

# Android Debug Bridge (adb) Shell Commands

---

## adb devices

Prints a list of all attached emulator/device

In response, return serial number and state

```
$ adb devices
List of devices attached
3a21d0e device
```

## Available commands

```bash
$ adb
Android Debug Bridge version 1.0.39
Version 27.0.0-4455170
Installed as /Users/jagadeesh/Library/Android/sdk/platform-tools/adb

global options:
-a          listen on all network interfaces, not just localhost
-d          use USB device (error if multiple devices connected)
-e          use TCP/IP device (error if multiple TCP/IP devices available)
-s SERIAL   use device with given serial (overrides $ANDROID_SERIAL)
-t ID       use device with given transport id
-H          name of adb server host [default=localhost]
-P          port of adb server [default=5037]
-L SOCKET   listen on given socket for adb server
[default=tcp:localhost:5037]

general commands:
devices [-l]          list connected devices (-l for long output)
help                  show this help message
version               show version num

networking:
connect HOST[:PORT]   connect to a device via TCP/IP [default
port=5555]
disconnect [HOST[:PORT]]
    disconnect from given TCP/IP device [default port=5555], or all
forward --list         list all forward socket connections
forward [--no-rebind] LOCAL REMOTE
    forward socket connection using:
    tcp:<port> (<local> may be "tcp:0" to pick any open 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)
forward --remove LOCAL    remove specific forward socket connection
forward --remove-all     remove all forward socket connections
ppp TTY [PARAMETER...]   run PPP over USB
reverse --list            list all reverse socket connections from device
reverse [--no-rebind] REMOTE LOCAL
    reverse socket connection using:
        tcp:<port> (<remote> may be "tcp:0" to pick any open port)
        localabstract:<unix domain socket name>
        localreserved:<unix domain socket name>
        localfilesystem:<unix domain socket name>
reverse --remove REMOTE   remove specific reverse socket connection
reverse --remove-all     remove all reverse socket connections from
device

```

#### file transfer:

```

push [--sync] LOCAL... REMOTE
    copy local files/directories to device
    --sync: only push files that are newer on the host than the device
pull [-a] REMOTE... LOCAL
    copy files/dirs from device
    -a: preserve file timestamp and mode
sync [system|vendor|oem|data|all]
    sync a local build from $ANDROID_PRODUCT_OUT to the device (default
all)
    -l: list but don't copy

```

#### shell:

```

shell [-e ESCAPE] [-n] [-Tt] [-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
emu COMMAND                run emulator console command

```

#### app installation:

```

install [-lrtsdg] PACKAGE
install-multiple [-lrtsdpg] PACKAGE...
    push package(s) to the device and install them
    -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 (install-multiple only)
    -g: grant all runtime permissions
uninstall [-k] PACKAGE
    remove this app package from the device
    '-k': keep the data and cache directories

```

#### backup/restore:

```

to show usage run "adb shell bu help"

```

```

debugging:
  bugreport [PATH]
    write bugreport to given PATH [default=bugreport.zip];
    if PATH is a directory, the bug report is saved in that directory.
    devices that don not support zipped bug reports output to stdout.
  jdwp
    list pids of processes hosting a JDWP transport
  logcat
    show device log (logcat --help for more)

security:
  disable-verity
    disable dm-verity checking on userdebug builds
  enable-verity
    re-enable dm-verity checking on userdebug builds
  keygen FILE
    generate adb public/private key; private key stored in FILE,
    public key stored in FILE.pub (existing files overwritten)

scripting:
  wait-for[[-TRANSPORT]]-STATE
    wait for device to be in the given state
    State: device, recovery, sideload, or bootloader
    Transport: usb, local, or any [default=any]
  get-state
    print offline | bootloader | device
  get-serialno
    print <serial-number>
  get-devpath
    print <device-path>
  remount
    remount /system, /vendor, and /oem partitions read-write
  reboot [bootloader|recovery|sideload|sideload-auto-reboot]
    reboot the device; defaults to booting system image but
    supports bootloader and recovery too. sideload reboots
    into recovery and automatically starts sideload mode,
    sideload-auto-reboot is the same but reboots after sideloading.
  sideload OTAPACKAGE
    sideload the given full OTA package
  root
    restart adbd with root permissions
  unroot
    restart adbd without root permissions
  usb
    restart adb server listening on USB
  tcpip PORT
    restart adb server listening on TCP on PORT

internal debugging:
  start-server
    ensure that there is a server running
  kill-server
    kill the server if it is running
  reconnect
    kick connection from host side to force
  reconnect
    reconnect device
    kick connection from device side to force
  reconnect
    reconnect offline
    reset offline/unauthorized devices to force
  reconnect

environment variables:
  $ADB_TRACE
    comma-separated list of debug info to log:
    all,adb,sockets,packets,rwx,usb,sync,sysdeps,transport,jdwp
  $ADB_VENDOR_KEYS
    colon-separated list of keys (files or
directories)
  $ANDROID_SERIAL
    serial number to connect to (see -s)

```

```
$ANDROID_LOG_TAGS      tags to be used by logcat (see logcat --help)
...
```

## Reference

[Official](https://developer.android.com/studio/command-line/adb)

[Github emoji](https://help.github.com/en/articles/basic-writing-and-formatting-syntax#using-emoji)

[adbshell](http://adbshell.com/commands/adb-devices)