**adb 常用命令大全**

**adb介绍: SDK的Tools文件夹下包含着Android模拟器操作的重要命令adb，adb的全称为(Android Debug Bridge就是调试桥的作用。**

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. 缷载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，做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**

**启动UC的时候进入debug模式，只需要如下命令：**

**adb shell am set-debug-app -w com.UCMobile --persistent**

**这里 -w 表示等待debugger。ok，启动UC就可以看到已经进入debug模式了，这样不管是第三方调用，还是直接启动都可以。**

**如果想每次都进入debug模式，加上 --persistent 就可以了。**

**清除的话也很简单：**

**adb shell am clear-debug-app com.UCMobile**

**Adb发广播**

**am broadcast -a com.android.vending.INSTALL\_REFERRER -n com.UCMobile/com.uc.browser.googleanalytics.GoogleAnalyticsReceiver --es "referrer" "utm\_source=(not%20set)&utm\_medium=(not%20set)"**

1. 获取序列号：

        adb get-serialno

2. 查看连接计算机的设备：

        adb devices

3. 重启机器：

        adb reboot

4. 重启到bootloader，即刷机模式：

        adb reboot bootloader

5. 重启到recovery，即恢复模式：

        adb reboot recovery

6. 查看log：

        adb logcat

7. 终止adb服务进程：

        adb kill-server

8. 重启adb服务进程：

        adb start-server

9. 获取机器MAC地址：

        adb shell  cat /sys/class/net/wlan0/address

10. 获取CPU序列号：

        adb shell cat /proc/cpuinfo

11. 安装APK：

        adb install <apkfile> //比如：adb install baidu.apk

12. 保留数据和缓存文件，重新安装apk：

        adb install -r <apkfile> //比如：adb install -r baidu.apk

13. 安装apk到sd卡：

        adb install -s <apkfile> // 比如：adb install -s baidu.apk

14. 卸载APK：

        adb uninstall <package> //比如：adb uninstall com.baidu.search

15. 卸载app但保留数据和缓存文件：

        adb uninstall -k <package> //比如：adb uninstall -k com.baidu.search

16. 启动应用：

        adb shell am start -n <package\_name>/.<activity\_class\_name>

17. 查看设备cpu和内存占用情况：

        adb shell top

18. 查看占用内存前6的app：

        adb shell top -m 6

19. 刷新一次内存信息，然后返回：

        adb shell top -n 1

20. 查询各进程内存使用情况：

        adb shell procrank

21. 杀死一个进程：

        adb shell kill [pid]

22. 查看进程列表：

        adb shell ps

23. 查看指定进程状态：

        adb shell ps -x [PID]

24. 查看后台services信息：

        adb shell service list

25. 查看当前内存占用：

        adb shell cat /proc/meminfo

26. 查看IO内存分区：

        adb shell cat /proc/iomem

27. 将system分区重新挂载为可读写分区：

        adb remount

28. 从本地复制文件到设备：

        adb push <local> <remote>

29. 从设备复制文件到本地：

        adb pull <remote>  <local>

30. 列出目录下的文件和文件夹，等同于dos中的dir命令：

        adb shell ls

31. 进入文件夹，等同于dos中的cd 命令：

        adb shell cd <folder>

32. 重命名文件：

        adb shell rename path/oldfilename path/newfilename

33. 删除system/avi.apk：

        adb shell rm /system/avi.apk

34. 删除文件夹及其下面所有文件：

        adb shell rm -r <folder>

35. 移动文件：

        adb shell mv path/file newpath/file

36. 设置文件权限：

        adb shell chmod 777 /system/fonts/DroidSansFallback.ttf

37. 新建文件夹：

        adb shell mkdir path/foldelname

38. 查看文件内容：

        adb shell cat <file>

39. 查看wifi密码：

        adb shell cat /data/misc/wifi/\*.conf

40. 清除log缓存：

        adb logcat -c

41. 查看bug报告：

        adb bugreport

42. 获取设备名称：

        adb shell cat /system/build.prop

43. 查看ADB帮助：

        adb help

44. 跑monkey：

        adb shell monkey -v -p your.package.name 500

查看包名：

aapt dump badging [xxx.apk]

am命令，am全称activity manager，你能使用am去模拟各种系统的行为，例如去启动一个activity，强制停止进程，发送广播进程，修改设备屏幕属性等等

force-stop <PACKAGE> ： 强制停止指定的package包应用。

kill [options] <PACKAGE> ：杀死指定package包应用进程，该命令在安全模式下杀死进程，不影响用户体验。参数选项：--user <USER\_ID> | all | current: 指定user进程杀死，如果不指定默认为所有users。（关于USER\_ID下面会介绍到）

kill-all ：杀死所有的后台进程。

broadcast [options] <INTENT> ：发送一个intent。具体intent参数参照start命令参数。参数选项：--user <USER\_ID> | all | current: 指定user进程杀死，如果不指定默认为所有users。

instrument [options] <COMPONENT> ：测试命令，不多作介绍。

profile start <PROCESS> <FILE> ：在<PROCESS>进程中运行profile，分析结果写到<FILE>里。

profile stop <PROCESS> ：停止profile。

set-debug-app [options] <PACKAGE> ：设置package包应用为debug模式。参数选项：-w|--persistent：等待进入调试模式，保留值。

clear-debug-app ：清空之前用set-debug-app命令设置的package包应用。

monitor [options]

screen-compat [on|off] <PACKAGE>

display-size [reset|<WxH>]

display-density <dpi>

to-uri <INTENT>

to-intent-uri <INTENT>

1. 拨打一个电话：

am start -a android.intent.action.CALL -d tel:10086

这里-a表示动作，-d表述传入的数据，还有-t表示传入的类型。

2. 打开一个网页：

am start -a android.intent.action.VIEW -d http://www.baidu.com （这里-d表示传入的data）

3. 打开音乐播放器：

am start -a android.intent.action.MUSIC\_PLAYER 或者

am start -n com.android.music/om.android.music.MusicBrowserActivity

4. 启动一个服务：

am startservice <服务名称>

例如：am startservice -n com.android.music/com.android.music.MediaPlaybackService (这里-n表示组件)

或者 am startservice -a com.smz.myservice (这里-a表示动作，就是你在Androidmanifest里定义的)

5. 发送一个广播：

am broadcast -a <广播动作>

例如： am broadcast -a com.smz.mybroadcast

6 启动一个Activity

# am start -n 包(package)名/包名.活动(activity)名称

启动的方法可以从每个应用的AndroidManifest.xml的文件中得到

Music 和 Video（音乐和视频）的启动方法为：

# am start -n com.android.music/com.android.music.MusicBrowserActivity

# am start -n com.android.music/com.android.music.VideoBrowserActivity

# am start -n com.android.music/com.android.music.MediaPlaybackActivity

Camera（照相机）的启动方法为：

# am start -n com.android.camera/com.android.camera.Camera

Browser（浏览器）的启动方法为：

# am start -n com.android.browser/com.android.browser.BrowserActivity

启动 google map 直接定位到北京 :

am start -a android.intent.action.VIEW geo:0,0?q=beijing

pm全称package manager，你能使用pm命令去模拟android行为或者查询设备上的应用等，当你在adb shell命令下执行pm命令：

pm <command>

你也可以在adb shell前执行pm命令：

adb shell pm uninstall com.example.MyApp

关于一些pm命令的介绍：

list packages [options] <FILTER> ：打印所有包，选择性的查询包列表。参数选项：-f：查看关联文件，即应用apk的位置跟对应的包名（如：package:/system/app/MusicPlayer.apk=com.sec.android.app.music）；-d：查看disabled packages；-e：查看enable package；-s：查看系统package；-3：查看第三方package；-i：查看package的对应安装者（如：1、package:com.tencent.qqmusic installer=null 2、package:com.tencent.qqpim installer=com.android.vending）；-u：查看曾被卸载过的package。（卸载后又重新安装依然会被列入）；--user<USER\_ID>：The user space to query。

list permission-groups ：打印所有已知的权限群组。

list permissions [options] <GROUP> ：选择性的打印权限。参数选项：

list features ：设备特性。硬件之类的性能。

list libraries ：当前设备支持的libs。

list users ：系统上所有的users。（上面提到的USER\_ID查询方式，如：UserInfo{0:Primary:3}那么USER\_ID为0）

path <PACKAGE> ：查询package的安装位置。

install [options] <PATH> ：安装命令。

uninstall [options] <PACKAGE> ：卸载命令。

clear <PACKAGE> ：对指定的package删除所有数据。

enable <PACKAGE\_OR\_COMPONENT> ：使package或component可用。（如：pm enable \" package/class\" ）

disable <PACKAGE\_OR\_COMPONENT> ：使package或component不可用。（如：pm disable \" package/class\" ）

disable-user [options] <PACKAGE\_OR\_COMPONENT> ：参数选项：--user <USER\_ID>: The user to disable.

grant <PACKAGE\_PERMISSION> ：授权给应用。

revoke <PACKAGE\_PERMISSION> ：撤销权限。

set-install-location <LOCATION> ：设置默认的安装位置。其中0：让系统自动选择最佳的安装位置。1：安装到内部的设备存储空间。2：安装到外部的设备存储空间。（这只用于调试应用程序，使用该命令可能导致应用程序退出或者其他不适的后果）。

get-install-location ：返回当前的安装位置。返回结果同上参数选项。

set-permission-enforced <PERMISSION> [true|false] ：使指定权限生效或者失效。

create-user <USER\_NAME> ：增加一个新的USER。

remove-user <USER\_ID> ：删除一个USER。

get-max-users ：该设备所支持的最大USER数。（