Find Class and Find Usages broken: "AssertionError: Symbols must be loaded"

+1 28 Hotlists (5) Mark as Duplicate

Comments (20) Dependencies Duplicates (7) Blocking (0) Resources (10)

Fixed Bug P1 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION kd...@google.com created issue #1

```
Android Studio Hedgehog | 2023.1.1 Canary 2
Build #AI-231.6890.12.2311.SNAPSHOT, built on April 27, 2023 at 06:50
Runtime version: 17.0.6+0-17.0.6b829.5-9884540 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.
Linux 6.1.20-1rodete1-amd64
GC: G1 Young Generation, G1 Old Generation
Memory: 1280M
Cores: 72
Registry:
    external.system.auto.import.disabled=true
    ide.text.editor.with.preview.show.floating.toolbar=false


Current Desktop: X-Cinnamon
```

Create a new studio project using the basic compose template.

Wait for everything to index and try to Ctrl+click to find usages of the `Greeting` function.

Before you even click, it should fail to have the popup show up and instead spew out:

```
Exception in thread "AWT-EventQueue-0" com.google.common.util.concurrent.ExecutionError: java.lang.AssertionError: Symbols must be loaded
        at com.google.common.cache.LocalCache$Segment.get(LocalCache.java:2053)
        at com.google.common.cache.LocalCache.get(LocalCache.java:3966)
        at com.google.common.cache.LocalCache$LocalManualCache.get(LocalCache.java:4863)
        at com.android.tools.ndk.jni.utils.JniUtilsKt.findAllJniCalls(JniUtils.kt:309)
        at com.android.tools.ndk.jni.utils.JniUtilsKt.findAllJniCalls$default(JniUtils.kt:300)
        at com.android.tools.ndk.jni.service.RegisterNativesProcessor.getAllCallsToRegisterNatives(RegisterNativesProcessor.kt:64)
        at com.android.tools.ndk.jni.service.JniMethodResolver.resolveManuallyRegistered(JniMethodResolver.kt:128)
        at com.android.tools.ndk.jni.service.JniMethodResolver.resolveJavaOrKotlinMethod(JniMethodResolver.kt:113)
        at com.android.tools.ndk.jni.reference.JniGotoDeclarationHandler.getGotoDeclarationTargets(JniGotoDeclarationHandler.kt:38)
        at com.intellij.codeInsight.navigation.impl.GtdProvidersKt.fromGTDProvidersInner(gtdProviders.kt:30)
        at com.intellij.codeInsight.navigation.impl.GtdProvidersKt.access$fromGTDProvidersInner(gtdProviders.kt:1)
        at com.intellij.codeInsight.navigation.impl.GtdProvidersKt$fromGTDProviders$1.invoke(gtdProviders.kt:17)
        at com.intellij.codeInsight.navigation.impl.GtdProvidersKt$fromGTDProviders$1.invoke(gtdProviders.kt:16)
        at com.intellij.codeInsight.navigation.impl.CommonKt.processInjectionThenHost(common.kt:25)
        at com.intellij.codeInsight.navigation.impl.GtdProvidersKt.fromGTDProviders(gtdProviders.kt:16)
        at com.intellij.codeInsight.navigation.actions.GotoDeclarationOrUsageHandler2.gotoDeclarationOrUsages(GotoDeclarationOrUsageHandler2.kt:32)
        at com.intellij.codeInsight.navigation.actions.GotoDeclarationOrUsageHandler2.getCtrlMouseData(GotoDeclarationOrUsageHandler2.kt:45)
        at com.intellij.codeInsight.navigation.actions.GotoDeclarationAction.getCtrlMouseData(GotoDeclarationAction.java:95)
        at com.intellij.codeInsight.navigation.CtrlMouseHandler2$computeInReadAction$1.invoke(CtrlMouseHandler.kt:233)
        at com.intellij.codeInsight.navigation.CtrlMouseHandler2$computeInReadAction$1.invoke(CtrlMouseHandler.kt:232)
        at com.intellij.lang.documentation.ide.impl.DocumentationTargetHoverInfoKt.injectedThenHost(DocumentationTargetHoverInfo.kt:76)
        at com.intellij.codeInsight.navigation.CtrlMouseHandler2.computeInReadAction(CtrlMouseHandler.kt:232)
        at com.intellij.codeInsight.navigation.CtrlMouseHandler2.access$computeInReadAction(CtrlMouseHandler.kt:74)
        at com.intellij.codeInsight.navigation.CtrlMouseHandler2$compute$2$1.invoke(CtrlMouseHandler.kt:220)
        at com.intellij.codeInsight.navigation.CtrlMouseHandler2$compute$2$1.invoke(CtrlMouseHandler.kt:219)
        at com.intellij.openapi.application.rw.InternalReadAction.insideReadAction(InternalReadAction.kt:110)
        at com.intellij.openapi.application.rw.InternalReadAction.access$insideReadAction(InternalReadAction.kt:15)
        at com.intellij.openapi.application.rw.InternalReadAction$tryReadCancellable$1.invoke(InternalReadAction.kt:94)
        at com.intellij.openapi.application.rw.InternalReadAction$tryReadCancellable$1.invoke(InternalReadAction.kt:93)
        at com.intellij.openapi.progress.CancellationKt.withCurrentJob$lambda$0(cancellation.kt:17)
        at com.intellij.openapi.progress.Cancellation.withCurrentJob(Cancellation.java:60)
        at com.intellij.openapi.progress.CancellationKt.withCurrentJob(cancellation.kt:17)
        at com.intellij.openapi.progress.CancellationKt.executeWithJobAndCompleteIt(cancellation.kt:126)
        at com.intellij.openapi.application.rw.CancellableReadActionKt.cancellableReadActionInternal$lambda$1$lambda$0(cancellableReadAction.kt:49)
        at com.intellij.openapi.application.impl.ApplicationImpl.tryRunReadAction(ApplicationImpl.java:1096)
        at com.intellij.openapi.application.rw.CancellableReadActionKt.cancellableReadActionInternal$lambda$1(cancellableReadAction.kt:47)
        at com.intellij.openapi.progress.util.ProgressIndicatorUtilService.runActionAndCancelBeforeWrite(ProgressIndicatorUtilService.java:63)
        at com.intellij.openapi.progress.util.ProgressIndicatorUtils.runActionAndCancelBeforeWrite(ProgressIndicatorUtils.java:129)
        at com.intellij.openapi.application.rw.CancellableReadActionKt.cancellableReadActionInternal(cancellableReadAction.kt:45)
        at com.intellij.openapi.application.rw.InternalReadAction.tryReadCancellable(InternalReadAction.kt:93)
        at com.intellij.openapi.application.rw.InternalReadAction.access$tryReadCancellable(InternalReadAction.kt:15)
        at com.intellij.openapi.application.rw.InternalReadAction$tryReadAction$2.invoke(InternalReadAction.kt:79)
        at com.intellij.openapi.application.rw.InternalReadAction$tryReadAction$2.invoke(InternalReadAction.kt:74)
        at com.intellij.openapi.progress.CancellationKt.withCurrentJob$lambda$0(cancellation.kt:17)
        at com.intellij.openapi.progress.Cancellation.withCurrentJob(Cancellation.java:60)
        at com.intellij.openapi.progress.CancellationKt.withCurrentJob(cancellation.kt:17)
        at com.intellij.openapi.progress.CoroutinesKt.blockingContext(coroutines.kt:193)
        at com.intellij.openapi.application.rw.InternalReadAction.tryReadAction(InternalReadAction.kt:74)
        at com.intellij.openapi.application.rw.InternalReadAction.readLoop(InternalReadAction.kt:66)
        at com.intellij.openapi.application.rw.InternalReadAction.access$readLoop(InternalReadAction.kt:15)
        at com.intellij.openapi.application.rw.InternalReadAction$runReadAction$5.invokeSuspend(InternalReadAction.kt:45)
        at kotlin.coroutines.jvm.internal.BaseContinuationImpl.resumeWith(ContinuationImpl.kt:33)
        at kotlinx.coroutines.DispatchedTask.run(DispatchedTask.kt:106)
        at kotlinx.coroutines.scheduling.CoroutineScheduler.runSafely(CoroutineScheduler.kt:570)
        at kotlinx.coroutines.scheduling.CoroutineScheduler$Worker.executeTask(CoroutineScheduler.kt:750)
        at kotlinx.coroutines.scheduling.CoroutineScheduler$Worker.runWorker(CoroutineScheduler.kt:677)
        at kotlinx.coroutines.scheduling.CoroutineScheduler$Worker.run(CoroutineScheduler.kt:664)
        Suppressed: kotlinx.coroutines.DiagnosticCoroutineContextException: [StandaloneCoroutine{Cancelling}@6d9d4dbe, EDT]
Caused by: java.lang.AssertionError: Symbols must be loaded
        at com.jetbrains.cidr.lang.symbols.symtable.AbstractGlobalProjectSymbolsCache.lambda$new$2(AbstractGlobalProjectSymbolsCache.java:59)
        at com.intellij.psi.impl.PsiCachedValueImpl.doCompute(PsiCachedValueImpl.java:39)
        at com.intellij.util.CachedValueBase.lambda$getValueWithLock$3(CachedValueBase.java:244)
        at com.intellij.util.CachedValueBase.computeData(CachedValueBase.java:42)
        at com.intellij.util.CachedValueBase.lambda$getValueWithLock$4(CachedValueBase.java:244)
        at com.intellij.openapi.util.RecursionManager$1.computePreventingRecursion(RecursionManager.java:112)
        at com.intellij.openapi.util.RecursionGuard.doPreventingRecursion(RecursionGuard.java:42)
        at com.intellij.openapi.util.RecursionManager.doPreventingRecursion(RecursionManager.java:66)
        at com.intellij.util.CachedValueBase.getValueWithLock(CachedValueBase.java:245)
        at com.intellij.psi.impl.PsiCachedValueImpl.getValue(PsiCachedValueImpl.java:28)
        at com.jetbrains.cidr.lang.symbols.symtable.AbstractGlobalProjectSymbolsCache.lambda$getGlobalShortNames$10(AbstractGlobalProjectSymbolsCache.java:13
        at com.intellij.openapi.progress.util.ProgressIndicatorUtils.computeWithLockAndCheckingCanceled(ProgressIndicatorUtils.java:299)
        at com.jetbrains.cidr.lang.symbols.symtable.AbstractGlobalProjectSymbolsCache.getGlobalShortNames(AbstractGlobalProjectSymbolsCache.java:136)
        at com.jetbrains.cidr.lang.symbols.symtable.AbstractGlobalProjectSymbolsCache.processByQualifiedName(AbstractGlobalProjectSymbolsCache.java:163)
        at com.jetbrains.cidr.lang.symbols.symtable.OCGlobalProjectSymbolsCache.processByQualifiedName(OCGlobalProjectSymbolsCache.java:152)
        at com.android.tools.ndk.jni.utils.JniUtilsKt$findAllJniCalls$1.call(JniUtils.kt:311)
        at com.android.tools.ndk.jni.utils.JniUtilsKt$findAllJniCalls$1.call(JniUtils.kt:309)
        at com.google.common.cache.LocalCache$LocalManualCache$1.load(LocalCache.java:4868)
        at com.google.common.cache.LocalCache$LoadingValueReference.loadFuture(LocalCache.java:3533)
        at com.google.common.cache.LocalCache$Segment.loadSync(LocalCache.java:2282)
        at com.google.common.cache.LocalCache$Segment.lockedGetOrLoad(LocalCache.java:2159)
        at com.google.common.cache.LocalCache$Segment.get(LocalCache.java:2049)
        ... 56 more
```

✓ Mentioned issues (4)   ✓ Links (4)

Mentioned issues (4)

-- -- "Seems similar to b/266179384 ."

P1 [Giraffe | 2022.3.1 Canary 2] On opening CPP file, showing infinite loader in editor. " … where we analyzed the previous manifestation of the same error, and documented the fix: b/266174935"

-- -- "https://issuetracker.google.com/280487158"

P1 Short-circuit C/C++ Analyses when there are no C/C++ projects "…he internal cleanup work (about replacing Android Studio fix with a more generic fix) will be on b/288443830 ."

Links (4)

"Fortunately, JetBrains made a recent change to prevent plugin exceptions from breaking Find Usages completely (like what happened here)."
"I also sent JetBrains/intellij-community/pull/2433 to do the same for the Find Class feature (b/279964342)."
"There is one stack not covered that JetBrains will need to handle: https://youtrack.jetbrains.com/issue/CPP-33974/ ."
"… is one piece of work remaining on our side. Once https://youtrack.jetbrains.com/issue/CPP-33974 is fixed and integrated then we can safely remove HotfixForOCInitialTablesBuildingActivity. I'm le

COMMENTS

kd...@google.com <kd...@google.com>

Assigned to an...@google.com.

kd...@google.com <kd...@google.com> #2

Reassigned to em...@google.com.

Seems similar to b/266179384.

em...@google.com <em...@google.com> #3

The issue where we analyzed the previous manifestation of the same error, and documented the fix: b/266174935

Since this started happening again in Hedgehog Canary 2, it's probably another IntelliJ merge related issue.

em...@google.com <em...@google.com> #4

Another relevant (or duplicate, not sure) issue: http://b/280487158

jo...@google.com <jo...@google.com>

Reassigned to jo...@google.com.

jo...@google.com <jo...@google.com> #5

I think we shouldn't even be hitting this code path (JniUtilsKt$findAllJniCalls) when there is no C++ in this project

gh...@google.com <gh...@google.com> #6

Due to the recent CIDR commits `990aeaf448` and `8cf2b7d5d3` (either one is sufficient I believe), `OCInitialTablesBuildingActivity` no longer builds native symbols for projects without
symbols are queried.

As far as I can tell we have two options:

1. Revert `990aeaf448` and `8cf2b7d5d3` so that native symbols are always built.
2. Or, make sure nobody queries for native symbols when `OCLanguageUtilsBase.isCppSupportDisabled(project)` returns true.

gh...@google.com <gh...@google.com> #7

The hotfix for AS Hedgehog Canary 3 was Change Ib17cb4ce9.

The hotfix for Canary 4 onward is Change Ia10f04128.

Remaining work: follow up with JetBrains to fix the root cause in CIDR.

gh...@google.com <gh...@google.com> #8

Fortunately, JetBrains made a recent change to prevent plugin exceptions from breaking Find Usages completely (like what happened here).

I also sent JetBrains/intellij-community/pull/2433 to do the same for the Find Class feature (b/279964342).

cp...@gmail.com <cp...@gmail.com> #9

Android Studio Hedgehog | 2023.1.1 Canary 6, Find Usages still broken :(

gh...@google.com <gh...@google.com> #10

comment#9 It may be a different issue, since I'm unable to reproduce this in Canary 6. Please file a new bug with `idea.log` attached and post the link here. Thank you!

cp...@gmail.com <cp...@gmail.com> #11

Canary 8 Ctrl + N still does not work

idea.log
88 KB View Download

cp...@gmail.com <cp...@gmail.com> #12

Canary 9 is still broken

jo...@google.com <jo...@google.com> #13

Marked as fixed.

This doesn't repro for me in canary 9. Looking at the idea.log from comment#11, the last callstack is from canary 6 not canary 8.

jo...@google.com <jo...@google.com> #14

I tried ctrl-N and ctrl-B. Please open a new bug with idea.log with canary 9 or later if this persists.

gh...@google.com <gh...@google.com> #15

Status: Assigned (reopened)

The idea.log from comment#11 contains logs from multiple versions. There's an exception in there from Canary 8:

```
2023-06-15 02:06:26,831 [  32938] SEVERE - #c.i.o.a.i.ApplicationImpl - Symbols must be loaded
java.lang.AssertionError: Symbols must be loaded
        at com.jetbrains.cidr.lang.symbols.symtable.AbstractGlobalProjectSymbolsCache.lambda$new$2(AbstractGlobalProjectSymbolsCache.java:59)
        ...
        at com.jetbrains.cidr.lang.symbols.symtable.OCGlobalProjectSymbolsCache.getAllSymbolNames(OCGlobalProjectSymbolsCache.java:171)
        at com.jetbrains.cidr.lang.navigation.OCGotoByNameContributor.processNames(OCGotoByNameContributor.java:44)
        ...
2023-06-15 02:06:26,832 [  32939] SEVERE - #c.i.o.a.i.ApplicationImpl - Android Studio Hedgehog | 2023.1.1 Canary 8  Build #AI-231.9011.34.2311.10290408
2023-06-15 02:06:26,832 [  32939] SEVERE - #c.i.o.a.i.ApplicationImpl - JDK: 17.0.6; VM: OpenJDK 64-Bit Server VM; Vendor: JetBrains s.r.o.
2023-06-15 02:06:26,832 [  32939] SEVERE - #c.i.o.a.i.ApplicationImpl - OS: Linux
```

gh...@google.com <gh...@google.com> #16

I can repro the exception when the "Android NDK Support" plugin is disabled (since disabling that plugin also disables `HotfixForOCInitialTablesBuildingActivity`).

jo...@google.com <jo...@google.com> #17

cpt.reese@gmail.com, have you disabled the "Android NDK Support" plugin? If so, you should be able to get rid of the call stack by reenabling it. We're working with JetBrains on a fix so C++ p
Support" isn't going to work since the hotfix for the callstack is hosted in the "Android NDK Support" plugin

jo...@google.com <jo...@google.com> #18

The part of this we can fix in Android Studio is now done (internal change id I3acbb1e4e93c0c1b817c0233ab2795d50a050611).

There is one stack not covered that JetBrains will need to handle: https://youtrack.jetbrains.com/issue/CPP-33974/.

JetBrains also confirmed that disabling the CLion plugin isn't going to work.

There is one piece of work remaining on our side. Once https://youtrack.jetbrains.com/issue/CPP-33974 is fixed and integrated then we can safely remove HotfixForOCInitialTablesBuildingA

cp...@gmail.com <cp...@gmail.com> #19

thanks, enabling Android NDK Support fixes issue, currently canary 12

em...@google.com <em...@google.com> #20

Marked as fixed.

Rest of the internal cleanup work (about replacing Android Studio fix with a more generic fix) will be on b/288443830.

For those who are impacted, please make sure you have the Android NDK plugin enabled. Otherwise the local fix would not be applied.