对接重签名步骤

**目的：** 由于使用到动态库技术，与证书挂钩，而动态库签名证书必须和游戏所使用的证书保持一致才能正常编译运行，所以厂商在运行SDKDEMO或者对接SDK之前必须对SDKDEMO中包含的动态库以及其他相关文件做重新签名，换成厂商自己的证书，具体见“步骤”。

**步骤：**

1. **重签名动态库二进制文件PPAppPlatformKitDylib.pp**

文件所在位置：

PPAppPlatformKit.bundle –> PPAppPlatformKitDylib.framework –>PPAppPlatformKitDylib.pp

查看动态库文件的签名信息命令：codesign –vv –d PPAppPlatformKitDylib.pp文件路径

重签名命令：codesign -f –s ‘证书常用名称’‘PPAppPlatformKitDylib.pp文件路径’



图1

命令如图2：

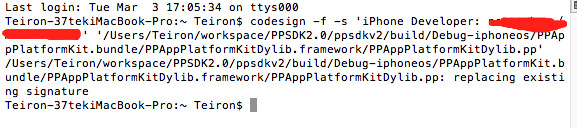


图2

1. **重新计算动态库二进制文件hash值，并写入配置文件中**

文件所在位置：

PPAppPlatformKit.bundle –> PPAppPlatformKitDylib.framework –>PPSource.bundle –> sdk\_dylib\_Info.plist ，可以查看到一个Hash值，如图3所示：

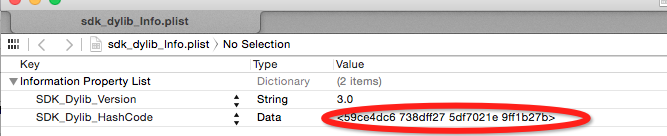


图3

使用GeneralHashCode工具（SDK包里）计算出Hash值方法：

如图4输入命令：GeneralHashCode工具文件路径 动态库文件路径

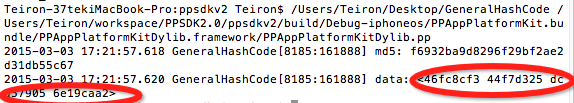


图4

1. **重签名动态库PPAppPlatformKitDylib.framework**

文件所在位置：

PPAppPlatformKit.bundle –> PPAppPlatformKitDylib.framework

查看动态库framework的签名信息命令：

codesign –vv –d PPAppPlatformKitDylib.framework文件路径

重签名命令：

codesign -f –s ‘证书常用名称’

‘PPAppPlatformKitDylib.framework文件路径’

1. **在重新查看证书正确之后可在真机上编译运行，不支持模拟器**