Documentation Language: Swift

SwiftUI / DocumentGroup

Structure

DocumentGroup

A scene that enables support for opening, creating, and saving documents.

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

```
struct DocumentGroup<Document, Content> where Content : View
```

Mentioned in

Building and customizing the menu bar with SwiftUI

Overview

Use a `DocumentGroup` scene to tell SwiftUI what kinds of documents your app can open when you declare your app using the App protocol.

Initialize a document group scene by passing in the document model and a view capable of displaying the document type. The document types you supply to `DocumentGroup` must conform to `FileDocument` or `ReferenceFileDocument`. SwiftUI uses the model to add document support to your app. In macOS this includes document-based menu support, including the ability to open multiple documents. On iOS this includes a document browser that can navigate to the documents stored on the file system and multiwindow support:

```
@main
struct MyApp: App {
    var body: some Scene {
        DocumentGroup(newDocument: TextFile()) { configuration in
            ContentView(document: configuration.$document)
        }
    }
}
```

Any time the configuration changes, SwiftUI updates the contents with that new configuration, similar to other parameterized builders.

Viewing documents

If your app only needs to display but not modify a specific document type, you can use the file viewer document group scene. You supply the file type of the document, and a view that displays the document type that you provide:

```
@main
struct MyApp: App {
    var body: some Scene {
        DocumentGroup(viewing: MyImageFormatDocument.self) {
            MyImageFormatViewer(image: $0.document)
        }
    }
}
```

Supporting multiple document types

Your app can support multiple document types by adding additional document group scenes:

```
@main
struct MyApp: App {
    var body: some Scene {
        DocumentGroup(newDocument: TextFile()) { group in
            ContentView(document: group.$document)
        }
        DocumentGroup(viewing: MyImageFormatDocument.self) { group in
            MyImageFormatViewer(image: group.document)
        }
    }
}
```

Accessing the document’s URL

If your app needs to know the document’s URL, you can read it from the `editor` closure’s `configuration` parameter, along with the binding to the document. When you create a new document, the configuration’s `fileURL` property is `nil`. Every time it changes, it is passed over to the `DocumentGroup` builder in the updated `configuration`. This ensures that the view you define in the closure always knows the URL of the document it hosts.

```
@main
struct MyApp: App {
    var body: some Scene {
        DocumentGroup(newDocument: TextFile()) { configuration in
            ContentView(
                document: configuration.$document,
                fileURL: configuration.fileURL
            )
        }
    }
}
```

The URL can be used, for example, to present the file path of the file name in the user interface. Don’t access the document’s contents or metadata using the URL because that can conflict with the management of the file that SwiftUI performs. Instead, use the methods that `FileDocument` and `ReferenceFileDocument` provide to perform read and write operations.

Topics

Creating a document group

`init(newDocument:editor:)`

Creates a document group for creating and editing file documents.

`init(viewing:viewer:)`

Creates a document group capable of viewing file documents.

Editing a document backed by a persistent store

`init(editing:contentType:editor:prepareDocument:)`

Instantiates a document group for creating and editing documents that store a specific model type.

`init(editing: UTType, migrationPlan: any SchemaMigrationPlan.Type, editor: () -> Content, prepareDocument: (ModelContext) -> Void)`

Instantiates a document group for creating and editing documents described by the last `Schema` in the migration plan.

Viewing a document backed by a persistent store

`init(viewing:contentType:viewer:)`

Instantiates a document group for viewing documents that store a specific model type.

`init(viewing: UTType, migrationPlan: any SchemaMigrationPlan.Type, viewer: () -> Content)`

Instantiates a document group for viewing documents described by the last `Schema` in the migration plan.

Relationships

Conforms To

`Scene`

See Also

Creating a document

{} Building a document-based app with SwiftUI

Create, save, and open documents in a multiplatform app.

{} Building a document-based app using SwiftData

Code along with the WWDC presenter to transform an app with SwiftData.