

Android Debug Bridge

Cheat Sheet

Selecting a device

<code>adb devices</code>	List of devices by serial number.
<code>adb devices -l</code>	List of devices by product/model.
<code>adb -s <serial> ...</code>	Command line selection.
<code>export ANDROID_SERIAL=<serial></code>	Env. variable selection.

If a command starts with `$` it has to be run from the Android shell or via `adb shell <command>`, or even better `adb shell <command> | less`.

Package installation

<code>adb install <apk></code>	Installs app.
<code>\$ pm install <path></code>	Install app from phone path.
<code>\$ pm install -r <path></code>	Reinstall app from phone path.
<code>\$ pm uninstall <name></code>	Remove the app.
<code>\$ pm get-install-location</code>	Install location: 0 - Auto 1 - Internal 2 - External

Package info

<code>\$ pm list packages</code>	List package names.
<code>\$ pm list packages -f</code>	As above + path to apks.
<code>\$ pm list packages -3</code>	Only third party packages.
<code>\$ pm list packages -s</code>	Only system packages.
<code>\$ pm list packages -u</code>	Also uninstalled packages.
<code>\$ dumpsys package packages</code>	List info on all apps.
<code>\$ pm dump <name></code>	List info on one package.
<code>\$ pm path <package></code>	Path to the apk file.

Permissions

<code>\$ pm permission groups</code>	Permission groups definitions.
<code>\$ pm list permissions -g -f</code>	List permissions details.

File operations

<code>adb push <local> <remote></code>	Copy file/dir to device.
<code>adb pull <remote> [<local>]</code>	Copy file/dir from device.
<code>adb backup -f <file></code>	Backup the phone.

If you want to access the private package files just use `run-as <package> cat <file>`.

Paths

<code>/data/data/<package></code>	App data, as described below.
<code> databases/</code>	App databases.
<code> shared_prefs/</code>	Shared preferences.
<code>/data/app</code>	APK files installed by user.
<code>/system/app</code>	Pre-installed APK files.
<code>/mnt/asec</code>	Encrypted apps (App2SD).
<code>/mnt/emmc</code>	Internal SD Card.
<code>/mnt/sdcard</code>	External/Internal SD Card.
<code>/mnt/sdcard/external_sd</code>	External SD Card.

Phone info

<code>\$ sqlite3 /data/data/com.android.providers.settings/databases/settings.db .dump</code>	Dump phone settings.
<code>\$ getprop</code>	Get properties (e.g. model).
<code>\$ dumpsys iphonesubinfo</code>	Get the IMEI.
<code>adb get-serialno</code>	Get the serial number.
<code>\$ dumpsys battery</code>	Battery status.
<code>\$ pm list users</code>	Lists phone users (4.1+).
<code>\$ pm list features</code>	List phone features.

Services & activities

<code>\$ service list</code>	List all services.
<code>\$ dumpsys activity <package>/<activity></code>	Dump activity info.

Logs

All logs are accessed by using either `$ logcat` or `adb logcat`.

Useful options are:

<code>-d</code>	Only dump logs (do not block).
<code>-c</code>	Flush the buffers.
<code>-b <buffer></code>	Buffer to display (default: system, main).
<code><tag>[:priority]</code>	Filter spec at the end of logcat.

Available priorities are:

V	Verbose
D	Debug
I	Info
W	Warn
E	Error
F	Fatal
S	Silent (suppress all output)

Miscellaneous

<code>\$ screencap -p <path>.png</code>	Screenshot (save to path on device).
<code>\$ screenrecord <path>.mp4</code>	Screen capture (path on device).

ADB daemon

adbd runs on TCP/5037.

<code>adb kill-server</code>	Kill the server if it is running.
<code>adb start-server</code>	Ensure that there is a server running.
<code>adb root</code>	Restarts the adbd with root permissions.

by @maldr0id

based on L^AT_EX cheat sheet by Winston Chang
<http://www.stdout.org/~winston/latex/>