Haskell and the JNI

This is a tutorial for making a Haskell module available for Java via the JNI

Perhaps you have a module Fibonacci where you have a function fibonacci n that you want to call in Java

First you compile the Haskell Module with the exported functions: ghc -c -O Fibonacci.hs

Then you write you Java Class which uses the functions of the Haskell Module:

public class HelloHaskell {

static {

System.loadLibrary("fibonacci"); //no .dll !!!

}

private native static int fibonacci(int n);

public static void main(String[] args) {

System.out.println(fibonacci(42));

}

}

Now Compile the Java class: javac HelloHaskell.java

You have to generate a Bridge.h file by javac -h $HelloHaskell

Then you implement the functions declared in Bridge.h in the file Bridge.c:

#include <HsFFI.h> //hs\_init, hs\_add\_root and hs\_exit

#ifdef \_\_GLASGOW\_HASKELL\_\_

#include "Fibonacci\_stub.h" //include

#include <jni.h>

extern void \_\_stginit\_Safe(void);

#endif

#include <stdio.h>

JNIEXPORT jint JNICALL Java\_HelloHaskell\_fibonacci(JNIEnv \* env, jclass c, jint n)

{

int argc;

char\*\*\* argv;

argc = 0;

hs\_init(&argc, argv);

#ifdef \_\_GLASGOW\_HASKELL\_\_

hs\_add\_root(\_\_stginit\_Safe);

#endif

int i;

i = fibonacci(n);

hs\_exit();

return i;

}

Now you make the fibonacci.dll by invoking ghc and compiling the Bridge.c file

ghc -I"%JAVA\_HOME%\include" -I"%JAVA\_HOME%\include\win32" -shared -o fibonacci.dll --make -no-hs-main -optc -O Bridge.c Fibonacci.o