



← ↻ ☆ Crash around Skia on Android 11

+1

Hotlists (2)

Mark as Duplicate



Comments (5) Dependencies Duplicates (0) Blocking (0) Resources (3)

Obsolete

Bug

P3

+ Add Hotlist



STATUS UPDATE No update yet.

Edit



DESCRIPTION ch...@amazon.com created issue #1

This is a crash reported from Google play console. We don't additional log and cannot reproduce this crash in house. Stack trace is as below. In the past 30 days, it has around 2.2K crash reports
It only happens on Android 11. See here: <https://drive.google.com/file/d/1pxs-Qvuh2X5ZroGQAU1ui9RIP9BYL9uD/view?usp=sharing>

pid: 0, tid: 0 >>> com.amazon.kindle <<<

```
backtrace: #00 pc 000000000010918 /system/lib64/libutils.so (android::RefBase::incStrong(void const*) const+8) #00 pc 0000000000f6a08 /system/lib64/libandroid_runtime.so (android::na
(android.view.SurfaceControl$Transaction.hide+80) #00 pc 000000000090b3cc /system/framework/arm64/boot-framework.oat (android.view.SurfaceView$1.positionLost+524) #00 pc 0000000
(art_quick_invoke_stub+548) #00 pc 00000000001a8a78 /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char const*)+200) #
std::__va_list)+468) #00 pc 0000000000556ae8 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithVarArgs<_jmethodID*>(art::ScopedObjectAccessAlreadyRunni
00000000001acc88 /system/lib64/libhwui.so (_JNIEnv::CallVoidMethod(_jobject*, _jmethodID*, ...) +124) #00 pc 00000000001b1084 /system/lib64/libhwui.so (_ZZN7androidL46android_view_F
00000000001f3230 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+2772) #00 pc 00000000
(android::uirenderer::RenderNode*, android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+632) #00 pc 00000000001f2b84 /system/lib64/libhwui.so (android::uirenderer::Rer
(android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool, std::__1::function<void (android::uirenderer::Re
android::uirenderer::TreeInfo&, bool)+1064) #00 pc 00000000001cdbcc /system/lib64/libhwui.so (android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::T
/system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+1064) #00 pc 00000000001cdbcc /system
android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+632) #00 pc 00000000001f2b84 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android
(android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool, std::__1::function<void (android::uirenderer::Re
android::uirenderer::TreeInfo&, bool)+1064) #00 pc 00000000001cdbcc /system/lib64/libhwui.so (android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::T
/system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+1064) #00 pc 00000000001cdbcc /system
android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+632) #00 pc 00000000001f2b84 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android
(android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool, std::__1::function<void (android::uirenderer::Re
android::uirenderer::TreeInfo&, bool)+1064) #00 pc 00000000001f25d4 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTree(android::uirenderer::TreeInfo&)+144) #00 pc 0000
(android::uirenderer::renderthread::CanvasContext::prepareTree(android::uirenderer::TreeInfo&, long*, long, android::uirenderer::RenderNode*)+380) #00 pc 0000000000220500 /system/lib64/libl
/system/lib64/libhwui.so (android::uirenderer::WorkQueue::process()+220) #00 pc 0000000000230024 /system/lib64/libhwui.so (android::uirenderer::renderthread::RenderThread::threadLoop()+
00000000000afce8 /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64) #00 pc 00000000000502c8 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)
```

✓ Links (3)

↔ Links (3)

"This is a crash reported from Google play console. We don't additional log and cannot reproduce this crash in house. Stack trace is as below. In the past 30 days, it has around 2.2K crash reports. In I
"It only happens on Android 11. See here: <https://drive.google.com/file/d/1pxs-Qvuh2X5ZroGQAU1ui9RIP9BYL9uD/view?usp=sharing>"
"Is "later releases" Android 12 or Android 11 patch? If it's Android 11 patch, as per <https://source.android.com/setup/start/build-numbers> , does the latest patch "RQ3A.210705.001" have the fix?"

COMMENTS



ch...@amazon.com <ch...@amazon.com> #2

Format stack trace as below:

```
*** ** pid: 0, tid: 0 >>> com.amazon.kindle <<<

backtrace:
#00 pc 000000000010918 /system/lib64/libutils.so (android::RefBase::incStrong(void const*) const+8)
#00 pc 0000000000f8628 /system/lib64/libandroid_runtime.so (android::nativeSetFlags(_JNIEnv*, _jclass*, long, long, int, int)+84)
#00 pc 0000000000202930 /system/framework/arm64/boot-framework.oat (art_jni_trampoline+176)
#00 pc 000000000091adf0 /system/framework/arm64/boot-framework.oat (android.view.SurfaceControl$Transaction.hide+80)
#00 pc 000000000091e62c /system/framework/arm64/boot-framework.oat (android.view.SurfaceView$1.positionLost+524)
#00 pc 00000000004bef50 /system/framework/arm64/boot-framework.oat (android.graphics.RenderNode$CompositePositionUpdateListener.positionLost+112)
#00 pc 0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#00 pc 00000000001a8a78 /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char cc
#00 pc 0000000000554fb4 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithVarArgs<art::ArtMethod*>(art::ScopedObject
#00 pc 00000000000555154 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithVarArgs<_jmethodID*>(art::ScopedObjectAcce
#00 pc 00000000003ab850 /apex/com.android.art/lib64/libart.so (art::JNI<false>::CallVoidMethodV(_JNIEnv*, _jobject*, _jmethodID*, std::__va_list)+660)
#00 pc 00000000001acc88 /system/lib64/libhwui.so (_JNIEnv::CallVoidMethod(_jobject*, _jmethodID*, ...) +124)
#00 pc 00000000001b1084 /system/lib64/libhwui.so (_ZZN7androidL46android_view_RenderNode_requestPositionUpdatesEP7_JNIEnvP8_jobjectIS3_EN26PositionLis
#00 pc 00000000001f3230 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirend
#00 pc 00000000001cdbcc /system/lib64/libhwui.so (android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::Tree
#00 pc 00000000001f2b84 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirend
#00 pc 00000000001cdbcc /system/lib64/libhwui.so (android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::Tree
#00 pc 00000000001f2b84 /system/lib64/libhwui.so (android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirend
```

Assigned to an...@google.com.



Reassigned to jr...@google.com.



Status: Won't Fix (Obsolete)

I believe this SurfaceView bug has already been fixed in later releases. There's not really anything we can do otherwise.



Hi, there

Is "later releases" Android 12 or Android 11 patch? If it's Android 11 patch, as per <https://source.android.com/setup/start/build-numbers>, does the latest patch "RQ3A.210705.001" have the fix?
Meanwhile, anything Application can do to avoid the crash? thanks.



I don't believe any of the fixes here were backported to Android 11, so it'd be a fixed in Android 12 issue.

But yes I believe apps can avoid the crash. If I remember correctly this crash results from adding & removing a SurfaceView in one frame (so add the SurfaceView in frame N, and remove it in