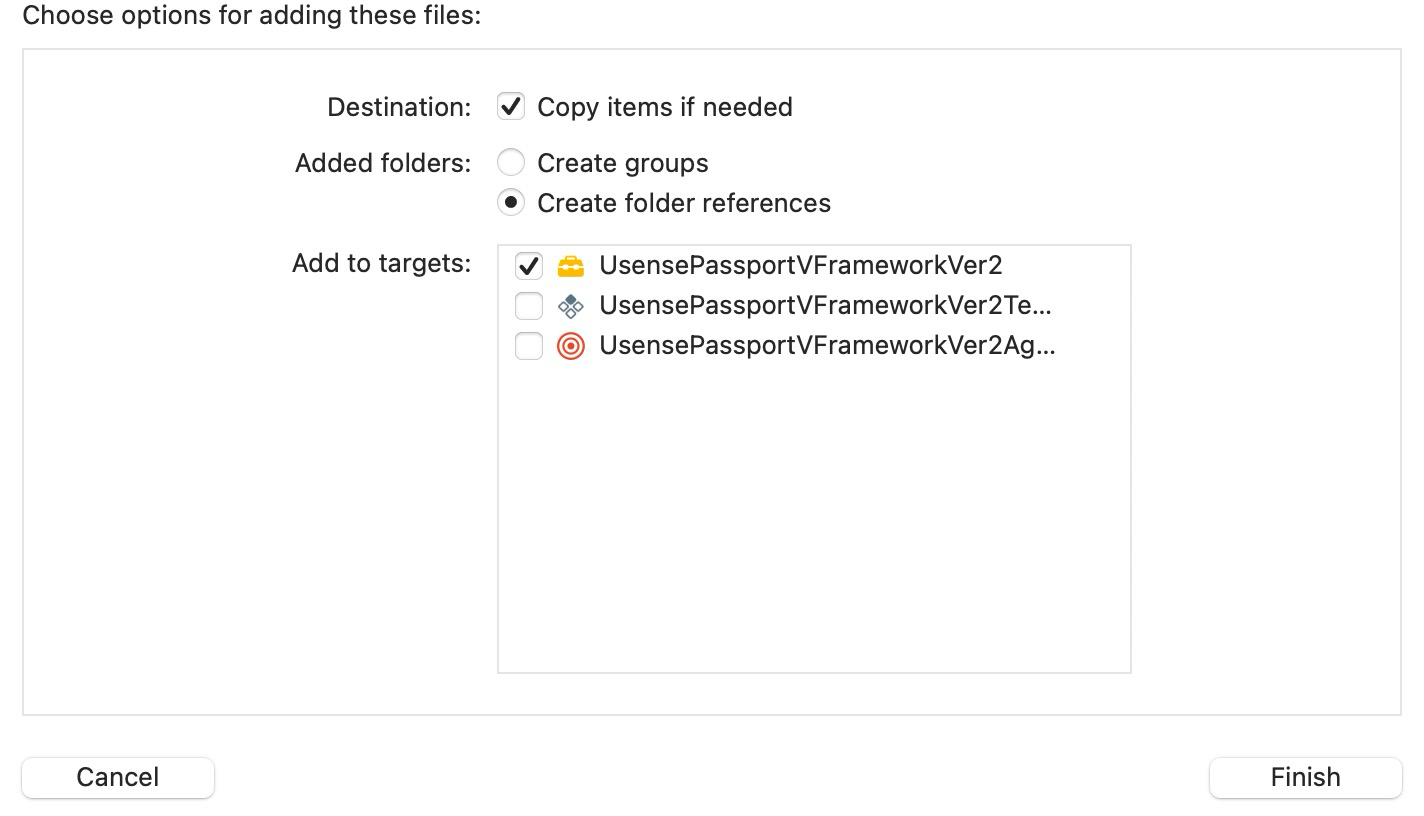
**IOS 一体式framework打包指北**

1. 创建一个framework项目。
2. 将app的项目代码直接移动到framework项目中，创建目录关联。



1. 删除桥接文件，通过framework中项目同名头文件import头文件，要用尖括号格式，且其他oc文件也按此格式导入，headers注意放到public中。

#import <UsensePassportVFrameworkVer2/newViewController.h>

1. 创建xcassets格式文件，将图片都导入其中，将storyboard文件改名同时更改info与代码中的storyboard名称，修改其中资源依赖，最后一定要把资源文件加入build phases中的copy bundle resources。
2. 通过podfile导入第三方依赖，可以在products文件夹下找到对应framework并在finder中复制放入本地项目，动态库一定要在framework and libraries中改为embed & sign，不然会提示没有该库。
3. 创建对外暴露接口，记得用public修饰相关类与方法，如果要带storyboard的ui需要指定对应storyboard然后将跳转方法封装在暴露接口中，不要从集成项目调用framework内部storyboard。

**public** **class** UsensePassportVInterface: NSObject {

**public** **static** **func** instantiateViewController() -> UIViewController {

        // 替换"MyStoryboard"为你的Storyboard名字，"MyViewController"为Storyboard中视图控制器的Storyboard ID

**let** bundle = Bundle(for: UsensePassportVInterface.**self**)

**let** storyboard = UIStoryboard(name: "Usense", bundle: bundle)

**return** storyboard.instantiateViewController(withIdentifier: "start")

    }

}

1. 把build setting中的build active architecture only修改为no，运行编译打包framework，并放入需要集成的项目中。
2. 集成项目中记得把第三方依赖都通过embed & sign加入framework目录中，修改info权限，现不允许直接用http所以App Transport Security Settings的Allow Arbitrary Loads要改成yes。
3. 运行demo，但是只能在真机上运行。