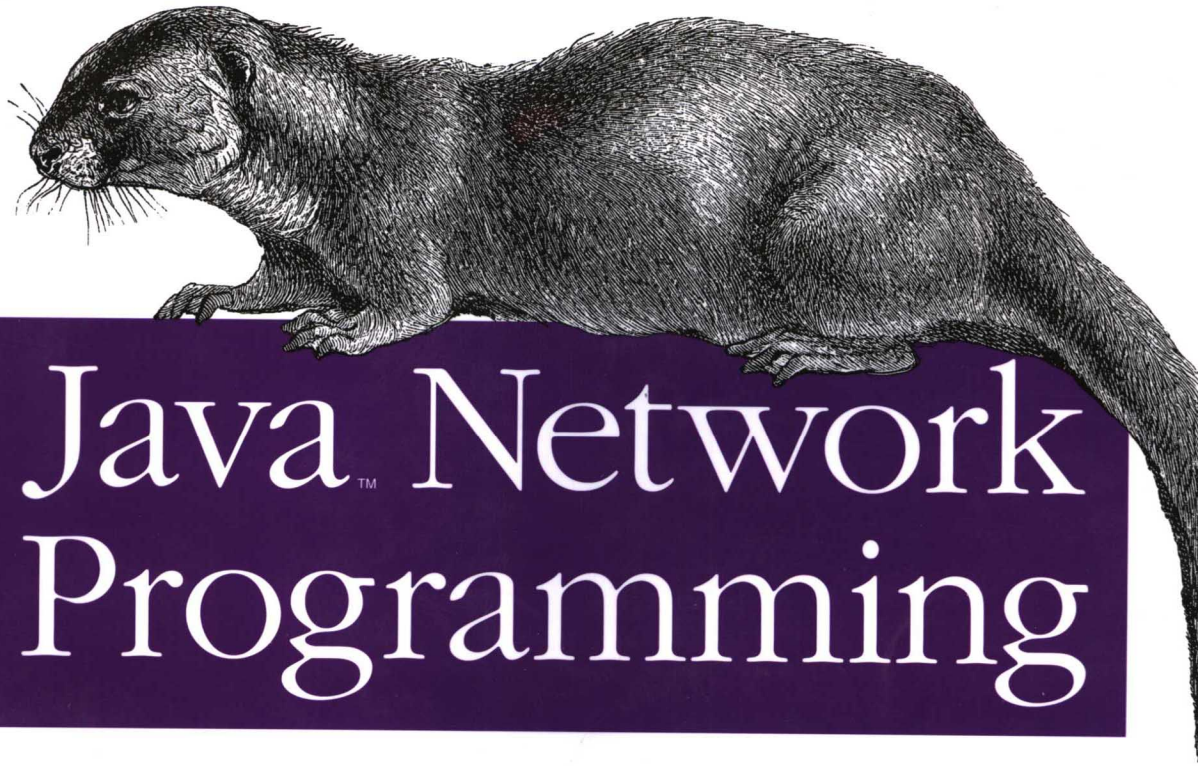
Java™网络编程(影印版)

3rd Edition
Covers Java 5.0



Java™ Network Programming

O'REILLY®
東南大學出版社

Elliotte Rusty Harold 著

第三版

Java™网络编程(影印版)

Java™ Network Programming

Elliotte Rusty Harold

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc.授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

Java™ 网络编程：第3版 /（美）哈罗德（Harold, R. E.）著 — 影印版 . — 南京：东南大学出版社，2005.6

书名原文：Java™ Network Programming, Third Edition

ISBN 7-5641-0042-7

Ⅰ.J... Ⅱ.哈... Ⅲ.JAVA 语言－程序设计－英文 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2005）第 047644 号

江苏省版权局著作权合同登记

图字：10-2005-081 号

书　　名 / Java™ 网络编程（影印版）
书　　号 / ISBN 7-5641-0042-7/TP · 3
责任编辑 / 张烨
封面设计 / Emma Colby，张健
出版发行 / 东南大学出版社
地　　址 / 南京四牌楼 2 号　　邮编 210096
印　　刷 / 扬中市印刷有限公司
开　　本 / 787 毫米 × 1092 毫米　16 开本　47.75 印张
版　　次 / 2005 年 6 月第 1 版　2005 年 6 月第 1 次印刷
印　　数 / 0001-2000 册
定　　价 / 98.00 元（册）

O'Reilly Media，Inc. 介绍

O'Reilly Media，Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的 The Whole Internet User's Guide & Catalog（被纽约公共图书馆评为 20 世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media，Inc. 一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media，Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media，Inc. 具有深厚的计算机专业背景，这使得 O'Reilly Media，Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media，Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media，Inc. 还有许多固定的作者群体 —— 他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media，Inc. 依靠他们及时地推出图书。因为 O'Reilly Media，Inc. 紧密地与计算机业界联系着，所以 O'Reilly Media，Inc. 知道市场上真正需要什么图书。

出版说明

随着计算机技术的成熟和广泛应用，人类正在步入一个技术迅猛发展的新时期。计算机技术的发展给人们的工业生产、商业活动和日常生活都带来了巨大的影响。然而，计算机领域的技术更新速度之快也是众所周知的，为了帮助国内技术人员在第一时间了解国外最新的技术，东南大学出版社和美国 O'Reilly Media，Inc. 达成协议，将陆续引进该公司的代表前沿技术或者在某专项领域享有盛名的著作，以影印版或者简体中文版的形式呈献给读者。其中，影印版书籍力求与国外图书"同步"出版，并且"原汁原味"展现给读者。

我们真诚地希望，所引进的书籍能对国内相关行业的技术人员、科研机构的研究人员和高校师生的学习和工作有所帮助，对国内计算机技术的发展有所促进，也衷心期望读者提出宝贵的意见和建议。

第一批影印图书共 10 本，涉及 Java，Unix/Linux，Python 等方面：

- 《核心 Java》(影印版)
- 《Jakarta Commons 经典实例》(影印版)
- 《Weblogic 权威指南》(影印版)
- 《Java 网络编程　第三版》(影印版)
- 《Linux 设备驱动程序　第三版》(影印版)
- 《LPI Linux 认证权威指南》(影印版)
- 《GNU Make 项目管理　第三版》(影印版)
- 《游戏开发中的人工智能》(影印版)
- 《学习 Python　第二版》(影印版)
- 《精通正则表达式　第二版》(影印版)

To Grandmama, a great grandmother.

Preface

Java's growth over the last 10 years has been nothing short of phenomenal. Given Java's rapid rise to prominence and the even more spectacular growth of the Internet, it's a little surprising that network programming in Java is still so mysterious to so many. It doesn't have to be. In fact, writing network programs in Java is quite simple, as this book will show. Readers with previous experience in network programming in a Unix, Windows, or Macintosh environment should be pleasantly surprised at how much easier it is to write equivalent programs in Java. The Java core API includes well-designed interfaces to most network features. Indeed, there is very little application-layer network software you can write in C or C++ that you can't write more easily in Java. *Java Network Programming, 3rd Edition* endeavors to show you how to take advantage of Java's network class library to quickly and easily write programs that accomplish many common networking tasks. Some of these include:

- Browsing the Web with HTTP
- Parsing and rendering HTML
- Sending email with SMTP
- Receiving email with POP and IMAP
- Writing multithreaded servers
- Installing new protocol and content handlers into browsers
- Encrypting communications for confidentiality, authentication, and guaranteed message integrity
- Designing GUI clients for network services
- Posting data to server-side programs
- Looking up hosts using DNS
- Downloading files with anonymous FTP
- Connecting sockets for low-level network communication
- Distributing applications across multiple systems with Remote Method Invocation

Java is the first language to provide such a powerful cross-platform network library, which handles all these diverse tasks. *Java Network Programming* exposes the power and sophistication of this library. This book's goal is to enable you to start using Java as a platform for serious network programming. To do so, this book provides a general background in network fundamentals, as well as detailed discussions of Java's facilities for writing network programs. You'll learn how to write Java programs that share data across the Internet for games, collaboration, software updates, file transfer, and more. You'll also get a behind-the-scenes look at HTTP, SMTP, TCP/IP, and the other protocols that support the Internet and the Web. When you finish this book, you'll have the knowledge and the tools to create the next generation of software that takes full advantage of the Internet.

About the Third Edition

In 1996, in the first chapter of the first edition of this book, I wrote extensively about the sort of dynamic, distributed network applications I thought Java would make possible. One of the most exciting parts of writing subsequent editions has been seeing virtually all of the applications I foretold come to pass. Programmers are using Java to query database servers, monitor web pages, control telescopes, manage multiplayer games, and more, all by using Java's native ability to access the Internet. Java in general and network programming in Java in particular has moved well beyond the hype stage and into the realm of real, working applications. Not all network software is yet written in Java, but it's not for a lack of trying. Efforts are well under way to subvert the existing infrastructure of C-based network clients and servers with pure Java replacements. Clients for newer protocols like Gnutella and Freenet are preferentially written in Java. It's unlikely that Java will replace C for all network programming in the near future. However, the mere fact that many people are willing to use web browsers, web servers, and more written in Java shows just how far we've come since 1996.

This book has come a long way, too. The third edition has one completely new chapter to describe the most significant development in network programming since readers and writers were introduced in Java 1.1. I refer of course to the new I/O APIs in the `java.nio` package. The ability to perform asynchronous, non-blocking I/O operations is critical for high-performance network applications, especially servers. It removes one of the last barriers to using Java for network servers. Many other chapters have been updated to take advantage of these new I/O APIs.

There've been lots of other small changes and updates throughout the `java.net` and supporting packages in Java 1.4 and 1.5, and these are covered here as well. New classes addressed in this edition include `CookieHandler`, `SocketAddress`, `Proxy`, `NetworkInterface`, and `URI`. IPv6 has become a reality, and is now covered extensively. Many other methods have been added to existing classes in the last two

releases of Java, and these are discussed in the relevant chapters. I've also rewritten large parts of the book to reflect changing fashions in Java programming in general and network programming in particular. Applets and CGI programs are emphasized much less. In their place, you'll find more generic discussion of remote code execution and server-side environments, however implemented.

Of course, the text has been cleaned up, too. There's only one completely new chapter here, but the 18 existing chapters have been extensively rewritten and expanded to bring them up-to-date with new developments as well as to make them clearer and more engaging. I hope you'll find this third edition an even stronger, longer-lived, more accurate, and more enjoyable tutorial and reference to network programming in Java than the last edition.

Organization of the Book

This book begins with three chapters that outline how networks and network programs work. Chapter 1, *Why Networked Java?*, is a gentle introduction to network programming in Java and the applications it makes possible. All readers should find something of interest in this chapter. It explores some of the unique programs that become feasible when networking is combined with Java. Chapter 2, *Basic Network Concepts*, and Chapter 3, *Basic Web Concepts*, explain in detail what a programmer needs to know about how the Internet and the Web work. Chapter 2 describes the protocols that underlie the Internet, such as TCP/IP and UDP/IP. Chapter 3 describes the standards that underlie the Web, such as HTTP, HTML, and REST. If you've done a lot of network programming in other languages on other platforms, you may be able to skip these two chapters.

The next two chapters throw some light on two parts of Java programming that are critical to almost all network programs but are often misunderstood and misused, I/O and threading. Chapter 4, *Streams*, explores Java's classic I/O models which, despite the new I/O APIs, aren't going away any time soon and are still the preferred means of handling input and output in most client applications. Understanding how Java handles I/O in the general case is a prerequisite for understanding the special case of how Java handles network I/O. Chapter 5, *Threads*, explores multithreading and synchronization, with a special emphasis on how they can be used for asynchronous I/O and network servers. Experienced Java programmers may be able to skim or skip these two chapters. However, Chapter 6, *Looking Up Internet Addresses*, is essential reading for everyone. It shows how Java programs interact with the domain name system through the `InetAddress` class, the one class that's needed by essentially all network programs. Once you've finished this chapter, it's possible to jump around in the book as your interests and needs dictate. There are, however, some interdependencies between specific chapters. Figure P-1 should allow you to map out possible paths through the book.

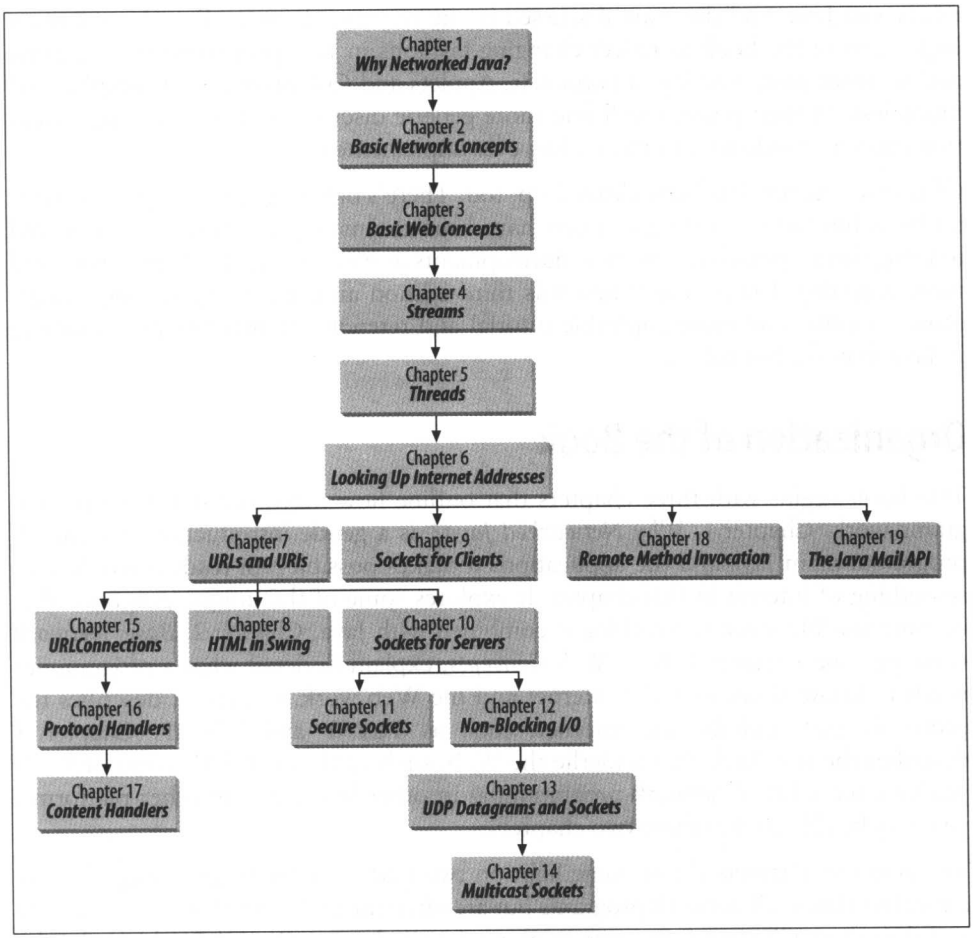


Figure P-1. Chapter prerequisites

Chapter 7, *URLs and URIs*, explores Java's `URL` class, a powerful abstraction for downloading information and files from network servers of many kinds. The `URL` class enables you to connect to and download files and documents from a network server without concerning yourself with the details of the protocol the server speaks. It lets you connect to an FTP server using the same code you use to talk to an HTTP server or to read a file on the local hard disk.

Once you've got an HTML file from a server, you're going to want to do something with it. Parsing and rendering HTML is one of the most difficult challenges network programmers can face. Chapter 8, *HTML in Swing*, introduces some little known classes for parsing and rendering HTML documents that take this burden off your shoulders and put it on Sun's.

Chapters 9 through 11 discuss Java's low-level socket classes for network access. Chapter 9, *Sockets for Clients*, introduces the Java sockets API and the `Socket` class in particular. It shows you how to write network clients that interact with TCP servers of all kinds including whois, finger, and HTTP. Chapter 10, *Sockets for Servers*, shows you how to use the `ServerSocket` class to write servers for these and other protocols in Java. Chapter 11, *Secure Sockets*, shows you how to protect your client server communications using the Secure Sockets Layer (SSL) and the Java Secure Sockets Extension (JSSE).

Chapter 12, *Non-Blocking I/O*, covers the new I/O APIs introduced in Java 1.4. These APIs were specifically designed for network servers. They enable a program to figure out whether a connection is ready before it tries to read from or write to the socket. This allows a single thread to manage many different connections simultaneously, thereby placing much less load on the virtual machine. The new I/O APIs don't help much for small servers or clients that don't open many simultaneous connections, but they provide huge performance boosts for high volume servers that want to transmit as much data as the network can handle as fast as the network can deliver it.

Chapter 13, *UDP Datagrams and Sockets*, introduces the User Datagram Protocol (UDP) and the associated `DatagramPacket` and `DatagramSocket` classes that provide fast, unreliable communication. Finally, Chapter 14, *Multicast Sockets*, shows you how to use UDP to communicate with multiple hosts at the same time. All the other classes that access the network from Java rely on the classes described in these five chapters.

Chapters 15 through 17 look more deeply at the infrastructure supporting the `URL` class. These chapters introduce protocol and content handlers, concepts unique to Java that make it possible to write dynamically extensible software that automatically understands new protocols and media types. Chapter 15, *URLConnections*, describes the class that serves as the engine for the `URL` class of Chapter 7. It shows you how to take advantage of this class through its public API. Chapter 16, *Protocol Handlers*, also focuses on the `URLConnection` class but from a different direction; it shows you how to subclass this class to create handlers for new protocols and URLs. Finally, Chapter 17, *Content Handlers*, explores Java's somewhat moribund mechanism for supporting new media types.

Chapters 18 and 19 introduce two unique higher-level APIs for network programs, Remote Method Invocation (RMI) and the JavaMail API. Chapter 18, *Remote Method Invocation*, introduces this powerful mechanism for writing distributed Java applications that run across multiple heterogeneous systems at the same time while communicating with straightforward method calls just like a nondistributed program. Chapter 19, *The JavaMail API*, acquaints you with this standard extension to Java, which offers an alternative to low-level sockets for talking to SMTP, POP,

IMAP, and other email servers. Both of these APIs provide distributed applications with less cumbersome alternatives to lower-level protocols.

Who You Are

This book assumes you are comfortable with the Java language and programming environment, in addition to object-oriented programming in general. This book does not attempt to be a basic language tutorial. You should be thoroughly familiar with the syntax of Java. You should have written simple applications and applets. You should also be comfortable with basic AWT and Swing programming. When you encounter a topic that requires a deeper understanding for network programming than is customary—for instance, threads and streams—I'll cover that topic as well, at least briefly.

You should also be an accomplished user of the Internet. I will assume you know how to FTP files and visit web sites. You should know what a URL is and how you locate one. You should know how to write simple HTML and be able to publish a home page that includes Java applets, although you do not need to be a super web designer.

However, this book doesn't assume that you have prior experience with network programming. You should find it a complete introduction to networking concepts and network application development. I don't assume that you have a few thousand networking acronyms (TCP, UDP, SMTP, etc.) at the tip of your tongue. You'll learn what you need to know about these here. It's certainly possible that you could use this book as a general introduction to network programming with a socket-like interface, and then go on to learn WSA (the Windows Socket Architecture) and figure out how to write network applications in C++. But it's not clear why you would want to: as I said earlier, Java lets you write very sophisticated applications with ease.

Java Versions

Java's network classes have changed a lot more slowly since Java 1.0 than other parts of the core API. In comparison to the AWT or I/O, there have been almost no changes and only a few additions. Of course, all network programs make extensive use of the I/O classes and many make heavy use of GUIs. This book is written with the assumption that you and your customers are using at least Java 1.1. In general, I use Java 1.1 features like readers and writers and the new event model freely without further explanation.

Java 2 is a bit more of a stretch. Although I wrote almost this entire book using Java 2, and although Java 2 has been available for most platforms for several years, no Java 2 runtime or development environment is yet available for MacOS 9. It is virtually certain that neither Apple nor Sun will ever port any version of Java 2 to MacOS 9.x or earlier, thus effectively locking out 60% of the current Mac-installed base from future

developments. This is not a good thing for a language that claims to be "write once, run anywhere." Furthermore, Microsoft's Java virtual machine supports Java 1.1 only and does not seem likely to improve in this respect for the foreseeable future. Thus, while I have not shied away from using Java 2–specific features where they seemed useful or convenient—for instance, the ASCII encoding for the `InputStreamReader` and the *keytool* program—I have been careful to point out my use of such features. Where 1.1 safe alternatives exist, they are noted. When a particular method or class is new in Java 1.2 or later, it is noted by a comment following its declaration like this:

```
public void setTimeToLive(int ttl) throws IOException // Java 1.2
```

To further muddy the waters, there are multiple versions of Java 2. At the time this book was completed, the current release was the "Java™ 2 SDK, Standard Edition, v 1.4.2_05". At least that's what it was called then. Sun seems to change names at the drop of a marketing consultant. In previous incarnations, this is what was simply known as the JDK. Sun also makes available the "Java™ 2 Platform, Enterprise Edition (J2EE™)" and "Java™ 2 Platform, Micro Edition (J2ME™)". The Enterprise Edition is a superset of the standard edition that adds features like the Java Naming and Directory Interface and the JavaMail API that provide high-level APIs for distributed applications. Most of these additional APIs are also available as extensions to the standard edition, and will be so treated here. The Micro Edition is a subset of the standard edition targeted at cell phones, set-top boxes, and other memory, CPU, and display-challenged devices. It removes a lot of the GUI APIs programmers have learned to associate with Java, although surprisingly it retains many of the basic networking and I/O classes discussed in this book. Finally, when this book was about half complete, Sun released a beta of the "Java™ 2 SDK, Standard Edition, v1.5". This added a few pieces to the networking API, but left most of the existing API untouched. Over the next few months Sun released several more betas of JDK 1.5. The finishing touches were placed on this book and all the code tested with JDK 1.5 beta 2. You shouldn't have any trouble using this book after 1.5 is released. With any luck at all, discrepancies between the final specification and what I discuss here will be quite minor.

To be honest, the most annoying problem with all these different versions and editions was not the rewriting they necessitated. It was figuring out how to identify them in the text. I simply refuse to write *Java™ 2 SDK, Standard Edition, v1.3* or even *Java 2 1.3* every time I want to point out a new feature in the latest release of Java. I normally simply refer to Java 1.1, Java 1.2, Java 1.3, Java 1.4, and Java 1.5. Overall, though, the networking API seems fairly stable. Java 1.1 through Java 1.3 are very similar, and there are a few only major additions in Java 1.4 and 1.5. Very little of the post-1.0 networking API has been deprecated.

About the Examples

Most methods and classes described in this book are illustrated with at least one complete working program, simple though it may be. In my experience, a complete working program is essential to showing the proper use of a method. Without a program, it is too easy to drop into jargon or to gloss over points about which the author may be unclear in his own mind. The Java API documentation itself often suffers from excessively terse descriptions of the method calls. In this book, I have tried to err on the side of providing too much explication rather than too little. If a point is obvious to you, feel free to skip over it. You do not need to type in and run every example in this book, but if a particular method does give you trouble, you are guaranteed to have at least one working example.

Each chapter includes at least one (and often several) more complex programs that demonstrate the classes and methods of that chapter in a more realistic setting. These often rely on Java features not discussed in this book. Indeed, in many of the programs, the networking components are only a small fraction of the source code and often the least difficult parts. Nonetheless, none of these programs could be written as easily in languages that didn't give networking the central position it occupies in Java. The apparent simplicity of the networked sections of the code reflects the extent to which networking has been made a core feature of Java, and not any triviality of the program itself. All example programs presented in this book are available online, often with corrections and additions. You can download the source code from *http://www.cafeaulait.org/books/jnp3/*.

This book assumes you are using Sun's Java Development Kit. I have tested all the examples on Linux and many on Windows and MacOS X. Almost all the examples given here *should* work on other platforms and with other compilers and virtual machines that support Java 1.2 (and most on Java 1.1, as well). The occasional examples that require Java 1.3, 1.4, or 1.5 are clearly noted.

Conventions Used in This Book

Body text is Times Roman, normal, like you're reading now.

A `monospaced typewriter font` is used for:

- Code examples and fragments
- Anything that might appear in a Java program, including keywords, operators, data types, method names, variable names, class names, and interface names
- Program output
- Tags that might appear in an HTML document

A **bold monospaced font** is used for:

- Command lines and options that should be typed verbatim on the screen

An *italicized* font is used for:

- New terms where they are defined
- Pathnames, filenames, and program names (however, if the program name is also the name of a Java class, it is given in a monospaced font, like other class names)
- Host and domain names (*java.oreilly.com*)
- URLs (*http://www.cafeaulait.org/slides/*)
- Titles of other chapters and books (*Java I/O*)

Significant code fragments and complete programs are generally placed into a separate paragraph, like this:

```
Socket s = new Socket("java.oreilly.com", 80);
if (!s.getTcpNoDelay()) s.setTcpNoDelay(true);
```

When code is presented as fragments rather than complete programs, the existence of the appropriate `import` statements should be inferred. For example, in the above code fragment you may assume that `java.net.Socket` was imported.

Some examples intermix user input with program output. In these cases, the user input will be displayed in bold, as in this example from Chapter 9:

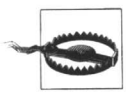
```
% telnet rama.poly.edu 7
Trying 128.238.10.212...
Connected to rama.poly.edu.
Escape character is '^]'.
This is a test
This is a test
This is another test
This is another test
9876543210
9876543210
^]
telnet> close
Connection closed.
```

The Java programming language is case-sensitive. `Java.net.socket` is not the same as `java.net.Socket`. Case-sensitive programming languages do not always allow authors to adhere to standard English grammar. Most of the time, it's possible to rewrite the sentence in such a way that the two do not conflict, and when possible I have endeavored to do so. However, on those rare occasions when there is simply no way around the problem, I have let standard English come up the loser. In keeping with this principle, when I want to refer to a class or an instance of a class in body text, I use the capitalization that you'd see in source code, generally an initial capital with internal capitalization—for example, `ServerSocket`.

Throughout this book, I use the British convention of placing punctuation inside quotation marks only when punctuation is part of the material quoted. Although I learned grammar under the American rules, the British system has always seemed far more logical to me, even more so than usual when one must quote source code where a missing or added comma, period, or semicolon can make the difference between code that compiles and code that doesn't.

Finally, although many of the examples used here are toy examples unlikely to be reused, a few of the classes I develop have real value. Please feel free to reuse them or any parts of them in your own code. No special permission is required. As far as I am concerned, they are in the public domain (although the same is most definitely not true of the explanatory text!). Such classes are placed somewhere in the `com.macfaq` package, generally mirroring the `java` package hierarchy. For instance, Chapter 4's `SafePrintWriter` class is in the `com.macfaq.io` package. When working with these classes, don't forget that the compiled *.class* files must reside in directories matching their package structure inside your class path, and that you'll have to import them in your own classes before you can use them. The book's web page at *http://www.cafeaulait.org/books/jnp3/* includes a *jar* file containing all these classes that can be installed in your class path.

Indicates a tip, suggestion, or general note.



Indicates a warning or caution.

Request for Comments

I enjoy hearing from readers, whether with general comments about this book, specific corrections, other topics you would like to see covered, or just war stories about your own network programming travails. You can reach me by sending email to *elharo@metalab.unc.edu*. Please realize, however, that I receive several hundred pieces of email a day and cannot personally respond to each one. For the best chances of getting a personal response, please identify yourself as a reader of this book. If you have a question about a particular program that isn't working as you expect, try to reduce it to the simplest case that reproduces the bug, preferably a single class, and paste the text of the entire program into the *body* of your email. Unsolicited attachments will be deleted unopened. And please, please send the message from the account you want me to reply to and make sure that your Reply-to address is properly set! There's nothing quite so frustrating as spending an hour or more

carefully researching the answer to an interesting question and composing a detailed response, only to have it bounce because my correspondent was sending from a public terminal and neglected to set the browser preferences to include their actual email address.

I also adhere to the old saying "If you like this book, tell your friends. If you don't like it, tell me." I'm especially interested in hearing about mistakes. This is my eighth book. I've yet to publish a perfect one, but I keep trying. As hard as I and the editors at O'Reilly worked on this book, I'm sure there are mistakes and typographical errors that we missed here somewhere. And I'm sure that at least one of them is a really embarrassing whopper of a problem. If you find a mistake or a typo, please let me know so I can correct it. I'll post it on the web page for this book at *http://www.cafeaulait.org/books/jnp3/* and on the O'Reilly web site at *http://www.oreilly.com/catalog/javanetwk/errata/*. Before reporting errors, please check one of those pages to see if I already know about it and have posted a fix. Any errors that are reported will be fixed in future printings.

Comments and Questions

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

There is a web page for this book, which lists errata, examples, and any additional information. You can access this page at:

http://www.oreilly.com/catalog/javanp3/

To comment on or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about books, conferences, software, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

http://www.oreilly.com

The author maintains a web site for the discussion of EJB and related distributed computing technologies at *http://www.jmiddleware.com*. jMiddleware.com provides news about this book as well as code tips, articles, and an extensive list of links to EJB resources.