
NSToolbar 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

NSToolbar Class Reference 5

Overview	5
Tasks	6
Creating an NSToolbar Object	6
Toolbar Attributes	6
Managing the Delegate	6
Managing Items on the Toolbar	7
Displaying the Toolbar	7
Toolbar Customization	7
Autosaving the Configuration	7
Working With Item Identifiers	8
Instance Methods	8
allowsUserCustomization	8
autosavesConfiguration	8
configurationDictionary	9
customizationPalettesRunning	9
delegate	10
displayMode	10
identifier	10
initWithIdentifier:	11
insertItemWithItemIdentifier:atIndex:	11
isVisible	12
items	12
removeItemAtIndex:	13
runCustomizationPalette:	13
selectedItemIdentifier	13
setAllowsUserCustomization:	14
setAutosavesConfiguration:	14
setConfigurationFromDictionary:	15
setDelegate:	15
setDisplayMode:	16
setSelectedItemIdentifier:	16
setShowsBaselineSeparator:	17
setSizeMode:	17
setVisible:	18
showsBaselineSeparator	18
sizeMode	19
validateVisibleItems	19
visibleItems	19
Delegate Methods	20
toolbar:itemForItemIdentifier:willBeInsertedIntoToolbar:	20

toolbarAllowedItemIdentifiers: 21
toolbarDefaultItemIdentifiers: 21
toolbarDidRemoveItem: 22
toolbarSelectableItemIdentifiers: 22
toolbarWillAddItem: 23

Constants 23

NSToolbarDisplayMode 23

NSToolbarSizeMode 24

Notifications 24

NSToolbarDidRemoveItemNotification 24

NSToolbarWillAddItemNotification 25

Document Revision History 27

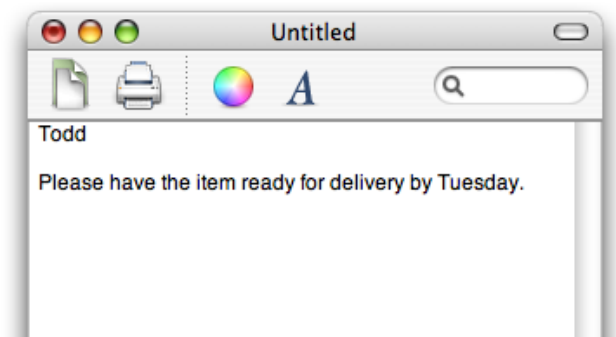
Index 29

NSToolbar Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Toolbar Programming Topics for Cocoa
Declared in	NSToolbar.h
Related sample code	EnhancedDataBurn GridCalendar iSpend Quartz Composer QCTV ToolbarSample

Overview

`NSToolbar` and `NSToolBarItem` provide the mechanism for a titled window to display a toolbar just below its title bar, as shown below:



Tasks

Creating an NSToolbar Object

- [initWithIdentifier:](#) (page 11)
Initializes a newly allocated toolbar with the specified identifier.

Toolbar Attributes

- [displayMode](#) (page 10)
Returns the receiver's display mode.
- [setDisplayMode:](#) (page 16)
Sets the receiver's display mode.
- [showsBaselineSeparator](#) (page 18)
Returns a Boolean value that indicates whether the toolbar shows the separator between the toolbar and the main window contents.
- [setShowsBaselineSeparator:](#) (page 17)
Sets whether the toolbar shows the separator between the toolbar and the main window contents.
- [allowsUserCustomization](#) (page 8)
Returns a Boolean value that indicates whether users are allowed to modify the toolbar.
- [setAllowsUserCustomization:](#) (page 14)
Sets whether users are allowed to modify the toolbar.
- [identifier](#) (page 10)
Returns the receiver's identifier.
- [items](#) (page 12)
Returns the receiver's current items, in order.
- [visibleItems](#) (page 19)
Returns the receiver's currently visible items.
- [sizeMode](#) (page 19)
Returns the receiver's size mode.
- [setSizeMode:](#) (page 17)
Sets the receiver's size mode.

Managing the Delegate

- [delegate](#) (page 10)
Returns the receiver's delegate.
- [setDelegate:](#) (page 15)
Sets the receiver's delegate.

Managing Items on the Toolbar

- [insertItemWithIdentifier:atIndex:](#) (page 11)
Inserts the specified item at the specified index.
- [removeItemAtIndex:](#) (page 13)
Removes the specified item.
- [toolbarWillAddItem:](#) (page 23) *delegate method*
Sent just before a new item is added to a toolbar.
- [toolbarDidRemoveItem:](#) (page 22) *delegate method*
Sent just after an item has been removed from a toolbar.
- [setSelectedItemIdentifier:](#) (page 16)
Sets the receiver's selected item to the specified toolbar item.
- [selectedItemIdentifier](#) (page 13)
Returns the identifier of the receiver's currently selected item, or `nil` if there is no selection.

Displaying the Toolbar

- [isVisible](#) (page 12)
Returns a Boolean value that indicates whether the receiver is visible.
- [setVisible:](#) (page 18)
Sets whether the receiver is visible or hidden.

Toolbar Customization

- [runCustomizationPalette:](#) (page 13)
Runs the receiver's customization palette.
- [customizationPaletteIsRunning](#) (page 9)
Returns a Boolean value that indicates whether the receiver's customization palette is running (in use).

Autosaving the Configuration

- [autosavesConfiguration](#) (page 8)
Returns a Boolean value that indicates whether the receiver autosaves its configuration.
- [setAutosavesConfiguration:](#) (page 14)
Sets whether the receiver autosaves its configuration.
- [configurationDictionary](#) (page 9)
Returns the receiver's configuration as a dictionary.
- [setConfigurationFromDictionary:](#) (page 15)
Sets the receiver's configuration using *configDict*.

Working With Item Identifiers

- [toolbar:itemForItemIdentifier:willBeInsertedIntoToolbar:](#) (page 20) *delegate method*
Sent to request a new toolbar item; returns a toolbar item of the identified kind for the specified toolbar.
- [toolbarAllowedItemIdentifiers:](#) (page 21) *delegate method*
Sent to discover the allowed item identifiers for a toolbar.
- [toolbarDefaultItemIdentifiers:](#) (page 21) *delegate method*
Sent to discover the default item identifiers for a toolbar.
- [toolbarSelectableItemIdentifiers:](#) (page 22) *delegate method*
Sent to discover the selectable item identifiers for a toolbar.
- [validateVisibleItems](#) (page 19)
Called on window updates to validate the visible items.

Instance Methods

allowsUserCustomization

Returns a Boolean value that indicates whether users are allowed to modify the toolbar.

- (BOOL)allowsUserCustomization

Return Value

YES if users are allowed to modify the toolbar, NO otherwise. The default is NO.

Discussion

If the value is NO, then the Customize Toolbar... menu item is disabled and other modification is disabled. This attribute does not affect the user's ability to show or hide the toolbar.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAllowsUserCustomization:](#) (page 14)
- [autosavesConfiguration](#) (page 8)

Declared In

NSToolbar.h

autosavesConfiguration

Returns a Boolean value that indicates whether the receiver autosaves its configuration.

- (BOOL)autosavesConfiguration

Return Value

YES if the receiver autosaves its configuration, otherwise NO. The default is NO.

Discussion

When autosaving is enabled, the receiver will automatically write the toolbar settings to user defaults if the toolbar configuration changes. The toolbar's configuration is identified in user defaults by the toolbar identifier. If there are multiple toolbars active with the same identifier, they all share the same configuration.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAutosavesConfiguration:](#) (page 14)
- [configurationDictionary](#) (page 9)

Declared In

NSToolbar.h

configurationDictionary

Returns the receiver's configuration as a dictionary.

- (NSDictionary *)configurationDictionary

Return Value

A dictionary containing configuration information for the toolbar.

Discussion

Contains `displayMode`, `isVisible`, and a list of the item identifiers currently in the toolbar.

Special Considerations

Do not depend on any details of the normal contents of a configuration dictionary.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setConfigurationFromDictionary:](#) (page 15)

Declared In

NSToolbar.h

customizationPalettelsRunning

Returns a Boolean value that indicates whether the receiver's customization palette is running (in use).

- (BOOL)customizationPaletteIsRunning

Return Value

YES if the receiver's customization palette is running, otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [runCustomizationPalette:](#) (page 13)

Declared In
NSToolbar.h

delegate

Returns the receiver's delegate.

- (id)delegate

Return Value
The receiver's delegate.

Discussion
Every toolbar must have a delegate, which must implement the required delegate methods.

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

See Also
- [setDelegate:](#) (page 15)

Declared In
NSToolbar.h

displayMode

Returns the receiver's display mode.

- (NSToolbarDisplayMode)displayMode

Return Value
The receiver's display mode.

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

See Also
- [setDisplayMode:](#) (page 16)

Declared In
NSToolbar.h

identifier

Returns the receiver's identifier.

- (NSString *)identifier

Return Value
The receiver's identifier, a string used by the class to identify the kind of toolbar.

Discussion

Within the application all toolbars with the same identifier are synchronized to maintain the same state, including for example, the display mode and item order. The identifier is used as the autosave name for toolbars that save their configuration.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `setAutosavesConfiguration:`

Declared In

`NSToolbar.h`

initWithIdentifier:

Initializes a newly allocated toolbar with the specified identifier.

```
- (id)initWithIdentifier:(NSString *)identifier
```

Parameters

identifier

A string used by the class to identify the kind of the toolbar.

Return Value

The initialized toolbar object.

Discussion

identifier is never seen by users and should not be localized. See [identifier](#) (page 10) for important information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [identifier](#) (page 10)

Declared In

`NSToolbar.h`

insertItemWithIdentifier:atIndex:

Inserts the specified item at the specified index.

```
- (void)insertItemWithIdentifier:(NSString *)itemIdentifier
    atIndex:(NSInteger)index
```

Parameters

itemIdentifier

The identifier of the item to insert.

index

The index at which to insert the item.

Discussion

If the toolbar needs a new instance, it will get it from [toolbar:itemForItemIdentifier:willBeInsertedIntoToolbar:](#) (page 20). Typically, you should not call this method; you should let the user reconfigure the toolbar. See [identifier](#) (page 10) for important information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [removeItemAtIndex:](#) (page 13)

Declared In

NSToolbar.h

isVisible

Returns a Boolean value that indicates whether the receiver is visible.

- (BOOL)isVisible

Return Value

YES if the receiver is visible, otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

NSToolbar.h

items

Returns the receiver's current items, in order.

- (NSArray *)items

Return Value

An array of the items in the toolbar.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [visibleItems](#) (page 19)

Declared In

NSToolbar.h

removeItemAtIndex:

Removes the specified item.

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

Parameters

index

The index of the item to remove.

Discussion

Typically, you should not call this method; you should let the user reconfigure the toolbar. See [identifier](#) (page 10) for important information.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [insertItemWithIdentifier:atIndex:](#) (page 11)

Declared In

NSToolbar.h

runCustomizationPalette:

Runs the receiver's customization palette.

```
- (void)runCustomizationPalette:(id) sender
```

Parameters

sender

The control sending the message.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [customizationPaletteIsRunning](#) (page 9)

Declared In

NSToolbar.h

selectedItemIdentifier

Returns the identifier of the receiver's currently selected item, or `nil` if there is no selection.

```
- (NSString *)selectedItemIdentifier
```

Return Value

The identifier of the receiver's currently selected item, or `nil` if there is no selection.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setSelectedItemIdentifier:](#) (page 16)

Declared In

NSToolbar.h

setAllowsUserCustomization:

Sets whether users are allowed to modify the toolbar.

- (void)setAllowsUserCustomization:(BOOL)*allowsCustomization*

Parameters

allowsCustomization

YES to allow users to modify the toolbar, NO otherwise.

Discussion

This value can be changed at any time. For instance, you may not want users to be able to customize the toolbar while some event is being processed. This attribute does not affect the user's ability to show or hide the toolbar.

If you set the toolbar to allow customization, be sure to also set the toolbar to autosave its configuration so the user's changes persist.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [allowsUserCustomization](#) (page 8)

- [setAutosavesConfiguration:](#) (page 14)

Related Sample Code

EnhancedDataBurn

iSpend

PDFKitLinker2

Quartz Composer QCTV

ToolbarSample

Declared In

NSToolbar.h

setAutosavesConfiguration:

Sets whether the receiver autosaves its configuration.

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

Parameters

flag

YES to indicate that the receiver should autosave its configuration, NO otherwise.

Discussion

Customizable toolbars should generally supporting autosaving. If you need to customize the saving behavior, you can use the [configurationDictionary](#) (page 9) to access the settings that should be saved.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [allowsUserCustomization](#) (page 8)
- [autosavesConfiguration](#) (page 8)

Related Sample Code

EnhancedDataBurn

iSpend

PDFKitLinker2

ToolbarSample

Declared In

NSToolbar.h

setConfigurationFromDictionary:

Sets the receiver's configuration using *configDict*.

```
- (void)setConfigurationFromDictionary:(NSDictionary *)configDict
```

Parameters

configDict

A dictionary with the toolbar's configuration information. If you want to provide a custom dictionary, you should first get the receiver's current configuration dictionary, then create a modified copy, rather than trying to construct one yourself.

Discussion

This method immediately affects toolbars with the same identifier in all windows of your application.

Special Considerations

Do not depend on any details of the normal contents of a configuration dictionary.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [configurationDictionary](#) (page 9)

Declared In

NSToolbar.h

setDelegate:

Sets the receiver's delegate.

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

Parameters*delegate*

The new delegate object.

Discussion

Every toolbar must have a delegate, which must implement the required delegate methods.

Availability

Available in Mac OS X v10.0 and later.

See Also- [delegate](#) (page 10)**Related Sample Code**

EnhancedDataBurn

iSpend

PDFKitLinker2

Quartz Composer QCTV

ToolbarSample

Declared In

NSToolbar.h

setDisplayMode:

Sets the receiver's display mode.

- (void)setDisplayMode:(NSToolbarDisplayMode)*displayMode***Parameters***displayMode*

The new display mode.

Availability

Available in Mac OS X v10.0 and later.

See Also- [displayMode](#) (page 10)**Related Sample Code**

EnhancedDataBurn

iSpend

PDFKitLinker2

Quartz Composer QCTV

ToolbarSample

Declared In

NSToolbar.h

setSelectedItemIdentifier:

Sets the receiver's selected item to the specified toolbar item.


```
- (void)setSelectedItemIdentifier:(NSString *)itemIdentifier
```

Parameters

itemIdentifier

The identifier of the item to select. *itemIdentifier* may be any identifier returned by `toolbarSelectableItemIdentifiers:`, even if it is not currently in the toolbar.

Discussion

Typically, a toolbar will manage the selection of items automatically. This method can be used to select identifiers of custom view items, or to force a selection change. See [toolbarSelectableItemIdentifiers:](#) (page 22) for more details. If *itemIdentifier* is not recognized by the receiver, the current selected item identifier does not change.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [selectedItemIdentifier](#) (page 13)
- [toolbarSelectableItemIdentifiers:](#) (page 22)

Declared In

NSToolbar.h

setShowsBaselineSeparator:

Sets whether the toolbar shows the separator between the toolbar and the main window contents.

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

Parameters

flag

YES if the toolbar should show the separator between the toolbar and the main window contents, otherwise NO.

Availability

Available in Mac OS X v10.4 and later.

See Also

- [showsBaselineSeparator](#) (page 18)

Declared In

NSToolbar.h

setSizeMode:

Sets the receiver's size mode.

```
- (void)setSizeMode:(NSToolbarSizeMode) sizeMode
```

Parameters

sizeMode

The new size mode.

Discussion

If there is no icon of the given size for a toolbar item, the toolbar item creates one by scaling an icon of another size.

Availability

Available in Mac OS X v10.2 and later.

See Also

- [sizeMode](#) (page 19)

Declared In

NSToolbar.h

setVisible:

Sets whether the receiver is visible or hidden.

- (void)setVisible:(BOOL)shown

Parameters

shown

YES to indicate the receiver should be made visible, NO to indicate it should be hidden.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [isVisible](#) (page 12)

Declared In

NSToolbar.h

showsBaselineSeparator

Returns a Boolean value that indicates whether the toolbar shows the separator between the toolbar and the main window contents.

- (BOOL)showsBaselineSeparator

Return Value

YES if the toolbar shows the separator between the toolbar and the main window contents, otherwise NO. The default is YES.

Availability

Available in Mac OS X v10.4 and later.

See Also

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

Declared In

NSToolbar.h

sizeMode

Returns the receiver's size mode.

```
- (NSToolbarSizeMode)sizeMode
```

Return Value

The receiver's size mode.

Availability

Available in Mac OS X v10.2 and later.

See Also

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

Declared In

NSToolbar.h

validateVisibleItems

Called on window updates to validate the visible items.

```
- (void)validateVisibleItems
```

Discussion

You typically use this method by overriding it in a subclass. The default implementation of this method iterates through the list of visible items, sending each a `validate` message. Override it and call `super` if you want to know when this method is called.

This method forces a validation of the entire visible set of items, so typically you should not invoke it directly. Instead, if necessary you should validate on a per item basis only those items that have auto-validation disabled (see `validate` and Validating Toolbar Items)—you generally do this for performance reasons if your validation code is slow.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSToolbar.h

visibleItems

Returns the receiver's currently visible items.

```
- (NSArray *)visibleItems
```

Return Value

An array of the toolbar's visible items.

Discussion

Items in the overflow menu are not considered visible.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [items](#) (page 12)

Declared In

NSToolbar.h

Delegate Methods

toolbar:itemForItemIdentifier:willBeInsertedIntoToolbar:

Sent to request a new toolbar item; returns a toolbar item of the identified kind for the specified toolbar.

```
- (NSToolbarItem *)toolbar:(NSToolbar *)toolbar itemForItemIdentifier:(NSString *)itemIdentifier willBeInsertedIntoToolbar:(BOOL)flag
```

Parameters

toolbar

The toolbar for which the item is being requested.

itemIdentifier

The identifier for the requested item.

flag

YES if the item will be immediately inserted into the toolbar. If *flag* is NO the toolbar item is being requested for display in the toolbar customization sheet and should always be enabled or provide some other canonical representation. If you ignore this parameter the same toolbar item will be used in the toolbar and in the customization sheet.

Return Value

The toolbar item for the specified toolbar and identifier. Return `nil` to indicate that the identified kind of toolbar item is not supported. When an item is requested again, you may return the same `NSToolbarItem` object returned earlier or a different instance.

Discussion

Implement this method to create new toolbar item instances. This method is called lazily on behalf of a toolbar instance, which must be the sole owner of the toolbar item. A toolbar may ask again for a kind of toolbar item already supplied to it, in which case this method may return the same toolbar item it returned before or a different one. If your delegate services multiple toolbars, each attached to a different window, it is best to return a different item for each toolbar—an `NSToolbarItem` object can only be in one toolbar at a time.

If the item is a custom view item, the `NSView` object must be fully formed when the item is returned. Do not assume that the returned item is going to be added as an active item in the toolbar, as it could be that it will be used only in the customization palette. (The customization palette makes a copy of the returned item.)

Implementation of this method is required.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSToolbar.h

toolbarAllowedItemIdentifiers:

Sent to discover the allowed item identifiers for a toolbar.

```
- (NSArray *)toolbarAllowedItemIdentifiers:(NSToolbar *)toolbar
```

Parameters

toolbar

The toolbar whose allowed item identifiers are to be returned.

Return Value

An array of toolbar item identifiers for *toolbar*, specifying the contents and the order of the items in the configuration palette.

Discussion

Every allowed item must be explicitly listed, even the standard ones. The identifiers returned should include all of those returned by `toolbarDefaultItemIdentifiers:`.

Implementation of this method is required.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSToolbar.h

toolbarDefaultItemIdentifiers:

Sent to discover the default item identifiers for a toolbar.

```
- (NSArray *)toolbarDefaultItemIdentifiers:(NSToolbar *)toolbar
```

Parameters

toolbar

The toolbar whose default item identifiers are to be returned.

Return Value

An array of toolbar item identifiers for *toolbar*, specifying the contents and the order of the items in the default toolbar configuration.

Discussion

During initialization of *toolbar*, this method is called only if a toolbar configuration for the identifier of *toolbar* is not found in the user preferences. This method is called during initialization of the toolbar customization palette.

Implementation of this method is required.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSToolbar.h

toolbarDidRemoveItem:

Sent just after an item has been removed from a toolbar.

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

Parameters

notification

The notification object sent to observers of [NSToolbarDidRemoveItemNotification](#) (page 24).

Discussion

This method allows you to remove information related to the item that may have been cached.

Implementation of this method is optional.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSToolbar.h

toolbarSelectableItemIdentifiers:

Sent to discover the selectable item identifiers for a toolbar.

```
- (NSArray *)toolbarSelectableItemIdentifiers:(NSToolbar *)toolbar
```

Parameters

toolbar

The toolbar whose selectable item identifiers are to be returned.

Return Value

An array of item identifiers that should indicate selection in the specified *toolbar*.

Discussion

Toolbars that need to indicate item selection should return an array containing the identifiers of the selectable toolbar items.

If implemented, *toolbar* will display the currently selected item with a visual highlight. Clicking on an item whose identifier is selectable will automatically update the toolbar's selected item identifier, when possible. Clicking an item whose identifier is not selectable will not update the toolbar's selected item identifier.

Implementation of this method is optional.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setSelectedItemIdentifier:](#) (page 16)

Declared In

NSToolbar.h

toolbarWillAddItem:

Sent just before a new item is added to a toolbar.

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

Parameters

notification

The notification object sent to observers of [NSToolbarWillAddItemNotification](#) (page 25).

Discussion

If you need to cache a reference to a toolbar item or need to set up some initial state before a toolbar item is added, this is where to do it.

Implementation of this method is optional.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSToolbar.h

Constants

NSToolbarDisplayMode

These constants specify toolbar display modes and are used by [displayMode](#) (page 10) and [setDisplayMode:](#) (page 16).

```
typedef enum {
    NSToolbarDisplayModeDefault,
    NSToolbarDisplayModeIconAndLabel,
    NSToolbarDisplayModeIconOnly,
    NSToolbarDisplayModeLabelOnly
} NSToolbarDisplayMode;
```

Constants

NSToolbarDisplayModeDefault

The default display mode.

Available in Mac OS X v10.0 and later.

Declared in NSToolbar.h.

NSToolbarDisplayModeIconAndLabel

The toolbar will display icons and labels.

Available in Mac OS X v10.0 and later.

Declared in NSToolbar.h.

NSToolbarDisplayModeIconOnly

The toolbar will display only icons.

Available in Mac OS X v10.0 and later.

Declared in NSToolbar.h.

`NSToolbarDisplayModeLabelOnly`

The toolbar will display only labels.

Available in Mac OS X v10.0 and later.

Declared in `NSToolbar.h`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSToolbar.h`

NSToolbarSizeMode

These constants specify toolbar display modes and are used by `sizeMode` (page 19) and `setSizeMode:` (page 17).

```
typedef enum {
    NSToolbarSizeModeDefault,
    NSToolbarSizeModeRegular,
    NSToolbarSizeModeSmall
} NSToolbarSizeMode;
```

Constants

`NSToolbarSizeModeDefault`

The toolbar uses the system-defined default size, which is `NSToolbarSizeModeRegular`.

Available in Mac OS X v10.2 and later.

Declared in `NSToolbar.h`.

`NSToolbarSizeModeRegular`

The toolbar uses regular-sized controls and 32 by 32 pixel icons.

Available in Mac OS X v10.2 and later.

Declared in `NSToolbar.h`.

`NSToolbarSizeModeSmall`

The toolbar uses small-sized controls and 24 by 24 pixel icons.

Available in Mac OS X v10.2 and later.

Declared in `NSToolbar.h`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`NSToolbar.h`

Notifications

NSToolbarDidRemoveItemNotification

Posted after an item is removed from a toolbar. The notification item is the `NSToolbar` object that had an item removed from it. The `userInfo` dictionary contains the following information:

Key	Value
@"item"	The NSToolBarItem object that was removed.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [toolbarDidRemoveItem:](#) (page 22)

Declared In

NSToolbar.h

NSToolbarWillAddItemNotification

Posted before a new item is added to the toolbar. The notification item is the NSToolbar object having an item added to it. The *userInfo* dictionary contains the following information:

Key	Value
@"item"	The NSToolBarItem object being added.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [toolbarWillAddItem:](#) (page 23)

Declared In

NSToolbar.h

Document Revision History

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

Date	Notes
2008-10-15	Corrected description of toolbar:itemForItemIdentifier: willBeInsertedIntoToolbar: method.
2006-06-28	Clarified the role of the insertion flag parameter in toolbar:itemForItemIdentifier: willBeInsertedIntoToolbar:.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

allowsUserCustomization **instance method** 8
autosavesConfiguration **instance method** 8

C

configurationDictionary **instance method** 9
customizationPaletteIsRunning **instance method** 9

D

delegate **instance method** 10
displayMode **instance method** 10

I

identifier **instance method** 10
initWithIdentifier: **instance method** 11
insertItemWithItemIdentifier:atIndex: **instance method** 11
isVisible **instance method** 12
items **instance method** 12

N

NSToolbarDidRemoveItemNotification **notification** 24
NSToolbarDisplayMode **data type** 23
NSToolbarDisplayModeDefault **constant** 23
NSToolbarDisplayModeIconAndLabel **constant** 23
NSToolbarDisplayModeIconOnly **constant** 23
NSToolbarDisplayModeLabelOnly **constant** 24
NSToolbarSizeMode **data type** 24

NSToolbarSizeModeDefault **constant** 24
NSToolbarSizeModeRegular **constant** 24
NSToolbarSizeModeSmall **constant** 24
NSToolbarWillAddItemNotification **notification** 25

R

removeItemAtIndex: **instance method** 13
runCustomizationPalette: **instance method** 13

S

selectedItemIdentifier **instance method** 13
setAllowsUserCustomization: **instance method** 14
setAutosavesConfiguration: **instance method** 14
setConfigurationFromDictionary: **instance method** 15
setDelegate: **instance method** 15
setDisplayMode: **instance method** 16
setSelectedItemIdentifier: **instance method** 16
setShowsBaselineSeparator: **instance method** 17
setSizeMode: **instance method** 17
setVisible: **instance method** 18
showsBaselineSeparator **instance method** 18
SizeMode **instance method** 19

T

toolbarAllowedItemIdentifiers: <NSObject> **delegate method** 21
toolbar:itemForItemIdentifier: willBeInsertedIntoToolbar: <NSObject> **delegate method** 20
toolbarDefaultItemIdentifiers: <NSObject> **delegate method** 21
toolbarDidRemoveItem: <NSObject> **delegate method** 22

toolbarSelectableItemIdentifiers: <NSObject>
 delegate method [22](#)
toolbarWillAddItem: <NSObject> **delegate method**
 [23](#)

V

validateVisibleItems **instance method** [19](#)
visibleItems **instance method** [19](#)