# ADB (Android Debug Bridge) Cheatsheet

A concise reference for ADB commands, covering device management, file transfer, shell interaction, and more, tailored for Android developers and enthusiasts.

## Device Management

### Basic Device Commands

| | |
|---|---|
| `adb devices` | Lists all connected Android devices. Shows serial number and state (device, offline, unauthorized). |
| `adb devices -l` | Lists connected devices with more details like transport ID and product/model information. |
| `adb get-state` | Prints the device's current state (device, offline, unauthorized, unknown). |
| `adb get-serialno` | Retrieves the serial number of the connected device. |
| `adb wait-for-device` | Blocks execution until a device is connected. Useful in scripts. |
| `adb kill-server` | Kills the ADB server process. The server will restart automatically when another ADB command is executed. |
| `adb start-server` | Starts the ADB server. Usually not required as ADB commands start the server if it's not running. |

### Device Selection

| | |
|---|---|
| `adb -s <serialNumber> <command>` | Specifies the device by serial number to execute the command on. Useful when multiple devices are connected.<br>**Example:** `adb -s emulator-5554 shell getprop ro.product.model` |
| `adb -e <command>` | Directs the command to the only connected emulator. Returns an error if more than one emulator is running. |
| `adb -d <command>` | Directs the command to the only connected hardware device. Returns an error if more than one device is connected. |

### Reboot Commands

| | |
|---|---|
| `adb reboot` | Reboots the Android device. |
| `adb reboot bootloader` | Reboots the device into the bootloader (fastboot) mode. |
| `adb reboot recovery` | Reboots the device into recovery mode. |

## File Management

### File Transfer

| | |
|---|---|
| `adb push <local> <remote>` | Copies a file or directory from your computer (local) to the Android device (remote).<br>**Example:** `adb push myfile.txt /sdcard/` |
| `adb pull <remote> <local>` | Copies a file or directory from the Android device (remote) to your computer (local).<br>**Example:** `adb pull /sdcard/myimage.jpg .` (copies to current directory) |

### Shell File Operations

| | |
|---|---|
| `adb shell ls <path>` | Lists files and directories in the specified path on the device. Use `-l` for detailed listing.<br>**Example:** `adb shell ls /sdcard/DCIM/Camera` |
| `adb shell mkdir <path>` | Creates a new directory at the specified path on the device.<br>**Example:** `adb shell mkdir /sdcard/new_folder` |
| `adb shell rm <path>` | Removes the file at the specified path on the device.<br>**Example:** `adb shell rm /sdcard/temp.txt` |
| `adb shell rm -r <path>` | Removes the directory recursively at the specified path on the device.<br>**Example:** `adb shell rm -r /sdcard/temp_folder` |
| `adb shell mv <source> <destination>` | Moves a file from source to destination path on the device.<br>**Example:** `adb shell mv /sdcard/test.txt /sdcard/backup/test.txt` |

## Shell Commands

### Basic Shell Interaction

| | |
|---|---|
| `adb shell <command>` | Executes a shell command on the Android device.<br>**Example:** `adb shell getprop ro.product.model` (gets the device model) |
| `adb shell` | Opens an interactive shell session on the Android device. Type `exit` to close the session. |
| `adb emu <command>` | Runs emulator console command.<br>**Example:** `adb emu geo fix -80 30` |

### Package Management

| | |
|---|---|
| `adb install <path_to_apk>` | Installs an Android application package (APK) from your computer to the device.<br><br>**Example:** `adb install myapp.apk` |
| `adb install -r <path_to_apk>` | Reinstalls an existing application, keeping its data.<br><br>**Example:** `adb install -r myapp.apk` |
| `adb uninstall <package_name>` | Uninstalls the application with the specified package name from the device.<br><br>**Example:** `adb uninstall com.example.myapp` |
| `adb shell pm list packages` | Lists all installed packages on the device. |
| `adb shell pm clear <package_name>` | Clear user data of a package.<br><br>**Example:** `adb shell pm clear com.example.myapp` |

### Screen and Media

| | |
|---|---|
| `adb shell screencap -p /sdcard/screen.png` | Takes a screenshot of the device screen and saves it as a PNG file on the device.<br><br>**Then use `adb pull /sdcard/screen.png` to copy it to your computer.** |
| `adb shell screenrecord /sdcard/screen.mp4` | Records a video of the device screen and saves it as an MP4 file on the device.<br><br>**Press Ctrl+C to stop recording. Then use `adb pull /sdcard/screen.mp4` to copy it to your computer.** |

## Debugging & Logging

### Logcat Commands

| | |
|---|---|
| `adb logcat` | Starts streaming system log messages to your console. |
| `adb logcat -c` | Clears the current log buffers. |
| `adb logcat -v <format>` | Specifies the log message format ( `brief` , `process` , `tag` , `thread` , `raw` , `time` , `threadtime` , `long` ).<br><br>**Example:** `adb logcat -v threadtime` |
| `adb logcat <tag>:<level>` | Filters log messages by tag and priority level ( `V` - Verbose, `D` - Debug, `I` - Info, `W` - Warning, `E` - Error, `F` - Fatal, `S` - Silent).<br><br>**Example:** `adb logcat MyTag:D *:S` (shows Debug messages from MyTag and suppresses all other tags) |
| `adb logcat -f <filename>` | Saves the log output to the specified file.<br><br>**Example:** `adb logcat -f mylog.txt` |
| `adb logcat \| grep <search_term>` | Filters logcat output using grep for specific strings.<br><br>**Example:** `adb logcat \| grep "MyApp"` |

### Debugging Applications

| | |
|---|---|
| `adb jdwp` | Lists the process IDs (PID) of all applications that are enabled for JDWP (Java Debug Wire Protocol) debugging. |
| `adb forward tcp:<local_port> jdwp:<remote_pid>` | Sets up port forwarding for debugging a specific application process.<br><br>**Example:** `adb forward tcp:8000 jdwp:1234` (forwards local port 8000 to process 1234) |
| `adb shell am start -D -n <package>/<activity>` | Starts an application in debug mode, waiting for a debugger to attach.<br><br>**Example:** `adb shell am start -D -n com.example.myapp/.MainActivity` |

### ADB over Network

| | |
|---|---|
| `adb tcpip <port>` | Restarts ADB in TCP/IP mode on the specified port (usually 5555).<br><br>**Run on the device itself (via USB connection).** |
| `adb connect <device_ip>:<port>` | Connects to the device over the network.<br><br>**Run on the computer after enabling `adb tcpip` on the device. Example: `adb connect 192.168.1.100:5555`** |
| `adb disconnect <device_ip>:<port>` | Disconnects from a network-connected device.<br><br>**Example:** `adb disconnect 192.168.1.100:5555` |