Before all the step, please make sure all the ndk related command is from the same NDK directory and the NDK folder is not stripped.

I just want to keep my way to enable the Eclipse to build the JNI

1. Add some flag in Application.mk (in jni folder)

                APP\_OPTIM := debug  
               APP\_CFLAG := -g -ggdb -O0

               APP\_PLATFORM := android-15

2. Modify the AndroidManifest.xml add [android:debuggable](http://androiddebuggable/)="true" to Application element (NOTE: This step isn't recommended if you intend to use Eclipse to debug the application,

as it categorizes this attribute as an error. You can skip this attribute in that case.)

3. Set the NDK Path in the environment variable.

4. Now you are able to run the "which ndk-build" command in the command window, it will shows the path to the file

5. Run command: ndk-build in the android application folder (which is the same level as AndroidManifest.xml)

             You will see some line looks like:

[armeabi-v7a] Gdbserver      : [arm-linux-androideabi-4.6] libs/armeabi-v7a/gdbserver

[armeabi-v7a] Gdbsetup       : libs/armeabi-v7a/gdb.setup

[x86] Gdbserver      : [x86-4.6] libs/x86/gdbserver

[x86] Gdbsetup       : libs/x86/gdb.setup

6. Run build\_debug.sh to generate the application (if you want to use eclipse to debug the application, please goes to **6b** at below)

7. Install the apk file by command: adb install bin/\*\*\*.apk

8. Run "ndk-gdb --start --nowait bin/" to start the application (assuming you were still in the top-level directory)

**6b**: Debug with eclipse

7. Set the NDK path in preference>Android>NDK

8. Right click on the Application in the eclipse Package Explorer, click Android Tools > Add Native Support (this will convert you application with C++ enabled project)

9. Click debug configurations (click the drop down icon on the toolbar – near the little bug)

10. Click on the Android Native Application, click the add icon on the toolbar, it will create a new debug configuration for you.

11. Name the configuration as C++, choose you project , now you are ready to go (click the debug button ... you will succeed, if you are lucky)

12. When you continue to this line, that means you are not the lucky guy, you need to do more configurations, let's do it.

First, did you get an error message said "Run-as \*\*\* failed", in that case we can trick the default configuration

Bring back the debug configuration page you just created, click the "Debugger" tab set the debugger value looks something like this

/Users/kevinli/Documents/Software/android/32bit/android-ndk-r9d/toolchains/arm-linux-androideabi-4.8/prebuilt/darwin-x86/bin/arm-linux-androideabi-gdb

Red part is your NDK-path, you need try to find arm\_linux\_androideabi-gdb in some similar folder structure

For shared libraries, you need to point it to the obj/local/\*\*\* in you android src folder

\*\*\* means the CPU architecture of your devices