

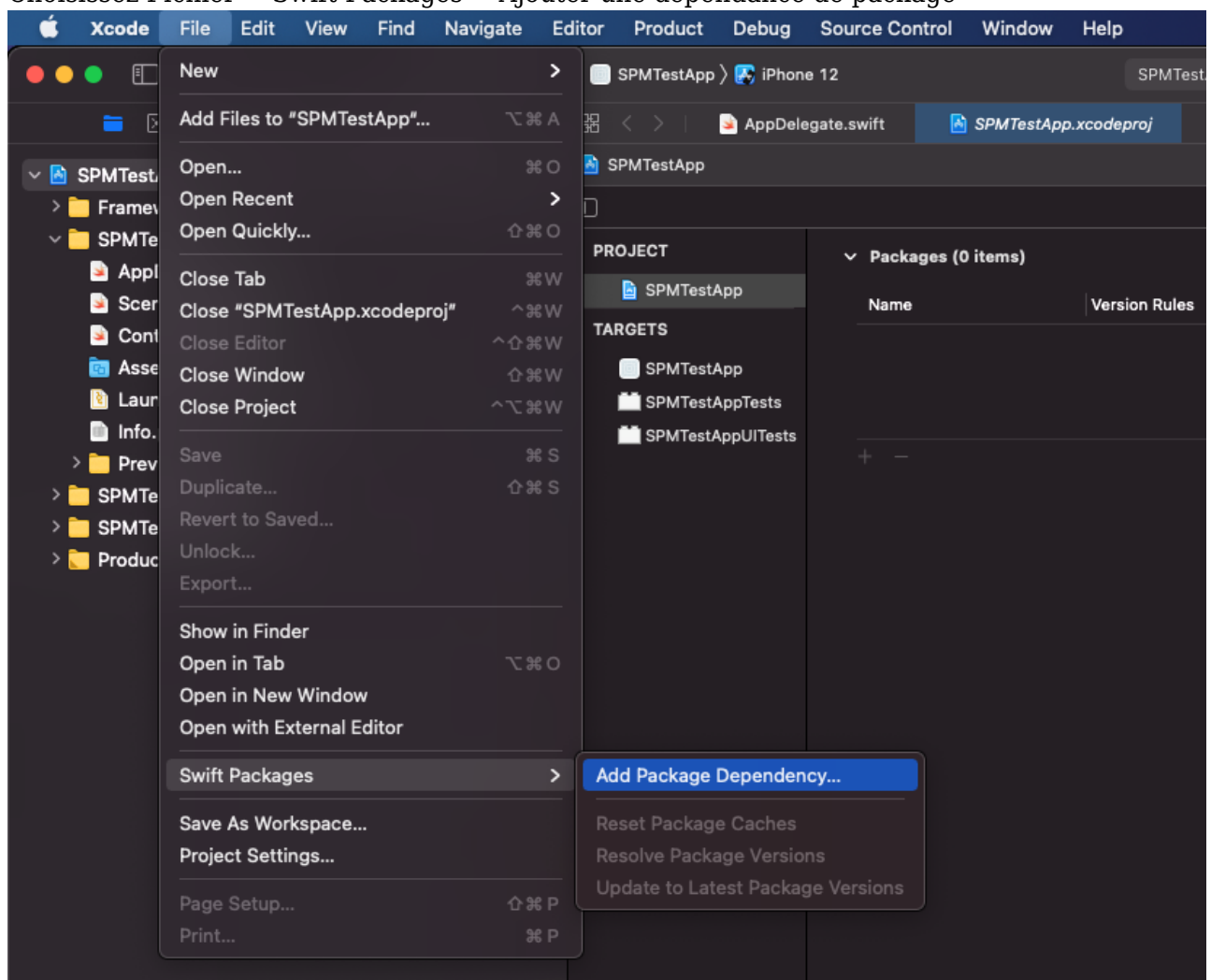
Comment intégrer le SDK iOS de WalkMe Mobile à l'aide de Swift Package Manager

Aperçu général

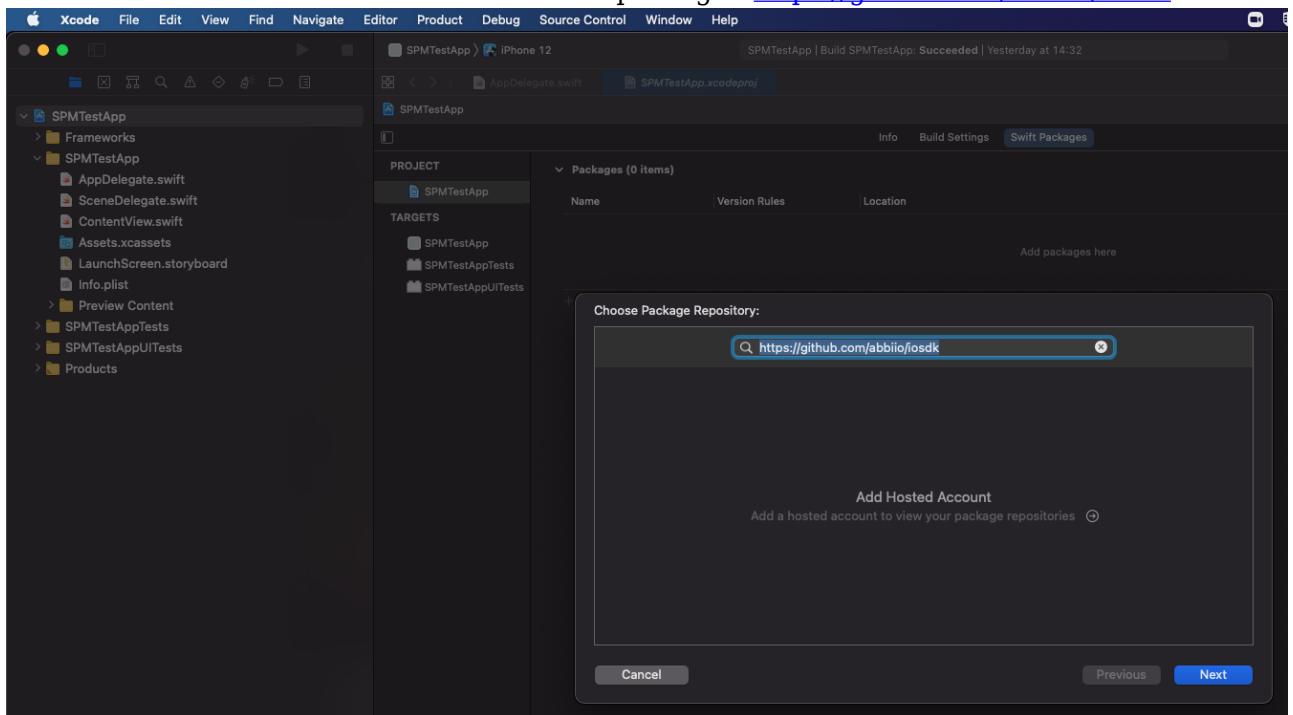
The SDK iOS de WalkMe est compatible avec Swift Package Manager. Cet article explique comment intégrer le SDK iOS de WalkMe avec Swift Package Manager.

Comment ça marche

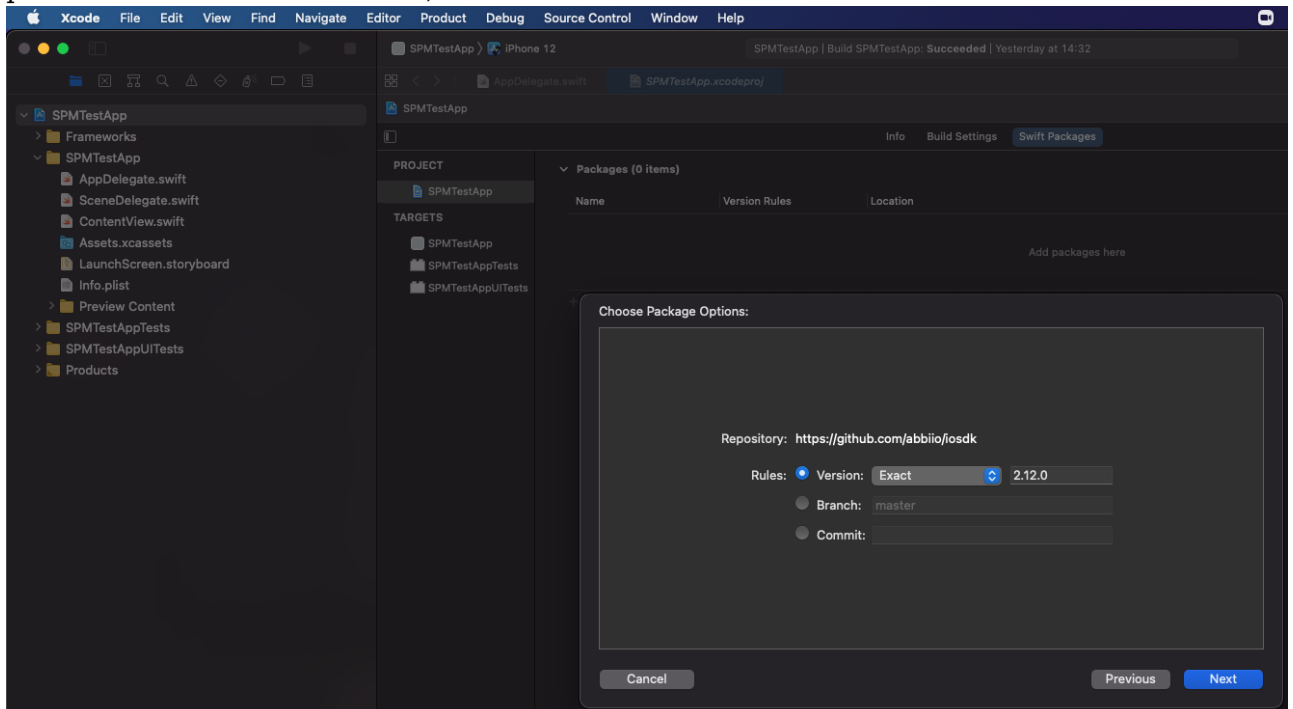
1. Ouvrez votre application dans Xcode.
2. Choisissez Fichier → Swift Packages → Ajouter une dépendance de package



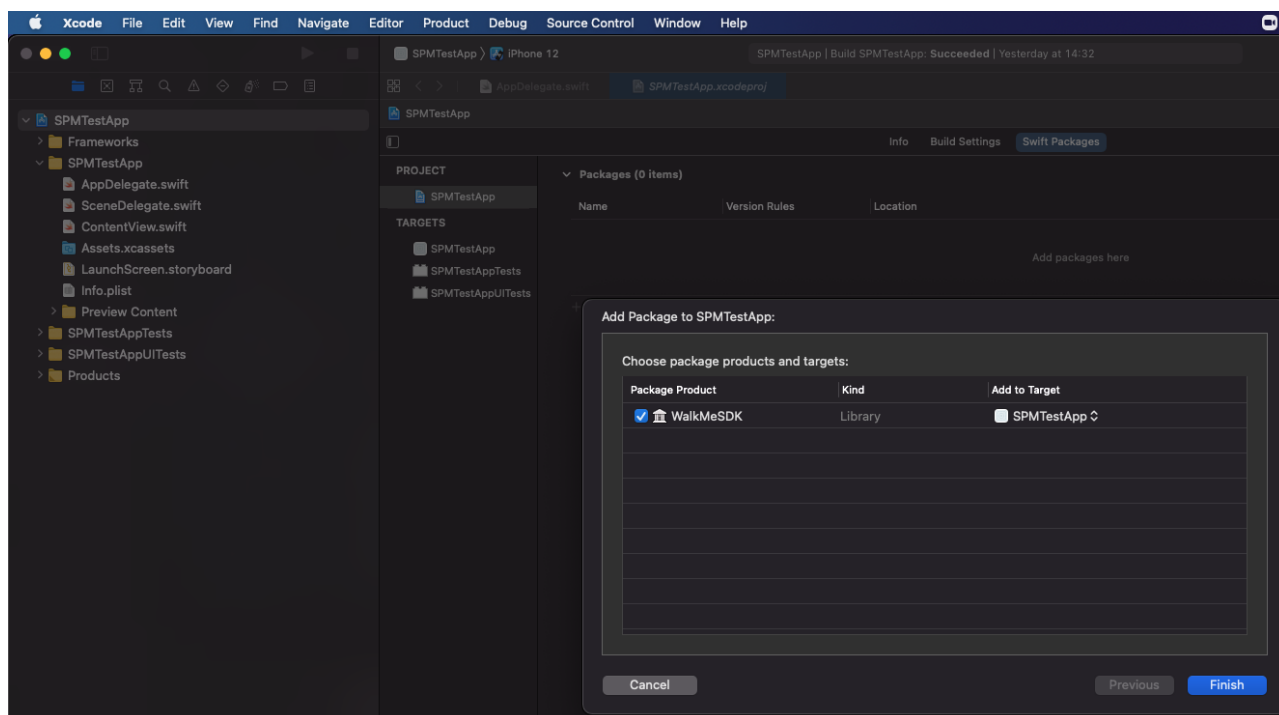
- Saisissez le chemin d'accès au référentiel du package : <https://github.com/abbiio/iosdk>



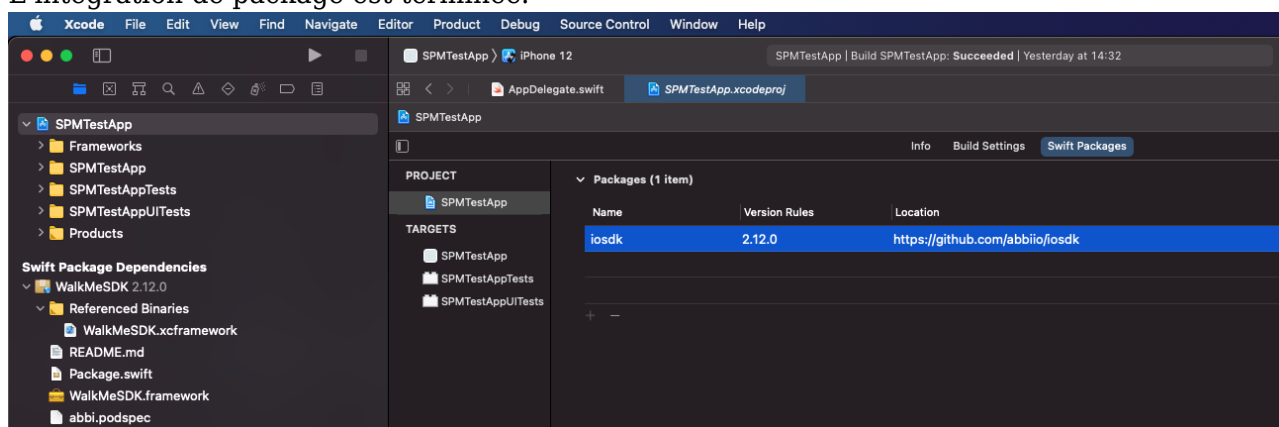
- Le référentiel sera récupéré.
- Choisissez la version du WalkMeSDK que vous souhaitez intégrer (le SPM est disponible à partir de la 2.12.0 et ultérieure)



- Assurez-vous que WalkMeSDK est vérifié et cliquez sur Finish.



7. L'intégration de package est terminée.



8. Dans votre fichier AppDelegate, importez WalkMeSDK

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

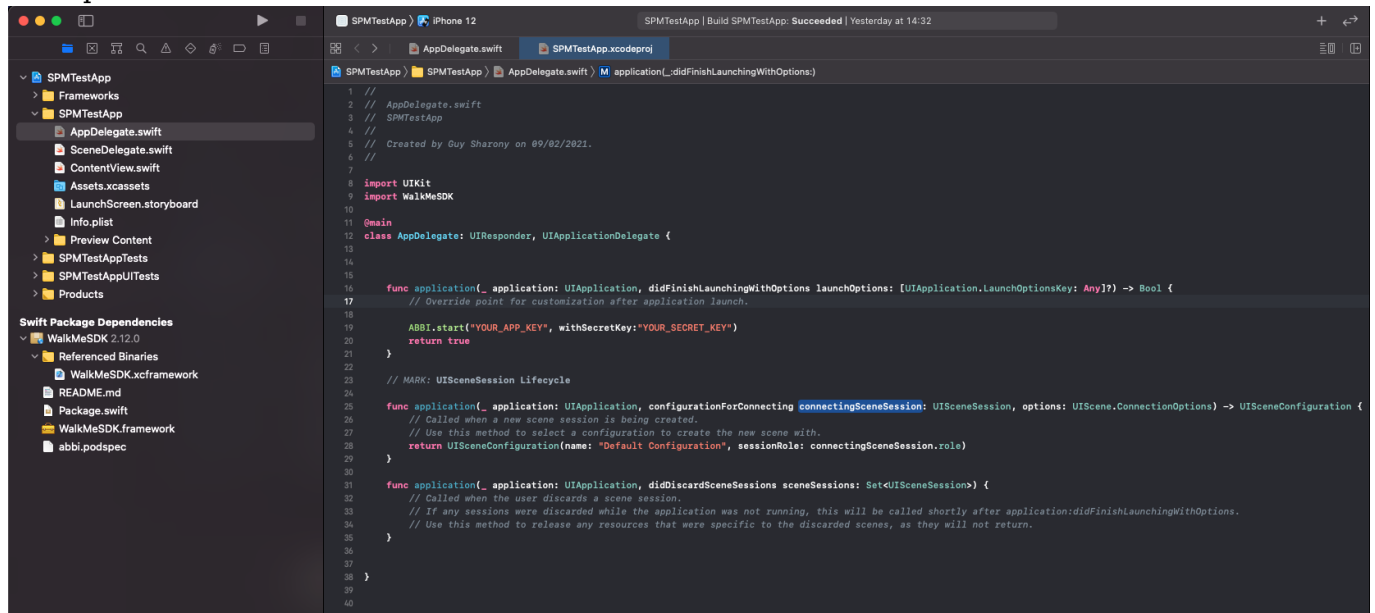
9. Démarrez WalkMeSDKObjC : Dans *(BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary*)launchOptions*, ajoutez les informations suivantes :

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

Swift : Dans `func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool`, ajoutez les informations suivantes :

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

Exemple de swift



```

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
17     // Override point for customization after application launch.
18
19     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
20         // MARK: UISceneSession Lifecycle
21         // MARK: UISceneSession Lifecycle
22
23         func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) -> UISceneConfiguration {
24             // Called when a new scene session is being created.
25             // Use this method to select a configuration to create the new scene with.
26             return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)
27         }
28
29         func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {
30             // Called when the user discards a scene session.
31             // If any sessions were discarded while the application was not running, this will be called shortly after application:didFinishLaunchingWithOptions.
32             // Use this method to release any resources that were specific to the discarded scenes, as they will not return.
33         }
34     }
35
36
37
38
39
40

```