Rockchip Android GMS用户配置指南

文件标识：RK-YH-YF-237

发布版本：V5.0.4

日期：2025-01-25

文件密级：□绝密　□秘密　□内部资料　■公开

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：　福建省福州市铜盘路软件园A区18号

网址：　www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

产品版本

芯片名称	内核版本
全平台	

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	卞金晨	2020-02-17	发布初稿
V1.0.1	卞金晨	2020-06-30	添加user版本相关说明
V2.0.0	卞金晨	2021-02-22	添加对Android 11的支持
V2.0.1	卞金晨	2021-03-29	添加对测试时机器锁定状态的描述
V2.0.2	蔡建清	2022-02-18	补充attestation key的相关说明
V3.0.0	卞金晨	2022-03-28	添加对Android 12的支持
V3.0.1	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
V3.0.2	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
V4.0.0	卞金晨	2023-04-01	添加对Android 13的支持
V5.0.0	黄超群	2024-04-12	添加对Android 14的支持
V5.0.1	黄超群	2024-07-10	添加GMS固件配置脚本和签名替换脚本
V5.0.2	黄超群	2024-11-20	添加GTS-ROOT和CTS-ON-GSI-SOC说明
V5.0.3	卞金晨	2025-01-24	补充64bit-only的完整配置说明
V5.0.4	黄超群	2025-01-25	BTS打包镜像说明

文档问题反馈：kenjc.bian@rock-chips.com

目录

1. 名词解释

- GMS (Google Mobile Service)：GMS认证包括MADA/EDLA两种类型

  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协议(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协议(Enterprise Device Licensing Agreement)

  和MADA基本一致，放宽了对屏幕和电池的要求，可以做18英寸~70英寸屏幕或无电池的设备，**但内存必须大于2GB**，且产品不能直接面向消费者。

- IR/MR/LR/SMR版本说明

类型	说明	用法
IR(Initial Release)	初始版本发布	首次发布和大规模生产
MR(Maintenance Release)	版本维护	修复问题和添加新功能
LR(Letter Upgrade)	升级Android版本	需要提交LR版本进行GMS认证
SMR(Security Maintenance Release)	安全补丁版本维护	已量产软件定期更新安全补丁

- 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/EDLA协议成为Google Partner后具有MADA/EDLA资质，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(Fingerprint必须相同)，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`run retry --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。

  ```
  报告：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_PHOBEXGDPG_***.txt
  结果说明：若测试结果为"ASSUMPTION_FAILURE"表示跳过该测试项，即认定该测试项Pass，Google只统计"FAILURE"项
  ```

2. 硬件要求

- 主板必须有硬件RTC(**不能是虚拟RTC**)且存在节点/sys/class/rtc/rtc0/
- 机器必须有Microphone，可以是内置Mic或Usb Mic(**若只有Usb Mic则必须切成内置Mic**)，测试前需要保证机器录音正常
- 机器必须有BLE功能(Bluetooth Low Energy)：Android 11及以上版本要求Nearby Share，该功能依赖BLE
- 机器若有Type-C口则**该Type-C口必须支持OTG功能**，否则GTS Interactive会有fail项，硬件设计时请务必确认清楚
- 机器相关feature若有变更，**则必须重新完整过GMS认证**，过完GMS认证的固件请不要再修改feature
- EDLA设备大多数使用Usb Camera，而非mipi这类的内置Camera。过认证时3PL测试不会指定某一种Camera型号，而CTS中测试相对比较严格，目前已经通过认证的sensor型号可以查看附录A

3. 版本要求

3.1 Android 13版本新要求

- Android 13+版本新设备DDR内存必须大于等于2GB，2GB以下DDR不能过GMS认证(包括Go版本)(Android Go配置要求)
- Android 13+版本新增RKP(Remote key Provisioning)要求

3.2 Android 14版本新要求

- Android 14+版本需要3GB及以上DDR内存才能够申请Go版本的GMS认证
- Android 14+版本要求**CPU必须有大核，否则无法过MADA/EDLA认证**
- Android 14+版本新增Device ID Attestation要求

- Android 14+版本**新设备**MADA/EDLA认证**只支持64-bit only设备**过认证，需要配置相关宏，**老版本LR升级不受此限制**

```
# 配置64 bit only的方法如下：
device/rockchip/rk3588/rk3588s_u/BoardConfig.mk
DEVICE_IS_64BIT_ONLY := true
# 并且需确保second配置都为空，可以通过get_build_var TARGET_2ND_ARCH确认
TARGET_2ND_ARCH :=
TARGET_2ND_ARCH_VARIANT :=
TARGET_2ND_CPU_ABI :=
TARGET_2ND_CPU_ABI2 :=
TARGET_2ND_CPU_VARIANT :=
```

3.3 Android 15版本新要求

- 现有RK各芯片平台添加GRF 14后Android 15+版本过GMS认证(MADA/EDLA)需要按照`Android 14 VENDOR + Android 15 SYSTEM`组合方式编译固件测试，具体参考随附的SDK文档：`Android_SDK/vendor/gms_tools/README.md`

平台认证版本	GRF版本	Android 15	Android 16	Android 17
RK3326	14	Y	Y	Y
RK3399	14	Y	Y	Y
RK3562	14	Y	Y	Y
RK356x	14	Y	Y	Y
RK3576	14	Y	Y	Y
RK3588	14	Y	Y	Y

- Android 15 GRF固件编译需要下载2套SDK工程(Android 14 SDK和Android 15 SDK)，然后按照`Android 15 GRF平台开发说明`用脚本`merge_unity.sh`配置固件（需要确保两边基本配置一致）和编译生成`update.img`镜像，该镜像是最终GMS认证测试固件也是发货固件

镜像	从哪个素材包生成
boot.img	Android 14 SDK
dtbo.img	Android 14 SDK
init_boot.img	Android 14 SDK
odm.img	Android 14 SDK
odm_dlkm.img	Android 14 SDK
product.img	**Android 15 SDK**
resource.img	Android 14 SDK
system.img	**Android 15 SDK**
system_dlkm.img	Android 14 SDK
system_ext.img	**Android 15 SDK**
uboot.img	Android 14 SDK
vbmeta.img	**由新镜像重新签名生成**
vendor.img	Android 14 SDK
vendor_boot.img	Android 14 SDK
vendor_dlkm.img	Android 14 SDK

- GRF固件是否编译成功可通过获取`api_level`属性值进行判断，需要保证属性值如下配置

```
console:/ $ getprop | grep api_level
[ro.board.first_api_level]: [34]
[ro.llndk.api_level]: [202404]
[ro.product.first_api_level]: [35]
[ro.vendor.api_level]: [34]
```

4. 测前需知

想要在设备搭载Google移动服务，必须通过MADA/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MADA/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。没有资质的客户请先联系具有MADA/EDLA资质的ODM公司，可以自行联系或通过业务同事引荐。

- 文档中绝大部分的外部链接，都需要MADA/EDLA资质才有访问权限。
- 目前2GB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2GB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
```

5. 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

6. 认证窗口

- Rockchip各芯片平台Android版本MADA/EDLA认证支持情况查询网址GMS Express/GO Ready Chipset
- Google严格把控**首次过MADA/EDLA认证产品的Android版本**，各Android版本认证窗口截止日期GMS approval windows

Android 14版本AOSP发布日期2023年10月4日，MADA认证窗口截止日期2025年1月31日，EDLA认证窗口截止日期2025年12月31日

7. 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-Verifier	1	1
GTS	1	2
GTS-Verifier	1	1
APTS	1	1
STS	1	1

8. 谷歌包获取

GMS认证需要集成以下三个包：GMS包、mainline包、prebuilts包

谷歌包	预置目录	更新方法
GMS包	vendor/partner_gms	比对更新
mainline包	vendor/partner_modules	删除更新
prebuilts包	prebuilts/module_sdk(**整个目录**必须同步替换**同个mainline包**文件)	删除更新

- 集成mainline包时，请务必确认mainline包文档中提到的补丁必须在SDK中存在，否则编译报错。
- 最新mainline包下载地址：https://releasetracker.partner.android.com/
- 最新mainline包及SDK补丁下载地址：https://docs.partner.android.com/mainline/release/release-notes
- GMS包版本有效期查询网址：`https://docs.partner.android.com/gms/building/integrating/gms-download`
- GMS包版本信息获取：`getprop | grep ro.com.google.gmsversion`

```
/vendor/partner_gms/products/gms_package_version.mk:     GMS_PACKAGE_VERSION_ID := 13_202310
/vendor/partner_gms/products/gms.mk:
ro.com.google.gmsversion=$(GMS_PACKAGE_VERSION_ID)
```

1. 没有MADA/EDLA资质的客户不能自己下载，我司也无法提供Google包，需要找3PL实验室获取
2. 拥有MADA/EDLA资质的客户建议使用我们的Express分支，同步下来的GMS包是符合GMS Express Plus规定的。同步方法见章节GMS Express
3. 拥有MADA/EDLA资质的客户想要自己下载GMS包，请按照以下步骤操作：

- 常规GMS包下载解压后放到`vendor/partner_gms`目录：`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/1gxxMa1TC99Y2SvyRDUfiDFvY_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/art/current/sdk/hiddenapi/filtered-stub-flags.csv
> out/soong/hiddenapi/hiddenapi-stub-flags.txt
```

9. 测试包获取

测试包版本：要求使用Google官方发布的最新测试包(测试包版本在线查询)，Google每季度更新一次(即2/5/8/11月)；

测试包下载链接查看附录B，或从安全补丁FTP服务器目录`GMS-Test-Suite/xTS`获取。

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
GTS	user固件	
STS	userdebug固件	测试包每月第1个周三更新
VTS	user固件	烧写Google签名system.img和固件vendor_boot-debug.img
CTS-Verifier	user固件	
GTS-Verifier	user固件	仅Android 13版本需要
GTS Interactive	user固件	Android 13+版本新增要求(android-gts-11-R2以上套件)
GTS Root	user固件	烧写固件vendor_boot-debug.img
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)

10. 安全补丁FTP服务器

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

11. GMS固件配置

11.1 固件配置检查脚本

Rockchip提供GMS固件配置检查脚本，在`lunch`项目后使用脚本检查固件配置，务必保证脚本所有项的检查结果都Pass

```
source build/envsetup.sh
lunch rk3588s_t-user

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

若u-boot相关宏已经按照文档配置，但脚本check_gms_env.sh检查结果仍然Failed，需要先编译u-boot再用脚本检查

```
Android_SDK$ 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."

解决方法：Android_SDK/u-boot$ ./make.sh rk3588
```

上述检查结果的Failed项需要根据下面章节进行GMS固件配置，直到脚本所有项的检查结果都Pass再进行GMS认证测试

11.2 固件配置脚本

Rockchip提供GMS固件配置脚本，工程source和lunch后执行脚本配置GMS固件，脚本主要是对下面章节相关宏进行配置

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

11.3 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

相关逻辑：Android_SDK/build.sh
if [ "$BUILD_KERNEL_WITH_CLANG" = true ] ; then
if [ "$KERNEL_ARCH" = "arm64" ]; then
      ADDON_ARGS="CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1"
else
      ADDON_ARGS="CC=clang LD=ld.lld"
```

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

11.4 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。(EEA固件配置说明)

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

11.5 GMS包使能

```
device/rockchip/rk3588/BoardConfig.mk
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2GB版本，添加如下修改：(Android 13及以上版本DDR 2GB以下不能过MADA/EDLA认证)

```
Android_SDK/device/rockchip/common$ git diff modules/gms.mk
    TMP_GMS_VAR := gms
    TMP_MAINLINE_VAR := mainline_modules
    ifeq ($(strip $(BUILD_WITH_GO_OPT)),true)
       # 2G A Go
-     #TMP_GMS_VAR := $(TMP_GMS_VAR)_go_2gb
+     TMP_GMS_VAR := $(TMP_GMS_VAR)_go_2gb
```

工程source和lunch后可通过命令查看GMS用哪个mk配置文件

```
Android_SDK$ get_build_var TMP_GMS_VAR
gms-mandatory                                                    ##
vendor/partner_gms/products/gms-mandatory.mk
```

若没有使用我们GMS Express分支即未配置宏'BUILD_WITH_GOOGLE_GMS_EXPRESS := true'则默认使用'gms.mk'

11.6 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
```

若机器没有GPS功能，也不能移除下面两个feature(网络定位需要使用)

```
android.hardware.location
android.hardware.location.network
```

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>
注:上述manifests配置文件生成目录/vendor/etc/vintf/manifest.xml
```

11.7 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
```

11.8 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：

```
Android_SDK/device/rockchip/rk3588$ git diff rk3588_s/BoardConfig.mk
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

```
Android_SDK/u-boot$ git diff configs/rk3568_defconfig
+CONFIG_ANDROID_AB=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'
```

11.9 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
```

```
$ adb root
$ adb shell
console:/ $ chmod +x /data/local/tmp/rockchip_selinux_tools64
console:/ $ ./data/local/tmp/rockchip_selinux_tools64
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
```

将扫描结果添加到宏`BOARD_SEPOLICY_DIRS`定义的权限文件(例如`device/rockchip/rk3399/sepolicy_vendor/genfs_contexts`)

```
Android_SDK$ 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`
- MADA/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
- MADA/EDLA认证**不能修改user固件默认adb root权限**(属性ro.debuggable)，否则可能会导致adbd进程挂死阻塞测试进展和测试报告/日志丢失

```
I init     : starting service 'adbd'...
I init     : Created socket '/dev/socket/adbd', mode 660, user 1000, group 1000
W adbd     : type=1400 audit(0.0:27602): avc: denied { setcurrent } for scontext=u:r:adbd:s0
tcontext=u:r:adbd:s0 tclass=process permissive=0
F adbd     : main.cpp:162] Could not set SELinux context
I init     : Service 'adbd' (pid 10075) received signal 6
I init     : Sending signal 9 to service 'adbd' (pid 10075) process group...
E init     : process with updatable components 'adbd' exited 4 times in 4 minutes
```

下面fail项处理需要用工具扫描SELinux权限，若添加权限后仍然测试fail机器可"擦除所有"重新烧写固件测试

```
netd_integration_test
NetdSELinuxTest#CheckProperMTULabels

SuspendSepolicyTests
SuspendSepolicyTests#SuspendSepolicyTests

CtsSecurityHostTestCases
android.security.cts.SELinuxHostTest#testNoBugreportDenials        ## 需要打开宏
BOARD_SELINUX_ENFORCING ?= true
```

11.10 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。

```
Android_SDK/device/rockchip/common$ git diff prebuild.mk
 -include $(TARGET_DEVICE_DIR)/prebuild.mk
+ROCKCHIP_BUILD_NUMBER := 202001 (写一串数字即可，自己定)
```

11.11 安全补丁等级 Security Patch level

安全补丁下载地址：Partner_Security_Bulletin 或 安全补丁FTP服务器

安全补丁等级查询：`getprop | grep ro.build.version.security_patch`

安全补丁指导文档：FTP服务器`/GMS-Test-Suite/xTS/tools/Rockchip_Android_Security_Patch_Guide.pdf`

安全补丁有效期一般是3个月，送测时固件的安全补丁不能过旧，请留意GMS认证窗口的要求。

```
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"
```

Android版本	Google安全补丁分支查询
Android 12(包括Android 12L)	android12-security-release
Android 13	android13-security-release
Android 14	android14-security-release

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

Android15_SDK/build/make/core/version_util.mk
PLATFORM_SECURITY_PATCH := $(RELEASE_PLATFORM_SECURITY_PATCH)

## Android15_SDK此处"ap3a"会跟着版本变动，可能是"ap4a"或"ap5a"
Android15_SDK/build/release/flag_values/ap3a/RELEASE_PLATFORM_SECURITY_PATCH.textproto
name: "RELEASE_PLATFORM_SECURITY_PATCH"
value: {
  string_value: "2024-10-05"
}
```

安全补丁等级必须是**完整"年-月-日"格式**，否则参数--rollback_index为空会导致vbmeta.img编译报错

```
[ 99% 8801/8803] Target vbmeta image: out/target/product/rk3588_t/vbmeta.img
FAILED: out/target/product/rk3588_t/vbmeta.img
```

```
usage: avbtool make_vbmeta_image [-h] [--output OUTPUT]
                                                [--padding_size NUMBER]
                                                [--algorithm ALGORITHM] [--key KEY]
                                                [--signing_helper APP]
                                                [--signing_helper_with_files APP]
                                                [--public_key_metadata KEY_METADATA]
                                                [--rollback_index ROLLBACK_INDEX]
                                                [--rollback_index_location
ROLLBACK_INDEX_LOCATION]
                                                [--append_to_release_string STR]
                                                [--prop KEY:VALUE]
                                                [--prop_from_file KEY:PATH]
                                                [--kernel_cmdline CMDLINE]
                                                [--setup_rootfs_from_kernel IMAGE]
                                                [--include_descriptors_from_image IMAGE]
                                                [--print_required_libavb_version]
                                                [--chain_partition
PART_NAME:ROLLBACK_SLOT:KEY_PATH]
                                                [--flags FLAGS]
                                                [--set_hashtree_disabled_flag]
avbtool make_vbmeta_image: error: argument --rollback_index: expected one argument
```

```
build/make/core/version_util.mk:
ifndef PLATFORM_SECURITY_PATCH_TIMESTAMP
   # Used to indicate the matching timestamp for the security patch string in
PLATFORM_SECURITY_PATCH.
  PLATFORM_SECURITY_PATCH_TIMESTAMP := $(shell date -d 'TZ="GMT" $(PLATFORM_SECURITY_PATCH)' +%s)
endif

device/rockchip/common/modules/avb.mk:
BOARD_AVB_ROLLBACK_INDEX ?= $(PLATFORM_SECURITY_PATCH_TIMESTAMP)

build/make/core/Makefile:
ifdef BOARD_AVB_ROLLBACK_INDEX
  BOARD_AVB_MAKE_VBMETA_IMAGE_ARGS += --rollback_index $(BOARD_AVB_ROLLBACK_INDEX)
endif

define build-vbmetaimage-target
   $(call pretty,"Target vbmeta image: $(INSTALLED_VBMETAIMAGE_TARGET)")
   $(hide) mkdir -p $(AVB_CHAIN_KEY_DIR)
   $(call extract-avb-chain-public-keys, $(AVB_CHAIN_KEY_DIR))
   $(hide) $(AVBTOOL) make_vbmeta_image \
      $(INTERNAL_AVB_MAKE_VBMETA_IMAGE_ARGS) \
      $(PRIVATE_AVB_VBMETA_SIGNING_ARGS) \
      $(BOARD_AVB_MAKE_VBMETA_IMAGE_ARGS) \
     --output $@
   $(hide) rm -rf $(AVB_CHAIN_KEY_DIR)
endef
```

11.12 Widevine

RK3326/RK3399/RK356X/RK3562/RK3576/RK3588都支持Widevine L1。平板设备一般L3即可，如果需要使用L1请与我司联系获取L1补丁。

```
BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 3
```

注意：Widevine key打包工具：FTP服务器`GMS-Test-Suite/xTS/tools/rkpacker2`

由于Google策略更新若没有MADA/EDLA资质我司不能直接提供widevine代码，因此SDK中默认删除/vendor/widevine仓库

百度网盘下载链接：https://pan.baidu.com/s/1LFSZMdG_Rzkf8wzMFgQxvA?pwd=68pp 提取码：68pp

```
Android_SDK/vendor$ unzip widevine-Idb769b95046c3de08d4d7424b7d06d3ffcbbfa3e.zip
```

Android 14确认widevine是否正常集成

```
console:/ $ mount | grep widevine
/dev/block/loop6 on /apex/com.google.android.widevine@341310000 type ext4
(ro,dirsync,seclabel,nodev,noatime)
/dev/block/loop6 on /apex/com.google.android.widevine type ext4
(ro,dirsync,seclabel,nodev,noatime)
```

11.13 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
```

11.14 Attestation key (Google keybox)

进行GMS测试前请烧写Attestation key。首先通过3PL向谷歌申请Google keybox，拿到keybox后使用瑞芯微提供的打包工具rkpacker打包成烧写的格式，再使用工具烧写到设备中，相关的工具在工程目录：
`RKTools/linux/Linux_AttestationKeyboxPack_Tool.rar`
`RKTools/windows/KeyBoxWrite_v1.53.zip` (可从FTP服务器`GMS-Test-Suite/xTS/tools/KeyBoxWrite_v1.53.zip`)

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：

```
Android_SDK/u-boot$ git diff configs/rk3568_defconfig
 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烧写到机器emmc的security分区，重新烧写固件Attestation key不会丢失，若擦除Flash则需要重新烧写
- 产线如何确认Attestation key烧写成功：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_key key_file=PrivateKey.RSA ret=0                          ## 0表示烧写完成
write_key key_file=PrivateKey.EC ret=0
或
write attestation key: RSA ret_rsa=0
```

- Android 14+版本可以通过开机串口logcat查看机器是否烧写Attestation key和Device ID Attestation

```
Provision google keybox:no,ID attestation:no                    ## yes表示已烧，no表示未烧
```

- 新测试包要求机器烧写Attestation key和Device ID Attestation，否则CTS/GTS测试有下面提示信息，导致测试报告信息不全无法上传Google服务器

```
W/ApkInstrumentationPreparer: Target preparation step
com.android.compatibility.common.deviceinfo.KeystoreAttestationDeviceInfo#testCollectDeviceInfo
failed.
junit.framework.AssertionFailedError: KeystoreAttestationDeviceInfo collection completed.
expected:<COMPLETED> but was:<FAILED>
```

如果要申请keybox，请提供以下材料(具体需求请和3pl确认)：按照Google说法每个DeviceID烧写一台机器

```
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：
1.gzip -cdk 2023-01-01_***.output.gz > attest_keyboxes.output
2.split -n 4 attest_keyboxes.output --additional-suffix=.xml
3.cat xaa.xml | grep -w "</Keybox>" | wc -l                    ## xaa.xml为切割后第一个文件
4.将上述Keybox数量查询结果赋值<NumberOfKeyboxes>100</NumberOfKeyboxes>
5.将最后一个"</Certificate></CertificateChain></Key></Keybox>"后面内容删掉，并补充最后一行"
</AndroidAttestation>"
6.rkpacker xaa.xml -o keybox.kdb

其中xaa.xml完整结构如下
<?xml version="1.0"?>
<AndroidAttestation>
<NumberOfKeyboxes>4</NumberOfKeyboxes>                         ## NumberOfKeyboxes值根据"
</Keybox>"数量确定
<Keybox DeviceID="BW10RS23090001"><Key algorithm="ecdsa"><PrivateKey format="pem">......
......
-----END CERTIFICATE-----
</Certificate></CertificateChain></Key></Keybox>
</AndroidAttestation>
```

注意：若使用工具rkpacker将*.output打包成.kdb时报错缺少库libcrypto.so.1.0.0

```
报错信息：error while loading shared libraries: libcrypto.so.1.0.0: cannot open shared object file: No such file or directory
解决方法：FTP服务器/GMS-Test-Suite/xTS/tools/libcrypto.so.1.0.0预置到Ubuntu主机"/usr/lib/x86_64-linux-gnu/libcrypto.so.1.0.0"
```

下面测试项需要机器烧写Attestation key且锁定(`fastboot oem at-lock-vboot`)，若仍然测试fail可以擦除机器再烧写锁定

```
CtsAppSecurityHostTestCases
CtsDevicePolicyManagerTestCases
CtsKeystorePerformanceTestCases
CtsKeystoreTestCases

VtsAidlKeyMintTargetTest
```

11.15 Android 13新增RKP要求

RKP(Remote key Provisioning)是Android 13+新增要求，需要在测试前提取机器信息并通过android-partner-api@company.com账号上传到Google服务器端。

详细介绍与操作流程请参考随附的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#testEcAttestationChainRemProvLength
com.google.android.gts.security.AttestationRootHostTest#testRsaAttestationChainRemProvLength
或
GtsGmscoreHostTestCases
com.google.android.gts.security.AttestationRootHostTest#testEcAttestationChainRemProvLengthTee
com.google.android.gts.security.AttestationRootHostTest#testRsaAttestationChainRemProvLengthTee
```

注意：RKP工具机器目录`/vendor/bin/rockchip_rkp_factory_extraction_tool`

注意：若机器SN码没有修改则不需要重新提取RKP CSR信息，**提取RKP前必须确认机器已经烧写SN码**

注意：如果需要同时烧写Attestation key且提取RKP CSR信息请使用工具`RKTools/windows/KeyboxWriter-v1.55.7z`

注意：如果分开烧写Attestation key和提取RKP CSR信息请使用FTP服务器工具GMS-Test-Suite/xTS/tools/KeyBoxWrite_v1.53.zip

注意：RKP账号android-partner-api@company.com申请周期较长(起码要一两周)请提前申请

11.16 Android 14新增Device ID Attestation要求

Android 14+版本需要同时烧写Device ID Attestation，只允许烧写一次且不能修改，若需要重新烧写则要先擦除机器(工具`RKDevTool.exe`)
操作步骤请参考随附的SDK文档:
`RKDocs/android/Rockchip_User_Guide_Android_Device_ID_Attestation_Flash_CN.pdf`

Device ID Attestation烧写工具：FTP服务器`/GMS-Test-Suite/xTS/tools/RKDevInfoWriteTool-V1.2.1.7z`

用工具烧写Attestation key和Device ID Attestation时，需要保证机器**提前烧写SN码，否则RKP提取失败**

```
10:54:48:013 ReadRkp[15]: reading sn from device fail
```

Device ID Attestation功能需要申请下面两个features:

```
pm list features | grep android.software.device_id_attestation
pm list features | grep android.hardware.device_unique_attestation
```

若没有烧写Device ID Attestation信息，会有下面几个fail项

```
CtsKeystoreTestCases
android.keystore.cts.KeyAttestationTest#testRsaAttestation
android.keystore.cts.KeyAttestationTest#testEcAttestation
android.keystore.cts.DeviceOwnerKeyManagementTest#testAllVariationsOfDeviceIdAttestation
报错信息：java.lang.UnsupportedOperationException: Device does not support Device ID attestation.
```

11.17 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

GKI固件需要使用Google官方Release的boot.img镜像，boot分区大小固定64MB

```
device/rockchip/common/build/rockchip/Partitions.mk:
   ifeq ($(BOARD_BUILD_GKI),true)
     BOARD_BOOTIMAGE_PARTITION_SIZE ?= 67108864
     BOARD_VENDOR_BOOTIMAGE_PARTITION_SIZE ?= 41943040
```

注意：查看机器是否启用GKI命令`ls -l /dev/block/by-name/vendor_boot*`

GKI相关测试项

```
CtsHardwareTestCases
android.hardware.input.cts.tests.SonyDualSenseBluetoothTest#testAllKeys

StsHostTestCases
android.security.sts.KernelLtsTest#testRequiredKernelLts_ENFORCING
android.security.sts.KernelLtsTest#testRequiredKernelLts_WARN

vts_fs_test
fs#ErofsSupported

VtsBootconfigTest
VtsBootconfigTest#ProcCmdlineAndroidbootTest
```

11.18 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，请注意要在正确的目录下操作，Android 13内核均为kernel-5.10。

11.19 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：判断`/sys/fs/incremental-fs/features/v2`节点是否存在或查看`pm list features | grep incremental`

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

相关测试项

```
GtsIncrementalInstallProxyHostTestCases
com.google.android.packageinstallerv2proxy.host.gts.ConcurrentInstallationHostTest#testConcurrentI
nstallWithOneFailure

报错信息：java.lang.AssertionError: expected:<SUCCESS> but was:<FAILURE>
关键打印：[incfs] Feature is not enabled
I InstallationHelper: Commit
I IncrementalService: createStorage: /data/app/vmdl1270450043.tmp | 5
W incfs    : [incfs] Feature is not enabled
E IncrementalService: Vold::mountIncFs() failed: Status(-8, EX_SERVICE_SPECIFIC): '-95: '
E InstallationHelper: Installation Failed. Error Code: 6, message:
INSTALL_FAILED_MEDIA_UNAVAILABLE: Couldn't create incremental storage at
/data/app/vmdl1270450043.tmp
```

11.20 固件签名的生成、替换方法

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

固件/apk签名生成方法参考如上说明，信息请自行更换，例如：

```
  development/tools/make_key testkey
'/C=CN/ST=Fujian/L=Fuzhou/O=Rockchip/OU=Rockchip/CN=Rockchip/emailAddress=gms@rock-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4组key(分别是`media platform shared testkey`)，每组有3个文件(分别是`pem/pk8/x509.pem`)，共计12个文件。

```
  签名替换目录由宏'OTA_KEY_DIR'定义
  device/rockchip/common/mkimage.sh:  OTA_KEY_DIR=device/rockchip/common/security

   # Android 11替换到如下目录
  build/target/product/security/
```

```
 # Android 12替换到如下目录
build/target/product/security/

 # Android 13替换到如下目录
device/rockchip/common/security/

 # Android 14替换到如下目录
build/target/product/security/
```

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.
```

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：

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

11.21 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#testApexPubKeyIsNotWellKnownKey
报错信息：java.lang.AssertionError: must not use well known pubkey
          Expected: must not match well known key
               but: was </tmp/ApexSignatureVerificationTest13030241211815479418/tests-apex_com.android.btservices874024/apex_pubkey>

GtsApexSignatureVerificationTest
android.appsecurity.cts.ApexSignatureVerificationTest#testApexPubKeyIsNotWellKnownKey
报错信息：1. com.rockchip.hardware.sensors must not use well known pubkey
                expected to be false
```

```
            7. com.android.rkpd must not use well known pubkey
                expected to be false
```

解决方法：根据报错信息替换apex_com.android.btservices签名(SDK目录packages/modules/Bluetooth/apex)

GtsPackageSignatureTest
android.security.cts.PackageSignatureTest#testPackageSignatures
报错信息：These packages should not be signed with a well known key: [com.android.bluetooth, com.android.btservices, com.android.i18n, com.android.rkpd, com.android.runtime, com.android.uwb, com.android.uwb.resources, com.android.virt, com.android.vndk.v34, com.rockchip.hardware.sensors]

这里拿`com.android.i18n`举例说明替换方法(注意：工程source和lunch后再执行)。生成方法可以在Google官网查看

```
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
apex_com.android.btservices			

注意：替换Apex签名后若编译报错，需要make clean后编译固件

```
报错信息：avbtool: Adding hashtree_footer failed: Key is wrong size for algorithm SHA256_RSA4096.
```

注意：替换Apex签名仍然测试fail，关闭Ubuntu主机终端再打开(Apex签名文件缓存目录`/tmp/ApexSignatureVerificationTest***`)

11.22 有关App权限的说明

编译user固件时，默认会有检查App权限的提示，如未配置宏`BUILD_WITH_GOOGLE_GMS_EXPRESS := true`，会编译报错：

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

编译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。

```
Android_SDK/device/rockchip/common$ git diff modules/gms.mk
    ifeq ($(strip $(TARGET_BUILD_VARIANT)), user)
       ifneq ($(strip $(BUILD_WITH_GOOGLE_GMS_EXPRESS)),true)
          $(warning ****************************************)
-         $(error Please note that all your apps MUST be able to get permissions, Otherwise android
cannot boot!)
          $(warning After confirming your apps, please remove the above error line!)
          $(warning ****************************************)
```

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:

```
Android_SDK/vendor/partner_gms$ git diff etc/permissions/privapp-permissions-google-system-ext.xml
+    <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。

11.23 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`

设备启动后查询加密状态：`Settings->Security->More security settings->Encryption & credentials->Encrypt tablet`

12. 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各芯片平台Android版本支持GMS Express情况如表格所示，实际请以Google网址GMS Express/GO Ready Chipset查询内容为准。

芯片平台	Android 12 (Regular/Go)	Android 13 (Regular/Go)	Android 14 (Regular/Go)
RK3288	Y/N	*/N	N/N
RK3326	Y/Y	*/Y	*/Y
RK3399	Y/N	*/N	*/N
RK3562	N/N	Y/Y	Y/Y
RK3566	Y/Y	Y/Y	Y/Y
RK3576	N/N	N/N	Y/N
RK3588	Y/N	Y/N	Y/N

12.1 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。

12.2 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

# Android 14请使用Android 14的xml
repo init -m Android14_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

# Android 14
repo init --repo-url https://gerrit.rock-chips.com:8443/repo-release/tools/repo
-u https://gerrit.rock-chips.com:8443/Android_U/manifests.git -m Android14_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：

```
Android_SDK/vendor/rockchip/common$ git diff gms-express.xml
  <!-- 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，前提是配置宏`BUILD_WITH_GOOGLE_GMS_EXPRESS := true`

```
Android_SDK/vendor/rockchip/common$ git diff gms-express.xml
  <!-- 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`

13. 主机配置

13.1 Ubuntu主机环境配置

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。

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

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

推荐使用安全补丁FTP服务器中的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
```

13.2 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'

SDK编译逻辑：
prebuilts/sdk/tools/Android.bp
cc_prebuilt_binary {
     name: "aapt2",
     host_supported: true,
     prefer: true,
     target: {
           linux: {
                 srcs: ["linux/bin/aapt2"],
           },
           darwin: {
                 srcs: ["darwin/bin/aapt2"],
           },
           windows: {
                 srcs: ["windows/bin/aapt2.exe"],
           },
     },
```

4. adb devices查询设备状态提示no permissions

```
$ adb devices
List of devices attached
CKPOKUIQQ1  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：## 添加adb设备注意替换1f3a:1001
$ sudo vim /etc/udev/rules.d/android.rules
SUBSYSTEM=="usb",ATTRS{idVendor}=="1f3a",ATTRS{idProduct}=="1001",MODE="0666"

解决方法步骤3：插拔adb线后查询
$ adb devices
List of devices attached
CKPOKUIQQ1  device
```

```
GtsUsbHostTestCases
com.google.android.gts.usb.AccessoryModeHostTest#usbAccessoryModeIsImplemented
com.google.android.gts.usb.AccessoryModeHostTest#switchToAccessoryModeFromOtherFunctions
```

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(FTP服务器/GMS-Test-Suite/openjdk-11+28_linux-x64_bin.tar.gz)
$ java --version
openjdk 11 2018-09-25
OpenJDK Runtime Environment 18.9 (build 11+28)
OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)

$ whereis java
java: /usr/share/java /home/Rockchip/Software/jdk-11/bin/java
```

7. Ubuntu主机堆内存不足

```
报错信息：E/CommandScheduler: Java heap space
          java.lang.OutOfMemoryError: Java heap space
报错信息：${JAVA_BINARY} $RDBG_FLAG -Xmx16g -XX:+HeapDumpOnOutOfMemoryError -cp ${JAR_PATH} -DCTS_ROOT=${CTS_ROOT}

解决方法：Ubuntu主机需要内存16GB及以上，可以重启电脑后再测试
```

13.3 使用Docker容器部署GMS认证测试环境

若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容器直接测试

14. GMS认证测试

14.1 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(**升级固件需要重新校验**)，测试VTS前务必确认校准状态
    按照下面步骤查看Sensor校准状态，未校准的机器水平静止放置(`user固件烧vendor_boot-debug.img`)

```
$ adb root
adbd is already running as root
$ adb shell
console:/ $ cat /sys/class/sensor_class/accel_calibration
read error                                                                    ## 未校
准
console:/ $ echo 1 > /sys/class/sensor_class/accel_calibration   ## 机器水平静止放置
console:/ $
console:/ $ cat /sys/class/sensor_class/accel_calibration
accel calibration: 166, -309, -177                                          ## 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. 测试前机器不能设置PIN锁屏密码(报错信息`Device must have screen lock disabled`)
17. 测试前需要**保证机器SN码重启后不会变化，即机器SN码是固定的**，SN码烧写参考(机器SN码命名规范)
18. 每次重新测试，都要将机器进行如上配置。

14.2 测试终端进入命令

```
Rockchip@Rockchip-pc:~/android-cts/tools$ ./cts-tradefed            ## CTS测试终端
Rockchip@Rockchip-pc:~/android-vts/tools$ ./vts-tradefed            ## VTS测试终端
Rockchip@Rockchip-pc:~/android-gts/tools$ ./gts-tradefed            ## GTS测试终端
Rockchip@Rockchip-pc:~/android-sts/tools$ ./sts-tradefed            ## STS测试终端
Rockchip@Rockchip-pc:~/android-apts-go/tools$ ./apts-tradefed    ## APTS测试终端
```

14.3 多台机器协作测试

下面测试大多支持多台机器协作测试，测试命令添加参数：`--shard-count`

```
run cts --shard-count 3 -s SN1 -s SN2 -s SN3                              ## 3台机器同时跑CTS
```

注意：多台机器协作测试时，如果其中一台机器掉电或死机则该台机器的报告将全部丢失，测试模块数量和总数量会异常！建议将机器调试稳定后，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！

14.4 跳过某个测试模块

单独跳过某个测试模块或测试项

```
run cts --exclude-filter "CtsViewTestCases"

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

14.5 压测某个测试模块

有些测试项概率性能测试Pass，可以按照下面命令循环压测某个测试模块或测试项

```
run cts -m module_name -t case_name --retry-strategy RETRY_ANY_FAILURE --max-testcase-run-count 20
```

14.6 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.img恢复出厂设置(否则机器开机可能卡在加载界面)，misc是固件一起编译出来的
fastboot flash misc misc.img

## VTS测试：GKI固件烧写vendor_boot-debug.img到vendor_boot分区；非GKI固件烧写boot-debug.img到boot分区
fastboot flash vendor_boot vendor_boot-debug.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2GB 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；(**机器不能锁定**)
- 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"
aosp-android14-gsi	Android14 PAB版GSI镜像下载地址 搜"aosp_arm64-img"
aosp-android15-gsi	Android15 PAB版GSI镜像下载地址 搜"signed-gsi_arm64-img"

14.7 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
```

- 机器锁定状态/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
```

14.8 VTS

需要额外烧写GSI(即AOSP的system-xxx-signed.img，可从Google官网/3PL或安全补丁FTP服务器获取，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`

14.9 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`

14.10 CTS-ON-GSI

需要额外烧写signed的GSI镜像(即AOSP的system-xxx-signed.img，可从Google官网/3PL或安全补丁FTP服务器)，烧写方法同上。

Android 15+版本OEM厂商测试

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`

14.11 CTS-ON-GSI-SOC

需要额外烧写signed的GSI镜像(即AOSP的system-xxx-signed.img，可从Google官网/3PL或安全补丁FTP服务器)，烧写方法同上。

Android 15+版本SOC原厂测试，OEM厂商不需要测试

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

14.12 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(环境配置说明网址)

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. 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+版本需要执行下面命令

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

4. Android 11+版本需要执行下面命令

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

5. Android 13+版本需要执行下面命令

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

6. Android 14+版本需要执行下面命令

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

7. 设置好系统的时间
8. 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
9. 测试报告路径：`/storage/emulated/0/VerifierReports/`

Controls Panel tests测试前需要先设置PIN锁屏密码

14.13 GTS

使用完整编译的user固件测试，固件配置按照文档<GMS固件配置>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：

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

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

- EDI白名单申请：将下面模块测试报告android-gts/results/2023.01.01_01.01.01.zip发给3PL上传到谷歌服务器申请白名单

```
run gts -m GtsEdiHostTestCases

机器报错: This device isn't Play Protect certified
测试报错: Test failed due to unrecognized service account for this product, please submit an
initial GTS report for this product with full EDI (run gts -m GtsEdiHostTestCases) at
https://partner.android.com/approvals/upload-report and try again.
```

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)
```

14.14 GTS-Verifier

测试说明：https://docs.partner.android.com/gms/testing/gts/gts-verifier?hl=en。Android 14+版本不再测试GTS-Verifier

14.15 GTS Interactive

Android 13+版本新增要求，替代GTS-Verifier手动测试项，测试过程根据机器界面提示信息操作

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

14.16 GTS Root

Android 15+版本新增要求，user固件解锁设备`unlock devices`和烧写`vendor_boot-debug.img`

没有烧写`vendor_boot-debug.img`跑测报错信息：

```
12-18 10:13:30 W/NativeDevice: 'adb root' on RK3326STB2 unsuccessful on attempt 1 of 3. Output:
'adbd cannot run as root in production builds
```

GTS-12-R3套件之前，**IR/LR**版本都需要测试`GTS Interactive`和`GTS Root`

GTS-12-R3套件之后，**IR/LR/MR**版本都需要测试`GTS Interactive`和`GTS Root`

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

14.17 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	`run sts-dynamic-full`	`run retry --retry 0`	`run sts-dynamic-full -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`

新测试套件要求下载最新的Ghidra工具

```
测试报错:
10-30 16:32:24 E/TestInvocation: Caught exception while running invocation
10-30 16:32:24 E/TestInvocation: Please manually download the latest version of Ghidra zip from: [https://github.com/NationalSecurityAgency/ghidra/releases] to /tmp/tradefed_ghidra/ghidra_11.2_PUBLIC_20240926.zip/ghidra_11.2_PUBLIC_20240926.zip. Make sure to rename the zip to ghidra_11.2_PUBLIC_20240926.zip. Ensure /tmp/ghidra_11.2_PUBLIC_20240926.zip/ghidra_11.2_PUBLIC_20240926.zip/ directory does not contain any other files.
```

14.18 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
```

14.19 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`

14.20 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。

Rockchip建议的BTS打包方法：将以下列表中的*.img镜像放到文件夹且命令为"Fingerprint"值(其中所有的":"和"/"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.img  dtbo.img  init_boot.img  misc.img  super.img  vbmeta.img  vendor_boot.img  vendor_boot-debug.img
```

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，上述boot-debug.img需要替换为vendor_boot-debug.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。

15. 本地媒体包测试

15.1 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. 模块单测命令(路径以实际位置为准):

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

15.2 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"
```

4. 模块单测命令(注意替换模块名`WvtsDeviceTestCases`)

```
run gts -m GtsExoPlayerTestCases \
--module-arg GtsExoPlayerTestCases:config-url:https://storage.googleapis.com/exoplayer-test-media-
1/gen-4/dynamic-config-sdcard-1.0.json \
--module-arg "GtsExoPlayerTestCases:instrumentation-arg:media-path:=/sdcard/gts"

run gts -m GtsYouTubeTestCases \
--module-arg GtsYouTubeTestCases:config-url:https://storage.googleapis.com/exoplayer-test-media-
1/gen-4/dynamic-config-sdcard-1.0.json \
--module-arg "GtsYouTubeTestCases:instrumentation-arg:media-path:=/sdcard/test"

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

常见GTS测试fail项需要用本地媒体包测试

```
WvtsDeviceTestCases
com.google.android.wvts.WidevineH264PlaybackTests#testL3WithUHD60
com.google.android.wvts.WidevineH264PlaybackTests#testClearWithUHD60
com.google.android.wvts.WidevineVP9WebMPlaybackTests#testVP9WebMCencSuperFramesL3WithUHD30
com.google.android.wvts.WidevineVP9WebMPlaybackTests#testVP9WebMProfile210bitCencSuperFramesL3With1080P30
com.google.android.wvts.WidevineVP9WebMPlaybackTests#testVP9WebMCencSubSampleL3WithUHD60
```

16. 用户客制修改

16.1 更新GMS包版本

```
Android_SDK$ unzip -o gms-oem-U-14-202403.zip -d vendor/
Android_SDK/vendor/partner_gms$ find . -name "*x86*apk" | xargs rm -rf   ## 删除32-bit应用
Android_SDK/vendor/partner_gms$ find . -name "*.apk" | xargs git add
Android_SDK/vendor/partner_gms$ git status ./                           ## 查到本地修改差异，手动比对再合入
```

16.2 查看系统配置项

FTP服务器/GMS-Test-Suite/xTS/tools/OverlayTools.apk可查看系统/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
```

16.3 查看RKSDK版本

获取属性`getprop | grep ro.rksdk.version`

```
device/rockchip/common/device.mk:           ro.rksdk.version=ANDROID$(PLATFORM_VERSION)_RKR5
```

16.4 机器SN码命名规范

烧写SN码时机器要进入bootloader状态，写号工具在SDK目录

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

GMS认证要求SN码以字母开头且长度为7-bit ASCII，RK限制长度在6~14个字节以内，SN码格式要求匹配正则表达式"^[A-Za-z0-9]+$"，**且不能包含"."","_"等符号**

```
hardware/rockchip/drmservice/Android.mk:
# Verify sn with pattern "^[A-Za-z0-9]+$"
ifeq ($(BUILD_WITH_GOOGLE_MARKET), true)
LOCAL_CFLAGS += -DENABLE_SN_VERIFY
endif

hardware/rockchip/drmservice/drmservice.c:
#define SERIALNO_PATTERN "^[A-Za-z0-9]+$"
#define SERIALNO_BUF_LEN 33
```

```
CtsAppSecurityHostTestCases
android.appsecurity.cts.AccessSerialNumberTest#testGetSerialReturnsExpectedFormat

CtsUsbTests
com.android.cts.usb.TestUsbTest#testUsbSerialReadOnDeviceMatches
```

16.5 修改系统API等级

通过属性`getprop | grep 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 := 34
build/make/core/main.mk:
     ifdef PRODUCT_SHIPPING_API_LEVEL
          ADDITIONAL_VENDOR_PROPERTIES +=
ro.product.first_api_level=$(PRODUCT_SHIPPING_API_LEVEL)
      endif

device/rockchip/rk3588/BoardConfig.mk:                BOARD_SHIPPING_API_LEVEL := 34
build/make/core/main.mk:
     ifdef BOARD_SHIPPING_API_LEVEL
          ADDITIONAL_VENDOR_PROPERTIES += ro.board.first_api_level=$(BOARD_SHIPPING_API_LEVEL)
      endif
```

LR版本需要保持宏`PRODUCT_SHIPPING_API_LEVEL`值不变，比如Android 13版本LR升级到Android 14版本，配置宏`PRODUCT_SHIPPING_API_LEVEL := 33`

Android 14+版本RK各芯片平台添加GRF功能，需要配置上述宏`BOARD_SHIPPING_API_LEVEL`，否则会有下面fail项

```
GtsOsTestCases
com.google.android.os.gts.GoogleRequirementFreezeTest#testChipsetInfo
```

Android 13及以前版本不属于Android 14 GRF范畴，不需要配置`BOARD_SHIPPING_API_LEVEL`宏，包括Android 13版本LR升级到Android 14版本也需要删掉该宏

```
build/make/core/main.mk:

# Vendors with GRF must define BOARD_SHIPPING_API_LEVEL for the vendor API level.
# This must not be defined for the non-GRF devices.
ifdef BOARD_SHIPPING_API_LEVEL
ADDITIONAL_VENDOR_PROPERTIES += \
    ro.board.first_api_level=$(BOARD_SHIPPING_API_LEVEL)
```

16.6 修改系统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

报错信息：FAILURE: junit.framework.AssertionFailedError: 36 failures in theme test
```

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)按照下面配置，且需要同时保证`PRODUCT_AAPT_CONFIG`列表包含tvdpi

```
device/rockchip/rk3588/rk3588_t/rk3588_t.mk:     PRODUCT_PROPERTY_OVERRIDES +=
ro.sf.lcd_density=213
device/rockchip/rk3588/rk3588_t/rk3588_t.mk:     PRODUCT_AAPT_PREF_CONFIG := tvdpi
device/rockchip/common/device.mk:                      PRODUCT_AAPT_CONFIG ?= normal large xlarge
hdpi tvdpi xhdpi xxhdpi
```

16.7 修改系统clientidbase属性

根据MADA/EDLA资质修改ro.com.google.clientidbase属性，**请不要使用我司的资质"android-rockchip"**

```
Android_SDK/vendor/rockchip/common$ git diff gms/gms.prop
+ro.com.google.clientidbase=android-rockchip

编译结果：out/target/product/rk3588s_t/system/build.prop
```

```
device/rockchip/common/modules/gms.mk:
ifeq ($(strip $(BUILD_WITH_GOOGLE_MARKET)), true)
  ......
   $(warning Please set client id with your own MADA ID!)
  TARGET_SYSTEM_PROP += vendor/rockchip/common/gms/gms.prop
```

16.8 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/product.mk:
     _product_single_value_vars += \
           PRODUCT_SYSTEM_NAME \
           PRODUCT_SYSTEM_MODEL \
           PRODUCT_SYSTEM_DEVICE \
           PRODUCT_SYSTEM_BRAND \
           PRODUCT_SYSTEM_MANUFACTURER \

build/make/core/sysprop.mk:
      $(if $(filter system,$(1)),\
            echo "ro.product.$(1).brand=$(PRODUCT_SYSTEM_BRAND)" >> $(2);\
            echo "ro.product.$(1).device=$(PRODUCT_SYSTEM_DEVICE)" >> $(2);\
            echo "ro.product.$(1).manufacturer=$(PRODUCT_SYSTEM_MANUFACTURER)" >> $(2);\
            echo "ro.product.$(1).model=$(PRODUCT_SYSTEM_MODEL)" >> $(2);\
            echo "ro.product.$(1).name=$(PRODUCT_SYSTEM_NAME)" >> $(2);\
        ,\
            echo "ro.product.$(1).brand=$(PRODUCT_BRAND)" >> $(2);\
            echo "ro.product.$(1).device=$(TARGET_DEVICE)" >> $(2);\
            echo "ro.product.$(1).manufacturer=$(PRODUCT_MANUFACTURER)" >> $(2);\
            echo "ro.product.$(1).model=$(PRODUCT_MODEL)" >> $(2);\
            echo "ro.product.$(1).name=$(TARGET_PRODUCT)" >> $(2);\             ## 工程lunch选项结
果

build/make/core/product_config.mk:
     TARGET_DEVICE := $(PRODUCT_DEVICE)                                        ## 工程
lunch选项结果

     ifndef PRODUCT_SYSTEM_NAME
        PRODUCT_SYSTEM_NAME := $(PRODUCT_NAME)
      endif
     ifndef PRODUCT_SYSTEM_DEVICE
        PRODUCT_SYSTEM_DEVICE := $(PRODUCT_DEVICE)
      endif
     ifndef PRODUCT_SYSTEM_BRAND
        PRODUCT_SYSTEM_BRAND := $(PRODUCT_BRAND)
      endif
     ifndef PRODUCT_SYSTEM_MODEL
        PRODUCT_SYSTEM_MODEL := $(PRODUCT_MODEL)
      endif
     ifndef PRODUCT_SYSTEM_MANUFACTURER
        PRODUCT_SYSTEM_MANUFACTURER := $(PRODUCT_MANUFACTURER)
      endif

device/rockchip/rk3588/rk3588_t/rk3588_t.mk:
     PRODUCT_NAME := rk3588_t
     PRODUCT_DEVICE := rk3588_t
```

```
    PRODUCT_BRAND := rockchip                                                    ##
修改PRODUCT_BRAND值
    PRODUCT_MODEL := rk3588_t
    PRODUCT_MANUFACTURER := rockchip

编译结果
out/target/product/rk3588s_t/system/build.prop
out/target/product/rk3588s_t/vendor/build.prop
```

16.9 修改系统Fingerprint指纹

```
Android14/build/make/core/sysprop.mk:
ifeq (,$(strip $(BUILD_FINGERPRINT)))
  ifdef ROCKCHIP_BUILD_NUMBER
    BF_BUILD_NUMBER := $(ROCKCHIP_BUILD_NUMBER)
   else
     ifeq ($(strip $(HAS_BUILD_NUMBER)),false)
       BF_BUILD_NUMBER := $(BUILD_USERNAME)$$($(DATE_FROM_FILE) +%m%d%H%M)
     else
       BF_BUILD_NUMBER := $(file <$(BUILD_NUMBER_FILE))
     endif
   endif
  BUILD_FINGERPRINT :=
$(PRODUCT_BRAND)/$(TARGET_PRODUCT)/$(TARGET_DEVICE):$(PLATFORM_VERSION)/$(BUILD_ID)/$(BF_BUILD_NUM
BER):$(TARGET_BUILD_VARIANT)/$(BUILD_VERSION_TAGS)
endif

Android15/build/make/core/sysprop.mk:
ifeq (,$(strip $(BUILD_FINGERPRINT)))
  BUILD_FINGERPRINT :=
$(PRODUCT_BRAND)/$(TARGET_PRODUCT)/$(TARGET_DEVICE):$(PLATFORM_VERSION)/$(BUILD_ID)/$(BUILD_NUMBER
_FROM_FILE):$(TARGET_BUILD_VARIANT)/$(BUILD_VERSION_TAGS)
endif
```

系统Fingerprint指纹定义如上，若需要修改各字段则按照下面修改相关宏定义

```
device/rockchip/rk3588/rk3588_t/rk3588_t.mk:           PRODUCT_BRAND := rockchip

build/make/envsetup.sh:                                      TARGET_PRODUCT=$product \

build/make/core/product_config.mk:                       TARGET_DEVICE := $(PRODUCT_DEVICE)
device/rockchip/rk3588/rk3588_t/rk3588_t.mk:           PRODUCT_DEVICE := rk3588_t

build/make/core/version_util.mk:                         PLATFORM_VERSION :=
$(PLATFORM_VERSION_LAST_STABLE)
build/make/core/version_defaults.mk:                   PLATFORM_VERSION_LAST_STABLE := 13

build/make/core/build_id.mk:                             BUILD_ID=TQ3C.230805.001.B2
build/make/envsetup.sh:                                     TARGET_BUILD_VARIANT=$variant \
```

说明：其中"TARGET_PRODUCT"和"TARGET_BUILD_VARIANT"宏是工程lunch选项结果，即"lunch <product_name>-<build_variant>"

- Google要求过MADA/EDLA认证产品的Fingerprint必须是唯一的(相关说明网址)，且拿到认证后Fingerprint不能修改。
- Google要求Fingerprint长度为7-bit ASCII，不能包含" "空格且第一个字段PRODUCT_BRAND不能包含"_"下划线，否则Google Play Store识别失败

```
CtsOsTestCases
android.os.cts.BuildVersionTest#testBuildFingerprint

当前Fingerprint值：
Rockchip_Worship_Player/rk3399_t/rk3399_t:13/TQ2B.230505.005.A1/202303:user/release-keys
报错信息：junit.framework.ComparisonFailure: expected:<Rockchip[ Worship ]Player> but was:
<Rockchip[_Worship_]Player>
错误说明：第一个字段"PRODUCT_BRAND := Rockchip_Worship_Player"不能包含"_"下划线
```

16.10 修改系统data分区文件格式

- MADA/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`且分区加密

```
$ 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(即userdata)必须使用f2fs文件格式(分区文件格式要求)

```
CtsSecurityHostTestCases
android.security.cts.MetadataEncryptionTest#testMetadataEncryptionIsEnabled

报错信息：java.lang.AssertionError: Metadata encryption must be enabled
解决方法：GMS认证要求userdata分区必须添加加密标识，默认userdata分区格式f2fs已经加密
fileencryption=aes-256-xts:aes-256-
cts:v2+inlinecrypt_optimized,keydirectory=/metadata/vold/metadata_encryption,checkpoint=fs

标识含义
fileencryption=aes-256-xts:aes-256-cts:v2+inlinecrypt_optimized  表示加密算法和优化选项的组合
keydirectory=/metadata/vold/metadata_encryption  表示密钥文件的存储目录
checkpoint=fs  表示UDC机制(User Data Checkpoint)即支持data分区回滚
```

注意：开启虚拟A/B后系统无法开机，需要保证userdata分区格式一致

```
报错信息：DEBUG: Cmdline: /system/bin/credstore /data/misc/credstore
DEBUG: pid: 720, tid: 720, name: credstore  >>> /system/bin/credstore <<<
DEBUG: uid: 1076
DEBUG: tagged_addr_ctrl: 0000000000000001 (PR_TAGGED_ADDR_ENABLE)
DEBUG: signal 6 (SIGABRT), code -1 (SI_QUEUE), fault addr --------
DEBUG: Abort message: 'Check failed: chdir(data_dir.c_str()) != -1 chdir: /data/misc/credstore: No
such file or directory'
```

```
解决方法:需要保证下面3个文件配置一致(同时为f2fs或ext4)
device/rockchip/rk3588/device.mk:
BOARD_USERDATAIMAGE_FILE_SYSTEM_TYPE := f2fs

device/rockchip/rk3588/rk3588_t/recovery.fstab_AB:
/dev/block/by-name/userdata     /data     f2fs     defaults     defaults

device/rockchip/common/scripts/fstab_tools/fstab.in:
/dev/block/by-name/userdata     /data     f2fs
noatime,nosuid,nodev,discard,reserve_root=32768,resgid=1065
```

16.11 Camera相关修改

Camera相关feature说明:

```
console:/ $ pm list features | grep camera
feature:android.hardware.camera                                    ## 后摄Camera
feature:android.hardware.camera.any                                ## 必须(只要有
Camera必须申明该feature)
feature:android.hardware.camera.external                           ## Usb Camera
feature:android.hardware.camera.front                              ## 前摄Camera
```

若需要将Usb Camera配成前摄或后摄，按照下面修改

```
Android_SDK/hardware/interfaces/camera$ git diff device/3.4/default/ExternalCameraDevice.cpp
-    const uint8_t facing = ANDROID_LENS_FACING_EXTERNAL;
+    const uint8_t facing = ANDROID_LENS_FACING_FRONT;              ## 后摄配置
ANDROID_LENS_FACING_BACK
      UPDATE(ANDROID_LENS_FACING, &facing, 1);
```

- 测试项BYOD Managed Provisioning - Camera support cross profile video capture (with extra output path)报错：空指针挂死

```
解决方法：Android_SDK/packages/apps/Camera2$ git diff src/com/android/camera/VideoModule.java
      if (mIsVideoCaptureIntent) {
            // if no file save is needed, we can show the post capture UI now
            showCaptureResult();
-    }
-    if (mVideoFileDescriptor != null) {
+    }else if (mVideoFileDescriptor != null) {
            saveVideo();
      }
```

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

```
解决方法：Android_SDK/packages/apps/Camera2$ git diff src/com/android/camera/VideoModule.java
       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
```

若机器没有Camera则CTS-Verifier要求不能出现下面3个测试项

```
BYOD Managed Provisioning - Camera support cross profile image capture
BYOD Managed Provisioning - Camera support cross profile video capture (with extra output path)
BYOD Managed Provisioning - Camera support cross profile video capture (without extra output path)
```

解决方法：移除上面Camera相关feature申明，移除Camera2应用

```
Android_SDK/build/make$ git diff target/product/handheld_product.mk
 PRODUCT_PACKAGES += \
       Browser2 \
       Calendar \
-      Camera2 \

Android_SDK/device/rockchip/common$ git diff
./overlay/frameworks/base/core/res/res/values/config.xml
       <!-- Whether this device is supporting the software microphone toggle -->
       <bool name="config_supportsMicToggle">true</bool>
       <!-- Whether this device is supporting the camera toggle -->
-      <bool name="config_supportsCamToggle">true</bool>
+      <bool name="config_supportsCamToggle">false</bool>
```

上述修改可以通过命令查询`adb shell cmd overlay lookup android android:bool/config_supportsCamToggle`

16.12 修改Usb Mic切成内置Mic

机器必须有Microphone，可以是内置Mic或Usb Mic，若只有Usb Mic则必须切成内置Mic

16.13 机器不带电池需要打补丁

- EDLA设备大多不带电池，若机器不带电池需要打FTP服务器补丁/GMS-Test-Suite/Exp+/batteryless.diff，相关文档说明Supporting Batteryless Devices
- 打完上述补丁编译kernel后GKI固件需要执行拷贝替换命令("kernel-5.10"根据实际kernel版本替换)

```
mv kernel-5.10/drivers/power/supply/test_power.ko
mkcombinedroot/vendor_ramdisk/lib/modules/test_power.ko
```

```
CtsStatsdAtomHostTestCases
android.cts.statsdatom.statsd.HostAtomTests#testFullBatteryCapacity
报错信息：FAILURE: expected to be greater than: 0        but was: 0
解决方法：保证电池驱动节点/sys/class/power_supply/***battery/charge_full值大于0，即配
置"battery_present=0"
```

```
CtsStatsdAtomHostTestCases
android.cts.statsdatom.statsd.HostAtomTests#testRemainingBatteryCapacity
报错信息：FAILURE: expected to be greater than: 0        but was: 0
解决方法：保证电池驱动节点/sys/class/power_supply/test_battery/charge_counter值大于0

CtsStatsdHostTestCases
android.cts.statsd.validation.ValidationTests#testPartialWakelockDuration
报错信息：FAILURE: No wakelocks with uid 10163 in statsd
解决方法：无电池产品需要保证节点值为1
cat /sys/class/power_supply/test_usb/online

hcq@sys2206:~/3_Android13_0321/kernel-5.10$ git diff drivers/power/supply/test_power.c
-static int usb_online                          = 0;
+static int usb_online                          = 1;
 static int battery_status                     = POWER_SUPPLY_STATUS_UNKNOWN;
```

```
VtsHalHealthV2_1TargetTest
PerInstance/HealthHidlTest#getHealthInfo_2_1/0_default

报错信息：Expected: ((long)value.batteryFullChargeDesignCapacityUah) > (((long)100 * 1000)),
actual: 100 vs 100000
          batteryFullChargeDesignCapacityUah should be greater than 100 mAh, or 0 if unknown
```

```
解决方法：Android_SDK/kernel-5.10/drivers/power/supply$ git diff test_power.c
-static int battery_status                    = POWER_SUPPLY_STATUS_DISCHARGING;
+static int battery_status                    = POWER_SUPPLY_STATUS_UNKNOWN;
 static int battery_health                    = POWER_SUPPLY_HEALTH_GOOD;
-static int battery_present                   = 1; /* true */
+static int battery_present                   = 0; /* true */                ## 该修改后电池图标显
示"？"

          case POWER_SUPPLY_PROP_CHARGE_FULL_DESIGN:
          case POWER_SUPPLY_PROP_CHARGE_FULL:
-                    val->intval = 100;
+                    val->intval = 0;
```

```
VtsHalHealthTargetTest
Health/HealthAidl#getChargeCounterUah/0_android_hardware_health_IHealth_default
报错信息：FAILURE: hardware/interfaces/health/aidl/vts/functional/VtsHalHealthTargetTest.cpp:170:
Failure
          Value of: value
          Expected: is >= 0
             Actual: -1000

Health/HealthAidl#getHealthInfo/0_android_hardware_health_IHealth_default
报错信息：FAILURE: hardware/interfaces/health/aidl/vts/functional/VtsHalHealthTargetTest.cpp:413:
Failure
          Value of: value
          Expected: is valid health info
             Actual: 200-byte object <01-01 00-00 A0-25 26-00 40-4B 4C-00 03-00 00-00 02-00 00-00
01-00 00-00

Health/BatteryTest#ConnectedAgainstStatusFromHal/0_android_hardware_health_IHealth_default
```

```
报错信息：FAILURE: hardware/interfaces/health/aidl/vts/functional/VtsHalHealthTargetTest.cpp:592:
Failure
          Value of: SucceedOnce(gBatteryTestTimeout, testOnce)
             Actual: false (BatteryStatus is DISCHARGING and power source is connected: ac=1,
usb=1, wireless=0)
          Expected: true
          You may want to try again later when battery_status becomes stable.

Health/BatteryTest#ConnectedAgainstStatusInHealthInfo/0_android_hardware_health_IHealth_default
报错信息：FAILURE: hardware/interfaces/health/aidl/vts/functional/VtsHalHealthTargetTest.cpp:604:
Failure
          Value of: SucceedOnce(gBatteryTestTimeout, testOnce)
             Actual: false (BatteryStatus is DISCHARGING and power source is connected: ac=1,
usb=1, wireless=0)
          Expected: true
          You may want to try again later when getHealthInfo becomes stable.
```

16.14 更新固件签名/Apex签名脚本

应用签名密钥配置逻辑

```
packages/apps/rkCamera2/Android.mk
LOCAL_CERTIFICATE := platform
或
packages/apps/Settings/Android.bp
certificate: "platform"

若应用没有指定'LOCAL_CERTIFICATE'或'certificate'值，则按照下面默认配置应用签名密钥
device/rockchip/common/BoardConfig.mk
PRODUCT_DEFAULT_DEV_CERTIFICATE := build/make/target/product/security/releasekey

build/make/core/config.mk
# The default key if not set as LOCAL_CERTIFICATE
ifdef PRODUCT_DEFAULT_DEV_CERTIFICATE
  DEFAULT_SYSTEM_DEV_CERTIFICATE := $(PRODUCT_DEFAULT_DEV_CERTIFICATE)
else
  DEFAULT_SYSTEM_DEV_CERTIFICATE := build/make/target/product/security/testkey      ## 直接修改可
能不生效
endif
.KATI_READONLY := DEFAULT_SYSTEM_DEV_CERTIFICATE
```

查看应用签名密钥信息

```
Android_SDK$ keytool -printcert -jarfile SearchLauncherQuickStep.apk
```

Rockchip提供脚本更新固件签名/Apex签名：`/GMS-Test-Suite/xTS/tools/gms_certificate_tools.zip`

```
Android_SDK/vendor$ unzip gms_certificate_tools.zip
Android_SDK/vendor$ cd gms_tools/
Android_SDK/vendor/gms_tools$ ./apex_sig_gen.sh                    ## 提示输入密码"Enter
password for"时直接Enter键
Android_SDK/vendor/gms_tools$ ./firmware_sig_gen.sh                ## 提示输入密码"Enter
password for"时直接Enter键
```

固件签名/Apex签名相关测试项

```
CtsAppSecurityHostTestCases
android.appsecurity.cts.ApexSignatureVerificationTest#testApexPubKeyIsNotWellKnownKey

GtsApexSignatureVerificationTest
android.appsecurity.cts.ApexSignatureVerificationTest#testApexPubKeyIsNotWellKnownKey

CtsSecurityTestCases
android.security.cts.PackageSignatureTest#testPackageSignatures

GtsPackageSignatureTest
android.security.cts.PackageSignatureTest#testPackageSignatures

GtsPermissionTestCases
com.google.android.permission.gts.DefaultPermissionGrantPolicyTest#testDefaultGrantsWithRemoteExce
ptions
com.google.android.permission.gts.DefaultPermissionGrantPolicyTest#testPreGrantsWithRemoteExceptio
ns
```

注意：替换Apex签名仍然测试fail，关闭Ubuntu主机终端再打开(Apex签名文件缓存目录`/tmp/ApexSignatureVerificationTest***`)

17. 测试常见问题

17.1 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(MediaParserHostSideTes
t.java:294)
    at
android.media.mediaparser.cts.MediaParserHostSideTest.setUp(MediaParserHostSideTest.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
```

17.2 VTS很多module没法跑测

- 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。

17.3 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
```

17.4 通过认证后，无法显示已认证问题

查看机器认证状态：Play Store -> Settings -> About -> Play Protect certification

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
```

17.5 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?
```

```
GtsFeaturesTestCases
com.google.android.features.gts.GoogleFeaturesTest#testFeatureTurboPreload
报错信息1：Device Health Services was missing but the "com.google.android.feature.TURBO_PRELOAD" feature was found.
          expected to be false
报错信息2：Device Health Services was found but the "com.google.android.feature.TURBO_PRELOAD" feature was not, please reach out to your Google contact

解决方法：同时集成应用Turbo.apk和TurboOverlay.apk，或同时不集成应用Turbo.apk和TurboOverlay.apk，两者是共存关系
若同时不集成应用Turbo.apk和TurboOverlay.apk，则需要添加如下修改删除"com.google.android.feature.TURBO_PRELOAD"
Android_SDK/vendor/partner_gms$ git diff etc/sysconfig/Android.bp
-prebuilt_etc {
-     name: "sysconfig_turbo",
-     product_specific: true,
-     sub_dir: "sysconfig",
-     src: "turbo.xml",
-     filename_from_src: true,
-}
```

- 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`，若查询结果为空需要确保安装TurboOverlay.apk
- `pm list package -f | grep turbo` 查询是否安装应用：Turbo.apk、TurboOverlay.apk

```
package:/product/priv-app/Turbo/Turbo.apk=com.google.android.apps.turbo
package:/product/overlay/TurboOverlay.apk=com.rockchip.android.overlay.turbo
```

- `pm list features | grep com.google.android.feature.TURBO_PRELOAD` 查询feature是否添加
- 需要保证config_autoBrightnessLevels和config_autoBrightnessDisplayValuesNits数组长度相同(工具OverlayTools.apk)

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

或
cmd overlay lookup android android:array/config_autoBrightnessLevels
cmd overlay lookup android android:array/config_autoBrightnessDisplayValuesNits
```

```
## 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
```

注意：应用Turbo.apk属于Google Device Health Services功能，是GMS Requirements的可选功能。

17.6 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

或
$ adb remount
restarting adbd as root
Device must be bootloader unlocked
```

17.7 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：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：Android_SDK/vendor/partner_gms/products$ cp gms.mk gms-mandatory.mk
           Android_SDK/vendor/partner_gms/products$ git diff gms-mandatory.mk
-# GMS comms suite
-$(call inherit-product, $(ANDROID_PARTNER_GMS_HOME)/products/google_comms_suite.mk)
```

- 编译报错BOARD_SUPER_PARTITION_WARN_LIMIT：修改super分区大小(必须是4MB的整数倍)

```
报错信息：RuntimeError: sum of sizes of ['system', 'system_dlkm', 'system_ext', 'vendor', 'vendor_dlkm', 'odm', 'odm_dlkm', 'product'] is greater than BOARD_SUPER_PARTITION_SIZE: 977747968+15765504+207147008+251867136+8318976+897024+348160+2484719616 == 3946811392 > 3932160000 == 3932160000
[ 99% 40908/40909] Target super fs image for debug: out/target/product/rk3588s_s/super.img
```

```
解决方法：Android_SDK/device/rockchip/common$ git diff BoardConfig_AB.mk
    ifeq ($(PRODUCT_USE_DYNAMIC_PARTITIONS), true)
        ifneq ($(PRODUCT_RETROFIT_DYNAMIC_PARTITIONS), true)
            ifeq ($(BOARD_ROCKCHIP_VIRTUAL_AB_ENABLE), true)
                ifeq ($(BUILD_WITH_GO_OPT), true)
                    ifeq ($(strip $(TARGET_ARCH)), arm64) # arm64 go
                        BOARD_SUPER_PARTITION_SIZE := 2390753280
                    else # arm go
                        BOARD_SUPER_PARTITION_SIZE := 1971322880
                    endif
                else # non-go
-                       BOARD_SUPER_PARTITION_SIZE := 3932160000
+                       BOARD_SUPER_PARTITION_SIZE := 3963617280
                endif
                BOARD_ROCKCHIP_DYNAMIC_PARTITIONS_SIZE := $(shell expr $(BOARD_SUPER_PARTITION_SIZE) - 4194304)
```

- 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：Android_SDK/vendor/partner_gms$ git diff apps_go/FilesGoogle/Android.mk
-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：Android_SDK/vendor/partner_gms$ git diff apps_go/FilesGoogle/Android.mk
-LOCAL_OPTIONAL_USES_LIBRARIES := org.apache.http.legacy
+LOCAL_ENFORCE_USES_LIBRARIES := false
```

- 编译报错manifest_level_32.xml

```
报错信息：FAILED: ninja: 'device/rockchip/common/manifests/manifest_level_32.xml', needed by
'out/target/product/rk3588_s/gen/ETC/vendor_manifest.xml_intermediates/manifest.xml', missing and
no known rule to make it

解决方法：first_api_level=31是Android12.0      first_api_level=32是Android12.1
Android_SDK/device/rockchip/common/manifests$ cp manifest_level_31.xml manifest_level_32.xml
```

- 集成Mainline包后编译报错：Target of symlink is not executable

```
报错信息1：warning:  Missing compilation artifacts. Dexpreopting is not working for some system
server jars
报错信息2：out/host/linux-x86/obj/EXECUTABLES/dexdump_intermediates/dexdump: error: Target of
symlink is not executable
             out/host/linux-x86/obj/EXECUTABLES/hiddenapi_intermediates/hiddenapi: error: Target
of symlink is not executable

解决方法：chmod a+x prebuilts/module_sdk/art/current/host-exports/x86_64/bin/*
           chmod a+x prebuilts/module_sdk/art/go/current/host-exports/x86_64/bin/*
```

17.8 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
CtsNetTestCases
android.net.cts.MultinetworkApiTest#testResNApi

CtsLibcoreTestCases
android.libcore.cts libcore.java.net.ConcurrentCloseTest#test_connect_timeout

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

GtsExoPlayerTestCases
com.google.android.exoplayer.gts.DashDownloadTest#download                    ## 该项不能使用本地
媒体包
java.lang.ClassCastException: sun.net.www.protocol.file.FileURLConnection cannot be cast to
java.net.HttpURLConnection
```

```
## IPv6
CtsNetTestCases
android.net.cts.DnsTest#testDnsWorks
android.net.cts.DnsResolverTest#testQueryForInetAddressIpv6Inline

CtsLibcoreTestCases
libcore.java.net.InetAddressTest#test_getByName_invalid[1]
libcore.java.net.SocketTest#testSocketTestAllAddresses
```

17.9 CTS常见fail项处理

- ConfigurationContainerProto报错：测试包版本不匹配

```
报错信息：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
```

- 测试项testGetMemoryClass报错：虚拟机内存dalvik.vm.heapgrowthlimit不满足当前机器dpi=640值所需虚拟机内存

```
CtsAppTestCases
android.app.cts.ActivityManagerMemoryClassTest#testGetMemoryClass

报错信息：AssertionFailedError: Expected to have at least 512mb of memory for screen density 640
解决方法：根据报错修改虚拟机内存dalvik.vm.heapgrowthlimit=512m，修改后机器串口查看"getprop | grep dalvik.vm.heap"
```

```
Android_SDK/frameworks/native/build$ git diff tablet-10in-xhdpi-2048-dalvik-heap.mk
 PRODUCT_VENDOR_PROPERTIES += \
        dalvik.vm.heapstartsize?=16m \
-     dalvik.vm.heapgrowthlimit?=192m \
+     dalvik.vm.heapgrowthlimit?=512m \
```

- 测试项testSensorFeatures报错：Accelerometer sensor软件/硬件/feature不匹配导致

```
CtsAppTestCases
android.app.cts.SystemFeaturesTest#testSensorFeatures

报错信息：java.lang.AssertionError:
PackageManager#hasSystemFeature(android.hardware.sensor.accelerometer) returns false but SensorManager#getSensorList(1) shows sensors [Accelerometer sensor] expected:<false> but was:<true>
解决方法：若硬件有"Accelerometer sensor"传感器则软件添加
feature:android.hardware.sensor.accelerometer
          若硬件没"Accelerometer sensor"传感器则软件关闭宏"BOARD_GRAVITY_SENSOR_SUPPORT := false"
相关逻辑：
hardware/rockchip/sensor/st/Android.mk
```

```
    ifeq ($(BOARD_GRAVITY_SENSOR_SUPPORT), true)
        LOCAL_CFLAGS += -DGRAVITY_SENSOR_SUPPORT
    endif

hardware/rockchip/sensor/st/sensors.c
    static const struct sensor_t sSensorList[] = {
        #ifdef GRAVITY_SENSOR_SUPPORT
        { .name          = "Accelerometer sensor",
          .vendor        = "The Android Open Source Project",
          .version       = 1,
          .handle        = SENSORS_HANDLE_BASE+ID_A,
          .type          = SENSOR_TYPE_ACCELEROMETER,
```

```
传感器软件宏控制
BOARD_GRAVITY_SENSOR_SUPPORT := false
BOARD_COMPASS_SENSOR_SUPPORT := false
BOARD_GYROSCOPE_SENSOR_SUPPORT := false
BOARD_PROXIMITY_SENSOR_SUPPORT := false
BOARD_LIGHT_SENSOR_SUPPORT := false
BOARD_PRESSURE_SENSOR_SUPPORT := false
BOARD_TEMPERATURE_SENSOR_SUPPORT := false
```

- 测试项testGenerateKeyWhenCredentialAndBiometricEnrolled报错：测试前需要保证机器PIN锁屏功能正常，即正常锁屏和解锁

```
CtsBiometricsTestCases
android.server.biometrics.BiometricCryptoTests#testGenerateKeyWhenCredentialAndBiometricEnrolled

报错信息：java.lang.IllegalStateException: Secure lock screen must be enabled to create keys requiring user authentication
解决方法：不能禁用系统锁屏功能，即需要配置项<bool name="def_lockscreen_disabled">false</bool>保留默认值
```

- 测试项testManagedProfileSupported报错：2GB的Go版本需要配置feature，但1.5GB及以下Go版本不需要配置

```
CtsContentTestCases
android.content.pm.cts.FeatureTest#testManagedProfileSupported

报错信息：Device should support managed profiles, but android.software.managed_users is not enabled
```

```
Android_SDK/device/rockchip/common$ git diff device.mk
 ifneq ($(strip $(BUILD_WITH_GO_OPT)),true)
       PRODUCT_COPY_FILES += \

frameworks/native/data/etc/tablet_core_hardware.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/tablet_core_hardware.xml
+else
+     PRODUCT_COPY_FILES += \
+
frameworks/native/data/etc/android.software.managed_users.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.software.managed_users.xml
```

- 测试项CtsDeqpTestCases报错：Abort: Test cannot be executed

```
CtsDeqpTestCases
dEQP-EGL.info#version

报错信息：Abort: Test cannot be executed
           dEQP  : Writing test log into /sdcard/TestLog.qpa
           dEQP  : ERROR: Unable to open test log output file '/sdcard/TestLog.qpa'.
           dEQP  : Failed to create activity: Failed to open test log file '/sdcard/TestLog.qpa'

解决方法：按照"主机配置"章节配置aapt/aapt2工具
CtsDeqpTestCases模块相关报错请先查看Mali库版本：getprop | grep sys.gmali.version

$ aapt version
Android Asset Packaging Tool, v0.2-6739378
$
$ aapt2 version
Android Asset Packaging Tool (aapt) 2.19-eng.20249
$
$ which aapt
/home/hcq/Software/sdk_tools/aapt
$
$ which aapt2
/home/hcq/Software/sdk_tools/aapt2
```

- 测试项setDefaultSmsApplication_notSmsCapable_unchanged报错：overlay配置项config_defaultSms

```
CtsDevicePolicyTestCases
android.devicepolicy.cts.DefaultSmsApplicationTest#setDefaultSmsApplication_notSmsCapable_unchanged[IncludeRunOnDeviceOwnerUser]

报错信息：expected: com.android.messaging
           but was : com.android.bedstead.testapp.SmsApp
```

```
解决方法：Android_SDK/device/rockchip/common$ git diff
overlay/frameworks/base/core/res/res/values/config.xml
 <resources>
+     <!-- The name of the package that will hold the SMS role by default. -->
+     <!-- Set it empty here -->
+     <string name="config_defaultSms" translatable="false"></string>
```

- 测试项testOpenGlEsDeqpLevel报错：根据报错信息更新android.software.opengles.deqp.level

```
CtsGraphicsTestCases
android.graphics.cts.OpenGlEsDeqpLevelTest#testOpenGlEsDeqpLevel

报错信息：Feature android.software.opengles.deqp.level must be present and have at least version 132383489.
解决方法：pm list features | grep android.software.opengles.deqp.level

Android_SDK/frameworks/native/data/etc$ cat android.software.opengles.deqp.level-2022-03-01.xml
<permissions>
    <feature name="android.software.opengles.deqp.level" version="132514561" />
</permissions>
```

```
Android_SDK/device/rockchip/common$ git diff modules/gms.mk
 ifneq ($(filter true yes, $(BUILD_WITH_GOOGLE_MARKET) $(PRODUCT_USE_PREBUILT_GTVS)),)
    PRODUCT_COPY_FILES += \
-     frameworks/native/data/etc/android.software.opengles.deqp.level-2021-03-01.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.software.opengles.deqp.level-2020-03-01.xml
+     frameworks/native/data/etc/android.software.opengles.deqp.level-2022-03-01.xml:$(TARGET_COPY_OUT_VENDOR)/etc/permissions/android.software.opengles.deqp.level-2022-03-01.xml
```

- 测试项testAllKeys报错：Android13平台开启GKI

```
CtsHardwareTestCases
android.hardware.input.cts.tests.SonyDualSenseBluetoothTest#testAllKeys
android.hardware.input.cts.tests.SonyDualSenseBluetoothTest#testVibrator
android.hardware.input.cts.tests.SonyDualSenseBluetoothTest#testAllMotions
android.hardware.input.cts.tests.SonyDualSenseUsbTest#testAllKeys
android.hardware.input.cts.tests.SonyDualSenseUsbTest#testVibrator
android.hardware.input.cts.tests.SonyDualSenseUsbTest#testAllMotions

报错信息：java.lang.RuntimeException: Did not receive device added notification in time
解决方法：需要开启GKI使能下面三个宏
kernel-5.10/arch/arm64/configs/gki_defconfig
CONFIG_HID_SONY=y
CONFIG_LEDS_CLASS=y
CONFIG_NEW_LEDS=y
```

- 测试项CtsMediaDecoderTestCases报错：根据报错信息修改frame rate帧率为实际测试帧率 Measured frame rate

```
CtsMediaDecoderTestCases
android.media.decoder.cts.VideoDecoderPerfTest#testPerf[0(c2.rk.avc.decoder:qvga)]

报错信息：java.lang.AssertionError: Expected achievable frame rates for c2.rk.avc.decoder video/avc 320x240: [600.0, 650.0].
          Measured frame rate: [134.16529367461462, 129.51021865979183].
解决方法：根据报错信息"c2.rk.avc.decoder"和"320x240"定位测试项，修改帧率"125-140"要包含测试帧率[134.16, 129.51]
```

```
Android_SDK/vendor/rockchip/common/vpu/etc$ git diff media_codecs_performance_rk3588.xml
            <MediaCodec name="c2.rk.avc.decoder" type="video/avc" update="true">
-                <Limit name="measured-frame-rate-320x240" range="600-650" />
+                <Limit name="measured-frame-rate-320x240" range="125-140" />
注:修改文件生成目录/vendor/etc/media_codecs_performance.xml
```

- 测试项SingleLayer_DepthTest_DepthAffectsDrawAcrossContexts_D32_FLOAT报错：/GMS-Test-Suite/Exp+/AHardwareBufferNativeTests

```
CtsNativeHardwareTestCases
android.hardware.nativehardware.cts.AHardwareBufferNativeTests#SingleLayer_DepthTest_DepthAffectsDrawAcrossContexts_D32_FLOAT
android.hardware.nativehardware.cts.AHardwareBufferNativeTests#SingleLayer_DepthTest_DepthCanBeSampled_D32_FLOAT
android.hardware.nativehardware.cts.AHardwareBufferNativeTests#MultipleLayers_DepthTest_DepthCanBeSampled_D32_FLOAT
```

- 测试项testSocketKeepaliveUnprivileged报错：补丁/GMS-Test-Suite/Exp+/DisableKeepAliveMainlineBuild

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

CtsHostsideNetworkTests
com.android.cts.net.HostsideVpnTests#testAutomaticOnOffKeepaliveModeClose
com.android.cts.net.HostsideVpnTests#testAutomaticOnOffKeepaliveModeNoClose
```

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

- 测试项testPruneUnusedStaticSharedLibraries_reboot_fullMode报错：需要保证硬件RTC功能正常

```
CtsOsHostTestCases
android.os.cts.StaticSharedLibsHostTests#testPruneUnusedStaticSharedLibraries_reboot_fullMode
android.os.cts.StaticSharedLibsHostTests#testPruneUnusedStaticSharedLibraries_reboot_instantMode

报错信息：java.lang.AssertionError: Library foo.bar.lib.recursive should be uninstalled
```

- 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SELinuxNeverallowRulesTest报错：MADA/EDLA认证不能修改/system/sepolicy目录下任何权限，**且不能用userdebug固件测试**

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

报错信息：junit.framework.AssertionFailedError: The following errors were encountered when
validating the SELinuxneverallow rule:
          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testGetInactiveApexFactoryPackagesAfterApexInstall_containsNoDuplicates报错：机器watchdogd进程异常挂死

```
CtsStagedInstallHostTestCases
com.android.tests.stagedinstall.host.StagedInstallTest#testGetInactiveApexFactoryPackagesAfterApex
Install_containsNoDuplicates
报错信息：APEX activation failed. Reason: Session reverted due to crashing native process:
watchdogd

GtsBootHealthHostTestCases
android.boothealth.gts.BootHealthHostTest#testUpdatableProcessCrashNotDetected
报错信息：Upadatable process crash detected: watchdogd
```

```
检查dts中是否有添加wdt
Android_SDK/kernel-6.1/arch/arm64/boot/dts/rockchip$ git diff rk3576-android.dtsi
 &vop {
          support-multi-area;
 };
+
+&wdt {
+          status= "okay";
+};
```

- 测试项testSystemMemoryAtom报错

```
CtsStatsdAtomHostTestCases
android.cts.statsdatom.memory.SystemMemoryStatsTests#testSystemMemoryAtom

报错信息：FAILURE: expected to be at least: 0        but was: -1
```

```
解决方法：若为kernel-4.19则关掉dmabuf（Android 13及以上版本不适用）
Android_SDK/kernel-4.19$ git diff arch/arm64/configs/rockchip_defconfig
-CONFIG_DMABUF_HEAPS=y
-CONFIG_DMABUF_HEAPS_SYSTEM=y
-CONFIG_DMABUF_HEAPS_CMA=y
```

- 测试项testRestart和testMultipleActivations报错：主板必须有硬件RTC(**不能是虚拟RTC**)且存在节点/sys/class/rtc/rtc0/

```
CtsStatsdHostTestCases
android.cts.statsd.metric.MetricActivationTests#testRestart
android.cts.statsd.metric.MetricActivationTests#testMultipleActivations

解决方法：根据实际节点状态添加SELinux权限(或用工具Android_Devices_SELinux_Tool扫描)，且需要保证"date -u"和"hwclock -r"查询时间一致
console:/ $ ls -laZ /sys/class/rtc/
total 0
drwxr-xr-x   2 root root u:object_r:sysfs_rtc:s0    0 2023-08-12 18:19 .
drwxr-xr-x 82 root root u:object_r:sysfs:s0          0 2023-08-12 18:19 ..
lrwxrwxrwx   1 root root u:object_r:sysfs:s0          0 2023-08-12 18:53 rtc0 ->
../../devices/platform/feaa0000.i2c/i2c-2/2-0051/rtc/rtc0
```

```
Android_SDK/device/rockchip/rk3588$ git diff sepolicy_vendor/genfs_contexts
 genfscon sysfs /devices/platform/fe2e0000.mmc/mmc_host/mmc0
u:object_r:sysfs_mmc:s0
+genfscon sysfs /devices/platform/feaa0000.i2c/i2c-2/2-0051/rtc/rtc0     u:object_r:sysfs_rtc:s0
```

- 测试项testRefreshRate报错：屏幕帧率不够导致fail

```
CtsViewTestCases
android.view.cts.DisplayRefreshRateTest#testRefreshRate

报错信息：D DisplayRefreshRateTest: claimed 75.0 fps, achieved 65.0 fps
          D DisplayRefreshRateTest: claimed 75.0 fps, achieved 60.63158 fps
          D DisplayRefreshRateTest: claimed 75.0 fps, achieved 61.125 fps
          E TestRunner: failed: testRefreshRate(android.view.cts.DisplayRefreshRateTest)
```

```
帧率计算公式：(1200+40+40+4)*(1920+150+40+4)*60=162862560，取整配置clock-frequency = <162000000>

clock-frequency = (hactive + hfront-porch + hback-porch + hsync-len) * (vactive + vfront-porch +
vback-porch + vsync-len) * fps

disp_timings0: display-timings {
    native-mode = <&dsi_timing0>;
    dsi_timing0: timing0 {
        clock-frequency = <162000000>;
        hactive = <1200>;
        vactive = <1920>;
        hback-porch = <40>;
        hfront-porch = <40>;
        vback-porch = <40>;
        vfront-porch = <150>;
```

```
        hsync-len = <4>;
        vsync-len = <4>;
        hsync-active = <0>;
        vsync-active = <0>;
        de-active = <0>;
        pixelclk-active = <0>;
    };
};
```

- 测试项AnrTests报错：不能去掉系统ANR弹框和修改ANR时间

```
CtsWindowManagerDeviceTestCases
android.server.wm.AnrTests#slowUiThreadWithKeyEventTriggersAnr
android.server.wm.AnrTests#slowOnCreateWithKeyEventTriggersAnr
android.server.wm.AnrTests#slowOnTouchEventHandleTriggersAnr
android.server.wm.AnrTests#slowOnKeyEventHandleTriggersAnr
android.server.wm.AnrTests#embeddedWindowTriggersAnr

报错信息：FAILURE: java.lang.AssertionError: Could not find anr dialog
```

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

```
CtsWindowManagerDeviceTestCases
android.server.wm.WindowInsetsControllerTests#testSystemUiVisibilityCallbackCausedByInsets
```

- 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#testActivityTransitionDurationNoShortenAsExpected
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：Android_SDK/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testMaxAspectRatioPreOActivity报错：屏幕宽高比不满足要求，切换横竖屏测试或调小导航栏高度navigation_bar_height

```
CtsWindowManagerSdk25TestCases
android.server.wm.AspectRatioSdk25Tests#testMaxAspectRatioPreOActivity

报错信息：Expected: a value less than or equal to <1.87F>
                  but: <1.9047619F> was greater than <1.87F>
解决方法：1.9047619 = 1920/(1080-72)      1.87 = 1920/(1080-53)
console:/ $ wm size
Physical size: 1920x1080
console:/ $ dumpsys SurfaceFlinger | grep NavigationBar
0xb400007eee0bd600: unknown        | 1080 (    0) x    72 |      1 |            1 |
0x40000000000b00 | NavigationBar0#0
```

```
Android_SDK/frameworks/base$ git diff core/res/res/values/dimens.xml
       <!-- Height of the bottom navigation / system bar. -->
-      <dimen name="navigation_bar_height">72dp</dimen>
+      <dimen name="navigation_bar_height">53dp</dimen>
```

17.10 CTS-V常见fail项处理

- 测试项Companion Device Service Test报错：需要将辅助机(BLE蓝牙设备、如手表)设置成可发现状态并放在测试机旁边

```
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"Advertisement duration"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"
```

- 测试项Device Suspend Tests报错：机器要进入深度休眠2秒且能够唤醒(测试时不接adb线)

```
com.android.cts.verifier.sensors.DeviceSuspendTestActivity#testTimestampClockSource

报错信息：SensorTest: [TestExecution] Delta between clock sources too small (0mS), device must
sleep more than 2 seconds
```

- 测试项Notification Listener Testb报错：开机向导设置PIN密码 或 设置PIN密码后`System->Reset options-> Reset app preferences`

```
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
```

- 测试项USB Accessory Test报错：建议使用Google手机作为辅助机

```
测试项USB Accessory Test
1.辅助机点击"Start accessory test companion"，测试机点击"USB Accessory Test"
2.辅助机作为HOST(接OTG口)，测试机作为DEVICE(接USB口)
3.辅助机弹框选择"OK"

测试项USB Device Test
1.辅助机点击"Start device test companion"，测试机点击"USB Device Test"
2.辅助机作为DEVICE(接USB口)，测试机作为HOST(接OTG口)
3.测试机弹框选择"OK"
```

17.11 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
```

- 测试项soundDoseReached_showsWarning报错：缺少android.permission.MODIFY_AUDIO_SETTINGS_PRIVILEGED权限

```
GtsAudioTestCases
com.google.android.audio.gts.SoundDoseTest#soundDoseReached_showsWarning
com.google.android.audio.gts.SoundDoseTest#momentaryExposure_showsWarning

报错信息：com.android.bedstead.nene.exceptions.NeneException: Can not get required permission

Running On User: User{id=0, name=Owner}
Permission: android.permission.MODIFY_AUDIO_SETTINGS_PRIVILEGED
Permission protection level: Permission not found
Permission state: -1
Instrumented Package: com.google.android.audio.gts

Requested Permissions:
[android.permission.QUERY_ALL_PACKAGES,
com.google.android.audio.gts.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION,
android.permission.REORDER_TASKS, android.permission.READ_EXTERNAL_STORAGE]

Can adopt shell permissions: true
Shell permissions:[android.permission.REQUEST_DELETE_PACKAGES,
android.permission.MANAGE_ROLLBACKS, ......
```

```
解决方法：参考https://android-review.googlesource.com/c/platform/frameworks/base/+/1261388
Android_SDK/frameworks/base$ git diff data/etc/privapp-permissions-platform.xml
       <privapp-permissions package="com.android.shell">
+           <permission name="android.permission.MODIFY_AUDIO_SETTINGS_PRIVILEGED"/>
            <!-- Needed for test only -->
            <permission name="android.permission.MODIFY_DAY_NIGHT_MODE"/>
Android_SDK/frameworks/base$ git diff packages/Shell/AndroidManifest.xml
       <!-- Permission required for testing system audio effect APIs. -->
       <uses-permission android:name="android.permission.MODIFY_DEFAULT_AUDIO_EFFECTS"/>
+    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS_PRIVILEGED"/>
```

- 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
```

```
解决方法：Android_SDK/vendor/partner_gms$ git diff products/gms.mk
-# GMS comms suite
-$(call inherit-product, $(ANDROID_PARTNER_GMS_HOME)/products/google_comms_suite.mk)
```

- 测试项testDisablingDeviceProvisioningPackage报错：Android12以上版本不能集成应用 OobConfig.apk

```
GtsGmscoreHostTestCases
com.google.android.gts.provisioning.ProvisioningHostTest#testDisablingDeviceProvisioningPackage

报错信息：android.content.pm.PackageManager$NameNotFoundException:
com.google.android.apps.work.oobconfig
```

- 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

报错信息：junit.framework.AssertionFailedError: The verified boot key must be non-trivial, at
least 32 bytes long.
报错信息：junit.framework.AssertionFailedError: The device's bootloader must be locked.
```

- 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testPolicy报错：根据报错信息修改应用包名

```
GtsMbaPrivilegedPermissionTestCases
com.google.android.mbaprivilegedpermission.gts.GtsPackageNamespaceTest#testPolicy

报错信息：java.lang.RuntimeException: Violation of the Package Namespace
          [Package Name: com.google.android.apps.customization.pixel, Certificates:
[3cc6c5f27c0b54f96507c2a5b31c404b3dbf065a2edb093eacd41147f2480f30]] contains Google, but not owned
by Google.
解决方法：修改应用包名"com.google.android.apps.customization.pixe" -->
"com.rockchip.android.apps.customization.pixel"
```

- 测试项testNetworkStack报错：

```
GtsNetworkStackHostTestCases
com.google.android.gts.networkstack.NetworkStackHostTest#testNetworkStack

报错信息：
com.google.android.networkstack.gts.NetworkStackTest#testCaptivePortalAppInstalledAndHasPostNotifi
cationsPermission:
java.lang.AssertionError: com.google.android.go.captiveportallogin does not have
android.permission.POST_NOTIFICATIONS permission. Check that its package name was configured in
config_defaultCaptivePortalLoginPackageName (typically via overlay). expected:<0> but was:<-1>
```

```
解决方法1：集成CaptivePortalLoginFrameworkOverlayGo.apk应用Overlay配置
项"config_defaultCaptivePortalLoginPackageName"
Android_SDK/vendor/rockchip/common/gms/CaptivePortalLoginFrameworkOverlayGo$ cat
res/values/config.xml
<string name="config_defaultCaptivePortalLoginPackageName"
translatable="false">com.google.android.go.captiveportallogin</string>

解决方法2：默认授予应用com.google.android.go.captiveportallogin权限
Android_SDK/vendor/partner_gms$ git diff etc/default-permissions/default-permissions-google.xml
+
+  <exception
+     package="com.google.android.go.captiveportallogin"
+     sha256-cert-
digest="5F:A4:22:12:AD:40:3E:22:DD:6E:FE:75:F3:F3:11:84:05:1F:EF:74:4C:0B:05:BE:5C:73:ED:F6:0B:F6:
2C:1E">
+     <!-- Notifications -->
+     <permission name="android.permission.POST_NOTIFICATIONS" fixed="false"/>
+  </exception>
 </exceptions>
```

- 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)
```

```
解决方法：Android_SDK/vendor/partner_gms$ git diff apps/SearchLauncher/AndroidManifest.xml
-     <uses-sdk android:targetSdkVersion="29" android:minSdkVersion="28"/>
+     <uses-sdk android:targetSdkVersion="30" android:minSdkVersion="28"/>
```

- 测试项testSystemAppsInstalled报错：非4G设备不集成应用AndroidAutoStub.apk

```
GtsPlacementTestCases
com.google.android.placement.gts.InstalledAppsTest#testSystemAppsInstalled

报错信息：java.lang.RuntimeException: Test failed for cases: [0]
          Reasons include:
          Message [0]: com.google.android.projection.gearhead should not be installed on devices without Telephony
```

```
解决方法：Android_SDK/vendor/partner_gms$ git diff products/gms.mk
 # GMS mandatory core packages
 PRODUCT_PACKAGES := \
-    AndroidAutoStub \
```

- 测试项testWidgetPresence报错：不能删掉桌面Google Search widget组件

```
GtsSearchHostTestCases
com.google.android.search.gts.GoogleSearchWidgetTest#testWidgetPresence

报错信息：FAILURE: java.lang.AssertionError: Launcher from package com.android.launcher3 does not contain search widget provided by com.google.android.googlequicksearchbox (Google App).
解决方法：相关修改变
量/packages/apps/Launcher3/src_build_config/com/android/launcher3/BuildConfig.java -->
QSB_ON_FIRST_SCREEN
```

- 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_zeroTouchWrapperLaunched

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.
```

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

```
WvtsDeviceTestCases
com.google.android.wvts.WidevineBusinessLogicTests#testWidevineVersion

报错信息:FAILURE: java.lang.AssertionError: Unknown build number with cdm version 16.1.0
```

17.12 STS常见fail项处理

- 测试项testPocBug_258188673报错：FTP服务器补丁/GMS-Test-Suite/Exp+/testPocBug_258188673

```
StsHostTestCases
android.security.sts.Bug_258188673#testPocBug_258188673

报错信息：java.lang.AssertionError: PoC exited with bad exit code.. Actual: 113
```

17.13 VTS常见fail项处理

- 测试项ProcCmdlineAndroidbootTest报错：需要开启GKI(宏BOARD_BUILD_GKI)，不再使用androidboot.***

```
VtsBootconfigTest
VtsBootconfigTest#ProcCmdlineAndroidbootTest

报错信息：Expected equality of these values:
              cmdline.find("androidboot")
                 Which is: 18
              cmdline.npos
                 Which is: 18446744073709551615
           "androidboot" parameters are not allowed in the kernel cmdline for devices using
kernel version 5.10 or greater with Android S and beyond. These parameters are to be placed in
bootconfig.
```

- 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
GenericBootImageTest#GenericRamdisk

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
KernelVersionTest#AgainstPlatformRelease

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

- 测试项VtsHalSensorsV1_0TargetTest报错信息：校准传感器(参考测前准备)

```
VtsHalSensorsV1_0TargetTest
PerInstance/SensorsHidlTest#AccelerometerStreamingOperationSlow/0_default
PerInstance/SensorsHidlTest#AccelerometerStreamingOperationNormal/0_default

报错信息：Value of: checker.check(sensorEvents, &s)
          Actual: false
          Expected: true
          Event @ 49513760914252 (8.6287, -0.603339, 1.13006) has norm 8.72328, which is beyond
range [8.80665, 10.8067]
解决方法：echo 1 > /sys/class/sensor_class/accel_calibration
```

- 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_libsnapshot_test
SnapshotUpdateTest#FullUpdateFlow
SnapshotUpdateTest#MergeCannotRemoveCow

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"idl_package_root"同级目录current.txt，文件可能
是/vendor/rockchip/hardware/interfaces/current.txt
若编译报错，根据报错信息将缺少项按照上述命令添加current.txt
```

17.14 Checklist常见fail项处理

- M13 - Default Home Screen Display
- M14 - Placement of Google Collections Folder测试项：需要集成应用GmsSampleIntegration.apk

```
vendor/partner_gms/apps/GmsSampleIntegration/res_dhs_full/xml/partner_default_layout.xml
<folder title=......</folder>
```

注意：Duo.apk和Meet.apk包名都是package="com.google.android.apps.tachyon"但图标不同，系统只能集成一个应用否则会出现两个应用图标

```
Android_SDK/vendor/partner_gms$ git diff products/gms.mk
 PRODUCT_PACKAGES += \
       Chrome \
       Drive \
       Gmail2 \
-      Duo \
+      Meet \
```

注意：Google文件夹需要包含"Google TV"图标(旧名称"Play Movies")，即应用Videos.apk

```
package:/product/app/Videos/Videos.apk=com.google.android.videos

Android_SDK/vendor/partner_gms$ git diff
apps/GmsSampleIntegration/res_dhs_full/xml/partner_default_layout.xml
-     <favorite packageName="com.google.android.videos"
className="com.google.android.youtube.videos.EntryPoint"/>
+     <favorite packageName="com.google.android.videos"
className="com.google.android.videos.GoogleTvEntryPoint"/>
       <favorite packageName="com.google.android.apps.tachyon"
className="com.google.android.apps.tachyon.MainActivity"/>
```

- M30 - Restore Experience in the Setup Wizard测试项：/GMS-Test-Suite/Exp+/M30_A13_USB Data
- M33 - Location Privacy测试项：若机器没有GPS则跳过测试
- M43 - Enterprise Setup Experience - QR Code provisioning测试项：/GMS-Test-Suite/Exp+/M58_Code scanner

```
Android_SDK/frameworks/base$ git diff core/res/res/values/config.xml
       <!-- Default component for QR code scanner -->
-     <string name="config_defaultQrCodeComponent"></string>
+     <string
name="config_defaultQrCodeComponent">com.google.android.gms/.mlkit.barcode.ui.PlatformBarcodeScann
ingActivityProxy</string>

Android_SDK/frameworks/base$ git diff packages/SystemUI/res/values/config.xml
       <!-- The default tiles to display in QuickSettings -->
       <string name="quick_settings_tiles_default" translatable="false">
-           wifi,bt,dnd,flashlight,rotation,battery,cell,airplane,night,screenrecord
+
wifi,bt,dnd,flashlight,rotation,battery,cell,qr_code_scanner,airplane,night,screenrecord
       </string>
```

状态栏显示"Scan QR Code"或"QR code scanner"图标需要机器申请`feature:android.hardware.camera`，若机器只有前摄或Usb Camera则修改

```
Android_SDK/frameworks/base$ git diff
packages/SystemUI/src/.../qrcodescanner/controller/QRCodeScannerController.java
@@ -237,7 +237,7 @@ public class QRCodeScannerController implements
       public boolean isCameraAvailable() {
             if (mIsCameraAvailable == null) {
                   mIsCameraAvailable = mContext.getPackageManager().hasSystemFeature(
-                               PackageManager.FEATURE_CAMERA);
+                               PackageManager.FEATURE_CAMERA_FRONT);
             }
             return mIsCameraAvailable;
```

排查方法：下拉状态栏点击右下角"笔"图标，进去后点击"RESET"

```
console:/ $ logcat -v time | grep QSTileHost
D/QSTileHost(1256):Destroying not available tile: qr_code_scanner
```

- M53 - Safety and Emergency settings测试项：/GMS-Test-Suite/Exp+/M53_Emergency
- M56 - Monochrome app icons测试项：/GMS-Test-Suite/Exp+/M56_A13_0828.zip

```
Android_SDK/vendor/partner_gms$ git diff products/gms.mk
 PRODUCT_PACKAGES += \
       GmsSampleIntegration \
-      SearchLauncherQuickStep
+      SearchLauncherQuickStep \
+      WallpaperGoogle

Android_SDK/vendor/partner_gms$ git diff etc/permissions/privapp-permissions-google-system-ext.xml
+      <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"/>
+            <permission name="android.permission.MODIFY_DAY_NIGHT_MODE"/>
+            <permission name="android.permission.BIND_WALLPAPER"/>
+      </privapp-permissions>
```

- M60 - Long press on power to Assistant Education测试项：/GMS-Test-Suite/Exp+/M60_Assistant Education
- EEA-V2固件SearchSelector测试项：SearchSelector app has no icon and cannot be disabled(EEA固件配置说明)

```
Android_SDK/packages/apps/Settings$ git diff
src/com/android/settings/applications/ApplicationFeatureProviderImpl.java
              keepEnabledPackages.add(euicc.packageName);
        }
+      keepEnabledPackages.add("com.google.android.apps.setupwizard.searchselector");
        keepEnabledPackages.addAll(getEnabledPackageAllowlist());
```

17.15 userdebug固件测试fail项

下面相关fail项需要使用user固件测试，而不能使用userdebug固件

```
CtsDrmTestCases
android.drm.cts.DRMTest#testForwardLockAccess
报错信息：FAILURE: java.lang.RuntimeException: setDataSource failed: status = 0x80000000

CtsOsHostTestCases
android.os.cts.BuildHostTests#testIsSecureUserBuild
报错信息：FAILURE: org.junit.ComparisonFailure: Must be a user build expected:<user[debug]> but
was:<user[]>

CtsOsTestCases
android.os.cts.BuildVersionTest#testBuildFingerprint
报错信息：FAILURE: junit.framework.ComparisonFailure: Variant expected:<user[]> but was:
<user[debug]>

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>

CtsPermissionPolicyTestCases
android.permissionpolicy.cts.PrivappPermissionsTest#privappPermissionsMustBeEnforced
报错信息：java.lang.AssertionError: ro.control_privapp_permissions is not set to enforce expected:
<enforce> but was:<null>
```

18. 调试方法

某些测试项可能跟页面布局或机器dpi值有关，测试时可先调节机器dpi值或切换横竖屏测试

```
wm density ***                                                    ## 调节机器dpi

settings get system user_rotation
settings put system user_rotation 1                               ## user_rotation值为0/1/2/3

GtsPermissionControllerHostTestCases
android.permissioncontroller.gts.PermissionControllerHostTest#testDenyAppPermissionFragment
报错信息：FAILURE: java.lang.AssertionError: Expected four atoms reported
```

CTS测试过程会设置机器为"Personal"和"Work"两种工作模式，去掉方法：Personal->Settings->Passwords & accounts->Work->Remove work profile

19. 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]

豁免条件说明：能申请豁免的前提是**CDD没有"MUST"强制要求**但测试套件测了导致的fail测试项，与CDD不符的可以申请豁免；但**CDD有"MUST"明确要求的无法申请豁免**，除非贵司出货量非常大可以从商务端找Google谈判，否则没戏。

20. 附录A

- 已知通过MADA/EDLA认证的型号

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

21. 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/1gxxMaITC99Y2SvyRDUfiDFvY_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	

Android版本	PAB包下载地址
aosp-android11-tests-release	Android11 PAB版CTS/VTS测试套件下载地址
aosp-android12-tests-release	Android12 PAB版CTS/VTS测试套件下载地址
aosp-android13-tests-release	Android13 PAB版CTS/VTS测试套件下载地址
aosp-android14-tests-release	Android14 PAB版CTS/VTS测试套件下载地址
aosp-android15-tests-release	Android15 PAB版CTS/VTS测试套件下载地址