Protocol

UIViewRepresentable

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

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | tvOS 13.0+ | visionOS 1.0+

```
@MainActor @preconcurrency
protocol UIViewRepresentable : View where Self.Body == Never
```

Overview

Use a `UIViewRepresentable` instance to create and manage a `UIView` object in your SwiftUI interface. Adopt this protocol in one of your app's custom instances, and use its methods to create, update, and tear down your view. The creation and update processes parallel the behavior of SwiftUI views, and you use them to configure your view with your app's current state information. Use the teardown process to remove your view cleanly from your SwiftUI. For example, you might use the teardown process to notify other objects that the view is disappearing.

To add your view into your SwiftUI interface, create your `UIViewRepresentable` instance and add it to your SwiftUI interface. The system calls the methods of your representable instance at appropriate times to create and update the view. The following example shows the inclusion of a custom `MyRepresentedCustomView` structure in the view hierarchy.

```
struct ContentView: View {
    var body: some View {
        VStack {
            Text("Global Sales")
            MyRepresentedCustomView()
        }
    }
}
```

The system doesn't automatically communicate changes occurring within your view to other parts of your SwiftUI interface. When you want your view to coordinate with other SwiftUI views, you must provide a `Coordinator` instance to facilitate those interactions. For example, you use a coordinator to forward target-action and delegate messages from your view to any SwiftUI views.

> **Warning**
>
> SwiftUI fully controls the layout of the UIKit view's `center`, `bounds`, `frame`, and `transform` properties. Don't directly set these layout-related properties on the view managed by a `UIViewRepresentable` instance from your own code because that conflicts with SwiftUI and results in undefined behavior.

Topics

Creating and updating the view

`func makeUIView(context: Self.Context) -> Self.UIViewType`

Creates the view object and configures its initial state.

Required

`func updateUIView(Self.UIViewType, context: Self.Context)`

Updates the state of the specified view with new information from SwiftUI.

Required

`typealias Context`

`associatedtype UIViewType : UIView`

The type of view to present.

Required

Specifying a size

`func sizeThatFits(ProposedViewSize, uiView: Self.UIViewType, context: Self.Context) -> CGSize?`

Given a proposed size, returns the preferred size of the composite view.

Required Default implementation provided.

Cleaning up the view

```
static func dismantleUIView(Self.UIViewType, coordinator: Self.
Coordinator)
```

Cleans up the presented UIKit view (and coordinator) in anticipation of their removal.

Required Default implementation provided.

Providing a custom coordinator object

```
func makeCoordinator() -> Self.Coordinator
```

Creates the custom instance that you use to communicate changes from your view to other parts of your SwiftUI interface.

Required Default implementation provided.

```
associatedtype Coordinator = Void
```

A type to coordinate with the view.

Required

Performing layout

```
typealias LayoutOptions
```

Relationships

Inherits From

`View`

See Also

Adding UIKit views to SwiftUI view hierarchies

```
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.