Customer Performance Tool Guideline 1.0

Documents Number: <<No.3>>

Revision History

Change Log

Revision	Date	Author	Change Log
1.0	Mar. 18th, 2019	Fuchuan Luo(罗辅川)	Initial document.
1.0	Apr.17th,2019	Chuan Chen(陈川)	Review and modify

Reviewer's Comments

Date	Reviewer	Comments

Table of Contents

Revision History **1**

Table of Contents **3**

Symbols and Abbreviations **4**

1 Introduction **5**

2 Rapid Screening Tool **7**

2.1 Testcases 7

2.2 Summary 17

2.3 Tool Entry 18

2.4 Configurations 19

2.5 Run Tool 21

3 Systrace Analysis Tool **22**

3.1 Supported Scenarios 22

3.2 Run Tool 25

Appendix I Advanced Configuration **27**

Symbols and Abbreviations

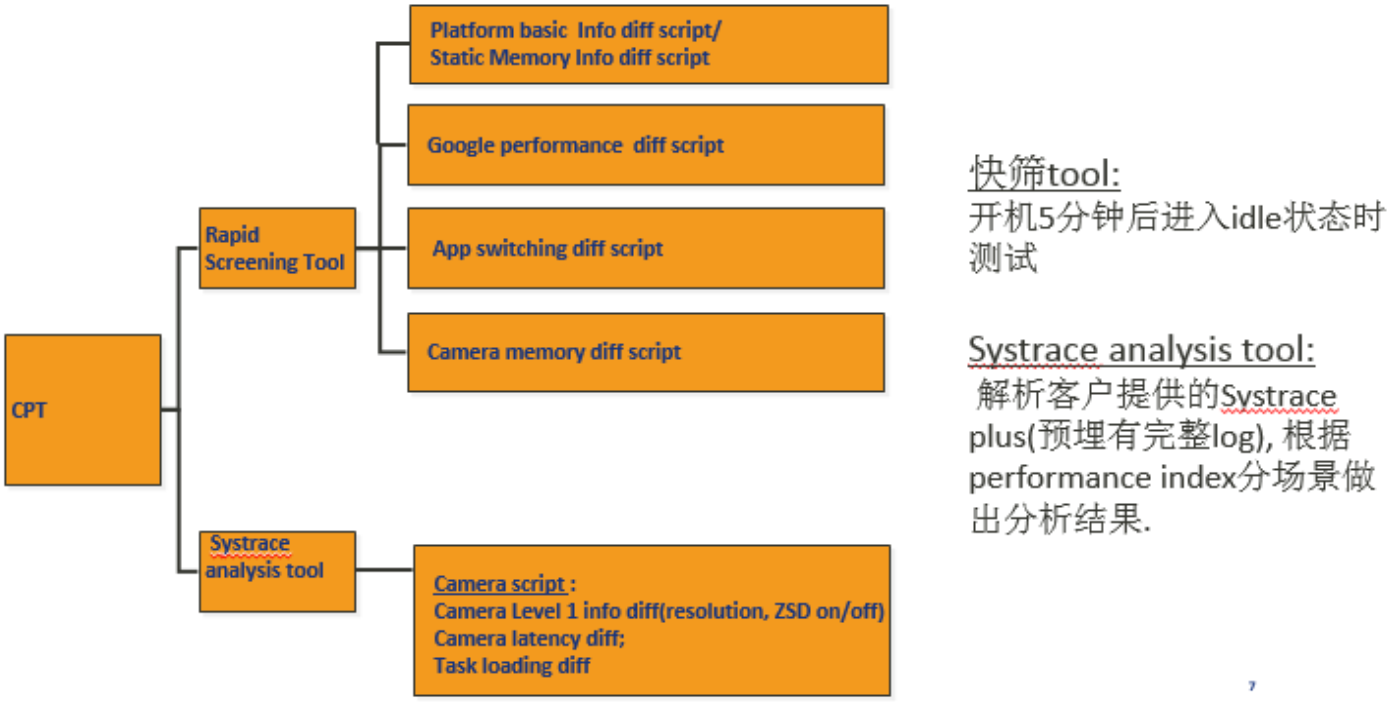
- MemoryTest
  - One of test cases in Rapid Screening Tool which is developed by MTK, it is used to check the memory whether meet Google and MTK memory requirement or not.
- Google Performance
  - One of test cases in Rapid Screening Tool which is converted from Google performance test script, it is used to check the performance whether meet Google requirement.
- App Switch
  - One of test cases in Rapid Screening Tool which is developed by MTK, it is used to test launch time of applications in several different scenarios which contains pure environment, App switch by cold launch and App switch by natural launch, this test case checks the CPU high and memory low healthy management.
- Pure Test
  - Launch time test scenario, make sure there is only one test application installed when do launch test, the test flow is install application, test the cold start launch time;uninstall it, and then test the next application.
- Switch Test
  - Launch time test scenario, all test applications are installed when do swtich test, the test flow is install all applications, launch these applications one by one circularly, make sure all the applications are launch with cold start.
- Switch Test Hot
  - Launch time test scenario which is similar with Switch Test, the difference is Switch Test Hot launch all applications in natural, it means tool do not kill any applications during test.

1 Introduction

Customer Performance Tool(CPT) is a performance issue quick analyzing tool which could quickly distinguish issue caused by customer or by MTK platform. CPT integrates a lot of analysis experiences from MTK engineers, the aim is to improve work efficiency and save resources. CPT consists of 2 parts, they are Rapid Screening Tool and Systrace Analysis Tool.

Run condition:

Windows 7 environment is recommanded, device with user load and **root** permission is better to get enough analysis info, otherwise there may be lack of some analysis info and can not do factory reset action to clean the device test environment.



Rapid Screening Tool is a tool which is used to check the platform basic info, system memory status, camera memory usage peak in several scenarios, Google performance and App switching for Android system. Tool runs several auto test cases and extract raw data on test device, and then produce a formatted, comparable result file for quick analysis of android system issues. There are serveral report files will be generated, such as “thrashing_check.xlsx” and “camera_memory_analysis_report.xlsx”, once you run the report diff option, it will generate “cpt_report.xlsx” and “camera_memory_diff_report.xlsx”. When customers of MTK suffer memory issue or performance issue, they could run this tool to get raw data and reports to find out the issue root cause by himself or send these files to MTK for advanced issue analysis.

Systrace Analysis Tool is a tool which is used to analyze the systrace and some log files produced by systrace record tool, such as Systrace+ and LTR(Long time recording tool). In customer version, it contains camera latency analysis . When customers of MTK suffer some unclear issue likes above, they just need to record the systrace by tool and pass the systrace file to Systrace Analysis Tool, then tool will generate a report containing the analysis result which may help them quickly identify the issue key point.

Before using the Customer Performance Tool you should read this document first to study how to use it.

This document is continuously improved. If you find there is lack of something, you could ask for more content from MTK directly.

2 Rapid Screening Tool

2.1 Testcases

2.1.1 Platform Basic Info Test

- App launch time performance is affected by the platform basic info (ex: RAM size,Screen size,CPU frequency….), which will be detected firstly after booting. We need to compare the result between test phone and reference phone.
- Test flow introduction
  - Reboot the device after initial setup (must not be the first boot after flash), wait for 5 minutes, keep screen on and the device unlocked when measuring memory values and getting the platform's basic informations.
- Sample output

Both thrash_check.xlsx and camera_memory_analysis_report.xlsx are contains these basic informations.

	Test Phone	Reference Phone
ProjectName	k79v1_64	k61v1_32_bsp_1g
GMS	No	No
AndroidGo	No	Yes
ROMSize	128GB	32GB
Resolution	1080x2160(FHD)	720x1280(HD)
AndroidVersion	9(Android P0)	9(Android P0)
GPUFreq	21 525000	2 390000
CpuInfo		
- MaxCpuFreq	2000000	2001000
- CpuNum	8	4
DuraSpeedInfo		
- Enable	Yes	Yes
- 监控CPU/Memory健康	Yes	Yes
- 抑制后台自启	Yes	Yes
- 抑制关联启动	Yes	Yes
- 特殊场景保护	Yes	Yes

2.1.2 Memory Test

Free Memory size directly impacts performance. We will reflect memory status between test phone and reference phone on boot(The timing is same as getting platform basic info). For example,

The following is the 3 layer memory info design. These informations are in thrashing_check.xlsx.

Layer 1: Memory overview

		Test Phone	Reference Phone	Diff	Diff(%)
MemInfo	RAMSize	6GB	1GB	-5.0GB	-83.33%
	MemTotal	5645MB	886MB	-4759.0MB	-84.30%
	Reserved	499MB	138MB	-361.0MB	-72.34%
	GoogleDefaultCacheFree	4767MB	562MB(standard is: 5	-4205.0MB	-88.21%
	Kernel	394MB	161MB	-233.0MB	-59.14%
	Persistent	94MB	100MB	6.0MB	6.38%
	SwapTotal	1023MB	664MB	-359.0MB	-35.09%
	SwapFree	1023MB	645MB	-378.0MB	-36.95%
	Swappiness	100	60	-40.0	-40.00%
	MemAvailable	4681MB	555MB	-4126.0MB	-88.14%
	MemFree	3625MB	14MB	-3611.0MB	-99.61%

Reserved: RAMSize – MemTotal

GoogleDefaultCacheFree: meminfo/MemAvailable + dumpsys/Cached/PrivateDirty + dumpsys/Cached/PrivateClean

Layer 2: Memory break down

Memory Total: xxx (MB)	Test Phone	Rference Phone	Diff	Diff(%)
MTK cache free(L1.MTK))	5058.4 MB	634.9 MB	-4423.5MB	-87.45%
Google default cache free with zram	4767.4 MB	559.4 MB	-4208.0MB	-88.27%
Kernel	231.9 MB	93.7 MB	-138.2MB	-59.59%
>> vmalloc	32.9 MB	11.4 MB	-21.5MB	-65.35%
>> PageTables	30.6 MB	10.8 MB	-19.8MB	-64.71%
>> KernelStack	23.4 MB	7.9 MB	-15.5MB	-66.24%
>> SUnreclaim	76.1 MB	37.6 MB	-38.5MB	-50.59%
>> SReclaimable	46 MB	20.9 MB	-25.1MB	-54.57%
>> Shmem	2.1 MB	1.9 MB	-0.2MB	-9.52%
>> Buffers	20.9 MB	3.2 MB	-17.7MB	-84.69%
Meminfo/Overall Memory Used	896.3 MB	466.4 MB	-429.9MB	-47.96%
>> Mapped	484.7 MB	289.7 MB	-195.0MB	-40.23%
>> AnonPages	411.7 MB	160 MB	-251.7MB	-61.14%
>> SwapTotal	1024 MB	664.1 MB	-359.9MB	-35.15%
>> SwapFree	1024 MB	647.3 MB	-376.7MB	-36.79%
meminfo/MemAvailable	4679 MB	551.8 MB	-4127.2MB	-88.21%

>> Active(file)	244.6 MB	269 MB	24.4MB	9.98%
>> Inactive(file)	888.8 MB	270.2 MB	-618.6MB	-69.60%
>> wmark_low	40.5 MB	4.4 MB	-36.1MB	-89.14%
>> meminfo/MemFree	3620.8 MB	13.1 MB	-3607.7MB	-99.64%
dumpsys/Cached	143.2 MB	34.7 MB	-108.5MB	-75.77%
meminfo/Cached	1119.2 MB	539.7 MB	-579.5MB	-51.78%
Unevictable	5.2 MB	2.4 MB	-2.8MB	-53.85%
Active(anon)	411.6 MB	85.5 MB	-326.1MB	-79.23%
Inactive(anon)	1.7 MB	76 MB	74.3MB	4370.59%
Extra_Available_Size	175.7 MB	49.2 MB	-126.5MB	-72.00%
ZRAM	0 MB	5.9 MB	5.9MB	0.00%
ZRAM Ratio	3.2	1.75	-1.5	-46.88%
HW Memory	276.2 MB	81 MB	-195.2MB	-70.67%
>> 3D Memory Usage	25.2 MB	20.7 MB	-4.5MB	-17.86%
>> ION Memory Usage	164.6 MB	37.3 MB	-127.3MB	-77.34%
>> Misc(Other)	86.3 MB	23 MB	-63.3MB	-73.35%
Persistent	94.4 MB	100.3 MB	5.9MB	6.25%
ProcessMemory(adj<0)	732.5 MB	440.4 MB	-292.1MB	-39.88%
ProcessMemory(adj=0)	61 MB	1.5 MB	-59.5MB	-97.54%
ProcessMemory(adj 1~2)	38.6 MB	18.1 MB	-20.5MB	-53.11%
ProcessMemory(adj 3~8)	0 MB	0 MB	0.0MB	0.00%
ProcessMemory(adj>=9)	142.6 MB	33.6 MB	-109.0MB	-76.44%
BG_Process_PSS_and_pSWAP	142.7 MB	32.9 MB	-109.8MB	-76.94%
>> BG_Process_PSS	142.7 MB	32 MB	-110.7MB	-77.58%
>> BG_Process_pSwap	0 MB	0.9 MB	0.9MB	0.00%
Reserved	498.3 MB	137.8 MB	-360.5MB	-72.35%
>> SCP	19 MB	9 MB	-10.0MB	-52.63%
>> SSPM	1 MB	1 MB	0.0MB	0.00%
>> KernelCode	14.4 MB	13 MB	-1.4MB	-9.72%
>> KernelData	15.8 MB	11.1 MB	-4.7MB	-29.75%
>> PageArray	96 MB	9 MB	-87.0MB	-90.62%
>> KernelOther	78 MB	1.1 MB	-76.9MB	-98.59%
>> FB	31.6 MB	12.6 MB	-19.0MB	-60.13%
>> MD	112.4 MB	70 MB	-42.4MB	-37.72%
>> Shared	9 MB	1.4 MB	-7.6MB	-84.44%
>> CCBBuffer	22 MB	0 MB	-22.0MB	-100.00%
>> HwOther	108.2 MB	11.1 MB	-97.1MB	-89.74%

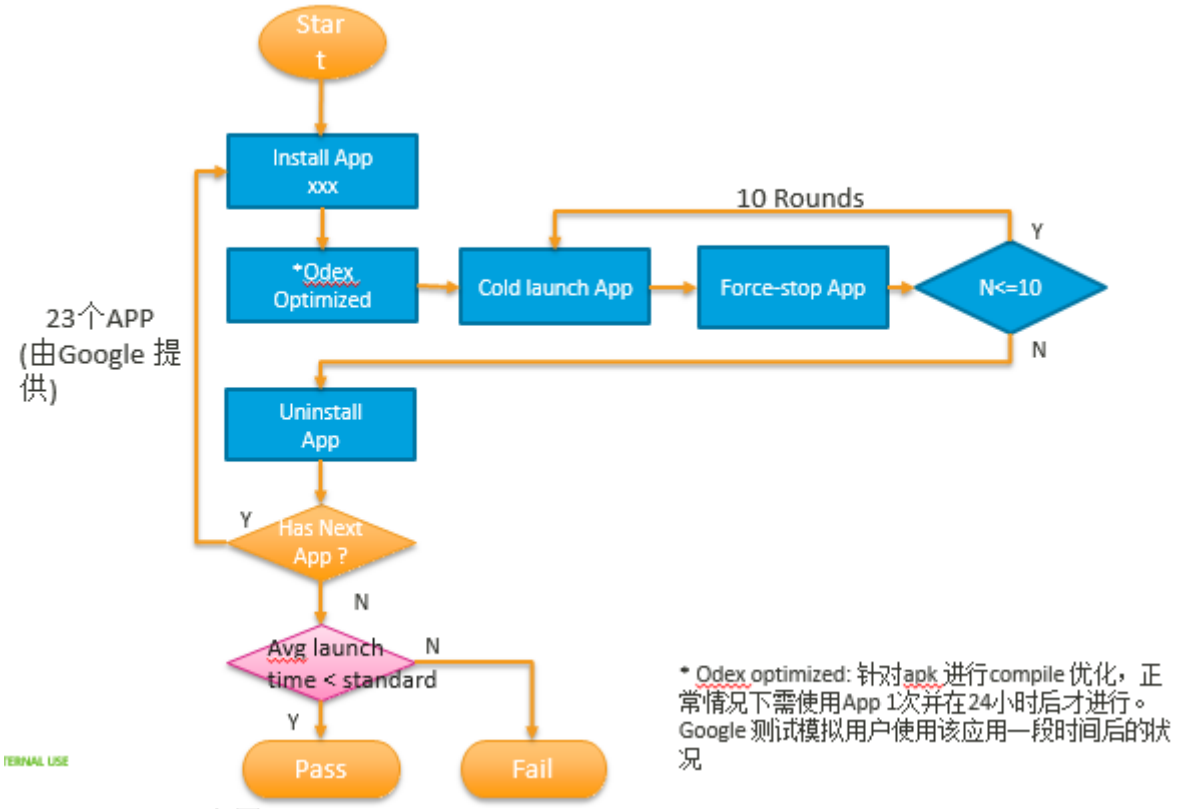
Layer 3: Process info

Process Info	Test Phone		Reference Phone		Adj Diff	Diff	Diff(%)
/init	-1000	3816	-1000	2683	N	-1133.0	-29.69%
/sbin/ueventd	-1000	1380	-1000	900	N	-480.0	-34.78%
/system/bin/GoogleOtaBinder	-1000	952			Y	952	
/system/bin/adbd	-1000	1510	-1000	1166	N	-344.0	-22.78%
/system/bin/aee_aed	-1000	1379	-1000	1029	N	-350.0	-25.38%
/system/bin/aee_aed64	-1000	1358			Y	1358	
/system/bin/audioserver	-1000	12231	-1000	10384	N	-1847.0	-15.10%
/system/bin/batterywarning	-1000	798			Y	798	
/system/bin/cameraserver	-1000	4851	-1000	3815	N	-1036.0	-21.36%
/system/bin/connsyslogger	-1000	1072	-1000	808	N	-264.0	-24.63%
/system/bin/drmserver	-1000	6773	-1000	4248	N	-2525.0	-37.28%
/system/bin/em_svr	-1000	5946			Y	5946	
/system/bin/emdlogger1	-1000	1615	-1000	1168	N	-447.0	-27.68%
/system/bin/gatekeeperd	-1000	1577	-1000	1163	N	-414.0	-26.25%
/system/bin/ged_srv	-1000	1723			Y	1723	
/system/bin/hw/android.hidl.allocator@1.0-service	-1000	743	-1000	567	N	-176.0	-23.69%
/system/bin/hwservicemanager	-1000	1721	-1000	1134	N	-587.0	-34.11%
/system/bin/incidentd	-1000	1311	-1000	982	N	-329.0	-25.10%
/system/bin/installd	-1000	2037	-1000	1382	N	-655.0	-32.16%
/system/bin/ip6tables-restore	-1000	1278	-1000	691	N	-587.0	-45.93%

2.1.3 Google Performance Test

This test case is used to check pure cold launch time of applications specificed by Google whether match Google Android Go performance requirement. If not match, you should optimize your platform until match the Google requirement, otherwise your platform can't pass the Google certification. At the same time, you can also evaluate your platform performance status via Googe performance on non Andoird Go project.

- Test flow introduction
  - Install one application and measure the cold launch times (the Apps is provided by Google performance script) 10 times, and then get the average code lauch time. Install another one application to repeat again. After all applications test done, calculate the total average time of all applications.
  - Flow chart



- Sample output :these informations are in thrashing_check.xlsx

Test condition: cold launch apps (the apps provided by google performance script) 10 times, calculate the average time.					
PackageName	ActivityName	Test Phone	Reference Phone	Diff	Diff(%)
com.adobe.reader	com.adobe.reader.AdobeReader	326	742	416.0	127.61%
com.airbnb.android	.authentication.ui.LoginActivity	412	936	524.0	127.18%
com.bbm	com.bbm.ui.activities.StartupActivity	792	1814	1022.0	129.04%
com.cam001.selfie	.MainActivity	504	1050	546.0	108.33%
com.domobile.applock	.GuideActivity	334	885	551.0	164.97%
com.facebook.katana	.LoginActivity	600	1522	922.0	153.67%
com.facebook.lite	com.facebook.lite.MainActivity	483	1170	687.0	142.24%
com.facebook.orca	com.facebook.messaging.auth.StartScreenAc	474	1099	625.0	131.86%
com.imo.android.imoim	.activities.SignupActivity2	297	713	416.0	140.07%
com.instagram.android	com.instagram.android.activity.MainTabActiv	404	947	543.0	134.41%
com.opera.mini.native	com.opera.mini.android.Browser	355	770	415.0	116.90%
com.outfit7.mytalkingangelafree	com.outfit7.mytalkingangela.MyTalkingAnge	738	1595	857.0	116.12%
com.picsart.studio	com.socialin.android.photo.picsinphoto.Main	599	1307	708.0	118.20%
com.pinterest	com.pinterest.activity.PinterestActivity	271	629	358.0	132.10%
com.quora.android	.components.activities.FullScreenActivity	476	1055	579.0	121.64%
com.skype.raider	com.skype.raider.Main	351	723	372.0	105.98%
com.snapchat.android	.app.main.activity.LoginAndSignupActivity	453	973	520.0	114.79%
com.viber.voip	.SplashActivity	422	1026	604.0	143.13%
com.vkontakte.android	com.vkontakte.android.MainActivity	314	784	470.0	149.68%
com.whatsapp	com.whatsapp.registration.EULA	334	731	397.0	118.86%
me.pou.app	.App	993	2135	1142.0	115.01%
AVG time		472	1076	604.0	127.97%

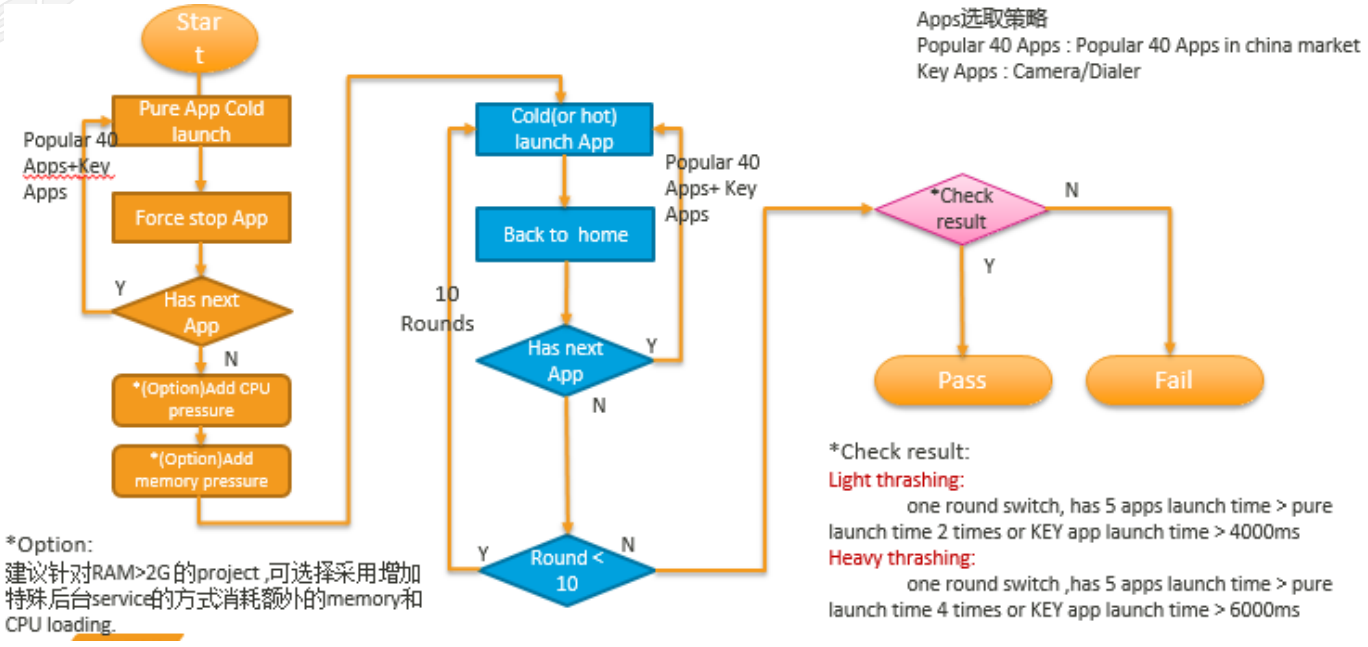
For Android Go project, Google doesn't release the explicit performance standard. But you can ask for help from MTK, because MTk has the pass experience for kinds of platforms . If your platform meets the MTK standard, it could pass the Google certification also.

For non Android Go project, there is no Google performance requirement.

2.1.4 App Switch Test

This test case is a heavy loading performance test which is used to check whether the device has the CPU high and memory low healthy management when there are a lot of applications run in background.

- Test flow introduction
  - Step 1, To run every App when there is no other Apps running in background, repeat 3 times and then get the average of cold launch time for every App, it calls pure test launch time.
  - Step 2, Switch the Apps when there are so many Apps running in background, then get the **cold** launch time or **natual(maybe cold or hot)** launch time, it depends on the test configuration. You can repeat 10 rounds or other specified rounds. Before switching, tool will add some CPU or memory pressures if necessary to emulate CPU high or memory low status.
  - Step 3, compare the launch times got by step 2 to the pure cold launch time got by stpe1, and then check the CPU high and memory low healthy management result.
  - Flow chart



- We define 3 levels for CPU high and memory low healthy management:

- Level 1(Light thrashing): If the result is that more than five consecutive Apps are between 200% and 400% compared to pure ,or the key App (Camera/Dialer)launch time > 4s, then we consider it as light thrashing level.
- Level 2(Heavy thrashing): If the result is that more than five consecutive Apps are more than 400% compared to pure,or the key App(Camera/Dialer) launch time> 6s, then we consider it as heavy thrashing level
- Level 3(Normal): Others.

If the level is 3, then pass the testcase, otherwise fail indicating your platform can't manage the CPU high and memory low healthily .

- Sample output,these informations are in thrashing_check.xlsx
  - App Switch cold start test result:

Test condition: after add memory pressure and CPU pressure, then switch 40 popular apps and key apps(camera/dialer), calculate the cold launch time.											
Package	**Pure(ms)**	**Round 1(**	**Round 2(**	**Round 3(**	**Round 4(**	**Round 5(**	**Round 6(**	**Round 7(**	**Round 8(**	**Round 9(**	**Round 10(ms)**
com.tencent.android.qqdownloader	349	100%	136%	127%	132%	137%	123%	126%	120%	133%	128%
com.tencent.mm	355	100%	108%	114%	110%	105%	119%	114%	107%	113%	111%
com.tencent.mobileqq	188	107%	130%	113%	118%	122%	126%	116%	118%	123%	114%
com.snda.wifilocating	332	108%	115%	120%	107%	104%	115%	116%	112%	103%	112%
com.tencent.mtt	204	111%	125%	128%	127%	128%	122%	123%	121%	148%	119%
com.taobao.taobao	851	114%	121%	121%	123%	121%	114%	114%	114%	111%	120%
com.tencent.qqlive	512	104%	124%	118%	114%	119%	121%	124%	122%	115%	128%
com.kugou.android	385	113%	117%	130%	120%	151%	139%	142%	127%	124%	133%
com.baidu.searchbox	323	104%	115%	110%	106%	112%	119%	117%	132%	114%	132%
com.eg.android.AlipayGphone	294	111%	112%	115%	111%	114%	120%	113%	136%	115%	118%
com.UCMobile	524	141%	178%	167%	153%	151%	154%	187%	160%	157%	149%
com.tencent.qqmusic	274	117%	120%	117%	113%	121%	126%	113%	122%	125%	122%
com.miui.video	263	111%	121%	122%	117%	122%	119%	119%	120%	122%	121%
com.youku.phone	528	114%	120%	113%	118%	139%	119%	111%	125%	121%	125%
com.baidu.BaiduMap	574	115%	125%	115%	111%	127%	117%	115%	124%	118%	117%
com.sina.weibo	509	146%	148%	142%	142%	160%	137%	137%	146%	148%	157%
com.nuomi	1059	130%	125%	123%	124%	139%	121%	119%	126%	122%	126%
com.ting.mp3.android	489	123%	122%	125%	132%	135%	121%	122%	128%	127%	133%
com.qiyi.video	980	116%	147%	141%	133%	145%	149%	159%	149%	144%	133%
com.tencent.news	1067	161%	130%	129%	123%	126%	134%	137%	137%	133%	132%
Average	**503**	**122%**	**129%**	**126%**	**124%**	**131%**	**127%**	**129%**	**129%**	**127%**	**128%**
App Switch Average	**642**										

  - App switch hot start test result, if app launch is hot start, the background color should be mark as red:

Package	Pure(ms)	Round 1(	Round 2(	Round 3(	Round 4(	Round 5(	Round 6(	Round 7(	Round 8(	Round 9(	Round 10
com.tencent.android.qqdownloade	344	113%	37%	25%	36%	30%	30%	34%	34%	34%	29%
com.tencent.mm	358	45%	34%	32%	30%	27%	32%	30%	30%	28%	31%
com.tencent.mobileqq	192	108%	58%	61%	61%	58%	63%	47%	58%	53%	61%
com.snda.wifilocating	344	104%	26%	34%	31%	34%	30%	34%	33%	34%	29%
com.tencent.mtt	219	100%	62%	58%	58%	60%	57%	57%	60%	56%	59%
com.taobao.taobao	832	107%	18%	14%	14%	14%	15%	13%	16%	14%	16%
com.tencent.qqlive	508	44%	18%	19%	24%	22%	20%	23%	23%	23%	22%
com.kugou.android	370	125%	44%	57%	40%	45%	45%	40%	42%	41%	42%
com.baidu.searchbox	320	68%	36%	30%	33%	30%	33%	29%	31%	31%	36%
com.eg.android.AlipayGphone	284	62%	46%	45%	45%	52%	45%	46%	54%	46%	49%
com.UCMobile	551	124%	28%	31%	30%	30%	28%	30%	27%	27%	31%
com.tencent.qqmusic	278	108%	58%	56%	48%	51%	48%	51%	46%	55%	48%
com.miui.video	268	110%	41%	44%	41%	48%	47%	36%	48%	42%	45%
com.youku.phone	538	113%	26%	33%	26%	24%	25%	29%	28%	29%	28%
com.baidu.BaiduMap	603	111%	34%	28%	24%	25%	25%	27%	25%	27%	28%
com.sina.weibo	528	143%	43%	39%	45%	45%	41%	39%	38%	40%	40%
com.nuomi	1086	116%	21%	20%	18%	16%	17%	17%	17%	17%	17%
com.ting.mp3.android	474	49%	47%	51%	50%	51%	47%	53%	53%	48%	51%
com.qiyi.video	945	95%	34%	28%	29%	37%	27%	27%	31%	35%	27%
com.tencent.news	1079	68%	13%	15%	16%	14%	14%	15%	14%	15%	15%
Average	**506**	**96%**	**31%**	**31%**	**30%**	**30%**	**29%**	**29%**	**30%**	**30%**	**30%**
App Switch Average	**187**										

Compared report of 2 cold start app switch result

	Test Phone	Reference Phone	Diff	Diff(%)
Pure Average	642	2216	1574.0	245.17%
Heavy loading Average	0	0	0.0	0.00%

Unit:ms

Package	Pure-MTK	Round-MTK-1	Round-MTK-2	Round-MTK-3	Round-MTK-4	Round-MTK-5	Round-MTK-6	Round-MTK-7	Round-MTK-8	Round-MTK-9	Round-MTK-10	Pure-Reference	Round-R
com.tencent.android.qqdownloader	349	100%	136%	127%	132%	127%	123%	126%	120%	133%	126%	997	126%
com.tencent.mm	355	100%	106%	114%	110%	105%	119%	114%	107%	115%	111%	911	126%
com.tencent.mobileqq	188	107%	130%	115%	118%	122%	126%	116%	118%	123%	114%	472	246%
com.snda.wifilocating	332	108%	115%	110%	107%	104%	115%	116%	112%	103%	112%	850	127%
com.tencent.mtt	204	111%	125%	125%	127%	126%	122%	123%	121%	148%	119%	480	205%
com.taobao.taobao	551	114%	121%	121%	123%	121%	114%	114%	114%	111%	120%	1941	143%
com.tencent.qqlive	512	104%	124%	115%	114%	119%	121%	124%	122%	115%	125%	1190	142%
com.kugou.android	365	113%	117%	130%	120%	151%	139%	142%	127%	124%	153%	946	204%
com.baidu.searchbox	325	104%	115%	110%	106%	112%	119%	117%	132%	114%	152%	815	199%
com.eg.android.AlipayGphone	294	111%	112%	115%	111%	114%	120%	113%	136%	115%	118%	695	229%
com.UCMobile	524	141%	175%	167%	153%	151%	154%	167%	160%	157%	149%	1340	196%
com.tencent.qqmusic	274	117%	120%	117%	115%	121%	126%	115%	122%	125%	122%	659	133%
com.miui.video	265	111%	121%	122%	117%	122%	119%	119%	120%	122%	121%	635	160%
com.youku.phone	525	114%	120%	115%	115%	139%	119%	111%	125%	121%	125%	1223	156%
com.baidu.BaiduMap	574	115%	125%	115%	111%	127%	117%	115%	124%	118%	117%	1326	160%
com.sina.weibo	509	146%	148%	142%	142%	160%	157%	157%	146%	148%	157%	1151	257%
com.nuomi	1059	130%	125%	125%	124%	139%	121%	119%	126%	122%	126%	2291	160%
com.ting.mp3.android	459	125%	122%	125%	132%	135%	121%	122%	126%	127%	133%	1145	197%
com.qiyi.video	930	136%	147%	141%	133%	145%	149%	139%	142%	144%	133%	2319	143%
com.tencent.news	1067	161%	130%	129%	125%	126%	134%	137%	137%	135%	132%	2490	200%
Average	503	122%	128%	124%	124%	131%	127%	129%	129%	127%	128%	1192	174%
App Switch Average	642											2216	
DDR_freq	0											0	

The level of this example is Normal level.

Pure: App cold launch time in pure .

Round1~10: App switch launch time divided by pure launch time, if performance of App launch time reduced above 2 times or slow than 10s, it marked in red color.

2.1.5 Camera Memory Peak Test

Camera memory peak usually means the maximum usage. In worst case, Low memory killer(LMK) will kill camera itself when peak usage is too much.

To grab the peak usage, tool need monitor memory usage as soon as possible then offline post processing the record log.

In here, we define camera memory usage scope should be total of three part: Camera HAL + AOSP cameraserver + Camera APP

Tool will monitor two kind of memory source, total peak means sum of those two kinds memory sources at same timing.

1. Process PSS usage– HAL/SRV/APP via /proc/xxx/smaps
2. ION total usage – HAL/SRV/APP and others (i.e. background app, notification bar, … usually, others is small)

Memory peak usage should be happen during any camera operation (UI feature switch on/off, capture/preview/recording….)

For simplify the tool, based on experience, we know the peak usage usually appear at Capture scenario:

Here we define some basic scenarios to monitor:

1) Home screen idle: to know the overhead of camera memory standby at background
2) Preview: to know when camera launch, how many memory should be allocated?
3) Capture: to know worst memory usage

After recording the memory log, tool will analyze the smaps and ION log file. Provide detail sorted break down for RD analysis.

- Sample output, these informations are in camera_memory_analysis_report.xlsx

Peak usage: xxxx mb (unit:mb)

Memory peak usage: (unit:MB)

	HAL	SRV	APP	ION
stay_by	61.4	6.2	40.8	568.5
preview	56.8	5.8	33.8	494
take_picture	60.8	6.4	41	586.6

Process PSS breakdown:

Take picture - HAL - smaps

take_picture - HAL - smaps	(unit:KB)	(unit:MB)
Total	50590	49.4
anon_inode:dmabuf	10812	10.6
/vendor/lib64/hw/vendor.mediatek.hardware.camera.lomoeffect@1.0-impl.so	5748	5.6
/vendor/lib64/libcameracustom.so	3996	3.9
/vendor/lib64/libcam.hal3a.v3.so	3788	3.7
/vendor/lib64/libmtkcam_hwnode.so	1664	1.6
/vendor/lib64/libmtkcam.featurepipe.streaming.so	1344	1.3
/vendor/lib64/libmtkcam_modulefactory_custom.so	1332	1.3
/vendor/lib64/libmtkcam.featurepipe.capture.so	1024	1
/vendor/lib64/libimageio_plat_drv.so	904	0.9
/vendor/lib64/libmtkcam_fdvt.so	784	0.8
/vendor/lib64/libmtkcam_metastore.so	536	0.5
/vendor/lib64/libdpframework.so	500	0.5
/vendor/lib64/libmtkcam_device3_app.so	460	0.4
/vendor/lib64/libmtkcam_pipeline.so	448	0.4
/vendor/lib64/libcam.halsensor.so	428	0.4
/vendor/lib64/libfeature.stereo.provider.so	420	0.4
/vendor/lib64/libcam.iopipe.so	404	0.4

ION breakdown:

take_picture - ion				
Client Detail	Item	Total (unit: KB)	Block Size(unit: Byte)	Block count
gralloc	32554752x-BLOB-Hal:Image:P1:Fullraw_main1	413296	32555008	13
gralloc	1920x1456-FG_BAYER10-Hal:Image:P1:Resizeraw_mai	66560	5242880	13
gralloc	SurfaceTexture-0-6222-0	32400	3317760	10
gralloc	29319948x-BLOB-capturenr	28636	29323264	1
gralloc	FramebufferSurface	27540	9400320	3
gralloc	ImageReader-4096x2304f100m5-6222-3	18540	18984960	1
gralloc	4096x2304-YUY2-Hal:Image:YuvJpeg	18432	18874368	1
from_kernel	com.mediatek.came_iatek.camera.CameraLauncher#0	18360	9400320	2
gralloc	4656x3496-YV12-CapturePipe	15896	16277504	1
gralloc	4656x3496-I420-CapturePipe	15896	16277504	1
gralloc	gralloc	13248	4521984	3
gralloc	com.android.systemui.ImageWallpaper#0	10532	10784768	1
gralloc	com.mediatek.came_iatek.camera.CameraLauncher#0	9180	9400320	1
gralloc	gralloc	9180	9400320	1
gralloc	4656x3496-I420-CapturePipe	7952	4071424	2
gralloc	4656x3496-YV12-CapturePipe	7952	4071424	2
from_kernel	from_kernel	6836	7000064	1
gralloc	gralloc	5124	5246976	1
gralloc	384x384-STA_2BYTE-Hal:Image:LCSraw_main1	3744	294912	13

Diff report, these informations are in camera_memory_diff_report.xlsx

Memory peak usage: (unit:MB)

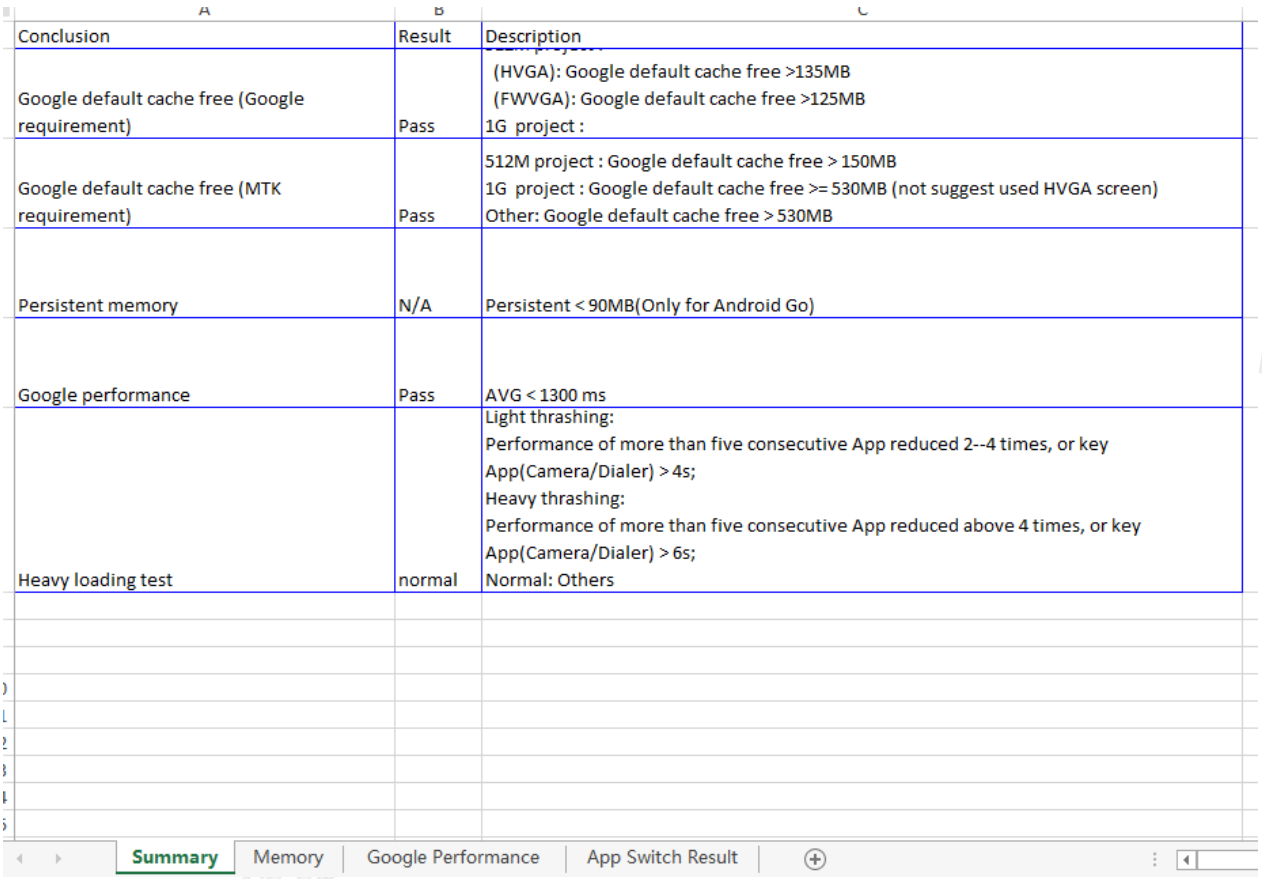
	HAL				SRV				APP				ION			
Scenario	Test Phone	Ref Phone	Diff	Diff(%)	Test Phone	Ref Phone	Diff	Diff(%)	Test Phone	Ref Phone	Diff	Diff(%)	Test Phone	Ref Phone	Diff	Diff(%)
stay_by	61.4	63.2	-1.8	-2.85%	6.2	4.6	1.6	34.78%	40.8	0	40.8		568.5	0	568.5	
preview	56.8	92.3	-35.5	-38.46%	5.8	5.8	0	0.0%	33.8	0	33.8		494	263.6	230.4	87.41%
take_picture	60.8	100.4	-39.6	-39.44%	6.4	6.3	0.1	1.59%	41	38.6	2.4	6.22%	586.6	274.8	311.8	113.46%

take_picture-HAL-smaps	Test Phone		Ref Phone		Diff		Diff(%)
	(unit:KB)	(unit:MB)	(unit:KB)	(unit:MB)	(unit:KB)	(unit:MB)	diff/ref
Total	50590	49.4	82726	80.8	-32136	-31.4	-38.85%
anon_inode:dmabuf	10812	10.6	9576	9.4	1236	1.2	12.91%
/vendor/lib64/hw/vendor.mediatek.hardware.camera.lomoeffect@1.0-impl.so	5748	5.6	32	0	5716	5.6	17862.5%
/vendor/lib64/libcameracustom.so	3996	3.9	4750	4.6	-754	-0.7	-15.87%
/vendor/lib64/libcam.hal3a.v3.so	3788	3.7	3176	3.1	612	0.6	19.27%
/vendor/lib64/libmtkcam_hwnode.so	1664	1.6	1648	1.6	16	0	0.97%
/vendor/lib64/libmtkcam.featurepipe.streaming.so	1344	1.3	1348	1.3	-4	0	-0.3%
/vendor/lib64/libmtkcam_modulefactory_custom.so	1332	1.3	16	0	1316	1.3	8225.0%
/vendor/lib64/libmtkcam.featurepipe.capture.so	1024	1	860	0.8	164	0.2	19.07%
/vendor/lib64/libimageio_plat_drv.so	904	0.9	1008	1	-104	-0.1	-10.32%
/vendor/lib64/libmtkcam_fdvt.so	784	0.8	68	0.1	716	0.7	1052.94%
/vendor/lib64/libmtkcam_metastore.so	536	0.5	328	0.3	208	0.2	63.41%
/vendor/lib64/libdpframework.so	500	0.5	539	0.5	-39	0	-7.24%
/vendor/lib64/libmtkcam_device3_app.so	460	0.4	432	0.4	28	0	6.48%
/vendor/lib64/libmtkcam_pipeline.so	448	0.4	444	0.4	4	0	0.9%
/vendor/lib64/libcam.halsensor.so	428	0.4	252	0.2	176	0.2	69.84%

take_picture-ion			Test Phone		Ref Phone		Diff		Diff(%)
Client Detail	Item	Block Size(unit: Byte)	Total (unit: KB)	Block count	Total (unit: KB)	Block count	Total (unit: KB)	Block count	total diff/ref
gralloc	32554752x-BLOB-Hal:Image:P1:Fullraw_main1	32555008	413296	13					
gralloc	1920x1456-FG_BAYER10-Hal:Image:P1:Resizeraw	5242880	66560	13	66560	13	0	0	0.0%
from_kernel	from_kernel	7000064	6836	1					
gralloc	gralloc	5246976	5124	1	5124	1	0	0	0.0%
gralloc	384x384-STA_2BYTE-Hal:Image:LCSraw_main1	294912	3744	13	3744	13	0	0	0.0%
DpIonHandler	DpIonHandler	528384	3096	6					
gralloc	gralloc	118784	2900	25					
gralloc	gralloc	2351104	2296	1					
gralloc	640x360-YUY2-Hal:Image:FD	462848	2260	5	2260	5	0	0	0.0%
StdIMemDrvImp.cpp	StdIMemDrvImp.cpp	2289664	2236	1					
gralloc	gralloc	528384	2064	4					
gralloc	gralloc	1048576	2048	2	2048	2	0	0	0.0%
gralloc	gralloc	1966080	1920	1					
gralloc	NavigationBar#0	626688	1224	2	1224	2	0	0	0.0%
gralloc	StatusBar#0	315392	924	3	924	3	0	0	0.0%
gralloc	819200x1-BLOB-FDTempBuf	819200	800	1	800	1	0	0	0.0%
from_kernel	NavigationBar#0	626688	612	1	612	1	0	0	0.0%

2.2 Summary

Show the conlusion for platform basic info, Memory test, Google performance test and App switch test.



Conclusion	Result	Description
Google default cache free (Google requirement)	Pass	(HVGA): Google default cache free >135MB (FWVGA): Google default cache free >125MB 1G project :
Google default cache free (MTK requirement)	Pass	512M project : Google default cache free > 150MB 1G project : Google default cache free >= 530MB (not suggest used HVGA screen) Other: Google default cache free > 530MB
Persistent memory	N/A	Persistent < 90MB(Only for Android Go)
Google performance	Pass	AVG < 1300 ms
Heavy loading test	normal	Light thrashing: Performance of more than five consecutive App reduced 2--4 times, or key App(Camera/Dialer) > 4s; Heavy thrashing: Performance of more than five consecutive App reduced above 4 times, or key App(Camera/Dialer) > 6s; Normal: Others

1. Google default cache free(Google requirement): if pass,means your platform meets Google requirement for Google cache free.
2. Google default cache free(MTK requirement): if pass,means your platform meets MTK requirement for Google cache free.If not, optimize your process to use more smaller memory, or process don't be auto launched after the device reboot, and so on.
3. Persistent: if pass,means your platform meets Google requirement for persistent process PSS. If not, please reduce the persistent process number, or optimize your process to use more smaller memory
4. Google performance:if pass, means your platform meets MTK suggestion performance value to pass the Google performance certification
5. Heavy loading test: show thrashing level for CPU high and memory low managemenmt.

2.3 Tool Entry

Rapid Screening Tool is packed into one .exe file, the tool entry is App_Switch.exe.

Tool have 3 major configure files:

- standard.ini

- stress.ini
- thrashing_check.ini

Standard.ini is used to setup the standards for result check.

Stress.ini is used to setup the extra system stress, such as CPU stress and memory stress.

Thrashing_check.ini is used to setup the test flow, which test case needs to be run and how to run it.

2.4 Configurations

2.4.1 Thrashing_check.ini

Thrashing_check.ini is used to setup the test flow, which test case needs to be run and how to run it, it contains basic configurations of 5 test cases and some advanced configurations. For usual use, you only need take care of the basic configurations likes below, the advanced configurations introduction are in the **Appendix**.

Configuration details:

[MemoryTest]

test=True ;default is True;Switch of memory test case

reboot=True ;default is True;Switch of reboot action before dump memory information, if True, test case will reboot and wait 5 minutes

;for reset the memory useage status, and then dump memory information. If False, test case dump memory

;info directly

[GooglePerformanceTest]

test=True ;default is True, Switch of google performance test

apk_path=D:\Test\apks ;no default path , need you to specify the apk path; Path of test apk folder, apks are come from google, tool contains these apks.

apk_config_path=D:\Test\test_app_config ; no default path , need you to specify the config file path; Path of test apk config, likes apks, tool contains this config file.

test_round=10 ;default is 10 round; The test round of each apk

[AppSwitchTest]

test=True ;default is True;Switch of App switch test

apk_path=D:\Test\testapk ;no default path, need you to specify the apk path;Path of 3rd part test apk folder, you can prepare your own test apk or you can use the apks which is provided by tool.

package_config_path=D:\Test\Config.txt ; no default path, need you to specify the config path; Path of configure file of internal Apps, config the internal App package and launch activity

;which you want to test. Configuration line starts with '#' will be skiped

;Example:

;#com.android.camera/com.android.camera.Camera

;com.android.contacts1/com.android.contacts.activities.TwelveKeyDialer

test_round=10 ;default is 10;App swich test round

[AppSwitchTestHot] ;Similar with [AppSwitchTest]

test=True ; default is True ;Switch of hot App switch test

factory_reset=True ;default is True; Switch for factory_reset, if True, Tool makes device do factory reset for clean up

;the test environment before this test case run.

apk_path=D:\Test\testapk ; no default path, need you to specify

package_config_path= D:\Test\Config.txt ; no default path, need you to specify

test_round=10 ;default is True; App swich test round

[CameraMemoryTest]

Test=True ;default is True;Switch of camera memory peak test

scenario=stay_by, preview, take_picture ;default is stay_by, preview, take_picture;Scenarios need to be checked, you can set one or more scenario;Scenarios currently contains stay_by, preview and take_picture

test_time=10 ;default is 10;The time period need to be check for each scenarios.

2.4.2 Stress.ini

Stress.ini is used to setup the extra system stress, such as CPU stress and memory Stress. Usually this part is no need to be modified, details please reference the Appendix.

2.4.3 Standard.ini

Standard.ini is used to setup the standards for result check. No special requirement, you should not modify this configuration file, details please reference the Appendix.

2.5 Run Tool

After all configurations be setup done, tool can be run by the next commands.

- Run Test
  - App_Switch.exe –s <device serial number>

  Example: App_Switch.exe –s 0123456789ABCDEF
- Diff thrashing report
  - App_Switch.exe --diff-thrashing –report <report path> --ref-report <reference report path>
- Diff camera report
  - App_Switch.exe --diff-camera-memory –report <report path> --ref-report <reference report path>

Also you can run App_Switch.exe –help to check the detail usage.

```
Usage: App_Switch.exe [options]

Options:
  -h, --help            show this help message and exit
  -s SERIAL, --serial-number=SERIAL
                        serial number of device
  -o OUTPUT_PATH, --output-path=OUTPUT_PATH
                        output path
  --diff-thrashing      diff 2 thrashing check report excel file
  --diff-camera-memory  diff 2 camera memory report excel file
  --report=REPORT       report file path
  --ref-report=REF_REPORT
                        reference report file path
```

3 Systrace Analysis Tool

3.1 Supported Scenarios

3.1.1 Camera Latency Analysis

Camera delay needs to match spec.

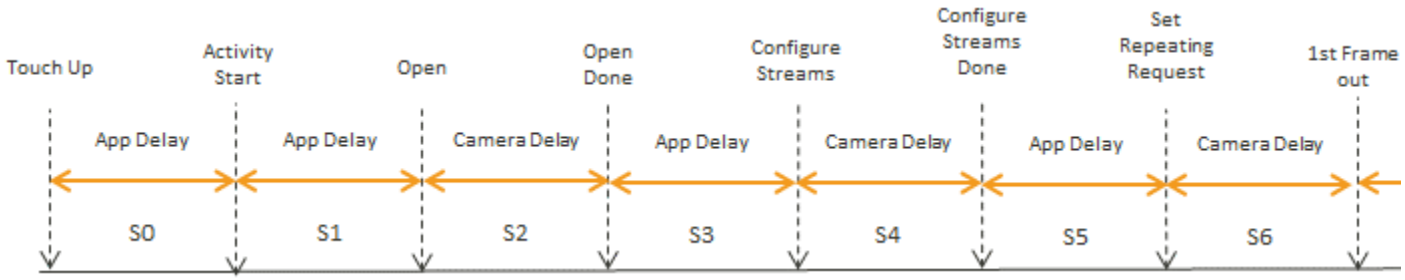
Each test item are separated to several stages, some stages are related to camera App, and others are related to Caemra MW(middleware)

If some stages are longer than reference phone, ask App or middleware(MW) owner for detail analysis.

- Stages

Stage	Behavior
Camera launch	Cold/Warm/hard launch
Camera switch	Main/sub cam switch
Mode switch	Preview/video switch
Take picture	Shot to jpeg

Camera launch:

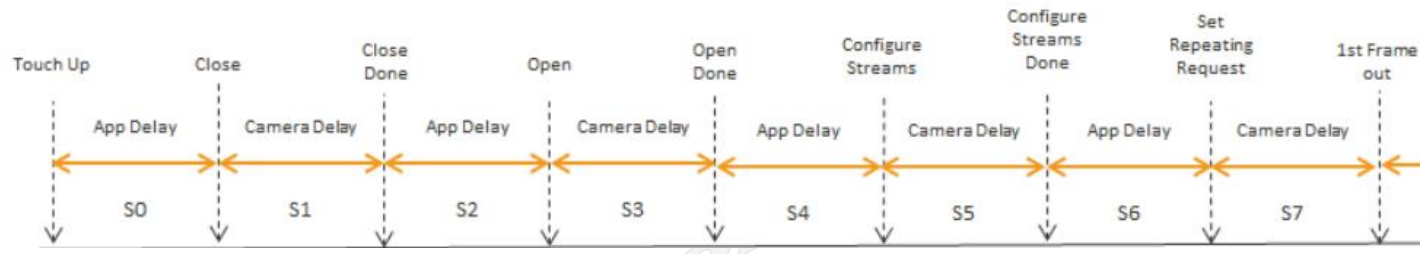


- **If stage X is larger than reference phone, next action**

	Next action
Stage 0+1	**Ask camera APP team to check why** **Input up → start open camera takes so much time**
Stage 2	**Check MW**
Stage 3	**If end of "LogicalDev power On" time < start time of "endConfiure time** **→ ask why app takes so much time after open done**

	If end of "LogicalDev power On" time > start time of "endConfiure time **→ check MW power one time**
Stage 4	**Check MW**
Stage 5	**Check Camera APP**
Stage 6	**Check MW**

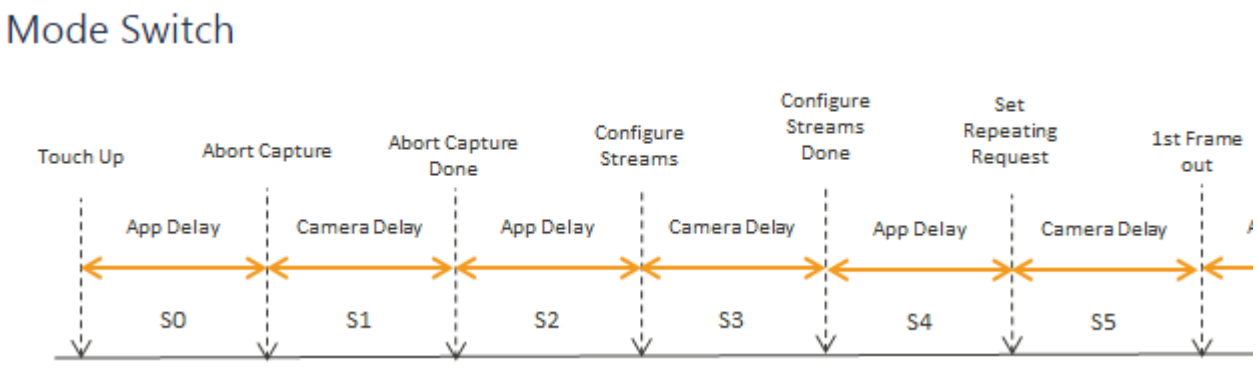
Camera Switch:



- If stage X is larger than reference phone, next action

	Next action
Stage 0	Ask APP to check time between touch up → start close Input up → start open camera takes so much time
Stage 1	Check MW
Stage 2	Ask APP to check time between close done → open camera
Stage 3	Check MW
Stage 4	If end of "LogicalDev power On" time < start time of "endConfiure time → ask why app takes so much time after open done If end of "LogicalDev power On" time > start time of "endConfiure time → check MW power one time
Stage 5	Check MW
Stage 6	Check Camera APP
Stage 7	Check MW

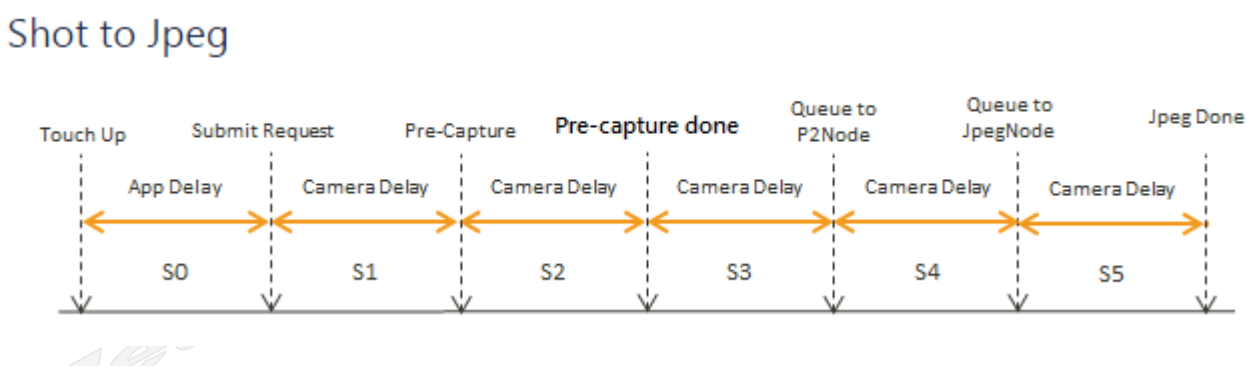
Mode switch:



- If stage X is larger than reference phone, next action

	Next action
Stage 0	Ask APP to check time between touch up → abort capture
Stage 1	Check MW
Stage 2	Ask APP to check time between abort capture done → configure stream
Stage 3	Check MW
Stage 4	Ask APP check time between configure stream done→ set request
Stage 5	Check MW

Take Picture:



	Next action
Stage 0	Ask camera APP team to check time during touch up → submit take picture request
Stage 1	MW: check why take more to do prepare pre-capture
Stage 2	Check MW (3A): Why pre-capture take more time
Stage 3	Check MW: Why enque to P2 after pre-capture done takes more time
Stage 4	Check MW: When P2 takes more time
Stage 5	Check MW: why jpeg encode takes more time

- Sample output

main to sub	s0	s1	s2	s3	s4	s5	s6	s7	total
Test phone	0.029031	0.130729	0.013987	0.016963	0.02081	0.01482	0.034034	0.090503	0.350877
Reference phone	0.029031	0.130729	0.013987	0.016963	0.02081	0.01482	0.034034	0.090503	0.350877
Diff	0	0	0	0	0	0	0	0	0
Diff(%)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

sub to ma	s0	s1	s2	s3	s4	s5	s6	s7	total
Test phone	0.011118	0.120805	0.006229	0.008133	0.020035	0.09248	0.03684	0.10766	0.403298
Reference	0.011118	0.120805	0.006229	0.008133	0.020035	0.09248	0.03684	0.10766	0.403298
Diff	0	0	0	0	0	0	0	0	0
Diff(%)	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Test phone raw data

main to sub	s0	s1	s2	s3	s4	s5	s6	s7	total
1	0.030408	0.127576	0.01241	0.01584	0.019682	0.014031	0.033914	0.091746	0.345607
2	0.027853	0.12252	0.014914	0.018398	0.023736	0.015889	0.032013	0.091622	0.346945
3	0.028831	0.142092	0.014638	0.016651	0.019012	0.014539	0.036176	0.08814	0.360079
Average	0.029031	0.130729	0.013987	0.016963	0.02081	0.01482	0.034034	0.090503	0.350877

Reference phone raw data

main to su	s0	s1	s2	s3	s4	s5	s6	s7	total
1	0.030408	0.127576	0.01241	0.01584	0.019682	0.014031	0.033914	0.091746	0.345607
2	0.027853	0.12252	0.014914	0.018398	0.023736	0.015889	0.032013	0.091622	0.346945
3	0.028831	0.142092	0.014638	0.016651	0.019012	0.014539	0.036176	0.08814	0.360079
Average	0.029031	0.130729	0.013987	0.016963	0.02081	0.01482	0.034034	0.090503	0.350877

sub to main	s0	s1	s2	s3	s4	s5	s6	s7	total
1	0.009497	0.11761	0.005742	0.007918	0.018156	0.093795	0.038228	0.106235	0.397181
2	0.014141	0.119987	0.007447	0.007931	0.021805	0.09204	0.037259	0.106602	0.407212
3	0.009716	0.124818	0.005498	0.008549	0.020143	0.091604	0.035032	0.110142	0.405502
Average	0.011118	0.120805	0.006229	0.008133	0.020035	0.09248	0.03684	0.10766	0.403298

sub to ma	s0	s1	s2	s3	s4	s5	s6	s7	total
1	0.009497	0.11761	0.005742	0.007918	0.018156	0.093795	0.038228	0.106235	0.397181
2	0.014141	0.119987	0.007447	0.007931	0.021805	0.09204	0.037259	0.106602	0.407212
3	0.009716	0.124818	0.005498	0.008549	0.020143	0.091604	0.035032	0.110142	0.405502
Average	0.011118	0.120805	0.006229	0.008133	0.020035	0.09248	0.03684	0.10766	0.403298

Camera analysis result contains task loading result same as launch analysis.

3.2 Run Tool

3.2.1 Tool Options

Run command: Trace_Analysis_Tool.exe --help, the option details description will be printed out.

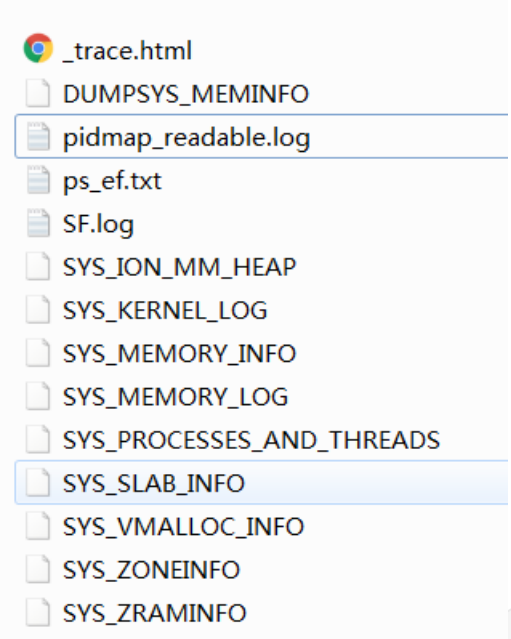
```
Usage: Trace_Analysis_Tool.exe [options]

Options:
  -h, --help            show this help message and exit
  -o OUTPUT_PATH, --output-path=OUTPUT_PATH
                        output path
  --trace=TRACE         trace file path
  --ref-trace=REF_TRACE
                        reference trace file path
  --cluster-types=CLUSTER_TYPES
                        cluster types, example: 62, means large cpu is 2 and
                        small cpu 6
  --ref-cluster-types=REF_CLUSTER_TYPES
                        cluster types, example: 62, means large cpu is 2 and
                        small cpu 6
  --camera-test-times=CAMERA_TEST_TIMES
                        the test repeat times in the trace
  --camera-test-category=CAMERA_TEST_CATEGORY
                        the camera test category, contains: cam_launch,
                        cam_switch, mode_switch, take_pic
```

3.2.2 Run Camera Latency Analysis

- Record traces

Camera systarce record flow is little different from others, it need to open some camera trace tag by particular batch script, which names systrace-setup-bsp-p - need.bat. Before run systrace+ or LTR, please run the setup batch script first, and you need ensure the trace tag

“input, camera” is added, such as “systrace.bat –t 10 input camera”. After systrace+ or LTR starts run, manually execute the particular camera test case 1 or 3 times. Also you can get the trace folder like next



- Command

  D:\Test\AppSwitchTest >Trace_Analysis_Tool.exe

  --camera-analysis

  --cluster-types 44 ;44 means 4 large CPU and 4 small CPU

  --ref-cluster-types 62 ;62 means 2 large CPU and 6 small CPU

  --camera-test-times 1 ;the test case executes time when record systarce

  --camera-test-category mode_switch ;test categories contains cam_launch, cam_switch, mode_switch, take_pic

  --trace D:\target_trace\trace.html

  --ref-trace D:\reference_trace\trace.html

Appendix I Advanced Configuration

- Advanced part of Thrashing_check.ini

```
[AppSwitchTest]

debug_mode=False                ;Switch of debug mode, if debug mode is Ture,
                                ;test case will skip install app and all of other
                                ;setup flow, and launch app directly.
add_pure_test=True              ;Switch of pure test.
pure_test_round=3               ;Pure test round.
add_cpu_stress=False        ;Switch of cpu stress, if True, test case will add extra cpu stress for system.
add_memory_stress=False         ;Switch of memory stress, if True, test case will add extra memory stress for system.
systrace_on=False               ;Switch of systrace, if True, catch the systrace during launch app,
                                ;you and modify the
                                ;TRACE_TAG file for specify the trace tags you are interested in.
memory_dump=False               ;Switch of memory dump, if True,
                                ;tool will dump memory info of current system after test end.
thermal_check=True              ;Switch of thermal check, if True, tool will record the thermal data during test.
```

- Stress.ini

```
[MemoryStress]
service=4               ; Service count
memory=4000             ; Total memory need been eat, unit is M

[CpuStress]
service=1               ;Service count
```

- Standard.ini

```
[KeyAppNormal]                  ;Standard for key app normal thrashing  check
com.mediatek.camera=4000
com.android.dialer=4000
com.tencent.mm=4000
[KeyAppBad]                     ; Standard for key app bad thrashing check
com.mediatek.camera=6000
com.android.dialer=6000
com.tencent.mm=6000

[ComparativeData]               ;Standard for common app
com.sankuai.meituan=12758
com.tencent.news=10735

[ThresholdNormal]
float=200%                      ;The threshold value of float for normal threshing status.
count = 5                       ;Count of apps which ones switch launch time float value are greater than threshold value.

[ThresholdBad]
float=400%                      ; The threshold value of float for bad threshing status.
count = 5                       ; Count of apps which ones switch launch time float value are greater than threshold value.

[GoogleMemory]                  ;standard of google default cache free
512M_HVGA=135
512M_FWVGA=125
1G_HVGA=600
1G_FWVGA=530
1G_QHD=510
```

1G_HD=430

Other=530

[MTKMemory] ;standard of mtk default cache free

512M=220

1G=600

Other=600

[Persistent] ;standard of persistent

Persistent=90

[GooglePerformance] ;standard of google performance

avg=1639 ;1639 is only for android p