What's the Android ADB shell “dumpsys” tool and what are its benefits?

I'm looking for the full list of ADB shell `dumpsys` commands with a full explanation of all of the commands.

Where can I find this information?

android shell adb

edited Nov 30 '14 at 0:07
Agamemnus
637 9 30

asked Jun 26 '12 at 6:14
Vishwanath.M
2,753 9 26 39

4 Answers

What's dumpsys and what are its benefit

dumpsys is an android tool that runs on the device and dumps interesting information about the status of system services.

Obvious benefits:

1. Possibility to easily get system information in a simple string representation.
2. Possibility to use dumped CPU, RAM, Battery, storage stats for a pretty charts, which will allow you to check how your application affects the overall device!

What information can we retrieve from dumpsys shell command and how we can use it

If you run dumpsys you would see a ton of system information. But you can use only separate parts of this big dump.

to see all of the "subcommands" of dumpsys do:

```
dumpsys | grep "DUMP OF SERVICE"
```

Output:

```
DUMP OF SERVICE SurfaceFlinger:
DUMP OF SERVICE accessibility:
DUMP OF SERVICE account:
DUMP OF SERVICE activity:
DUMP OF SERVICE alarm:
DUMP OF SERVICE appwidget:
DUMP OF SERVICE audio:
DUMP OF SERVICE backup:
DUMP OF SERVICE battery:
DUMP OF SERVICE batteryinfo:
DUMP OF SERVICE clipboard:
DUMP OF SERVICE connectivity:
DUMP OF SERVICE content:
DUMP OF SERVICE cpuinfo:
DUMP OF SERVICE device_policy:
DUMP OF SERVICE devicestoragemonitor:
DUMP OF SERVICE diskstats:
DUMP OF SERVICE dropbox:
DUMP OF SERVICE entropy:
DUMP OF SERVICE hardware:
DUMP OF SERVICE input_method:
DUMP OF SERVICE iphonesubinfo:
```

```
DUMP OF SERVICE isms:
DUMP OF SERVICE location:
DUMP OF SERVICE media.audio_flinger:
DUMP OF SERVICE media.audio_policy:
DUMP OF SERVICE media.player:
DUMP OF SERVICE meminfo:
DUMP OF SERVICE mount:
DUMP OF SERVICE netstat:
DUMP OF SERVICE network_management:
DUMP OF SERVICE notification:
DUMP OF SERVICE package:
DUMP OF SERVICE permission:
DUMP OF SERVICE phone:
DUMP OF SERVICE power:
DUMP OF SERVICE reboot:
DUMP OF SERVICE screenshot:
DUMP OF SERVICE search:
DUMP OF SERVICE sensor:
DUMP OF SERVICE simphonebook:
DUMP OF SERVICE statusbar:
DUMP OF SERVICE telephony.registry:
DUMP OF SERVICE throttle:
DUMP OF SERVICE usagestats:
DUMP OF SERVICE vibrator:
DUMP OF SERVICE wallpaper:
DUMP OF SERVICE wifi:
DUMP OF SERVICE window:
```

Some Dumping examples and output

1) Getting all possible battery statistic:

```
$~ adb shell dumpsys battery
```

You will get output:

```
Current Battery Service state:
AC powered: false
AC capacity: 500000
USB powered: true
status: 5
health: 2
present: true
level: 100
scale: 100
voltage:4201
temperature: 271 <---------- Battery temperature! %)
technology: Li-poly <---------- Battery technology! %)
```

2)Getting wifi informations

```
~$ adb shell dumpsys wifi
```

Output:

```
Wi-Fi is enabled
Stay-awake conditions: 3

Internal state:
interface tiwlan0 runState=Running
SSID: XXXXXXX BSSID: xx:xx:xx:xx:xx:xx, MAC: xx:xx:xx:xx:xx:xx, Supplicant state:
COMPLETED, RSSI: -60, Link speed: 54, Net ID: 2, security: 0, idStr: null
ipaddr 192.168.1.xxx gateway 192.168.x.x netmask 255.255.255.0 dns1 192.168.x.x
dns2 8.8.8.8 DHCP server 192.168.x.x lease 604800 seconds
haveIpAddress=true, obtainingIpAddress=false, scanModeActive=false
lastSignalLevel=2, explicitlyDisabled=false

Latest scan results:

Locks acquired: 28 full, 0 scan
Locks released: 28 full, 0 scan

Locks held:
```

3) Getting CPU info

```
~$ adb shell dumpsys cpuinfo
```

Output:

```
Load: 0.08 / 0.4 / 0.64
CPU usage from 42816ms to 34683ms ago:
system_server: 1% = 1% user + 0% kernel / faults: 16 minor
kdebuglog.sh: 0% = 0% user + 0% kernel / faults: 160 minor
tiwlan_wq: 0% = 0% user + 0% kernel
usb_mass_storag: 0% = 0% user + 0% kernel
pvr_workqueue: 0% = 0% user + 0% kernel
+sleep: 0% = 0% user + 0% kernel
+sleep: 0% = 0% user + 0% kernel
TOTAL: 6% = 1% user + 3% kernel + 0% irq
```

4)Getting memory usage informations

```
~$ adb shell dumpsys meminfo 'your apps package name'
```

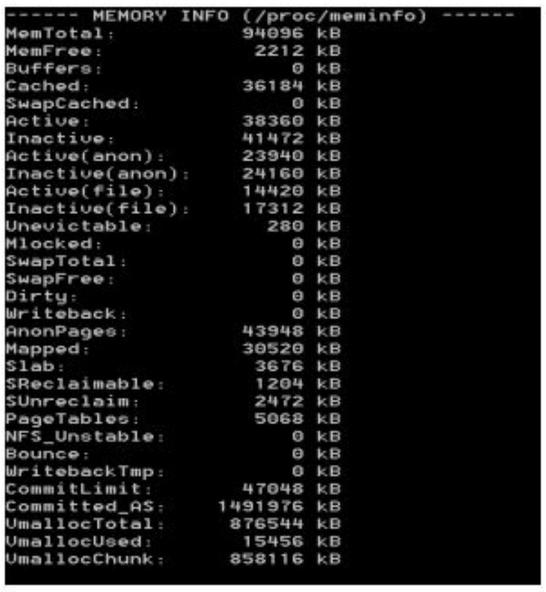
Output:

```
** MEMINFO in pid 5527 [com.sec.android.widgetapp.weatherclock] **
                    native   dalvik    other    total
            size:     2868     5767      N/A     8635
       allocated:     2861     2891      N/A     5752
            free:        6     2876      N/A     2882
           (Pss):      532       80     2479     3091
  (shared dirty):      932     2004     6060     8996
    (priv dirty):      512       36     1872     2420

 Objects
           Views:        0        ViewRoots:        0
     AppContexts:        0       Activities:        0
          Assets:        3    AssetManagers:        3
   Local Binders:        2    Proxy Binders:        8
Death Recipients:        0
 OpenSSL Sockets:        0


 SQL
               heap:         0         MEMORY_USED:         0
 PAGECACHE_OVERFLOW:         0         MALLOC_SIZE:         0
```

If you want see the info for all processes, use `~$ adb shell dumpsys meminfo`



```
------ MEMORY INFO (/proc/meminfo) ------
MemTotal:              94096 kB
MemFree:                2212 kB
Buffers:                   0 kB
Cached:                36184 kB
SwapCached:                0 kB
Active:                38360 kB
Inactive:              41472 kB
Active(anon):          23940 kB
Inactive(anon):        24160 kB
Active(file):          14420 kB
Inactive(file):        17312 kB
Unevictable:             280 kB
Mlocked:                   0 kB
SwapTotal:                 0 kB
SwapFree:                  0 kB
Dirty:                     0 kB
Writeback:                 0 kB
AnonPages:             43948 kB
Mapped:                30520 kB
Slab:                   3676 kB
SReclaimable:           1204 kB
SUnreclaim:             2472 kB
PageTables:             5068 kB
NFS_Unstable:              0 kB
Bounce:                    0 kB
WritebackTmp:              0 kB
CommitLimit:           47048 kB
Committed_AS:        1491976 kB
VmallocTotal:         876544 kB
VmallocUsed:           15456 kB
VmallocChunk:         858116 kB
```

dumpsys is ultimately flexible and useful tool!

If you want to use this tool do not forget to add permission into your android manifest automatically `android.permission.DUMP`

Try to test all commands to learn more about dumpsys. Happy dumping!

edited Sep 16 '15 at 21:07
Tiny Giant
11.4k 6 26 48

answered Jun 26 '12 at 6:19
K_Anas
24.1k 6 47 73

2 Is this from a document somewhere or just your own experimentation? – brianestey Oct 25 '12 at 3:19

9 Also very nice is: adb shell dumpsys activity broadcasts , shows all broadcasts – joecks Mar 19 '13 at 10:26

starting with Kitkat (I think), "batteryinfo" was replaced with "batterystats" , and its output and usage also seems different. – android developer Aug 29 '14 at 15:53

dumpsys is incredibly powerful. It reports private values of core Android services that you would never be able to typically obtain. – Kevin Mark Oct 26 '14 at 5:41

unfortunately it requires root or system level permission to use in java program. – umesh Jul 17 '15 at 10:21

According to official Android information about *dumpsys*:

The *dumpsys* tool runs on the device and provides information about the status of system services.

To get a list of available services use

```
adb shell dumpsys -l
```

edited Aug 2 '16 at 11:17

answered Aug 2 '16 at 10:33
lujop
6,491 6 40 74

1 I can't see why this was downvoted - it's a useful link. The list of available services `-l` works on my Lollipop but not on my Jelly Bean. – starfry Aug 18 '16 at 13:26

This answer has an actual pointer to the actual documentation. I believe this is what the OP was looking for – Robert R Evans Sep 23 '16 at 17:46

i use **dumpsys** to catch if app is crashed and process is still active. situation i used it is to find about remote machine app is crashed or not.

```
dumpsys | grep myapp | grep "Application Error"
```

or

```
adb shell dumpsys | grep myapp | grep Error
```

or anything that helps...etc

if app is not running you will get nothing as result. When app is stoped messsage is shown on screen by android, process is still active and if you check via "**ps**" command or anything else, you will see process state is not showing any error or crash meaning. But when you click button to close message, app process will cleaned from process list. so catching crash state without any code in application is hard to find. but **dumpsys** helps you.

answered Nov 10 '15 at 9:39
Engin OZTURK
1,145 1 14 12

Looking at the source code for dumpsys and service, you can get the list of services available by executing this:

```
adb shell service list
```

You can then supply the service name you are interested in to dumpsys to get the specific information. For example (note that not all services provide dump info):

```
adb shell dumpsys activity
adb shell dumpsys cpuinfo
adb shell dumpsys battery
```

As you can see in the code (and in K_Anas's answer), if you call dumpsys without any service name, it will dump the info on all services in one big dump:

```
adb shell dumpsys
```

Some services can receive additional arguments on what to show which normally is explained if you supplied a `-h` argument, for example:

```
adb shell dumpsys activity -h
adb shell dumpsys window -h
adb shell dumpsys meminfo -h
adb shell dumpsys package -h
adb shell dumpsys batteryinfo -h
```

answered Mar 2 '13 at 0:46
Joe
12.3k 1 28 44