密级状态：绝密( ) 秘密( ) 内部( ) 公开(√)

Rockchip Android GMS用户配置指南

文件状态:
[ ] 草稿
[√] 正式发布
[ ] 正在修改

文件标识:	RK-YH-YF-237
当前版本:	V2.1
作者:	卞金晨
完成日期:	2023-04-01
审核:	金华君
审核日期:	2023-04-01

版本号	作者	修改日期	修改说明	备注
V1.0	卞金晨	2020-02-17	发布初稿	
V1.1	卞金晨	2020-06-30	添加user版本相关说明	
V1.2	卞金晨	2021-02-22	添加对Android 11的支持	
V1.3	卞金晨	2021-03-29	添加对测试时机器锁定状态的描述	
v1.4	蔡建清	2022-02-18	补充attestation key的相关说明	
v1.5	卞金晨	2022-03-28	添加对Android 12的支持	
v1.6	卞金晨	2022-03-30	添加对Android 12 incrementalfs的说明 更换固件、Apex签名的说明 添加目录	
v1.7	卞金晨	2022-04-01	添加对adb/fastboot的版本说明 更正一些Android 12的细节	
v2.0	卞金晨	2023-04-01	添加对Android 13的支持	

文档问题反馈：kenjc.bian@rock-chips.com

目录

名词解释

- GMS (Google Mobile Service)
  Google移动设备服务，包含Google服务框架，Play商店，Chrome浏览器等一系列应用。
- MADA (Mobile Application Distribution Agreement)

  主要针对手机平板移动设备，满足手持或平板设备类型的所有CDD要求，屏幕尺寸在3.3英寸~18英寸且**设备必须带电池。**
- EDLA (Enterprise Device Licensing Agreement)

  和MADA基本一致，放宽了对屏幕电池的要求，可以做18英寸~70英寸屏幕/无电池的设备，**但内存必须大于2G**，且产品不能直接面向消费者。
- 3PL
  第三方实验室，它会根据Google CDD要求以及GMS Requirments，协助Google在全球不同区域的GMS配置要求对产品进行认证。常见的有富士康/哈曼/风河/和硕等，一般来说自行联系，也可以找业务引荐。
- Google Partner
  签署MADA协议/EDLA协议成为Google partner后，才能够访问Google Partner文档，才能够联系3PL进行GMS的申请及测试。
- EEA
  卖往欧盟国家的设备需要通过EEA的GMS认证，代码中配置的EEA类型则需要针对不同的情况和签订的协议EMADA (European Mobile Application Distribution Agreement)进行配置。有关EEA的相关说明，请参考Google Partner文档：
  https://support.google.com/androidpartners_gms/answer/9071728?hl=en
- xTS
  执行对应目录下的binary后，会进入xTS-tradefed命令行，此后执行下列的测试命令或retry命令即可。
  具体可参考Google官方网站：https://source.android.com/compatibility/cts/run
- retry
  指在已有结果的基础上继续测试，进入tradefed后，执行`l r`即可查看之前的测试结果，第一列为`session id`，执行retry时接参数`--retry session_id`即可续测指定的结果。**retry测试时不要添加多余参数，例如：**`--retry-type`

- 单测
  指单独测某些测试项，用于debug，测试后会产生测试结果及log，分别位于`results`及`logs`目录下。

  ```
  报错：android-cts/logs/2023.05.04_14.13.15/inv_1787362909717145343 4/host_log_***.txt
  日志：android-cts/logs/2023.05.04_14.13.15/inv_1787362909717145343 4/device_logcat_test_PH0BEXGDPG_***.txt
  ```

测前需知

想要在设备搭载Google移动服务，必须通过GMS/EDLA认证，通过认证后能够享受Google的移动服务，设备的安全性及兼容性也能得到极大的保证。想要通过GMS/EDLA认证，必须拥有资质，一般来说有2种方式能够通过GMS认证，使设备最终能够搭载GMS服务：

- 自己拥有MADA/EDLA资质
  这种情况下，会专门有一个Google的联系人，有关Partner的文档权限问题以及各种账号问题都能够咨询他，想要通过GMS认证时需要联系Google联系人或是联系3PL来申请。通过本文档的详细配置和自测通过所有xTS测试后，就能够搭载GMS服务了。
- 自己没有MADA/EDLA资质
  只有具有MADA/EDLA资质才能申请并通过GMS认证。没有MADA资质时，请先联系具有ODM MADA资质的公司，可以自行联系或通过业务同事引荐。

注意事项

- 文档中绝大部分的外部链接，都需要MADA/EDLA权限才可以访问。
- 目前2G及以下容量的DDR，均只能以Android Go过认证，Android 13及以上版本不再允许2G以下容量DDR过认证。体现在编译时选择`lunch *go`，例如:

  ```
  lunch rk3326_rgo-user
  ```

- 关于PX30/RK3568过GMS认证
  PX30为RK3326的工业版本，功能大致相同，且Google备案处仅有RK3326，所以如果想要用PX30去过认证，请使用RK3326；RK3568同属RK356x系列，使用RK3566或RK3568均可；**RK3562和RK356x不是同一系列的芯片，注意不要以RK356x申请。**以PX30为例：

1. 向实验室提交sub-license申请时，与RK3326保持一致。
2. 编译请直接使用RK3326，例如PX30-2GB DDR版本，`lunch rk3326_rgo-user`
3. 如需使用一键编译(`例如：./build.sh -UCA`)，请先修改makefile中的`PRODUCT_UBOOT_CONFIG`

   ```
   device/rockchip/rk3326/BoardConfig.mk
   PRODUCT_UBOOT_CONFIG := px30
   ```

- EDLA相关说明
  EDLA设备大多数使用USB Camera，而非mipi这类的内置Camera。这种情况下，3PL测试时不会指定某一种Camera型号，而CTS中测试相对比较严格，目前已知已经通过认证的sensor型号可以查看附录A。

测前申请

为了加快GMS认证测试进展，在进行测试前请先申请以下内容：

1. 向3PL申请sub-license，相关内容请参考Keymaster & Optee部分详细说明
2. 向3PL申请keybox，相关内容请参考Attestation key部分详细说明
3. 向3PL申请GTS测试APE_API_KEY，相关内容请参考GTS测试详细说明
4. 向Google申请android-partner-api@company.com账号并申请RKP，相关内容请参考RKP部分详细说明

认证窗口

Google严格把控**首次**过GMS/EDLA认证产品的Android版本，相关Android版本过GMS/EDLA认证窗口截止日期见网址GMS approval windows

例如：Android12L版本AOSP释放日期2022年3月7日，过GMS认证窗口截止日期2023年2月28日，过EDLA认证窗口截止日期2023年12月31日

认证周期

GMS一般为平板设备，硬件设计较为简单，一般获取稳定版本SDK，且在**整机做出来，机器没有明显的死机问题**后，处理问题所需周期约3-4周。

Android 13及以上版本要求使用GKI，此时内核中的驱动需要额外注意，如果有用到默认abi列表之外的接口，则需要提交到Google的服务器，审核+合并通常需要3-4天。Google每月15号会截断当月的版本进行测试，释放签名版`boot.img`则通常在每月的月末。因此，如果你的提交在15号之后，最快需要等约40天才能拿到release版本。因此，**需要额外关注时间节点。**测试阶段可临时下载Google每日构建PAB版boot.img。

Android 13及以上版本需要使用RKP，涉及步骤比较复杂，且需要Google端协助，请尽量早申请，**以免耽误项目周期。**参考有关RKP部分的详细说明。

EDLA设备通常硬件设计比较复杂，涉及到的驱动更多，GKI部分的时间很有可能要拉长4-6周的时间。参考有关GKI部分的详细说明。

测试必须使用整机测试，EDLA设备中如果配套多个屏幕，则按照最大尺寸屏幕进行测试。通常所有测试如果同步进行且机器无死机等问题，5天基本能测试完成。顺利测试所需时间可参考：

测试项	建议整机数量	测试周期
CTS	3+	2-4天
CTS-ON-GSI	2	2
VTS	1	1
CTS-V	1	1
GTS	1	2
GTS-V	1	1
APTS	1	1
STS	1	1

谷歌包获取

GMS认证需要以下三个包：GMS包、mainline包、prebuilts包

谷歌包	预置目录
GMS包	vendor/partner_gms
mainline包	vendor/partner_modules
prebuilts包	prebuilts/module_sdk

1. 拥有MADA/EDLA资质的客户，建议直接使用我们的Express分支，同步下来的GMS包是符合GMS Express Plus规定的，无需做太多修改。同步方法请移步至文档结尾处的GMS Express章节。
2. 没有MADA/EDLA资质或想要自己下载GMS包的客户，请按以下步骤操作：

- 常规GMS包请到以下网址直接下载，解压后放到`vendor/partner_gms`：
  `https://docs.partner.android.com/gms/building/integrating/gms-download`
- mainline包请到网盘下载，Android 11及以上版本没有打包好的压缩包，只有说明文档，需要自己使用repo进行同步，网盘/说明文档地址如下：
  `https://drive.google.com/drive/folders/1gxxMalTC99Y2SvyRDUfiDFvY_k4OzjO5`
  例如：

```
# 下载manifest文件
repo init -u https://partner-android.googlesource.com/platform/manifest -b
r-aml-prebuilt-release
# 开始同步，同步前，请先检查.repo/manifests/default.xml中是否有内容
repo sync -c -j8
```

下载指定版本（一般测试都需要使用最新的Approved版本），查看当前最新approved的文档的`Release Summary`，找出`git tag`，例如：

```
git tag mainline_m_2020_dec_preload_5
```

此时，需要手动将manifest中的默认同步分支修改为对应的tag：

```
diff --git a/default.xml b/default.xml
index 7be1f37..01ccba9 100644
--- a/default.xml
+++ b/default.xml
@@ -4,7 +4,7 @@
   <remote  name="ohd"
            fetch=".."
            review="https://partner-android-review.googlesource.com/" />

-  <default revision="r-aml-prebuilt-release"
+  <default revision="refs/tags/mainline_m_2020_dec_preload_5"
            remote="ohd" />

   <!-- Google-signed Mainline prebuilt module projects !-->
```

- Android 12及以上版本还需要prebuilts包，请务必下载和mainline包相同的版本，否则编译会报错。例如：

```
vendor/partner_modules
T1003604/mainline_m_2022_feb_preload_5

prebuilts/module_sdk
T1003604/mainline_m_2022_feb_preload_5
```

编译报错示例：

```
ERROR: Hidden API flags are inconsistent:
< prebuilts/module_sdk/IPsec/current/hiddenapi/all-flags.csv
> out/soong/hiddenapi/hiddenapi-flags.csv

ERROR: Hidden API flags are inconsistent:
< prebuilts/module_sdk/Media/current/hiddenapi/stub-flags.csv
> out/soong/hiddenapi/hiddenapi-stub-flags.txt

ERROR: Hidden API flags are inconsistent:
< prebuilts/module_sdk/Media/current/hiddenapi/stub-flags.csv
> out/soong/hiddenapi/hiddenapi-stub-flags.txt

ERROR: Hidden API flags are inconsistent:
< prebuilts/module_sdk/Connectivity/current/hiddenapi/all-flags.csv
> out/soong/hiddenapi/hiddenapi-flags.csv

ERROR: Hidden API flags are inconsistent:
< prebuilts/module_sdk/Media/current/hiddenapi/all-flags.csv
> out/soong/hiddenapi/hiddenapi-flags.csv

ERROR: Hidden API flags are inconsistent:
< prebuilts/module_sdk/Wifi/current/hiddenapi/stub-flags.csv
> out/soong/hiddenapi/hiddenapi-stub-flags.txt
```

- 集成mainline时，请务必确认mainline的文档，文档中提到的补丁必须在SDK中存在，否则编译报错。
- GMS包版本信息的获取：`getprop | grep ro.com.google.gmsversion`
  GMS包版本有效期查询网址：
  `https://docs.partner.android.com/gms/building/integrating/gms-download`

测试包获取

测试包版本要求：一般要求用Google官方发布的最新测试包(测试包版本要求在线查询)
测试包下载链接查看附录B，或从Rockchip安全补丁FTP服务器目录`GMS-Test-Suite/xTS`获取但可能不是最新

MADA/EDLA认证测试包括以下模块：

模块	固件类型	说明
CTS	user固件	
CTS-ON-GSI	user固件	烧写Google签名system.img

模块	固件类型	说明
GTS	user固件	
STS	userdebug固件	
VTS	user固件	烧写Google签名system.img和固件boot-debug.img
CTS-Verifier	user固件	
GTS-Verifier	user固件	仅Android 13及以上版本需要
Checklist	user固件	EDLA认证也需要测试Checklist
BTS	user固件	
GOATS	userdebug固件	仅Android 11及以下Go版本需要
APTS	userdebug固件	仅Android 12及以上Go版本需要

- PAB包：每日构建版本(可下载谷歌包和测试包，用于排查某些fail项Google是否已经修复)

GMS SDK参数设置

内核编译

在编译kernel时，请确认使用Clang进行编译（一键编译命令`./build.sh -CK`），否则VTS测试会产生如下fail:

```
vts_kernel_toolchain
KernelVersionTest#IsClang
```

如果是编译Android Go版本，请使用android-10-go.config代替android-10.config，Android 11版本请把10更换为11，Android 13则更换为android-13.config例如：

```
make ARCH=arm64 rockchip_defconfig android-10-go.config rk3326.config
make ARCH=arm64 rockchip_defconfig android-11-go.config rk3326.config
```

如果提示rkxxxx.config不存在，可以不加。

以下部分涉及Android编译，注意到device目录中检查device/rockchip/rkxxxx/BoardConfig.mk的配置

Widevine

注意只有RK3588/RK3399/RK356x已经支持Widevine L1，而平板设备一般L3即可满足要求。如果需要使用L1，请与我司联系获取最新L1支持情况以及L1补丁。

```
BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 3
```

EEA

针对欧盟出售的设备，只有做EEA的设备才需要配置，其他设备请留空

```
BUILD_WITH_EEA := true
BUILD_WITH_EEA_TYPE := type1 (根据自己的EEA类型配置)
```

GMS包使能

```
BUILD_WITH_GOOGLE_MARKET := true
# 含radio的设备（即带有4G通讯功能）请配置以下配置，其余设备保持false
BUILD_WITH_GOOGLE_MARKET_ALL := true
# 编译user固件时，请把下面的宏设成false，RK的压力测试工具将会移除，否则无法正常启动。
PRODUCT_HAVE_RKAPPS := false
```

如果需要使用Android Go的2G版本，请`device/rockchip/common/device.mk`中添加这个修改：

```
diff --git a/device.mk b/device.mk
index 63bb4ac..4340170 100644
--- a/device.mk
+++ b/device.mk
@@ -899,7 +899,7 @@ ifeq ($(strip $(BUILD_WITH_GOOGLE_MARKET)), true)
     OVERRIDE_TARGET_FLATTEN_APEX := true
     PRODUCT_PROPERTY_OVERRIDES += ro.apex.updatable=false

     # 2G A Go
-    #TMP_GMS_VAR := $(TMP_GMS_VAR)_2gb
+    TMP_GMS_VAR := $(TMP_GMS_VAR)_2gb
   endif
   ifeq ($(strip $(BUILD_WITH_EEA)),true)
     BUILD_WITH_GOOGLE_MARKET_ALL := true
```

Keymaster & Optee

申请sub-license时，填写Trust OS时，要注意只有RK3326/RK3562/RK356x/RK3588平台填写Optee V2，其他平台均为V1。

```
PRODUCT_HAVE_OPTEE := true
```

Hardware Features

请仔细和硬件及驱动同事确认设备所支持的硬件，如果硬件不支持（如陀螺仪，BLE等），软件上一定要移除。**注意：Android 11及以上版本要求Nearby Share，该功能依赖BLE，设计硬件时务必注意！**可以通过以下的宏控制，没有宏的，请到`frameworks/native/data/etc/`下将对应的feature.xml删除：

```
BOARD_GRAVITY_SENSOR_SUPPORT := true
BOARD_COMPASS_SENSOR_SUPPORT := true
```

例如，删除BLE:

```
BOARD_BLUETOOTH_LE_SUPPORT := false
```

Usb Camera所使用的Feature配置：

```
android.hardware.camera.any
android.hardware.camera.external
```

过GMS/EDLA认证机器**必须带Microphone**，可以是内置Mic或者Usb Mic，测试前需要保证机器录音正常

若机器没有任何传感器，即`pm list features | grep sensor`查询为空，则软件上要移除传感器服务

```
device/rockchip/common/device.mk:
-# Sensor HAL
-PRODUCT_PACKAGES += \
-    android.hardware.sensors@1.0-service \
-    android.hardware.sensors@1.0-impl \
-    sensors.$(TARGET_BOARD_HARDWARE)

device/rockchip/common/manifests/manifest_level_33.xml:
-    <hal format="hidl">
-        <name>android.hardware.sensors</name>
-        <transport>hwbinder</transport>
-        <version>1.0</version>
-        <interface>
-            <name>ISensors</name>
-            <instance>default</instance>
-        </interface>
-    </hal>
```

FRP (Factory Reset Protection)

GMS需要启用FRP。有关此功能的详细介绍，请参阅随附的SDK文档：
`RKDocs/android/Rockchip_Introduction_Android_Factory_Reset_Protection_CN&EN.pdf`

```
BUILD_WITH_GOOGLE_FRP := true
```

AVB (Android Verified Boot)，A/B升级

GMS需要启用AVB。有关此功能的详细介绍，请参阅随附的SDK文档：
`RKDocs/android/Rockchip_Introduction_Android_Verify_Boot_CN&EN.pdf`

```
BOARD_AVB_ENABLE := true                                    ## 若没有启用AVB则不能锁定机器，否则会导致机器无法开机
```

为方便开发与调试，SDK默认代码没有开启AVB，也没有锁定bootloader。这会导致通过认证后也无法显示已认证，请确认在量产前在uboot目录打上以下补丁，以确保刷key时能够同时锁定bootloader，正常显示已认证，调试阶段建议先不加补丁，否则烧写不是同一套固件时，会无法启动。

```
RKDocs/android/patches/gms/0001-libavb-Lock-the-device-when-the-device-init-or-write.patch
```

- 在Android 11及以上版本中需要启用虚拟A/B，例如:

```
BOARD_USES_AB_IMAGE := true
BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := true

# 下面这段在配置时如果没有，请一起添加
ifeq ($(strip $(BOARD_USES_AB_IMAGE)), true)
  include device/rockchip/common/BoardConfig_AB.mk
  TARGET_RECOVERY_FSTAB := device/rockchip/rk356x/rk3566_rgo/recovery.fstab_AB
endif
```

- 在Android 13使用GKI的平台，需要启用压缩虚拟A/B，例如：

```
BOARD_ROCKCHIP_VIRTUAL_AB_COMPRESSION := true
```

如果编译时提示默认的分区大小不够，可以手动配置：

```
ifeq ($(strip $(BOARD_USES_AB_IMAGE)), true)
  include device/rockchip/common/BoardConfig_AB.mk
  TARGET_RECOVERY_FSTAB := device/rockchip/rk356x/rk3566_rgo/recovery.fstab_AB
endif
+ # 以下这段overlay必须要放到`include device/rockchip/common/BoardConfig_AB.mk`之后
+BOARD_SUPER_PARTITION_SIZE := 4294967296
+BOARD_ROCKCHIP_DYNAMIC_PARTITIONS_SIZE := $(shell expr
$(BOARD_SUPER_PARTITION_SIZE) - 4194304)
```

- 开启A/B时，uboot中也要一起打开A/B的配置, 例如：RK3566/RK3568

```
diff --git a/configs/rk3568_defconfig b/configs/rk3568_defconfig
index b7d433a..a87e2e3 100644
--- a/configs/rk3568_defconfig
+++ b/configs/rk3568_defconfig
@@ -204,6 +204,7 @@ CONFIG_AVB_LIBAVB=y
 CONFIG_AVB_LIBAVB_AB=y
 CONFIG_AVB_LIBAVB_ATX=y
 CONFIG_AVB_LIBAVB_USER=y
+CONFIG_ANDROID_AB=y
 CONFIG_RK_AVB_LIBAVB_USER=y
 CONFIG_OPTEE_CLIENT=y
 CONFIG_OPTEE_V2=y
```

注意：查看机器是否启用AVB命令`pm list features | grep android.software.verified_boot`

```
CtsSecurityTestCases
android.security.cts.VerifiedBootTest#testVerifiedBootSupport

测试报错：java.lang.AssertionError: Verified boot must be supported on the device
```

注意：查看机器是否启用A/B命令`ls -l /dev/block/by-name/`

注意：若没有启用AVB但锁定机器(`fastboot oem at-lock-vboot`)，会导致机器无法开机，用下面两种方法可解锁开机

```
报错信息：Hit key to stop autoboot('CTRL+C'):  0
        ANDROID: reboot reason: "(none)"
        Vboot=0, AVB images, AVB verify
        read_is_device_unlocked() ops returned that device is LOCKED
        avb_slot_verify.c:763: ERROR: vbmeta: Error verifying vbmeta image:
OK_NOT_SIGNED
        AVB verify failed                                   ## AVB校验失败
        Android boot failed, error -1.

解决方法1：固件烧写工具(RKDevTool.exe)--高级功能--擦除所有--机器重启
解决方法2：开机串口按'CTRL+C'进入cmdline模式，串口输入'fastboot usb 0'进入fastboot模式，
adb口输入'fastboot oem at-unlock-vboot'
```

SELinux

GMS需要启用SELinux。开启SELinux后可能会导致某些功能不正常，具体说明及修改方法请参考随附的SDK文档：
`RKDocs/android/Rockchip_Developer_Guide_Android_SELinux(Sepolicy)_CN.pdf`

```
device/rockchip/common/BoardConfig.mk:
BOARD_SELINUX_ENFORCING ?= true
```

对于Android 12及以上版本，我们提供了一个工具可以处理常见的问题，在开始测试前请务必在userdebug固件上使用该工具反复扫描SELinux权限，具体使用方法请参照工具目录下的`README.md`，工具位于：

```
RKTools/linux/Android_Devices_SELinux_Tool
```

```
console:/ $ chmod +x ./data/local/rockchip_selinux_tools
console:/ $ ./data/local/rockchip_selinux_tools
Please USE the following cmd to confirm which files need to be update!
get_build_var BOARD_SEPOLICY_DIRS                       ## 工程source和lunch后执行
该命令
Add the following result to file: [genfs_contexts]
----------------------------------------------------    ## 将下面扫描结果添加到
genfs_contexts
genfscon sysfs /devices/platform/av1d-master/wakeup
u:object_r:sysfs_wakeup:s0
genfscon sysfs /devices/platform/es8388-sound/extcon
u:object_r:sysfs_extcon:s0
```

将扫描结果添加到命令`get_build_var BOARD_SEPOLICY_DIRS`获取的权限文件(例如`device/rockchip/rk3399/sepolicy_vendor/genfs_contexts`)

```
hcq@sys2_206:~/Android12-0928$ get_build_var BOARD_SEPOLICY_DIRS
device/rockchip/common/sepolicy/vendor device/rockchip/rk3399/sepolicy_vendor
....
```

- 检查上述修改是否添加成功：`cat /vendor/etc/selinux/vendor_sepolicy.cil`
- GMS/EDLA认证**不能修改/system/sepolicy目录下任何权限**，SELinux权限相关修改请到/device/rockchip目录

```
下面fail项处理需要用工具扫描SELinux权限
netd_integration_test
NetdSELinuxTest#CheckProperMTULabels

SuspendSepolicyTests
SuspendSepolicyTests#SuspendSepolicyTests

CtsSecurityHostTestCases
android.security.cts.SELinuxHostTest#testNoBugreportDenials
```

固定fingerprint（可选）

在测试过程中可能需要调整固件。建议在测试之前用以下补丁固定指纹，以免在更新固件后无法继续retry测试。

```
device/rockchip/common/prebuild.mk:
diff --git a/prebuild.mk b/prebuild.mk
index 28391f6..7f38922 100644
--- a/prebuild.mk
+++ b/prebuild.mk
@@ -4,3 +4,4 @@ $(warning You can disable this by removing this and setting
BOARD_RECORD_COMMIT_
 $(shell test -d .repo && .repo/repo/repo manifest -r -o
$(OUT_DIR)/commit_id.xml)
 -include $(TARGET_DEVICE_DIR)/prebuild.mk

+ROCKCHIP_BUILD_NUMBER := 202001 (写一串数字即可，自己定)
```

Google要求过GMS/EDLA认证产品的fingerprint必须是唯一的，且拿到认证后fingerprint不能修改(相关说明网址)

安全补丁等级 Security Patch level

安全补丁下载地址：Partner Security Bulletin
当前系统安全补丁等级的查询：`getprop | grep ro.build.version.security_patch`
对于安全补丁的获取及说明，请参考随附的SDK文档：
`Rockchip_Android_Security_Patch_Guide.pdf`
过GMS认证时请留意GMS认证窗口的要求，一般来说安全补丁有效期为三个月，送测时固件的安全补丁不能过旧。

```
GtsOsTestCases
com.google.android.os.gts.SecurityPatchTest#testSecurityPatchDate

报错信息：ro.build.version.security_patch should be "2023-03" or later. Found
"2022-07-05"
```

按**时间顺序**打完安全补丁后需要根据实际情况更新系统安全补丁等级属性：

```
build/make/core/version_defaults.mk
PLATFORM_SECURITY_PATCH := 2020-03-05
```

Attestation key (Keybox)

进行GMS测试前请烧Attestation key。首先要通过3PL向Google申请keybox。拿到keybox后使用瑞芯微提供的打包工具rkpacker进行打包。打包成烧写的格式，再使用烧写工具写到设备中，相关的工具在工程目录：

`RKTools/linux/Linux_AttestationKeyboxPack_Tool.rar`

`RKTools/windows/KeyBoxWrite_v1.53.zip` (务必使用1.53版本，可从FTP服务器`GMS-Test-Suite/xTS/tools`)

详细的操作步骤请参考随附的SDK文档：

`RKDocs/common/RKTools manuals/Rockchip_User_Guide_Keybox_Burning_EN.pdf`

编译固件前，请检查uboot中烧写attestation key的功能有没有打开，例如在3566/3568中打开烧写功能：

```
diff --git a/configs/rk3568_defconfig b/configs/rk3568_defconfig
index 46440b6..22831d2 100644
--- a/configs/rk3568_defconfig
+++ b/configs/rk3568_defconfig
@@ -205,6 +205,10 @@ CONFIG_AVB_LIBAVB=y
 CONFIG_AVB_LIBAVB_AB=y
 CONFIG_AVB_LIBAVB_ATX=y
 CONFIG_AVB_LIBAVB_USER=y
 CONFIG_RK_AVB_LIBAVB_USER=y
 CONFIG_OPTEE_CLIENT=y
 CONFIG_OPTEE_V2=y
+CONFIG_ANDROID_WRITE_KEYBOX=y
+CONFIG_ANDROID_KEYMASTER_CA=y
+CONFIG_OPTEE_ALWAYS_USE_SECURITY_PARTITION=y            ## RK3399不配置该宏
```

- Attestation key烧写到系统security分区，重新烧写固件Attestation key不会丢失，若擦除Flash则需要重新烧录
- 产线如何确认Attestation key烧录成功：u-boot下打补丁`/GMS-Test-Suite/Exp+/Attestation key`获取属性值`ro.boot.deviceid`
- 测试阶段Attestation key通用(即不管Android版本和Soc类型)，但量产时需要烧写项目单独申请的keybox
- 使用工具烧写key时，设备要处于bootloader状态，提示烧写完成后不要直接断电，需要等待8秒左右使烧写相关流程执行完成。可以通过串口打印查看烧写状态(需要确保u-boot代码更新到2022-03)。

```
write attestation key: RSA ret_rsa=0                    ## 0表示烧写完成
```

如果要申请keybox，请提供以下材料(具体需求请和3pl确认)：

```
1. 一份报告（一条case即可，截止到文档更新时间，测试是GtsEdiHostTestCases）
2. 设备信息（问3pl要模板）
3. 一个txt文件（一行一个device id对应一个设备）

Google Request to use Device ID to apply the key:
Device ID within the file should meet the following properties:
    1. Unique and cannot be duplicate of another Device IDs within the file
    2. Must be between 1-32 characters in length
    3. Only following characters allowed [a-z][A-Z][0-9][_][-][.]
```

```
    4. No whitespaces allowed

Device ID files have the following requirements:
    1. ASCII text file in unix format.
    2. File name should be created with the following characters [a-z][A-Z][0-9]
[_][-][.] in a meaningful way (e.g. Make_Model_Date_Quantity.txt)
    3. Must only contain Device IDs. No comments, headers, or other information
    4. One Device ID per line
    5. No duplicate Device IDs within file
    6. No blank lines
    7. No white spaces
```

注意：若向Google申请10万个keybox用工具rkpacker将*.output打包成.kdb时可能会报错"out of memory at line 1"

```
$ chmod +x rkpacker
$ ./rkpacker attest_keyboxes.output -o keybox.kdb
55488 Data File Type : KEYBOX
Output File : keybox.kdb
Data File Num : 1
Total Key Num : 100000
out of memory at line 1

解决方法：split -n 4 attest_keyboxes.output 用命令将10万个keybox分割成4个文件并整理后用rkpacker重新打包
```

Android 13新增RKP要求

RKP(Remote key Provisioning)是Android 13+新增的一项要求，需要在测试之前将机器的信息抽取出来，通过Google Api帐号上传到Google服务器端。该功能较为复杂，详细的介绍与操作请移步至SDK文档:
`RKDocs/android/Rockchip_Android_Remote_key_Provisioning_Guide.pdf`
未使用RKP上传机器信息到Google服务器时GTS测试会有下面2项fail

```
GtsGmscoreHostTestCases
com.google.android.gts.security.AttestationRootHostTest#testEcAttestationChainRem
ProvLength
com.google.android.gts.security.AttestationRootHostTest#testRsaAttestationChainRe
mProvLength
```

GKI要求

Android 12及以上版本中，如果Linux内核版本大于等于5.4，则必须使用GKI(Generic Kernel Image)。简单来说，RK3588 Android 12以及Android 13所有平台都需要使用GKI。

```
BOARD_BUILD_GKI := true
```

该功能较为复杂，详细的介绍与操作请移步至在线文档:
`https://blog.csdn.net/weixin_43245753/article/details/129308364?spm=1001.2014.3001.5502`
参考随附的SDK文档：`RKDocs/android/Rockchip_Developer_Guide_GKI_CN.pdf`

Google官方Release通用内核映像(GKI)下载地址查看附录B

有关Android 12的内核版本使用说明

Android 12同时支持RK3588，但内核版本于其他平台有所不同。内核源码目录如下表：

平台	kernel-4.19	kernel-5.10
RK3326	Y	N
RK356x	Y	N
RK3399	Y	N
RK3588	N	Y
RK3288	Y	N

编译、修改内核源码，驱动时，请注意不要把目录搞错。Android 13内核均为kernel-5.10。

有关incrementalfs的说明

Android 12版本中(**有且仅有Android 12版本需要此步骤**)，GMS认证要求支持`incrementalfs v2`，这需要内核驱动支持。为保证兼容所有平台，我司没有将V2的更新合并到`kernel-4.19`的主线，而是将`incrementalfs_v2`编译成了ko，放在以下目录：

```
vendor/rockchip/common/modular_kernel/4.19$ tree user*
user
`-- incrementalfs.ko
userdebug
`-- incrementalfs.ko

0 directories, 2 files
```

内核版本更新，ko有可能更新不及时，导致无法正确被加载。可以用以下方法确认驱动是否被正确加载，列表中包含`incrementalfs`即加载成功：

```
adb shell lsmod
Module                  Size  Used by
bcmdhd               1150976  0
incrementalfs          77824  0
```

如果没有被正确加载，则需要手动更新，所有V2的补丁`*.patch`以及补丁基于的提交点`base.txt`，都可以在以下位置找到：

```
vendor/rockchip/common/modular_kernel/4.19/incrementalfs_patch
```

kernel-4.19目录下确认`base`提交点的方法：

```
git log fs/incfs/
```

打补丁需要先把该目录提交点切到补丁包`base.txt`提交点。如果您不考虑兼容其他Android 12以前的版本，您也可以将这些补丁默认合并到自己的主线中，做`built-in`编译。确认编译时是否做了`built-in`的方法如下，显示`is not set`即未做`built-in`，显示`=y`则做了`built-in`，这时可以不需要编译`ko`：

```
cat kernel-4.19/.config |grep CONFIG_INCREMENTAL_FS
# CONFIG_INCREMENTAL_FS is not set
```

查看incfs功能是否正常：`/sys/fs/incremental-fs/features/v2`判断节点是否存在
或`pm list features|grep inc`：

```
feature:android.software.incremental_delivery=2
```

固件签名的生成、替换方法

固件/apk签名生成方法，参考如下说明，信息请自行更换：

```
build/target/product/security/README
```

例如：

```
development/tools/make_key testkey
'/C=CN/ST=Fujian/L=Fuzhou/O=Rockchip/OU=Rockchip/CN=Rockchip/emailAddress=gms@roc
k-chips.com'
生成：testkey.pk8 testkey.x509.pem                          ## 文件在根目录下

openssl pkcs8 -inform DER -nocrypt -in testkey.pk8 -out testkey.pem
生成：testkey.pem
```

一般需要自己生成替换以下四组key, 每组3个文件，分别是`pem/pk8/x509.pem`：
`media platform shared testkey`，共计12个文件。直接替换到如下目录：

```
build/target/product/security/
```

Android 13中，签名被统一放到`device/rockchip/common/security`目录，请自行生成替换。注意，自己生成后，请注意替换如下几个签名到apk目录，否则可能会导致蓝牙功能无法使用！即，这几个文件要一致：

- vendor/partner_modules/build/certificates/bluetooth.x509.pem
- packages/modules/Bluetooth/android/app/certs/com.android.bluetooth.x509.pem
- device/rockchip/common/security/bluetooth.x509.pem

注意：替换签名生成.pk8和.x509.pem时不要设置密码，否则openssl命令生成*.pem时会解密失败

```
Enter password for 'testkey' (blank for none; password will be visible):
 ## 不设密码直接回车Enter

报错信息：Error decrypting key
139:error:0D0680A8:asn1 encoding routines:asn1_check_tlen:wrong
tag:../crypto/asn1/tasn_dec.c:1149:
139:error:0D06C03A:asn1 encoding routines:asn1_d2i_ex_primitive:nested asn1
error:../crypto/asn1/tasn_dec.c:713:
139:error:0D08303A:asn1 encoding routines:asn1_template_noexp_d2i:nested asn1
error:../crypto/asn1/tasn_dec.c:646:Field=version, Type=PKCS8_PRIV_KEY_INFO
```

注意：若没有替换固件签名则BTS测试时会产生如下fail项，需要替换`platform.pk8`和`platform.x509.pem`

```
Blocked app /system/framework/framework-res.apk

报错信息：This build contains pre-installed "android" at
/system/framework/framework-res.apk which has been signed by a certificate that
is known to be compromised and should be replaced.
```

Apex签名的生成、替换方法

一般来说，做Regular GMS设备，需要修改Apex签名，否则会产生如下fail:

```
CtsAppSecurityHostTestCases
android.appsecurity.cts.ApexSignatureVerificationTest#testApexPubKeyIsNotWellKnow
nKey
  java.lang.AssertionError: must not use well known pubkey
Expected: must not match well known key
```

生成方法可以在Google官网查看，此处拿`com.android.i18n`举例(执行前，请先source和lunch选择编译目标)：

```
cd libcore/apex/
rm com.android.i18n.pem com.android.i18n.avbpubkey
openssl genrsa -out com.android.i18n.pem 4096            ## 注意换行
avbtool extract_public_key --key com.android.i18n.pem --output
com.android.i18n.avbpubkey
```

通常需要修改以下三个Apex，具体可以测试结果的log为准，log会直接打印在测试终端：

Apex module name	Path-Android 13.0	Path-Android 12.0	Path-Android 11.0
apex_com.android.i18n	packages/modules/RuntimeI18n/apex	和13相同	libcore/apex
apex_com.android.runtime	bionic/apex	和13相同	和13相同
apex_com.android.art	不需要替换	不需要替换	art/build/apex

注意：如果更换后仍不能pass，可以换测试主机试下，有部分客户遇到PC主机导致的fail.

有关App权限的说明

编译user固件时，如未配置宏`BUILD_WITH_GOOGLE_GMS_EXPRESS := true`，编译会提示以下错误：

```
error: Please note that all your apps MUST be able to get permissions, Otherwise
android cannot boot!
```

编user版本必须确认自己的app都符合标准api，且能获取对应的权限，否则会起不来。权限确认无误后，可以把报错位置注释。

```
$(error Please note that all your apps MUST be able to get permissions, Otherwise
android cannot boot!)
```

通常是由于权限导致，可以将权限模式改为log，开机时观察logcat打印，再将这些有问题的app删除或添加权限。**user固件默认不能使用adb**，可以烧写`boot-debug.img`到`boot`分区(A/B固件到`boot_a/boot_b`， GKI则烧写`vendor_boot-debug.img`到`vendor_boot_a/vendor_boot_b`)：

```
device/rockchip/common/modules/gms.mk:
# Enforce privapp-permissions whitelist only for user build.
PRODUCT_PROPERTY_OVERRIDES += \
    ro.control_privapp_permissions=enforce  --> 改为
ro.control_privapp_permissions=log
```

搜索log关键字`not in privapp-permissions allowlist`，可以看到类似如下log：

```
W PackageManager: Privileged permission
android.permission.READ_WALLPAPER_INTERNAL for package
com.google.android.apps.wallpaper (/system_ext/priv-app/WallpaperGoogle) not in
privapp-permissions allowlist
W PackageManager: Privileged permission
android.permission.SET_WALLPAPER_COMPONENT for package
com.google.android.apps.wallpaper (/system_ext/priv-app/WallpaperGoogle) not in
privapp-permissions allowlist
W PackageManager: Privileged permission
android.permission.CHANGE_OVERLAY_PACKAGES for package
com.google.android.apps.wallpaper (/system_ext/priv-app/WallpaperGoogle) not in
privapp-permissions allowlist
W PackageManager: Privileged permission android.permission.WRITE_SECURE_SETTINGS
for package com.google.android.apps.wallpaper (/system_ext/priv-
app/WallpaperGoogle) not in privapp-permissions allowlist
```

根据apk的安装位置，添加对应权限。例如 GMS包中的system_ext下的wallpaper：

```
diff --git a/etc/permissions/privapp-permissions-google-system-ext.xml
b/etc/permissions/privapp-permissions-google-system-ext.xml
index a46cb6c..a187d10 100644
--- a/etc/permissions/privapp-permissions-google-system-ext.xml
+++ b/etc/permissions/privapp-permissions-google-system-ext.xml
@@ -75,4 +75,13 @@
         <permission name="android.permission.WRITE_APN_SETTINGS"/>
         <permission name="android.permission.WRITE_SECURE_SETTINGS"/>
     </privapp-permissions>
```

```
+
+    <privapp-permissions package="com.google.android.apps.wallpaper">
+        <permission name="android.permission.READ_WALLPAPER_INTERNAL"/>
+        <permission name="android.permission.SET_WALLPAPER_COMPONENT"/>
+        <permission name="android.permission.CHANGE_OVERLAY_PACKAGES"/>
+        <permission name="android.permission.WRITE_SECURE_SETTINGS"/>
+    </privapp-permissions>
 </permissions>
```

注：请不要编译RK压力测试apk（`RKDeviceTest/Lightning/StressTest/RKUpdateService`），GMS认证不允许使用。

检查脚本

对于以上大部分要求，Android 12及以上版本已增加一键检查脚本，请在`lunch`自己的项目后，使用该脚本进行检查，务必保证脚本所有项的检查结果都PASS。例如：

```
source build/envsetup.sh
lunch rk3326_sgo-user

source device/rockchip/common/scripts/check_gms_env.sh
```

其中，`BOARD_WIDEVINE_OEMCRYPTO_LEVEL`等级根据自己产品需要配置即可。
`PRODUCT_HAVE_RKAPPS`自己确认APK权限都符合即可。
若uboot相关要求已经按照文档配置固件，但脚本check_gms_env.sh检查仍然fail，则需要先编译uboot再用脚本检查

```
hcq@sys2_206:~/Android12-0928$ source
device/rockchip/common/scripts/check_gms_env.sh
grep: u-boot/.config: No such file or directory
 [Failed]       "CONFIG_ANDROID_AB=y" Check Failed! Expect "CONFIG_ANDROID_AB=y"
 "Android 11+ MUST enable A/B update."
grep: u-boot/.config: No such file or directory
 [Failed]       "CONFIG_ANDROID_WRITE_KEYBOX=y" Check Failed! Expect
"CONFIG_ANDROID_WRITE_KEYBOX=y"
 "Attestation key MUST be enabled."
grep: u-boot/.config: No such file or directory
 [Failed]       "CONFIG_ANDROID_KEYMASTER_CA=y" Check Failed! Expect
"CONFIG_ANDROID_KEYMASTER_CA=y"
 "Keymaster in uboot MUST be enabled."
grep: u-boot/.config: No such file or directory
 [Failed]       "CONFIG_OPTEE_CLIENT=y" Check Failed! Expect
"CONFIG_OPTEE_CLIENT=y"
 "Optee MUST be enabled."
```

对于Android SDK lunch和编译，默认会有检查App权限的提示，请确认所有App没有申请特殊权限后再注释掉该报错！例如：

```
device/rockchip/common/modules/gms.mk:31: warning:
****************************************
device/rockchip/common/modules/gms.mk:32: error: Please note that all your apps
MUST be able to get permissions, Otherwise android cannot boot!.
17:38:09 dumpvars failed with: exit status 1
```

送测前需要检查的部分

送到实验室正式测试之前，一定要根据GMS Requirements仔细核对，以节约时间，对于部分要求给出检查方法：

- 安全模式
  在开机动画阶段按住`音量-`或`重启时按住重启的图标`，即可进入安全模式。
- Lockdown 模式

1. 添加一个锁屏密码
2. 打开`设置`
3. 下滑滚动并点击`安全和位置`
4. 在设备安全下方点击`锁屏偏好`
5. 点击`显示lockdown选项`

- 磁盘加密/文件加密 `Full Disk Encryption/File Base Encryption`
  设备启动后请在以下选项查询状态：
  `settings->security & location->Encryption & credentials->Encrypt tablet`

测前准备

1. 请按Rockchip Android SDK发布说明中的刷机方法，为测试机烧写固件；
2. 由于集成了GMS包，烧写结束后第一次启动会比较慢，请耐心等待，开机启动后完成GMS Setup Wizard中的设置，设置默认语言为United States English，wifi部分请先跳过，时区请选择美国，不能登录账号，进入Home主界面；
3. 确保机器按以下配置，Settings->Wi-Fi连接wifi，连接vpn网络环境，主机也要连接vpn(Ubuntu主机要直连VPN，不能用SSR方式连接VPN)；
4. 开始测试CTS/GTS之前，切记不要登录GMS账号，否则会有fail；如果是EEA Go项目，其中`GtsRegulationComplianceTestCases`单项测试要求，测试前登陆EEA测试帐号，并选择非Google搜索引擎（例如：DuckDuckGo）。具体请以EEA官方文档为准：

   ```
   https://docs.partner.android.com/gms/policies/restricted/eea-builds?hl=en#choice-screen
   ```

5. Settings->Security->Screenlock选择None；
6. 如果产品（如laptop类产品）带物理键盘，Languages & input->Physical keyboard->Show virtual keyboard，勾选该选项；
7. 连续点击Settings->About tablet（phone）->Build Number，使被隐藏的Developer Options显示出来；
8. Settings->Developeroptions->Stayawake，勾选该选项；
9. Settings->Location打开定位服务（默认是打开的，不要关闭）；
10. Settings->Display->Sleep设置成最长时间，将亮度调节到最暗（测试时间较长，以节省电量）；
11. Settings->Display->Adaptive brightness打开自适应亮度(Android13版本要求)；
12. 需使用android-cts-media-version新版的媒体资源包(具体操作查看CTS本地媒体包测试)。

13. 拷贝媒体资源文件，执行媒体包下的脚本`source copy_media.sh && source copy_image.sh`拷贝媒体文件(或将媒体包放到以下位置：`/tmp/android-cts-media/android-cts-media-1.4`)。android 11及以上版本需要`media-1.5`。
14. 查看Sensor校准状态，校准状态永久有效，如果该机器做过校准则不需要再做。测VTS前务必要确认校准状态，查看方法：`cat /sys/class/sensor_class/accel_calibration`，如果有值打印，类似`accel calibration: -604, 131, 535`，则说明校准成功。未校准的机器将其水平静止放置，输入命令`echo 1 > /sys/class/sensor_class/accel_calibration`即可校准，校准后请确认是否校准成功；
15. 物理竖屏的机器要竖屏放置，物理横屏的机器横屏放置；
16. 每次重新测试，都要将机器进行如上配置。
    注：Android 11及以上版本测试GTS前，必须在向导中连接wifi，帐号不要登陆，否则Gts会有fail.

主机配置

需要使用Ubuntu主机(不能使用Windows电脑)并自行配置好java，Python，adb，fastboot，aapt环境。

Rockchip安全补丁FTP服务器：

地址	账户	密码
ftp://ftp.rock-chips.com	xzj-guest	V3qTwMQ4YV

对Android 12及以上版本，我们将在FTP中持续更新客户遇到的常见问题及工具，可以从FTP的以下位置获取：

```
GMS-Test-Suite/Exp+
```

推荐使用RK-FTP中的Android_O_cts_env.tar.gz，解压后配置环境变量：

```
Ubuntu下可以加载到~/.bashrc或者编辑以下内容到文件中如env，测试前source env:

export JAVA_HOME=/home/Your_Name/Software/jdk1.8.0_77     ## 注意替换Your_Name
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
export PATH=/home/Your_Name/Software/android-sdk-linux/tools:$PATH
export PATH=/home/Your_Name/Software/android-sdk-linux/platform-tools:$PATH
export PATH=/home/Your_Name/Software/android-sdk-linux/build-tools/19.0.0:$PATH
```

配置后，请按照下面Q&A中的方法升级adb、aapt及fastboot工具，如果不满足最低版本要求，测试时将会产生各种问题！

```
$ aapt version
$ adb --version
$ fastboot --version
$ java --version
```

各个Android版本要求的adb/aapt/fastboot版本如下：

Android版本	adb版本最低要求	fastboot版本最低要求
Android 10.0	29.0.6+	29.0.6+
Android 11.0	30	30
Android 12.0	31	31
Android 12.1 (Android 12L)	31	31
Android 13	33	33

如果出现aapt不可用，请安装c++兼容库：(aapt不可用会导致CtsDeqpTestCases测试失败)

```
$ sudo apt-get install lib32stdc++6 lib32z1
```

1. 安装Python开发包：

```
$ sudo apt-get install python-dev
```

2. 安装 Protocol Buffer工具：

```
$ sudo apt-get install python-protobuf
$ sudo apt-get install protobuf-compiler
```

3. 安装 Python 虚拟环境相关工具：

```
$ sudo apt-get install python-virtualenv
$ sudo apt-get install python-pip
```

4. 如果你是python3，可能需要单独装下面这个：

```
$ sudo apt install virtualenv
```

主机环境常见问题及注意事项

1. 出现无法创建虚拟环境
   VTS测试时无法创建虚拟环境，可以尝试以下命令强制指定pip为pip2并通过pip安装virtualenv：

```
$ sudo apt autoremove python-virtualenv
$ sudo apt autoremove virtualenv
$ sudo ln -sf ~/.local/bin/pip2 /usr/bin/pip
$ pip install --user virtualenv
```

2. 升级adb、aapt及fastboot
   使用SDK中提供的adb、aapt和fastboot工具替换。参考随附的SDK文档：
   `Rockchip_Android_*_development_guide_V*_CN.pdf`

```
Q：替换adb/fastboot
A：以adb为例，在终端键入：
$ whereis adb
adb: /home/rockchip/Software/android-sdk-linux/platform-tools/adb
$ adb kill-server
得知adb/aapt/fastboot的位置后，替换adb/aapt/fastboot的binary，binary可在FTP服务器GMS-Test-Suite/sdk_tools.tar.gz获取。

若fastboot工具版本过旧，在GSI烧写过程执行命令'fastboot reboot fastboot'时会报错
报错信息：fastboot: usage: unknown reboot target fastboot
```

3. 新测试包需要使用aapt2

```
报错信息1：java.io.IOException: Cannot run program "aapt2": error=2, No such file or directory
报错信息2：E/AaptParser: aapt2 dump badging stderr: W/ziparchive(2284292): Unable to open 'badging': No such file or directory

解决方法：
$ which aapt
/home/rockchip/Software/sdk_tools/aapt
将SDK目录out/host/linux-x86/bin/aapt2文件添加到Ubuntu主机/home/rockchip/Software/sdk_tools/aapt2并给权限'chmod 775 aapt2'
```

4. adb devices查询设备状态提示no permissions

```
$ adb devices
List of devices attached
CKP0KUIQQ1  no permissions(missing udev rules? user is in the plugdev group); see http://developer.android.com/tools/device.html

解决方法步骤1：
$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 070: ID 1f3a:1001 Onda (unverified)
## 获取adb设备识别码1f3a:1001
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

解决方法步骤2：
$ sudo vim /etc/udev/rules.d/android.rules
SUBSYSTEM=="usb",ATTRS{idVendor}=="1f3a",ATTRS{idProduct}=="1001",MODE="0666"
 ## 添加adb设备注意替换1f3a:1001

解决方法步骤3：插拔adb线后查询
$ adb devices
List of devices attached
CKP0KUIQQ1  device
```

5. fastboot devices查询设备状态提示no permissions

```
$ fastboot devices
no permissions(missing udev rules? user is in the plugdev group); see [......]
Android Fastboot

解决方法:在fastboot模式下执行'lsusb'并添加fastboot设备识别码，注意fastboot模式下设备vid/pid和adb模式不同
```

6. 测试模块CtsUsesNativeLibraryTest报错：A JNI error has occurred

```
报错信息：java.lang.RuntimeException: Error: A JNI error has occurred, please check your installation and try again
        Exception in thread "main" java.lang.UnsupportedClassVersionError: com/android/signapk/SignApk has been compiled by a more recent version of the Java Runtime (class file version 55.0), this version of the Java Runtime only recognizes class file versions up to 52.0

解决方法：更新jdk工具版本
$ java --version
```

7. Ubuntu主机内存不足

```
报错信息：E/CommandScheduler: Java heap space
        java.lang.OutOfMemoryError: Java heap space

解决方法：Ubuntu主机需要内存8GB以上，可重启电脑后测试
```

使用Docker容器部署GMS认证测试环境

如果不想按照上述流程搭建Ubuntu主机环境，可从FTP服务器`GMS-Test-Suite/xTS/tools/Docker`下载Docker容器直接运行测试

多台机器协作测试

在以下的测试中大多支持多台机器协作测试，命令为：`--shard-count`

- 如3台机器同时跑cts：

```
run cts --shard-count 3 -s SN1 -s SN2 -s SN3
```

注意：多台机器在测试时，如果其中一台机器掉电或死机，那这台机器中的报告将全部丢失，测试总数量会异常！建议先将机器调试稳定后，先以一台机器测试，测十分钟就拔掉，生成一份完整报告，再基于这个报告多台机器retry测试！

GSI烧写的方法

确认机器解锁后，进入fastbootd，只需要烧写GSI中的system.img及固件中的misc.img，烧写后会进入recovery进行恢复出厂设置。下面附上整个烧写流程：

1. 重启至bootloader，未解锁->机器解锁：

```
adb reboot bootloader
fastboot oem at-unlock-vboot ## 对于烧写过avb公钥的客户，请参考对应的文档解锁。
```

2. 恢复出厂设置，重启至fastbootd：

```
fastboot reboot fastboot     ## 此时将进入fastbootd
```

3. 开始烧写GSI

```
## (可选)对于分区空间紧张的设备，可以先执行本条命令删除product分区后再烧写GSI
fastboot delete-logical-partition product
## 对于使用A/B，虚拟A/B的设备，需要同时删除product_a/product_b
fastboot delete-logical-partition product_a
fastboot delete-logical-partition product_b
fastboot flash system system.img
## 烧misc，恢复出厂设置，misc是固件一起编译出来的
fastboot flash misc misc.img
## 烧写成功后，重启
fastboot reboot
```

- 注1：也可以使用DSU(Dynamic System Updates)烧写GSI，目前Rockchip平台已经默认支持DSU。由于该功能需要消耗大量内存，不建议2G DDR及以下的设备使用，有关DSU的说明和使用，请参考Android官网；
- 注2：VTS测试时，需要同时烧写编译出的`boot-debug.img`到boot分区；如果是GKI，则烧`vendor_boot-debug.img`到vendor_boot分区；
- 注3：CTS-ON-GSI测试时则不需要烧`boot-debug.img`或`vendor_boot-debug.img`；
- 注4：测试时请使用Google官方发布的，以`signed-`开头的GSI镜像和Release build版本的GKI boot镜像；
- 注5：刷完GSI无法启动时，请刷boot-debug或vendor_boot-debug，刷完后adb和串口都可以输入，可以抓logcat分析启动失败的原因。

Android版本	下载地址
aosp-android11-gsi	Android11 PAB版GSI镜像下载地址 搜"aosp_arm64-img"
aosp-android12-gsi	Android12 PAB版GSI镜像下载地址 搜"aosp_arm64-img"
aosp-android13-gsi	Android13 PAB版GSI镜像下载地址 搜"aosp_arm64-img"

测试VTS/CTS-ON-GSI

根据Google GMS文档的要求，测试时需要将机器按下列情况锁定/解锁。

- CTS/GTS: 原有的OEM固件，即正常编译出的完整固件，锁定机器测试(命令`fastboot oem at-lock-vboot`)。
- CTS-on-GSI: 解锁机器，刷写GSI，锁定机器后再进行CTS-on-GSI的测试。(**根据Google最新要求，Android 13+烧GSI后不需要再锁定**)
- VTS: 解锁机器，刷写boot-debug和GSI，再进行VTS的测试（不需要锁定机器）。
  注意：在CTS-on-GSI的情况时，务必要确保GSI为签名(signed-开头)版本，且GSI的安全补丁等级要等于或高于SDK的安全补丁等级，否则烧写后再锁定设备，系统将无法启动！

  注意：烧写GSI镜像后Security Patch显示的是GSI镜像的安全补丁等级，而非原始固件的安全补丁等级。

注意：Android 11/12版本测试CTS或CTS-on-GSI时，务必将设备按上面所说的进行锁定，否则会产生如下fail项。Android 13+的CTS-ON-GSI则不需要锁定

```
CtsKeystoreTestCases
android.keystore.cts.KeyAttestationTest#testEcAttestation
android.keystore.cts.KeyAttestationTest#testRsaAttestation
android.keystore.cts.KeyAttestationTest#testEcAttestation_DeviceLocked
android.keystore.cts.KeyAttestationTest#testRsaAttestation_DeviceLocked
```

- 机器锁定/解锁状态判断：

```
console:/ $ getprop | grep ro.boot.verifiedbootstate
[ro.boot.verifiedbootstate]: [orange]         ## 属性值orange表示机器未锁定
[ro.boot.verifiedbootstate]: [green]          ## 属性值green表示机器已锁定

console:/ $ getprop | grep ro.boot.vbmeta.device_state
[ro.boot.vbmeta.device_state]: [unlocked]     ## 属性值unlocked表示机器未锁定
[ro.boot.vbmeta.device_state]: [locked]       ## 属性值locked表示机器已锁定
```

VTS

需要额外烧写GSI(即AOSP的system-xxx-signed.img，可从Google官网/3PL或Rockchip安全补丁FTP获取，要使用signed的镜像)以及boot-debug.img（开启AVB后打包固件会打包到rockdev/Image-xxx中）。

测试包	测试命令	retry命令	单测命令
android-vts-xxx-arm_64.zip	`run vts`	`run retry --retry 0`	`run vts -m module_name -t case_name`

CTS-ON-GSI

需要额外烧写GSI(即AOSP的system-xxx-signed.img，可从Google官网/3PL或Rockchip安全补丁FTP获取，要使用signed的镜像)，烧写方法同上。

测试包	测试命令	retry命令	单测命令
android-cts-xxx-arm_64.zip	`run cts-on-gsi`	`run retry --retry 0`	`run cts-on-gsi -m module_name -t case_name`

在Android 10及以前版本中，按以下格式：

测试包	测试命令	retry命令	单测命令
android-vts-xxx-arm_64.zip	`run cts-on-gsi`	`run retry --retry 0`	`run cts-on-gsi -m module_name -t case_name`

CTS

使用完整编译出的user固件进行测试，固件配置请按文档前面所说检查和配制好。

测试包	测试命令	retry命令	单测命令
android-cts-_xxx-linux_x86-arm.zip	`run cts`	`run retry --retry 0`	`run cts -m module_name -t case_name`

CTS-Verifier

使用完整编译出的user固件进行测试，固件配置请按文档前面所说检查和配制好。

测试包	测试命令	retry命令	单测命令
android-cts-verifier-xx_xx-linux_x86-arm.zip	手动测试	手动测试	手动测试

1. Android 10.0及以上版本，请在安装CtsVerifier前，执行以下命令:

```
adb shell settings put global hidden_api_policy 1
```

2. 安装CtsVerifier，请务必使用这个命令安装，且在上一步执行之后

```
adb install -r -g CtsVerifier.apk
```

3. Android 10.0及以上版本需要额外执行

```
adb shell appops set com.android.cts.verifier android:read_device_identifiers
allow
```

4. Android 11.0及以上版本需要额外执行

```
adb shell appops set com.android.cts.verifier MANAGE_EXTERNAL_STORAGE 0
```

5. Android 13.0及以上版本需要额外执行(相关说明网址)

```
adb shell am compat enable ALLOW_TEST_API_ACCESS com.android.cts.verifier
```

6. 设置好系统的时间
7. CTS-Verifier测试时烧完固件在开机向导要设置PIN密码，否则测试项Notification Listener Test会找不到条目而fail

GTS

使用完整编译出的user固件进行测试，固件配置请按文档前面所说检查和配制好。**开机向导时需要连接VPN-Wifi，不要登陆Google帐号。**

测试前务必确认好主机环境，是否配置好gts key，可通过`echo $APE_API_KEY`查看。

- GTS key配置方法：从3PL获取后将key.json添加到环境变量中，如：

```
$ vi .bashrc
export APE_API_KEY=/path/to/key.json                    ## 注意替换/path/to

测试报错: Test failed due to failed service account authentication, please ensure
service account key is installed and try again.
```

- EDI白名单申请：将下面模块测试报告发给3PL上传到谷歌服务器申请白名单

```
run gts -m GtsEdiHostTestCases

测试报错: Test failed due to unrecognized service account for this product, please
submit an initial GTS
机器报错: This device isn't Play Protect certified
```

测试包	测试命令	retry命令	单测命令
android-gts-xx.zip	`run gts`	`run retry --retry 0`	`run gts -m module_name -t case_name`

机器未烧Attestation key开机向导连接VPN-Wifi时可能会报错：

```
E/DynamiteLoaderV2Impl( 4625): gzh: Can't load code for GoogleCertificates.apk
E/GoogleCertificates( 4625): Failed to get Google certificates from remote
E/GoogleCertificates( 4625): amh: Remote load failed. No local fallback found.
D/AndroidRuntime( 4813): Shutting down VM
E/AndroidRuntime( 4813): FATAL EXCEPTION: main
E/AndroidRuntime( 4813): Process: com.google.android.apps.restore, PID: 4813
E/AndroidRuntime( 4813): java.lang.SecurityException: GoogleCertificatesRslt:
module init: Remote load failed. No local fallback found. (go/gsrlt)
```

GTS-Verifier

测试说明：https://docs.partner.android.com/gms/testing/gts/gts-verifier?hl=en

STS

使用完整编译出的userdebug固件进行测试，固件配置请按文档前面所说检查和配制好，有关fail项的处理，请参考随附的SDK文档`Android安全补丁说明文档`

测试包	测试命令	retry命令	单测命令
android-sts-xx.zip	`sts-dynamic-full`	`run retry --retry 0`	`run sts -m module_name -t case_name`

注：STS测试包`android-sts-xx.zip`解压密码：`sts`
注：某个fail项所需安全补丁查询网址：
`https://source.android.com/docs/security/bulletin/asb-overview`

GOATS (Performance Test)

性能测试，使用完整编译出的userdebug固件进行测试，固件配置请按文档前面所说检查和配制好。
编译内核必须用Clang编译（编译方法请看开发文档）
kernel中打入如下补丁

```
diff --git a/drivers/video/rockchip/rga2/Kconfig
b/drivers/video/rockchip/rga2/Kconfig
index efc1ef6..3458051 100644
--- a/drivers/video/rockchip/rga2/Kconfig
+++ b/drivers/video/rockchip/rga2/Kconfig
@@ -1,5 +1,5 @@
 # SPDX-License-Identifier: GPL-2.0
-menu "RGA2"
+menu "RGA2G"
        depends on ARCH_ROCKCHIP

 config ROCKCHIP_RGA2
```

APTS

性能测试，使用完整编译出的userdebug固件进行测试，固件配置请按文档前面所说检查和配制好。
Android 12及以上版本的Go版本机器，需要测试新的性能测试，不再测试GOATS。

测试包	测试命令（填写对应user固件的fingerprint）	单测命令，不支持retry
android-apts-go.zip	`run test/approval-go --fingerprint-swap google/wembley/wembley:12/SP1A.210712.001/7539480:user/release`	`run test/approval-go --test-case app-start-cold-3p`

BTS

自2018年4月1日起，所有的设备过认证都要进行Android BTS (Build Test Suite) 测试，请参考Google Partner文档：
https://support.google.com/androidpartners_gms/answer/9027630?hl=en

上传固件给Google时，**不要上传打包后的** `update.img`，否则会出现Google无法解析镜像的问题，具体细节请自行查看文档和询问3pl。RK建议的BTS打包方法为：将以下列表中的镜像放到文件夹中，命名为fingerprint(其中所有的":", "/"请更换为"~")，然后再压缩为zip格式。

```
rockchip~rk3326_qgo~rk3326_qgo~10~QD1A.190821.014.C2~201911~user~release-keys$ ls
boot-debug.img  boot.img  dtbo.img  misc.img  recovery.img  super.img  vbmeta.img
```

关于PHA：有客户曾经出现过GMS测试全部通过，但因BTS中检测出PHA（有害应用）被拒绝的情况，根据Google香港峰会中安全会议所说，可以先将需要预置的app上传到Google Play进行检测，具体咨询3PL或TAM，避免BTS出现类似的情况。

注1：GMS固件启用虚拟A/B，因此固件没有recovery.img
注2：GKI固件，需要额外打包vendor_boot.img/init_boot.img

其他测试命令

单独跳过某一项：

```
run cts --exclude-filter "CtsViewTestCases
android.view.cts.SurfaceViewSyncTests#testEmptySurfaceView"
```

添加fail子项plan：
help add查看

```
add subplan --session xx -n plan_name --result-type type
```

按新建的子项plan测试：

```
run cts -o -a arm64-v8a --skip-all-system-status-check --subplan xxxx
```

GMS Express

为建立良好的Android生态，使得市面上所有的Android设备均能及时应用到Google每月release的安全更新，Google推出了GMS Express Program。
Rockchip全系列平台Android 13.0平板方案将全面支持，如以下表格所示，本文档介绍Rockchip GMS Express技术相关内容。名单会不断更新，请以Google网站名单为准。

芯片平台	Android 12 (Go/Regular)	Android 13 (Go/Regular)
RK3326	Y/Y	即将支持
RK3566	Y/Y	即将支持
RK3399	N/Y	N/即将支持
RK3588	N/Y	N/Y
RK3288	N/Y	N/即将支持

注意事项

- 使用Rockchip GMS Express Basline的客户，请务必确保与Google有签署MADA协议，若是集成Android Go请务必确保有签署Go的补充MADA；集成EDLA，请确保签署EDLA协议，不按照协议滥用设备所造成的一切法律风险需自行承担。
- Rockchip GMS Express baseline，会按Google要求每月都会持续更新（包括安全更新、GMS包的更新、Google提供的AOSP重要补丁等），每个月的baseline正式发布后会正式通知，请务必跟进更新，并将更新按ota方式推送给终端用户。若未及时更新我司将不予提供GMS相关技术支持。
- GMS Express的SDK不是RK常规发布的SDK，如需开通下载权限，请联系RK业务对接人申请，请同时附上能够证明MADA/EDLA资质的材料。

工程下载及配置

Android 10及以上版本中，所有平台代码统一，获得Rockchip SDK后，更新至Express baseline的方法如下，不必重新下载：

```
repo init -m Android10_Express.xml

# Android 11请使用Android 11的xml
repo init -m Android11_Express.xml

# Android 12请使用Android 12的xml
repo init -m Android12_express.xml

# Android 13请使用Android 13的xml
repo init -m Android13_Express.xml
```

- 直接获取Express baseline的方法如下：

```
repo init --repo-url=ssh://git@www.rockchip.com.cn:2222/repo-
release/tools/repo.git  -u
ssh://git@www.rockchip.com.cn:2222/Android_Qt/manifests.git -m
Android10_Express.xml

# Android 11
repo init --repo-url=ssh://git@www.rockchip.com.cn:2222/repo-
release/tools/repo.git  -u
ssh://git@www.rockchip.com.cn:2222/Android_R/manifests.git -m
Android11_Express.xml

# Android 12
repo init --repo-url=ssh://git@www.rockchip.com.cn:2222/repo-
release/tools/repo.git  -u
ssh://git@www.rockchip.com.cn:2222/Android_S/manifests.git -m
Android12_express.xml

# Android 13
repo init --repo-url https://gerrit.rock-chips.com:8443/repo-release/tools/repo
-u https://gerrit.rock-chips.com:8443/Android_T/manifests.git -m
Android13_Express.xml
```

- 请确保产品目录Makefile中做如下配置：

```
BUILD_WITH_GOOGLE_GMS_EXPRESS := true
```

这个宏配置打开后，**我们不预置任何RK的应用，有需要自行添加。**编译结束后，默认会配置`com.google.android.feature.GMSEXPRESS_BUILD`的flag，我们的baseline满足Express Plus的要求。
谷歌鼓励Express Plus的产品，过GMS认证时可以拿到相关补贴支持(不包括EDLA认证)，详情请咨询3PL。在具体项目二次开发后，请参考Google Exp+要求文档确认是否满足Express Plus的要求。若满足，修改以下文件：
`vendor/rockchip/common/gms-express.xml`

```
diff --git a/gms-express.xml b/gms-express.xml
index 78f4d99..de93557 100644
--- a/gms-express.xml
+++ b/gms-express.xml
@@ -16,5 +16,5 @@
  -->
  <!-- These are configurations that should exist on GMS Express devices. -->
  <config>
-     <feature name="com.google.android.feature.GMSEXPRESS_BUILD " />
+     <feature name="com.google.android.feature.GMSEXPRESS_PLUS_BUILD" />
  </config>
```

EDLA客户无法使用GMS Express，但可以在此文件中配置符合自己形态的feature，例如:

```
## 确保打开宏BUILD_WITH_GOOGLE_GMS_EXPRESS := true
diff --git a/gms-express.xml b/gms-express.xml
index 78f4d99..de93557 100644
--- a/gms-express.xml
+++ b/gms-express.xml
@@ -16,5 +16,5 @@
  -->
  <!-- These are configurations that should exist on GMS Express devices. -->
  <config>
-     <feature name="com.google.android.feature.GMSEXPRESS_BUILD " />
+     <feature name="com.google.android.feature.ENTERPRISE_DEVICE" />
+     <feature name="com.google.android.feature.batteryless_device"/>
  </config>
```

EDLA类型需要根据实际产品型态配置相关features：[EDLA类型配置features说明网址]

Device type	Features definition to be included
device lacks a battery	com.google.android.feature.other_form_factor com.google.android.feature.batteryless_device
device has a battery but the screen size is greater than 18 inches	com.google.android.feature.other_form_factor com.google.android.feature.large_display
device lacks a battery and the screen size is greater than 18 inches	com.google.android.feature.other_form_factor com.google.android.feature.batteryless_device com.google.android.feature.large_display
device lacks a battery and is a headless unit with a separate display unit	com.google.android.feature.other_form_factor com.google.android.feature.batteryless_device com.google.android.feature.headless_device

EDLA设备大多不带电池，需要打FTP服务器补丁`GMS-Test-Suite/Exp+/batteryless.diff`，相关说明网址Supporting Batteryless Devices

本地媒体包测试

CTS本地媒体包测试

- 使用本地的媒体包最大的好处就是能够排除网络问题，实现视频秒播。
- CTS测试中使用本地媒体包，可以访问Google说明源地址或从FTP服务器下载`GMS-Test-Suite/xTS/CTS_local_media_package`

1. 下载media媒体包1.5 及在线测试媒体包
2. push在线测试媒体包到机器sdcard
3. 解压media媒体包1.5放到Ubuntu主机目录/tmp/android-cts-media/android-cts-media-1.5

```
$ adb push CtsMediaTestCases /sdcard/
```

4. 单测命令(路径以实际位置为准):

```
$run cts --module-arg CtsMediaTestCases:config-
url:https://storage.googleapis.com/cts_media/DynamicConfig_local.json --module-
arg CtsMediaTestCases:local-media-path:/tmp/android-cts-media/android-cts-media-
1.5
```

GTS本地媒体包测试

- 使用本地的媒体包最大的好处就是能够排除网络问题，实现视频秒播。
- GTS测试中使用本地媒体包，可以访问Google说明源地址或从FTP服务器下载`GMS-Test-Suite/xTS/GTS_local_media_package`

1. 下载GtsExoPlayerTestCases、GtsYouTubeTestCases以及GtsMediaTestCases
2. push本地媒体包到机器sdcard

```
$ adb push gts test wvmedia /sdcard/
```

3. 使用命令测试:

```
run gts \
--module-arg GtsExoPlayerTestCases:config-
url:https://storage.googleapis.com/exoplayer-test-media-1/gen-4/dynamic-config-
sdcard-1.0.json \
--module-arg "GtsYouTubeTestCases:skip-media-download:true" \
--module-arg "GtsYouTubeTestCases:instrumentation-arg:media-path:=/sdcard/test" \
--module-arg "GtsMediaTestCases:instrumentation-arg:media-
path:=file:///sdcard/wvmedia"
```

常见GTS测试fail项需要用本地媒体包测试

```
GtsMediaTestCases
com.google.android.media.gts.WidevineH264PlaybackTests#testL3WithUHD60
com.google.android.media.gts.WidevineH264PlaybackTests#testClearWithUHD60
```

测试常见问题

ConfigurationContainerProto报错

```
测试报错：java.lang.NullPointerException: Attempt to read from field
'com.android.server.wm.nano.ConfigurationContainerProto
com.android.server.wm.nano.WindowContainerProto.configurationContainer' on a null
object reference

解决方法：确认测试包版本是否正确，例如Android12L版本(ro.build.version.sdk=32)需要用
12.1_r1测试包而不是12_r1测试包
```

protobuf报错

- 报错log:

```
java.lang.NoSuchMethodError: 'com.google.protobuf.Parser
com.android.os.StatsLog$ConfigMetricsReportList.parser()'
    at
android.media.mediaparser.cts.MediaParserHostSideTest.getAndClearReportList(Media
ParserHostSideTest.java:294)
    at
android.media.mediaparser.cts.MediaParserHostSideTest.setUp(MediaParserHostSideTe
st.java:79)
    at junit.framework.TestCase.runBare(TestCase.java:140)
```

- 解决方法:
  该问题和终端软件配置有关，且此配置可以通过SSH传到服务端。查看终端是否有配置如下环境：

```
env|grep LC_ALL
```

如果有，请去掉此配置再进xTS测试包测试，例如：

```
unset LC_ALL
./cts-tradefed
```

VTS很多module不跑

- 报错log:

```
xxx.config syntax error: unexpected 'newline'
```

- 解决方法
  可写权限或Windows格式编码会修改文件的结束符。请直接使用Ubuntu测试，不要使用WSL(Windows Subsystem Linux)或任何的虚拟机测试，**测试包直接在Ubuntu中使用unzip命令解压**，不要做包括修改权限（chmod 777）在内的任何操作。

开机向导Wi-Fi设置概率无法skip

性能差导致，可以打这个补丁：

```
patch/frameworks/base/increase_waiting_time_for_setup_wizard.diff
```

通过认证后，无法显示已认证问题

不能显示已认证，一般有两种情况:

1. 机器未烧key
2. 机器未锁定
   为方便开发及调试，默认将机器的状态设为解锁，这里提供一个补丁，默认在烧写key时将机器锁定。
   待生产前，请务必加上补丁以保证烧attestation key时能够锁定机器。

```
RKDocs/android/patches/gms/0001-libavb-Lock-the-device-when-the-device-init-or-write.patch
```

data分区文件系统

- GMS/EDLA认证不支持`ext4`文件格式，需要保证userdata分区的文件格式为`f2fs`：

```
$ adb shell cat vendor/etc/fstab.rk30board | grep userdata
```

- Android 14版本且内核kernel-5.15以上要求/metadata分区和/data分区必须使用F2FS文件格式(分区文件格式要求)

修改系统API等级

- 通过属性`getprop ro.product.first_api_level`获取当前系统API等级，若需要修改参照下面

```
device/rockchip/rk3588/rk3588_t/rk3588_t.mk:     PRODUCT_SHIPPING_API_LEVEL := 33
```

修改系统ro.product.*属性

```
[ro.product.brand]: [rockchip]
[ro.product.device]: [rk3588_t]
[ro.product.manufacturer]: [rockchip]
[ro.product.model]: [rk3588_t]
[ro.product.name]: [rk3588_t]

build/make/core/sysprop.mk:
echo "ro.product.$(1).brand=$(PRODUCT_SYSTEM_BRAND)" >> $(2);\          ## 修改PRODUCT_SYSTEM_BRAND值
echo "ro.product.$(1).device=$(PRODUCT_SYSTEM_DEVICE)" >> $(2);\
```

光感功能(自适应亮度)

```
CtsDisplayTestCases
android.display.cts.BrightnessTest#testGetDefaultCurve

CtsDisplayTestCases
android.display.cts.BrightnessTest#testSetAndGetPerDisplay
测试报错：java.lang.AssertionError: Failed to fetch first slider event. Is the
ambient brightness sensor working?
```

- 打开自适应亮度：Settings->Display->Adaptive brightness(Android13版本要求) -- config_automatic_brightness_available
- `dumpsys display | grep mAutoBrightnessAvailable` 查询结果是否为true
- `dumpsys display | grep MappingStrategy` 查询结果是否为 `PhysicalMappingStrategy`
- `pm list package -f | grep turbo` 查询是否安装应用：Turbo.apk、TurboOverlay.apk
- 需要保证config_autoBrightnessLevels和config_autoBrightnessDisplayValuesNits数组长度相同

```
## 下面6个fail项需要：先调节环境光较亮测试，再调节环境光较暗retry测试
arm64-v8a CtsDisplayTestCases
android.display.cts.BrightnessTest#testNoColorSampleData
android.display.cts.BrightnessTest#testSliderEventsReflectCurves

armeabi-v7a CtsDisplayTestCases
android.display.cts.BrightnessTest#testSetAndGetPerDisplay
android.display.cts.BrightnessTest#testBrightnessSliderTracking
android.display.cts.BrightnessTest#testNoColorSampleData
android.display.cts.BrightnessTest#testSliderEventsReflectCurves
```

若机器没有光感传感器，则需要删掉光感feature:android.hardware.sensor.light，即配置宏

```
BOARD_LIGHT_SENSOR_SUPPORT := false
```

系统dpi和aapt匹配

当系统dpi和aapt不匹配时会产生如下fail项，需要保证 `ro.sf.lcd_density` 和 `PRODUCT_AAPT_PREF_CONFIG` 匹配且符合Google标准

```
CtsThemeHostTestCases
android.theme.cts.ThemeHostTest#testThemes
```

根据产品的屏幕物理尺寸和分辨率 `wm size` 计算系统最理想的dpi值：dpi在线计算器
标准dpi值可查看Google在线文档：dpi和aapt匹配关系
例如计算出dpi=200则就近选择213 dpi(tvdpi)，则需要配置 `ro.sf.lcd_density=213` 和 `PRODUCT_AAPT_PREF_CONFIG := tvdpi`，同时需要保证 `PRODUCT_AAPT_CONFIG` 列表包含tvdpi

查看RK发布SDK版本

获取属性`getprop | grep ro.rksdk.version`

系统配置项查询

FTP服务器/GMS-Test-Suite/xTS/tools/OverlayTools.apk可查询系统/应用相关配置项(注意选择Data Type)

```
am start -n cn.jcassistant.overlaytools/.MainActivity --es pkg android --es cfg
config_automatic_brightness_available
```

机器remount失败

若机器锁定状态下remount则会报下面错误，需要先解锁机器再remount

```
$ adb root
restarting adbd as root
$ adb remount
Skipping /system for remount
Skipping /vendor for remount
Skipping /product for remount
No partitions to remount
remount failed
```

机器SN码命名规范

SN码必须以字母开头且长度在14个字节以内。SN码烧写时机器要进入bootloader状态，写号工具在SDK目录下

```
RKTools/windows/RKDevInfoWriteTool-1.3.0.7z
```

编译报错fail项处理

- 非4G设备配置`BUILD_WITH_GOOGLE_MARKET_ALL := false`编译报错：没有使用RK的GMS Express分支缺少gms-mandatory.mk

```
报错信息：which would you like? [aosp_arm-eng] rk3588_s-userdebug
        device/rockchip/common/modules/gms.mk:100: error:
..."vendor/partner_gms/products/gms-mandatory.mk" does not exist.

解决方法1：脚本check_gms_env.sh检查错误但仍然配置`BUILD_WITH_GOOGLE_MARKET_ALL :=
true`
解决方法2：hcq@sys2_206:~/Android12-0928/vendor/partner_gms/products$ cp gms.mk
gms-mandatory.mk
         hcq@sys2_206:~/Android12-0928/vendor/partner_gms/products$ git diff ./
diff --git a/products/gms.mk b/products/gms.mk
index 2fc7d93..13c07e4 100644
--- a/products/gms.mk
+++ b/products/gms.mk
@@ -74,8 +74,6 @@ PRODUCT_PACKAGES += \
     Videos \
```

```
    YouTube

-# GMS comms suite
-$(call inherit-product,
$(ANDROID_PARTNER_GMS_HOME)/products/google_comms_suite.mk)
```

- 编译报错`mismatch in the <uses-library> tags between the build system and the manifest`

```
报错信息：vendor/partner_gms/apps_go/FilesGoogle/FilesGoogle_arm64.apk )"
error: mismatch in the <uses-library> tags between the build system and the
manifest:
        - required libraries in build system: []
                        vs. in the manifest: []
        - optional libraries in build system: [org.apache.http.legacy]
                        vs. in the manifest: [org.apache.http.legacy,
androidx.window.extensions, androidx.window.sidecar]
        - tags in the manifest
(vendor/partner_gms/apps_go/FilesGoogle/FilesGoogle_arm64.apk):
                uses-library-not-required:'org.apache.http.legacy'
                uses-library-not-required:'androidx.window.extensions'
                uses-library-not-required:'androidx.window.sidecar'
```

```
解决方法1：hcq@sys2_206:~/Android12-0928/vendor/partner_gms$ git diff ./
diff --git a/apps_go/FilesGoogle/Android.mk b/apps_go/FilesGoogle/Android.mk
index fd88fd1..c036fd4 100644
--- a/apps_go/FilesGoogle/Android.mk
+++ b/apps_go/FilesGoogle/Android.mk
@@ -16,7 +16,7 @@ LOCAL_PRODUCT_MODULE := true
 LOCAL_CERTIFICATE := PRESIGNED
 LOCAL_SRC_FILES := $(LOCAL_MODULE)_$(my_src_arch).apk
 #LOCAL_OVERRIDES_PACKAGES :=
-LOCAL_OPTIONAL_USES_LIBRARIES := org.apache.http.legacy
+LOCAL_OPTIONAL_USES_LIBRARIES := org.apache.http.legacy
androidx.window.extensions androidx.window.sidecar

解决方法2：hcq@sys2_206:~/Android12-0928/vendor/partner_gms$ git diff ./
diff --git a/apps_go/FilesGoogle/Android.mk b/apps_go/FilesGoogle/Android.mk
index fd88fd1..fb83106 100644
--- a/apps_go/FilesGoogle/Android.mk
+++ b/apps_go/FilesGoogle/Android.mk
@@ -16,7 +16,7 @@ LOCAL_PRODUCT_MODULE := true
 LOCAL_CERTIFICATE := PRESIGNED
 LOCAL_SRC_FILES := $(LOCAL_MODULE)_$(my_src_arch).apk
 #LOCAL_OVERRIDES_PACKAGES :=
-LOCAL_OPTIONAL_USES_LIBRARIES := org.apache.http.legacy
+LOCAL_ENFORCE_USES_LIBRARIES := false
```

CTS常见fail项处理

- 测试项testSocketKeepaliveUnprivileged报错：FTP服务器/GMS-Test-Suite/Exp+/DisableKeepAliveMainlineBuild

```
CtsNetTestCases
android.net.cts.ConnectivityManagerTest#testSocketKeepaliveUnprivileged
android.net.cts.ConnectivityManagerTest#testCreateTcpKeepalive
android.net.cts.ConnectivityManagerTest#testSocketKeepaliveLimitWifi
```

```
device/rockchip/common/device.mk:
PRODUCT_PACKAGES += \
    RockchipTetheringNoKeepAliveOverlay \
    RockchipTetheringNoKeepAliveMainlineOverlay
```

- 测试项testProfiles和testRequestNotifications报错：把BLE蓝牙设备(如手表)设置成可发现状态放旁边测试

```
CtsOsTestCases
android.os.cts.CompanionDeviceManagerTest#testProfiles
android.os.cts.CompanionDeviceManagerTest#testRequestNotifications

测试报错：FAILURE: java.lang.AssertionError: Test requires a discoverable bluetooth
device nearby
```

- 测试项privappPermissionsMustBeEnforced报错：CTS测试必须用user固件(CTS测试)

```
CtsPermission2TestCases
android.permission2.cts.PrivappPermissionsTest#privappPermissionsMustBeEnforced

报错信息：java.lang.AssertionError: ro.control_privapp_permissions is not set to
enforce expected:<enforce> but was:<null>

user固件'ro.control_privapp_permissions'属性配置逻辑
device/rockchip/common/modules/gms.mk
```

- 测试项SELinuxNeverallowRulesTest报错：GMS/EDLA认证不能修改/system/sepolicy目录下任何权限

```
CtsSecurityHostTestCases
android.security.cts.SELinuxNeverallowRulesTest#testNeverallowRules199

报错信息：libsepol.report_failure: neverallow violated by allow hal_keymint_default
serialno_prop:file { read getattr map open };
        libsepol.check_assertions: 1 neverallow failures occurred
```

- 测试项testNoRemovedCertificates报错：缺失CA证书，Google安全补丁引入

```
CtsSecurityTestCases
android.security.cts.CertificateTest#testNoRemovedCertificates

报错信息：junit.framework.AssertionFailedError: Missing CA certificates expected:
<[]> but was:<[B8:BE:6D:CB:56:F1...]>
解决方法：回退Google安全补丁bulletin_2023_08_preview/.../0001-Drop-TrustCor-
certificates.bulletin.patch
```

- 测试项CtsSensorTestCases报错：根据报错信息调节传感器数据上报频率

```
报错信息1：VerifySensorOperation | sensor='Gravity Sensor', samplingPeriod=10000us,
maxReportLatency=0us | Didn't collect any measurements

报错信息2：VerifySensorOperation | sensor='Accelerometer sensor',
samplingPeriod=20000us, maxReportLatency=10000000us | Frequency out of range:
Requested "Accelerometer sensor" at 50.00hz (expecting between 45.00Hz and
110.00Hz, measured 40.01Hz)
```

```
hcq@sys2_206:~/Android12-0928/hardware/rockchip/sensor$ git diff ./
diff --git a/st/sensors.c b/st/sensors.c
index 1817b28..f7ac231 100755
--- a/st/sensors.c
+++ b/st/sensors.c
@@ -40,7 +40,7 @@ static const struct sensor_t sSensorList[] = {
            .maxRange   = 4.0f*9.80f,
            .resolution = (4.0f*9.80f)/4096.0f,
            .power      = 0.2f,
-           .minDelay   = 7000,
+           .minDelay   = 10000,

hcq@sys2_206:~/Android12-0928/kernel-5.10/drivers/input/sensors$ git diff ./
diff --git a/drivers/input/sensors/sensor-dev.c b/drivers/input/sensors/sensor-
dev.c
index eb93e19157ea..c25f848ea4fe 100644
--- a/drivers/input/sensors/sensor-dev.c
+++ b/drivers/input/sensors/sensor-dev.c
@@ -491,10 +491,10 @@ static int sensor_reset_rate(struct i2c_client *client, int
rate)
        dev_info(&client->dev, "set sensor poll time to %dms\n", rate);

        /* work queue is always slow, we need more quickly to match hal rate */
-       if (sensor->pdata->poll_delay_ms == (rate - 4))
+       if (sensor->pdata->poll_delay_ms == (rate - 6))
                return 0;

-       sensor->pdata->poll_delay_ms = rate - 4;
+       sensor->pdata->poll_delay_ms = rate - 6;

hcq@sys2_206:~/Android12-0928/kernel-5.10/drivers/input/sensors$ git diff ./
diff --git a/drivers/input/sensors/sensor-dev.c b/drivers/input/sensors/sensor-
dev.c
index eb93e19157ea..11b815a1eacd 100644
--- a/drivers/input/sensors/sensor-dev.c
+++ b/drivers/input/sensors/sensor-dev.c
```

```
@@ -1771,7 +1771,7 @@ static int sensor_probe(struct i2c_client *client, const
struct i2c_device_id *d
        client->irq = pdata->irq_pin;
        type = pdata->type;
        pdata->irq_flags = irq_flags;
-       pdata->poll_delay_ms = 30;
+       pdata->poll_delay_ms = 20;                          ## 根据实际情况修改
```

- 测试项CtsWindowManagerDeviceTestCases若报错：需要修改window动画参数为1.0

```
CtsWindowManagerDeviceTestCases
android.server.wm.ActivityTransitionTests#testTaskTransitionOverride
android.server.wm.ActivityTransitionTests#testActivityTransitionDurationNoShorten
AsExpected
android.server.wm.ActivityTransitionTests#testTaskTransitionOverrideDisabled

报错信息：java.lang.AssertionError: Actual transition duration should be in the
range &lt;1900, 2900&gt; ms, actual=1012
```

```
解决方法：hcq@sys2_206:~/Android12-0928/device/rockchip/rk3588$ git diff ./
--- a/overlay/frameworks/base/packages/SettingsProvider/res/values/defaults.xml
+++ b/overlay/frameworks/base/packages/SettingsProvider/res/values/defaults.xml
@@ -36,6 +36,6 @@
     <integer name="def_wifi_sleep_policy">2</integer>

     <!-- Decrease animation duration. -->
-    <fraction name="def_window_animation_scale">50%</fraction>
-    <fraction name="def_window_transition_scale">50%</fraction>
+    <fraction name="def_window_animation_scale">100%</fraction>
+    <fraction name="def_window_transition_scale">100%</fraction>
```

- 测试项BYOD Managed Provisioning - Camera support cross profile video capture (without extra output path)报错

```
解决方法：hcq@sys2_206:~/Android12-0928/packages/apps/Camera2$ git diff
diff --git a/src/com/android/camera/VideoModule.java
b/src/com/android/camera/VideoModule.java
index 2aa6bd9d4..45502d92b 100644
--- a/src/com/android/camera/VideoModule.java
+++ b/src/com/android/camera/VideoModule.java
@@ -1780,7 +1780,7 @@ public class VideoModule extends CameraModule
             mUI.setOrientationIndicator(0, true);
             mActivity.enableKeepScreenOn(false);
             if (shouldAddToMediaStoreNow && !fail) {
-                if (mIsVideoCaptureIntent) {
+                if (mIsVideoCaptureIntent && mActivity.getIntent().getExtras()
!= null) {
                     // if no file save is needed, we can show the post capture
UI now
```

- 测试项Companion Device Service Test报错：BLE蓝牙设备(如手表)设置成可发现状态放旁边

```
com.android.cts.verifier.companion.CompanionDeviceServiceTestActivity

解决方法：FTP服务器/GMS-Test-Suite/xTS/tools/nrfconnect_v4.24.3_itmop.com.apk
1.在辅助机器上安装应用nrfconnect_v4.24.3_itmop.com.apk
2.点击"nRF Connect"应用图标并进入到ADVERTISER选项
3.点击右下角"+"按钮添加蓝牙设备："Display name"随便填写，选择"ADD RECORD"并选择第一项"Complete Local Name"，勾选"Connectable"后点击"OK"按钮
4.点击右边使能开关，在弹框界面直接选择"OK"
5.在测试机器执行测试项"Companion Device Service Test"
6.测试过程等待蓝牙设备消失可能需要2分钟时间

提示：测试过程必须要能弹框"Device appeared"和"Device disappeared"
```

GTS常见fail项处理

- 测试项GtsAssistantHostTestCases报错：根据产品Camera情况把多余的camera feature去掉(**测试前先打开Camera授权拍照后退出再测试**)

```
GtsAssistantHostTestCases
com.google.android.assistant.gts.AssistantTest#testAssistantTakePhotoWithVoiceInteraction
com.google.android.assistant.gts.AssistantTest#testAssistantOpenRearCameraWithoutVoiceInteraction
com.google.android.assistant.gts.AssistantTest#testAssistantTakePhotoWithoutVoiceInteraction
com.google.android.assistant.gts.AssistantTest#testAssistantOpenRearCameraWithVoiceInteraction

console:/ $ pm list features | grep camera
feature:android.hardware.camera                         ## 后置Camera
feature:android.hardware.camera.any                     ## 必须
feature:android.hardware.camera.external                ## Usb Camera
feature:android.hardware.camera.front                   ## 前置Camera
```

- 测试项testTelephonyAudioAvailable报错：若机器不带通话功能(非4G)，需要把通话组件去掉

```
GtsGmscoreHostTestCases
com.google.android.gts.audio.AudioHostTest#testTelephonyAudioAvailable

GtsNmgiarcTestCases
com.google.android.comms.MessagesTests#testMessagesVersionCheck
报错信息：java.lang.RuntimeException: AM should be a privileged system app      ##应用Messages.apk
```

```
hcq@sys2_206:~/Android12-0928/vendor/partner_gms/products$ git diff ./gms.mk
-# GMS comms suite
-$(call inherit-product,
$(ANDROID_PARTNER_GMS_HOME)/products/google_comms_suite.mk)
```

- 测试项testSerialNumberAttestation报错：机器必须烧Attestation key且锁定(`fastboot oem at-lock-vboot`)

```
GtsGmscoreHostTestCases
com.google.android.gts.security.DeviceIdAttestationHostTest#testSerialNumberAttestation
```

- 测试项testMbaPrivilegedPermission报错：应用AndroidManifest.xml不能申明android:debuggable="true"

```
GtsMbaPrivilegedPermissionTestCases
com.google.android.mbaprivilegedpermission.gts.GtsDebugCertificateTest#testMbaPrivilegedPermission

报错信息：java.lang.RuntimeException: Violation of the Debug certificate policy
        [com.rockchip.mirror] is debuggable.
```

- 测试项testDefaultGrantsWithRemoteExceptions报错：根据报错信息删除应用对应权限

```
GtsPermissionTestCases
com.google.android.permission.gts.DefaultPermissionGrantPolicyTest#testDefaultGrantsWithRemoteExceptions

报错信息：java.lang.AssertionError: packageName: com.google.android.contacts {
  priv app: false
  targetSDK: 33
  uid: 10101
  persistent: false
  signature: 917F8719D4C2C892DBF26BBE2BDB179FFF0E0DDBEB8FF9F31A4DD98BB4A4D25D
  on system image: true
  has platform signature: false
  message: cannot be granted by default to package {
    permission: android.permission.POST_NOTIFICATIONS
  }
}
```

- 测试项testPreloadedAppsTargetSdkVersion报错：根据报错信息修改应用`android:targetSdkVersion`

```
GtsPermissionTestCases
com.google.android.permission.gts.PreloadAppsTargetSdkVersionTest#testPreloadedAppsTargetSdkVersion

报错信息：All apps preloaded on DEVICEs launching with Android 11 MUST target API level 29 or higher.
        All apps preloaded on DEVICEs launching with Android 12 or 12L MUST target API level 30 or higher.
        All apps preloaded on DEVICEs launching with Android 13 MUST target API level 31 or higher.
        com.android.launcher3 must target 30 or higher, but targets 29

若修改'android:targetSdkVersion'且生效但测试项仍然fail，可修改'compileSdkVersion'到最新'ro.product.first_api_level'属性值
变量关系：minSdkVersion(lowest possible) <= targetSdkVersion <= compileSdkVersion(latest SDK)
```

- 测试项testZeroTouch_zeroTouchWrapperLaunched报错：开机向导时需要连接VPN-Wifi，不要登陆Google帐号(务必确保机器烧Attestation key)

```
GtsSetupWizardHostTestCases
com.google.android.gts.setupwizard.SetupWizardZeroTouchTest#testZeroTouch_zeroTou
chWrapperLaunched

报错信息：FAILURE: junit.framework.AssertionFailedError: SetupWizard didn't launch
ZeroTouchWrapper.
On Android R+ or when com.google.android.apps.work.oobconfig is installed,
SetupWizard MUST launch ZeroTouchWrapper with intent action
com.google.android.setupwizard.ZERO_TOUCH_SETUP during normal SetupWizard flow
when WIFI/mobile data is connected.
```

VTS常见fail项处理

- 测试项audioPolicyConfigurationValidation报错：根据报错信息修改audio_policy_configuration.xml

```
VtsHalAudioV7_1TargetTest
CheckConfig#audioPolicyConfigurationValidation

报错信息：1 error occurred during xml validation:
        XML is not valid according to the xsd
          While validating: xmlFilePath
                  Which is: /vendor/etc/audio_policy_configuration.xml
        Against the schema: xsd
                  Which is: /data/local/tmp/audio_policy_configuration_V7_0.xsd
        Libxml2 errors:
        Error: Element 'devicePort', attribute 'type': 'AUDIO_DEVICE_OUT_HDMI_1'
is not a valid value of the union type 'extendableAudioDevice'.

解决方法：配置宏BOARD_SUPPORT_MULTIAUDIO ?= false 关闭多声卡输出不集成
audio_policy_configuration_multiaudio.xml
```

- 测试项KernelReleaseFormat报错：需使用Google官方Release的boot.img镜像(GKI Release boot.img)

```
vts_generic_boot_image_test
GenericBootImageTest#KernelReleaseFormat

报错信息：Value of: KernelRelease::Parse(release, true ).has_value()
          Actual: false
        Expected: true
        Kernel release '5.10.168-maybe-dirty' does not have generic kernel image
(GKI) release format. It must match this regex:
        ^(?P<w>\d+)[.](?P<x>\d+)[.](?P<y>\d+)-(?P<z>android\d+)-(?P<k>\d+).*$
        Example: 5.4.42-android12-0-something
```

- 测试项GkiComplianceV2和GkiComplianceV2_kernel报错：需使用Google官方Release的boot.img镜像(GKI Release boot.img)

```
vts_gki_compliance_test
GkiComplianceTest#GkiComplianceV2
GkiComplianceTest#GkiComplianceV2_kernel

报错信息：The GKI image descriptor is not signed by an official key.
```

- 测试项vts_kernel_net_tests报错：FTP服务器补丁/GMS-Test-Suite/Exp+/vts_kernel_net_tests

```
vts_kernel_net_tests
vts_kernel_net_tests#vts_kernel_net_tests

报错信息：testVtiRekey_IPv4_in_IPv4 (xfrm_tunnel_test.XfrmVtiTest) ... ERROR
        testVtiRekey_IPv4_in_IPv6 (xfrm_tunnel_test.XfrmVtiTest) ... ERROR
        testVtiRekey_IPv6_in_IPv4 (xfrm_tunnel_test.XfrmVtiTest) ... ERROR
        testVtiRekey_IPv6_in_IPv6 (xfrm_tunnel_test.XfrmVtiTest) ... ERROR
```

- 测试项Enabled和vts_virtual_ab_test报错：需要启用虚拟A/B和压缩虚拟A/B，保证属性 `ro.virtual_ab.userspace.snapshots.enabled=true`

```
vts_ota_config_test
VAB#Enabled

vts_virtual_ab_test
VirtualAbRequirementTest#EnabledOnLaunchT
```

```
build/make/target/product/virtual_ab_ota/compression.mk:
PRODUCT_VENDOR_PROPERTIES += ro.virtual_ab.userspace.snapshots.enabled=true

build/make/target/product/virtual_ab_ota/compression_with_xor.mk:
$(call inherit-product, $(SRC_TARGET_DIR)/product/virtual_ab_ota/compression.mk)

BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE := true
BOARD_ROCKCHIP_VIRTUAL_AB_COMPRESSION := true
BOARD_BOOT_HEADER_VERSION := 4                                          ##
BOARD_BUILD_GKI := true
```

- 测试项vts_treble_vintf_vendor_test报错：根据报错信息has an empty hash需要Frozen current

```
vts_treble_vintf_vendor_test
DeviceManifest/SingleManifestTest#InterfacesAreReleased/0

测试报错：rockchip.hardware.tv.input@1.0::ITvInput has an empty hash. This is
because it was compiled without being frozen in a corresponding current.txt file.

解决方法：hidl-gen -L hash
            -r rockchip.hardware:vendor/rockchip/hardware/interfaces    ## 替换
hidl_package_root定义的name:path
            -r android.hardware:hardware/interfaces
            -r android.hidl:system/libhidl/transport
            rockchip.hardware.tv.input@1.0::ITvInput                    ## 根据测
试报错替换该行

将上述命令执行结果添加到hidl_package_root同级目录current.txt
可能是文件/vendor/rockchip/hardware/interfaces/current.txt
```

网络常见fail项处理

- 网络环境问题：下面测试项需要VPN/IPv6网络环境(Ubuntu主机要直连VPN，不能用SSR方式连接VPN)

```
## VPN
libcore.java.net.InetAddressTest#test_getByName_invalid
android.net.cts.MultinetworkApiTest#testNativeDatagramTransmission
NativeDnsAsyncTest#Async_NXDOMAIN

GtsExoPlayerTestCases
com.google.android.exoplayer.gts.DashDownloadTest#download

## IPv6
CtsNetTestCases
android.net.cts.DnsTest#testDnsWorks

CtsLibcoreTestCases
libcore.java.net.SocketTest#testSocketTestAllAddresses
```

已知补丁

- 测试项testSystemUiVisibilityCallbackCausedByInsets补丁：/GMS-Test-Suite/Exp+/12.1_CtsWindowManagerDeviceTestCases/testSystemUiVisibilityCallbackCausedByInsets

豁免申请

若机器产品型态不满足某个测试用例条件可考虑向Google申请豁免，申请豁免相关流程如下

- 单侧某个fail项的测试用例，并提供单侧报告results和logs
- 命令`adb bugreport`抓取信息并提供报告bugreport-rk3399_Android12-SQ3A.220705.003.A1-2023-05-10-04-58-08.zip
- 登录@***.corp-partner.google.com账号向Google豁免申请[IssueTracker]

附录A

- 已知通过GMS/EDLA认证的型号

Modules	Sensors
Camera	AR0230 / 索尼IMS 179
WiFi	AP6275P(PCIE) / ap6275S(sdio)

附录B

- 下载地址汇总

测试包	下载地址	需要MADA/EDLA
GMS	https://docs.partner.android.com/gms/building/integrating/gms-download	Y
Mainline	https://drive.google.com/drive/folders/1gxxMalTC99Y2SvyRDUfiDFvY_k4OzjO5	Y
CTS/CTSV	https://source.android.com/docs/compatibility/cts/downloads	N
VTS/GSI	https://docs.partner.android.com/gms/testing/vts	Y
STS	https://drive.google.com/drive/folders/1xqPTtC6MWiQizfFVdG7Ho0f2oGsmH0e-	Y
GTS	https://docs.partner.android.com/gms/testing/gts?hl=en	Y
GTSV	https://docs.partner.android.com/gms/testing/gts/gts-verifier?hl=en#download	
GKI	https://source.android.com/docs/core/architecture/kernel/gki-release-builds	N
APTS	https://docs.partner.android.com/gms/testing/apts?hl=en	Y
GOATS	https://docs.partner.android.com/gms/testing/goats?hl=en	Y
PAB	https://ci.android.com/builds/branches/aosp-master/grid	N
Checklist	https://docs.partner.android.com/gms/policies/overview/manual-checklist?hl=en#	Y
安全补丁	https://drive.google.com/drive/u/0/folders/0B85mEDAGzAbsckRrZFhhV3YwTEk?resourcekey=0-6zNsbY0nWUmYsJO2TF-RZQ	