Show 1 additional field

COMMENTS All comments → Oldest first ey...@gmail.com <ey...@gmail.com><u>#2</u> Jan 24, 2020 07:53PM Just found this native crash on the play store console. Same device model and os version (Samsung S8 dreamqltesq running Android 9). *** *** *** *** *** *** *** *** *** *** *** *** *** pid: 0, tid: 0 >>> co.twenty.android <<< backtrace: #00 pc 00000000012ae20 /system/vendor/lib64/egl/libGLESv2_adreno.so (EsxFramebufferObject::DetachAll(EsxContext*)+56) #01 pc 0000000001fd9fc /system/vendor/lib64/egl/libGLESv2_adreno.so (EglContext::UnmakeCurrentEsx()+308) #02 pc 0000000001fd870 /system/vendor/lib64/egl/libGLESv2_adreno.so (EglContext::UnmakeCurrent()+128) #03 pc 0000000001fe838 /system/vendor/lib64/egl/libGLESv2_adreno.so (EglThreadList::SearchCurrentDisplayAndUnmake(EglDisplay*)+224) #04 pc 0000000001fe418 /system/vendor/lib64/egl/libGLESv2_adreno.so (EglDisplay::Terminate(int)+168) #05 pc 0000000001f4814 /system/vendor/lib64/egl/libGLESv2_adreno.so (EglApi::Terminate(void*)+84) #06 pc 0000000001115c /system/lib64/libEGL.so (android::egl_display_t::terminate()+128) $\#07\ pc\ 00000000006f918\ / system/lib64/libgui.so\ (and roid::GLC on sumer::Egllmage:: \sim Egllmage() + 80)$

Brand: samsung Model: Galaxy S8 Orientation: Portrait

RAM free: 998.49 MB Disk free: 38.54 GB Operating System Version: 9 Orientation: Portrait Rooted: No Crash

Date: Jan 23, 2020, 11:39:00 PM

```
#08 pc 00000000006f75c /system/lib64/libgui.so (android::GLConsumer::abandonLocked()+68)
 #09 pc 000000000065a40 /system/lib64/libqui.so (android::ConsumerBase::onLastStrongRef(void const*)+48)
 #10 pc 0000000001687bc /system/lib64/libandroid_runtime.so (android::SurfaceTexture_release(_JNIEnv*, _jobject*)+80)
 #11 pc 00000000042982c /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.graphics.Camera.nativeConstructor [DEDUPED]+124)
 #12 pc 000000000088ebb8 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.graphics.SurfaceTexture.release+40)
 #13 pc 000000000d99d1c /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.TextureView.releaseSurfaceTexture+252)
 #14 pc 0000000000d9a788 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.TextureView.onDetachedFromWindowInternal+248)
 #15 pc 000000000cd22e0 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.View.dispatchDetachedFromWindow+192)
 #16 pc 000000000da1a00 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewGroup.dispatchDetachedFromWindow+240)
 #17 pc 000000000da1a00 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewGroup.dispatchDetachedFromWindow+240)
 #18 pc 000000000da1a00 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewGroup.dispatchDetachedFromWindow+240)
 #19 pc 000000000da1a00 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewGroup.dispatchDetachedFromWindow+240)
 #20 pc 000000000da1a00 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewGroup.dispatchDetachedFromWindow+240)
 #21 pc 000000000da1a00 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewGroup.dispatchDetachedFromWindow+240)
 #22 pc 000000000000564c /system/framework/arm64/boot-framework.oat (offset 0x420000)
(and roid.view. ViewRootImpl. dispatchDetachedFromWindow + 188)\\
 #23 pc 0000000000d05fc4 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewRootImpl.doDie+180)
 #24 pc 000000000d050f4 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.ViewRootImpl.die+68)
 #25 pc 0000000000c0cc0c /system/framework/arm64/boot-framework.oat (offset 0x420000)
(and roid.view. Window Manager Global.remove View Locked + 316)\\
 #26 pc 0000000000c0e14c /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.WindowManagerGlobal.removeView+220)
 #27 pc 000000000dae124 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.view.WindowManagerImpl.removeViewImmediate+68)
 #28 pc 000000000090980c /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.app.ActivityThread.handleDestroyActivity+316)
 #29 pc 0000000000af9b88 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.app.servertransaction.DestroyActivityItem.execute+136)
 #30 pc 000000000813480 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.app.servertransaction.TransactionExecutor.executeLifecycleState+416)
 #31 pc 000000000813b10 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.app.servertransaction.TransactionExecutor.execute+304)
 #32 pc 0000000008f6918 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.app.ActivityThread$H.handleMessage+1112)
 #33 pc 000000000b271a4 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.os.Handler.dispatchMessage+180)
 #34 pc 000000000b2a2a0 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.os.Looper.loop+1264)
 #35 pc 00000000009037e8 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(android.app.ActivityThread.main+680)
 #36 pc 00000000055844c /system/lib64/libart.so (art_quick_invoke_static_stub+604)
 #37 pc 0000000000cfce8 /system/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int,
art::JValue*. char const*)+232)
 #38 pc 00000000045e62c /system/lib64/libart.so (art::(anonymous
namespace)::InvokeWithArgArray(art::ScopedObjectAccessAlreadyRunnable const&, art::ArtMethod*, art::(anonymous
namespace)::ArgArray*, art::JValue*, char const*)+104)
 #39 pc 000000000460080 /system/lib64/libart.so (art::InvokeMethod(art::ScopedObjectAccessAlreadyRunnable const&,
iobiect*, iobiect*, iobiect*, unsigned long)+1440)
 #40 pc 0000000003ef51c /system/lib64/libart.so (art::Method_invoke(_JNIEnv*, _jobject*, _jobject*, _jobjectArray*)+52)
 #41 pc 00000000011f7e4 /system/framework/arm64/boot.oat (offset 0x115000)
(iava.lang.Class.getDeclaredMethodInternal [DEDUPED]+180)
 #42 pc 0000000000e12708 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run+136)
 #43 pc 000000000e192b0 /system/framework/arm64/boot-framework.oat (offset 0x420000)
(com.android.internal.os.Zygotelnit.main+2208)
 #44 pc 00000000055844c /system/lib64/libart.so (art_quick_invoke_static_stub+604)
 #45 pc 0000000000cfce8 /system/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int,
art::JValue*. char const*)+232)
 #46 pc 00000000045e62c /system/lib64/libart.so (art::(anonymous
namespace)::InvokeWithArgArray(art::ScopedObjectAccessAlreadyRunnable const&, art::ArtMethod*, art::(anonymous
namespace)::ArgArray*, art::JValue*, char const*)+104)
 #47 pc 000000000045e28c /system/lib64/libart.so (art::InvokeWithVarArgs(art::ScopedObjectAccessAlreadyRunnable
const&. iobiect*. imethodID*. std:: va list)+424)
 #48 pc 000000000362d88 /system/lib64/libart.so (art::JNI::CallStaticVoidMethodV(_JNIEnv*, _jclass*, _jmethodID*,
std:: va list)+652)
 #49 pc 00000000000b934c /system/lib64/libandroid_runtime.so (_JNIEnv::CallStaticVoidMethod(_jclass*, _jmethodID*,
 #50 pc 0000000000bbfa4 /system/lib64/libandroid_runtime.so (android::AndroidRuntime::start(char const*,
android::Vector<android::String8> const&, bool)+768)
 #51 pc 0000000000004b44 /system/bin/app_process64 (main+1832)
 #52 pc 00000000000c9e24 /system/lib64/libc.so (__libc_init+88)
```

Hi There,

Thanks for filing this issue, we'll be taking a look in the coming months. Thanks!

sc...@google.com <sc...@google.com><u>#4</u>

Feb 5, 2020 01:18AM

Hi eygraber,

Can you share your code snippet which calls imageCapture.takePicture and the OnImageSavedCallback callback implementation?

This throwable CameraClosedException is wrapped in a ImageCaptureException which is passed to OnImageSavedCallback.onError.

The purpose is to simply notify the takePicture call fails due to CameraClosedException. It doesn't look like it will raise a crash.

Scott

er...@google.com <er...@google.com>

Feb 5, 2020 05:32AM

Reassigned to ey...@gmail.com.

ey...@gmail.com <ey...@gmail.com>#5

Feb 5, 2020 05:56AM

Just did some further investigation. It looks like this is happening in a scenario where the user takes a photo using the front lens with flash enabled. We animate a bright white screen for a few hundred milliseconds to emulate a flash, and during that time, the user managed to click the button that changes the camera lens.

sc...@google.com <sc...@google.com>#6

Feb 5, 2020 09:28PM

Hi eygraber,

Specifically, I'd like to know what you app did in the onError of ImageCapture.onImageSavedCallback because CameraX don't cause a crash when the CameraClosedException happens. CameraClosedException is just used to notify apps why the takePicture fails. Is it possible that your app throw the exception in onError()?

ey...@gmail.com <ey...@gmail.com><u>#7</u>

Feb 6, 2020 05:37AM

Yes, we would throw the error there.

 $sc...@google.com < sc...@google.com > \underline{\#8} \\$

Feb 7, 2020 03:19PM :

then I would say it is working as intended. It is expected if camera is closed in the middle of takePicture flow, onError is called with CameraClosedException. So I think it does not make sense for you app to throw the exception. Instead you should just do proper clean up or show failure dialog only only when your app is still in foreground.

er...@google.com <er...@google.com>

Feb 11, 2020 10:26AM

Status: Won't Fix (Intended Behavior)