# 1 ndk环境安装

ndk linux安装包下载地址：

http://dl.google.com/android/ndk/android-ndk-r9b-linux-x86.tar.bz2

或

<http://dl.google.com/android/ndk/android-ndk-r9b-linux-x86_64.tar.bz2>

已下载到251：/home/ubuntu/ndk

下载解压缩。解压方法：tar -jxvf

按照下面配置一下即可。

在~/.bashrc（Linux下配置环境变量的文件）文件末尾加上：

export NDK\_HOME=/home/xxx/android\_env/android-ndk-r9b

export PATH=$NDK\_HOME:$PATH

到这里 NDK 的环境就已经配置好了,在任意目录下就能使用 NDK 提供的工具， NDK 提供的主要的工具是 ndk-build,在任意目录下执行：ndk-build

执行结果：

AndroidNDK: Could not find application project directory !     
Android NDK: Please define the NDK\_PROJECT\_PATH variable topoint to it.     
/home/xxx/android\_env/android-ndk-r8/build/core/build-local.mk:130:\*\*\* Android NDK: Aborting    .  Stop.

说明配置成功。

# 2 生成jni方法

## 2.1 创建jni工程（此步略）

此步略，我已建好，位置：/home/ubuntu/ndk/my-jni

将android-ndk-r9b/samples下的hello-jni拷贝到任一目录，重命名为my-jni。

## 2.2 放入c/cpp代码文件并修改makefile

A、将你要生成so的c/cpp文件拷贝到jni目录下。

注意该c/cpp文件必须是符合jni的文件。参见后面。

B、修改jni目录中的Android.mk文件，内容像下面这样：

LOCAL\_PATH := $(call my-dir)  
include $(CLEAR\_VARS)  
LOCAL\_MODULE    := jnitest  
LOCAL\_SRC\_FILES := jnitest.c  
include $(BUILD\_SHARED\_LIBRARY)  
#include $(BUILD\_EXECUTABLE)

C、在jni目录中创建Application.mk（略，我已创建好），内容如下：

APP\_ABI := armeabi

APP\_STL := stlport\_static

## 2.3 生成so文件

返回到my-jni目录，执行ndk-build即可。按照提示可以找到生成的so文件。

## 2.4 c/cpp代码的特别要求



因为通过jni的方式编译出来的so是供java/android调用的，所以被android/java调用的那些c/cpp函数、参数需要进行适当处理。即上面所说的要添加包名、类名及其它处理。可以参看之前我提供的静态检查代码的例子：

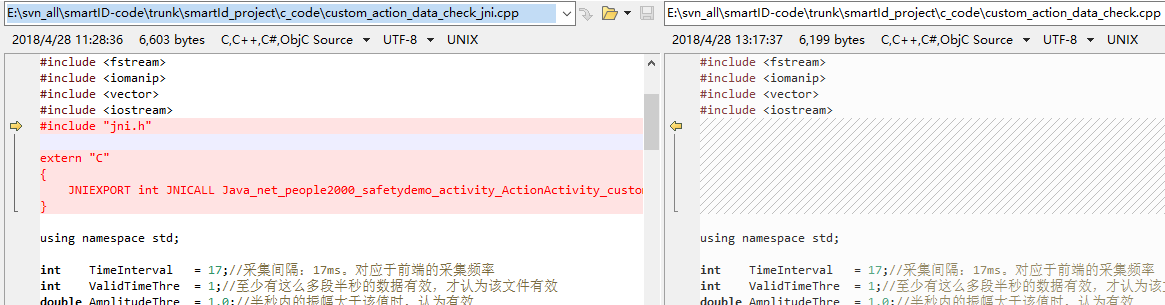
没有进行jni处理的cpp代码：

svn://192.168.1.150/smartID/trunk/smartId\_project/c\_code/custom\_action\_data\_check.cpp

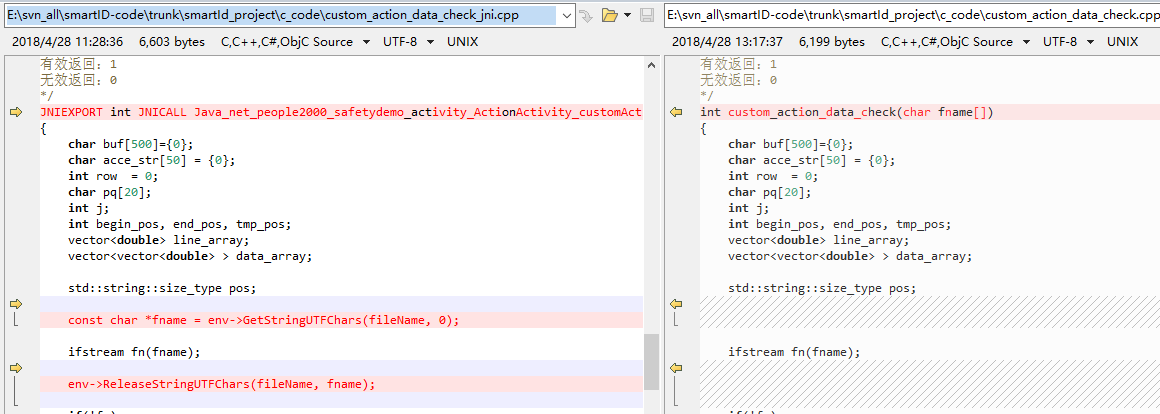
进行jni处理的cpp代码：

svn://192.168.1.150/smartID/trunk/smartId\_project/c\_code/custom\_action\_data\_check\_jni.cpp

区别：



区别：



刘敬民