

iOS framework 制作

前言

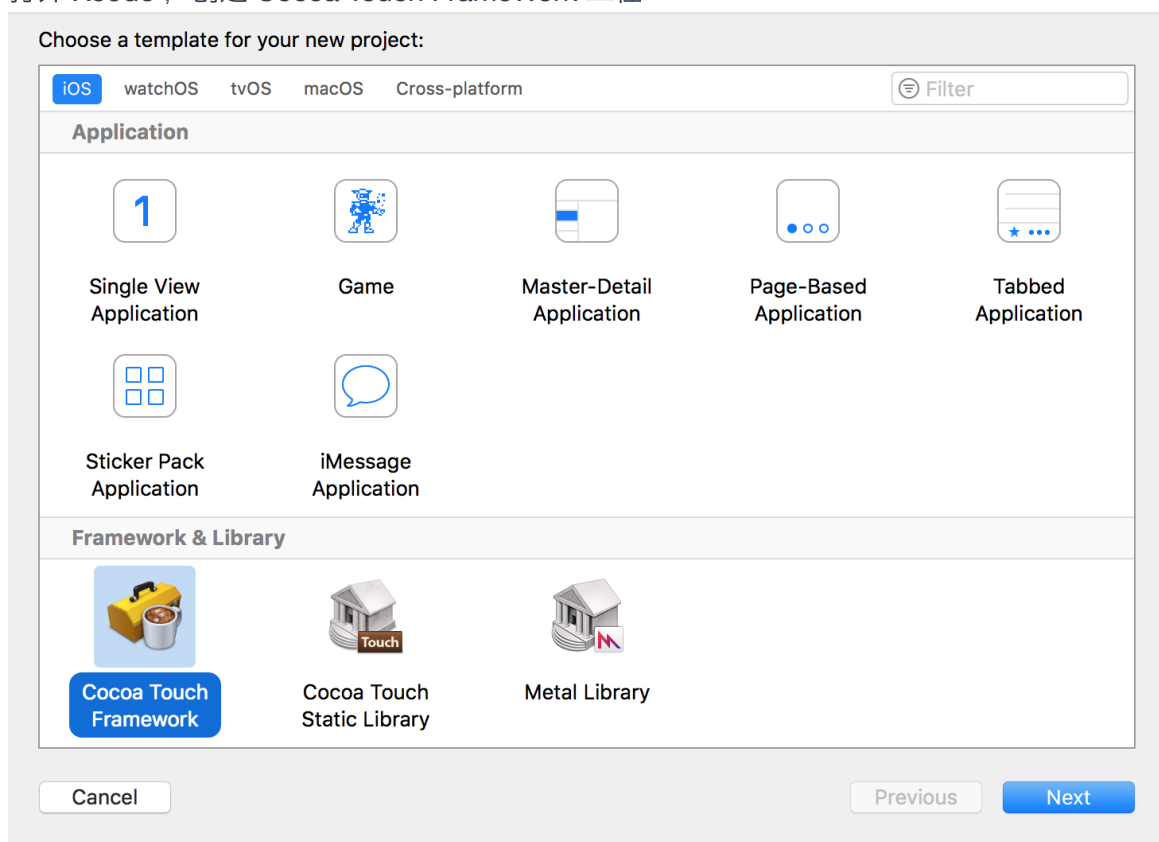
如果你想将你开发的控件与别人分享，一种方法是直接提供源代码文件。然而，这种方法并不是很优雅。它会暴露所有的实现细节，而这些实现你可能并不想开源出来。此外，开发者也可能并不想看到你的所有代码，因为他们可能仅仅希望将你的这份代码的一部分植入自己的应用中。

另一种方法是将你的代码编译成静态库（library），让其他开发者添加到自己的项目中。然而，这需要你一并公布所有的公开的头文件，实在是非常不方便。

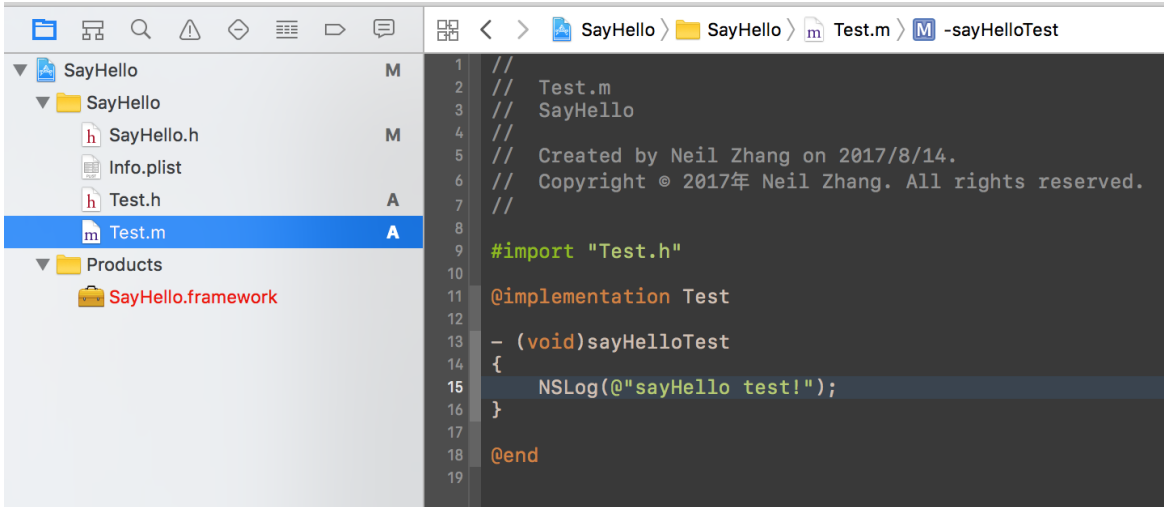
你需要一种简单的方法来编译你的代码，这种方法应该使得你的代码易分享，并且在多个工程中易复用。你需要的是一种方法来打包你的静态库，将所有的头文件放到一个单元中，这样你就可以立刻将其加入到你的项目中并使用。

步骤

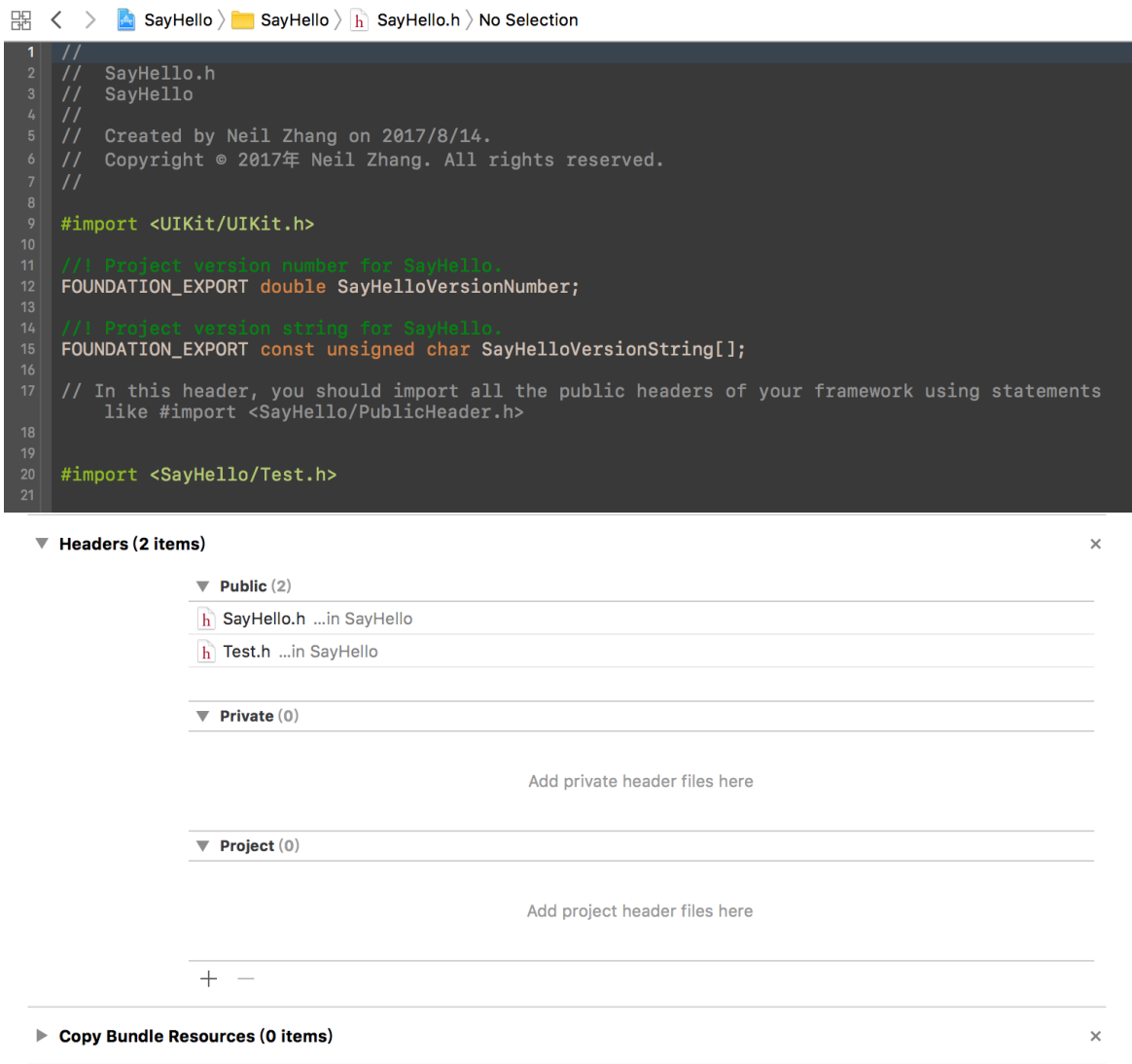
1. 打开 Xcode，创建 Cocoa Touch Framework 工程



2. 创建并实现功能类



3. 将需要公开的头文件集中到 Framework 同名的自动生成的头文件中，并设置相关的 Build Phase



4. 修改 Build Setting 相关设置

▼ Architectures	
Setting	SayHello
Additional SDKs	
Architectures	\$(ARCHS_STANDARD) armv7s - \$(ARCHS_STANDARD) armv7s ⚙
Base SDK	Latest iOS (iOS 10.3) ⚙
▼ Build Active Architecture Only	
Debug	No ⚙
Release	No ⚙
Supported Platforms	iOS ⚙
Valid Architectures	arm64 armv7 armv7s

▼ Linking	
Setting	SayHello
Bundle Loader	
Compatibility Version	1
Current Library Version	1
Dead Code Stripping	No ⚙
Display Mangled Names	No ⚙
Don't Dead-Strip Inits and Terms	No ⚙
Dynamic Library Install Name	@rpath/SayHello.framework/SayHello
Dynamic Library Install Name Base	@rpath
Exported Symbols File	
Generate Position-Dependent Executable	No ⚙
Initialization Routine	
Link With Standard Libraries	No ⚙
▶ Mach-O Type	Static Library ⚙
Order File	
Other Librarian Flags	
Other Linker Flags	

▼ Deployment	
Setting	SayHello
Additional Strip Flags	
Deployment Postprocessing	No ⚙
Strip Debug Symbols During Copy	No ⚙
Strip Linked Product	Yes ⚙
▶ Strip Style	Non-Global Symbols ⚙
Use Separate Strip	No ⚙

5. commend + B

如果需要支持模拟器和手机，则需要在连接手机时和模拟器下分别编译。

打包 Framework

打包 Framework 有两种方式：

1. 手动编译打包合并

- 测试编译出的库所支持的架构

- 右键点击 Product 目录下的 XXX.framework Show in Finder
- 进入到 XXX.framework 下，执行 `lipo -info` 命令，可以看出其支持的 CPU 架构

```
neilZhang:SayHello.framework neilzhang$ cd /Users/neilzhang/Library/Developer/Xcode/DerivedData/SayHello-eniqnaarugcswlgmzjflfvqmsre/Build/Products/Debug-iphonesimulator/SayHello.framework
neilZhang:SayHello.framework neilzhang$ ls
Headers      Info.plist  Modules    SayHello
neilZhang:SayHello.framework neilzhang$ lipo -info SayHello
Architectures in the fat file: SayHello are: armv7 arm64 armv7s
neilZhang:SayHello.framework neilzhang$ cd ../../Debug-iphonesimulator/SayHello.framework/
neilZhang:SayHello.framework neilzhang$ ls
Headers      Info.plist  Modules    SayHello      _CodeSignature
neilZhang:SayHello.framework neilzhang$ lipo -info SayHello
Architectures in the fat file: SayHello are: i386 x86_64
neilZhang:SayHello.framework neilzhang$
```

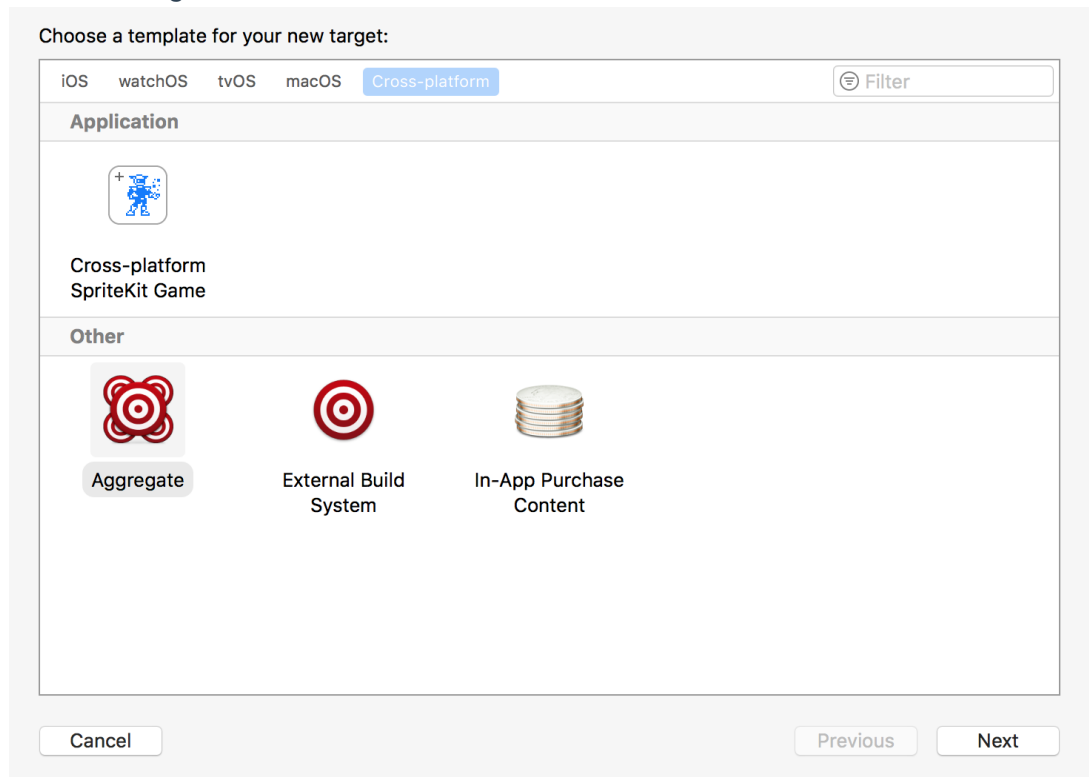
- 如果需要该 Framework 同时支持模拟器和手机，则需要将编译出的两个 Framework 合并，可以使用 `lipo -create xxx xxx -output xxx` 命令

```
neilZhang:SayHello.framework neilzhang$ lipo -create SayHello ../../Debug-iphoneos/SayHello.framework/SayHello -output ~/Desktop/SayHello
neilZhang:SayHello.framework neilzhang$ lipo -info ~/Desktop/SayHello
Architectures in the fat file: /Users/neilzhang/Desktop/SayHello are: i386 armv7
armv7s x86_64 arm64
neilZhang:SayHello.framework neilzhang$
```

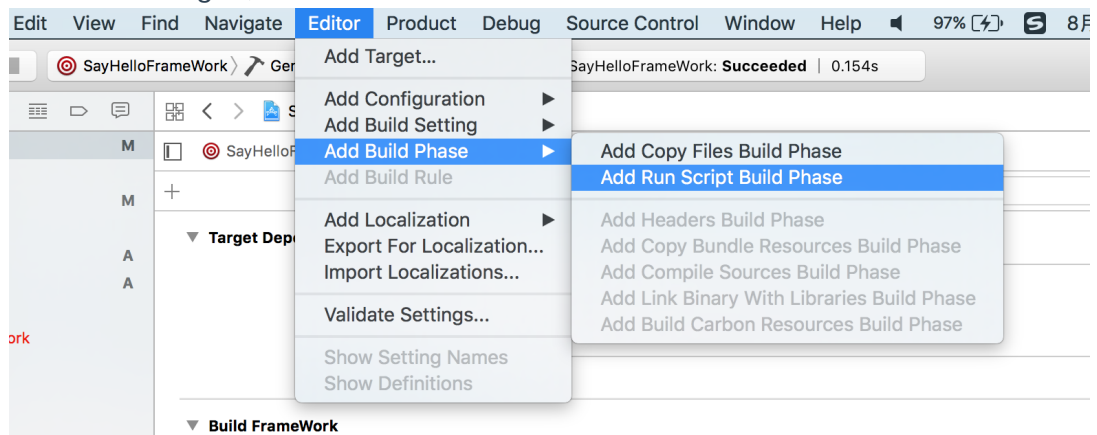
- 将合并后的库文件替换掉原来只支持单一架构的库文件
- 大功告成

2. 创建新的 Target 打包

- 创建新的 Target



- 选择新建的 Target，添加脚本



添加脚本

Build Framework

Shell /bin/sh

20 cp -R "\${DEVICE_DIR}/" "\${INSTALL_DIR}/"

21 #ditto "\${DEVICE_DIR}/Headers" "\${INSTALL_DIR}/Headers"

22

23 lipo -create "\${DEVICE_DIR}/\${FMK_NAME}" "\${SIMULATOR_DIR}/\${FMK_NAME}" -

24 output "\${INSTALL_DIR}/\${FMK_NAME}"

25 #这个是合并完成后打开对应的文件夹，你就可以直接看到文件了

☒ Show environment variables in build log

☐ Run script only when installing

Input Files

Add input files here

- command + B, 保存生成的 Framework

Framework 的使用

- 1. 直接将 Framework 添加到工程中
- 2. 添加 Target Build Phase 选项

Link Binary With Libraries (2 items)

Name	Status
libSayHelloA.a	Required
SayHello.framework	Required

Drag to reorder frameworks

Framework Search Paths

Header Search Paths

Library Search Paths

Rez Search Paths

Sub-Directories to Exclude in Recursive Searches

Sub-Directories to Include in Recursive Searches

System Framework Search Paths

/Users/neilzhang/Desktop/Basic/TestForFramework/**

/Users/neilzhang/Desktop/Basic/TestForFramework/..../SavHelloA/**

\$(PROJECT_DIR)

\$(PROJECT_DIR)/TestForFramework

Text Settings

non-recursive

recursive

non-recursive

- 3. 导入公开的头文件

```
//
#import "AppDelegate.h"
#import <SayHello/SayHello.h>

@interface AppDelegate ()
@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    Test *test = [Test new];
    [test sayHelloTest];

    return YES;
}
```