

GMS Express 2.0 项目介绍

文档版本

V1.2

发布日期

2020-03-31

版权所有 © 紫光展锐科技有限公司。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，紫光展锐均不负责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。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。

紫光展锐科技有限公司

前言

概述

本文档主要介绍展锐 **GMS 2.0** 方案。

读者对象

本文档主要适用于 **GMS 2.0** 项目的相关人员，以及采用 **GMS 2.0** 方案的客户。

缩略语

缩略语	英文全名	中文解释
GMS	Google Mobile Service	
GMS 2.0	GMS Express 2.0	

修改记录

文档版本	发布日期	修改说明
V1.0	2020-03-09	第一次正式发布。
V1.1	2020-03-20	更新 system.img 的命名规则和发布流程，补充 product 分区应用的名称
V1.2	2020-03-31	删除原 5.1 节 Target 包产生 ota 升级包，增加第 6 章 GMS2.0 项目特有 feature

目 录

1	GMS 2.0 项目 Overview	4
1.1	GMS Express 2.0 项目的优点	4
1.2	GMS Express 2.0 Partitions	5
1.3	基础合作框架	8
1.4	简单参考样例	9
1.5	各阶段预计周期	11
1.6	BUG fix 流程	12
2	Finger Print 问题注意事项	13
3	编译版本注意事项	14
4	Image 签名要求	15
5	OTA 升级包制作	16
5.1	通过 pac 包产生升级包	16
6	GMS 2.0 项目特有 feature	17

1 GMS 2.0 项目 Overview

GMS 2.0 项目是 Google, Unisoc 以及有意愿参与该项目的 Unisoc 客户三方合作的项目。

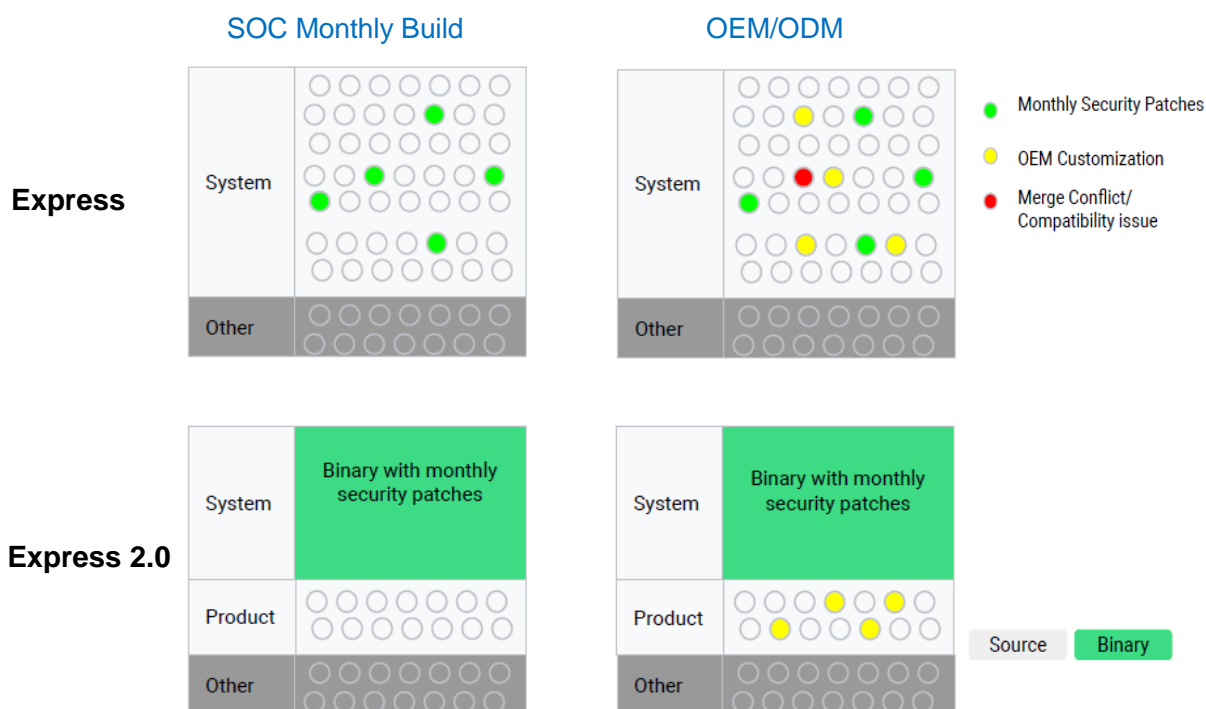
Unisoc 负责将 GMS 应用, Unisoc 自研应用和客户定制应用移动到 product 分区, 并解决应用移动到 product 分区后出现的不兼容问题。Unisoc 给参与该项目的客户发布 binary system image。要求参与 GMS 2.0 项目的客户, 使用 Unisoc 发布的 system image 制作版本, 如有定制需求需要和 Unisoc 协作实现。

1.1 GMS Express 2.0 项目的优点

- 优化 Android 碎片化;
- 优化 Android MR/SMR 升级速度;
- 即提高了系统安全又可减小 ODM/OEM 维护和测试成本;
- Security updates 将由 SOC 来完成, 除了 OEM/ODM 自己所修改引入的问题;
- Google 可以与 SOC 一起进行质量管控;
- 如果系统发生任何问题, Google 会与 SOC 一起进行解决, ODMs 只需要更新相关 binary 文件;
- 可以和谷歌 Variant build approval policy 结合, 减少合规测试的成本 (长远规化);
- 随着 GMS 2.0 的成熟, 后续客户只需进行简单的应用部分的定制, 其余大部分工作都会由 SOC 来完成, 会大大节省项目的量产周期;

1.2 GMS Express 2.0 Partitions

图1-1 分区图



根据和 Google 的沟通结论，Unisoc 在 GMS 2.0 项目上移动到 product 分区应用的原则是：

- 所有的 GMS 应用；
- 所有的 unisoc 自研应用；
- 其他客户有定制需求的应用；

目前 Unisoc 已经移动到 product 分区的应用列表如下，后续如有更新也会及时同步，客户基于以下应用的客制化需求可直接自行修改。

表1-1 移到 product 分区的应用列表

包名	应用名称
com. sprd. firewall	来电屏蔽
com. sprd. cameracalibration	相机标定与验证工具
com. android. camera2	相机
com. android. fmradio	FM 电台
com. android. soundrecorder	录音机
com. sprd. engineerinternal	EngineerInternal
com. sprd. engineermode	EngineerMode
com. android. providers. partnerbookmarks	
com. android. dreams. phototable	照片屏幕保护程序
com. sprd. autoslt	SLT
com. sprd. commlog	
com. android. deskclock	时钟
com. spreadst. validator	芯片筛选
com. android. stk	SIM 卡工具包
com. sprd. systemupdate	系统更新
com. sprd. uplmnsettings	UPLMN Preference
com. sprd. validationtools	工厂测试工具
com. spreadst. validdat	
com. android. carrierconfig	
com. android. cellbroadcastreceiver	紧急警报
com. android. dialer	电话
com. android. emergency	急救信息
com. spreadtrum. ims	ImsApp
com. android. providers. media	
com. sprd. omacp	OtaOmaHandler
com. sprd. powersavemodelauncher	PowerSaveModeLauncher
com. sprd. quickcamera	
com. android. sprd. telephony. server	
com. android. settings	设置
com. android. settings. intelligence	Settings Suggestions
com. android. providers. calendar	
com. android. contacts	通讯录
com. android. providers. contacts	
com. unisoc. storageclearmanager	存储空间管理
com. android. storagemanager	存储空间管理器
com. android. systemui	系统界面
com. android. wallpapercropper	剪裁壁纸
com. android. wallpaper	壁纸

如果客户有自己的应用需要移动到 product 分区，可参考如下代码修改：

Android.mk 中编译的模块

- ✓ 在模块内添加: `LOCAL_PRODUCT_MODULE := true`

Android.bp 中编译的模块

- ✓ 在模块里添加: `product_specific: true,`

Property 编译到 product 分区

- ✓ `FEATURES.PRODUCT_PRODUCT_PROPERTIES += \`

配置文件编译到 product 分区

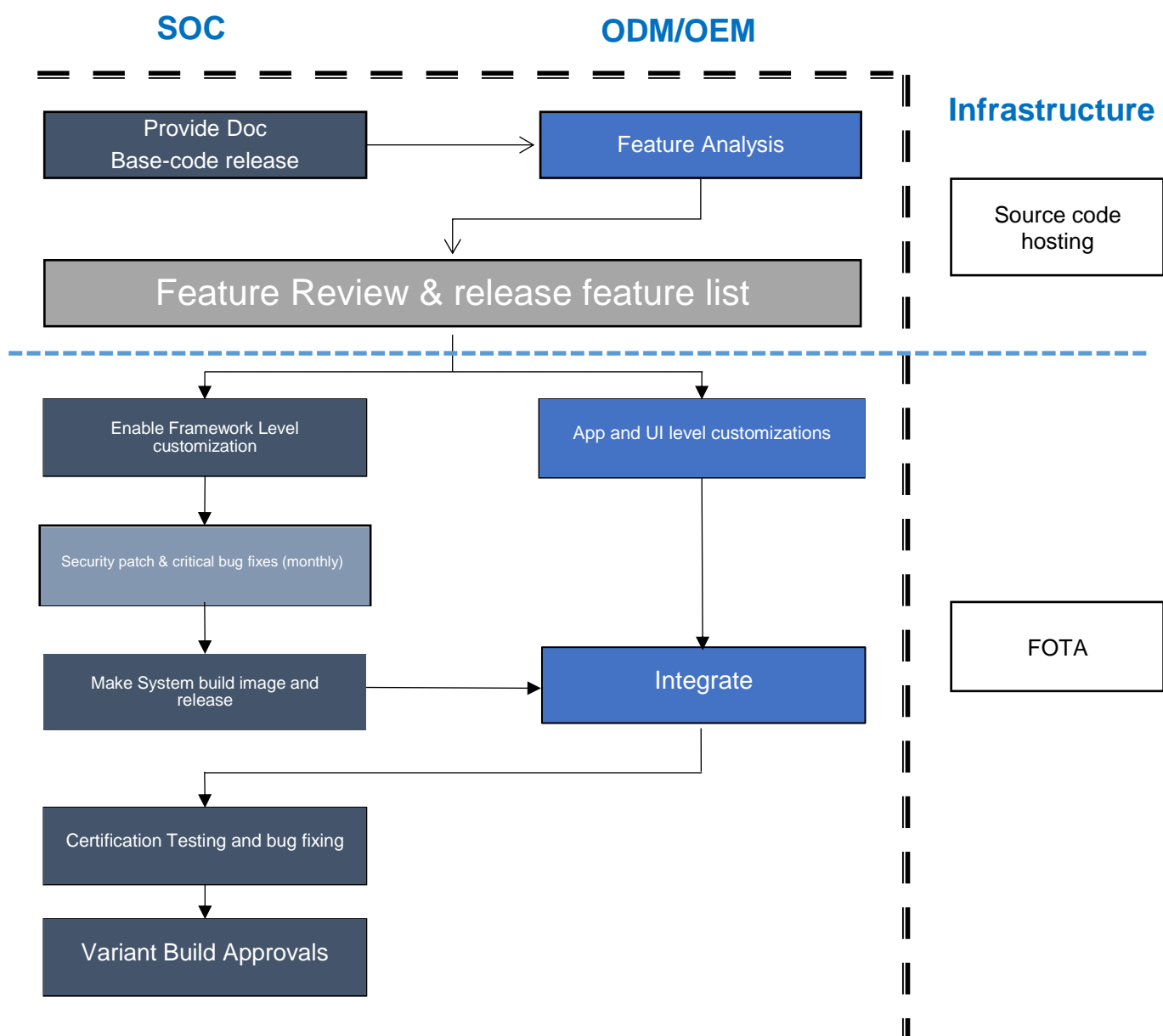
- ✓ `$(TARGET_COPY_OUT_PRODUCT)` 宏对应的的 product 目录

下面章节重点描述 GMS2.0 项目的注意事项，合作客户需要重点关注。

1.3 基础合作框架

对于 Unisoc 能收集过来的需求都是需要符合 GMS Express 2.0 基本理念，并且相对 common 或者有继承和维护的价值，如果只是特别单一性的需求，Unisoc 可能无法进行继承，也无法使用 single system image 来完成，对于已继承并且只有部分客户使用的，我们也会指导客户可以在 product 分区配置 run time feature flag 来控制此 feature.

图1-2 分工图



1.4 简单参考样例

- RRO

```
637 // Environment.getRootDirectory() is a fancy way of saying
    ANDROID_ROOT or "/system".

638 File confFile = new File(Environment.getRootDirectory(),
    PARTNER_APNS_PATH);

639 File oemConfFile = new File(Environment.getOemDirectory(),
    OEM_APNS_PATH);

640 File updatedConfFile = new
    File(Environment.getDataDirectory(), OTA_UPDATED_APNS_PATH);

641 File productConfFile = new
    File(Environment.getProductDirectory(), PARTNER_APNS_PATH);

642 confFile = pickSecondIfExists(confFile, oemConfFile);

643 confFile = pickSecondIfExists(confFile, productConfFile);

644 confFile = pickSecondIfExists(confFile, updatedConfFile);

645 return confFile;

646 }
```

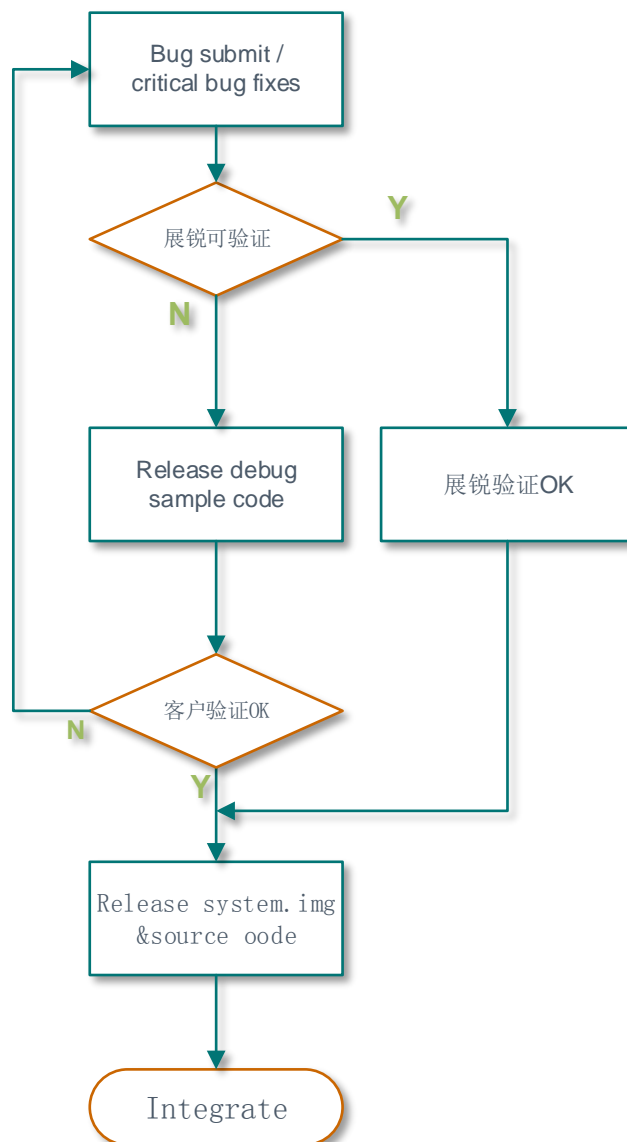
1.5 各阶段预计周期

表1-2 周期表

Stage	Unisoc GE2.0	Estimated lead time (First GE2.0 device)
Preparation Stage	<p>OEM/ODM provides feature list to Unisoc</p> <p>Unisoc reviews the feature list</p> <p>Unisoc moves the required features from system partition to product partition if needed.</p> <p>Or ODM/OEM upstream related to SoC system partition if those feature cannot move to product partition</p>	<p>Two weeks</p> <p>(Based on simultaneous communication and feedback from both parties)</p>
Dev. Stage	<p>Unisoc merge required security patch, functional patch, and internal patches to system.img</p> <p>Release system.img and other partition' s codes to OEM/ODM</p> <p>ODM/OEM modifies the codes in product partition for customization</p> <p>IR build approval</p>	<p>According to the list of customer features</p> <p>And give an estimated development completion timeline within one week after review the feature list</p>
MR/SMR Stage	<p>Unisoc integrate required security patch, function patch and internal patches to system.img and run xTS</p> <p>Release system.img to ODM/OEM</p>	<p>Security patch & critical bug fixes monthly release.</p>

1.6 BUG fix 流程

图1-3 bug 处理流程



2

Finger Print 问题注意事项

GMS 2.0 方案，使用 product 分区 Finger Print 替换 system 分区 Finger Print 的解决方案，需要遵守下面两个原则：

1. 必须同时编译 product image 和 vendor image，保证 product 分区和 vendor 分区的 finger-print 相同；
2. 做 OTA 升级包时，必须同时更新 product 分区和 vendor 分区的 finger print，保证 OTA 升级后 product 分区和 vendor 分区 finger print 随着 OTA 升级更新。

3

编译版本注意事项

GMS Express 2.0 项目要求，客户要使用 Unisoc release 的 system image。Unisoc 在 release system image 的同时也会 release 源码。客户在使用源码编译 product image 时，**必须保证不能修改编译到 system image 中的源码**，否则可能会出现 product image 中编译时生成的 oat 文件无法使用问题。

这是因为首次开机时虚拟机会使用 product 分区中应用 oat 文件中的 checksum 和 system 分区中 boot.oat 的 checksum 进行比较，如果有编译到 system image 相关代码被修改，会导致 checksum 值不相等，虚拟机就在首次开机时对 product image 中的应用重新做 dex 优化，首次开机时间加长。影响 oat 文件中 checksum 的参数包括 instruction_set_，instruction_set_features_bitmap_ 等等，如果客户拿到 unisoc 的代码没有对 system 做过修改，这些值和 unisoc 给的 system image 中应该相同，也就是最终的 checksum 能够匹配上。

目前在 GMS 2.0 代码推送中，Unisoc 已经将 system image 打包到每个工程的属性包中并随版本一起发布，客户只需要拿到 Unisoc release 的属性包（位于 out/target/product/\${target_product}/ 目录下）后进行解压，就可以进行使用。

unisoc release 的 system image 命名规则如下：

unisoc_\${TARGET_ARCH}-\${TARGET_BUILD_VARIANT}-system.img

例子: unisoc_arm64-userdebug-system.img

4 Image 签名要求

System.img 的发布流程如下：

- 1) Unisoc 平台基于 Unisoc key 编译 System.img;
- 2) 客户拿到 System.img 后，使用脚本对 System.img 中的所有 APK 重新签名;
- 3) 客户在拿到 Unisoc 发布的 system.img 后，需要使用 resign_image.py 脚本进行二次签名，在运行该脚本前，首先需要对代码进行一次整体编译。然后再运行脚本，使用方法如下：

```
./build/tools/image_tools/resign_image.py /path/to/system.img /path/to/mntpoint /  
path/to/top/sourcedir /path/to/apksigning/keys /path/to/systemsigning/keys
```

这里几个参数含义如下：

/path/to/system.img: 指定需要重新签名的 system.img 路径

/path/to/mntpoint: 指定 system.img 的挂载点

/path/to/top/sourcedir: 指定代码的根路径

/path/to/apksigning/keys: 指定客户 APK 对应的 key 的路径

/path/to/systemsigning/keys: 指定对 system.img 重新签名时对应的 key 的路径

在 resign_image.py 中有一组默认参数配置，可以参考

```
if len(sys.argv) < 6:  
  
    image_path = os.environ["OUT"] + "/system.img"  
  
    mount_dir = "/mnt/systemdir/"  
  
    build_top = os.environ["ANDROID_BUILD_TOP"]  
  
    key_path_prefix = "/vendor/sprd/partner/CusKeys/"  
  
    image_key_path = "/vendor/sprd/proprieties-  
source/packimage_scripts/signimage/sprd/config/rsa4096_system.pem"
```

- 4) 重签 system image 后，super.img 和 vbmeta_system.img 会重新生成，做 pac 包时需要使用重新生成 vbmeta_system.img 和 super.img (还在原目录下)。

5 OTA 升级包制作

由于我司释放的固定的 system.img 中会添加文件，而 release 代码中不会包含该文件产出，在使用 target 包做升级包时会因检索不到该新添加文件而失败，故该项目只能使用 pac 包产生 ota 升级包。

5.1 通过 pac 包产生升级包

预制 system.img 重新签好名，并做出能开机合格 pac 包后，可直接通过 pac 包产生 ota 升级包。

步骤为：

1. 工程 source lunch，并执行 make otatools，会产生 otatools，在目录 out/target/product/s9863alh10/otatools.zip，该工具可一次产生后放至别处一直使用。
2. 将该工具在 linux 环境下解压并按照里面 Guide.txt 执行命令产生升级包，也可直接使用如下命令：

■ 制作 OTA 整包

```
./build/make/tools/releasetools/ota_from_pac_files -p product_name -k sign_key -t pac_target full_update.zip
```

■ 制作 OTA 差分升级包

```
./build/make/tools/releasetools/ota_from_pac_files -p product_name -k sign_key -b pac_base -t pac_target delta_base-to-target.zip
```

■ 制作降级差分升级包

```
./build/make/tools/releasetools/ota_from_pac_files -d -p product_name -k sign_key -b pac_newer_build -t pac_oldrget delta_update_newer-to-old.zip
```

■ 其中各参数意义：

1 sign_key:	user 版本为 build/target/product/security/release/releasekey
2	userdebug 版本为 build/target/product/security/testkey
3 delta_base-to-target.zip:	基准版本到目标版本的差分 ota 升级包
4 full_update.zip:	目标版本整包
5 full_update_repart.zip:	目标版本分区变更的 ota 整包
6 product_name:	base 对应的项目名称，只能是小写：shark13 sharkle pike2 shark15 roc1
7 pac_base:	基准版本 pac 包，即可为.gz 的压缩格式，也可为.pac 的非压缩格式
pac_target:	目标版本 pac 包，压缩非压缩皆可

6

GMS 2.0 项目特有 feature

为了区分 GMS 项目与非 GMS 项目，在 GMS 项目上增加了特有的 feature:

com.google.android.feature.GMSEXPRESS_V2_BUILD

检查方式:

- 1、 在手机/system/etc/permissions/目录下有文件 gms_express_feature.xml

```
phone:/system/etc/permissions # ls | grep gms
```

```
gms_express_feature.xml
```

- 2、 在手机 adb shell 下执行命令: pm list features | grep GMSEXPRESS_V2_BUILD

```
phone:/ # pm list features | grep GMSEXPRESS_V2_BUILD
```

```
feature:com.google.android.feature.GMSEXPRESS_V2_BUILD
```

GMSEXPRESS_V2_BUILD 这个 feature 由展锐配置。目前版本已经包含此 feature。

根据 google 的说明，如果客户的设备满足 GMS Express Plus 的要求，还需要配置以下 feature:

com.google.android.feature.GMSEXPRESS_PLUS_BUILD

这个 feature 必须放到 product 分区。路径: /product/etc/permissions/gms_express_plus_feature.xml

“GMS Express Plus Device” means a device which meets the following conditions:

- a) Uses the baseline GMS Express Build; and
- b) Default intents in core app categories from the baseline GMS Express build are not overridden.
- c) The Google “Feed” product will remain on the -1 screen(Home left) after it’s been integrated into the baseline GMS Express build