SwiftUI / Menus and commands

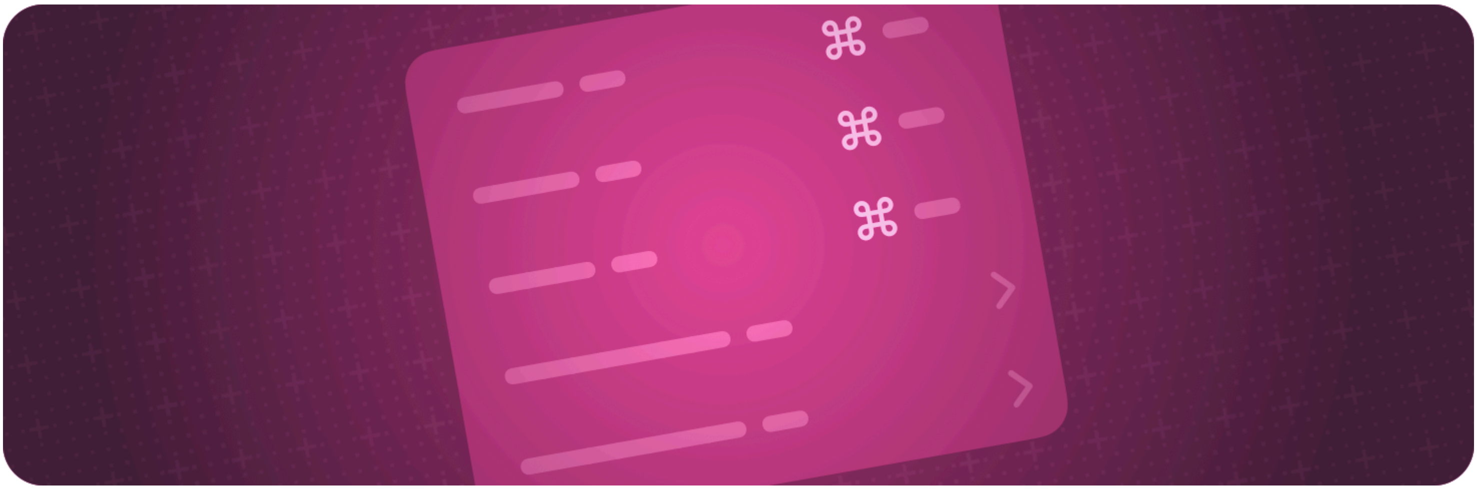
API Collection

Menus and commands

Provide space-efficient, context-dependent access to commands and controls.

Overview

Use a menu to provide people with easy access to common commands. You can add items to a macOS or iPadOS app’s menu bar using the `commands(content:)` scene modifier, or create context menus that people reveal near their current task using the `contextMenu(menuItems:)` view modifier.



Create submenus by nesting `Menu` instances inside others. Use a `Divider` view to create a separator between menu elements.

For design guidance, see Menus in the Human Interface Guidelines.

Topics

Building a menu bar

Building and customizing the menu bar with SwiftUI

Provide a seamless, cross-platform user experience by building a native menu bar for iPadOS and macOS.

Creating a menu

Populating SwiftUI menus with adaptive controls

Improve your app by populating menus with controls and organizing your content intuitively.

```
struct Menu
```

A control for presenting a menu of actions.

```
func menuStyle<S>(S) -> some View
```

Sets the style for menus within this view.

Creating context menus

```
func contextMenu<MenuItems>(menuItems: () -> MenuItems) -> some View
```

Adds a context menu to a view.

```
func contextMenu<M, P>(menuItems: () -> M, preview: () -> P) -> some View
```

Adds a context menu with a custom preview to a view.

```
func contextMenu<I, M>(forSelectionType: I.Type, menu: (Set<I>) -> M, primaryAction: ((Set<I>) -> Void)?) -> some View
```

Adds an item-based context menu to a view.

Defining commands

```
func commands<Content>(content: () -> Content) -> some Scene
```

Adds commands to the scene.

```
func commandsRemoved() -> some Scene
```

Removes all commands defined by the modified scene.

```
func commandsReplaced<Content>(content: () -> Content) -> some Scene
```

Replaces all commands defined by the modified scene with the commands from the builder.

`protocol Commands`

Conforming types represent a group of related commands that can be exposed to the user via the main menu on macOS and key commands on iOS.

`struct CommandMenu`

Command menus are stand-alone, top-level containers for controls that perform related, app-specific commands.

`struct CommandGroup`

Groups of controls that you can add to existing command menus.

`struct CommandsBuilder`

Constructs command sets from multi-expression closures. Like `ViewBuilder`, it supports up to ten expressions in the closure body.

`struct CommandGroupPlacement`

The standard locations that you can place new command groups relative to.

Getting built-in command groups

`struct SidebarCommands`

A built-in set of commands for manipulating window sidebars.

`struct TextEditingCommands`

A built-in group of commands for searching, editing, and transforming selections of text.

`struct TextFormattingCommands`

A built-in set of commands for transforming the styles applied to selections of text.

`struct ToolbarCommands`

A built-in set of commands for manipulating window toolbars.

`struct ImportFromDevicesCommands`

A built-in set of commands that enables importing content from nearby devices.

`struct InspectorCommands`

A built-in set of commands for manipulating inspectors.

`struct EmptyCommands`

An empty group of commands.

Showing a menu indicator

`func menuIndicator(Visibility) -> some View`

Sets the menu indicator visibility for controls within this view.

`var menuIndicatorVisibility: Visibility`

The menu indicator visibility to apply to controls within a view.

Configuring menu dismissal

`func menuActionDismissBehavior(MenuActionDismissBehavior) -> some View`

Tells a menu whether to dismiss after performing an action.

`struct MenuActionDismissBehavior`

The set of menu dismissal behavior options.

Setting a preferred order

`func menuOrder(MenuOrder) -> some View`

Sets the preferred order of items for menus presented from this view.

`var menuOrder: MenuOrder`

The preferred order of items for menus presented from this view.

`struct MenuOrder`

The order in which a menu presents its content.

Deprecated types

~~struct MenuButton~~

A button that displays a menu containing a list of choices when pressed.

Deprecated

~~typealias PullDownButton~~ Deprecated

~~struct ContextMenu~~

A container for views that you present as menu items in a context menu.

Deprecated

See Also

Views

- View fundamentals

  Define the visual elements of your app using a hierarchy of views.

- View configuration

  Adjust the characteristics of views in a hierarchy.

- View styles

  Apply built-in and custom appearances and behaviors to different types of views.

- Animations

  Create smooth visual updates in response to state changes.

- Text input and output

  Display formatted text and get text input from the user.

- Images

  Add images and symbols to your app’s user interface.

- Controls and indicators

  Display values and get user selections.

- Shapes

  Trace and fill built-in and custom shapes with a color, gradient, or other pattern.

- Drawing and graphics

  Enhance your views with graphical effects and customized drawings.