NAME
adb − Android Debug Bridge

SYNOPSIS
adb [−d|−e|−s *serialNumber*] *command*

DESCRIPTION
WARNING: This manual might be outdated, please refer to the official documentation.

Android Debug Bridge (**adb**) is a versatile command line tool that lets you communicate with an emulator instance or connected Android−powered device. 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 shell by issuing an **adb** command. Other Android tools such as DDMS also create **adb** clients.

- **A daemon**, which runs commands on a device. The daemon runs as a background process on each emulator or device instance.

- **A server**, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

If there's only one emulator running or only one device connected, the **adb** command is sent to that device by default. If multiple emulators are running and/or multiple devices are attached, you need to use the **−d**, **−e**, or **−s** option to specify the target device to which the command should be directed.

OPTIONS
−a   Directs **adb** to listen on all interfaces for a connection.

−d   Directs command to the only connected USB device. Returns an error if more than one USB device is present.

−e   Directs command to the only running emulator. Returns an error if more than one emulator is running.

−s *specific device*
       Directs command to the device or emulator with the given serial number or qualifier. Overrides **ANDROID_SERIAL** environment variable.

−p *product name or path*
       Simple product name like **sooner**, or a relative/absolute path to a product out directory like **out/target/product/sooner**. If **−p** is not specified, the **ANDROID_PRODUCT_OUT** environment variable is used, which must be an absolute path.

−H   Name of adb server host (default: **localhost**)

−P   Port of adb server (default: **5037**)

COMMANDS
adb devices [−l]
       List all connected devices. **−l** will also list device qualifiers.

adb connect *host*[:*port*]
       Connect to a device via TCP/IP. Port **5555** is used by default if no port number is specified.

adb disconnect [*host*[:*port*]**]**
       Disconnect from a TCP/IP device. Port **5555** is used by default if no port number is specified. Using this command with no additional arguments will disconnect from all connected TCP/IP devices.

Device commands
adb push *local... remote*
       Copy file/dir to device.

adb pull [−a] *remote* [*local*]
    Copy file/dir from device. **−a** means copy timestamp and mode.

adb sync [−l] [*directory*]
    Copy host−>device only if changed. **−l** means list but don't copy.

If *directory* is not specified, **/system**, **/vendor** (if present), **/oem** (if present) and **/data** partitions will be up-dated.

If it is **system**, **vendor**, **oem** or **data**, only the corresponding partition is updated.

adb shell [−e *escape*] [−n] [−T|−t] [−x] [*command*]
    Run remote shell command (interactive shell if no command given)

- −e: Choose escape character, or **none**; default **~**

- −n: Don't read from stdin

- −T: Disable PTY allocation

- −t: Force PTY allocation

- −x: Disable remote exit codes and stdout/stderr separation

adb emu *command*
    Run emulator console command

adb logcat [*filter−spec***]**
    View device log.

adb forward −list
    List all forward socket connections. The format is a list of lines with the following format: ***serial*** **" "** ***local*** **" "** ***remote*** **""**

adb forward *local remote*
    Forward socket connections.

Forward specs are one of:

- tcp:*port*

- localabstract:*unix domain socket name*

- localreserved:*unix domain socket name*

- localfilesystem:*unix domain socket name*

- dev:*character device name*

- jdwp:*process pid* (remote only)

adb forward −no−rebind *local remote*
    Same as "adb forward *local remote*" but fails if *local* is already forwarded

adb forward −remove *local*
    Remove a specific forward socket connection.

adb forward −remove−all
    Remove all forward socket connections.

adb reverse −list
    List all reverse socket connections from device.

adb reverse *remote local*
    Reverse socket connections.

Reverse specs are one of:

- tcp:*port*

- localabstract:*unix domain socket name*

- localreserved:*unix domain socket name*

- localfilesystem:*unix domain socket name*

adb reverse –no–rebind *remote local*
    Same as 'adb reverse *remote local*' but fails if *remote* is already reversed.

adb reverse –remove *remote*
    Remove a specific reversed socket connection.

adb reverse –remove–all
    Remove all reversed socket connections from device.

adb jdwp
    List PIDs of processes hosting a JDWP transport.

adb install [–lrtsdg] *file*
    Push this package file to the device and install it.

- **–l**: Forward lock application.

- **–r**: Replace existing application.

- **–t**: Allow test packages.

- **–s**: Install application on sdcard.

- **–d**: Allow version code downgrade (debuggable packages only).

- **–g**: Grant all runtime permissions.

adb install–multiple [–lrtsdpg] *file*...
    Push this package file to the device and install it.

- **–l**: Forward lock application.

- **–r**: Replace existing application.

- **–t**: Allow test packages.

- **–s**: Install application on sdcard.

- **–d**: Allow version code downgrade (debuggable packages only).

- **–p**: Partial application install.

- **–g**: Grant all runtime permissions.

adb uninstall [–k] *package*
    Remove this app package from the device. **–k** means keep the data and cache directories.

adb bugreport [*zipfile***]**
    Return all information from the device that should be included in a bug report.

adb backup [–f *file*] [–apk|–noapk] [–obb|–noobb] [–shared|–noshared] [–all] [–system|–nosystem]
[*packages*...]
    Write an archive of the device's data to *file*. If no **–f** option is supplied then the data is written to **back-up.ab** in the current directory.

–apk | –noapk enable/disable backup of the .apks themselves in the archive; the default is noapk.

–obb | –noobb enable/disable backup of any installed apk expansion (aka .obb) files associated with each
application; the default is noobb.

–shared | –noshared enable/disable backup of the device's shared storage / SD card contents; the default is
noshared.

–all means to back up all installed applications.

–system | –nosystem toggles whether **–all** automatically includes system applications; the default is to in-

clude system apps.

packages... is the list of applications to be backed up. If the **−all** or **−shared** flags are passed, then the package list is optional. Applications explicitly given on the command line will be included even if **−nosystem** would ordinarily cause them to be omitted.

adb restore *file*
> Restore device contents from the *file* backup archive.

adb disable−verity
> Disable dm−verity checking on USERDEBUG builds.

adb enable−verity
> Re−enable dm−verity checking on USERDEBUG builds.

adb keygen *file*
> Generate adb public/private key. The private key is stored in *file*, and the public key is stored in *file*.pub. Any existing files are overwritten.

adb help
> Show help message.

adb version
> Show version number.

Scripting

adb wait−for−[−*transport***]−***state*
> Wait for device to be in the given state: **device**, **recovery**, **sideload**, or **bootloader**. *transport* is: **usb**, **local** or **any** (default = **any**)

adb start−server
> Ensure that there is a server running.

adb kill−server
> Kill the server if it is running.

adb get−state
> Prints: **offline | bootloader | device**

adb get−serialno
> Prints: *serial−number*.

adb get−devpath
> Prints: *device−path*.

adb remount
> Remounts the **/system**, **/vendor** (if present) and **/oem** (if present) partitions on the device read−write.

adb reboot [bootloader|recovery]
> Reboots the device, optionally into the bootloader or recovery program.

adb reboot sideload
> Reboots the device into the sideload mode in recovery program (adb root required).

adb reboot sideload−auto−reboot
> Reboots into the sideload mode, then reboots automatically after the sideload regardless of the result.

adb sideload *file*
> Sideloads the given package.

adb root
> Restarts the adbd daemon with root permissions.

adb unroot
> Restarts the adbd daemon without root permissions.

adb usb
> Restarts the adbd daemon listening on USB.

adb tcpip *port*
> Restarts the adbd daemon listening on TCP on the specified port.

Networking

adb ppp *tty* [*parameters*]
> Run PPP over USB.

parameters: E.g. **defaultroute debug dump local notty usepeerdns**

Note: you should not automatically start a PPP connection. *tty* refers to the tty for PPP stream. E.g. **dev:/dev/omap_csmi_tty1**

Internal Debugging

adb reconnect
> Kick current connection from host side and make it reconnect.

adb reconnect device
> Kick current connection from device side and make it reconnect.

ENVIRONMENT VARIABLES

ADB_TRACE
> Print debug information. A comma separated list of the following values **1** or **all**, **adb**, **sockets**, **packets**, **rwx**, **usb**, **sync**, **sysdeps**, **transport**, **jdwp**

ANDROID_SERIAL
> The serial number to connect to. **−s** takes priority over this if given.

ANDROID_LOG_TAGS
> When used with the logcat option, only these debug tags are printed.

SEE ALSO
> https://developer.android.com/tools/help/adb.html

AUTHORS
> The Android Open Source Project.