

[SwiftUI](#) / [Scene](#) / commands(content:)

## Instance Method

# commands(content:)

Adds commands to the scene.

iOS 14.0+ | iPadOS 14.0+ | Mac Catalyst 14.0+ | macOS 11.0+ | visionOS 1.0+

nonisolated

```
func commands<Content>(@Commands  
Builder content: () -> Content) -> some Scene where Content : Commands
```

## Mentioned in

 Building and customizing the menu bar with SwiftUI

## Discussion

Commands are realized in different ways on different platforms. On macOS, the main menu uses the available command menus and groups to organize its main menu items. Each menu is represented as a top-level menu bar menu, and each command group has a corresponding set of menu items in one of the top-level menus, delimited by separator menu items.

On iPadOS, commands with keyboard shortcuts are exposed in the shortcut discoverability HUD that users see when they hold down the Command (⌘) key.

## See Also

[Defining commands](#)

```
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.