前期准备：

下载Cygwin，要求版本至少在1.7以上。

Cygwin地址：<http://www.cygwin.com/>

下载Android-NDK，R4出现重大更新，要求版本R5以上

Android-NDK地址：<http://developer.android.com/sdk/ndk/index.html>

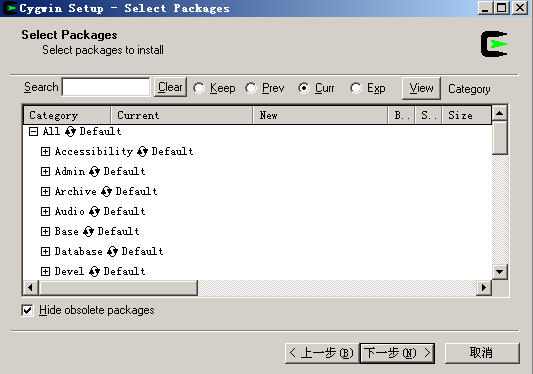
获得已有第三方库以及头文件

环境安装：

Cygwin安装：

双击安装文件，按步骤进行，保证安装路径无中文无空格无特殊字符

默认package是不安装的，需要手动选择必须的package，**编译所须的binutils，gcc，make都在 Devel库中，请选择安装**。

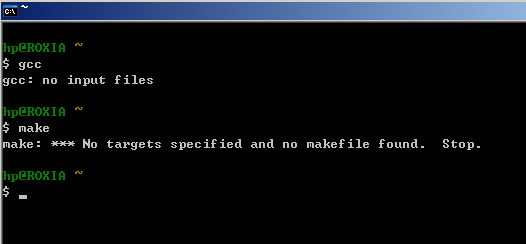


**点击Default，将其变为Install，并选择下一步，直到完成安装**



打开 cygwin.bat 等待初始化直到出现linux命令行

输入gcc,make 出现正常反应则安装完成



解压缩Android-NDK待用，同样路径不包含空格和特殊字符中文等

配置NDK环境变量

首先找到cygwin的安装目录，找到一个home\<你的用户名>\.bash\_profile文件，我的是：E:\cygwin\home\Administrator\.bash\_profile，(注意：我安装的时候我的home文件夹下面神马都没有，解决 的办法：首先打开环境变量，把里面的用户变量中的HOME变量删掉，在E:\cygwin\home文件夹下建立名为Administrator的文件夹（是用户名），然后把E:\cygwin\etc\skel\.bash\_profile拷贝到该文件夹下)。

打开bash\_profile文件，添加NDK=/cygdrive/<你的盘符>/<android ndk 目录> 例 如：NDK=/cygdrive/e/android-ndk-r7

export NDK

NDK这个名字是随便取的，为了方面以后使用方便，选个简短的名字，然后保存

打开cygwin，输入cd $NDK，如果输出上面配置的/cygdrive/e/android-ndk-r7信息，则表明环境变量设置成功了。

Eclipse集成及编译

新建Android工程，**保证工程路径中不包含中文符号和空格**，在工程中新建Java类，其中包含封装后的API函数，用来生成jni需要的h头文件。

**public** **class** AndroidARHeader {

**public** **native** **int** loadImages(**char**[][] images, **int**[] widths, **int**[] heights, **int** num);

**public** **native** **int** DetectImage(**char**[] cameraframe, **int** width, **int** height, **double**[][] matrix, **int**[] nMatchID );

**public** **native** **int** addtest(**int** i,**int** j);

}

所有接口方法均为native，参数保证为基本类型。

打开windows cmd命令行

进入工程文件夹的src目录

运行编译命令

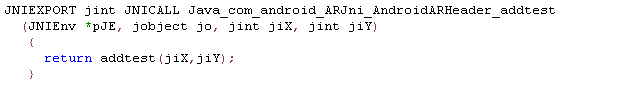




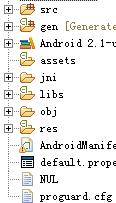
生成JNI用的 h 头文件

新建C文件，完整复制h头文件内容，引用第三方so库的头文件，并实现接口方法，也就是回调第三方库方法。

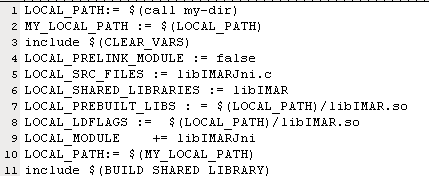




在工程中src同级目录新建jni目录



将 刚刚所写的c文件，第三方头文件以及so库放入该文件夹，并且在该文件夹下创建Android.mk文件



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

获取local\_path 路径

MY\_LOCAL\_PATH := $(LOCAL\_PATH)

存储一个根路径的指针

include $(CLEAR\_VARS)

清除所有local变量

LOCAL\_PRELINK\_MODULE := false

不引入前制模块

LOCAL\_SRC\_FILES := libIMARJni.c

本地文件路径

LOCAL\_SHARED\_LIBRARIES := libIMAR

第三方库文件，不用加后缀.so

LOCAL\_PREBUILT\_LIBS : = $(LOCAL\_PATH)/libIMAR.so

预载入库文件，第三方库绝对路径

LOCAL\_LDFLAGS := $(LOCAL\_PATH)/libIMAR.so

这一句是必须的，加载库文件标识，第三方库绝对路径

LOCAL\_MODULE += libIMARJni

当前模块名称，因为引用了别的模块所以使用 +=

LOCAL\_PATH:= $(MY\_LOCAL\_PATH)

重置LOCAL\_PATH指针

include $(BUILD\_SHARED\_LIBRARY)

创建动态库

完成后保存

在eclipse中集成jni的编译器

选择工程 选择属性，选择builders

新建立一个编译器

Location 指向 Cygwin安装目录下bin文件夹下的 bash.exe文件

Working Directory 选择 bin文件夹

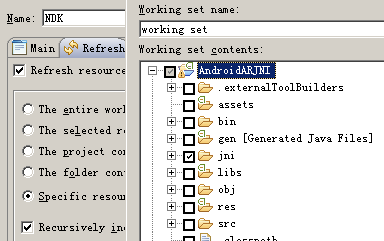
Arguments配置参数

--login -c "cd /cygdrive/d/jni/AndroidARJNI && $ANDROID\_NDK\_ROOT/ndk-build"

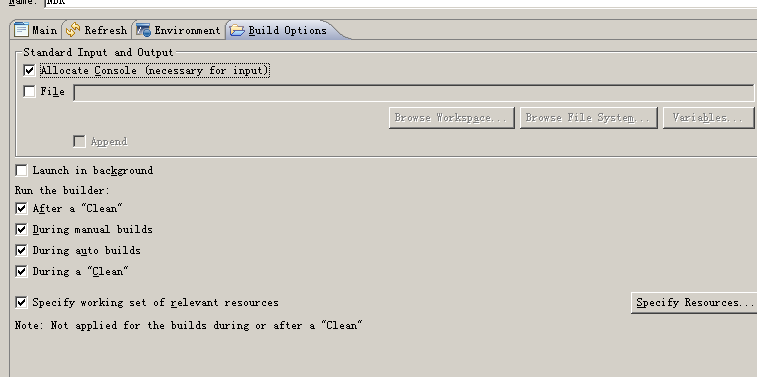
--login –c 登陆cygwin环境

其中 cd /cygdrive 指向电脑硬盘，然后找到自己路径下的工程地址 $ANDROID\_NDK\_ROOT 是开始配置的NDK环境变量，运行ndk-build文件（只有r5及以后版本ndk才有）

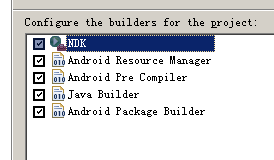
在refresh选项卡，选择specific resource 并关联 工程下的 jni文件夹



Build Options选项卡 选择全部，并关联jni文件夹



将新builder 移动到最上方，这样才能优先编译c文件



如果控制台编译so成功，那配置完成。

在Java主文件的activity 中，通过

Static

{

System.Loadlibrary(“XXXXX”);

}

按顺序载入第三方库和自己新生成的库文件，XXXX是不包含前缀lib和后缀.so的库名

打包前，将第三方so库复制进入jni编译生成的libs\armeabi\ 文件夹下，生成apk时。So库会被自动打包进apk，结果如果运行调用成功。