

Comments (5)DependenciesDuplicates (0)Blocking (0)Resources (1)

Fixed


Bug

P3

+ Add Hotlist

 STATUS UPDATE No update yet. 

Edit

 DESCRIPTION kn...@gmail.com created issue [#1](#)

Au

...

#####

Please provide all of the following information, otherwise we may not be able to route your bug report.

#####

1. Describe the bug or issue that you're seeing.  
The IDE starts updating symbols very regularly (after each build/clean), during which the "Make project" and "Clean project" buttons are disabled. The "Run" (and "Debug") button is enabled which can build while the symbols are updating, but after updating a library version in my build.gradle need to clean or I will hit errors.

2. Attach log files from Android Studio  
2A. In the IDE, select the Help..Collect Logs and Diagnostic Data menu option.  
2B. Create a diagnostic report and save it to your local computer.  
2C. Attach the report to this bug using the Add attachments button.

3. If you know what they are, write the steps to reproduce:  
  
3A.  
3B.  
3C.

In addition to logs, please attach a screenshot or recording that illustrates the problem.


For more information on how to get your bug routed quickly, see <https://developer.android.com/studio/report-bugs.html>


Build: AI-231.9225.16.2311.10572941, 202307271245

AI-231.9225.16.2311.10572941, JRE 17.0.7+0-17.0.7b1000.6-10550314x64 JetBrains s.r.o., OS Mac OS X(aarch64) v13.5, screens 1920.0x1200.0, 1920.0x1200.0

AS: Hedgehog | 2023.1.1 Canary 15  
Kotlin plugin: 231-1.8.21-release-380-AS9225.16.2311.10572941  
Android Gradle Plugin: 7.4.2  
Gradle: 7.5  
Gradle JDK: JetBrains Runtime version 17.0.7  
NDK: from module: 23.1.7779620, from module: 23.1.7779620, from module: 23.1.7779620, from local.properties: (not specified), latest from SDK: 25.1.8937393  
CMake: from local.properties: (not specified), latest from SDK: 3.22.1-g37088a8, from PATH: (not found)

...

 **deleted**  
0 B

 **Tue Aug 01 2023 17:17:59 GMT-0700 (Pacific Daylight Time).png**  
102 KB [View](#) [Download](#)

COMMENTS

All comments



**ag...@google.com** <ag...@google.com>

Assigned to an...@google.com.



**an...@google.com** <an...@google.com> [#2](#)

Aug

I see a number of indexing errors related to JNI functions. Example:

```
2023-08-01 16:19:39,452 [4175016] INFO - #c.i.u.i.FileBasedIndexImpl - Rebuild requested for index android.ndk.jni.nativemethodindex
com.intellij.util.indexing.StorageException: com.intellij.util.indexing.StorageException: Error while applying update data for 277248 of android.ndk.jni.r
    at com.intellij.util.indexing.impl.MapReduceIndex.updateWithMap(MapReduceIndex.java:359)
    at com.intellij.util.indexing.storage.MapReduceIndexBase.updateWithMap(MapReduceIndexBase.java:105)
    at com.intellij.util.indexing.impl.MapReduceIndex$IndexUpdateComputable.compute(MapReduceIndex.java:388)
    at com.intellij.util.indexing.impl.MapReduceIndex$IndexUpdateComputable.compute(MapReduceIndex.java:370)
    at com.intellij.openapi.util.Computable.get(Computable.java:16)
    at com.intellij.util.indexing.FileBasedIndexImpl.lambda$runUpdateForPersistentData$27(FileBasedIndexImpl.java:1762)
```

```
at com.intellij.openapi.progress.impl.CoreProgressManager.registerIndicatorAndRun(CoreProgressManager.java:679)
at com.intellij.openapi.progress.impl.CoreProgressManager.computeUnderProgress(CoreProgressManager.java:635)
at com.intellij.openapi.progress.impl.CoreProgressManager.lambda$computeInNonCancelableSection$4(CoreProgressManager.java:230)
at com.intellij.openapi.progress.Cancellation.computeInNonCancelableSection(Cancellation.java:99)
at com.intellij.openapi.progress.impl.CoreProgressManager.computeInNonCancelableSection(CoreProgressManager.java:230)
at com.intellij.util.indexing.FileBasedIndexImpl.lambda$runUpdateForPersistentData$28(FileBasedIndexImpl.java:1762)
at com.intellij.util.indexing.StorageBufferingHandler.runUpdate(StorageBufferingHandler.java:22)
at com.intellij.util.indexing.FileBasedIndexImpl.runUpdateForPersistentData(FileBasedIndexImpl.java:1761)
at com.intellij.util.indexing.SingleIndexValueApplier.doApply(SingleIndexValueApplier.java:65)
at com.intellij.util.indexing.SingleIndexValueApplier.apply(SingleIndexValueApplier.java:53)
at com.intellij.util.indexing.FileIndexesValuesApplier.apply(FileIndexesValuesApplier.java:111)
at com.intellij.util.indexing.contentQueue.IndexUpdateRunner.doWriteIndexesForFile(IndexUpdateRunner.java:365)
at com.intellij.util.indexing.contentQueue.IndexUpdateRunner.writeIndexesForFile(IndexUpdateRunner.java:353)
at com.intellij.util.indexing.contentQueue.IndexUpdateRunner.indexOneFileOfJob(IndexUpdateRunner.java:327)
at com.intellij.util.indexing.contentQueue.IndexUpdateRunner.lambda$indexJobsFairly$3(IndexUpdateRunner.java:243)
at com.intellij.openapi.progress.impl.ProgressSuspender.executeNonSuspendableSection(ProgressSuspender.java:80)
at com.intellij.util.indexing.contentQueue.IndexUpdateRunner.indexJobsFairly(IndexUpdateRunner.java:246)
at com.intellij.util.indexing.contentQueue.IndexUpdateRunner.lambda$doIndexFiles$2(IndexUpdateRunner.java:177)
at com.intellij.util.concurrent.BoundedTaskExecutor.doRun(BoundedTaskExecutor.java:248)
at com.intellij.util.concurrent.BoundedTaskExecutor.access$200(BoundedTaskExecutor.java:28)
at com.intellij.util.concurrent.BoundedTaskExecutor$1.executeFirstTaskAndHelpQueue(BoundedTaskExecutor.java:221)
at com.intellij.util.ConcurrencyUtil.runUnderThreadName(ConcurrencyUtil.java:227)
at com.intellij.util.concurrent.BoundedTaskExecutor$1.run(BoundedTaskExecutor.java:210)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
at java.base/java.util.concurrent.ThreadPoolExecutor$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$1.run(Unknown Source)
at java.base/java.lang.Thread.run(Unknown Source)
Caused by: com.intellij.util.indexing.StorageException: Error while applying update data for 277248 of android.ndk.jni.nativemethodindex
at com.intellij.util.indexing.impl.UpdateData.iterateKeys(UpdateData.java:55)
at com.intellij.util.indexing.impl.MapReduceIndex.updateWithMap(MapReduceIndex.java:351)
... 35 more
Caused by: java.io.IOException: Error reading data for index android.ndk.jni.nativemethodindex
at com.intellij.util.indexing.impl.InputIndexDataExternalizer.read(InputIndexDataExternalizer.java:54)
at com.intellij.util.indexing.impl.InputIndexDataExternalizer.read(InputIndexDataExternalizer.java:28)
at com.intellij.util.indexing.InputMapExternalizer.read(InputMapExternalizer.java:89)
at com.intellij.util.indexing.InputMapExternalizer.read(InputMapExternalizer.java:24)
at com.intellij.util.indexing.impl.forward.AbstractForwardIndexAccessor.deserializeFromByteSeq(AbstractForwardIndexAccessor.java:87)
at com.intellij.util.indexing.impl.forward.AbstractForwardIndexAccessor.deserializeData(AbstractForwardIndexAccessor.java:29)
at com.intellij.util.indexing.impl.forward.AbstractForwardIndexAccessor.getDiffBuilder(AbstractForwardIndexAccessor.java:35)
at com.intellij.util.indexing.impl.MapReduceIndex.getKeysDiffBuilder(MapReduceIndex.java:287)
at com.intellij.util.indexing.impl.storage.TransientFileContentIndex.getKeysDiffBuilder(TransientFileContentIndex.java:77)
at com.intellij.util.indexing.impl.MapReduceIndex.lambda$prepareUpdate$3(MapReduceIndex.java:259)
at com.intellij.util.indexing.impl.UpdateData.iterateKeys(UpdateData.java:52)
... 36 more
Caused by: java.lang.IllegalArgumentException: `classes` must not be empty. Details: packages=[com, microsoft, odsp, crossplatform, core, coreJNI], classe
at com.android.tools.ndk.jni.model.FullyQualifiedClassName.<init>(JvmType.kt:358)
at com.android.tools.ndk.jni.model.FullyQualifiedClassName$Companion.fromString(JvmType.kt:410)
at com.android.tools.ndk.jni.service.SimpleNativeMethodId$Descriptor.read(GlobalNativeMethodIndex.kt:105)
at com.android.tools.ndk.jni.service.SimpleNativeMethodId$Descriptor.read(GlobalNativeMethodIndex.kt:94)
at com.intellij.util.indexing.impl.CollectionDataExternalizer.read(CollectionDataExternalizer.java:57)
at com.intellij.util.indexing.impl.CollectionDataExternalizer.read(CollectionDataExternalizer.java:30)
at com.intellij.util.indexing.impl.InputIndexDataExternalizer.read(InputIndexDataExternalizer.java:51)
... 46 more
```

**em...@google.com** <em...@google.com> [#3](#)

Aug 1

*Reassigned to em...@google.com.*

```
Caused by: java.lang.IllegalArgumentException: `classes` must not be empty. Details: packages=[com, microsoft, odsp, crossplatform, core, coreJNI], classe
at com.android.tools.ndk.jni.model.FullyQualifiedClassName.<init>(JvmType.kt:358)
```

The class name starting with a lowercase letter breaks our assumption here. We can start by fixing this particular exception.

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

Aug 1

I fixed the offending class name, but IDE seemed to be stuck in a bad state. Launching Android Studio Preview still resulted in a seemingly endless symbol update.

I uninstalled Android Studio Preview, manually deleted the cache and support library folders for the Android Studio Preview, reinstalled and the issue is now gone.

So yes, looks like the class name took care of it! Thanks!

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

Aug 1

*Marked as fixed.*

Thank you for verifying that the class name was what caused Android Studio to choke here.

We fixed the class name parser. The fix will be in the next Android Studio canary (Android Studio I Canary 1) and all subsequent I + builds.

Message last modified on Aug 11, 2023 10:05AM