

UI/Application Exerciser Monkey

The Monkey is a program that runs on your [emulator](#) (/tools/help/emulator) or device and generates pseudo-random streams of user events such as clicks, touches, or gestures, as well as a number of system-level events. You can use the Monkey to stress-test applications that you are developing, in a random yet repeatable manner.

Overview

The Monkey is a command-line tool that you can run on any emulator instance or on a device. It sends a pseudo-random stream of user events into the system, which acts as a stress test on the application software you are developing.

The Monkey includes a number of options, but they break down into four primary categories:

- Basic configuration options, such as setting the number of events to attempt.
- Operational constraints, such as restricting the test to a single package.
- Event types and frequencies.
- Debugging options.

When the Monkey runs, it generates events and sends them to the system. It also *watches* the system under test and looks for three conditions, which it treats specially:

- If you have constrained the Monkey to run in one or more specific packages, it watches for attempts to navigate to any other packages, and blocks them.
- If your application crashes or receives any sort of unhandled exception, the Monkey will stop and report the error.
- If your application generates an *application not responding* error, the Monkey will stop and report the error.

Depending on the verbosity level you have selected, you will also see reports on the progress of the Monkey and the events being generated.

Basic use of the Monkey

You can launch the Monkey using a command line on your development machine or from a script. Because the Monkey runs in the emulator/device environment, you must launch it from a shell in that environment. You can do this by prefacing `adb shell` to each command, or by entering the shell and entering Monkey commands directly.

The basic syntax is:

```
$ adb shell monkey [options] <event-count>
```

With no options specified, the Monkey will launch in a quiet (non-verbose) mode, and will send events to any (and all) packages installed on your target. Here is a more typical command line, which will launch your application and send 500 pseudo-random events to it:

```
$ adb shell monkey -p your.package.name -v 500
```

Command options reference

The table below lists all options you can include on the Monkey command line.

Category Option		Description
General	<code>--help</code>	Prints a simple usage guide.

-v Each -v on the command line will increment the verbosity level. Level 0 (the default) provides little information beyond startup notification, test completion, and final results. Level 1 provides more details about the test as it runs, such as individual events being sent to your activities. Level 2 provides more detailed setup information such as activities selected or not selected for testing.

Events	-s <seed>	Seed value for pseudo-random number generator. If you re-run the Monkey with the same seed value, it will generate the same sequence of events.
	--throttle <milliseconds>	Inserts a fixed delay between events. You can use this option to slow down the Monkey. If not specified, there is no delay and the events are generated as rapidly as possible.
	--pct-touch <percent>	Adjust percentage of touch events. (Touch events are a down-up event in a single place on the screen.)
	--pct-motion <percent>	Adjust percentage of motion events. (Motion events consist of a down event somewhere on the screen, a series of pseudo-random movements, and an up event.)
	--pct-trackball <percent>	Adjust percentage of trackball events. (Trackball events consist of one or more random movements, sometimes followed by a click.)
	--pct-nav <percent>	Adjust percentage of "basic" navigation events. (Navigation events consist of up/down/left/right, as input from a directional input device.)
	--pct-majornav <percent>	Adjust percentage of "major" navigation events. (These are navigation events that will typically cause actions within your UI, such as the center button in a 5-way pad, the back key, or the menu key.)
	--pct-syskeys <percent>	Adjust percentage of "system" key events. (These are keys that are generally reserved for use by the system, such as Home, Back, Start Call, End Call, or Volume controls.)

--pct-appswitch **<percent>** Adjust percentage of activity launches. At random intervals, the Monkey will issue a `startActivity()` call, as a way of maximizing coverage of all activities within your package.

--pct-anyevent **<percent>** Adjust percentage of other types of events. This is a catch-all for all other types of events such as keypresses, other less-used buttons on the device, and so forth.

Constraints **-p** **<allowed-package-name>** If you specify one or more packages this way, the Monkey will *only* allow the system to visit activities within those packages. If your application requires access to activities in other packages (e.g. to select a contact) you'll need to specify those packages as well. If you don't specify any packages, the Monkey will allow the system to launch activities in all packages. To specify multiple packages, use the `-p` option multiple times — one `-p` option per package.

-c **<main-category>** If you specify one or more categories this way, the Monkey will *only* allow the system to visit activities that are listed with one of the specified categories. If you don't specify any categories, the Monkey will select activities listed with the category `Intent.CATEGORY_LAUNCHER` or `Intent.CATEGORY_MONKEY`. To specify multiple categories, use the `-c` option multiple times — one `-c` option per category.

Debugging **--dbg-no-events** When specified, the Monkey will perform the initial launch into a test activity, but will not generate any further events. For best results, combine with `-v`, one or more package constraints, and a non-zero throttle to keep the Monkey running for 30 seconds or more. This provides an environment in which you can monitor package transitions invoked by your application.

--hprof If set, this option will generate profiling reports immediately before and after the Monkey event sequence. This will generate large (~5Mb) files in `data/misc`, so use with care. For information on analyzing profiling reports, see [Profile your app performance](#) (`/studio/profile`).

--ignore-crashes Normally, the Monkey will stop when the application crashes or experiences any type of unhandled exception. If you specify this option, the Monkey will continue to send events to the system, until the count is completed.

--ignore-timeouts	Normally, the Monkey will stop when the application experiences any type of timeout error such as a "Application Not Responding" dialog. If you specify this option, the Monkey will continue to send events to the system, until the count is completed.
--ignore-security-exceptions	Normally, the Monkey will stop when the application experiences any type of permissions error, for example if it attempts to launch an activity that requires certain permissions. If you specify this option, the Monkey will continue to send events to the system, until the count is completed.
--kill-process-after-error	Normally, when the Monkey stops due to an error, the application that failed will be left running. When this option is set, it will signal the system to stop the process in which the error occurred. Note, under a normal (successful) completion, the launched process(es) are not stopped, and the device is simply left in the last state after the final event.
--monitor-native-crashes	Watches for and reports crashes occurring in the Android system native code. If --kill-process-after-error is set, the system will stop.
--wait-dbg	Stops the Monkey from executing until a debugger is attached to it.

Content and code samples on this page are subject to the licenses described in the [Content License](#) (/license).
Java and OpenJDK are trademarks or registered trademarks of Oracle and/or its affiliates.

Last updated 2023-04-12 UTC.