selected, this method synchronizes the pop-up buttons displayed item with the first menu item. If the pop-up button cell does not get its displayed item from a menu item, this method does nothing.

For pull-down menus, this method sets the displayed item to the title first menu item.

If the pop-up button's menu does not contain any menu items, this method sets the pop-up button's displayed item to `nil`, resulting in nothing being displayed in the control.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPopUpButtonCell.h

## titleOfSelectedItem

Returns the title of the item last selected by the user.

- (NSString \*)titleOfSelectedItem

### Return Value

The title of the selected menu item, or an empty string if no item is selected.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSPopUpButtonCell.h

## usesItemFromMenu

Returns a Boolean value indicating whether the pop-up button uses an item from the menu for its own title.

- (BOOL)usesItemFromMenu

### Return Value

YES if the button uses the first menu item as its own title; otherwise, NO. YES is the default value.

### Discussion

If this option is set, pull-down menus use the title of the first menu item, while pop-up menus use the title of the currently selected menu.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

### Declared In

NSPopUpButtonCell.h

## Constants

### NSPopUpArrowPosition

These constants are defined for use with the [arrowPosition](#) (page 10) and [setArrowPosition:](#) (page 26) methods.

```
typedef enum {
 NSPopUpNoArrow = 0,
 NSPopUpArrowAtCenter = 1,
 NSPopUpArrowAtBottom = 2
} NSPopUpArrowPosition;
```

**Constants**

NSPopUpNoArrow

Does not display any arrow in the receiver.

Available in Mac OS X v10.0 and later.

Declared in NSPopUpButtonCell.h.

NSPopUpArrowAtCenter

Arrow is centered vertically, pointing toward the [preferredEdge](#) (page 20).

Available in Mac OS X v10.0 and later.

Declared in NSPopUpButtonCell.h.

NSPopUpArrowAtBottom

Arrow is drawn at the edge of the button, pointing toward the [preferredEdge](#) (page 20).

Available in Mac OS X v10.0 and later.

Declared in NSPopUpButtonCell.h.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPopUpButtonCell.h

## Notifications

**NSPopUpButtonCellWillPopUpNotification**

This notification is posted just before a pop-up menu is attached to its window frame. You can use this notification to lazily construct your part's menus, thus preventing unnecessary calculations until they are needed. The notification object can be either a pop-up button or its enclosed pop-up button cell. This notification does not contain a *userInfo* dictionary.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSPopUpButtonCell.h



# Document Revision History

---

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

| Date       | Notes                                                                                                                                                                                                          |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2008-10-15 | Replaced parameter types of "id <NSMenuItem>" with "NSMenuItem *".<br>Corrected description of <code>attachPopUpWithFrame:inView:</code> method. Included code example for <code>setUsesItemFromMenu:</code> . |
| 2006-05-23 | First publication of this content as a separate document.                                                                                                                                                      |

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

addItemWithTitle: [instance method 8](#)  
addItemWithTitle: [instance method 9](#)  
altersStateOfSelectedItem [instance method 10](#)  
arrowPosition [instance method 10](#)  
attachPopUpWithFrame:inView: [instance method 11](#)  
autoenablesItems [instance method 11](#)

## D

---

dismissPopUp [instance method 12](#)

## I

---

indexOfItem: [instance method 12](#)  
indexOfItemWithRepresentedObject: [instance method 12](#)  
indexOfItemWithTag: [instance method 13](#)  
indexOfItemWithTarget:andAction: [instance method 14](#)  
indexOfItemWithTitle: [instance method 14](#)  
indexOfSelectedItem [instance method 15](#)  
initTextCell:pullsDown: [instance method 15](#)  
insertItemWithTitle:atIndex: [instance method 16](#)  
itemArray [instance method 17](#)  
itemAtIndex: [instance method 17](#)  
itemTitleAtIndex: [instance method 17](#)  
itemTitles [instance method 18](#)  
itemWithTitle: [instance method 18](#)

## L

---

lastItem [instance method 19](#)

## M

---

menu [instance method 19](#)

## N

---

NSPopUpArrowAtBottom [constant 32](#)  
NSPopUpArrowAtCenter [constant 32](#)  
NSPopUpArrowPosition [data type 31](#)  
NSPopUpButtonCellWillPopUpNotification [notification 32](#)  
NSPopUpNoArrow [constant 32](#)  
numberOfItems [instance method 19](#)

## O

---

objectValue [instance method 20](#)

## P

---

performClickWithFrame:inView: [instance method 20](#)  
preferredEdge [instance method 20](#)  
pullsDown [instance method 21](#)

## R

---

removeAllItems [instance method 21](#)  
removeItemAtIndex: [instance method 22](#)  
removeItemWithTitle: [instance method 22](#)

## S

---

selectedItem [instance method 22](#)

`selectItemAtIndex`: instance method [23](#)  
`selectItem`: instance method [23](#)  
`selectItemWithTag`: instance method [24](#)  
`selectItemWithTitle`: instance method [25](#)  
`setAltersStateOfSelectedItem`: instance method [25](#)  
`setArrowPosition`: instance method [26](#)  
`setAutoenablesItems`: instance method [26](#)  
`setImage`: instance method [26](#)  
`setMenu`: instance method [27](#)  
`setObjectValue`: instance method [27](#)  
`setPreferredEdge`: instance method [28](#)  
`setPullsDown`: instance method [28](#)  
`setTitle`: instance method [29](#)  
`setUsesItemFromMenu`: instance method [29](#)  
`synchronizeTitleAndSelectedItem`: instance method [30](#)

## T

---

`titleOfSelectedItem`: instance method [31](#)

## U

---

`usesItemFromMenu`: instance method [31](#)