

fengsh998的专栏

blog因你而精彩。

☰ 目录视图

☰ 摘要视图

RSS 订阅

个人资料



边缘998



访问：1064451次

积分：9348

等级：

BLOG > 6

排名：第1358名

原创：48篇

转载：50篇

译文：5篇

评论：186条

联系方式

QQ:19985430(满)/3031702

fengsh998@163.com

文章搜索

文章分类

- lphone (65)
- mac os (4)
- 网络 (0)
- 网络设置 (1)
- DOS (4)
- Delphi (7)
- Delphi 线程 (3)
- android 资料 (4)
- Object-C (4)

【CSDN技术主题月】深度学习框架的重构与思考

【观点】有了深度学习，你还学传统机器学习算法么？

【知识库】深度学

习知识图谱上线啦

初探swift语言的学习笔记九(OC与Swift混编)

标签：

object-c swift 混编 OC

2014-06-25 22:26

48272人阅读

评论(5)

收藏

举报

☰ 分类：

swift (12) ▼

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

作者：fengsh998

原文地址：<http://blog.csdn.net/fengsh998/article/details/34440159>

转载请注明出处

如果觉得文章对你有所帮助，请通过留言或关注微信公众帐号fengsh998来支持我，谢谢！

swift 语言出来后，可能新的项目直接使用swift来开发，但可能在过程中会遇到一些情况，某些已用OC写好的类或封装好的模块，不想再在swift 中再写一次，哪就使用混编。这个在 I O S 8中是允许的。

先中简单的入手，先研究在同一个工程目录下混合使用的情况。

为了演示。先准备两个类

第一个是swift语言写的类,文件名为 act.swift

[cpp] C

```
01. import Foundation
02.
03. class Act : NSObject
04. {
05.     func hasAct(tag:Int) -> String
06.     {
07.         switch (tag)
08.         {
09.             case 1:return "Movie"
10.             case 2:return "CCTV"
11.             case 3:return "Sport TV"
12.             default:return "Area TV"
13.         }
14.     }
15.
16.     init()
17.     {
18.         println("act constructor is called.")
19.     }
20.
```

设置 (5)
角色 (5)
设计模式 (5)
安全 (6)
UML (2)
C++ (21)
gcc/g++ (11)
架构 (5)
编码 (7)
WCF(C#) (3)
Linux (2)
医学 (1)
swift (13)

文章存档

2016年08月 (1)
2015年05月 (2)
2014年07月 (1)
2014年06月 (13)
2014年05月 (1)

展开

阅读排行

IOS SEL (@selector) 原 (67477)
IOS CoreText.framework (48454)
初探swift语言的学习笔记 (48249)
写给喜欢用Block的朋友((45547)
IOS 世界标准时间UTC / (43450)
ios设备抓包方法 (36396)
IOS CoreText.framework (30505)
WINHTTP的API接口说明 (23894)
IOS Emoji 编码打印 (22045)
ios 防止按钮快速点击 (21884)

评论排行

基于智能手机的加速计进 (21)
IOS CoreText.framework (18)
写给喜欢用Block的朋友((16)
IOS CoreText.framework (13)
详释(常见UITextView 输 (10)
IOS SEL (@selector) 原 (10)
IOS CoreText.framework (9)
自荐亲自写的一套http访 (7)
UITextView的动态适应高 (7)
初探swift语言的学习笔记 (6)

推荐文章

* 2016 年最受欢迎的编程语言是什么?
* Chromium扩展 (Extension) 的页面 (Page) 加载过程分析
* Android Studio 2.2 来啦
* 手把手教你做音乐播放器

```
21.     deinit
22.     {
23.         println("act destroyed is called.")
24.     }
25. }
```

第二个是用OC写的类 头文件为OCChannel.h,实现文件为OCChannel.m
头文件

```
[cpp]
01. #import <Foundation/Foundation.h>
02.
03. @interface OCChannel : NSObject
04.
05. @property (nonatomic,retain) NSString *ChannelName;
06.
07. - (NSString *)ChannelChange:(NSInteger) channels;
08.
09. @end
```

实现文件

```
[cpp]
01. #import "OCChannel.h"
02. #import "SwiftModule-swift.h"
03.
04. @interface OCChannel()
05. {
06.     Act *act; //swift的类
07. }
08. @end
09.
10. @implementation OCChannel
11.
12. - (id)init
13. {
14.     if (self = [super init]) {
15.         NSLog(@"OC Constructor is called.");
16.         //使用Swift类
17.         act = [[Act alloc] init];
18.     }
19.     return self;
20. }
21.
22. - (void)dealloc
23. {
24.     NSLog(@"OC Destroyed is called.");
25.     //act release;//ARC not use
26.     //super dealloc;//ARC not use
27. }
28.
29. - (NSString *)ChannelChange:(NSInteger) channels
30. {
31.     return [act hasAct:channels];
32. }
33.
34. @end
```

这个OCChannel为中引用了swift 写的类Act 。主要是为了演示在同一个工程项目里, swift类调用OC,同时OC类也调用Swift。从而形成一种混合编写的模式。

下面是具体步骤:

(二) 技术原理与框架设计

* JVM 性能调优实战之：使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码

最新评论

基于智能手机的加速计进行计步
kezhende: 楼主可以把demo代码发下看看嘛，最近也在做计步，谢谢~邮箱：229888209@qq.com

关于cocoa pod 管理代码时，第一小小不懂道：Specs怎么改名为master，Specs路径怎么找到

关于cocoa pod 管理代码时，第一边缘998: 如果没有说明你把git的隐藏文件删了

关于cocoa pod 管理代码时，第一边缘998: @qq_19293569:git clone https://github.com/CocoaPod...

关于cocoa pod 管理代码时，第一qq_19293569: 输入 pod repo 出现下面的显示master- Type: local- URL: - Pa...

IOS CoreText.framework --- 行 C qq_18939335: 你好，请问怎么在最后一行添加省略号呢

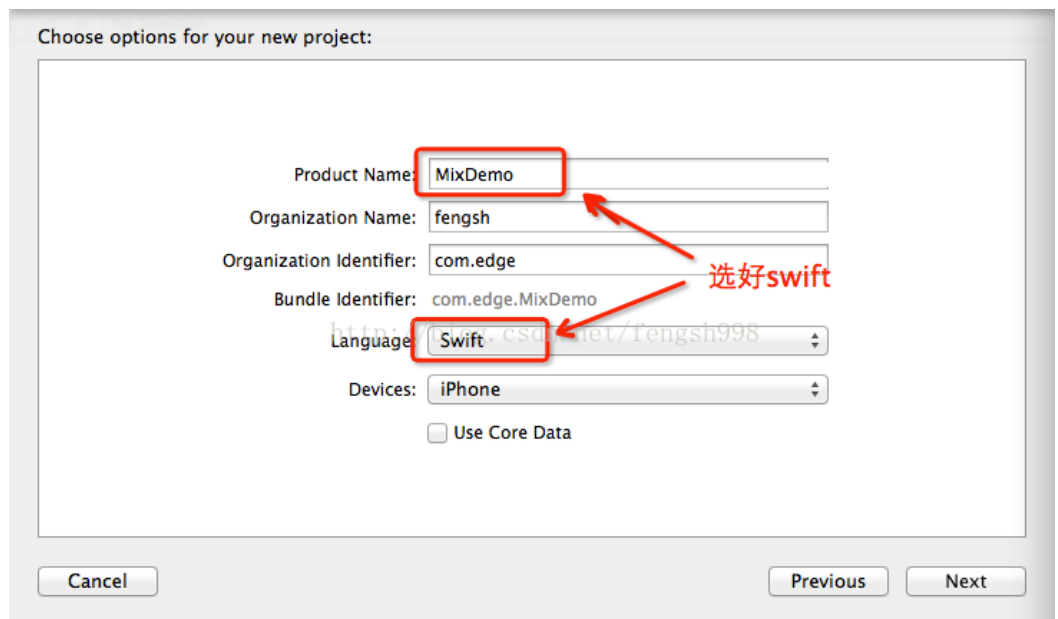
IOS SEL (@selector) 原理及使用 haxwn: s.methodTest = @selector(SubMethod);//这句就在运行时期，会先寻...

UITextView的动态适应高度处理 wangbaoan1986511: 在吗

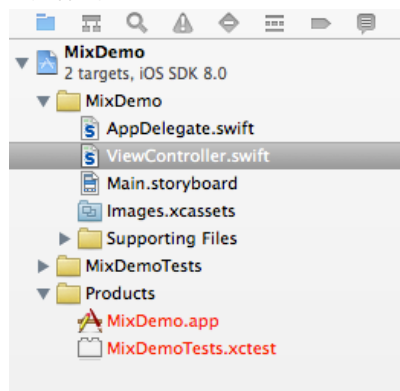
写给喜欢用Block的朋友(ios Bloc JobsHero: 有钻研精神，思考后的劳动成果。

基于智能手机的加速计进行计步 baidu_34203140: 求来份demo, 1007352884@qq.com

1.新建一个Swift工程：我这里工程名为MixDemo



建好后工程：



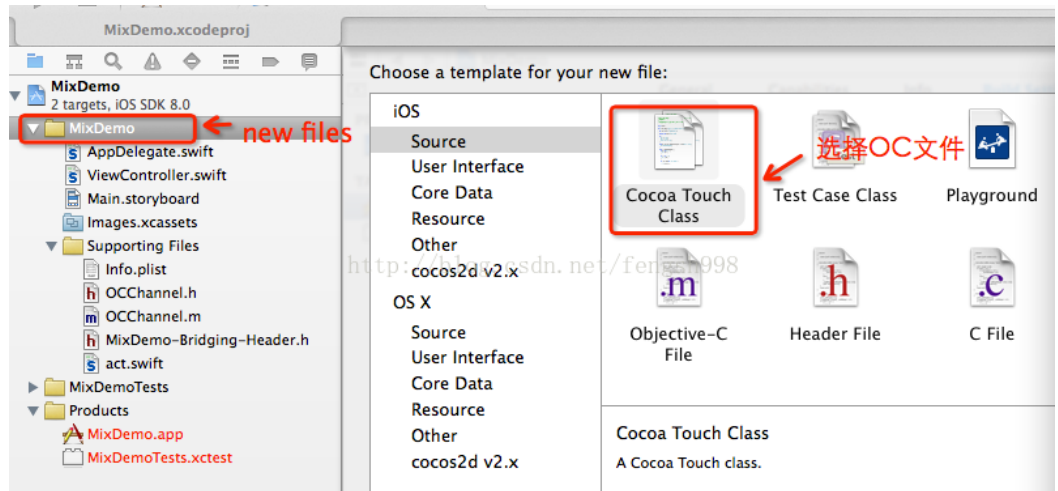
2.就是分别引入前面的两个类，咱先一个来。因为建的是Swift，所以，咱先以Swift工程中引用OC文件进行一次混编

Swift中使用OC

首先Swift中不再使用头文件和.m文件的方式了。所以也不需要import ""来导入头文件。哪swift 如何能访问到O C的类声明的。

其实，swift也是需要使用头文件进行访问的，只不过不再需要使用显式的方式使用import进行导入。有两种方式来实现这个头文件的生成。

方式一：在一个全新的Swift，利用第一次新建提示的方式自动添加桥接头文件。



se options for your new file:

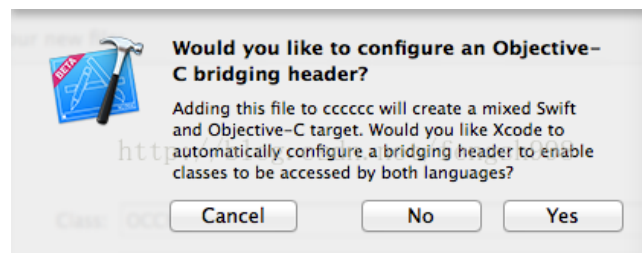
Class: **OCChannel**

Subclass of: NSObject

Also create XIB file

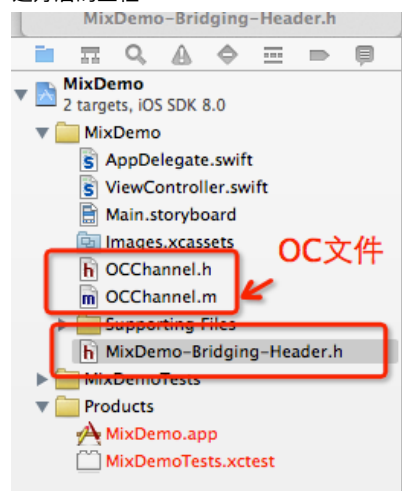
iPhone

Language: **Objective-C**

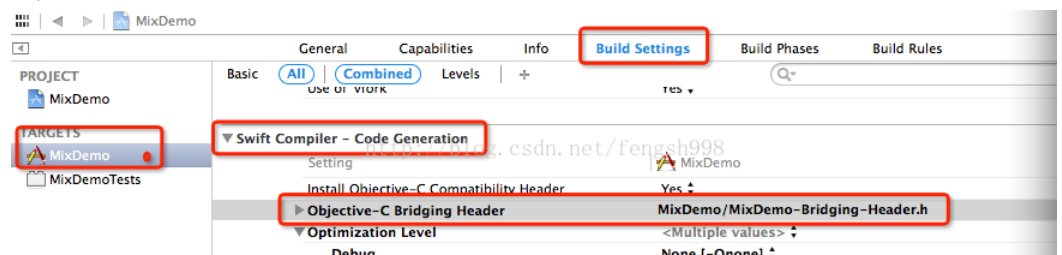


点确定这后就会生成一个以<produceName-Bridging-Header.h>的头文件。

建好后的工程:



这里有一个地方需要注意的就是在targets->build settings->Object-C Bridging Header 设为哪个桥接的头文件即可。



经过上述步骤，桥接文件弄好了就可以

尽情的把想要在swift类中调用的OC头文件放使用import "" 写到这个桥接文件中吧。就像：

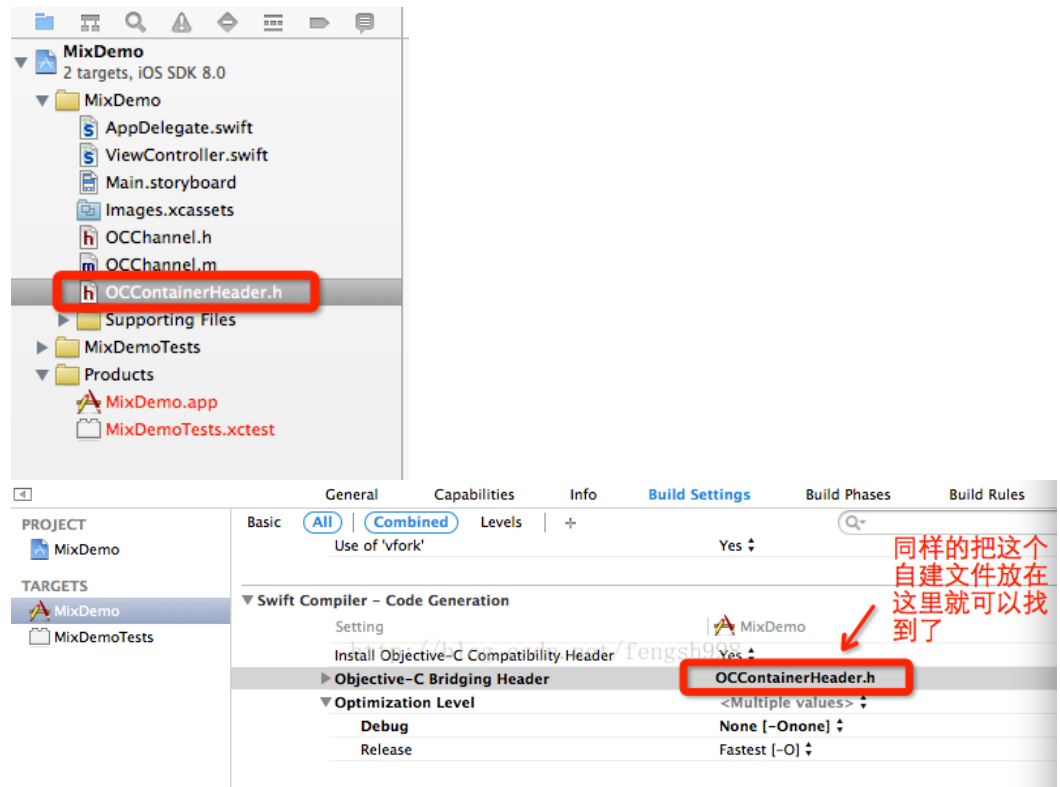
```
[cpp]
01. //
02. // Use this file to import your target's public headers that you would like to expose to Swift
```

```
03. //MixDemo/MixDemo-Bridging-Header.h
04.
05. #import "OCChannel.h"
```

同样的，当你知道这个swift搜索头文件的关系后，就不需要再理会这个-Bridging-Header.h的文件了。完全可以手工建一个并取自己喜欢的名字。如：

方式二：

新建一个头文件，名为:OCContainerHeader.h



好了，以上的设置后就完全满足了Swift使用OC写的类了。

```
[cpp]
01. import UIKit
02.
03. class ViewController: UIViewController {
04.
05.     override func viewDidLoad() {
06.         super.viewDidLoad()
07.         // Do any additional setup after loading the view, typically from a nib.
08.
09.         //调用OC类
10.         var channel = OCChannel()
11.         println(channel.ChannelChange(10))
12.         println(channel.ChannelChange(2))
13.     }
14.
15.     override func didReceiveMemoryWarning() {
16.         super.didReceiveMemoryWarning()
17.         // Dispose of any resources that can be recreated.
18.     }
19.
20.
21. }
```

好了下面再来看一下OC如何调用Swift写的类。（事实上，如果你是一比一抄我这个DEMO，那么恭喜你，在



以上你将编译不通过。因为我的OC类中引用了swift写的类，所以你要想运行，就必须把哪个Act的类注释了才行。)

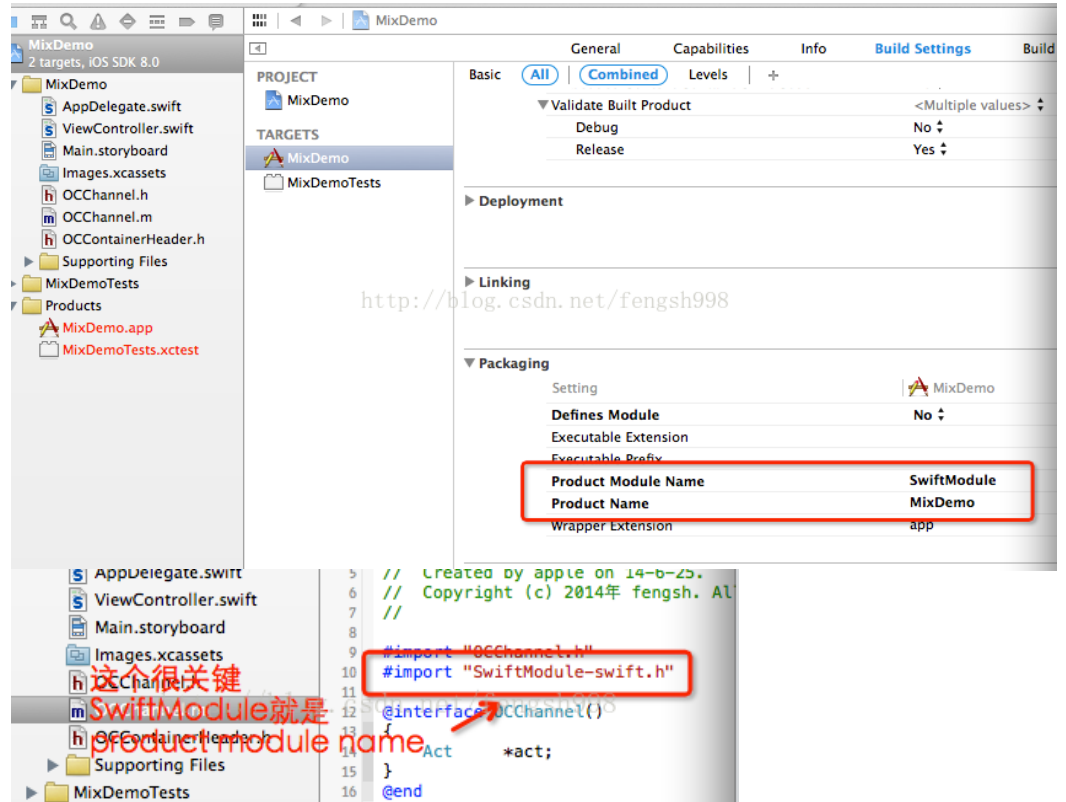
OC如何调用Swift写的类

OC要想使用，必须有头文件。而swift文件却没有头文件，所以咱们想必也需要产生一个头文件。但对于OC调用swift的头文件比较特殊。因头文件里面的机制是自动生成的，在不熟悉的，不建议手写。

哪如何产生这个头文件。（注意，系统设置的头文件，在工程中是看不到的。）

产生步骤：

选中targets->build settings ->packing->Product Module Name 中设置模块名，这个名称很重要 swift的头文件就是根据这个来命名的。



虽然你看图中有这个import "SwiftModule-swift.h"但你在整个工程中是找不到这个文件的，但可以使用CMD+ 鼠标点击可看这个头文件中的内容。



这样，工程中如查Swift要使用OC,则把需要使用的OC类的头文件，全写在MixDemo-Bridging-Header.h里。同样如果OC中所使用的swift类，只需要Clean一把，再编就可以了，但不要忘了导入SwiftModule-swift.h哦（名称自取，但-swift.h是固定的），另外还有一个需要读者注意的。

注：

凡是用Swift写的类，如果不继承自NSObject或NSObject的派生类，那么编译后将不会生成对应的转换类。从而使得OC中找不到相应的声明。

如我的例子中 class Act 这样不会被编译到SwiftModule-swift.h中，但写为 class Act : NSObject，就可以编译出相应的声明。另外可以使用@objc加以声明，但这个还是一样，类最好继承NSObject下来。就像下面：

```

[cpp]
01. import Foundation
02.
03. @objc(Act)
04.
05. class Act
06. {
07.     func hasAct(tag:Int) -> String
08.     {
09.         switch (tag)
10.         {
11.             case 1:return "Movie"
12.             case 2:return "CCTV"
13.             case 3:return "Sport TV"
14.             default:return "Area TV"
15.         }
16.     }
17.
18. @objc(init)//原本以为加上这个alloc就可以找到，但不行的。。。
19. init()
20. {
21.     println("act constructor is called.")
22. }
23.
24. deinit
25. {
26.     println("act destroyed is called.")
  
```



```
27.     }  
28. }
```

但是在使用时你就会发现

```
act = [[Act alloc]init]; //报错, 找不到alloc, 因此建议大家还是继承NSObject.
```

好吧, 今天就写到这里, 再研究一下framework之间的混编。。。。。

顶 8
踩 1

上一篇 初探swift语言的学习笔记八(保留了许多O C的实现)

下一篇 Swift实战

我的同类文章

swift (12)			
• 初探swift语言的学习笔记...	2014-06-29	• 初探swift语言的学习笔记...	2014-06-29
阅读 10846		阅读 15604	
• Swift实战	2014-06-25 阅读 4153	• 初探swift语言的学习笔记...	2014-06-20 阅读 4584
• 初探swift语言的学习笔记...	2014-06-18 阅读 8564	• 初探swift语言的学习笔记...	2014-06-18 阅读 7363
• 初探swift语言的学习笔记...	2014-06-12	• 初探swift语言的学习笔记...	2014-06-12 阅读 1357
阅读 10623		• 初探swift语言的学习笔记...	2014-06-08 阅读 6204
• 初探swift语言的学习笔记...	2014-06-09 阅读 4747		

猜你在找

使用Swift语言开发iOS 9应用程序实战

初探swift语言的学习笔记八保留了许多O C的实现

iOS8开发视频教程Swift语言版-Part 9:iOS分层架构

初探swift语言的学习笔记八保留了许多O C的实现

iOS8开发视频教程Swift语言版-Part 1:第一个iOS应

详解swift和OC以及C语言的混编不看后悔

iOS开发Swift语言学习教程

初探swift语言的学习笔记二可选类型和隐式可选类

Swift语言进阶

初探swift语言的学习笔记十block



胡歌那么忙, 怎么学英语

[上海免费] 订阅每日英语, 随时随地轻松学

查看评论

5楼 From now on 2015-12-09 15:22发表



很好, 谢谢博主。

4楼 刘小壮 2015-06-28 16:29发表



非常感谢博主分享这么好的一篇博客，正好用到了。

3楼 xianzufu 2015-03-03 16:22发表



感谢

2楼 机智的白菜 2014-10-09 14:55发表



引用"xiaoquan_2001"的评论：

framework之间的混编，楼主有研究吗。按照上述的方法swift调用OC是没有问题，但是...

大意了... 编译抛异常

using bridging headers with framework targets is unsupported.

好吧。

1楼 机智的白菜 2014-10-09 14:36发表



framework之间的混编，楼主有研究吗。按照上述的方法swift调用OC是没有问题，但是反过来OC中调用swift时候在：

```
#import "$(product module name)-swift.h"
```

提示不存在...

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved

