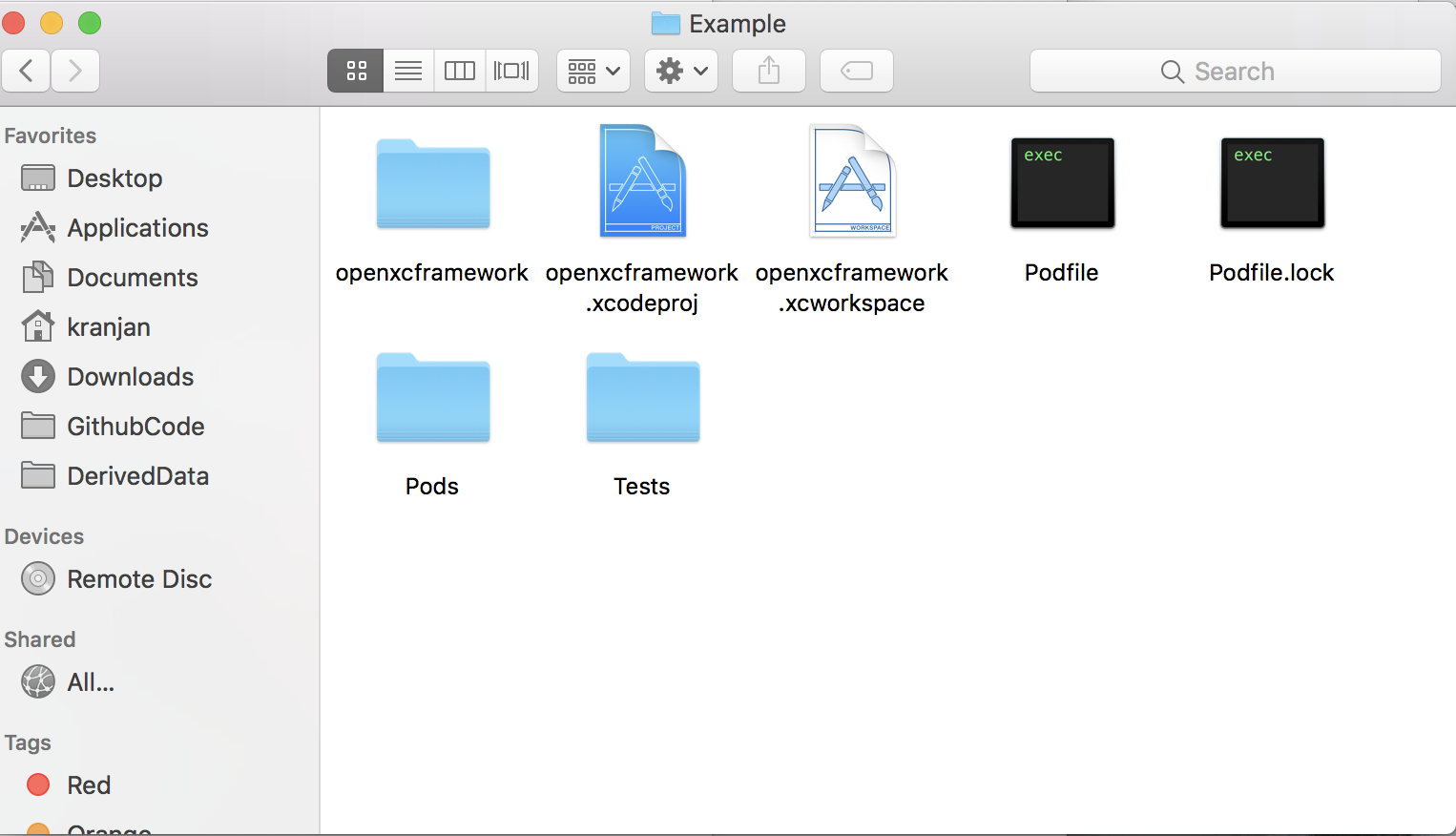
Build OpenXC enabler iOS app

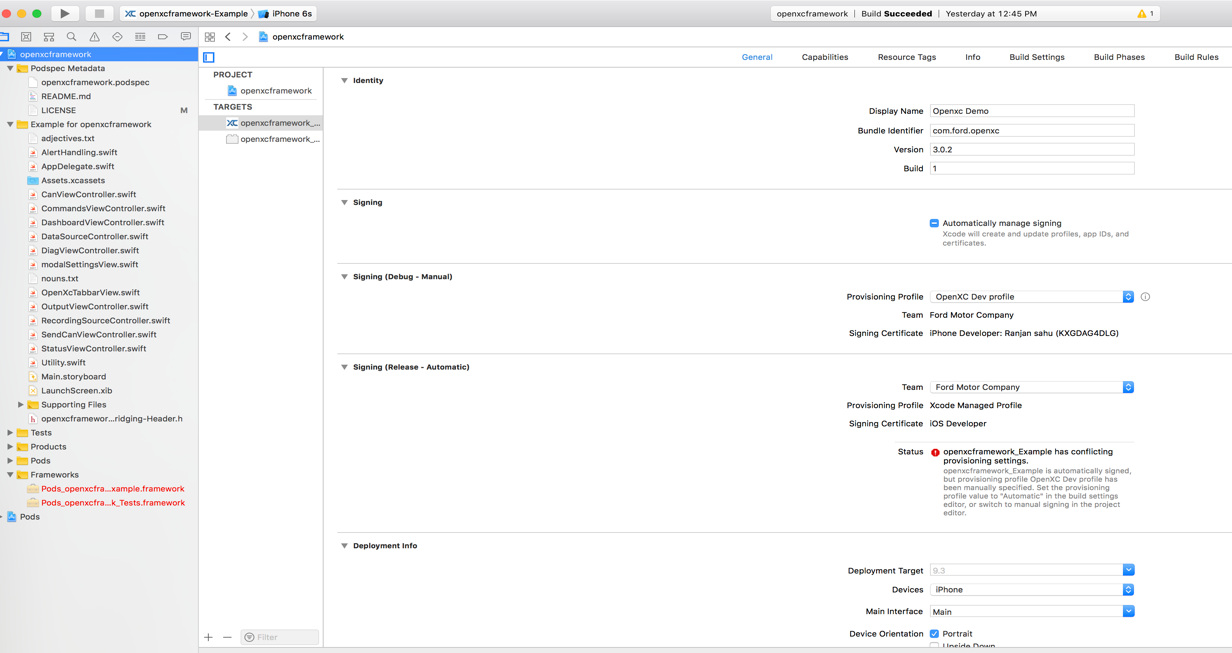
# **Steps for Openxc Enabler app setup Using Cocoa pod:**

* Install XCode 8 or latest in your machine if it is not installed.
* Create an empty folder on your machine and name it accordingly.
* Open terminal and go to that folder “**cd <folder Path>**” .
* Copy the link from GitHub which you want to clone (master or next ) branch .
* Use command in terminal - “**git clone <link>**” and press enter. It will start cloning the project inside your folder.
* After cloning, we need to go to Example folder “**cd <Example>**”.
* Their need to run “**pod install**” from terminal.
* After that open the open **.xcworkspace** file and run the app in device or simulator.

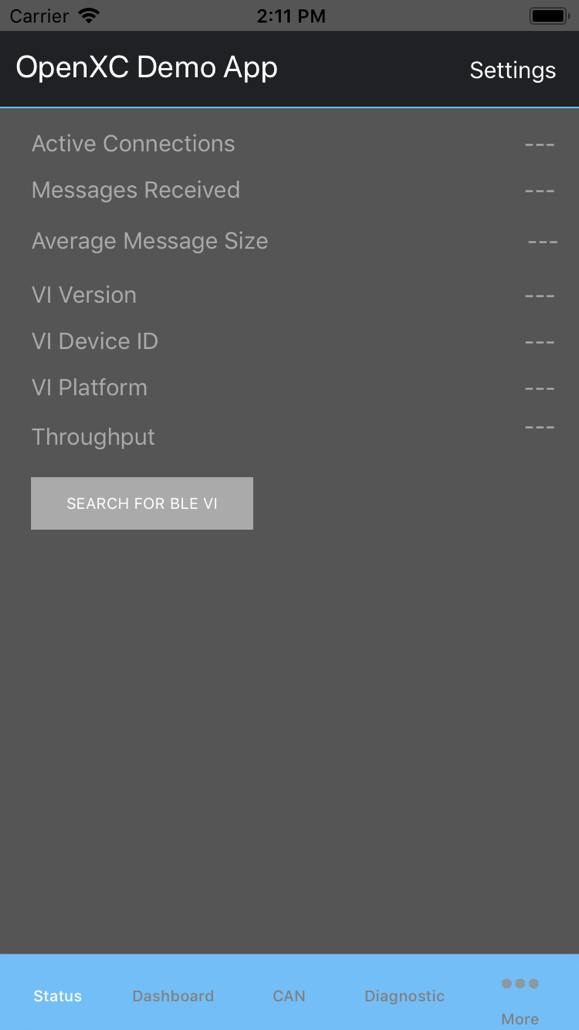
In example folder you will find openxcframework.xcworkspace



After Opening the openxcframework.xcworkspace file



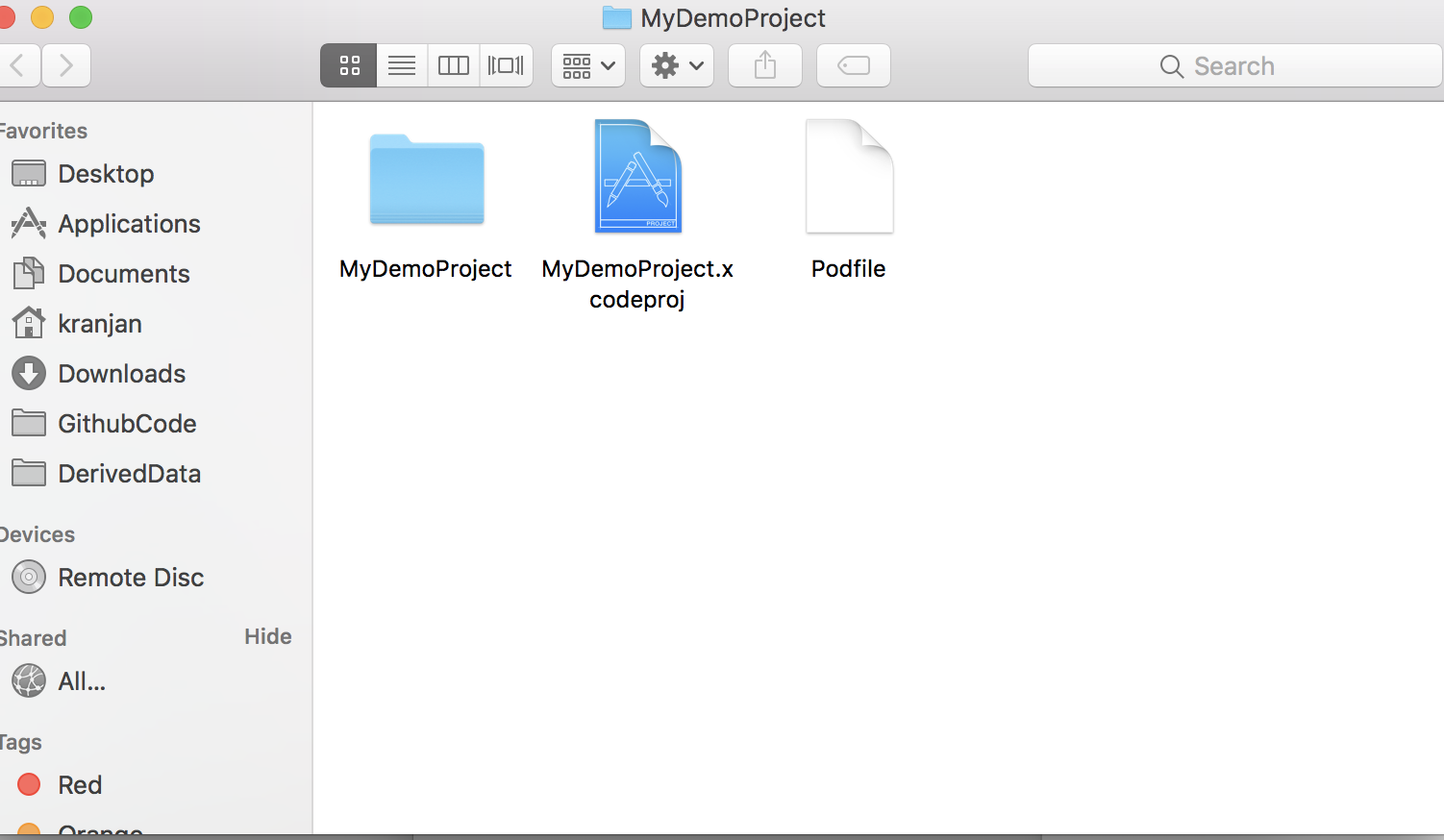
Once you run the app you will get this screen:



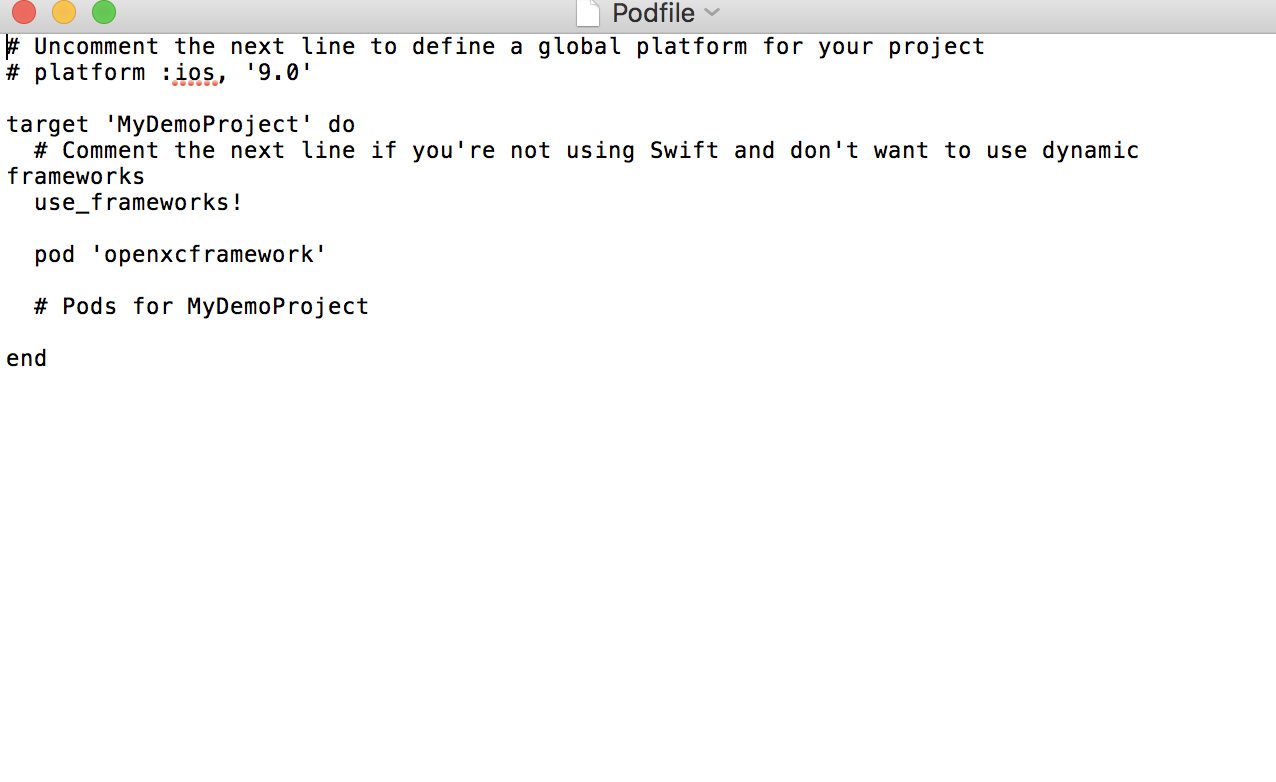
# **Steps for Openxc-ios-library setup in new app Using Cocoa pod:**

* Install XCode 8 or latest in your machine if it is not installed.
* Open xcode create a new project, if you have existing project.
* Open terminal and go to that folder(new project/existing project) “**cd <folder Path>**” .
* Run the command from terminal **“pod init”**
* After that you will find a pod file in your project folder.
* Open the pod file and add **“pod openxcframework”.**
* Save the file and run the command **“pod install”** in terminal.
* After this you will get the **.xcworkspace** file in your project folder.
* Open the file you will find that pod has added the **openxcframework** to your project
* For time to time update just run **“pod update”** in terminal it will update the framework if update available.

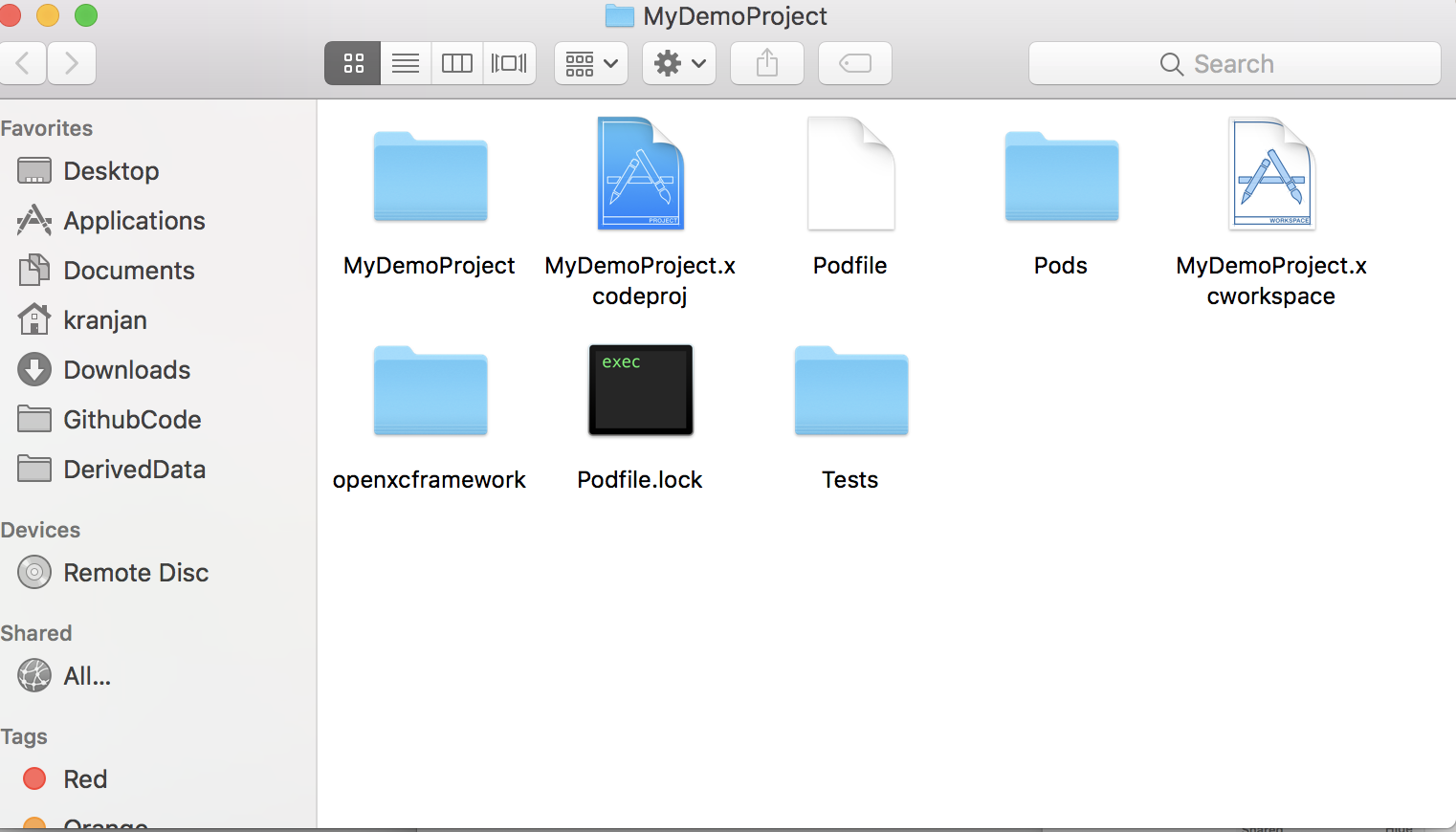
After running the command pod int



After adding pod into pod file .



After pod install command the project file Structure.

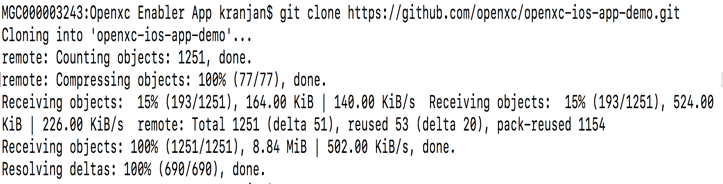


# **Steps for Openxc Enabler app setup Using submodule:**

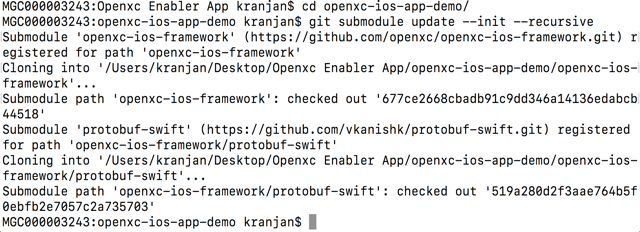
* Install Xcode 8 or latest in your machine if it is not installed.
* Create an empty folder on your machine and name it accordingly.
* Open terminal and go to that folder “**cd <folder Path>**” .
* Copy the link from GitHub which you want to clone (master or next ) branch .
* Use command in terminal - “**git clone <link>**” and press enter. It will start cloning the project inside your folder.
* After cloning,we need to get the frameworks(submodules)
* In terminal, **cd** **Openxc-ios-app-demo**.
* Run “ **git submodule update --init --recursive** ” .
* Your demo app,openxc iOS framework and protobuf framework are ready to use now.
* Now simply open the openxc Enabler project file in XCode, build and run the app in device or simulator.

Below are the screen shots from terminal for your reference.

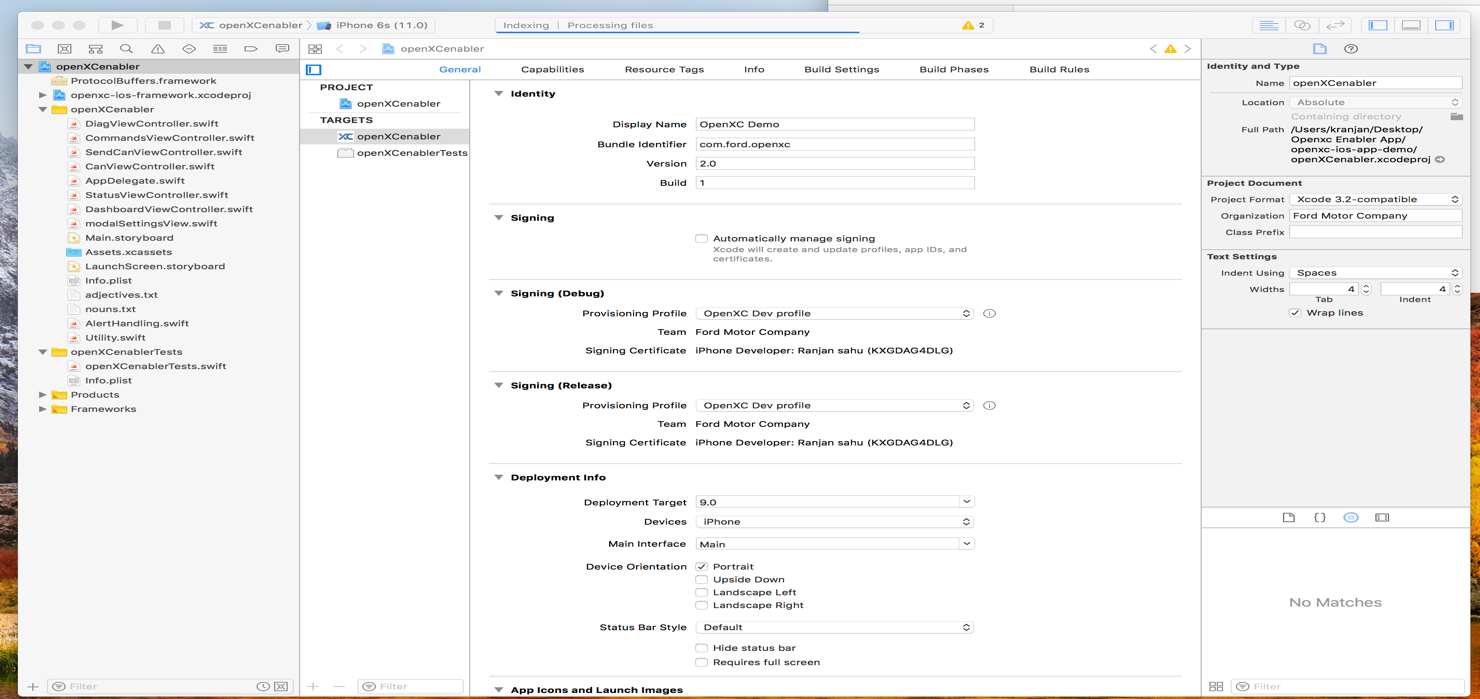
**Clone:**



**Submodule update:**



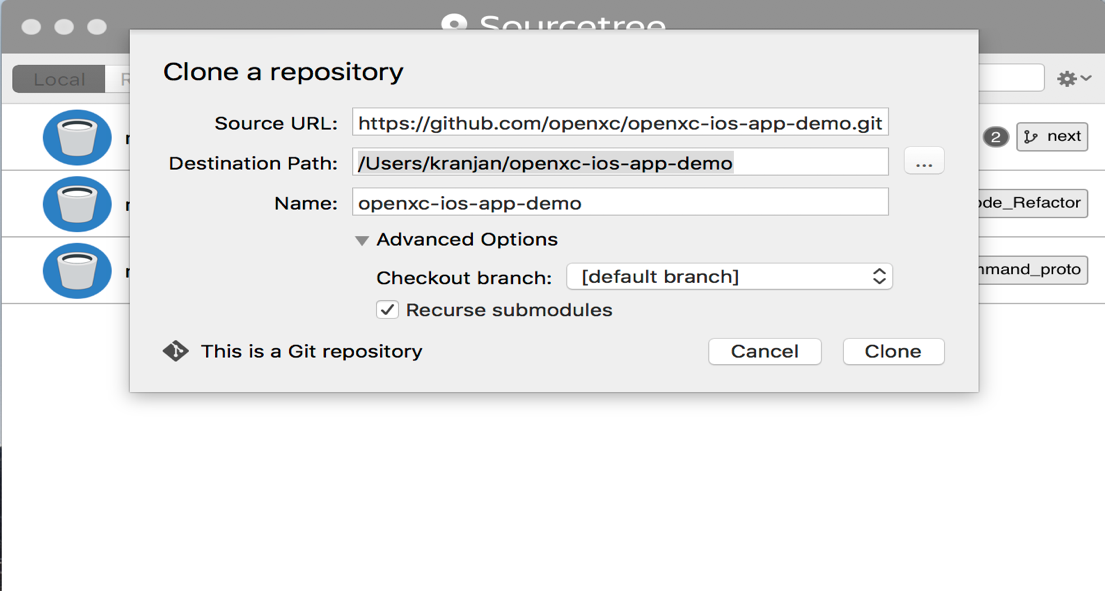
After following all these Steps when you open OpenXC app in XCode, the screen will be like below.



# **Steps to use OpenXC Frame work without demo app:**

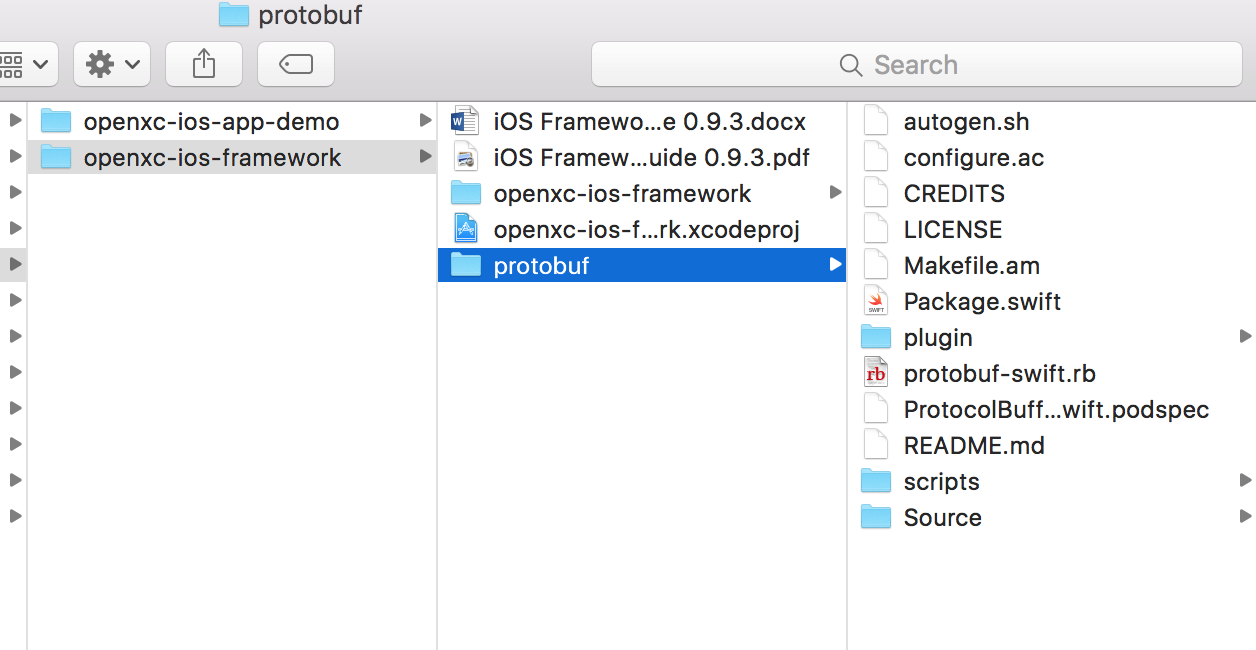
* Download or clone OpenXC framework from Github.Then drag the openxc framework project file to your App.
* Same process for protobuf .Download or clone the particular this branch “**protobuf 2.6 swift 3.0** “ and then drag the protobuf projectfile to your app.
* Then follow the below step to build openxc frame work and protobuf to run in your project .

**NOTE:-** If you are using **“SourceTree”** instead of terminal to clone openxc enabler app then no need to run git submodule separately it will automatically do it for you. Just put tick mark in the “**Recurse Submodule box”**



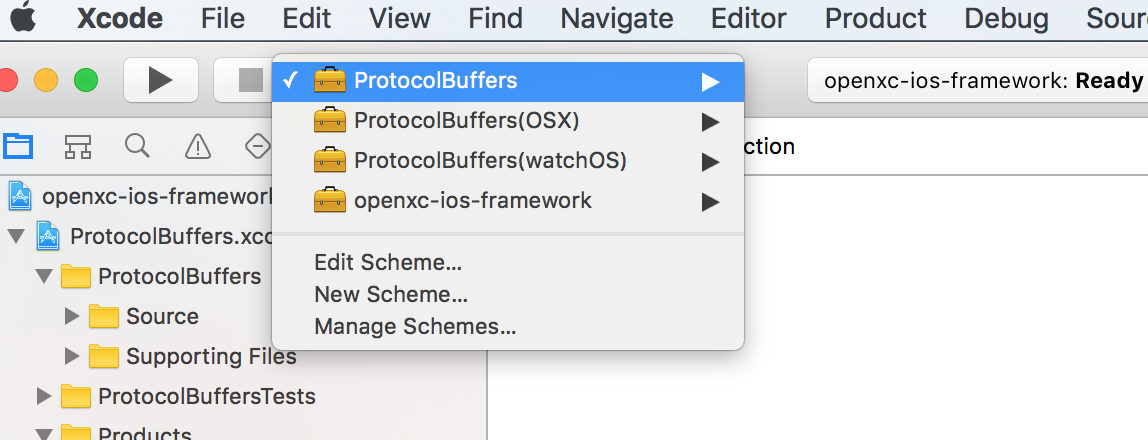
# **Steps:**

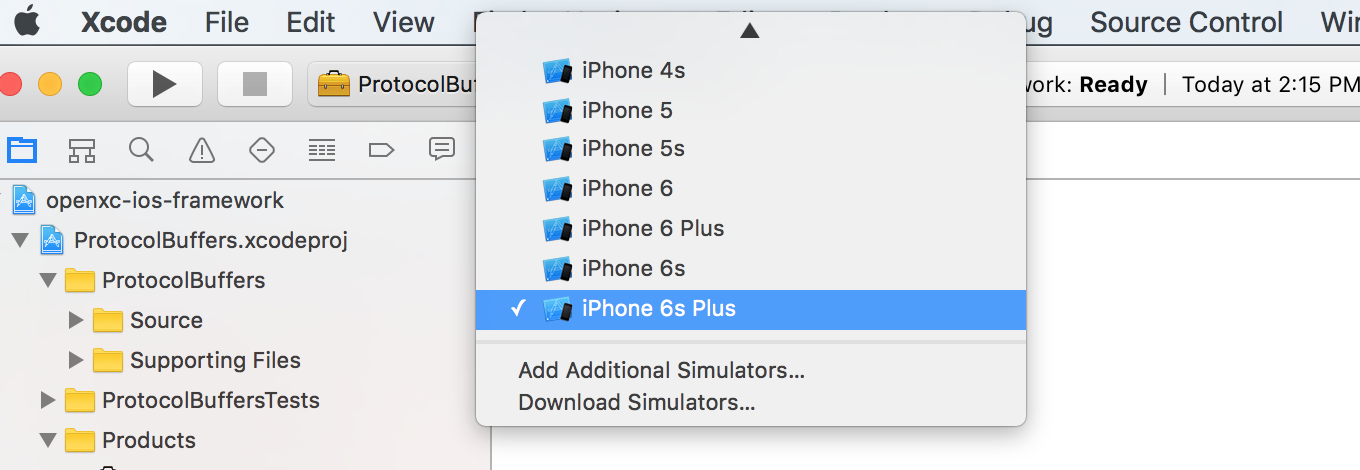
* Fork the framework & demo app repository on GitHub and follow steps from [here](https://github.com/openxc/openxc-ios-app-demo/blob/master/CONTRIBUTING.mkd). Once forked, you will see the repository structure as below.



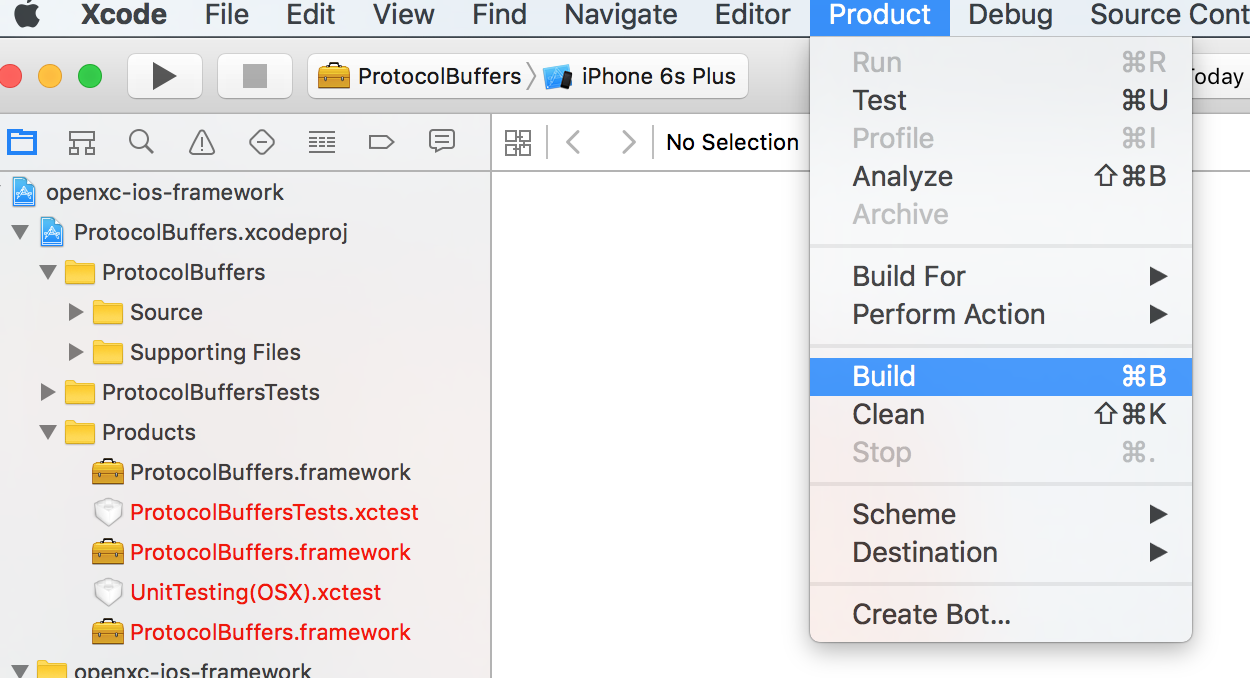
Make sure “protobuf” framework is also cloned with these files.

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

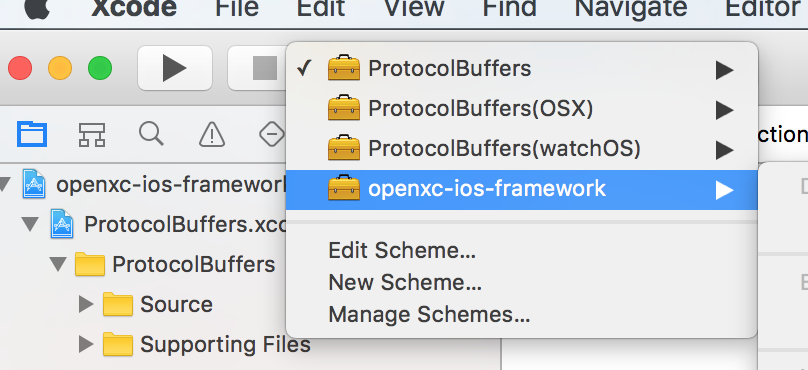


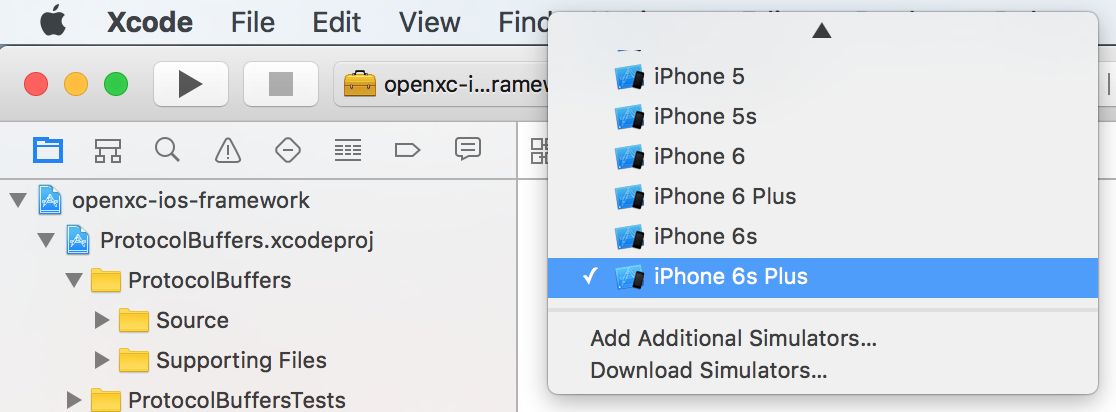


* Build the “ProtocolBuffer” framework for the selected simulator/device version.

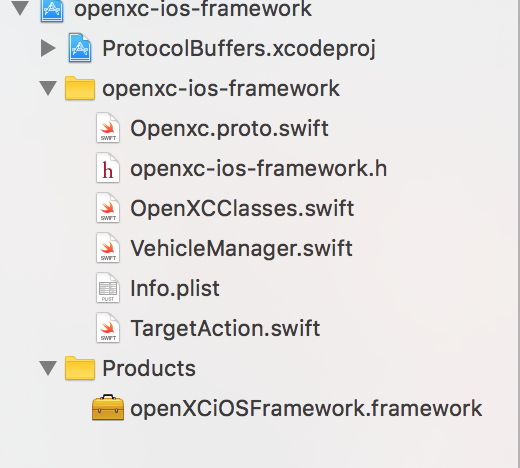


* “Build Success” message should be seen on the screen
* Once “ProtocolBuffer” framework is built, select “openxc-ios-framework” scheme for same simulator/device version and build.

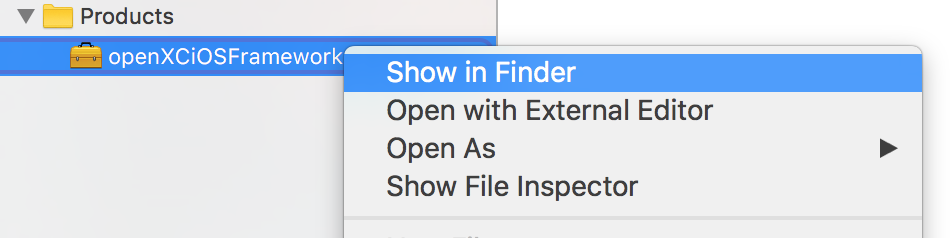


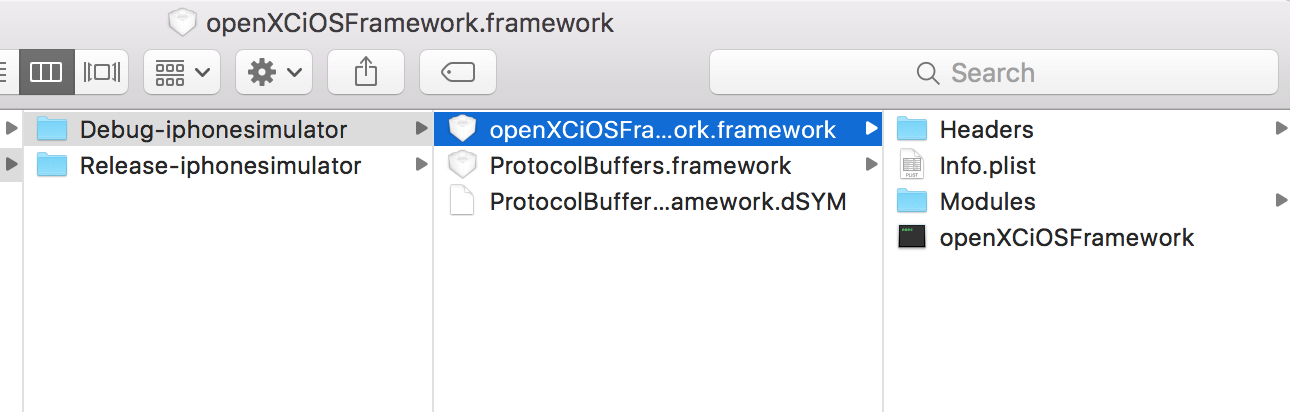


* Once this framework is built, color of the “Products-openXCiOSFramework.framework” is built and color changes from red to black.



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



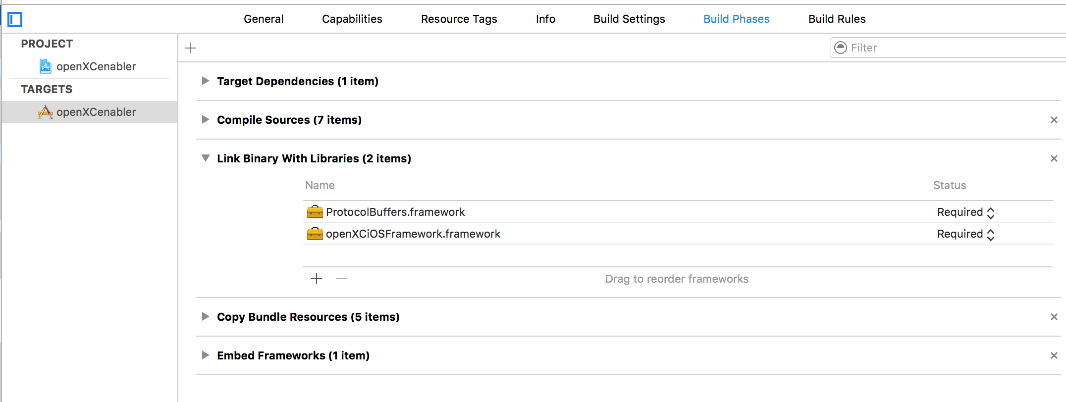


This framework can be included in any application that needs to use openXC-iOS-Framework.

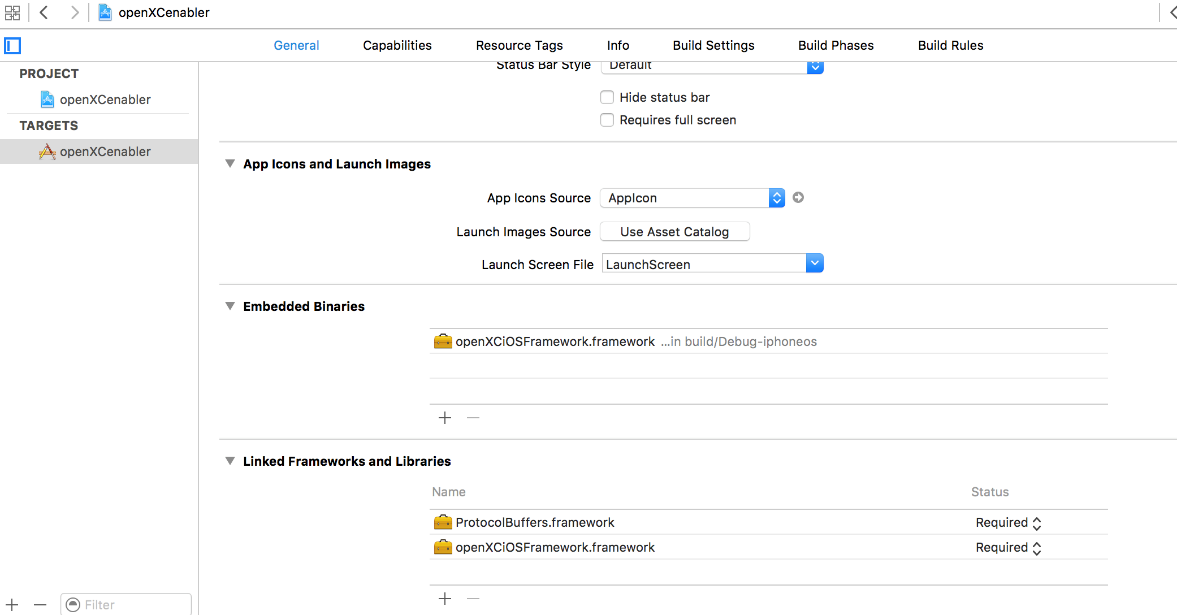
Build OpenXC iOS Demo App

# **Steps:**

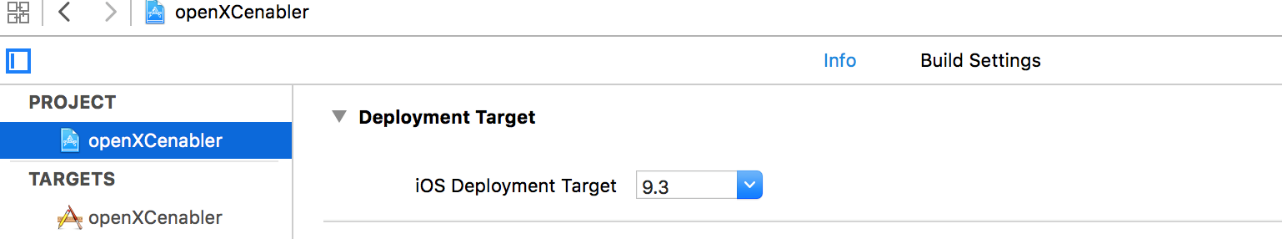
* Add the above frameworks – openXC-iOS-Framework & Protocol.Buffers Framework in “openXCenabler target - Build Phases- Link Binary With Libraries”.



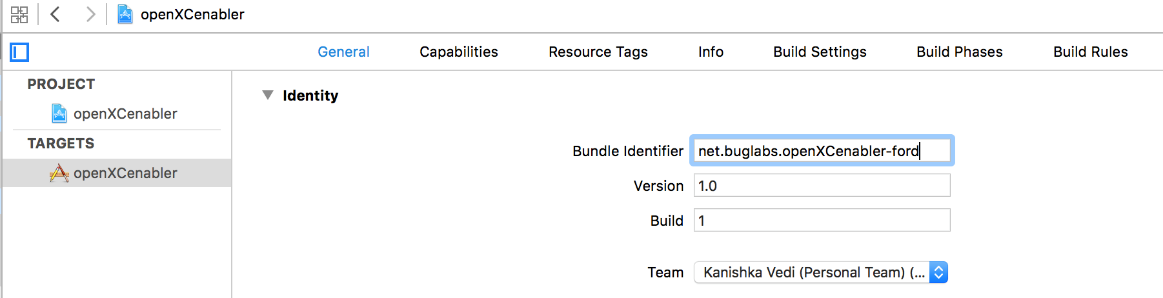
* Also add it as “Embedded Binaries” in “openXCenabler target- General”



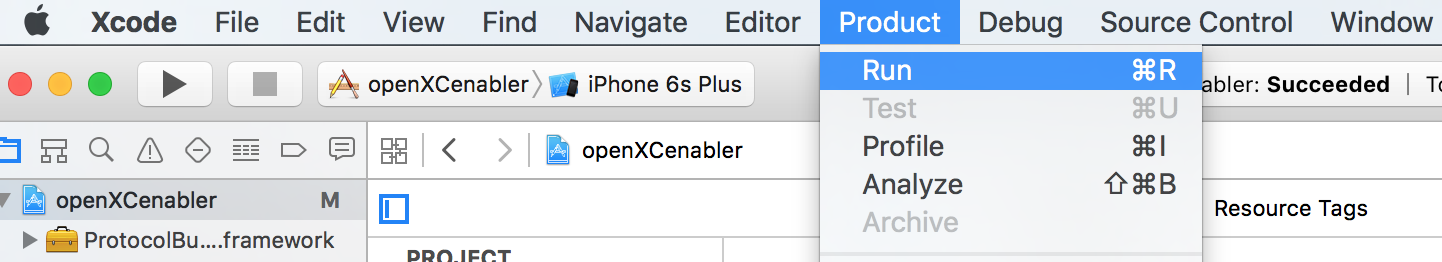
* Make sure the deployment target is 9.3



* 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.

