

android adb shell 命令大全

1. 显示系统中全部 Android 平台:

```
android list targets
```

2. 显示系统中全部 AVD (模拟器):

```
android list avd
```

3. 创建 AVD (模拟器):

```
android create avd --name 名称 --target 平台编号
```

4. 启动模拟器:

```
emulator -avd 名称 -sdcard ~/名称.img (-skin 1280x800)
```

5. 删除 AVD (模拟器):

```
android delete avd --name 名称
```

6. 创建 SDCard:

```
mksdcard 1024M ~/名称.img
```

7. AVD(模拟器)所在位置:

```
Linux(~/.android/avd)      Windows(C:\Documents and  
Settings\Administrator\.android\avd)
```

8. 启动 DDMS:

```
ddms
```

9. 显示当前运行的全部模拟器:

```
adb devices
```

10. 对某一模拟器执行命令:

```
abd -s 模拟器编号 命令
```

11. 安装应用程序:

```
adb install -r 应用程序.apk
```

12. 获取模拟器中的文件:

```
adb pull <remote> <local>
```

13. 向模拟器中写文件:

```
adb push <local> <remote>
```

14. 进入模拟器的 shell 模式:

```
adb shell
```

15. 启动 SDK, 文档, 实例下载管理器:

```
android
```

16. 卸载 apk 包:

```
adb shell
cd data/app
rm apk 包
exit
adb uninstall apk 包的主包名
adb install -r apk 包
```

17. 查看 adb 命令帮助信息:

```
adb help
```

18. 在命令行中查看 LOG 信息:

```
adb logcat -s 标签名
```

19. adb shell 后面跟的命令主要来自:

源码\system\core\toolbox 目录和源码\frameworks\base\cmds 目录。

20. 删除系统应用:

```
adb remount （重新挂载系统分区，使系统分区重新可写）。
adb shell
cd system/app
rm *.apk
```

21. 获取管理员权限:

```
adb root
```

22. 启动 Activity:

```
adb shell am start -n 包名/包名+类名 (-n 类名, -a action, -d date, -m MIME-TYPE, -c category, -e 扩展数据, 等)。
```

23. 发布端口:

你可以设置任意的端口号，作为主机向模拟器或设备的请求端口。如：
adb forward tcp:5555 tcp:8000

24. 复制文件:

你可向一个设备或从一个设备中复制文件，
复制一个文件或目录到设备或模拟器上：
adb push <source> <destination></destination></source>
如：adb push test.txt /tmp/test.txt
从设备或模拟器上复制一个文件或目录：
adb pull <source> <destination></destination></source>
如：adb pull /addroid/lib/libwebcore.so .

25. 搜索模拟器/设备的实例:

取得当前运行的模拟器/设备的实例的列表及每个实例的状态：
adb devices

26. 查看 bug 报告:

adb bugreport

27、记录无线通讯日志:

一般来说,无线通讯的日志非常多,在运行时没必要去记录,但我们还是可以通过命令,设置记录:

```
adb shell
logcat -b radio
```

28、获取设备的 ID 和序列号:

```
adb get-product
adb get-serialno
```

29、访问数据库 SQLite3

```
adb shell
sqlite3
```

#cd system/sd/data //进入系统内指定文件夹

#ls //列表显示当前文件夹内容

#rm -r xxx //删除名字为 xxx 的文件夹及其里面的所有文件

#rm xxx //删除文件 xxx

#rmdir xxx //删除 xxx 的文件夹

<http://xuxd32.blog.163.com/blog/static/4812193320118133049753/>

```
adb shell am instrument -w com.android.music.tests/.MusicPlayerFunctionalTestRunner
```

```
adb shell am startservice -n com.youcan/com.youcan.GetInfoService
```

```
adb shell am start -n
```

```
adb shell am broadcast -a android.provider.Telephony.SECRET_CODE -d
```

```
android_secret_code://28346
```

```
adb shell am broadcast -a android.provider.Telephony.SECRET_CODE -d
```

```
android_secret_code://767*3855
```

```
usage: am [subcommand] [options]
```

start an Activity: am start [-D] [-W] <INTENT>

-D: enable debugging

-W: wait for launch to complete

start a Service: am startservice <INTENT>

send a broadcast Intent: am broadcast <INTENT>

start an Instrumentation: am instrument [flags] <COMPONENT>

-r: print raw results (otherwise decode REPORT_KEY_STREAMRESULT)

-e <NAME> <VALUE>: set argument <NAME> to <VALUE>

-p <FILE>: write profiling data to <FILE>

-w: wait for instrumentation to finish before returning

start profiling: am profile <PROCESS> start <FILE>

stop profiling: am profile <PROCESS> stop

start monitoring: am monitor [--gdb <port>]

--gdb: start gdbserv on the given port at crash/ANR

<INTENT> specifications include these flags:

[-a <ACTION>] [-d <DATA_URI>] [-t <MIME_TYPE>]

[-c <CATEGORY> [-c <CATEGORY>] ...]

[-e|--es <EXTRA_KEY> <EXTRA_STRING_VALUE> ...]

[--esn <EXTRA_KEY> ...]

[--ez <EXTRA_KEY> <EXTRA_BOOLEAN_VALUE> ...]

[-e|--ei <EXTRA_KEY> <EXTRA_INT_VALUE> ...]

[-n <COMPONENT>] [-f <FLAGS>]