

Texas Instruments  
amsdk\_android

---



AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test Report

Project: amsdk\_android

Author: gt\_amsdk\_lead

Printed by TestLink on 07/11/2011

2009 (c) Testlink Community

# Table Of Contents

## Compliance

Google's Compliance Test Suite(CTS) Automated

## Compatibility

## Reference Software

SDK's Calculator App

SDK's LunarLander App

SDK's ApiDemos App

Dalvik's Unit Tests

Apps for android Amazed App

Apps for android AndroidGlobalTime App

Apps for android Clickin2DaBeat App

Apps for android DivideAndConquer App

Apps for android HeightMapProfiler App

Apps for android LOLcat Builder App

Apps for android Photostream App

Apps for android Samples App

Apps for android SpriteMethodTest App

Apps for android Translate App

Apps for android WebViewDemo App

Apps for android WikiNotes App

## Replica Island

## Development Tools

ADB Ethernet

DDMS

Multimedia

Audio

Decode

MP3

Image

Decode

JPEG

PNG

GIF

BMP

Video

Decode

H.264

MPEG4 SP

Performance

System

Browser Launch Time

Simultaneous Applications

Boot time

Quadrant Benchmark

0xBench

0xBench Math Linpack test

0xBench Math Scimark2 test

0xBench 2D Draw Canvas test

0xBench 2D Draw Circle test

0xBench 2D Draw Circle2 test

Table Of Contents

0xBench 2D Draw Rect test

0xBench 2D Draw Arc test

0xBench 2D Draw Image test

0xBench 2D Draw Text test

0xBench 3D OpenGL Cube test

0xBench 3D OpenGL Blending test

0xBench 3D OpenGL Fog test

0xBench 3D OpenGL Flying Teapot test

0xBench VM Garbage Collection test

0xBench Native LibMicro test

0xBench Native UnixBench test

Netperf

TCP Stream, Buffer size 16

TCP Stream, Buffer size 32

TCP Stream, Buffer size 64

TCP Stream, Buffer size 128

TCP Stream, Buffer size 256

TCP Stream, Buffer size 512

TCP Stream, Buffer size 1024

TCP Stream, Buffer size 4096

TCP Stream, Buffer size 8192

Browser

Acid3 tests

Sunspider test

V8 Browser performance test

TheManInBlue Animation

Table Of Contents

RowboPerf

Dhrystone

Whetstone

Linpack

adb

adb USB Performance

adb ethernet Performance

Storage

USB

USB vfat partition write/read test with a block size of 512 bytes and a file of size 104857600 bytes

USB vfat partition write/read test with a block size of 4096 bytes and a file of

USB vfat partition write/read test with a block size of 16384 bytes and a file o

USB vfat partition write/read test with a block size of 65536 bytes and a file o

USB vfat partition write/read test with a block size of 524288 bytes and a file

USB vfat partition write/read test with a block size of 1048576 bytes and a file

MMC/SD

MMC/SD vfat partition write/read test with a block size of 512 bytes and a file

MMC/SD vfat partition write/read test with a block size of 4096 bytes and a file

MMC/SD vfat partition write/read test with a block size of 16384 bytes and a fil

MMC/SD vfat partition write/read test with a block size of 65536 bytes and a fil

MMC/SD vfat partition write/read test with a block size of 524288 bytes and a fi

MMC/SD vfat partition write/read test with a block size of 1048576 bytes and a f

Database

TestIndex

TestIndex Benchmarks

Power

Table Of Contents

DVFS-Performance

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

DVFS-Powersave

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

DVFS-userspace

1MHz

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

800KHz

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

600KHz

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

300KHz

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

DVFS-Ondemand(default)

Idle power performance with FULL WAKE LOCK

Idle power performance with SCREEN BRIGHT WAKE LOCK

Idle power performance with SCREEN DIM WAKE LOCK

Idle power performance with PARTIAL WAKE LOCK

Dhrystone power performance with PARTIAL WAKE LOCK

3D Graphics power performance

Audio + Video power performance

CPU Idle

SUSPEND MODE power consumption sleep while idle disabled and enable off mode disabled

SUSPEND MODE power consumption sleep while idle enabled and enable off mode enabled

SUSPEND MODE power consumption sleep while idle disabled and enable off mode enabled

SUSPEND MODE power consumption sleep while idle enabled and enable off mode disabled

WLAN

Non-secure

WLAN Non-secure, TCP Stream, Buffer size 1024

WLAN Non-secure, TCP Stream, Buffer size 4096

WLAN Non-secure, TCP Stream, Buffer size 8192

WEP 40 bits

WLAN WEP 40 bits, TCP Stream, Buffer size 1024

WLAN WEP 40 bits, TCP Stream, Buffer size 4096

WLAN WEP 40 bits, TCP Stream, Buffer size 8192

WEP 128 bits

WLAN WEP 128 bits, TCP Stream, Buffer size 1024

WLAN WEP 128 bits, TCP Stream, Buffer size 4096

WLAN WEP 128 bits, TCP Stream, Buffer size 8192

WPA-PSK

WLAN WPA-PSK, TCP Stream, Buffer size 1024

WLAN WPA-PSK, TCP Stream, Buffer size 4096



[WLAN WPA-PSK, TCP Stream, Buffer size 8192](#)

[WPA2-PSK](#)

[WLAN WPA2-PSK, TCP Stream, Buffer size 1024](#)

[WLAN WPA2-PSK, TCP Stream, Buffer size 4096](#)

[WLAN WPA2-PSK, TCP Stream, Buffer size 8192](#)

[Stress](#)

[Monkey](#)

[Monkey System Stress](#)

[USB](#)

[Enumeration](#)

[Long-Term Graphics test](#)

[Documentation](#)

[DevKit Users Guide](#)

[Release Notes](#)

[Porting Guide](#)

[CTS Report](#)

[DevKit Test Report](#)

[Android Rowboat Manifest](#)

[Datasheet](#)

[Eclipse Setup](#)

[ADB over Ethernet Setup](#)

[ADB over USB Setup](#)

[ADB .apk File Download](#)

[Eclipse APK File Download](#)

[DevKit Developers Guide](#)

[Document Format](#)

[Table Of Contents](#)

[Packages List](#)

[PinMux Utility Usage](#)

[Serial Flash Utility Usage](#)

[Kitting](#)

[Review DevKit components with the legal team](#)

[Review SW Manifest with OSRB](#)

[DevKit Content](#)

[Android Devkit apk file](#)

[Download Page](#)

[arowboat.org Download Link](#)

[Tools](#)

[Pinmux Utility](#)

[Flashing Utility](#)

[Bootable-MMC/SD-Card-Generation script](#)

[Fastboot utility](#)

[Functionality](#)

[SGX SDK](#)

[SGX Demos](#)

[System](#)

[System boot](#)

[System boot w/ console](#)

[OOB Demos](#)

[RootFS over NFS](#)

[Video](#)

[Hardware-VRFB](#)

[S-Video output](#)

[Table Of Contents](#)

## Graphics

## Maximum Display Resolution

## Bluetooth

## Bluetooth Object push

---

# 1 Test Suite : Compliance

## **Test Case amsdkA-403: Google's Compliance Test Suite(CTS) Automated**

### Summary:

This is to verify platform MUST pass the most recent version of the Android Compatibility Test Suite (CTS) available at the time of the device implementation's software is completed.

### Steps:

- 1) download latest CTS and install on your PC(TEE)
- 2) update this test case parameters like cts\_dir and cts\_res\_dir using your new installation dir.
- 3) assign the test plan you want run(default is CTS) for the variable test\_plan.
- 4) start staf and others.

### Expected Results:

Compliance test must pass with percentage greater than 95.

### Test

execution vatf

### engine:

Test script  
or logic: android/cts/cts\_test.rb

### Required

hardware dut1 = ["<platform>",android]

### assets:

DUT shell\_script=BSP\CTS\cts\_test.bat,cts\_dir=/home/gtadwlan001/android-cts/tools/,cmdline=startcts  
parameters: start --plan, test\_plan=CTS, cts\_res\_dir=/home/gtadwlan001/android-cts/repository/results

### Application

### parameters:

### Test

### Equipment

### parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 2 Test Suite : Compatibility

### 2.1 Test Suite : Reference Software

#### Test Case amsdkA-9: SDK's Calculator App

Summary:

Run Calculator app (from Google's SDK)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### Test Case amsdkA-10: SDK's LunarLander App

Summary:

Run LunarLander app (from Google's SDK)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead

### **Test Case amsdkA-12: SDK's ApiDemos App**

Summary:

Run ApiDemos app (from Google's SDK)

Expected Results:

Application APK is properly installed and runs OK

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead

### **Test Case amsdkA-13: Dalvik's Unit Tests**

Summary:

Run Dalvik VM unit tests (from /dalvik/tests/)

Expected Results:

All Dalvik VM tests passed

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-384: Apps for android Amazed App**

Summary:

Run Amazed app (from <http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-385: Apps for android AndroidGlobalTime App**

Summary:

Run AndroidGlobalTime app (from  
<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

**Test Case amsdkA-387: Apps for android Clickin2DaBeat App**

Summary:

Run Clickin2DaBeat app (from

<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

**Test Case amsdkA-388: Apps for android DivideAndConquer App**

Summary:

Run DivideAndConquer app (from  
<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

**Test Case amsdkA-389: Apps for android HeightMapProfiler App**

Summary:

Run HeightMapProfiler app (from  
<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead



### Test Case amsdkA-390: Apps for android LOLcat Builder App

Summary:

Run LOLcat Builder app (from

<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-392: Apps for android Photostream App

Summary:

Run Photostream app (from

<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

### **Test Case amsdkA-395: Apps for android Samples App**

Summary:  
Run Samples app (from <http://code.google.com/p/apps-for-android/>)  
Expected Results:  
Application APK is properly installed and runs OK  
Test execution  
engine:  
Test script or  
logic:  
Required hardware  
assets:  
DUT parameters:  
Application  
parameters:  
Test Equipment  
parameters:  
test case approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

### **Test Case amsdkA-396: Apps for android SpriteMethodTest App**

Summary:  
Run SpriteMethodTest app (from  
<http://code.google.com/p/apps-for-android/>)  
Expected Results:  
Application APK is properly installed and runs OK  
Test execution  
engine:  
Test script or  
logic:  
Required hardware  
assets:  
DUT parameters:  
Application  
parameters:  
Test Equipment  
parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

#### **Test Case amsdkA-397: Apps for android Translate App**

Summary:

Run Translate app (from <http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

#### **Test Case amsdkA-398: Apps for android WebViewDemo App**

Summary:

Run WebViewDemo app (from  
<http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### **Test Case amsdkA-399: Apps for android WikiNotes App**

Summary:

Run WikiNotes app (from <http://code.google.com/p/apps-for-android/>)

Expected Results:

Application APK is properly installed and runs OK

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### **Test Case amsdkA-233: Replica Island**

Summary:

Run Replica Island Game

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 2.2 Test Suite : Development Tools

### Test Case amsdkA-15: ADB Ethernet

Summary:

Use Android Debug Bridge (adb) tool to connect to the target via ethernet port and install an application (.apk)

Steps:

On the host machine run the following commands from terminal shell:  
\$ export ADBHOST= \$ adb kill-server \$ adb start-server On the target,  
type the following commands to avoid ADBD defaulting to USB  
transport. Restart ADBD to take the changed settings.: # setprop  
service.adb.tcp.port 5555 # stop adbd # start adbd

Expected Results:

adb recognizes the device (adb devices) and can connect to it (adb  
shell)

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-16: DDMS

Summary:

Use Dalvik Debug Monitor Service (DDMS) to watch processes running in the target, see process' threads, etc. Try to capture the device screen and to kill one process using DDMS.

Steps:

It is recommended to install Eclipse and the Android development (ADT) plugin to use DDMS, however it is not mandatory

Expected Results:

DDMS can connect to the device debug data is shown to the user

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 2.3 Test Suite : Multimedia

### 2.3.1 Test Suite : Audio

#### 2.3.1.1 Test Suite : Decode

##### Test Case amsdkA-33: MP3

Summary:

Mono/Stereo 8-320Kbps constant (CBR) or variable bit-rate (VBR) in a MP3 (.mp3) container

Expected Results:

Audio file plays fine

Test execution  
engine: vatf

android/media/media\_player.rb

Test script or  
logic:

Required hardware  
assets: dut1 = ["<platform>",android];server1 = ["linux\_server"]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/video,  
file\_name=Mixed33\_48KHz\_320kbps\_Stereo.mp3, target\_file\_path=  
/mnt/sdcard, testname=MP3,intent=shell am start -W -n  
com.android.music/.MediaPlayerActivity -a  
com.android.music.PLAYBACK\_VIEWER -d

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 2.3.2 Test Suite : Image

### 2.3.2.1 Test Suite : Decode

#### Test Case amsdkA-39: JPEG

Summary:

Display JPEG files using the Gallery app.

Steps:

Use the media app to display .jpg files, if no JPEG files in dut:

- Push a jpeg file to the dut via adb, "adb push <path to jpeg file> /sdcard/Images/<jpef file name>".

- Go to Launcher->Dev tools -> Media Scanner.

- Open the jpeg file with the Gallery app.

Expected Results:

File displays fine

Test execution  
engine: vatf

android/media/media\_player.rb

Test script or  
logic:

Required hardware  
assets: dut1 = ["<platform>",android]; server1 = ["linux\_server"]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/image,  
file\_name=cafeteria\_2048x3172\_420p.jpg, target\_file\_path= /mnt/sdcard,  
testname=JPEG,intent=shell am start -W -n com.cooliris.media/.Gallery -a  
action.intent.ancion.VIEW -d

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-40: PNG

Summary:

Display PNG image with Galllery app.

Steps:

Use the media app to display .png files, if no PNG files in dut:

- Push a .png file to the dut via adb, "adb push <path to png file> /sdcard/Images/<png file name>.
- Go to Launcher->Dev tools -> Media Scanner.
- Open the png file with the Gallery app.

Expected Results:

File displays fine

Test execution  
engine: vatf

Test script or  
logic: android/media/media\_player.rb

Required hardware  
assets: dut1 = ["<platform>",android]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/image, file\_name=o-png24.png,  
target\_file\_path= /mnt/sdcard, testname=PNG

Application  
parameters:

Test Equipment  
parameters:



test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

### Test Case amsdkA-41: GIF

Summary:

Display GIF image with Gallery app.

Steps:

Use the media app to display .gif files, if no GIF files in dut:

- Push a .gif file to the dut via adb, "adb push <path to gif file> /sdcard/Images/<gif file name>."
- Go to Launcher->Dev tools -> Media Scanner.
- Open the gif file with the Gallery app.

Expected Results:

File displays fine

Test execution engine: vatf

Test script or logic: android/media/media\_player.rb

Required hardware assets: dut1 = ["<platform>",android]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/image,  
file\_name=water\_asymmetric\_o-h\_stretch.gif, target\_file\_path= /mnt/sdcard,  
testname=GIF

Application parameters:

Test Equipment parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

### Test Case amsdkA-42: BMP

Summary:

Display BMP Image with Gallery app.

Steps:

Use the media app to display .bmp files, if no BMP files in dut:

- Push a .bmp file to the dut via adb, "adb push <path to bmp file> /sdcard/Images/<bmp file name>".

- Go to Launcher->Dev tools -> Media Scanner.

- Open the bmp file with the Gallery app.

Expected Results:

File displays fine

Test execution  
engine: vatf

Test script or  
logic: android/media/media\_player.rb

Required hardware  
assets: dut1 = ["<platform>", android]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/image, file\_name=blackbuck.bmp,  
target\_file\_path= /mnt/sdcard, testname=BMP

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 2.3.3 Test Suite : Video

### 2.3.3.1 Test Suite : Decode

#### Test Case amsdkA-45: H.264

Summary:

H.264 files in 3GPP (.3gp) and MPEG-4 (.mp4) container

Expected Results:

Video file plays fine

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test execution engine: vatf

Test script or logic: android/media/media\_player.rb

Required hardware assets: dut1 = ["<platform>",android]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/video,  
file\_name=omap4ad1\_h264\_aac\_cif\_288kbps\_63kbps.3gp, target\_file\_path=  
/mnt/sdcard, testname=H264,intent=shell am start -W -n  
com.cooliris.media/.MovieView -a action.intent.ancion.VIEW -d

Application parameters:

Test Equipment parameters:

test case approver:

Last Result: **Passed**

Build: BD\_Gin\_2.3.4.DEVKIT

Tester: gt\_amsdk\_lead

Testing notes: Test case PASS.

### LOG PATH

## Test Case amsdkA-46: MPEG4 SP

Summary:

MPEG4 Simple Profile files in 3GPP (.3gp) container

Expected Results:

Video file plays fine

Test execution engine: vatf

Test script or logic: android/media/media\_player.rb

Required hardware assets: dut1 = ["<platform>",android]

DUT parameters: host\_file\_path=/mnt/gtautoftp/android/data/video,  
file\_name=omap4ad1\_mpeg4\_aac\_cif\_528kbps\_63kbps.3gp,  
target\_file\_path= /mnt/sdcard, testname=MPEG4-SP,intent=shell am start -W  
-n com.cooliris.media/.MovieView -a action.intent.ancion.VIEW -d

Application parameters:

Test Equipment parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Test case PASS.

LOG PATH

## 3 Test Suite : Performance

### 3.1 Test Suite : System

#### Test Case amsdkA-47: Browser Launch Time

Summary:

The launch time is measured as the total time to complete loading the default activity for the application, including the time it takes to start the Linux process, load the Android package into the Dalvik VM, and call onCreate.

Expected Results:

Browser should launch in less than 1300ms

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes 950ms

#### Test Case amsdkA-49: Simultaneous Applications

Expected Results:

When multiple applications have been launched, re-launching an

alreadyrunning application after it has been launched must take less than the original launch time.

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead

### Test Case amsdkA-117: Boot time

Summary:

Measure the time it takes since kernel image starts being downloaded until Android home screen appears.

Steps:

Boot the DUT and measure the boot time.

Expected Results:

Less or equal than previous release

Test execution  
engine:           vatf

Test script or  
logic:            android/boot/boottime\_test.rb

Required hardware  
assets:           dut1 = ["<platform>",android]

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
--------------	---------------

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Test case PASS.

LOG PATH

**Test Case amsdkA-593: Quadrant Benchmark**

Summary:

Install and run aurorasoftworks Quadrant benchamrk

Steps:

Install and run Qudrant, and save the results

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 3.2 Test Suite : 0xBench

**Test Case amsdkA-89: 0xBench Math Linpack test**

Summary:

0xBench Math Linpack test.

Test  
execution vatf  
engine:

Test script or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

### Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

DUT class org.zeroxlab.benchmark.test.BenchmarkTest#testMathLinpack

parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

### Application

parameters:

### Test

### Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing MathLinpack performance data collected successfully

notes

LOG PATH

## Test Case amsdkA-90: 0xBench Math Scimark2 test

### Summary:

0xBench Math Scimark2 test.

### Test

execution vatf  
engine:

Test script android/performance/0xLabBenchmark/zeroxbench\_perf.rb  
or logic:

### Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

DUT class org.zeroxlab.benchmark.test.BenchmarkTest#testMathScimark2

parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

### Application

parameters:

### Test

### Equipment

parameters:

test case

approver:

Last Result: **Passed**

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing MathScimark2 performance data collected successfully  
notes  
LOG PATH

### Test Case amsdkA-91: 0xBench 2D Draw Canvas test

Summary:

0xBench 2D Draw Canvas test.

Test

execution vatf

engine:

Test script  
or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

DUT class org.zeroxlab.benchmark.test.BenchmarkTest#test2DDrawCanvas

parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 2DDrawCanvas performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-92: 0xBench 2D Draw Circle test

Summary:

0xBench 2D Draw Circle test.

Test

execution vatf

engine:



## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test script  
or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required  
hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:

DUT test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark  
class org.zeroxlab.benchmark.test.BenchmarkTest#test2DDrawCircle  
parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application  
parameters:

Test  
Equipment  
parameters:

test case  
approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 2DDrawCircle performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-93: 0xBench 2D Draw Circle2 test

Summary:

0xBench 2D Draw Circle2 test.

Test  
execution vatf  
engine:

Test script  
or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required  
hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:

DUT test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark  
class org.zeroxlab.benchmark.test.BenchmarkTest#test2DDrawCircle2  
parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application  
parameters:

Test  
Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 2DDrawCircle2 performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-94: 0xBench 2D Draw Rect test

Summary:

0xBench 2D Draw Rect test.

Test

execution vatf

engine:

Test script or logic: android/performance/0xLabBenchmark/zerobench\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zerolab.benchmark.test;Benchmark-debug.apk:org.zerolab.benchmark

DUT class org.zerolab.benchmark.test.BenchmarkTest#test2DDrawRect

parameters: org.zerolab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I

\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 2DDrawRect performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-95: 0xBench 2D Draw Arc test

Summary:

0xBench 2D Draw Arc test.

Test

execution vatf

engine:

Test script  
or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

DUT class org.zeroxlab.benchmark.test.BenchmarkTest#test2DDrawArc

parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 2DDrawArc performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-96: 0xBench 2D Draw Image test

Summary:

0xBench 2D Draw Image test.

Test

execution vatf

engine:

Test script  
or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

DUT class org.zeroxlab.benchmark.test.BenchmarkTest#test2DDrawImage

parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test  
Equipment  
parameters:  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing 2DDrawImage performance data collected successfully  
notes  
LOG PATH

### Test Case amsdkA-97: 0xBench 2D Draw Text test

Summary:

0xBench 2D Draw Text test.

Test  
execution vatf  
engine:  
Test script or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb  
Required hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:  
test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark  
DUT class org.zeroxlab.benchmark.test.BenchmarkTest#test2DDrawText  
parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application  
parameters:

Test  
Equipment  
parameters:  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing 2DDrawText performance data collected successfully  
notes  
LOG PATH

### Test Case amsdkA-98: 0xBench 3D OpenGL Cube test

#### Summary:

0xBench 3D OpenGL Cube test.

Test

execution     vatf

engine:

Test script  
or logic:     android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

DUT           class org.zeroxlab.benchmark.test.BenchmarkTest#test3DOpenGLCube

parameters:   org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I

\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build          BD\_Gin\_2.3.4.DEVKIT

Tester        gt\_amsdk\_lead

Testing       3DOpenGLCube performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-99: 0xBench 3D OpenGL Blending test

#### Summary:

0xBench 3D OpenGL Blending test.

Test

execution     vatf

engine:

Test script  
or logic:     android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

DUT           test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchmark

parameters:   class org.zeroxlab.benchmark.test.BenchmarkTest#test3DOpenGLBlending

org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I

\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 3DOpenGLBlending performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-100: 0xBench 3D OpenGL Fog test

Summary:

0xBench 3D OpenGL Fog test.

Test

execution vatf

engine:

Test script or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zeroxlab.benchmark.test;Benchmark-debug.apk:org.zeroxlab.benchma

DUT class org.zeroxlab.benchmark.test.BenchmarkTest#test3DOpenGLFog

parameters: org.zeroxlab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I

\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing 3DOpenGLFog performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-101: 0xBench 3D OpenGL Flying Teapot test

Summary:

0xBench 3D OpenGL Flying Teapot test.

Test

execution     vatf

engine:

Test script  
or logic:     android/performance/0xLabBenchmark/zerobench\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zerolab.benchmark.test;Benchmark-debug.apk:org.zerolab.benchmark

DUT           class org.zerolab.benchmark.test.BenchmarkTest#test3DOpenGLTeapot

parameters:   org.zerolab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build          BD\_Gin\_2.3.4.DEVKIT

Tester        gt\_amsdk\_lead

Testing       3DOpenGLTeapot performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-102: 0xBench VM Garbage Collection test

Summary:

0xBench VM Garbage Collection test.

Test

execution     vatf

engine:

Test script  
or logic:     android/performance/0xLabBenchmark/zerobench\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

DUT test\_libs=BenchmarkTest.apk:org.zerolab.benchmark.test;Benchmark-debug.apk:org.zerolab.benchmark.test  
parameters: class org.zerolab.benchmark.test.BenchmarkTest#testVMGC  
org.zerolab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing VMGC performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-103: 0xBench Native LibMicro test

Summary:

0xBench Native LibMicro test.

Test

execution vatf

engine:

Test script or logic: android/performance/0xLabBenchmark/zerobench\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

test\_libs=BenchmarkTest.apk:org.zerolab.benchmark.test;Benchmark-debug.apk:org.zerolab.benchmark.test  
DUT class org.zerolab.benchmark.test.BenchmarkTest#testNativeLibMicro  
parameters: org.zerolab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead



Testing NativeLibMicro performance data collected successfully  
notes

LOG PATH

### Test Case amsdkA-104: 0xBench Native UnixBench test

Summary:

0xBench Native UnixBench test.

Test  
execution vatf  
engine:

Test script  
or logic: android/performance/0xLabBenchmark/zeroxbench\_perf.rb

Required  
hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:

test\_libs=BenchmarkTest.apk:org.zerolab.benchmark.test;Benchmark-debug.apk:org.zerolab.benchmark  
DUT class org.zerolab.benchmark.test.BenchmarkTest#testNativeUnixBench  
parameters: org.zerolab.benchmark.test/android.test.InstrumentationTestRunner,log\_option=Benchmark:I  
\*:S,res\_file=/sdcard/0xBenchmark.xml

Application  
parameters:

Test  
Equipment  
parameters:

test case  
approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing NativeUnixBench performance data collected successfully  
notes

LOG PATH

## 3.3 Test Suite : Netperf

### Test Case amsdkA-105: TCP Stream, Buffer size 16

Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 16"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=16,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.81 Mb/s for a buffer size of 16.

### Test Case amsdkA-106: TCP Stream, Buffer size 32

Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 32

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=32,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.58 Mb/s for a buffer size of 32.

**Test Case amsdkA-107: TCP Stream, Buffer size 64**

Summary:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 64"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=64,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.56 Mb/s for a buffer size of 64.

### Test Case amsdkA-108: TCP Stream, Buffer size 128

Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 128"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=128,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.61 Mb/s for a buffer size of 128.

## Test Case amsdkA-109: TCP Stream, Buffer size 256

### Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

### Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 256"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=256,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.63 Mb/s for a buffer size of 256.

### Test Case amsdkA-110: TCP Stream, Buffer size 512

#### Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

#### Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 512"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=512,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.47 Mb/s for a buffer size of 512.

### Test Case amsdkA-111: TCP Stream, Buffer size 1024

#### Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

#### Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 1024"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=1024,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 30.0 Mb/s. the avg throughput measured was 22.47 Mb/s for a buffer size of 1024.



**Test Case amsdkA-112: TCP Stream, Buffer size 4096**

Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 4096"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=4096,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Buffer Size Throughput 4096 54.92

LOG PATH

### Test Case amsdkA-113: TCP Stream, Buffer size 8192

#### Summary:

Measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

#### Steps:

1) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

2) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

3) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the default filesystem)

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 8192"

Test execution engine:	vatf
Test script or logic:	android/performance/netperf/netperf.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"]
DUT parameters:	
Application parameters:	time=60,buffer_size=8192,port_number=22115,ip_version=4,min_bw=30
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	Buffer Size Throughput 8192 55.06

LOG PATH

## 3.4 Test Suite : Browser

### Test Case amsdkA-262: Acid3 tests

Summary:

Measure Browser functionality and performance by running <http://acid3.acidtests.org/> tests

Steps:

Run automated test or manually open the browser and go to <http://acid3.acidtests.org/>

Expected Results:

Score 100 out of 100.

Test  
execution vatf  
engine:

Test script  
or logic: android/performance/webbrowser/browser\_test.rb

Required  
hardware dut1 = ["<platform>",android]  
assets:

DUT  
parameters: enable\_eth=netcfg eth0 up,enable\_dhcp=netcfg eth0 dhcp,set\_dns=setprop net.dns1 158.218.108.21,set\_L  
<http://www.gate.ti.com:80>,app\_name=WebViewBrowserTest.apk,test\_type=acid,host\_file\_path=/mnt/gta  
am start -W -n com.android.mWebView/.WebViewBrowserTest -a android.intent.action.MAIN -c androi

Application  
parameters: min\_score=100

Test  
Equipment  
parameters:

test case  
approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Test case PASS.

notes

LOG PATH

## Summary:

Measure Javascript performance by running <http://www2.webkit.org/perf/sunspider/sunspider.html> tests

## Steps:

Run automated test or manually open the browser and go to <http://www2.webkit.org/perf/sunspider-0.9/sunspider.html>

## Test

execution vatf

## engine:

Test script or logic: android/performance/webbrowser/browser\_test.rb

## Required

hardware dut1 = ["<platform>",android]

## assets:

## DUT

parameters: enable\_eth=netcfg eth0 up,enable\_dhcp=netcfg eth0 dhcp,set\_dns=setprop net.dns1 158.218.108.21,set\_p  
http://wwwgate.ti.com:80,app\_name=WebViewBrowserTest.apk,test\_type=sunspider,host\_file\_path=/mr  
am start -W -n com.android.mWebView/.WebViewBrowserTest -a android.intent.action.MAIN -c android

## Application

parameters: max\_exec\_time=5000

## Test

## Equipment

## parameters:

## test case

## approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Test case PASS.

## notes

LOG PATH

## Test Case amsdkA-264: V8 Br

## Summary:

Measure Javascript performance by running <http://v8.googlecode.com/svn/data/benchmarks/v6/run.html> tests

## Steps:

Run automated test or manually open the browser and go to <http://v8.googlecode.com/svn/data/benchmarks/v6/run.html>

Expected Results:

At least a score of 100.

Test

execution vatf

engine:

Test script or logic: android/performance/webbrowser/browser\_test.rb

Required

hardware dut1 = ["<platform>",android]

assets:

DUT enable\_eth=netcfg eth0 up,enable\_dhcp=netcfg eth0 dhcp,set\_dns=setprop net.dns1 158.218.108.21,set\_p  
parameters: http://wwwgate.ti.com:80,app\_name=WebViewBrowserTest.apk,test\_type=v8,host\_file\_path=/mnt/gtaut  
am start -W -n com.android.mWebView/.WebViewBrowserTest -a android.intent.action.MAIN -c androi

Application

parameters: min\_score=100

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Test case PASS.

notes

LOG PATH

**Test Case amsdkA-**

Summary:

Measure browser performance using <http://www.themaninblue.com/experiment/AnimationBenchmark/>

Steps:

Open the browser and go to <http://http://www.themaninblue.com/experiment/AnimationBenchmark/>

Test

execution vatf

engine:

Test script or logic: android/performance/webbrowser/browser\_test.rb

Required

hardware     dut1 = ["<platform>",android]

assets:

DUT

parameters:

enable\_eth=netcfg eth0 up,enable\_dhcp=netcfg eth0 dhcp,set\_dns=setprop net.dns1 158.218.108.21,set\_p  
http://wwwgate.ti.com:80,app\_name=WebViewBrowserTest.apk,test\_type=themaninblue,host\_file\_path:  
am start -W -n com.android.mWebView/.WebViewBrowserTest -a android.intent.action.MAIN -c androi

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build        BD\_Gin\_2.3.4.DEVKIT

Tester       gt\_amsdk\_lead

Testing      Test case PASS.

notes

LOG PATH

## 3.5 Test Suite : RowboPerf

### Test Case amsdkA-118: Dhrystone

Summary:

Measure Dhrystone bechmark

Steps:

Run RowboPerf's Dhrystone application

Expected Results:

As good or better than previous

Test execution

engine:        vatf

Test script or

logic:          android/performance/rowboperf/rowboperf.rb

Required hardware

assets:        dut1 = ["<platform>",android]

DUT parameters: app\_name=runDhrystone

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Test case PASS.

LOG PATH

**Test Case amsdkA-119: Whetstone**

Summary:

Measure Whetstone metric

Steps:

Run RowboPerf's Whetstone application

Expected Results:

As good or better than previous release

Test execution  
engine: vatf

Test script or  
logic: android/performance/rowboperf/rowboperf.rb

Required hardware  
assets: dut1 = ["<platform>", android]

DUT parameters: app\_name=runWhetstone

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Test case PASS.

LOG PATH

**Test Case amsdkA-120: Linpack**

Summary:

Measure Linpack metrics

Steps:

Run RowboPerf's Linpack application

Expected Results:

As good or better than previous release

Test execution  
engine: vatf

Test script or  
logic: android/performance/rowboperf/rowboperf.rb

Required hardware  
assets: dut1 = ["<platform>",android]

DUT parameters: app\_name=runLinpack

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Test case PASS.

LOG PATH

## 3.6 Test Suite : adb

### Test Case amsdkA-121: adb USB Performance

Summary:

Measure Android Debug bridge performance using USB connection

Steps:

Push and pull a 20MB file 10 times and measure the throughput

Expected Results:

As good or better than previous release

Test execution  
engine: vatf

Test script or  
logic: android/performance/adb/adb\_perf.rb

Required hardware  
assets: dut1 = ["<platform>",android]; server1 = ["linux\_server"]

DUT parameters:



## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Application parameters: file\_size=20,iterations=10,min\_bw=4000  
Test Equipment parameters:  
test case approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Functionality is good, performance does not meet expected value of 4000 KB/s. the avg throughputs measured were Mean-TX=3753.2  
Mean-RX=4244.0

### Test Case amsdkA-122: adb ethernet Performance

Summary:

Measure Android Debug bridge performance using ethernet connection

Steps:

Push and pull a 20MB file 10 times and measure the throughput

Expected Results:

As good or better than previous release

Test execution engine: vatf  
Test script or logic: android/performance/adb/adb\_perf.rb  
Required hardware assets: dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
DUT parameters:  
Application parameters: file\_size=20,iterations=10,min\_bw=4000  
Test Equipment parameters:  
test case approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Functionality is good, performance does not meet expected value of 4000 KB/s. the avg throughputs measured were Mean-TX=2770.3  
Mean-RX=3155.3

## 3.7 Test Suite : Storage

### 3.7.1 Test Suite : USB

#### Test Case amsdkA-265: USB vfat partition write/read test with a block size of 512 bytes

Summary:

USB vfat partition write/read test with a block size of 512 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a USB vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 512 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release

Test

execution     vfat

engine:

Test script  
or logic:     android/performance/StorageIO/storageio\_perf.rb

Required     dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
hardware

assets:

DUT device=USB,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIO.apk  
parameters: fileSize 104857600 -e blkSize 512 -e class com.ti.android.apps.storage.test.StorageIOTest#testIOSpeed  
StorageIOTest:com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\\.E\\+\\-]+)\s\*(.\*?bytes/sec);StorageIOTest

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-266: USB vfat partition write/read test with a block size of 4096 bytes

Summary:

---- Warning ----

TestLink Warning

test case name is too long (101 chars) > 100 => has been truncated

Original name

USB vfat partition write/read test with a block size of 4096 bytes and a file of size 104857600 bytes

---- \*\*\* ----

USB vfat partition write/read test with a block size of 4096 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a USB vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 4096 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen

8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as goog or better than the last release

Test

execution vfat

engine:

Test script  
or logic: android/performance/StorageIO/storageio\_perf.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

DUT device=USB,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIO.apk  
parameters: fileSize 104857600 -e blkSize 4096 -e class com.ti.android.apps.storage.test.StorageIOTest#testIOSpeed  
com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\.E\+|-])\s\*(.\*?bytes/sec);StorageIOTest

Application  
parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully

notes

LOG PATH

**Test Case amsdkA-267: USB vfat partition write/read test with a block size of 16384 bytes**

Summary:

---- Warning ----

TestLink Warning

test case name is too long (102 chars) > 100 => has been truncated

Original name

USB vfat partition write/read test with a block size of 16384 bytes and a file of size 104857600 bytes

---- \*\*\* ----

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

USB vfat partition write/read test with a block size of 16384 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a USB vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 16384 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as goog or better than the last release

Test

execution     vatf

engine:

Test script  
or logic:     android/performance/StorageIO/storageio\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

device=USB,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIO.apk

DUT     fileSize 104857600 -e blkSize 16384 -e class com.ti.android.apps.storage.test.StorageIOTest#testIOSpeed

parameters: com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([d\E\+|-]+)\s\*(.\*?bytes/sec);StorageIOTest

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully  
notes

LOG PATH

### Test Case amsdkA-268: USB vfat partition write/read test with a block size of 65536 bytes and a file of size 104857600 bytes

Summary:

---- Warning ----

TestLink Warning

test case name is too long (102 chars) > 100 => has been truncated

Original name

USB vfat partition write/read test with a block size of 65536 bytes and a file of size 104857600 bytes

---- \*\*\* ----

USB vfat partition write/read test with a block size of 65536 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a USB vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 65536 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release

vfat

Test execution engine:	
Test script or logic:	android/performance/StorageIO/storageio_perf.rb
Required hardware assets:	dut1 = ["<platform>", android]; server1 = ["linux_server"]
DUT parameters:	device=USB,file_system=vfat,test_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIO.apk fileSize 104857600 -e blkSize 65536 -e class com.ti.android.apps.storage.test.StorageIOTest#testIOSpeed com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log_option=StorageIOTest:I,perf_matches=StorageIOTest.*?(Write\s*rate)\s*:\s*([\d\.E\+\- ]+)\s*(.*?bytes/sec);StorageIOTest:I,perf_matches=StorageIOTest.*?(Write\s*rate)\s*:\s*([\d\.E\+\- ]+)\s*(.*?bytes/sec);StorageIOTest:I,perf_matches=StorageIOTest.*?(Write\s*rate)\s*:\s*([\d\.E\+\- ]+)\s*(.*?bytes/sec)
Application parameters:	
Test Equipment parameters:	
test case approver:	
Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead
Testing notes	StorageIO performance data collected successfully
	LOG PATH

```
Summary:
---- Warning ----
TestLink Warning
test case name is too long (103 chars) > 100 => has been truncated
Original name
USB vfat partition write/read test with a block size of 524288 bytes and a file of size 104857600 bytes
----- *** -----
```

## Manual execution

- ### 3.7.1 Test Suite : USB

- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 524288 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as goog or better than the last release

Test

execution     vatf

engine:

Test script  
or logic:     android/performance/StorageIO/storageio\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

DUT             device=USB,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIO.apk  
parameters:     fileSize 104857600 -e blkSize 524288 -e class com.ti.android.apps.storage.test.StorageIOTest#testIOSpec  
com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\.E\+\-]+)\s\*(.\*?bytes/sec);StorageIOTest

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build             BD\_Gin\_2.3.4.DEVKIT

Tester            gt\_amsdk\_lead

Testing           StorageIO performance data collected successfully

notes

LOG PATH

**Test Case amsdkA-270: USB vfat partition write/read test with a block size of 512 bytes**



Summary:

---- Warning ----

TestLink Warning

test case name is too long (104 chars) > 100 => has been truncated

Original name

USB vfat partition write/read test with a block size of 1048576 bytes and a file of size 104857600 bytes

---- \*\*\* ----

USB vfat partition write/read test with a block size of 1048576 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a USB vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 1048576 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release

Test

execution     vfat

engine:

Test script  
or logic:     android/performance/StorageIO/storageio\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

DUT           device=USB,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIO.apk

parameters:   fileSize 104857600 -e blkSize 1048576 -e class com.ti.android.apps.storage.test.StorageIOTest#testIOSp  
com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTestRunner

StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\.E\+\-]+\s\*\s\*(.\*?bytes/sec);Stora

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully

notes

LOG PATH

## 3.7.2 Test Suite : MMC/SD

### Test Case amsdkA-277: MMC/SD vfat partition write/read test with a block s

Summary:

---- Warning ----

TestLink Warning

test case name is too long (103 chars) > 100 => has been truncated

Original name

MMC/SD vfat partition write/read test with a block size of 512 bytes and a file of size 104857600 bytes

---- \*\*\* ----

MMC/SD vfat partition write/read test with a block size of 512 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a MMC/SD vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 512 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen

## 8) Collect the Write and Read Throughput

### Expected Results:

Throughput should be as goog or better than the last release

#### Test

execution vatf

#### engine:

Test script  
or logic: android/performance/StorageIO/storageio\_perf.rb

#### Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

#### assets:

device=MMC/SD,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;Storage

DUT MMC/SD -e fileSize 104857600 -e blkSize 512 -e class com.ti.android.apps.storage.test.StorageIOTest#t

parameters: com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=Stora  
StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\.E\+|-])\s\*(.\*?bytes/sec);Stora

#### Application

#### parameters:

#### Test

#### Equipment

#### parameters:

#### test case

#### approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully

#### notes

LOG PATH

## Test Case amsdkA-278: MMC/SD vfat partition write/read test with a block s

### Summary:

---- Warning ----

TestLink Warning

test case name is too long (104 chars) > 100 => has been truncated

Original name

MMC/SD vfat partition write/read test with a block size of 4096 bytes and a file of size 104857600 bytes

---- \*\*\* ----

## testreport AM37x Gingerbread 2.3.4 Devkit 2.1

MMC/SD vfat partition write/read test with a block size of 4096 bytes and a file of size 104857600 bytes

Steps:

## Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a MMC/SD vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 4096 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release

Test

execution      vatf

engine:

Test script or logic: android/performance/StorageIO/storageio\_perf.rb

Required

```
hardware    dut1 = ["<platform>",android]; server1 = ["linux_server"]
```

assets:

```
device=MMC/SD,file_system=vfat,test_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;Storage
```

```
DUT MMC/SD -e fileSize 104857600 -e blkSize 4096 -e class com.ti.android.apps.storage.test.StorageIOTest#
```

```
parameters: com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log_option=StorageIOTest:I,perf_matches=StorageIOTest.*?(Write\s*rate)\s*:\s*(\d\.E|\+|-|+)\s*(.*?bytes/sec);StorageIOTest
```

## Application

parameters:

Test

## Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully  
notes

LOG PATH

### **Test Case amsdkA-279: MMC/SD vfat partition write/read test with a block s**

Summary:

---- Warning ----

TestLink Warning

test case name is too long (105 chars) > 100 => has been truncated

Original name

MMC/SD vfat partition write/read test with a block size of 16384 bytes and a file of size 104857600 bytes

---- \*\*\* ----

MMC/SD vfat partition write/read test with a block size of 16384 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a MMC/SD vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 16384 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release  
vaf

Test  
execution  
engine:  
Test script or logic: android/performance/StorageIO/storageio\_perf.rb  
Required hardware: dut1 = ["<platform>",android]; server1 = ["linux\_server"]  
assets:  
device=MMC/SD,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIOTest.apk  
DUT MMC/SD -e fileSize 104857600 -e blkSize 16384 -e class com.ti.android.apps.storage.test.StorageIOTest  
parameters: com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTest:L,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\.E\+\-]+\s\*(.)\*?bytes/sec);StorageIOTest.\*?  
Application parameters:  
Test Equipment parameters:  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing StorageIO performance data collected successfully  
notes  
LOG PATH

### Test Case amsdkA-280: MMC/SD vfat partition write/read test with a block size of 65536 bytes and a file of size 104857600 bytes

Summary:  
---- Warning ----  
TestLink Warning  
test case name is too long (105 chars) > 100 => has been truncated  
Original name  
MMC/SD vfat partition write/read test with a block size of 65536 bytes and a file of size 104857600 bytes  
---- \*\*\* ----

MMC/SD vfat partition write/read test with a block size of 65536 bytes and a file of size 104857600 bytes  
Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a MMC/SD vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut

- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 65536 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release

Test

execution      vatf

engine:

## Test script

or logic: `android/performance/StorageIO/storageio_perf.rb`

Required

```
hardware    dut1 = ["<platform>",android]; server1 = ["linux_server"]
```

assets:

```
device=MMC/SD,file_system=vfat,test_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;Storage
```

DUT

```
MMC/SD -e fileSize 104857600 -e blkSize 65536 -e class com.ti.android.apps.storage.test.StorageIOTes
```

parameters:

```
com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log_option=StorageIOTest:I,perf_matches=StorageIOTest.*?(Write\s*rate)\s*:\s*(\\d\\.E\\+\\-|\\+)|)\s*(.*?bytes/sec);StorageIOTest
```

## Application

parameters:

Test

## Equipment

parameters:

test case

approver:

**Last Result: Passed**

## Build

BD Gin 2.3.4.DEVKIT

Tester

gt\_amsdk\_lead

## Testing

StorageIO performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-281: MMC/SD vfat partition write/read test with a block size

Summary:

---- Warning ----

TestLink Warning

test case name is too long (106 chars) > 100 => has been truncated

Original name

MMC/SD vfat partition write/read test with a block size of 524288 bytes and a file of size 104857600 bytes

---- \*\*\* ----

MMC/SD vfat partition write/read test with a block size of 524288 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a MMC/SD vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 524288 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as goog or better than the last release

Test

execution     vfatf

engine:

Test script  
or logic:     android/performance/StorageIO/storageio\_perf.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

DUT           device=MMC/SD,file\_system=vfat,test\_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;StorageIOTest.apk

parameters:   MMC/SD -e fileSize 104857600 -e blkSize 524288 -e class com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTestRunner



StorageIOTest:I,perf\_matches=StorageIOTest.\*?(Write\s\*rate)\s\*:\s\*([\d\\.E\+|-])\s\*(.\*?bytes/sec);Stora

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully

notes

LOG PATH

### Test Case amsdkA-282: MMC/SD vfat partition write/read test with a block si

Summary:

---- Warning ----

TestLink Warning

test case name is too long (107 chars) > 100 => has been truncated

Original name

MMC/SD vfat partition write/read test with a block size of 1048576 bytes and a file of size 104857600 bytes

---- \*\*\* ----

MMC/SD vfat partition write/read test with a block size of 1048576 bytes and a file of size 104857600 bytes

Steps:

Manual execution

- 1) Verify that you have StorageIO installed in the dut
- 2) Mount a MMC/SD vfat partition on the dut's file system, if not already mounted
- 3) Start StorageIO on the dut
- 4) Select the partition mounted in step 2) from the External Device: Spinner
- 5) Enter 1048576 in the Block Size: field
- 6) Enter 104857600 in the File Size: field
- 7) Click the Run button, and wait for the results screen
- 8) Collect the Write and Read Throughput

Expected Results:

Throughput should be as good or better than the last release

Test

execution      vatif

engine:

Test script  
or logic: android/performance/StorageIO/storageio\_perf.rb

Required

```
hardware    dut1 = ["<platform>",android]; server1 = ["linux_server"]
```

assets:

```
device=MMC/SD,file_system=vfat,test_libs=StorageIOTest.apk:com.ti.android.apps.storage.test;Storage
```

```
DUT MMC/SD -e fileSize 104857600 -e blkSize 1048576 -e class com.ti.android.apps.storage.test.StorageIOT
```

parameters: com.ti.android.apps.storage.test/com.ti.android.apps.storage.test.StorageIOTestRunner,log\_option=StorageIOTestRunner

```
StorageIOTest:Iperf_matches=StorageIOTest.*?(Write\s*rate)\s*:\s*(\[\d\.E\+|-]+\)\s*(.*?bytes/sec);Stora
```

## Application

parameters:

Test

## Equipment

parameters:

test case

approver:

**Last Result: Passed**

Build BD Gin 2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing StorageIO performance data collected successfully

notes

LOG PATH

### 3.8 Test Suite : Database

### 3.8.1 Test Suite : TestIndex

### Test Case amsdkA-124: TestIndex Benchmarks

### Summary:

Run Testindex benchmark application to measure database performance.

Steps:

- 1) Install Testindex (Benchmark.apk) file available at  
android/common/testindex/benchmark.apk
- 2) Start benchmark application
- 3) Press F1 or the menu button and select sqllite tp start the test
- 4) Wait few minutes (~15min) until the test completes.

#### Expected Results:

Performance should be as good or better than previous releases

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Elapsed time 10000 =549433ms

2000 index elapsed time search = 17702 ms

iteratingthrough 20000 record = 1014 ms

database length before delete 891904

elapsed data for deleting 10000 record 542863ms

database file length 8192

## 3.9 Test Suite : Power

### 3.9.1 Test Suite : DVFS-Performance

**Test Case amsdkA-315: Idle power perfor**

## Summary:

Acquire FULL WakeLock and measure power w/out running any other application

## Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

## Expected Results:

100 power samples will be collected in the performance table

## Test

execution vatf

## engine:

## Test script

or logic: android/performance/power/power\_perf.rb

## Required

hardware dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

## assets:

## DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

## Application

parameters: loop\_count=100

## Test

Equipment sample\_count=5,timeout=2

## parameters:

## test case

## approver:

Last Result: **Passed**

## Build

BD\_Gin\_2.3.4.DEVKIT

## Tester

gt\_amsdk\_lead

## Testing

Power Performance data collected

## notes

LOG PATH**Test Case amsdkA-316: Idle power performance**

## Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

## Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

## Expected Results:

100 power samples will be collected in the performance table

## Test

execution vatf

## engine:

Test script or logic: android/performance/power/power\_perf.rb

## Required

hardware assets: dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

## DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

## Application parameters:

loop\_count=100

## Test

Equipment parameters: sample\_count=5,timeout=2

## test case

## approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing      Power Performance data collected  
notes

LOG PATH

**Test Case amsdkA-317: Idle power performance**

Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test  
execution      vatf  
engine:

Test script      android/performance/power/power\_perf.rb  
or logic:

Required  
hardware      dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT      test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters:      com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters:      loop\_count=100

Test  
Equipment      sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build      BD\_Gin\_2.3.4.DEVKIT

Tester       gt\_amsdk\_lead  
Testing      Power Performance data collected  
notes  
LOG PATH

## Test Case amsdkA-318: Idle power performance

### Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

### Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

### Test

execution    vatf  
engine:

Test script    android/performance/power/power\_perf.rb  
or logic:

### Required

hardware      dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

### DUT

parameters:   test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters:

loop\_count=100

### Test

Equipment    sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing Power Performance data collected  
notes LOG PATH

## Test Case amsdkA-319: Dhrystone power performance

### Summary:

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

### Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

### Test

execution vatf  
engine:

Test script android/performance/power/power\_perf.rb  
or logic:

Required hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

### Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:



Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-320: 3D Gra

Summary:

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-321: Audio +

Summary:

Measure power while running video and audio decode and playback

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware  
assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1=["multimeter"]

DUT  
parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test sample\_count=5,timeout=2

Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## 3.9.2 Test Suite : DVFS-Powersave

**Test Case amsdkA-322: Idle power perform**

Summary:

Acquire FULL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled.  
Application  
parameters: loop\_count=100  
Test  
Equipment sample\_count=5,timeout=2  
parameters:  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing Power Performance data collected  
notes  
LOG PATH

### Test Case amsdkA-323: Idle power performance v

Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test  
execution vatf  
engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

hardware

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-324: Idle power performance

Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script android/performance/power/power\_perf.rb

or logic:

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT

parameters:

test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters:

loop\_count=100

Test

Equipment

sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build

BD\_Gin\_2.3.4.DEVKIT

Tester

gt\_amsdk\_lead

Testing

Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-325: Idle power performance

Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

vatf

execution

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required hardware assets: dut1 = [<platform>,android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

DUT parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

Test Equipment parameters: sample\_count=5,timeout=2

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-326: Dhrystone power perform

Summary:

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test  
execution vatf  
engine:  
Test script or logic: android/performance/power/power\_perf.rb  
Required hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled.  
Application parameters: loop\_count=100  
Test  
Equipment parameters: sample\_count=5,timeout=2  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing Power Performance data collected  
notes  
LOG PATH

### Test Case amsdkA-327: 3D Graph

Summary:

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:



100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-328: Audio + V

Summary:

Measure power while running video and audio decode and playback

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## 3.9.3 Test Suite : DVFS-userspace

### 3.9.3.1 Test Suite : 1MHz

Test Case amsdkA-339: Idle power

Summary:

Acquire FULL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters:

loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-340: Idle power perfo

#### Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1=["multimeter"]  
assets:

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-341: Idle power per

Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

Steps:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-342: Idle power**

Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script

or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.

com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build

BD\_Gin\_2.3.4.DEVKIT

Tester

gt\_amsdk\_lead

Testing

Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-343: Dhrystone pow**

Summary:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-344:**

Summary:

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected  
notes

LOG PATH



## Summary:

Measure power while running video and audio decode and playback

## Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

## Expected Results:

100 power samples will be collected in the performance table

## Test

execution vatf

## engine:

Test script or logic: android/performance/power/power\_perf.rb

## Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

## DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

## Application

parameters: loop\_count=100

## Test

Equipment parameters: sample\_count=5,timeout=2

## test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

## notes

LOG PATH

### 3.9.3.2 Test Suite : 800KHz

Test Case amsdkA-346: Idle power

Summary:

Acquire FULL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester       gt\_amsdk\_lead  
Testing       Power Performance data collected  
notes  
              LOG PATH

**Test Case amsdkA-347: Idle power perfor**

Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution     vatf

engine:

Test script    android/performance/power/power\_perf.rb  
or logic:

Required

hardware       dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1=["multimeter"]  
assets:

DUT

parameters:    test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters:    loop\_count=100

Test

Equipment     sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing Power Performance data collected  
notes LOG PATH

### Test Case amsdkA-348: Idle power per

#### Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

#### Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

#### Expected Results:

100 power samples will be collected in the performance table

#### Test

execution vatf  
engine:

Test script or logic: android/performance/power/power\_perf.rb

#### Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

#### DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

#### Application

parameters: loop\_count=100

#### Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-349: Idle power p

Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-350: Dhrystone power

Summary:

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

Test sample\_count=5,timeout=2

Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-351: 3**

Summary:

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application loop\_count=100

parameters:

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-352: A**

Summary:

Measure power while running video and audio decode and playback

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.

parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo



com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### 3.9.3.3 Test Suite : 600KHz

**Test Case amsdkA-353: Idle power**

Summary:

Acquire FULL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

### Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

### DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

### Application

parameters: loop\_count=100

### Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-354: Idle power perfor

### Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test script  
or logic: android/performance/power/power\_perf.rb

Required  
hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-355: Idle power per

Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test vatf

execution

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-356: Idle power p

Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected  
notes

LOG PATH

## Test Case amsdkA-357: Dhrystone power

Summary:

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-358: 3**

Summary:

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

**Test Case amsdkA-359: A**

Summary:

Measure power while running video and audio decode and playback

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### 3.9.3.4 Test Suite : 300KHz

**Test Case amsdkA-360: Idle power**

Summary:

Acquire FULL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,



chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

#### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected  
notes

LOG PATH

#### Test Case amsdkA-361: Idle power perfor

Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters:

loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-362: Idle power per

#### Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1=["multimeter"]  
assets:

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-363: Idle power p

Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

Steps:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-364: Dhrystone power

Summary:

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script

or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.

com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build

BD\_Gin\_2.3.4.DEVKIT

Tester

gt\_amsdk\_lead

Testing

Power Performance data collected

notes

LOG PATH

Test Case amsdkA-365: 3

Summary:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

Test Case amsdkA-366: A

Summary:

Measure power while running video and audio decode and playback

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected  
notes

LOG PATH

### 3.9.4 Test Suite : DVFS-Ondemand(default)

#### Test Case amsdkA-308: Idle power perform

Summary:

Acquire FULL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlFull  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT



Tester       gt\_amsdk\_lead  
Testing      Power Performance data collected  
notes  
              LOG PATH

**Test Case amsdkA-309: Idle power performance v**

Summary:

Acquire SCREEN\_BRIGHT WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution    vatf

engine:

Test script    android/performance/power/power\_perf.rb  
or logic:

Required

hardware      dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT

parameters:    test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlBright

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters:    loop\_count=100

Test

Equipment    sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing Power Performance data collected  
notes  
LOG PATH

## Test Case amsdkA-310: Idle power performance

### Summary:

Acquire SCREEN\_DIM WakeLock and measure power w/out running any other application

### Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

### Expected Results:

100 power samples will be collected in the performance table

### Test

execution vatf  
engine:

Test script or logic: android/performance/power/power\_perf.rb

### Required

hardware assets: dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

### DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlDim  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

### Application

parameters: loop\_count=100

### Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-311: Idle power performance

Summary:

Acquire PARTIAL WakeLock and measure power w/out running any other application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.  
parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlPartial  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-312: Dhrystone power perfor

Summary:

Acquire PARTIAL WakeLock and measure power while running Dhrystone benchmark

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>", android\_power]; server1 = ["linux\_server"]; multimeter1 = ["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlCpu

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application

parameters: loop\_count=100

Test

sample\_count=5,timeout=2

Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-313: 3D Graphics

Summary:

Measure power while running 3D graphics application

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware assets: dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

DUT

parameters: test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlGraphics  
com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application loop\_count=100

parameters:

Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-314: Audio + V

Summary:

Measure power while running video and audio decode and playback

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT test\_libs=PowerControllerTest.apk:com.ti.android.apps.powercontroller.test;PowerController.apk:com.ti.

parameters: com.ti.android.apps.powercontroller.test.PowerControllerTest#testPowerCtrlVideo

com.ti.android.apps.powercontroller.test/com.ti.android.apps.powercontroller.test.TiTestRunner,disabled

Application  
parameters: loop\_count=100

Test

Equipment parameters: sample\_count=5,timeout=2

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## 3.9.5 Test Suite : CPU Idle

### Test Case amsdkA-335: SUSPEND MODE power consumption sleep\_while\_idle d

Summary:

Measure power while system is in SUSPEND mode

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script or logic: android/performance/power/power\_perf.rb

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

### Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

### DUT

parameters: bypass\_dut=yes,bypass\_dut\_wait=120,disabled\_cpu\_idle\_modes=sleep\_while\_idle;enable\_off\_mode,dv

### Application

parameters: loop\_count=100

### Test

Equipment sample\_count=5,timeout=2

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

## Test Case amsdkA-336: SUSPEND MODE power consumption sleep\_while\_idle e

Summary:

Measure power while system is in SUSPEND mode

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

Test script android/performance/power/power\_perf.rb



or logic:

Required

hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]

assets:

DUT

parameters: bypass\_dut=yes,bypass\_dut\_wait=120,enabled\_cpu\_idle\_modes=enable\_off\_mode;sleep\_while\_idle,dv

Application

parameters: loop\_count=100

Test

Equipment sample\_count=5,timeout=10

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-337: SUSPEND MODE power consumption sleep

Summary:

Measure power while system is in SUSPEND mode

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test

execution vatf

engine:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test script  
or logic: android/performance/power/power\_perf.rb

Required  
hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT  
parameters: bypass\_dut=yes,bypass\_dut\_wait=120,disabled\_cpu\_idle\_modes=sleep\_while\_idle,enabled\_cpu\_idle\_m

Application  
parameters: loop\_count=100

Test  
Equipment sample\_count=5,timeout=2  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected

notes

LOG PATH

### Test Case amsdkA-338: SUSPEND MODE power consumption sleep

Summary:

Measure power while system is in SUSPEND mode

Steps:

Connect Keithley 2000 Multimeter as described in attached doc.

chan1 measures voltage drop at vdd1,

chan2 measures voltage drop at vdd2,

chan3 is ignored,

chan4=vdd1 and chan5=vdd2

Run automated power\_perf.rb script using VATF TEE

Expected Results:

100 power samples will be collected in the performance table

Test vatf  
execution

engine:

Test script  
or logic: android/performance/power/power\_perf.rb

Required  
hardware dut1 = ["<platform>",android\_power]; server1 = ["linux\_server"]; multimeter1 =["multimeter"]  
assets:

DUT  
parameters: bypass\_dut=yes,bypass\_dut\_wait=120,enabled\_cpu\_idle\_modes=sleep\_while\_idle,disabled\_cpu\_idle\_m

Application  
parameters: loop\_count=100

Test  
Equipment sample\_count=5,timeout=2  
parameters:

test case  
approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Power Performance data collected  
notes

LOG PATH

## 3.10 Test Suite : WLAN

### 3.10.1 Test Suite : Non-secure

#### Test Case amsdkA-292: WLAN Non-secure, TCP Stream, Buffer size 1024

Summary:

WLAN Non-secure test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

- 1) Configure a Non-secure wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60

Test

execution     vatf

engine:

Test script

or logic:     android/performance/wlan/wlan.rb

Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]

assets:

DUT

parameters:     test\_sequence=remove\_all;add;select;test,ssid=gtaccess-open,auth\_alg=OPEN,key\_mgmt=NONE

Application

parameters:     time=60,buffer\_size=1024,port\_number=22115,ip\_version=4,min\_bw=6,cpu\_load\_samples=10,wlan\_co

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build         BD\_Gin\_2.3.4.DEVKIT

Tester        gt\_amsdk\_lead

Testing

notes         Functionality is good, performance does not meet expected value of 6.0 Mb/s. the avg throughput measur

SDOCM00082457

### Test Case amsdkA-293: WLAN Non-secure, TCP Stream, Buffer size 4096

Summary:

WLAN Non-secure test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

1) Configure a Non-secure wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60

Test  
execution     vatf  
engine:

Test script  
or logic:     android/performance/wlan/wlan.rb

Required  
hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
assets:

DUT  
parameters:   test\_sequence=remove\_all;add;select;test,ssid=gtaccess-open,auth\_alg=OPEN,key\_mgmt=NONE

Application  
parameters:   time=60,buffer\_size=4096,port\_number=22115,ip\_version=4,min\_bw=11,cpu\_load\_samples=10,wlan\_c

Test  
Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build         BD\_Gin\_2.3.4.DEVKIT

Tester        gt\_amsdk\_lead

Testing  
notes         Functionality is good, performance does not meet expected value of 11.0 Mb/s. the avg throughput meas  
size of 4096.

SDOCM00082457

**Test Case amsdkA-294: WLAN Non-secure, TCP Stream, Buffer size 8192**

## Summary:

WLAN Non-secure test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

## Steps:

## Manual Verification:

1) Configure a Non-secure wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

## Test

execution     vatf  
engine:

Test script     android/performance/wlan/wlan.rb  
or logic:

## Required

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
assets:

## DUT

parameters:     test\_sequence=remove\_all;add;select;test,ssid=gtaccess-open,auth\_alg=OPEN,key\_mgmt=NONE

Application  
parameters:

time=60,buffer\_size=8192,port\_number=22115,ip\_version=4,min\_bw=14,cpu\_load\_samples=10,wlan\_c

## Test

Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build             BD\_Gin\_2.3.4.DEVKIT

Tester	gt_amsdk_lead
Testing notes	Functionality is good, performance does not meet expected value of 14.0 Mb/s. the avg throughput measured size of 8192.
	SDOCM00082457

## 3.10.2 Test Suite : WEP 40 bits

### Test Case amsdkA-295: WLAN WEP 40 bits, TCP Stream, Buffer size 1024

Summary:

WLAN WEP 40 bits test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

- 1) Configure a WEP 40 bits wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

- 4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

Test execution engine:	vatf
Test script or logic:	android/performance/wlan/wlan.rb
Required hardware assets:	dut1 = ["<platform>",android]; server1 = ["linux_server"];ap1 = ["access_point"]

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

DUT  
parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wep40,auth\_alg=OPEN,wep\_key0=0123456789  
Application  
parameters: time=60,buffer\_size=1024,port\_number=22115,ip\_version=4,min\_bw=8,cpu\_load\_samples=10,wlan\_co  
Test  
Equipment  
parameters:  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Functionality is good, performance does not meet expected value of 8.0 Mb/s. the avg throughput measur  
size of 1024.  
SDOCM00082457

### Test Case amsdkA-296: WLAN WEP 40 bits, TCP Stream, Buffer size 4096

#### Summary:

WLAN WEP 40 bits test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

#### Steps:

#### Manual Verification:

- 1) Configure a WEP 40 bits wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

- 4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the  
netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"



## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test  
execution     vatf  
engine:  
Test script  
or logic:     android/performance/wlan/wlan.rb  
Required  
hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
assets:  
DUT  
parameters:   test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wep40,auth\_alg=OPEN,wep\_key0=0123456789  
Application  
parameters:   time=60,buffer\_size=4096,port\_number=22115,ip\_version=4,min\_bw=11,cpu\_load\_samples=10,wlan\_c  
Test  
Equipment  
parameters:  
test case  
approver:  
Last Result: **Passed**  
Build         BD\_Gin\_2.3.4.DEVKIT  
Tester        gt\_amsdk\_lead  
Testing  
notes         Functionality is good, performance does not meet expected value of 11.0 Mb/s. the avg throughput measu  
size of 4096.  
  
SDOCM00082457

### Test Case amsdkA-297: WLAN WEP 40 bits, TCP Stream, Buffer size 8192

Summary:

WLAN WEP 40 bits test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT)

Steps:

Manual Verification:

- 1) Configure a WEP 40 bits wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

- 4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the  
netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

Test  
 execution     vatf  
 engine:  
 Test script     android/performance/wlan/wlan.rb  
 or logic:  
 Required  
 hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
 assets:  
 DUT  
 parameters:     test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wep40,auth\_alg=OPEN,wep\_key0=0123456789  
 Application  
 parameters:     time=60,buffer\_size=8192,port\_number=22115,ip\_version=4,min\_bw=14,cpu\_load\_samples=10,wlan\_c  
 Test  
 Equipment  
 parameters:  
 test case  
 approver:  
 Last Result: **Passed**  
 Build            BD\_Gin\_2.3.4.DEVKIT  
 Tester          gt\_amsdk\_lead  
 Testing        Functionality is good, performance does not meet expected value of 14.0 Mb/s. the avg throughput measu  
 notes          size of 8192.  
  
 SDOCM00082457

### 3.10.3 Test Suite : WEP 128 bits

#### Test Case amsdkA-298: WLAN WEP 128 bits, TCP Stream, Buffer size 10

Summary:

WLAN WEP 128 bits test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT)  
 Steps:

Manual Verification:

1) Configure a WEP 128 bits wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60

Test

execution vatf

engine:

Test script  
or logic: android/performance/wlan/wlan.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
assets:

DUT

parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wep128,auth\_alg=OPEN,wep\_key0=123456789

Application

parameters: time=60,buffer\_size=1024,port\_number=22115,ip\_version=4,min\_bw=8,cpu\_load\_samples=10,wlan\_co

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Functionality is good, performance does not meet expected value of 8.0 Mb/s. the avg throughput measur  
1024.

SDOC M00082457

**Test Case amsdkA-299: WLAN WEP 128 bits, TCP Stream, Buffer size 4096**

Summary:

WLAN WEP 128 bits test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT)  
Steps:

Manual Verification:

- 1) Configure a WEP 128 bits wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

- 4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

Test

execution vatf

engine:

Test script or logic: android/performance/wlan/wlan.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
assets:

DUT

parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wep128,auth\_alg=OPEN,wep\_key0=123456789

Application

parameters: time=60,buffer\_size=4096,port\_number=22115,ip\_version=4,min\_bw=11,cpu\_load\_samples=10,wlan\_c

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Functionality is good, performance does not meet expected value of 11.0 Mb/s. the avg throughput measu  
4096.

SDOCM00082457

**Test Case amsdkA-300: WLAN WEP 128 bits, TCP Stream, Buffer size 8192****Summary:**

WLAN WEP 128 bits test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT)

Steps:

**Manual Verification:**

1) Configure a WEP 128 bits wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

**Test**

execution     vatf

**engine:****Test script  
or logic:**

android/performance/wlan/wlan.rb

**Required**

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]

assets:

**DUT**

parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wep128,auth\_alg=OPEN,wep\_key0=123456789

**Application  
parameters:**

time=60,buffer\_size=8192,port\_number=22115,ip\_version=4,min\_bw=14,cpu\_load\_samples=10,wlan\_c

**Test****Equipment  
parameters:**

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Functionality is good, performance does not meet expected value of 14.0 Mb/s. the avg throughput measured is 8192.

SDOCM00082457

## 3.10.4 Test Suite : WPA-PSK

### Test Case amsdkA-301: WLAN WPA-PSK, TCP Stream, Buffer size 1024

Summary:

WLAN WPA-PSK test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

- 1) Configure a WPA-PSK wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

- 4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the test suite)  
netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60 -- -s 1024"

Test

execution vatf

engine:

android/performance/wlan/wlan.rb

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test script  
or logic:

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]

assets:

DUT

parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wpa-psk,psk=q1w2e3r4,key\_mgmt=WPA-PSK

Application

parameters: time=60,buffer\_size=1024,port\_number=22115,ip\_version=4,min\_bw=8,cpu\_load\_samples=10,wlan\_co

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Functionality is good, performance does not meet expected value of 8.0 Mb/s. the avg throughput measur  
size of 1024.

SDOCM00082457

### Test Case amsdkA-302: WLAN WPA-PSK, TCP Stream, Buffer size 4096

Summary:

WLAN WPA-PSK test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

1) Configure a WPA-PSK wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

Test  
 execution engine: vatf  
 Test script or logic: android/performance/wlan/wlan.rb  
 Required hardware assets: dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
 DUT parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wpa-psk,psk=q1w2e3r4,key\_mgmt=WPA-PSK  
 Application parameters: time=60,buffer\_size=4096,port\_number=22115,ip\_version=4,min\_bw=11,cpu\_load\_samples=10,wlan\_c  
 Test Equipment parameters:  
 test case approver:  
 Last Result: **Passed**  
 Build BD\_Gin\_2.3.4.DEVKIT  
 Tester gt\_amsdk\_lead  
 Testing notes: Functionality is good, performance does not meet expected value of 11.0 Mb/s. the avg throughput measu  
 size of 4096.  
 SDOCM00082457

### Test Case amsdkA-303: WLAN WPA-PSK, TCP Stream, Buffer size 8192

#### Summary:

WLAN WPA-PSK test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).  
 Steps:

#### Manual Verification:

- 1) Configure a WPA-PSK wlan in the access point and the dut
- 2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

- 3) Start netserver in the Host Machine (Linux preferably)



## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60

Test  
execution engine: vatf  
Test script or logic: android/performance/wlan/wlan.rb  
Required hardware assets: dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
DUT parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wpa-psk,psk=q1w2e3r4,key\_mgmt=WPA-PSK  
Application parameters: time=60,buffer\_size=8192,port\_number=22115,ip\_version=4,min\_bw=14,cpu\_load\_samples=10,wlan\_c  
Test Equipment parameters:  
test case  
approver:  
Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead  
Testing notes Functionality is good, performance does not meet expected value of 14.0 Mb/s. the avg throughput measu  
size of 8192.

SDOCM00082457

### 3.10.5 Test Suite : WPA2-PSK

#### Test Case amsdkA-304: WLAN WPA2-PSK, TCP Stream, Buffer size 1024

Summary:

WLAN WPA2-PSK test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

1) Configure a WPA2-PSK wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60

Test

execution     vatf  
engine:

Test script     android/performance/wlan/wlan.rb  
or logic:

Required     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]  
hardware     assets:

DUT     test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wpa2-psk,psk=psk12345,key\_mgmt=WPA-PSK  
parameters:

Application     time=60,buffer\_size=1024,port\_number=22115,ip\_version=4,min\_bw=8,cpu\_load\_samples=10,wlan\_co  
parameters:

Test  
Equipment  
parameters:

test case  
approver:

Last Result: **Passed**

Build     BD\_Gin\_2.3.4.DEVKIT

Tester     gt\_amsdk\_lead

Testing     Functionality is good, performance does not meet expected value of 8.0 Mb/s. the avg throughput measur  
notes     size of 1024.

**Test Case amsdkA-305: WLAN WPA2-PSK, TCP Stream, Buffer size 4096****Summary:**

WLAN WPA2-PSK test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

**Steps:****Manual Verification:**

1) Configure a WPA2-PSK wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60

**Test**

execution     vatf

**engine:****Test script**

or logic:     android/performance/wlan/wlan.rb

**Required**

hardware     dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]

**assets:****DUT**

parameters:     test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wpa2-psk,psk=psk12345,key\_mgmt=WPA-PSK

**Application**

parameters:     time=60,buffer\_size=4096,port\_number=22115,ip\_version=4,min\_bw=11,cpu\_load\_samples=10,wlan\_c

**Test****Equipment****parameters:**

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Functionality is good, performance does not meet expected value of 11.0 Mb/s. the avg throughput measured size of 4096.

SDOCM00082457

### Test Case amsdkA-306: WLAN WPA2-PSK, TCP Stream, Buffer size 8192

Summary:

WLAN WPA2-PSK test, measures TCP bandwidth between Server (Running on Host PC) and Client (Android DUT).

Steps:

Manual Verification:

1) Configure a WPA2-PSK wlan in the access point and the dut

2) Verify that you have netperf installed in your host machine by typing "netperf -h"

If you get an error, you need to install netperf. On a ubuntu system, you may type "sudo apt-get install netperf"

3) Start netserver in the Host Machine (Linux preferably)

sudo netserver -p 22115 -4. Where -p specifies the listening port number and -4 sets the ip protocol version to IPV4.

4) Start netperf on the device under test (Note: There is no need to install an APK as netperf is already provided in the

netperf -H <host machine> -l <test time in secs> -- -s <tcp buffer size>. For example "netperf -H 158.218.103.64 -l 60"

Test

execution vatf

engine:

Test script or logic: android/performance/wlan/wlan.rb

Required hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"];ap1 = ["access\_point"]

assets:

DUT

parameters: test\_sequence=remove\_all;add;select;test,ssid=gtaccess-wpa2-psk,psk=psk12345,key\_mgmt=WPA-PSK

Application

parameters: time=60,buffer\_size=8192,port\_number=22115,ip\_version=4,min\_bw=14,cpu\_load\_samples=10,wlan\_c

Test

Equipment

parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes Functionality is good, performance does not meet expected value of 14.0 Mb/s. the avg throughput measu  
size of 8192.

SDOCM00082457

## 4 Test Suite : Stress

### 4.1 Test Suite : Monkey

#### Test Case amsdkA-307: Monkey System Stress

Summary:

Stress Test the system using the monkey tool

Steps:

Manual Verification:

1) Run the monkey tool for the given number of events, with the specified flags

2) Verify that there are no crashes

Test

execution vatf

engine:

Test script or logic: android/ui/ui.rb

Required

hardware dut1 = ["<platform>",android]; server1 = ["linux\_server"]

assets:

black\_list=com.android.providers.telephony;com.android.phone;com.android.camera,event\_count=5000

DUT

parameters:

Application

parameters:

Test

Equipment

parameters:

test case

approver:

Last Result: **Failed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Crash(es) reported for [{"com.ti.android.apps.launcher", "pid 6155", "java.lang.RuntimeException:"}] No

notes response(s) reported []

LOG PATH

## 4.2 Test Suite : USB

Summary:

Tests USB enumeration capability.

Steps:

- 1) Connect the test switch(Extron) to the test PC using serial port and to the platform using USB.
- 2) connect devices to the HUB. Devices are like Key Board, Mous, JoyStick etc
- 3) Update you test bench to include the swtich, change DNS workaround file to include the switch.
- 4) Execute the test case. It should run automaticaly

Expected Results:

Check pass or fail status.

Test

execution vatf

engine:

Test script android/stability/usb/usb\_enumeration.rb  
or logic:

Required dut1 = ["<platform>",android]; server1 = ["linux\_server"]; usb\_sw=["usb\_switch"]  
hardware

assets:

DUT  
parameters: iterations=100,wait\_after\_connect=3,wait\_after\_disconnect=3,pass\_rate=100,enum\_strings=USB\s+hub\

Application  
parameters:

Test

Equipment  
parameters:

test case

approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing Success Enumeration rate=100.0

notes

LOG PATH

### Test Case amsdkA-235: Long-Term Graphics test

Summary:

Run Graphics demos for 48 hrs

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 5 Test Suite : Documentation

### Test Case amsdkA-54: DevKit Users Guide

Summary:

Verify that a DevKit Users Guide document is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

**Test Case amsdkA-55: Release Notes**

Summary:

Verify that a Release Notes are provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

**Test Case amsdkA-56: Porting Guide**

Summary:



Verify that an Android Rowboat Porting Guide document is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-57: CTS Report

Summary:

Verify that a CTS report is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-58: DevKit Test Report

Summary:

Verify that a DevKit Test Report is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-59: Android Rowboat Manifest**

Summary:

Verify that an Android Rowboat Manifest document t is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-60: Datasheet**

Summary:

Verify that a Datasheet document is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-72: Eclipse Setup

Summary:

Verify that procedure to setup Eclipse for Android development is provided or referenced in the DevKit documentation

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-73: ADB over Ethernet Setup

Summary:

Verify that the procedure to setup Android Debug Bridge (ADB) over Ethernet is provided or referenced in the DevKit documentation

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-74: ADB over USB Setup**

Summary:

Verify that the procedure to setup Android Debug Bridge (ADB) over USB is provided or referenced in the DevKit documentation

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-75: ADB .apk File Download**

Summary:

Verify that procedure to download .apk files using ADB is documented

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-76: Eclipse APK File Download**

Summary:

Verify that procedure to download .apk files using Eclipse is documented

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### **Test Case amsdkA-78: DevKit Developers Guide**

Summary:

Verify that a DevKit Developers Guide document is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

#### Test Case amsdkA-81: Document Format

Summary:

Verify that all documents follow consistent template for same/similar information

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

Testing notes

#### Test Case amsdkA-82: Packages List

Summary:

Verify that the DevKit includes a list of packages contained in each filesystem image.

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead

#### Test Case amsdkA-83: PinMux Utility Usage

Summary:

Verify that the procedure to use the PinMux utility is provided

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead

#### Test Case amsdkA-84: Serial Flash Utility Usage

Summary:

Verify that the procedure to use the Serial Flash utility is provided

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 6 Test Suite : Kitting

### Test Case amsdkA-51: Review DevKit components with the legal team

Summary:

All components in the DevKit shall be reviewed by the legal team to identify any possible incompatibility.

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead



### Test Case amsdkA-52: Review SW Manifest with OSRB

Summary:

Software manifests shall be reviewed and approved by the OSRB

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-53: DevKit Content

Summary:

Devkit content should be complete (see expected results section)

Expected Results:

Â· Source  
Â· 2.6.32 Kernel  
Â· u-boot  
Â· x-loader  
Â· SGX SDK Installer  
Â· Pre-built binaries  
Â· AM35x\_EVM  
Â· uImage  
Â· u-boot  
Â· x-loader.bin.ift  
Â· MLO  
Â· AM37x\_OMAP35x\_EVM  
Â· uImage  
Â· u-boot  
Â· x-loader.bin.ift  
Â· MLO  
Â· Beagleboard  
Â· uImage  
Â· u-boot  
Â· x-loader.bin.ift

- Â· MLO
- Â· Filesystem
- Â· rootfs.tar.gz (no integrated SGX, need to install separately)
- Â· Tools
- Â· ARM Tool Chain (pre-built - pulled from Android)
- Â· PinMux-utility
- Â· AM35x
- Â· AM37x
- Â· OMAP35x
- Â· Flashing utility
- Â· OMAP35x\_AM37x
- Â· AM35x
- Â· mk-mmc-image.script
- Â· Documentation
- Â· DevKit user guide
- Â· Release notes
- Â· Android Rowboat Porting Guide
- Â· CTS Report
- Â· DevKit Test Report
- Â· Android Rowboat Manifest
- Â· Datasheet

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-77: Android Devkit apk file

Summary:

Verify that Android Package (.apk) file is provided for the DevKit

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-79: Download Page

Summary:

Verify that the DevKit installer is distributed from TI's download page and that md5 checksums are provided for all the downloadable files

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-80: arowboat.org Download Link

Summary:

Verify that a link to TI's product download page is provided on arowboat.org

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 7 Test Suite : Tools

### Test Case amsdkA-61: Pinmux Utility

Summary:

Verify that a PinMux Utility is provided and it works

Steps:

Use the pinmux utility to generate a header file and use the generated-header file to built Uboot.

Expected Results:

Uboot should build and the enabled IPs should work after booting the DUT

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-62: Flashing Utility

Summary:

Verify that a Flashing Utility is provided and the primary/secondary bootloaders can be flashed to the DUT

Steps:

Flash Uboot to DUT and verify the DUT boots fine.

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result:	<b>Passed</b>
Build	BD_Gin_2.3.4.DEVKIT
Tester	gt_amsdk_lead

### Test Case amsdkA-63: Bootable-MMC/SD-Card-Generation script

Summary:

Verify that a script to generate a bootable MMC/SD card is provided and works fine.

Steps:

Generate a bootable MMC/SD card using the script and boot the DUT from MMC/SD

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-291: Fastboot utility

Summary:

Validate fastboot

Steps:

Following the steps in the user guide, validate fastboot functionality

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 8 Test Suite : Functionality

### 8.1 Test Suite : SGX SDK

#### Test Case amsdkA-69: SGX Demos

Summary:

Validate that Graphics Demo Application(s) is provided and runs fine.

Steps:

Run provided SGX demos

Expected Results:

Graphics display w/out artifacts and an appropriate rate

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 8.2 Test Suite : System

### Test Case amsdkA-70: System boot

Summary:

Verify that DUT boots fine w/ provided x-loader, u-boot, uImage and root filesystem

Steps:

1. Flash x-loader and u-boot to DUT using serial flashing utility
2. Set uboot environment to load provided uImage and use provided root filesystem
3. Boot the DUT

Expected Results:

DUT should boot fine and Android Home page should be shown

Test execution  
engine:

Test script or  
logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-71: System boot w/ console

Summary:

Verify that DUT boots fine w/ provided x-loader, u-boot, uImage and root filesystem and upon booting the Android console is available in the UART port

Steps:

1. Flash x-loader and u-boot to DUT using serial flashing utility
2. Set uboot environment to load provided uImage and use provided root filesystem
3. Boot the DUT
4. type "ls" in the UART console

Expected Results:

DUT should boot fine and Android console should be available in the UART port.

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**



Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

### Test Case amsdkA-86: OOB Demos

#### Summary:

Validate that the system provides icons to Demo Apps in the wallpaper upon booting

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

### Test Case amsdkA-87: RootFS over NFS

#### Summary:

Validate that the DUT boots fine when using root filesystem over NFS

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

## 8.3 Test Suite : Video

### Test Case amsdkA-283: Hardware-VRFB

Summary:

Test hardware based VRFB functionality.

Steps:

Using the Gallery app display a video in portrait and landscape orientations:

1)Send the video decode intent using the following command:

```
adb shell am start -W -n com.cooliris.media/.MovieView -a  
action.intent.ancion.VIEW -d <video path in dut> --ei  
android.intent.extra.screenOrientation <0 (landscape) or 1 (portrait)>.
```

For example,

```
adb shell am start -W -n com.cooliris.media/.MovieView -a  
action.intent.ancion.VIEW -d  
/mnt/sdcard/Video/big_buck_bunny_cif_mpeg4_aac.mp4 --ei  
android.intent.extra.screenOrientation 0
```

2)Verify that the video is displayed with the orientation specified.

Expected Results:

The video should be displayed with the orientation specified

Test execution  
engine:

Test script or  
logic:

Required hardware  
assets:

DUT parameters:

Application  
parameters:

Test Equipment  
parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

#### **Test Case amsdkA-290: S-Video output**

Summary:

Check the S-Video output

Steps:

Follow the steps in the User's guide and set the system for S-Video output

Test execution engine:

Test script or logic:

Required hardware assets:

DUT parameters:

Application parameters:

Test Equipment parameters:

test case approver:

Last Result: **Passed**  
Build BD\_Gin\_2.3.4.DEVKIT  
Tester gt\_amsdk\_lead

## **8.4 Test Suite : Graphics**

#### **Test Case amsdkA-412: Maximum Display Resolution**

Steps:

Connect the boards output to a TV.

Verify that the TV indicates the maximum resolution supported by the board

Verify that the quality of the Graphics displayed is good and that they are displayed without problems.

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead

## 8.5 Test Suite : Bluetooth

### Test Case amsdkA-477: Bluetooth Object push

Summary:

Verify that you can transfer files to the device via a bluetooth connection

Steps:

- Pair the dut with the host
- Send files to/from the host to/from dut.
- Verify that you can open the received files without any problems

Expected Results:

- The received should open without problems.

Test execution

engine:

Test script or

logic:

Required hardware

assets:

DUT parameters:

Application

parameters:

## testreport AM37x\_Gingerbread\_2.3.4\_Devkit\_2.1

Test Equipment

parameters:

test case approver:

Last Result: **Passed**

Build BD\_Gin\_2.3.4.DEVKIT

Tester gt\_amsdk\_lead