



Essential Jni: Java Native Interface (Essential Java)

By Robert Gordon, Rob Gordon



Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon

With Java maturing and branching out into so many different APIs, programmers are finding it difficult to find good, solid coverage of the many new need-to-know APIs in the many "catch all" Java books on the market. This title answers that need by presenting no-nonsense, nuts-and-bolts coverage of the new "niche" APIs like JNI, JTAPI, and the Java Wallet in a task-oriented reference format that gives coders the answers they need. .

 [Download Essential Jni: Java Native Interface \(Essential Ja ...pdf](#)

 [Read Online Essential Jni: Java Native Interface \(Essential ...pdf](#)

Essential Jni: Java Native Interface (Essential Java)

By Robert Gordon, Rob Gordon

Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon

With Java maturing and branching out into so many different APIs, programmers are finding it difficult to find good, solid coverage of the many new need-to-know APIs in the many "catch all" Java books on the market. This title answers that need by presenting no-nonsense, nuts-and-bolts coverage of the new "niche" APIs like JNI, JTAPI, and the Java Wallet in a task-oriented reference format that gives coders the answers they need. .

Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon Bibliography

- Sales Rank: #138829 in Books
- Published on: 1998-03
- Original language: English
- Number of items: 1
- Dimensions: .93" h x 6.10" w x 8.97" l,
- Binding: Paperback
- 528 pages

 [Download Essential Jni: Java Native Interface \(Essential Ja ...pdf](#)

 [Read Online Essential Jni: Java Native Interface \(Essential ...pdf](#)

Download and Read Free Online Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon

Editorial Review

Amazon.com Review

The Java Native Interface (JNI) enables programmers to share data and make function calls between programs written in different languages. A program created in Java can refer to the contents of a program written in C++, and vice versa. The communication also works between Java and C. This book explains how to use the JNI to make Java programs communicate with their neighbors.

First things first. As is customary in programming books, the author starts with a "Hello, World" example, except here, the program is written in both C and Java. Gordon walks you through the source code in each language, explains compilation, and quickly gets to the mechanics of using the javah utility to generate an include-able .h file.

After the procedural walk-through, Gordon strides into specifics, such as how to get a C/C++ program to set a variable that exists in a Java program and how to deal with native types. He compares object architectures and gets into some detail on arrays, strings, exceptions, and producer-consumer relationships. You'll find plenty of information on input and output, too--the author shows the means of writing to POSIX and Win32 ports. A cool chapter has to do with setting up the Java Developer's Kit as a Windows NT service via the Service Control Manager.

A full JNI class reference with minimal annotation ends the book.

From the Inside Flap

Before we plow an unfamiliar patch
It is well to be informed about the winds,
About the variations in the sky,
The native traits and habits of the place,
What each locale permits, and what denies
Virgil The Georgics

Preface

What This Book is About

The subject of this book is the Java Native Interface (JNI) Application Programming Interface (API). The JNI was introduced in release 1.1 of the Java Development Kit (JDK) as distributed by JavaSoft. This book covers the entire API for the JNI including the enhancements introduced in release 1.2 of the JDK. Where there are minor differences between various 1.1 point releases, these are discussed.

Who Should Read This Book?

This book is written for the software engineer who needs to make Java and C or C++ talk to one another. Experience with C/C++ and Java is assumed. This book also assumes some familiarity with both UNIX and Win32 platforms.

If you are a Java programmer who needs to step outside the Java Virtual Machine to take advantage of some platform-specific functionality, this book will show you how. If you are a C programmer responsible for putting a Java front-end on a legacy application, this book will show you how. If you are a C++ programmer wanting to take advantage of an existing C++ class library, this book will show you how.

Further, this book covers these topics for both UNIX and Win32 platforms.
Okay, now turn around and walk to the sales counter.

Structure of This Book

This book can be thought of as having three distinct parts. The first part, roughly the first eight chapters, covers the JNI API in great detail. The second part, the remaining chapters, covers some general issues involved with native method programming. The third part, a series of appendices, contains reference material, both for the JNI and for tools introduced in this book. There is also an appendix that compares the JNI with the old-style native method programming model introduced in JDK 1.0. The last appendix offers a brief discussion of native methods, applets and security issues.

The early chapters contain plenty of simple examples intended to highlight the essential features of the API. No attempt is made to place JNI function calls into large, complex examples that obscure their salient features.

The first part of the book takes a walk-before-you-run approach. After an overview of the JNI in Chapter 1. Chapter 2 presents a JNI version of the classic "Hello World" example. Chapter 3 then follows with examples of some of the more common JNI operations before plunging into the syntactical details of the JNI in Chapter 4. With all that work behind you, Chapter 5 through Chapter 8 provides detailed coverage of the remaining JNI functions.

The second half of the book deals with a series of general topics on using the JNI to integrate Java code with non-Java code. Chapter 9 presents an approach for mirroring existing C++ classes in Java. Chapter 10 introduces a tool for the automatic conversion of C structures into Java classes and an accompanying set of adapter functions for copying data between an instance of a C structure and a Java object. Chapter 11 starts with a collection of Java classes that provide a high-level interface to serial and parallel ports. Throughout this chapter the native code for targeting these classes for both POSIX and Win32 platforms is presented and discussed. The Java package used in this chapter is the portio package which is freely-available from Central Data (cd) as well as being included with the examples at the Prentice-Hall ftp site.

Chapter 12 and Chapter 13 deal with the Invocation API. Chapter 12 is a broad discussion of the mechanics involved with starting a Java Virtual Machine from a C/C++ application. Chapter 13 provides a very specific example of this facility, namely, starting the Java Virtual Machine (JVM) as an NT service. This chapter is for NT developers who wish to use Sun's JVM as an engine for their Java applications. Chapter 14 presents some approaches to debugging a Java application that includes native code. Finally, Chapter 15 is dedicated to changes and enhancements to the JNI in the JDK 1.2.

Whose Java?

This book covers the Java Native Interface API. To use the Java Native Interface API, you will need to run your Java code on a Java Virtual Machine that supports the JNI. The list of JVM vendors supporting the JNI is growing, but a sure bet is the JVM distributed by Sun. The Sun JVM, the Java Core classes and the JNI are available as part of the Java Development Kit (JDK). The JNI is supported starting in release 1.1 of the JDK. The JDK 1.

This site is available via anonymous ftp. The following sequence of steps illustrates how to download the Essential JNI examples. You type the bold.

```
% ftp ftp.prenhall
```

```
Connected to iq-ss3.prenhall.
```

```
220 iq-ss3 FTP server(UNIX(r) System V Release 4.0) ready.
User (iq-ss3.prenhall:(none)): anonymous
331 Guest login ok, send ident as password.
Password:
230 Guest login ok, access restrictions apply.
ftp> cd /pub/ptr/professional_computer_science.w-022/gordon/essential_jni
250 CWD command successful.
ftp> get ejni_ex.tar
200 PORT command successful.
150 ASCII data connection for ejni_examples.tar (38.11.232.6,1099) (19818 bytes).
226 ASCII Transfer complete.
20184 bytes received in 5.93 seconds (3.40 Kbytes/sec)
ftp> bye
```

Goodbye.

The ftp directory named in line 1 contains two examples files described in the following table.

File Name

Format

Size

ejni_ex.zip

zip

452KB

ejni_ex.tar

tar

1.55MB

These two files include makefiles for both Win32 and Solaris.

After downloading the examples, unzip or untar the file in a directory of your choice. Users of a UNIX zip utility should be careful to use the -a flag with unzip so that CR/NL sequences are properly converted.

Extracting the files will create a directory named ejni. Within the ejni directory will be another directory, examples.

The directory ejni/examples is known as the examples directory throughout this book.

Building the Examples

Before the examples can be built and run, some environment must be set up.

Setting Up the Environment

Two files are included with the example files that describe the appropriate environment settings. The file csh.env is provided for UNIX csh users. It is easily modified for other shells. The file win32.bat is provided for Win32 users. Both files must be modified to specify the location of your JDK and the location of your examples directory. Specifically, two environment variables are provided, JDK_HOME and EJNI_HOME, which point to these locations.

Additionally, these files define an appropriate setting for the CLASSPATH environment variable, adding the directories required to run the examples.

Finally, there are some platform-specific environment variables that need to be set.

Solaris Environment

The environment variable LD_LIBRARY_PATH is set within the csh.env file. The directory in which the

example native libraries are installed is added to your existing value.

The Solaris build assumes you are using Sun's compilers. Be certain to modify the CC settings in EJNI_HOME/Makefile.master to point to the proper installation directory, or comment it out and use your environment settings.

The examples in Chapter 9 require Rogue Wave's Tools.h++ class library. If you have this software, you will need to set the relevant macros in EJNI_HOME/Makefile.master to the appropriate values. If you are a UNIX user and don't have the Tools.h++ software, remove the target chap9 from the SUBDIRS macro in EJNI_HOME/Makefile. Your enjoyment of this example will be necessarily limited to viewing the source.

Win32 Environment

Win32 users can use the batch file win32.bat located in EJNI_HOME to set up their environment for building the examples. The file win32.bat batch assumes you are building using Microsoft's compiler, specifically Visual C++ and cl.exe. The appropriate environment for running cl from the command line is set using the vcvars32.bat batch file shipped with Visual C++. This file is located in your Visual C++ bin directory, typically Program Files\DevStudio\VC in. vcvars32.bat should be run before running win32.bat. If you get the message "Out of environment space," you will need to use the command command from the DOS prompt as shown below.

Increasing DOS Environment Space

```
C:> command /e:8192
```

This setting will only be in effect for the current DOS shell. To set this value permanently, add the following line to your CONFIG.SYS file.

```
shell=command /e:8192 /p
```

How To Build

Once the environment is set up you are ready to build. On Solaris, the following make commands, executed from the EJNI_HOME directory, build the Java class files and native libraries.

Building on Solaris

```
% make all; make install
```

On Win32 systems, the nmake command is used to build the example Java class files and native DLLs. This command must be executed from the EJNI_HOME directory.

Building on Win32

```
% nmake TARGET=all /f nmake.mak all
```

When the build is complete, all the class files live in \$EJNI_HOME/classes and all the native libraries live in \$EJNI_HOME/platform/lib where platform is either sparc or win32. Tilt the slashes appropriately for Win32 platforms.

A Word About the Examples

All of native coding examples are written in either C or C++. As you will learn, the syntax for calling a JNI function differs across the two languages.

In the first part of the book, through Chapter 8, all of the examples are written in C++. Therefore, whenever a reference is made to the first argument to a JNI function, it is understood not to count the JNIEnv pointer argument that is present in a C call of a JNI function.

The examples directory contains a directory for each chapter. Within the chapter directory is a directory for each example. Example 1 in Chapter 5 resides in ejni/examples/chap5/example1.

A README file in each example directory describes how to run the example since the book does not actually show all the examples running.

Some Conventions

All code listings, output, all variable, method and function names that appear in the text, command names,

directory names, and all URLs appear in courier.

User input appears in bold courier.

Special terms, upon their introduction, and parameters to be substituted for appear in italics.

Pseudo-code and labels within figures appear in helvetica.

Help From Fellow Travellers

There are at least two list-servers available for JNI programmers. The first deals directly with native method programming. On the `jnative-l` list you will find discussions of JNI as well as old-style JDK native method programming, Microsoft's Runtime Native Interface (RNI), Netscape's Java Runtime Environment (JRE) and, to a lesser extent, problems with incorporating native code with various browsers. To subscribe to this list, send an e-mail message to `majordomo@lists.teleport` with the text `subscribe jnative-l` in the body of the message.

It is not unusual for questions related to native method programming to appear on the Advanced Java list. To subscribe to this list, send mail to `majordomo@xcf.berkeley` with an empty subject and the following body:

```
subscribe advanced-java your_email_address
```

```
end
```

Do not use this latter list in judiciously. It is heavily patrolled by self-appointed experts who spend as much time telling people their question is not sufficiently "advanced" as they do providing answers to those question that they deem worthy of their consideration.

To stay in step with updates and enhancements to the JDK and the JNI, you may want to register with the Java Developer's Connection (JDC). To do so, follow the link off the JavaSoft home page (`java.sun`). As a member of the JDC, you are eligible for Early Access APIs. This is also the place to submit and track bugs. The JDC is a free service from JavaSoft but you do have to register.

A Word About Borrowed Words

Reading the quotes at the beginning of each chapter is a requirement for successful completion of this book. Too many late nights were spent finding them and too many phone calls were made acquiring permission to use them for you to ignore them.

Why include them at all? A guy who takes writing seriously has to do something to catch the eye of the reviewers at The New York Times Review of Books. Besides, it was fun trying to decorate the often bland minutia of a programming API with the distinguished dress of great ideas. I don't know if it worked, but I highly recommend all the quoted authors. Well, I can take or leave Nietzsche. As for Hobbes, I think he had it backwards. We are heading toward where he thinks we began.

From the Back Cover

Leverage platform-specific features without compromising binary compatibility!

With the Java Native Interface API, you get the best of both worlds: access to the native platform features you need to get the job done plus full binary compatibility across multiple Java Virtual Machines. And there is no better way to master the JNI than this fast-paced, example-rich guide, which focuses on proven techniques.

Walk step-by-step through writing Java and native code components, building libraries, loading and invoking native methods, and more. Discover powerful techniques for creating Java Virtual Machines from within any native application. Learn proven approaches to integrating Java and C++ code, converting C structures, and providing serial I/O from Java on both POSIX and Win32 platforms. Get comprehensive debugging help. There's even a complete, easy-to-use API reference!

With JNI, you can leverage Java's extraordinary capabilities without sacrificing proven legacy code.

Essential JNI: Java Native Interface gives you the insight, proven techniques, and real-world sample code you'll need to get results with JNI, plus:

- A complete, task-oriented reference to the entire Java Native Interface API
- Tools and strategies for integrating legacy code with new Java applications
- Extensive code examples and detailed debugging tips
- Clear explanations of JNI native types, signatures, references, and object and class functions
- Specific techniques for managing C++ code and converting C structures

Users Review

From reader reviews:

Bethany Christiansen:

In this 21st millennium, people become competitive in most way. By being competitive right now, people have do something to make these survives, being in the middle of often the crowded place and notice simply by surrounding. One thing that often many people have underestimated the item for a while is reading. Sure, by reading a book your ability to survive raise then having chance to stand than other is high. To suit your needs who want to start reading the book, we give you this Essential Jni: Java Native Interface (Essential Java) book as starter and daily reading e-book. Why, because this book is more than just a book.

Jodi Harper:

Hey guys, do you desires to finds a new book to learn? May be the book with the concept Essential Jni: Java Native Interface (Essential Java) suitable to you? The actual book was written by well-known writer in this era. The particular book untitled Essential Jni: Java Native Interface (Essential Java)is a single of several books that everyone read now. This specific book was inspired a lot of people in the world. When you read this guide you will enter the new dimensions that you ever know just before. The author explained their thought in the simple way, therefore all of people can easily to recognise the core of this reserve. This book will give you a lot of information about this world now. In order to see the represented of the world on this book.

Julie Chambers:

Typically the book Essential Jni: Java Native Interface (Essential Java) has a lot of knowledge on it. So when you read this book you can get a lot of help. The book was authored by the very famous author. Tom makes some research previous to write this book. That book very easy to read you can obtain the point easily after reading this article book.

Robert Jones:

Do you like reading a e-book? Confuse to looking for your best book? Or your book ended up being rare?

Why so many query for the book? But virtually any people feel that they enjoy intended for reading. Some people likes studying, not only science book but also novel and Essential Jni: Java Native Interface (Essential Java) or even others sources were given expertise for you. After you know how the truly great a book, you feel need to read more and more. Science e-book was created for teacher or students especially. Those ebooks are helping them to increase their knowledge. In some other case, beside science publication, any other book likes Essential Jni: Java Native Interface (Essential Java) to make your spare time a lot more colorful. Many types of book like this one.

Download and Read Online Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon #0JFMOR1GA4C

Read Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon for online ebook

Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon books to read online.

Online Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon ebook PDF download

Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon Doc

Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon Mobipocket

Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon EPub

0JFMOR1GA4C: Essential Jni: Java Native Interface (Essential Java) By Robert Gordon, Rob Gordon