Exploring Android Command Line Tools

Bladymir Tellez
www.OCAndroid.org
August 26, 2020

Agenda

1. 3 Primary Development Use Cases
    - Building an APK
    - Interacting with the Android OS
    - Creating Virtual Devices
2. Memorable Mentions
3. Project Case Study: Mirror/Mirror-Knock Off
4. Questions

“Pre-Requesites”

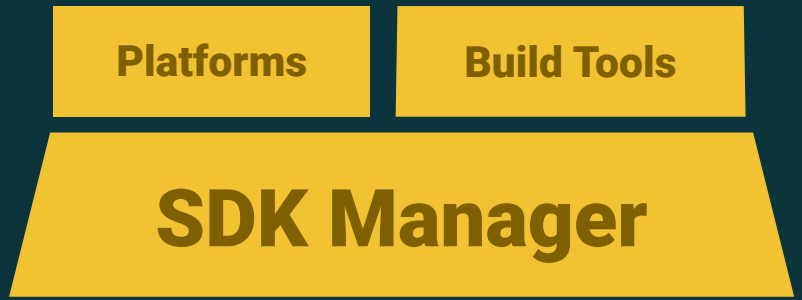
- Familiarity with configuring your preferred shell. ie: bash, zsh, csh, fish…
- Familiar with navigating a shell session.
- Downloaded Android SDK and/or Android Studio
- Familiarity with Android Development Process

Use Case 1:

Building an APK

Building an APK: SDK Manager

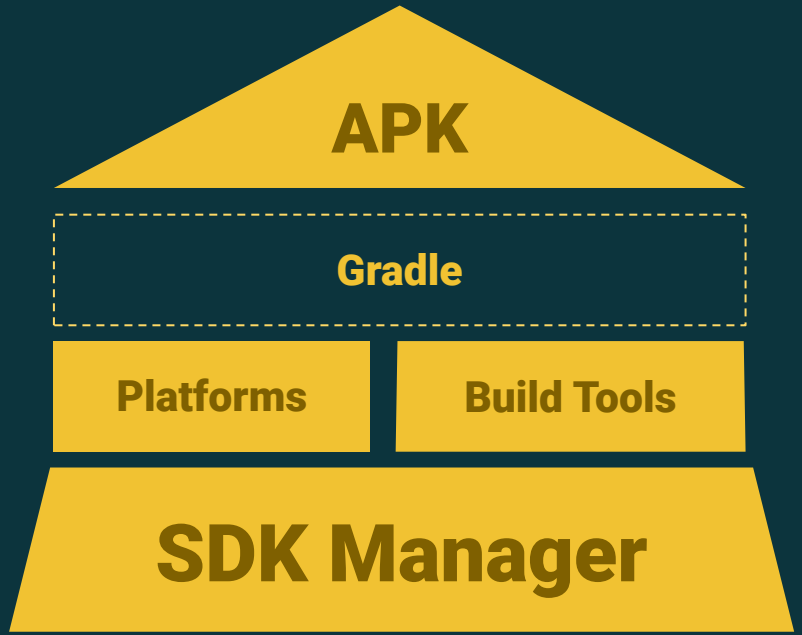
Building an APK: SDK Manager



Platforms -> API Versions

Build Tools -> Package Binaries

Building an APK: SDK Manager



Environment Variables

SDKManager: Environment

```
~/project $ echo $ANDROID_HOME
/Users/xyz/Library/Android/sdk

~/project $ cat <<EOF >> ~/.profile
export ANDROID_HOME=$HOME/Library/Android/sdk
export ANDROID_SDK_ROOT=$ANDROID_HOME                    (Needed by Gradle)
export PATH=$PATH:$ANDROID_HOME/emulator                 (emulator first!)
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/tools/bin
export PATH=$PATH:$ANDROID_HOME/platform-tools
EOF
```

https://developer.android.com/studio/command-line/variables

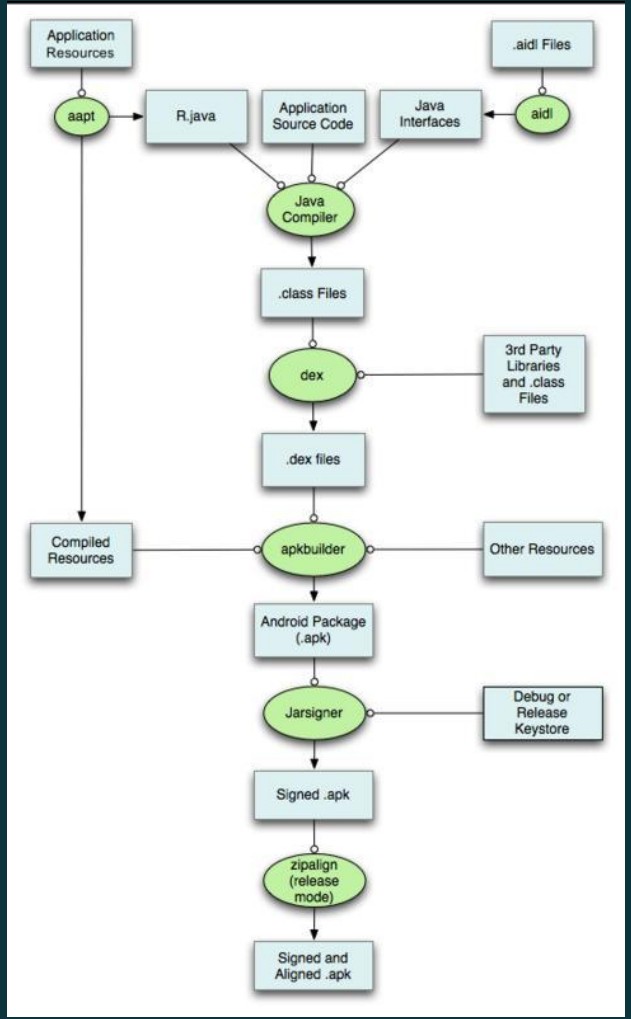
SDK Manager

SDKManager: List, Install, Uninstall

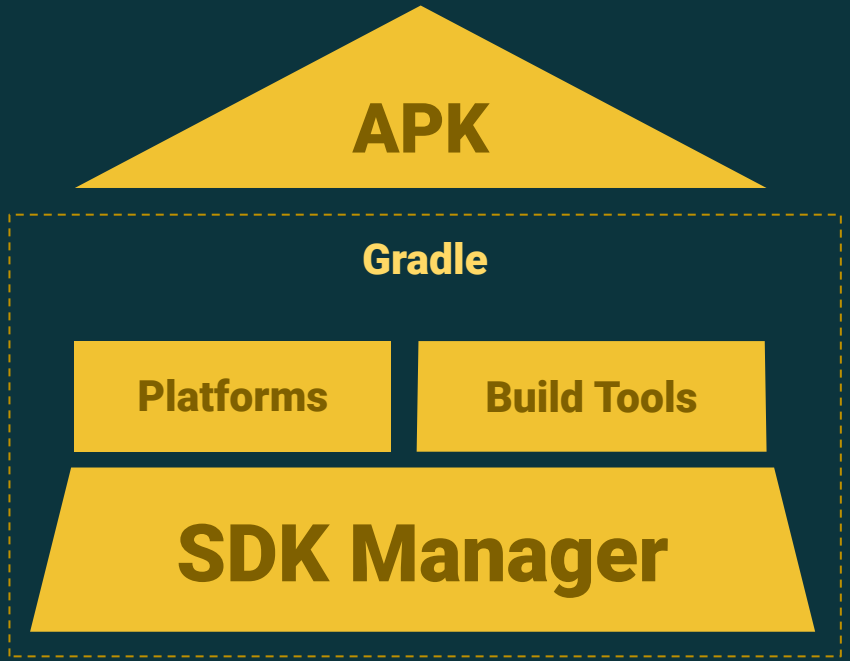
```
[1] ~/project $ sdkmanager

[2] ~/project $ sdkmanager --list

[3] ~/project $ sdkmanager --list --list-obsolete

[4] ~/project $ sdkmanager --install 'platforms:android-29'

[5] ~/project $ sdkmanager --install 'build-tools:30.0.0'

[6] ~/project $ sdkmanager --uninstall 'platforms:android-25'
```

SDK Home: Platforms & Build Tools

```
[1] ~/project $ ls $ANDROID_HOME/platforms/{version}

[2] ~/project $ ls $ANDROID_HOME/build-tools/{version}
-rwxrwxr-x. 1 blad blad 1.5M Aug 16 16:24 aapt
-rwxrwxr-x. 1 blad blad 5.7M Aug 16 16:24 aapt2
-rwxrwxr-x. 1 blad blad 3.5M Aug 16 16:24 aidl
-rwxrwxr-x. 1 blad blad 2.6K Aug 16 16:24 apksigner
-rwxrwxr-x. 1 blad blad 2.6K Aug 16 16:24 d8
-rwxrwxr-x. 1 blad blad 4.0M Aug 16 16:24 dexdump
-rwxrwxr-x. 1 blad blad 2.6K Aug 16 16:24 dx
-rwxrwxr-x. 1 blad blad 234K Aug 16 16:24 zipalign
```


Application Resources
.aidl Files
aapt
R.java
Application Source Code
Java Interfaces
aidl
Java Compiler
.class Files
dex
3rd Party Libraries and .class Files
.dex files
Compiled Resources
apkbuilder
Other Resources
Android Package (.apk)
Jarsigner
Debug or Release Keystore
Signed .apk
zipalign (release mode)
Signed and Aligned .apk

Building an APK: SDK Manager



Compiling an APK: Gradle to the Rescue

```
[1] ~/project $ ./gradlew assemble{variant}
```

https://developer.android.com/studio/build/building-cmdline

Use Case 2:

Interacting with the Android OS

Interacting with the OS: SDK Manager

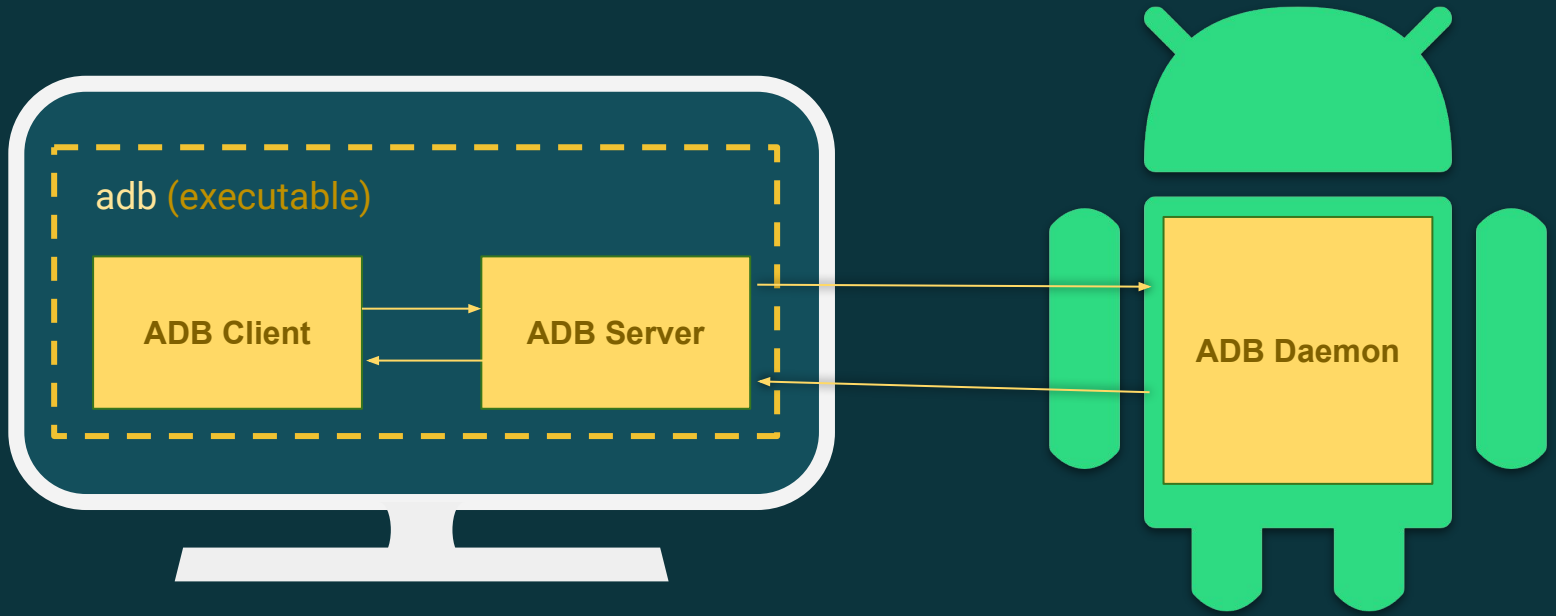
Interacting with the OS: Platform Tools

Interacting with the OS: ADB

SDKManager: Platform Tools

```
[1] ~/project $ sdkmanager --install 'platforms-tools'

[2] ~/project $ ls $ANDROID_HOME/platform-tools
-rwxrwxr-x. 1 - - 7.0M Aug 16 10:34 adb
-rwxrwxr-x. 1 - -  59K Aug 16 10:34 dmtracedump
-rwxrwxr-x. 1 - - 1.6M Aug 16 10:34 e2fsdroid
-rwxrwxr-x. 1 - - 303K Aug 16 10:34 etc1tool
-rwxrwxr-x. 1 - - 1.9M Aug 16 10:34 fastboot
-rwxrwxr-x. 1 - - 249K Aug 16 10:34 make_f2fs
-rwxrwxr-x. 1 - - 852K Aug 16 10:34 mke2fs
```

ADB: Android Debug Bridge


adb (executable)
ADB Client
ADB Server
ADB Daemon

Platform Tools: ADB

```
[1] ~/project $ adb
* daemon not running; starting now at tcp:5037
* daemon started successfully
...

[2] ~/project $ adb devices
List of devices attached

0KAYG00GLE      unauthorized
1KAYG00GLE      device
2KAYG00GLE      offline
emulator-3      device
```

Platform Tools: ADB

```
[1] ~/project $ adb -s 1KAYGOOGLE ...

[2] ~/project $ adb -d ...

[3] ~/project $ adb install my-app.apk          ./gradlew installDebug

[4] ~/project $ adb uninstall <package>         ./gradlew uninstallDebug

[5] ~/project $ adb push my-file /sdcard/my-file

[6] ~/project $ adb pull /sdcard/my-file my-file

[7] ~/project $ adb shell
```

adb shell: The Portal into Your Android

It’s not bash/zsh...it’s just **sh**

It’s just Linux...mostly.

It’s not the exact tools you GNU.

- Toolbox (Older)
- Toybox (newer Android Devices)

ADB Shell: From the Outside

```
[1] ~/project $ adb shell [args]

[2] ~/project $ adb shell ls /system/bin

[3] ~/project $ adb shell ls -hl /system/bin | grep 'toolbox'
lrwxr-xr-x 1 root shell 7    2008-12-31 18:00 getevent -> toolbox
lrwxr-xr-x 1 root shell 7    2008-12-31 18:00 getprop -> toolbox
lrwxr-xr-x 1 root shell 7    2008-12-31 18:00 modprobe -> toolbox
lrwxr-xr-x 1 root shell 7    2008-12-31 18:00 setprop -> toolbox
lrwxr-xr-x 1 root shell 7    2008-12-31 18:00 start -> toolbox
lrwxr-xr-x 1 root shell 7    2008-12-31 18:00 stop -> toolbox
-rwxr-xr-x 1 root shell 115K 2008-12-31 18:00 toolbox
```

ADB Shell: From the Outside

```
[1] ~/project $ adb shell getevent > events.txt

[2] ~/project $ adb shell sendevent /device/input/event0 100 1 2

[3] ~/project $ adb shell am | less

[4] ~/project $ adb shell pm | less
```

ADB Shell: From the Inside

```
[1] ~/project $ adb shell
[2] shell:/ $
[3] shell:/ $ ls /system/bin
[4] shell:/ $ whoami
shell
[5] shell:/ $ am | more
[6] shell:/ $ pm | more
[7] shell:/ $ getevent -t > /sdcard/events.txt
[8] shell:/ $ sendevent /dev/input/event0 100 1 2
[9] shell:/ $ exit

[1] ~/project $
```

ADB: Activity Manager: am

```
[1] shell:/ $ am
[2] shell:/ $ am start-activity <INTENT>
[3] shell:/ $ am start-service <INTENT>
[4] shell:/ $ am start-foreground-service <INTENT>
[5] shell:/ $ am stop-service <INTENT>
[6] shell:/ $ am kill <PACKAGE>
[7] shell:/ $ am crash <PACKAGE>
[8] shell:/ $ am profile …

...and lots more!
```

ADB: Package Manager: pm

```
[1] shell:/ $ pm
[1] shell:/ $ pm list packages
[1] shell:/ $ pm list permissions
[1] shell:/ $ pm list permission-groups
[1] shell:/ $ pm dump <PACKAGE>

[1] shell:/ $ pm query-{activities|services|receivers} <INTENT>

[1] shell:/ $ pm install <PATH>                adb install ...
[1] shell:/ $ pm uninstall <PACKAGE>           adb uninstall ...

[1] shell:/ $ pm grant <PACKAGE> <PERMISSION>
[1] shell:/ $ pm revoke <PACKAGE> <PERMISSION>
[1] shell:/ $ pm reset-permissions -p <PACKAGE>
```

ADB: IO; getevent, sendevent, input

```
[1] shell:/ $ getevent -t
[  102733.050250] /dev/input/event2: 0000 0000 00000000
[  102733.058841] /dev/input/event2: 0003 0035 0000040e
[  102733.058841] /dev/input/event2: 0003 0036 000003dd

[2] shell:/ $ sendevent /dev/input/event2 3 23 1038


[3] shell:/ $ input touchscreen text "pizza near me"
```

ADB: Utils

```
[1] shell:/ $ screenrecord screen.mp4

[2] shell:/ $ screencap -p screen.png (can be piped from the “outside”)
[2] ~project/ $ adb shell screencap -p > screen.png

[3] shell:/ $ logcat                    adb logcat
```

abb acpi am apexd app_process app_process32 app_process64 appops appwidget atrace audioserver auditctl **awk** **base64** basename bc bcc blank_screen blkid blockdev bmgr bootanimation bootstat bootstrap boringssl_self_test32 boringssl_self_test64 bpfloader bu bugreport bugreportz bunzip2 bzcat **bzip2** cal cameraserver cat charger chattr chcon chgrp chmod chown chroot chrt **cksum** clatd clear cmd **cmp** comm content **cp** cpio cppreopts.sh crash_dump32 crash_dump64 credstore cut dalvikvm date **dd** debuggerd defrag.f2fs device_config devmem dex2oat df diff dirname disable-verity dmctl dmesg dnsmasq dos2unix dpm drmserver du dump.f2fs dumpstate dumpsys e2freefrag e2fsck e2fsdroid **echo** egrep enable-verity **env** expand expr fallocate false fgrep **file** find flags_health_check flock fmt free fsck.f2fs fsck_msdos fsverity_init fsync gatekeeperd getconf getenforce getevent getprop gpuservice grep groups gsi_tool gsid gunzip gzip **head** heapprofd hid hostname hw hwclock hwservicemanager i2cdetect i2cdump i2cget i2cset icm iconv id idmap2 idmap2d ifconfig ime incident incident-helper-cmd incident_helper incidentd init inotifyd input insmod install installd ionice iorap.cmd.compiler iorap.cmd.maintenance iorap.inode2filename iorap.prefetcherd iorapd iorenice ip ip-wrapper-1.0 ip6tables ip6tables-restore ip6tables-save ip6tables-wrapper-1.0 iptables iptables-restore iptables-save iptables-wrapper-1.0 **keystore** keystore_cli_v2 kill killall ld.mc ldd librank linker linker64 linker_asan linker_asan64 linkerconfig llkd lmkd ln load_policy locksettings log logcat logd logname logwrapper losetup lpdump lpdumpd ls lsattr lshal lsmod lsof lspci lsusb make_f2fs md5sum mdnsd mediaextractor mediametrics mediaserver microcom migrate_legacy_obb_data.sh mini-keyctl mkdir mke2fs mkfifo mkfs.ext2 mkfs.ext3 mkfs.ext4 mknod mkswap **mktemp** modinfo modprobe monkey more mount mountpoint move_time_data.sh mtpd mv nc ndc ndc-wrapper-1.0 netcat netd netstat netutils-wrapper-1.0 newfs_msdos nice nl nohup notify_traceur.sh nproc nsenter od otapreopt otapreopt_chroot otapreopt_script otapreopt_slot paste patch perfetto pgrep pidof **ping** ping6 pkill pktlogconf pm pmap pppd preloads_copy.sh preopt2cachename printenv **printf** procrank **ps** pwd racoon readelf readlink realpath reboot recovery-persist recovery-refresh renice requestsync resize.f2fs resize2fs restorecon rm rmdir rmmod rss_hwm_reset run-as runcon schedtest screencap screenrecord sdcard secdiscard secilc sed sendevent sensorservice seq service servicemanager set-verity-state setenforce setprop setsid settings sgdisk sh **sha1sum** sha224sum sha256sum sha384sum sha512sum showmap simpleperf simpleperf_app_runner **sleep** sload_f2fs sm snapshotctl sort split ss start stat stop storaged strings stty surfaceflinger svc swapoff swapon sync sysctl tac tail **tar** taskset tc tc-wrapper-1.0 tcpdump tee telecom test time timeout tombstoned **toolbox** **top** touch **toybox** tr traced traced_perf traced_probes trigger_perfetto true truncate tty tune2fs tzdatacheck ueventd uiautomator ulimit umount uname uncrypt uniq unix2dos unlink unshare **unzip** update_engine update_verifier uptime usbd usleep uudecode uuencode uuidgen vdc vendor_cmd_tool viewcompiler vmstat vold vold_prepare_subdirs vr wait_for_keymaster watch watchdogd **wc** which whoami wificond wm xargs **xxd** yes zcat zipinfo ziptool

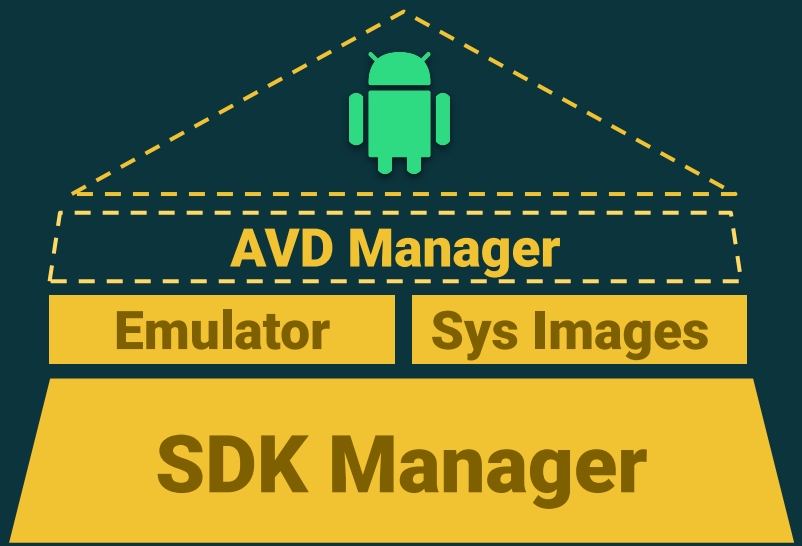
387 commands

Use Case 3:

Creating Virtual Devices

Creating Virtual Devices: SDK Manager

Creating Virtual Devices: SDK Manager



SDKManager: Emulator and System Images

```
[1] ~/project $ sdkmanager --install 'emulator'

[2] ~/project $ ls $ANDROID_HOME/emulator

[3] ~/project $ sdkmanager --install 'system-images;android-29;default;x86'

[4] ~/project $ ls $ANDROID_HOME/system-images/...
```

AVD Manager

```
[1] ~/project $ avdmanager

[2] ~/project $ avdmanager list avd

[3] ~/project $ avdmanager list device

[4] ~/project $ avdmanager list target

[5] ~/project $ avdmanager list
```

AVD Manager: Create an Virtual Device

```
[1] ~/project $ avdmanager create avd \
--device 'Nexus 5X' \
--package 'system-images;android-25;google_apis;x86 \
--name 'vNexus5_29'

[2] ~/project $ avdmanager list avd
Available Android Virtual Devices:
      Name: vNexus5-29
    Device: Nexus 5X (Google)
      Path: /home/yxz/.android/avd/v10-29.avd
    Target: Default Android System Image
      Based on: Android 10.0 (Q) Tag/ABI: default/x86
    Sdcard: 512 MB
```

Emulator: Up and Running

```
[1] ~/project $ emulator

[2] ~/project $ emulator -list-avds

[3] ~/project $ emulator -avd vNexus5_29

[4] ~/project $ emulator -avd vNexus5_29 -sd-card sd.img -no-window ...
```

Memorable Mentions

Memorable Mentions

```
[1] ~/project $ apkanalyzer                          (sdkmanager)
```

https://developer.android.com/studio/build/apk-analyzer

```
[2] ~/project $ adb shell dumpsys -l                 (platform-tools)
[3] ~/project $ adb shell dumpsys activity {activities|services|...}
```

https://developer.android.com/studio/command-line/dumpsys

```
[4] ~/project $ mksdcard                             (emulator)
```

https://developer.android.com/studio/command-line/mksdcard

Project Case Study

adb-event-mirror

...

https://github.com/JakeWharton/adb-event-mirror

android-event-playback

...

https://github.com/blad/android-event-playback

Thank You!

...

Blad Tellez
Twitter: @bctellez
GitHub: @blad

Questions?

...

Blad Tellez
Twitter: @bctellez
GitHub: @blad


Misc.

Additional Links

ADB Cheat Sheet:
https://www.automatetheplanet.com/adb-cheat-sheet/

ADB Source:
https://android.googlesource.com/platform/system/core/+/master/adb/
https://android.googlesource.com/platform/system/core/+/master/adb/OVERVIEW.TXT
https://android.googlesource.com/platform/system/core/+/master/adb/SERVICES.TXT

Can I use my Android device as a computer?

Short: No.

Medium: Linux system is heavily restricted. You'd need an ADB client to drop into a shell.

Alternative answer: Use Termux! https://termux.com/

“No Permissions on Linux”

Add `udev` rules for the device.

https://www.janosgyerik.com/adding-udev-rules-for-usb-debugging-android-devices/

Installing a bare SDK

Solution: Move: “tools/” into “cmdline-tools/tools/”

https://stackoverflow.com/questions/60440509/android-command-line-tools-sdkmanager-always-shows-warning-could-not-create-se

Issues:
Warning: Could not create settings
java.lang.IllegalArgumentException
at
com.android.sdklib.tool.sdkmanager.SdkManagerCliSettings.<init>(SdkManagerCliSettings.java:428)