**adb命令**

db 命令的基本语法如下：  
adb [-d|-e|-s ]   
如果只有一个设备/模拟器连接时，可以省略掉 [-d|-e|-s ] 这一部分，直接使用 adb 。  
参数 （含义）  
-d | 指定当前唯一通过 USB 连接的 Android 设备为命令目标  
-e | 指定当前唯一运行的模拟器为命令目标  
-s | 指定相应 serialNumber 号的设备/模拟器为命令目标

1.查看设备adb devices  
这个命令是查看当前连接的设备, 连接到计算机的android设备或者模拟器将会列出显示；  
命令：  
adb devices  
输出示例：  
List of devices attached  
cf264b8f device  
emulator-5554 device

输出格式为 [serialNumber] [state]，serialNumber 即我们常说的 SN，state 有如下几种：  
offline —— 表示设备未连接成功或无响应。  
device —— 设备已连接。注意这个状态并不能标识 Android 系统已经完全启动和可操作，在设备启动过程中设备实例就可连接到 adb，但启动完毕后系统才处于可操作状态。  
no device —— 没有设备/模拟器连接。

以上输出显示当前已经连接了两台设备/模拟器，cf264b8f 与 emulator-5554分别是它们的 SN。从 emulator-5554 这个名字可以看出它是一个 Android 模拟器。

常见异常输出：  
没有设备/模拟器连接成功。  
List of devices attached  
设备/模拟器未连接到 adb 或无响应。  
List of devices attached  
cf264b8f offline

2.安装软件adb install <apk文件路径>  
这个命令将指定的apk文件安装到设备上  
参数-r,它是更新安装（允许覆盖安装）的意思，  
参数 -s ,将应用安装到sdcard.  
参数 -d ,允许降级覆盖安装  
如: adb install com.rc.gebao.apk。  
如果见到类似如下输出（状态为 Success）代表安装成功：  
3.0 MB/s (12358418 bytes in 3.893s)  
pkg: /data/local/tmp/app-release.apk  
Success  
而如果状态为 Failure 则表示安装失败。常见安装失败输出代码、含义及可能的解决办法如下：

输出 | 含义 | 解决办法;  
INSTALL\_FAILED\_ALREADY\_EXISTS | 应用已经存在 | 使用 -r 参数;  
INSTALL\_FAILED\_INVALID\_APK | 无效的 APK 文件 |  ;  
INSTALL\_FAILED\_INVALID\_URI | 无效的 APK 文件名 | 确保 APK 文件名里无中文;  
INSTALL\_FAILED\_INSUFFICIENT\_STORAGE | 空间不足 | 清理空间;  
INSTALL\_FAILED\_DUPLICATE\_PACKAGE | 已经存在同名程序 |  ;  
INSTALL\_FAILED\_NO\_SHARED\_USER | 请求的共享用户不存在 |  ;  
INSTALL\_FAILED\_UPDATE\_INCOMPATIBLE | 已经安装过签名不一样的同名应用，且数据没有移除 |  ;  
INSTALL\_FAILED\_SHARED\_USER\_INCOMPATIBLE | 请求的共享用户存在但签名不一致 |  ;  
INSTALL\_FAILED\_MISSING\_SHARED\_LIBRARY | 安装包使用了设备上不可用的共享库 |  ;  
INSTALL\_FAILED\_REPLACE\_COULDNT\_DELETE | 替换时无法删除 |  ;  
INSTALL\_FAILED\_DEXOPT | dex 优化验证失败或空间不足 |  ;  
INSTALL\_FAILED\_OLDER\_SDK | 设备系统版本低于应用要求 |  ;  
INSTALL\_FAILED\_CONFLICTING\_PROVIDER | 设备里已经存在与应用里同名的 content provider |  ;  
INSTALL\_FAILED\_NEWER\_SDK | 设备系统版本高于应用要求 |  ;  
INSTALL\_FAILED\_TEST\_ONLY | 应用是 test-only 的，但安装时没有指定 -t 参数 |  ;  
INSTALL\_FAILED\_CPU\_ABI\_INCOMPATIBLE | 包含不兼容设备 CPU 应用程序二进制接口的 native code |  ;  
INSTALL\_FAILED\_MISSING\_FEATURE | 应用使用了设备不可用的功能 |  ;  
INSTALL\_FAILED\_CONTAINER\_ERROR | sdcard 访问失败 | 确认 sdcard 可用，或者安装到内置存储;  
INSTALL\_FAILED\_INVALID\_INSTALL\_LOCATION | 不能安装到指定位置 | 切换安装位置，添加或删除 -s 参数;  
INSTALL\_FAILED\_MEDIA\_UNAVAILABLE | 安装位置不可用 | 一般为 sdcard，确认 sdcard 可用或安装到内置存储;  
INSTALL\_FAILED\_VERIFICATION\_TIMEOUT | 验证安装包超时 |  ;  
INSTALL\_FAILED\_VERIFICATION\_FAILURE | 验证安装包失败 |  ;  
INSTALL\_FAILED\_PACKAGE\_CHANGED | 应用与调用程序期望的不一致 |  ;  
INSTALL\_FAILED\_UID\_CHANGED | 以前安装过该应用，与本次分配的 UID 不一致 | 清除以前安装过的残留文件;  
INSTALL\_FAILED\_VERSION\_DOWNGRADE | 已经安装了该应用更高版本 | 使用 -d 参数;  
INSTALL\_FAILED\_PERMISSION\_MODEL\_DOWNGRADE | 已安装 target SDK 支持运行时权限的同名应用，要安装的版本不支持运行时权限 |  ;  
INSTALL\_PARSE\_FAILED\_NOT\_APK | 指定路径不是文件，或不是以 .apk结尾 |  ;  
INSTALL\_PARSE\_FAILED\_BAD\_MANIFEST | 无法解析的 AndroidManifest.xml 文件 |  ;  
INSTALL\_PARSE\_FAILED\_UNEXPECTED\_EXCEPTION | 解析器遇到异常 |  ;  
INSTALL\_PARSE\_FAILED\_NO\_CERTIFICATES | 安装包没有签名 |  ;  
INSTALL\_PARSE\_FAILED\_INCONSISTENT\_CERTIFICATES | 已安装该应用，且签名与 APK 文件不一致 | 先卸载设备上的该应用，再安装;  
INSTALL\_PARSE\_FAILED\_CERTIFICATE\_ENCODING | 解析 APK 文件时遇到CertificateEncodingException |  ;  
INSTALL\_PARSE\_FAILED\_BAD\_PACKAGE\_NAME | manifest 文件里没有或者使用了无效的包名 |  ;  
INSTALL\_PARSE\_FAILED\_BAD\_SHARED\_USER\_ID | manifest 文件里指定了无效的共享用户 ID |  ;  
INSTALL\_PARSE\_FAILED\_MANIFEST\_MALFORMED | 解析 manifest 文件时遇到结构性错误 |  ;  
INSTALL\_PARSE\_FAILED\_MANIFEST\_EMPTY | 在 manifest 文件里找不到找可操作标签（instrumentation 或 application） |  ;  
INSTALL\_FAILED\_INTERNAL\_ERROR | 因系统问题安装失败 |  ;  
INSTALL\_FAILED\_USER\_RESTRICTED | 用户被限制安装应用 |  ;  
INSTALL\_FAILED\_DUPLICATE\_PERMISSION | 应用尝试定义一个已经存在的权限名称 |  ;  
INSTALL\_FAILED\_NO\_MATCHING\_ABIS | 应用包含设备的应用程序二进制接口不支持的 native code |  ;  
INSTALL\_CANCELED\_BY\_USER | 应用安装需要在设备上确认，但未操作设备或点了取消 | 在设备上同意安装;  
INSTALL\_FAILED\_ACWF\_INCOMPATIBLE | 应用程序与设备不兼容 |  ;  
does not contain AndroidManifest.xml | 无效的 APK 文件 |  ;  
is not a valid zip file | 无效的 APK 文件 |  ;  
Offline | 设备未连接成功 | 先将设备与 adb 连接成功;  
unauthorized | 设备未授权允许调试 |  ;  
error: device not found | 没有连接成功的设备 | 先将设备与 adb 连接成功;  
protocol failure | 设备已断开连接 | 先将设备与 adb 连接成功;  
Unknown option: -s | Android 2.2 以下不支持安装到 sdcard | 不使用 -s 参数;  
No space left on devicerm | 空间不足 | 清理空间;  
Permission denied ... sdcard ... | sdcard 不可用 |  ;

1. 卸载软件  
   　　adb uninstall <包名>  
   如果加 -k 参数,为卸载软件但是保留配置和缓存文件.  
   如: adb uninstall com.rc.gebao  
   adb uninstall com.qihoo360.mobilesafe 表示卸载 360 手机卫士。
2. 登录设备shell  
   　　adb shell  
   　　adbshell <command命令>  
   　　这个命令将登录设备的shell.  
   　　后面加<command命令>将是直接运行设备命令, 相当于执行远程命令  
   如: adb shell cat /proc/kmsg
3. 从电脑上发送文件到设备  
   　　adb push <电脑上的文件路径> <设备里的目录>  
   　　用push命令可以把本机电脑上的文件或者文件夹复制到设备(手机)  
   如: adb push /local/build.prop /system/build.prop  
   adb push ~/sr.mp4 /sdcard/  
   小技巧：设备上的文件路径普通权限可能无法直接写入，如果你的设备已经 root 过，可以先 adb push /path/on/pc /sdcard/filename，然后 adb shell 和su 在 adb shell 里获取 root 权限后，cp /sdcard/filename /path/on/device。
4. 从设备上下载文件到电脑  
   　　adb pull <设备里的目录> <电脑上的目录>  
   　　用pull命令可以把设备(手机)上的文件或者文件夹复制到本机电脑;  
   其中 电脑上的目录 参数可以省略，默认复制到当前目录。  
   如: adb pull /system/build.prop /local/  
   adb pull /sdcard/sr.mp4 ~/tmp/  
   小技巧：设备上的文件路径可能需要 root 权限才能访问，如果你的设备已经 root 过，可以先使用 adb shell 和 su 命令在 adb shell 里获取 root 权限后，先 cp /path/on/device /sdcard/filename 将文件复制到 sdcard，然后 adb pull /sdcard/filename /path/on/pc。
5. 同步更新  
   adb sync [ ]  
   如果不指定目录,将同时更新 /data 和 /system/  
   如: adb sync /data/
6. 显示帮助信息  
   　　adb help  
   　　这个命令将显示帮助信息
7. 重新挂载  
   adb remount  
   重新挂载系统 分区 用于读写
8. 启动和停止  
   adb start-server - ensure that there is a server running  
   （一般无需手动执行此命令，在运行 adb 命令时若发现 adb server 没有启动会自动调起。）  
   adb kill-server - kill the server if it is running
9. 重启设备  
   adb reboot [bootloader|recovery]  
   - reboots the device, optionally into the bootloader or recovery program
10. 查看Log  
    [adb] logcat [] ... [] ..  
    -b 加载一个可使用的日志缓冲区供查看，比如event 和radio . 默认值是main 。具体查看Viewing Alternative Log Buffers.  
    -c 清楚屏幕上的日志.  
    -d 输出日志到屏幕上.  
    -f 指定输出日志信息的 ，默认是stdout .  
    -g 输出指定的日志缓冲区，输出后退出.  
    -n 设置日志的最大数目 .，默认值是4，需要和 -r 选项一起使用。  
    -r 每 时输出日志，默认值为16，需要和-f 选项一起使用.  
    -s 设置默认的过滤级别为silent.  
    -v 设置日志输入格式，默认的是brief 格式

where is a log component tag (or \* for all) and priority is:  
V Verbose（最低，输出得最多）  
D Debug  
I Info  
W Warn  
E Error  
F Fatal  
S Silent (supress all output)（最高，啥也不输出）  
'' means ':d' and by itself means :v  
按某级别过滤日志则会将该级别及以上的日志输出。  
比如，命令：adb logcat \*:W  
会将 Warning、Error、Fatal 和 Silent 日志输出。

按 tag 和级别过滤日志：adb logcat ActivityManager:I MyApp:D \*:S  
表示输出 tag ActivityManager 的 Info 以上级别日志，输出 tag MyApp 的 Debug 以上级别日志，及其它 tag 的 Silent 级别日志（即屏蔽其它 tag 日志）。

日志格式  
可以用 adb logcat -v 选项指定日志输出格式。

日志支持按以下几种 ：  
--brief  
默认格式。格式为：  
/():   
示例：  
D/HeadsetStateMachine( 1785): Disconnected process message: 10, size: 0

--process  
格式为：  
()   
示例：  
D( 1785) Disconnected process message: 10, size: 0 (HeadsetStateMachine)  
--tag  
格式为：  
/:   
示例：  
D/HeadsetStateMachine: Disconnected process message: 10, size: 0  
--raw  
格式为：  
  
示例：  
Disconnected process message: 10, size: 0  
--time  
格式为：  
/():   
示例：  
08-28 22:39:39.974 D/HeadsetStateMachine( 1785): Disconnected process message: 10, size: 0  
--threadtime  
格式为：  
:   
示例：  
08-28 22:39:39.974 1785 1832 D HeadsetStateMachine: Disconnected process message: 10, size: 0  
--long  
格式为：  
[ : / ]  
  
示例：  
[ 08-28 22:39:39.974 1785: 1832 D/HeadsetStateMachine ]  
Disconnected process message: 10, size: 0  
指定格式可与上面的过滤同时使用。比如：  
adb logcat -v long ActivityManager:I \*:S

13、查看bug报告：  
adb bugreport  
14.查看adb版本：  
adb version  
15.清除应用数据与缓存  
adb shell pm clear   
表示应用名包，这条命令的效果相当于在设置里的应用信息界面点击了「清除缓存」和「清除数据」。  
命令示例：  
adb shell pm clear com.qihoo360.mobilesafe  
表示清除 360 手机卫士的数据和缓存。

16.查看前台 Activity  
adb shell dumpsys activity activities | grep mFocusedActivity  
输出示例：  
mFocusedActivity: ActivityRecord{f9cd503 u0 com.rc.gebao/.MainActivity t59}  
其中的 com.rc.gebao/.MainActivity 就是当前处于前台的 Activity。

17.清空日志adb logcat -c  
18.内核日志adb shell dmesg  
输出示例：  
<6>[14201.684016] PM: noirq resume of devices complete after 0.982 msecs  
<6>[14201.685525] PM: early resume of devices complete after 0.838 msecs  
<6>[14201.753642] PM: resume of devices complete after 68.106 msecs  
<4>[14201.755954] Restarting tasks ... done.  
<6>[14201.771229] PM: suspend exit 2016-08-28 13:31:32.679217193 UTC  
<6>[14201.872373] PM: suspend entry 2016-08-28 13:31:32.780363596 UTC  
<6>[14201.872498] PM: Syncing filesystems ... done.  
中括号里的 [14201.684016] 代表内核开始启动后的时间，单位为秒。

通过内核日志我们可以做一些事情，比如衡量内核启动时间，在系统启动完毕后的内核日志里找到 Freeing init memory 那一行前面的时间就是。

19.查看设备信息  
--型号命令：  
adb shell getprop ro.product.model  
输出示例：  
NX511J  
--电池状况  
adb shell dumpsys battery  
输出示例：  
Current Battery Service state:  
AC powered: false  
USB powered: true  
Wireless powered: false  
status: 2  
health: 2  
present: true  
level: 100  
scale: 100  
voltage: 4387  
temperature: 336  
technology: Li-ion  
其中 scale 代表最大电量，level 代表当前电量。上面的输出表示100% 的电量满。  
--屏幕分辨率  
adb shell wm size  
输出示例：  
Physical size: 1080x1920  
该设备屏幕分辨率为 1080px \* 1920px。  
--屏幕密度  
adb shell wm density  
输出示例：  
Physical density: 480  
该设备屏幕密度为 480dpi。  
--显示屏参数  
adb shell dumpsys window displays  
输出示例：  
WINDOW MANAGER DISPLAY CONTENTS (dumpsys window displays)  
Display: mDisplayId=0  
init=1080x1920 420dpi cur=1080x1920 app=1080x1794 rng=1080x1017-1810x1731  
deferred=false layoutNeeded=false  
其中 mDisplayId 为 显示屏编号，init 是初始分辨率和屏幕密度，app 的高度比 init 里的要小，表示屏幕底部有虚拟按键，高度为 1920 - 1794 = 126px 合 42dp。  
--android\_id  
adb shell settings get secure android\_id  
输出示例：  
51b6be48bac8c569  
---IMEI  
在 Android 4.4 及以下版本可通过如下命令获取 IMEI：  
adb shell dumpsys iphonesubinfo  
输出示例：  
Phone Subscriber Info:  
Phone Type = GSM  
Device ID = 860955027785041  
其中的 Device ID 就是 IMEI。

而在 Android 5.0 及以上版本里这个命令输出为空，得通过其它方式获取了（需要 root 权限）：  
adb shell  
su  
service call iphonesubinfo 1  
输出示例：  
Result: Parcel(  
0x00000000: 00000000 0000000f 00360038 00390030 '........8.6.0.9.'  
0x00000010: 00350035 00320030 00370037 00350038 '5.5.0.2.7.7.8.5.'  
0x00000020: 00340030 00000031 '0.4.1... ')  
把里面的有效内容提取出来就是 IMEI 了，比如这里的是 860955027785041。

20.Android 系统版本  
adb shell getprop ro.build.version.release  
输出示例：  
5.0.2

21.Mac 地址  
adb shell cat /sys/class/net/wlan0/address  
输出示例：  
f8:a9:d0:17:42:4d

22.CPU 信息  
adb shell cat /proc/cpuinfo  
输出示例：  
Processor : ARMv7 Processor rev 0 (v7l)  
processor : 0  
BogoMIPS : 38.40

processor : 1  
BogoMIPS : 38.40

processor : 2  
BogoMIPS : 38.40

processor : 3  
BogoMIPS : 38.40

Features : swp half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt  
CPU implementer : 0x51  
CPU architecture: 7  
CPU variant : 0x2  
CPU part : 0x06f  
CPU revision : 0

Hardware : Qualcomm MSM 8974 HAMMERHEAD (Flattened Device Tree)  
Revision : 000b  
Serial : 0000000000000000  
这是 Nexus 5 的 CPU 信息，我们从输出里可以看到使用的硬件是 Qualcomm MSM 8974，processor 的编号是 0 到 3，所以它是四核的，采用的架构是 ARMv7 Processor rev 0 (v71)。

23.更多硬件与系统属性  
adb shell cat /system/build.prop  
这会输出很多信息，包括前面几个小节提到的「型号」和「Android 系统版本」等。

输出里还包括一些其它有用的信息，它们也可通过 adb shell getprop <属性名> 命令单独查看，列举一部分属性如下：  
属性名 | 含义  
ro.build.version.sdk | SDK 版本  
ro.build.version.release | Android 系统版本  
ro.build.version.security\_patch | Android 安全补丁程序级别  
ro.product.model | 型号  
ro.product.brand | 品牌  
ro.product.name | 设备名  
ro.product.board | 处理器型号  
ro.product.cpu.abilist | CPU 支持的 abi 列表  
persist.sys.isUsbOtgEnabled | 是否支持 OTG  
dalvik.vm.heapsize | 每个应用程序的内存上限  
ro.sf.lcd\_density | 屏幕密度

实用功能  
24.屏幕截图  
adb shell screencap -p /sdcard/sc.png  
然后将 png 文件导出到电脑：  
adb pull /sdcard/sc.png  
可以使用 adb shell screencap -h 查看 screencap 命令的帮助信息，下面是两个有意义的参数及含义：  
参数 | 含义  
-p | 指定保存文件为 png 格式  
-d display-id | 指定截图的显示屏编号（有多显示屏的情况下）  
实测如果指定文件名以 .png 结尾时可以省略 -p 参数；否则需要使用 -p 参数。如果不指定文件名，截图文件的内容将直接输出到 stdout。

25.录制屏幕  
录制屏幕以 mp4 格式保存到 /sdcard：  
adb shell screenrecord /sdcard/filename.mp4  
需要停止时按 Ctrl-C，默认录制时间和最长录制时间都是 180 秒。  
如果需要导出到电脑：  
adb pull /sdcard/filename.mp4  
可以使用 adb shell screenrecord --help 查看 screenrecord 命令的帮助信息，下面是常见参数及含义：  
参数 | 含义  
--size WIDTHxHEIGHT | 视频的尺寸，比如 1280x720，默认是屏幕分辨率。  
--bit-rate RATE | 视频的比特率，默认是 4Mbps。  
--time-limit TIME | 录制时长，单位秒。  
--verbose | 输出更多信息。