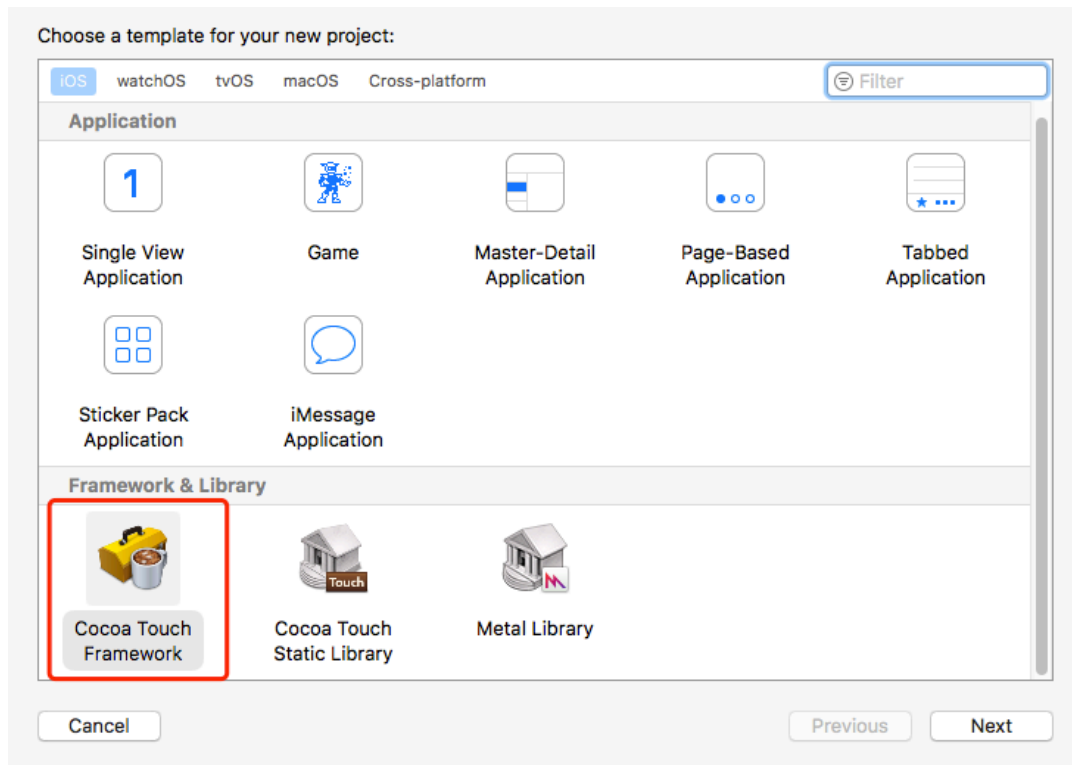
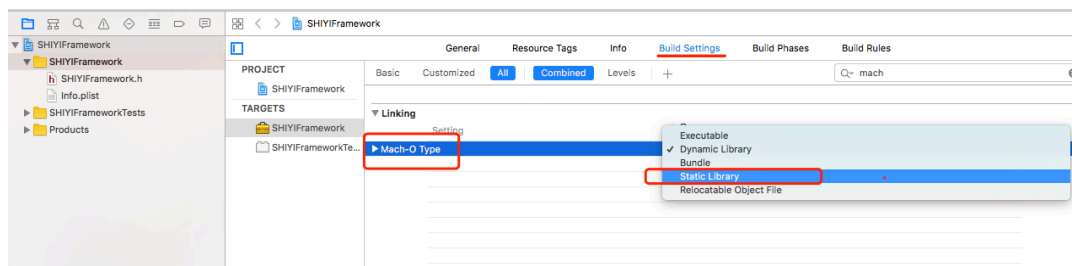


## 一：制作静态库framework

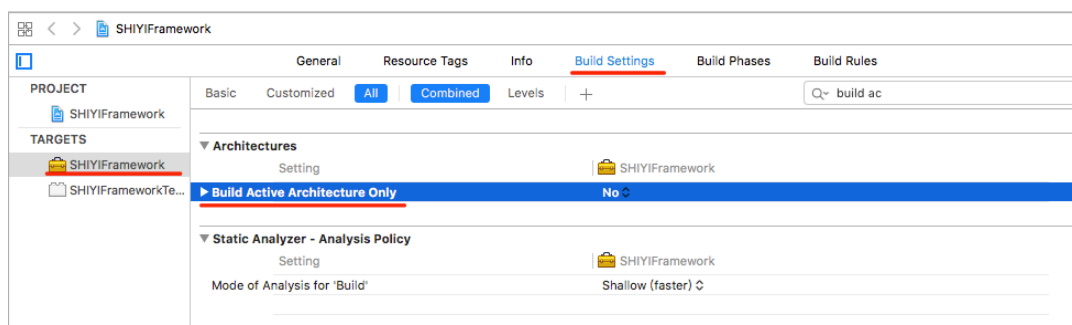
### 1, 创建framework



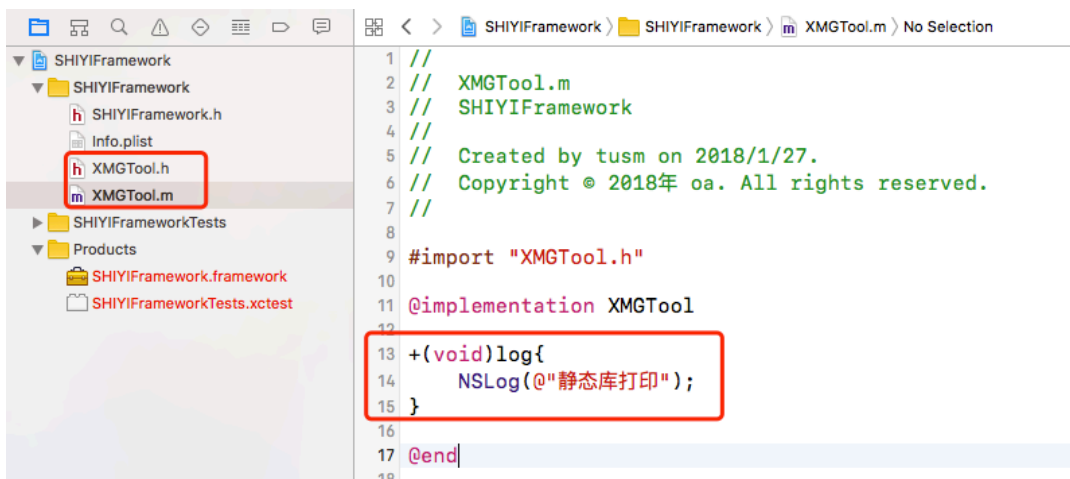
### 2, 设置mach-O Type为 静态库 Static Library



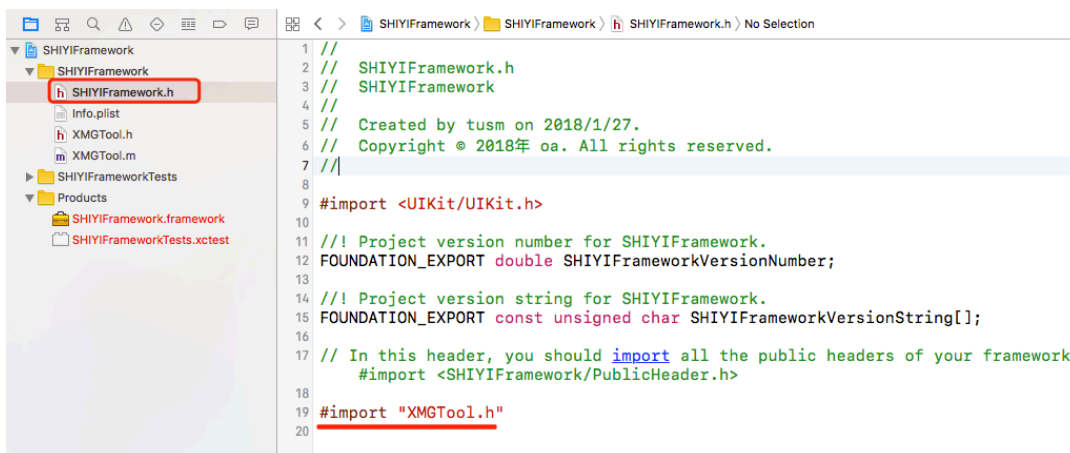
### 3, 设置编译的时候，不能只编译活跃的架构：



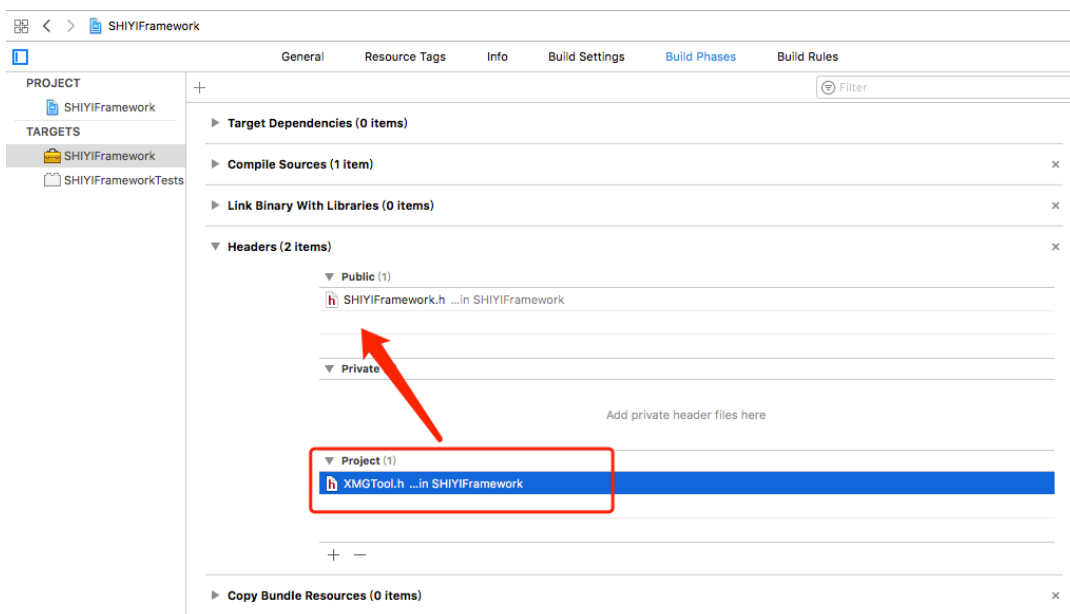
### 4, 创建XMGTool工具类，并编写类方法log



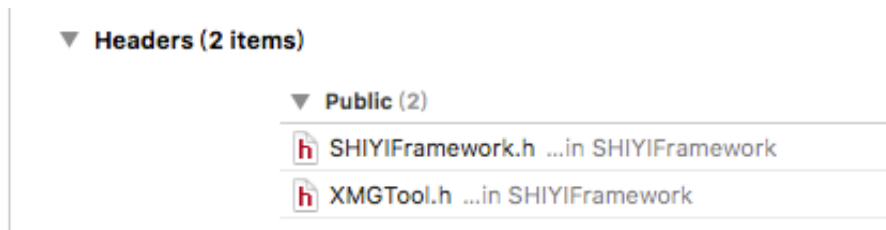
5, 在公共头文件中引入:



6, 将XMGTool.h 头文件放出去, 外边可以直接引用。

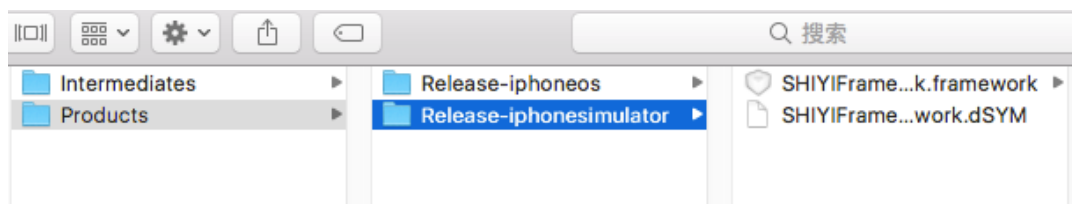


直接拖拽到Public目录中：

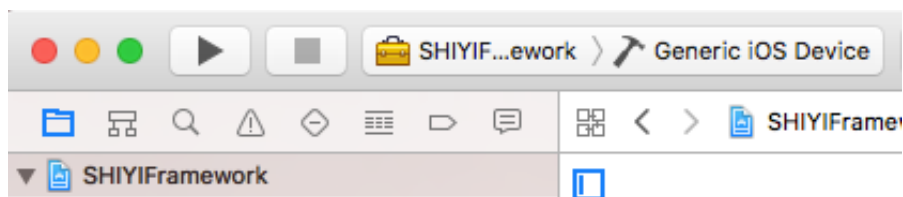


这样打成framework以后，外边就可以直接看到.h文件了。

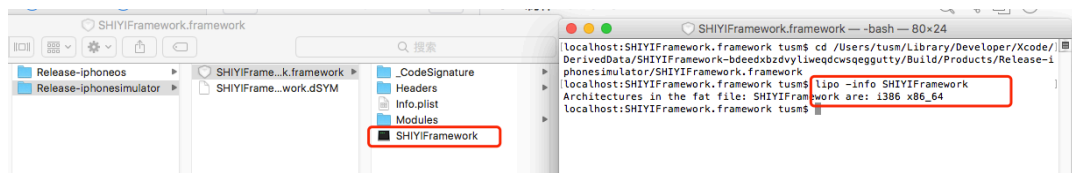
7，选择真机和模拟器编译。（设置模式为release）



选择真机编译：



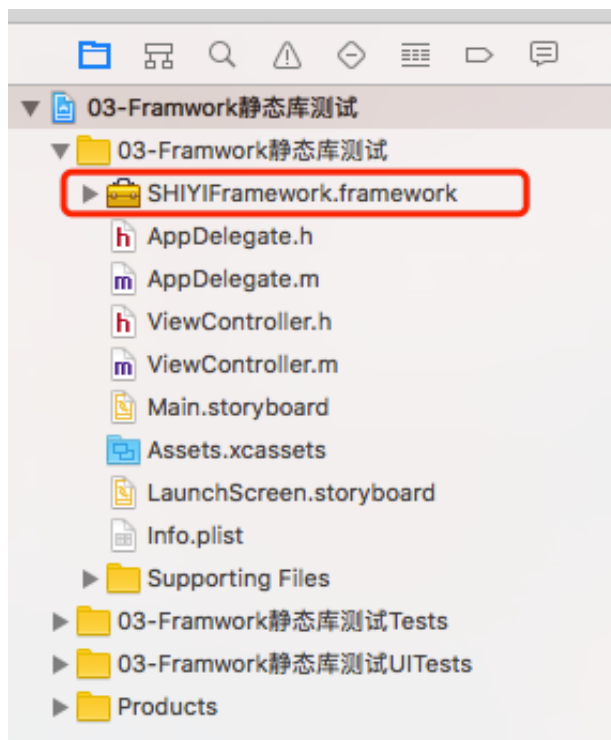
8，查看.framework支持的架构：



注意：在查看支持的架构时，直接去找framework里边的可执行文件

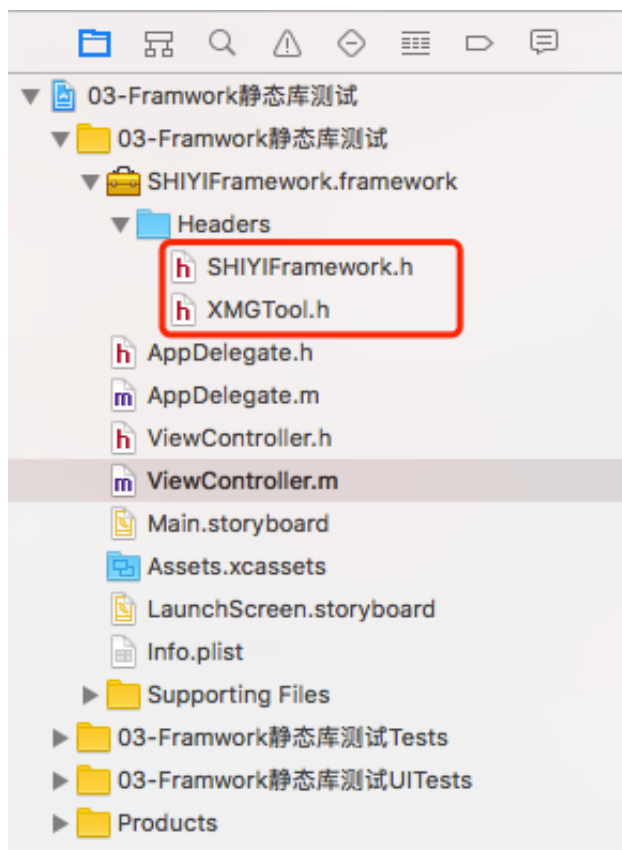
二：创建项目测试静态库

1，创建测试项目，并直接拖进去framework



直接编译，通过

2，查看静态库暴露出来的.h文件



3, 静态库的使用:

直接引入:

```
9 #import "ViewController.h"
10 #import <SHIYIFramework/SHIYIFramework.h>
11
12 @interface ViewController ()
13
14 @end
15
16 @implementation ViewController
17
18 - (void)viewDidLoad {
19     [super viewDidLoad];
20
21     [XMGTool log];
22 }
23
24
25 @end
```

直接运行, 运行成功:

