

tldr pages book

Simplified and community-driven man pages

Generated on Mon Oct 23 23:48:53 2023

Website: <https://tldr.sh>

GitHub: <https://github.com/tldr-pages/tldr>

Android

am

Android activity manager.

More information: <https://developer.android.com/studio/command-line/adb#am>.

- Start a specific activity:

```
am start -n {{com.android.settings/.Settings}}
```

- Start an activity and pass [d]ata to it:

```
am start -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

- Start an activity matching a specific action and [c]ategory:

```
am start -a {{android.intent.action.MAIN}} -c  
{{android.intent.category.HOME}}
```

- Convert an intent to a URI:

```
am to-uri -a {{android.intent.action.VIEW}} -d {{tel:123}}
```

bugreport

Show an Android bug report.

This command can only be used through **adb shell**.

More information: [https://cs.android.com/android/platform/superproject/+main:frameworks/native/cmds/bugreport](https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreport).

- Display a complete bug report of an Android device:

```
bugreport
```

bugreportz

Generate a zipped Android bug report.

This command can only be used through **adb shell**.

More information: [https://cs.android.com/android/platform/superproject/+main:frameworks/native/cmds/bugreportz](https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/bugreportz).

- Generate a complete zipped bug report of an Android device:

```
bugreportz
```

- Show the progress of a running bugreportz operation:

```
bugreportz -p
```

- Show the version of bugreportz:

```
bugreportz -v
```

- Display help:

```
bugreportz -h
```

cmd

Android service manager.

More information: [https://cs.android.com/android/platform/superproject/+main:frameworks/native/cmds/cmd/](https://cs.android.com/android/platform/superproject/+/main:frameworks/native/cmds/cmd/).

- List all running services:

```
cmd -l
```

- Call a specific service:

```
cmd {{service}}
```

- Call a service with specific arguments:

```
cmd {{service}} {{argument1 argument2 ...}}
```

dalvikvm

Android Java virtual machine.

More information: <https://source.android.com/devices/tech/dalvik>.

- Start a specific Java program:

```
dalvikvm -classpath {{path/to/file.jar}} {{classname}}
```

dumpsys

Provide information about Android system services.

This command can only be used through **adb shell**.

More information: <https://developer.android.com/studio/command-line/dumpsys>.

- Get diagnostic output for all system services:

```
dumpsys
```

- Get diagnostic output for a specific system service:

```
dumpsys {{service}}
```

- List all services dumpsys can give information about:

```
dumpsys -l
```

- List service-specific arguments for a service:

```
dumpsys {{service}} -h
```

- Exclude a specific service from the diagnostic output:

```
dumpsys --skip {{service}}
```

- Specify a timeout period in seconds (defaults to 10s):

```
dumpsys -t {{8}}
```


getprop

Show information about Android system properties.

More information: <https://manned.org/getprop>.

- Display information about Android system properties:

```
getprop
```

- Display information about a specific property:

```
getprop {{property}}
```

- Display the SDK API level:

```
getprop {{ro.build.version.sdk}}
```

- Display the Android version:

```
getprop {{ro.build.version.release}}
```

- Display the Android device model:

```
getprop {{ro.vendor.product.model}}
```

- Display the OEM unlock status:

```
getprop {{ro.oem_unlock_supported}}
```

- Display the MAC address of the Android's Wi-Fi card:

```
getprop {{ro.boot.wifimacaddr}}
```

input

Send event codes or touchscreen gestures to an Android device.

This command can only be used through **adb shell**.

More information: https://developer.android.com/reference/android/view/KeyEvent.html#constants_1.

- Send an event code for a single character to an Android device:

```
input keyevent {{event_code}}
```

- Send a text to an Android device (%s represents spaces):

```
input text "{{text}}"
```

- Send a single tap to an Android device:

```
input tap {{x_position}} {{y_position}}
```

- Send a swipe gesture to an Android device:

```
input swipe {{x_start}} {{y_start}} {{x_end}} {{y_end}}  
{{duration_in_ms}}
```

- Send a long press to an Android device using a swipe gesture:

```
input swipe {{x_position}} {{y_position}} {{x_position}}  
{{y_position}} {{duration_in_ms}}
```

logcat

Dump a log of system messages, including stack traces when an error occurred, and information messages logged by applications.

More information: <https://developer.android.com/studio/command-line/logcat>.

- Display system logs:

```
logcat
```

- Write system logs to a file:

```
logcat -f {{path/to/file}}
```

- Display lines that match a regular expression:

```
logcat --regex {{regular_expression}}
```

- Display logs for a specific PID:

```
logcat --pid={{pid}}
```

- Display logs for the process of a specific package:

```
logcat --pid=$(pidof -s {{package}})
```

pkg

Package management utility for Termux.

More information: https://wiki.termux.com/wiki/Package_Management.

- Upgrade all installed packages:

```
pkg upgrade
```

- Install a package:

```
pkg install {{package}}
```

- Uninstall a package:

```
pkg uninstall {{package}}
```

- Reinstall a package:

```
pkg reinstall {{package}}
```

- Search for a package:

```
pkg search {{package}}
```

pm

Display information about apps on an Android device.

More information: <https://developer.android.com/studio/command-line/adb#pm>.

- List all installed apps:

```
pm list packages
```

- List all installed system apps:

```
pm list packages -s
```

- List all installed 3rd-Party apps:

```
pm list packages -3
```

- List apps matching specific keywords:

```
pm list packages {{keyword1 keyword2 ...}}
```

- Display a path of the APK of a specific app:

```
pm path {{app}}
```

screencap

Take a screenshot of a mobile display.

This command can only be used through **adb shell**.

More information: <https://developer.android.com/studio/command-line/adb#screencap>.

- Take a screenshot:

```
screencap {{path/to/file}}
```

settings

Get information about the Android OS.

More information: <https://adbinstaller.com/commands/adb-shell-settings-5b670d5ee7958178a2955536>.

- Display a list of settings in the global namespace:

```
settings list {{global}}
```

- Get a value of a specific setting:

```
settings get {{global}} {{airplane_mode_on}}
```

- Set a specific value of a setting:

```
settings put {{system}} {{screen_brightness}} {{42}}
```

- Delete a specific setting:

```
settings delete {{secure}} {{screensaver_enabled}}
```

wm

Show information about the screen of an Android device.

This command can only be used through **adb shell**.

More information: <https://adbinstaller.com/commands/adb-shell-wm-5b672b17e7958178a2955538>.

- Display the physical size of an Android device's screen:

```
wm {{size}}
```

- Display the physical density of an Android device's screen:

```
wm {{density}}
```


Sunos

devfsadm

Administration command for **/dev**. Maintains the **/dev** namespace.

More information: <https://www.unix.com/man-page/sunos/1m/devfsadm>.

- Scan for new disks:

```
devfsadm -c disk
```

- Cleanup any dangling /dev links and scan for new device:

```
devfsadm -C -v
```

- Dry-run - output what would be changed but make no modifications:

```
devfsadm -C -v -n
```

dmesg

Write the kernel messages to **stdout**.

More information: <https://www.unix.com/man-page/sunos/1m/dmesg>.

- Show kernel messages:

```
dmesg
```

- Show how much physical memory is available on this system:

```
dmesg | grep -i memory
```

- Show kernel messages 1 page at a time:

```
dmesg | less
```

prctl

Get or set the resource controls of running processes, tasks, and projects.

More information: <https://www.unix.com/man-page/sunos/1/prctl>.

- Examine process limits and permissions:

```
prctl {{pid}}
```

- Examine process limits and permissions in machine parsable format:

```
prctl -P {{pid}}
```

- Get specific limit for a running process:

```
prctl -n process.max-file-descriptor {{pid}}
```

prstat

Report active process statistics.

More information: <https://www.unix.com/man-page/sunos/1m/prstat>.

- Examine all processes and reports statistics sorted by CPU usage:

```
prstat
```

- Examine all processes and reports statistics sorted by memory usage:

```
prstat -s rss
```

- Report total usage summary for each user:

```
prstat -t
```

- Report microstate process accounting information:

```
prstat -m
```

- Print out a list of top 5 CPU using processes every second:

```
prstat -c -n {{5}} -s cpu {{1}}
```

snoop

Network packet sniffer.

SunOS equivalent of tcpdump.

More information: <https://www.unix.com/man-page/sunos/1m/snoop>.

- Capture packets on a specific network interface:

```
snoop -d {{e1000g0}}
```

- Save captured packets in a file instead of displaying them:

```
snoop -o {{path/to/file}}
```

- Display verbose protocol layer summary of packets from a file:

```
snoop -V -i {{path/to/file}}
```

- Capture network packets that come from a hostname and go to a given port:

```
snoop to port {{port}} from host {{hostname}}
```

- Capture and show a hex-dump of network packets exchanged between two IP addresses:

```
snoop -x0 -p4 {{ip1}} {{ip2}}
```

svcadm

Manipulate service instances.

More information: <https://www.unix.com/man-page/linux/1m/svcadm>.

- Enable a service in the service database:

```
svcadm enable {{service_name}}
```

- Disable service:

```
svcadm disable {{service_name}}
```

- Restart a running service:

```
svcadm restart {{service_name}}
```

- Command service to re-read configuration files:

```
svcadm refresh {{service_name}}
```

- Clear a service from maintenance state and command it to start:

```
svcadm clear {{service_name}}
```

svccfg

Import, export, and modify service configurations.

More information: <https://www.unix.com/man-page/linux/1m/svccfg>.

- Validate configuration file:

```
svccfg validate {{path/to/smf_file.xml}}
```

- Export service configurations to file:

```
svccfg export {{servicename}} > {{path/to/smf_file.xml}}
```

- Import/update service configurations from file:

```
svccfg import {{path/to/smf_file.xml}}
```


SVCS

List information about running services.

More information: <https://www.unix.com/man-page/linux/1/svcs>.

- List all running services:

```
svcs
```

- List services that are not running:

```
svcs -vx
```

- List information about a service:

```
svcs apache
```

- Show location of service log file:

```
svcs -L apache
```

- Display end of a service log file:

```
tail $(svcs -L apache)
```

truss

Troubleshooting tool for tracing system calls.

SunOS equivalent of strace.

More information: <https://www.unix.com/man-page/linux/1/truss>.

- Start tracing a program by executing it, following all child processes:

```
truss -f {{program}}
```

- Start tracing a specific process by its PID:

```
truss -p {{pid}}
```

- Start tracing a program by executing it, showing arguments and environment variables:

```
truss -a -e {{program}}
```

- Count time, calls, and errors for each system call and report a summary on program exit:

```
truss -c -p {{pid}}
```

- Trace a process filtering output by system call:

```
truss -p {{pid}} -t {{system_call_name}}
```

