CREDENCEID MOBILE

INTRODUCTION

In this POC we have implemented Java Native Interface and add some native functionalities to the existing project. And then called the native methods in our android studio project. We have created a simple display() and tried implementing in our android studio project.

Index

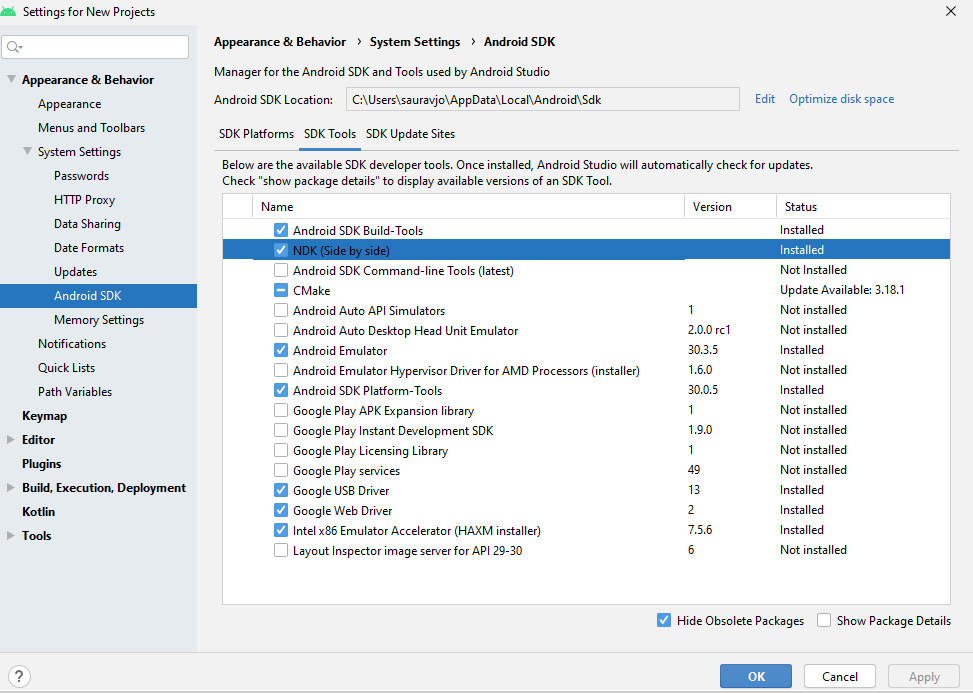
* Setting up the environment.
* Defining native method.
* Adding native method to your project.
* Calling native method.

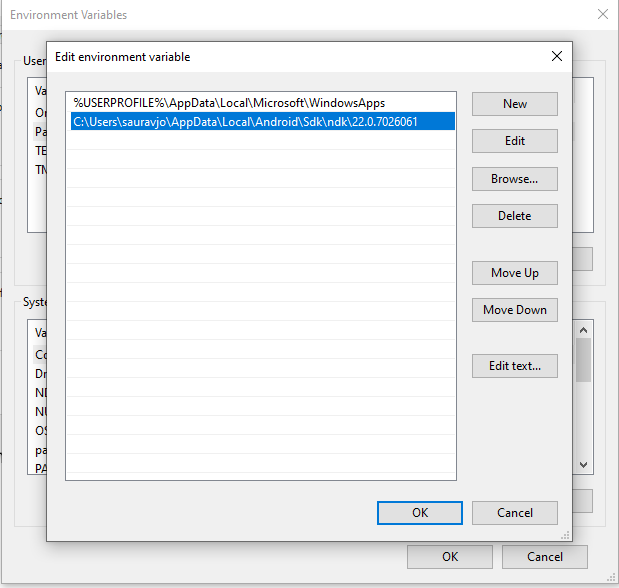
Setting up the environment.

1.Firstly you will need to download ndk in your android studios.

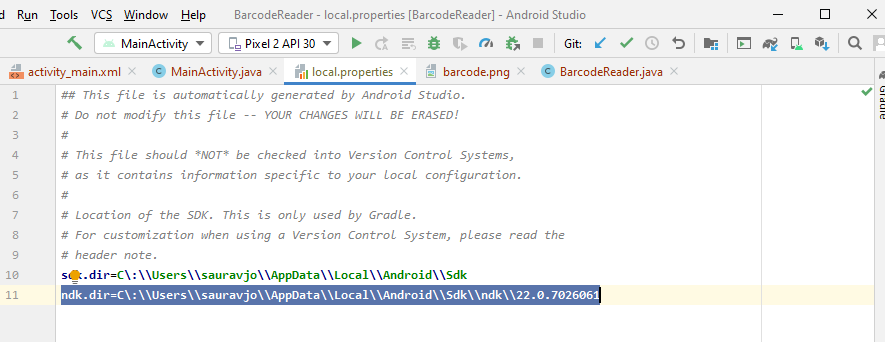
Go to: Tools->SDK manager->sdk tools->NDK(Side by Side).

Checked box on ndk(side by side) and clicked download.



2.Copy your ndk path once your are finished with the download. Add it in your system environment variable.

3.Defining your ndk path in your local properties file of android studio project.



Defining native method:

Following is the syntax for defining the native method.

extern "C" JNIEXPORT return\_type JNICALL

Java\_packagename\_classname\_nativefunctname (JNIEnv \* env, jobject obj){

}

NOTE: Here packagename and classname is of file in which you will call your native function.

For eg:

extern "C" JNIEXPORT jstring JNICALL

Java\_com\_zxing\_BarcodeReader\_display(JNIEnv \* env, jobject obj) {

return env->NewStringUTF("CALLING NATIVE FUNCTION");

//return("CALLING NATIVE FUNCTION");

}

Adding native method to your project:

1.Navigate through your project folder and reach .cpp file in which you want to add native functionality.(wrappers/android/jni/BarcodeReader.cpp).

2. Add your native code to the file with correct syntax.

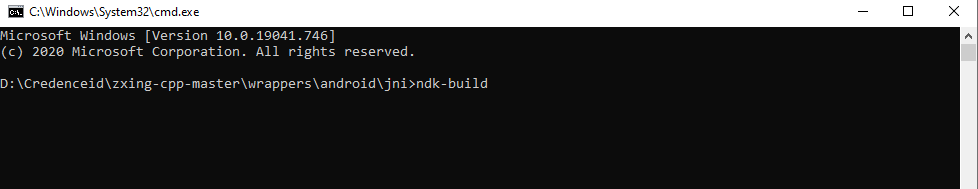
3.Now you need to delete your previous .so files which are in “libs” folder.

(you can create backup for previous .so file if you feel it’s neccessary)

4.Generating new .so files

a) Go to the path in which your libs folder is present and open command prompt from there and fire ndk-build or ndk-build -j <number of your CPU cores> command.

(In this case: cd into wrappers/android/jni and fire above mentioned command).



b) Copy generated files from ‘libs’ into corresponding folder of your android project. (In this case: create folder ‘jniLibs’ in your main directory of android studio project and copy them in it.)

c) You are ready to call/implement your native functionality.

Calling native method.

Your native function can be called just like any other methods in java.

(In our project we have created show() method in java file and called our native method display() in MainActivity.java of android studio project by using object of java class ).