

2017 Android of WMNTAA

Android Debug Bridge

ADB

- Introduction
- Setting up
- Install an App
- Copy files to/from a device
- Issue shell commands
- Activity manager (am)
- Package Manager (pm)
- Take a screenshot
- Record a video

Introduction

Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device (an emulator or a connected Android device). The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device.

Introduction

It is a client-server program that includes three components:

- A client, which sends commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an `adb` command.
- A daemon (`adb`), which runs commands on a device. The daemon runs as a background process on each device.
- A server, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

Set up

- Link : <https://developer.android.com/studio/releases/platform-tools.html>
- Decompress the downloaded file
- Switch to decompressed folder in command line interface

```
→ platform-tools ll
total 10896
drwx-----@ 13 eric  staff    442 Mar 16 15:42 .
drwx-----+ 57 eric  staff   1938 Apr 13 11:51 ..
-rw-r-----@ 1 eric  staff  253025 Mar 13 17:11 NOTICE.txt
-rwxr-x---@ 1 eric  staff 2326236 Mar 13 17:10 adb
drwxr-x---@ 4 eric  staff   136 Mar 16 15:40 api
-rwxr-x---@ 1 eric  staff  60504 Mar 13 17:10 dmtracedump
-rwxr-x---@ 1 eric  staff  238028 Mar 13 17:10 etc1tool
-rwxr-x---@ 1 eric  staff 1429540 Mar 13 17:10 fastboot
-rwxr-x---@ 1 eric  staff  17536 Mar 13 17:10 hprof-conv
drwxr-x---@ 3 eric  staff   102 Mar 16 15:42 lib
-rw-r-----@ 1 eric  staff    63 Mar 16 14:23 source.properties
-rwxr-x---@ 1 eric  staff 1233284 Mar 13 17:10 sqlite3
drwxr-x---@ 6 eric  staff   204 Mar 16 15:42 systrace
```

Set up

- List devices : `adb devices`
 - `$ adb devices`
List of devices attached
`emulator-5556 device product:sdk_google_phone_x86_64`
`model:Android_SDK_built_for_x86_64 device:generic_x86_64`
`emulator-5554 device product:sdk_google_phone_x86`
`model:Android_SDK_built_for_x86 device:generic_x86`
`0a388e93 device usb:1-1 product:razor model:Nexus_7 device:flo`
- Send commands to a specific device : `adb -s "Serial number" "cmd"`

Checkout

- Use command :

```
adb -s "your device serial number" shell cat /proc/cpuinfo
```

- Output :

```
→ platform-tools adb -s E8ATBC015221 shell cat /proc/cpuinfo
Processor      : ARMv7 Processor rev 1 (v7l)
processor      : 0
BogoMIPS       : 38.40
```

Install an App

- Generate a debug apk : Android Studio → Build → Build Apk
 - The generated apk should be at */path_to_your_project/app/build/outputs/apk*
- Install an apk : `adb install path_to_apk`

```
→ platform-tools adb install app-debug.apk
app-debug.apk: 1 file pushed. 5.8 MB/s (10387754 bytes in 1.713s)
    pkg: /data/local/tmp/app-debug.apk
Success
```

Copy files to/from a device

- Copy from devices : `adb pull remote_path local_path`
- Copy to devices : `adb push local_path remote_path`

```
→ platform-tools adb push file /sdcard/  
file: 1 file pushed.  
→ platform-tools adb shell ls /sdcard/  
ASUS  
Alarms  
Android  
Bluetooth spp pro  
DCIM  
Download  
DownloadAndUploadCache  
Movies  
Music  
Notifications  
Pictures  
Podcasts  
Ringtones  
asus_log  
file  
workspace
```

Issue shell commands

- Command without entering a remote shell :

```
adb [-d | -e | -s serial_number] shell shell_command
```

- Enter a remote shell on a device :

```
adb [-d | -e | -s serial_number] shell
```

Checkout

- Enter a shell on your device, echo \$PATH, and check what commands are native on your device.

```
shell@ASUS_T00N:/ $ echo $PATH
/sbin:/vendor/bin:/system/sbin:/system/bin:/system/xbin:/bin
shell@ASUS_T00N:/ $ ls /system/bin
ATFWD-daemon
DxHDCP_Provisioning
GammaSetting
HSVSetting
PktRspTest
```

Activity manager (am)

- Shell command : `(adb shell) am command`
- Start an Activity with Intent :

```
am start -a intent(ex: android.intent.action.MAIN)
```

- Start an Activity with activity name :

```
am start -n package_name/package_name.activity_name
```

```
(ex: binaron.com.helloworld/binaron.com.helloworld.MainActivity)
```

Activity manager (am)

- Broadcast :
 - Broadcast lower battery : `am broadcast -a Android.intent.action.BATTERY_LOW --ei "level" 3 --ei "scale" 100`
- List Activity stack :
 - `dumpsys activity activities (| grep -i your.app.package.name)`

Checkout

- See how the activity stack changes by switching between activities.

Package Manager (pm)

- Shell command : `pm command`
- Common commands :
 - `list packages [options] filter`
 - `list features`
 - `list libraries`
 - `install [options] path`
 - `uninstall [options] package`
 - `clear package`

Checkout

- Push an apk file to your device and install it with package manager (pm)

Take a screenshot

- Shell command : `screencap filename`
 - `adb shell screencap /sdcard/screen.png`

Record a video

- Shell command : `screenrecord [options] filename`
 - `adb shell screenrecord --verbose /sdcard/demo.mp4`
- Options :
 - `--help` :
 - Displays command syntax and options
 - `--size widthxheight` :
 - Sets the video size: 1280x720. The default value is the device's native display resolution (if supported), 1280x720 if not. For best results, use a size supported by your device's Advanced Video Coding (AVC) encoder.

Record a video

- Options :
 - `--bit-rate rate:`
 - The default value is 4Mbps. You can increase the bit rate to improve video quality, but doing so results in larger movie files.
 - `--time-limit time :`
 - Sets the maximum recording time, in seconds. The default and maximum value is 180 (3 minutes).
 - `--verbose:`
 - Displays log information on the command-line screen.

Checkout

- Try to record a video by yourself !

Reference

- [Android Debug Bridge](#)