


Fixed

Bug


P1

+ Add Hotlist

 STATUS UPDATE

No update yet.

Edit

 DESCRIPTION

jo...@google.com created issue [#1](#)

This bug tracks the work to meet with JetBrains and figure out how to disable the C/C++ plugin when the project has no C/C++.

This is related to [b/279964342](#), which exhibits a callstack like:

```
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)
    ...
```


There is a related hot-fix called `HotfixForOCInitialTablesBuildingActivity` which tactically reenabled table building so that the above callstack wouldn't block users. However, we'd like a permanent

Assigning initially to Emre because he's setting up the JetBrains meet. Emre, please assign back to me after we've met with JetBrains and have an idea how to proceed.

In the meantime, it is possible we'll need other tactical fixes. Those should go in to separate bugs with their own repro steps. This bug is about the long term fix.


✓ Mentioned issues (1)

✓ Links (1)

 Mentioned issues (1)


P1

Find Class and Find Usages broken: "AssertionError: Symbols must be loaded" "This is related to [b/279964342](#) , which exhibits a callstack like:"


 Links (1)

"If I remove above (and `OCGotoByNameContributor.java`) from the query, there are no remaining callstacks that users have hit: https://crash.corp.google.com/browse?q=product_name%3D%27Android

COMMENTS

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


Reassigned to em...@google.com.

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

The call to `isCppSupportDisabled` triggers this problem. Symbol tables aren't built, but other code assumes they are built.

```
internal class OCInitialTablesBuildingActivity : RunAfterOCWorkspaceIsLoaded {
    override fun runActivity(project: Project) {
        if (OCLanguageUtilsBase.isCppSupportDisabled(project)) return

        DumbService.getInstance(project).queueTask(object : DumbModeTask() {
            override fun performInDumbMode(indicator: ProgressIndicator) {
                indicatorRunBlockingCancellable(indicator) {
                    for (extension in RequiredForCidrSmartMode.EP_NAME.extensions) {
                        extension.initialize(project)
                    }
                    OCSymbolTablesBuildingActivity.getInstance(project).setInitialized()
                }
                OCSymbolTablesBuildingActivity.getInstance(project).rebuildSymbols()
            }
        })
    }
}
```

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

On the Android Studio side, I identified several entry points that can trigger this assert: `JniGotoDeclarationHandler.kt` `JniReferencesSearch.kt` `KotlinJniMissingFunctionInspection.kt` `JniRelate`

If I remove above (and `OCGotoByNameContributor.java`) from the query, there are no remaining callstacks that users have hit: https://crash.corp.google.com/browse?q=product_name%3D%27Android

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

Reassigned to jo...@google.com.

Summary of JetBrains meeting was that JB will fix these on a case-by-case basis. There's no deadline assigned for this particular issue on the JetBrains side, if we need some fix for this before
Back to jomof@ to lead further discussions on this.

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

On studio-main (Iguana), I get:

```
java.lang.Throwable: Symbol building is not allowed: 1692395663474 .  
Loaded: false  
Dumb: false  
Previous activities:
```

This error happens if building symbols is provoked outside of centralized symbol building activity before it has finished for the first time, or if this activity did not start at all (see `OCSymbolTablesBuildingActivity#rebuildSymbols`, `FileSymbolTablesCache#shouldBuildTables`).

In order to fix this exception, you need to make sure that the client code does check `FileSymbolTablesCache.areSymbolsLoaded()`.

Previously, this was not necessary because it was guaranteed that we finish symbol loading in smart mode.

But after replacing components with services, it is not guaranteed any more.

```
at com.intellij.openapi.diagnostic.Logger.error(Logger.java:370)  
at com.jetbrains.cidr.lang.symbols.symtable.building.OCBuildingActivityExecutionService.assertParsingAndSymbolBuildingAllowed(OCBuildingActivityExecutionService.java:228)  
at com.jetbrains.cidr.lang.symbols.symtable.FileSymbolTable.forFile(FileSymbolTable.java:220)  
at com.jetbrains.cidr.lang.psi.impl.OCFileImpl.getSymbolTable(OCFileImpl.java:252)  
at com.jetbrains.cidr.lang.psi.impl.OCFileImpl.processSymbolsRecursively(OCFileImpl.java:197)  
at com.jetbrains.cidr.lang.psi.impl.OCFileImpl.processSymbolsRecursively(OCFileImpl.java:185)  
at com.android.tools.ndk.jni.cosmetics.JniMangledFunctionNameFoldingBuilder.buildFoldRegions(JniMangledFunctionNameFoldingBuilder.kt:41)  
at com.intellij.lang.folding.LanguageFolding.buildFoldingDescriptorsNoPlaceholderCaching(LanguageFolding.java:92)  
at com.intellij.lang.folding.CompositeFoldingBuilder.buildFoldRegions(CompositeFoldingBuilder.java:41)  
at com.intellij.lang.folding.LanguageFolding.buildFoldingDescriptorsNoPlaceholderCaching(LanguageFolding.java:92)  
at com.intellij.lang.folding.LanguageFolding.buildFoldingDescriptors(LanguageFolding.java:74)  
at com.intellij.codeInsight.folding.impl.FoldingUpdate.getFoldingsFor(FoldingUpdate.java:272)  
at com.intellij.codeInsight.folding.impl.FoldingUpdate.getFoldingsFor(FoldingUpdate.java:240)  
at com.intellij.codeInsight.folding.impl.CodeFoldingManagerImpl.buildInitialFoldings(CodeFoldingManagerImpl.java:138)  
at com.intellij.openapi.fileEditor.impl.text.PsiAwareTextEditorImpl$loadEditorInBackground$2$1$1.invoke(PsiAwareTextEditorImpl.kt:78)  
at com.intellij.openapi.fileEditor.impl.text.PsiAwareTextEditorImpl$loadEditorInBackground$2$1$1.invoke(PsiAwareTextEditorImpl.kt:77)  
at com.intellij.openapi.progress.CoroutinesKt.jobToIndicator$lambda$1(coroutines.kt:386)  
at com.intellij.openapi.progress.ProgressManager.lambda$runProcess$0(ProgressManager.java:71)  
at com.intellij.openapi.progress.impl.CoreProgressManager.lambda$runProcess$2(CoreProgressManager.java:186)  
at com.intellij.openapi.progress.impl.CoreProgressManager.lambda$executeProcessUnderProgress$13(CoreProgressManager.java:604)  
at com.intellij.openapi.progress.impl.CoreProgressManager.registerIndicatorAndRun(CoreProgressManager.java:679)  
at com.intellij.openapi.progress.impl.CoreProgressManager.computeUnderProgress(CoreProgressManager.java:635)  
at com.intellij.openapi.progress.impl.CoreProgressManager.executeProcessUnderProgress(CoreProgressManager.java:603)  
at com.intellij.openapi.progress.impl.ProgressManagerImpl.executeProcessUnderProgress(ProgressManagerImpl.java:61)  
at com.intellij.openapi.progress.impl.CoreProgressManager.runProcess(CoreProgressManager.java:173)  
at com.intellij.openapi.progress.ProgressManager.runProcess(ProgressManager.java:71)  
at com.intellij.openapi.progress.CoroutinesKt.jobToIndicator(coroutines.kt:374)  
at com.intellij.openapi.progress.CoroutinesKt.contextToIndicator(coroutines.kt:347)  
at com.intellij.openapi.progress.CoroutinesKt.blockingContextToIndicator(coroutines.kt:340)  
at com.intellij.openapi.fileEditor.impl.text.PsiAwareTextEditorImpl$loadEditorInBackground$2.invoke(PsiAwareTextEditorImpl.kt:77)  
at com.intellij.openapi.fileEditor.impl.text.PsiAwareTextEditorImpl$loadEditorInBackground$2.invoke(PsiAwareTextEditorImpl.kt:74)  
at com.intellij.openapi.application.rw.InternalReadAction.insideReadAction(InternalReadAction.kt:105)  
at com.intellij.openapi.application.rw.InternalReadAction.access$insideReadAction(InternalReadAction.kt:14)  
at com.intellij.openapi.application.rw.InternalReadAction$tryReadCancellable$2.invoke(InternalReadAction.kt:95)  
at com.intellij.openapi.application.rw.InternalReadAction$tryReadCancellable$2.invoke(InternalReadAction.kt:94)  
at com.intellij.openapi.application.rw.CancellableReadActionKt$cancellableReadActionInternal$1.invoke$lambda$1$lambda$0(cancellableReadAction.kt:39)  
at com.intellij.openapi.application.impl.ApplicationImpl.tryRunReadAction(ApplicationImpl.java:1133)  
at com.intellij.openapi.application.rw.CancellableReadActionKt$cancellableReadActionInternal$1.invoke$lambda$1(cancellableReadAction.kt:39)  
at com.intellij.openapi.progress.util.ProgressIndicatorUtilService.runActionAndCancelBeforeWrite(ProgressIndicatorUtilService.java:63)  
at com.intellij.openapi.progress.util.ProgressIndicatorUtils.runActionAndCancelBeforeWrite(ProgressIndicatorUtils.java:133)  
at com.intellij.openapi.application.rw.CancellableReadActionKt$cancellableReadActionInternal$1.invoke(cancellableReadAction.kt:37)  
at com.intellij.openapi.progress.CoroutinesKt.blockingContext(coroutines.kt:248)  
at com.intellij.openapi.application.rw.CancellableReadActionKt.cancellableReadActionInternal(cancellableReadAction.kt:34)  
at com.intellij.openapi.application.rw.InternalReadAction.tryReadCancellable(InternalReadAction.kt:94)  
at com.intellij.openapi.application.rw.InternalReadAction.tryReadAction(InternalReadAction.kt:76)  
at com.intellij.openapi.application.rw.InternalReadAction.readLoop(InternalReadAction.kt:63)  
at com.intellij.openapi.application.rw.InternalReadAction.access$readLoop(InternalReadAction.kt:14)  
at com.intellij.openapi.application.rw.InternalReadAction$runReadAction$4.invokeSuspend(InternalReadAction.kt:42)  
at com.intellij.openapi.application.rw.InternalReadAction$runReadAction$4.invoke(InternalReadAction.kt)  
at com.intellij.openapi.application.rw.InternalReadAction$runReadAction$4.invoke(InternalReadAction.kt)  
at kotlinx.coroutines.intrinsics.UndispatchedKt.startUndispatchedOrReturn(Undispatched.kt:78)  
at kotlinx.coroutines.BuildersKt__Builders_commonKt.withContext(Builders.common.kt:167)  
at kotlinx.coroutines.BuildersKt.withContext(Unknown Source)  
at com.intellij.openapi.application.rw.InternalReadAction.runReadAction(InternalReadAction.kt:38)
```

```
at com.intellij.openapi.application.rw.PlatformReadWriteActionSupport.executeReadAction(PlatformReadWriteActionSupport.kt:38)
at com.intellij.openapi.application.ReadWriteActionSupport.executeReadAction$default(ReadWriteActionSupport.kt:15)
at com.intellij.openapi.application.CoroutinesKt.constrainedReadAction(coroutines.kt:58)
at com.intellij.openapi.application.CoroutinesKt.readAction(coroutines.kt:25)
at com.intellij.openapi.fileEditor.impl.text.PsiAwareTextEditorImpl.loadEditorInBackground$suspendImpl(PsiAwareTextEditorImpl.kt:74)
at com.intellij.openapi.fileEditor.impl.text.PsiAwareTextEditorImpl.loadEditorInBackground(PsiAwareTextEditorImpl.kt)
at com.intellij.openapi.fileEditor.impl.text.AsyncEditorLoader$start$continuationDeferred$1.invokeSuspend(AsyncEditorLoader.kt:116)
at kotlin.coroutines.jvm.internal.BaseContinuationImpl.resumeWith(ContinuationImpl.kt:33)
at kotlinx.coroutines.DispatchedTask.run(DispatchedTask.kt:106)
at kotlinx.coroutines.scheduling.CoroutineScheduler.runSafely(CoroutineScheduler.kt:584)
at kotlinx.coroutines.scheduling.CoroutineScheduler$Worker.executeTask(CoroutineScheduler.kt:793)
at kotlinx.coroutines.scheduling.CoroutineScheduler$Worker.runWorker(CoroutineScheduler.kt:697)
at kotlinx.coroutines.scheduling.CoroutineScheduler$Worker.run(CoroutineScheduler.kt:684)
```

Adding that one-liner fixes it: `if (!FileSymbolTablesCache.areSymbolsLoaded(root.project)) return emptyArray()`

Message last modified on Aug 19, 2023 07:57AM



em...@google.com <em...@google.com> [#6](#)

Marked as fixed, reassigned to em...@google.com.

If you observe any new cases of "Symbol building is not allowed", please reopen this bug.