附带两个实现例子

一个输出一串字符

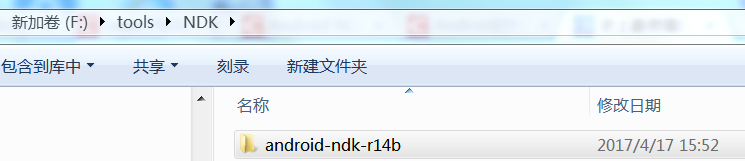
一个实现两个数字的加法

NDK提供了一系列的工具，帮助开发者快速开发C（或C++）的动态库，并能自动将so和java应用一起打包成apk。这些工具对开发者的帮助是巨大的。

NDK集成了交叉编译器，并提供了相应的mk文件隔离CPU、平台、ABI等差异，开发人员只需要简单修改mk文件（指出“哪些文件需要编译”、“编译特性要求”等），就可以创建出so。

下载NDK到本地，并将其路径配置到环境变量的path路径中

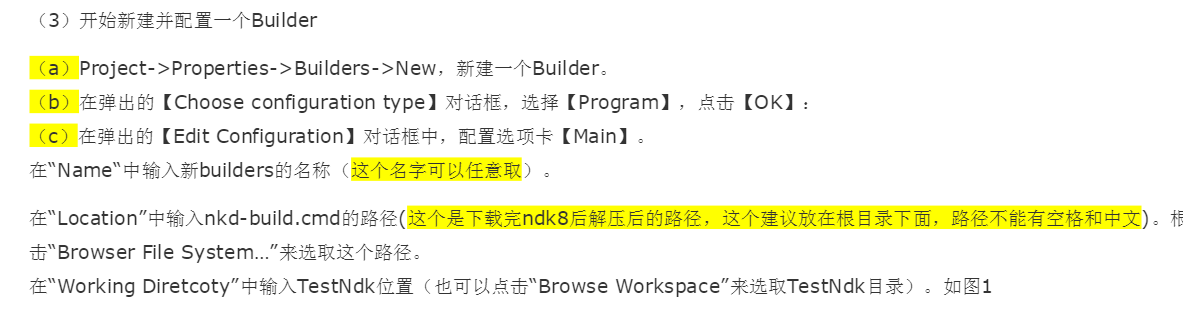
;F:\tools\NDK\android-ndk-r14b;

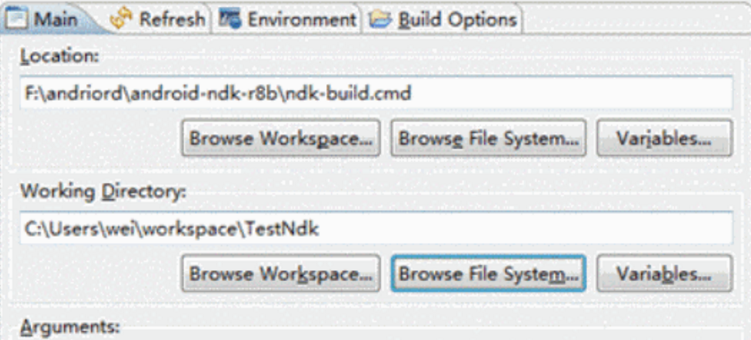


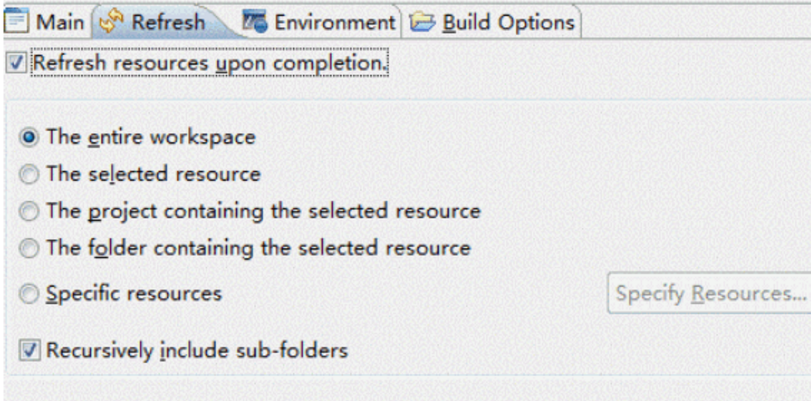
进入cmd查看是否配置成功ndk-build

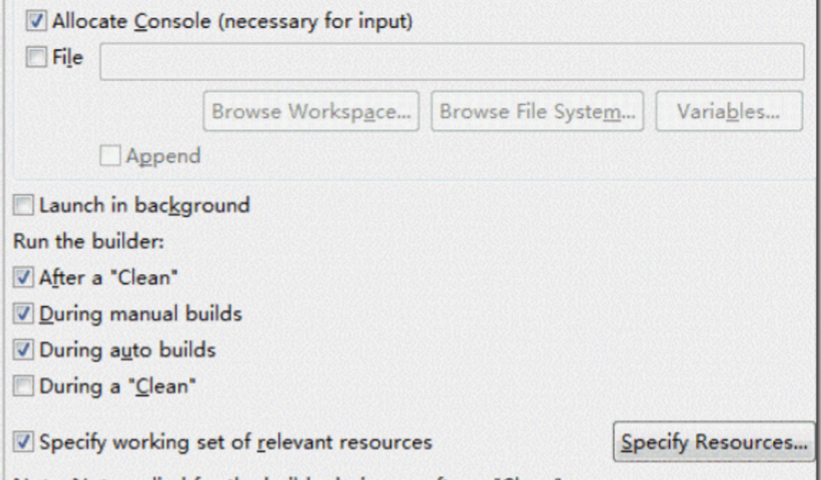
新建Android项目，并新建Android目录下的jni文件夹

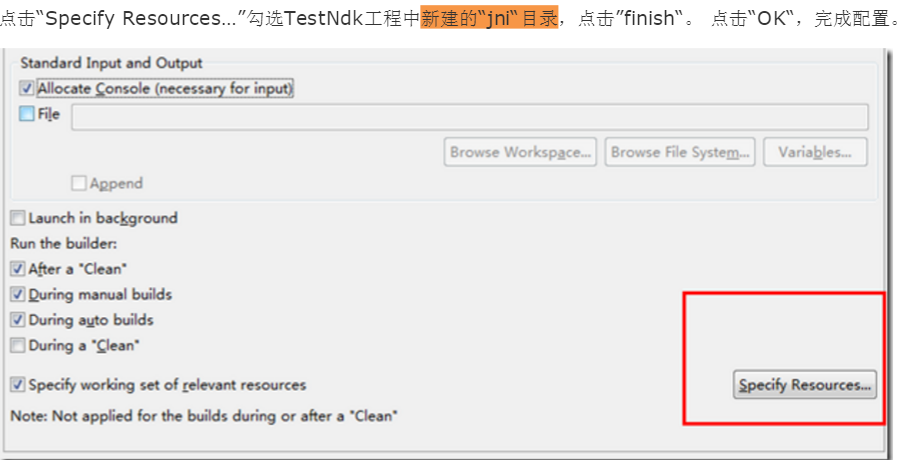
将下载的NDK集成到eclipse中











将上述操作 按照步骤完成，就完成了jni的编译环境了

接下来就可以自己写一个jni的小例子练习练习了

1、在Android项目中创建一个.class文件，注意native关键字声明 ，用于调用C/C++代码例如：

package com.ndk.test;

public class JniClient {

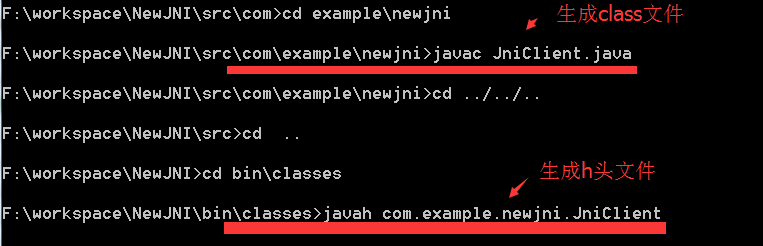
static public native String AddStr(String strA, String strB); }

2、在cmd中通过命令生成相关的.class 与.h文件

注意：

在生成.h文件前可以clean或者build下文件是，其实已经自动生成为.class文件，并且也生成了对应目录下bin\classes\+包名（已经含有.class文件，可以省略javac步骤）

然后可以进入相关目录通过javah命令生成相关的.h头文件



将生成的.h头文件拷贝到新建项目的jni目录下，并新建Android.mk文件（注意A为大写）

内容如下

LOCAL\_PATH := $(call my-dir)

include $(CLEAR\_VARS)

LOCAL\_MODULE := TestNdk //定义生成的.so文件的名字

LOCAL\_SRC\_FILES := com\_ndk\_test\_JniClient.c//此为.h文件对应的实现类的.c文件的文件名

include $(BUILD\_SHARED\_LIBRARY)

添加实现方法的对应的.c/.c++ 文件

注意：C与C++文件中的实现方法还是有一定的区别的

#include "com\_ndk\_test\_JniClient.h"

#include <stdlib.h>

#include <stdio.h>

#ifdef \_\_cplusplus

extern "C"

{

#endif

/\*

\* Class: com\_ndk\_test\_JniClient

\* Method: AddStr

\* Signature: (Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;

\*/

JNIEXPORT jstring JNICALL Java\_com\_ndk\_test\_JniClient\_AddStr

(JNIEnv \*env, jclass arg, jstring instringA, jstring instringB)

{

jstring str = (\*env)->NewStringUTF(env, "HelloWorld from JNI !");

return str;

}

#ifdef \_\_cplusplus

}

#endif

Activity实现方法

**public** **class** MainActivity **extends** Activity {

**public** TextView tv;

**static** {

System.*loadLibrary*("TestNdk");

}

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

tv = (TextView) findViewById(R.id.*tv*);

String str = testjni.*AddStr*("test", "test");

tv.setText(str);

}

}