

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

2011-08-19 14:11:14 | 分类: [Mac 使用和开发](#) | 标签: | 字号大中小 订阅

开发环境使用的是目前为止最新的稳定版软件: Mac OS X Lion 10.7 + Xcode 4.1
目前 Xcode 4.2 Preview 版也已经发布, 据说其修改方法跟 4.1 非常类似, 只改动了一行代码, 请参看参考文章的第二篇。本文仍以 4.1 版本为例。

**更新: 现在 Xcode 4.2 正式版和 iOS 5 均已发布, 下面补充上 4.2 的修改方法。
(2011-10-24)**

各步骤会标明版本, 比如 (Xcode4.1 请执行) 和 (Xcode4.2 请执行)

未标明的步骤为两个版本均需执行的步骤!

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。

未经许可, 请勿转载。本文首次发表于 <http://kwd.blog.163.com/>

参考的文章有:

<http://laolang.cn/back-end-develop/helloworld-iphone-developer.html>

<http://ccmos.tw/blog/2011/06/30/xcode4-port-program-to-idevice-without-idp/>

<http://www.cnblogs.com/flyingzl/articles/2207717.html>

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使用这个伪造的证书
3. 使用自定义的步骤为应用程序签名
4. 使用一点小 trick 来生成 IPA 文件

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/PrivateP
lugIns/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.alexwhittemore.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！

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。我们只能用个小 trick 来完成这个操作了。

先将代码生成 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。

文章好长... 欢迎各路大侠指正。

评论这张



转发至微博



转发至微博



0 人

分享到: _____

阅读 (2264) | 评论 (39) | 引用 (0) | 举报

[自己动手修理 MacBook\(小熊猫\) Multi-touch 触控板的微动开关](#)

[Mac 系统下显示隐藏文件](#)

历史上的今天


相关文章

最近读者

评论



<#--最新日志，群博日志-->

-  <#--推荐日志-->

•






<#--引用记录-->

•



<#--博主推荐-->

-  <#--随机阅读-->
-  <#--首页推荐-->
-  <#--相关文章-->

•



<#--历史上的今天-->

•



<#--右边模块结构-->



<#--评论模块结构-->



<#--引用模块结构-->



<#--博主发起的投票-->



.

页脚

[公司简介](#) - [联系方法](#) - [招聘信息](#) - [客户服务](#) - [隐私政策](#) - [博客风格](#) - [手机博客](#)
- [VIP 博客](#) - 订阅此博客

网易公司版权所有 ©1997-2011

