

[SwiftUI](#) / Toolbars

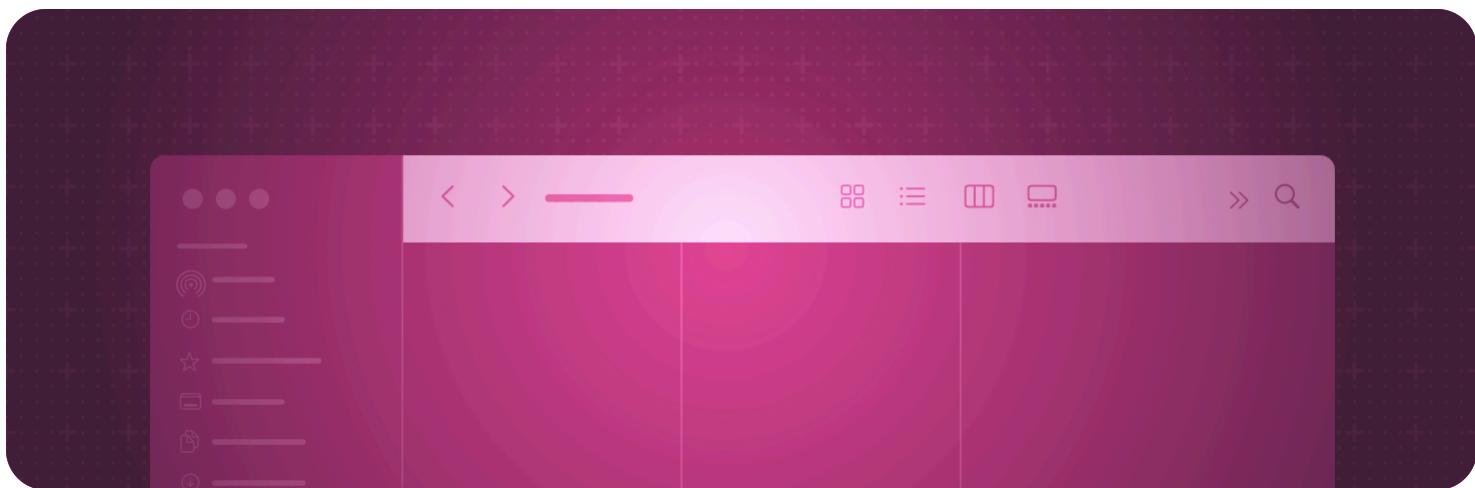
API Collection

Toolbars

Provide immediate access to frequently used commands and controls.

Overview

The system might present toolbars above or below your app's content, depending on the platform and the context.



Add items to a toolbar by applying the [`toolbar\(content:\)`](#) view modifier to a view in your app. You can also configure the toolbar using view modifiers. For example, you can set the visibility of a toolbar with the [`toolbar\(_:for:\)`](#) modifier.

For design guidance, see [Toolbars](#) in the Human Interface Guidelines.

Topics

[Populating a toolbar](#)

```
func toolbar(content:)
```

Populates the toolbar or navigation bar with the specified items.

```
struct ToolbarItem
```

A model that represents an item which can be placed in the toolbar or navigation bar.

```
struct ToolbarItemGroup
```

A model that represents a group of ToolbarItems which can be placed in the toolbar or navigation bar.

```
struct ToolbarItemPlacement
```

A structure that defines the placement of a toolbar item.

```
protocol ToolbarContent
```

Conforming types represent items that can be placed in various locations in a toolbar.

```
struct ToolbarContentBuilder
```

Constructs a toolbar item set from multi-expression closures.

```
struct ToolbarSpacer
```

A standard space item in toolbars.

```
struct DefaultToolbarItem
```

A toolbar item that represents a system component.

Populating a customizable toolbar

```
func toolbar<Content>(id: String, content: () -> Content) -> some View
```

Populates the toolbar or navigation bar with the specified items, allowing for user customization.

```
protocol CustomizableToolbarContent
```

Conforming types represent items that can be placed in various locations in a customizable toolbar.

```
struct ToolbarCustomizationBehavior
```

The customization behavior of customizable toolbar content.

```
struct ToolbarCustomizationOptions
```

Options that influence the default customization behavior of customizable toolbar content.

```
struct SearchToolbarBehavior
```

The behavior of a search field in a toolbar.

Removing default items

```
func toolbarremoving: ToolbarDefaultItemKind?) -> some View
```

Remove a toolbar item present by default

```
struct ToolbarDefaultItemKind
```

A kind of toolbar item a View adds by default.

Setting toolbar visibility

```
func toolbar(Visibility, for: ToolbarPlacement...) -> some View
```

Specifies the visibility of a bar managed by SwiftUI.

```
func toolbarVisibility(Visibility, for: ToolbarPlacement...) -> some View
```

Specifies the visibility of a bar managed by SwiftUI.

```
func toolbarBackgroundVisibility(Visibility, for: ToolbarPlacement...) -> some View
```

Specifies the preferred visibility of backgrounds on a bar managed by SwiftUI.

```
struct ToolbarPlacement
```

The placement of a toolbar.

```
struct ContentToolbarPlacement
```

Specifying the role of toolbar content

```
func toolbarRole(ToolbarRole) -> some View
```

Configures the semantic role for the content populating the toolbar.

```
struct ToolbarRole
```

The purpose of content that populates the toolbar.

Styling a toolbar

```
func toolbarBackground(_ : for:)
```

Specifies the preferred shape style of the background of a bar managed by SwiftUI.

```
func toolbarColorScheme(ColorScheme?, for: ToolbarPlacement...) -> some View
```

Specifies the preferred color scheme of a bar managed by SwiftUI.

```
func toolbarForegroundColor<S>(S, for: ToolbarPlacement...) -> some View
```

Specifies the preferred foreground style of bars managed by SwiftUI.

```
func windowToolbarStyle<S>(S) -> some Scene
```

Sets the style for the toolbar defined within this scene.

```
protocol WindowToolbarStyle
```

A specification for the appearance and behavior of a window's toolbar.

```
var toolbarLabelStyle: ToolbarLabelStyle?
```

The label style to apply to controls within a toolbar.

```
struct ToolbarLabelStyle
```

The label style of a toolbar.

```
struct SpacerSizing
```

A type which defines how spacers should size themselves.

Configuring the toolbar title display mode

```
func toolbarTitleDisplayMode(ToolbarTitleDisplayStyle) -> some View
```

Configures the toolbar title display mode for this view.

```
struct ToolbarTitleDisplayStyle
```

A type that defines the behavior of title of a toolbar.

Setting the toolbar title menu

```
func toolbarTitleMenu<C>(content: () -> C) -> some View
```

Configure the title menu of a toolbar.

```
struct ToolbarTitleMenu
```

The title menu of a toolbar.

Creating an ornament

```
func ornament(visibility:attachmentAnchor:contentAlignment:ornament:)
```

Presents an ornament.

```
struct OrnamentAttachmentAnchor
```

An attachment anchor for an ornament.

See Also

App structure

- ☰ App organization
 - Define the entry point and top-level structure of your app.
- ☰ Scenes
 - Declare the user interface groupings that make up the parts of your app.
- ☰ Windows
 - Display user interface content in a window or a collection of windows.
- ☰ Immersive spaces
 - Display unbounded content in a person's surroundings.
- ☰ Documents
 - Enable people to open and manage documents.
- ☰ Navigation
 - Enable people to move between different parts of your app's view hierarchy within a scene.
- ☰ Modal presentations
 - Present content in a separate view that offers focused interaction.
- ☰ Search
 - Enable people to search for text or other content within your app.
- ☰ App extensions
 - Extend your app's basic functionality to other parts of the system, like by adding a Widget.