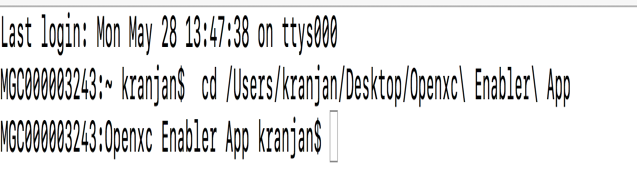
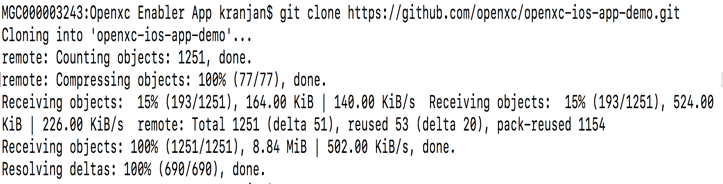
Build OpenXC enabler iOS app

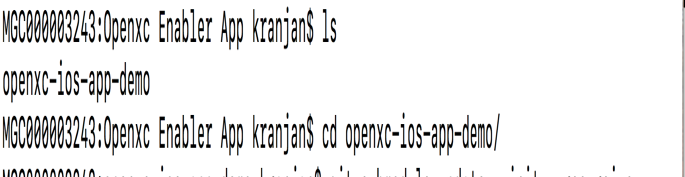
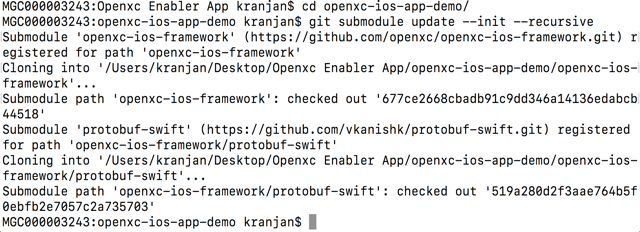
# **Steps for OpenXc Enabler app setup in your mac machine:**

* Install Xcode 9 or latest in your machine if it is not installed.
* Create an empty folder on your machine name it accordingly.
* Open terminal and go to that folder “**cd <folder Path>**” or you can simply drag the folder to terminal after **cd space** .
* Then copy the link from GitHub which you want to clone (master or next ) branch .
* Then type “**git clone <link>**” and press enter it will start cloning the project inside the folder.
* Once the clone done if you check there is OpenXC framework folder which is empty To include framework to project we need to run git submodule command inside “**Openxc-ios-app-demo** “folder.
* Go to **OpenXC-ios-app-demo** folder using command cd <folder name> After that run this command “ **git submodule update --init --recursive** ” inside folder.
* Once it finish now you can check inside the OpenXC frame work folder now you will find the files and protobuf folder also .Now you have did the setup and everything is included .
* Now simply open the OpenXC Enabler project file in Xcode build the app and then run the app in device or simulator which ever you want to

Below here you can follow these screen shots in terminal.

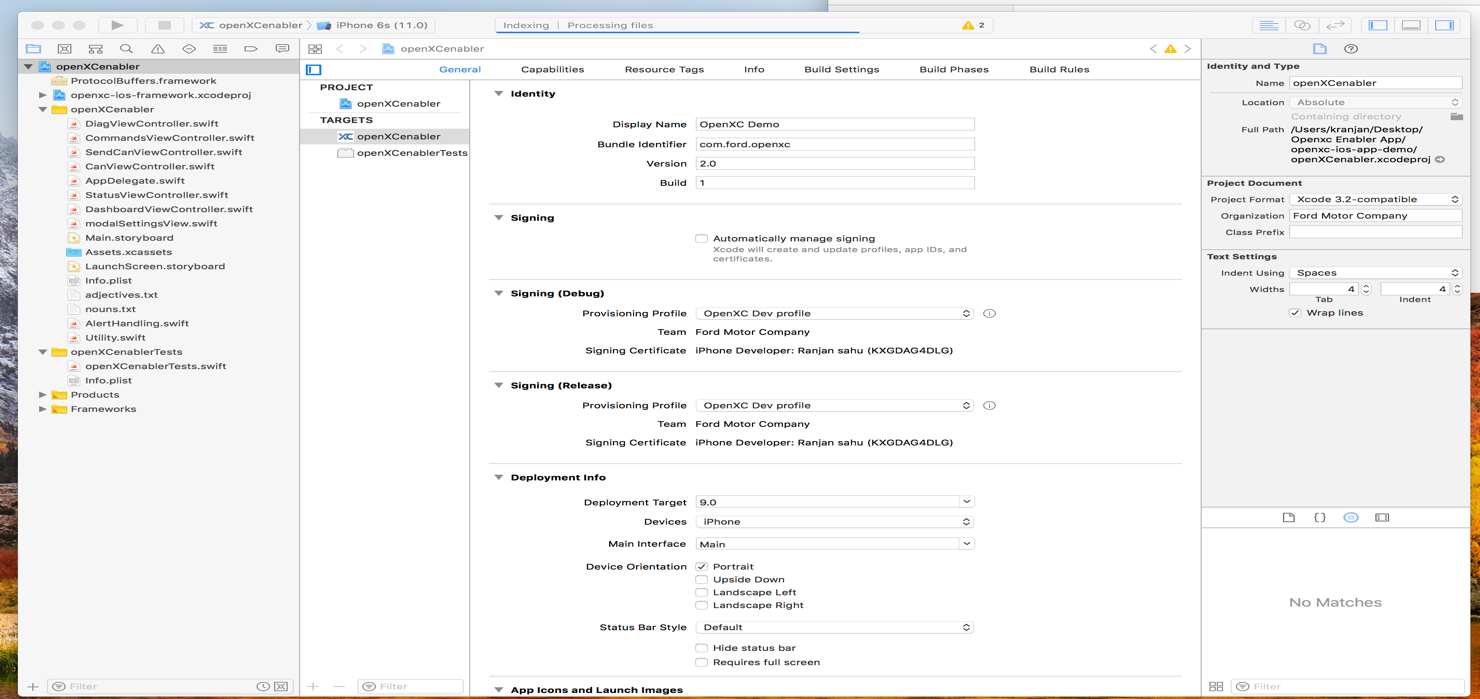
 

(Step 1) (step 2)

(Step 3) (Step 4)

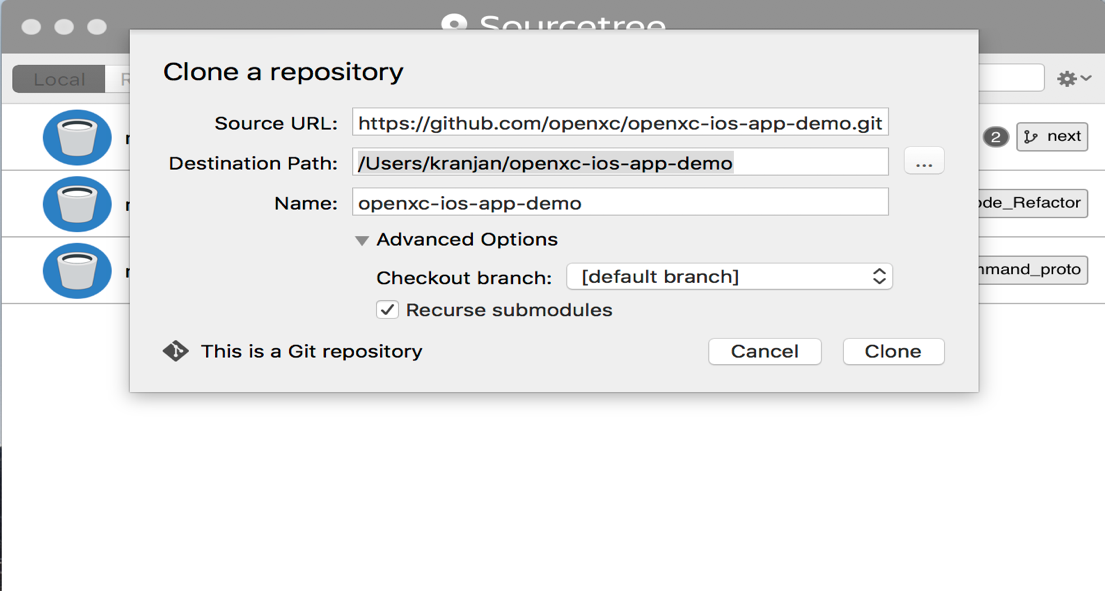
After following all these Steps when you open OpenXC app in your project file in xcode the screen will be like below that means you have done the setup .



# **Steps to include OpenXC & protobuf Frame work in other app:**

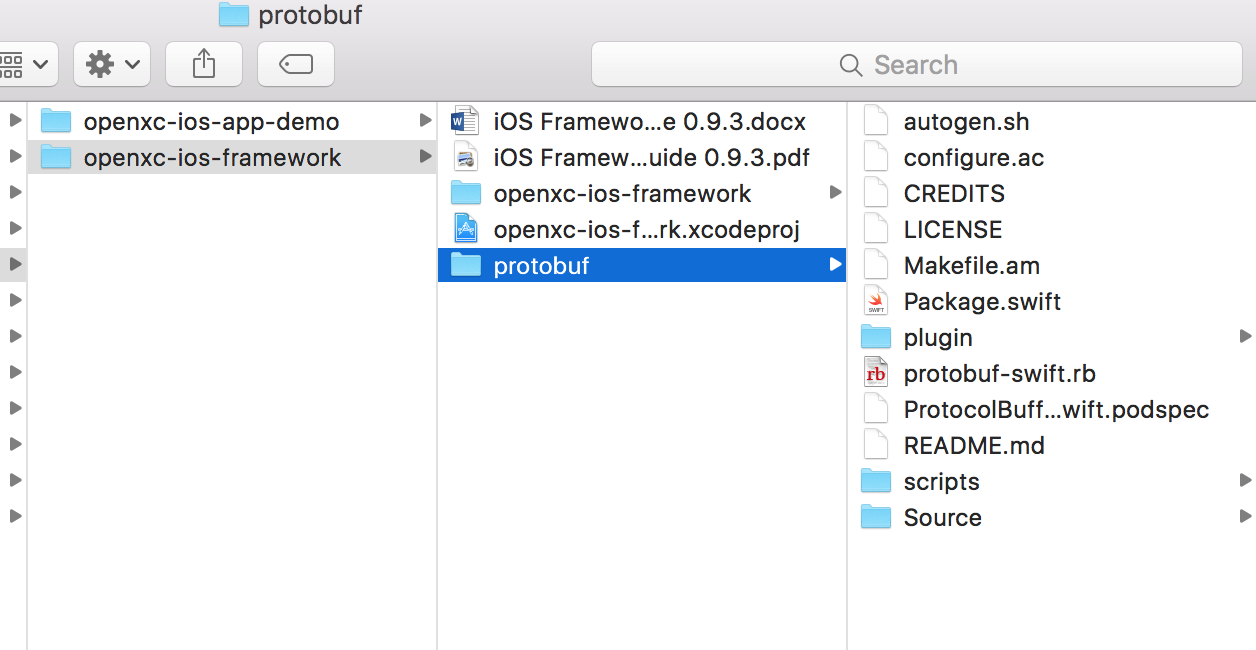
* Download or clone OpenXC framework from Github.Then drag the openxc framework project file to your App.
* Same process for protobuf .Download or clone the particular this branch “**protobuf 2.0 swift 3.0** “ and then drag the protobuf projectfile to your app.
* Then follow the below step to build openxc 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 “**Recurce Submodule box”**



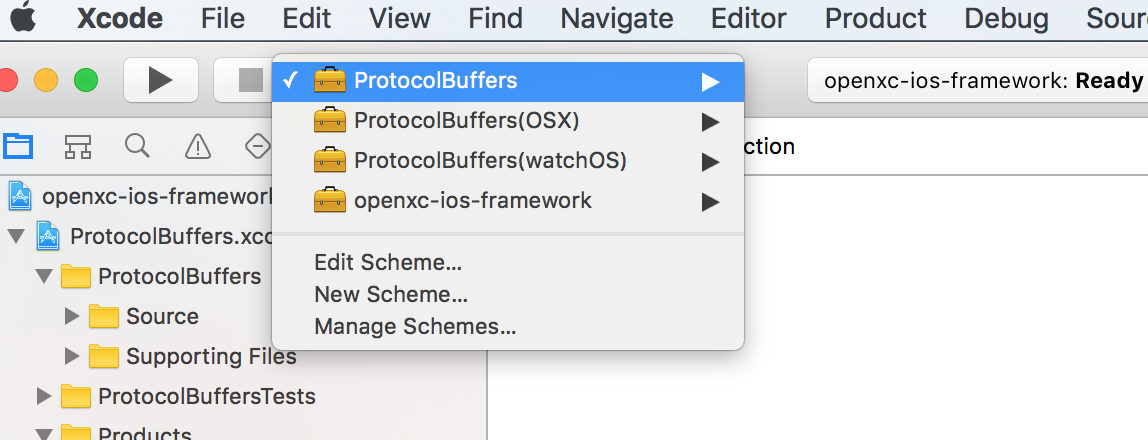
# **Steps:**

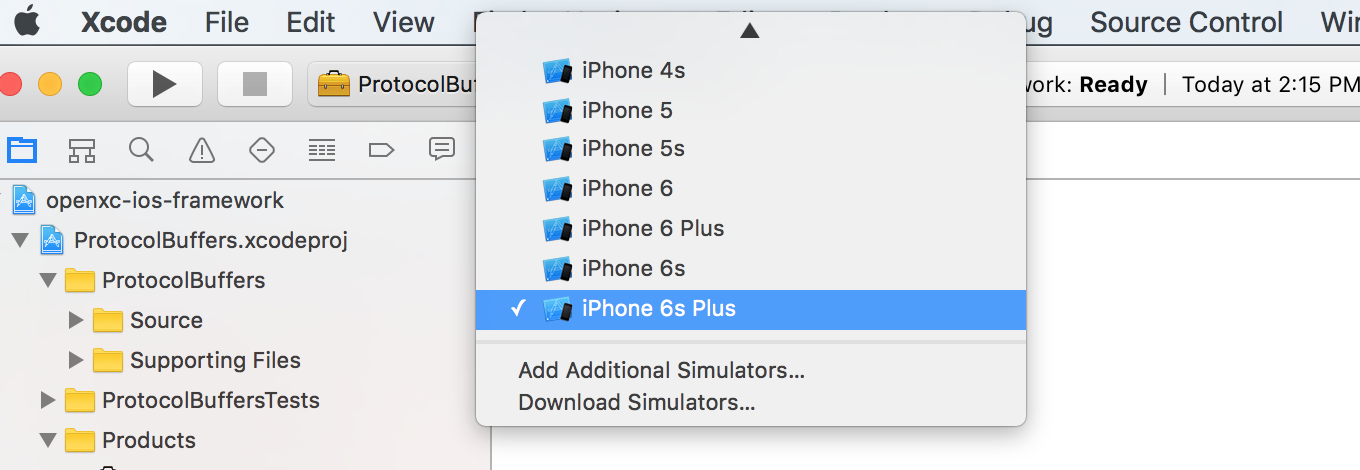
* Fork the framework & demo app repository on GitHub and follow steps from [here](https://github.com/openxc/openxc-ios-app-demo/blob/master/CONTRIBUTING.mkd). Once forked, you will see the repository structure as below.



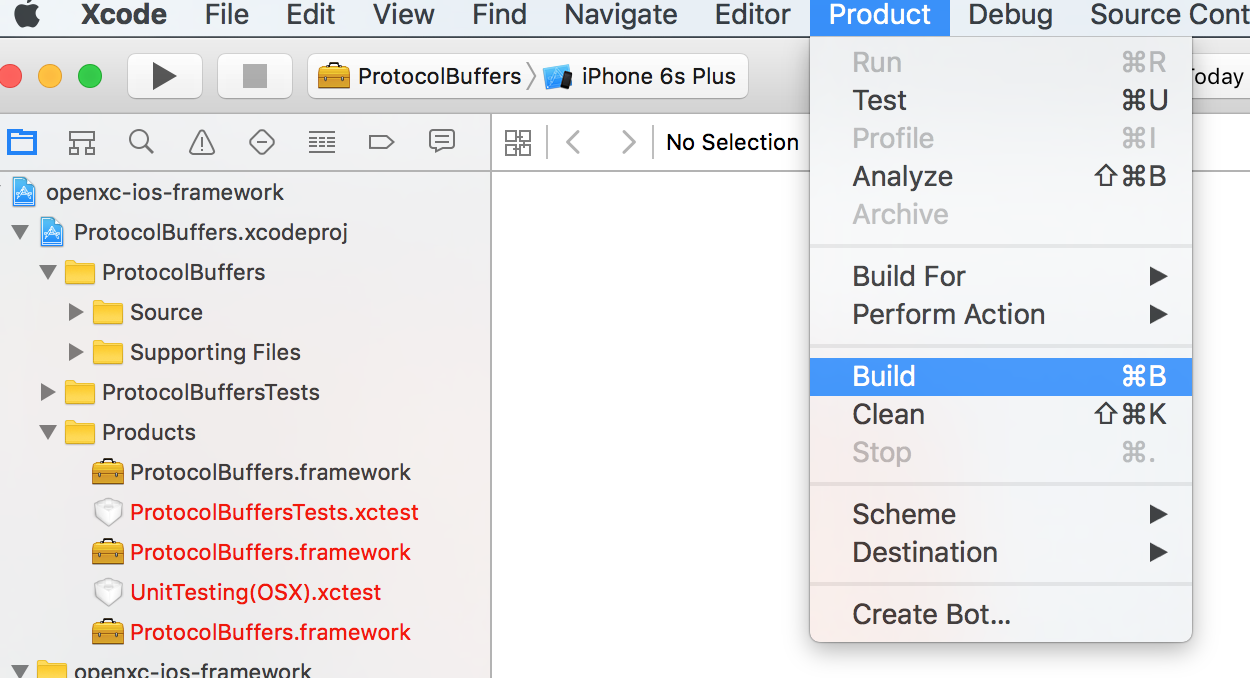
Make sure “protobuf” framework is also cloned with these files.

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

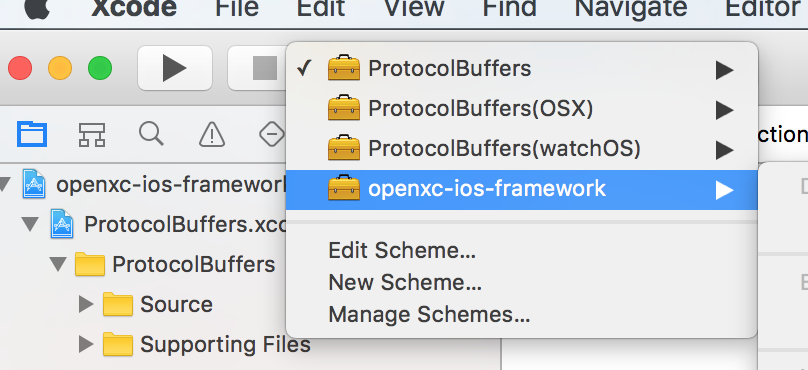


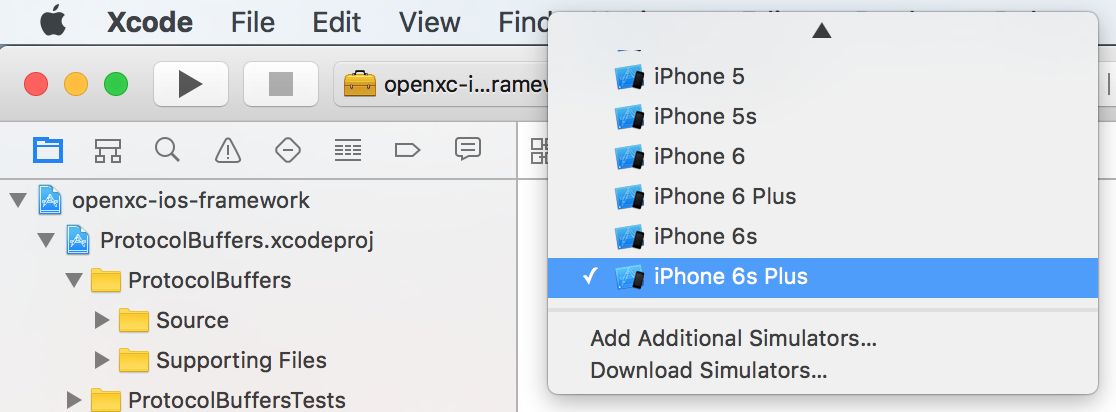


* Build the “ProtocolBuffer” framework for the selected simulator/device version.

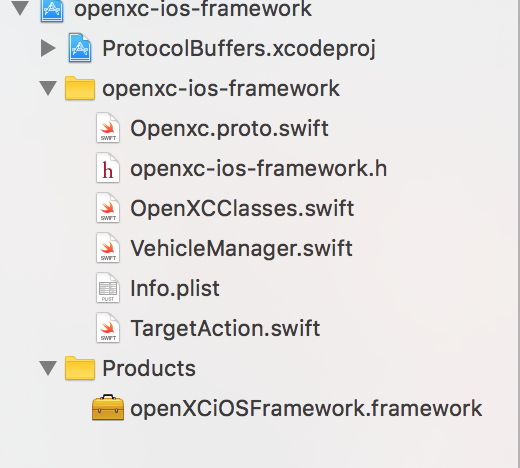


* “Build Success” message should be seen on the screen
* Once “ProtocolBuffer” framework is built, select “openxc-ios-framework” scheme for same simulator/device version and build.

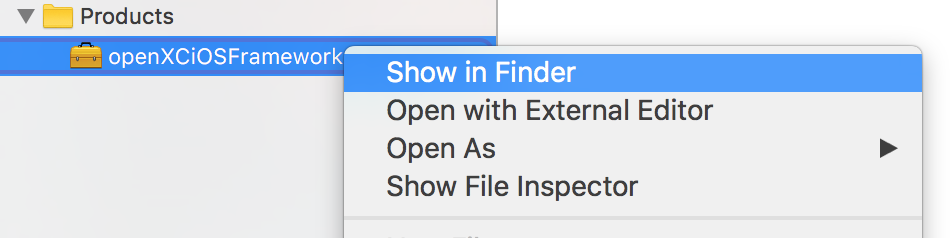


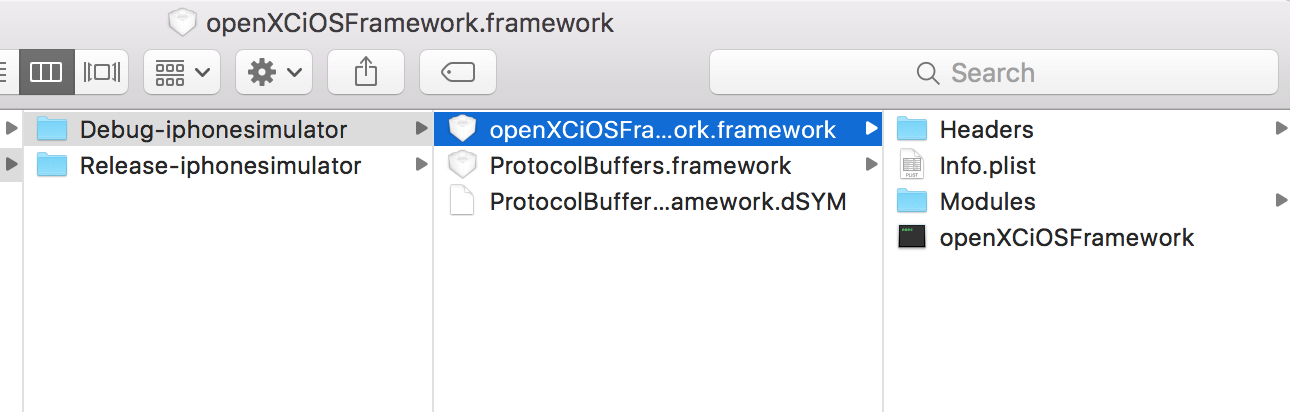


* Once this framework is built, color of the “Products-openXCiOSFramework.framework” is built and color changes from red to black.



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



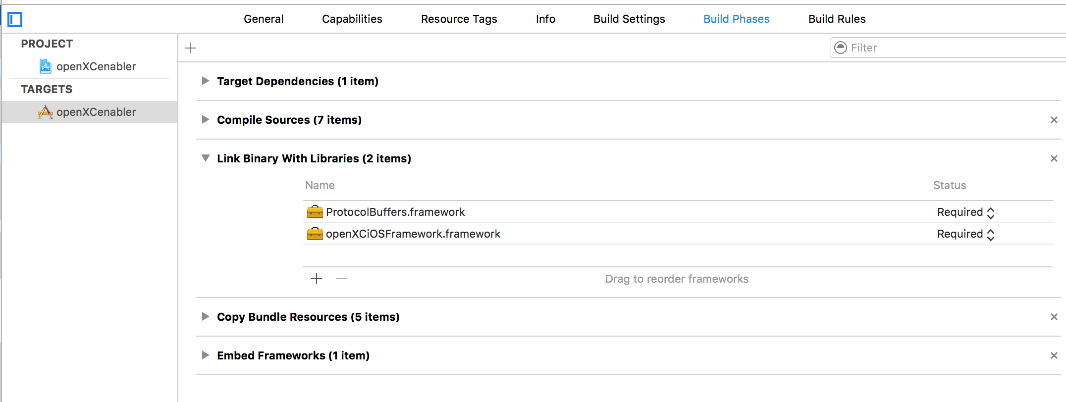


This framework can be included in any application that needs to use openXC-iOS-Framework.

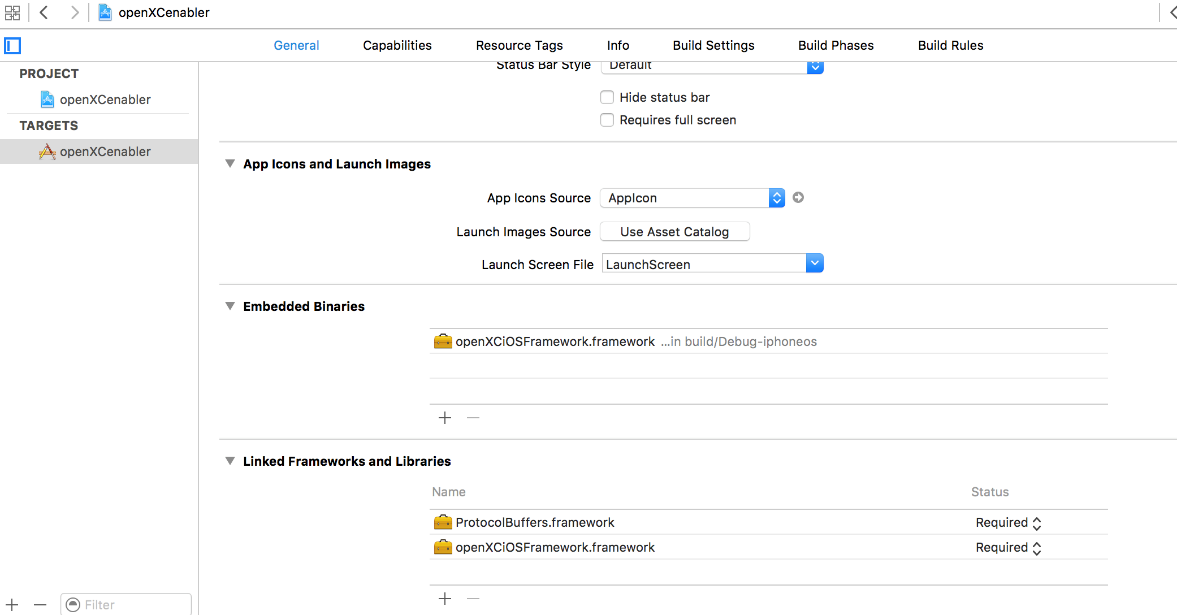
Build OpenXC iOS Demo App

# **Steps:**

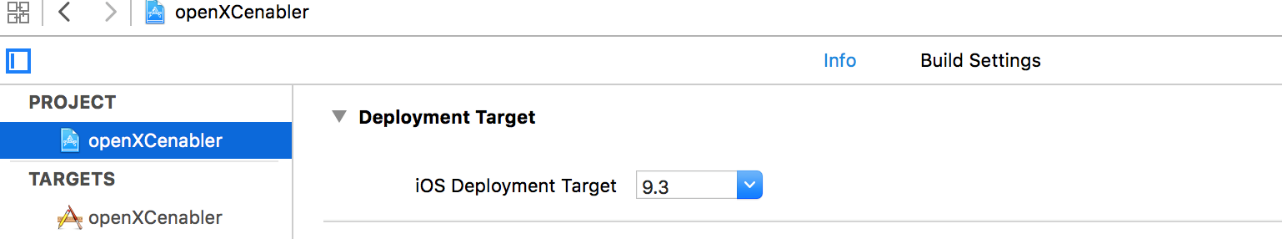
* Add the above frameworks – openXC-iOS-Framework & Protocol.Buffers Framework in “openXCenabler target - Build Phases- Link Binary With Libraries”.



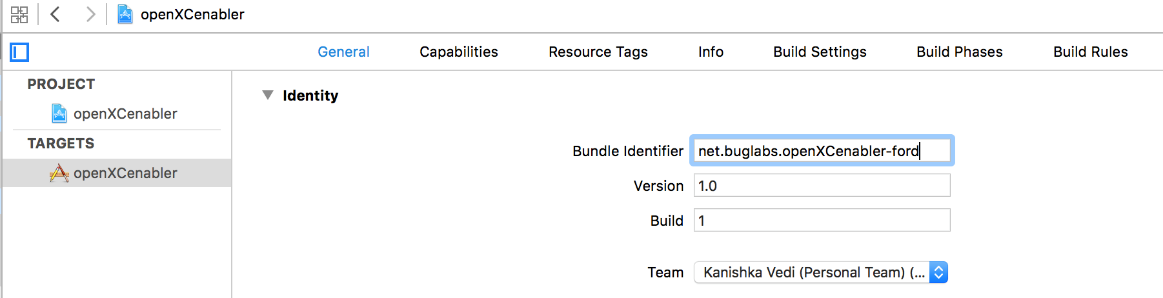
* Also add it as “Embedded Binaries” in “openXCenabler target- General”



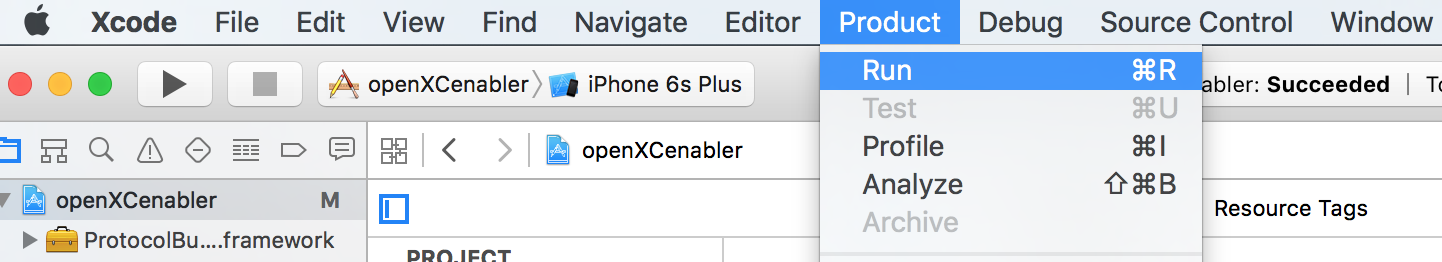
* Make sure the deployment target is 9.3



* 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.

