

[SwiftUI / UIKit integration](#)

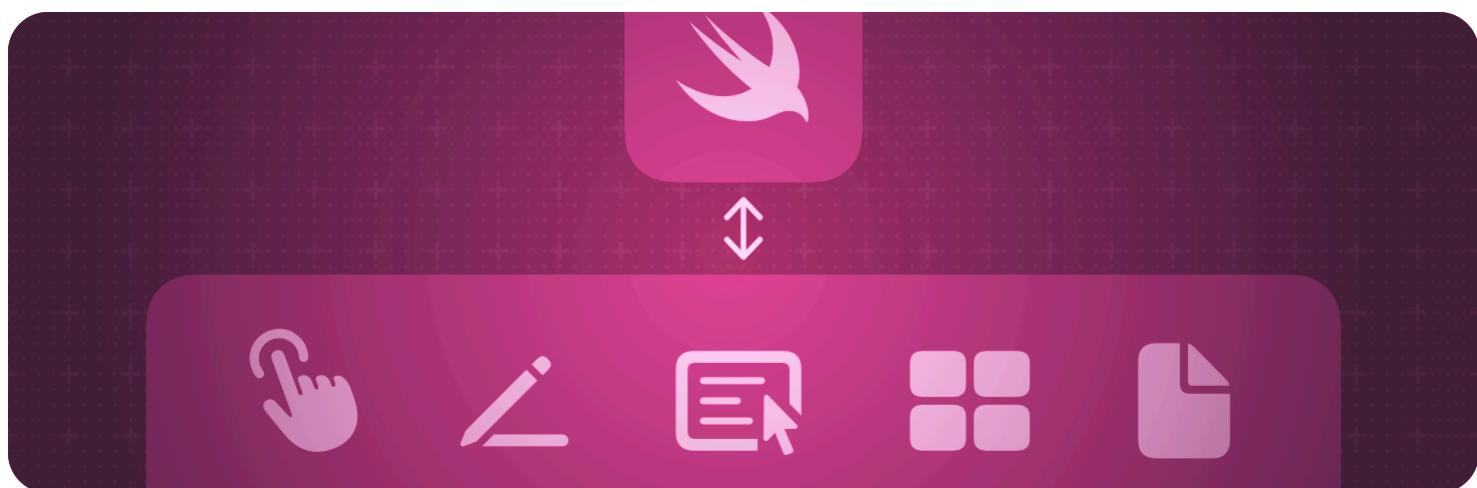
[API Collection](#)

# UIKit integration

Add UIKit views to your SwiftUI app, or use SwiftUI views in your UIKit app.

## Overview

Integrate SwiftUI with your app's existing content using hosting controllers to add SwiftUI views into UIKit interfaces. A hosting controller wraps a set of SwiftUI views in a form that you can then add to your storyboard-based app.



You can also add UIKit views and view controllers to your SwiftUI interfaces. A representable object wraps the designated view or view controller, and facilitates communication between the wrapped object and your SwiftUI views.

For design guidance, see the following sections in the Human Interface Guidelines:

- [Designing for iOS](#)
- [Designing for iPadOS](#)
- [Designing for tvOS](#)

# Topics

## Displaying SwiftUI views in UIKit

### {} Using SwiftUI with UIKit

Learn how to incorporate SwiftUI views into a UIKit app.

### 📄 Unifying your app's animations

Create a consistent UI animation experience across SwiftUI, UIKit, and AppKit.

### `class UIHostingController`

A UIKit view controller that manages a SwiftUI view hierarchy.

### `struct UIHostingControllerSizingOptions`

Options for how a hosting controller tracks its content's size.

### `struct UIHostingConfiguration`

A content configuration suitable for hosting a hierarchy of SwiftUI views.

### `protocol UIHostingSceneDelegate`

Extends UIKit/UISceneDelegate to bridge SwiftUI scenes.

## Adding UIKit views to SwiftUI view hierarchies

### `protocol UIViewRepresentable`

A wrapper for a UIKit view that you use to integrate that view into your SwiftUI view hierarchy.

### `struct UIViewRepresentableContext`

Contextual information about the state of the system that you use to create and update your UIKit view.

### `protocol UIViewControllerRepresentable`

A view that represents a UIKit view controller.

### `struct UIViewControllerRepresentableContext`

Contextual information about the state of the system that you use to create and update your UIKit view controller.

## Adding UIKit gesture recognizers into SwiftUI view hierarchies

```
protocol UIGestureRecognizerRepresentable
```

A wrapper for a `UIGestureRecognizer` that you use to integrate that gesture recognizer into your SwiftUI hierarchy.

```
struct UIGestureRecognizerRepresentableContext
```

Contextual information about the state of the system that you use to create and update a represented gesture recognizer.

```
struct UIGestureRecognizerRepresentableCoordinateSpaceConverter
```

A proxy structure used to convert locations to/from coordinate spaces in the hierarchy of the SwiftUI view associated with a [UIGestureRecognizerRepresentable](#).

## Sharing configuration information

```
typealias UITraitBridgedEnvironmentKey
```

## Hosting an ornament in UIKit

```
class UIHostingOrnament
```

A model that represents an ornament suitable for being hosted in UIKit.

```
class UIOrnament
```

The abstract base class that represents an ornament.

---

## See Also

### Framework integration

- ☰ AppKit integration

Add AppKit views to your SwiftUI app, or use SwiftUI views in your AppKit app.

- ☰ WatchKit integration

Add WatchKit views to your SwiftUI app, or use SwiftUI views in your WatchKit app.

- ☰ Technology-specific views

Use SwiftUI views that other Apple frameworks provide.