Can't recognise prefab native library in app native library (no JNI)

+1 Hotlists Mark as Duplicate

Comments (5) Dependencies Duplicates (0) Blocking (0) Resources (3)

Can't Repro Bug P3 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION ps...@gmail.com created issue #1 Aug 23, 2021 04

Hi, I can't seem to get Android Studio prefab native library to be recognised inside another native library app.

When I try to build MyApp I get the following error:

CMake Error: The following variables are used in this project, but they are set to NOTFOUND.
Please set them or make sure they are set and tested correctly in the CMake files:
my-library
linked by target "myapp" in directory /Users/paul/Desktop/TEST/MyApp/app/src/main/cpp

I've searched online and found similar people's problems.

What I'm trying to achieve is: use the native library C++ class inside the app native C++ class. I don't think I need JNI for this because I'm not exposing any of the C++ to managed code in this exa would only use JNI in MyApp later on to expose the C++ to that project's managed code.

I've attached a minimum example, the unzipped folder contains the App folder and a Library folder. I set up prefab in both and the library compiles and I can see the prefab details in the .aar file. this aar as a module dependency ("implementation files" in Gradle of the App). But when I try to use the library C++ class inside the app C++ class it doesn't recognise the class.

One work around which is not very satisfying is to include the AAR (which exposes the headers) and add a nativeBuild to my external C++ project (which enables linking), but that's the opposite o prefab is supposed to do, i.e. the AAR should be enough.

Can you help?

The attached zip of Android Studio projects should work out of the box, except for the main error described above, as all paths are relative.

Thanks
Paul

Build: AI-203.7717.56.2031.7583922, 202107261959,

AI-203.7717.56.2031.7583922, JRE 11.0.10+0-b96-7281165x64 JetBrains s.r.o., OS Mac OS X(x86_64) v11.5.1, screens 2880.0x1800.0; Retina

AS: Arctic Fox | 2020.3.1; Kotlin plugin: 203-1.5.21-release-328-AS7717.8; Android Gradle Plugin: 7.0.0; Gradle: 7.0.2; Gradle JDK: version 11.0.10; NDK: from local.properties: (not specified), lates (not found); LLDB: pinned revision 3.1 not found, latest from SDK: (package not found); CMake: from local.properties: (not specified), latest from SDK: 3.10.2, from PATH: 3.21.1Source: user_sentiment_feedback

IMPORTANT: Please read https://developer.android.com/studio/report-bugs.html carefully and supply all required information.

TEST.zip
29 MB Download

✓ Links (3)

Links (3)

“IMPORTANT: Please read https://developer.android.com/studio/report-bugs.html carefully and supply all required information.”
“ …brary needs to be built with ANDROID_STL=c++_shared or else the the one STL rule will be broken. The error message you get for this is hidden and also terrible. It's something we sho…”
“https://github.com/pshdevio/android-studio-native-aar-s… ”

COMMENTS All comments ↓ C

ra...@google.com <ra...@google.com> Aug 24, 202

Assigned to an...@google.com.

gi...@google.com <gi...@google.com> Sep 9, 202

Reassigned to jo...@google.com.

jo...@google.com <jo...@google.com> #2 Sep 14, 2021 04:3

Status: Won't Fix (Not Reproducible)

The attached project needs a few changes to work correctly.

MyLibrary Side

build.gradle

The library needs to be built with ANDROID_STL=c++_shared or else the the one STL rule will be broken. The error message you get for this is hidden and also terrible. It's something we sh

```
externalNativeBuild {
  cmake {
    arguments '-DANDROID_STL=c++_shared'
    cppFlags '-std=c++14'
  }
}
```

You could also go the other way and change MyLibrary CMakeLists.txt to produce a static libmylibrary.a.

MyApp Side

build.gradle

Also needs ANDROID_STL=c++_shared.

```
externalNativeBuild {
  cmake {
    arguments '-DANDROID_STL=c++_shared'
    ...
  }
}
```

CMakeLists.txt

Should use find_package() instead of find_library().

```
find_package(app REQUIRED CONFIG)
```

Note that it's called 'app' because that's the name of the module in 'MyLibrary'. If you change it to 'lib' or similar then you can have a more intuitive find_package(...) call.

In order to use the package, you have to modify target_link_libraries(...) to reference it.

```
target_link_libraries(
        myapp
        ${log-lib}
        app::mylibrary)
```

I'm uploading a fixed version of the project.

Note

Starting in recent Android Studio Bumblebee canaries, you can have MyApp and MyLibrary in the same project and use a module-to-module dependency:

```
dependencies {
  implementation project(":lib")
}
```

TEST-fixed.zip
1.3 MB Download

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

Sep 14, 2021 06:4

Thank you, that has solve my problem. I had actually tried

```
externalNativeBuild {
  cmake {
    arguments '-DANDROID_STL=c++_shared'
    cppFlags '-std=c++14'
  }
}
```

before but forgot to put that into the projects I sent you (this only seems to work when placed inside `defaultConfig { externalNativeBuild { ... } }`)

The part I was missing was: `find_package(app REQUIRED CONFIG)` and `target_link_libraries(myapp ${log-lib} app::mylibrary)`

Again, I had tried that too before at some point, but it was not clear to me from the online docs that I had to use `app` as you pointed out. I was using `mylibrary` instead. Bugger !

Thanks for sorting this.

PS I adapted my original projects and put a working demo on my github: `https://github.com/pshdevio/android-studio-native-aar-sample`

Message last modified on Sep 14, 2021 07:28PM

ps...@gmail.com <ps...@gmail.com> #4

Sep 14, 2021 07:2

Comment has been deleted.

Message last modified on Sep 14, 2021 07:27PM

ps...@gmail.com <ps...@gmail.com> #5

Sep 14, 2021 09:1

There's a final problem:

If I rename my module library (as you suggested) to e.g. mylibraryname using right click / Refactor on "app" then I get the following error on build (I cleaned etc beforehand):

```
CMake Error at CMakeLists.txt:46 (find_package):
  Could not find a package configuration file provided by "mylibraryname" with any of
  the following names:

    mylibrarynameConfig.cmake
    mylibraryname-config.cmake

  Add the installation prefix of "mylibraryname" to CMAKE_PREFIX_PATH or set "mylibraryname_DIR"
  to a directory containing one of the above files.  If "mylibraryname" provides a
  separate development package or SDK, be sure it has been installed.

ninja: error: rebuilding 'build.ninja': subcommand failed
```

I was able to fix this by renaming as `app` and cleaning projects and invalidation caches etc.

Message last modified on Sep 14, 2021 09:47PM