



**FREE eBook**

**LEARNING**

**jni**

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#jni**

# Table of Contents

About.....	1
Chapter 1: Getting started with jni.....	2
Remarks.....	2
Examples.....	2
Getting started with JNI.....	2
Credits.....	4

---

# About

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](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with jni

## Remarks

This section provides an overview of what jni is, and why a developer might want to use it.

It should also mention any large subjects within jni, and link out to the related topics. Since the Documentation for jni is new, you may need to create initial versions of those related topics.

## Examples

### Getting started with JNI

JNI means Java Native Interface. It is a mechanism how to call a native code from Java application running under JVM control and vice versa. Native code means code compiled for the target platform. Native code code is typically written in C or C++ but it may be written in any language that has compiler for the target platform.

JNI is useful when

- A java application need to access a platform specific resources and there is no java library with the required functionality. The resource may be a specific hardware, sensors or whatever.
- A java application has to do a performance critical task and native code may be faster or with less footprint than java bytecode. Nevertheless do be too self-confident JVM is able to do a lot of optimization and a naive implementation in C/C++ will be probably slower.
- An application in C/C++ (or another language) wants to use a feature provided in a java library.

To start with JNI you need

- JDK or something that is able to compile java to bytecode.
- Compiler for compiling the native code.

The following hello world example is a simple java application that calls a C function. The example can be compiled by javac from JDK and gcc C compiler.

Java code:

```
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);

}
```

## C code:

```
#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 environment 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);
}
```

Read Getting started with jni online: <https://riptutorial.com/jni/topic/7728/getting-started-with-jni>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with jni	<a href="#">Community</a> , <a href="#">Zaboj Campula</a>