



android.opengl.GLES20.glTexImage2D crash with buffer with offset

+1 1 Hotlists (3) Mark as Duplicate

Comments (6) Dependencies Duplicates (0) Blocking (0) Resources (1)

Fixed Bug P3 + Add Hotlist [AOSP] assigned

STATUS UPDATE No update yet. Edit

DESCRIPTION ga...@gmail.com created issue #1 Jul 3, 2019 12:05AM

When I call glTexImage2D with a buffer with offset 0, there is no problem. But when I change the position or use a sub-buffer, it crashes.

Android Version: 8.1.0

Crash brief:

signal 6 (SIGABRT), code -6 (SI_TKILL)
Abort message: 'Invalid address 0xc1d2400c passed to free: value not allocated'
07-02 20:35:16.031 19705 19705 F DEBUG : backtrace:
07-02 20:35:16.031 19705 19705 F DEBUG : #00 pc 0001a7ee /system/lib/libc.so (abort+63)
07-02 20:35:16.031 19705 19705 F DEBUG : #01 pc 0006c0f5 /system/lib/libc.so (ifree+600)
07-02 20:35:16.031 19705 19705 F DEBUG : #02 pc 0006c2f7 /system/lib/libc.so (je_free+70)
07-02 20:35:16.031 19705 19705 F DEBUG : #03 pc 002afa71 /system/lib/libart.so
(art::JNI::ReleasePrimitiveArrayCritical(_JNIEnv*, _jarray*, void*, int)+556)
07-02 20:35:16.031 19705 19705 F DEBUG : #04 pc 000d1ebf /system/lib/libart.so
(art::CheckJNI::ReleasePrimitiveArrayCritical(_JNIEnv*, _jarray*, void*, int)+502)
07-02 20:35:16.031 19705 19705 F DEBUG : #05 pc 00087a2b /system/lib/libandroid_runtime.so
(android_glTexImage2D_IIIIIIIIJava_nio_Buffer_2(_JNIEnv*, _jobject*, int, int, int, int, int, int, int, int, _jobject*)+126)
07-02 20:35:16.031 19705 19705 F DEBUG : #06 pc 00d2a28d /system/framework/arm/boot-framework.oat (offset 0x5f2000) (android.opengl.GLES10.glCompressedTexSubImage2D [DEDUPED]+180)
.....

I examined the source, and found a problem with android_glTexImage2D_IIIIIIIIJava_nio_Buffer_2.
https://android.googlesource.com/platform/frameworks/base/+master/core/jni/android_opengl_GLES20.cpp:
4574

```
static void
android_glTexImage2D_IIIIIIIIJava_nio_Buffer_2
(JNIEnv * _env, jobject _this, jint target, jint level, jint internalformat, jint width, jint height, jint border, jint format,
jint type, jobject pixels_buf) {
    jint _exception = 0;
    const char * _exceptionType = NULL;
    const char * _exceptionMessage = NULL;
    jarray _array = (jarray) 0;
    jint _bufferOffset = (jint) 0;
    jint _remaining;
    GLvoid *pixels = (GLvoid *) 0;
    if (pixels_buf) {
        pixels = (GLvoid *)getPointer(_env, pixels_buf, (jarray*)&_array, &_remaining, &_bufferOffset);
    }
    if (pixels_buf && pixels == NULL) {
        char * _pixelsBase = (char *)_env->GetPrimitiveArrayCritical(_array, (jboolean *) 0); //! _pixelsBase not
saved
        pixels = (GLvoid *) (_pixelsBase + _bufferOffset);
    }
    glTexImage2D(
        (GLenum)target,
        (GLint)level,
        (GLint)internalformat,
        (GLsizei)width,
        (GLsizei)height,
        (GLint)border,
        (GLenum)format,
        (GLenum)type,
        (GLvoid *)pixels
    );
    if (_array) {
        releasePointer(_env, _array, pixels, JNI_FALSE); //! calling ReleasePrimitiveArrayCritical, _pixelsBase shall
be passed instead of pixels
    }
}
```


Reporter ga...@gmail.com
Type Bug
Priority P3
Severity S3
Status Fixed
Access Default access View
Assignee am...@google.com
Verifier --
Collaborators
CC am...@google.com
ga...@gmail.com
AOSP ID --
ReportedBy Developer
Found In --
Targeted To --
Verified In --
In Prod

```
}  
if (_exception) {  
    jniThrowException(_env, _exceptionType, _exceptionMessage);  
}  
}
```

COMMENTS

All comments ▼

↓ Oldest first

 **am...@google.com** <am...@google.com> [#2](#) Jul 13, 2019 09:13PM ⋮
Assigned to am...@google.com.

Thank you for reporting this issue. We've shared this with our product and engineering teams and will continue to provide updates as more information becomes available.

 **am...@google.com** <am...@google.com> [#3](#) Jul 20, 2019 01:53PM ⋮

gaolerofhell@,
Please help us with the below information.
We need instructions for how to reproduce, a bug report, which device is being used, what revision of Android (e.g. build number).

 **ga...@gmail.com** <ga...@gmail.com> [#4](#) Jul 28, 2019 09:58PM ⋮

To reproduce, run the project in the attachment.

```
byte[] bytes = new byte[1280 * 720];  
for (int k = 0; k < bytes.length; k += 1)  
{  
    bytes[k] = (byte)(k);  
}  
GLES20.glActiveTexture(GLES20.GL_TEXTURE0);  
GLES20.glBindTexture(GLES20.GL_TEXTURE_2D, background_texture_id.get(0));  
  
ByteBuffer buffer = ByteBuffer.wrap(bytes);  
GLES20.glTexImage2D(GLES20.GL_TEXTURE_2D, 0, GLES20.GL_LUMINANCE, 8, 8, 0,  
GLES20.GL_LUMINANCE, GLES20.GL_UNSIGNED_BYTE, buffer);  
  
buffer.position(1280 * 360);  
//crashes on the following line  
GLES20.glTexImage2D(GLES20.GL_TEXTURE_2D, 0, GLES20.GL_LUMINANCE, 8, 8, 0,  
GLES20.GL_LUMINANCE, GLES20.GL_UNSIGNED_BYTE, buffer);
```

Device: BlackBerry Key2 BBF100-4

Android version: 8.1.0

I can not find a revision for this, but I think the bug is on all devices with a recent Android version, as the code on master is not fixed.

 **deleted**
0 B ?

 Restricted

 **ga...@gmail.com** <ga...@gmail.com> [#5](#) Dec 10, 2019 10:23PM ⋮

As you don't have any progress in the past months, and I recently get some Pixel test devices, I can now provide the following logs on Android 9 and Android 10 on Pixel devices.

Model: Pixel

Android version 10 Build Number QP1A.190711.020

```
12-10 06:14:39.025 6539 6576 F libc : Fatal signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x0 in tid 6576 (GLThread 562), pid 6539 (com.test.opengl)  
12-10 06:14:39.124 6588 6588 F DEBUG : *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***  
12-10 06:14:39.124 6588 6588 F DEBUG : Build fingerprint:  
'google/sailfish/sailfish:10/QP1A.190711.020/5800535:user/release-keys'  
12-10 06:14:39.125 6588 6588 F DEBUG : Revision: '0'  
12-10 06:14:39.125 6588 6588 F DEBUG : ABI: 'arm64'  
12-10 06:14:39.126 6588 6588 F DEBUG : Timestamp: 2019-12-10 06:14:39-0500  
12-10 06:14:39.126 6588 6588 F DEBUG : pid: 6539, tid: 6576, name: GLThread 562 >>>  
com.test.opengl <<<
```

```
12-10 06:14:39.126 6588 6588 F DEBUG : uid: 10000
12-10 06:14:39.126 6588 6588 F DEBUG : signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x0
12-10 06:14:39.126 6588 6588 F DEBUG : Cause: null pointer dereference
12-10 06:14:39.126 6588 6588 F DEBUG : x0 0000007ee546cc08 x1 0000007ef7800980 x2
0000000000000000 x3 0000007ee5365940
12-10 06:14:39.126 6588 6588 F DEBUG : x4 0000000000000001 x5 0000007f7821b000 x6
0000000000000001 x7 0000000000000000
12-10 06:14:39.126 6588 6588 F DEBUG : x8 d46443ad0f6cf376 x9 d46443ad0f6cf376 x10
00000000000000bf x11 0000000000000200
12-10 06:14:39.126 6588 6588 F DEBUG : x12 0000007ef780fd60 x13 0000007ec0000000 x14
0000000000000001 x15 0000007ee546cd40
12-10 06:14:39.126 6588 6588 F DEBUG : x16 0000007f782138d0 x17 0000007f782068d8 x18
0000007e8f7d8000 x19 0000007ee6cfb610
12-10 06:14:39.126 6588 6588 F DEBUG : x20 0000007f7841d580 x21 0000007ee546cdd8 x22
0000007ee546cc08 x23 0000007ef7800980
12-10 06:14:39.126 6588 6588 F DEBUG : x24 0000007ee6cfb5d0 x25 0000000000000000 x26
0000007ee6cfb5d8 x27 ffffffff
12-10 06:14:39.127 6588 6588 F DEBUG : x28 ffffffff x29 0000007ee6cfb840
12-10 06:14:39.127 6588 6588 F DEBUG : sp 0000007ee6cfb5d0 lr 0000007f7819b9b8 pc
0000007f7819b958
12-10 06:14:39.133 6588 6588 F DEBUG :
12-10 06:14:39.133 6588 6588 F DEBUG : backtrace:
12-10 06:14:39.133 6588 6588 F DEBUG : #00 pc 000000000079958
/apex/com.android.runtime/lib64/bionic/libc.so (je_tcache_bin_flush_small+1024) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #01 pc 00000000007b558
/apex/com.android.runtime/lib64/bionic/libc.so (tcache_flush_cache+40) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #02 pc 00000000007b944
/apex/com.android.runtime/lib64/bionic/libc.so (tcache_destroy+52) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #03 pc 00000000007d320
/apex/com.android.runtime/lib64/bionic/libc.so (je_tsd_cleanup+84) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #04 pc 00000000007da8c
/apex/com.android.runtime/lib64/bionic/libc.so (tsd_cleanup_wrapper+32) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #05 pc 0000000000e3dec
/apex/com.android.runtime/lib64/bionic/libc.so (pthread_key_clean_all()+116) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #06 pc 0000000000e384c
/apex/com.android.runtime/lib64/bionic/libc.so (pthread_exit+72) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #07 pc 0000000000e3730
/apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+40) (BuildId:
5812256023147338b8a9538321d4c456)
12-10 06:14:39.133 6588 6588 F DEBUG : #08 pc 000000000084004
/apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64) (BuildId:
5812256023147338b8a9538321d4c456)
```

Model: Pixel 3

Android version 9 Build Number PQ1A.181105.017.A1

```
12-10 06:15:55.576 29691 29717 F libc : Fatal signal 7 (SIGBUS), code 1 (BUS_ADRALN), fault addr
0x78f4b814 in tid 29717 (GLThread 1172), pid 29691 (com.test.opengl)
12-10 06:15:55.577 29691 29719 E libc : Access denied finding property
"vendor.gralloc.enable_ahardware_buffer"
12-10 06:15:55.584 625 625 E libc : Access denied finding property "vendor.debug.egl.swapinterval"
12-10 06:15:55.610 29724 29724 F DEBUG : *** *** *** *** *** *** *** *** *** *** ***
12-10 06:15:55.610 29724 29724 F DEBUG : Build fingerprint:
'google/blueline/blueline:9/PQ1A.181105.017.A1/5081125:user/release-keys'
12-10 06:15:55.610 29724 29724 F DEBUG : Revision: 'MP1.0'
12-10 06:15:55.610 29724 29724 F DEBUG : ABI: 'arm64'
12-10 06:15:55.610 29724 29724 F DEBUG : pid: 29691, tid: 29717, name: GLThread 1172 >>>
com.test.opengl <<<
12-10 06:15:55.610 29724 29724 F DEBUG : signal 7 (SIGBUS), code 1 (BUS_ADRALN), fault addr
0x78f4b814
12-10 06:15:55.610 29724 29724 F DEBUG : x0 0000006fdf68a338 x1 0000000078f4b824 x2
0000000078f4b824 x3 0000000000000004
12-10 06:15:55.610 29724 29724 F DEBUG : x4 0000000000000000 x5 00000000ffffff x6
00000000ffffff x7 000007f000000440
12-10 06:15:55.610 29724 29724 F DEBUG : x8 0000000078f4b814 x9 0000000000000001 x10
0000000000000000 x11 0000006ff63fbec0
12-10 06:15:55.610 29724 29724 F DEBUG : x12 0000000000000000 x13 0000000000000000 x14
000000000000000d x15 0000000000000000
12-10 06:15:55.610 29724 29724 F DEBUG : x16 000000707a070be0 x17 000000707a00395c x18
```

```
000000005def7e6b x19 0000006fefa401e0
12-10 06:15:55.610 29724 29724 F DEBUG : x20 0000000000000004 x21 0000006ff6237040 x22
0000000078f4b80c x23 0000006fdf68c588
12-10 06:15:55.610 29724 29724 F DEBUG : x24 0000006fdf68c588 x25 0000000000000001 x26
000000000000007ec x27 00000000000000438
12-10 06:15:55.610 29724 29724 F DEBUG : x28 0000000078f4b824 x29 0000006fdf68a3a0
12-10 06:15:55.610 29724 29724 F DEBUG : sp 0000006fdf68a330 lr 0000006fecd09bbc pc
0000006fecd09bbc
12-10 06:15:55.675 29724 29724 F DEBUG :
12-10 06:15:55.675 29724 29724 F DEBUG : backtrace:
12-10 06:15:55.675 29724 29724 F DEBUG : #00 pc 00000000000003bcc
/vendor/lib64/libgralloccore.so (gralloc::BufferManager::RegisterHandleLocked(private_handle_t const*,
int, int)+140)
12-10 06:15:55.675 29724 29724 F DEBUG : #01 pc 00000000000003ec8
/vendor/lib64/libgralloccore.so (gralloc::BufferManager::ImportHandleLocked(private_handle_t*)+116)
12-10 06:15:55.675 29724 29724 F DEBUG : #02 pc 0000000000000429c
/vendor/lib64/libgralloccore.so (gralloc::BufferManager::RetainBuffer(private_handle_t const*)+236)
12-10 06:15:55.675 29724 29724 F DEBUG : #03 pc 000000000000043c4
/vendor/lib64/hw/android.hardware.graphics.mapper@2.0-impl-qt-
display.so (vendor::qti::hardware::display::Mapper::V1_0::implementation::QtiMapper::importBuffer(android
d::hardware::hidl_handle const&, std::__1::function<void
(android::hardware::graphics::mapper::V2_0::Error, void*)>)+92)
12-10 06:15:55.675 29724 29724 F DEBUG : #04 pc 00000000000012184
/system/lib64/android.hardware.graphics.mapper@2.1.so (android::hardware::graphics::mapper::V2_1::
BsMapper::importBuffer(android::hardware::hidl_handle const&, std::__1::function<void
(android::hardware::graphics::mapper::V2_0::Error, void*)>)+156)
12-10 06:15:55.675 29724 29724 F DEBUG : #05 pc 0000000000001536c /system/lib64/libui.so
(android::Gralloc2::Mapper::importBuffer(android::hardware::hidl_handle const&, native_handle const**)
const+88)
12-10 06:15:55.675 29724 29724 F DEBUG : #06 pc 00000000000018808 /system/lib64/libui.so
(android::GraphicBufferMapper::importBuffer(native_handle const*, unsigned int, unsigned int, unsigned
int, int, unsigned long, unsigned int, native_handle const**)+136)
12-10 06:15:55.675 29724 29724 F DEBUG : #07 pc 00000000000017788 /system/lib64/libui.so
(android::GraphicBuffer::unflatten(void const*&, unsigned long&, int const*&, unsigned long&)+636)
12-10 06:15:55.675 29724 29724 F DEBUG : #08 pc 000000000000142f50
/system/lib64/libandroid_runtime.so
(android::Parcel::FlattenableHelper<android::GraphicBuffer>::unflatten(void const*, unsigned long, int
const*, unsigned long)+56)
12-10 06:15:55.675 29724 29724 F DEBUG : #09 pc 00000000000069908 /system/lib64/libbinder.so
(android::Parcel::read(android::Parcel::FlattenableHelperInterface&) const+476)
12-10 06:15:55.675 29724 29724 F DEBUG : #10 pc 000000000000738d0 /system/lib64/libgui.so
(android::BpGraphicBufferProducer::requestBuffer(int, android::sp<android::GraphicBuffer>*)+236)
12-10 06:15:55.675 29724 29724 F DEBUG : #11 pc 00000000000084124 /system/lib64/libgui.so
(android::Surface::dequeueBuffer(ANativeWindowBuffer**, int*)+744)
12-10 06:15:55.675 29724 29724 F DEBUG : #12 pc 000000000000b028
/vendor/lib64/egl/eglSubDriverAndroid.so (EglAndroidWindowSurface::DequeueBuffer()+132)
12-10 06:15:55.675 29724 29724 F DEBUG : #13 pc 000000000000a82c
/vendor/lib64/egl/eglSubDriverAndroid.so (EglAndroidWindowSurface::GetBuffer(EglSubResource*,
EglMemoryDesc*)+64)
12-10 06:15:55.675 29724 29724 F DEBUG : #14 pc 0000000000002a414c
/vendor/lib64/egl/libGLESv2_adreno.so (EglWindowSurface::UpdateResource(EsxContext*)+112)
12-10 06:15:55.675 29724 29724 F DEBUG : #15 pc 0000000000002a3a98
/vendor/lib64/egl/libGLESv2_adreno.so (EglWindowSurface::GetResource(EsxContext*, EsxResource**,
EsxResource**, int)+56)
12-10 06:15:55.675 29724 29724 F DEBUG : #16 pc 0000000000002a2100
/vendor/lib64/egl/libGLESv2_adreno.so (EglSurface::UpdateAuxResource(EsxContext*,
EglSurface::EglAuxResource, int, EsxResource**)+116)
12-10 06:15:55.675 29724 29724 F DEBUG : #17 pc 0000000000001ef3a4
/vendor/lib64/egl/libGLESv2_adreno.so (EsxContext::AcquireBackBuffer(int)+2252)
12-10 06:15:55.675 29724 29724 F DEBUG : #18 pc 0000000000002c5b0c
/vendor/lib64/egl/libGLESv2_adreno.so (EsxCmdMgr::Flush(EsxFlushReason)+3004)
12-10 06:15:55.675 29724 29724 F DEBUG : #19 pc 0000000000001f138c
/vendor/lib64/egl/libGLESv2_adreno.so (EsxContext::SwapBuffers(EsxRect const*, unsigned int)+172)
12-10 06:15:55.675 29724 29724 F DEBUG : #20 pc 0000000000002927f4
/vendor/lib64/egl/libGLESv2_adreno.so (EglContext::SwapBuffers(EsxRect const*, unsigned int)+84)
12-10 06:15:55.675 29724 29724 F DEBUG : #21 pc 0000000000002a48a4
/vendor/lib64/egl/libGLESv2_adreno.so
(EglWindowSurface::PerformContextSwapOperation(EglContext*, EsxRect const*, unsigned int,
EglResourceAccessInfo*)+92)
12-10 06:15:55.675 29724 29724 F DEBUG : #22 pc 0000000000002a393c
/vendor/lib64/egl/libGLESv2_adreno.so (EglWindowSurface::SwapBuffers(EglContext*, EsxRect const*,
unsigned int)+52)
12-10 06:15:55.675 29724 29724 F DEBUG : #23 pc 00000000000028ec80
/vendor/lib64/egl/libGLESv2_adreno.so (EglApi::SwapBuffers(void*, void*)+388)
12-10 06:15:55.675 29724 29724 F DEBUG : #24 pc 00000000000017080 /system/lib64/libEGL.so
(eglSwapBuffersWithDamageKHR+572)
12-10 06:15:55.675 29724 29724 F DEBUG : #25 pc 000000000000b743c
```

```

/system/lib64/libandroid_runtime.so (android::jni_eglSwapBuffers(_JNIEnv*, _object*, _object*,
_jobobject*)+104)
12-10 06:15:55.675 29724 29724 F DEBUG : #26 pc 00000000003de034
/system/framework/arm64/boot-framework.oat (offset 0x3cd000)
(android.hardware.usb.UsbDeviceConnection.native_open [DEDUPED]+180)
12-10 06:15:55.675 29724 29724 F DEBUG : #27 pc 0000000000545b88 /system/lib64/libart.so
(art_quick_invoke_stub+584)
12-10 06:15:55.675 29724 29724 F DEBUG : #28 pc 0000000000cf698 /system/lib64/libart.so
(art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char const*)+200)
12-10 06:15:55.675 29724 29724 F DEBUG : #29 pc 000000000027a978 /system/lib64/libart.so
(art::interpreter::ArtInterpreterToCompiledCodeBridge(art::Thread*, art::ArtMethod*, art::ShadowFrame*,
unsigned short, art::JValue*)+344)
12-10 06:15:55.675 29724 29724 F DEBUG : #30 pc 00000000002749b8 /system/lib64/libart.so
(bool art::interpreter::DoCall<false, false>(art::ArtMethod*, art::Thread*, art::ShadowFrame&,
art::Instruction const*, unsigned short, art::JValue*)+948)
12-10 06:15:55.675 29724 29724 F DEBUG : #31 pc 0000000000516760 /system/lib64/libart.so
(MterpInvokeInterface+1392)
12-10 06:15:55.675 29724 29724 F DEBUG : #32 pc 0000000000538314 /system/lib64/libart.so
(ExecuteMterpImpl+14740)
12-10 06:15:55.675 29724 29724 F DEBUG : #33 pc 0000000000ad4caa /system/framework/boot-
framework.vdex (android.opengl.GLSurfaceView$EglHelper.swap+12)
12-10 06:15:55.675 29724 29724 F DEBUG : #34 pc 000000000024e938 /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadERKNS_20CodeItemDataAccessorERNS_11ShadowFrame
ENS_6JValueEb.llvm.3830762457+488)
12-10 06:15:55.675 29724 29724 F DEBUG : #35 pc 0000000000254098 /system/lib64/libart.so
(art::interpreter::ArtInterpreterToInterpreterBridge(art::Thread*, art::CodeItemDataAccessor const&,
art::ShadowFrame*, art::JValue*)+216)
12-10 06:15:55.675 29724 29724 F DEBUG : #36 pc 000000000027499c /system/lib64/libart.so
(bool art::interpreter::DoCall<false, false>(art::ArtMethod*, art::Thread*, art::ShadowFrame&,
art::Instruction const*, unsigned short, art::JValue*)+920)
12-10 06:15:55.675 29724 29724 F DEBUG : #37 pc 00000000005157f0 /system/lib64/libart.so
(MterpInvokeVirtual+588)
12-10 06:15:55.675 29724 29724 F DEBUG : #38 pc 0000000000538114 /system/lib64/libart.so
(ExecuteMterpImpl+14228)
12-10 06:15:55.675 29724 29724 F DEBUG : #39 pc 0000000000ad53d8 /system/framework/boot-
framework.vdex (android.opengl.GLSurfaceView$GLThread.guardedRun+1142)
12-10 06:15:55.675 29724 29724 F DEBUG : #40 pc 000000000024e938 /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadERKNS_20CodeItemDataAccessorERNS_11ShadowFrame
ENS_6JValueEb.llvm.3830762457+488)
12-10 06:15:55.675 29724 29724 F DEBUG : #41 pc 0000000000254098 /system/lib64/libart.so
(art::interpreter::ArtInterpreterToInterpreterBridge(art::Thread*, art::CodeItemDataAccessor const&,
art::ShadowFrame*, art::JValue*)+216)
12-10 06:15:55.675 29724 29724 F DEBUG : #42 pc 000000000027499c /system/lib64/libart.so
(bool art::interpreter::DoCall<false, false>(art::ArtMethod*, art::Thread*, art::ShadowFrame&,
art::Instruction const*, unsigned short, art::JValue*)+920)
12-10 06:15:55.675 29724 29724 F DEBUG : #43 pc 0000000000516b24 /system/lib64/libart.so
(MterpInvokeDirect+296)
12-10 06:15:55.675 29724 29724 F DEBUG : #44 pc 0000000000538214 /system/lib64/libart.so
(ExecuteMterpImpl+14484)
12-10 06:15:55.675 29724 29724 F DEBUG : #45 pc 0000000000ad5880 /system/framework/boot-
framework.vdex (android.opengl.GLSurfaceView$GLThread.run+48)
12-10 06:15:55.675 29724 29724 F DEBUG : #46 pc 000000000024e938 /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadERKNS_20CodeItemDataAccessorERNS_11ShadowFrame
ENS_6JValueEb.llvm.3830762457+488)
12-10 06:15:55.675 29724 29724 F DEBUG : #47 pc 00000000005064ac /system/lib64/libart.so
(artQuickToInterpreterBridge+1032)
12-10 06:15:55.675 29724 29724 F DEBUG : #48 pc 000000000054ecfc /system/lib64/libart.so
(art_quick_to_interpreter_bridge+92)
12-10 06:15:55.675 29724 29724 F DEBUG : #49 pc 0000000000545b88 /system/lib64/libart.so
(art_quick_invoke_stub+584)
12-10 06:15:55.675 29724 29724 F DEBUG : #50 pc 0000000000cf698 /system/lib64/libart.so
(art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char const*)+200)
12-10 06:15:55.675 29724 29724 F DEBUG : #51 pc 000000000044f524 /system/lib64/libart.so (art::
(anonymous namespace)::InvokeWithArgArray(art::ScopedObjectAccessAlreadyRunnable const&,
art::ArtMethod*, art::(anonymous namespace)::ArgArray*, art::JValue*, char const*)+104)
12-10 06:15:55.675 29724 29724 F DEBUG : #52 pc 00000000004505f0 /system/lib64/libart.so
(art::InvokeVirtualOrInterfaceWithJValues(art::ScopedObjectAccessAlreadyRunnable const&, _jobobject*,
_jmethodID*, jvalue*)+424)
12-10 06:15:55.675 29724 29724 F DEBUG : #53 pc 000000000047add0 /system/lib64/libart.so
(art::Thread::CreateCallback(void*)+1260)
12-10 06:15:55.675 29724 29724 F DEBUG : #54 pc 00000000000821e0 /system/lib64/libc.so
(__pthread_start(void*)+36)
12-10 06:15:55.675 29724 29724 F DEBUG : #55 pc 0000000000023178 /system/lib64/libc.so
(__start_thread+68)

```



The issue has been fixed and it will be available in a future release.