**华为安全类授权开放SDK开发指导书**

版权所有 © 华为技术有限公司2018保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

附件3-版权声明页图和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

**修改记录**

|  |  |  |
| --- | --- | --- |
| **日期** | **版本** | **修改描述** |
| 2015-11-2 | v1.0.0 | 初稿 |
| 2016-6-20 | v1.0.1 | 增加支持平台范围说明；  增加开发证书有效期说明；  增加申请DeviceIds区间说明；  增加Android v2签名方式说明；  增加了相关FAQ说明； |
| 2016-10-8 | V1.1.0 | 增加32个接口扩大支持平台范围； |
| 2018-3-2 | V2.0 | 1. 其他管理类更新至EMUI8.0版本 2. 另外增加FAQ内容 3. 更新开发者联盟证书申请步骤 4. 删除无用管理权限 |
| 2018-9-15 | V9.0.0.1 | 1. 删除不完整的功能接口介绍 2. 新增关于Android V1 V2签名介绍 3. 刷新部分证书申请流程截图 4. 增加开发证书和商用证书申请流程 5. 增加FAQ |

目录

[1 概述 7](#_Toc529263328)

[1.1 权限使用声明、免责声明与软件许可协议（强制实现） 7](#_Toc529263329)

[1.1.1 权限使用声明 7](#_Toc529263330)

[1.2 架构 8](#_Toc529263331)

[1.3 交付组件 9](#_Toc529263332)

[2 华为证书介绍 9](#_Toc529263333)

[2.1 华为证书的基本格式 9](#_Toc529263334)

[2.2 华为证书的分类 10](#_Toc529263335)

[3 华为安全类授权开放SDK使用指南 10](#_Toc529263336)

[3.1 开发者应用修改 10](#_Toc529263337)

[3.1.1 AndroidManifest.xml修改 10](#_Toc529263338)

[3.1.2导入hwsdk-mdm-openapi-9.0.0.1.jar 10](#_Toc529263339)

[3.2 华为证书申请流程 11](#_Toc529263340)

[3.2.1开发证书的申请流程 11](#_Toc529263341)

[3.2.2 V1商用证书的申请流程 12](#_Toc529263342)

[3.2.3 V2商用证书的申请流程 12](#_Toc529263343)

[3.3 申请华为证书 13](#_Toc529263344)

[3.4 将华为证书打包到开发者应用中 17](#_Toc529263345)

[4 华为安全类授权开放SDK使用详解 17](#_Toc529263346)

[4.1 获取当前hwMdm版本号 18](#_Toc529263347)

[4.2 设备管理类 18](#_Toc529263348)

[4.2.1目前支持的管理类 18](#_Toc529263349)

[4.2.2使用方法 18](#_Toc529263350)

[5 避免混淆打包 19](#_Toc529263351)

[6 FAQ 19](#_Toc529263352)

[6.1 在其他非华为手机上是否可以使用华为安全类授权开放SDK？ 19](#_Toc529263353)

[6.2 如果华为开发者联盟网站的界面，与本文档中界面不一致，是什么原因？ 19](#_Toc529263354)

[6.3 如何获取设备的IMEI号、WIFI MAC地址和MEID？ 20](#_Toc529263355)

[6.4 如果手机无IMEI号，如何申请开发证书？ 20](#_Toc529263356)

[6.5 证书申请周期一般有多长时间？ 20](#_Toc529263357)

[6.6 如果通过MDM应用禁用了USB文件传输功能，忘记了开启，并且把MDM应用删除了，该怎么办？ 20](#_Toc529263358)

[6.7 如何判断APK是否使用了Android v2签名？ 20](#_Toc529263359)

[6.8 如果已经申请了商用证书，后来APK版本更新了，还需要再次申请商用证书吗 20](#_Toc529263360)

[6.9 最常见报错信息 21](#_Toc529263361)

[6.10 推荐使用签名工具介绍—apkSigner 23](#_Toc529263362)

[6.11 如果v2签名的apk打包进去证书重新做签名，apk的hash值不就变了么？ 23](#_Toc529263363)

[6.12 V1签名V2签名的apk文件名的区别 23](#_Toc529263364)

[6.13 商用证书和开发证书在开发阶段的区别 23](#_Toc529263365)

[6.14 addPersistentApp和setSuperWhiteListForHwSystemManger两个保活接口有什么区别。 23](#_Toc529263366)

# 概述

华为安全类授权开放SDK，包括两部分API，分别是设备管理类和应用权限管理类。这两部分API为在华为手机上使用的、经过华为证书授权的应用，提供了深度管理华为手机的功能。

## 权限使用声明、免责声明与软件许可协议（强制实现）

**注意：**任何使用“华为安全类授权开放SDK”的应用，都**必须**提供如下界面：权限使用声明。否则，会导致应用无法通过华为证书审核。

其中，“权限使用声明” 界面，既可以按照华为提供的（如下图或Sample例子）方式和内容进行显示，也可以由开发者自己拟定方式和内容，但是内容**必须**包含调用SDK接口对应的业务，不能缺失。

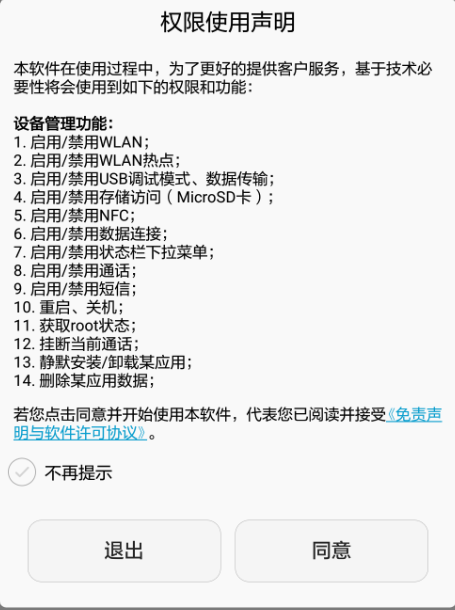
如：开发者使用到SDK中的WLAN和USB相关接口，在提示内容中必须包含此两个业务说明，给用户明确提示，具体内容华为不做限制。

另外，应用开发者需要根据实际使用到的接口和权限进行提示。

### 1.1.1 权限使用声明

权限使用声明对话框，列出了“华为安全类授权开放SDK”所提供具体功能的简要描述。开发者可以根据自己实际使用到功能和权限，对当前用户进行提醒。

具体的内容请查看文档sample例子。



**注意：**此对话框的显示规则如下，

* 用户点击应用后，此对话框即弹出；
* 只要当用户选中“不再提醒”，同时点击“同意”后，此对话框就不用再显示；

但是如果应用有了重大功能更新（开发者自行判断），此对话框还要再次弹出；

* 请根据实际用到的功能进行显示，未用到功能不要显示；
* 在用户点击“同意”之前，应用不能调用“华为安全类授权开放SDK”的任何接口；

“免责声明与软件许可协议”，由开发者根据具体业务场景，来声明对应的免责协议，此协议与华为公司无关。

## 架构

下图展示了华为安全类授权开放SDK的架构。



架构组成：

* 应用：应用调用华为安全类授权开放SDK。
* hwsdk-mdm-openapi-9.0.0.1.jar：华为安全类授权开放SDK的jar包，具体见[1.3交付组件](#_交付组件)

在华为手机上，开发者应用通过hwsdk-mdm-openapi-9.0.0.1.的接口调用，能直接完成功能设置；

而在非华为手机上，开发者应用仅仅调用hwsdk-mdm-openapi-9.0.0.1提供的接口，此时是空实现，不起作用；

## 交付组件

组件：

* hwsdk-mdm-openapi-9.0.0.1.jar
* DevEco.exe

用于将华为证书打包到开发者应用的工具

# 华为证书介绍

## 华为证书的基本格式

华为证书，是保证华为安全类授权开放SDK能够正常使用的前提，开发者调用华为SDK时的权限校验，都是基于华为证书的授权和校验。开发者需要按照如下“3.2”节申请华为证书。

目前华为证书的格式如下：

华为证书是一个以.cer后缀结尾的文件，内容包含了

* DeveloperKey：开发者签名，与android.content.pm.Signature.toCharsString()的输出相同；
* PackageName：开发者应用包名；
* Permissions：开发者想要申请的权限；
* DeviceIds：开发者应用被限制使用的设备，仅限开发证书有效，商用证书时无效；
* ValidPeriod：华为证书有效期，如果有效期过期，则应用在华为手机上安装失败，

目前开发证书的有效期为6个月；

* ApkHash：开发者应用HASH值，在商用证书中必选；
* Signature：前6项SHA 256 HASH值的签名；

## 华为证书的分类

华为证书目前分为两类：开发证书和商用证书。

开发证书是专门为开发者提供的，是一种调试适配阶段的证书，DeviceIds字段会被严格限制在一定数量设备上，ApkHash字段不做限制。

而商用证书，是作为开发者应用使用开发证书，功能调试完成之后，处于正式商用阶段的证书。一般DeviceIds字段不会做出限制，ApkHash字段必选。

**注意：**开发者在使用开发证书，调试完成之后，一定要申请商用证书，才能正式发布。

另外，开发者在申请华为开发证书和商用证书之前，需要按照“HW安全类授权开放SDK功能介绍及展示文档.pdf”中的相关流程，与华为商务或市场部门签订相关协议，否则华为证书无法申请成功。

# 华为安全类授权开放SDK使用指南

## 开发者应用修改

## 3.1.1 AndroidManifest.xml修改

在开发者的AndroidManifest.xml中，增加

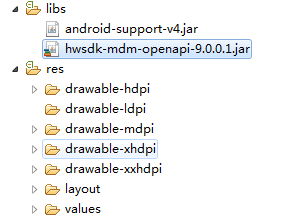
“<uses-permission android:name="com.huawei.permission.sec.MDM" />”权限；

## 3.1.2导入hwsdk-mdm-openapi-9.0.0.1.jar

开发者需要将hwsdk-mdm-openapi-9.0.0.1.jar包集成到开发者应用中，建议按照如下步骤操作：

1. 如果开发者使用“Android Developer Tools”开发工具（以v21.0.1-543035为例），

需要将hwsdk-mdm-openapi-9.0.0.1.jar放到开发者工程的libs文件夹下；



1. 如果开发者使用“Android Studio”开发工具（以1.4.1版本为例），只需要将hwsdk-mdm-openapi-9.0.0.1.jar拷贝到开发者工程的app/libs中，Android Studio工具会自动将此jar包导入到工程中。

## 华为证书申请流程

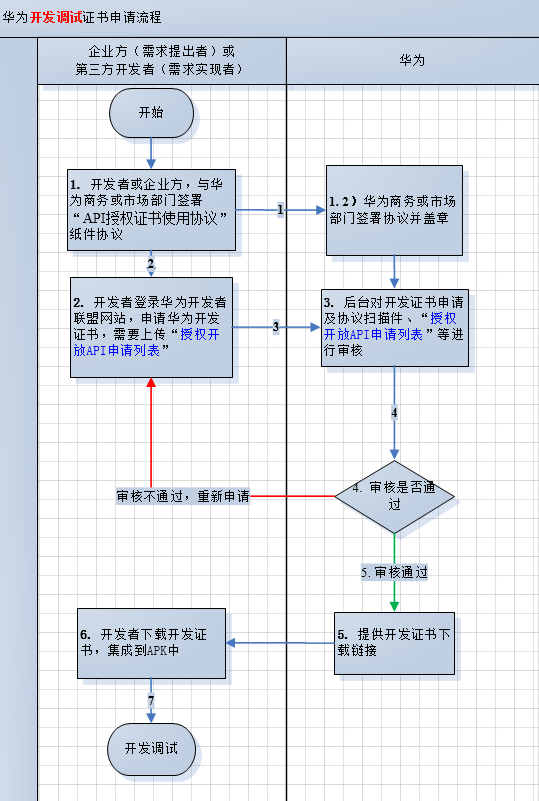
华为证书分为两种：开发证书和商用证书。开发证书，可以方便开发者在华为设备上进

行功能的开发与调试。商用证书，是开发者在APP的开发调试功能基本完成后，准备批量预置或商用发布之前，需要将商用证书替换开发证书集成在APP中。 注：在申请华为证书之前，需要与华为商务或市场部门签署相关协议。

## 3.2.1开发证书的申请流程

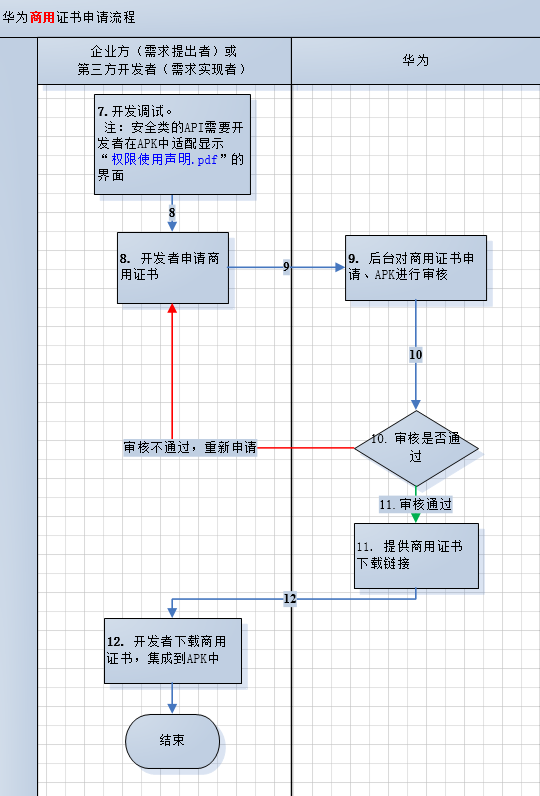
开发证书的申请之前，需与华为商务或市场部门签署《API授权证书使用协议》，

登陆华为开发者联盟申请，并提交《授权开放API申请列表》。



## 3.2.2 V1商用证书的申请流程

申请商用证书，需要上传APK进行审核。开发者需要按照《HW安全类授权开放SDK开发指导书》要求，完成“权限使用声明”界面的适配。



## 3.2.3 V2商用证书的申请流程

前面步骤同V1证书申请流程，在V1商用证书申请流程基础上，步骤11得到的是商用临时证书的下载链接，然后步骤12下载临时商用证书集成到apk中，使用该集成临时证书的apk重复以上8,9,10三个步骤，申请得到正式使用的商用证书，将该正式商用证书集成到apk中使用，如下图：



## 申请华为证书

1. 华为安全类授权开放SDK的使用，是基于华为证书的授权，需要开发者首先要进行华为证书申请。

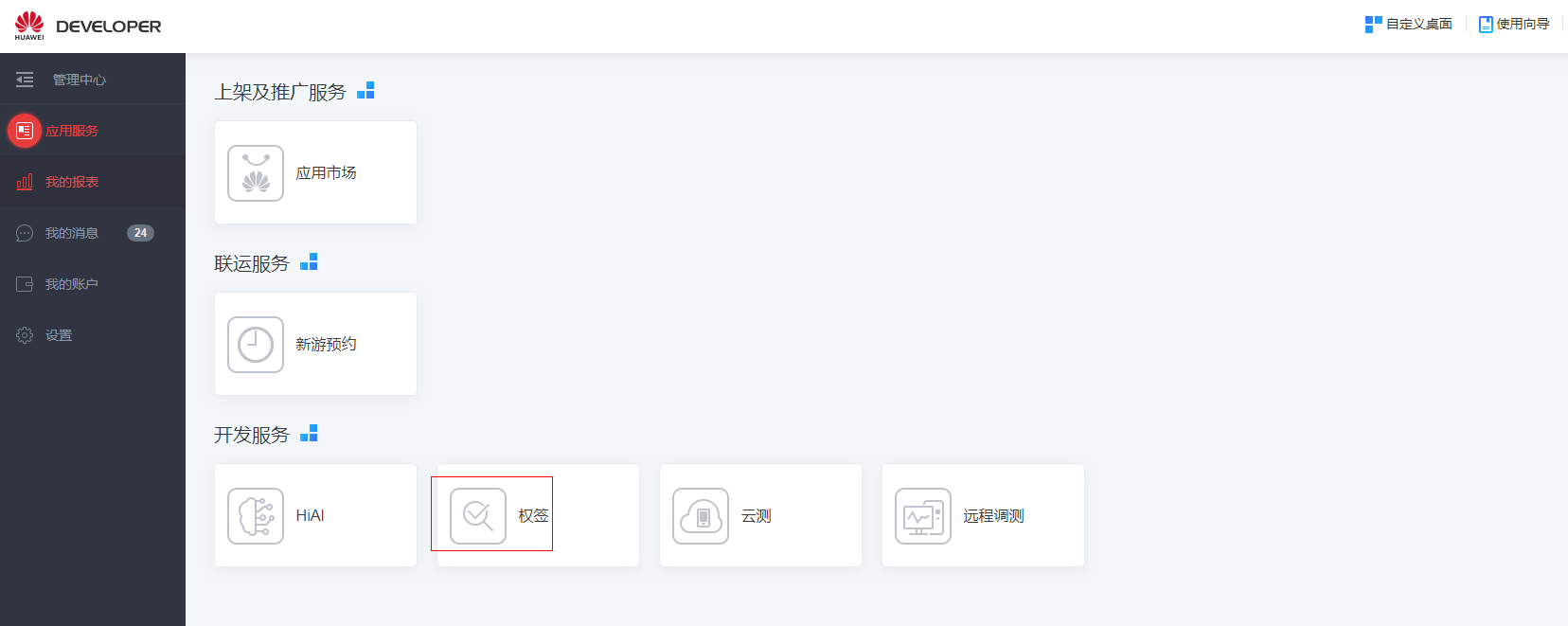
网页地址：http://developer.huawei.com/cn/consumer/



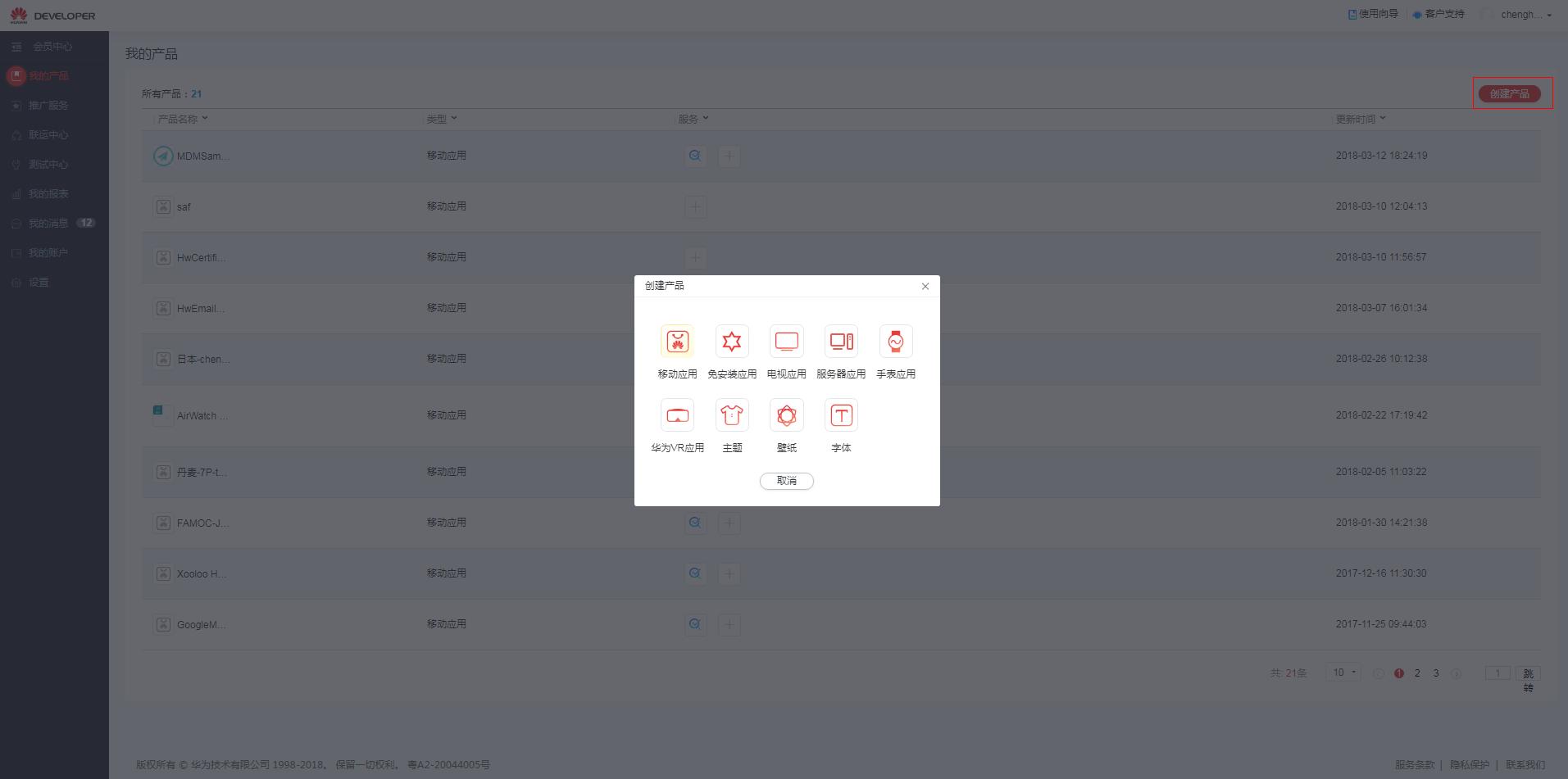
在使用设备管理类和应用权限管理类功能之前，开发者需要到华为开发者联盟去申请华为证书。

1. 注册并登陆华为账号后，进入如下网址：<https://developer.huawei.com/consumer/cn/console#/serviceCards/>

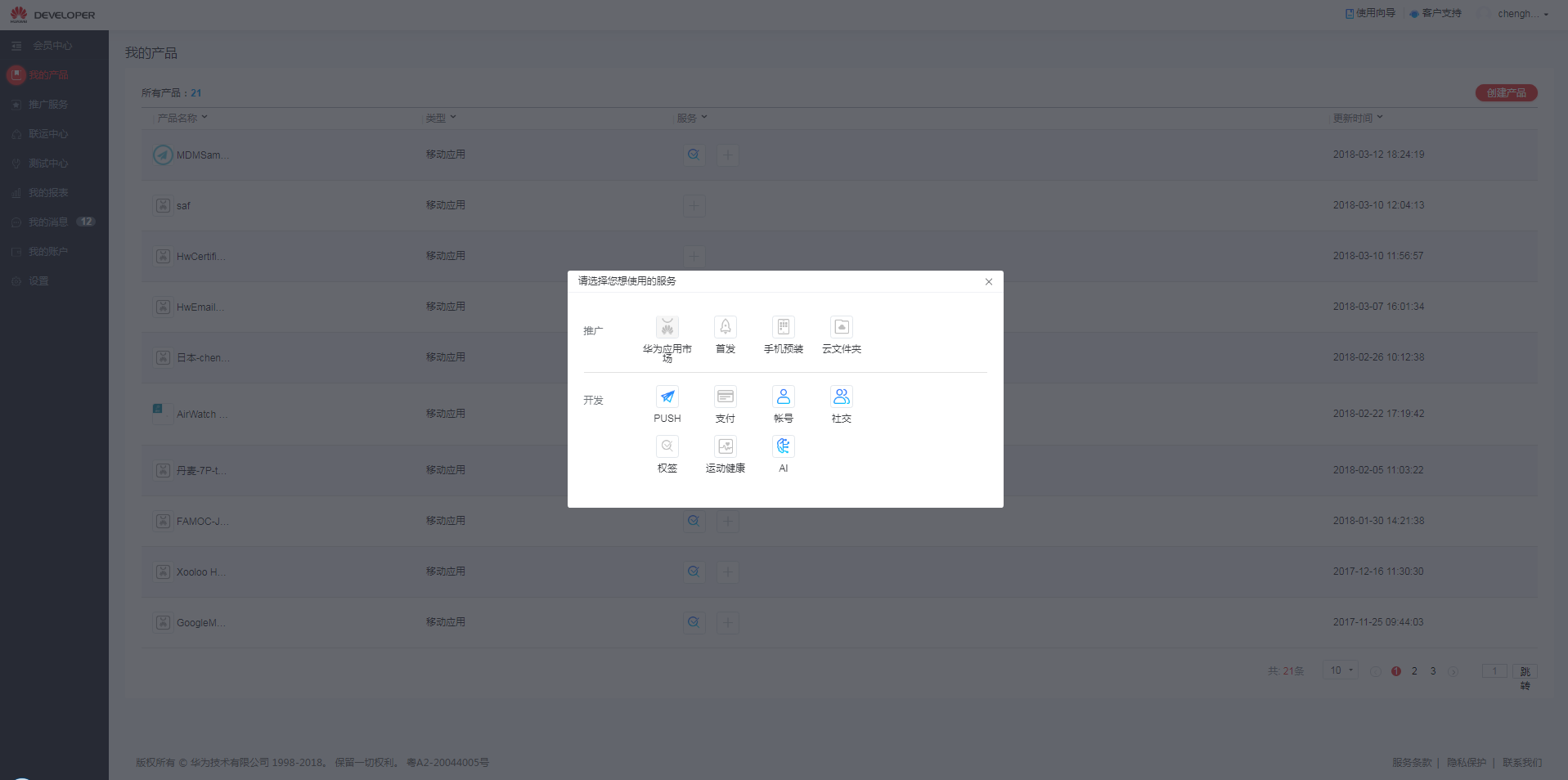
选择应用服务->开发服务->权签。



1. 开发者需要先按照下图，创建移动应用。



1. 在创建移动应用后，点击“”，为此开发者应用创建权签平台申请。

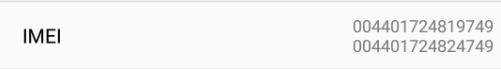


1. 点击“下一步”，

**注意：**填写“开发设备列表”项时，如果填写的是连续的IMEI号（即区间），请使用“-”分隔，且必须小值在前，大值在后，否则无法申请开发证书。

例如：用户想申请IMEI号的区间范围是：450514654130000-450514654139999的设备，则固定格式必须是：450514654130000-450514654139999。

另外，如果当前设备有两个IMEI号，则设备号可以并列申请，不必按照区间申请，如下图：



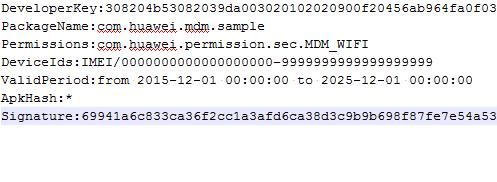
则DeviceIds字段可以申请为：IMEI/004401724819749,IMEI/004401724824749



1. 按照提示要求，申请权签平台的华为证书，当申请审核通过后，点击刚才的权签平台按钮，这时会有下载链接，开发者点击此链接下载开发者证书。



下面截图是一个已经申请成功的开发证书：



## 将华为证书打包到开发者应用中

使用上面提到的“DevPack.exe”工具，将刚刚下载的华为证书打包到开发者APK中：



点击“OK”后，会提示保存 “\*\*\*\_update.apk”，证书在“META-INF”目录下。

**注意：**如果开发者的APK是采用Android v2（Android N版本新增）签名方式进行签名的，那么将证书集成到APK中后，还需要再次对此APK按照Android v2进行签名（可参考FAQ）。

# 华为安全类授权开放SDK使用详解

**注意：**无论是使用设备管理类，还是应用权限管理类，前提都是开发者应用集成了正确的华为证书（使用了第2章的步骤），当应用安装到华为手机上后，此应用就具有了使用上述两类的权限。

## 获取当前hwMdm版本号

hwMdm提供了一个VersionInfo类，可以获取当前的版本号：

|  |
| --- |
| String version = VersionInfo.getApiVersion(); |

## 设备管理类

## 4.2.1目前支持的管理类

* DeviceApplicationManager
* DevicePackageManager
* DeviceControlManager
* DeviceEmailManager
* DevicePhoneManager
* DeviceRestrictionManager
* DeviceNetworkManager
* DeviceWifiPolicyManager
* DeviceTelephonyManager
* DeviceHwSystemManager
* DeviceCameraManager
* DeviceVpnManager
* DeviceFirewallManager
* DeviceSettingsManager
* DevicePasswordManager
* DeviceStorageManagerEx
* DeviceLocationManager
* DeviceInfraredManager
* DeviceP2PManager

以上管理类包含的接口，可以参看JAVA DOC文档。

## 4.2.2使用方法

**注意：**由于开发者应用集成的hwsdk-mdm-openapi-9.0.0.1.jar，在非华为设备、华为EMUI 4.0及以下设备上无具体实现，其接口会抛出NoAPIException，所以开发者应用需要对此异常的捕获，只有在华为设备上运行后，才能调用具体实现。

* + - 1. 首先激活开发者应用

|  |
| --- |
| import com.huawei.android.app.admin;  //使用Android原生DevicePolicyManager  mDevicePolicyManager = (DevicePolicyManager) getSystemService(Context.DEVICE\_POLICY\_SERVICE);  //创建当前APK的组件名称  mAdminName = new ComponentName(this, SampleDeviceReceiver.class);  ………  //如果当前APK没有被设备管理激活，则先进行激活  if (mDevicePolicyManager !=null && !mDevicePolicyManager.isAdminActive(mAdminName)) {  Intent intent = new Intent(  DevicePolicyManager.ACTION\_ADD\_DEVICE\_ADMIN);  intent.putExtra(DevicePolicyManager.EXTRA\_DEVICE\_ADMIN, mAdminName);  mActivity.startActivityForResult(intent, REQUEST\_ENABLE);  } |

* + - 1. 使用相关管理类实现对应功能

|  |
| --- |
| DeviceRestrictionManager restriction = new DeviceRestrictionManager();  restriction.setWifiDisabled(adminName, true); |

**注意：**如果在使用设备管理的管理类之前，没有激活开发者应用，那么会抛出SecurityException异常。

详细内容请参考Sample例子和JAVA DOC。

# 避免混淆打包

开发者编译应用时请不要混淆本SDK，避免功能异常。在配置文件中加入：

-keep class com.huawei.android.app.\*\*{\*;}

仅供参考。

# FAQ

## 在其他非华为手机上是否可以使用华为安全类授权开放SDK？

答：不可以，只能支持华为设备的功能。

## 如果华为开发者联盟网站的界面，与本文档中界面不一致，是什么原因？

答：这可能因为华为开发者联盟界面有过更新，请以开发者联盟实际为准。

## 如何获取设备的IMEI号、WIFI MAC地址和MEID？

答：IMEI号和MEID可以在设置中查询，在“设置”->“关于手机”->“IMEI”或“MEID”查看；

WIFI MAC地址可以在“设置”->“WLAN”->“菜单”->“高级设置”->“MAC地址”查看；

## 如果手机无IMEI号，如何申请开发证书？

答：由于某些电信制式的手机，可能无IMEI号，此时建议开发者可以使用WIFI MAC地址或MEID。

## 证书申请周期一般有多长时间？

答：开发证书一般审核时间为2~3天，商用证书一般为1~2周。

## 如果通过MDM应用禁用了USB文件传输功能，忘记了开启，并且把MDM应用删除了，该怎么办？

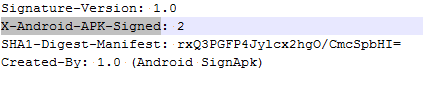
答：当MDM应用被取消激活，或被删除后，此MDM之前所设置的所有状态，都会被取消掉。此时想打开USB文件传输功能，需要用户手动打开USB调试，然后重新拔插USB即可。

## 如何判断APK是否使用了Android v2签名？

答：Android v2签名，是Google在Android N版本上引入的一种签名方式，目的是为了提升APK的安装速度。

查看Android v2签名的方法：

1）开发者可以参看APK的根目录“META-INF/CERT.SF”文件，如果其文件头有“X-Android-APK-Signed”字段，则为Android v2签名，没有则为原有签名方式。如下图：



2）使用Android N版本的新增类和接口判断：

ApkSignatureSchemeV2Verifier. hasSignature

ApkSignatureSchemeV2Verifier. verify

## 如果已经申请了商用证书，后来APK版本更新了，还需要再次申请商用书吗？

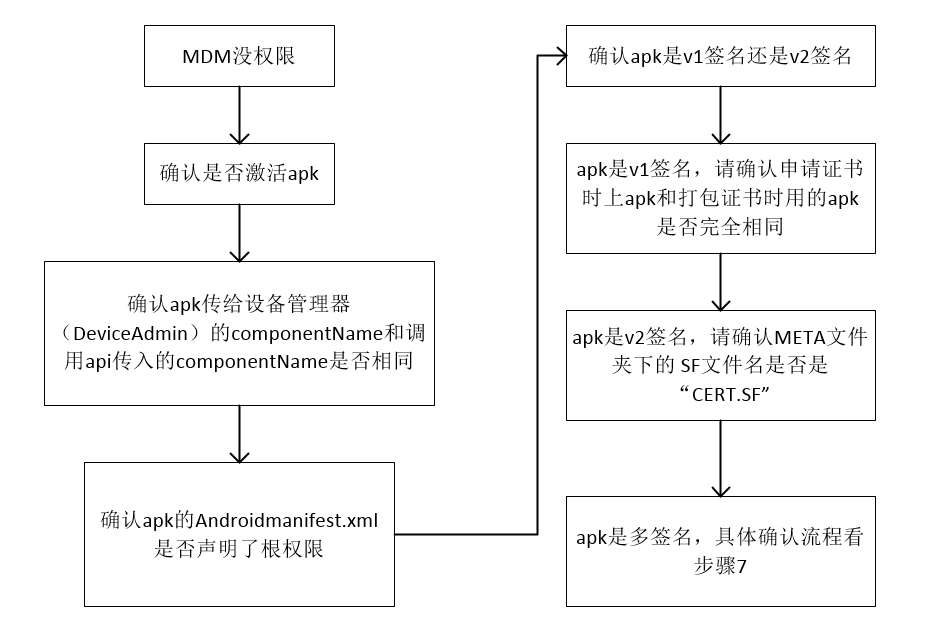
答：是的，由于商用证书会与APK版本进行绑定（商用证书APKHASH必选），所以开发者APK更新后，需要申请新的商用证书。

## 最常见报错信息



**含有does not have device\_manager MDM permission字段的log就是该apk没有权限调用MDM api**

1. **一般自查流程**



提示没有MDM权限--》查看应用是否已经激活--》确认apk传给设备管理器（DeviceAdmin）的ComponentName和调用api传入的ComponentName是否相同--》确认apk的AndroidManifest.xml是否申明了权限--》确认apk是V1签名还是V2签名--》apk是V1签名，请确认申请证书时的apk和打包证书时的apk是否完全相同--》apk是V2签名，请确认MATE文件夹下的SF文件名是否是“CERT.SF”--》apk是多签名的，具体看流程d

1. **定位问题流程**
2. 请确认apk是否在设置-安全与隐私-设备管理器中激活
3. 请确认apk传给设备管理器（DeviceAdmin）的componentName和调用api传入的componentName是否相同？ 因为Android系统会通过componentName判断apk是否激活，如果两次传入的componentName不一致，系统会认为激活的apk和调用api的apk不是同一个，当做未激活处理，报SecurityException。
4. 请确认apk的Androidmanifest.xml中是否申请了com.huawei.permission.sec.MDM或者com.huawei.permission.sec.MDM.v2根权限，因为PMS会通过这两个根权限判断是否需要解析华为证书，如果没有这两个根权限，设备会认为该apk不包含华为证书，不需要解析，如果华为证书没有解析，PMS就不会赋予apk MDM权限，调用api会报SecurityException，一定要保证Androidmanifest.xml的格式是正确的，建议不要把根权限放在声明权限的后方，如果前面的格式不对，PMS不会读到根权限。
5. 请确认apk是v1签名还是v2签名,如果是V1签名且apk只有一个.SF文件请看步骤e，如果是V2签名且apk只有一个.SF文件请看步骤f，如果多个.SF文件请查看步骤g。

确认方法：

1. 用WinRAR解压缩apk；
2. 用NotePad（任意一个文本编辑器）打开META文件夹下后缀名是.SF的文件，文件名默认为CERT.SF（第三方可能自定义SF文件名），查看前十行是否有X-Android-APK-Signed: 2这个字符串，如果有，说明这个apk是v2签名的，没有则是v1。
3. 如果是v1签名且apk只有一个.SF文件，apk没有权限请确认申请证书时上apk和打包证书时用的apk是否完全相同，因为设备会校验apk的hash值，如果apk发生改动了，APK的hash值与证书上记录的hash值不匹配，系统服务就不会赋予该APK MDM权限
4. 如果是v2签名且apk只有一个.SF文件，apk没有权限请确认，META文件夹下的 SF文件名是否是“CERT.SF”，当前所有版本的华为设备，只支持解析“CERT.SF”的签名文件，自定义文件名的.SF签名文件不会被解析。请重新签名apk，保持默认的命名方式不要自定义SF签名文件的名称，**key Alias要命名为cert**，重新签名请保证两次签名采用同一个密钥文件，否则证书解析会失败。
5. 如果有多个.SF文件，请按照以下思路确认：
6. 判断多个SF文件是否会出现一个是V1签名，一个是V2签名的情况，如果出现这种情况，说明业务方先用v2签名（Android Studio默认签名方式），然后又用v1签名的方式将证书打包到apk中，安装会失败，因为签名不统一，请业务方统一用v1或者v2进行签名。
7. 如果多个SF文件确实是统一的v1或者v2签名，请排查签名文件的算法是否统一，例如是否一个是SHA-1，一个是SHA-256，如果不一致，请让它们一致。
8. 如果前两步没问题了，如果是v1签名，请确认申请证书时上传的apk和打包证书时用的apk是否完全相同。如果是v2签名，请确认META文件夹下面是否有CERT.SF文件，如果没有，请让它有，重新签名，签名方法见后面科普知识

综上3条，即有多个SF文件要确保这些SF签名方式、签名算法相同。

1. **v2签名的apk打包证书后无法安装**

出现这种情况，绝大多数都是开发者采用了错误的签名方式导致的，请按如下思路确认问题：

1. 没打包证书的时候能不能安装，如果可以看步骤（2）
2. 解压缩apk，查看META文件下的后缀是SF的文件，是否有多个，如果有看步骤（3）
3. 是否同时存在v1、v2两种签名方式。出现这种情况往往是开发者用android studio打包时采用默认的v2签名，第二次手动签名的时候，采用某工具默认的v1签名方式，导致两种签名文件同时存在，设备会认为这个apk是非法的，原生系统会阻止这个apk的安装，如果没有这种情况请看步骤（4）
4. 请查看多个签名文件的hash算法是否一致（查看SF文件中是否有SHA-1，SHA-256的字段，对比是否一致），推荐采用同一种hash算法对同一个apk进行签名。

## 推荐使用签名工具介绍—apkSigner

apksigner.jar是Android Studio自带的一个签名工具。可在android sdk中获取到该签名工具，使用apksigner.jar签名时需要提供签名文件，业务方需要自己生成testkey.pk8与testkey.x509.pem 或者jks文件 ，签名方法可以参照这个https://developer.android.com/studio/command-line/apksigner.html#options-sign-general，网上类似的指导有很多，建议采用谷歌官方文档的方法！

## 如果v2签名的apk打包进去证书重新做签名，apk的hash值不就变了么？

apk的hash值肯定会变，重新签名是为了保证apk的完整性，保证apk可以在安卓系统中正常的安装（如果安卓系统认为你的v2 签名apk不完整、做了改动，系统会阻止apk的安装）。v1和v2最大的不同点是v2签名要校验安装包的完整性，我们把未打包证书的apk当做一个完整的整体，打包证书以后会破坏apk的完整性，为了保证apk的完整性，需要对apk再次签名。华为证书保存的是第一次签名时某些文件（信息安全）hash值，校验时不会把新打包进来的华为证书计算在内，因此能够兼容v2签名的apk，开发者要注意两次签名要采用相同的密钥，采用相同的hash算法，采用相同的签名方式（不要一次v2，一次v1），签名之后要保证“META/”文件夹下的SF文件文件名为“CERT.SF”，key Alias要命名为cert。

## V1签名V2签名的apk文件名的区别

如果是V2签名，META-INF中文件的命名分别为：MANIFEST.MF，CERT.SF，CERT.RSA，HUAWEI.CER；如果V1签名文件命名不用关心，可不遵守命名规则。

## 商用证书和开发证书在开发阶段的区别

一份商用证书绑定一份apk，该apk有任何的修改，都必须得重新申请商用证书，因为商用证书任何不一致都会导致解析不成功，进而导致没有权限调接口。开发证书则是在开发期间除了权限、hash值、设备号的修改不用重新申请外，其他修改都必须重新申请开发证书，否则影响接口调用。

## addPersistentApp和setSuperWhiteListForHwSystemManger两个保活接口有什么区别。

1) addPersistentApp:本接口对应开发指导书“保持某应用始终运行”功能，添加后，保活单中的应用不能被任何方式停止运行。Android DOZE模式和自启动、关联启动不在此范围内。需要申请com.huawei.permission.sec.MDM\_APP\_MANAGEMENT权限才能调用此接口。  
2) setSuperWhiteListForHwSystemManger:设置应用为信任应用，允许应用自启动、关联启动，默认开启忽略电池优化，允许后台运行，从而达到保活目的。需要申请 com.huawei.permission.sec.MDM\_APP\_MANAGEMENT 权限才能调用此接口。

注：建议两个接口配合使用。