Rockchip GMS测试 多媒体模块帮助文档

文件标识：RK-PC-YF-E06

发布版本：V1.0.0

日期：2025-02-13

文件密级：□绝密 □秘密 □内部资料 ■公开

免责声明

本文档按"现状"提供，瑞芯微电子股份有限公司（"本公司"，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

"Rockchip"、"瑞芯微"、"瑞芯"均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2025 瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址： 福建省福州市铜盘路软件园A区18号

网址： www.rock-chips.com

客户服务电话： +86-4007-700-590

客户服务传真： +86-591-83951833

客户服务邮箱： fae@rock-chips.com

前言

概述

本文提供了GMS测试过程中多媒体模块常见fail项的处理方法。

产品版本

Android 13及以上版本

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V1.0.0	刘兴亮	2025-02-13	初始版本

目录

1. 概述

GMS测试包中包含大量多媒体相关（media/audio）的模块，测试过程中会出现各种各样的fail，一些fail项需要找代码开发者去解决，一些fail项可以通过修改配置自行解决。本文提供了测试模块快速测试方法以及常见fail项解决办法。

2. 命令行测试

GMS测试正常流程是需要机器连接VPN并通过GMS测试包完成测试，测试时间相对比较久，在debug过程中如果通过正常流程去验证修改，效率偏低。

可以通过adb shell用命令行的方式通过am instrument启动Instrumentation测试。

以下以CtsMediaV2TestCases为例详细说明。

2.1 测试环境准备

2.1.1 资源包下载及push到机器

CtsMediaV2TestCases对应资源包可以通过CtsMediaV2TestCases.dynamic查看

文件路径：

```
android-cts-15-r2/android-cts/testcases/CtsMediaV2TestCases
```

具体内容如下：

```
<dynamicConfig>
    <entry key="media_files_url">

 <value>https://dl.google.com/android/xts/cts/tests/media/CtsMediaV2TestCases-
5.0.zip</value>
    </entry>
</dynamicConfig>
```

下载CtsMediaV2TestCases-5.0.zip后解压，执行里面的copy_media.sh，会将资源文件push到测试机器对应目录下。

2.1.2 测试应用安装及权限设置

apk路径：

```
android-cts-15-r2/android-
cts/testcases/CtsMediaV2TestCases/arm64/CtsMediaV2TestCases.apk
```

apk安装：

```
adb install CtsMediaV2TestCases.apk
```

权限设置：

```
Settings--Apps--All apps--android.mediav2.cts--Permissions--PHOTOS and videos--
Allow--CONFIRM
```

2.2 测试命令

以CtsMediaV2TestCases中某一个case为例单测说明。

某一单项case：

```
android.mediav2.cts.EncoderColorAspectsTest#testColorAspectsEndToEnd[176_c2.rk.av
c.encoder_video/avc_unknown_bt709_unknown_surfacergb888_0-bframes]
```

测试完整命令：

```
am instrument -w -r -e class
android.mediav2.cts.EncoderColorAspectsTest#testColorAspectsEndToEnd[176_c2.rk.av
c.encoder_video/avc_unknown_bt709_unknown_surfacergb888_0-bframes]
android.mediav2.cts/androidx.test.runner.AndroidJUnitRunner
```

测试结果：

```
rk3588_u:/ # am instrument -w -r -e class
android.mediav2.cts.EncoderColorAspectsTest#testColorAspectsEndToEnd[176_c2.rk.av
c.encoder_video/avc_unknown_bt709_unknown_surfacergb888_0-bframes]
android.mediav2.cts/androidx.test.runner.AndroidJUnitRunner
INSTRUMENTATION_STATUS: class=android.mediav2.cts.EncoderColorAspectsTest
INSTRUMENTATION_STATUS: current=1
INSTRUMENTATION_STATUS: id=AndroidJUnitRunner
INSTRUMENTATION_STATUS: numtests=1
INSTRUMENTATION_STATUS: stream=
android.mediav2.cts.EncoderColorAspectsTest:
INSTRUMENTATION_STATUS:
test=testColorAspectsEndToEnd[176_c2.rk.avc.encoder_video/avc_unknown_bt709_unkno
wn_surfacergb888_0-bframes]
INSTRUMENTATION_STATUS_CODE: 1
INSTRUMENTATION_STATUS: class=android.mediav2.cts.EncoderColorAspectsTest
INSTRUMENTATION_STATUS: current=1
INSTRUMENTATION_STATUS: id=AndroidJUnitRunner
INSTRUMENTATION_STATUS: numtests=1
INSTRUMENTATION_STATUS: stream=.
```

```
INSTRUMENTATION_STATUS:
test=testColorAspectsEndToEnd[176_c2.rk.avc.encoder_video/avc_unknown_bt709_unkno
wn_surfacergb888_0-bframes]
INSTRUMENTATION_STATUS_CODE: 0
INSTRUMENTATION_RESULT: stream=

Time: 0.315

OK (1 test)


INSTRUMENTATION_CODE: -1
```

2.2.1 命令行说明

2.2.1.1 命令行格式

```
am instrument -w -r -e class testcase test_package/runner_class
```

2.2.1.2 参数说明

```
-w:保持adb shell打开直至测试完成
-r:以原始形式输出测试结果
-e <NAME> <VALUE>: set argument <NAME> to <VALUE>.提供过滤器和参数
```

2.2.1.3 获取test_package/runner_class

通过pm list instrumentation找test_package/runner_class

```
rk3588_u:/ # pm list instrumentation
instrumentation:android.mediav2.cts/androidx.test.runner.AndroidJUnitRunner
(target=android.mediav2.cts)
```

CtsMediaV2TestCases的test_package/runner_class是：
android.mediav2.cts/androidx.test.runner.AndroidJUnitRunner

2.2.1.4 testcase

- 整个模块

  testcase：android.mediav2.cts
- 模块中的一个大类集合

  testcase：android.mediav2.cts.EncoderColorAspectsTest
- 模块中的一个大类集合中的某一个case

testcase：
android.mediav2.cts.EncoderColorAspectsTest#testColorAspectsEndToEnd[176_c2.rk.avc.encoder_video/avc_unknown_bt709_unknown_surfacergb888_0-bframes]

3. CTS常见fail项处理办法

3.1 CtsMediaPlayerTestCases

3.1.1 网络相关

```
android.media.player.cts.StreamingMediaPlayerTest#testHTTP_H264Base_AAC_Video1
android.media.player.cts.StreamingMediaPlayerTest#testHTTP_H264Base_AAC_Video2
android.media.player.cts.StreamingMediaPlayerTest#testHTTP_MPEG4SP_AAC_Video1
android.media.player.cts.StreamingMediaPlayerTest#testHTTP_MPEG4SP_AAC_Video2
android.media.player.cts.StreamingMediaPlayerTest#testHTTP_H263_AMR_Video1
android.media.player.cts.StreamingMediaPlayerTest#testHTTP_H263_AMR_Video2
```

- 报错

  ```
  org.junit.runners.model.TestTimedOutException: test timed out after 600000
  milliseconds
  ```

- 原因分析

  这个case是播放在线视频，对网络要求比较高，如果网络环境不好，会出现这种10分钟超时的报错。

- 解决办法

  更换vpn或者通过本地媒体包的方式去测试

```
下载 https://storage.googleapis.com/cts_media/CtsMediaTestCases.zip
下载 https://storage.googleapis.com/android_media/cts/tests/tests/media/player/CtsMediaPlayerTestCases-1.0.zip

adb push CtsMediaTestCases /sdcard/
cd CtsMediaPlayerTestCases-1.0
./copy_media.sh

run cts --module-arg CtsMediaPlayerTestCases:config-url:https://storage.googleapis.com/cts_media/DynamicConfig_local.json -m CtsMediaPlayerTestCases
```

3.2 CtsMediaCodecTestCases

3.2.1 网络相关

```
android.media.codec.cts.MediaCodecCapabilitiesTest#testAvcHigh31
android.media.codec.cts.MediaCodecCapabilitiesTest#testAvcHigh40
android.media.codec.cts.MediaCodecCapabilitiesTest#testAvcBaseline12
android.media.codec.cts.MediaCodecCapabilitiesTest#testAvcBaseline3
```

解决办法：

更换vpn或者使用本地媒体包的方式测试：

```
下载 https://storage.googleapis.com/cts_media/CtsMediaTestCases.zip
下载
https://storage.googleapis.com/android_media/cts/tests/tests/media/codec/CtsMedia
CodecTestCases-1.0.zip

adb push CtsMediaTestCases /sdcard/
cd CtsMediaCodecTestCases-1.0
./copy_media.sh

run cts --module-arg CtsMediaCodecTestCases:config-
url:https://storage.googleapis.com/cts_media/DynamicConfig_local.json -m
CtsMediaCodecTestCases
```

3.3 CtsVideoTestCases

3.3.1 performance-point问题

- 测试项

  ```
  android.video.cts.CodecDecoderPerformanceTest#testPerformanceOfHardwareVideoD
  ecoders[112(c2.rk.vp9.decoder_1_2.5)]
  ```

- 报错

  ```
  java.lang.AssertionError: Unable to achieve the expected rate. DecodeMime:
  video/x-vnd.on2.vp9, Decoder: c2.rk.vp9.decoder, resolution: 2160p, Key-
  priority: 1 :: act/exp fps: 41.586958329867755/57.0
  ```

- 分析

  rk356x对应的vendor/rockchip/common/vpu/etc/media_codecs_c2_rk356x.xml文件中c2.rk.vp9.decoder的配置如下：

```
        <MediaCodec name="c2.rk.vp9.decoder" type="video/x-vnd.on2.vp9">
            <Limit name="size" min="320x182" max="4096x2160" />
            <Limit name="alignment" value="2x2" />
            <Limit name="block-size" value="16x16" />
            <Limit name="block-count" range="1-32768" /> <!-- max 4096x2160 -
->
            <Limit name="blocks-per-second" range="1-1966080" />
            <Limit name="frame-rate" range="1-60" />
            <Limit name="bitrate" range="1-20000000" />
            <Feature name="adaptive-playback" />
            <Limit name="concurrent-instances" max="32" />
            <Limit name="performance-point-3840x2160" value="60" />
        </MediaCodec>
```

Android支持的standard point如下：

```
List<VideoCapabilities.PerformancePoint> standardPoints = Arrays.asList(
      VideoCapabilities.PerformancePoint.UHD_240,/*3840x2160@240*/
      VideoCapabilities.PerformancePoint.UHD_200,/*3840x2160@200*/
      VideoCapabilities.PerformancePoint.UHD_120,/*3840x2160@120*/
      VideoCapabilities.PerformancePoint.UHD_100,/*3840x2160@100*/
      VideoCapabilities.PerformancePoint.UHD_60, /*3840x2160@60*/
      VideoCapabilities.PerformancePoint.UHD_50, /*3840x2160@50*/
      VideoCapabilities.PerformancePoint.UHD_30, /*3840x2160@30*/
      VideoCapabilities.PerformancePoint.UHD_25, /*3840x2160@25*/
      VideoCapabilities.PerformancePoint.UHD_24, /*3840x2160@24*/
      VideoCapabilities.PerformancePoint.FHD_240,/*1920x1080@240*/
      VideoCapabilities.PerformancePoint.FHD_200,/*1920x1080@200*/
      VideoCapabilities.PerformancePoint.FHD_120,/*1920x1080@120*/
      VideoCapabilities.PerformancePoint.FHD_100,/*1920x1080@100*/
      VideoCapabilities.PerformancePoint.FHD_60, /*1920x1080@60*/
      VideoCapabilities.PerformancePoint.FHD_50, /*1920x1080@50*/
      VideoCapabilities.PerformancePoint.FHD_30, /*1920x1080@30*/
      VideoCapabilities.PerformancePoint.FHD_25, /*1920x1080@25*/
      VideoCapabilities.PerformancePoint.FHD_24, /*1920x1080@24*/
      VideoCapabilities.PerformancePoint.HD_240, /*1280x720@240*/
      VideoCapabilities.PerformancePoint.HD_200, /*1280x720@200*/
      VideoCapabilities.PerformancePoint.HD_120, /*1280x720@120*/
      VideoCapabilities.PerformancePoint.HD_100, /*1280x720@100*/
      VideoCapabilities.PerformancePoint.HD_60,  /*1280x720@60*/
      VideoCapabilities.PerformancePoint.HD_50,  /*1280x720@50*/
      VideoCapabilities.PerformancePoint.HD_30,  /*1280x720@30*/
      VideoCapabilities.PerformancePoint.HD_25,  /*1280x720@25*/
      VideoCapabilities.PerformancePoint.HD_24,  /*1280x720@24*/
      VideoCapabilities.PerformancePoint.SD_60,  /*720x480@60*/
      VideoCapabilities.PerformancePoint.SD_50,  /*720x576@50*/
      VideoCapabilities.PerformancePoint.SD_48,  /*720x480@48*/
      VideoCapabilities.PerformancePoint.SD_30,  /*720x480@30*/
      VideoCapabilities.PerformancePoint.SD_25,  /*720x576@25*/
      VideoCapabilities.PerformancePoint.SD_24); /*720x480@24*/
```

报错分析如下：

```
exp: 60 * 0.95 = 57  act: 41.5
测试实际能达到的fps = 41.5 / 0.95 = 43.1
43介于30和50之间，若想测试通过且符合谷歌标准，performance-point-3840x2160的value需改成30
```

- 解决办法

```
diff --git a/vpu/etc/media_codecs_c2_rk356x.xml
b/vpu/etc/media_codecs_c2_rk356x.xml
index b3288c6..0fb8e75 100644
--- a/vpu/etc/media_codecs_c2_rk356x.xml
+++ b/vpu/etc/media_codecs_c2_rk356x.xml
@@ -50,7 +50,7 @@
             <Limit name="bitrate" range="1-20000000" />
             <Feature name="adaptive-playback" />
             <Limit name="concurrent-instances" max="32" />
-            <Limit name="performance-point-3840x2160" value="60" />
+            <Limit name="performance-point-3840x2160" value="30" />
         </MediaCodec>
         <MediaCodec name="c2.rk.vp8.decoder" type="video/x-vnd.on2.vp8">
             <Limit name="size" min="2x2" max="1920x1088" />
```

注意：以上是以rk356x为例说明，其它芯片平台修改方法一致，请修改对应的media_codecs_c2_rkxxx.xml

3.4 CtsMediaDecoderTestCases

3.4.1 帧率问题

- 测试项

```
android.media.decoder.cts.VideoDecoderPerfTest#testPerf[0(c2.rk.avc.decoder:qvga)]
```

- 报错

```
E TestRunner: failed: testPerf[0(c2.rk.avc.decoder:qvga)]
(android.media.decoder.cts.VideoDecoderPerfTest)
E TestRunner: ----- begin exception -----
E TestRunner: java.lang.AssertionError: Expected achievable frame rates for
c2.rk.avc.decoder video/avc 320x240: [500.0, 560.0].
E TestRunner: Measured frame rate: [135.4609736934789, 135.4713543799203].
E TestRunner:  expected null, but was:<Expected achievable frame rates for
c2.rk.avc.decoder video/avc 320x240: [500.0, 560.0].
E TestRunner: Measured frame rate: [135.4609736934789, 135.4713543799203].
E TestRunner: >
E TestRunner:   at org.junit.Assert.fail(Assert.java:89)
E TestRunner:   at org.junit.Assert.failNotNull(Assert.java:756)
E TestRunner:   at org.junit.Assert.assertNull(Assert.java:738)
```

- 解决办法

报错信息会提示配置，以rk3568对应的xml为例，配置的是[500.0, 560.0]，实际测试才[135.4609736934789, 135.4713543799203]，说明原来配置的偏大，重新配置将实际测试的结果包含进去即可，可配置[130, 140]，如下补丁所示：

```
diff --git a/vpu/etc/media_codecs_performance_rk356x.xml
b/vpu/etc/media_codecs_performance_rk356x.xml
index e763b0f..d5dafac 100644
--- a/vpu/etc/media_codecs_performance_rk356x.xml
+++ b/vpu/etc/media_codecs_performance_rk356x.xml
@@ -88,7 +88,7 @@
         </MediaCodec>
         <!-- c2_decoder performance list -->
         <MediaCodec name="c2.rk.avc.decoder" type="video/avc" update="true">
-            <Limit name="measured-frame-rate-320x240" range="500-560" />
+            <Limit name="measured-frame-rate-320x240" range="130-140" />
             <Limit name="measured-frame-rate-720x480" range="480-520" />
             <Limit name="measured-frame-rate-1280x720" range="310-315" />
             <Limit name="measured-frame-rate-1920x1080" range="164-164" />
```

xml路径vendor/rockchip/common/vpu/etc，其它芯片平台修改对应xml即可，所有这类问题修改方法都一致。

3.5 CtsMediaAudioTestCases

3.5.1 testTimestamp

- 测试项

```
android.media.audio.cts.AudioRecordTest#testTimestamp
```

- 报错

```
java.lang.AssertionError: AudioRecordTest expect
maxAbsJitterMs(31.848597038199177) &lt; 6.0
```

- 解决办法

```
//hardware/rockchip/audio/tinyalsa_hal
static struct pcm_config pcm_config_in = {
    #if PCM_REFERENCE_CHANNELS
    .channels = PCM_CAPTURE_CHANNELS + PCM_REFERENCE_CHANNELS,
#else
    .channels = PCM_CAPTURE_CHANNELS,
#endif
    .rate = 48000,
    .period_size = 480,    // 10ms
    .period_count = 4,
    .format = PCM_FORMAT_S16_LE,
};
```

.period_size改成240或者768

3.5.2 usb相关

- 测试项

```
android.media.audio.cts.AudioNativeTest#testRecordStreamData
android.media.audio.cts.AudioRecordSharedAudioTest#testCapturesMatch
```

- 报错

```
E modules.usbaudio.audio_hal: adev_release_audio_patch cannot find stream
with patch handle as 189
E modules.usbaudio.audio_hal: adev_create_audio_patch() the patch handle(202)
does not match recorded one(0) for stream with handle(5918) when creating
audio patch
```

- 解决办法

  hardware/libhardware/modules/usbaudio

```
diff --git a/modules/usbaudio/audio_hal.c b/modules/usbaudio/audio_hal.c
old mode 100644
new mode 100755
index fe921d69..43229938
--- a/modules/usbaudio/audio_hal.c
+++ b/modules/usbaudio/audio_hal.c
@@ -1338,6 +1338,7 @@ static int adev_open_input_stream(struct
audio_hw_device *hw_dev,
     int num_open_inputs = in->adev->inputs_open;
     device_unlock(in->adev);
+    num_open_inputs = 0;
     /* Check if an input stream is already open */
     if (num_open_inputs > 0) {
         if (!profile_is_cached_for(&device_info->profile, card, device)) {
@@ -1687,6 +1688,7 @@ static int adev_create_audio_patch(struct
audio_hw_device *dev,
     if (are_devices_the_same(
                 num_configs, cards, devices, num_saved_devices, saved_cards,
saved_devices)) {
         // The new devices are the same as original ones. No need to update.
+        *patch_handle = *handle;
         stream_unlock(lock);
         return 0;
     }
```

4. GTS相关

4.1 GTS本地媒体包测试

- 使用本地的媒体包最大的好处就是能够排除网络问题，实现视频秒播。
- GTS测试中使用本地媒体包，可以访问Google说明源地址或从FTP服务器下载`GMS-Test-Suite/xTS/GTS_local_media_package`

1. 下载GtsExoPlayerTestCases、GtsYouTubeTestCases以及GtsMediaTestCases
2. push本地媒体包到机器sdcard

```
adb push gts test wvmedia /sdcard/
```

3. 使用命令测试：

```
run gts \
--module-arg GtsExoPlayerTestCases:config-
url:https://storage.googleapis.com/exoplayer-test-media-1/gen-4/dynamic-config-
sdcard-1.0.json \
--module-arg "GtsYouTubeTestCases:skip-media-download:true" \
--module-arg "GtsYouTubeTestCases:instrumentation-arg:media-path:=/sdcard/test" \
--module-arg "GtsMediaTestCases:instrumentation-arg:media-
path:=file:///sdcard/wvmedia"
```

4. 模块单测命令(注意替换模块名`WvtsDeviceTestCases`)

```
run gts -m GtsExoPlayerTestCases \
--module-arg GtsExoPlayerTestCases:config-
url:https://storage.googleapis.com/exoplayer-test-media-1/gen-4/dynamic-config-
sdcard-1.0.json \
--module-arg "GtsExoPlayerTestCases:instrumentation-arg:media-path:=/sdcard/gts"

run gts -m GtsYouTubeTestCases \
--module-arg GtsYouTubeTestCases:config-
url:https://storage.googleapis.com/exoplayer-test-media-1/gen-4/dynamic-config-
sdcard-1.0.json \
--module-arg "GtsYouTubeTestCases:instrumentation-arg:media-path:=/sdcard/test"

run gts -m WvtsDeviceTestCases \
--module-arg WvtsDeviceTestCases:config-
url:https://storage.googleapis.com/exoplayer-test-media-1/gen-4/dynamic-config-
sdcard-1.0.json \
--module-arg "WvtsDeviceTestCases:instrumentation-arg:media-
path:=file:///sdcard/wvmedia"
```

常见GTS测试fail项需要用本地媒体包测试

```
WvtsDeviceTestCases
com.google.android.wvts.WidevineH264PlaybackTests#testL3WithUHD60
com.google.android.wvts.WidevineH264PlaybackTests#testClearWithUHD60
com.google.android.wvts.WidevineVP9WebMPlaybackTests#testVP9WebMCencSuperFramesL3WithUHD30
com.google.android.wvts.WidevineVP9WebMPlaybackTests#testVP9WebMProfile210bitCencSuperFramesL3With1080P30
com.google.android.wvts.WidevineVP9WebMPlaybackTests#testVP9WebMCencSubSampleL3WithUHD60
```