

← ↻ ☆ Unable to debug JNI since upgrade to Flamingo

+1 4

Hotlists

Unmark Duplicate


🔔

⋮

Comments (17) Dependencies Duplicates (0) Blocking (0) Resources (4)

Duplicate of 268213434 Bug P3 + Add Hotlist

 STATUS UPDATE No update yet. Edit

 DESCRIPTION kv...@justin.tv created issue #1 Apr 1

DESCRIBE THE ISSUE IN DETAIL:

We have an existing project that is a mix of C++, Java and Kotlin.

Since upgrading to Flamingo - and keeping everything the same - Android Studio fails to attach to native debugger.

I get an error when launching: No symbol directories found.

Our project does have:

```
buildTypes {
    debug {
        debuggable true
        jniDebuggable true
    }
}
```

for both the app and the library that it's using. The C++ code is in the library.

Native debugging worked just fine in Electric Eel and prior versions, but upgrading to Flamingo broke it.

We have not upgraded Gradle yet (using 7.3 wrapper) or Android Gradle Plugin (using 7.0.4).


ATTACH LOG FILES (Select Help > Show Log in Files, or Show Log in Finder on a Mac)

Attached


-----

Android Studio Flamingo | 2022.2.1  
Build #AI-222.4459.24.2221.9862592, built on March 31, 2023  
Runtime version: 17.0.6+0-17.0.6b802.4-9586694 aarch64  
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.  
macOS 13.3.1  
GC: G1 Young Generation, G1 Old Generation  
Memory: 6144M  
Cores: 10  
Metal Rendering is ON  
Registry:  
external.system.auto.import.disabled=true  
ide.text.editor.with.preview.show.floating.toolbar=false  
gradle.version.catalogs.dynamic.support=true

Gradle: 7.3 wrapper  
Android Gradle Plugin: 7.0.4

 idea.log 1.6 MB [View](#) [Download](#)


✓ Mentioned issues (3) ✓ Links (1)

 Mentioned issues (3)

P1 Could not get applicationId for ... Project type: PROJECT\_TYPE\_LIBRARY "I'm not convinced the issue is a dup. [b/262868739](#) is marked as an internal cleanup while this bug is a failure to debug

P1 Error when trying to debug android test of library from AS "[https://issuetracker.google.com/268213434](#)"

P1 Native debugger doesn't load symbols after Flamingo update "[https://issuetracker.google.com/279489616](#)"

 Links (1)

"[https://github.com/kmansoft/native-debug-fail](#)"

Same issue with a different much smaller project that has C++ code

Android Studio is NOT running sandboxed (I mention this since I'm running it on MacOS)

**kv...@justin.tv** <kv...@justin.tv> [#3](#)

Apr 18

Tried reinstalling Android Studio, deleting .idea and .cxx directories - no help.

Same thing with a totally unrelated project under Linux (Ubuntu 23.04 x86\_64).

This one has been fully upgraded to Gradle 8.1 and Android Gradle Plugin 8.0.0

Looks like C++ debugging is just broken in Flamingo. Hope it can be fixed soon.

**jv...@google.com** <jv...@google.com>

A

*Assigned to an...@google.com.*

**km...@gmail.com** <km...@gmail.com> [#4](#)

Apr 18

One more thing that's broken is:

Automatic debug launch target type no longer enables native debugging. I had to edit the target and explicitly set debug type to dual (java + native).

Message last modified on Apr 18, 2023 07:55PM

**aa...@google.com** <aa...@google.com> [#5](#)

Apr 19

From log file:

```
2023-04-17 17:44:39,005 [ 371230] SEVERE - #c.a.t.i.r.GradleApplicationIdProvider - Could not get applicationId for android.broadcast. Project type: PROJFI
java.lang.Throwable
    at com.android.tools.idea.run.GradleApplicationIdProvider.getPackageName(GradleApplicationIdProvider.kt:111)
    at com.android.tools.idea.execution.common.debug.utils.FacetFinder.findFacetForProcess(FacetFinder.kt:48)
    at com.android.tools.ndk.run.editor.NativeAndroidDebugger.getDebugProcessStarterForNewProcess(NativeAndroidDebugger.java:158)
    at com.android.tools.ndk.run.editor.NativeAndroidDebugger.getDebugProcessStarterForNewProcess(NativeAndroidDebugger.java:66)
    at com.android.tools.idea.execution.common.debug.DebugSessionStarter$attachDebuggerToStartedProcess$1.fun(DebugSessionStarter.kt:79)
    at com.android.tools.idea.execution.common.debug.DebugSessionStarter$attachDebuggerToStartedProcess$1.fun(DebugSessionStarter.kt:78)
    at org.jetbrains.concurrency.AsyncPromise$thenAsync$1.apply(AsyncPromise.kt:127)
    at org.jetbrains.concurrency.AsyncPromise$thenAsync$1.apply(AsyncPromise.kt:16)
    at java.base/java.util.concurrent.CompletableFuture$UniCompose.tryFire(Unknown Source)
    at java.base/java.util.concurrent.CompletableFuture.postComplete(Unknown Source)
    at java.base/java.util.concurrent.CompletableFuture.complete(Unknown Source)
    at org.jetbrains.concurrency.AsyncPromise.setResult(AsyncPromise.kt:146)
    at com.android.tools.idea.execution.common.debug.DebugSessionStarter$clientAsyncPromise$1.run(DebugSessionStarter.kt:240)
    at com.intellij.openapi.progress.impl.CoreProgressManager.startTask(CoreProgressManager.java:442)
    at com.intellij.openapi.progress.impl.ProgressManagerImpl.startTask(ProgressManagerImpl.java:114)
    at com.intellij.openapi.progress.impl.CoreProgressManager.lambda$runProcessWithProgressAsynchronously$5(CoreProgressManager.java:493)
    at com.intellij.openapi.progress.impl.ProgressRunner.lambda$submit$3(ProgressRunner.java:252)
    at com.intellij.openapi.progress.impl.CoreProgressManager.lambda$runProcess$2(CoreProgressManager.java:188)
    at com.intellij.openapi.progress.impl.CoreProgressManager.lambda$executeProcessUnderProgress$12(CoreProgressManager.java:608)
    at com.intellij.openapi.progress.impl.CoreProgressManager.registerIndicatorAndRun(CoreProgressManager.java:683)
    at com.intellij.openapi.progress.impl.CoreProgressManager.computeUnderProgress(CoreProgressManager.java:639)
    at com.intellij.openapi.progress.impl.CoreProgressManager.executeProcessUnderProgress(CoreProgressManager.java:607)
    at com.intellij.openapi.progress.impl.ProgressManagerImpl.executeProcessUnderProgress(ProgressManagerImpl.java:60)
    at com.intellij.openapi.progress.impl.CoreProgressManager.runProcess(CoreProgressManager.java:175)
    at com.intellij.openapi.progress.impl.ProgressRunner.lambda$submit$4(ProgressRunner.java:252)
    at java.base/java.util.concurrent.CompletableFuture$AsyncSupply.run(Unknown Source)
    at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
    at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
    at java.base/java.util.concurrent.Executors$PrivilegedThreadFactory$1$1.run(Unknown Source)
    at java.base/java.util.concurrent.Executors$PrivilegedThreadFactory$1$1.run(Unknown Source)
    at java.base/java.security.AccessController.doPrivileged(Unknown Source)
    at java.base/java.util.concurrent.Executors$PrivilegedThreadFactory$1.run(Unknown Source)
    at java.base/java.lang.Thread.run(Unknown Source)
```

**km...@gmail.com** <km...@gmail.com> [#6](#)


Apr 19

I'm attaching another Android Studio log, from my personal project which I'm running on Linux.

Same issue, but this project has been fully upgraded to new Gradle and Android Gradle Plugin.

I believe the log has the same error.

In this one, C++ code is in a module called Resize (or maybe image\_resizer).

 **idea.log**  
870 KB [View](#) [Download](#)

**aa...@google.com** <aa...@google.com> [#7](#)

Apr 19

*Reassigned to ko...@google.com.*

Is this by chance a project you can share?

**ko...@google.com** <ko...@google.com>

A

*Status: Duplicate of [262868739](#)*

**km...@gmail.com** <km...@gmail.com> [#8](#)

Apr 19

Sorry I can't share either project but you should be able to reproduce it easily - since for me it happens with 3 different unrelated projects on two different platforms.

One more thing from the log, seems relevant:

```
2023-04-17 20:00:52,503 [ 57048] INFO - #c.a.t.n.r.l.ShellSessionStarterImpl - Command: cat /data/local/tmp/start_lldb_server.sh | run-as org.kman.email2 sh -c 'cat > /data/data/org.kman.email2/lldb/bin/start_lldb_server.sh && chmod 700 /data/data/org.kman.email2/lldb/bin/start_lldb_server.sh' 2023-04-17 20:00:52,538 [ 57083] INFO - #c.a.t.n.r.l.ShellS Starting LLDB server : /data/data/org.kman.email2/lldb/bin/start_lldb_server.sh /data/data/org.kman.email2/lldb unix-abstract /org.kman.email2-0 platform-1681786851985.sock "lldb proc packets" 2023-04-17 20:00:52,539 [ 57084] INFO - #c.a.t.n.r.l.AndroidLLDBDriver - Connecting to LLDB server: unix-abstract-connect://[emulator-5554]/org.kman.email2-0/platform-16817868 17 20:00:52,542 [ 57087] WARN - #c.a.t.n.r.l.AndroidLLDBDriver - Failed to connect platform (attempt 1 of 10) - retrying. Error was: Connection shut down by remote side while waiting for rep packet
```

**km...@gmail.com** <km...@gmail.com> [#9](#)

Apr 19

Re: Status: Duplicate of 262868739

The other bug is for a "Could not get applicationId" exception, are you sure that's the root cause of inability to debug C++ native code?

It has been there since Android Gradle Tools 7.4 but I had no issues debugging native with Electric Eel.

Message last modified on Apr 19, 2023 04:12AM

**aa...@google.com** <aa...@google.com> [#10](#)

Apr 19

I'm not convinced the issue is a dup. b/262868739 is marked as an internal cleanup while this bug is a failure to debug in native.

Polina, can you double check?

**ko...@google.com** <ko...@google.com> [#11](#)

Apr 19

Yes, see `Repro` instructions

**kv...@justin.tv** <kv...@justin.tv> [#12](#)

Apr 19

I will have some time today to create a reproducer project

**ko...@google.com** <ko...@google.com>

A

*Status: Duplicate of [268213434](#)*

**km...@gmail.com** <km...@gmail.com> [#13](#)

Apr 19

Here is a minimal sample:

<https://github.com/kmansoft/native-debug-fail>

Two issues:

1 - When debug type is set to Automatic (the default), Android Studio Flamingo doesn't even try to start a native debug session

2 - If you manually set debug type to Dual (Java + Native), then you'll get the error saying "No symbol directories found" - AS won't be able to attach to native C++ code and debugging native C work

**va...@adyen.com** <va...@adyen.com> [#14](#)

Apr 21

Adding symbol directories manually does work, though. Breakpoints start working afterwards.

em...@google.com <em...@google.com> [#15](#)

Apr 27

When using Android Studio Flamingo:

Are you trying to debug an `androidTest` instrumented test?

- This is tracked in <https://issuetracker.google.com/268213434>
- It's fixed in Giraffe.
- No plans for Flamingo cherry-pick for now.

Or, are you trying to debug an `app`?

- This is tracked in <https://issuetracker.google.com/279489616>
- We are considering a Flamingo cherry-pick to fix this (Possibly Flamingo Patch 2).

Or, something else?

km...@gmail.com <km...@gmail.com> [#16](#)

Apr 27

Trying to debug the app, please see my sample.

Glad to hear this is being fixed. And thank you for considering a patch release.

km...@gmail.com <km...@gmail.com> [#17](#)

May 5

Issue still present in

```
Android Studio Flamingo | 2022.2.1 Patch 1
Build #AI-222.4459.24.2221.9971841, built on April 19, 2023
Runtime version: 17.0.6+0-17.0.6b802.4-9586694 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.
Linux 6.3.1-x64v3-xanmod1
GC: G1 Young Generation, G1 Old Generation
Memory: 4096M
Cores: 20
Registry:
  external.system.auto.import.disabled=true
  ide.text.editor.with.preview.show.floating.toolbar=false
  gradle.version.catalogs.dynamic.support=true
```

Current Desktop: ubuntu:GNOME