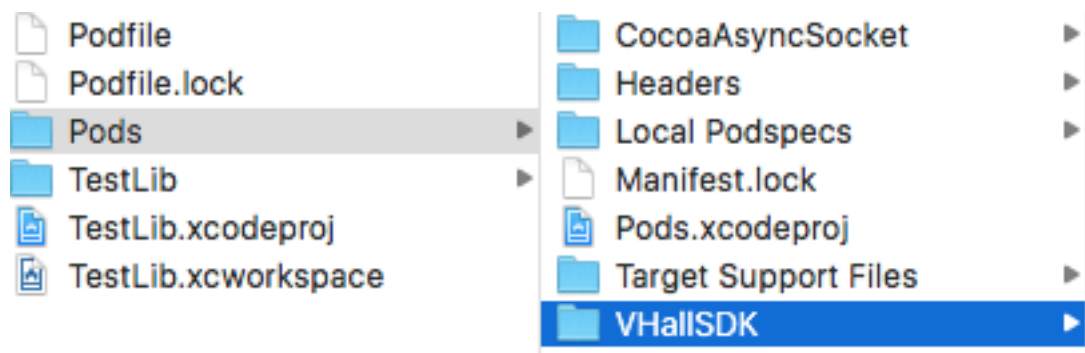


IOS 使用多个第三方库导致冲突的解决办法 : ( 适用于所有第三方库连接冲突 )

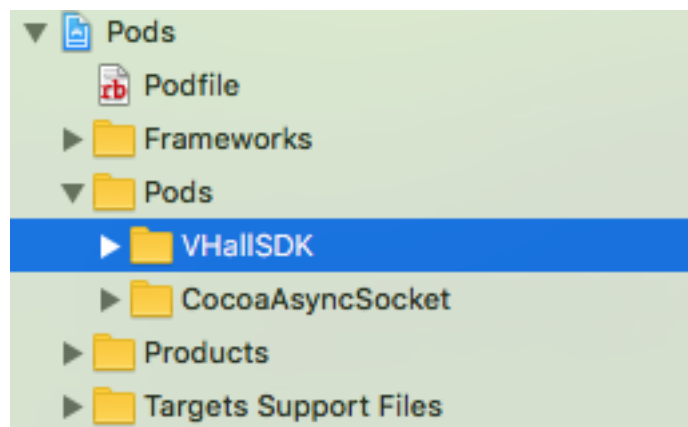
如果使用 CocoaPods ( 以下简称 Pods ) 管理第三方库导致和未使用 Pods 进行管理的第三方库冲突。

方案一 :

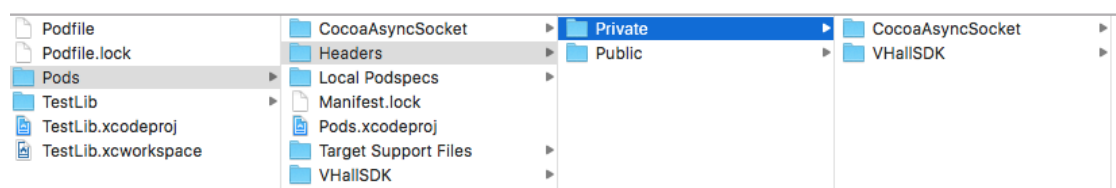
1. 将未使用 Pods 的第三方库文件复制到 Pods 项目中。

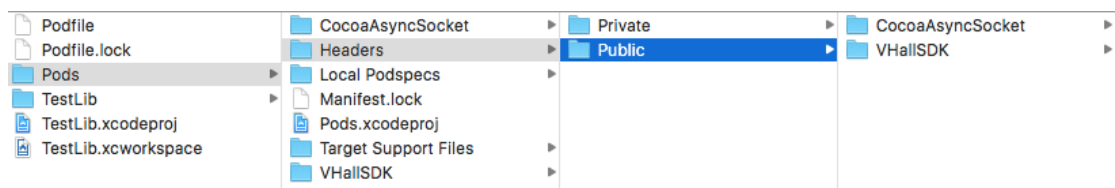


2. 删除原项目中对未使用第三方库文件的引用。
3. 将未使用 Pods 的第三方库添加到 Pods 项目中。

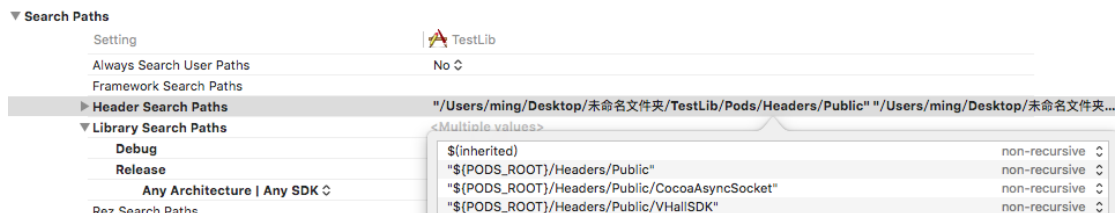


4. 将未使用 Pods 的第三方库头文件复制到 Pods 的 Headers 目录 ( 可忽略 )。





5. 根据未使用 Pods 的第三方库的头文件位置来设置头文件搜索目录。



6. 此时编译可以通过，运行时如果不成功需要设置库文件搜索目录。



方案二：

将 Pods 的 build 目录和未使用 Pods 进行管理的第三方库(即项目)的 build 设置到相同的位置。

方案三：

设置项目的 Other Linker Flags 为 `-force_load` 并依次指定每一个连接库。此时不要设置 `-ObjC` 和 `-all_load` 标记。

方案四：

不用 Pods 进行第三方库管理，删除 Pods 后正常导入所有需要的库文件。

方案五：

将有冲突的其中一个或多个库的.o 去除掉 ( 只保留一个库的.o )。参考以下网址：<http://blog.csdn.net/djl4104804/article/details/43099061>