am commands

force-stop *<PACKAGE>*
Force stop everything associated with <PACKAGE> (the app's package name)

kill-all
Kill all background processes

profile start *<PROCESS>* *<FILE>*
Start profiler on <PROCESS>, write results to <FILE>

profile stop *<PROCESS>*
Stop profiler on <PROCESS>

screen-compat [on|off] <PACKAGE>
Control screen compatibility mode of <PACKAGE>

display-size [reset|<WxH>]
Override emulator/device display size

display-density *<dpi>*
Override emulator/device display density

to-uri *<INTENT>*
Print the given intent specification as a URI.

to-intent-uri *<INTENT>*
Print the given intent specification as an intent: URI.

clear-debug-app
Clear the package previous set for debugging with *set-debug-app*

am command : set-debug-app

set-debug-app *[options]* **<PACKAGE>**
Set application <PACKAGE> to debug.

-w
Wait for debugger when application starts

--persistent
Retain this value

am command : broadcast

broadcast [options] <INTENT>
Issue a broadcast intent.

--user [<USER_ID> | all | current]
Specify which user to send to; if not specified then send to all users.

am command : instrument

instrument *[options]* **<COMPONENT>**
Start monitoring with an Instrumentation instance. Typically the target <COMPONENT> is the form <TEST_PACKAGE> / <RUNNER_CLASS>.

-r
Print raw results (otherwise decode <REPORT_KEY_STREAMRESULT>). Use with [-e perf true] to generate raw output for performance measurements.

-e <NAME> <VALUE>
Set argument <NAME> to <VALUE>. For test runners a common form is -e <testrunner_flag> <value>[,<value> ...]

-p <FILE>
Write profiling data to <FILE>

-w
Wait for instrumentation to finish before returning. Required for test runners.

--no-window-animation
Turn off window animations while running

--user [<USER_ID>]
Specify which user instrumentation runs in; current user if not specified

am command : monitor

monitor *[options]*
Start monitoring for crashes or ANRs

--gdb
Start gdbserv on the given port at crash/ANR

am command : dumpheap

dumpheap *[options]* **<PROCESS> <FILE>**
Dump the heap of <PROCESS>, write to <FILE>

--user [<USER_ID>]
When supplying a process name, specify user of process to dump; uses current user if not specified.

-n
Dump native heap instead of managed heap

am command : start

start *[options]* **<INTENT>**
Start an *Activity* specified by <INTENT>

-D
Enable Debugging

-W
Wait for launch to complete

--start-profiler <FILE>
Start profiler and send results to *<FILE>*

-P <FILE>
Like *--start-profiler*, but profiling stops when the app goes idle.

-R <COUNT>
Repeat the activity launch *<COUNT>* times. Prior to each repeat, the top activity will be finished.

-S
Force stop the target app before starting the activity.

--opengl-trace
Enable tracing of OpenGL functions

--user [<USER_ID>]
Specify which user to run as; if not specified, then run as the current user.

am command : startservice

startservice *[options]* **<INTENT>**
Start the *Service* specified by <INTENT>

--user [<USER_ID>]
Specify which user to run as; if not specified, then run as the current user.

am command : kill

kill [options] <PACKAGE>
Kill all processes associated with <PACKAGE> (the app's package name). This command kills only processes that are safe to kill and that will not impact the user experience.

--user [<USER_ID> | all | current]
Specify user whose processes to kill; all users if not specified.