 DESCRIPTION sn...@snapchat.com created issue [#1](#)

The Profiler in Android Studio seems to create one JNI global ref for each class in the target app. When the target app has more than 51200 classes, this overflows the global ref table size and c

A similar issue in the debugger was fixed in <https://android-review.googlesource.com/c/platform/external/oj-libjdwpl/+656777>.


Repro steps:

- 1) start profiler session on a large app
- 2) profiler will work for around 5 seconds, then will crash with the JNI error

Android Studio 4.1.3
Build #AI-201.8743.12.41.7199119, built on March 10, 2021
Runtime version: 1.8.0_242-release-1644-b3-6915495 x86_64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
macOS 10.15.7
GC: ParNew, ConcurrentMarkSweep
Memory: 8166M
Cores: 16
Registry: ide.new.welcome.screen.force=true, external.system.auto.import.disabled=true, caches.indexerThreadsCount=8
Non-Bundled Plugins: IdeaVIM, org.jetbrains.kotlin, com.google.idea.bazel.aswb

Studio Build: AI-201.8743.12.41.7199119
Version of Gradle Plugin: 4.1.0
Version of Gradle: 6.8.1
Version of Java: bundled with android studio
OS: macos

✓ Links (3)


 Links (3)

"A similar issue in the debugger was fixed in <https://android-review.googlesource.com/c/platform/external/oj-libjdwpl/+656777>."

"http://com.snapchat.android.dev/code_cache/libjvmtiagent..."


"...uncovers a different bug. libjvmtiagent_arm.so needs to be built with "-Wl,-exclude-library,libunwind.a" but isn't (see <https://github.com/android/ndk/issues/816#issuecomment-472949097>), so it

COMMENTS



sn...@snapchat.com <sn...@snapchat.com> [#2](#)

The global refs are being created in libjvmtiagent_arm.so.




sn...@snapchat.com <sn...@snapchat.com> [#3](#)

A similar issue in the debugger was fixed in <https://android-review.googlesource.com/c/platform/external/oj-libjdwpl/+656777>.

In case you have symbols available internally, here's the stack trace that adds global refs and causes the crash:

```
#00 pc 000388b4 /apex/com.android.runtime/lib/bionic/libc.so (abort+172) (BuildId: 9036f24afe515a61c5d660b744229f1f)
#01 pc 0040327f /apex/com.android.art/lib/libart.so (art::Runtime::Abort(char const*)+1770) (BuildId: 2462cffbee76f3601ba0aacfa326bc6d)
#02 pc 0000d993 /system/lib/libbase.so (android::base::SetAborter(std::__1::function<void (char const*)>&&)::$_3::__invoke(char const*)+46) (BuildId: cc46a3ceeee28e1299032244e9cfb59b)
#03 pc 0000d2b1 /system/lib/libbase.so (android::base::LogMessage::~~LogMessage()+224) (BuildId: cc46a3ceeee28e1299032244e9cfb59b)
#04 pc 00296e67 /apex/com.android.art/lib/libart.so (art::JavaVMExt::AddGlobalRef(art::Thread*, art::ObjPtr<art::mirror::Object>)+174) (BuildId: 2462cffbee76f3601ba0aacfa326bc6d)
#05 pc 002f92d1 /apex/com.android.art/lib/libart.so (art::JNI<true>::NewGlobalRef(_JNIEnv*, _jobject*)+492) (BuildId: 2462cffbee76f3601ba0aacfa326bc6d)
#06 pc 0028ac07 /apex/com.android.art/lib/libart.so (art::(anonymous namespace)::CheckJNI::NewRef(char const*, _JNIEnv*, _jobject*, art::IndirectRefKind)+614) (BuildId: 2462cffbee76f3601ba0aacfa326bc6d)
#07 pc 00042c7f /data/user/0/com.snapchat.android.dev/code_cache/libjvmtiagent_arm.so
#08 pc 00042917 /data/user/0/com.snapchat.android.dev/code_cache/libjvmtiagent_arm.so
#09 pc 00042809 /data/user/0/com.snapchat.android.dev/code_cache/libjvmtiagent_arm.so
#10 pc 0006adfb /data/user/0/com.snapchat.android.dev/code_cache/libjvmtiagent_arm.so
#11 pc 001c978f /data/user/0/com.snapchat.android.dev/code_cache/libjvmtiagent_arm.so
#12 pc 00080a9f /apex/com.android.runtime/lib/bionic/libc.so (__pthread_start(void*)+40) (BuildId: 9036f24afe515a61c5d660b744229f1f)
#13 pc 00039dc5 /apex/com.android.runtime/lib/bionic/libc.so (__start_thread+30) (BuildId: 9036f24afe515a61c5d660b744229f1f)
```



os...@google.com <os...@google.com>

Assigned to an...@google.com.

ph...@google.com <ph...@google.com> [#4](#)

Thank you for reporting. This bug is supposed to be fixed in the latest version of 4.2. Could you please check if it's still in 4.2? (The latest 4.2 is RC 1 as of this message)

yi...@google.com <yi...@google.com>

Reassigned to sn...@snapchat.com.

sn...@snapchat.com <sn...@snapchat.com> [#5](#)

It's fixed, but uncovers a different bug. libjvmtiagent_arm.so needs to be built with "-Wl,--exclude-library,libunwind.a" but isn't (see <https://github.com/android/ndk/issues/816#issuecomment>)

tr...@squareup.com <tr...@squareup.com> [#6](#)

My team is seeing this issue as well. We've seen it with AS 4.2.2 and Arctic Fox, both with AGP 4.2.2. It only happens on emulators and not on real devices.

Here's the error:

```
JNI ERROR (app bug): weak global reference table overflow (max=51200)weak global reference table dump:
Last 10 entries (of 51200):
51199: 0x132c1e88 java.lang.String "kotlin.coroutine... (33 chars)
51198: 0x132c1e20 java.lang.String "kotlinx.coroutines... (52 chars)
51197: 0x132c1db8 java.lang.String "kotlinx.coroutines... (52 chars)
51196: 0x132c1d50 java.lang.String "kotlinx.coroutines... (49 chars)
51195: 0x132c1ce8 java.lang.String "kotlin.coroutine... (51 chars)
51194: 0x132c1c88 java.lang.String "kotlin.coroutine... (46 chars)
51193: 0x14d62c50 java.lang.String "class com.square... (71 chars)
51192: 0x13a80a20 java.lang.String "d"
51191: 0x14d62b30 java.lang.String "ListScrollPositi... (44 chars)
51190: 0x14d60960 java.lang.String "LibraryItemsPage... (4172 chars)
Summary:
51078 of java.lang.Class (51078 unique instances)
46 of java.lang.String (46 unique instances)
33 of java.lang.DexCache (33 unique instances)
9 of dalvik.system.PathClassLoader (2 unique instances)
9 of java.lang.Thread (9 unique instances)
```

Some devs have been reporting crashes when attaching the debugger to our app. I ran this and it seems to indicate there's 51078 Class references?

```
> git ls-files | grep -E "(.kt$|.java$)" | wc -l
32772
```

We have 32k files, so pretty likely we're near the 52k limit with sealed classes and such, so I wonder if we're literally loading every class in the codebase in the debugger, and it can't handle th

tr...@squareup.com <tr...@squareup.com> [#7](#)

I can only reproduce this on API 28 emulators. On 29 and 30, there is no problem.