

Android 10

GTS 测试操作指南

1.0

2020.02.26

文档履历

版本号	日期	制/修订人	内容描述
1.0	2020.02.26		Android 10 GTS 测试操作 指南

目录

1. 测试环境搭建	1
1.1 获取 GTS 测试包	1
1.2 Ubuntu 主机测试环境搭建	1
1.3 网络环境	2
2. 测试前平板配置	3
3. 启动 GTS 测试	5
3.1 GTS 完整测试	5
3.1.1 常用测试命令	5
3.1.2 完整测试指令	7
3.2 GTS 补测	9
3.2.1 Retry 测试	9
3.2.2 创建补测测试计划以及启动补测	9
3.3 针对性测试	10
3.4 跳过某些有问题的测试项	10
3.5 循环测试	11
4. 测试结果分析与调试	12
4.1 从 GTS 报告中获取出错信息	12
4.2 查找相关 log 信息	12
5. Declaration	13

1. 测试环境搭建

1.1 获取 GTS 测试包

GTS 工具是不开源的，由 GMS 送测渠道提供。

1.2 Ubuntu 主机测试环境搭建

对测试主机的具体要求如下：

1. 安装 Ubuntu16.04 LTS 64bit 系统
2. 添加 adb 工具，配置系统使主机能够通过 adb 连接平板。
3. 安装 JDK1.8。
4. 安装 aapt 工具。
5. 安装 GTS 测试工具。将 1.1 节中获取的 zip 压缩文件解压到测试主机中，解压后得到名字为 android-gts 的文件夹，注意不要修改此文件夹及其子目录和文件的名字。
6. 主机需要连接外网。在 GTS 5.1r2 测试包之后测试主机也需要连接可以访问谷歌的网络，否则测试无法跑起来。
7. 配置主机端测试 key。将获取到的 key 文件保存在本地，该 key 文件的获取方式请联系相关的技术客户经理。增加环境变量 APE_API_KEY，其值指向该 key 的路径。

如 key 的名称为：gts-allwinner.json，存放路径如下：/home/pdc/Test/GTS/gts-allwinner.json

则可以在 vim ~/.bashrc，其中配置如下：

```
export APE_API_KEY=/home/pdc/Test/GTS/gts-allwinner.json
```

注意：需要设置环境变量，使系统能够找到 adb，java 版本和 aapt 工具。

1.3 网络环境

GTS 测试过程中会连接国外的网络如：youtube。使用国内的网络无法访问这些网站。需要使用能正常访问这些网站的网络环境进行测试。

测试过程会测试 ipv6 网络，所以 wifi 需要支持 ipv6。



2. 测试前平板配置

平板固件烧录完成后需要进行相关配置才能测试 GTS。进行 GTS 测试的设备都必须是安全机器，并已经烧写 google attestation key。

主要的配置如下：

1. 恢复出厂设置。这一项在有必要的情况下进行。刚烧好固件运行起来的可跳过此步骤。如果烧录好固件后被用作其它用途再进行 GTS 测试，则需要先恢复出厂设置或重烧固件。

2. 在设置中选择语言为 English(United States)。

3. Settings > Display > Brightness 设置为最小。

4. Settings > Display > Sleep 设置最长休眠时间。

5. 选项 Settings > Security > Screen Lock 为 none，确保设备上未设置锁定图案或密码。

6. 选项 Setting > Wi-fi，连接支持 ipv6 和能连接国外网络的网络。

7. 勾选 USB 调试。Settings > Developer options > USB debugging。(注意，在 4.2 之后的系统中 Developer options 默认是不显示的，需要进入 Settings > About tablet，然后迅速连续敲击 Build number 七次，返回上一级菜单查找开发者选项)。

8. 勾选 Settings > Developer options > Stay Awake，保持屏幕常亮。

9. 去掉勾选 Settings > Developer options > Verify apps over USB。

10. 打开浏览器，跳过浏览器设置界面。

11. 打开相机 app，跳过相机设置界面，使其在测试的时候能正常打开相机，如果弹出窗口 "Allow Camera to access this device's location"，请选择 DENY，。

12. GTS 需要多媒体文件，若没有配置本地多媒体的参数，则会自动下载。目前 Android 10 版本的 GTS 支持指定本地多媒体文件的参数，请参考 3.1.2 章节的介绍。

13. 连接能够上国外网络的 Wifi AP。

14. 将平板通过 USB 连接到测试主机。在 USB debugging 弹框中勾选 Always allow from this computer，点击 OK。

15. 在 Settings>System>Backup 中，打开 backup 设置。



3. 启动 GTS 测试

3.1 GTS 完整测试

启动 gts 测试，需要在终端进入 1.2 步骤所解压的 android-gts 文件夹下的 tools 目录下，执行命令 ./gts-tradefed，会启动测试平台。

3.1.1 常用测试命令

Host	Description
help	Display a summary of the most commonly used commands
help all	Display the complete list of available commands
version	Show the version.
exit	Gracefully exit the GTS console. Console will close when all currently running tests are finished.
Run	Description
run gts	Run the default GTS plan (that is, the full GTS invocation). The GTS console can accept other commands while tests are in progress. If no devices are connected, the GTS desktop machine (or host) will wait for a device to be connected before starting tests. If more than one device is connected, the GTS host will choose a device automatically.
--plan <test_plan_name>	Run the specified test plan.

Host	Description
-- module/-m [--module/-m <test_module2>...]	Run the specified test module or modules. For example, run gts --module GtsHomeHostTestCases executes the gesture test module (this can be shortened to run gts -m HomeHost). run gts -m HomeHost --test com.google.android.home.gts.ScreenshotTest#testHomeScreen runs the specific package, class, or test.
-- module/-m -- test	Run the specified module and test. For example, run gts -m HomeHost --test com.google.android.home.gts.ScreenshotTest#testHomeScreen runs the specific package, class, or test.
--retry	Retry all tests that failed or were not executed from the previous sessions. Use list results to get the session id.
--shards <number_of_shards>	Shard a GTS run into given number of independent chunks, to run on multiple devices in parallel.
--serial/-s <deviceID>	Run GTS on the specific device.
--include-filter [--include-filter <module2>...]	Run only with the specified modules.
--exclude-filter [--exclude-filter <module2>...]	Exclude the specified modules from the run.
--log-level-display/-l	Run with the minimum specified log level displayed to STDOUT. Valid values: [VERBOSE, DEBUG, INFO, WARN, ERROR, ASSERT].
--abi <abi_name>	Force the test to run on the given ABI, 32 or 64. By default GTS runs a test once for each ABI the device supports.
--logcat, --bugreport, and --screenshoot-on-failure	Give more visibility into failures and can help with diagnostics.
--device-token	Specifies a given device has the given token eg. --device-token 1a2b3c4d:sim-card..
--skip-device-info	Skips collection of information about the device. Note: do not use this option when running GTS for approval.

Host	Description
--skip-preconditions	Bypasses verification and setup of the device's configuration, such as pushing media files or checking for Wi-Fi connection.
List	Description
list modules	List all available test modules in the repository.
list plans or list configs	List all available test plans (configs) in the repository.
list invocations	List 'run' commands currently being executed on devices.
list commands	List all 'run' commands currently in the queue waiting to be assigned to devices.
list results	List GTS results currently stored in repository.
list devices	List currently connected devices and their state. 'Available' devices are functioning, idle devices, available for running tests. 'Unavailable' devices are devices visible via adb, but are not responding to adb commands and won't be allocated for tests. 'Allocated' devices are devices currently running tests.
Dump	Description
dump logs	Dump the tradefed logs for all running invocations.

3.1.2 完整测试指令

测试命令：run gts

参数：

-s : 平板序列号可通过"l d" (list device 的首字母缩写) 命令查看

--logcat-on-failure : 抓取 fail 项 log

--shard-count x : 使用 x 台机器并行测试

--abi(-a) arm64-v8a 或者 --abi(-a) armeabi-v7a : 指定测试 64/32 系统

指定本地多媒体：准备一张 32G 的 TF 卡，插入待测设备，由于多媒体文件过大，至少要保证有 32G 的容量大小。

可在以下网址分别下载 exoplayer、youtube、wvmedia 多媒体文件：

<https://www.google.com/url?q=https://storage.googleapis.com/exoplayer-test-media-1/gen-4/exoplayer-gts-media.zip&ust=1572677040000000&usg=AFQjCNEBo0D67jzc3ct6VsUWDGMOh3PVNw&hl=zh-CN>

<https://storage.googleapis.com/youtube-test-media/gts/GtsYouTubeTestCases-media-1.2.zip>

https://www.google.com/url?q=https://storage.googleapis.com/gts_media/wvmedia-gts-media.zip&ust=1572677040000000&usg=AFQjCNEBo0D67jzc3ct6VsUWDGMOh3PVNw&hl=zh-CN

解压三个多媒体文件 wvmedia-gts-media.zip, exoplayer-gts-media.zip, GtsYouTubeTestCases-media-1.2.zip 到本地电脑，解压出来的文件夹分别是 gts、test, wvmedia

拷贝多媒体到设备中的 TF 卡，adb -s [serialNum] push ...[绝对路径]/gts/ /storage/9016-4EF8/, adb -s [serialNum] push ...[绝对路径]/test/ /storage/9016-4EF8/, adb -s [serialNum] push ...[绝对路径]/wvmedia/ /storage/9016-4EF8/, 注意 9016-4EF8 是指 TF 卡在设备中的名字。

测试指令添加参数：

1, 修改 gts 文件夹中的 dynamic-config-sdcard-1.0.json 文件，把 ``file:" 的内容改为指向 TF 中的 gts 文件夹路径，如 ``file:///storage/9016-4EF8/gts/exoplayer/"，把该文件拷贝到本地电脑，如/home/user/gts_media/dynamic-config-sdcard-1.0.json,

测试参数：--module-arg GtsExoPlayerTestCases:config-url:file:///home/user/gts_media/dynamic-config-sdcard-1.0.json, 注意 home/user/gts_media 是指测试电脑的路径。

2, --module-arg ``GtsYouTubeTestCases:skip-media-download:true" --module-arg ``GtsYouTubeTestCases:instrumentation-arg:media-path:=/storage/9016-4EF8/test"

3, --module-arg ``GtsMediaTestCases:instrumentation-arg:mediapath:=file:///storage/9016-4EF8/wvmedia"

比如：

启动 GTS 32bit (armeabi-v7a) 环境测试：

run gts --shard-count 2 -s < 平板序列号 1> -s < 平板序列号 2> -a armeabi-v7a --logcat-on-failure

启动 GTS 64bit (arm64-v8a) 环境测试:

```
run gts --shard-count 2 -s < 平板序列号 1> -s < 平板序列号 2> -a arm64-v8a --logcat-on-failure
```

3.2 GTS 补测

3.2.1 Retry 测试

除谷歌允许的 FAIL 外, 如果 GTS 测试 fail 项超过允许的 FAIL 项, 要进行补测。

测试命令:

```
run retry --retry <session_id> -s <serial>
```

session_id: 可通过 "l r" 命令查看

比如通过如下命令启动补测:

```
run retry -r session_id --shard-count 2 -s 平板序列号 1 -s 平板序列号 2
```

3.2.2 创建补测测试计划以及启动补测

Add: 可以通过 help add 命令查看帮助

a/add s/subplan: create a subplan from a previous session

Options:

--session <session_id>: The session used to create a subplan.

--name/-n <subplan_name>: The name of the new subplan.

--result-type <status>: Which results to include in the subplan. One of passed, failed, not_executed.Repeatable.

例: a s --session 2 --name 2 --result-type failed --result-type not_executed

通过如下命令启动补测:

`run gts --subplan subplan_name` (subplan_name : 创建测试计划中创建的名字)

3.3 针对性测试

可以针对某个测试包，测试类或者具体测试用例进行测试。如下图所示。

`run <plan> --module/-m <module> --test/-t <test_name>`: run a specific test from the module. Test name can be `<package>.<class>`, `<package>.<class>#<method>` or `<native_name>`.

例：

1. 测试整个 module

`run gts -m GtsNetTestCases`

2. 测试整个 class

`run gts -m GtsNetTestCases -t com.google.android.net.gts.policy.DataSaverTest`

3. 测试一项 test

`run gts -m GtsNetTestCases -t com.google.android.net.gts.policy.DataSaverTest#testRequiredWhitelist`

Test	Result	Details
com.google.android.net.gts.policy.DataSaverTest#testRequiredWhitelist	fail	junit.framework.AssertionFailedError

图 1: 测试 module, class, testcase 分布

3.4 跳过某些有问题的测试项

比如测试的时候 `GtsNetTestCases` 项目出现了问题，导致机器卡死或者测试中断，导致无法进行其他的项目的测试，这个时候可以选择跳过测试该项目，在执行的命令加上：

`--exclude-filter GtsNetTestCases` 或者 `--exclude-filter "GtsNetTestCases"`

`com.google.android.net.gts.policy.DataSaverTest#testRequiredWhitelist`", 注意, 跳过某个用例时, 模块与用例名要用空格隔开, 且要有双引号包住。

其他与此类推, 跳过多项时, 重复输入参数。

3.5 循环测试

如需多次执行测试, 无需等待正在执行的测试完成, 直接输入多次测试命令即可, 这些命令会被缓存起来, 被依次调用。

4. 测试结果分析与调试

4.1 从 GTS 报告中获取出错信息

测试后的结果保存在 `android-gts\results` 目录下，可以通过浏览器打开 `test_result_failures_suite.html` 文件显示出测试的结果，在错误项的 `Details` 栏中给出了基本的错误信息。

测试过程抓取的 log 保存在 `android-gts\logs` 目录下。

4.2 查找相关 log 信息

测试时加上 `--logcat-on-failure` 参数，`gts` 工具会自动将 fail 的测试 log 保存在 `android-gts\logs` 下，打开 log 文件，搜索 `TestRunner`。测试过程的 log 以 `TestRunner: started:....` 开始，以 `TestRunner: finished:` 结束。

5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgment to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This document neither states nor implies warranty of any kind, including fitness for any particular application. tates nor implies warranty of any kind, including fitness for any particular application.