Developing Networked Applications

# Java Network Programming

O'REILLY®

Elliotte Rusty Harold

Anne Bernard Programming Third Edition

en constant, per al name there, inc. All again energy

#### THIRD EDITION

## Java<sup>®</sup> Network Programming



QUÀ TẶNG CỦA QUỸ CHÂU Á KHÔNG ĐƯỢC BÁN LẠI



C.C. An

## Elliotte Rusty Harold

#### O'REILLY\*

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

## **Table of Contents**

T.D.		
Prefa	ce	 XI
1.	Why Networked Java?	 1
	What Can a Network Program Do?	2
	Security	16
	But Wait! There's More!	18
2	Pasis Natural Concents	10
2.	Network	. 19
	The Lavers of a Network	21
	IP TCP and UDP	21
	The Internet	31
	The Client/Server Model	36
	Internet Standards	38
	internet standards	50
3.	Basic Web Concepts	 . 46
	URIs	46
	HTML, SGML, and XML	53
	HTTP	55
	MIME Media Types	59
	Server-Side Programs	63
01		
4.	Streams	 . 6/
	Output Streams	68
	Input Streams	72
	Filter Streams	76
	Readers and writers	91

5.	Threads		105
	Running Threads		107
	Returning Information from a Thread		111
	Synchronization		123
	Deadlock		129
	Thread Scheduling		130
	Thread Pools		144
6.	Looking Up Internet Addresses		150
	The InetAddress Class		152
	Inet4Address and Inet6Address		169
	The NetworkInterface Class		170
	Some Useful Programs		173
7.	URLs and URIs	Why Networked Java?	184
	The URL Class		184
	The URLEncoder and URLDecoder Classes		208
	The URI Class		213
	Proxies		222
	Communicating with Server-Side Programs T	hrough GET	225
	Accessing Password-Protected Sites	The Layers of a Network	229
8	HTML in Swing		237
0.	HTML on Components	The Internet	237
	IEditorPane		239
	Parsing HTMI		248
	Cookies		266
	COORIES		200
9.	Sockets for Clients		275
	Socket Basics		275
	Investigating Protocols with Telnet		276
	The Socket Class	MUMB Media Types	278
	Socket Exceptions		304
	Socket Addresses		305
	Examples	Streams	306
	Launpies		
10.	Sockets for Servers	input Streams	325
	The ServerSocket Class	Filter Streams	325
	Some Useful Servers		342

11.	Secure Sockets	363
	Secure Communications	364
	Creating Secure Client Sockets	367
	Methods of the SSLSocket Class	371
	Creating Secure Server Sockets	376
	Methods of the SSLServerSocket Class	381
12.	Non-Blocking I/O	384
	An Example Client	385
	An Example Server	389
	Buffers	395
	Channels	415
	Readiness Selection	420
12	UDP Datagrams and Eackats	172
15.	The LIDE Protocol	423
	The DatagramPacket Class	425
	The DatagramSocket Class	435
	Some Useful Applications	448
	DatagramChannel	461
	Dutugramonamier	
14.	Multicast Sockets	470
	What Is a Multicast Socket?	471
	Working with Multicast Sockets	479
	Two Simple Examples	487
620	and Amberedand	1
15.	URLConnections	493
	Opening URLConnections	494
	Reading Data from a Server	495
	Reading the Header	497
	Configuring the Connection	506
	Configuring the Client Request HTTP Header	515
	Writing Data to a Server	517
	Content Handlers	522
	The Object Methods	524
	Security Considerations for URLConnections	524
	Guessing MIME Content Types	525
	HttpURLConnection	528
	Caches	544
	JarURLConnection	549

16	Protocol Handlers	552
10.	What Is a Protocol Handler?	552
	The URLStreamHandler Class	556
	Writing a Protocol Handler	564
	More Protocol Handler Examples and Techniques	567
	The URLStreamHandlerFactory Interface	576
17.	Content Handlers	580
2	What Is a Content Handler?	582
	The ContentHandler Class	584
	The ContentHandlerFactory Interface	595
	A Content Handler for the FITS Image Format	598
	Readiness Selection	(10
18.	Remote Method Invocation	. 010
	What Is Remote Method Invocation?	610
	Implementation	610
	Loading Classes at Runtime	629
	The java.rmi Package	624
	The java.rmi.registry Package	635
	The java.rmi.server Package	033
		. 641
19.	Ine JavaMall API	642
	What is the Javaman All:	644
	Sending Email	654
	Receiving Mail	659
	Password Authentication	663
	Addresses	667
	The URLINAME Class	670
	The Message Class	682
	The Part Interface	692
	Multipart Messages and The Treatment	696
	MIME Messages	698
	Folders	742
Inde	av	/13
mue	Security Considerations for L'RLConnactions	

## 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.