

Xcode 4.1/4.2/4.2.1 免证书(iDP)开发+真机调试+生成IPA全攻略

当然您需要先越狱您的设备并通过Cydia安装AppSync

本文参考了多篇文章，综合其中的方法，并修正其中的错误，整理而成。

众所周知，在Xcode上开发的程序只能在模拟器中运行，如果要放到真机上则要花费99美金购买开发者证书iDP。这严重阻碍了我等草根开发者探索的脚步。写个小程序，同学间分享一下这个小小的愿望都不能满足，自然不能善罢甘休。

在没有iDP的情况下，要想将程序放到iPhone上调试，并最终发布IPA用于分享，需要以下几个步骤：

- 1.自己为自己颁发一个证书用于为生成的程序签名
- 2.修改工程配置以及Xcode的配置文件和二进制文件以阻止其验证和签名
- 3.通过自定义生成步骤，用伪造的证书为应用程序签名
- 4.使用一点小trick来生成IPA文件

开始之前的友情提示：

1. 本文给有一定基础的朋友看，各种命令是免不了的。**UNIX**中的基本命令比如**cp**，**mv**，**cd**，**chmod**，**sudo**啥的还有**vim**编辑器，如果您都没听说过，强烈建议您不要尝试本文提到的修改。一旦改错了轻则**Xcode**挂掉，重则系统崩溃。
2. 本文的每一个步骤都是必须的，如果您哪一步没有得到预期的结果，请不要继续，以免发生更加意外的事情。。。
3. 如果遇到解决不了的问题欢迎留言询问，务必给出详细的错误信息，否则无法判断。

1.创建证书

创建证书的过程比较简单，打开实用工具-钥匙串访问。然后在菜单栏里点击钥匙串访问-证书助理-创建证书来打开向导。第一个步骤比较重要，必须要把名称命名为**iPhone Developer**，将类型设定为代码签名，将"让我覆盖这些默认值"选中。之后的步骤无需更改，一路点击“确定”和“继续”来完成这个向导就可以。



2.修改Xcode的配置文件和二进制文件

下面的步骤稍微有点繁琐，您应该了解UNIX命令行的基本操作，并了解一种命令行文本编辑器，本文使用的是vim。尽管这里会给出完整的代码，但是关于修改和保存代码的基本操作，不再赘述。下面的操作请先将**Xcode**按**Command+Q**完全关闭。

(1)修改配置文件

进入目录并备份原文件（**4.1**和**4.2**在这里主要的差别是**SDK**的目录名不同）

（**Xcode4.1**请执行）`cd /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS4.3.sdk/`

（**Xcode4.2**请执行）`cd /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS5.0.sdk/`

`sudo cp SDKSettings.plist SDKSettings.plist.orig`

进行编辑

`sudo vim SDKSettings.plist`

将以下两段中的YES改为NO

```
<key>CODE_SIGNING_REQUIRED</key>
<string>YES</string>
```

和

```
<key>ENTITLEMENTS_REQUIRED</key>
<string>YES</string>
```

下面修改另外一个文件

进入目录并备份原文件

`cd /Developer/Platforms/iPhoneOS.platform/`

`sudo cp Info.plist Info.plist.orig`

进行编辑

`sudo vi Info.plist`

将全部的**XCiPhoneOSCodeSignContext** 修改成 **XCCodeSignContext**，网上的大部分文章说有2处，但我找到了3处，可能是Xcode 4.1要多一处？（**Xcode 4.2**也有三处）总之都改掉了。提示：在在vim中输入/要搜索的内容来搜索，按n键是搜索下一处。

(2)二进制补丁

#在桌面上建立script这个脚本

cd ~/Desktop

vim script

#(**Xcode 4.1**执行)在编辑器中输入如下内容

#!/bin/bash

**cd /Developer/Platforms/iPhoneOS.platform/Developer/Library/Xcode/Plug-ins/
iPhoneOS\ Build\ System\ Support.xcplugin/Contents/MacOS/**

dd if=iPhoneOS\ Build\ System\ Support of=working bs=500 count=255

printf "xc3x26x00x00" >> working

**/bin/mv -n iPhoneOS\ Build\ System\ Support iPhoneOS\ Build\ System\
Support.original**

/bin/mv working iPhoneOS\ Build\ System\ Support

chmod a+x iPhoneOS\ Build\ System\ Support

#(**Xcode 4.2**执行)在编辑器中输入如下内容

#!/bin/bash

**cd /Developer/Platforms/iPhoneOS.platform/Developer/Library/Xcode/
PrivatePlugIns/iPhoneOS\ Build\ System\ Support.xcplugin/Contents/MacOS/**

dd if=iPhoneOS\ Build\ System\ Support of=working bs=500 count=255

printf "xc3x26x00x00" >> working

**/bin/mv -n iPhoneOS\ Build\ System\ Support iPhoneOS\ Build\ System\
Support.original**

/bin/mv working iPhoneOS\ Build\ System\ Support

chmod a+x iPhoneOS\ Build\ System\ Support

保存并退出。（4.1和4.2在这里的区别也是目录名不同，就是代码中绿色的部分。4.1是**Plug-ins**而4.2是**PrivatePlugIns**）

授予这个脚本执行权限并执行它

chmod 777 script

./script

正常的话应该输出(具体的数字可能有差别)

231+1 records in

231+1 records out

115904 bytes transferred in 0.001738 secs (66694555 bytes/sec)

至此，对SDK中配置文件和二进制文件的修改就完成了

3.准备自定义的生成后脚本

连接互联网后执行

mkdir /Developer/iphoneentitlements401

cd /Developer/iphoneentitlements401

curl -O http://www.alexwhitemore.com/iphone/gen_entitlements.txt

mv gen_entitlements.txt [gen_entitlements.py](#)

chmod 777 [gen_entitlements.py](#)

如果您已经联网，则请直接转到步骤4，如果您没有联网，那么请手动创建/Developer/iphoneentitlements401/gen_entitlements.py并授予其执行权限，这个文件的内容为

#!/usr/bin/env python

import sys

import struct

```

if len(sys.argv) != 3:
    print "Usage: %s appname dest_file.xcent" % sys.argv[0]
    sys.exit(-1)

APPNAME = sys.argv[1]
DEST = sys.argv[2]

if not DEST.endswith('.xml') and not DEST.endswith('.xcent'):
    print "Dest must be .xml (for ldid) or .xcent (for codesign)"
    sys.exit(-1)

entitlements = """
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>application-identifier</key>
    <string>%s</string>
    <key>get-task-allow</key>
    <true/>
</dict>
</plist>
""" % APPNAME

f = open(DEST, 'w')
if DEST.endswith('.xcent'):
    f.write("\xfa\xde\x71\x71")
    f.write(struct.pack('>L', len(entitlements) + 8))
    f.write(entitlements)
f.close()

```

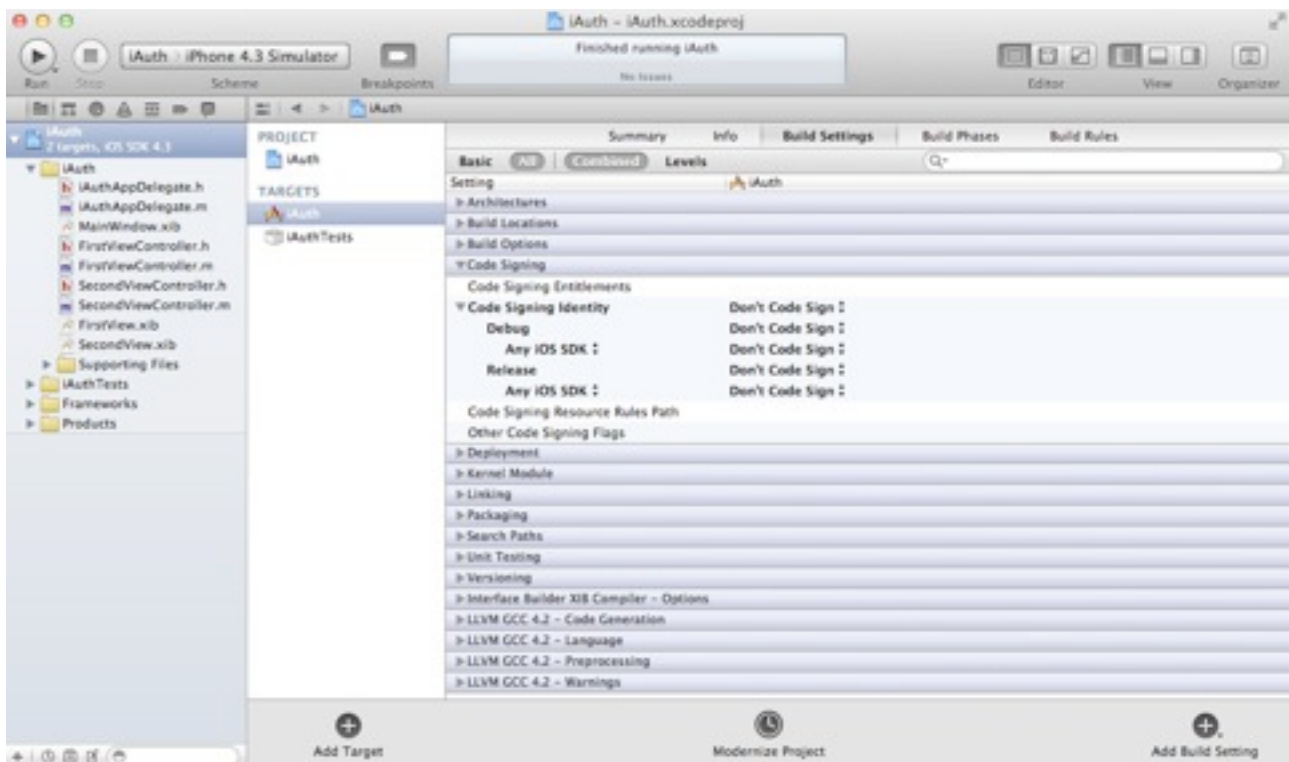
4. 修改工程设置

特别注意：本阶段之前的修改配置文件、准备脚本等，只需要做一次。但本阶段的操作，对每个需要真机调试的工程都要做一遍。

这个步骤的主要作用是支持真机调试，如果不做这个步骤，仍然可以通过步骤5来生成ipa在真机上运行，但是无法使用Xcode内置的调试器对在真机上运行的程序进行单步跟踪。如果您的程序在点击Run真机调试时秒退，请检查此步骤是否正确完成。

(1) 禁用Xcode自动的签名操作

将工程配置中所有的Code Signing选项全部设为Don't Code Sign，如图。可能需要先点击“All”让这个选项显示出来

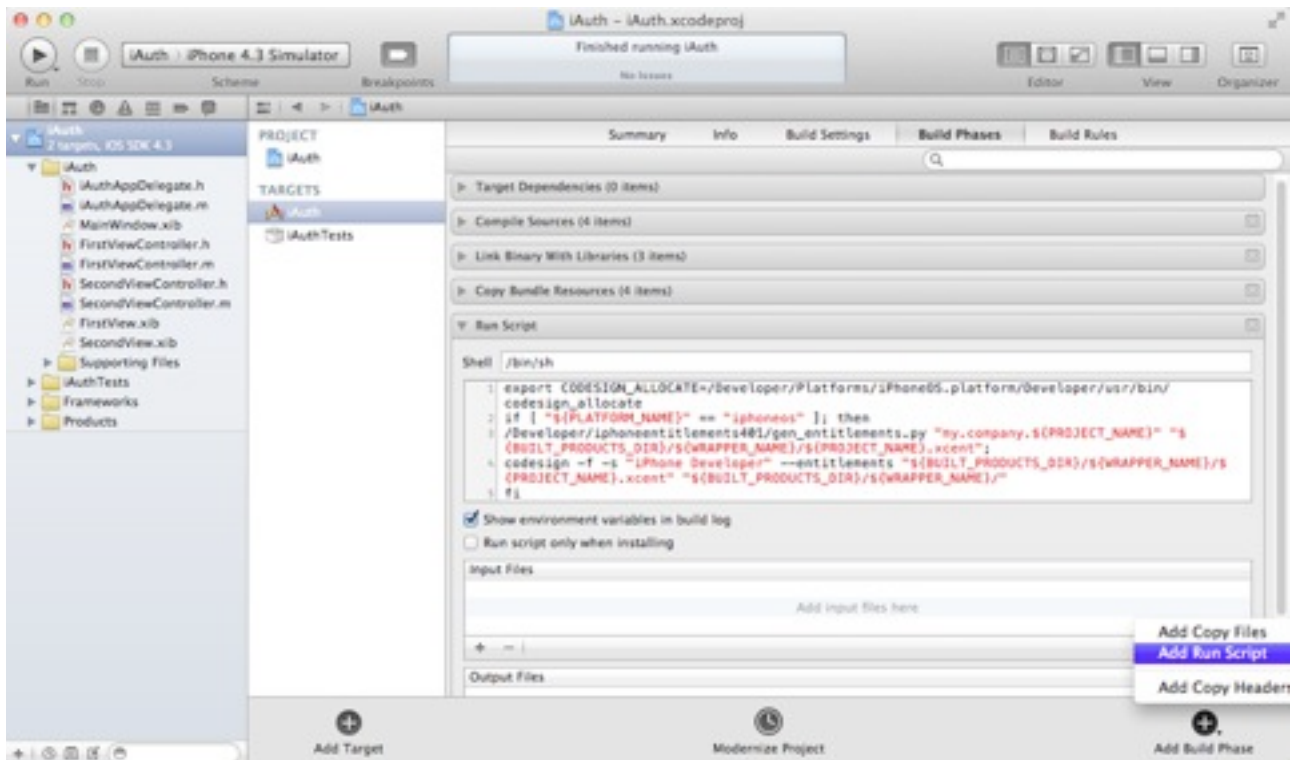


(2)添加自定义的生成后脚本

在Build Phases中添加一个Phase，右下角的Add Build Phase，然后单击Add Run Script，输入以下脚本

```
export CODESIGN_ALLOCATE=/Developer/Platforms/iPhoneOS.platform/Developer/
usr/bin/codesign_allocate
if [ "${PLATFORM_NAME}" == "iphoneos" ]; then
/Developer/iphoneentitlements401/gen_entitlements.py "my.company.$
${PROJECT_NAME}" "${BUILT_PRODUCTS_DIR}/${WRAPPER_NAME}/${
${PROJECT_NAME}.xcent";
codesign -f -s "iPhone Developer" --entitlements "${BUILT_PRODUCTS_DIR}/${
WRAPPER_NAME}/${PROJECT_NAME}.xcent" "${BUILT_PRODUCTS_DIR}/${
WRAPPER_NAME}/"
fi
```

如图所示：



至此配置全部完成，下面就可以插上iPhone，重新选择生成目标来测试一下在线调试了！如果是第一次使用该设备调试，请先在Organizer中将设备切换为开发模式，具体操作请见常见问题5。

5.旁门左道生成IPA文件

如果我的程序调试好了，怎么才能发给别人用呢？正常情况下IPA文件是从Xcode的Organizer中输出的，但是我们没有证书，这样输出会产生错误。我们只能用小trick来完成这个操作了。

先将代码生成成为Release目标，然后打开工程的输出文件夹，通常情况下这个目录是 **/Users/你都用户名/Library/Developer/Xcode/DerivedData/以工程名打头的文件夹/Build/Products/Release-iphoneos**

很纠结吧~这个目录下有个.app的文件，就是生成的程序了。把这个.app拖到iTunes中，它会出现现在应用程序那个列表中，然后再把它从iTunes的那个列表中拖出来（比如拖到桌面），发生了什么？哈哈，它就这样变成.ipa了！

把这个.ipa发给朋友，大家就可以跟您一起分享这个程序了。

文章发表以来点击量已经超过了5000次，也收到了不少热心博友的反馈和疑问。现加以整理，列为常见问题。感谢大家的关注！（2011年12月2日）

常见问题：

不成功的先看这里！！本文点击量超过**6000**，无数网友亲测成功，如果您失败了基本可以确定是您没有严格按照文中的方法操作，下面列出常见问题，切望各位朋友提问前先看！（2011年12月14日）

1. 程序运行时报错，或者根本不能通过编译。错误信息类似（感谢博友rinzeng提出）ARC forbids explicit message send of 'release'

'release' is unavailable: not available in automatic reference counting mode

解答：这个错误与本文讨论的问题无关，即使在模拟器上运行也照样会报错。但我愿意在这里予以解答。这个错误是由在iOS 5中最新引进的ARC内存管理机制造成的。解决方法有2种，一是不要加release语句。二是在Build Settings中关闭Objective-C Automatic Reference Counting

2. 各种错误，错误信息中包含“No such file or directory”这句话

解答：错误信息的含义非常清楚，就是“没有这个文件或者目录”。这类错误通常是由于您拷贝代码时不全或者开发环境安装错误等问题导致的。请再次检查报错的目录是否存在，检查代码与本文给出的是否严格一致，各种检查吧~总之是低级错误

3. 联机调试时程序秒退，或者任何时候报错，错误信息中包含“code sign”、“CERT”或者“certificate”字样的

解答：证书错误或者签名错误，这种问题就不要问了，肯定是因为您没有严格文中的步骤做。提醒您检查的地方有(1)有没有设置为**Don't Code signing** (2)生成后事件的代码是否已经正确粘贴 (3)那个**Python**脚本是否已经成功执行

4. iPad能用吗？

解答：我没有iPad，所以没法准确回答。但从热心博友的回复来看，iPad无疑是可以用的。

如果要用ipad 需要把第四步脚本文件里的 **platform_name == iphones** 换成 **platform_name == ipads** 不然会报错的~（感谢博友xyishao的建议）

5. 为什么我的Build for Archiving选项是灰色的？

解答：是因为您没有把设备插到电脑上。虽然这个问题挺蠢的，但我曾经也有过这个疑问，哈哈~~

经热心网友xc7296815提醒，设备未开启开发模式也会导致Build for Archiving不可用，开启开发模式的方法是：插上设备，点Xcode右上角的Organizer图标，选中您的设备，在右边窗口中点击“Use for Development” (2011年12月25日)

6. 我的iOS版本/Xcode版与你的不同，能用吗？

解答：我只在iPod Touch 4 + iOS 4.3.3 & 5.0 + Xcode 4.1/4.2上测试可用，在iPhone 4和iPad上虽未测试，但根据网友的反馈是可用的。其他环境我没有测试过，也没有条件测试，因此当您的环境与我的不同时，别问我不可用，您可以试一下，然后把结果告诉我，也为其他的网友提供方便，在此先谢谢您了！

7. 真机调试时出现Error launching remote program: failed to get the task for process xxx错误（2012年1月26日更新）

解答：我在调试中貌似没有遇到过这个问题，但是不少网友指出会有这个问题，谁来告诉我什么情况下会出这个问题呢？如果您遇到了这个问题，请阅读Apple官方的帮助文档http://developer.apple.com/library/ios/#qa/qa1710/_index.html