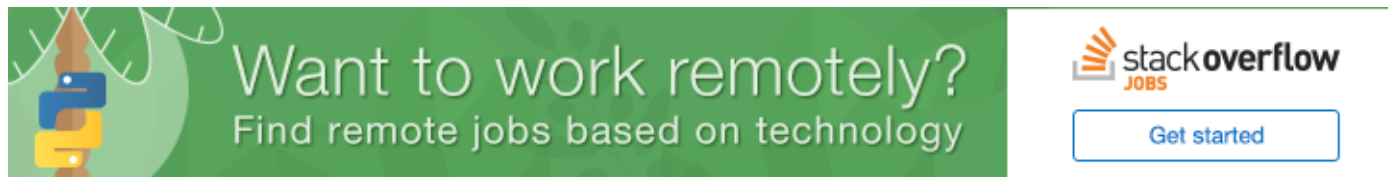


Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

The results are in! [See the 2018 Developer Survey results.](#) »



Force CMake in verbose mode via Gradle and the Android NDK



I'm using Gradle and CMake to compile an Android NDK project from the command line. Previously, I was using Ant and `ndk-build` but I'm trying to migrate the project completely to Gradle and CMake.

In my `build.gradle` I have the following lines to invoke CMake:

```
externalNativeBuild {
    cmake {
        path "src/main/cpp/CMakeLists.txt"
    }
}
```

Now how can I force CMake to print all compiler calls to the console before it makes them? Specifically, I want to see just how CMake runs the compiler and linker.

I've already tried the following, all to no avail:

1) In my `CMakeLists.txt` I have put the following line:

```
set(CMAKE_VERBOSE_MAKEFILE on)
```

Didn't have any effect.

2) I have started the build like this:

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

Google

Facebook



Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

3) And like this:

```
./gradlew build --debug
```

Gradle printed lots of stuff, but no compiler calls.

So none of those three attempts did what I wanted which makes me wonder how can I see how CMake runs clang on my individual source files?

gradle  android-ndk cmake

asked Apr 16 '17 at 16:43



Andreas

1,633 15 31

1 Check `.externalNativeBuild` directory under project root; it may contain all information relevant for you – [Alex Cohn](#) Apr 16 '17 at 18:49

1 Indeed, great, thanks! There's a `build.ninja` log file which contains all the essential information. If you want to post your comment as an answer, I'll accept it. – [Andreas](#) Apr 16 '17 at 18:57

See also some update: I finally reached my desktop to test the flags manually. – [Alex Cohn](#) Apr 19 '17 at 10:47

1 Answer

In Android Studio, **gradle** creates directory `.externalNativeBuild` under the module root, for each module that has NDK integration, via CMake or ndk-build.

For CMake, the gradle plugin is quite verbose. For each build variant it creates separate subdirectory, e.g. `.externalNativeBuild/cmake/debug/x86` OR `.externalNativeBuild/cmake/release/armeabi` , etc.

Each directory contains some useful files: **cmake_build_command.txt** describes the actual parameters passed to CMake; **android_gradle_build.json** shows what parameters the gradle plugin derived for your binaries; from **build.ninja** you can deduce the how these parameters were applied for each compilation or linkage step.

For `ndk-build` , the `android_gradle_build.json` file is also quite useful. `ndkBuild_build_command.txt` lists all parameters passed to **ndk-build** command, and `ndkBuild_build_output.txt` is the unabridged

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
externalNativeBuild {
    ndkBuild {
        cppFlags "-std=c++11"
        arguments "APP_STL=gnustl_static", "NDK_TOOLCHAIN_VERSION=4.9",
        "APP_OPTIM=release", "NDK_DEBUG=0", "V=1"
        abiFilters "armeabi"
    }
}
```

For `cmake`, the relevant argument is `"-DCMAKE_VERBOSE_MAKEFILE:BOOL=ON"` (see [explanation and alternatives](#)):

```
externalNativeBuild {
    cmake {
        cppFlags "-std=c++11"
        arguments "-DCMAKE_VERBOSE_MAKEFILE:BOOL=ON"
        abiFilters "armeabi"
    }
}
```

Without `CMAKE_VERBOSE_MAKEFILE`, the Gradle Console displays:

```
:app:externalNativeBuildDebug
Build native-lib armeabi
[1/2] Building CXX object CMakeFiles/native-lib.dir/src/main/cpp/native-lib.cpp.o
[2/2] Linking CXX shared library
../../../../build/intermediates/cmake/debug/obj/armeabi/libnative-lib.so
```

With `"-DCMAKE_VERBOSE_MAKEFILE:BOOL=ON"`, I get tons of output:

```
:app:externalNativeBuildDebug
Build native-lib armeabi

[1/2] /Users/alex/Library/Android/sdk/ndk-bundle/toolchains/llvm/prebuilt/darwin-x86_64/bin/clang++ -target=armv5te-none-linux-androideabi --gcc-toolchain=/Users/alex/Library/Android/sdk/ndk-bundle/toolchains/arm-linux-androideabi-4.9/prebuilt/darwin-x86_64 --sysroot=/Users/alex/Library/Android/sdk/ndk-bundle/platforms/android-14/arch-arm -Dnative_lib_EXPORTS -isystem /Users/alex/Library/Android/sdk/ndk-bundle/sources/cxx-stl/gnu-libstdc++/4.9/include -isystem /Users/alex/Library/Android/sdk/ndk-bundle/sources/cxx-stl/gnu-libstdc++/4.9/libs/armeabi/include -isystem /Users/alex/Library/Android/sdk/ndk-bundle/sources/cxx-stl/gnu-libstdc++/4.9/include/backward -g -DANDROID -ffunction-sections -funwind-tables -fstack-protector-strong -no-canonical-
```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
sections -funwind-tables -fstack-protector-strong -no-canonical-prefixes -
march=armv5te -mtune=xscale -msoft-float -fno-integrated-as -mthumb -Wa,--
noexecstack -Wformat -Werror=format-security -O0 -fno-limit-debug-info -O0 -
fno-limit-debug-info -fPIC -MD -MT CMakeFiles/native-
lib.dir/src/main/cpp/native-lib.cpp.o -MF CMakeFiles/native-
lib.dir/src/main/cpp/native-lib.cpp.o.d -o CMakeFiles/native-
lib.dir/src/main/cpp/native-lib.cpp.o -c
/Users/alex/test/egl/app/src/main/cpp/native-lib.cpp
[2/2] : && /Users/alex/Library/Android/sdk/ndk-
bundle/toolchains/llvm/prebuilt/darwin-x86_64/bin/clang++ --target=armv5te-none-
linux-androideabi --gcc-toolchain=/Users/alex/Library/Android/sdk/ndk-
bundle/toolchains/arm-linux-androideabi-4.9/prebuilt/darwin-x86_64 --
sysroot=/Users/alex/Library/Android/sdk/ndk-bundle/platforms/android-14/arch-arm
-fPIC -g -DANDROID -ffunction-sections -funwind-tables -fstack-protector-strong -
no-canonical-prefixes -march=armv5te -mtune=xscale -msoft-float -fno-integrated-
as -mthumb -Wa,--noexecstack -Wformat -Werror=format-security -g -DANDROID -
ffunction-sections -funwind-tables -fstack-protector-strong -no-canonical-
prefixes -march=armv5te -mtune=xscale -msoft-float -fno-integrated-as -mthumb -
Wa,--noexecstack -Wformat -Werror=format-security -O0 -fno-limit-debug-info -O0
-fno-limit-debug-info -Wl,--build-id -Wl,--warn-shared-textrel -Wl,--fatal-
warnings -Wl,--no-undefined -Wl,-z,noexecstack -Qunused-arguments -Wl,-z,relro -
Wl,-z,now -Wl,--build-id -Wl,--warn-shared-textrel -Wl,--fatal-warnings -Wl,--no-
undefined -Wl,-z,noexecstack -Qunused-arguments -Wl,-z,relro -Wl,-z,now -shared -
Wl,-soname,libnative-lib.so -o
../../../../build/intermediates/cmake/debug/obj/armeabi/libnative-lib.so
CMakeFiles/native-lib.dir/src/main/cpp/native-lib.cpp.o -llog -lEGL -lGLESv2 -lm
"/Users/alex/Library/Android/sdk/ndk-bundle/sources/cxx-stl/gnu-
libstdc++/4.9/libs/armeabi/libgnustl_static.a" "-latomic" && :
```

The obvious file `.externalNativeBuild/cmake/debug/armeabi/cmake_build_output.txt` does not contain interesting information (unless you have problems with CMake configuration *per se*). Gradle does not write this output to file, but you can use command line redirection, see [Where are Gradle logs?](#).

edited May 23 '17 at 11:47



Community ♦

1 1

answered Apr 16 '17 at 21:14



Alex Cohn

33.9k 5 47 149

gives me an error: Could not find method arguments() for arguments on object of type com.android.build.gradle.internal.dsl.CmakeOptions – [peetonn](#) Jul 6 '17 at 19:50

- 1 @peetonn: your mistake is that you put it in the **externalNativeBuild** block with **path** to CMakeLists.txt; but **arguments** should go into the other block, which is inside **android.defaultConfig** – [Alex Cohn](#) Jul 6 '17 at 21:09

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

entities in android.build.gradle DSL – [Alex Cohn](#) Jul 6 '17 at 21:20

- 1 Note: "-DCMAKE_VERBOSE_MAKEFILE:BOOL=ON" does not seem to work anymore. "-DCMAKE_VERBOSE_MAKEFILE=ON" worked for me. Not sure what has changed, I just installed android studio. – [Fake Name](#) Oct 11 '17 at 21:21
-

@FakeName: cool, for me both work – [Alex Cohn](#) Oct 12 '17 at 12:20
