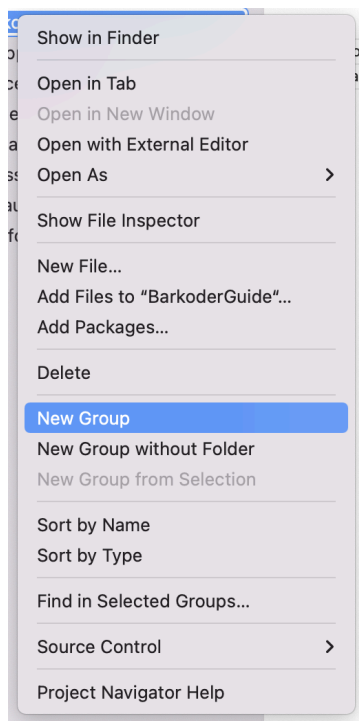


Barkoder Guide for iOS

Installation guide

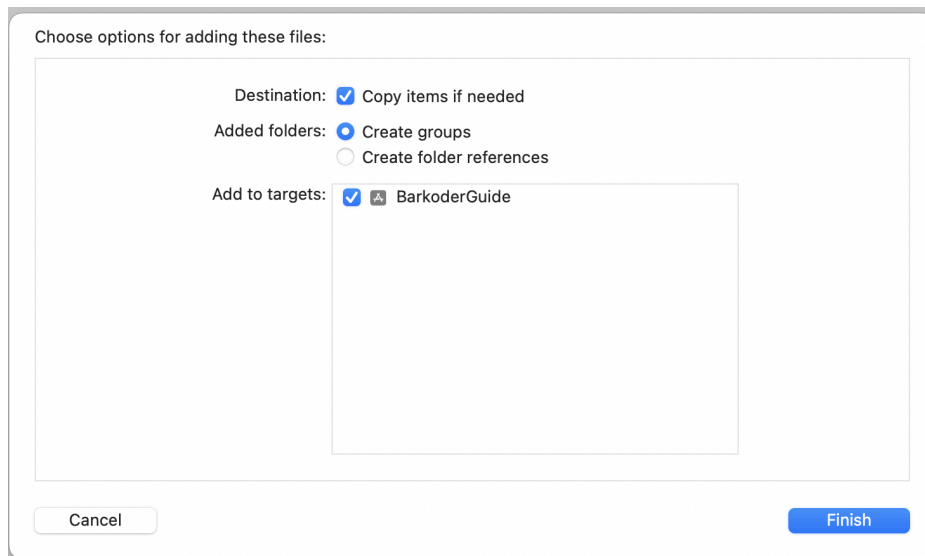
Please follow these simple steps to integrate our SDK into your iOS project

1. Create new group and name it “frameworks” (optional)

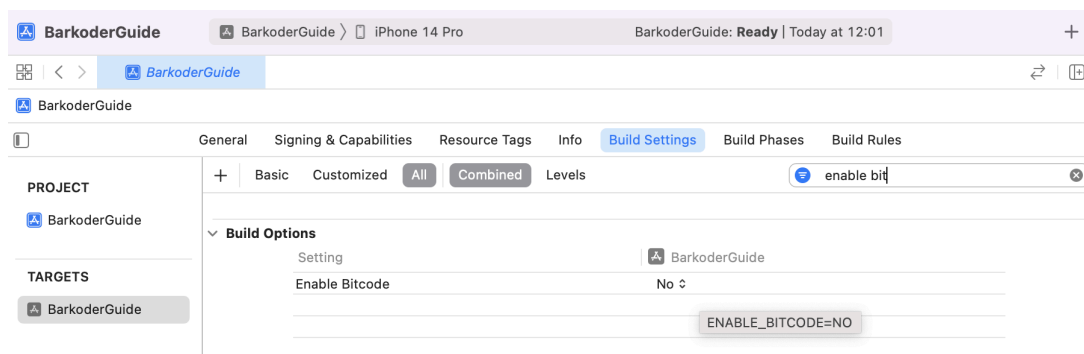


2. Add **Barkoder.xcframework** and **BarkoderSDK.xcframework** into frameworks

**Copy items if needed, Create groups and Add to desired targets)*



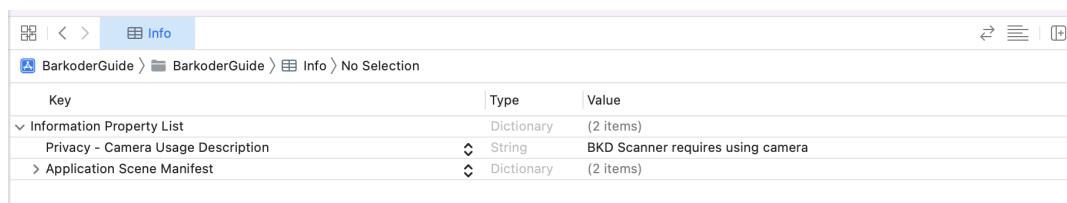
3. In **Build Settings** set **Enable Bitcode** to **NO**



4. Open the **Info.plist** with Source Code and add these changes

```
<key>NSCameraUsageDescription</key>  
<string>BKD Scanner requires using camera</string>
```

** Or using **Property List** add “Privacy - Camera Usage Description” and key “BKD Scanner requires using camera”*



5. Add **BarkoderView** as a view

```
import UIKit
import BarkoderSDK

class ViewController: UIViewController {

    private var barkoderView: BarkoderView!

    override func viewDidLoad() {
        super.viewDidLoad()
        setupUI()
        setupConstraints()
    }

    private func setupUI() {
        barkoderView = BarkoderView()

        view.addSubview(barkoderView)
    }

    private func setupConstraints() {
        barkoderView.translatesAutoresizingMaskIntoConstraints = false
        barkoderView.leadingAnchor.constraint(equalTo: view.leadingAnchor).isActive = true
        barkoderView.topAnchor.constraint(equalTo: view.topAnchor).isActive = true
        barkoderView.trailingAnchor.constraint(equalTo: view.trailingAnchor).isActive = true
        barkoderView.bottomAnchor.constraint(equalTo: view.bottomAnchor).isActive = true
    }
}
```

6. Create **Barkoder Config** per your needs

```
private func createBarkoderConfig() {
    // In order to perform scanning, config property need to be set before
    // If license key is not valid you will receive results with asterisks inside
    barkoderView.config = BarkoderConfig(licenseKey: "LICENSE_KEY") { licenseResult in
        print("Licensing SDK: \(licenseResult)")
    }

    // Enable QR barcode type
    guard let decoderConfig = barkoderView.config?.decoderConfig else { return }
    decoderConfig.qr.enabled = true
}
```

7. Implement **BarkoderResultDelegate** protocol where you will receive scanned results

```
extension ViewController: BarkoderResultDelegate {  
  
    func scanningFinished(_ decoderResults: [DecoderResult], thumbnails: [UIImage]?, image: UIImage?) {  
        if let textualData = decoderResults[0].textualData {  
            print("Scanned result: ", textualData)  
        }  
    }  
}
```

8. Start the scanning process

```
try? barkoderView.startScanning(self)
```

API description

The code documentation, including all supported platforms, can be found at <https://barkoder.com/docs/v1/ios/ios-sdk-api-reference>