IML file format not recognized after upgrade to 4.1.1

+1 2 Hotlists (1) Mark as Duplicate

Comments (15) Dependencies Duplicates (0) Blocking (0) Resources (3)

Can't Repro Bug P3 + Add Hotlist NeedsInfo

STATUS UPDATE No update yet. Edit

DESCRIPTION ja...@gmail.com created issue #1

After upgrading from last latest version to Android Studio 4.1.1 x86_64 for Linux + NDK Side By Side,
I have lost the ability to build my large 15+ year old project with Android Studio , because it now fails
to recognize my IML files or find my project Android.manifest or gradle build variants & flavors.

I am running on a Fedora-32 Linux (5.9) x86_64 laptop.

Now when I try to build my large existing Java + C++ NDK project , which built fine with previous
AS version, the error :

"Error:Android Source Generator: [POC] AndroidManifest.xml file not found"

is raised, with this message in the Log:

```
14:27     Load Settings
            Cannot load settings from file '${MY_PROJECT}${A_MODULE}.iml' does not exist
            Please correct the file content
```

where project & file names are elided. The IML files exist, and are checked in to GIT . I can share the problematic iml files with a human @google.com email address, but I cannot post them here

But how can I see the problems with my IML files ? When I load them into the editor, no problems are displayed .

We are reliant on our Gradle Build Flavors / Configuration Variants being picked up from our IML files -
they are now not read in at all so we cannot build our project with new Android Studio .

It is really not enough to say "Please correct the file content" and refuse to build without saying which
file content needs to be corrected - that in itself is a bug and shows disrespect for your users.

Please help, this is urgent, I am in the middle of an active sprint.

Studio Build: #AI-201.8743.12.41.6953283
Version of Gradle Plugin: 2.4.17
Version of Gradle: 5.8
Version of Java: 1.8
OS: Linux

Steps to Reproduce:
1. Try to build large existing old code with own IML files under version 4.1.1
2. Build fails to parse IML files with no hint as to why.

✓ Links (3)

Links (3)

"where project & file names are elided. The IML files exist, and are checked in to GIT . I can share the problematic iml files with a human @ google.com email address, but I cannot post them here sin
"...so that we can advise you and help troubleshoot the situation. In the mean time you can install older versions of studio to get you back on track. Look for the version that previously worked for you
" ...uture, if you encounter this or any other issue, please read https://developer.android.com/studio/report-bugs.html and file a new bug report with all the required information. This will help ensure t

COMMENTS

uc...@google.com <uc...@google.com>

Assigned to an...@google.com.

ja...@gmail.com <ja...@gmail.com> #2

Thanks, any help or advice on this would be much appreciated.
I have checked in to GIT :
./POCAPP/POCAPP.iml
./POCAPP/app/app.iml
./POCLIB/POCLIB.iml
./POCLIB/poc_core_lib/poc_core_lib.iml

These set up our Build Flavors & Configuration Variants.

POCLIB is a C++ NDK project with JNI Java wrapper classes.

POCAPP is a large VOIP SIP/RDP/UDP OMA Voice Messaging application which uses POCLIB .

I can build neither since installing AS 4.1.1 because those IML files are not parsed, instead are created
many more IML files , which do not build our project at all.

ja...@gmail.com <ja...@gmail.com> #3

Here is what AS 4.1.1 writes to stderr / stdout :

```
2020-12-08 17:37:19,160 [    862]   WARN - Container.ComponentManagerImpl - Do not use constructor injection (requestorClass=com.android.tools.idea.AndroidInitialConfigurator)
2020-12-08 17:37:19,679 [   1381]   WARN - .AbstractProgressIndicatorBase - This progress indicator is indeterminate, this may lead to visual inconsistency. Please call setIndeterminate(fals
com.intellij.openapi.project.impl.ProjectImpl.setProgressDuringInit(ProjectImpl.java:286)
2020-12-08 17:37:19,782 [   1484]   WARN - nSystem.impl.ActionManagerImpl - keymap "ReSharper" not found [Plugin: com.android.tools.ndk]
2020-12-08 17:37:19,851 [   1553]   WARN - nsions.impl.ExtensionPointImpl - Extension to be removed not found: class org.jetbrains.plugins.gradle.execution.test.runner.TestClassGradleCon
2020-12-08 17:37:20,622 [   2324]   WARN - Container.ComponentManagerImpl - Do not use constructor injection (requestorClass=org.jetbrains.android.compose.AndroidComposeAutoDocu
2020-12-08 17:37:20,702 [   2404]   WARN - Container.ComponentManagerImpl - Do not use constructor injection (requestorClass=com.android.tools.idea.apk.ApkProjectComponent)
2020-12-08 17:37:20,702 [   2404]   WARN - Container.ComponentManagerImpl - Do not use constructor injection (requestorClass=com.android.tools.idea.apk.issues.SetupIssueReporter)
2020-12-08 17:37:21,637 [   3339]   WARN - openapi.wm.impl.ToolWindowImpl - ToolWindow icons should be 13x13. Please fix ToolWindow (ID:  Problems View) or icon jar:file:/home/jvd/FW
2020-12-08 17:37:21,880 [   3582]   WARN - Container.ComponentManagerImpl - Do not use constructor injection (requestorClass=com.android.tools.idea.apk.symbols.DebugSymbolNotifica
2020-12-08 17:37:22,045 [   3747]   WARN - ugins.textmate.TextMateService - Missing builtin bundles, checked:
/home/jvd/.local/share/Google/AndroidStudio4.1/textmate/lib/bundles
/home/jvd/FW/android-studio/plugins/textmate/lib/bundles
2020-12-08 17:37:22,753 [   4455]   WARN - ctRoots.impl.UnknownSdkTracker - SDK with name 1.8 already exists: clash=1.8: java version "1.8.0_242" (/home/jvd/FW/android-studio/jre), new
2020-12-08 17:37:43,799 [  25501]   WARN - openapi.wm.impl.ToolWindowImpl - ToolWindow icons should be 13x13. Please fix ToolWindow (ID:  Android Emulator) or icon jar:file:/home/jvd/
windows/emulator.svg
2020-12-08 17:37:44,118 [  25820]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:44,562 [  26264]   WARN -            #com.android.ddmlib - * daemon not running; starting now at tcp:5037
2020-12-08 17:37:44,839 [  26541]   WARN -            #com.android.ddmlib - * daemon started successfully
2020-12-08 17:37:45,306 [  27008]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:45,416 [  27118]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:45,610 [  27312]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:45,710 [  27412]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:45,796 [  27498]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:45,898 [  27600]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:46,007 [  27709]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:37:46,105 [  27807]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:38:00,316 [  42018]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:38:00,475 [  42177]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:38:00,546 [  42248]   WARN - ctivity.DefaultActivityLocator - Cannot locate default activity when indices are not available
2020-12-08 17:38:00,648 [  42350]   WARN - com.intellij.util.xmlb.Binding - no accessors for class org.jetbrains.kotlin.idea.core.script.configuration.utils.ScriptClassRootsStorage
2020-12-08 17:39:10,091 [ 111793]  ERROR - l.BuildProcessClasspathManager - Cannot add 'layoutlib-api.jar' from 'Android 10.4.1.1' to compiler classpath
java.lang.Throwable: Cannot add 'layoutlib-api.jar' from 'Android 10.4.1.1' to compiler classpath
	at com.intellij.openapi.diagnostic.Logger.error(Logger.java:146)
	at com.intellij.compiler.server.impl.BuildProcessClasspathManager.computeCompileServerPluginsClasspath(BuildProcessClasspathManager.java:127)
	at com.intellij.compiler.server.impl.BuildProcessClasspathManager.getStaticClasspath(BuildProcessClasspathManager.java:54)
	at com.intellij.compiler.server.impl.BuildProcessClasspathManager.getBuildProcessPluginsClasspath(BuildProcessClasspathManager.java:38)
	at com.intellij.compiler.server.BuildManager.launchBuildProcess(BuildManager.java:1240)
	at com.intellij.compiler.server.BuildManager.lambda$null$10(BuildManager.java:800)
	at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
	at java.util.concurrent.FutureTask.run(FutureTask.java:266)
	at com.intellij.util.concurrency.BoundedTaskExecutor.doRun(BoundedTaskExecutor.java:222)
	at com.intellij.util.concurrency.BoundedTaskExecutor.access$200(BoundedTaskExecutor.java:29)
	at com.intellij.util.concurrency.BoundedTaskExecutor$1.execute(BoundedTaskExecutor.java:201)
	at com.intellij.util.ConcurrencyUtil.runUnderThreadName(ConcurrencyUtil.java:210)
	at com.intellij.util.concurrency.BoundedTaskExecutor$1.run(BoundedTaskExecutor.java:190)
	at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
	at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
	at java.lang.Thread.run(Thread.java:748)
2020-12-08 17:39:10,093 [ 111795]  ERROR - l.BuildProcessClasspathManager - Android Studio 4.1.1  Build #AI-201.8743.12.41.6953283
2020-12-08 17:39:10,093 [ 111795]  ERROR - l.BuildProcessClasspathManager - JDK: 1.8.0_242-release; VM: OpenJDK 64-Bit Server VM; Vendor: JetBrains s.r.o
2020-12-08 17:39:10,093 [ 111795]  ERROR - l.BuildProcessClasspathManager - OS: Linux
2020-12-08 17:39:10,093 [ 111795]  ERROR - l.BuildProcessClasspathManager - Last Action: MakeModule
```

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

Yuriy, do you have context on how we handle IML differently?

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

Hi, thanks for the report.

Are you using Gradle at all? Normally `iml` file are regenerated during sync and you should not have to bother with them. If you are not using Gradle then please be aware that this is not a pat

We have not fully closed the door on native IDEA projects (using `iml` files) inside Studio, but it's something we have discussed more and more and something we'll probably do soon.

We do need to know what your setup is so that we can advise you and help troubleshoot the situation. In the mean time you can install older versions of studio to get you back on track. Look

ja...@gmail.com <ja...@gmail.com> #6

Yes, the project uses Gradle . I have a POCLIB project with its own build.gradle and IML file & Android.Manifest, which builds a C++ NDK library, and a JNI Java Wrapper library, both of which
build.gradle and IML file . The C++ libraries need to be built for both aarch64 and armv7a, and the Java Libraries & Application need to be build for 16
different Build Flavors, each with Debug and Release Variants - all using Gradle.
It looks AS 4.1.1 does not read my build.gradle (which sets up the build flavors and invokes the C++ builds) or IML files at all, if I try to 'Open Existing' project, if I try to 'Import' the project, it d
it uses Gradle at all, and wrongly creates a Subproject for each Build Flavor, and neither work to allow me to build.
The project builds fine from the command line with Gradle 6.7 (though internally it uses Gradle 5.6.4).
But I have lost the ability to build from Android Studio or use AS debugging facilities .
I will have to look into getting the debugger working outside AS and abandon use of AS for the project
because of this issue.

ja...@gmail.com <ja...@gmail.com> #7

The Project has the following structure when checked out from GIT :

```
    ${git checkout}/.git
                    POCAPP/
                            POCAPP.iml
                            build.gradle
                            settings.gradle
                            app/app.iml
                                build.gradle
                                src/main/{AndroidManifest.xml , assets , java , res , resources }
                                src/{ FLAVORS : one of 16 different flavors }/ {AndroidManifest.xml , assets, java is link to ../main/java }
                     POCLIB/
                             POCLIB.iml
                             build.gradle
                             settings.gradle
                             poc_core_lib/
                                         poc_core_lib.iml
                                         build.gradle
                                         src/main/AndroidManifest.xml
                                                 java # : JNI Wrapper code lives here
                                                 jni   # : C++ code lives here
                                                 res
```

I can execute './gradlew -b build.gradle -c settings.gradle build ' from the command line, which works in the POCLIB directory and POCAPP directory.
The POCAPP directory gradle scripts check that POCLIB has been built, and build it if not.
It used to be that AS would allow us to 'Import' the top-level '.git' containing directory, and a toplevel project was created with POCAPP and POCLIB
sub-projects, and both would get built, just as on the command line .
Now it complains that the IML files have 'invalid contents', not specifying which invalid contents , and does not build our C++ code at all , so the java code can't build.
I am downloading the AS Community Edition GIT repository and will try to build a version of AS that parses our IML & gradle files on Import and post the patch here, eventually, I guess ,
when just being able to build from the command line is not enough.

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

Reassigned to so...@google.com.

I'm guessing that `POCAPP/settings.gradle` refer to the lib via `../POCLib/poc_core_lib`? I wonder if that's the problem.

Yuriy, haven't we encountered something like this?

xa...@google.com <xa...@google.com> #9

Also, maybe try to delete the `iml` files and reimport the project inside Studio to see if generates new one. As I mentioned earlier you should not put these files in git.

ja...@gmail.com <ja...@gmail.com> #10

If AS honored our existing gradle scripts and was able to build the whole project without our custom IML files, we'd love to get rid of them -
unfortunately, it cannot - without them it interprets each one of our dummy build flavor directories, which differ only in AndroidManifest EDITs and
"asset" / resource files from our real "main" Java source directory, as fully-fledged Sub-Projects, which they definitely are not, and on failing
to build any of them fails to even attempt to build our main/* sources .
This would happen on a fresh build after checkout from GIT, with no custom IML files or with them and AS 4.1.1.
If we build first from the command line, and then attempt to 'Open Existing' or Import the project, and uncheck the checkboxes
next to the Build Flavor Android Manifest files, at least the project structure
appears as it did under the previous AS release, but still no build variants are recognized either, and without selection of one of our
build flavors, which we were able to do from the GUI with pre-AS 4.1.1, the project cannot build .
On attempting to "Open Existing Project" after a fresh rebuild from command line of just the POCLIB project, AS appears to ignore the fact
that some huge C++ libraries have been produced, which the POCAPP main/ java code depends on having been produced, so as AS 4.1.1 cannot
build POCLIB , (it does not know it produces a C++ library) it cannot build any of our sources; the whole project and its 32 APKs build fine
from the command line with ' cd POCAPP; ./gradlew -b build.gradle -c settings.gradle build' .
With our custom IML files , up to 4.1.1, at least we could load the project into AS and all source files (including C++ ones) are recognized,
browsable and editable and we can build / clean / debug the project with one click, and debug into our Java and JNI calls code .

Will this ever be possible with AS again ? Else we need to move the project to CLion / Keil / SEGGER Studio / TI Code Composer / CodeWarrior / XCode /
Visual Studio or back to Eclipse, from which it originated . It is about 60/40 Java / C++ now, but that 40% C++ is very important, and also works on
Windows / CE / Mobile or Linux or any POSIX OS . It might be cleaner to move the Java code to IntellIJ Idea and the C++ code to CLion, in the longterm ,
but I need a short-term solution also, which for me now is just makefiles which run gradle & Emacs. It would be nice to at least be able to use AS's debugging facilities on
a project it cannot build, but I don't think this is possible ?

so...@google.com <so...@google.com> #11

Re comment #2:

Just having `.iml` files somewhere does not make them part of the project or even if they are listed in `.idea/modules.xml` they may still not be recognised as modules representing your Gra

Therefore every time you try to open a project which is recognised as an Android project built with Gradle the IDE tries to sync it with the Gradle build configuration and it just comes up with r to represent your Gradle project.

so...@google.com <so...@google.com> #12

`.iml` files are not required to build an Android Gradle project. Can you still build your project from the command line using Gradle?

ja...@gmail.com <ja...@gmail.com> #13

Yes, as I said in Comments #7 & #8, the project builds fine from the command line using gradlew, but fails to import or be opened as an Android Studio project, which it used to do OK .

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

What happens if you:

1. Close Android Studio
2. Backup your project if needed
3. Delete all `.iml` files and `.idea/modules.xml`
4. Re-open the project?

It should take some time for Android Studio to sync the project structure with the Gradle configuration and re-create modules.

`.iml` *modules will likely NOT appear in your project's source tree. Android Studio/IDEA maintains* `.iml` *files for new projects in the system or* `.idea/modules` *directory.*

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

Status: Won't Fix (Not Reproducible)

Our team had requested additional information for this issue which was not provided within 30 days. Unfortunately there is not enough information for us to proceed and this issue is now clo

In the future, if you encounter this or any other issue, please read https://developer.android.com/studio/report-bugs.html and file a new bug report with all the required information. This will h

Thank you!