



## 刘坤的技术博客

- [BlueBox](#)
- [所有文章](#)

嗨，我是刘坤，一名来自中国的 IOS 开发者，现就职于杭州阿里，花名‘念纪’，沉淀技术，寻求创意

[GitHub](#) [RSS](#)

### 友情链接

• [Casatwy Taloyum](#)  
• [gf&zjの盗梦空间](#)  
• [明弈](#)

## 在framework或子工程中使用xib

在工程拆分的架构中，一般会有一个主工程（上图中的MainProject），很多个子工程(上图中的subProject). 它们可能是在一个workspace中，也可能作为subProject等. 但是一般子工程的代码都会打包为静态库.a或framework来使用。

考虑一个使用场景：在subProject中使用resource怎么处理呢？这个方案大家应该很容易想到，使用bundle来管理resource. 没错，我们也是这样用的，但是如果我们要更近一步，使用xib呢。本文给出我们使用的方案，文中的顺序会按照我发现解决问题的思路来陈述。

### xcode如何处理resource

我们都知道，xib是作为资源文件来管理的，在 Build Phases 中，有一项“Copy Bundle Resources”. 在这一项中的内容会在build过程中被拷贝到product中。

- Tip: 如果你观察下Xcode的编译日志，会发现“Build Phases”中的项是按顺序执行的。

也就是说我们需要把xib添加或拖拽到这个项目下面，才会被作为资源文件打包进app. 但是子工程中的xib怎么办呢，难道也要一个一个拖到MainProject中吗，这样也太不优雅了，本来工程拆分就是要拆分结构，这样岂不是又揉到一块了么。那么我们考虑到把xib放到bundle中来处理。

那么问题来了。

### 如何让bundle中的xib能够编译呢？

xib不同于普通resource的一点是，它也需要编译，/Applications/Xcode.app/Contents/Developer/usr/bin/ibtool 工具就是用来编译xib的，被编译完的xib会生成一个叫 \*\*.nib 的文件夹，这个才是代码最终加载nib的东西。

你可能会说，这个编译好像和放在bundle里面没啥关系吧，但是经过测试证明，放在bundle或Folder中的xib在编译时，xcode是发现不了的，也就是不会被编译。我想到两个方案：\* 1.手动添加编译脚本，编译bundle中的xib \* 2.xib在工程中不放在bundle中，在编译完成后，再拷贝到bundle中去。

最终我选择了第二个方案，因为第一个方案要复杂些。。

知道这个方案了，总不可能手动copy吧，那么我们可以通过脚本完成这个事情，那么如何在导出的framework中将nib拷贝到bundle中呢，在如果你是用Run Script添加的framework。可以在编译脚本中添加(在lipo -create前后添加)：

```
1 BUNDLE_DIR=${DEVICE_DIR}/${FRAMEWORK_NAME}.bundle
2
3 if [ -d "${BUNDLE_DIR}" ];then
4     if ls ${DEVICE_DIR}/*.nib >/dev/null 2>&1;then
5         rm -rf ${BUNDLE_DIR}/*.nib
6         cp -rf ${DEVICE_DIR}/*.nib ${BUNDLE_DIR}
7     fi
8 rm -rf "${INSTALL_DIR}/${FRAMEWORK_NAME}.bundle"
9 cp -R "${BUNDLE_DIR}" "${INSTALL_DIR}/${FRAMEWORK_NAME}.bundle"
10 fi
```

其中DEVICE\_DIR是编译目录，INSTALL\_DIR是输出framework目录

这块脚本就是做了拷贝的事情，如果你不懂要加在哪，那是因为你还没有试过IOS的framework吧，推荐链接 [IOS-Universal-Framework](#).

另：Xcode6已经支持ios上的framework了，可以自选为静态动态framework. 原理差不多，上面的脚本同样使用。

### 如何让xib在DEBUG中及时生效

上面的方案已经完成了将编译过的nib拷贝到bundle中，同样代码中加载nib时也要添加bundle路径，如 @"MyFramework.bundle/MyViewController"

但是又有了新的问题，上面的目录只是更改了共享的frameworks目录下的bundle管理xib. 但是如果我们要DEBUG子工程，那么就需要把子工程拖到workspace中。那么在平时的开发DEBUG中，共享的framework目录是不会实时改动的(Release使用，上面的脚本做的事情)，那么怎么实现这样的效果：

更改了一个xib，然后DEBUG，在模拟器中可以看到刚才的更改呢。

当子工程被添加到workspace之后，在编译时也会被编译，编译完成后的build目录会和主工程共享DerivedData目录。在知道这一点后，我就思考，能不能通过在主工程添加Run Script，通过脚本在每次Build完成后把nib拷贝到product的对应的bundle中呢，而且根据前文提到的Build Phases是顺序执行的，我就可以控制该Run Scripts执行的时机。想到就开始实施，最终证明成功有效。

最后贴出MainProject中添加的Run Script(添加在Copy Bundle Resources 之后)：

```
1 RESOURCE_DIR=${CONFIGURATION_BUILD_DIR}/${UNLOCALIZED_RESOURCES_FOLDER_PATH}
2
3 mkdir -p "${RESOURCE_DIR}"
4
5 for file in `ls ${CONFIGURATION_BUILD_DIR}`;do
6     if [[ $file == *.framework ]];then
7         framework_name=`echo $file | cut -d "." -f 1`
8         framework_folder=${CONFIGURATION_BUILD_DIR}/${file}
9         if ls ${framework_folder}/${framework_name}.bundle >/dev/null 2>&1;then
10             echo "copy ${framework_name} bundle"
11             rsync -avr --copy-links --no-relative --exclude '*/.svn/*' ${framework_folder}/${framework_name}.bundle ${RESOURCE_DIR}
12             if ls ${framework_folder}/*.nib >/dev/null 2>&1;then
```

```
13             echo "copy ${framework_name} nibs to bundle"
14             rm -rf ${RESOURCE_DIR}/${framework_name}.bundle/*.nib
15             cp -rf ${framework_floder}/*.nib ${RESOURCE_DIR}/${framework_name}.bundle
16         fi
17     fi
18 fi
19 done
```

### 小结

或许我的方案不是好的，但是解决了问题，而且解决这个问题过程中让我对Xcode的编译过程有了进一步的学习和了解。还知道了神器同步命令 `rsync`，这里就不讲了。

### Comments

Copyright © 2016 刘坤 Design credit: [Shashank Mehta](#)