

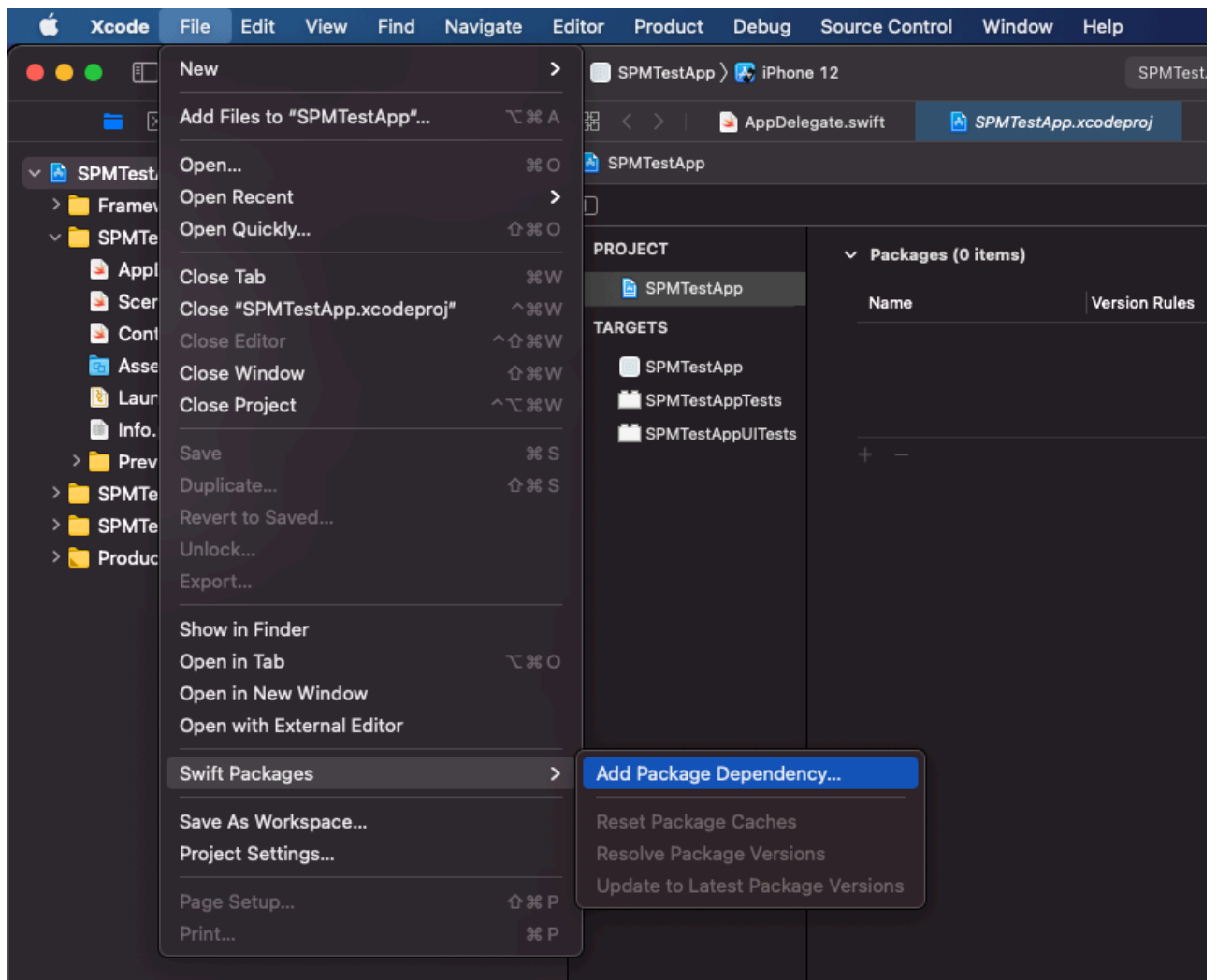
So integrieren Sie das WalkMe Mobile iOS SDK mit dem Swift Package Manager

Kurzübersicht

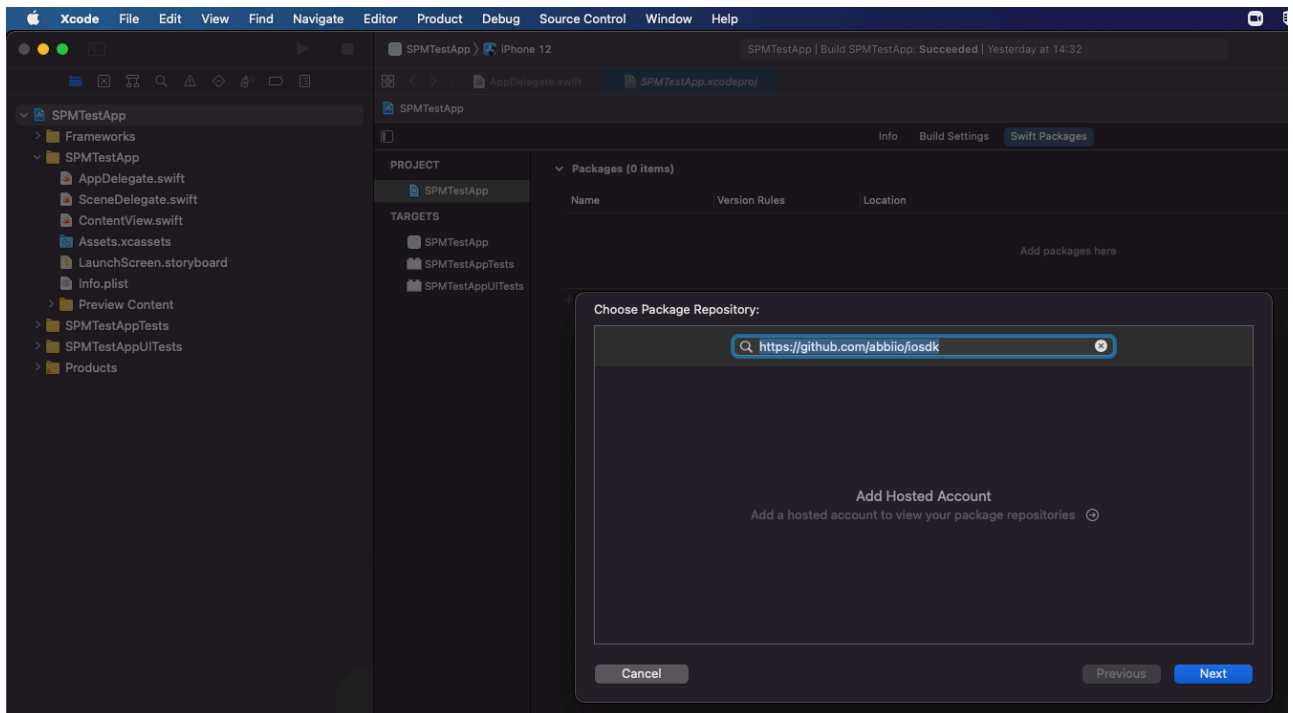
Das WalkMe iOS SDK ist mit dem Swift Package Manager kompatibel. Dieser Artikel erklärt, wie Sie WalkMe iOS SDK in den Swift Package Manager integrieren können.

Funktionsweise

1. Öffnen Sie Ihre App in Xcode
2. Choose File (Datei auswählen) → Swift Packages → Add Package Dependency
(Paketabhängigkeit hinzufügen)

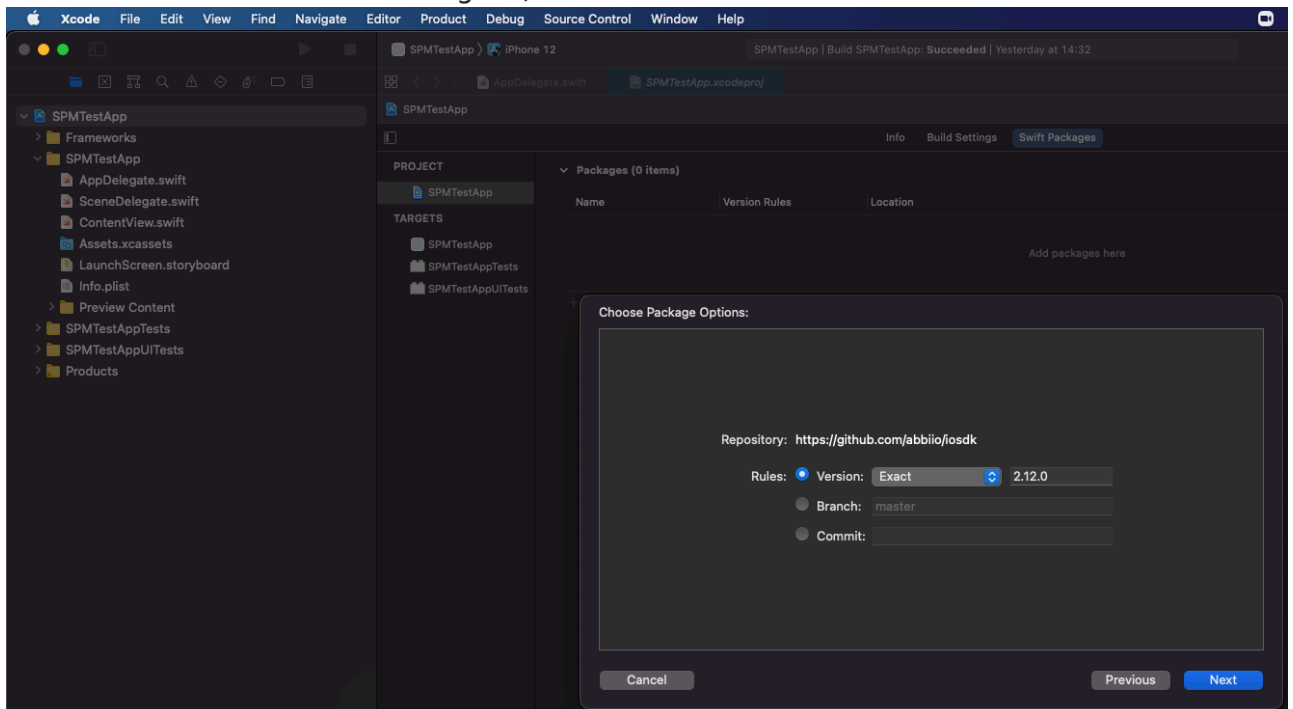


3. Geben Sie den Pfad zum Paket-Repository ein: <https://github.com/abbiio/iosdk>

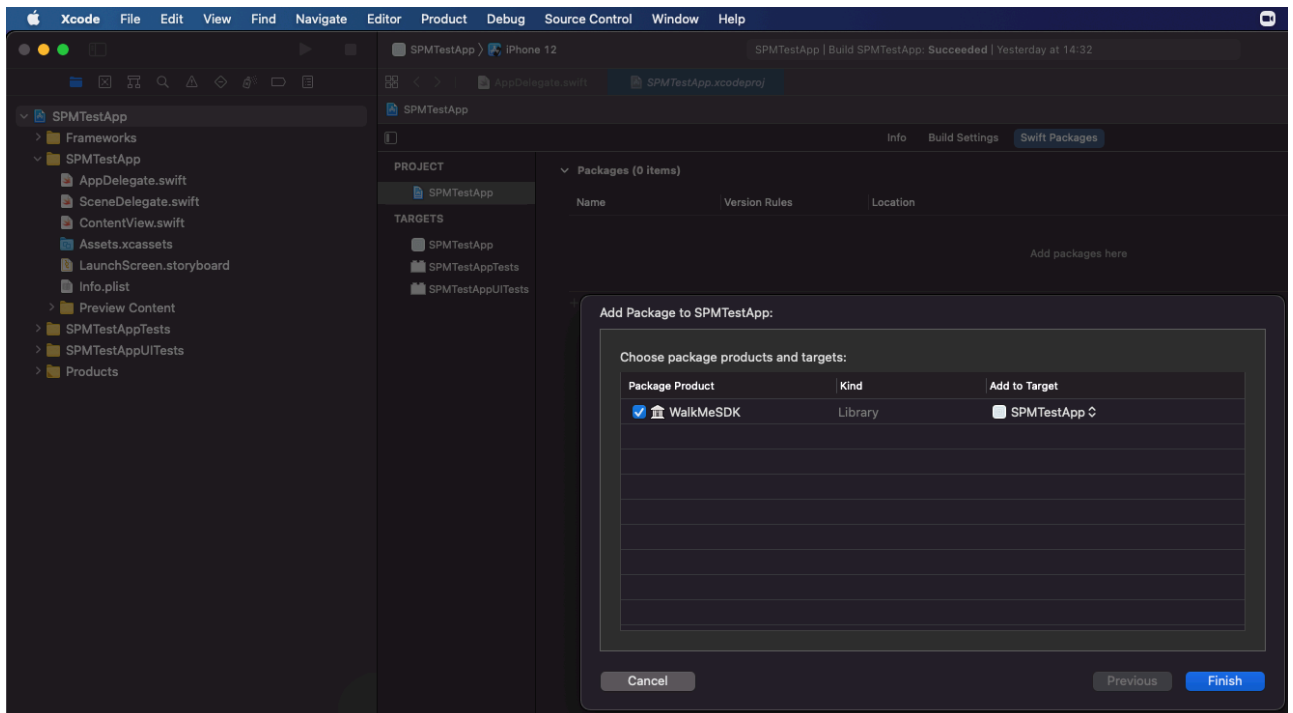


4. Das Repo wird abgerufen.

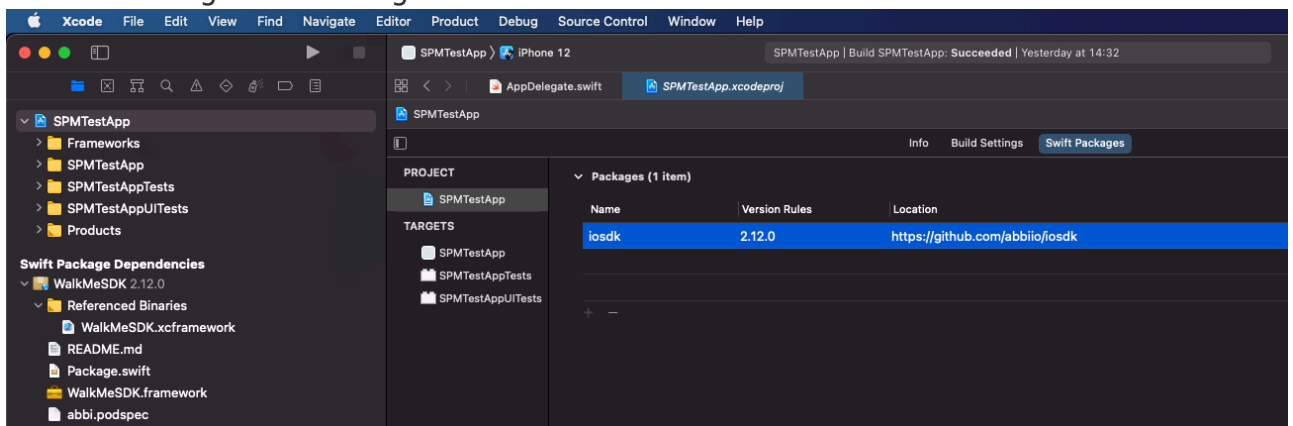
5. Wählen Sie die Version des WalkMeSDK, das Sie integrieren möchten (SPM ist ab Version 2.12.0 und höher verfügbar).



6. Vergewissern Sie sich, dass WalkMeSDK aktiviert ist, und klicken Sie auf Finish (Fertigstellen)



7. Die Paketintegration ist abgeschlossen.



8. Importieren Sie in Ihrer AppDelegate-Datei WalkMeSDK.

ObjC: `#import <WalkMeSDK/WalkMeSDK.h>` Swift: `Import WalkMeSDK` (WalkMeSDK importieren)

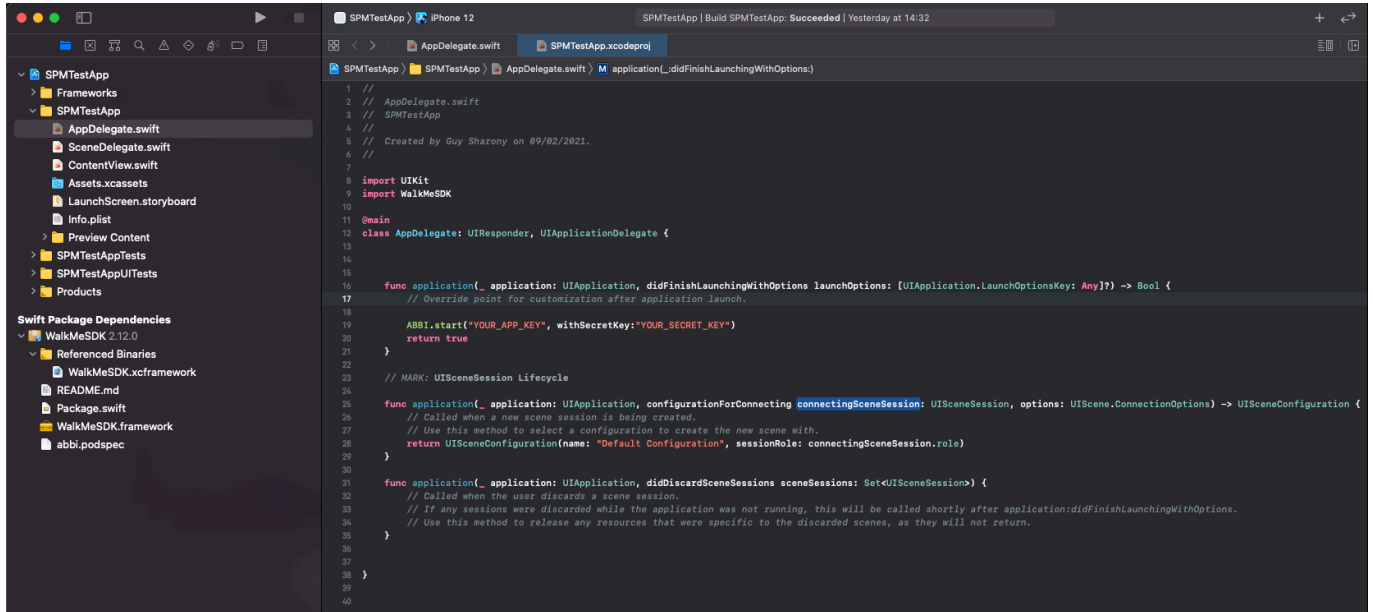
9. Starten Sie WalkMeSDKObjC: In der (BOOL)-Anwendung:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary*)launchOptions, fügen Sie Folgendes hinzu:

```
[ABBI start:@"YOUR_APP_KEY" withSecretKey:@"YOUR_SECRET_KEY"];
```

Swift: In der func `application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool`, fügen Sie Folgendes hinzu:

```
ABBI.start("YOUR_APP_KEY", withSecretKey:"YOUR_SECRET_KEY")
```

Swift-Beispiel:



```
1 //
2 // AppDelegate.swift
3 // SPMTestApp
4 //
5 // Created by Guy Sharony on 09/02/2021.
6 //
7
8 import UIKit
9 import WalkMeSDK
10
11 @main
12 class AppDelegate: UIResponder, UIApplicationDelegate {
13
14
15
16     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
17         // Override point for customization after application launch.
18
19         ABBI.start("YOUR_APP_KEY", withSecretKey:"YOUR_SECRET_KEY")
20         return true
21     }
22
23     // MARK: UISceneSession Lifecycle
24
25     func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration {
26         // Called when a new scene session is being created.
27         // Use this method to select a configuration to create the new scene with.
28         return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)
29     }
30
31     func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {
32         // Called when the user discards a scene session.
33         // If any sessions were discarded while the application was not running, this will be called shortly after application:didFinishLaunchingWithOptions.
34         // Use this method to release any resources that were specific to the discarded scenes, as they will not return.
35     }
36
37
38 }
39
40
```