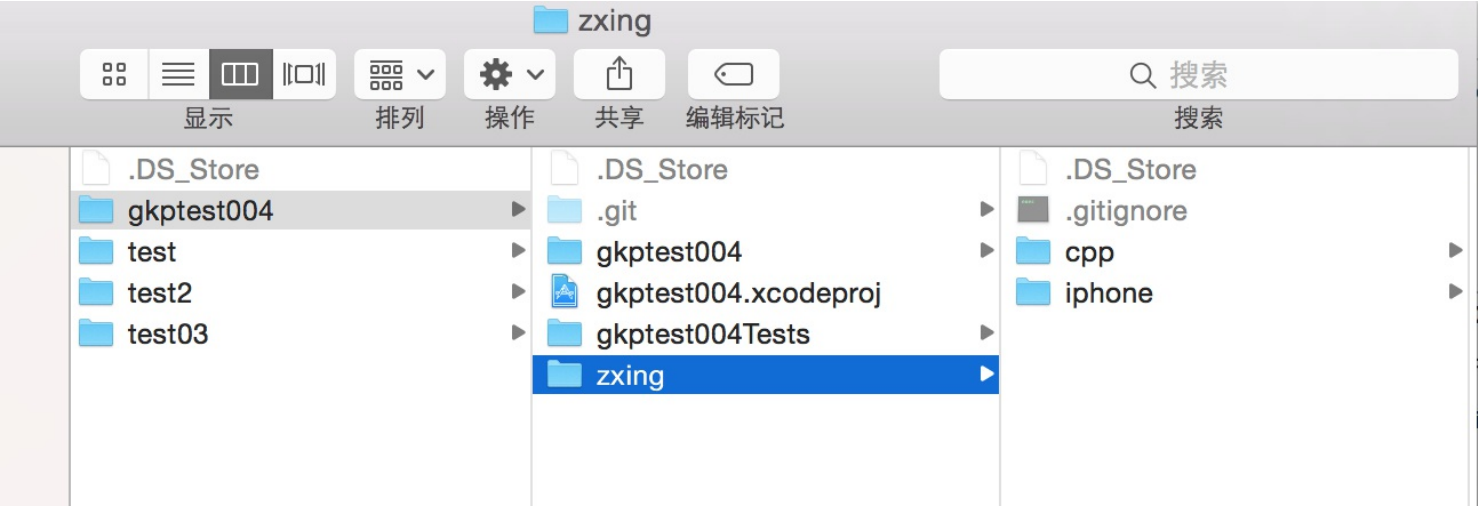
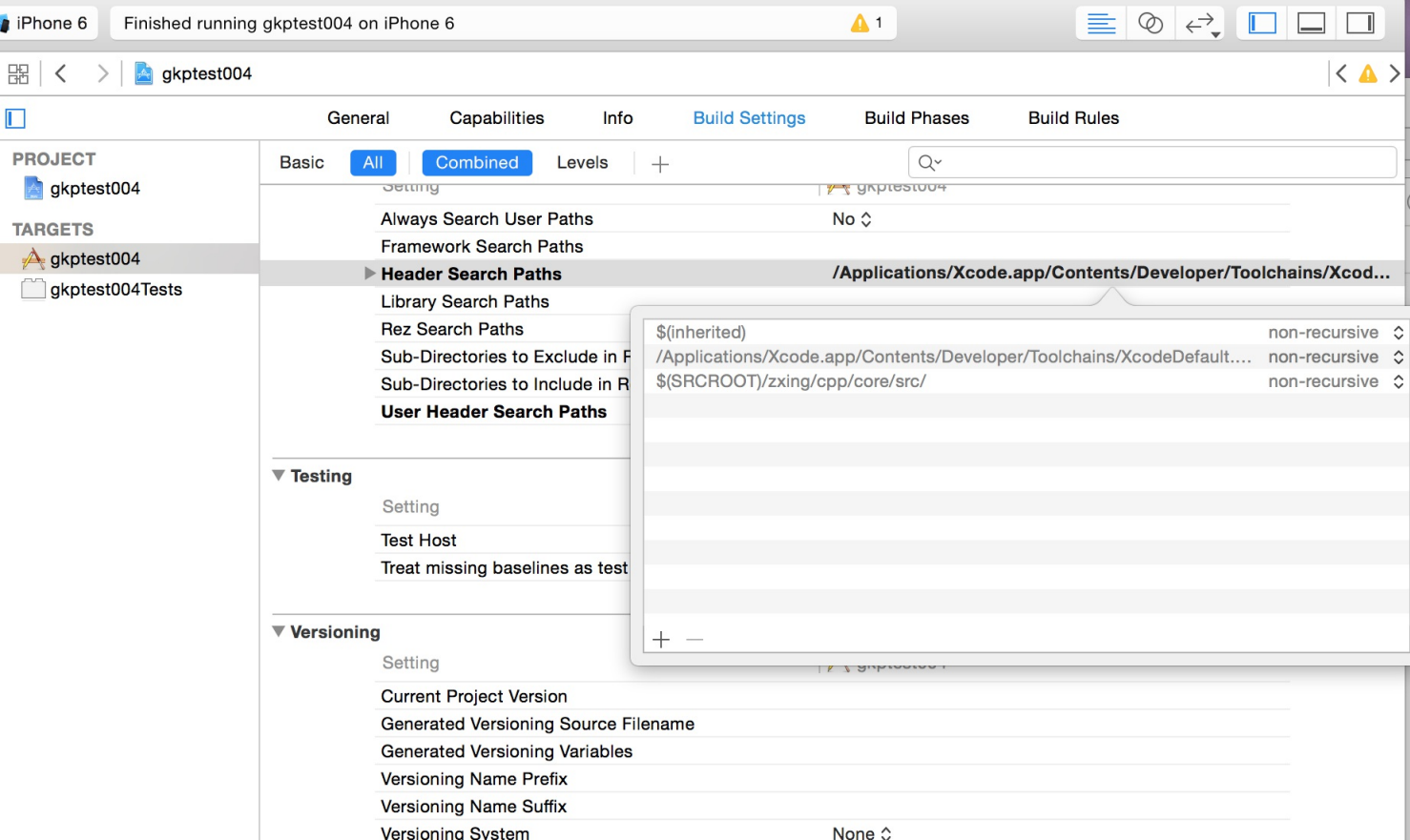


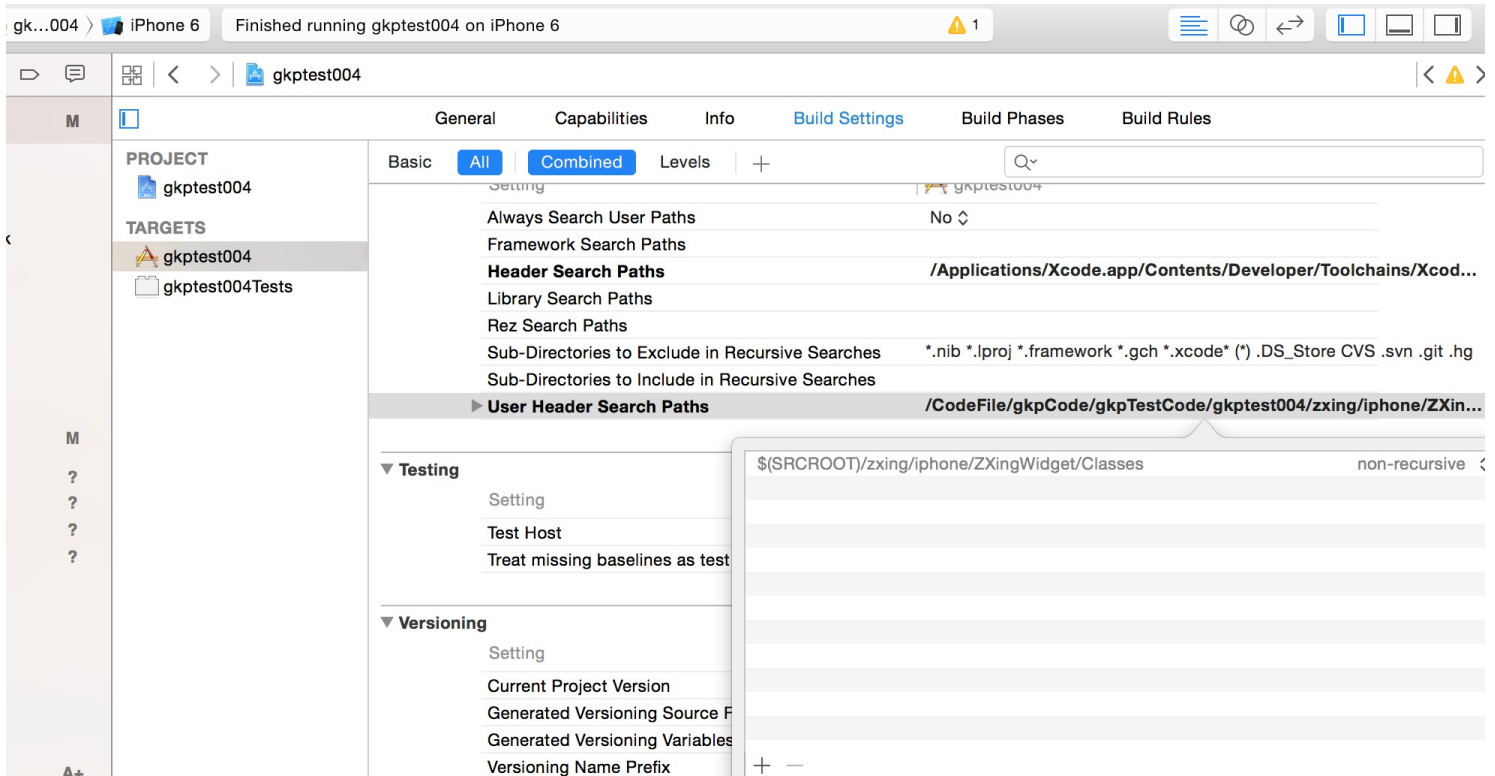
1、目录下存放zxing项目



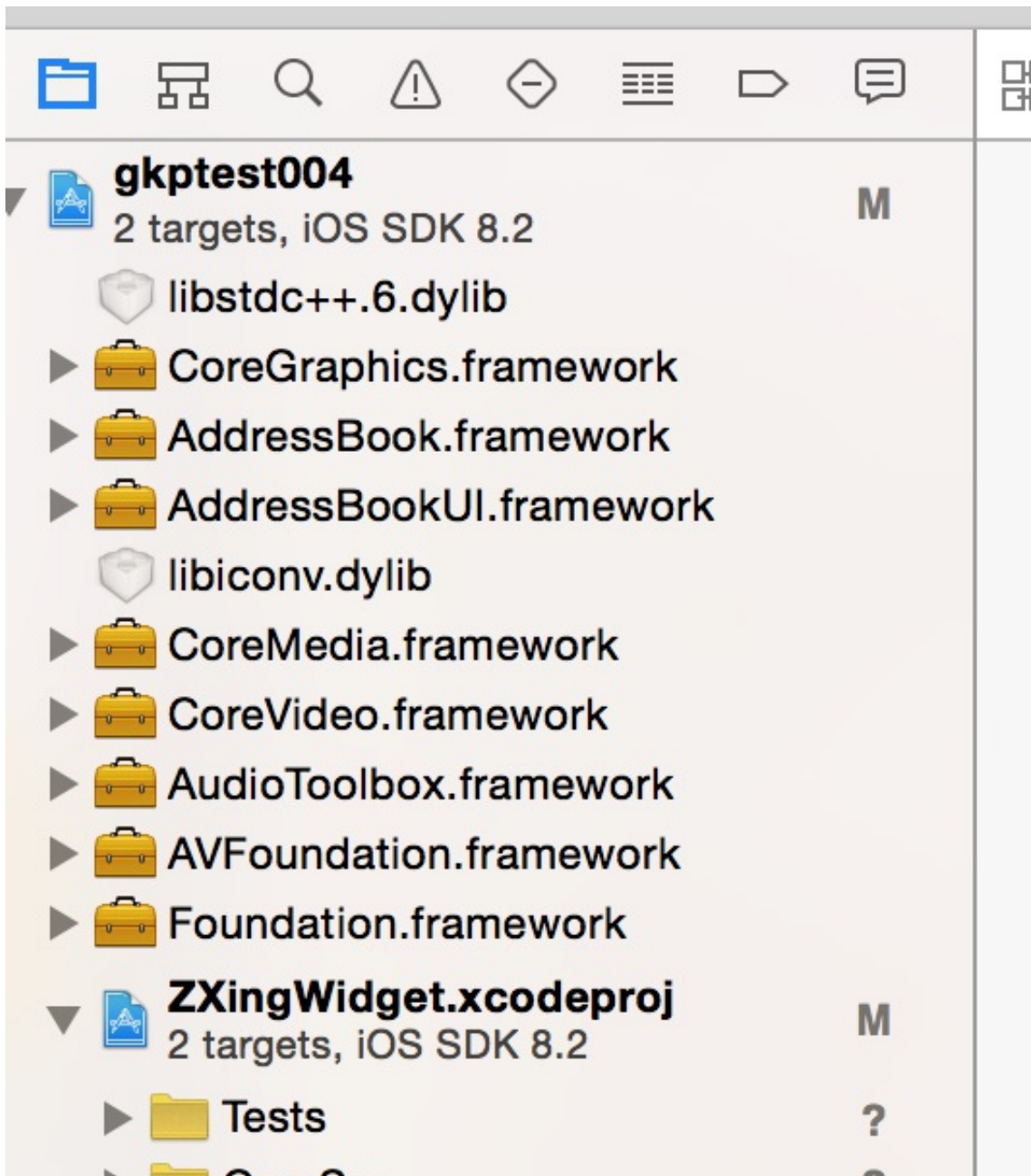
2、修改引用的路径

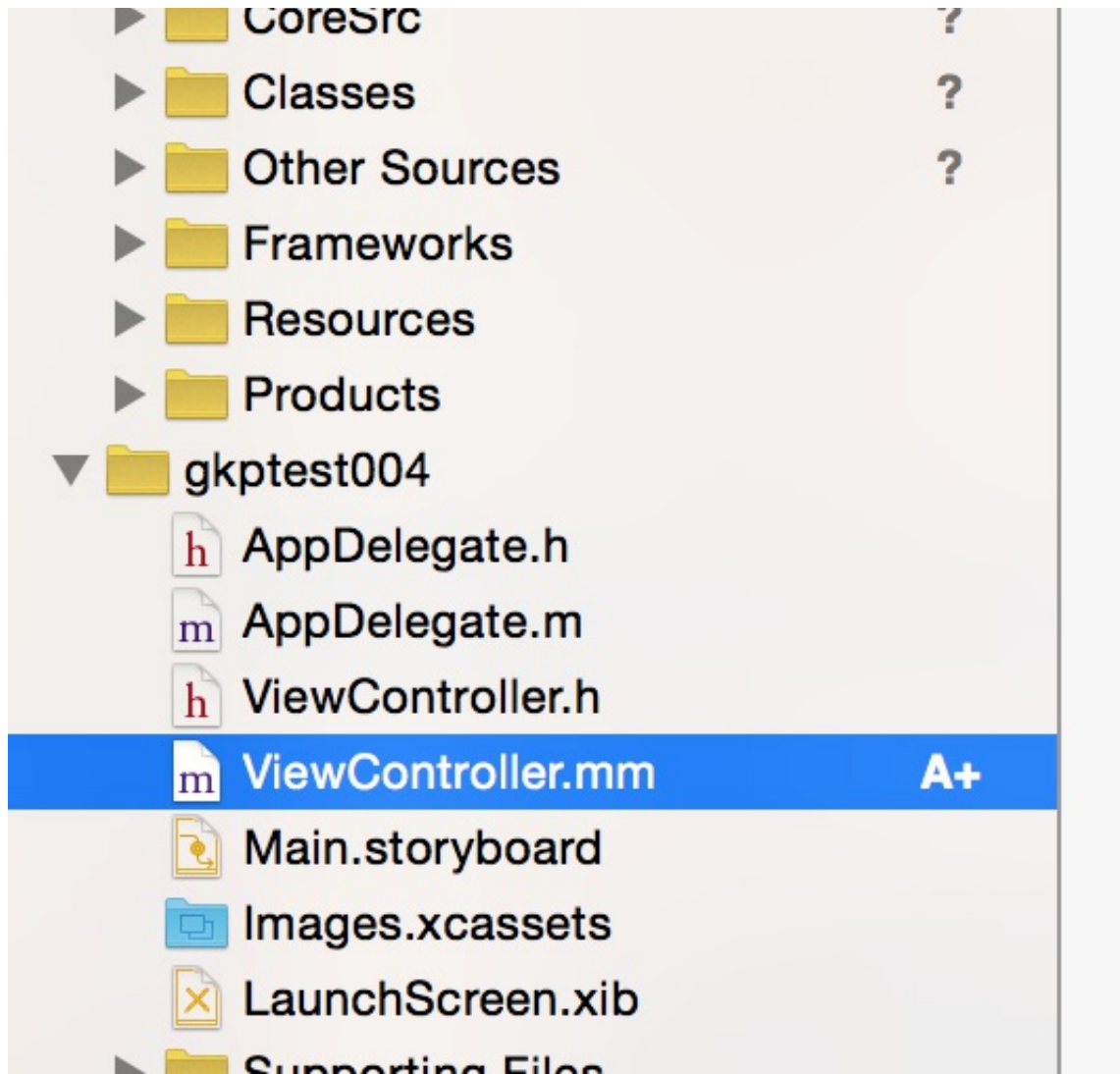
`$(SRCROOT)/zxing/cpp/core/src/`
`$(SRCROOT)/zxing/iphone/ZXingWidget/Classes`



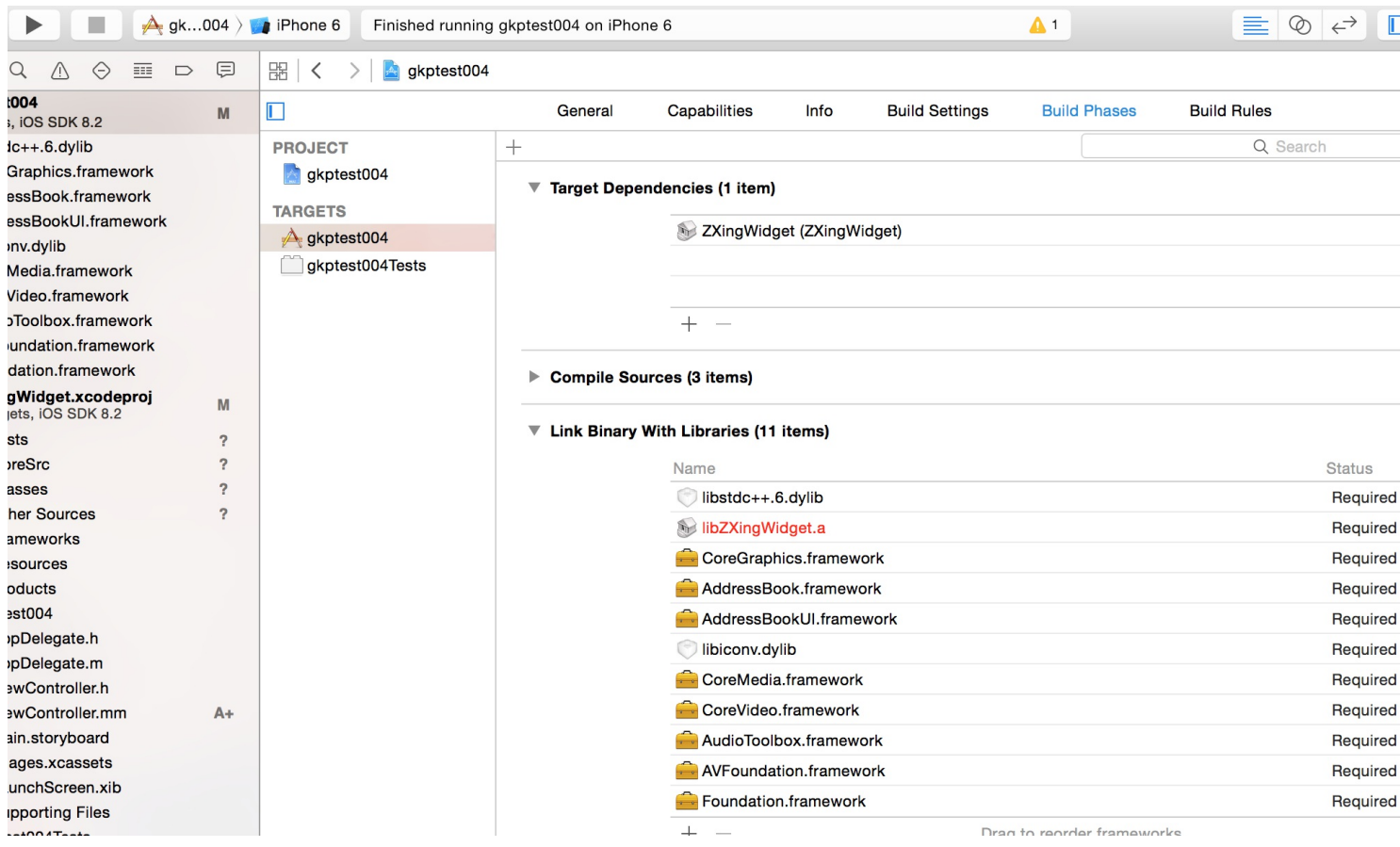


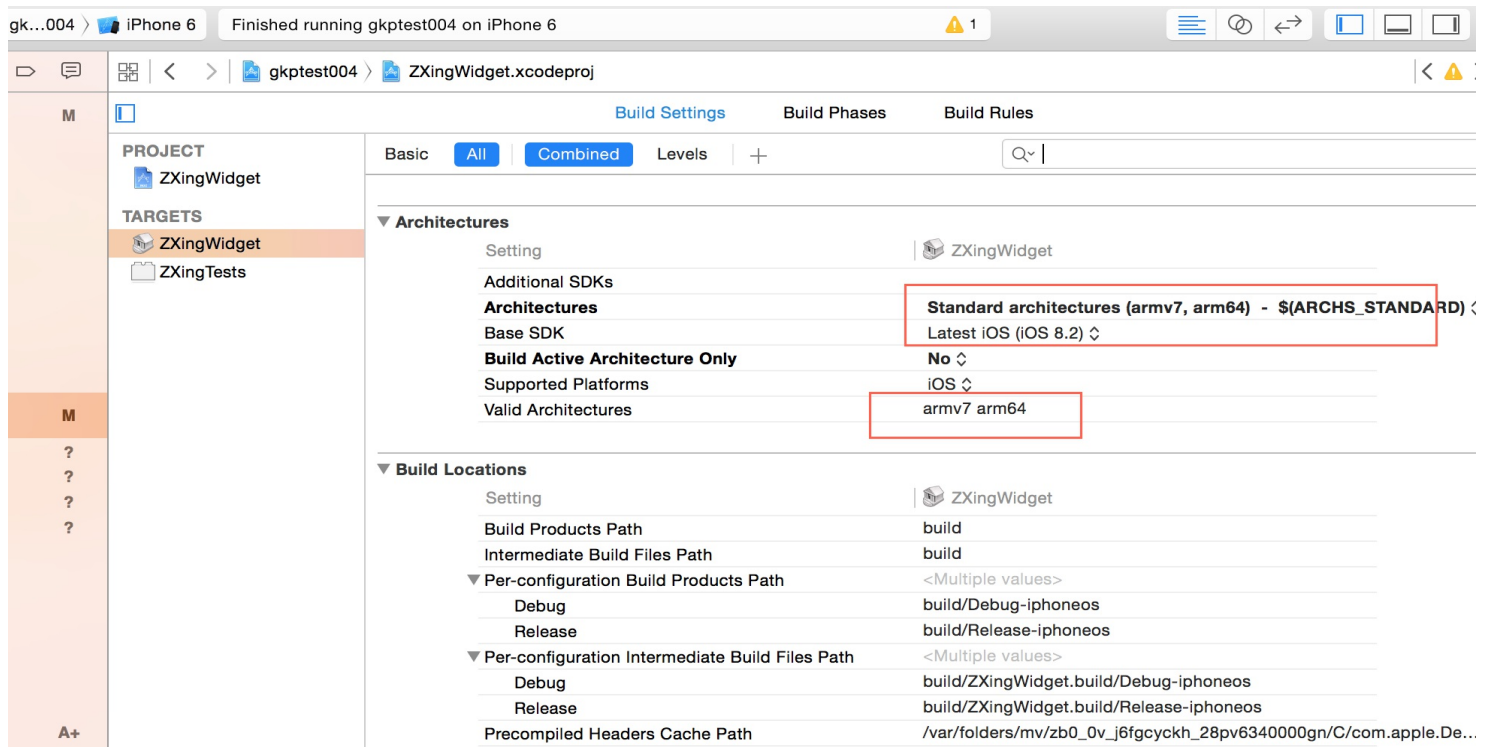
3、引用的标记为.mm文件





4、引用。a文件





参考: <http://thierry-xing.iteye.com/blog/1815295>

[在iOS和Android中使用二维码ZXing库及常见问题解决和整合后的代码](#)

最近做Android和iOS项目用到了ZXing进行二维码扫描, 其中在iOS整合比较麻烦, 遇到不少问题。

文章前半部分基础部分摘自此文章: <http://blog.devtang.com/blog/2012/12/23/use-zxing-library/> 写的比较清楚。

后面是自己在整合过程中遇到的问题和解决方案, 整合后的代码可以在附件中看到(包括, iOS和Android), 在此和大家进行分享。

[ZXing\(Github镜像地址\)](#)是一个开源的条码生成和扫描库(开源协议为[Apache2.0](#))。它不但支持众多的条码格式, 而且有各种语言的实现版本, 它支持的语言包括: Java, C++, C#, Objective-C, ActionScript和Ruby。

iOS篇

集成步骤

首先去Google Code或Github将ZXing的代码下载下来, 整个工程比较大, 我们只需要其中涉及iOS的部分, 所以最好做一些裁剪。简单来说, 我们只需要保留cpp和iphone这2个文件夹, 其余的全部删掉。如下图所示:

名称	修改日期
▼ 文件夹 zxing	今天 下午5:22
▶ 文件夹 zxing.appspot.com	2012-12-11 下午8:59
▶ 文件夹 zxingorg	2012-12-11 下午8:59
▶ 文件夹 cpp	2012-12-11 下午8:59
▶ 文件夹 csharp	2012-12-11 下午8:59
▶ 文件夹 iphone	2012-12-11 下午9:49
▶ 文件夹 javase	2012-12-11 下午8:59
▶ 文件夹 jruby	2012-12-11 下午8:59
▶ 文件夹 objc	今天 下午5:22
▶ 文件夹 symbian	2012-12-11 下午8:59
▶ 文件夹 actionscript	2012-12-11 下午8:58
▶ 文件夹 android	2012-12-11 下午8:58
▶ 文件夹 android-integration	2012-12-11 下午8:58
▶ 文件夹 androidtest	2012-12-11 下午8:58
文件 AUTHORS	2012-12-11 下午8:58
文件 build.properties	2012-12-11 下午8:58
文件 build.xml	2012-12-11 下午8:58
文件 CHANGES	2012-12-11 下午8:58
文件 COPYING	2012-12-11 下午8:58
▶ 文件夹 core	2012-12-11 下午8:58
文件 NOTICE	2012-12-11 下午8:58

只需保留这2个目录

接着我们继续裁剪，对于cpp这个目录，只保留cpp/core/src/zxing下面的内容，其余内容也可以删掉了。但是整个目录结构必须保持原样。裁剪完后，整个目录结构如下所示：

名称
▼ 文件夹 zxing
▼ 文件夹 iphone
▼ 文件夹 ZXingWidget
文件 ZXingWidget.xcodeproj
文件 ZXingWidget_Prefix.pch
文件 ZXingTests-Info.plist
▶ 文件夹 Tests
文件 exportList
▶ 文件夹 Classes
文件 beep-beep.caf
文件 README
▼ 文件夹 cpp
▼ 文件夹 core
▼ 文件夹 src
▶ 文件夹 zxing

接下来，我们把裁剪后的zxing目录整个移动到我们的iOS项目的目录下，并且把上图中可以看到的ZXingWidget.xcodeproj文件拖动到我们的iOS工程中。

下一步，我们需要设置ZXing项目和我们原本的iOS项目之间的依赖关系。在我们的iOS项目的设置中，点击build

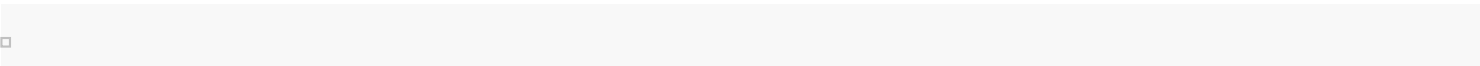
phases tab，然后增加 Target Dependencies 和 Link binary，并且增加这些framework依赖：



Java代码 ☆

```
<span style="font-size: 14px;">a. AVFoundation  
b. AudioToolbox  
c. CoreVideo  
d. CoreMedia  
e. libconv  
f. AddressBook  
g. AddressBookUI</span>
```

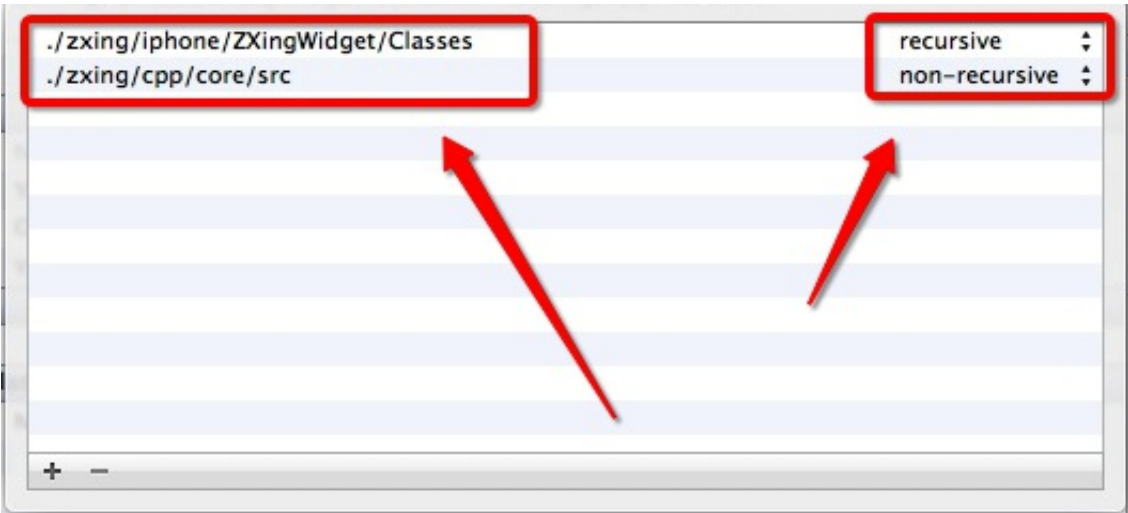
完成之后如下图所示：



最后一步，在设置中增加如下2个header search path:

- ./zxing/iphone/ZXingWidget/Classes
- ./zxing/cpp/core/src

需要注意的是，第一个path要设置成循环查找子目录，而第二个不循环查找，如下图所示：



恭喜你，完成这步之后，你就已经完成ZXing库的集成了。下面谈谈如何使用ZXing库来做二维码识别。

二维码识别

ZXing直接提供了一个扫描二维码的View Controller，即ZXingWidgetController。在需要使用的界面代码中，加入文件依赖：

Cpp代码 ☆

```
<span style="font-size: 14px;">#import <ZXingWidgetController.h>
#import <QRCodeReader.h> </span>
```

然后在需要扫描的时候，调用如下代码即可：

Cpp代码 ☆

```
<span style="font-size: 14px;">- (IBAction)scanPressed:(id)sender {
    ZXingWidgetController *widController = [[ZXingWidgetController alloc] initWithDelegate:self
showCancel:YES OneDMode:NO];
    NSMutableSet *readers = [[NSMutableSet alloc] init];
    QRCodeReader* qrcodeReader = [[QRCodeReader alloc] init];
    [readers addObject:qrcodeReader];
    [qrcodeReader release];
    widController.readers = readers;
    [readers release];
    [self presentViewController:widController animated:YES];
    [widController release];
}</span>
```

在ZXing扫描有结果时，会调用如下回调函数：

Cpp代码 ☆

```
<span style="font-size: 14px;">@protocol ZXingDelegate
- (void)zxingController:(ZXingWidgetController*)controller didScanResult:(NSString *)result;
- (void)zxingControllerDidCancel:(ZXingWidgetController*)controller;
@end</span>
```

遇到的问题和解决方案：

我在使用中遇到了一些问题，主要是编译的问题。

一个是找不到头文件。解决方法：把用到ZXing的源文件扩展名由.m改成.mm。

报错：Undefined symbols for architecture armv7s，[解决方法](#)：（1）把ZXingWidget的一个build target参数：“Build Active Architecture Only”修改成“NO”。（2）把main.m改为main.mm

报错：No such file or directory，出现该错误可能是你的Header Search Path写错了，或者就是你的zxing库的目录结构不是我上面强调的，好好检查一下吧。

如果你需要生成二维码做测试，推荐一个不错的在线生成二维码的网站：<http://cli.im/>

报错：**IOStream File not found**，解决方案把引入ZXingWidgetController的ViewController以及AppDelegate和main也需要修改成.mm，修改完后报错：Undefined symbols for architecture i386: "std::string::c_str() const"修改：TARGETS -> Build Settings -> Apple LLVM compiler 4.2 - Language 设置C++ Standard Library的值为Compiler Default
ZXingWidget项目：
TARGETS -> Build Settings -> Architectures
设置Architectures的值为Standard(armv7, armv7s)

报错：**clang: error: invalid deployment target for -stdlib=libc++ (requires iOS 5.0 or later)**:解决办法：将target 中的CLANG_CXX_LIBRARY项设置为libstdc++ 或者 引用libstdc++.6.dylib

错误：Undefined symbols for architecture i386: "_OBJC_CLASS_\$_ZXingWidgetController", referenced from: objc-class-ref in AttendanceVC.o "_OBJC_CLASS_\$_QRCodeReader", referenced from: objc-class-ref in AttendanceVC.o ld: symbol(s) not found for architecture i386 clang: error: linker command failed with exit code 1 (use -v to see invocation)：解决方法：在build phase中把libZXingWidget.a加入

和PhoneGap混用的时候会出现bug (error: default initialization of an object of const type 'NSString *const')：解决方法把：NSString* const CDVPageDidLoadNotification;改为extern NSString* const CDVPageDidLoadNotification;

如果大家还遇到过别的错，可以和我沟通，我们一起来解决。
整合好的iOS代码见附件，点击扫描，进入扫描功能，扫描结束后会在Console中打印出扫描出的文字。