NativeLibCompression

====================

There are lots of NDK apps on Google software market. To reduce package size, some ISV will only release Separate APK. A native library compression sdk is given to solve the apk size problem. It is easy to integrate and will get max 50% size decreasing. Beside sdk, a Java tool for package is provided to convert normal apk to compressed apk.

**HOW TO USE IT:**

1.Include DecRawso into your project (if you use ant, please copy **DecRawso\_Jar**to your project , and add the Decrawso.jar, do not use the jar in the sdk bin folder)

2.Call DecRawso.NewInstance **before any native library loading!!!**

3.Replace all system.loadlibrary(\*\*\*) to system.load(DecRawso . GetInstance ().GetPath(\*\*\*))

now, it is recommend to change to system.load, but system.loadlibrary also work.

--- build your apk as usual, and run your apk as usual when in your development, the apk is not compressed.

**HOW TO COMPRESS THE APK: -- Use compress tool : ApkLibCompress**

1.You can use it as: **ComPressApk.jar -a C:/my/test.apk -k c:/key storepass keypass alias [your keyname] -x86 http://www.test.com**

2.if “-k” is missing, eclipse default test key will be used to sign this apk.

3.[you keyname] is optional, if not have it. the defalt **CERT**will be used

4.If -x86 with link is used, then x86 library will be stored on http://www.test.com/cloudrawso\_x86,   you must store the lib on the network bu manuanlly.

5.you can put **arm lib on x86 folder**to avoid library miss on x86 devices, use -**nox86check**to forbidden the check (x86 directly cal arm lib is **unsafed**)

6.you can copy all of "**DecRawso\_Jar**" into your project if you use "ant" to package your project

7.new flag:

  -o outputfilename      define the finaly output file name

    -slience                   no popup window, that is suitable for ant package

    -nosign                    do not sign the apk, that is suitable for ant package , due to the ant will sign apk

    -nox86check    do not check x86 library missing and mix use of arm issue (x86 directly call arm library is forbidden default)

  8.how to know the result

    now will create 3 files in the ApkLibCompress.jar folder

    :Done.flag   you can check whether the file is exist , if exist , then **compression is ok**

    :error.log    if generation fail, the log will has the reason

    :porting.log  it will show the x86/arm mix using or x86 lib missing issue