



Unisoc Confidential For hiar

Android VTS 测试指导手册

文档版本 V1.5
发布日期 2021-03-04

版权所有 © 紫光展锐（上海）科技有限公司。保留一切权利。

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

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

Unisoc Confidential For hiar

紫光展锐（上海）科技有限公司



前言

概述

本文档详细地描述了 Android VTS 测试的环境搭建、测试方法及注意事项，便于测试工程师更好的开展 VTS 测试。

读者对象




本文档主要适用于进行 Android VTS 的测试工程师。

缩略语

缩略语	英文全名	中文解释
VTS	Vendor Test Suite	供应商测试套件
CTS	Compatibility Test Suite	兼容性测试套件
GSI	Generic System Image	谷歌提供 AOSP 的 system.img
AOSP	Android Open-Source Project	Android 开放源代码项目
HAL	Hardware Abstraction Layer	硬件抽象层，是位于操作系统 内核与硬件电路之间的接口层
Lib	library	文件库

符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。
 注意	用于突出容易出错的操作。 “注意”不是安全警示信息，不涉及人身、设备及环境伤害。
 警告	用于可能无法恢复的失误操作。 “警告”不是危险警示信息，不涉及人身及环境伤害。

变更信息

文档版本	发布日期	修改说明
V1.0	2019-09-02	第一次正式发布。
V1.1	2019-12-16	增加 Android10.0 相关内容，更新到新模板格式
V1.2	2020-03-02	内容无变更，适用产品增加 SharkL5 Pro(UMS512(T))、SharkL5(UMS312)
V1.3	2020-08-10	技术更新： -增加 Android 11.0 相关 log 设置的内容 文字更新： -调整文档格式，优化文档描述
V1.4	2020-10-25	完善 VTS 环境配置步骤，调整文档格式，优化文档描述。
V1.5	2021-03-04	<ul style="list-style-type: none">• 增加硬件 NFC 模组。• 优化文档内容，更新到新模板样式。

关键字

VTS。

Unisoc Confidential For hiar

目 录

1 概述.....	1
1.1 VTS 简介.....	1
1.2 测试目标	1
2 测试环境搭建.....	2
2.1 PC 端测试环境	2
2.2 手机端测试环境	4
2.3 网络环境要求	5
3 VTS 测试及测试方法	6
3.1 测试场景及用例说明	6
3.2 测试方法指导	6
3.2.1 VTS 测试步骤	6
3.2.2 VTS 常用命令	6
3.3 测试结果判定	8
4 VTS 测试注意事项	9
4.1 VTS 测试须知.....	9
4.2 GSI 工程说明.....	9
4.3 GSI user 工程 log 设置	9
4.3.1 Android 8.0 上 VTS 测试的 Ylog 命令.....	9
4.3.2 Android 9.0 上 VTS 测试的 Ylog 命令.....	10
4.3.3 Android 10.0 上 VTS 测试的 Ylog 命令.....	10
4.3.4 Android 11.0 上 VTS 测试的 Ylog 命令.....	11
4.4 VTS 测试特殊情况.....	12
4.5 VTS 测试 Android 原生问题	13
4.6 Google 对豁免用例的处理策略.....	13
5 参考文档.....	14

图目录

图 1-1 treble_blog_before.....	1
图 1-2 treble_blog_after.....	1
图 2-1 python 环境变量配置	3
图 3-1 VTS 命令图示.....	7
图 3-2 VTS 执行结束.....	8
图 3-3 VTS 测试报告.....	8

Unisoc Confidential For hiar

表目录

表 3-1 VTS 常用命令	6
表 4-1 Android 8.0 上 VTS 测试的 Ylog 命令	9
表 4-2 Android 9.0 上 VTS 测试的 Ylog 命令	10
表 4-3 Android 10.0 上 VTS 测试的 Ylog 命令	10
表 4-4 Android 11.0 上 VTS 测试的 Ylog 命令	11
表 4-5 VTS 测试特殊情况	12
表 4-6 VTS 测试 Android 原生问题	13

Unisoc Confidential For hiar

1 概述

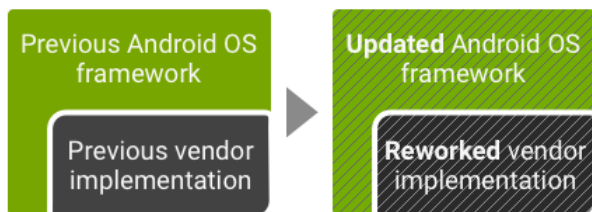
1.1 VTS 简介

为了能更快的将设备升级到新的 Android 版本，Android 8.0 开始新引入了 Project Treble。Project Treble 适用于搭载 Android 8.0 及后续版本的所有新设备。

Android 8.0 之前的版本中没有正式的 Vendor 层接口如图 1-1 所示，因此每次更新系统都相对耗时和困难。

图1-1 treble_blog_before

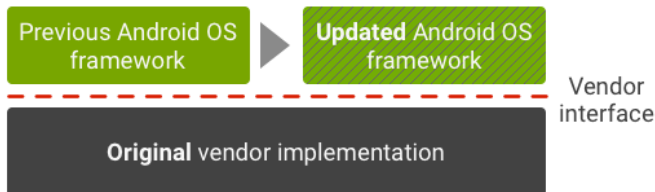
ANDROID UPDATES BEFORE TREBLE



Android 8.0 之后，Treble 提供了稳定的 Vendor 层接口如图 1-2 所示，供设备制造商访问 Android 代码中特定于硬件的部分。这样就可以只更新框架层，减少升级系统带来的成本和困难。

图1-2 treble_blog_after

ANDROID UPDATES WITH TREBLE



1.2 测试目标

Google 主要通过 CTS 测试来检测 Android 系统的兼容性，但 CTS 主要测试 FW 接口，并不能全面覆盖所有的兼容性要求，因此 Google 又推出了 VTS (Vendor Test Suite)测试。VTS 主要测试 Vendor 接口，具体包括 Kernel、Lib、HAL 等。从 Android 8.0 开始，Google 要求所有 Android 设备不仅需要通过 CTS，还必须通过 VTS，而且 VTS 测试必须在 CTS 之前完成。

2

测试环境搭建

VTS 测试必备的测试环境包括 PC 环境，手机配置，网络环境等。

2.1 PC 端测试环境

执行 VTS 测试的电脑，需要满足如下要求：

1. PC 安装 Ubuntu 系统，版本为 14.04 以上。
2. 安装 jdk1.8 版本及以上，Android 11.0 平台请更新到 jdk1.9 及以上。

升级 jdk 的命令如下：

```
sudo add-apt-repository ppa:openjdk-r/ppa5/11
sudo apt-get updatesudo apt-get install openjdk-9-jdk
```

安装完成之后，需更改默认 java，配置/etc/profile 文件即可（如下面前 3 行，改变下注释即可）。

```
#export JAVA_HOME=/usr/java/jdk1.6.0_29
#export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export JAVA_HOME=/usr/lib/jvm/java-9-openjdk-amd64
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
export CLASSPATH=$CLASSPATH:.$JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

配置完成后记得 source /etc/profile。

3. 安装 python，Android 10.0 及其以前安装 python2.7 即可，Android 11.0 需要安装 python3.0。

如果遇到 python 模块 pip 等无法安装问题，可以使用 anaconda 替代 python。anaconda 具体安装方法请自行查询或找 IT 协助。anaconda 集成了常用 python 环境，且更新方便，更新 python 库的命令如下。

```
conda update python
```

4. 下载 adb、fastboot 工具。请下载最新版本，platform-tools 包含 adb、fastboot，官网路径：

<https://developer.android.com/studio/releases/platform-tools.html>

5. 配置环境变量，包含 adb、fastboot、python、jdk 的环境变量。

其中 python 环境变量配置较为特殊，请参照下述说明。

6. python 环境配置

VTS 测试中，每个 module 开始测试前，均会从 pypi 网站下载并安装 vtspython 库模块安装包。由于从网址“<https://pypi.python.org/simple/>”获取安装源，受网络影响比较大，若网络不稳定，VTS 测试会高概率出现安装失败导致 VTS 测试失败。因此，建议本地化配置安装源。

- a 打开 terminal 窗口，执行 sudo gedit /etc/profile 打开配置文件 profile。
- b 添加“export VTS_PYPI_PATH=/usr/local/bin/vtspython”，如图 2-1 所示。

图2-1 python 环境变量配置

```

File Edit View Search Tools Documents Help
Open Save Undo Cut Copy Paste Find
profile x
. $!
fi
done
unset i
fi

if [ "$PS1" ]; then
if [ "$BASH" ]; then
PS1='\u@h:\w\$ '
if [ -f /etc/bash.bashrc ]; then
. /etc/bash.bashrc
fi
else
if [ "`id -u`" -eq 0 ]; then
PS1='# '
else
PS1='$ '
fi
fi
fi

umask 022

#export JAVA_HOME=/usr/java/jdk1.6.0_29
#export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-and64
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-and64
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH
export CLASSPATH=$CLASSPATH:.$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export VTS_PYPI_PATH=/usr/local/bin/vtspython

export USE_CCACHE=1

export LANG='en_US.UTF-8'
export LC_ALL='en_US.UTF-8'

```

c 获取并解压缩 vtspython 工具包

请从官网“<https://pypi.python.org/simple/>”获取 vtspython 库模块安装源，当前已经获取到的 vtspython 工具包本地保存路径请参见官网。

d 拷贝 vtspython 工具包

拷贝 vtspython 工具包到 VTS_PYPI_PATH 环境变量路径（如/usr/local/bin/vtspython，目录可更改为其它目录，但该目录必须有读写权限）。

拷贝命令：

```
$ cp -a (vtspython 文件夹路径) /usr/local/bin/
```

e 配置 vtspython 包权限。

打开 terminal 窗口，执行赋予权限命令：

```
sudo chmod 777 -R /usr/local/bin/vtspython
```

f 使能配置，生效方式有如下两种，请二选一。

- 重启电脑。
- 在 Terminal 中执行：\$ source /etc/profile。

g 验证配置是否生效。

在 terminal 中执行\$ echo \$VTS_PYPI_PATH，若返回值与 VTS_PYPI_PATH 环境变量路径一致（如/usr/local/bin/vtspython），则配置已生效。

h vtspython 库模块安装如下文所示，包括 python-setuptools、dev、protobuf、virtualenv、pip 等模块。

拷贝 vtspython 安装包到本地并解压，赋文件夹目录权限（上述步骤已完成）。

i 配置 VTS 工具包。

拷贝 VTS 测试包至 PC 上，并配置 VTS 测试包权限，如 sudo chmod 777 -R */VTS 包文件夹。

vtspython 库模块安装

- 方法一：

步骤 1 安装 python-setuptools 模块。

- 解压缩并进入 python-setuptools 安装包目录，如 `cd */setuptools-40.6.2`。
- 执行 `sudo python setup.py install` 进行 python-setuptools 模块安装。

步骤 2 如上述安装步骤所示，逐次安装 dev、protobuf、virtualenv、pip 等模块，安装方式同[步骤 1 安装 python-setuptools 模块](#)。

----结束

- 方法二：

步骤 1 安装\$ `sudo apt-get install python-setuptools`。

步骤 2 安装\$ `sudo easy_install dev`。

步骤 3 安装\$ `sudo easy_install protobuf`。

步骤 4 安装\$ `sudo easy_install virtualenv`。

步骤 5 安装\$ `sudo easy_install pip`。

----结束

说明

由于 PC 的初始 Python 环境不一定相同，使用方法二安装时在不同的 PC 会出现不同的报错，故建议优先采用方法一的安装方式。

2.2 手机端测试环境

执行 VTS 测试的手机，需要满足如下条件，进行相关设置：

- 手机校准 Pass（建议通过 GNSS OTA 校准）。
- 硬件安装指纹模组。
- 硬件安装 NFC 模组。
- 硬件安装 camera 模组。
- 手机需要 unlock。
 - unlock 必要性：

VTS 测试要求刷入 Google 提供 AOSP 的 system.img (GSI)。在 user 版本中，如果直接使用 flash tool 单独烧录 GSI，会导致无法开机。这是因为在 user/userdebug 版本中，dm-verity 是使能的，替换 GSI 后导致 dm-verity 不能通过。如果要解决这个问题，就需要进行 unlock 操作，并且要用 fastboot 来刷入谷歌提供的 system.img。
 - unlock 操作：
 - 方法一：使用展锐工具实现 unlock。
 - 方法二：使用官方方法。

官方 unlock 具体的操作如下：

- a 在设置中打开 OEM unlocking 选项。
 - b 在设置中打开 USB debugging 选项。
 - c 长按音量 + 和电源键进入 fastboot 模式。
 - d 连接到电脑上，分别执行 fastboot flashing unlock 和 fastboot oem unlock。执行命令后需要选择音量 + 来确认 unlock。
 - e 执行 fastboot 命令刷入 Google 提供的 system.img (GSI)：fastboot flash system system.img。需要根据软件版本的信息来选择 GSI 版本。Android 9.0 版本需要执行 fastboot flash vbmeta vbmeta.img。
 - f 重启：fastboot reboot。
- 休眠 30 分钟，保持唤醒，无锁屏（非强制要求）。
 - 连接翻墙 Wi-Fi。
 - SIM 卡：
 - Android 9.0 以上项目插双 SIM 卡。
 - 项目支持 SecureElement HAL 需求时，部分 case 需要使用支持 SE（Secure Element，安全元件）的 SIM 卡复测。
 - 建议插 SD 卡。
 - 手机刷 GSI 的版本。展锐 VTS 验收测试均直接下载版本正式发布的带 GSI 的 user 工程的 pac 包。（不建议自行烧录）如果自行先下载 user 版本后再烧录 GSI，需要注意如下内容：
 - GSI 刷机所需文件：
 - 32 位只需刷与 pac 包安全 patch 对应的谷歌 gsi system.img。
 - 64 位需要同时刷版本编译出来的 vbmeta.img、与 pac 包安全 patch 对应的谷歌 GSI system.img。
 - GSI system.img 选择：
 - adb shell "getprop |grep -I patch"查询使用哪个日期的版本。
 - adb shell "getprop |grep -i update"显示[ro.build.ab_update]: [true]即支持 a/b 升级。若支持 a/b 升级使用 arm_ab，否则使用 arm_a。
 - 展锐 Android 8.0 都选用 arm 且不带_a 或_ab 的版本。

2.3 网络环境要求

- 测试环境有稳定翻墙 Wi-Fi。
- 对于 Android 9.0 及以后版本，测试环境 GPS 信号稳定可用。测试过程中手机保持竖立状态，减少信号衰减，手机的 GPS 天线一般在顶部。需要注意室内还是室外测试：
 - 使用移动笔记本配置的电脑环境，在室外空旷环境测试，可以满足该要求。
 - 如果室内测试，需要依赖 GPS 信号放大器。

3 VTS 测试及测试方法

3.1 测试场景及用例说明

VTS 用例为 Google 官方发布的测试包，需要根据 VTS 官方发布对应更新。VTS 测试包目前区分 Android 版本。

3.2 测试方法指导

3.2.1 VTS 测试步骤

步骤 1 将 VTS 测试包复制到需要执行 VTS 测试的电脑上。

步骤 2 执行以下命令修改文件夹权限为 777。

```
sudo chmod 777 -R *installation-path*/android-vts/
```

步骤 3 打开终端命令行。

步骤 4 切换到 android-vts 路径下，命令如下。

```
$ cd *installation-path*/android-vts/tools
```

步骤 5 通过下面的命令来进入到 vts 的控制台。

```
$.vts-tradefed
```

步骤 6 运行 vts 的测试命令。

```
> run vts /*VTS 整包测试命令*/
```

---结束

3.2.2 VTS 常用命令

VTS 常用命令如表 3-1 所示。

表3-1 VTS 常用命令

VTS 常用命令	含义	备注
> lr	列出所有跑测结果	无
> ld	列出所有检测到或已知的设备	无
> run vts -m <模块名>	单跑某一个模块	无

VTS 常用命令	含义	备注
> run vts -m <模块名> -t <测试项名>	单跑某个测试项	无
> run vts -s <device_id> --logcat-on-failure --screenshot-on-failure --shard-count <shards> --exclude-filter <模块名>	可用选项	<ul style="list-style-type: none"> • -s <device_id>: 可指定运行测试的设备 • --logcat-on-failure: 当测试失败时捕获 logcat • --screenshot-on-failure: 当测试失败时截取屏幕截图 • --shard-count <shards>: 用多个设备上同时跑测, 可节省时间 • --exclude-filter: 过滤特定的已知问题的模块, 可以节省时间。比如 Android 10.0 新增无法测到的模块 VtsHalNeuralnetworksV1_2Target 会导致测试延长
run vts --retry <seddion id> 或 run retry -- retry <seddion id> (需要根据 Android 版本确定)	retry 选项	<p>Android 9.0 及其之前的 Android 版本: run vts --retry <seddion id></p> <p>Android 10.0 及其之后的 Android 版本: run retry -- retry <seddion id></p>

更多的 VTS 命令, 如“启动 VTS”等可通过 help all 来查看, 如图 3-1 所示。

图3-1 VTS 命令图示

```

~/CTS/VTS/8.1/android-vts/tools$ sudo ./vts-tradefed
Android Vendor Test Suite 8.1_rc2 (4499397)
Use "help" or "help all" to get more information on running commands.
vts-tf >
vts-tf >
vts-tf > help all
l(:?|st)? help:
  l[ocations]          List all invocation threads
  d[evices]           List all detected or known devices
  c[ommands]          List all commands currently waiting to be executed
  c[ommands] [pattern] List all commands matching the pattern and currently waiting to be executed
  c[onfigs]           List all known configurations

  ----- Vendor Test Suite specific options -----
  p[ans]             List all plans available
  m[odules]          List all modules available
  r[esults]           List all results

d(:?|unp)? help:
  s[ack]             Dump the stack traces of all threads
  l[ogs]             Dump the logs of all invocations to files
  b[ugreport]        Dump a bugreport for the running Tradefed instance
  c[onfig] <config>  Dump the content of the specified config
  commandQueue       Dump the contents of the command execution queue
  commands           Dump all the config XML for the commands waiting to be executed
  commands [pattern] Dump all the config XML for the commands matching the pattern and waiting to be executed
  e[nv]              Dump the environment variables available to test harness process
  u[ptime]           Dump how long the Tradefed process has been running

r(:?|un)? help:
  command <config> [options] Run the specified command
  <config> [options]         Shortcut for the above: run specified command
  cmdfile <cmdfile.txt>     Run the specified commandfile
  commandAndExit <config> [options] Run the specified command, and run 'exit -c' immediately afterward
  cmdfileAndExit <cmdfile.txt> Run the specified commandfile, and run 'exit -c' immediately afterward

  ----- Vendor Test Suite specific options -----
  <plan> --module/-m <module> Run a test module
  <plan> --module/-m <module> --test/-t <test_name> Run a specific test from the module. Test name can be <package>

```

3.3 测试结果判定

当出现如图 3-2 所示界面时，VTS 测试结束。测试完成会在测试包路径下生成 logs 和 results 文件夹。

图3-2 VTS 执行结束

```
result_failures.html
08-15 12:22:21 I/ResultReporter: Test Logs: /home/local/SPREADTRUM/intern/Downloads/VTS/android-vts/logs/2017.08.15_09.44.38
08-15 12:22:21 D/ResultReporter: Full Result: /home/local/SPREADTRUM/intern/Downloads/VTS/android-vts/results/2017.08.15_09.44.38.zip
08-15 12:22:21 I/ResultReporter: Invocation finished in 2h 37m 43s. PASSED: 2720, FAILED: 263, MODULES: 49 of 49
vts-tf >
```

results 和 logs 文件夹下分别生成以日期命名的文件夹，如：2019.04.02_18.06.06。打开 results-日期文件夹-test_result.xml 文件查看测试结果，如图 3-3 所示。

图3-3 VTS 测试报告

android
vendor test suite

Summary	
Suite / Plan	VTS / vts
Suite / Build	9.0_R9 / 5512091
Host Info	Result/@start sh02197tmp474 (Linux - 4.2.0-27-generic)
Start time / End Time	Wed Jun 19 16:59:17 CST 2019 / Wed Jun 19 17:02:54 CST 2019
Tests Passed	7685
Tests Failed	0
Modules Done	214
Modules Total	214
Fingerprint	SPRD/ [REDACTED]_1h10_oversea [REDACTED]_1h10:9/PPR1.180610.011/25225:user/release-keys
Security Patch	2019-06-05
Release (SDK)	9 (28)
ABIs	arm64-v8a,armeabi-v7a,armeabi

Module	Passed	Failed	Total Tests	Done
arm64-v8a VtsFirmwareBootHeaderVerification	2	0	2	true
arm64-v8a VtsFirmwareDtboVerification	2	0	2	true
arm64-v8a VtsHalAudioEffectV2_0Target	0	0	0	true
arm64-v8a VtsHalAudioEffectV4_0Target	39	0	39	true
arm64-v8a VtsHalAudioV2_0Target	0	0	0	true
arm64-v8a VtsHalAudioV4_0Target	1051	0	1051	true

4 VTS 测试注意事项

4.1 VTS 测试须知

- VTS 测试前，检查手机硬件和测试环境是否满足要求，如“[4.4 VTS 测试特殊情况](#)”所示。
- 第一轮测试失败的项，建议复测以排除网络或 GPS 等因素导致的概率性失败，提高认证效率。
- 翻墙 Wi-Fi 不稳定会导致失败，建议控制 VTS 测试环境 Wi-Fi 的接入设备数量，以保证网络稳定。
- GPS 信号：建议在室外空旷环境测试，或者使用信号转发器将信号转发到室内。参照“[4.4 VTS 测试特殊情况](#)”中的“测试环境 GPS 信号稳定可用”说明。

4.2 GSI 工程说明

- Android 9.0 以后 VTS 会主动触发 panic 导致测试停止，Debug 工程有 sysdump，因此 VTS 整包测试不建议用 debug 工程。
- VTS 测试过程中或测试完成，手机定屏都属于正常现象（已在 Google 对比机验证过）。

4.3 GSI user 工程 log 设置

一般情况下，仅提供 VTS 自身 log 即可定位问题。对于需要借助展锐 Ylog 定位的问题，此时请开启对应类型 log。一般 VTS 测试很少有 modem 问题，可以不用开启 modem log。刷好 GSI user 工程开机后，连接 Ubuntu PC，依次输入对应 Android 平台的命令设置 Ylog。

4.3.1 Android 8.0 上 VTS 测试的 Ylog 命令

Android 8.0 平台时，开始有 VTS 测试，其中 Android 8.1 开始大规模开展 VTS 测试，log 设置步骤如[表 4-1](#)所示。

表4-1 Android 8.0 上 VTS 测试的 Ylog 命令

步骤	Log 类型	命令	备注
1	开 root 权限	adb root	无
2	开 AP Log	adb shell setprop persist.ylog_lite.enabled 1	无
3	开 WCN Log	adb shell cplogctl enable wcn	无
4	开 GNSS Log	adb shell cplogctl enable gnss	无
5	重启手机	adb reboot	手机重启后生效

4.3.2 Android 9.0 上 VTS 测试的 Ylog 命令

Android 9.0 平台 VTS 测试，log 设置步骤如表 4-2 所示。

表4-2 Android 9.0 上 VTS 测试的 Ylog 命令

步骤	Log 类型	命令	备注
1	开 root 权限	adb root	无
2	开 AP Log	adb shell setprop persist.vendor.ylog_lite.enabled 1	无
3	开 Modem Log	adb shell cplogctl enable 5mode wcn	开 Modem Log 的第一句命令
4	开 Modem log 源控制	adb shell "echo "AT+ARMLOG=1\r" > /dev/stty_lte12" adb shell "echo "AT+SPDSPOP=2\r" > /dev/stty_lte12"	开 Modem Log 命令执行后，依次执行命令
5	关闭 modem reset	adb shell setprop persist.vendor.sys.sprd.modemreset 0	modem 异常后不自动重启，便于抓取 modem dump 数据
6	Modem Log 存 T 卡	adb shell setprop persist.vendor.modem.log_dest 2	modem log 可能较大，建议设置存储到 T 卡
7	开 WCN Log	adb shell cplogctl enable wcn	无
8	开 GNSS Log	adb shell cplogctl enable gnss	无
9	重启手机	adb reboot	手机重启后生效
10	Ylog 存储路径	SD 卡	无

4.3.3 Android 10.0 上 VTS 测试的 Ylog 命令

Android 10.0 平台 VTS 测试，log 设置步骤如表 4-3 所示。

表4-3 Android 10.0 上 VTS 测试的 Ylog 命令

步骤	Log 类型	命令	备注
1	开 root 权限	adb root	无
2	开 AP Log	adb shell start yloglite adb shell yloglitectl enable	依次执行命令。
3	开 Modem Log	adb shell cplogctl enable 5mode wcn	开 Modem Log 的第一句命令。

步骤	Log 类型	命令	备注
4	开 Modem log 源控制	adb shell "echo "AT+ARMLOG=1\r" > /dev/stty_lte27" adb shell "echo "AT+SPDSPOP=2\r" > /dev/stty_lte27"	开 Modem Log 命令执行后，依次执行命令。
5	关闭 modem reset	adb shell setprop persist.vendor.sys.sprd.modemreset 0	modem 异常后不自动重启，便于抓取 modem dump 数据。
6	Modem Log 存 T 卡	adb shell setprop persist.vendor.modem.log_dest 2	modem log 可能较大，建议设置存储到 T 卡。
7	开 WCN Log	adb shell cplogctl enable wcn	无。
8	开 GNSS Log	adb shell cplogctl enable gnss	无。
9	重启手机	adb reboot	手机重启后生效。
10	Ylog 存储路径	SD 卡或内卡	可存在 SD 卡中也可存手机内卡中。

4.3.4 Android 11.0 上 VTS 测试的 Ylog 命令

Android 11.0 平台 VTS 测试，log 设置步骤如表 4-4 所示。

表4-4 Android 11.0 上 VTS 测试的 Ylog 命令

步骤	Log 类型	命令	备注
1	开 root 权限	adb root	无
2	开 AP Log	adb shell start yloglite adb shell yloglitectl enable	依次命令执行
3	开 modem Log	adb shell cplogctl enable 5mode wcn	开 modem Log 的第一句命令
4	开 modem log 源控制	adb shell #"echo "AT+ARMLOG=1\r" > /dev/stty_lte27"	开 modem Log 命令执行后，依次执行命令
5	关闭 modem reset	adb shell setprop persist.vendor.sys.sprd.modemreset 0	modem 异常后不自动重启，便于抓取 modem dump 数据
6	开 WCN Log	adb shell cplogctl enable wcn	无
7	开 GNSS Log	adb shell cplogctl enable gnss	无
8	重启手机	adb reboot	手机重启后生效
9	Ylog 存储路径	/data/vendor/ylog	内部存储
10	Ylog 导出	adb pull /data/vendor/ylog/ 本地路径	Log 导出

4.4 VTS 测试特殊情况

通常，VTS 测试失败与特殊要求有关，如硬件、Wi-Fi、GPS 环境等。测试前注意检查测试环境，避免失败，提高 VTS 测试一次性通过率。已知的特殊情况总结如表 4-5 所示。

表4-5 VTS 测试特殊情况

特殊要求	建议与说明	VTS Module	Android 版本
手机 Unlock	VTS 基本要求，Android 8.0 开始就有该要求。未 unlock 下载 GSI 的 pac 无法开机。	All	All
安装指纹模组	VTS 基本要求，Android 8.0 开始就有该要求。手机安装指纹模组，否则指纹相关 case 会 Fail。	VtsHalBiometricsFingerprintV2_1Target	All
		VtsHalBiometricsFingerprintV2_1TargetReplay	All
硬件校准通过	硬件校准通过，OTA Pass 更好，可确保天线性能良好排除硬件问题，避免 GPS 模块 Fail	VtsHalGnssV1_1Target	Android 9.0 及之后
插双 SIM 卡	Android 9.0 新增要求。支持的 SIM 卡都可以（移动、联通或电信）。	VtsHalRadioV1_1Target	Android 9.0 及之后
		VtsHalRadioV1_0Target	NA
稳定翻墙 Wi-Fi 环境	VTS 基本要求，Android 8.0 开始就有该要求。测试环境翻墙 Wi-Fi 稳定，网络差时会概率性 Fail。	VtsHalDrmV1_0Target	All
		VtsHalDrmV1_1Target	All
稳定的 GPS 信号环境	Android 9.0 上 GNSS 模块新增 GnssV1_1。确保测试环境 GPS 信号稳定，以 GPS 信号正常搜星定位为准。空旷室外，或信号转发器（远距离需串联信号放大器）将室外 GPS 信号转发到室内。	VtsHalGnssV1_1Target	Android 9.0 及之后
支持 SE 白卡	准备支持 SE（SecureElement）的 SIM 卡，ZTE 新需求 SecureElement HAL，引入的 VTS 测试项，SE 测试项需要支持 SE 的 SIM 卡才能测试通过。	VtsHalSecureElementV1_0Target	Android 9.0 及之后

说明

表 4-5 中“Android 版本”列，特指该特殊情况会影响哪个 Android 版本的 VTS 测试。

4.5 VTS 测试 Android 原生问题

VTS 测试 Fail，除了代码问题导致的 case Fail 外，还有 Android 原生问题（如 VTS 用例编写问题等）导致的 case Fail，如表 4-6 所示。Android 原生问题需要 Google 来解决，需要提交 Google issue 来跟踪问题的解决。

表4-6 VTS 测试 Android 原生问题

特殊要求	建议与说明	VTS Module	Android 版本
Google 问题	Android 9.0 新增 Google 原生问题。VTS 包 9.0_R4 Fail, 9.0_R5 Pass	VtsTrebleVendorVintfTest	Android 9.0
Google 问题 (979625)	9.0 R5/ R6 VTS 包，VtsKernelNetTest 执行 testAnycastNetdeviceUnregister 时在如下 106 行因 thread.finished 值为 0 导致 Fail	VtsKernelNetTest	Android 9.0
Google 问题	VTS 9.0 R6 测试包存在 Google 内部服务器问题，bug 1013761 -TR	VtsHalDrmV1_0Target	Android 9.0
Google 问题 (1168928 1173795)	10.0 R1 VTS 包中 Google VTS 测试选择 BAND_ANY 是指同时支持 5G Band & 2.4G Band 不合理，修改兼容芯片不支持 5G Band \ BAND_ANY 的情况，10.0 R2 Pass	VtsHalWifiHostapdV1_1Target	Android 10.0
Google 问题 (1157870)	Android 10.0 新增 Google 原生问题。VTS 包 10.0_R1 Fail, 10.0 R2 Pass	VtsHalRadioV1_4Target	Android 10.0
Google 问题 (1157867)	Android 10.0 新增 Google 原生问题。VTS 包 10.0_R1 Fail, 10.0 R2 Pass	VtsHalRadioV1_2Target	Android 10.0
Google 问题 (1130957)	Android 10.0 新增 Google 原生问题。VTS 包 10.0_R1 Fail, 测不到	VtsHalNeuralnetworksV1_2Target	Android 10.0

4.6 Google 对豁免用例的处理策略

对于豁免用例，现在 Google 要求在 dev 包（Google 提供的包含修改的 VTS 测试包）测试通过后才能有针对性的豁免，否则不同意豁免。

也就是说，对于 VTS 测试中的 Fail case，是 Android 原生问题需要 Google 来解决的，提交 Google issue 后，在 VTS dev 包提供验证完毕之前，都需要重点跟踪，直到问题验证通过才算通过 Google 豁免。

如果有相关无法豁免的问题，需要提交 Bug 跟踪解决。

5

参考文档

1. VTS 测试 Google 官网网址:
 - a) <https://source.android.com/compatibility/vts>
 - b) <https://source.android.com/compatibility/cts/setup>
 - c) <https://source.android.com/devices/tech/vts/>
 - d) https://source.android.com/devices/tech/test_infra/tradefed/fundamentals/vts.html
2. ADB 相关工具 Google 官网网址:
<https://developer.android.com/studio/releases/platform-tools.html>

Unisoc Confidential For hiar