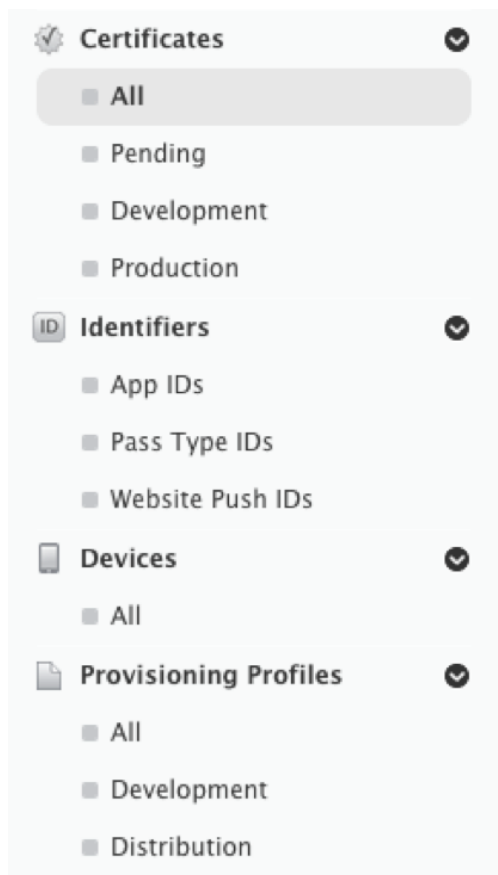


刚接触iOS开发的人难免会对苹果的各种证书、配置文件等不甚了解，可能你按照网上的教程一步一步的成功申请了真机调试，但是还是对其中的缘由一知半解。这篇文章就对Certificate、Provisioning Profile等做个总结。

1.概念介绍

如果你拥有一个开发者账户的话，在iOS Dev Center打开Certificates, Identifiers & Profiles，你就可以看到如下的列表：



Profile Portal改版有一段时间了，改版之后的结构比以前更清晰明了，易于理解和管理。

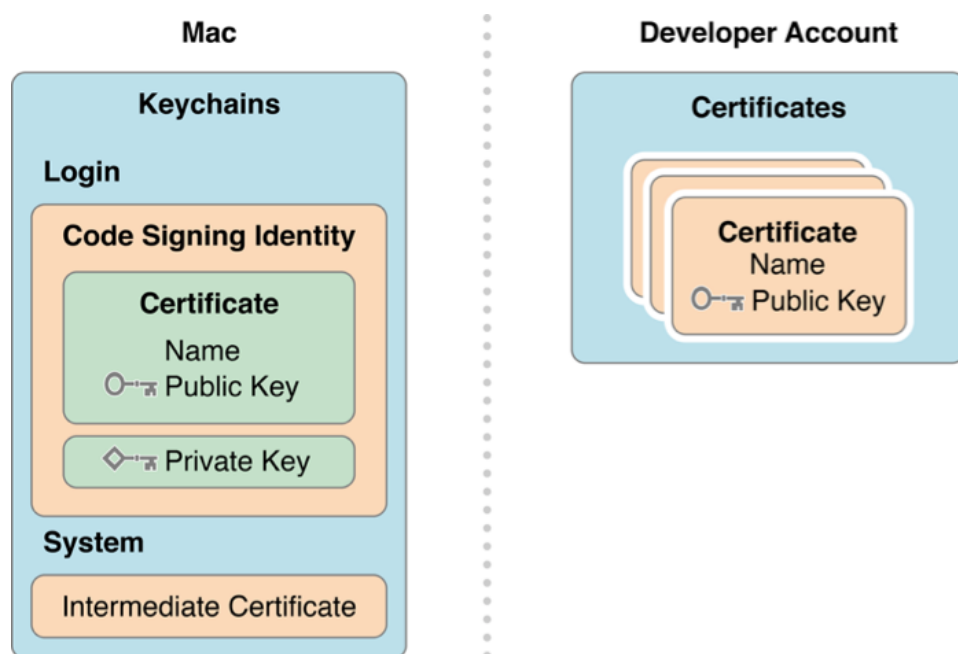
上面的列表就包含了开发、调试和发布iOS应用程序所需的所有内容：Certificates、Identifiers、Devices、Provisioning Profiles。下面将一一解释这几个东东。

Certificate

证书是用来给应用程序签名的，只有经过签名的应用程序才能保证他的来源是可信任的，并且代码是完整的，未经修改的。在Xcode Build Setting的Code Signing Identity中，你可以设置用于为代码签名的证书。

众所周知，我们申请一个Certificate之前，需要先申请一个Certificate Signing Request (CSR) 文件，而在这个过程中实际上是生成了一对公钥和私钥，保存在你Mac的Keychain中。代码签名正是使用这种基于非对称密钥的加密方式，用私钥进行签名，用公钥进行验证。如下图所示，在你Mac的keychain的login中存储着相关的公钥和私钥，而证书中包含了公钥。你只能用私钥来进行签名，所以如果没有了私钥，就意

意味着你不能进行签名了，所以就无法使用这个证书了，此时你只能revoke之前的证书再申请一个。因此在申请完证书时，最好导出并保存好你的私钥。当你想与其他人或设备共享证书时，把私钥传给它就可以了。私钥保存在你的Mac中，而苹果生成的Certificate中包含了公钥。当你用自己的私钥对代码签名后，苹果就可以用证书中的公钥来进行验证，确保是你对代码进行了签名，而不是别人冒充你，同时也确保代码的完整性等。



证书主要分为两类：Development和Production，Development证书用来开发和调试应用程序，Production主要用来分发应用程序（根据证书种类有不同作用），下面是证书的分类信息：（括号内为证书有效期）

（注：不同类型的开发者账户所能创建的证书种类不同，关于开发者账户的对比和InHouse证书相关的内容，请见我的另一篇文章）

- Development
 - App Development (1年)：用来开发和真机调试应用程序。
 - Push Development (1年)：用来调试Apple Push Notification
- Production
 - In-House and Ad Hoc (3年)：用来发布In-House和AdHoc的应用程序。
 - App Store：用来发布提交App Store的应用程序。
 - MDM CSR
 - Push Production (1年)：用来在发布版本中使用Apple Push Notification。
 - Pass Type ID Certificate
 - Website Push ID Certificate

有一些类型的证书我没有使用过，所以也不了解具体的作用。

App ID

App ID用于标识一个或者一组App，App ID应该是和Xcode中的Bundle ID是一致的或者匹配的。App ID主要有以下两种：

- Explicit App ID：唯一的App ID，这种App ID用于唯一标识一个应用程序，例如com.ABC.demo1，标识Bundle ID为com.ABC.demo1的程序。
- Wildcard App ID：通配符App ID，用于标识一组应用程序。例如*可以表示所有应用程序，而com.ABC.*可以表示以com.ABC开头的所有应用程序。

每创建一个App ID，我们都可以设置该App ID所使用的APP Services，也就是其所使用的额外服务。每种额外服务都有着不同的要求，例如，如果要使用Apple Push Notification Services，则必须是一个explicit App ID，以便能唯一标识一个应用程序。下面是目前所有可选的服务和相应的配置要求。

	Create Explicit App ID	Enable App ID	Set Entitlements	Edit Info.plist	Configure iTunes Connect
APNs	✓	✓			
Game Center	✓	✓			✓
iCloud		✓	✓		
In-App Purchase	✓	✓			✓
Data protection		✓			
Newsstand				✓	✓
Passbook		✓			
Routing apps				✓	✓

如果你的App使用上述的任何一种service，就要按照要求去配置。

Device

Device最简单了，就是iOS设备。Devices中包含了该账户中所有可用于开发和测试的设备。 每台设备使用UDID来唯一标识。

每个账户中的设备数量限制是100个。Disable 一台设备也不会增加名额，只能在membership year 开始的时候才能通过删除设备来增加名额。

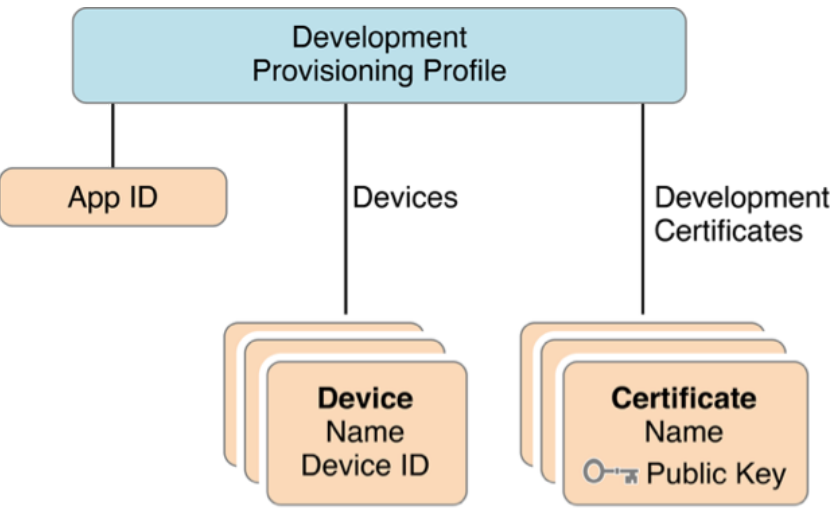
关于设备数量的问题，详见[这篇文章](#)。

Provisioning Profile

一个Provisioning Profile文件包含了上述的所有内容：证书、App ID、设备。

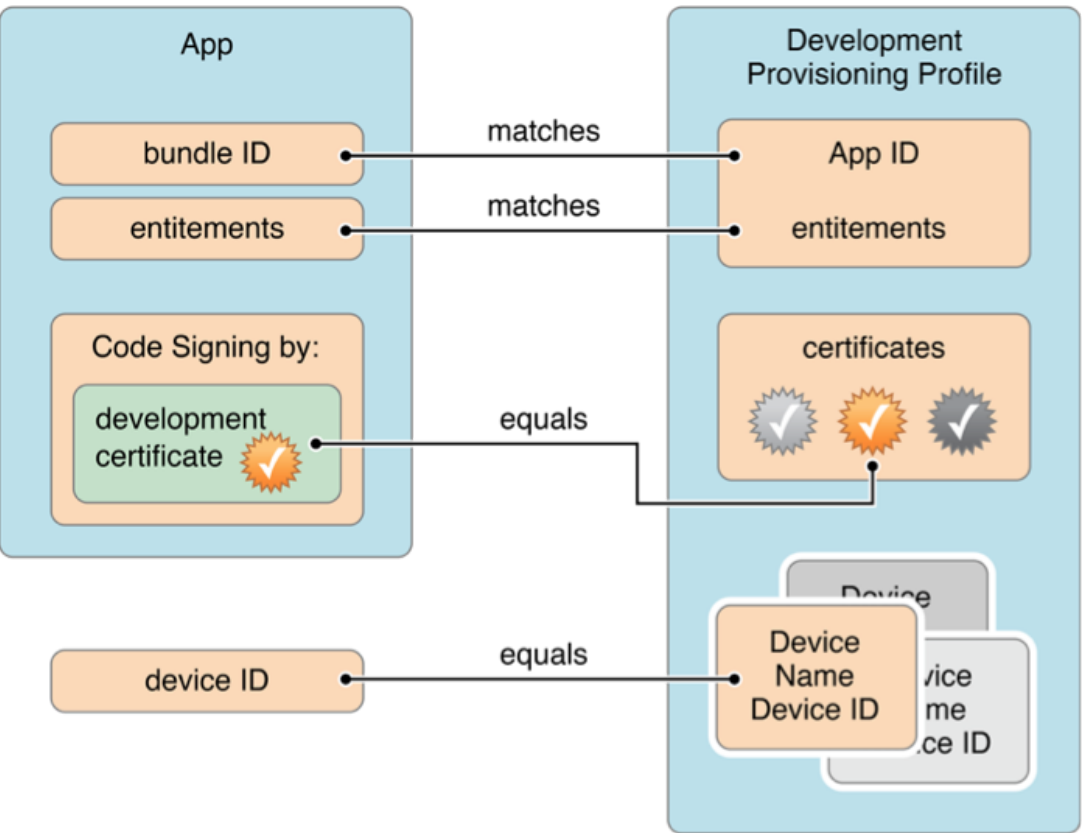
试想一下，如果我们要打包或者在真机上运行一个应用程序，我们首先需要证书来进行签名，用来标识这个应用程序是合法的、安全的、完整的等等；然后需要指明它的App ID，并且验证Bundle ID是否与其一致；再次，如果是真机调试，需要确认这台设备能否用来运行程序。而Provisioning Profile就把这些信息全部打包在一起，方便我们在调试和发布程序打包时使用，这样我们只要在不同的情况下选择不同的profile文件就可以了。而且这个Provisioning Profile文件会在打包时嵌入.ipa的包里。

例如，如下图所示，一个用于Development的Provisioning Profile中包含了该Provisioning Profile对应的App ID，可使用的证书和设备。这意味着使用这个Provisioning Profile打包程序必须拥有相应的证书，并且是将App ID对应的程序运行到Devices中包含的设备上去。



如上所述，在一台设备上运行应用程序的过程如下：

Your app launches on a device if:



与证书一样，Provisioning Profile也分为Development和Distribution两种：

（注：前面提到不同账户类型所能创建的证书种类不同，显然Profile文件的种类是和你所能创建的证书种类相关的）

- Development (1年)
- Distribution (1年)
 - In House
 - Ad Hoc
 - App Store

In House 与Ad Hoc的不同之处在于：In House没有设备数量限制，而Ad Hoc是用来测试用的，Ad Hoc的包只能运行在该账户内已登记的可用设备上，显然是有最多100个设备的数量限制。所以这两种Provisioning Profile文件的区别就在于其中的设备限制不一样而已，而他们所使用的Certificate是相同的。

2.开发/发布流程

了解了上面的概念，再来看开发及发布流程就非常简单了，而且相信你不用看教程也能一步步完成所有的操作了。

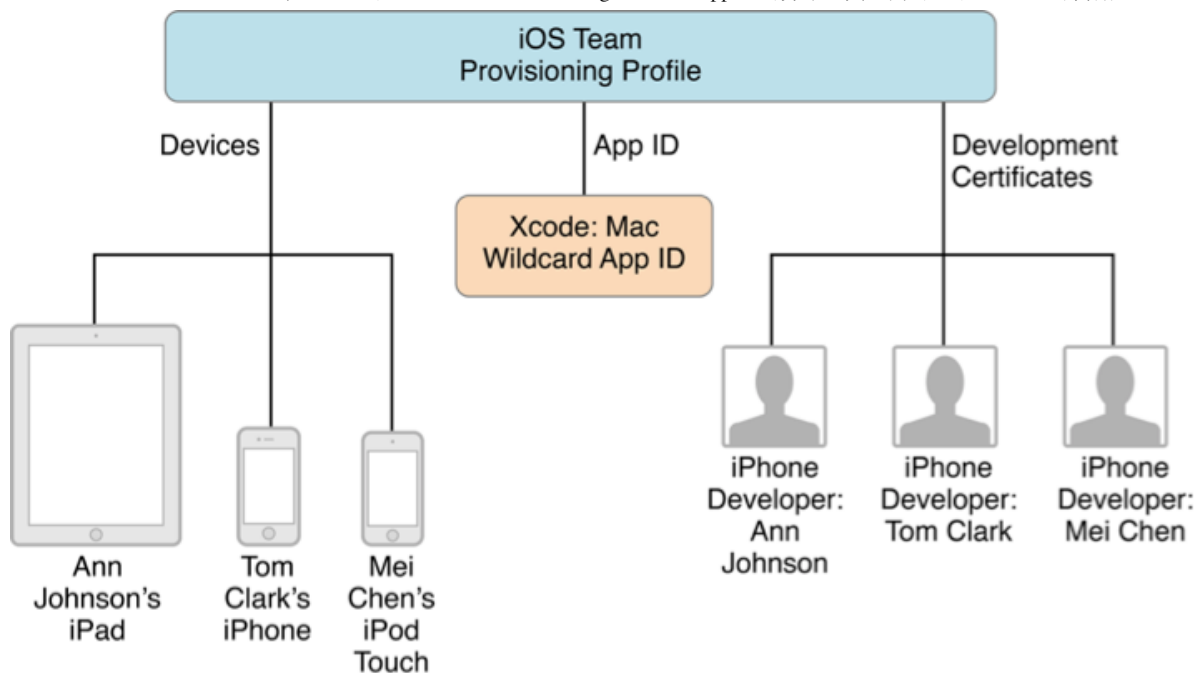
开发/真机调试流程

根据上面的介绍，可以知道进行Development主要有以下几个步骤：

- 申请证书
- 加入设备
- 生成Provisioning Profile
- 设置Xcode Code Sign Identifier

事实上第三步通常是不需要的，因为我们通常都是用Xcode生成和管理的iOS Team Provisioning Profile来进行开发，因为它非常方便，所以不需要自己手动生成Provisioning Profile。

iOS Team Provisioning Profile是第一次使用Xcode添加设备时，Xcode自动生成的，它包含了Xcode生成的一个Wildcard App ID（*，匹配所有应用程序），账户里面所有的Devices和所有Development Certificates，如下图所示。因此，team中的所有成员都可以使用这个iOS Team Provisioning Profile在team中的所有设备上调试所有的应用程序。并且当有新设备添加进来时，Xcode会更新这个文件。



发布流程

网上有很多关于发布App Store的流程，我就不赘述了，不过根据上面的概念介绍，不管是App Store、In-House还是Ad-Hoc，打包流程都是差不多的，都包括了以下几个关键步骤：

- 创建发布证书
- 创建App ID
- 创建对应的Provisioning Profile文件
- 设备Bundle ID和App ID一致
- 设置Xcode Code Sign Identifier，选择合适的Profile和证书进行签名，打包

以上就是对证书、Provisioning Profile、App ID等的介绍，下一篇文章会介绍以下In-House证书相关的内容。