Android Debug Bridge (adb)

1. Definition

A. The Android Debug Bridge (adb) is a development tool that facilitates communication between an Android device and a personal computer.

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

1. Client
2. Server
3. Daemon

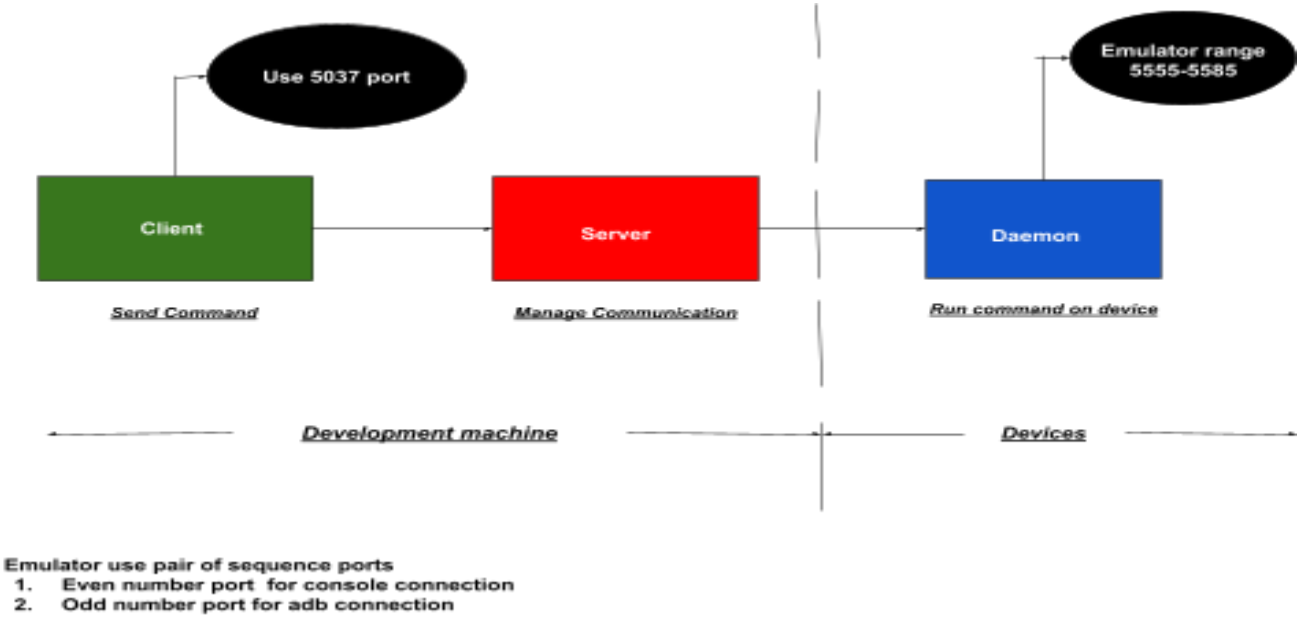


Image:-Android debug bridge components

2. Installing adb

1. Microsoft Windows ADB Setup

1. Download the Platform Tools ZIP file for Windows
2. Extract the contents of this ZIP file into an easily accessible folder (such as C:\adb)
3. Open Windows Explorer and browse to where you extracted the contents of this ZIP file
4. Then open up a Command Prompt from the same directory as this ADB binary. This can be done by holding *Shift* and *Right-clicking* within the folder then click the *"Open Command Prompt here"* option. (Some Windows 10 users may see "PowerShell" instead of "Command Prompt".)
5. In the Command Prompt window, enter the following command to launch the ADB daemon: *adb devices*. You can see your device's serial number in the command prompt. You can now run any ADB command on your device! Now go forth and start modding your phone by following our extensive list of tutorials!

2. How to Install ADB on Linux

1. Download the Platform Tools ZIP file for Linux
2. Extract the ZIP to an easily accessible location (like the Desktop for example).
3. Open a Terminal window.
4. Enter the following command: cd /path/to/extracted/folder/
5. This will change the directory to where you extracted the ADB files.
6. So for example:cd /Users/Doug/Desktop/platform-tools/
7. Once the Terminal is in the same folder your ADB tools are in, you can execute the following command to launch the ADB daemon: *adb devices*
8. You should now see your device's serial number in the Terminal window output. Congrats! You can now run any ADB command on your device! Now go forth and start modding your phone by following our extensive list of tutorials!

Some Linux users should be aware that there can be an easier way to install ADB on their computers. The guide above will certainly work for you, but those who own a Debian or Fedora/SUSE-based distro of Linux can skip steps 1 and 2 of the guide above and use one of the following commands:

- Debian-based Linux users can type the following command to install ADB: *sudo apt-get install adb*
- Fedora/SUSE-based Linux users can type the following command to install ADB: *sudo yum install android-tools*

However, it is always better to opt for the latest binary from the Android SDK Platform Tools release, since the distro-specific packages often contain outdated builds.

3. ADB Commands

Flags

A. -d (directs command to the only connected USB device)
B. -e (directs command to the only running emulator)
C. -s (serial number)
D. -p (product name or path)

The flag you decide to use has to come before the actual adb command

ADB Basic Commands

1. adb (give options and detail about adb command)
1. adb devices (lists connected devices)
2. adb root (restarts adb with root permissions)
3. adb start-server (starts the adb server)
4. adb kill-server (kills the adb server)
5. adb reboot (reboots the device)
6. adb devices -l (list of devices by product/model)
7. adb shell (starts the background(will enter device) terminal)
8. exit (exits the background terminal)
9. adb help (list all commands)
10. adb -s <deviceName> <command> (redirect command to specific device)

    ex:-adb -s emulator-5554 shell ls

11. adb –d <command> (directs command to only attached USB device)

    ex:-adb -d shell ls

12. adb –e <command> (directs command to only attached emulator)

    ex:-adb -e shell ls

Package Installation

1. adb install <apk> (install app)

```
adb install C:\Users\kakh272961\Downloads\example.apk
```

2. adb install <path> (install app from phone path)

```
adb install example.apk
```

3. adb install -r <path> (-r: Reinstall an existing app, keeping its data,install app from phone path)

```
adb install -r C:\Users\kakh272961\Downloads\example.apk
```

4. adb uninstall <package_name> (remove the app,package name you can find in manifest file of app)

```
adb uninstall com.example.adb_test1
```

Paths

1. /data/data/<package>/databases (app databases)

2. /data/data/<package>/shared_prefs/ (shared preferences)

```
generic_x86_arm:/data/data/com.google.android.apps
.maps/shared_prefs # ls
FirebaseAppHeartBeat.xml
_has_set_default_values.xml
settings_preference.xml
ThirdPartyAppsClicksPreference.xml
com.google.android.gms.appid.xml
ThirdPartyAppsRecencyPreference.xml
com.google.android.gms.measurement.prefs.xml
```

3. /data/app (apk installed by user)

```
generic_x86_arm:/data/app # ls
~~SG68XkJ9ad1i7E4IFMUZkw==
~~YkMq5E2w1-xXUsSO0hIFyQ==
```

4. /system/app (pre-installed APK files)

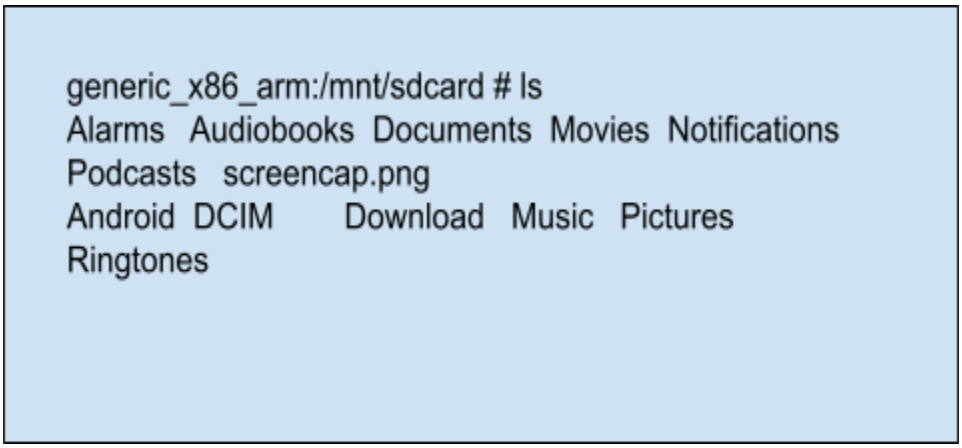
```
generic_x86_arm:/system/app # ls
BasicDreams             CompanionDeviceManager
NfcNci                  SecureElement
Bluetooth               EasterEgg
PacProcessor            SimAppDialog
BluetoothMidiService    GoogleExtShared
PartnerBookmarksProvider  Stk
BookmarkProvider
GooglePrintRecommendationService  PresencePolling
Traceur
CaptivePortalLoginGoogle  HTMLViewer
PrintSpooler            WallpaperBackup
CarrierDefaultApp       KeyChain
RcsService
CertInstaller           LiveWallpapersPicker
SafetyRegulatoryInfo
```

5. /mnt/asec (encrypted apps) (App2SD)(mnt:**mount** points for different devices)

```
generic_x86_arm:/mnt/asec # ls
generic_x86_arm:/mnt/asec #
```

6. /mnt/sdcard (internal SD Card)

7. /mnt/sdcard (external/Internal SD Card)



8. /mnt/sdcard/external_sd (external SD Card)
9. adb shell ls (list directory contents)
10. adb shell ls -s (print size of each file)
11. adb shell ls -R (list subdirectories recursively)

File Operations

1. adb push <local> <remote> (copy file/dir to device)

   adb push C:\Users\kakh272961\Pictures\Untitled9.png sdcard

2. adb pull <remote> <local> (copy file/dir from device)

```
D:\software\platform-tools>adb pull sdcard/screencap.png C:\Users\kakh272961\Pictures

sdcard/screencap.png: 1 file pulled, 0 skipped. 63.4 MB/s (579274 bytes in 0.009s)
```

3. run-as <package> cat <file> (access the private package files)

Phone Info

1. adb get-state (print device state)

   ```
   D:\software\platform-tools>adb get-state

   device
   ```

2. adb get-serialno (get the serial number)

   ```
   D:\software\platform-tools>adb get-serialno

   emulator-5554
   ```

3. adb shell dumpsys iphonesubinfo (get the IMEI:International Mobile Equipment Identity)
4. adb shell netstat (list TCP connectivity)
5. adb shell pwd (print current working directory)
6. adb shell dumpsys battery (battery status)
7. adb shell pm list features (list phone features)
8. adb shell service list (list all services)
9. adb shell dumpsys activity <package>/<activity> (activity info)
10. adb shell ps (print process status)
11. adb shell wm size (displays the current screen resolution)
12. adb shell dumpsys window windows | grep -E 'mCurrentFocus|mFocusedApp' (print current app's opened activity)

Package Info

1. adb shell cmd package list packages OR adb shell pm list packages (list package names)
2. adb shell pm path com.android.bluetooth (list package name + path to apks)
3. adb shell pm list packages -3 (list third party package names)

```
adb shell pm list packages -3
```

```
package:at.markushi.reveal

package:com.example.adb_test1
```

4. adb shell pm list packages -s (list only system packages)
5. adb shell list packages -u (list package names + uninstalled)
6. adb shell dumpsys package packages (list info on all apps)
7. adb shell dumpsys package <name> (list info on one package)

```
adb shell dumpsys package com.example.adb_test1
```

8. adb shell pm path <package> (path to the apk file)

Configure Settings Commands

1. adb shell dumpsys battery set level <n> (change the level from 0 to 100)

```
adb shell dumpsys battery set level 100
```

2. adb shell dumpsys battery set status<n> (change the level to unknown, charging, discharging, not charging or full)

   adb shell dumpsys battery set status 1

3. adb shell dumpsys battery reset (reset the battery)
4. adb shell dumpsys battery set usb <n> (change the status of USB connection. ON or OFF)

   adb shell dumpsys battery set usb 1

5. adb shell wm size WxH (sets the resolution to WxH)

   adb shell wm size 100x100

6. adb shell wm (give options wm:window manager)

Device Related Commands

1. adb reboot-recovery (reboot device into recovery mode)
2. adb reboot fastboot (reboot device into recovery mode)
3. adb shell screencap -p "/path/to/screenshot.png" (capture screenshot)

4. adb shell screenrecord "/path/to/record.mp4" (record device screen)
5. adb backup -apk -all -f backup.ab (backup settings and apps)
6. adb backup -apk -shared -all -f backup.ab (backup settings, apps and shared storage)
7. adb backup -apk -nosystem -all -f backup.ab (backup only non-system apps)
8. adb restore backup.ab (restore a previous backup)
9. adb shell am start|startservice|broadcast <INTENT>[<COMPONENT>]
10. -a <ACTION> e.g. android.intent.action.VIEW
11. -c <CATEGORY> e.g. android.intent.category.LAUNCHER (start activity intent)
12. adb shell am start -a android.intent.action.VIEW -d URL (open URL)
13. adb shell am start -t image/* -a android.intent.action.VIEW (opens gallery)

Logs

1. adb logcat [options] [filter] [filter] (view device log)

   adb logcat > logcat.log

   https://www.xda-developers.com/how-to-take-logs-in-android/

2. adb bugreport (print bug reports)

Permissions

1. adb shell list permissions -g -r (list permissions details status by group)

```
adb shell pm list permissions -s
```

```
adb shell pm list permissions -g
```

4. Links used

1. https://wiki.lineageos.org/adb_fastboot_guide.html#popular-adb-commands
2. https://developer.android.com/studio/command-line/adb
3. https://www.xda-developers.com/what-is-adb/
4. https://www.automatetheplanet.com/adb-cheat-sheet/
5. https://developer.android.com/studio/command-line/adb
6. https://gist.github.com/Pulimet/5013acf2cd5b28e55036c82c91bd56d8
7. https://proandroiddev.com/helpful-adb-commands-d195f13e93dd