

 eBook Gratuit

APPRENEZ

jni

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#jni

Table des matières

À propos.....	1
Chapitre 1: Commencer avec jni.....	2
Remarques.....	2
Exemples.....	2
Démarrer avec JNI.....	2
Crédits.....	4

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [jni](#)

It is an unofficial and free jni ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jni.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Commencer avec jni

Remarques

Cette section fournit une vue d'ensemble de ce qu'est jni et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans jni, et établir un lien avec les sujets connexes. La documentation de jni étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Démarrer avec JNI

JNI signifie Java Native Interface. Il s'agit d'un mécanisme permettant d'appeler un code natif à partir d'une application Java exécutée sous le contrôle JVM et inversement. Le code natif signifie un code compilé pour la plate-forme cible. Le code de code natif est généralement écrit en C ou C ++, mais il peut être écrit dans n'importe quel langage doté d'un compilateur pour la plate-forme cible.

JNI est utile quand

- Une application Java doit accéder à des ressources spécifiques à une plate-forme et il n'existe pas de bibliothèque Java avec les fonctionnalités requises. La ressource peut être un matériel spécifique, des capteurs ou autre.
- Une application Java doit effectuer une tâche critique en termes de performances et le code natif peut être plus rapide ou moins encombrant que le bytecode Java. Néanmoins, soyez trop sûr de vous JVM est capable de faire beaucoup d'optimisation et une implémentation naïve en C / C ++ sera probablement plus lente.
- Une application en C / C ++ (ou un autre langage) veut utiliser une fonctionnalité fournie dans une bibliothèque Java.

Pour commencer avec JNI, vous avez besoin de

- JDK ou quelque chose qui est capable de compiler Java en bytecode.
- Compilateur pour compiler le code natif.

L'exemple suivant de hello world est une application Java simple qui appelle une fonction C. L'exemple peut être compilé par javac à partir du compilateur JDK et gcc C.

Code Java:

```
public class JNIExample {  
  
    public static void main(String[] args) {
```

```

    // The loadLibrary search for the native library (libnative.so in this case)
    System.loadLibrary("native");
    String s = "Hello JNI";
    JNIExample example = new JNIExample();
    example.doPrint(s);
}

// The method with native code (written in C) must be declared with native prefix
public native void doPrint(String message);

}

```

Code C:

```

#include <jni.h>
#include <stdio.h>

/* the function that is called from java must be declared with decorators
 * JNIEXPORT and JNICALL.
 * The function name is constructed as Java_ClassName_MethodName
 * Function parameters correspond parameters in java but there are 2 extra parameters
 * JNIEnv is a pointer to java environmet and jobject is a reference to caller object.
 * Caller object is the instance of the JNIExample in this case.
 */
JNIEXPORT void JNICALL Java_JNIExample_doPrint(JNIEnv *e, jobject obj, jstring message) {
    const char *c_message;
    /* It is necessary to convert java objects like string to something C native */
    c_message = (*e)->GetStringUTFChars(e, message, NULL);
    printf("%s\n", c_message);
    /* in the end it is necessary to free resources allocated by Get above */
    (*e)->ReleaseStringUTFChars(e, message, c_message);
}

```

Lire Commencer avec jni en ligne: <https://riptutorial.com/fr/jni/topic/7728/commencer-avec-jni>

Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec jni	Community , Zaboj Campula