Android Public Tracker > App Development > Android Studio > Build Tools > Dexer (D8) 80166962

android.enableD8=true cause native crash on some Android 6.0/8.0 device

+1 25 Hotlists (4) Mark as Duplicate

Comments (57) Dependencies Duplicates (2) Blocking (0) Resources (14)

Assigned Bug P2 + Add Hotlist device specific JIT bugs affecting D8

STATUS UPDATE No update yet. Edit

DESCRIPTION ja...@gmail.com created issue #1 May 23, 2018 08:00PM

All the following crash appear after update Android Studio 3.1 (stable), with D8 enabled by default.
My current temporary workaround is: set `android.enableD8=false` in `gradle.properties`

Note:
Crash log 1: ALL devices with Android 6.0
Crash log 2: ALL devices with Android 6.0
Crash log 3: ALL devices with Android 8.0
Crash log 4: ALL devices with Android 6.0
Crash log in Pre-launch report: Galaxy S7 Edge with Android 6.0 (attachment logcat.txt)

I found a lot native crash in "Android Vitals - ANRs & crashes"

~~~~
### Crash log 1 ###

Crash title: /
Crash description: in tgkill
```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> com.satispay.customer <<<

backtrace:
  #00  pc 000000000006abe4  /system/lib64/libc.so (tgkill+8)
  #01  pc 0000000000068374  /system/lib64/libc.so (pthread_kill+68)
  #02  pc 00000000000212f8  /system/lib64/libc.so (raise+28)
  #03  pc 000000000001ba98  /system/lib64/libc.so (abort+60)
  #04  pc 0000000000434f74  /system/lib64/libart.so (_ZN3art7Runtime5AbortEv+324)
  #05  pc 0000000000137cd0  /system/lib64/libart.so (_ZN3art10LogMessageD2Ev+3136)
  #06  pc 000000000053ee04  /system/lib64/libart.so (artInvokeInterfaceTrampoline+1492)
  #07  pc 000000000011e1a8  /system/lib64/libart.so (art_quick_invoke_interface_trampoline+104)
  #08  pc 0000000073a7fd50  /data/dalvik-cache/arm64/system@framework@boot.oat
```
~~~~

Devices (all with Android 6.0):
Redmi Note 3 (kenzo)
P8 Lite (hwALE-H)
Y6II (HWCAM-H)
Honor 7 (HWPLK)
HM Note 2 (hermes)

~~~~
### Crash log 2 ###

Crash title: signal 11 (SIGSEGV), code 2 (SEGV_ACCERR)
Crash description: art_quick_imt_conflict_trampoline
```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> com.satispay.customer <<<

backtrace:
  #00  pc 00000000000ea768  /system/lib/libart.so (art_quick_imt_conflict_trampoline+7)
  #01  pc 0000000002f916f3  /system/framework/arm/boot.oat
```
~~~~

Devices (all with Android 6.0):
Galaxy Note4 (trlte)
Desire 816 (htc_a5ul)
LG G4c (c90n)
LG K10 (m216n)
JERRY (v2806)

~~~~
### Crash log 3 ###

Crash title: signal 11 (SIGSEGV), code 1 (SEGV_MAPERR)
Crash description:
_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8DexCacheEEE
```
~~~~

Reporter	ja...@gmail.com
Type	Bug
Priority	P2
Severity	S2
Status	Assigned
Access	Default access View
Assignee	sg...@google.com
Verifier	--
Collaborators	
CC	ag...@google.com ch...@google.com ga...@google.com ja...@gmail.com le...@google.com ... and 5 more (show a
AOSP ID	--
Blocking Release	--
Release Status	--
Found In	--
Targeted To	--
Verified In	--
In Prod	

Show 1 additional field

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> com.satispay.customer <<<

backtrace:
  #00  pc 0000000000266a4c  /system/lib64/libart.so
(_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8DexCacheEEE+352)
  #01  pc 0000000000299310  /system/lib64/libart.so
(_ZN3art11interpreterL8DoInvokeILNS_10InvokeTypeE0ELb0ELb1EEEbPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+504)
  #02  pc 000000000028e178  /system/lib64/libart.so
(_ZN3art11interpreter17ExecuteSwitchImplILb1ELb0EEENS_6JValueEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameES2_b+32968)
  #03  pc 000000000025fd40  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+664)
  #04  pc 00000000004e9bf4  /system/lib64/libart.so (artQuickToInterpreterBridge+1500)
  #05  pc 0000000000517a1c  /system/lib64/libart.so (art_quick_to_interpreter_bridge+92)
  #06  pc 000000000052f30c  /system/framework/arm64/boot.oat (java.util.TimSort.countRunAndMakeAscending+396)
  #07  pc 0000000000533da8  /system/framework/arm64/boot.oat (java.util.TimSort.sort+440)
  #08  pc 000000000044c020  /system/framework/arm64/boot.oat (java.util.Arrays.sort+128)
  #09  pc 00000000003cb684  /data/app/com.satispay.customer-kC9Wgt_afxHgi_tVYi9o0w==/oat/arm64/base.odex
```

Devices (all with Android 8.0):
10 (htc_pmeuhl)
Xperia XA1 (G3112)
Mi A1 (tissot_sprout)
Galaxy S8 (dreamlte)

Crash log 4

Crash title: signal 11 (SIGSEGV), code 1 (SEGV_MAPERR)
Crash description: base.odex

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> com.satispay.customer <<<

backtrace:
  #00  pc 0000000001b14540  /data/app/com.satispay.customer-2/oat/arm/base.odex
  #01  pc 00000000031cb921  /system/framework/arm/boot.oat
```

Devices (all with Android 6.0):
One (A0001)
Galaxy Note Edge (tblte)
Galaxy Note4 (trlte)

Crash log in Pre-launch report

Also in "Pre-launch report" from Play Console
Device: Galaxy S7 Edge

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
Build fingerprint: 'samsung/hero2ltexx/hero2lte:6.0.1/MMB29K/G935FXXU1APB6:user/release-keys'
Revision: '9'
ABI: 'arm64'
pid: 17144, tid: 20023, name: RxComputationSc  >>> com.satispay.customer <<<
signal 6 (SIGABRT), code -6 (SI_TKILL), fault addr --------
Abort message: 'art/runtime/entrypoints/quick/quick_trampoline_entrypoints.cc:2077] Check failed: instr_code == Instruction::INVOKE_INTERFACE
|| instr_code == Instruction::INVOKE_INTERFACE_RANGE Unexpected call into interface trampoline: invoke-virtual {v1, v127, v1615328776, v127,
v48}, thing@13752'
    x0   0000000000000000  x1   0000000000004e37  x2   0000000000000006  x3   0000000000000000
    x4   0000000000000000  x5   0000000000000001  x6   0000000000000000  x7   0000000000000000
    x8   0000000000000083  x9   000000000003b500  x10  00000000008f1a68  x11  00000000008f1ba8
    x12  0000000000000000  x13  0000007f7d302000  x14  0000000000000000  x15  0000000000000000
    x16  0000007f7d2f4568  x17  0000007f7d287378  x18  0000007f7d305f50  x19  0000007f59b93500
    x20  0000007f59b93440  x21  000000000000000d  x22  0000000000000006  x23  0000007f7977a000
    x24  0000007f6047b500  x25  0000007f77f13600  x26  0000007f797b0b80  x27  0000007f7969c000
    x28  0000007f6047b280  x29  0000007f59b8c560  x30  0000007f7d284b14
    sp   0000007f59b8c560  pc   0000007f7d287380  pstate 0000000020000000
backtrace:
    #00 pc 0000000000069380  /system/lib64/libc.so (tgkill+8)
    #01 pc 0000000000066b10  /system/lib64/libc.so (pthread_kill+68)
    #02 pc 0000000000023950  /system/lib64/libc.so (raise+28)
    #03 pc 000000000001e280  /system/lib64/libc.so (abort+60)
    #04 pc 000000000043281c  /system/lib64/libart.so (art::Runtime::Abort()+324)
    #05 pc 0000000000136364  /system/lib64/libart.so (art::LogMessage::~LogMessage()+3136)
    #06 pc 0000000000130e08  /system/lib64/libart.so (art::Barrier::~Barrier()+296)
    #07 pc 0000000000467984  /system/lib64/libart.so (art::ThreadList::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char> >&)+252)
    #08 pc 0000000000432908  /system/lib64/libart.so (art::Runtime::Abort()+560)
    #09 pc 0000000000136364  /system/lib64/libart.so (art::LogMessage::~LogMessage()+3136)
    #10 pc 000000000053c3bc  /system/lib64/libart.so (artInvokeInterfaceTrampoline+1492)
    #11 pc 000000000011db28  /system/lib64/libart.so (art_quick_invoke_interface_trampoline+104)
    #12 pc 0000000002f67fa8  /system/framework/arm64/boot.oat (offset 0x2f25000)
```

logcat - Galaxy S7 Edge.txt
4.2 MB View Download

✓ Mentioned issues (3) ✓ Links (8) Hide all

Mentioned issues (3)

P2	SIGSEGV in exception handler when D8 enabled “https://issuetracker.google.com/111337896”		ra...@ #5, ja...@ #15
P1	APK compiled with D8 crashes whole Android device when installing (even from Play Store) “https://issuetracker.google.com/111960171”		ri...@ #6
--	--	“Possibly the same issue reported in b/132103478”	ze...@ #45

Links (8)

“https://drive.google.com/file/d/1MSPcN0vikD2PYs1asVPuRcaBYV8yK_cU...”	ja...@ #3
“apk: https://drive.google.com/file/d/1sStvio2WU5O97wlMTf8CkTxG9_ec6zsB/vie... ”	ja...@ #4
“PS: related StackOverflow https://stackoverflow.com/questions/52517587/native-crash-in-system-lib64-libart-so”	ke...@ #10
“Re #11 : Yes, https://play.google.com/store/apps/details?id=ch.alpsoft.qoq... ”	ke...@ #16
“https://drive.google.com/drive/u/0/folders/1Vd-3wwUTepyrK0pnrS... ”	ja...@ #20, ja...@ #28

See all related links

COMMENTS All comments ▾ ↓ Oldest first

sg...@google.com <sg...@google.com> #2 May 24, 2018 12:22AM

Thank you for the report. Do you know of any way to reproduce these issues? I can install Satispay app, and can see that the version from May 13 is built with D8. However I cannot use the app as Danish bank accounts are not supported. However, if the "Pre-launch report" can produce a crash, then it can probably happen without creating an account. I don't know the "Pre-launch report", but does it contain the simulated user actions to get to the crash?

I am running the MMB29K build on a Nexus 5 (hammerhead-userdebug 6.0.1 BK29K 2419427 dev-keys), but that might be slightly different than the Samsung S7 image.

ja...@gmail.com <ja...@gmail.com> #3 May 24, 2018 01:15AM

For now: I cannot reproduce. I'm working on it.

Video from "Pre-launch report" with Version 3.1.3069
https://drive.google.com/file/d/1MSPcN0vikD2PYs1asVPuRcaBYV8yK_cU/view?usp=sharing
crash between: 0:53 - 0:58

ja...@gmail.com <ja...@gmail.com> #4 May 24, 2018 02:33AM

attachment: app v3.1.3069 (build with D8)
apk: https://drive.google.com/file/d/1sStvio2WU5O97wlMTf8CkTxG9_ec6zsB/view?usp=sharing

ra...@gmail.com <ra...@gmail.com> #5 Jul 12, 2018 11:52AM

Just posted a possibly related bug:

https://issuetracker.google.com/111337896 SIGSEGV in exception handler when D8 enabled

ag...@google.com <ag...@google.com> Jul 19, 2018 07:37PM

Assigned to sg...@google.com.

ri...@google.com <ri...@google.com> #6 Jul 30, 2018 09:57PM

also, possibly related:
https://buganizer.corp.google.com/issues/111960171

ze...@google.com <ze...@google.com> #7 Aug 9, 2018 09:41PM

Omar: are you still seeing this issue? If so, could you share the APK and any information about which method is active and causing the ART failure?

ja...@gmail.com <ja...@gmail.com> #8 Aug 9, 2018 11:32PM

Yes, it still happens.
In the latest version of the app, although there is `android.enabledD8 = false`, the problem seems to have come back.

Latest version of the app was build with Android Studio 3.2 Beta 5
Build #AI-181.5281.24.32.4913314, built on July 26, 2018
JRE: 1.8.0_152-release-1136-b04 x86_64
JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
macOS 10.13.6
gradle: gradle-4.8-all

The only information I have is the log of the Play Store. unfortunately I do not have the devices on which the problem happens.

ja...@gmail.com <ja...@gmail.com> #9 Aug 22, 2018 01:34AM

The reason why `android.enabledD8 = false` was ignored is because I was using the new "Signed Bundle" format instead of the classic "apk" (and probably it use D8 for build APK)

ke...@qoqa.com <ke...@qoqa.com> #10 Sep 26, 2018 11:36PM

I see the same crash reported in the Developer Console for my application. (Crash log 3: ALL devices with Android 8.0)
I use "Signed Bundle" too.

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> my.package <<<

backtrace:
  #00  pc 0000000000264298  /system/lib64/libart.so
(_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8
DexCacheEEE+352)
  #01  pc 00000000002978a8  /system/lib64/libart.so
(_ZN3art11interpreterL8DoInvokeILNS_10InvokeTypeE0ELb0ELb1EEEbPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JV
alueE+508)
  #02  pc 000000000028c824  /system/lib64/libart.so
(_ZN3art11interpreter17ExecuteSwitchImplILb1ELb0EEENS_6JValueEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameES2_b
+36540)
  #03  pc 000000000025d6fc  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+664)
  #04  pc 00000000004e4a88  /system/lib64/libart.so (artQuickToInterpreterBridge+1468)
  #05  pc 000000000051281c  /system/lib64/libart.so (art_quick_to_interpreter_bridge+92)
  #06  pc 000000000052da6c  /system/framework/arm64/boot.oat (java.util.TimSort.countRunAndMakeAscending+396)
  #07  pc 0000000000532508  /system/framework/arm64/boot.oat (java.util.TimSort.sort+440)
  #08  pc 000000000044a740  /system/framework/arm64/boot.oat (java.util.Arrays.sort+128)
  #09  pc 0000000000811f0c  /data/app/my.package-Py1SDcJah4ZBAmxAxgjfww==/oat/arm64/base.odex
```

PS: related StackOverflow https://stackoverflow.com/questions/52517587/native-crash-in-system-lib64-libart-so

ag...@google.com <ag...@google.com> #11 Oct 9, 2018 02:33AM

Kevin, can you provide a link to your app in a revision where you see this happening? Have you been able to reproduce, either on emulators or on devices?

ag...@google.com <ag...@google.com> #12 Oct 9, 2018 02:36AM

Looping in a couple of Art engineers. Nicolas, Andreas, do these stack traces (for example the one in comment #10) provide any hint as to what could be going on here?

ag...@google.com <ag...@google.com> #13 Oct 9, 2018 02:42AM

Interfaces being involved usually points towards an incompatible classpath, with duplication that isn't seen at compile time. It's strange though that we'd see this in 8.0 (unless secondary dex files are involved).

ag...@google.com <ag...@google.com> #14 Oct 9, 2018 02:42AM

Re #4: tells me the requested file doesn't exist. :-(

ja...@gmail.com <ja...@gmail.com> #15 Oct 9, 2018 02:45AM

Note: now (in the last few days) happen also for regular "apk", probably because we use "App signing" feature of Google Play Store.

I think this bug is related to https://issuetracker.google.com/issues/111337896

ke...@qoqa.com <ke...@qoqa.com> #16 Oct 9, 2018 06:24AM

Re #11 : Yes, https://play.google.com/store/apps/details?id=ch.alpsoft.qoqa.ui
I'm unable to reproduce this crash :/

ng...@google.com <ng...@google.com> #17 Oct 9, 2018 07:47PM

#1: Omar, the logcat is super useful, thanks for sharing. Do you mind also taking a bugreport and attaching it to this bug?

ja...@gmail.com <ja...@gmail.com> #18 Oct 9, 2018 08:08PM

Unfortunately I can not, these logs are extracted from Google Play Console (Android Vitals> Crashes). I have no reports even on Crashlytics.
Lately it only happens on Android 8.
It does not happen on any of our devices, even if we have models that seem to be afflicted by the problem.

Now, my most common issue are (screenshot attached):

Issue #1: signal 11 (SIGSEGV), code 1 (SEGV_MAPERR)
_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8DexCacheEEE

Description:

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> com.satispay.customer <<<

backtrace:
  #00  pc 000000000004b748  /system/lib/libc.so (tgkill+12)
  #01  pc 000000000001a4c3  /system/lib/libc.so (abort+54)
  #02  pc 0000000000339d53  /system/lib/libart.so (_ZN3art7Runtime5AbortEPKc+370)
  #03  pc 000000000033a34b  /system/lib/libart.so (_ZN3art7Runtime7AborterEPKc+10)
  #04  pc 00000000003f1f31  /system/lib/libart.so (_ZN7android4base10LogMessageD1Ev+456)
  #05  pc 00000000000aa635  /system/lib/libart.so (_ZN3art9ArtMethod19GetObsoleteDexCacheEv+328)
  #06  pc 00000000000ac6e5  /system/lib/libart.so (_ZN3art9ArtMethod12PrettyMethodEb+984)
  #07  pc 00000000000ad9c3  /system/lib/libart.so (_ZN3art9ArtMethod23GetOatQuickMethodHeaderEj+326)
  #08  pc 0000000000158591  /system/lib/libart.so (_ZN3art12FaultManager17IsInGeneratedCodeEP7siginfoPvb+696)
  #09  pc 00000000001580f9  /system/lib/libart.so (_ZN3art12FaultManager11HandleFaultEiP7siginfoPv+44)
  #10  pc 0000000000002623  /system/bin/app_process32 (_ZN3art11SignalChain7HandlerEiP7siginfoPv+382)
  #11  pc 0000000000018b4c  /system/lib/libc.so
  #12  pc 00000000001dac82  /system/lib/libart.so
(_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8
DexCacheEEE+225)
  #13  pc 0000000000201ff5  /system/lib/libart.so
(_ZN3art11interpreterL8DoInvokeILNS_10InvokeTypeE0ELb0ELb1EEEbPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JV
alueE+316)
  #14  pc 00000000001fa755  /system/lib/libart.so
(_ZN3art11interpreter17ExecuteSwitchImplILb1ELb0EEENS_6JValueEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameES2_b
+26604)
  #15  pc 00000000001d559f  /system/lib/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+570)
  #16  pc 00000000001da759  /system/lib/libart.so
(_ZN3art11interpreter30EnterInterpreterFromEntryPointEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameE+92)
  #17  pc 00000000003bfa5d  /system/lib/libart.so (artQuickToInterpreterBridge+944)
  #18  pc 00000000003e3171  /system/lib/libart.so (art_quick_to_interpreter_bridge+32)
  #19  pc 00000000004d6861  /system/framework/arm/boot.oat (java.util.TimSort.countRunAndMakeAscending+304)
  #20  pc 00000000004db1f7  /system/framework/arm/boot.oat (java.util.TimSort.sort+302)
  #21  pc 0000000000404d1d  /system/framework/arm/boot.oat (java.util.Arrays.sort+132)
  #22  pc 000000000037b541  /data/app/com.satispay.customer-_w0nZQSMFf8v6uerOLOyvQ==/oat/arm/base.odex
```

Issue #2: tgkill

Description:

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> com.satispay.customer <<<

backtrace:
  #00  pc 000000000004b748  /system/lib/libc.so (tgkill+12)
  #01  pc 000000000001a4c3  /system/lib/libc.so (abort+54)
  #02  pc 0000000000339d53  /system/lib/libart.so (_ZN3art7Runtime5AbortEPKc+370)
  #03  pc 000000000033a34b  /system/lib/libart.so (_ZN3art7Runtime7AborterEPKc+10)
  #04  pc 00000000003f1f31  /system/lib/libart.so (_ZN7android4base10LogMessageD1Ev+456)
  #05  pc 00000000000aa635  /system/lib/libart.so (_ZN3art9ArtMethod19GetObsoleteDexCacheEv+328)
  #06  pc 00000000000ac6e5  /system/lib/libart.so (_ZN3art9ArtMethod12PrettyMethodEb+984)
  #07  pc 00000000000ad9c3  /system/lib/libart.so (_ZN3art9ArtMethod23GetOatQuickMethodHeaderEj+326)
  #08  pc 0000000000158591  /system/lib/libart.so (_ZN3art12FaultManager17IsInGeneratedCodeEP7siginfoPvb+696)
  #09  pc 00000000001580f9  /system/lib/libart.so (_ZN3art12FaultManager11HandleFaultEiP7siginfoPv+44)
  #10  pc 0000000000002623  /system/bin/app_process32 (_ZN3art11SignalChain7HandlerEiP7siginfoPv+382)
  #11  pc 0000000000018b4c  /system/lib/libc.so
  #12  pc 00000000001dac82  /system/lib/libart.so
(_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8
DexCacheEEE+225)
  #13  pc 0000000000201ff5  /system/lib/libart.so
(_ZN3art11interpreterL8DoInvokeILNS_10InvokeTypeE0ELb0ELb1EEEbPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JV
alueE+316)
  #14  pc 00000000001fa755  /system/lib/libart.so
(_ZN3art11interpreter17ExecuteSwitchImplILb1ELb0EEENS_6JValueEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameES2_b
+26604)
  #15  pc 00000000001d559f  /system/lib/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+570)
```

```
 #16  pc 00000000001da759  /system/lib/libart.so
(_ZN3art11interpreter30EnterInterpreterFromEntryPointEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameE+92)
 #17  pc 00000000003bfa5d  /system/lib/libart.so (artQuickToInterpreterBridge+944)
 #18  pc 00000000003e3171  /system/lib/libart.so (art_quick_to_interpreter_bridge+32)
 #19  pc 00000000004d6861  /system/framework/arm/boot.oat (java.util.TimSort.countRunAndMakeAscending+304)
 #20  pc 00000000004db1f7  /system/framework/arm/boot.oat (java.util.TimSort.sort+302)
 #21  pc 0000000000404d1d  /system/framework/arm/boot.oat (java.util.Arrays.sort+132)
 #22  pc 000000000037b541  /data/app/com.satispay.customer-_w0nZQSMFf8v6uerOLOyvQ==/oat/arm/base.odex
```

Schermata 2018-10-09 alle 10.58.44.png
271 KB View Download

Schermata 2018-10-09 alle 11.07.02.png
156 KB View Download

ng...@google.com <ng...@google.com> #19 Oct 9, 2018 08:30PM

Do you perform dex file loading yourself, or use a library that does?

ja...@gmail.com <ja...@gmail.com> #20 Oct 9, 2018 08:52PM

No, I do not. I do not think some library does it, I add my build.gradle attached.

https://drive.google.com/drive/u/0/folders/1Vd-3wwUTepyrK0pnrS_mG-vSFDQAIzpT
(please request access)

ag...@google.com <ag...@google.com> #21 Oct 10, 2018 12:55AM

Kevin, I played around with your APK and could not reproduce either. Do you have information about the devices experiencing these crashes? Have you been able to test on an actual device? If possible, if you can provide a list of devices I can see if I can get a hold of one of the devices this is happening on in an attempt to reproduce on the actual device.

ke...@qoqa.com <ke...@qoqa.com> #22 Oct 10, 2018 01:17AM

I didn't be able to reproduce the error. I attach the reports from the PlayStore dev console.

screencapture-play-google-apps-publish-2018-10-09-16_15_00.png
548 KB View Download

ag...@google.com <ag...@google.com> #23 Oct 10, 2018 09:10PM

Thanks Kevin. I've ordered a Samsung S8 device and will attempt to repro on device when it arrives. I cannot reproduce on emulators, so it could be an issue with specific devices.

ag...@google.com <ag...@google.com> #24 Oct 16, 2018 07:38PM

Unfortunately, I cannot reproduce this on a Samsung S8 device running Android 8.0 either. Kevin, have you had any reports of the issue from your users? Do you know which steps they go through in the app when it crashes?

ke...@qoqa.com <ke...@qoqa.com> #25 Oct 16, 2018 08:08PM

No, I don't have any report about these crashes. I just see these on the console.

ja...@gmail.com <ja...@gmail.com> #26 Oct 22, 2018 08:03PM

As a temporary solution, I tried to load APKs with the original certificate (instead of the upload certificate for the Play Store) and our users no longer have native crashes.

At this point I think the application is recompiled with D8 if it is loaded with the upload certificate on the Play Store.

ag...@google.com <ag...@google.com> #27 Oct 22, 2018 08:31PM

Thanks for the update Omar. That is very puzzling. The dex code is not changed by bundle tool on the play store when shipping to Android 8.0 devices, the dex code that is in your bundle is what ends up in the APK.

Reading this again, it looks like you are seeing:

1. compile the app to a bundle specifying android.enableD8=false leads to crashes
2. compile the app to an APK with android.enableD8=false does not lead to crashes
3. compile the app to an APK with android.enableD8=true leads to crashes

Is that correct?

I would be very interested in having the APK for 1 and 2 above. Any chance you can share those with us?

That should lead to the same dex code in the APK. I would like to check that this is indeed the case. If it is something else could be going on here. Maybe a bug in bundle tool.

ja...@gmail.com <ja...@gmail.com> #28 Oct 22, 2018 09:05PM

Yes, it's correct

I build and upload in all 3 format

https://drive.google.com/drive/u/0/folders/1Vd-3wwUTepyrK0pnrS_mG-vSFDQAIzpT
(please request access)

1. 3135-without-D8.aab
2. 3135-without-D8.apk
3. 3135-with-D8.apk

ag...@google.com <ag...@google.com> #29 Oct 22, 2018 11:38PM

Great, thanks. I have requested access. :)

le...@google.com <le...@google.com> #30 Oct 23, 2018 12:54AM

Re #26: I can help investigating the part around Play Signing. We do *not* touch the dex files when an APK is signed with the upload key, so I assume this was just a coincidence, but I'd be happy to take a closer look.

Could you give me the two versions of your app:
- The one signed with upload key that crashes.
- The one signed with deployment key that doesn't crash.

It would help if you hadn't touched the source code or the value of "android.enableD8" between the two versions for easier comparison of the dex files, but anything you can provide helps.

ag...@google.com <ag...@google.com> #31 Oct 23, 2018 01:00AM

Hmm, that is strange. The difference in the two builds without D8 is that the .apk build has the classes

com.satispay.customer.R$attr
com.satispay.customer.R$styleable

in the main dex file whereas the .aab has it in classes2.dex.

I don't think there should have been a difference in those two builds. We should see if we can figure out how the dexing pipleline can be different for those two builds.

I still don't understand how that difference can lead to crashes on 8.0 devices that have native multi-dex support. But, there could of course be a bug. :-\

ag...@google.com <ag...@google.com> #32 Oct 23, 2018 01:13AM

Both of those classes are also in classes2.dex and *not* classes.dex in the D8 version. That doesn't guarantee that this is the problem, but it is interesting.

ja...@gmail.com <ja...@gmail.com> #33 Oct 23, 2018 02:21AM

Re #30: I upload all possibile build from the same source code:
- Signed with: original cert / upload cert
- Build: with D8 / without D8
- Format: ABB / APK

3136-original-cert-with-D8.aab (It probably makes no sense to use the original certificate with the ABB format)
3136-original-cert-with-D8.apk
3136-original-cert-without-D8.aab (It probably makes no sense to use the original certificate with the ABB format)
3136-original-cert-without-D8.apk
3136-upload-cert-with-D8.aab
3136-upload-cert-with-D8.apk
3136-upload-cert-without-D8.aab
3136-upload-cert-without-D8.apk

Currently, in this list, the only version that does not cause crashes is 3136-original-cert-without-D8.apk

le...@google.com <le...@google.com> #34 Oct 24, 2018 01:33AM

I notice that the dex files between "3136-original-cert-with-D8.apk" and "3136-upload-cert-with-D8.apk" are different.

```
3136-upload-cert-with-D8.apk
 10138808  ./classes2.dex
   574552  ./classes3.dex
  8752376  ./classes.dex
```

```
3136-original-cert-with-D8.apk
 10333764  ./classes2.dex
  1647544  ./classes3.dex
 10053468  ./classes.dex
```

That confirms that Play Signing is not at the source of the issue at least.

ga...@google.com <ga...@google.com> #35 Oct 24, 2018 05:41AM

I am not sure if #34 contains a typo, but all bundles/APKs produced using D8 have the same dex files, and all bundles/APKs produced using DX (without-D8) have the same dex files. Across D8 and DX, dex files differ, which is expected.

le...@google.com <le...@google.com> #36 Oct 24, 2018 08:50AM

Yeah, my bad, I seem to have mixed up the APKs.

I have however checked the last APK that was signed with the upload cert and uploaded to Play was version 3130 and the dex files are identical in the uploaded APK and the APK re-signed with Play, so I stand by the conclusion :)

ke...@qoqa.com <ke...@qoqa.com> #37 Dec 5, 2018 06:59PM

Hi! Any updates about this issue?

ag...@google.com <ag...@google.com> #38 Dec 5, 2018 08:18PM

We are still unable to reproduce the issue and therefore have no idea what is going on. This looks like an obscure VM bug to me, but we haven't been able to get a reproduction and therefore it is hard both for us and for the runtime team to actually figure out what is going on. I'm going to try again today with another APK from another reporter. Kevin, if you can reproduce this, please let us know and provide us with reproduction steps. :-\

ke...@qoqa.com <ke...@qoqa.com> #39 Dec 6, 2018 07:36PM

No, I can reproduce it myself 😟. But I have been in contact with one user that have this issue with our application. I generate an APK instead of a Bundle and... it works and doesn't crash anymore.
The device is an Sony XZ1 compact with Android 8.0.

It seems to be related to App Bundle.

ag...@google.com <ag...@google.com> #40 Dec 6, 2018 09:17PM

Thanks for the update Kevin. Is the apk and the bundle build from the same sources or is this two different source code versions? If it is the same source it would be interesting to get the APK that you get from the play store after uploading the bundle as well as the locally built APK built from the same source to see if we can figure out what the differences are. The code and the dex files should be exactly the same for those two build pipelines for Android 8.0. But if there is a difference we should understand why and that might give us more to go on.

We have reports on this issue from users that do not use app bundles and play signing, just plain old APK uploads, so it doesn't seem to be caused by app bundles, but maybe your bundle APKs are subtly different from the locally built one which might give us a hint as to what is going on.

ke...@qoqa.com <ke...@qoqa.com> #41 Dec 6, 2018 11:37PM

Same source, same commit.

ch.alpsoft.qoqa.ui-263-G8441.zip is from Play Console
qoqa-4.71.2 - ba87558d7f-release.apk is from Android Studio

https://drive.google.com/drive/folders/1kA7beIKhe5evXWztDgt7o7oxkpo-4trv?usp=sharing

ag...@google.com <ag...@google.com> #42 Dec 7, 2018 02:31AM

Hmm, that is interesting. The code in these two apks is exactly the same, except for two things:

1. Line numbers are different for a bunch of methods. The line numbers that are different look like proguard produced line numbers. I can't imagine how that could cause any issues.

2. The class initializer for the class that has been renamed to: com.qoqa.qoqa.a is different. It looks like proguard has decided to reshuffle the initialization which is a little odd.

Nicolas, Andreas, do the stack traces from the crash reports give any indication of whether class initialization is involved when the VM crashes?

ag...@google.com <ag...@google.com> #43 Dec 7, 2018 02:49AM

One more question Kevin. The user that you were in contact with: what did they do in the app when it crashed? Having more information about what might trigger the crash could be helpful. Is this at startup or is it when performing particular tasks in the app?

[Deleted User] <[Deleted User]> #44 Jan 2, 2019 01:12PM

delete

ze...@google.com <ze...@google.com> #45 May 7, 2019 03:56PM

Possibly the same issue reported in b/132103478

ni...@gmail.com <ni...@gmail.com> #46 Oct 2, 2019 06:22AM

I am having a similar backtrace to that of Crash log 3 on Android 8 devices only.
This issue is not systematic across all our Android 8 user base, but it is 100% reproducible with the same action on the app once it starts crashing.
I have a test device (OnePlus 5T) where this issue started appearing on a version of the app downloaded from the Play Store.

We are delivering apks signed with a signing key (no aab and no upload + signing key), have d8 enabled and using Proguard.

What kind of info can I gather from the device that can help you understand the issue?

ze...@google.com <ze...@google.com> #47 Oct 2, 2019 05:18PM

Thanks for the report!

We would like to see the stack trace of the crash.

Also, can you attach the apk that has the issue and a description of the actions needed to trigger the issue?

Do you have any indication of which method or context the app is in when the crash happens?

ni...@gmail.com <ni...@gmail.com> #48 Oct 2, 2019 05:29PM

Here is a crash dump from the OnePlus 3T device logcat output (I have redacted the app id).
I am going to share the apk with you later this morning.

The app crashes when the users tries to open an Activity which shows search filters: in this phase the app uses a native library through JNI to manipulate some data from an asset file.

Do you think a bugreport after the crash occurs could be of help?

Please note that about 2k users (in the last 60 days) out of 139k using Android 8.0 are experiencing this issue (as per Google Play stats).

```
    --------- beginning of crash
2019-10-01 10:16:53.384 18316-18316/? A/libc: Fatal signal 11 (SIGSEGV), code 1, fault addr 0x40 in tid 18316
2019-10-01 10:16:53.421 4714-5041/? D/AudioTrack: Uid 1000 AudioTrack::setVolume left 0.000000 right 0.000000
2019-10-01 10:16:53.417 18626-18626/? W/crash_dump64: type=1400 audit(0.0:243384): avc: denied { search } for
name="com.google.android.gms" dev="dm-1" ino=48971 scontext=u:r:crash_dump:s0:c512,c768
tcontext=u:object_r:app_data_file:s0:c512,c768 tclass=dir permissive=0
2019-10-01 10:16:53.417 18626-18626/? I/chatty: uid=10330(it.redacted.android) crash_dump64 identical 1 line
2019-10-01 10:16:53.417 18626-18626/? W/crash_dump64: type=1400 audit(0.0:243386): avc: denied { search } for
name="com.google.android.gms" dev="dm-1" ino=48971 scontext=u:r:crash_dump:s0:c512,c768
tcontext=u:object_r:app_data_file:s0:c512,c768 tclass=dir permissive=0
2019-10-01 10:16:53.441 18626-18626/? I/crash_dump64: obtaining output fd from tombstoned
2019-10-01 10:16:53.441 3164-3164/? I//system/bin/tombstoned: received crash request for pid 18316
2019-10-01 10:16:53.442 18626-18626/? I/crash_dump64: performing dump of process 18316 (target tid = 18316)
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: Build fingerprint:
'OnePlus/OnePlus3/OnePlus3T:8.0.0/OPR1.170623.032/05171658:user/release-keys'
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: Revision: '0'
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: ABI: 'arm64'
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: pid: 18316, tid: 18316, name: it.redacted.android  >>> it.redacted.android <<<
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x40
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG: Cause: null pointer dereference
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x0   0000000000000000  x1   00000073a1ec0a00  x2   0000000000000000  x3
00000000e316cbe0
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x4   0000000013055bc0  x5   0000008000000000  x6   6a4b0cff3a682e67  x7
7f7f7f7f7f7f7f7f
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x8   e1ff1ac6ccd66514  x9   e1ff1ac6ccd66514  x10  0000000000000000  x11
 000000009b529640
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x12  000000000000002f  x13  0000000000000002  x14  001679820025b5fd  x15
 00000073a1ec0a00
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x16  00000073a1bb85f0  x17  00000073a2b40190  x18  0000000000000000  x19
 000000009b5f0368
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x20  000000009b52ea28  x21  000000009b529638  x22  0000000000000000  x23
 0000007fc2efffc0
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x24  00000073a1bbdb70  x25  000000009b5f0368  x26  000000000000d427  x27
 000000009b529638
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     x28  0000000000002071  x29  0000007fc2effcd0  x30  00000073a183c7c8
2019-10-01 10:16:53.442 18626-18626/? A/DEBUG:     sp   0000007fc2effc90  pc   00000073a183c7cc  pstate 0000000060000000
2019-10-01 10:16:53.459 18626-18626/? A/DEBUG: backtrace:
2019-10-01 10:16:53.459 18626-18626/? A/DEBUG:     #00 pc 00000000002637cc  /system/lib64/libart.so
```

```
(_ZN3art6mirror5Class24ResolvedMethodAccessTestILb1ELb1ELNS_10InvokeTypeE0EEEbNS_6ObjPtrIS1_EEPNS_9ArtMethodEjNS4_INS0_8
DexCacheEEE+352)
2019-10-01 10:16:53.459 18626-18626/? A/DEBUG:     #01 pc 0000000000296ddc  /system/lib64/libart.so
(_ZN3art11interpreterL8DoInvokeILNS_10InvokeTypeE0ELb0ELb1EEEbPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JV
alueE+508)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #02 pc 000000000028bd58  /system/lib64/libart.so
(_ZN3art11interpreter17ExecuteSwitchImplILb1ELb0EEENS_6JValueEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameES2_b
+36540)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #03 pc 000000000025cc30  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+664)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #04 pc 0000000000263254  /system/lib64/libart.so
(_ZN3art11interpreter33ArtInterpreterToInterpreterBridgeEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameEPNS_6JValueE+
212)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #05 pc 000000000027bdb8  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb0EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+6
40)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #06 pc 00000000004ef6ac  /system/lib64/libart.so (MterpInvokeStatic+468)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #07 pc 00000000004f8414  /system/lib64/libart.so (ExecuteMterpImpl+14612)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #08 pc 000000000025cb54  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+444)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #09 pc 00000000004e2008  /system/lib64/libart.so
(artQuickToInterpreterBridge+1468)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #10 pc 000000000050fe1c  /system/lib64/libart.so
(art_quick_to_interpreter_bridge+92)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #11 pc 0000000000506c38  /system/lib64/libart.so
(art_quick_invoke_static_stub+600)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #12 pc 00000000000d80f8  /system/lib64/libart.so
(_ZN3art9ArtMethod6InvokeEPNS_6ThreadEPjjPNS_6JValueEPKc+260)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #13 pc 0000000000281710  /system/lib64/libart.so
(_ZN3art11interpreter34ArtInterpreterToCompiledCodeBridgeEPNS_6ThreadEPNS_9ArtMethodEPKNS_7DexFile8CodeItemEPNS_11ShadowF
rameEPNS_6JValueE+352)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #14 pc 000000000027bdd8  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb0EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+6
72)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #15 pc 00000000004ef6ac  /system/lib64/libart.so (MterpInvokeStatic+468)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #16 pc 00000000004f8414  /system/lib64/libart.so (ExecuteMterpImpl+14612)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #17 pc 000000000025cb54  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+444)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #18 pc 0000000000263254  /system/lib64/libart.so
(_ZN3art11interpreter33ArtInterpreterToInterpreterBridgeEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameEPNS_6JValueE+
212)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #19 pc 000000000027bdb8  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb0EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+6
40)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #20 pc 00000000004ef3f4  /system/lib64/libart.so (MterpInvokeDirect+504)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #21 pc 00000000004f8394  /system/lib64/libart.so (ExecuteMterpImpl+14484)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #22 pc 000000000025cb54  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+444)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #23 pc 0000000000263254  /system/lib64/libart.so
(_ZN3art11interpreter33ArtInterpreterToInterpreterBridgeEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameEPNS_6JValueE+
212)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #24 pc 000000000027bdb8  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb0EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+6
40)
2019-10-01 10:16:53.460 18626-18626/? A/DEBUG:     #25 pc 00000000004ef3f4  /system/lib64/libart.so (MterpInvokeDirect+504)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #26 pc 00000000004f8394  /system/lib64/libart.so (ExecuteMterpImpl+14484)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #27 pc 000000000025cb54  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+444)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #28 pc 0000000000263254  /system/lib64/libart.so
(_ZN3art11interpreter33ArtInterpreterToInterpreterBridgeEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameEPNS_6JValueE+
212)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #29 pc 000000000027bdb8  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb0EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+6
40)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #30 pc 00000000004f12ac  /system/lib64/libart.so (MterpInvokeVirtualQuick+680)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #31 pc 00000000004fc014  /system/lib64/libart.so (ExecuteMterpImpl+29972)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #32 pc 000000000025cb54  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+444)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #33 pc 0000000000263254  /system/lib64/libart.so
(_ZN3art11interpreter33ArtInterpreterToInterpreterBridgeEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameEPNS_6JValueE+
212)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #34 pc 000000000027bdb8  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb0EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+6
40)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #35 pc 00000000004f12ac  /system/lib64/libart.so (MterpInvokeVirtualQuick+680)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #36 pc 00000000004fc014  /system/lib64/libart.so (ExecuteMterpImpl+29972)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #37 pc 000000000025cb54  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+444)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #38 pc 0000000000263254  /system/lib64/libart.so
(_ZN3art11interpreter33ArtInterpreterToInterpreterBridgeEPNS_6ThreadEPKNS_7DexFile8CodeItemEPNS_11ShadowFrameEPNS_6JValueE+
212)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #39 pc 000000000027c548  /system/lib64/libart.so
(_ZN3art11interpreter6DoCallILb0ELb1EEEbPNS_9ArtMethodEPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JValueE+1
520)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #40 pc 0000000000296188  /system/lib64/libart.so
(_ZN3art11interpreterL8DoInvokeILNS_10InvokeTypeE4ELb0ELb1EEEbPNS_6ThreadERNS_11ShadowFrameEPKNS_11InstructionEtPNS_6JV
alueE+2084)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #41 pc 000000000028d00c  /system/lib64/libart.so
(_ZN3art11interpreter17ExecuteSwitchImplILb1ELb0EEENS_6JValueEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameES2_b
+41328)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #42 pc 000000000025cc30  /system/lib64/libart.so
(_ZN3art11interpreterL7ExecuteEPNS_6ThreadEPKNS_7DexFile8CodeItemERNS_11ShadowFrameENS_6JValueEb+664)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #43 pc 00000000004e2008  /system/lib64/libart.so
(artQuickToInterpreterBridge+1468)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #44 pc 000000000050fe1c  /system/lib64/libart.so
(art_quick_to_interpreter_bridge+92)
2019-10-01 10:16:53.461 18626-18626/? A/DEBUG:     #45 pc 0000000000350438  /data/app/it.redacted.android-
JpYctCyeW1M2zN_ZYseMhA==/oat/arm64/base.odex (offset 0x18b000)
```

mk...@google.com <mk...@google.com> #49 Oct 2, 2019 06:00PM

ninniuz@, we will attempt to recover a device such that we can reproduce the error.

If you cannot attach the APK here, then feel free to share it privately with sgjesse@google.com, mkroghj@google.com or/and zerny@google.com

Meanwhile, it would be interesting to learn a bit more about this:

1) Previous reports touches on the error may being related to app-bundles. Are you using app-bundles or any kind of resource modification. If yes, can you try and build it as an APK and see if the issue still reproduces? If you were using bundles, and the error went away when generating an APK, could you then share the two different builds? It may then be that we cannot find the asset after the app has been bundled.

2) If it is possible to modify the code, can you maybe list all the resources in the directory for where you expect to find a resource? That should be able to be done with:
AAssetDir_getNextFileName(AAssetDir *assetDir). It would be interesting to see if the expected asset is actually there.

3) You mention "native library through JNI". Is that an open source library we can take a look at our one that is internal?

ni...@gmail.com <ni...@gmail.com> #50 Oct 2, 2019 06:34PM

Please request access to this drive folder for the apk:
https://drive.google.com/drive/folders/1CbAmvntrh205WawFQ7qj-LGTOt02gMpR?usp=sharing

Re your questions:

1) Never used app-bundles on this app.
2) Actually the asset file is read into a String by the app and handed to 3rd party library which wraps it into a ByteArrayInputStream before feeding it into the JNI code
3) It is a slight modification of com.naef.jnlua library

mk...@google.com <mk...@google.com> #51 Oct 3, 2019 11:14PM

it seems like error is coming from the optimized dex (.odex) which is generated at some point when the phone is not in use and plugged in. Therefore, uninstalling and installing will fix the issue, but maybe only temporarily.

When running this on a Samsung device with a freshly installed APK I get the following ClassNotFoundException printed - but the app is not crashing so this may be unrelated.

```
10-03 14:39:43.199 16506 16506 D ViewRootImpl@b59d33e[ZonePolygonsActivity]: ViewPostIme pointer 1
10-03 14:39:43.207  3756  4360 E Parcel  : Class not found when unmarshalling: it.redacted.android.model.b.c
10-03 14:39:43.207  3756  4360 E Parcel  : java.lang.ClassNotFoundException: it.redacted.android.model.b.c
10-03 14:39:43.207  3756  4360 E Parcel  :      at java.lang.Class.classForName(Native Method)
10-03 14:39:43.207  3756  4360 E Parcel  :      at java.lang.Class.forName(Class.java:453)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.Parcel.readParcelableCreator(Parcel.java:2843)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.Parcel.readParcelable(Parcel.java:2797)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.Parcel.readValue(Parcel.java:2700)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.Parcel.readArrayMapInternal(Parcel.java:3067)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.BaseBundle.unparcel(BaseBundle.java:257)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.BaseBundle.getString(BaseBundle.java:1086)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.content.Intent.getStringExtra(Intent.java:7706)
10-03 14:39:43.207  3756  4360 E Parcel  :      at com.android.server.am.ActivityStarter.startActivity(ActivityStarter.java:468)
10-03 14:39:43.207  3756  4360 E Parcel  :      at com.android.server.am.ActivityStarter.startActivityLocked(ActivityStarter.java:419)
10-03 14:39:43.207  3756  4360 E Parcel  :      at com.android.server.am.ActivityStarter.startActivityMayWait(ActivityStarter.java:1289)
10-03 14:39:43.207  3756  4360 E Parcel  :      at
com.android.server.am.ActivityManagerService.startActivityAsUser(ActivityManagerService.java:6014)
10-03 14:39:43.207  3756  4360 E Parcel  :      at
com.android.server.am.ActivityManagerService.startActivity(ActivityManagerService.java:5867)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.app.IActivityManager$Stub.onTransact(IActivityManager.java:121)
10-03 14:39:43.207  3756  4360 E Parcel  :      at
com.android.server.am.ActivityManagerService.onTransact(ActivityManagerService.java:3844)
10-03 14:39:43.207  3756  4360 E Parcel  :      at android.os.Binder.execTransact(Binder.java:682)
10-03 14:39:43.207  3756  4360 E Parcel  : Caused by: java.lang.ClassNotFoundException: it.redacted.android.model.b.c
10-03 14:39:43.207  3756  4360 E Parcel  :      at java.lang.Class.classForName(Native Method)
10-03 14:39:43.207  3756  4360 E Parcel  :      at java.lang.BootClassLoader.findClass(ClassLoader.java:1355)
10-03 14:39:43.207  3756  4360 E Parcel  :      at java.lang.BootClassLoader.loadClass(ClassLoader.java:1415)
10-03 14:39:43.207  3756  4360 E Parcel  :      at java.lang.ClassLoader.loadClass(ClassLoader.java:312)
10-03 14:39:43.207  3756  4360 E Parcel  :      ... 17 more
10-03 14:39:43.207  3756  4360 E Parcel  : Caused by: java.lang.NoClassDefFoundError: Class not found using the boot class loader; no stack
```

```
trace available
10-03 14:39:43.208  3756  4360 W Bundle  : Failed to parse Bundle, but defusing quietly
10-03 14:39:43.208  3756  4360 W Bundle  : android.os.BadParcelableException: ClassNotFoundException when unmarshalling:
it.redacted.android.model.b.c
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.Parcel.readParcelableCreator(Parcel.java:2871)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.Parcel.readParcelable(Parcel.java:2797)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.Parcel.readValue(Parcel.java:2700)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.Parcel.readArrayMapInternal(Parcel.java:3067)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.BaseBundle.unparcel(BaseBundle.java:257)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.BaseBundle.getString(BaseBundle.java:1086)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.content.Intent.getStringExtra(Intent.java:7706)
10-03 14:39:43.208  3756  4360 W Bundle  :      at com.android.server.am.ActivityStarter.startActivity(ActivityStarter.java:468)
10-03 14:39:43.208  3756  4360 W Bundle  :      at com.android.server.am.ActivityStarter.startActivityLocked(ActivityStarter.java:419)
10-03 14:39:43.208  3756  4360 W Bundle  :      at com.android.server.am.ActivityStarter.startActivityMayWait(ActivityStarter.java:1289)
10-03 14:39:43.208  3756  4360 W Bundle  :      at
com.android.server.am.ActivityManagerService.startActivityAsUser(ActivityManagerService.java:6014)
10-03 14:39:43.208  3756  4360 W Bundle  :      at
com.android.server.am.ActivityManagerService.startActivity(ActivityManagerService.java:5867)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.app.IActivityManager$Stub.onTransact(IActivityManager.java:121)
10-03 14:39:43.208  3756  4360 W Bundle  :      at
com.android.server.am.ActivityManagerService.onTransact(ActivityManagerService.java:3844)
10-03 14:39:43.208  3756  4360 W Bundle  :      at android.os.Binder.execTransact(Binder.java:682)
10-03 14:39:43.210  3756  4360 I ActivityManager: START u0 {act=null typ=null flg=0x4000000
cmp=ComponentInfo{it.redacted.android/it.redacted.app.presentation.filters.FiltersActivity}} from uid 10204
```

it.redacted.android.model.b.c is implements Landroid/os/Parcelable; and is passed to FilterActivity, where the above exception is thrown. Can you see if a similar error is thrown on your test device before the actual crash is reported?

Is your it.redacted.android.model.b.c coming from kotlin or something else? Are you doing anything interesting with these parcable's?

ni...@gmail.com <ni...@gmail.com> #52

Oct 3, 2019 11:40PM

No exception is thrown on the test device before the actual crash occurs.

it.redacted.android.model.b.c is a Java class; FiltersActivity is a Kotlin class.
it.redacted.android.model.b.c has got one field which is an Enum and another one is a List<LatLng>. Can it be a problem with this?

In FiltersActivity I am simply reading the parcel from Intent's extras. There are checks for nullability.

sb...@opera.com <sb...@opera.com> #53

Mar 28, 2022 07:41PM

Although this bug hasn't seen much action in quite some time it seems to still be open so posting some info as we've recently seen an unexpected increase in these types of crash reports for Android 6.0. We have not been able to reproduce locally yet but what we've seen so far is Play Console stack traces very similar to the original reports above (also only on Android 6.0 devices):

```
backtrace:
  #00  pc 00000000000ea728  /system/lib/libart.so (art_quick_imt_conflict_trampoline+7)
  #00  pc 0000000002416a83  /system/framework/arm/boot.oat (offset 0x23e7000)
```

Without much else to go on we decided to revert back to an R8 version used in a previous release and at least for now we no longer see the crash reports. So:
- R8 v3.1.66: seeing many crash reports from the wild.
- R8 v3.1.45: no such reports seen yet.

The actual crash (SEGV_MAPERR) is typically followed by a SIGABRT, in the ART runtime as well a bit later:

```
backtrace:
  #00  pc 00000000000418e0  /system/lib/libc.so (tgkill+12)
  #00  pc 000000000003ff29  /system/lib/libc.so (pthread_kill+32)
  #00  pc 000000000001c73f  /system/lib/libc.so (raise+10)
  #00  pc 00000000000198f1  /system/lib/libc.so (__libc_android_abort+34)
  #00  pc 00000000000174b0  /system/lib/libc.so (abort+4)
  #00  pc 0000000000320a59  /system/lib/libart.so (art::Runtime::Abort()+212)
  #00  pc 00000000000f4179  /system/lib/libart.so (art::LogMessage::~LogMessage()+2092)
  #00  pc 00000000000f09a7  /system/lib/libart.so (art::Barrier::~Barrier()+182)
  #00  pc 00000000003472c5  /system/lib/libart.so (art::ThreadList::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char> >&)+144)
  #00  pc 0000000000320b15  /system/lib/libart.so (art::Runtime::Abort()+400)
  #00  pc 00000000000f4179  /system/lib/libart.so (art::LogMessage::~LogMessage()+2092)
  #00  pc 00000000000f09a7  /system/lib/libart.so (art::Barrier::~Barrier()+182)
  #00  pc 00000000003472c5  /system/lib/libart.so (art::ThreadList::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char> >&)+144)
  #00  pc 0000000000348273  /system/lib/libart.so (art::UnsafeLogFatalForThreadSuspendAllTimeout()+210)
  #00  pc 0000000000348b45  /system/lib/libart.so (art::ThreadList::SuspendAll(char const*, bool)+2116)
  #00  pc 00000000001a5caf  /system/lib/libart.so (art::gc::Heap::GetObjectsAllocated() const+322)
  #00  pc 00000000001a9669  /system/lib/libart.so (art::gc::Heap::DumpForSigQuit(std::__1::basic_ostream<char, std::__1::char_traits<char>
>&)+240)
  #00  pc 0000000000320d75  /system/lib/libart.so (art::Runtime::DumpForSigQuit(std::__1::basic_ostream<char, std::__1::char_traits<char>
>&)+40)
  #00  pc 0000000000328379  /system/lib/libart.so (art::SignalCatcher::HandleSigQuit()+896)
  #00  pc 0000000000328be9  /system/lib/libart.so (art::SignalCatcher::Run(void*)+324)
  #00  pc 000000000003f82b  /system/lib/libc.so (__pthread_start(void*)+30)
  #00  pc 0000000000019f75  /system/lib/libc.so (__start_thread+6)
```

We're using app bundles.

sg...@google.com <sg...@google.com> #54 Mar 28, 2022 09:31PM

Simon, as you can see we don't have much to go on for this issue, but looking forward to hearing if using 3.1.45 will result in these types of crashes disappearing. I would be a bit surprised, if that turns out to be then case, but please keep us posted on you statistics.

sb...@opera.com <sb...@opera.com> #55 Mar 29, 2022 05:26PM

Thanks for letting me know you're still monitoring this. We're starting to see some similar reports also in the new version, time will tell if it ramps up to the same levels. At least for now the report count is much lower. Not sure if it is relevant but one log indication we've seen in some reports look like this:

```
=====
14547 14547 I dex2oat : /system/bin/dex2oat --compiler-filter=interpret-only -j3 --dex-file=/data/user/0/com.our.app/code_cache/.       --oat-file=/data/user/0/com.our.app/code_cache/.
14547 14547 I dex2oat : dex2oat took 1.533s (threads: 3)
14338 14477 W art     : Method processed more than once: ...
14568 14568 I dex2oat : /system/bin/dex2oat --compiler-filter=interpret-only -j3 --dex-file=/data/user/0/com.our.app/code_cache/.      --oat-file=/data/user/0/com.our.app/code_cache/.
14568 14568 I dex2oat : dex2oat took 1.026s (threads: 3)
14338 14343 F art     : art/compiler/dex/mir_method_info.cc:104] Check failed: invoke_type == kVirtual (invoke_type=direct, kVirtual=virtual)
...
14338 14343 E art     : Unexpected time out during dump checkpoint.
14338 14343 F art     : art/runtime/barrier.cc:90] Check failed: count_ == 0 (count_=-1, 0=0) Attempted to destroy barrier with non zero count
14338 14343 F art     : art/runtime/runtime.cc:366] Runtime aborting --- recursively, so no thread-specific detail!
14338 14343 F art     : art/runtime/runtime.cc:366]
=====
```

So process 14338 in this example is the actual app process which is doing some 'art' work which triggers a Check inside mir_method_info.cc and eventually terminates the runtime. While this process is running there are also other processes being executed to run dex2oat (in case that matters).

de...@gmail.com <de...@gmail.com> #56 Jun 10, 2022 10:25AM

Same here.

The crash only occurs on Android 8.0 version and once the crash occurs, the user can't get out of it because it occurs repeatedly.
I suffered a lot from this crash and did some research and found something interesting.

1) R8 3.2.53 & gradle version 7.1.3 -> No such crash report yet.
2) R8 3.3.50 & gradle version 7.1.3 -> Can see many crash reports.
3) R8 3.2.53 & gradle version 7.2.1 -> Can see many crash reports.
4) R8 3.3.50 & gradle version 7.2.1 -> Can see many crash reports.

The only workaround I found is to revert both the R8 version and the gradle version to the old version.
The above experimental results are all using app bundles and Kotlin.

+ (Jun 22, 2022 modified)

The Kotlin version also seems to have an effect on error occurrence.

1) R8 3.2.53 & gradle version 7.1.3 & Kotlin 1.6.10 -> No such crash report yet.
2) R8 3.2.53 & gradle version 7.1.3 & Kotlin 1.7.0 -> Can see many crash reports.

Message last modified on Jun 22, 2022 01:55PM

sg...@google.com <sg...@google.com> #57 Aug 4, 2022 06:31PM

@dentksh@gmail.com, do you have a stack trace from the crashes that you can share?