1)Android Monkey Runner Testing :-

The UI/Application Exerciser Monkey, usually called "monkey", is a command-line tool that sends pseudo-random streams of keystrokes, touches, and gestures to a device. You run it with the Android Debug Bridge (adb) tool.

You use it to stress-test your application and report back errors that are encountered. You can repeat a stream of events by running the tool each time with the same random number seed.

**Monkey features**

Monkey has many features, but it can be all be summed up to these four categories.

1. Basic configuration options
2. Operational constraints
3. Event types and frequencies
4. Debugging options

**Monkey Usage**

In order to use monkey , open up a command prompt and just naviagte to the following directory.

**Monkey Usage**

In order to use monkey , open up a command prompt and just naviagte to the following directory.

android->sdk->platform-tools

Once inside the directory , attach your device with the PC , and run the following command.

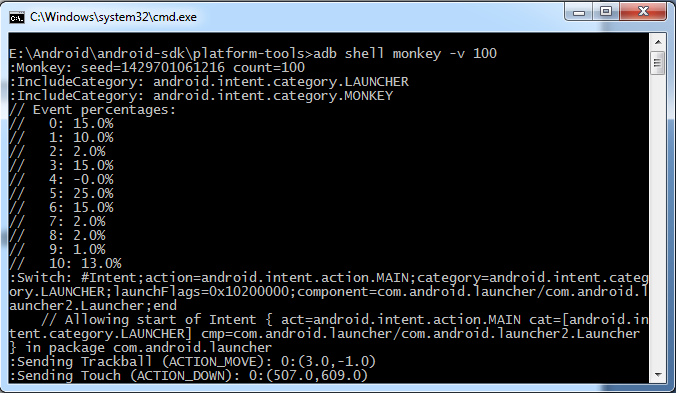
adb shell monkey -v 100

**Syntax :- adb shell monkey [options] <event-count>**

This command can be broken down into these steps.

* adb - Android Debug Bridge. A tool used to connect and sends commands to your Android phone from a desktop or laptop computer.
* shell - shell is just an inteface on the device that translates our commands to system commands.
* monkey - monkey is the testing tool.
* v - v stands for verbose method.
* 100- it is the frequency conut or the number of events to be sent for testing.

This is also shown in the figure:



In the above command, you run the monkey tool on the default android UI application. Now in order to run it to your application , here what you have to do.

First run the example code given in the example section in your device. After running , follow the steps of monkey usage and finally type this command.

adb shell monkey -p com.example.test -v 500

e.g

adb shell monkey –p adb shell monkey -p com.maheshwar.example -v 100

This has also been shown in the figure below. By typing this command , you are actually generating 500 random events for testing.

The below example demonstrates the use of Testing. It crates a basic application which can be used for monkey.

To experiment with this example , you need to run this on an actual device and then follow the monkey steps explained in the beginning.

Create Simple Android app with Two activity in eclipse or Android studio for simple Testing the Android app with Monkey Test Runner on Windows using Command Tool.

Before Run the TestCase we have Run the App on emulator.

Then execute the above command by replacing the Package name with your package name.

Usage of Monkey Options This are as fallows

usage: monkey [-p ALLOWED\_PACKAGE [-p ALLOWED\_PACKAGE] ...]

[-c MAIN\_CATEGORY [-c MAIN\_CATEGORY] ...]

[--ignore-crashes] [--ignore-timeouts]

[--ignore-security-exceptions]

[--monitor-native-crashes] [--ignore-native-crashes]

[--kill-process-after-error] [--hprof]

[--pct-touch PERCENT] [--pct-motion PERCENT]

[--pct-trackball PERCENT] [--pct-syskeys PERCENT]

[--pct-nav PERCENT] [--pct-majornav PERCENT]

[--pct-appswitch PERCENT] [--pct-flip PERCENT]

[--pct-anyevent PERCENT] [--pct-pinchzoom PERCENT]

[--pkg-blacklist-file PACKAGE\_BLACKLIST\_FILE]

[--pkg-whitelist-file PACKAGE\_WHITELIST\_FILE]

[--wait-dbg] [--dbg-no-events]

[--setup scriptfile] [-f scriptfile [-f scriptfile] ...]

[--port port]

[-s SEED] [-v [-v] ...]

[--throttle MILLISEC] [--randomize-throttle]

[--profile-wait MILLISEC]

[--device-sleep-time MILLISEC]

[--randomize-script]

[--script-log]

[--bugreport]

[--periodic-bugreport]

COUNT