CMake Transitive Dependency Issue

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

```
Build: AI-221.6008.13.2211.9619390, 202302170051,

AI-221.6008.13.2211.9619390, JRE 11.0.15+0-b2043.56-9505619x64 JetBrains s.r.o., OS Windows 11(amd64) v10.0 , screens 2400.0x1350.0

AS: Electric Eel | 2022.1.1 Patch 2
Kotlin plugin: 221-1.8.0-release-for-android-studio-AS5591.52
Android Gradle Plugin: 7.4.2
Gradle: 7.6.1
Gradle JDK: version 17.0.6
NDK: from module: 25.2.9519653, from local.properties: (not specified), latest from SDK: (not found)
CMake: from local.properties: (not specified), latest from SDK: 3.22.1-g37088a8-dirty, from PATH: (not found)


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

Hello, I'm a maintainer of Android port of LOVE game framework. Sorry if the category is appropriate as I think this issue is specifically inbetween AGP and CMake interaction.

In the upcoming major version (12.0), I switched the build system from legacy ndk-build to CMake. The switch is almost flawless because we already used CMake to compile for Windows. Howe

The project repository is located here: https://github.com/love2d/love-android/tree/12.x

In there there's "love_android" CMake (dummy) target which depends on these:

- love
- OpenAL (added to try to workaround this issue)
- ... additional user-supplied 3rd-party dependencies in `lua-modules` folder ...

The "love" CMake target has these dependencies, all of them are bundled in our source tree:

- zlib (static; not from NDK public API)
- libogg (static)
- libvorbis (static)
- libtheora (static)
- freetype (static)
- OpenAL (shared)
- SDL2 (shared)
- LuaJIT (shared, prebuilt)

Then "OpenAL" CMake target depends on these:

- OpenSLES (from NDK public API)
- Oboe (from AAR)

Before replicating the issue, edit the `app/build.gradle` and change the `cmake.targets` entry to only `"love_android"`, as I pushed change to workaround this issue there.

Here's the issue. When the project successfully compiles (assume setting `cmake.targets` to `"love_android"` alone), looking inside the APK, Oboe shared library and LuaJIT shared library is **nc** this probably indicate that it's related to transitive imported CMake target.

This is reproducible across local builds (Windows) and GitHub CI builds (Linux), as can be seen here https://github.com/love2d/love-android/actions/runs/4272864616 (logging in to GitHub is re

This has been an issue for me for at least few months ago, but have no confidence on filling the bug report until now.

✓ Links (4)

Links (4)

"The project repository is located here: https://github.com/love2d/love-android/tree/12.x"
"This is reproducible across local builds (Windows) and GitHub CI builds (Linux), as can be seen here https://github.com/love2d/love-android/actions/runs/4272864616 (logging in to GitHub is requir
" ... say that the only thing add_dependencies does is enforce build ordering: https://cmake.org/cmake/help/latest/command/add_dependencies.html . CMake needs to know that liboboe.so is a runti
"Possibly https://cmake.org/cmake/help/latest/command/install.html#runtime-dependency-set , and possibly just plain ol' target_link_libraries is the correct way to do that?"

COMMENTS

cm...@google.com <cm...@google.com>

Assigned to an...@google.com.

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

Can you please provide more specific instructions to reproduce this issue?

I tried these:

1. git clone https://github.com/love2d/love-android/tree/12.x
2. Use Android Studio Electric Eel Patch 2 to open the project `love-android`
3. ???

I see that:

1. `app/build.gradle` does not have any native build.
2. `love/build.gradle` has native build, but it's `externalNativeBuild { ndkBuild { path 'src/jni/Android.mk' } }`

I wanted to convert it from ndkBuild to CMake, but I don't see a top-level CMakeLists to use. Is 12.x the correct branch / did you submit your CMake changes?

---

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

Sorry for the missing detailed clone instructions.

1. Clone the repository with `git clone --recurse-submodules -b 12.x https://github.com/love2d/love-android`, with `--depth 1` if needed.
2. Perform the necessary edit I mentioned in the first comment.
3. Use either Android Studio Electric Eel Patch 2 or simply run `./gradlew assembleEmbedNoRecordDebug` to build the project.
4. Inspect the resulting APK doesn't have `liboboe.so`, which is required by `libopenal.so`, and `libluajit.so` which is required by `liblove.so`.

Message last modified on Mar 22, 2023 02:29PM

---

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

Thank you for the detailed instructions. I can reproduce this issue.

I created a minimal project that reproduces the problem:

1. `toplevel` custom cmake target: `add_custom_target(toplevel)`
2. `toplevel` dependencies: `add_dependencies(toplevel lib1 lib2)`
3. `lib2` depends on Oboe prefab.

Building target `toplevel` does not put `liboboe.so` into the APK. Adding `lib2` to the targets list puts `liboboe.so` into the APK.

Jomo/Dan do you have any pointers to where the issue might be?

TransitiveDependency.zip
1.1 MB Download

---

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

The docs say that the only thing `add_dependencies` does is enforce build ordering: https://cmake.org/cmake/help/latest/command/add_dependencies.html. CMake needs to know that libo

---

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

Possibly https://cmake.org/cmake/help/latest/command/install.html#runtime-dependency-set, and possibly just plain ol' `target_link_libraries` is the correct way to do that?

---

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

If I add `target_link_libraries(toplevel lib1 lib2)`, I get:

```
Utility target "toplevel" must not be used as the target of a
 target_link_libraries call.
```

---

gi...@google.com <gi...@google.com>

Reassigned to da...@google.com.

---

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

Reassigned to an...@google.com.

Not something I have time to dive deeper on. My guess was already disproven. Doing any more probably requires someone digging through what the cmake API is telling us about those targe

---

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

Status: New

At this point, we can say that this is a new "known issue", i.e., transitive dependencies via `add_dependency()` are not added into the APK.

Considering the simple workaround (add transitive dependencies to `targets` in `build.gradle`), fixing it does not look like a high priority item at this moment. We'll keep it alive to see if we