

[visionOS](#) / [Introductory visionOS samples](#) / Creating SwiftUI windows in visionOS

Sample Code

Creating SwiftUI windows in visionOS

Display and manage multiple SwiftUI windows in your visionOS app.

[Download](#)

visionOS 2.0+ | Xcode 16.0+



Overview

This sample code project demonstrates how to open a new SwiftUI view and a separate window group to manage multiple windows, assigning a unique `id` to each newly created window. In the sample, the app displays a SwiftUI window with a [Button](#) that a person can tap to open a new window view, as the following image illustrates:



Create a variable to track window IDs

To manage multiple windows for an app, the app uses the [Identifiable](#) protocol to establish an ID value for each new window view.

```
import SwiftUI

struct NewWindowID: Identifiable {
    /// The unique identifier for the window.
    var id: Int
}
```

Add a button for creating windows

The `OpenNewWindow` view creates and displays a SwiftUI button. When a person taps the button, a new SwiftUI window appears.

```
import SwiftUI

struct OpenWindowView: View {
    /// The `id` value that the main view uses to identify the SwiftUI window.
    @State var nextWindowID = NewWindowID(id: 1)

    /// The environment value for getting an `OpenWindowAction` instance.
    @Environment(\.openWindow) private var openWindow

    var body: some View {
        // Create a button in the center of the window that
        // launches a new SwiftUI window.
        Button("Open a new window") {
            // Open a new window with the assigned ID.
            openWindow(value: nextWindowID.id)

            // Increment the `id` value of the `nextWindowID` by 1.
            nextWindowID.id += 1
        }
    }
}
```

The [openWindow](#) instance property invokes a new window view in an app's environment.

Create a view for the new window

The NewWindow view displays the window's id value in a [Text](#) instance.

```
import SwiftUI

struct NewWindow: View {
    /// Acts as the main identifier for the new view.
    let id: Int

    var body: some View {
        // Create a text view that displays
        // the window's `id` value.
        Text("New window number \u{20}{(id)}")
    }
}
```

Add new windows to a window group

The `EntryPoint` provides a specific `WindowGroup` to create a window view and add the new window to the app's main view.

```
import SwiftUI

@main
struct EntryPoint: App {
    var body: some Scene {
        WindowGroup {
            MainView()
        }

        // A `WindowGroup` for each newly created window in the app's main view.
        WindowGroup("New Window", for: NewWindowID.ID.self) { $id in
            NewWindow(id: id ?? 1)
        }
    }
}
```

See Also

Related samples

{ } Creating 3D models as movable windows

Display 3D content with a volumetric window that people can move.