

# GMS (CTS、GTS、CTSVerifier) 4.4\_r3

## 认证测试完全教程总结

GMS 认证测试包括：CTS、GTS、CTSVerifier 三个测试。现将这段时间测试的注意事项做一个总结。

### 一、测试环境搭建

PC 必须是 linux 系统，最好是 64 位的 ubuntu 系统，系统请使用英语版本，不然生成的测试报告中会有中文字符。推荐使用 ubuntu-10.04-desktop-i386，而 12.04 版本不推荐。

1、安装 java6 环境(重点，搭建 CTS 环境成败在此)

#### 下载 java6

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

终端输入：sudo su -

输入密码

sudo apt-get install sun-java-jdk

当出现如下错误：

```
root@stu-system:/home# sudo apt-get install sun-java6-jdk
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

Package sun-java6-jdk is not available, but is referred to by another package.

This may mean that the package is missing, has been obsoleted, or is only available from another source

E: Package 'sun-java6-jdk' has no installation candidate

解决办法:

1.命令行输入 `sudo gedit /etc/apt/sources.list`

2.打开源列表在最后一行添加:

`deb http://us.archive.ubuntu.com/ubuntu/ hardy multiverse`

作用是在此网站下载 **Java**, 如果不行, 需要增加最新的可用的 网站。

3.保存退出

4.更新 apt

命令行输入: `sudo apt-get update`

5.安装 sun-java6-jdk

命令行输入: `sudo apt-get install sun-java6-jdk`

安装过程中根据提示选择"Y", 或"OK".

2、下载安装文件

1.下载 CTS 测试包 `android-cts-4.4_r3-linux_x86-arm.zip`,

`android-cts-media-1.0`

网址: <http://source.android.com/compatibility/downloads.html>

最近谷歌被封，请使用 VPN 登录下载。

## 2. 下载 android-sdk\_r23.0.2-linux.tgz

网址：<http://developer.android.com/sdk/index.html>

3. 先在 Windows 环境下把三个文件解压缩，在 home 文件夹下面新建一个文件夹，把这三个文件复制到其中。

## 配置 SDK 路径

### 1. 用 vi 编辑器打开

android-cts-4.0.3\_r1-linux\_x86-arm\android-cts\tools 下面的  
cts-tradefed 脚本。

### 2. 在代码最前面添加如下代码：

```
if [ -z "${SDK_ROOT}" ]; then
```

```
# CONFIGURATION
```

```
# Set this variable to the root of your Android SDK installation.
```

```
export
```

```
SDK_ROOT=/home/nick/Desktop/CTS/adt-bundle-linux-x86_64-2014  
0321/
```

```
fi;
```

保存。

### 3. 在

```
checkPath adb
```

```
checkPath java
```

之前，插入如下代码：

```
# Add SDK_ROOT to the PATH for backwards compatibility with prior
startcts

# commands that required SDK_ROOT to find adb.

if [ -n "${SDK_ROOT}" ]; then

PATH=${SDK_ROOT}/platform-tools:${SDK_ROOT}/tools:${PATH}

fi;
```

保存，并在 command line 输入命令 `chmod 777 cts-tradefed`。

4.记事本或vi编辑器打开 android-cts-media-1.0\下面的 copy\_media.sh 脚本。

5.在代码最前面添加如下代码：

```
if [ -z "${SDK_ROOT}" ]; then

# CONFIGURATION

# Set this variable to the root of your Android SDK installation.

export

SDK_ROOT=/home/nick/Desktop/CTS/adt-bundle-linux-x86_64-2014

0321/

fi;
```

保存，并在 command line 输入命令 `chmod 777 copy_media.sh`。

## 二、准备平台

CTS 测试前手机端需要做的设置：

0. 测试开始前执行一次回复出厂设置或者重新刷机；

1. 写 IMEI 号：353844060000423，353844060000431；

2. 插入 SIM 卡,且要给 SIM 卡写入本机号码(这样可以通过 sms 测试);
3. 语言设置英文, 选择 android 默认输入法;
4. 连接 Wifi, 连接 VPN 帐号;
5. 插入 T 卡 (如果手机自带存储空间足够, 1G 以上, 就不用 T 卡);
6. 开启 USB 调试、Stay awake、allow mock locations ;
7. adb install android-cts/repository/testcases/CtsDeviceAdmin.apk 安装

选择设备管理员 (Settings>Security>Select device administrators>Activate two android.deviceadmin) ;

这其中有三项, 请选择前面两项即可。

8. 设置正确的日期、时间(重点, 不然很多项不过)。
9. 从 android-cts/repository/testcases/取出 CtsDeviceAdmin.apk 安装到手机上去;
- 10 将屏幕超时时间设置为 30 分钟以上;
11. 测试过程中不要按键或触摸屏幕;
12. 不添加任何账户 (包括 google 账户);
13. 开始测试前手机处于待机界面;
14. 编译软件要使用 release-keys 进行签名  
(MTK\_SIGNATURE\_CUSTOMIZATION=yes)
15. 拷贝视频文件到 T 卡

在 window 环境下加压 android-cts-media-1.0, 然后拷贝到在 Linux 环境下, 使用命令安装到手机端中:

```
cd /home/nick/Desktop/media
```

```
./copy_media.sh
```

16. 设置-->安全-->禁止安全未知来源的应用

17. 安装 CTS 测试环境检查工具：CTSAssistant\_V1.apk

### 三、测试

#### 1、copy media 文件

1) 平台用 USB 线连接 LinuxPC （连通后平台右下方会显示 usb 和 debugicon）

2) 打开终端

3) 输入 `sudo su -`

输入 root 密码

4) 输入 `cd ../..`

5) 输入 `cd /home/user/cts4.0/android-cts-media-1.0`

//这个是 `copy_media.sh` 文件所在的路径，请输入您的实际路径。

输入 `./copy_media.sh`

。。。。脚本开始 copy media 文件到平台根目录下。。。。。

#### 2、完全测试

1) 平台用 USB 线连接 LinuxPC （连通后平台右下方会显示 usb 和 debugicon）

2) 打开终端

3) 输入 `sudo su -`

输入 root 密码

4) 输入 `cd ../../`

输入 `cd`

`/home/linux/cts/android-cts-4.0.3_r1-linux_x86-arm/android-cts/tools`

//这个是 `cts-tradefed` 文件所在路径，请输入您的实际路径

输入 `./cts-tradefed`

终端显示：

01-1615:44:05I/DeviceManager:Detectednewdevice0123456789ABC  
DEF

//如果没有这行，说明设备没有和 PC 联通

Android CTS 4.0.3\_r1

cts-tf >

5) 输入 cts 完全测试命令：`run cts --plan CTS`

cts-tf >`run cts --plan CTS`

01-16 15:45:34 I/TestInvocation: Starting invocation for 'cts' on  
build'4.0.3\_r1' on device 0123456789ABCDEF

01-16 15:45:35I/0123456789ABCDEF: Created result dir

2012.01.16\_15.45.34

。。。。开始测试 CTS。。。。

### 3、测试未完成的 plan

1) 平台用 USB 线连接 LinuxPC （连通后平台右下方会显示 usb 和 debugicon）

2) 打开终端

3) 输入 `sudo su -`

输入 root 密码

4) 输入 `cd ../../`

输入 `cd`

`/home/nick/Desktop/CTS/android-cts-4.4_r1-linux_x86-arm/android-cts/tools`

//这个是 `cts-tradefed` 文件所在路径，请输入您的实际路径

输入 `./cts-tradefed`

终端显示:



01-1615:44:05I/DeviceManager:Detectednewdevice0123456789ABC

DEF

//如果没有这行，说明设备没有和 PC 联通

Android CTS 4.4\_r3

cts-tf >

## 5) 输入 l r

终端显示:

Session	Pass	Fail	Not Executed	Start time	Planname
0	7	0	0	2012.01.16_16.09.19	NA
1	13	0	0	2012.01.16_16.13.01	NA
2	19	9	17120	2012.01.18_13.43.56	CTS

(根据 start time 时间，确定你要的任务的 session 编号)

## 6) 输入 run cts -continue-session session\_id (session 编号)

终端显示

01-1813:48:45I/TestInvocation:Startinginvocationfor'cts'onbuild'4.0.3

\_r1'ondevice0123456789ABCDEF

。。。。继续测试未完成的 plan。。。。

## 4、测试某一个 plan 的全部 fail/notExecuted/timeout 项

1)连接平台和 PC

2)终端输入 `sudo su -`

输入 root 密码

3)cd 到 android-cts/tools 下输入 `./cts-tradefed`

4)确认平台和 PC 连通后在 `cts-tf >` 输入 `l r`

终端显示:

	Session	Pass	Fail	Not Executed	Start time
Planname					
	0	7	0	0	2012.01.16_16.09.19
NA					
	1	13	0	0	2012.01.16_16.13.01
NA					
	2	19	9	17120	2012.01.18_13.43.56
CTS					

输入 `add derivedplan --plan plan_name -s sessionID -r fail`

//plan\_name 可以自己定义, sessionID 就是上面查看结果的 Session 编号。测试全部 fail 项-r 后面是 fail, 测试 notExecuted 项或者 timeout 项, -r 后面就是 notExecuted 或者 timeout。

例: 测试上面 SessionID 为 2 的所有 fail 项, 输入命令应为:

`add derivedplan --plan`

输入 `run cts --plan plan_name`

//plan\_name 即您前面定义的。

终端显示

```
cts-tf> 01-18 13:48:45 I/TestInvocation: Starting invocation  
for 'cts' on build'4.0.3_r1' on device 0123456789ABCDEF
```

。。。。开始测试 fail/notExecuted/timeout 项。。。。

## 5、测试某一个 package

1)连接平台和 PC

2)终端输入 `sudo su -`

输入 root 密码

3)cd 到 `android-cts/tools` 下输入

输入 `./cts-tradefed`

4)输入命令： `lp` 查看所有 package。

5)输入命令： `run cts -c package_name`

如测试 `android.bluetooth`，输入命令： `run cts -c  
android.bluetooth`

终端显示

```
cts-tf> 01-18 13:48:45 I/TestInvocation: Starting invocation  
for 'cts' on build'4.0.3_r1' on device 0123456789ABCDEF
```

。。。。开始测试一个 package。。。。

## 6、测试摸一个 case

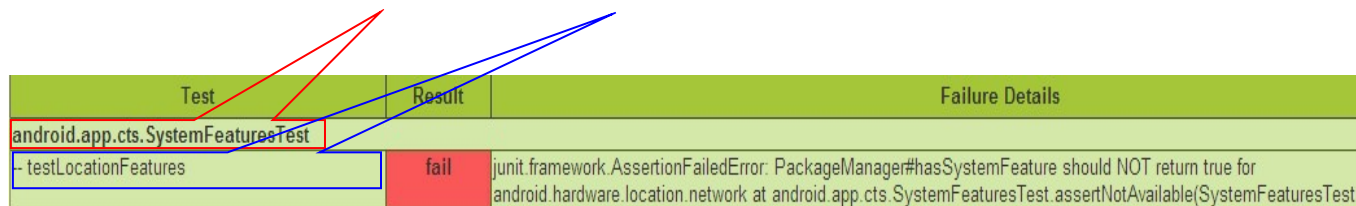
1)连接平台和 PC

2)终端输入 `sudo su -`

输入 root 密码

3)cd 到 `android-cts/tools` 下输入`./cts-tradefed`

4) 输入命令: `run cts -c package name --m method`



Test	Result	Failure Details
android.app.cts.SystemFeaturesTest		
- testLocationFeatures	fail	junit.framework.AssertionFailedError: PackageManager#hasSystemFeature should NOT return true for android.hardware.location.network at android.app.cts.SystemFeaturesTest.assertNotAvailable(SystemFeaturesTest

例：测试上面的一个 case，输入命令: `run cts -c`

`android.app.cts.SystemFeaturesTest --m testLocationFeatures`

终端显示

```
cts-tf> 01-18 13:48:45 I/TestInvocation: Starting invocation
for 'cts' on build'4.0.3_r1' on device 0123456789ABCDEF
```

。。。。开始测试一个 case。。。。

## 四、注意事项

1、如果忘记命令，可以在 `cts-tf>`后输入 `help` 查看命令。

2、测试完成后，自动生成的报告存放在目录：

android-cts-4.0.3\_r1-linux\_x86-arm/android-cts/repository/result 。

3、报告中包括 5 个文件，用浏览器打开 **testResult.xml** 文件查看完整报告。

4、所有 log 信息存放在目录：

android-cts-4.0.3\_r1-linux\_x86-arm/android-cts/repository/logs。

### 一些技巧总结：

1. 处理 **Not executed** 项的方法：

android 4.4 full test 耗时将近 12 小时，如果中间出现部分 **not executed** 项，不需要重新全部去测试，可以使用一下命令继续针对 **not executed** 项测试，并自动更新测试报告（这点很重要）

**run cts --continue-session session\_ID** // 要知道它的 **session\_ID**;

查看 **session\_ID** 可以输入命令：**l r**

如果仍然有，可以再进行一次上面的测试，直到所有的 **notExecuted** 项都已经 **run** 过 。

独家秘笈：开始执行后，**CTS** 已经开始在跑了，输入命令：**l r**,

出来所有的 **session\_ID**，这时可以输入命令：

**run cts --continue-session session\_ID+1**,

按回车，这样输入三遍，保证第一遍测试不会出现 **Not Executed** 项。

2. 处理因操作不当导致出现失败项的方法：

有时候我们测试完了， **Failed** 项目比较多，可以针对 **Failed** 项测试

命令 `add derivedplan --plan plan_name -s sessionID -r`  
`[pass/fail/notExecuted]`

`//plan_name` 可以自己定义，`sessionID` 就是 `lr` 查看结果的 **Session** 编号。测试全部 **fail** 项 `-r` 后面是 **fail**，测试 **notExecuted** 项或者 **timeout** 项，`-r` 后面就是 **notExecuted** 或者 **timeout**。

例：测试 **SessionID** 为 2 的所有 **fail** 项，输入命令应为：

```
add derivedplan --plan user123 -s 2 -r fail
```

```
run cts --plan plan_name
```

`//plan_name` 即你前面定义的。

例：测试上面例子中添加的 **fail** 项的 **plan**，输入命令应为：

```
run cts --plan user123
```

这种测试不会更新之前的测试报告，而是重新针对这些失败项生成新的报告，所以这个只能作为检查 **Failed** 项是否有操作不当，重新 **run** 一遍。

### 3. 需要连接 **google** 等外网的测试项处理方法：

因为 **VPN** 的不稳定性，不能保证我们在测试过程的 12 小时中一直处于连接，出现连接外网的测试项不能通过在所难免，测试结束后单独处理这些项。

打开浏览器，将生成的测试报告里的 **testResult.xml** 文件拖进浏览器，找到测试项里面失败原因是 “**www.... com**” 的项，然后用 **UE** 打开 **testResult.xml** 文件，在 **UE** 里也找到这个失败项，将测试结果 **fail** 改为 **notExecuted** 并保存，然后连接 **VPN** 执行上面的第一步。

在手机设置正确情况下、处理完 **Not executed** 项、验证过 **Failed** 项后，才可以将测试报告发出。

## 二、GTS 测试：

1. **GTS** 测试工具需要自行从 **Google** 获取，但是一般公司没有资格获取，只能到处去找；
2. **GTS** 和 **CTS** 测试很相似，命令都是几乎一致的，但 **GTS** 测试时间只有一个半小时；
3. 很多项要在线播放视频，对 **VPN** 环境要求高。

## 三、CTS Verifier 测试：

1. 首先测试 **Data backup test** ,否则后面就无法测试该项；

```
adb shell bmgr enable true
```

```
adb shell bmgr transport
```

```
android/com.android.internal.backup.LocalTransport
```

```
adb shell bmgr run
```

```
adb uninstall com.android.cts.verifier
```

adb install

F:\CTS4.0\CTS4.0.3\_r2\apk4.0.3\_r2\CtsVerifier\_4.0.3\_r1.apk

2. CTS Verifier 测试工具在 Google 官网下载

android-cts-verifier-4.4\_r3-linux\_x86-arm ;

3. CTS Verifier 测试需要 VPN 在线播放视频 ;

### 实例展示：

一次实际的 CTS 测试报告，第一次执行一遍后，还有 58 个错误。

Test Summary	
CTS version	4.4_r3
Test timeout	600000 ms
Host Info	ubuntu (Linux - 2.6.32-38-generic)
Plan name	CTS
Start time	Fri Aug 15 17:57:17 PDT 2014
End time	Sat Aug 16 01:54:23 PDT 2014
Tests Passed	24839
Tests Failed	58
Tests Timed out	0
Tests Not Executed	0

经过分析：

有 10 项左右是播放音视频文件；

有 30 多项是需要 VPN 连接，但测试过程汇总，VPN 会被断开；

有一项需要屏幕没有锁屏；



还有一项是操作点击开关按键，如果会 **WIFI** 是打开的，把 **WIFI** 关闭，所以连不上网，造成失败。所以测试前，需要先把 **WIFI** 开关按键关闭。

失败项验证：

测试后，发现 **Media** 文件被删除，需要重新复制一份 **Media** 到手机中，并且重启手机；

需要重新连接上 **VPN**，可以测试过 30 项；

重启手机；

经过验证失败项后，只有 18 个失败项，这些是谷歌允许的失败项。即是测试报告通过。

Test Summary	
CTS version	4.4_r3
Test timeout	600000 ms
Host Info	ubuntu (Linux - 2.6.32-38-generic)
Plan name	CTS
Start time	Fri Aug 15 17:57:17 PDT 2014
End time	Sat Aug 16 11:38:23 PDT 2014
Tests Passed	24879
Tests Failed	18
Tests Timed out	0
Tests Not Executed	0

测试心得总结：

1.CTS 测试不难，搭建一个 CTS 测试环境需要一点功夫；

2.顺利的话，CTS 测试，需要 10 小时；

CTS verifier 测试，需要 2 小时；

GTS 测试，需要 1.5 小时。

3.实际过程中，CTS 测试，在晚上执行第一遍，第二天上午使用两个小时验证完失败项。

耗时是一个晚上加白天两小时；

4.要有稳定的 WiFi 网速环境；

5.要有稳定的 VPN 账号，最好是申请一个付费的，因为免费的 VPN 账号每 20 分钟会自动断开；

6.最好要有三台机器，其中 CTS verifier 测试需要两台机器；

7.有一项测试需要读写大量的文件，在虚拟机上测试，读写速度慢，容易超时报错；需要在一台纯 Linux 系统主机下测试才能通过；

8.有时插 T 卡时，会有一些测试项因为读写速度慢，超时报错，就不要插 T 卡；

9.要修改测试报告 XML 文件的办法，使用命令：`chmod 777`，后面把 XML 文件拖进来，回车即可；

10.小技巧，命令终端粘贴操作：复制文本后，按一下鼠标中间的滑轮即可。

11.通过 CTS 测试，发现自己的 Linux 环境操作熟练了不少。