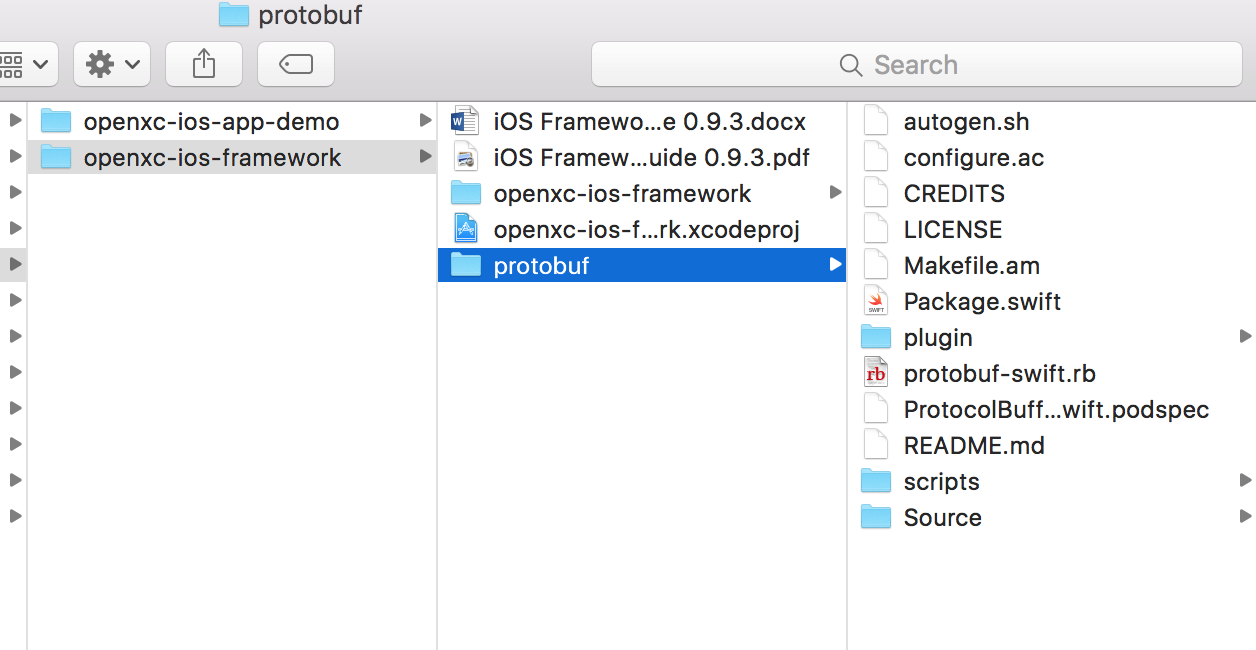
Build OpenXC iOS Framework

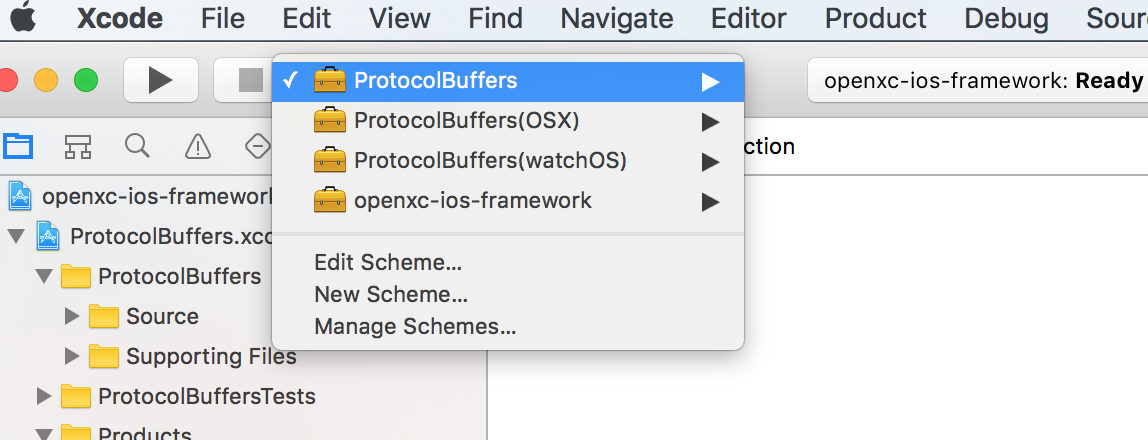
# **Steps:**

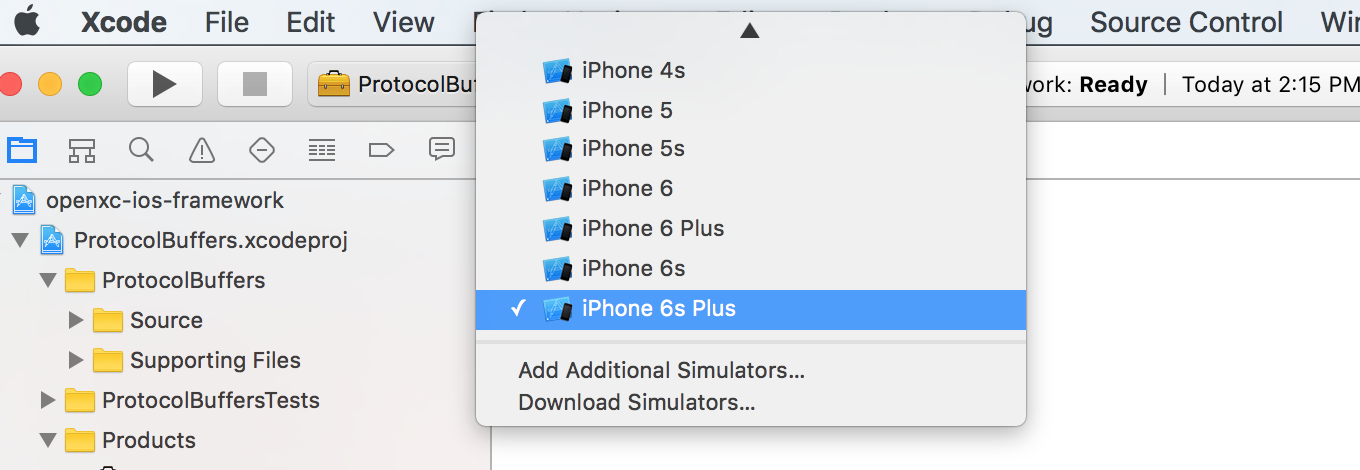
* Fork the framework & demo app repository on GitHub and follow steps from [here](https://github.com/openxc/openxc-ios-app-demo/blob/master/CONTRIBUTING.mkd). Once forked, you will see the repository structure as below. You may also download the repository through [SourceTree](https://www.sourcetreeapp.com/) using URL link.



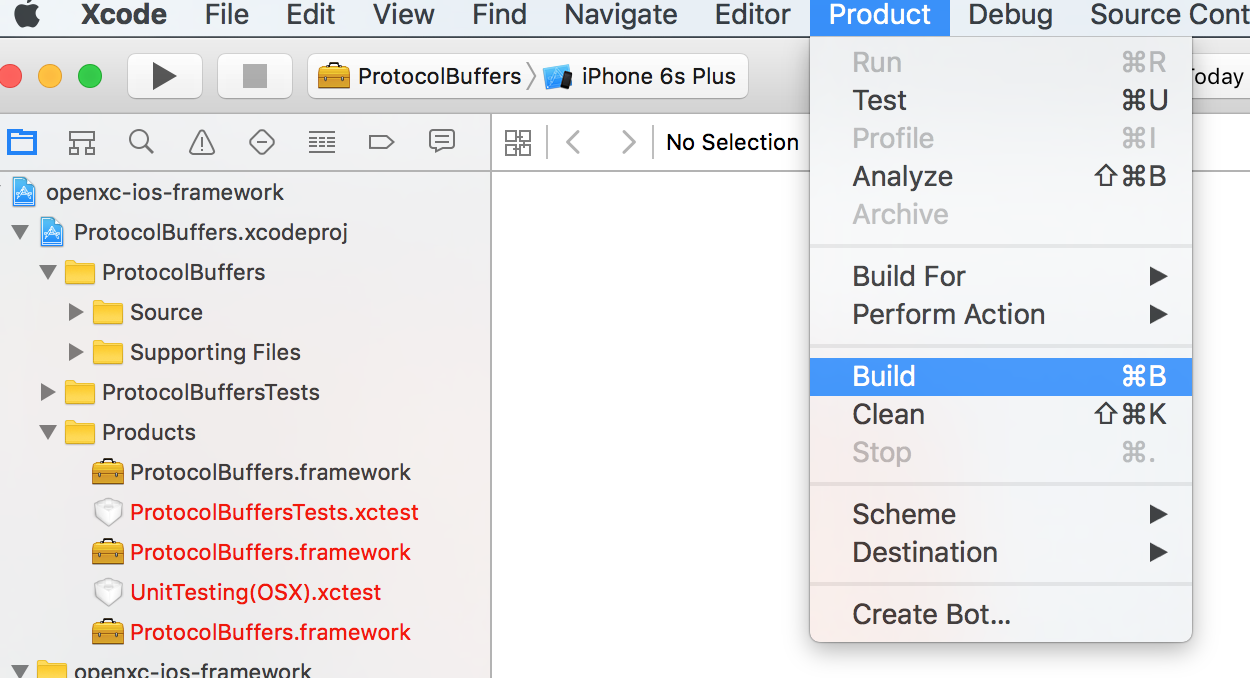
Make sure “protobuf” framework is also cloned with these files. If its not cloned, please run “git submodule update --init --recursive”

* Open the “openxc-ios-framework,xcodeproj” using XCode
* Select “ProtocolBuffer” scheme and a simulator/device version for which the build is required.

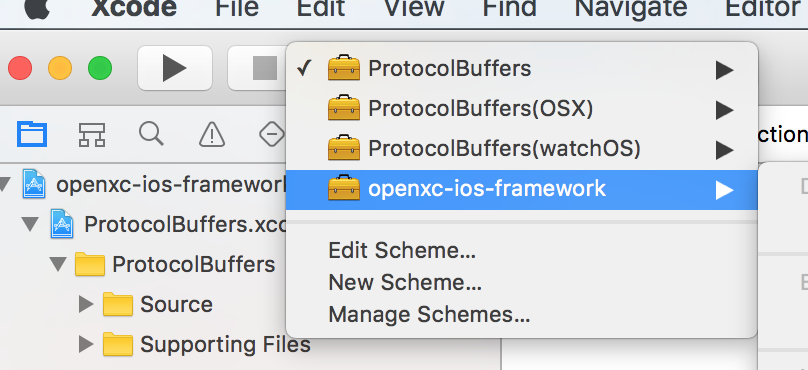


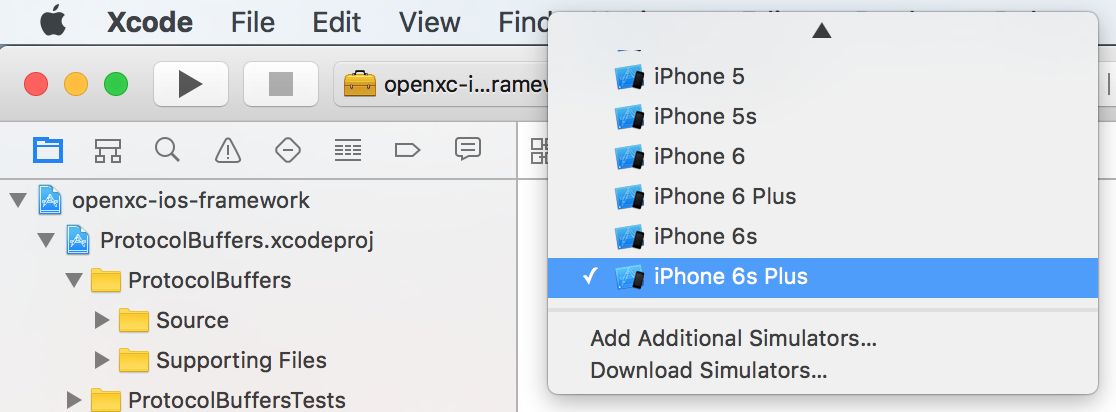


* Build the “ProtocolBuffer” framework for the selected simulator/device version.

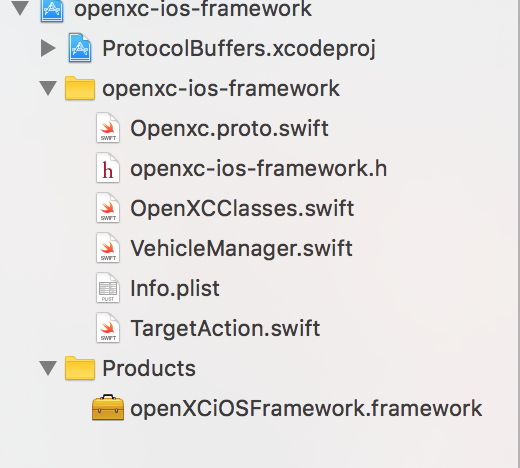


* “Build Success” message should be seen on the screen
* Once “ProtocolBuffer” framework is built, select “openxc-ios-framework” scheme for same simulator/device version and build.

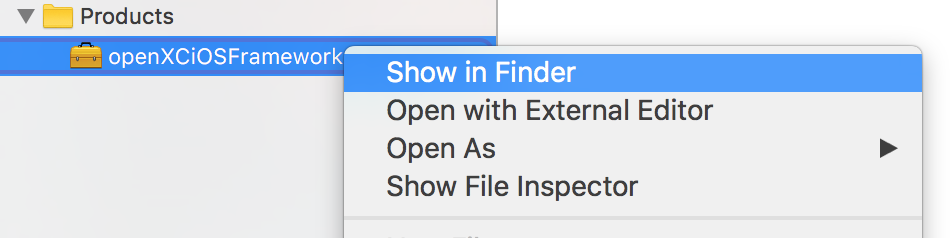


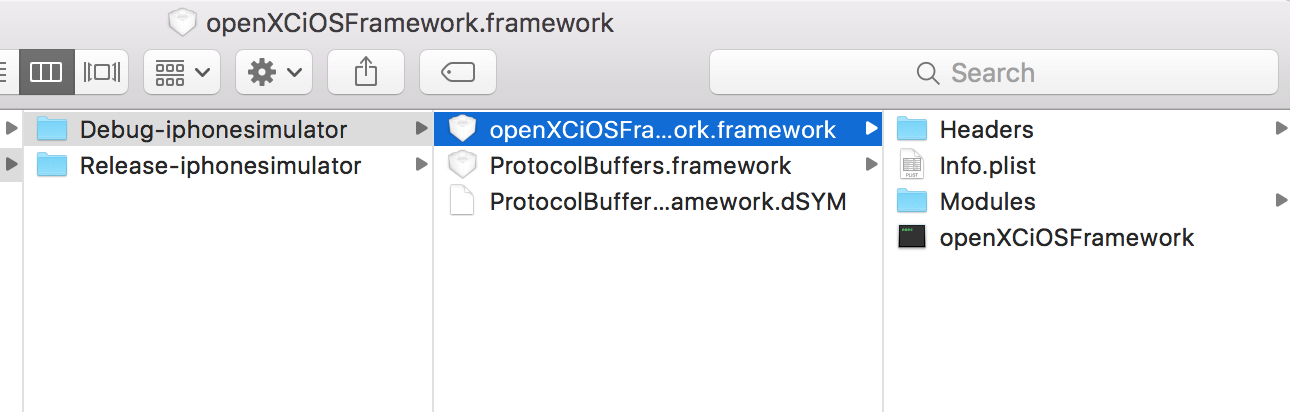


* Once this framework is built, color of the “Products-openXCiOSFramework.framework” is built and color changes from red to black.



* Right click on this product – “Show in Finder” will give you the product location.



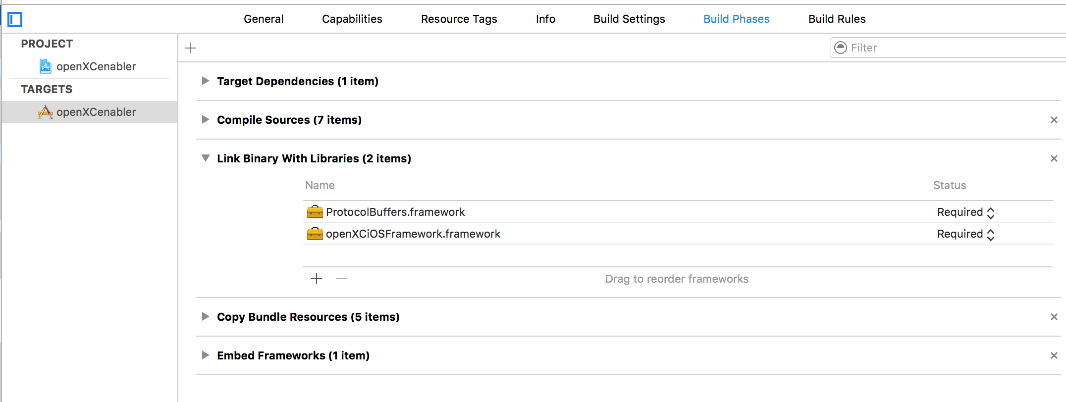


This framework can be included in any application that needs to use openXC-iOS-Framework.

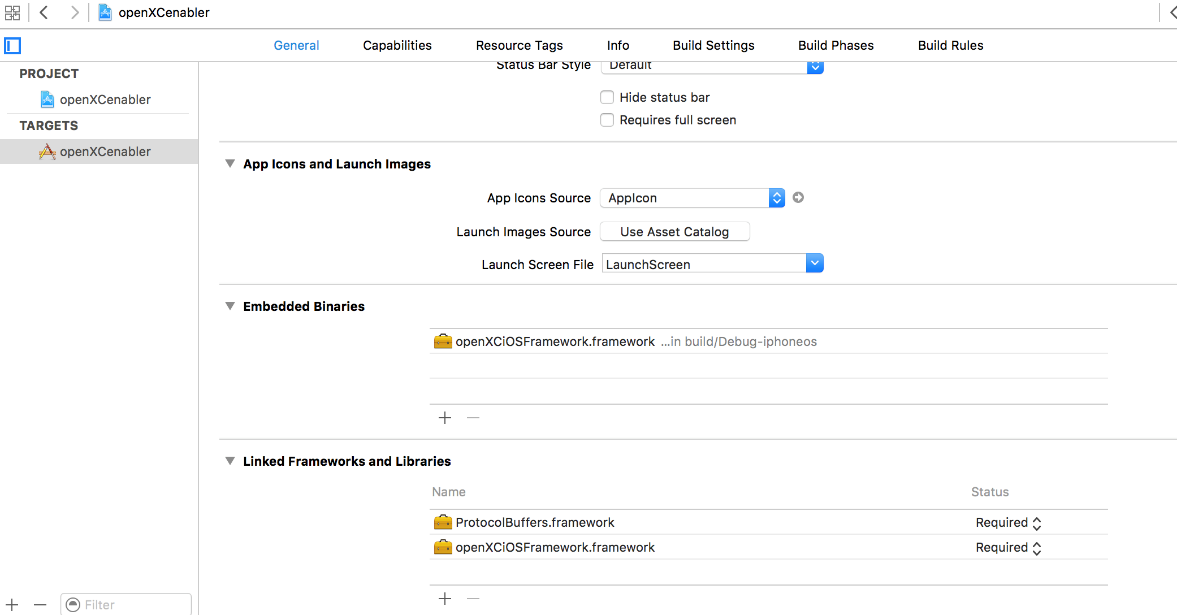
Build OpenXC iOS Demo App

# **Steps:**

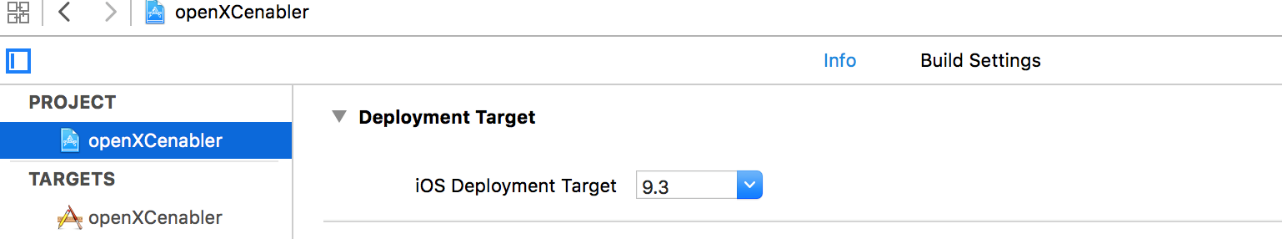
* Add the above frameworks – openXC-iOS-Framework & Protocol.Buffers Framework in “openXCenabler target - Build Phases- Link Binary With Libraries”.



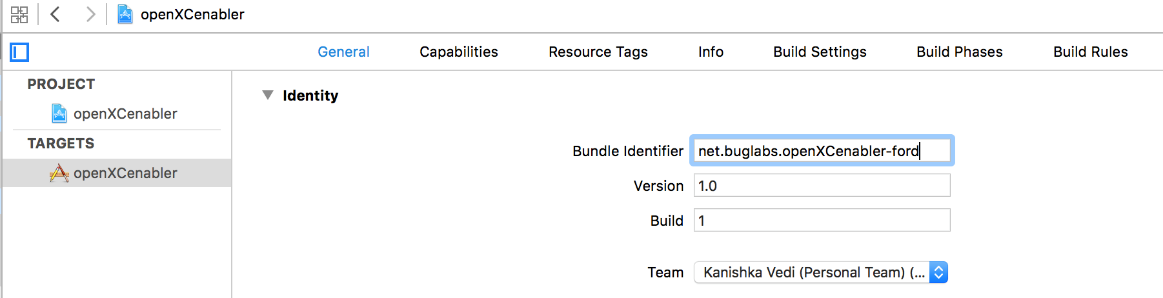
* Also add it as “Embedded Binaries” in “openXCenabler target- General”



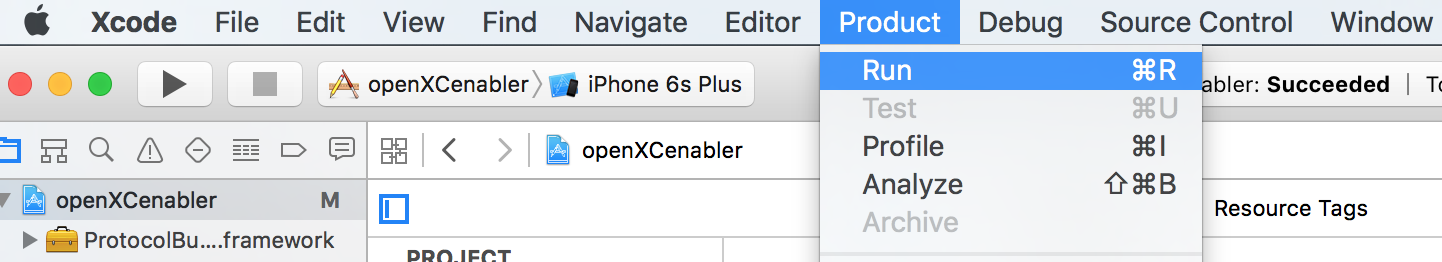
* Make sure the deployment target is 9.3



* Change the bundle identifier and profile/team as per your apple account details



* Now run the application



This way the framework can be added into any application that requires OpenXC iOS Framework.

Once the app is built successfully and runs on the simulator, this is how the app will look like.

