neo's Weblog

Just got to be lucky to read me…….

- Home
- About
-

Type text to search here...

Home > Tech > Working with JNI- Java Native Interface in Ubuntu

Working with JNI- Java Native Interface in Ubuntu

August 10, 2009 Vishwanath Kamath Leave a comment Go to comments

To work with JNI we need certain essential tools. They are:

1. A Java Compiler "javac" which ships with the SDK
2. A Java Virtual Machine "java" which ships with the SDK
3. A native method c header file generator "javah" which ships with the SDK.
4. A C/C++ compiler "cc" that can create shared library.

Essential steps to be followed to call C/C++ from the java code

1. Write a java code. (Example as mentioned below) (filename:JNIDemo.java)

```
public class JNIDemo
{
public native void display();
static
{
System.loadLibrary("JNIDemo");
}
public static void main(String args[])
{
try
{
JNIDemo jdo = new JNIDemo();
jdo.display();
}catch(Exception e)
{
System.out.println("Alert Alert Alert: " + e.getMessage());
}
}
}
```

2. Compile the java code. (Change the current folder to point to Source folder using Chage directory command in terminal window) (compilation as mentioned below)

```
javac JNIDemo.java
```

This will create a JNIDemo.Class file in the same folder level as of JNIDemo.java file.

3.Create C/C++ header file. (Compilation as mentioned below)

(Use the Java class file name without the ".class" extention. It would look as below in terminal)

```
javah JNIDemo
```

This would create a C/C++ header file with the native function signatures that we want to call.

The JNIDemo.h file would look as below.

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include
/* Header for class JNIDemo */

#ifndef _Included_JNIDemo
#define _Included_JNIDemo
#ifdef __cplusplus
extern "C" {
#endif
/*
* Class:     JNIDemo
* Method:    display
* Signature: ()V
*/
JNIEXPORT void JNICALL Java_JNIDemo_display (JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif
```

4.Write a C/C++ source code with the implementation.

(Implementation example as mentioned below) (filename:JNIDemo.c)

```
#include "JNIDemo.h"
JNIEXPORT void JNICALL Java_JNIDemo_display(JNIEnv *env, jobject obj)
{
char name[30];
printf("What is your name?\n");
scanf("%s",name);
printf("Hello %s, you are running JNIDemo\n", name);
}
int main()
{
return  0;
}
```

5.Create a shared library file using the C/C++ file created using terminal.

```
cc -o libJNIDemo.so -shared -I/JDK/include -I/JDK/include/linux JNIDemo.c
```

libJNIDemo.so —> This could be considered as the shared library file name you need/want to create. Template of the same could be considered as .so

/JDK —> This represents the JDK installation path

6.Check the LD_LIBRARY_PATH using the command:

```
echo $LD_LIBRARY_PATH
```

In Ubuntu, this path is not set to any location. Hence this would return empty.

Set LD_LIBRARY_PATH to the shared library created by you (libJNIDemo.so) using the command

```
export LD_LIBRARY_PATH='[Folder path of the shared library file]'
```

7.Run the java program .

```
java JNIDemo
```

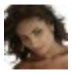
Output would be:

```
What is your name?
neo
Hello neo, you are running JNIDemo
```

Like

Be the first to like this.

Categories: Tech Tags: interface, Java, javac, javah, JNI, native, UBUNTU
Comments (3) Trackbacks (0) Leave a comment Trackback

1. 
   Iresha
   August 17, 2009 at 5:57 am
   Reply

   Good stuff, Here is another JNI tutorial, but using a windows dll.

   http://codediaries.blogspot.com/2009/07/java-native-interface-jni-example-using.html

2. Vishwanath Kamath
   August 17, 2009 at 10:43 am
   Reply

   Thanks for the additional resource Iresha…. 🙂

3. Mahesh
   October 16, 2012 at 7:08 am
   Reply

   Nice tutorial …

1. No trackbacks yet.

Leave a Reply

Enter your comment here...

Resolve OutOfMemory with Eclipse when source base is huge
RSS feed

Advertisements

AdChoices
arm
The Economist Internet of Things Report
Learn what 800 executives think about IoT.
Download Report

Report this ad

Recent Posts

- EMMA code coverage of Android
- Customize your CTS
- Android: Supporting the new music Voice Action
- Track EditText data….
- Sound Solutions for Ubuntu 9.04 (Jaunty) Users
- Hashtable Vs Hashmap, Vector Vs ArrayList
- Steps to build Donut sdk for Eclipse
- Resolve OutOfMemory with Eclipse when source base is huge
- Working with JNI- Java Native Interface in Ubuntu

Categories

- Android
- Linux
- Tech
- Uncategorized

Blogroll

- WordPress.com
- WordPress.org

Archives

- May 2011
- September 2010
- January 2010
- December 2009
- November 2009
- September 2009
- August 2009

Meta

- Register
- Log in

Top