

Unisoc Confidential For hiar

中国移动 Android 手机终端 DM 适配使用指南

文档版本
发布日期

V2.2
2020-08-27

版权所有 © 紫光展锐（上海）科技有限公司。保留一切权利。

本文件所含数据和信息都属于紫光展锐（上海）科技有限公司（以下简称紫光展锐）所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

Unisoc Confidential For hiar

紫光展锐（上海）科技有限公司



前言

概述

本文档介绍了中国移动 Android 手机终端 DM2.0 适配方案。DM（Device Management，终端管理），是运营商为了方便维护和管理终端、数据而制定的规范，是手机终端入库的基础。

读者对象

本文档适合以下人员阅读和参考：


- 维护中国移动 Android 手机终端 DM 方案的相关人员
- 负责中国移动 Android 手机终端 DM 测试的相关人员

缩略语

缩略语	英文全名	中文解释
DM	Device Management	终端管理
IMSI	International Mobile Subscriber Identification Number	国际移动用户识别码
ICCID	Integrate Circuit Card Identity	集成电路卡识别码（固化在手机 SIM 卡） ICCID 为 IC 卡的唯一识别号码，由 20 位数字组成
IMEI	International Mobile Equipment Identity	国际移动设备身份码
VoLTE	Voice over LTE	基于 LTE 的语音业务

符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

变更信息

文档版本	发布日期	修改说明
V1.0	2018-08-10	第一次正式发布。
V2.0	2019-05-22	增加 Android 4.4、Android 9.0 DM2.0 适配指导说明
V2.1	2020-04-02	结构调整、内容优化、格式更新等
V2.2	2020-08-27	增加 Android 10.0、Android 11.0 DM2.0 适配指导说明

关键字

Appkey、功耗、商用版本、调试版本、SDK。

Unisoc Confidential For hiar

目 录

1 中国移动 Android 手机终端 DM 介绍	1
1.1 中国移动 Android 手机终端 DM 的简介	1
1.2 Android 手机终端客户端 DM 集成方式说明	1
1.3 紫光展锐 Android 手机平台 DM 的支持情况	2
2 Android 4.4 移动 DM 方案适配说明	3
2.1 替换 AppKey	3
2.2 DM SDK 调试版本集成	3
2.3 APK 的适配	4
2.4 参数配置	4
2.5 DM SDK 商用版本集成	5
3 Android 7.0/8.1 移动 DM 方案适配说明	6
3.1 替换 AppKey	6
3.2 DM SDK 调试版本集成	6
3.3 APK 的适配	7
3.4 参数配置	7
3.5 DM SDK 商用版本集成	8
4 Android 9.0 移动 DM 方案适配说明	10
4.1 替换 AppKey	10
4.2 DM SDK 调试版本集成	10
4.3 APK 的适配	11
4.4 参数配置	11
4.5 DM SDK 商用版本集成	12
5 Android 10.0 移动 DM 方案适配说明	14
5.1 替换 AppKey	14
5.2 DM SDK 调试版本集成	14
5.3 APK 的适配	15
5.4 参数配置	16
5.5 SDK 升级修改	16
5.6 DM SDK 商用版本集成	17
6 Android 11.0 移动 DM 方案适配说明	19
6.1 替换 AppKey	19
6.2 DM SDK 调试版本集成	19
6.3 APK 的适配	20
6.4 参数配置	21

6.5 DM SDK 商用版本集成	21
7 DM 自测试说明及常见问题	23
7.1 概述	23
7.2 集成测试常见问题	23
7.3 集成测试注意事项	24
7.4 功耗测试建议	24

Unisoc Confidential For hiar

图目录

图 1-1 终端客户端 SDK 集成方案	1
----------------------------	---

Unisoc Confidential For hiar

表目录

表 1-1 各版本的工程支持情况	2
------------------------	---

Unisoc Confidential For hiar

1 中国移动 Android 手机终端 DM 介绍

1.1 中国移动 Android 手机终端 DM 的简介

DM（Device Management，终端管理）是运营商为了方便维护和管理终端、数据而制定的规范，是手机终端入库的基础。如果某手机终端厂商的手机终端产品需要录入中国移动的产品库，手机终端必须要满足中国移动的 DM 规范要求。

DM 对手机终端的要求，包含以下功能：

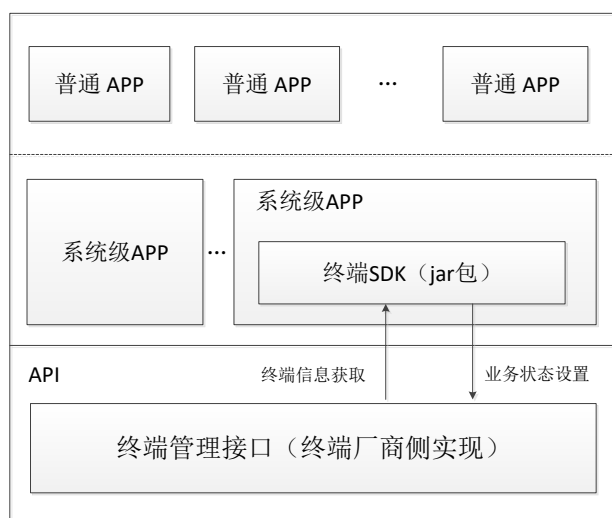
- 终端自注册
- 数据采集
- FOTA
- 业务配置

中国移动要求在 2017 年及以后，所有入中国移动产品库的定制、非定制 Android 手机终端均要支持终端管理功能。终端管理及服务系统正式发布之后，Android 手机产品需进行适配、测试、预装后方可录入中国移动产品库。

1.2 Android 手机终端客户端 DM 集成方式说明

Android 手机终端客户端 DM 集成方式说明，请参见图 1-1。

图1-1 终端客户端 SDK 集成方案



终端厂商将 SDK 终端内嵌至系统应用中，且应用用户不可见，同时确保其终端自注册信息采集正常，业务处理正常，保证客户端具备开机自启动权限，具备涉及联网、数据采集所需要的所有系统权限，保证客户端进程不会被系统安全管理软件清除。

其中终端客户端 SDK 主要实现相关数据采集及业务功能设置，实现终端数据上报、终端管理及服务平台。终端厂商需要实现终端管理接口，完成终端 SDK 内置集成。目前集成的 DM SDK 版本均为 2.0。

1.3 紫光展锐 Android 手机平台 DM 的支持情况

紫光展锐各平台移动 DM 的支持情况，请参见表 1-1。表中所列紫光展锐 Android 手机平台当前版本及之后的版本均支持移动 DM2.0 功能。

表1-1 各版本的工程支持情况

Android 版本	软件版本	工程支持	编译目标	默认开启
Android 4.4	MOCOR5_SFPHONE_17F_RLS1_W19.16.3	CMCC 工程	User 版本	否
			Debug 版本	是
Android 7.0	MOCORDROID7.0_Trunk_k44_SharkL2_W18.29.1	CMCC 工程	User 版本	否
			Debug 版本	是
	MOCORDROID7.0_Trunk_K44_17E_W18.28.5	CMCC 工程	User 版本	否
			Debug 版本	是
Android 8.1	MOCORDROID8.1_Trunk_18B_SHARKL3_W18.32.5	CMCC 工程	User 版本	否
			Debug 版本	是
Android 9.0	MOCORDROID9.0_Trunk_W18.51.4.	CMCC 工程	User 版本	否
			Debug 版本	是
Android 10.0	所有已发布的量产版本	CMCC 工程	User 版本	否
			Debug 版本	是
Android 11.0	所有已发布的量产版本	CMCC 工程	User 版本	否
			Debug 版本	是

2 Android 4.4 移动 DM 方案适配说明

以版本 MOCOR5_SFPHONE_17F_RLS1_W19.16.3 修改为例说明 DM2.0 方案的适配。考虑到 DM SDK 调试版本对平台软件功耗的影响较大，Android 手机 User 版本默认关闭移动 DM 功能，仅在 Userdebug 版本默认开启 DM 功能。建议在 Android 手机 Userdebug 版本上集成调试版 DM SDK 做 DM 功能测试，自测试通过后，User 版本再集成商用版本 DM SDK 送中国移动入库测试。

2.1 替换 AppKey

AppKey 一般由终端管理及服务平台生成，可通过测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取。如果无法获取，请联系移动接口人来获取登录测试平台的帐号密码及 AppKey。将文件 vendor/sprd/operator/cmcc/packages/OpManager/res/values/strings.xml 的 value 修改为获取的 Appkey。

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
<string name="appKey">A100000860</string>
</resources>
```

2.2 DM SDK 调试版本集成

步骤 1 登录移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取最新的 SDK 调试版本（紫光展锐平台默认集成的 SDK 调试版本为 dm_sdk_test_2.0.50_jdk1.6_v2.jar）

步骤 2 将以下路径的 SDK 替换为移动测试平台获取的最新的 SDK

vendor/sprd/operator/cmcc/packages/OpManager/libs/dm_sdk_test_2.0.50_jdk1.6_v2.jar

步骤 3 登陆移动测试平台获取并替换以下库文件：

- vendor/sprd/operator/cmcc/packages/OpManager/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/OpManager/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/OpManager/libs/armeabi/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/OpManager/libs/x86/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/OpManager/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/operator/cmcc/packages/OpManager/Android.mk 中 SDK 配置信息修改为替换的 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs/dm_sdk_test_2.0.50_jdk1.6_v2.jar
```

---结束

2.3 APK 的适配

根据移动对 APK 包名和应用名的要求，需要进行代码调整，操作步骤如下：

步骤 1 以下文件中，将 com.sprd.opm 替换为移动要求的实际包名 “com.终端厂家名称.regservice”

- frameworks/base/core/java/android/app/ApplicationPackageManager.java
- packages/apps/Settings/src/com/android/settings/applications/ApplicationsState.java
- packages/apps/Settings/src/com/android/settings/location/RecentLocationApps.java
- vendor/sprd/operator/cmcc/packages/OpManager/AndroidManifest.xml
- vendor/sprd/operator/cmcc/packages/OpManager/src/com/sprd/opm/OPApplication.java
- frameworks/base/core/res/res/values/config.xml

步骤 1 修改以下文件，将 OpManager 替换为 RegService

- device/sprd/sharkle/sp9820e_2h10ll / sp9820e_2h10ll_cmcc.mk


```
ifeq ($(strip $(DM_SUPPORT)), true)
PRODUCT_PACKAGES += \
    OpManager \
    Provision2
```
- vendor/sprd/operator/cmcc/packages/OpManager/Android.mk


```
LOCAL_SRC_FILES := $(call all-java-files-under, src)
LOCAL_PACKAGE_NAME := OpManager
LOCAL_CERTIFICATE := platform
```

步骤 2 修改 APK 目录结构 vendor/sprd/operator/cmcc/packages/OpManager

- OpManager 目录名替换为 RegService
- com/sprd/opm 目录结构替换为 com/终端厂家名称/regservice

说明

根据移动要求，固件版本号需对应手机的版本号：“设置”->“关于手机”->“版本号”（对应属性：ro.build.display.id），且“固件版本号”字符串最大长度为 32 个字符。

---结束

2.4 参数配置

在项目工程主 mk 中，配置送测手机“设备型号”、“ROM”、“RAM”信息。

以 Mocor5 SC9820E 为例，修改 device/sprd/sharkle/sp9820e_2h10ll / sp9820e_2h10ll_cmcc.mk，添加如下信息：

```
PRODUCT_PROPERTY_OVERRIDES += \
    ro.cta.hardware.model=SPREADTRUM \
    ro.cta.hardware.rom.size=512M \
    ro.cta.hardware.ram.size=256M
```

- ro.cta.hardware.model：工信部报备时填写的“设备型号”

- ro.cta.hardware.rom.size: 项目 ROM 大小
- ro.cta.hardware.ram.size: 项目 RAM 大小

2.5 DM SDK 商用版本集成

SDK 调试版测试通过后，终端入库送测以及正式发售必须集成商用版 SDK（移动 DM SDK 商用版本通过移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取）。

步骤 1 User 版本开启 DM 功能：

修改 device/sprd/sharkle/sp9820e_2h10ll / sp9820e_2h10ll_cmcc.mk 中的 DM 功能配置信息。将 true 改为 false。

```
# DM statement start
DM_SUPPORT_ONLY_FOR_USERDEBUG := true ➡ false
```

步骤 2 商用版本替换 SDK：

将 vendor/sprd/operator/cmcc/packages/apps/RegService/libs/下 dm_sdk_test_2.0.50_jdk1.6_v2.jar 替换为移动 DM SDK 商用版本。

步骤 3 移动发布的正式 sdk 若包含以下库文件，请依次替换。若不包含，请忽略此步骤。

- vendor/sprd/operator/cmcc/packages/RegService/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/RegService/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/RegService/libs/armeabi/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/RegService/libs/x86/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/RegService/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/operator/cmcc/packages/RegService/Android.mk 中 SDK 配置信息修改为商用版 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs, dm_sdk_test_2.0.50_jdk1.6_v2.jar
```

---结束

📖 说明

- 调试版本通过测试并在测试平台提交过测试报告的手机，相同 IMEI 号的手机在测试平台将无法上报信息（移动为控制测试平台无效数据上报，对已经通过测试平台测试并上传自测试报告的手机做 DM 信息上报屏蔽处理），如果需要继续做测试，需要更换手机 IMEI 号。
- IMEI 号有效期为 1 个月，从第一次上报的时间开始计算。
- 集成 DM SDK 商用版本后，DM 相关数据将上报到移动 DM 商用平台，移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 无法查看相关数据，仅移动可查看。

3 Android 7.0/8.1 移动 DM 方案适配说明

以版本 MOCORDROID7.0_Trunk_k44_SharkL2_W18.26.5 修改为例说明 DM 方案的适配。考虑到 DM SDK 调试版本对平台软件功耗的影响较大，Android 手机 User 版本默认关闭移动 DM 功能，仅在 Userdebug 版本默认开启 DM 功能。建议在 Android 手机 Userdebug 版本上集成调试版 DM SDK 做 DM 功能测试，自测试通过后，User 版本再集成商用版本 DM SDK 送中国移动入库测试。

3.1 替换 AppKey

AppKey 一般由终端管理及服务平台生成，可通过测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取。如果没有，请联系移动接口人获取登录测试平台的帐号密码及 AppKey。将文件 vendor/sprd/operator/cmcc/packages/apps/OpManager/res/values/strings.xml 的 value 修改为获取的 Appkey。

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
<string name="appKey">A100000860</string>
</resources>
```

3.2 DM SDK 调试版本集成

步骤 1 登录移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取最新的 SDK 调试版本（紫光展锐平台默认集成的 SDK 调试版本为 dm_sdk_test_2.0.47_jdk1.8.jar）

步骤 2 将以下路径的 SDK 替换为移动测试平台获取的最新的 SDK

vendor/sprd/operator/cmcc/packages/apps/OpManager/libs/dm_sdk_test_2.0.47_jdk1.8.jar

步骤 3 登陆移动测试平台获取并替换以下库文件：

- vendor/sprd/operator/cmcc/packages/apps/OpManager/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/OpManager/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/OpManager/libs/armeabi/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/OpManager/libs/x86/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/OpManager/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/operator/cmcc/packages/apps/OpManager/Android.mk 中 SDK 配置信息修改为替换的 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs/dm_sdk_test_2.0.47_jdk1.8.jar
```

---结束

3.3 APK 的适配

根据移动对 APK 包名和应用名的要求，需要做如下代码调整。

步骤 1 以下文件中，将 com.sprd.opm 替换为移动要求的实际包名 “com.终端厂家名称.regservice”

- frameworks/base/core/java/android/app/ApplicationPackageManager.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/applications/ApplicationsState.java
- frameworks/base/services/core/java/com/android/server/pm/PackageManagerService.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/location/RecentLocationApps.java
- vendor/sprd/operator/cmcc/packages/apps/OpManager/AndroidManifest.xml
- vendor/sprd/operator/cmcc/packages/apps/OpManager/src/com/sprd/opm/OPApplication.java
- vendor/sprd/platform/frameworks/base/core/res/res/values/config_ex.xml

步骤 2 以下文件将 OpManager 替换为 RegService

- vendor/sprd/operator/cmcc/configs/spec_common.mk

```
ifeq ($(strip $(DM_SUPPORT)), true)
PRODUCT_PACKAGES += \
    OpManager \
    Provision2
endif
# DM statement end
```

- vendor/sprd/operator/cmcc/packages/apps/OpManager/Android.mk

```
LOCAL_SRC_FILES := $(call all-java-files-under, src)
LOCAL_PACKAGE_NAME := OpManager
LOCAL_CERTIFICATE := platform
```

步骤 3 修改 APK 目录结构 vendor/sprd/operator/cmcc/packages/apps/OpManager

- OpManager 目录名替换为 RegService
- com/sprd/opm 目录结构替换为 com/终端厂家名称/regservice

说明

根据移动要求，固件版本号需对应手机的版本号：“设置”->“关于手机”->“版本号”（对应属性：ro.build.display.id），且“固件版本号”字符串最大长度为 32 个字符。

---结束

3.4 参数配置

在项目工程主 mk 中，配置送测手机“设备型号”、“ROM”、“RAM”信息。

以 SP9850KA 为例，修改 device/sprd/sharkl2/sp9850ka_1h10/sp9850ka_1h10_cmccll.mk，添加如下信息：


```
PRODUCT_PROPERTY_OVERRIDES := \
persist.radio.modem.config=TL_LF_TD_W_G,TL_LF_TD_W_G.\
persist.radio.modem.workmode=9,9.\
ro.radio.modem.capability=TL_LF_TD_W_G,TL_LF_TD_W_G.\
ro.cta.hardware.model=.SPREADTRUM.\
ro.cta.hardware.rom.size=.16GB.\
ro.cta.hardware.ram.size=.2GB.\
$(PRODUCT_PROPERTY_OVERRIDES)
```

- ro.cta.hardware.model: 工信部报备时填写的“设备型号”
- ro.cta.hardware.rom.size: 项目 ROM 大小
- ro.cta.hardware.ram.size: 项目 RAM 大小

3.5 DM SDK 商用版本集成

SDK 调试版测试通过后，终端入库送测以及正式发售必须集成商用版 SDK（移动 DM SDK 商用版本通过移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取）。

步骤 1 User 版本开启 DM 功能：

修改 vendor/sprd/operator/cmcc/configs/spec_common.mk DM 功能的配置信息。将 true 改为 false。

```
# DM statement start
DM_SUPPORT_ONLY_FOR_USERDEBUG := true ➡ false
```

步骤 2 商用版本 SDK 替换：

将 vendor/sprd/operator/cmcc/packages/apps/RegService/libs/中 dm_sdk_test_2.0.47_jdk1.8.jar 替换为移动 DM SDK 商用版本。

步骤 3 移动发布的正式 sdk 若包含以下库文件，请依次替换。若不包含，请忽略此步骤。

- vendor/sprd/operator/cmcc/packages/apps/RegService/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/RegService/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/RegService/libs/armeabi/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/RegService/libs/x86/libaes-jni.so
- vendor/sprd/operator/cmcc/packages/apps/RegService/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/operator/cmcc/packages/apps/RegService/Android.mk 中 SDK 配置信息修改为商用版 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs/dm_sdk_test_2.0.47_jdk1.8.jar
```

---结束

说明

- 调试版本通过测试并在测试平台提交过测试报告的手机，相同 IMEI 号的手机在测试平台将无法上报信息（移动为控制测试平台无效数据上报，对已经通过测试平台测试并上传自测试报告的手机做 DM 信息上报屏蔽处理），如果还需要继续做测试，需要更换手机 IMEI 号。

- IMEI 号有效期为 1 个月，从第一次上报的时间开始计算。
- 集成 DM SDK 商用版本后，DM 相关数据将上报到移动 DM 商用平台，移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 无法查看相关数据，仅移动可查看。

Unisoc Confidential For hiar

4 Android 9.0 移动 DM 方案适配说明

以版本 MOCORDROID9.0_Trunk_W18.51.4 修改为例说明 DM2.0 方案的适配。考虑到 DM SDK 调试版本对平台软件功耗的影响较大，Android 手机 User 版本默认关闭移动 DM 功能，仅在 Userdebug 版本默认开启 DM 功能。建议在 Android 手机 Userdebug 版本上集成调试版 DM SDK 做 DM 功能测试，自测试通过后，User 版本再集成商用版本 DM SDK 送中国移动入库测试。

4.1 替换 AppKey

AppKey 一般由终端管理及服务平台生成，可通过测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取。如果没有，请联系移动接口人获取登录测试平台的帐号密码及 AppKey。

将文件 vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/res/values/strings.xml 的 value 修改为获取的 Appkey。

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
<string name="appKey">A100000860</string>
</resources>
```

4.2 DM SDK 调试版本集成

步骤 1 登录移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取最新的 SDK 调试版本（紫光展锐平台默认集成的 SDK 调试版本为 dm_sdk_test_2.0.50_jdk1.8_v2.jar）

步骤 2 将以下路径的 SDK 替换为移动测试平台获取的最新的 SDK

vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/libs/dm_sdk_test_2.0.50_jdk1.8_v2.jar

步骤 3 登陆移动测试平台获取并替换以下库文件：

- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/libs/armeabi/libaes-jni.so
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/libs/x86/libaes-jni.so
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/Android.mk 中 SDK 配置信息修改为替换的 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs/dm_sdk_test_2.0.50_jdk1.8_v2.jar
```

---结束

4.3 APK 的适配

根据移动对 APK 包名和应用名的要求，需要做如下代码调整。

步骤 1 以下文件中将 com.sprd.opm 替换为移动要求的实际包名 “com.终端厂家名称.regservice”

- frameworks/base/core/java/android/app/ApplicationPackageManager.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/applications/ApplicationsState.java
- frameworks/base/services/core/java/com/android/server/pm/permission/PermissionManagerService.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/location/RecentLocationApps.java
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/AndroidManifest.xml
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/src/com/sprd/opm/OPApplication.java
- frameworks/base/core/res/res/values/config_ex.xml

步骤 2 以下文件中将 OpManager 替换为 RegService

- vendor/sprd/feature_configs/carriers/cmcc/config.mk


```

ifeq ($(strip $(DM_SUPPORT)), true)
FEATURES.PRODUCT_PACKAGES += \
  OpManager
endif
      
```
- vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager/Android.mk


```

LOCAL_SRC_FILES := $(call all-java-files-under, src)

LOCAL_PACKAGE_NAME := OpManager

LOCAL_CERTIFICATE := platform
      
```

步骤 3 修改 APK 目录结构 vendor/sprd/feature_configs/carriers/cmcc/apps/OpManager

- OpManager 目录名替换为 RegService
- com/sprd/opm 目录结构替换为 com/终端厂家名称/regservice

说明

根据移动要求，固件版本号对应手机“设置”->“系统”->“关于手机”->“版本号”（对应属性：ro.build.display.id）且“固件版本号”字符串最大长度为 32 个字符。

---结束

4.4 参数配置

在项目工程主 mk 中，配置送测手机“设备型号”、“ROM”、“RAM”信息。

添加如下信息：

```
PRODUCT_PROPERTY_OVERRIDES := \
persist.radio.modem.config=TL_LF_TD_W_G,TL_LF_TD_W_G \
persist.radio.modem.workmode=9,9 \
ro.radio.modem.capability=TL_LF_TD_W_G,TL_LF_TD_W_G \
ro.cta.hardware.model=.SPREADTRUM \
ro.cta.hardware.rom.size=.16GB \
ro.cta.hardware.ram.size=.2GB \
$(PRODUCT_PROPERTY_OVERRIDES)
```

- ro.cta.hardware.model: 工信部报备时填写的“设备型号”
- ro.cta.hardware.rom.size: 项目 ROM 大小
- ro.cta.hardware.ram.size: 项目 RAM 大小

4.5 DM SDK 商用版本集成

SDK 调试版测试通过后，终端入库送测以及正式发售必须集成商用版 SDK（移动 DM SDK 商用版本通过移动测试平台 <https://shipei.fxitsbl.com/terminalsrv> 获取）。

步骤 1 User 版本开启 DM 功能：

修改文件 vendor/sprd/feature_configs/carriers/cmcc/config.mk DM 功能的配置信息。将 true 改为 false。

```
# DM statement start
DM_SUPPORT_ONLY_FOR_USERDEBUG := true ➡ false
```

步骤 2 商用版本 SDK 替换：

将 vendor/sprd/feature_configs/carriers/cmcc/apps/RegService/libs/中 dm_sdk_test_2.0.50_jdk1.8_v2.jar 替换为移动 DM SDK 商用版本。

步骤 3 移动发布的正式 sdk 若包含以下库文件，请依次替换。若不包含，请忽略此步骤。

- vendor/sprd/feature_configs/carriers/cmcc/apps/RegService/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/feature_configs/carriers/apps/RegService/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/feature_configs/carriers/apps/RegService/libs/armeabi/libaes-jni.so
- vendor/sprd/feature_configs/carriers/apps/RegService/libs/x86/libaes-jni.so
- vendor/sprd/feature_configs/carriers/apps/RegService/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/feature_configs/carriers/cmcc/apps/RegService/Android.mk 中 SDK 配置信息修改为商用版 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs/dm_sdk_test_2.0.50_jdk1.8_v2.jar
```

---结束

📖 说明

- 调试版本通过测试并在测试平台提交过测试报告的手机，相同 IMEI 号的手机在测试平台将无法上报信息（移动为控制测试平台无效数据上报，对已经通过测试平台测试并上传自测试报告的手机做 DM 信息上报屏蔽处理），如果需要继续做测试，需要更换手机 IMEI 号。

- IMEI 号有效期为 1 个月，从第一次上报的时间开始计算。
- 集成 DM SDK 商用版本后，DM 相关数据将上报到移动 DM 商用平台，移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 无法查看相关数据，仅移动可查看。

Unisoc Confidential For hiar

5 Android 10.0 移动 DM 方案适配说明

考虑到 DM SDK 调试版本对平台软件功耗的影响较大，Android 手机 User 版本默认关闭移动 DM 功能，仅在 Userdebug 版本默认开启 DM 功能。建议在 Android 手机 Userdebug 版本上集成调试版 DM SDK 做 DM 功能测试，自测试通过后，User 版本再集成商用版本 DM SDK 送中国移动入库测试。

5.1 替换 AppKey

AppKey 一般由终端管理及服务平台生成，可通过测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取。如果没有，请联系移动接口人获取登录测试平台的帐号密码及 AppKey。

将文件 vendor/sprd/carriers/cmcc/packages/apps/OpManager/res/values/strings.xml 的 value 修改为获取的 Appkey。

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
<string name="appKey">A100000860</string>
</resources>
```

5.2 DM SDK 调试版本集成

步骤 1 登录移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取最新的 SDK 调试版本（紫光展锐平台默认集成的 SDK 调试版本为 dm_sdk_test_2.0.50_jdk1.8_v2.jar）

步骤 2 将以下路径的 SDK 替换为移动测试平台获取的最新的 SDK

vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/dm_sdk_test_2.0.50_jdk1.8_v2.jar

步骤 3 将以下路径的 android-async-http 库替换为移动测试平台获取的最新库文件

vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/android-async-http-1.4.8.jar

步骤 4 登陆移动测试平台获取并替换以下库文件：

- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/arm64-v8a/libaes-jni.so
- 8vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/armeabi/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/x86/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/x86_64/libaes-jni.so

步骤 5 将 vendor/sprd/carriers/cmcc/packages/apps/OpManager/Android.mk 中

- SDK 配置信息修改为替换的 SDK 名称
- android-async-http 库信息修改为替换的 android-async-http 库名称

```
include $(CLEAR_VARS)
```

```
LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES :=libdm_sdk_test:libs/dm_sdk_test_2.0.50_jdk1.8_v2.jar
```

```
LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES +=android-async-http:libs/android-async-http-1.4.8.jar
```

---结束

5.3 APK 的适配

根据移动对 APK 包名和应用名的要求，需要做如下代码调整。

步骤 1 以下文件将 com.sprd.opm 替换为移动要求的实际包名 “com.终端厂家名称.regservice”

- frameworks/base/core/java/android/app/ApplicationPackageManager.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/applications/ApplicationsState.java
- frameworks/base/services/core/java/com/android/server/pm/permission/PermissionManagerService.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/location/RecentLocationApps.java
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/AndroidManifest.xml
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/src/com/sprd/opm/OPApplication.java
- frameworks/base/core/res/res/values/config_ex.xml
- frameworks/base/telephony/java/android/telephony/DmykAbsTelephonyManager.java
- frameworks/base/api/current.txt

步骤 2 以下文件将 OpManager 替换为 RegService

- vendor/sprd/feature_configs/carriers/cmcc/config.mk


```
ifeq ($(strip $(DM_SUPPORT)), true)
FEATURES.PRODUCT_PACKAGES += \
  OpManager
endif
```
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/Android.mk


```
LOCAL_SRC_FILES := $(call all-java-files-under, src)

LOCAL_PACKAGE_NAME := OpManager

LOCAL_CERTIFICATE := platform
```

步骤 3 修改 APK 目录结构 vendor/sprd/carriers/cmcc/packages/apps/OpManager

- OpManager 目录名替换为 RegService
- com/sprd/opm 目录结构替换为 com/终端厂家名称/regservice

说明

根据移动要求，固件版本号对应手机“设置”->“系统”->“关于手机”->“版本号”(对应属性: ro.build.display.id) 且“固件版本号”字符串最大长度为 32 个字符。

---结束

5.4 参数配置

在项目工程主 mk 中，配置送测手机“设备型号”、“ROM”、“RAM”信息。

添加如下信息：

```
PRODUCT_PRODUCT_PROPERTIES += \  
    ro.cta.hardware.model = UNISOC \  
    ro.cta.hardware.rom.size = 32G \  
    ro.cta.hardware.ram.size = 4G
```

- ro.cta.hardware.model：工信部报备时填写的“设备型号”
- ro.cta.hardware.rom.size：项目 ROM 大小
- ro.cta.hardware.ram.size：项目 RAM 大小

5.5 SDK 升级修改

升级 dm_sdk_test_2.0.56 及以后版本，需要做如下修改：

- 新增接口实现

- frameworks/base/telephony/java/android/telephony/DmykAbsTelephonyManager.java 中增加接口说明：

```
public abstract String getModel();
```

- frameworks/base/telephony/java/android/telephony/DmykImplTelephonyManager.java 中增加接口实现：

```
@Override  
public String getModel() {  
    return Build.MODEL;  
}
```

- 通过编译命令“make update-api”更新 frameworks/base/api/current.txt 文件，更新后截图参考如下：

```
package com.dmyk.android.telephony {  
  
    public abstract class DmykAbsTelephonyManager {  
        ctor protected DmykAbsTelephonyManager(android.content.Context);  
        method public abstract android.net.Uri getAPNContentUri(int);  
        method public abstract String getCPUModel();  
        method public abstract String getCTAModel();  
        method public abstract String getCdmaDeviceId();  
        method public abstract int getCellId(int);  
        method public abstract int getDataState(int);  
        method public static com.dmyk.android.telephony.DmykAbsTelephonyManager getDefault(android.content.Context);  
        method public abstract String getDeviceSoftwareVersion();  
        method public abstract int getDeviceType();  
        method public abstract String getGsmDeviceId(int);  
        method public abstract String getIccId(int);  
        method public abstract int getLac(int);  
        method public abstract com.dmyk.android.telephony.DmykAbsTelephonyManager.LocationMsg getLocationMsg();  
        method public abstract String getMacAddress();  
        method public abstract int getMasterPhoneId();  
        method public abstract long getMobileTotalBytes(int, long, long);  
        method public abstract String getModel();  
        method public abstract int getNetworkType(int);  
    }  
}
```



```
public class DmykImplTelephonyManager extends com.dmyk.android.telephony.DmykAbsTelephonyManager {
    ctor public DmykImplTelephonyManager(android.content.Context);
    method public android.net.Uri getAPNContentUri(int);
    method public String getCPUModel();
    method public String getCTAModel();
    method @RequiresPermission("android.permission.READ_PRIVILEGED_PHONE_STATE") public String getCdmaDeviceId();
    method @RequiresPermission(android.Manifest.permission.ACCESS_FINE_LOCATION) public int getCellId(int);
    method public int getDataState(int);
    method public String getDeviceSoftwareVersion();
    method public int getDeviceType();
    method @RequiresPermission("android.permission.READ_PRIVILEGED_PHONE_STATE") public String getGsmDeviceId(int);
    method @RequiresPermission("android.permission.READ_PRIVILEGED_PHONE_STATE") public String getIccId(int);
    method public int getLac(int);
    method public com.dmyk.android.telephony.DmykAbsTelephonyManager.LocationMsg getLocationMsg();
    method public String getMacAddress();
    method public int getMasterPhoneId();
    method public long getMobileTotalBytes(int, long, long);
    method public String getModel();
    method public int getNetworkType(int);
```

- 修改 proguard.flag 配置。在 vendor/sprd/carriers/cmcc/packages/apps/OpManager/proguard.flag 增加如下配置：

```
-dontwarn com.loopj.android.http.**
-dontwarn cn.dm.volley.**
-dontwarn cn.richinfo.dm.**
-dontwarn cz.msebera.android.httpclient.**
```

修改后截图参考如下：

```
-dontwarn org.apache.http.**
-dontwarn com.loopj.android.http.**
-dontwarn cn.dm.volley.**
-dontwarn cn.richinfo.dm.**
-dontwarn cz.msebera.android.httpclient.**
```

5.6 DM SDK 商用版本集成

SDK 调试版测试通过后，终端入库送测以及正式发售必须集成商用版 SDK（移动 DM SDK 商用版本通过移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取）。

步骤 1 User 版本开启 DM 功能：

修改文件 vendor/sprd/feature_configs/carriers/cmcc/config.mk DM 功能的配置信息。将 true 改为 false。

```
# DM statement start
DM_SUPPORT_ONLY_FOR_USERDEBUG := true ➡ false
```

步骤 2 商用版本 SDK 替换：

将 vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/中 dm_sdk_test_2.0.50_jdk1.8_v2.jar 替换为移动 DM SDK 商用版本。

步骤 3 移动发布的正式 sdk 若包含以下库文件，请依次替换。若不包含，请忽略此步骤。

- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/armeabi/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/x86/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/carriers/cmcc/packages/apps/ RegService/Android.mk 中 SDK 配置信息修改为商用版 SDK 名称:

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES :=libdm_sdk_test:libs/dm_sdk_test_2.0.50_jdk1.8_v2.jar
```

---结束

📖 说明

- 调试版本通过测试并在测试平台提交过测试报告的手机，相同 IMEI 号的手机在测试平台将无法上报信息（移动为控制测试平台无效数据上报，对已经通过测试平台测试并上传自测试报告的手机做 DM 信息上报屏蔽处理），如需要继续做测试，需要更换手机 IMEI 号。
- IMEI 号有效期为 1 个月，从第一次上报的时间开始计算。
- 集成 DM SDK 商用版本后，DM 相关数据将上报到移动 DM 商用平台，移动测试平台 <https://shiwei.fxitsbl.com/terminalservice> 无法查看相关数据，仅移动可查看。

Unisoc Confidential For hiar

6 Android 11.0 移动 DM 方案适配说明

考虑到 DM SDK 调试版本对平台软件功耗的影响较大，Android 手机 User 版本默认关闭移动 DM 功能，仅在 Userdebug 版本默认开启 DM 功能。建议在 Android 手机 Userdebug 版本上集成调试版 DM SDK 做 DM 功能测试，自测试通过后，User 版本再集成商用版本 DM SDK 送中国移动入库测试。

6.1 替换 AppKey

AppKey 一般由终端管理及服务平台生成，可通过测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取。如果没有，请联系移动接口人获取登录测试平台的帐号密码及 AppKey。

将文件 vendor/sprd/carriers/cmcc/packages/apps/OpManager/res/values/strings.xml 的 value 修改为获取的 Appkey。

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
<string name="appKey">A100000860</string>
</resources>
```

6.2 DM SDK 调试版本集成

步骤 1 登录移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 获取最新的 SDK 调试版本（紫光展锐平台默认集成的 SDK 调试版本为 dm_sdk_test_2.0.56_jdk1.8.jar）

步骤 2 将以下路径的 SDK 替换为移动测试平台获取的最新的 SDK

vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/dm_sdk_test_2.0.56_jdk1.8.jar

步骤 3 将以下路径的 android-async-http 库替换为移动测试平台获取的最新库文件

vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/android-async-http-1.4.10.jar

步骤 4 登陆移动测试平台获取并替换以下库文件：

- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/armeabi/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/x86/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/libs/x86_64/libaes-jni.so

步骤 5 将 vendor/sprd/carriers/cmcc/packages/apps/OpManager/Android.mk 中

- SDK 配置信息修改为替换的 SDK 名称
- android-async-http 库信息修改为替换的 android-async-http 库名称

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES :=libdm_sdk_test:libs/dm_sdk_test_2.0.56_jdk1.8.jar

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES +=android-async-http:libs/android-async-http-1.4.10.jar
```

---结束

6.3 APK 的适配

根据移动对 APK 包名和应用名的要求，需要做如下代码调整。

步骤 1 以下文件将 com.sprd.opm 替换为移动要求的实际包名 “com.终端厂家名称.regservice”

- frameworks/base/core/java/android/app/ApplicationPackageManager.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/applications/ApplicationsState.java
- frameworks/base/services/core/java/com/android/server/pm/permission/PermissionManagerService.java
- frameworks/base/packages/SettingsLib/src/com/android/settingslib/location/RecentLocationApps.java
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/AndroidManifest.xml
- vendor/sprd/carriers/cmcc/packages/apps/OpManager/src/com/sprd/opm/OPApplication.java
- frameworks/base/core/res/res/values/config_ex.xml
- vendor/sprd/platform/frameworks/base/unisoc-common/src/com/dmyk/android/telephony/DmykAbsTelephonyManager.java
- vendor/sprd/platform/frameworks/uniframework/base/telephony/java/com/android/internal/telephony/SimStateTracker.java

步骤 2 以下文件将 OpManager 替换为 RegService

```
- vendor/sprd/feature_configs/carriers/cmcc/config.mk
ifeq ($(strip $(DM_SUPPORT)), true)
FEATURES.PRODUCT_PACKAGES += \
    OpManager
endif

- vendor/sprd/carriers/cmcc/packages/apps/OpManager/Android.mk
LOCAL_SRC_FILES := $(call all-java-files-under, src)

LOCAL_PACKAGE_NAME := OpManager

LOCAL_CERTIFICATE := platform
```

步骤 3 修改 APK 目录结构 vendor/sprd/carriers/cmcc/packages/apps/OpManager

- OpManager 目录名替换为 RegService
- com/sprd/opm 目录结构替换为 com/终端厂家名称/regservice

说明

根据移动要求，固件版本号需对应手机的版本号：“设置”->“系统”->“关于手机”->“版本号”（对应属性：ro.build.display.id），且“固件版本号”字符串最大长度为 32 个字符。

---结束

6.4 参数配置

项目工程主 mk 配置送测手机“设备型号”、“ROM”、“RAM”信息。

添加如下信息：

```
PRODUCT_PRODUCT_PROPERTIES += \
    ro.cta.hardware.model = UNISOC \
    ro.cta.hardware.rom.size = 32G \
    ro.cta.hardware.ram.size = 4G
```

- ro.cta.hardware.model：工信部报备时填写的“设备型号”
- ro.cta.hardware.rom.size：项目 ROM 大小
- ro.cta.hardware.ram.size：项目 RAM 大小

6.5 DM SDK 商用版本集成

SDK 调试版测试通过后，终端入库送测以及正式发售必须集成商用版 SDK（移动 DM SDK 商用版本通过移动测试平台 <https://shipei.fxitsbl.com/terminalsdk> 获取）。

步骤 1 User 版本开启 DM 功能：

修改文件 vendor/sprd/feature_configs/carriers/cmcc/config.mk DM 功能的配置信息。将 true 改为 false。

```
# DM statement start
DM_SUPPORT_ONLY_FOR_USERDEBUG := true → false
```

步骤 2 商用版本 SDK 替换：

将 vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/中 dm_sdk_test_2.0.56_jdk1.8.jar 替换为移动 DM SDK 商用版本。

步骤 3 移动发布的正式 sdk 若包含以下库文件，请依次替换。若不包含，请忽略此步骤。

- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/arm64-v8a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/armeabi-v7a/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/armeabi/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/x86/libaes-jni.so
- vendor/sprd/carriers/cmcc/packages/apps/RegService/libs/x86_64/libaes-jni.so

步骤 4 将 vendor/sprd/carriers/cmcc/packages/apps/RegService/Android.mk 中 SDK 配置信息修改为商用版 SDK 名称：

```
include $(CLEAR_VARS)

LOCAL_PREBUILT_STATIC_JAVA_LIBRARIES := libdm_sdk_test:libs/dm_sdk_test_2.0.56_jdk1.8.jar
```

----结束

说明

- 调试版本通过测试并在测试平台提交过测试报告的手机，相同 IMEI 号的手机在测试平台将无法上报信息（移动为控制测试平台无效数据上报，对已经通过测试平台测试并上传自测试报告的手机做 DM 信息上报屏蔽处理），如果还需要继续做测试，需要更换手机 IMEI 号。
- IMEI 号有效期为 1 个月，从第一次上报的时间开始计算。
- 集成 DM SDK 商用版本后，DM 相关数据将上报到移动 DM 商用平台，移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 无法查看相关数据，仅移动可查看。

Unisoc Confidential For hiar

7

DM 自测试说明及常见问题

7.1 概述

DM 的适配工作完成后，即可开展相关的测试工作。请先登陆移动测试平台 <https://shipei.fxitsbl.com/terminalservice>，仔细阅读移动测试平台“终端管理测试平台说明”。同时请务必先仔细阅读中国移动测试平台发布的相关文档。

7.2 集成测试常见问题

Q1：心跳/应用信息上报异常重试周期？

A：默认 10 分钟*N（N 为重试次数，N 最大值默认为 10）。

Q2：DM 各业务周期如何修改？

A：心跳周期/应用上报周期/异常最大重试次数/异常重试时间，均可通过移动测试平台 <https://shipei.fxitsbl.com/terminalservice> 进行设置。

Q3：为什么 GPS 经纬度无法获取？

A：GPS 经纬度获取条件：

- GPS 开关开启
- Wifi 或数据网络开启且连接正常（可正常上网）
- GPS 可正常搜星（室外）

Q4：为什么 GPS 无法获取地理位置？

A：地理位置（省份名称、地市/区县、地址）信息需要第三方支持。此功能项不影响 DM 入库测试。

Q5：为什么有些应用信息未上报？

A：系统集成的第三方应用（比如和飞信/和彩云…），如果开机未运行过，不会在手机中生成相关应用信息，DM SDK 无法采集。因此，如果要采集到系统集成的第三方应用信息，请确保该应用至少正常运行过一次。

Q6：为什么长时间无心跳上报？

A：请先确认刷机首次开机前是否已经通过移动测试网站配置心跳周期。如未配置，请优先通过移动测试网站配置心跳周期后重新刷机验证。

Q7: 为什么 Android10.0 以上版本, 手机根目录下找不到 DMLog?

A: 新版本 DM sdk (dm_sdk_test_2.0.56) 为解决 Android10.0 及以上版本存储设备访问限制, DMLog 迁移到/storage/emulated/0/Android/data/com.***.***/files 目录下, 其中 com.***.***为 DM 应用包名。

7.3 集成测试注意事项

- 首次开机测试前请务必先登录移动 DM 测试服务器“系统参数配置”中配置对应 IMEI 的相关参数
- 测试异常重试的相关 case 时, 心跳和应用信息上报实际时间要确保同时大于异常重试时间的(1+2+...+Nmax) 倍。
- 请务必确保测试手机的 IMEI 号独一无二且有效 (IMEI 有效期为一个月, 从第一次上报的时间开始计算)。
- 移动平台参数配置为避免对测试结果造成困扰, 测试参数设置“配置维度”务必选择“IMEI 号”, 即仅对特定 IMEI 的手机起作用。
- 确保手机系统时间为当前时间。
- 心跳周期验证请确保数据网络或 wifi 连接稳定。
- 应用信息上报周期验证请确保连接稳定的 wifi (wifi 连接通畅)。
- 移动测试平台延时属正常现象, 一般是 10 分钟以内。如果长时间测试平台未更新, 可先查看 DMLog 是否正常上报。如果正常上报, 则说明是移动测试平台同步延迟导致, 建议咨询中移动平台。如果未正常上报, 请根据实际问题分析对应 log。
- 如果需要实时查看心跳/APN/应用信息等上报状态, 可将手机连接 pc, 输入如下 adb 命令查看:

```
adb logcat -v threadtime -s RegisterMsgReport HeartbeatMsgReport PostApplyMsgReport ApnInfoReport currentRetryCount0 test DMServiceTimeRelatedTrafficMsgReport
```
- 测试前建议开启 slog/ylog (同时 modem log 勾选上), 出异常时提供 slog/ylog+DMLog, 并尽可能说明出现异常的时间点。

7.4 功耗测试建议

由于移动 DM SDK 调试版本心跳上报周期以分钟为单位, 会使功耗变大, 因此建议不要使用带有调试版 SDK 的软件平台测试功耗。

移动 DM SDK 商用版本心跳上报周期以天为单位, 对功耗影响相对较小, 因此建议集成移动 DM SDK 商用版本后再做完整的功耗测试。

说明

由于移动 DM 第一次开机上报和上报失败 (如网络连接失败等) 重试等会产生瞬间功耗增大, 项目做功耗排查时, 建议先关闭移动 DM 功能以避免对项目功耗排查的干扰。