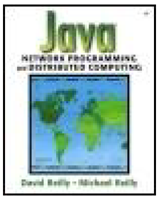
# Java™

## NETWORK PROGRAMMING and DISTRIBUTED COMPUTING

David Reilly • Michael Reilly



[Table of Contents](#)

**Java™ Network Programming and Distributed Computing**

By David Reilly, Michael Reilly

Publisher : Addison Wesley
Pub Date : March 25, 2002
ISBN : 0-201-71037-4
Pages : 496

*Java(TM) Network Programming and Distributed Computing* is an accessible introduction to the changing face of networking theory, Java(TM) technology, and the fundamental elements of the Java networking API. With the explosive growth of the Internet, Web applications, and Web services, the majority of today's programs and applications require some form of networking. Because it was created with extensive networking features, the Java programming language is uniquely suited for network programming and distributed computing.

Whether you are a Java devotee who needs a solid working knowledge of network programming or a network programmer needing to apply your existing skills to Java, this how-to guide is the one book you will want to keep close at hand. You will learn the basic concepts involved with networking and the practical application of the skills necessary to be an effective Java network programmer. An accelerated guide to networking API, Java(TM) Network Programming and Distributed Computing also serves as a comprehensive, example-rich reference.

You will learn to maximize the API structure through in-depth coverage of:

- The architecture of the Internet and TCP/IP
- Java's input/output system
- How to write to clients and servers using the User Datagram Protocol (UDP) and TCP
- The advantages of multi-threaded applications
- How to implement network protocols and see examples of client/server implementations
- HTTP and how to write server-side Java applications for the WebDistributed computing technologies such as Remote Method Invocation (RMI) and CORBA
- How to access e-mail using the extensive and powerful JavaMail(TM) API

This book's coverage of advanced topics such as input/output streaming and multi-threading allows even the most experienced Java developers to sharpen their skills. Java(TM) Network Programming and Distributed Computing will get you up-to-speed with network programming today; helping you employ innovative techniques in your own software development projects.

# Table of Content

## Copyright

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for special sales. For more information, please contact:

Pearson Education Corporate Sales Division

201 W. 103rd Street

Indianapolis, IN 46290

(800) 428-5331

corpsales@pearsoned.com

Visit Addison-Wesley on the Web: www.awl.com/cseng/

Library of Congress Control Number:

2002101206

For information on obtaining permission for use of material from this work, please submit a written request to:

Pearson Education, Inc.

Rights and Contracts Department

75 Arlington Street, Suite 300

Boston, MA 02116

Fax: (617) 848-7047

Text printed on recycled paper

12345678910—CRS—0605040302

First printing, March 2002

## Dedication

*To the memory of Countess Ada Lovelace, the world's first computer programmer, and to Myrtle Irene Daley, my beloved grandmother. A gracious thanks goes out to two former instructors, Mr. Terry Bell and Dr. Zheng da Wu, whose encouragement and faith in writing and networking, respectively, guided me to what I am today.*

*—David Reilly*

# PREFACE

Welcome to *Java Network Programming and Distributed Computing.* The goal of this book is to introduce and explain the basic concepts of networking and discuss the practical aspects of Java network programming.

This book will help readers get up to speed with network programming and employ the techniques learned in software development. If you've had some networking experience in another language and want to apply your existing skills to Java, you'll find the book to be an accelerated guide and a comprehensive reference to the networking API. This book does not require you to be a networking guru, however, as Chapters 1–4 provide a gentle introduction to networking theory, Java, and the most basic elements of the Java networking API. In later chapters, the Java API is covered in greater detail, with a discussion supplementing the documentation that Sun Microsystems provides as a reference.

## What You'll Learn

In this book, readers will learn how to write applications in Java that make use of network programming. The Java API provides many ways to communicate over the Internet, from sending packets and streams of data to employing higher-level application protocols such as HTTP and distributed computing mechanisms.

Along the way, you'll read about:

- How the Internet works, its architecture and the TCP/IP protocol stack
- The Java programming language, including a refresher course on topics such as exception handling
- Java's input/output system and how it works
- How to write clients and servers using the User Datagram Protocol (UDP) and the Transport Control Protocol (TCP)
- The advantages of multi-threaded applications, which allow network applications to perform multiple tasks concurrently
- How to implement network protocols, including examples of client/server implementations
- The HyperText Transfer Protocol (HTTP) and how to access the World Wide Web using Java

- How to write server-side Java applications for the WWW
- Distributed computing technologies including remote method invocation (RMI) and CORBA
- How to access e-mail using the extensive JavaMail API

## What You'll Need

A reasonable familiarity with Java programming is required to get the most out of this book. You'll need to be able to compile and run Java applications and to understand basic concepts such as classes, objects, and the Java API. However, you don't need to be an expert with respect to the more advanced topics covered herein, such as I/O streams and multi-threading. All examples use a text interface, so there's no need to have GUI experience.

You'll also need to install the Java SDK, available for free from Sun Microsystems (http://java.sun.com/j2se/). Java programmers will no doubt already have access to the SDK, but readers should be aware that some examples in this text will require JDK 1.1, and the advanced sections on servlets, RMI and CORBA, and JavaMail will require Java 2.

A minimal amount of additional software is required, and most of the tools for Java programming are available for free and downloadable via the WWW. Chapter 2 includes an overview of Java development tools, but readers can also use their existing code editor. Readers will be advised when examples feature additional Sun Microsystems software.

## Companion Web Site

As a companion to the material covered in this book, the book's Web site offers the source code in downloadable form (no need to wear out your fingers!), as well as a list of Frequently Asked Questions about Java Networking, links to networking resources, and additional information about the book. The site can be found at

http://www.davidreilly.com/jnpbook/.

## Contacting the Authors

We welcome feedback from readers, be it comments on specific chapters or sections or an evaluation of the book as a whole. In particular, reader input about whether topics were clearly conveyed and sufficiently comprehensive would be appreciated. While we'd love to receive only praise, honest opinions are valued (as well as suggestions about coverage of new networking topics).

Feel free to contact us directly. While we can't guarantee an individual reply, we'll do our best to respond to your query. Please send questions and feedback via e-mail to: jnpbook@davidreilly.com.

David Reilly and Michael Reilly
September 2001

# ACKNOWLEDGMENTS

This book would not have been possible without the assistance of our peer reviewers, who contributed greatly to improving its quality and allowing us to deliver a guide to Java network programming that is both clear and comprehensive. Our thanks go to Michael Brundage, Elisabeth Freeman, Bob Kitzberge, Lak Ming Lam, Ian Lance Taylor, and John J. Wegis.

We'd like to make special mention of two reviewers who contributed detailed reviews and offered insightful recommendations: Howard Lee Harkness and D. Jay Newman. Most of all, we would like to thank Amy Fong, whose thoroughness and invaluable suggestions, including questions that the inquisitive reader might have about TCP/IP and Java, helped shape the book that you are reading today.

We'd also like to thank our editorial team at Addison-Wesley, including Karen Gettman, whose initial encouragement and persistence convinced us to take on the project, Mary Hart, Marcy Barnes-Henrie, Melissa Dobson, and Emily Frey. Their support throughout the process of writing, editing, and preparing this book for publication is most heartily appreciated.

# Chapter 1. Networking Theory

This chapter provides an overview of the basic concepts of networking and discusses essential topics of networking theory. Readers experienced with networking may choose to skip over some of these preliminary sections, although a refresher course on basic networking concepts will be useful, as later chapters presume a knowledge of this theory on the part of the reader. A solid understanding of the relationship between the various protocols that make up the TCP/IP suite is required for network programming.

## 1.1 What Is a Network?

Put simply, a network is a collection of devices that share a common communication protocol and a common communication medium (such as network cables, dial-up connections, and wireless links). We use the term *devices* in this definition rather than *computers,* even though most people think of a network as being a collection of computers; certainly the basic concept of a network in most peoples' mind is of an assembly of network servers and desktop machines.

However, to say that networks are merely a collection of computers is to limit the range of hardware that can use them. For example, printers may be shared across a network, allowing more than one machine to gain access to their services. Other types of devices can also be connected to a network; these devices can provide access to information, or offer services that may be controlled remotely. Indeed, there is a growing movement toward connecting noncomputing devices to networks. While the technology is still evolving, we're moving toward a network-centric as opposed to a computing-centric model. Services and devices can be distributed across a network rather than being bound to individual machines. In the same way, users can move from machine to machine, logging on as if they were sitting at their own familiar terminal.

One fun and popular example from very early on in the history of networking is the soda machine connected to the Internet, allowing people around the world to see how many cans of a certain flavor of drink were available. While a trivial application, it served to demonstrate the power of networking devices. Indeed, as home networks become easier to use and more affordable, we may even see regular household appliances such as telephones, televisions, and home stereo systems connected to local networks or even to the Internet.

Network and software standards such as Sun's Jini already exist to help devices and hardware talk to each other over networks and to allow instant plug-and-play functionality. Devices and services can be added and removed from the network (as, for example, when you unplug your printer and take it to the next room) without the need for complex administration and configuration. It is anticipated that over the course of the next few years, users will become just as comfortable and familiar with network-centric computing as they are with the Internet.

In addition to devices that provide services are devices that keep the network going. Depending on the complexity of a network and its physical architecture, elements forming it may include network cards, routers, hubs, and gateways. These terms are defined below.

- *Network cards* are hardware devices added to a computer to allow it to talk to a network. The most common network card in use today is the Ethernet card. Network cards usually connect to a network cable, which is the link to the network and the medium through which data is transmitted. However, other media exist, such as dial-up connections through a phone line, and wireless links.
- *Routers* are machines that act as switches. These machines direct packets of data to the next "hop" in their journey across a network.