

MICROOH 麦可网

# Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

C02\_b

# 认识JNI开发与NDK(b)

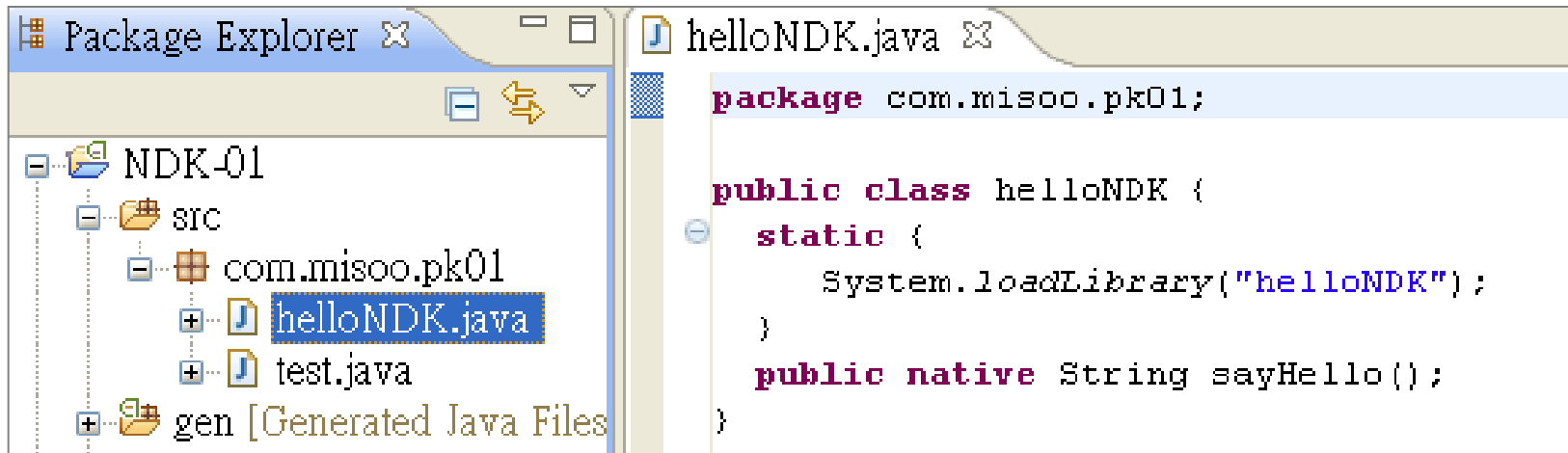
By 高煥堂

## 2、使用Android NDK

- 当你安装好NDK环境之后，就能动手利用NDK环境来开发本地(Native)的C程序了。于此，兹举例说明开发程序。

## Step-1. 在Android SDK环境里， 建立一个开发项目

- 例如，建立一个名称为NDK-01的应用程序开发项目，内含helloNDK.java和test.java程序。其中，helloNDK.java的内容如下：



Java

\*.class 類定義

native String sayHello();

C

\*.h 頭文件

- 把焦点放在接口(Interface)，就是函数：  
native String sayHello();

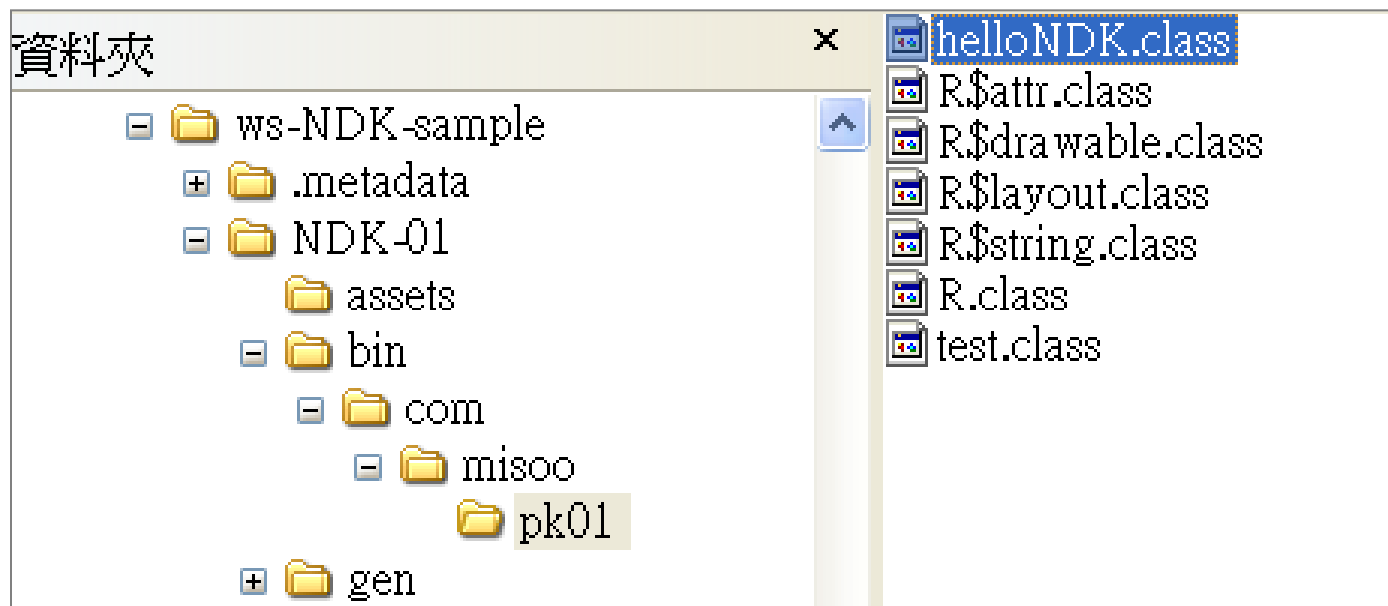
## 同一个接口的两份文档

- 可以将这个Java类定义，看成为这项接口的Java方叙述文件。
- 由于这项接口，涉及两种语言，所以应该有两份文件，两种语言各一份。
- 所以，我们需要替这项接口产出一份C语言方的叙述文件，其形式就是C的头文件(Header File)。



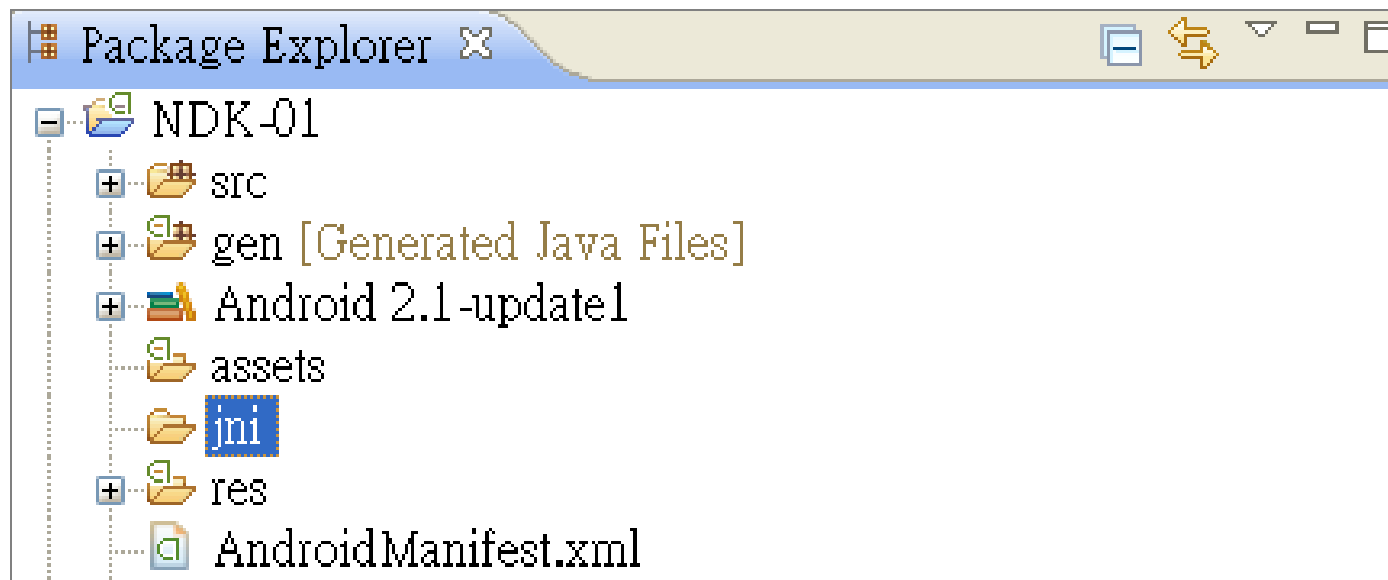
## Step-2. 进行编译， 产出helloNDK.class档案

- 编译上述的项目，产生\*.class档案。

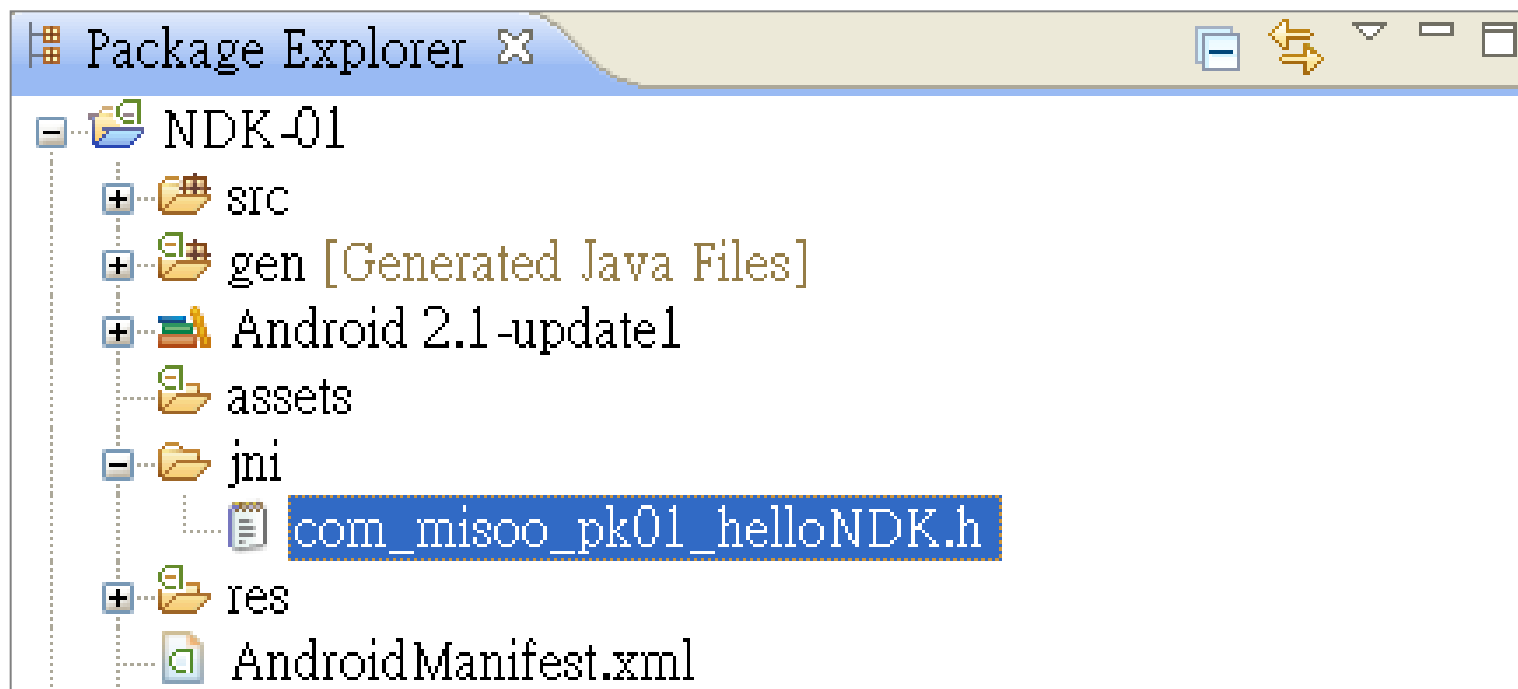


## Step-3. 使用javah工具， 产出C语言的\*.h头文件

- 返回Android SDK环境，建立一个名称为/jni/的新档案夹(Folder)如下：



- 进入/jni/目录区，执行javah去读取 /bin/helloNDK.class档案，然后产出 com\_misoo\_pk01\_helloNDK.h头文件。
- 返回到Android SDK环境，刷新(Refresh)之后，可在Eclipse画面上看到该头文件如下：



- 可以打开com\_misoo\_pk01\_helloNDK.h头文件，其内容如下：

## Package Explorer x

NDK-01

src

com.misoo.pk01

helloNDK.java

test.java

gen [Generated Java]

com.misoo.pk01

R.java

Android 2.1-update1

assets

jni

com\_misoo\_pk01

res

drawable-hdpi

drawable-ldpi

drawable-mdpi

layout

values

AndroidManifest.xml

default.properties

## com\_misoo\_pk01\_helloNDK.h - 記事本

檔案(E) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class com_misoo_pk01_helloNDK */

#ifndef _Included_com_misoo_pk01_helloNDK
#define _Included_com_misoo_pk01_helloNDK
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class:      com_misoo_pk01_helloNDK
 * Method:     sayHello
 * Signature:  ()Ljava/lang/String;
 */
JNIEXPORT jstring JNICALL
    Java_com_misoo_pk01_helloNDK_sayHello
        (JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif
```

## Step-4. 依据\*.h头文件而撰写 \*.c程序码

- 产出com\_misoo\_pk01\_helloNDK.h头文件之后，就可以将NDK-01开发项目内容拷贝(或只拷贝/jni/目录区内容)，拷贝到NDK的/samples/目录里如下：

## 資料夾

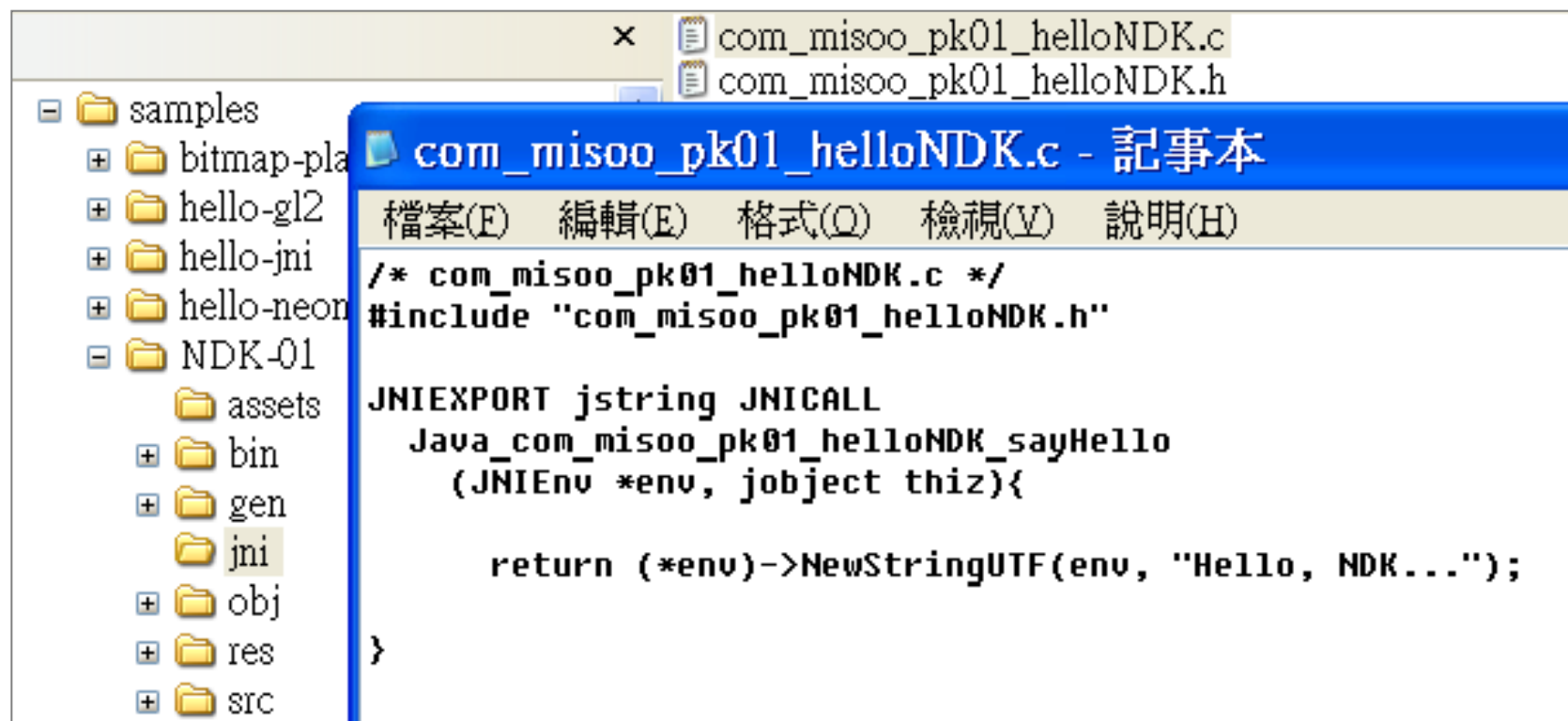


- [-] android-ndk-r4b
  - [+] build
  - [+] docs
  - [-] samples
    - [+] bitmap-plasma
    - [+] hello-gl2
    - [+] hello-jni
    - [+] hello-neon
    - [+] **NDK-01**
    - [+] san-angeles
    - [+] two-libs
  - [+] sources



- 接着，本地C开发者就能使用C语言，结合JNI(Java Native Interface)语法，撰写com\_misoo\_pk01\_helloNDK.c程序码，如下：





## Step-5. 编译及连结本地程序

- 必须先开启Cygwin。也就是，从桌面或<开始/所有程序/Cygwin>里，点选<Cygwin bash shell>，进行编译和连结动作。
- 就完成编译和连结任务，产出libhelloNDK.so本地程序库，并放置于/libs/armeabi/里，如下：

資料夾



- [-] samples
  - + bitmap-plasma
  - + hello-gl2
  - + hello-jni
  - + hello-neon
  - [-] NDK-01
    - assets
    - + bin
    - + gen
    - jni
    - [-] libs
      - armeabi
    - + obj



libhelloNDK.so  
SO 檔案  
14 KB

- 当C开发端完成libhelloNDK.so程序库之后，就可以将/samples/NDK-01内容(或是只拷贝/libs/目录区内容)，拷贝回去Android SDK环境里。于是在Android SDK环境里可以看到libhelloNDK.so本地程序库，如下：

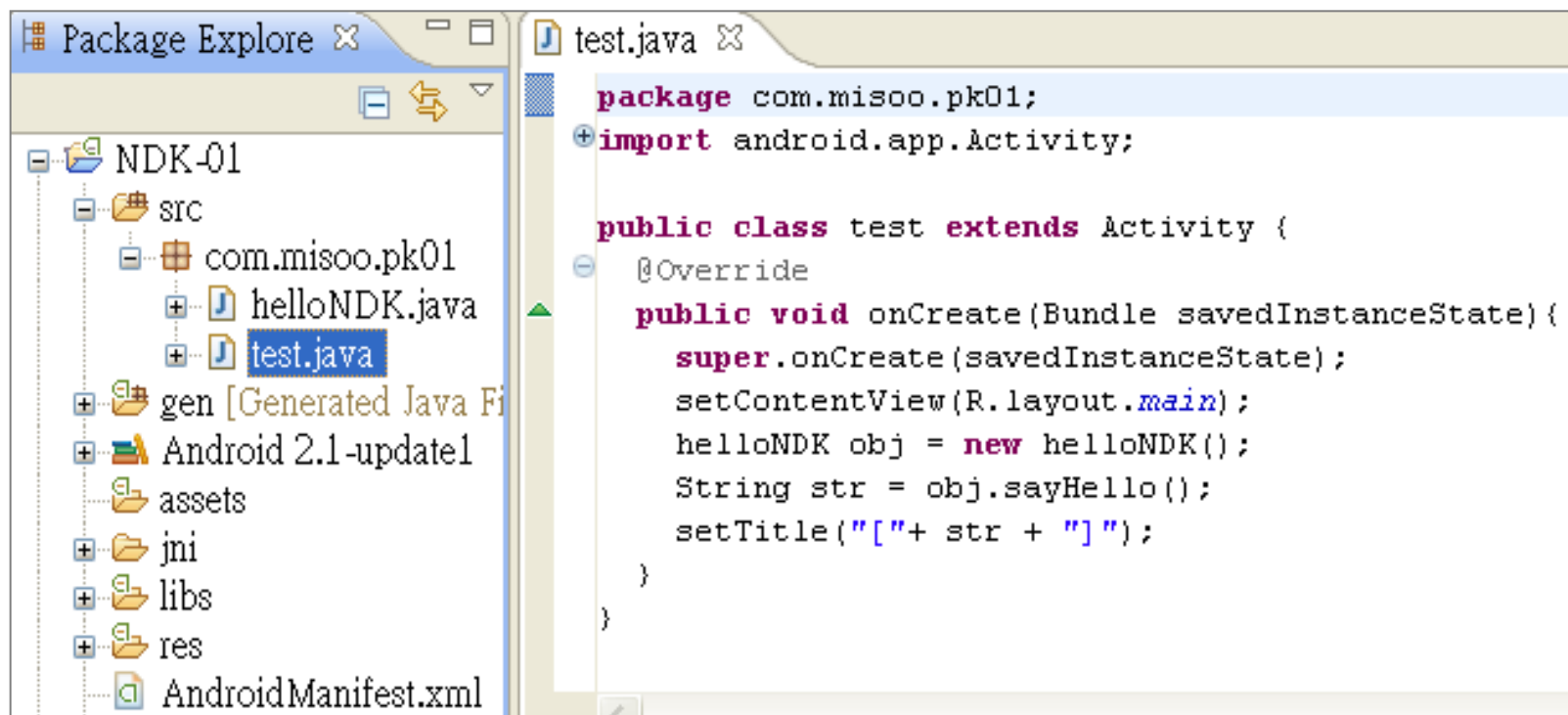
Package Explorer



- NDK-01
  - src
  - gen [Generated Java Files]
  - Android 2.1-update1
  - assets
  - jni
  - libs
    - armeabi
      - libhelloNDK.so
  - res
  - AndroidManifest.xml
  - default.properties

## Step-6. 编执行NDK-01范例程序

- 此时，就可以撰写test.java的内容，让它调用helloNDK.java类别的本地函数，如下：



## Step-7. 将\*.so打包到\*.apk

- 接着编译NDK-01项目，将\*.so本地程序库打包到\*.apk里，并且执行该\*.apk。执行到指令：`obj.sayHello()`时，就调用到\*.so程序库里的本地C程序。于是，`test.java`就将本地C程序回传值显示于画面，如下：



5554:ASPIRE



11:53 PM

[Hello, NDK...]

Welcome to NDK!





~ Continued ~