Monkey Test

Standard Operation Procedure

Customer Support

On the Android devices

Doc No:	
Version:	V1.0
Release date:	2016-08-29
Classification:	Standard

Specifications are subject to change without notice.

Keywords

System Test Environment

MediaTek Inc.

Postal address

No. 1, Dusing 1st Rd. , Hsinchu Science
Park, Hsinchu City, Taiwan 30078

MTK support office address

No. 1, Dusing 1st Rd. , Hsinchu Science
Park, Hsinchu City, Taiwan 30078

Internet

http://www.mediatek.com/

Document Revision History

Revision	Date	Author	Description
V1.0	2016-09-25	Jinghe Zhan	Initial Release from DS6000-D5A-DMT-V1.2EN

Table of Contents

Document Revision History....3
Table of Contents....4
Lists of Tables....6
Lists of Figures....7
1 Introduction....8
1.1 Purpose....8
1.2 Scope....8
1.3 Who Should Read This Document....8
1.4 How to Use This Manual....8
1.4.1 Terms and Conventions....9
2 References....10
3 Definitions....11
4 Abbreviations....12
5 Overview....13
6 System Preparation....14
6.1 Device Setup....14
6.1.1 Check configuration/options....14
6.1.2 Enable MTKlogger....14
6.2 Workstation Preparation....14
6.2.1 Checking adb connection to Android device....14
6.2.2 Checking UART connection to Android device....14
7 Use of the Monkey....15
7.1 Where is the monkey....15
7.2 Basic Use of the Monkey....15
7.3 Command Options Reference....15
8 Prerequisite and Collect of Logs....18
8.1 Before Monkey Test....18
8.2 After Monkey Test....18
9 Preliminary log analysis....19

9.1 Locate the when/where is failure 19
9.2 Use E-Consulter to conduct a preliminary analysis 19

Lists of Tables

Table 1-1. Reference Information beyond Scope 8
Table 1-2. Chapter Overview 8
Table 1-3. Conventions 9
Table 4-1. Abbreviations 12
Table 9-1. Sample Table. .... **Error! Bookmark not defined.**
Table 9-2. Sample Register Table .... **Error! Bookmark not defined.**

Lists of Figures

Figure 9-1. Title-Cased Caption Text. Centered. ....Error! Bookmark not defined.
Figure 9-2a. Sleepy. b. Tired..Error! Bookmark not defined.
Figure 9-3. Chipmunks and Flowers and Alligators ....Error! Bookmark not defined.

1 Introduction

1.1 Purpose

This document provides the operation guidelines for Monkey Test. It describes how to run Monkey Test on the Android platform. This manual also elaborates the preparation required to run Monkey Test.

1.2 Scope

The document provide the operation details of Monkey Test.

Table 1-1 presents the reference information of the modules which are used but beyond the scope.

Table 1-1. Reference Information beyond Scope

Modules	Reference information
Audio driver	The Android Multimeda audio driver**Error! Reference source not found.**].

1.3 Who Should Read This Document

This document is primarily intended for:

- QA Engineers who runs Monkey Test
- RD Engineers who debug Monkey Test issue

1.4 How to Use This Manual

This segment explains how information is distributed in this document, and presents some cues and examples to simplify finding and understanding information in this document. Table 1-2 presents an overview of the chapters and appendices in this document.

Table 1-2. Chapter Overview

#	Chapter	Contents
1	Introduction	Describes the scope and layout of this document.

1.4.1 Terms and Conventions

This document uses special terms and typographical conventions to help you easily identify various information types in this document. These cues are designed to simply finding and understanding the information this document contains.

Table 1-3. Conventions

Convention	Usage	Example
[1]	Serial number of a document in the order of appearance in the References topic	Look up Chapter 2: System Architecture in [1]
void xx(zz)	Source code	static int __stdcall cb_download_bloader_init(void *usr_arg){}
☞	Important	

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

[1] UI/Application Exerciser Monkey, https://developer.android.com/studio/test/monkey.html

[2] MTKLogger_Introduction.pptx, http://dms.mediatek.inc/

3 Definitions

For the purposes of the present document, the following terms and definitions apply:

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets.

AOSP (Android Open Source Project): Android has an active community of developers and enthusiasts who use the Android Open Source Project (AOSP) source code to develop and distribute their own modified versions of the operating system.

4 Abbreviations

Please note the abbreviations and their explanations provided in Table 4-1. They are used in many fundamental definitions and explanations in this document and are specific to the information that this document contains.

Table 4-1. Abbreviations

Abbreviations	Explanation
MTK	MediaTek, Asia's largest fabless IC design company.
AOSP	Android Open Source Project
ABTC	ALPS Build Tree Concentration. All intermediate files that are generated during building process will be concentrated in one place (e.g. located at alps/out folder)

5 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.

6 System Preparation

The following style guidelines are for device and workstation prepartion.

6.1 Device Setup

6.1.1 Check configuration/options

The following is suggested to be checked before monkey test running.

No	items	Description
1	Enough storage space for logs	Acctually, we suggest restore factory settings before test
2	Screen timeout	Set Screen timeout to maximum value (Settings → Display → Screen timeout → 30 minutes)
3	USB debugging	Enable USB debugging
4	Unlock screen	Keep Screen unlocked before Monkey Test start
5	UART Connection	Output UART log for ongoing anlysis and debugging

6.1.2 Enable MTKlogger

To avoid log missed, please turn on MTKlogger [2]

6.2 Workstation Preparation

A Windows, Linux or Mac machine with the Android SDK and ADB installed. Windows, Linux and Mac OS X systems can all be used for Monkey startup. Able to receive and record UART log from device.

6.2.1 Checking adb connection to Android device

Once you have an Android device available the adb command is the simplest and fastest way to check whether the Android device can be found. In a command window, at the command line type adb devices. If your device is connected correctly you should see the unique id of the device listed.

```
adb devices

List of devices attached

0123456789ABCDEF
```

6.2.2 Checking UART connection to Android device

Use HyperTerminal to check whether UART log from the Android device can be found.

7 Use of the Monkey

7.1 Where is the monkey

No	Location	Path
1	Source Code	/development/cmds/monkey/src/com/android/commands/monkey/
2	Device	/system/bin/monkey (sh)
3	Device	/system/framework/monkey.jar

7.2 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
```

7.3 Command Options Reference

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

Category	Option	Description
General	`--help`	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`	Inserts a fixed delay between events. You can use this option to slow down the Monkey. If not

	`<milliseconds>`	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. See Traceview for more information on trace files.
	--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.

8 Prerequisite and Collect of Logs

For the ongoing debugging, log keeping and collection are very important.

8.1 Before Monkey Test

No	items	Description
1	Clear logs before testing	To avoid log missed caused by not enough space
2	Enable MTKLogger	Record the whole process during Monkey test
3	Enable UART logging	Use HyperTerminal

8.2 After Monkey Test

No	items	Description
1	Observe/record whether restart during testing	For the reference to the ongoing debugging
2	Pull logs from MTKLogger	Pull the directory of /data/aee_exp and /data/anr as well
3	Use E-Consulter for Preliminary log analysis	http://econsulter.mediatek.com/ http://59.37.126.236:3000/

9 Preliminary log analysis

If we can’t see Monkey Finished at the end of monkey log, it means something wrong happened during testing.

9.1 Locate the when/where is failure

```
** Monkey aborted due to error.

Events injected: 178810

:Sending rotation degree=0, persist=false

:Dropped: keys=23 pointers=74 trackballs=0 flips=0 rotations=0

## Network stats: elapsed time=24532668ms (0ms mobile, 23138615ms wifi, 1394053ms not connected)

** System appears to have crashed at event 178810 of 400000 using seed 0
```

Please search with calendar_time as the keywords, the last record is the last time

```
…

Line 459233:    //[calendar_time:2013-01-24 03:01:23.783  system_uptime:25447119]

Line 459239:    //[calendar_time:2013-01-24 03:01:24.284  system_uptime:25447620]

Line 459517:    //[calendar_time:2013-01-24 03:01:43.569  system_uptime:25466905]
```

9.2 Use E-Consulter to conduct a preliminary analysis

E-Consulter is MediaTek Inc. proprietary design used for automatic log analysis. More detailed, please visit http://econsulter.mediatek.com/