

[SwiftUI](#) / [Scene](#) / `commandsRemoved()`

Instance Method

commandsRemoved()

Removes all commands defined by the modified scene.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst 16.0+ | macOS 13.0+ | visionOS 1.0+

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

Return Value

A scene that excludes any commands defined by its children.

Discussion

`WindowGroup`, `Window`, and other scene types all have an associated set of commands that they include by default. Apply this modifier to a scene to exclude those commands.

For example, the following code adds a scene for presenting the details of an individual data model in a separate window. To ensure that the window can only appear programmatically, we remove the scene's commands, including `File > New Note Window`.

```
@main
struct Example: App {
    var body: some Scene {
        ...

        WindowGroup("Note", id: "note", for: Note.ID.self) {
            NoteDetailView(id: $0)
        }
    }
}
```

```
.commandsRemoved()
```

```
}
```

See Also

Defining commands

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

Adds commands to the 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.