

Comments (4)DependenciesDuplications (0)Blocking (0)Resources (5)

Fixed

Bug

P2

+ Add Hotlist

STATUS UPDATE

No update yet.

Edit

DESCRIPTION

ni...@motorolasolutions.com created issue [#1](#)

Feb 9, 2021 01:19AM

Build: AI-201.8743.12.41.7042882, 202012200039,

AI-201.8743.12.41.7042882, JRE 1.8.0_242-release-1644-b01x64 JetBrains s.r.o, OS Windows 10(amd64) v10.0 , screens 1366x768

AS: 4.1.2; Kotlin plugin: 1.3.72-release-Studio4.1-5; Android Gradle Plugin: 4.1.2; Gradle: 6.8.2; NDK: from local.properties: (not specified), latest from SDK: 22.0.7026061; 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: (not found)

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

Trying ndk-build C code debug in Android studio(v4.1.2).Below is build.gradle setup.

jni , C/C++ Source files Dirs. Actual C files are not under jni folder but outside it but referred in [android.mk](#) file.

sourceSets.main.jniLibs.srcDirs = ['D:/ccodefolder/jni/']

2)Android.mk Builds a shared library from C , c++ code and also links inbuilt shared and static libraries.

```
externalNativeBuild {
    ndkBuild {
        path file('D:/ccodefolder/jni/Android.mk')
    }
}
ndkVersion '21.1.6352462'
```

3)debug {
 debuggable true
 jniDebuggable true
 minifyEnabled false
 shrinkResources false
 //ndk.debugSymbolLevel = 'FULL'
}

Able to run the project and shared library get generated along with other prebuilt .SOs and apk works, also CPP folder is created and able to see C code files of my project.

DEBUGGING ISSUE:

LLDB server gets started and Debugger attached to process, But debug any C file is failing with below error.

"Breakpoint will not currently be hit. No executable code is associated with this line"

Link(<https://stackoverflow.com/questions/66069240/android-ndk-build-c-c-debug-in-android-studio>) has more details.

Note that google sample ndk-build project debug work on the same machine.

Please help to understand where is the issue and let us know what other details are required?

Thanks

✓ Links (4)

Hide all

↔ Links (4)

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

ni...@ [#1](#)

"[http://android.mk](#)"

ni...@ [#1](#), ni...@ [#3](#)

"Link(<https://stackoverflow.com/questions/66069240/android-ndk-build-c-c-debug-in-android-studio>) has more details."

ni...@ [#1](#)

"...need to make a small addition to your wrap.sh file. See the following section: https://developer.android.com/ndk/guides/wrap-script#debugging_when_using_wrapsh"

em...@ [#4](#)

COMMENTS

All comments

↓ Oldest first

la...@google.com <la...@google.com>

Feb 10, 2021 04:07PM

ar...@google.com <ar...@google.com>

Feb 11, 2021 04:21AM

Reassigned to em...@google.com.

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

Feb 11, 2021 10:41AM ⋮

Which .so file are you trying to debug?

- For the .so file(s) built by ndk-build:
 - Android Studio can automatically identify the location of the debugging symbol files.
 - Breakpoints should work without any additional effort.
- For the .so file(s) that are included as prebuilt libraries using jniLibs.srcDirs:
 - Android Studio does not know where to find the debugging symbol files.
 - Breakpoints set on C++ files that were compiled into these .so libraries will show up as Breakpoint will not be hit. No executable code is associated with this line warnings, and won't hit.
 - You need to tell Android Studio where to find the symbol files for these .so files.
 - You can achieve this by adding the directory of your symbol files to Edit Configurations... | Debugger | Symbol Directories.

Can you please try the above suggestion?

ni...@motorolasolutions.com <ni...@motorolasolutions.com> #3

Feb 19, 2021 09:15AM ⋮

Thanks for suggestion it really helped!

Real problem for me was in my [android.mk](#) file having "LOCAL_LDFLAGS --strip-all" which didn't generate symbols for debugging. Once it was removed debugging is working.

Please help to understand below items.

1)How to disable c assert crash when (APP_OPTIM:=debug)debugging mode?

2)How to enable address sanitizer such that when in a project c/cpp file code does memory corruption, during debugging android studio should catch it and show the backtrace of files which caused it.?

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

Mar 10, 2021 11:28AM ⋮

Marked as fixed.

1)How to disable c assert crash when (APP_OPTIM:=debug)debugging mode?

You can add the -DNDEBUG flag to achieve this.

- For CMake:

```
add_definitions(-DNDEBUG)
```

- For ndk-build:

```
LOCAL_CFLAGS += -DNDEBUG
```

Please be warned that this will also disable code blocks within #ifdef DEBUG ... #endif.

2)How to enable address sanitizer such that when in a project c/cpp file code does memory corruption, during debugging android studio should catch it and show the backtrace of files which caused it.?

You will need to make a small addition to your wrap.sh file. See the following section: https://developer.android.com/ndk/guides/wrap-script#debugging_when_using_wrapsh