

# CocoaPods vs. Carthage



Dennis Parussini

Dec 13, 2015 · 3 min read

I've been dealing a lot with both CocoaPods and Carthage lately and I would like to share my thoughts about both of them.

3rd Party Frameworks are a big deal in both iOS and OS X development. There's a ton of ready to use libraries on Github and most of them can be added to your project using either **Carthage** or **CocoaPods** (You could also add them manually as a submodule, but I like the ability to just add them via aforementioned dependency managers).

CocoaPods is here since the early days, when iOS Apps still have been written in Objective-C only. It's pretty easy to add a library to your project. Just create a text file in the root folder of you application named 'Podfile' and add 'pod "FrameworkName"' (where FrameworkName has to be replaced with the actual name of the framework) to the file. Run 'pod install' in Terminal and CocoaPods does the rest for you. After that you'll have to open your projects via a .xcworkspace file in your app's root folder which has all the pods in it you just added, ready to use.

Carthage does a similar job. You add a text file named 'Cartfile' to your app's root folder and add 'github "FrameworkName"'. After

that run a simple 'carthage update' in Terminal and carthage adds and builds the libraries. After that you'll find a folder named Carthage in your app's root folder. In that folder there's a 'Build' folder from where you have to drag the frameworks you want to use into the "Linked Frameworks and Libraries" section in Xcode. So, with that here are the differences:

First of all the setup. To set up CocoaPods for your machine, you have to run 'sudo gem install cocoapods' in Terminal. This adds all necessary gems required to install pods. For Carthage there are two ways: Either you run 'brew install carthage' when you've got Homebrew installed, or you download the .pkg from <https://github.com/Carthage/Carthage/releases> and install it manually.

Next is usage. While CocoaPods is pretty easy to use, Carthage is more flexible. CocoaPods creates a .xcworkspace file in which all your pods are listed. This makes it easier to use, but also less error proof. If one library fails to build, your whole project doesn't work. In Carthage you have to add the frameworks you really want to use into your project manually. This means you have to do some extra work, but it also adds a lot more flexibility. Also if you want to get rid off Carthage, you just have to delete the Cartfile and Cartfile.resolved files and the Carthage folder. I couldn't find such an elegant solution for CocoaPods. First of all you have to delete the .xcworkspace, the Podfile and Podfile.lock files, the Pods folder and a bunch of other stuff inside Xcode.

The only downside I found about Carthage is that not all frameworks are available for Carthage. Especially older frameworks aren't available for Carthage, and if you don't want to use CocoaPods for them you have to add those manually. Just run

terminal and run ‘git submodule add (link to framework)’.

## Conclusion

Whatever dependency manager you search for, both CocoaPods and Carthage do a pretty good job. If you want something easy to use, which has been there for years, take CocoaPods. If you want a fresh approach, which is almost completely written in Swift and gives you more flexibility, go for Carthage. It already works great and over time more authors will support Carthage.

I hope I could help a little with making a choice between two great tools, that are designed to make our lives better.

[Swift](#)[Cocoapods](#)[Carthage](#)

**Medium**

[About](#)[Help](#)[Legal](#)