

[UIKit](#) / NSToolbar

## Class

# NSToolbar

An object that manages the space above your app's custom content and either below or integrated with the window's title bar.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.1+ | macOS

```
@MainActor
class NSToolbar
```

## Overview

An [NSToolbar](#) object manages the controls and views that apply to the main window's content area. Toolbars provide convenient access to the commands and features people use most often. Toolbars are also user-configurable and support the display of an interactive customization palette.

Create and configure your toolbar programmatically or using Interface Builder. Add items to the toolbar that correspond to the commands you want to feature in your window. Each item has a corresponding [NSToolbarItem](#) object, which you use to make changes. Each toolbar manages a unique set of items, but you can synchronize the items and state of multiple toolbars by assigning the same value to their [identifier](#) properties.

For more information about how to use toolbars, see [Integrating a Toolbar and Touch Bar into Your App](#).

## Topics

## Creating an toolbar object

```
init(identifier: NSToolbar.Identifier)
```

Creates a newly allocated toolbar with the specified identifier.

```
convenience init()
```

Creates a new toolbar with an empty identifier string.

## Configuring the toolbar contents

```
var delegate: (any NSToolbarDelegate)?
```

The object you use to customize the toolbar contents and configuration.

```
protocol NSToolbarDelegate
```

A set of optional methods you use to configure the toolbar and respond to changes.

## Getting the toolbar's identity

```
var identifier: NSToolbar.Identifier
```

The value you use to identify the toolbar in your app.

```
typealias Identifier
```

A string value that you use to differentiate your app's toolbars.

## Configuring the toolbar's behavior

```
var isVisible: Bool
```

A Boolean value that indicates whether the toolbar is visible.

```
var displayMode: NSToolbar.DisplayMode
```

A value that indicates whether the toolbar displays items using a name, icon, or combination of elements.

```
enum DisplayMode
```

Constants that indicate whether the toolbar displays items using a name, icon, or combination of elements.

```
var showsBaselineSeparator: Bool
```

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

Deprecated

```
var allowsUserCustomization: Bool
```

A Boolean value that indicates whether users can modify the contents of the toolbar.

```
var allowsExtensionItems: Bool
```

A Boolean value that indicates whether the toolbar can add items for Action extensions.

## Managing items on the toolbar

```
var items: [NSToolbarItem]
```

An array containing the toolbar's current items, in order.

```
var visibleItems: [NSToolbarItem]?
```

An array containing the toolbar's currently visible items.

```
var centeredItemIdentifiers: Set<NSToolbarItem.Identifier>
```

The set of custom items to display in the center of the toolbar.

```
var selectedItemIdentifier: NSToolbarItem.Identifier?
```

The identifier of the toolbar's currently selected item.

```
class let willAddItemNotification: NSNotification.Name
```

Posts before the toolbar adds a new item.

```
class let didRemoveItemNotification: NSNotification.Name
```

Posted after an item is removed from a toolbar.

```
func insertItem(withItemIdentifier: NSToolbarItem.Identifier, at: Int)
```

Inserts an item into the toolbar at the specified index.

```
func removeItem(at: Int)
```

Removes the item at the specified index in the toolbar.

## Autosaving the configuration

```
var autosavesConfiguration: Bool
```

A Boolean value that indicates whether the toolbar autosaves its configuration.

~~var configuration: [String : Any]~~

A dictionary containing the current configuration details for the toolbar.

Deprecated

~~func setConfiguration([String : Any])~~

Specifies the new configuration details for the toolbar.

Deprecated

## Displaying the customization palette

~~func runCustomizationPalette(Any?)~~

Displays the toolbar's customization palette and handles any user-initiated customizations.

~~var customizationPaletteIsRunning: Bool~~

A Boolean value that indicates whether the toolbar's customization palette is in use.

## Validating visible items

~~func validateVisibleItems()~~

Validates the toolbar's visible items during a window update.

## Deprecated

~~var centeredItemIdentifier: NSToolbarItem.Identifier?~~

The item to display in the center of the toolbar.

Deprecated

~~var fullScreenAccessoryView: NSView?~~

The toolbar's full screen accessory view.

Deprecated

~~var fullScreenAccessoryViewMinHeight: CGFloat~~

The minimum height of the toolbar's full screen accessory view.

Deprecated

~~var fullScreenAccessoryViewMaxHeight: CGFloat~~

The maximum height of the toolbar's full screen accessory view, in points.

Deprecated

~~var sizeMode: NSToolbar.SizeMode~~

The toolbar's size mode.

Deprecated

~~enum SizeMode~~

Constants that specify toolbar display modes.

Deprecated

## Instance Properties

`var allowsDisplayModeCustomization: Bool`

Whether or not the user is allowed to change display modes at run time. This functionality is independent of customizing the order of the items themselves. Only disable when the functionality or legibility of your toolbar could not be improved by another display mode. The user's selection will be persisted using the toolbar's `identifier` when `autosavesConfiguration` is enabled. The default is YES for apps linked on macOS 15.0 and above.

`var itemIdentifiers: [NSToolbarItem.Identifier]`

An array of `itemIdentifiers` that represent the current items in the toolbar. Setting this property will set the current items in the toolbar by diffing against items that already exist. Use this with great caution if `allowsUserCustomization` is enabled as it will override any customizations the user has made. This property is key value observable.

## Instance Methods

`func removeItem(identifier: NSToolbarItem.Identifier)`

Removes the item with matching `itemIdentifier` in the receiving toolbar. If multiple items share the same identifier (as is the case with space items) all matching items will be removed. To remove only a single space item, use `-removeItemAtIndex:` instead.

---

## Relationships

### Inherits From

`NSObject`

### Conforms To

CVarArg  
CustomDebugStringConvertible  
CustomStringConvertible  
Equatable  
Hashable  
NSObjectProtocol  
Sendable

---

## See Also

### View

`{}` [Integrating a Toolbar and Touch Bar into Your App](#)

Provide users quick access to your app’s features from a toolbar and corresponding Touch Bar.

`protocol` [NSToolbarItemValidation](#)

Validation of a toolbar item.