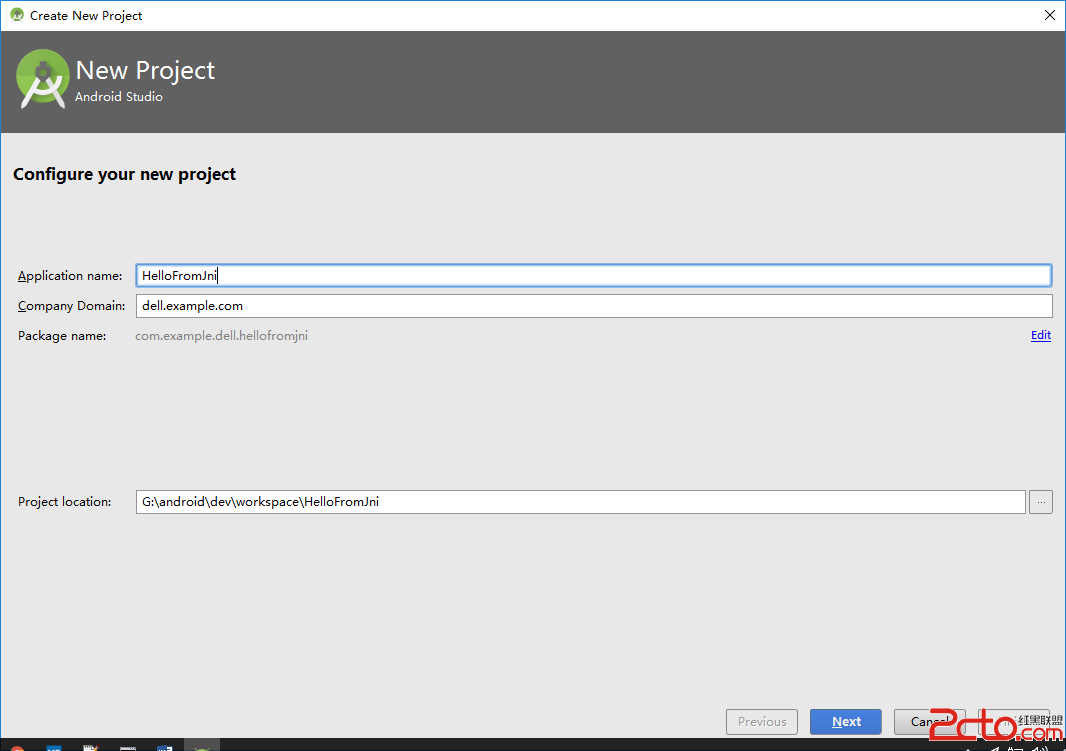
## **Android Studio 2.0下创建jni应用调用本地C函数**

## 开发环境

jdk1.8.0\_77 sdk 25.1.1 ndk android-ndk-r11c-windows-x86\_64 Android Studio 2.0

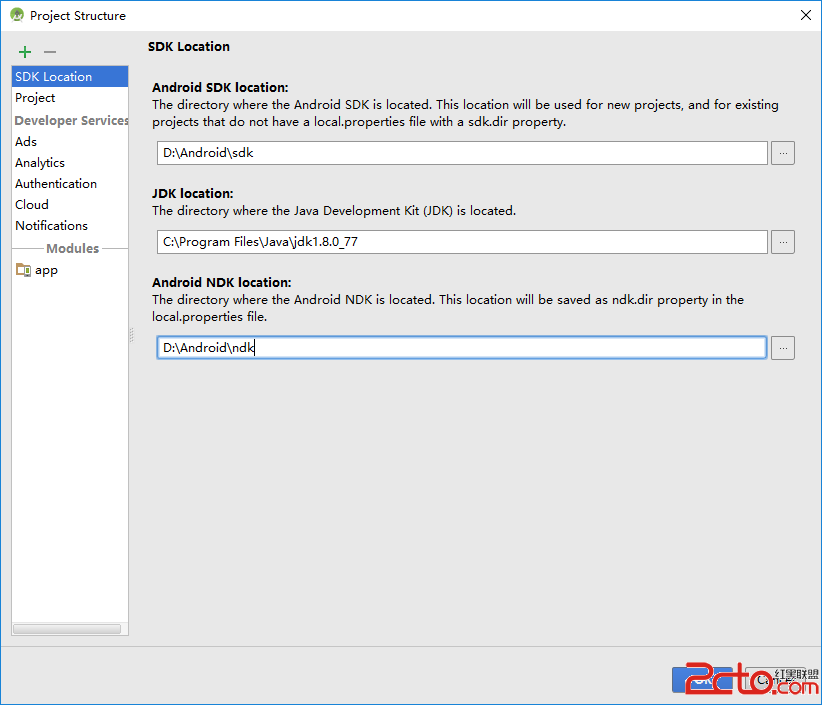
## 详细步骤

### 一：新建工程HelloFromJni



### 二：配置工具

#### 1 切换到project视图右键打开Module setting，添加NDK目录



#### 2 在build.gradle文件的defaultConfig节点中类似添加

defaultConfig {

ndk {

moduleName "hello-jni"

}

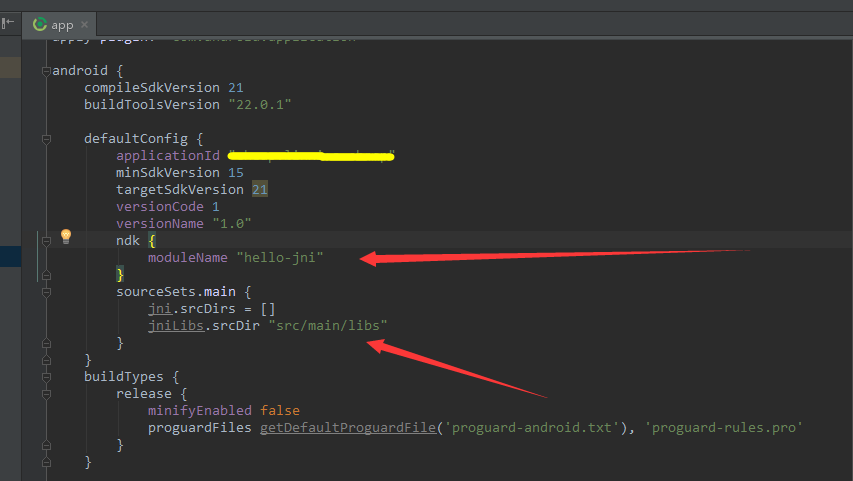
sourceSets.main {

jni.srcDirs = []

jniLibs.srcDir "src/main/libs"

}

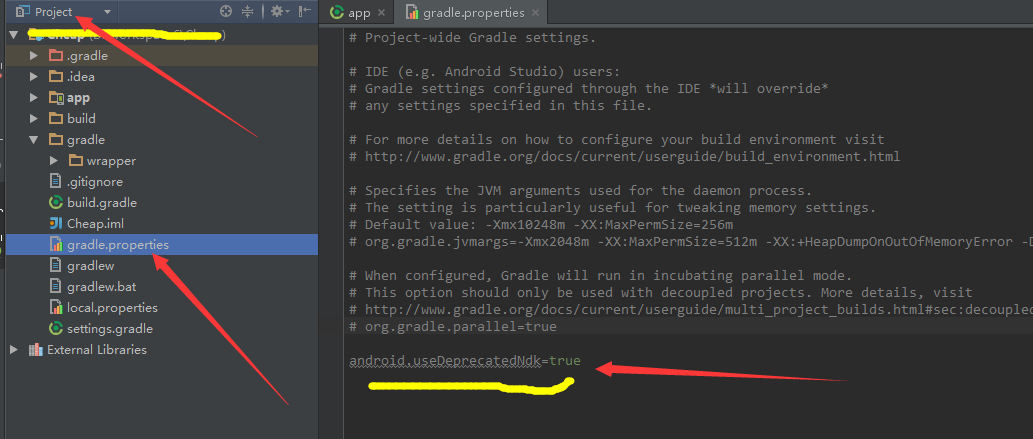
}



如果添加上面的代码后上方提示：

NDK support is an experimental features and all use cases are not yet supported

那就填一句： android.useDeprecatedNdk=true

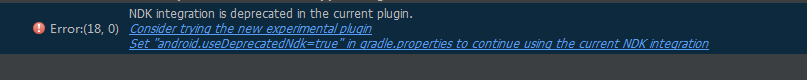


或者一下报错：和上面的改法一样

Error:(18, 0) NDK integration is deprecated in the current plugin.

Consider trying the new experimental plugin

Set "android.useDeprecatedNdk=true" in gradle.properties to continue using the current NDK integration



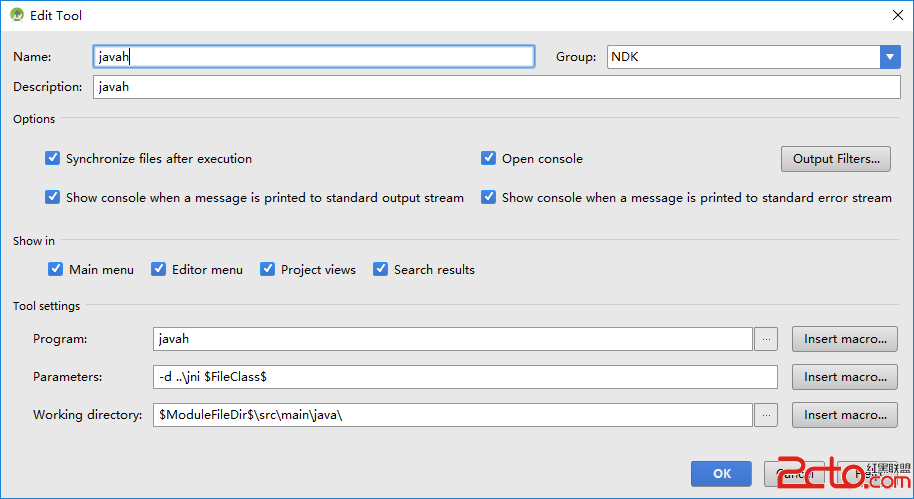
#### 3 在Settings > Tools > External Tools中添加命令行工具（NDK）如下：

##### 1 )添加javah （以便根据MainActivity生成相应头文件）

Javah

-d ..\jni $FileClass$

$ModuleFileDir$\src\main\java

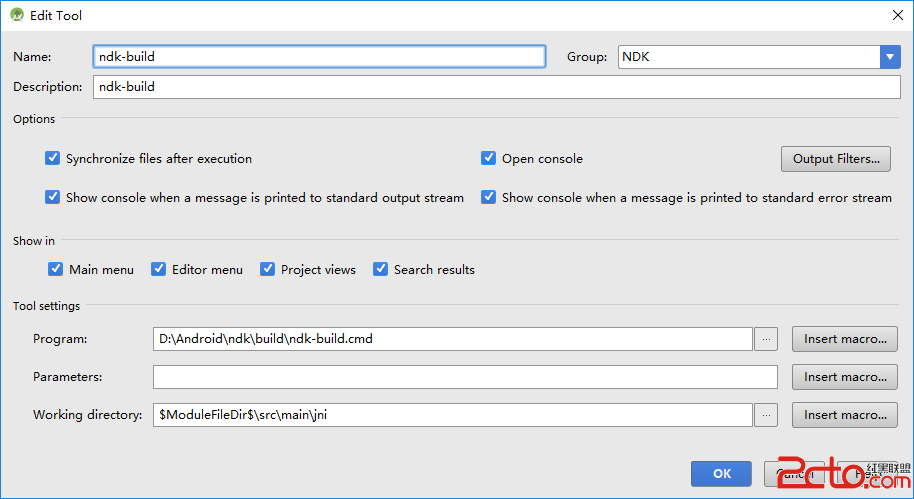


##### 2 )添加ndk-build.cmd编译命令工具

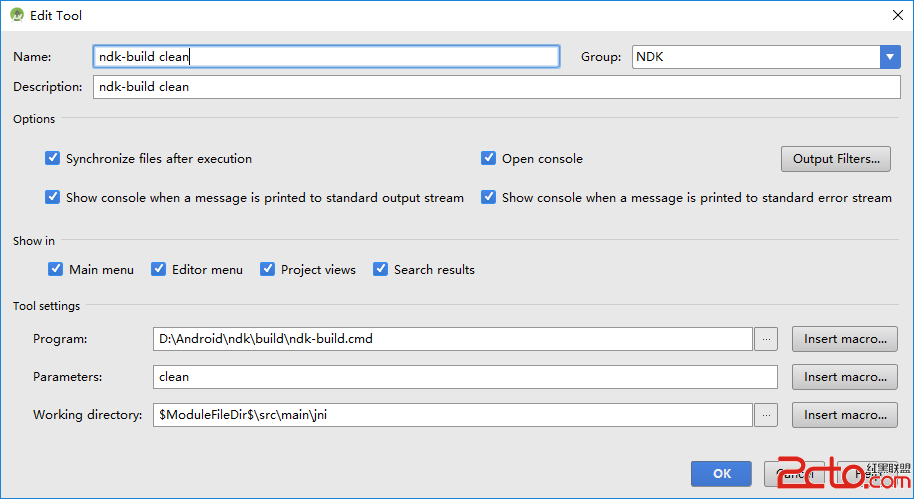
ndk-build

ndk-build.cmd 所在路径

$ModuleFileDir$\src\main\jni



##### 3 添加ndk build clean工具



#### 4 编辑MainActivity.java文件添加本地方法声明，并加载类库（此处为hello-jni）,示例代码如下:

package com.example.dell.hellofromjni;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

static{

System.loadLibrary("hello-jni");

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

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

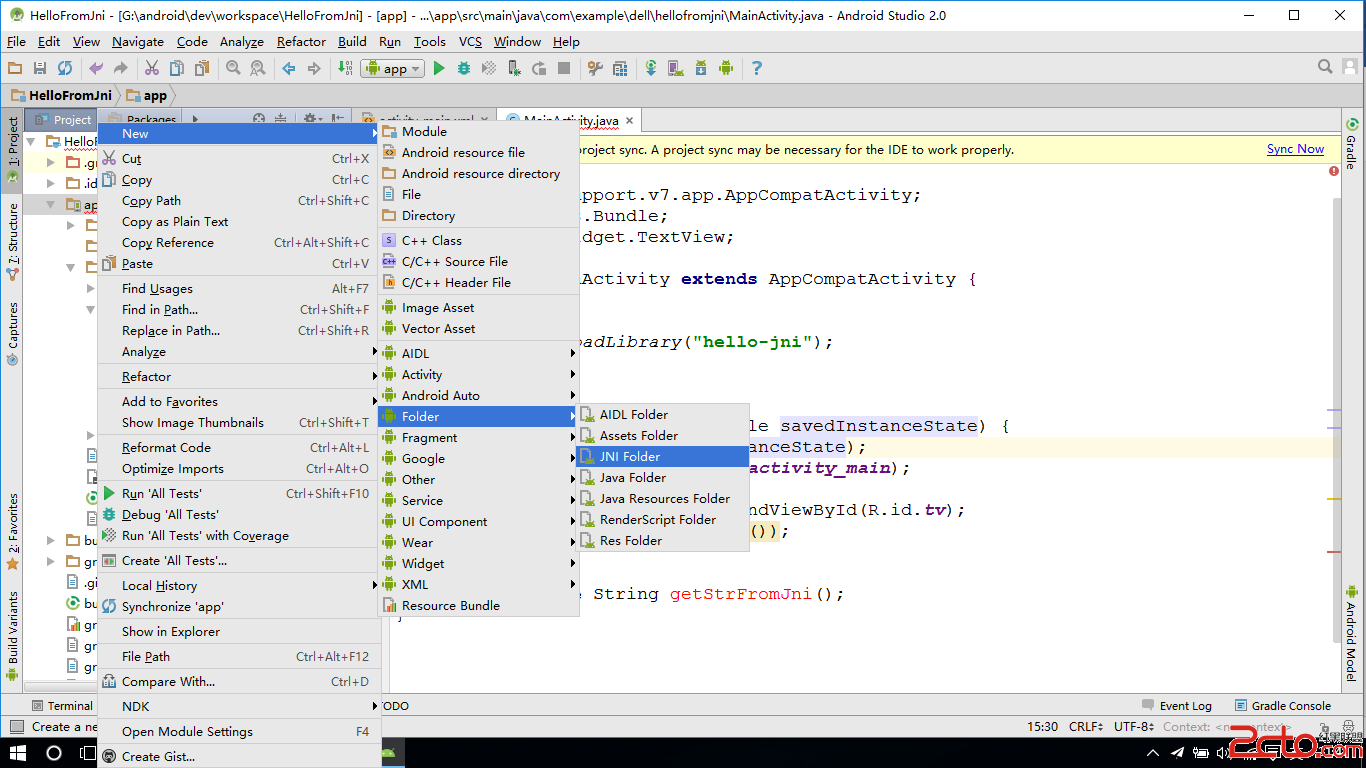
tv.setText(getStrFromJni());

}

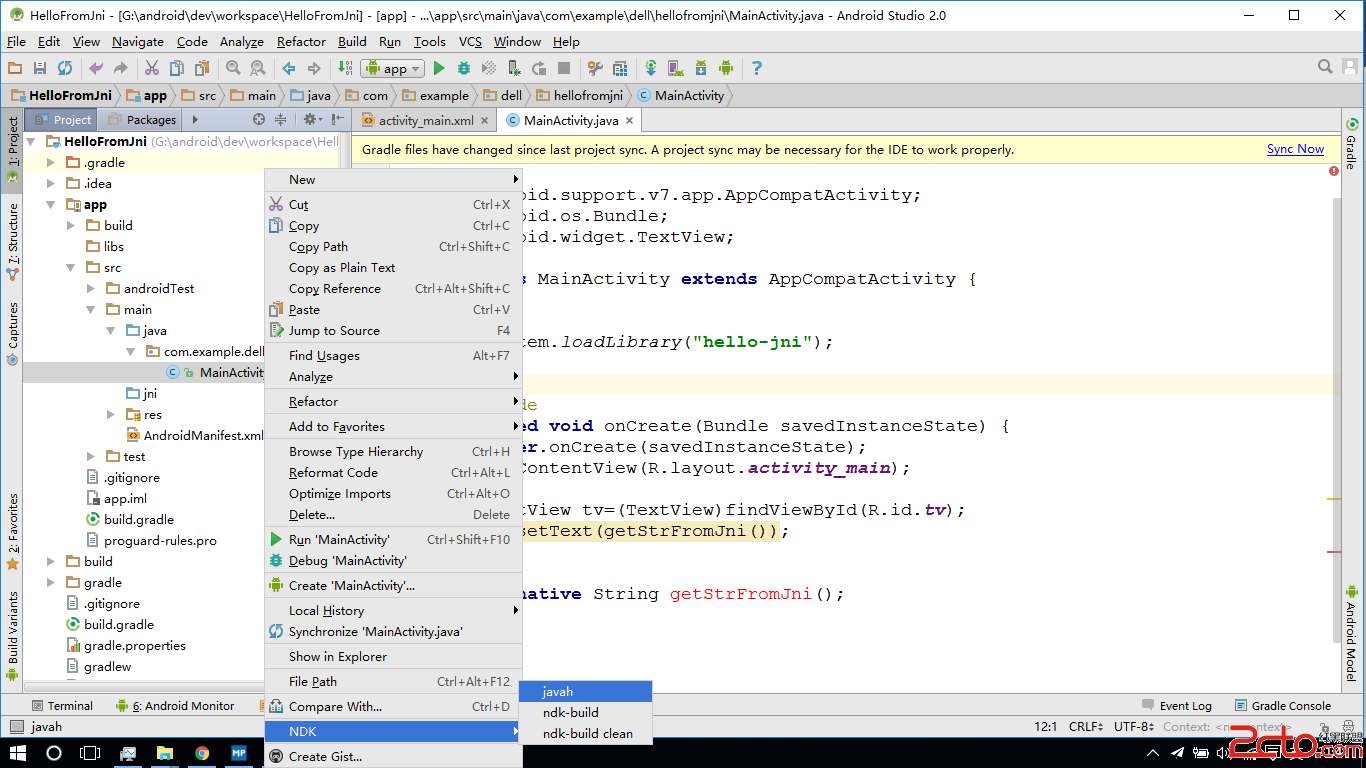
public native String getStrFromJni();

}

#### 5 在app上右键生成jni目录



#### 6 在MainActivity.java上右键选择NDK工具javah，在jni目录中生成com\_example\_dell\_hellofromjni\_MainActivity.h文件



#### 7 在jni目录中新建并编写hello-jni.c文件，函数可以直接在刚才生成的头文件中靠过来并添加参数和函数体：

#include <jni.h>

#include "com\_example\_dell\_hellofromjni\_MainActivity.h"

JNIEXPORT jstring JNICALL Java\_com\_example\_dell\_hellofromjni\_MainActivity\_getStrFromJni

(JNIEnv \* env, jobject obj){

char\* cstr = "hello from c";

return (\*env)->NewStringUTF(env, cstr);

}

#### 8 在jni中新建编译配置文件Android.mk和Application.mk

#### Android.mk

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

include $(CLEAR\_VARS)

LOCAL\_MODULE := hello-jni

LOCAL\_SRC\_FILES := hello-jni.c

include $(BUILD\_SHARED\_LIBRARY)

Application.mk

APP\_MODULES := hello-jni

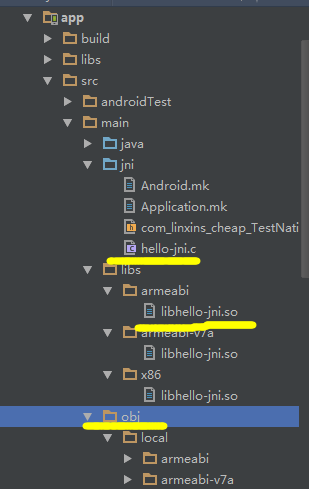
APP\_ABI := armeabi armeabi-v7a x86

或者

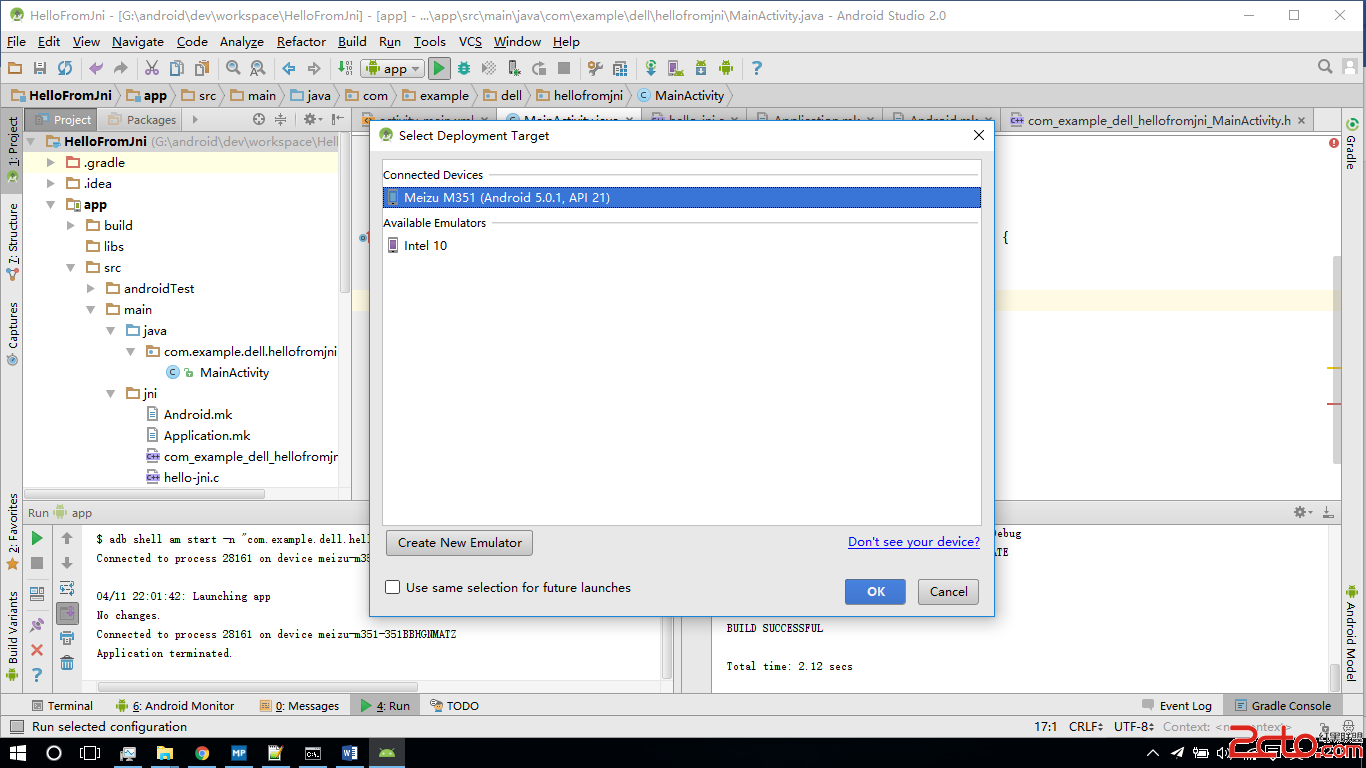
APP\_ABI := all

#### 9 在jni文件夹上右键选择NDK> ndk-build编译c代码，如果发生错误应该用ndk build clean一下清楚编译生成的类库再修改错误IMG_256

**成功就有如下图所示的结构：**



#### 虚拟机或者真机上测试运行了">10 现在就可以在虚拟机或者真机上测试运行了：





来自： 