

[登录](#) | [注册](#)

# 曾梦想仗剑走天涯

[目录视图](#)[摘要视图](#)[RSS](#) [订阅](#)

## 个人资料



弦苦

访问: 2586676次

积分: 9047

等级: [BLOG > 6](#)

排名: 第1490名

原创: 118篇 转载: 25篇

译文: 3篇 评论: 275条

## 个人独立博客

[col.dog](#)

## 文章搜索

## 文章分类

[嵌入式开发](#) (15)  
[VxWorks](#) (9)  
[Linux](#) (2)  
[网络通信/流媒体](#) (31)  
[VC/MFC/Windows](#) (21)  
[C/C++基础](#) (20)  
[多线程/分布式/并行/云](#) (9)  
[编程工具&生产效率](#) (11)  
[数据结构、算法/设计模式](#) (4)  
[计算机及网络安全](#) (2)  
[C#.NET](#) (1)  
[数据库](#) (2)  
[Android](#) (2)  
[MacTalk](#) (7)

[移动信息安全的漏洞和逆向原理](#) [【观点】世界上最好的语言是什么](#) [58到家周俊鹏: webpack PK fis, 实现前端工程化我更喜](#)  
[欢前者](#) [晒知识图谱, 享技术荣誉](#)

## iOS Provisioning Profile(Certificate)与Code Signing详解

标签: [Certificate](#) [App\\_ID](#) [Provisioning\\_Profile](#) [App\\_Group](#) [CodeSigning](#)

2015-01-13 22:01

208977人阅读

[评论\(33\)](#)[收藏](#)[举报](#)

### 分类:

[iOS \(8\)](#)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。

[目录\(?\)](#)[\[+\]](#)

## 引言

关于开发证书配置 ( Certificates & Identifiers & Provisioning Profiles ), 相信做 **iOS** 开发的同学没少被折腾。对于一个 iOS 开发小白、半吊子 ( 比如像我自己 ) 抑或老兵, 或多或少会有或曾有过以下不详、疑问、疑惑甚至困惑:

1. 什么是App ID ? Explicit/Wildcard App ID有何区别 ? 什么是App Group ID ?
2. 什么是证书 ( Certificate ) ? 如何申请 ? 有啥用 ?
3. 什么是Key Pair ( 公钥/私钥 ) ? 有啥用 ? 与证书有何关联 ?
4. 什么是签名 ( Signature ) ? 如何签名 ( CodeSign ) ? 怎样校验 ( Verify ) ?
5. 什么是 ( Team ) Provisioning Profiles ? 有啥用 ?
6. Xcode如何配置才能使用iOS真机进行开发调试 ?
7. 多台机器如何共享开发者账号或证书 ?
8. 遇到证书配置问题怎么办 ?
9. Xcode 7免证书调试真机调试

本文将围绕相关概念及背景做个系统的梳理串烧, 于条分缕析中对证书体系进行抽丝剥茧, 逐步揭开签名机制的神秘面纱。图穷匕首见, 水落而石出, 包教不包会, 不会请再来。

从 Xcode 7 开始支持普通 Apple 账号进行**免证书真机调试**, 详情参考最新官方文档《Launching Your App on Devices》, 或参考本文最后一节简介。

## 写在前面

1. 假设你使用过 Apple 设备 ( iMac/iPad/iPhone ) 且注册过 Apple ID ( Apple Account ), 详情参考《[创建和开始使用 Apple ID](#)》。
2. 假设你或你所在的开发组已加入苹果开发者计划 ( Enroll in iOS Developer Program to become a member ), 即已注册开发者账号 ( Apple Developer Account ) 。

- 只有拥有开发者账号, 才可以申请开发/发布证书及相关配置授权文件, 进而在 iOS 真机上开发调试 Apps 或发布到 App Store。

iOS (9)  
SCM (4)  
git (2)

#### 文章存档

2016年05月 (1)  
2016年02月 (2)  
2015年11月 (1)  
2015年09月 (3)  
2015年07月 (2)

展开

#### 阅读排行

VC操作INI文件 (1115148)  
iPhone屏幕尺寸、分辨率 (236180)  
iOS Provisioning Profile( (208917)  
Mac OS X上使用Wiresh: (104838)  
Xcode基本操作 (90903)  
TCP通信流程解析 (73174)  
WinSock编程基础 (47995)  
VLAN原理详解 (30434)  
Mac下git通过SSH进行免 (19798)  
使用Lua脚本为wireshark (18215)

#### 评论排行

iOS Provisioning Profile( (33)  
Mac OS X上使用Wiresh: (22)  
VLAN原理详解 (19)  
Xcode基本操作 (19)  
iPhone屏幕尺寸、分辨率 (16)  
TCP通信流程解析 (10)  
iOS8扩展插件开发配置 (9)  
应用层通信协议开发浅析 (8)  
WinSock编程基础 (8)  
VC项目配置基础 (8)

#### 推荐文章

\* 程序员10月书讯, 评论得书  
\* Android中Xposed框架篇--修改系统位置信息实现自身隐藏功能  
\* Chromium插件(Plugin) 模块(Module) 加载过程分析  
\* Android TV开发总结--构建一个TV app的直播节目实例  
\* 架构设计: 系统存储--MySQL简单主从方案及暴露的问题

#### 最新评论

iOS Provisioning Profile(Certifica  
kharis: 新建项目的时候报了2个  
error, 1.Failed to create  
provisioning p...  
Mac下git通过SSH进行免密码安  
wmymartin: 想问下如果没有设置  
公钥密码, 弹出窗口直接回车,

- 开发者账号分为 Individual 和 Company/Organization 两种类型。如无特别交代, 下文基于 \$99/Year 的普通个人开发者 ( Individual ) 账号展开。

3.若要真机调试实践, 你必须至少拥有一台装有 Mac OS X/Xcode 的 Mac 开发机 ( iMac or MacBook ), 其上自带原生的 Keychain Access。

## 一.App ID ( bundle identifier )

在苹果官方的开发者计划 ( Apple Developer Member Center ) 层面, App ID 即 Product ID, 用于标识一个或者一组 App。

在 Mac/iOS 开发语境中, **bundle** ( 捆绑 ) 是指一个内部结构按照标准规则组织的特殊目录。在 Mac OS 应用程序目录下的某个 \*.app 上可右键显示包内容 ( Contents ), 其本质上就是可执行二进制文件 ( MacOS/ ) 及其资源 ( Resources/ ) 的**打包组合**。因此, 在 Xcode 中配置的 Bundle Identifier 必须和 App ID 是一致的 ( Explicit ) 或匹配的 ( Wildcard )。

App ID 字符串通常以**反域名** ( reverse-domain-name ) 格式的 Company Identifier ( Company ID ) 作为前缀 ( Prefix/Seed ), 一般不超过 255 个 ASCII 字符。

App ID 全名会被追加 Application Identifier Prefix ( 一般为 TeamID. ), 分为两类:

- **Explicit App ID**: 唯一的 App ID, 用于唯一标识一个应用程序。例如 “com.apple.garageband” 这个 App ID, 用于标识 Bundle Identifier 为 “com.apple.garageband” 的 App。
- **Wildcard App ID**: 含有通配符的 App ID, 用于标识一组应用程序。例如 “\*” ( 实际上是 Application Identifier Prefix ) 表示所有应用程序; 而 “com.apple.\*” 可以表示 Bundle Identifier 以 “com.apple.” 开头 ( 苹果公司 ) 的所有应用程序。

用户可在 Developer Member Center 网站上注册 ( Register ) 或删除 ( Delete ) 已注册的 App IDs。App ID **被配置到**【XcodeTarget】InfoBundle Identifier 下; 对于 Wildcard App ID, 只要 bundle identifier 包含其作为 Prefix/Seed 即可。

## 二.设备 ( Device )

Device 就是运行 iOS 系统用于开发调试 App 的设备。每台 Apple 设备使用 **UDID** ( Unique Device Identifier ) 来唯一标识。

iOS 设备连接 Mac 后, 可通过 iTunes->Summary 或者 Xcode->Window->Devices 查看其**UDID**。Apple Member Center 网站个人账户下的 **Devices** 中包含了注册过的所有可用于开发和测试的设备, 普通个人开发账号每年累计最多只能注册**100**个设备。

- Apps signed by you or your team run only on **designated** development devices.
- Apps run only on the test devices you **specify**.

用户可在网站上注册或启用/禁用 ( Enable/Disable ) 已注册的Device。

本文的 Devices 是指**连接到 Xcode** 被授权用于开发测试的iOS设备 ( iPhone/iPad )。

## 三.开发证书 ( Certificates )

### 1.证书的概念

**证书**是由公证处或认证机关开具的证明资格或权力的**证件**, 它是表明 ( 或帮助断定 ) 事理的一个**凭证**。证件或凭证的尾部通常会烙印**公章**。

每个中国人一生可能需要70多个证件, 含15种身份证明。证件中 “必需的” 有30到40个。将这些证件按时间顺序铺开, 那就是一个天朝子民的一生——持**准生证**许可落地, 以**户籍证明**入籍, 以**身份证**认证身份, 持**结婚证**以合法同居, 最终以**死亡证明**注销。

### 2.数字证书的概念

**数字证书**就是互联网通讯中**标志**通讯各方**身份信息**的一串数字, 提供了一种在 Internet 上验证通信**实体身份**的方式, 其作用类似于司机的驾驶执照或日常生活中的身份证。它是由一个由权威机构——**CA机构**, 又称为证书授权中心 ( Certificate Authority ) 发行的, 人们可以在网上用它来识别对方的身份。

- 数字证书是一个经证书授权中心数字签名的包含**公开密钥拥有者信息**以及**公开密钥**的文件。最简单的证书包含一个公开密钥、名称以及证书授权中心的数字签名。

这样不行的。该怎么解决

iPhone屏幕尺寸、分辨率及适配  
Mango\_ios: 楼主好, 写得真棒

VLAN原理详解  
zsk900413: 很好的文章

VLAN原理详解  
Pfc小书生: 简单明了

VLAN原理详解  
\_zSir: 感谢分享

iOS Provisioning Profile(Certifica  
AiN\_Hy: 感谢楼主分享

iOS Provisioning Profile(Certifica  
avi9111: 再次路过 看看

iPhone屏幕尺寸、分辨率及适配  
Ylem: 希望允许转载下, 非常感谢!

iOS Provisioning Profile(Certifica  
罗先雄: @jeffahjd:已经解决了, 是签名证书的问题, 重写生成了 就没问题了。谢谢你。

- 数字证书还有一个重要的特征就是**时效性**：只在特定的时间段内有效。

数字证书中的**公开密钥**（公钥）相当于**公章**。

某一认证领域内的**根证书**是 CA 认证中心给自己颁发的证书，是信任链的**起始点**。安装根证书意味着对这个 CA 认证中心的信任。

为了防止 **GFW** 进行中间人攻击(MitM)，例如篡改 **github** 证书，导致无法访问 github 网站等问题，可选择**不信任 CNNIC**：

- 在 [ 钥匙串-系统 ] 中双击 **CNNIC ROOT**，在【信任】|【使用此证书时】下拉选择【永不信任】。



在天朝子民的一生中，户籍证明可理解为等效的**根证书**：有了户籍证明，才能办理身份证；有了上流的身份证，才能办理下游居住证、结婚证、计划生育证、驾驶执照等认证。

### 3.iOS (开发) 证书

iOS 证书是用来证明 iOS App 内容（bundle with executable and resources）的合法性和完整性的**数字证书**。对于想安装到真机或发布到 AppStore 的应用程序（App），只有经过**签名验证**（Signature Validated）才能确保来源可信，并且保证 App 内容是完整、未经篡改的。

iOS 证书分为两类：**Development** 和 **Production**（Distribution）。

- **Development** 证书用来开发和调试应用程序：A **development certificate** identifies you, as a team member, in a development provisioning profile that allows apps signed by you to **launch** on devices.
- **Production** 主要用来分发应用程序（根据证书种类有不同作用）：A **distribution certificate** identifies your team or organization in a distribution provisioning profile and allows you to **submit** your app to the store. Only a team agent or an admin can create a distribution certificate.

普通个人开发账号最多可注册 iOS Development/Distribution 证书各**2**个，用户可在网站上删除（Revoke）已注册的 Certificate。

下文主要针对 iOS App 开发调试过程中的**开发证书**（Certificate for Development）。

### 4.iOS (开发) 证书的根证书

那么，iOS 开发证书是谁颁发的呢？或者说我们是从哪个 CA 申请到用于 Xcode 开发调试 App 的证书呢？

iOS 以及 Mac OS X 系统（在安装 Xcode 时）将自动安装 **AppleWWDRCA.cer** 这个中间证书（**Intermediate Certificates**），它实际上就是 iOS（开发）证书的证书，即**根证书**（Apple Root

Certificate)。

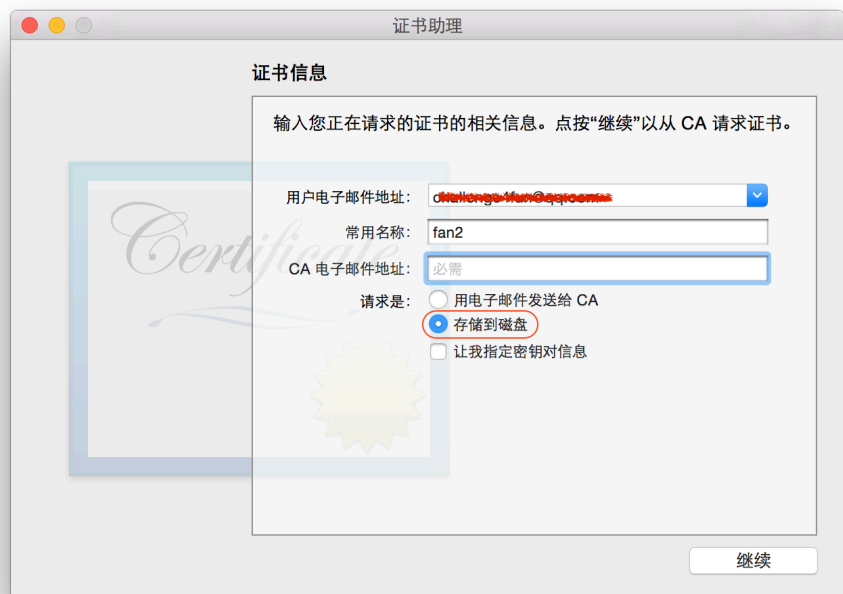
AppleWWDRCA ( Apple Root CA ) 类似注册管理户籍的公安机关户政管理机构, [AppleWWDRCA.cer](#) 之于 iOS ( 开发 ) 证书则好比户籍证之于身份证。

如果 Mac Keychain Access 证书助理在申请证书时尚未安装过该证书, 请先下载安装 ( Signing requires that you have both the signing identity and the intermediate certificate installed in your keychain ) 。

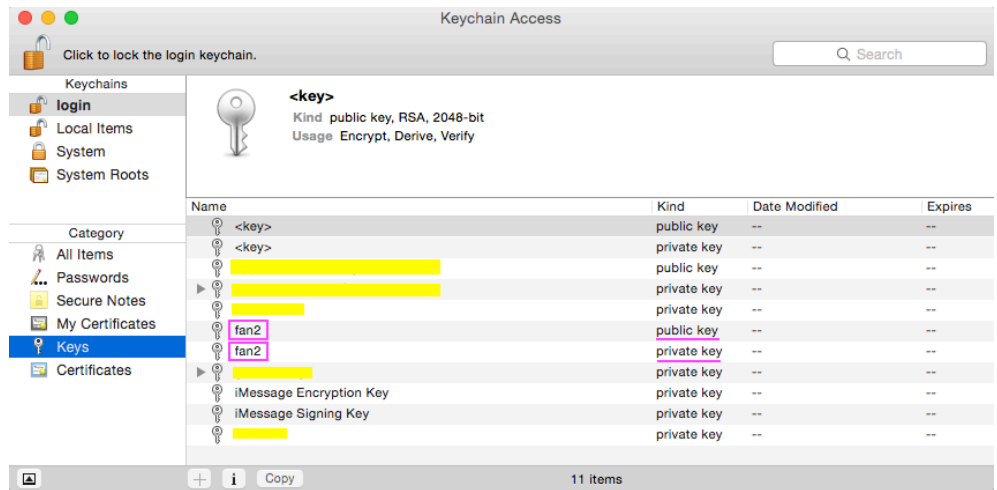


### 5. 申请证书 ( CSR : Certificate Signing Request )

可以在缺少证书时通过 Xcode Fix Issue 自动请求证书, 但是这会掩盖其中的具体流程细节。这里通过 Keychain 证书助理从证书颁发机构请求证书: 填写开发账号邮件和常用名称, 勾选【存储到磁盘】。



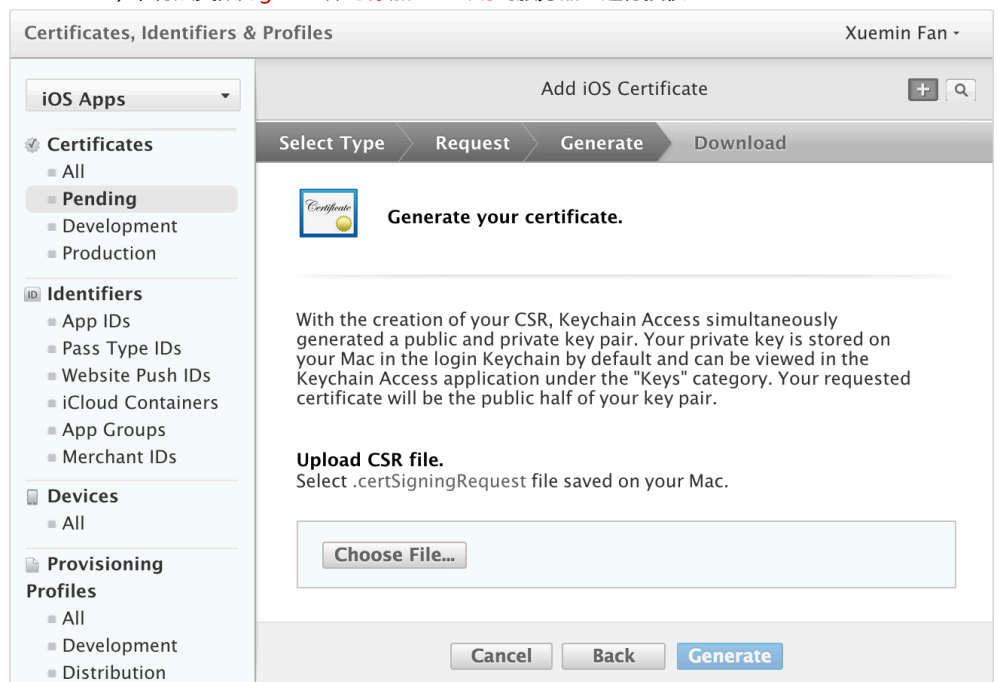
Keychain Access|Keys 中将新增一对非对称密钥对 Public/Private **Key Pair** ( This signing identity consists of a public-private key pair that Apple issues ) 。同时, keychain 将生成一个包含开发者身份信息和公钥的**CSR** ( Certificate Signing Request ) 文件——CertificateSigningRequest.certSigningRequest。



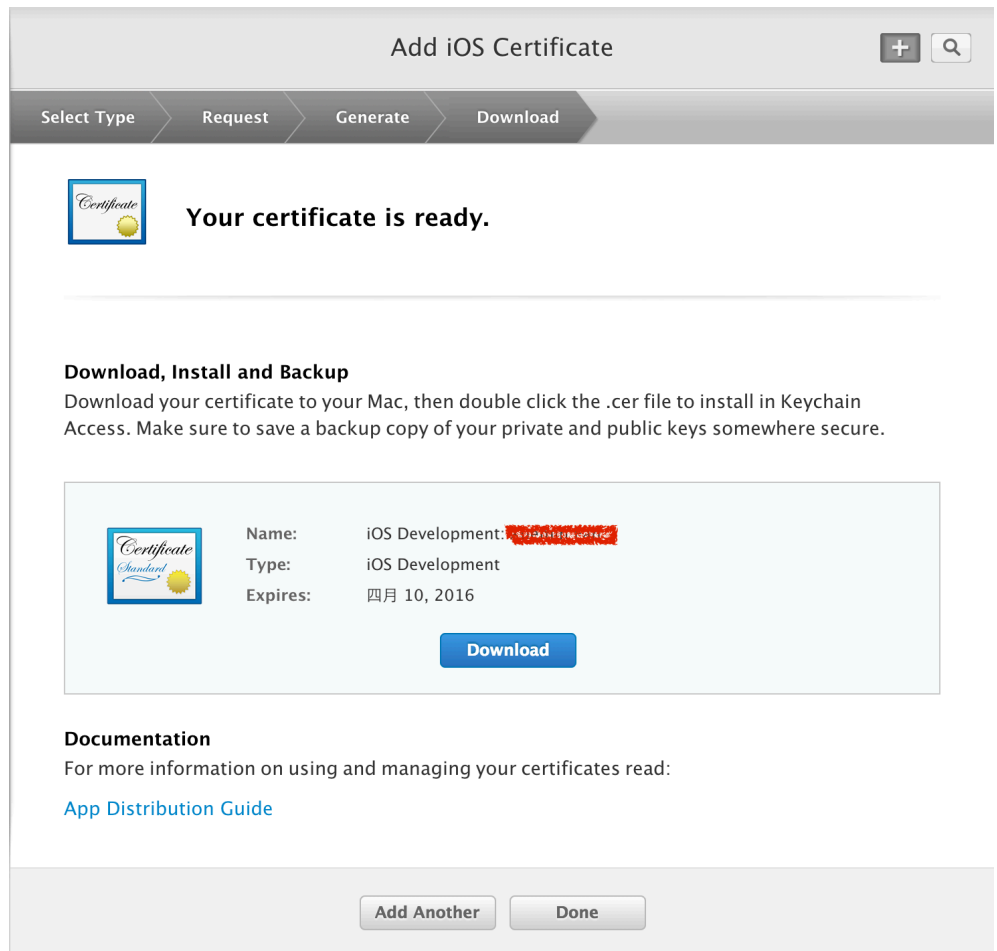
**私钥 private key** 始终保存在 Mac OS 的 Keychain Access 中，用于签名（CodeSign）本机对外发布的 App；**公钥 public key** 一般随证书（随Provisioning Profile，随App）散布出去，对 App 签名进行校验认证。用户必须妥善保存本地 Keychain 中的 private key，以防假冒。

- Keep a secure backup of your public-private key pair. If the private key is lost, you' ll have to create an **entirely new** identity to sign code.
- Worse, if someone else has your private key, that person may be able to **impersonate** you.

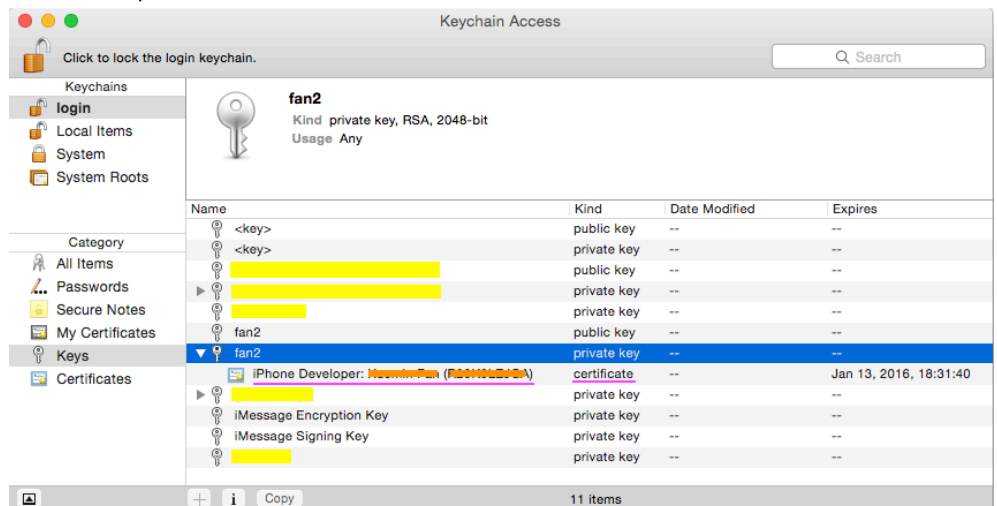
在 Apple 开发网站上传包含公钥的 CSR 文件作为换取证书的凭证（Upload CSR file to generate your certificate），有点类似为github账号添加SSH公钥到服务器上进行授权。



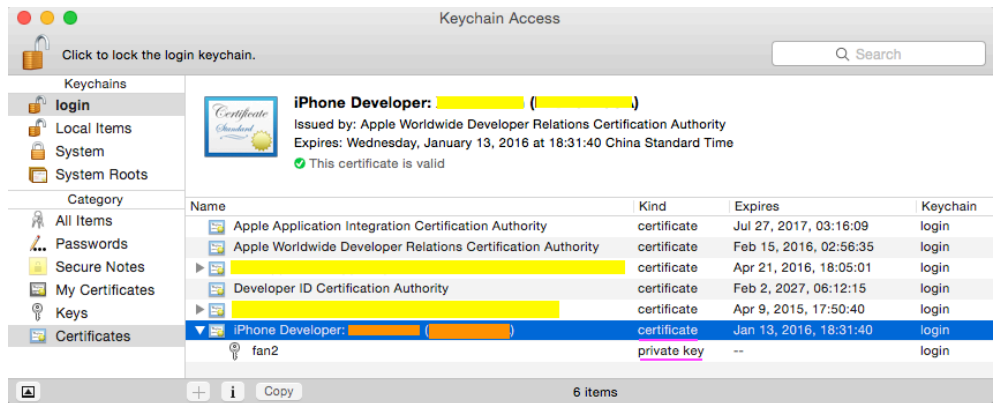
Apple 证书颁发机构 WWDRCA(*Apple Worldwide Developer Relations Certification Authority*) 将使用其 private key 对 CSR 中的 public key 和一些身份信息进行加密签名生成**数字证书**（ios\_development.cer）并记录在案（Apple Member Center）。



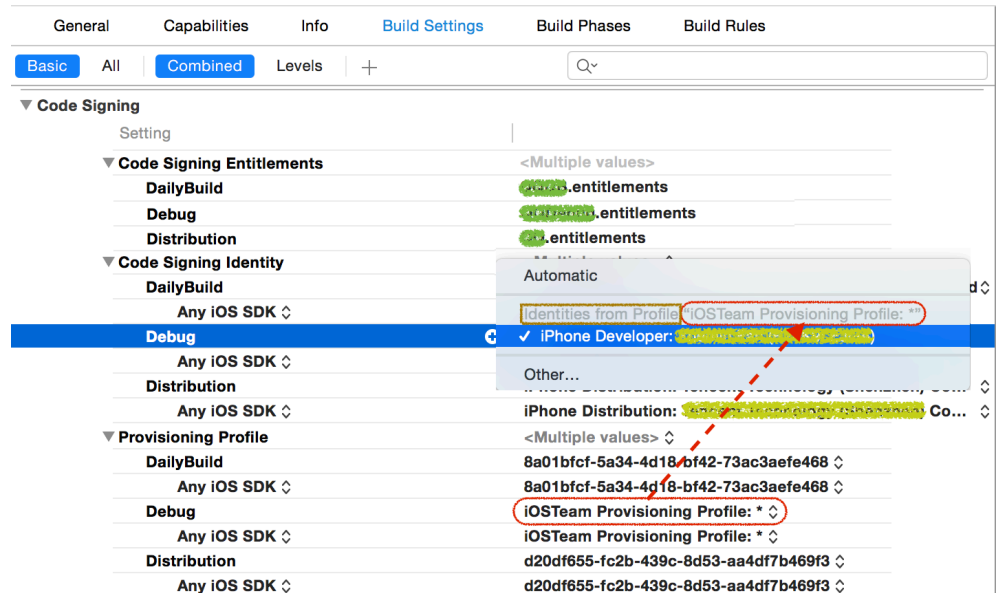
从 Apple Member Center 网站下载证书到 Mac 上双击即可安装（当然也可在 Xcode 中添加开发账号自动同步证书和[生成]配置文件）。证书安装成功后，在 KeychainAccess|Keys 中展开创建 CSR 时生成的 Key Pair 中的私钥前面的箭头，可以查看到包含其对应公钥的证书（Your requested certificate will be the public half of the key pair.）；在 Keychain Access|Certificates 中展开安装的证书（ios\_development.cer）前面的箭头，可以看到其对应的私钥。







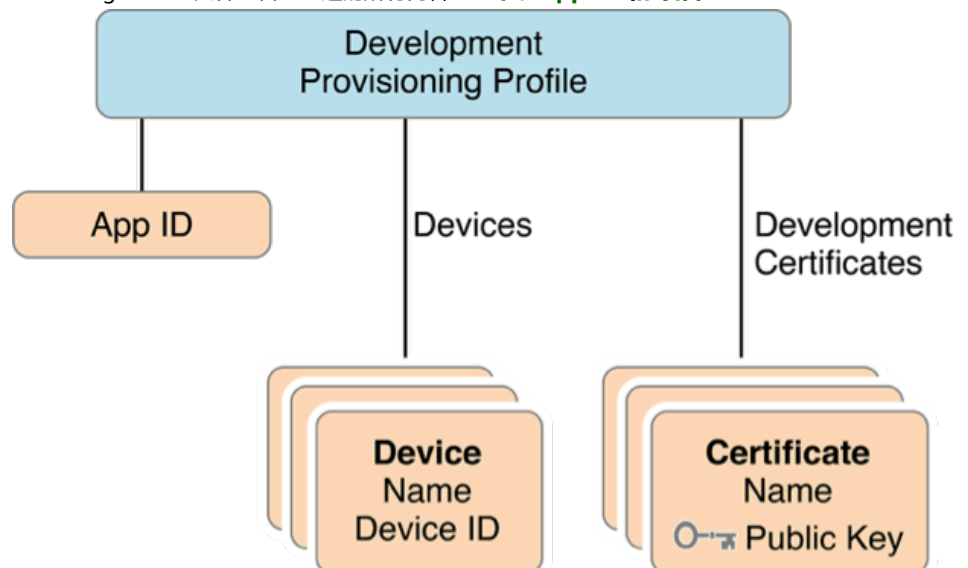
Certificate 应被配置到【Xcode Target|Build Settings|Code Signing|Code Signing Identity】下，下拉选择 Identities from Profile "..."（一般先配置 Provisioning Profile）。以下是 Xcode 配置示例：



#### 四.供应配置文件 (Provisioning Profiles)

##### 1.Provisioning Profile 的概念

Provisioning Profile 文件包含了上述的所有内容：证书、App ID 和设备 ID。



一个 Provisioning Profile 对应一个 Explicit App ID 或 Wildcard App ID（一组相同 Prefix/Seed 的 App IDs）。在网站上手动创建一个 Provisioning Profile 时，需要依次指定 App ID（单选）、证书（Certificates，可多选）和设备（Devices，可多选）。用户可在网站上删除（Delete）已注册的 Provisioning Profiles。

Provisioning Profile 决定 Xcode 用哪个证书（公钥）/私钥组合（Key Pair/Signing Identity）来签署应用程序（Signing Product），并将在应用程序打包时嵌入到 .ipa 包里。安装应用程序时，Provisioning Profile 文件被拷贝到 iOS 设备中，运行该 iOS App 的设备通过它来认证安装的程序。如果要打包到真机上运行一个APP，一般要经历以下三步：

- **首先**，需要指明它的 App ID，并且验证 Bundle ID 是否与其一致；
- **其次**，需要证书对应的私钥来进行签名，用于标识这个 APP 是合法、安全、完整的；
- **然后**，如果是真机调试，需要确认这台设备是否授权运行该 APP。

Provisioning Profile 把这些信息全部打包在一起，方便我们在调试和发布程序打包时使用。这样，只要在不同的情况下选择不同的 Provisioning Profile 文件就可以了。

Provisioning Profile 也分为 Development 和 Distribution 两类，有效期同 Certificate 一样。Distribution 版本的 Provisioning Profile 主要用于提交 App Store 审核，其中不指定开发测试的 Devices（0，unlimited）。App ID 为 Wildcard App ID（\*）。App Store 审核通过上架后，允许所有 iOS 设备（Deployment Target）上安装运行该App。

Xcode 将全部供应配置文件（包括用户手动下载安装的和 Xcode 自动创建的 Team Provisioning Profile）放在目录 `~/Library/MobileDevice/Provisioning Profiles` 下。

## 2.Provisioning Profile 的构成

以下为典型供应配置文件 \*.mobileprovision 的构成简析：

- (1) **Name**：该mobileprovision的文件名。
- (2) **UUID**：该mobileprovision文件的真实文件名。
- (3) **TeamName**：Apple ID账号名。
- (4) **TeamIdentifier**：Team Identity。
- (5) **AppIDName**：explicit/wildcard App ID name（ApplicationIdentifierPrefix）。
- (6) **ApplicationIdentifierPrefix**：完整App ID的前缀（TeamIdentifier.\*）。
- (7) **DeveloperCertificates**：包含了可以为使用该配置文件应用签名的所有证书<data><array>。

证书是基于 Base64 编码，符合 PEM(PrivacyEnhanced Mail, RFC 1848) 格式的，可使用 OpenSSL 来处理（opensslx509 -text -in file.pem）。

从 DeveloperCertificates 提取 <data></data> 之间的内容到文件 cert.cer（cert.perm）：

```
-----BEGIN CERTIFICATE-----
```

将<data></data>之间的内容拷贝至此

```
-----END CERTIFICATE-----`
```

Mac 下右键 QuickLook 查看 cert.cer（cert.perm），在 Keychain Access 中右键 Get Info 查看对应证书 ios\_development.cer，正常情况（公钥 KeyPair 配对）应吻合；Windows 下没有足够信息（WWDRCA.cer），无法验证该证书。

如果你用了一个不在这个列表中的证书进行签名，无论这个证书是否有效，这个应用都将 CodeSign Fail。

- (8) **Entitlements** 键<key>对应的<dict>：

**keychain-access-groups**：\$(AppIdentifierPrefix)，参见**Code Signing Entitlements**(\*entitlements)。

每个应用程序都有一个可以用于安全保存一些如密码、认证等信息的 **keychain**，一般而言自己的程序只能访问自己的 keychain。通过对应用签名时的一些设置，还可以利用keychain的方式实现同一开发者签证（就是相同 bundle seed）下的不同应用之间共享信息的操作。比如你有一个开发者帐户，并开发了两个不同的应用A和B，然后通过A和B的 keychain access group 这个东西指定共用的访问分组，就可以实现共享此 keychain 中的内容。

**application-identifier**：带前缀的全名，例如\$(AppIdentifierPrefix)com.apple.garageband。

**com.apple.security.application-groups**：App Group ID（group.com.apple），参见**Code Signing Entitlements**(\*entitlements)。

**com.apple.developer.team-identifier**：同Team Identifier。

- (9) **ProvisionedDevices**：该mobileprovision授权的开发设备的UDID <array>。

Provisioning Profile被配置到【XcodeTarget|Build Settings|Code Signing|Provisioning Profile】下，然后在**Code Signing Identity**下拉可选择Identities from Profile "...”（即Provisioning Profile中包



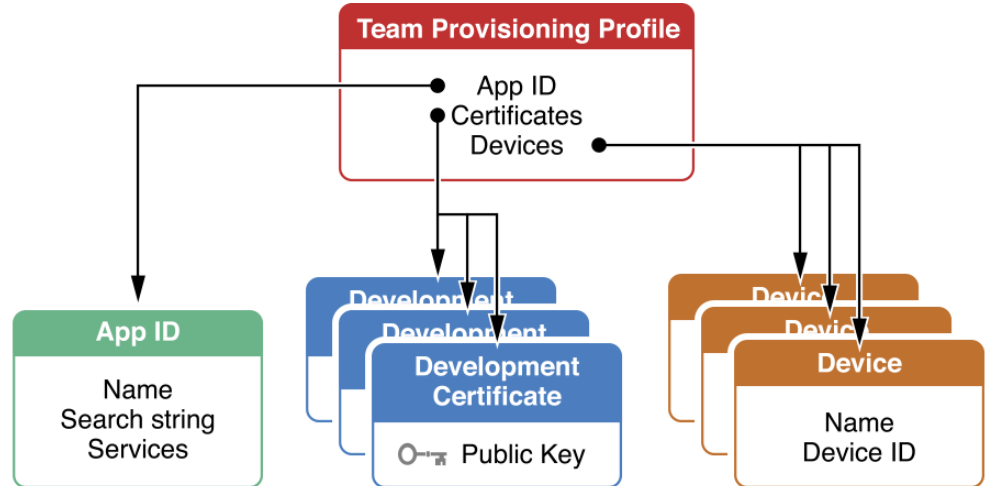
含的Certificates)。

## 五.开发组供应配置文件 ( Team Provisioning Profiles )

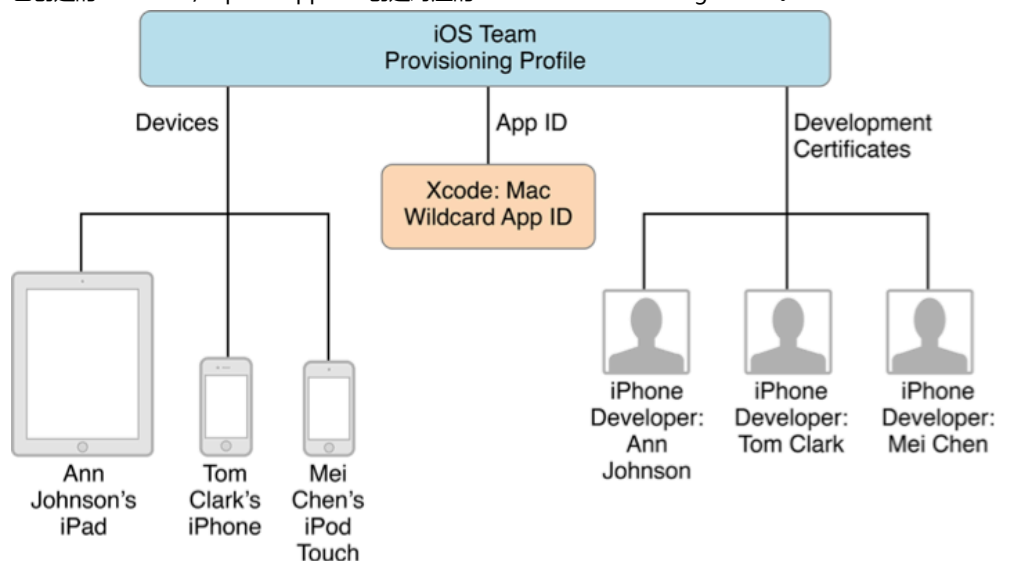
### 1.Team Provisioning Profile的概念

每个 Apple 开发者账号都对应一个唯一的 **Team ID** , Xcode3.2.3 预发布版本中加入了 Team Provisioning Profile 这项新功能。

在 Xcode 中添加 Apple Developer Account 时, 它将与 Apple Member Center 后台勾选**自动生成** iOS Team Provisioning Profile ( **Managed by Xcode** ) 。



Team Provisioning Profile 包含一个为 **Xcode iOS Wildcard App ID(\*)** 生成的 **iOS Team Provisioning Profile:\*** ( 匹配所有应用程序 ), 账户里所有的 Development Certificates 和 Devices 都可以使用它在这个 team 注册的所有设备上调试应用程序 ( 不管bundle identifier是什么 )。同时, 它还会为开发者自己创建的 Wildcard/Explicit App IDs 创建对应的 iOS Team Provisioning Profile。



### 2.Team Provisioning Profile 生成/更新时机

- Add an Apple ID account to Xcode
- Fix issue "No Provisioning Profiles with a valid signing identity" in Xcode
- Assign Your App to a Team in Xcode project settings of General|Identity
- Register new device on the apple development website or Xcode detected new device connected

利用 Xcode 生成和管理的 iOS Team Provisioning Profile 来进行开发非常方便, 可以不需要上网站手动生成下载 Provisioning Profile。

Team Provisioning Profile 同 Provisioning Profile, 只不过是 Xcode 自动生成的, 也**被配置到** **【XcodeTarget|Build Settings|Code Signing|Provisioning Profile】** 下, 同时需要在

【XcodeTarget|General|Identity】下指定 Team 账号 ID。

## 六.App Group ( ID )

### 1.App Group 的概念

WWDC14 除了发布了 OS X v10.10 和 swift 外，iOS 8.0 也开始变得更加开放了。说到开放，当然要数应用扩展 ( App Extension ) 了。顾名思义，应用扩展允许开发者扩展应用的自定义功能和内容，能够让用户在使用其他应用程序时使用该项功能，从而实现各个应用程序间的功能和资源共享。可以将扩展理解为一个轻量级 ( nimble and lightweight ) 的分身。

扩展和其 Containing App 各自拥有自己的沙盒，虽然扩展以插件形式内嵌在 Containing App 中，但是它们是独立的二进制包，不可以互访彼此的沙盒。为了实现 Containing App 与扩展的数据共享，苹果在 iOS 8 中引入了一个新的概念——App Group，它主要用于同一 Group 下的 APP 实现数据共享，具体来说是通过创建使用以 App Group ID 标识的共享资源区——App Group **Container** 来实现的。App Group ID 同 App ID 一样，一般不超过255个ASCII字符。用户可在网站上编辑 Explicit App IDs，将其纳入 App Group ( Assignment )；也可删除 ( Delete ) 已注册的App Group ( ID )。

### 2.App Group 的配置

Containing App 与 Extension 的 Explicit App ID 必须 Assign 到**同一 App Group** 下才能实现数据共享，并且 Containing App 与 Extension 的 App ID 命名必须符合规范：

1. 置于同一App Group 下的 App IDs 必须是**唯一**的 ( Explicit , not Wildcard )
2. Extension App ID 以 Containing App ID 为前缀 ( Prefix/Seed )

例如 Garageband 这个 App ID 为 “com.apple.garageband”，则支持从语音备忘录导入到 Garageband 应用的插件的 App ID 可能形如 “com.apple.garageband.**extImportRecording**”。

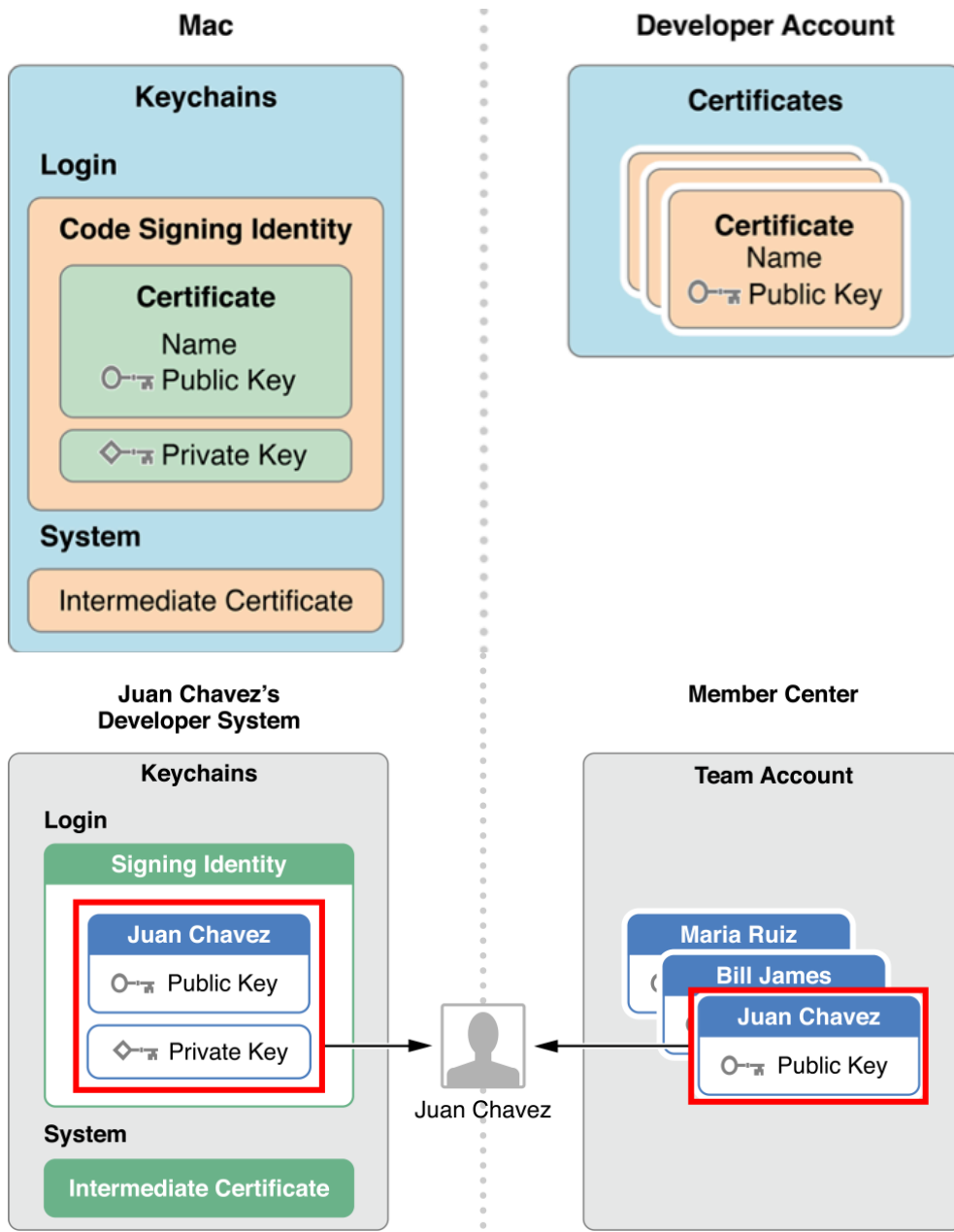
App(ex)	App Group ID	Provisioning Profile	
		Code Signing Identity ( Certificate Key Pair )	App ID ( bundle identifier )
<b>GarageBand</b>	置于同一分组： group.com.apple	( 1 ) 共用同一证书：ios_development.cer	com.apple.garageband
<b>GarageBand扩展插件</b>		( 2 ) 共用证书Key Pair中的Private Key进行CodeSign	com.apple.garageband. <b>extImportRecording</b>

关于Provisioning Profile，可以使用自己手动生成的，也可以使用 Xcode 自动生成的 Team Provisioning Profile。

App Group 会被配置到【Xcode Target|Build Settings|Code Signing|Code Signing Entitlements】文件 ( \*.entitlements ) 的键com.apple.security.application-groups下，不影响 Provisioning Profile 生成流程。

## 七.证书与签名 ( Certificate& Signature )

### 1.Code Signing Identity



Xcode 中配置的 Code Signing Identity ( entitlements、certificate ) 必须与 Provisioning Profile 匹配，并且配置的 Certificate 必须在本机 Keychain Access 中存在对应 Public / Private Key Pair，否则编译会报错。

Xcode 所在的 Mac 设备（系统）使用 CA 证书（WWDRCA.cer）来判断 Code Signing Identity 中 Certificate 的合法性：

- 若用 WWDRCA 公钥能成功解密出证书并得到公钥（Public Key）和内容摘要（Signature），证明此证书确乃 AppleWWDRCA 颁布，即证书来源可信；
- 再对证书本身使用哈希算法计算摘要，若与上一步得到的摘要一致，则证明此证书未被篡改过，即证书完整。

## 2.Code Signing

每个证书（其实是公钥）对应 Key Pair 中的**私钥**会被用来对内容（executable code，resources such as images and nib files aren't signed）进行数字**签名**（CodeSign）——使用哈希**算法**生成内容**摘要**（digest）。

Xcode 使用指定证书配套的私钥进行签名时需要授权，选择【始终允许】后，以后使用该私钥进行签名便不会再弹出授权确认窗口。

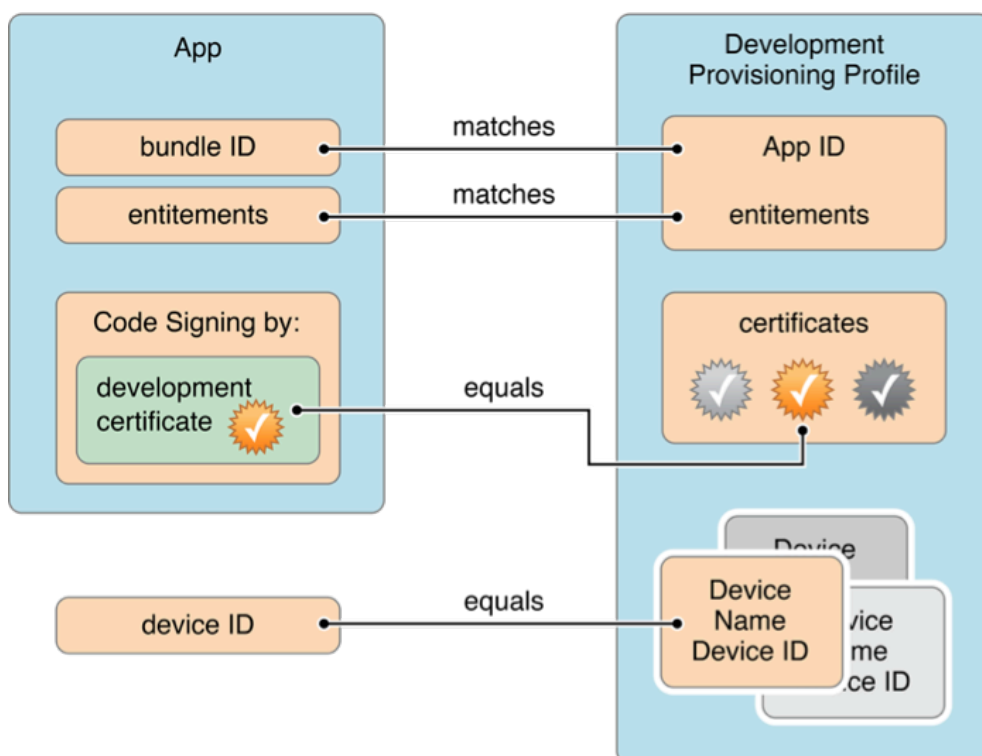


### 3. Verify Code Signature with Certificate

上面已经提到，公钥被包含在数字证书里，数字证书又被包含在描述文件(Provisioning File)中，描述文件在应用被安装的时候会被拷贝到 iOS 设备中。

第一步，App 在 Mac / iOS 真机上启动时，需要对配置的 bundle ID、entitlements 和 certificate 与 Provisioning Profile 进行匹配校验：

Your app launches on a device if:



第二步，iOS/Mac 真机上的 ios\_development.cer 被 AppleWWDRCA.cer 中的 public key 解密校验合法后，获取每个开发证书中可信任的公钥对 App 的可靠性和完整性进行校验。

**iOS/Mac 设备（系统）使用 App Provisioning Profile（Code Signing Identity）中的开发证书来判断App的合法性：**

- 若用证书公钥能成功解密出 App（executable bundle）的内容摘要（\_CodeSignature），证明此 App 确乃认证开发者发布，即来源可信；
- 再对 App（executable bundle）本身使用哈希算法计算摘要，若与上一步得到的摘要一致，则证明此 App 未被篡改过，即内容完整。

### 小结：

- 基于 Provisioning Profile 校验了 CodeSign 的一致性；
- 基于 Certificate 校验 App 的可靠性和完整性；
- 启动时，真机的 device ID ( UUID ) 必须在 Provisioning Profile 的 **ProvisionedDevices** 授权之列。
- 无论是 Xcode 对 APP 进行签名打包还是真机运行 APP 进行校验，都使用了基于证书体系的**非对称加密机制**。


## 八.在多台机器上共享开发账户/证书

### 1.Xcode 导出开发者账号(\*.developerprofile) 或 PKCS12 文件(\*.p12)

进入 Xcode Preferences|Accounts：

- 选中 Apple IDs 列表中对 Account 的 Email，点击+之后的|Export Accounts，可导出包含 account/code signing identity/provisioning profiles 信息的 **\*.developerprofile** ( Exporting a Developer Profile ) 文件供其他机器上的 Xcode 开发使用 ( Import 该 Account )。

选中右下列表中某行 Account Name 条目|ViewDetails，可以查看 Signing Identities 和 Provisioning Profiles。

- 选中欲导出的 Signing Identity 条目，点击栏底+之后的|Export，必须输入密码，并需授权 export key "privateKey" from keychain，将导出 **Certificates.p12**。

点击左下角的刷新按钮可从 Member Center 同步该账号下所有的 Provisioning Profile 到本地。

选中右击列表中某个 Provisioning Profile 可以【Show in Finder】到

[~/Library/MobileDevice/Provisioning\ Profiles]目录，其中 Provisioning Profile 的真实名称为 \$(UUID).mobileprovision，名如"2488109f-ff65-442e-9774-fd50bd6bc827.mobileprovision"，其中 <key>Name</key>中为 Xcode 中看到的描述性别名。

### 2.Keychain Access 导出 PKCS12 文件(\*.p12)

在 Keychain Access|Certificates 中选中欲导出的 certificate 或其下 private key，右键 Export 或者通过菜单 File|Export Items 导出 **Certificates.p12**——PKCS12 file holds the private key and certificate。

其他 Mac 机器上双击 Certificates.p12 ( 如有密码需输入密码 ) 即可安装该共享证书。有了共享证书之后，在开发者网站上将欲调试的 iOS 设备注册到该开发者账号名下，并下载对应证书授权了 iOS 调试设备的 Provisioning Profile 文件，方可在 iOS 真机设备上开发调试。

## 九.证书配置常见错误

### 1.no such provisioning profile was found

Xcode Target|General|Identity Team下提示"**Your build settings specify a provisioning profile with the UUID "xxx",however, no such provisioning profile was found.**"

Xcode Target|BuildSettings|Code Signing|当前配置的指定UDID的provisioning profile在本地不存在，此时需要更改Provisioning Profile。必要时手动去网站下载或重新生成Provisioning Profile或直接在Xcode中Fix issue予以解决 ( 可能自动生成iOS Team ProvisioningProfile ) ！

### 2.No identities from profile

Build Settings|CodeSigning的Provisioning Profile中选择了本地安装的provisioning profile之后，Code Signing Identity中下拉提示**No identities from profile "..."** or No identities from keychain。Xcode配置指定UDID的provisioning profile中的DeveloperCertificates在本地KeyChain中不存在 ( **No identities are available** ) 或不一致 ( KeyPair中的Private Key丢失 )，此时需去网站检查 ProvisioningProfile中的App ID-Certificate-Device配置是否正确。如果是别人提供的共享账号 ( \*.developerprofile ) 或共享证书(\*.p12)，请确保导出了对应Key Pair中的Private Key。必要时也直接在Xcode中Fix issue予以解决 ( 可能自动生成iOS Team ProvisioningProfile )。

### 3.Code Signing Entitlements file do not match profile

"Invalid application-identifier Entitlement" or "Code Signing Entitlements file do not match those specified in your provisioning profile.(0xE8008016)."



( 1 ) 检查对应版本 ( Debug ) 指定的\*.entitlements文件中的 "Keychain Access Groups" 键值是否与ProvisioningProfile中的Entitlements项相吻合 ( 后者一般为前者的Prefix/Seed )。

( 2 ) 也可以将Build Settings|Code Signing的Provisioning Profile中对应版本 ( Debug ) 的Entitlements置空。

#### 4.The app ID cannot be registered to your development team


▼ Signing


☒ Automatically manage signing  
Xcode will create and update profiles, app IDs, and certificates.

Team  (Personal Team) 

Provisioning Profile Xcode Managed Profile

Signing Certificate iOS Developer

Status  Failed to create provisioning profile.  
The app ID "com.[redacted].fan2" cannot be registered to your development team. Change your bundle identifier to a unique string to try again.

 No profiles for 'com.[redacted].fan2' were found  
Xcode couldn't find a provisioning profile matching 'com.[redacted].fan2'.



出现该问题通常是 app ID 冲突，即该 app ID 已经有人注册过，此时可以按照提示换一个 app ID或基于已有app ID添加后缀——Change your bundle identifier to a unique string to try again.

#### 5.The 'In-App Purchase' feature is only available to users enrolled in Apple Developer Program

只有开发者账号才能真机调试 'In-App Purchase' 特性，所以需要在工程配置 (Capabilities) 中禁用普通 Apple ID不支持的特性。


▼ Signing


☒ Automatically manage signing  
Xcode will create and update profiles, app IDs, and certificates.

Team  (Personal Team) 

Provisioning Profile Xcode Managed Profile

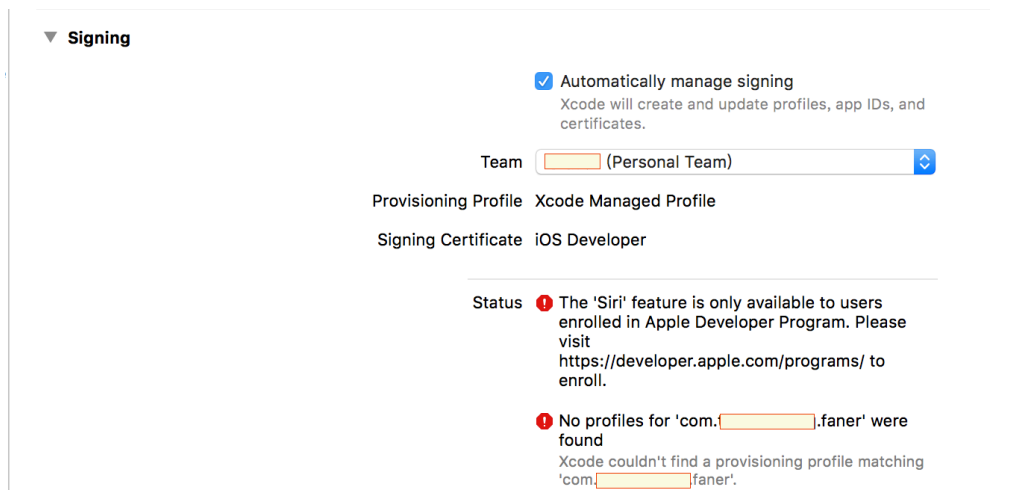
Signing Certificate iOS Developer

Status  The 'In-App Purchase' feature is only available to users enrolled in Apple Developer Program. Please visit <https://developer.apple.com/programs/> to enroll.

 No profiles for 'com.[redacted].faner' were found  
Xcode couldn't find a provisioning profile matching 'com.[redacted].faner'.

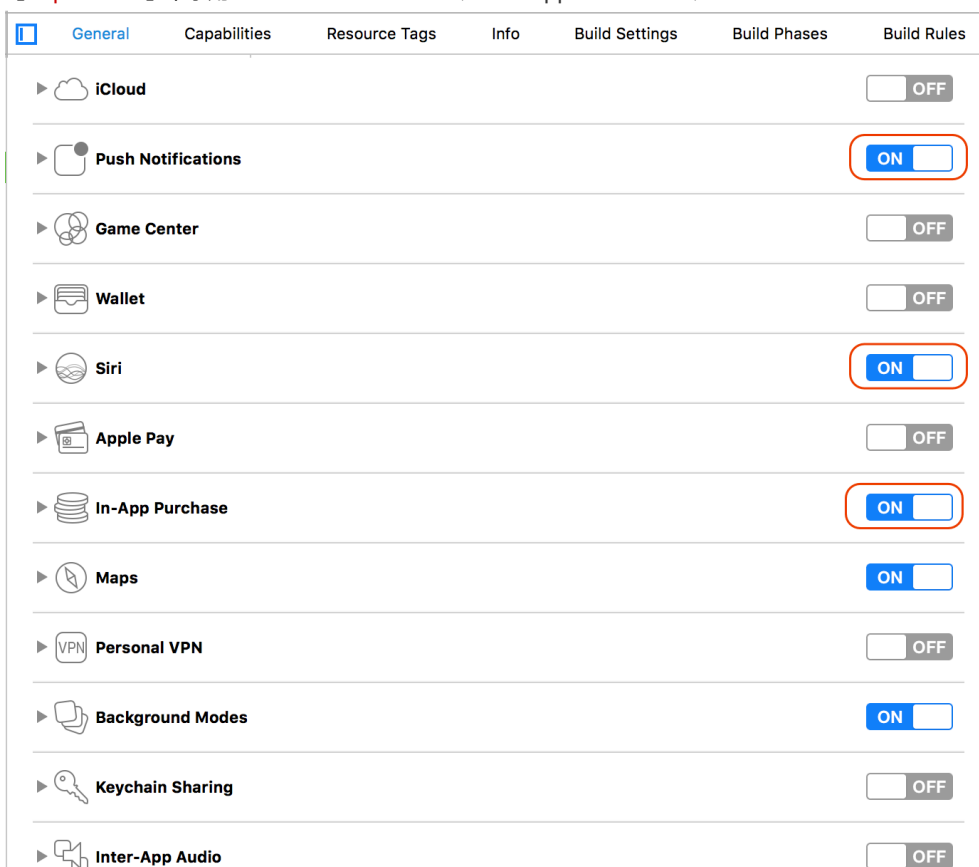
同样，在 Xcode 8 中，也只有开发者账号才能真机调试The 'Siri' feature，否则也会报错 "The 'Siri' feature is only available to users enrolled in Apple Developer Program." 而无法完成签名认证。





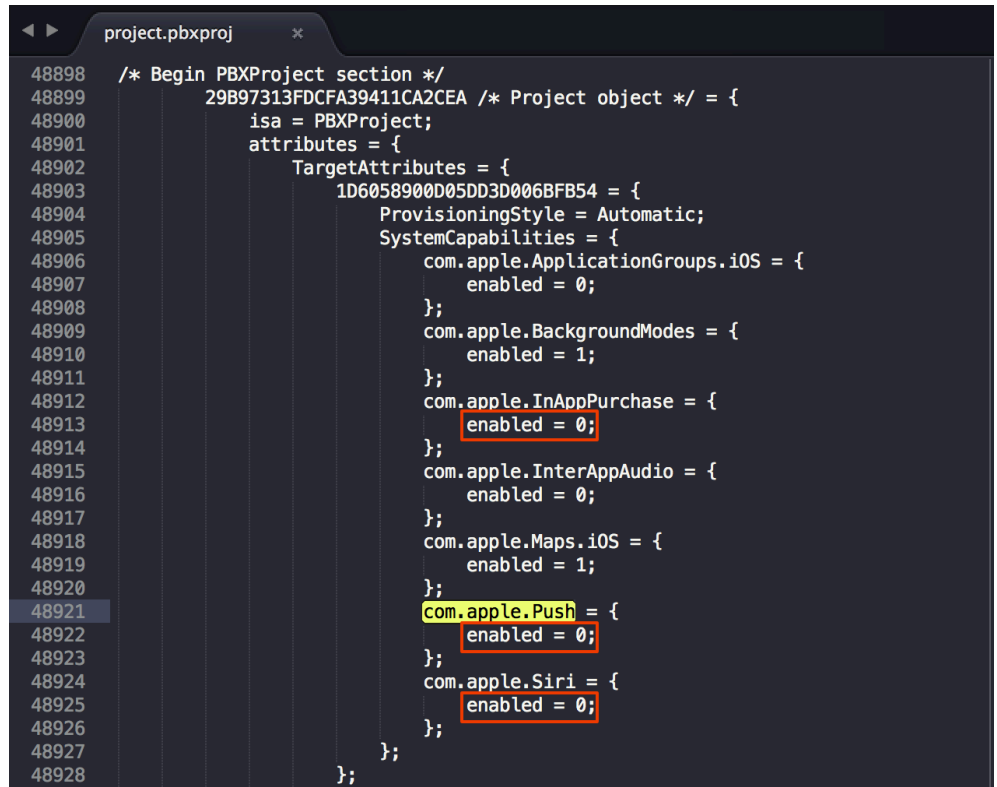
#### 解决方法：

在Xcode的 Project Navigator 中点击\*.xcodproj，在右侧的 Targets 下选择点击目标 Target，在【Capabilities】中禁用“Push Notifications”、“In-App Purchase”、“Siri”。



如果找不到这些配置项，可尝试直接修改项目配置文件中相应feature的配置开关，步骤如下：

关闭 Xcode 正在打开的 Project，在 Finder 中右键项目配置文件 \*.xcodproj 显示包内容，使用文本编辑器（例如 Sublime Text）打开 project.pbxproj 文件，搜索“SystemCapabilities”，依次找到“com.apple.Push”、“com.apple.InAppPurchase”、“com.apple.Siri”将其 enabled 键值从1修改为0，关闭退出使用 Xcode 重新打开该项目。



6.Xcode配置反应有时候不那么及时，可刷新、重置相关配置项开关（若有）或重启Xcode试试。

## 十. Xcode7+ 免证书真机调试

在 Xcode 7 中，苹果改变了自己在许可权限上的策略：

1. 此前 Xcode 只开放给注册开发者下载，现在 Xcode 7 改变了这种惯有的做法，无需注册开发者账号，仅使用普通的Apple ID就能下载和上手体验。
2. 此前开发者需每年支付99美元的费用成为注册开发者才能在 iPhone/iPad 真机上运行调试APP，苹果新的开发者计划则放宽要求，无需购买，只要你感兴趣同样可以在设备上测试app。  
——Developers would be able to test apps on devices without a paid Apple developer account in Xcode 7.

所谓“免证书”真机调试，并不是真的不需要证书，Xcode真机调试原有的证书配置体系仍在——All iOS, tvOS, and watchOS apps **must be** code signed and provisioned to launch on a device. 所以，上文啰嗦几千字还是有点用的。

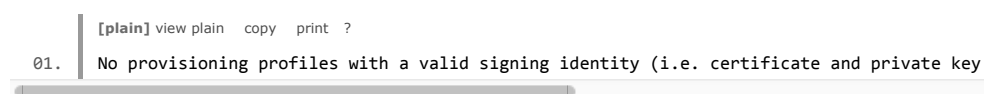
自 Xcode7 开始，原来基于付费开发者账号及自助生成证书及配置文件的繁琐过程被苹果简化，Xcode将针对任何普通账号自动为联调真机生成所需相关的证书及配置文件。当你打算向 App Store 提交发布应用，才需要付费。

**第一步：**进入 Xcode Preferences|Accounts，添加自己的 Apple ID 账号。

**第二步：**Build Settings|Code Signing 下的 Provisioning Profile 选择 **Automatic**，Code Signing Identity 选择 Automatic 下的**iOS Developer**。

**第三步：**General 配置 Bundle identifier，Team 下拉选择苹果Member Center自动为你的账号生成的 Personal Team ID。

自己的账号在调试公司或其他第三方APP代码时，若填写 Bundle identifier 为他人账号注册的 APP ID（例如苹果相机应用 **com.apple.camera**），会报错：

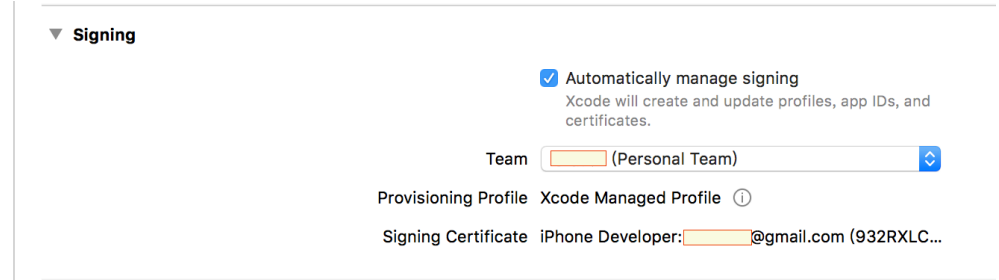


即使编译通过了，可能运行时APP自身与服务器校验也可能会报签名错误，肿么办？？？

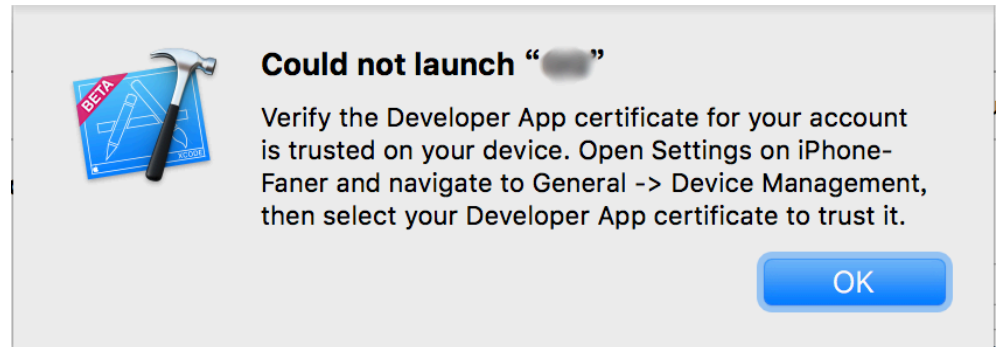
Her skill：此时，可以在他人原有App ID基础上添加后缀（例如**com.apple.camera.extension**），配

置成应用的衍生插件（相当于置于同一 **App Group** 下）就可以快乐的玩耍了。

解决了所有的开发配置问题，Xcode 8 Automatically manage signing 配置成功应该是这样子的：



如果启动APP时，Xcode报错 **“process launch failed: Security”** 或iPhone报错【**不受信任的开发者**】。



此时需要到iPhone通用配置中的**描述文件**（最新系统中可能叫**设备管理**）中，在描述文件（开发商应用）中选择对应的**描述文件**（你的Apple ID）点击 **信任** 或 **验证** 即可。



OK , All Done !

#### 参考：

《iPhone真机调试应用程序》《iOS Developer：真机测试》《Xcode5 & iOS 7 及以下版本免证书真机调试记录》

《iOS Development--Certificates, Provisioning Profiles》《关于Certificate、Provisioning Profile、App ID的介绍及其关系》

《数字签名和数字证书》《iOS keyChain 研究》

《苹果开发者账号那些事儿》《iOS關於Provisioning Profiles這些事》

《iOS Code Signing 学习笔记》《代码签名探析/Inside Code Signing》

《iOS Code Signing: 解惑/iOS Code Signing: Under The Hood》

《iOS行货自动打包》《解决Xcode无法生成Archive的问题》《iOS程序完成后如何生成ipa进行真机测试》

《发布iOS应用程序(Application Loader)》《iOS发布遇到的一些问题》

《Xcode打包ipa包》《iOS程序生成ipa进行真机测试》

顶

47

踩

0

上一篇 iPhone屏幕尺寸、分辨率及适配

下一篇 iOS8扩展插件开发配置

## 我的同类文章

### iOS (8)

- |  |  |
|--|--|
| • NSDictionary&NSMutable... 2015-07-26 阅读 5918 | • NSString&NSMutableStrin... 2015-07-19 阅读 2304    |
| • NSArray&NSMutableArray... 2015-05-24 阅读 7414 | • MultipeerConnectivity.fram... 2015-02-03 阅读 6472 |
| • iPhone/iOS开启个人热点的... 2015-01-21 阅读 14395     | • iOS8扩展插件开发配置 2015-01-14 阅读 14744                 |
| • iPhone屏幕尺寸、分辨率... 2014-12-26                 | • resizableImageWithCapIns... 2014-12-25 阅读 1433   |
| 阅读 236174                                      |  |

## 参考知识库



### iOS知识库

1679 关注 | 1277 收录



### Docker知识库

4524 关注 | 200 收录



### Swift知识库

3209 关注 | 464 收录



### 算法与数据结构知识库

8939 关注 | 2263 收录

## 猜你在找

精通iOS移动开发(Xcode7&Swift2;) 初识Xcode7.0

iOS开发从入门到精通(Xcode8和Swift3)

iOS移动开发从入门到精通(Xcode7 & Swift2)

iOS8开发技术(Swift版): iOS基础知识

精通iOS移动开发(Xcode7&Swift2): 常用控件的使用

	<b>1950.00/件</b> 供应:GPX80 不锈钢 288芯 ODF光纤交接		<b>60.00/件</b> 16E室外光纤分纤箱 SMC 楼道箱 光纤入户		<b>0.04/个</b> 光纤保护管 生产制造 工厂
---	--	---	--	---	-----------------------------------

广告

查看评论

30楼 [kharis](#) 3天前 09:44发表



新建项目的时候报了2个error, 1.Failed to create provisioning profile.

There are no devices registered in your account on the developer website. Plug in and select a device to have Xcode register it.

2.No profiles for 'cyw.Calculator' were found

Xcode couldn't find a provisioning profile matching 'cyw.Calculator'.

不知道楼主有没有遇到过? 求教。

29楼 [AiN\\_Hy](#) 2016-09-16 11:49发表



感谢楼主分享

28楼 [avi9111](#) 2016-09-14 14:18发表



再次路过 看看

27楼 [hsintung](#) 2016-06-25 10:07发表



写的非常详细, 各个关系描述的很清楚, 大神~~

26楼 [a350927646](#) 2016-06-20 21:53发表



感谢楼主, 非常详细

25楼 [zoro1101](#) 2016-06-20 15:32发表



extension大法好!

24楼 [ITCandy-5](#) 2016-04-27 15:41发表



赞, 解析的超级清楚明白, 谢楼主分享

23楼 [xhhusthdq](#) 2016-04-14 15:53发表



写得太好了, 终于看明白了

22楼 [罗先雄](#) 2016-04-05 20:46发表



Code Sign error: No code signing identities found: No valid, non-expired code signing identities (i.e. certificate and private key pair) were found.

这个问题怎么解决?

Re: [龙渊之源](#) 2016-04-21 18:21发表



回复罗先雄: 你说的那个问题你还找到解决方法了啊? 我好奇啊我所有的证书和对应的ID都设置好了, 可是就是没办法解决问题

Re: [罗先雄](#) 2016-08-01 17:55发表



回复龙渊之源: 已经解决了, 是签名证书的问题, 重写生成了就没问题了。谢谢你。

Re: [龙渊之源](#) 2016-04-21 18:20发表



回复罗先雄: 你说的那个问题你还找到解决方法了啊? 我好奇啊我所有的证书和对应的ID都设置好了, 可是就是没办法解决问题

21楼 [LeoMobileDeveloper](#) 2016-03-29 12:19发表

写的真不错





20楼 [3h3k](#) 2016-03-28 14:47发表



非常清楚明白的详尽解释。是我看过的少有的说得非常清楚的文章。

19楼 [avi9111](#) 2016-03-28 12:31发表



在网页没有找到创provisioning profile的地方，只能选择前几个，xcode activie的选项，或者.\*

18楼 [冰翔不败传说](#) 2016-03-18 13:58发表



谢谢楼主分享

17楼 [四毛五](#) 2016-03-16 14:18发表



写的真挺好呀...

16楼 [avi9111](#) 2016-03-07 10:50发表



强帖留名

15楼 [花花猪](#) 2016-02-26 17:28发表



国人都能拿出博主这样的态度写博客，何愁水平参差不齐啊！！怒赞！！向楼主看齐，认真写每一篇博客~~

14楼 [sno\\_guo](#) 2016-01-14 12:36发表



写的挺好，猛一看,还有点看不明白，不过实际测试后，还是可以的。

直接上科普知识，有点生猛，哈哈。

谢谢！

13楼 [ft3807154](#) 2016-01-07 17:29发表



UDID 会变啊。我没吃在开发者中心加上这个UDID，然后生成新的证书，再看这个ID 就变了

12楼 [RL\\_LEEE](#) 2015-12-31 14:21发表



写的太好了，科普扫盲文，已收藏

11楼 [at9009](#) 2015-10-10 14:28发表



博主能不能转

10楼 [阳光在冬季](#) 2015-09-14 14:47发表



好文~多谢

9楼 [洋小葱](#) 2015-08-27 18:12发表



写的很好很详细 我是小白看完之后就知道了我的问题所在 3Q

8楼 [yyk\\_ios](#) 2015-08-11 17:10发表



赞！！果断收藏！！

7楼 [小雄哥](#) 2015-07-23 14:19发表



不错 好想转载

6楼 [greenboy1](#) 2015-06-15 17:34发表



真心不错 我先去看看

5楼 [巾令页](#) 2015-06-15 15:18发表



写得很好。目前看到最新，最详细的。

4楼 [jiazhuangaini](#) 2015-04-29 14:38发表



写的非常棒，推荐大家阅读，3Q

3楼 [WaterGJ](#) 2015-04-28 14:06发表



mark ,thanks

2楼 [Kamto](#) 2015-04-24 10:35发表



mark

1楼 [跑在今日](#) 2015-04-18 11:47发表



这一篇文章还比较接近最新的

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](mailto:webmaster@csdn.net)   400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

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