



- 


em...@google.com <em...@google.com> [#6](#) Sep 21, 2020 05:27PM ⋮

Reassigned to en...@google.com.

It seems to me that this is not an Android Studio (or debugger) bug, but rather a behavior of Android runtime (ART) where a call to JNI `AttachCurrentThread(JNIEnv* p_env, void* thr_args)` method without a `thr_args` or `thr_args.name` argument resets the thread's name.


Assigning to NDK team to get their advice.
- 

en...@google.com <en...@google.com> Sep 22, 2020 02:15AM

Reassigned to ar...@google.com.
- 

al...@google.com <al...@google.com> [#7](#) Sep 23, 2020 07:58AM ⋮

Reassigned to al...@google.com.

I'm art-sheriff this week so I'll do this.
- 

al...@google.com <al...@google.com> [#8](#) Sep 23, 2020 08:32AM ⋮

I think this should be closed as infeasible or WAI.


We have (AFAICT) always tried to make sure that the pthread and java thread names are the same. We do this by setting the pthread name whenever `Thread.setName()` is called. This is in contrast to the RI where the pthread and java names are entirely separate.

If we want to keep this behavior but also allow custom pthread names to be picked up by java-threads we attach to them we'd need some way to differentiate between pthreads which have never had their names changed (which should get the java assigned `Thread-<number>` name) and ones which have (which should override the java name). AFAICT there is no way to distinguish these situations.


This being the case I think our current behavior is fine. If epic or whoever wants to work around this they should just pass in a `JavaVMAttachArgs` struct with the name filled in appropriately when calling `AttachCurrentThread`.

I'll upload a little test to codify the current behavior.

Any comments?

Message last modified on Sep 23, 2020 08:41AM
- 


al...@google.com <al...@google.com> [#9](#) Sep 23, 2020 08:41AM ⋮

<https://android-review.googlesource.com/c/platform/art/+/-1434753> is the new test I've added, codifying the current behavior as correct.
- 

en...@google.com <en...@google.com> [#10](#) Sep 24, 2020 02:20AM ⋮

the original claim was that the *parent* thread got renamed. that was incorrect, right?


If epic or whoever wants to work around this they should just pass in a `JavaVMAttachArgs` struct with the name filled in appropriately when calling `AttachCurrentThread`.

yeah, there's a workaround that works all the way back to dalvik, and it's not obvious what change we could make in the platform so just adding the test to codify this makes sense to me.
- 

al...@google.com <al...@google.com> [#11](#) Sep 24, 2020 02:26AM ⋮

the original claim was that the parent thread got renamed. that was incorrect, right?

I interpreted it to mean that the thread named 'cpp-parent' got renamed when it called `AttachCurrentThread`.

At the very least that's the only thing that really makes sense from the code and comments they provided.
- 

al...@google.com <al...@google.com> Oct 3, 2020 05:58AM

Status: Won't Fix (Intended Behavior)