

Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



Wrox Programmer to Programmer™



Professional

# XMPP

*Programming with JavaScript® and jQuery*

Jack Moffitt



Programmer to Programmer™

# Get more out of wrox.com

## Interact

---

Take an active role online by participating in our P2P forums @ [p2p.wrox.com](http://p2p.wrox.com)

## Wrox Online Library

---

Hundreds of our books are available online through [Books24x7.com](http://Books24x7.com)

## Wrox Blox

---

Download short informational pieces and code to keep you up to date and out of trouble!

## Join the Community

---

Sign up for our free monthly newsletter at [newsletter.wrox.com](http://newsletter.wrox.com)

## Browse

---

Ready for more Wrox? We have books and e-books available on .NET, SQL Server, Java, XML, Visual Basic, C#/C++, and much more!

## Contact Us.

We always like to get feedback from our readers. Have a book idea?

Need community support? Let us know by e-mailing [wrox-partnerwithus@wrox.com](mailto:wrox-partnerwithus@wrox.com)

# PROFESSIONAL XMPP PROGRAMMING WITH JAVASCRIPT AND JQUERY

---

INTRODUCTION.....	xix
<b>► PART I XMPP PROTOCOL AND ARCHITECTURE</b>	
<b>CHAPTER 1</b> Getting to Know XMPP.....	3
<b>CHAPTER 2</b> Designing XMPP Applications.....	23
<b>► PART II THE APPLICATIONS</b>	
<b>CHAPTER 3</b> Saying Hello: The First Application.....	39
<b>CHAPTER 4</b> Exploring the XMPP Protocol: A Debugging Console.....	63
<b>CHAPTER 5</b> Microblogging in Real Time: An Identica Client.....	87
<b>CHAPTER 6</b> Talking with Friends: One-on-One Chat.....	103
<b>CHAPTER 7</b> Exploring Services: Service Discovery and Browsing.....	145
<b>CHAPTER 8</b> Group Chatting: A Multi-User Chat Client.....	165
<b>CHAPTER 9</b> Publishing and Subscribing: A Shared Sketch Pad Introduction.....	203
<b>CHAPTER 10</b> Writing with Friends: A Collaborative Text Editor.....	251
<b>CHAPTER 11</b> Playing Games: Head to Head Tic-Tac-Toe.....	299
<b>► PART III ADVANCED TOPICS</b>	
<b>CHAPTER 12</b> Getting Attached: Bootstrapping BOSH.....	377
<b>CHAPTER 13</b> Deploying XMPP Applications.....	387
<b>CHAPTER 14</b> Writing Strophe Plug-ins.....	401
<b>APPENDIX A</b> Getting Started with jQuery.....	419
<b>APPENDIX B</b> Setting Up a BOSH Connection Manager.....	429
<b>INDEX.....</b>	<b>441</b>



PROFESSIONAL

# **XMPP Programming with JavaScript<sup>®</sup> and jQuery**



PROFESSIONAL

# XMPP Programming with JavaScript<sup>®</sup> and jQuery

Jack Moffitt



WILEY

Wiley Publishing, Inc.

# Professional XMPP Programming with JavaScript® and jQuery

Published by  
Wiley Publishing, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
[www.wiley.com](http://www.wiley.com)

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-54071-8

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

**Library of Congress Control Number:** 2009900000

**Trademarks:** Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. JavaScript is a registered trademark of Sun Microsystems, Inc. All other trademarks are the property of their respective owners. Wiley Publishing, Inc. is not associated with any product or vendor mentioned in this book.



*Dedicated to my wife Kimberly and our son  
Jasper, whose loves, hugs, and smiles make  
every day the best day ever.*

# CREDITS

**EXECUTIVE EDITOR**

Carol Long

**PROJECT EDITOR**

Ed Connor

**TECHNICAL EDITOR**

Dave Cridland

**PRODUCTION EDITOR**

Kathleen Wisor

**COPY EDITOR**

Kim Cofer

**EDITORIAL DIRECTOR**

Robyn B. Siesky

**EDITORIAL MANAGER**

Mary Beth Wakefield

**MARKETING MANAGER**

David Mayhew

**PRODUCTION MANAGER**

Tim Tate

**VICE PRESIDENT AND****EXECUTIVE GROUP PUBLISHER**

Richard Swadley

**VICE PRESIDENT AND EXECUTIVE PUBLISHER**

Barry Pruett

**ASSOCIATE PUBLISHER**

Jim Minatel

**PROJECT COORDINATOR, COVER**

Lynsey Stanford

**COMPOSITOR**

Craig Johnson, Happenstance Type-O-Rama

**PROOFREADER**

Carrie Hunter, Word One

**INDEXER**

Robert Swanson

**COVER DESIGNER**

Michael E. Trent

**COVER IMAGE**

© Punchstock/Glowimages

## ABOUT THE AUTHOR



**JACK MOFFITT** is a hacker and entrepreneur based in Albuquerque, New Mexico. He has founded several startups built on XMPP technology including Chesspark, a real-time, multi-user gaming platform, and Collecta, a real-time search engine for the Web. He has started and contributed to numerous XMPP related open source and free software projects including the Strophe XMPP client libraries, the Punjab XMPP connection manager, the Palaver multi-user chat component, the Speeqe group chat application. He also has served several terms on both the XSF Board of Directors and the XSF Council. Previous to his XMPP work, he created the Icecast streaming media server, managed the Ogg, Vorbis, and Theora codec projects, and co-founded the Xiph.org Foundation, a standards organization for royalty-free multimedia technologies for the Internet. He is passionate about free software and open source, open standards, and Internet technology. His favorite programming languages include JavaScript, Erlang, and Python. You can find him at <http://metajack.im>, blogging about start-ups and code, as @metajack on Twitter and Identica, or often spreading the word of XMPP at technology conferences.

# ACKNOWLEDGMENTS

**WE ALL STAND ON THE SHOULDERS OF GIANTS**, and I am fortunate to have stood on many friendly ones throughout my career and while writing this book. Thanks to Carol Long and Ed Connor for the encouragement, handholding, reminders, and patience that every author needs. Thanks also to Jason Salas who not only encouraged me on this project but made the appropriate introductions. Thanks also to Dave Cridland for his work ensuring the technical quality of this book and his tireless humor. I'm hugely indebted to Peter Saint-Andre, patron saint of XMPP, and the rest of the XMPP Standards Foundation members for their advice, criticism, and friendship over the years. Finally, the biggest thanks of all to my wife; not only did she encourage me in this project and put up with my long hours and absence, she also worked hard as my first reader and made many helpful suggestions to the text.

# CONTENTS

*INTRODUCTION*

*XIX*

## **PART I: XMPP PROTOCOL AND ARCHITECTURE**

<b>CHAPTER 1: GETTING TO KNOW XMPP</b>	<b>3</b>
<b>What Is XMPP?</b>	<b>4</b>
<b>A Brief History of XMPP</b>	<b>5</b>
<b>The XMPP Network</b>	<b>6</b>
Servers	6
Clients	7
Components	7
Plug-ins	8
<b>XMPP Addressing</b>	<b>8</b>
<b>XMPP Stanzas</b>	<b>9</b>
Common Attributes	10
Presence Stanzas	11
Message Stanzas	13
IQ Stanzas	15
Error Stanzas	17
<b>The Connection Life Cycle</b>	<b>18</b>
Connection	18
Stream Set Up	19
Authentication	20
Disconnection	20
<b>Summary</b>	<b>20</b>
<b>CHAPTER 2: DESIGNING XMPP APPLICATIONS</b>	<b>23</b>
<b>Learning from Others</b>	<b>24</b>
<b>XMPP versus HTTP</b>	<b>27</b>
Advantages of XMPP	28
Disadvantages of XMPP	29
<b>Bridging XMPP and the Web</b>	<b>30</b>
Long Polling	30
Managing the Connection	31
Making JavaScript Speak XMPP	32

<b>Making XMPP Applications</b>	<b>33</b>
The Browser Platform	33
Basic Infrastructure	33
Protocol Design	34
<b>Summary</b>	<b>36</b>

## **PART II: THE APPLICATIONS**

### **CHAPTER 3: SAYING HELLO: THE FIRST APPLICATION** **39**

---

<b>Application Preview</b>	<b>40</b>
<b>Hello Design</b>	<b>40</b>
<b>Preparing the Way</b>	<b>41</b>
jQuery and jQuery UI	42
Strophe	42
fXHR	43
XMPP Accounts	43
<b>Starting Your First Application</b>	<b>43</b>
User Interface	43
Application Code	45
<b>Making Connections</b>	<b>47</b>
The Connection Life Cycle	47
Creating a Connection	48
Connecting Hello	49
Running the Application	53
<b>Creating Stanzas</b>	<b>53</b>
Strophe Builders	54
Saying Hello	56
<b>Handling Events</b>	<b>57</b>
Adding and Removing Handlers	57
Stanza Matching	57
Stanza Handler Functions	58
Handling Responses in Hello	59
<b>More Hellos</b>	<b>62</b>
<b>Summary</b>	<b>62</b>

### **CHAPTER 4: EXPLORING THE XMPP PROTOCOL: A DEBUGGING CONSOLE** **63**

---

<b>Application Preview</b>	<b>64</b>
<b>Peek Design</b>	<b>64</b>

---

<b>Building the Console</b>	<b>65</b>
User Interface	66
Displaying Traffic	68
Making XML Pretty	71
Dealing with XML Input	74
Making Input Easier	76
<b>Exploring XMPP</b>	<b>81</b>
Controlling Presence	81
Probing Versions	81
Dealing with Errors	82
<b>Better Debugging</b>	<b>85</b>
<b>Summary</b>	<b>85</b>
<b>CHAPTER 5: MICROBLOGGING IN REAL TIME: AN IDENTICA CLIENT</b>	<b>87</b>
<hr/>	
<b>Application Preview</b>	<b>88</b>
<b>Arthur Design</b>	<b>88</b>
<b>Microblogging with Identica</b>	<b>89</b>
Creating Your Account	90
Turning on XMPP	90
<b>Building Arthur</b>	<b>90</b>
Getting Started	91
Receiving Messages	93
XHTML-IM	94
Adding XHTML-IM to Arthur	95
Sending Messages	96
<b>Offline Messages</b>	<b>97</b>
<b>Creating a Better Micrologger</b>	<b>101</b>
<b>Summary</b>	<b>102</b>
<b>CHAPTER 6: TALKING WITH FRIENDS: ONE-ON-ONE CHAT</b>	<b>103</b>
<hr/>	
<b>Application Preview</b>	<b>104</b>
<b>Gab Design</b>	<b>104</b>
Presence	105
Messages	105
Chat Area	105
Roster Area	106
<b>Making the Interface</b>	<b>106</b>

<b>Building the Roster</b>	<b>109</b>
Requesting Rosters	111
Handling IQs	113
Updating Presence Status	114
Adding New Contacts	116
Responding to Roster Changes	117
Dealing with Subscription Requests	119
<b>Building the Chats</b>	<b>122</b>
Working with Tabs	122
Creating New Chats	123
Sending Messages	124
<b>Best Practices for Instant Messaging</b>	<b>127</b>
Understanding Message Routing	127
Addressing Messages Better	127
<b>Adding Activity Notifications</b>	<b>130</b>
Understanding Chat States	130
Sending Notifications	131
Receiving Notifications	132
<b>Final Touches</b>	<b>133</b>
<b>Gabbing More</b>	<b>143</b>
<b>Summary</b>	<b>143</b>

---

## **CHAPTER 7: EXPLORING SERVICES: SERVICE DISCOVERY AND BROWSING**

---

**145**

<b>Application Preview</b>	<b>146</b>
<b>Dig Design</b>	<b>146</b>
<b>Finding Information</b>	<b>147</b>
Disco#info Queries	147
Disco#items Queries	148
Disco Nodes	149
<b>Creating Dig</b>	<b>149</b>
Initial Disco Queries	153
Browsing the Disco Tree	155
<b>Digging into Services</b>	<b>160</b>
Finding a Proxy Server	161
Discovering Features	162
Looking for a Chat	162
<b>Discovering More</b>	<b>163</b>
<b>Summary</b>	<b>163</b>



---

<b>CHAPTER 8: GROUP CHATTING: A MULTI-USER CHAT CLIENT</b>	<b>165</b>
<b>Application Preview</b>	<b>166</b>
<b>Groupie Design</b>	<b>167</b>
<b>Public Speaking</b>	<b>167</b>
Group Chat Services	167
Entering and Leaving a Room	168
Sending and Receiving Messages	171
Anonymity	171
Creating Rooms	172
Understanding Roles and Affiliations	173
<b>Building the Interface</b>	<b>175</b>
<b>Joining the Room</b>	<b>179</b>
<b>Dealing with Presence and Messages</b>	<b>183</b>
Handling Room Messages	183
Tracking Presence Changes	185
Chat History	186
Keeping It Private	187
Describing Actions	190
<b>Managing the Room</b>	<b>191</b>
Changing Topics	191
Dealing with Troublemakers	192
Recruiting Help	194
<b>Improving Groupie</b>	<b>201</b>
<b>Summary</b>	<b>201</b>
<b>CHAPTER 9: PUBLISHING AND SUBSCRIBING: A SHARED SKETCH PAD INTRODUCTION</b>	<b>203</b>
<b>SketchCast Preview</b>	<b>204</b>
<b>SketchCast Design</b>	<b>205</b>
Everything Is Pubsub	205
Presenter's Flow	205
Audience's Flow	206
<b>Filling Out Forms</b>	<b>206</b>
What Is The Data Forms Extension?	206
Form Elements, Fields, and Types	207
Standardized Form Fields	210

---

<b>Working with Pubsub Nodes</b>	<b>211</b>
Creating Nodes	211
Configuring Nodes	213
Pubsub Events	215
Publishing to a Node	215
Subscribing and Unsubscribing	216
Retrieving Subscriptions	218
Retrieving Items	219
Subscription Management	221
<b>Broadcasting Sketches Using Pubsub</b>	<b>222</b>
Building the Interface	222
Sketching with Canvas	225
Logging In and Making Nodes	228
Publishing and Receiving Sketch Events	234
<b>Summary</b>	<b>249</b>
<b>CHAPTER 10: WRITING WITH FRIENDS: A COLLABORATIVE TEXT EDITOR</b>	<b>251</b>
<hr/>	
<b>Application Preview</b>	<b>252</b>
<b>NetPad Design</b>	<b>252</b>
<b>Operational Transformation</b>	<b>253</b>
Basic Principles	253
Details of the Algorithm	255
Implementation	256
<b>Extending the XMPP Protocol</b>	<b>267</b>
Ignoring the Unknown	267
XML Namespaces	268
Extended Elements	268
Extended Attributes	270
Contributing Extensions	271
<b>Designing the Protocol</b>	<b>271</b>
Testing for Support	272
Requesting and Controlling Sessions	272
Editing Operations	273
<b>Building the Editor</b>	<b>274</b>
The Initial Skeleton	274
Starting Editing Sessions	278
Chatting About Work	284
Making Edits	287
<b>Expanding NetPad</b>	<b>298</b>
<b>Summary</b>	<b>298</b>

---

<b>CHAPTER 11: PLAYING GAMES: HEAD TO HEAD TIC-TAC-TOE</b>	<b>299</b>
<b>Application Preview</b>	<b>300</b>
<b>Toetem Design</b>	<b>301</b>
<b>Designing the Game Protocol</b>	<b>302</b>
Keeping Track of Users	303
Managing Players	304
Managing Games	305
Playing and Watching the Game	308
<b>Getting Started on Toetem</b>	<b>311</b>
<b>Implementing Sessions and the Waiting List</b>	<b>317</b>
Referee Version One	317
Toetem Client Version One	322
<b>Implementing Game Management</b>	<b>325</b>
Referee Version Two	325
Toetem Client Version Two	334
<b>Implementing the Game Logic</b>	<b>338</b>
The Tic-Tac-Toe Library	338
Referee Version Three	343
Toetem Client Version Three	345
<b>Making the Game More Fun</b>	<b>372</b>
<b>Summary</b>	<b>373</b>

### **PART III: ADVANCED TOPICS**

---

<b>CHAPTER 12: GETTING ATTACHED: BOOTSTRAPPING BOSH</b>	<b>377</b>
<b>Session Attachment</b>	<b>378</b>
The Mechanics of Sessions	378
Use Cases	379
<b>Automatic Logins with Session Attachment</b>	<b>380</b>
Creating the Django Project	381
<b>Summary</b>	<b>385</b>
<b>CHAPTER 13: DEPLOYING XMPP APPLICATIONS</b>	<b>387</b>
<b>Growing Horizontally</b>	<b>387</b>
Multiple Connection Managers	388
Clustering XMPP Servers	391
Spreading Out Components	392
Federating Internally	393
Becoming a Server	393

<b>Growing Vertically</b>	<b>394</b>
Reducing Latency	394
Minimizing XML Serialization	397
Optimizing DOM Operations	398
<b>Summary</b>	<b>399</b>
<b>CHAPTER 14: WRITING STROPHE PLUG-INS</b>	<b>401</b>
<hr/>	
<b>Using Plug-ins</b>	<b>402</b>
Loading Plug-ins	402
Accessing Plug-in Functionality	402
<b>Building Plug-ins</b>	<b>403</b>
<b>Creating a Roster Plug-in</b>	<b>404</b>
Storing Contacts	405
Getting and Maintaining the Roster	407
Manipulating the Roster	411
<b>Taking the Plug-in for a Spin</b>	<b>412</b>
Improving the Roster Plug-in	417
<b>Summary</b>	<b>417</b>
<b>Appendix A: Getting Started with jQuery</b>	<b>419</b>
<b>Appendix B: Setting Up a BOSH Connection Manager</b>	<b>429</b>
<b>INDEX</b>	<b>441</b>

# INTRODUCTION

**XMPP POWERS A WIDE RANGE OF APPLICATIONS** including instant messaging, multi-user chat, voice and video conferencing, collaborative spaces, real-time gaming, data synchronization, and even search. Although XMPP started its life as an open, standardized alternative to proprietary instant messaging systems like ICQ and AOL Instant Messenger, it has matured into an extremely robust protocol for all kinds of exciting creations.

Facebook uses XMPP technology as part of its chat system. Google uses XMPP to power Google Talk and its exciting new Google Wave protocol. Collecta has built a real-time search engine based extensively on XMPP's publish-subscribe system. Several web browsers are experimenting with XMPP as the basis of their synchronization and sharing systems. Dozens of other companies have XMPP-enabled their web applications to provide enhanced user experiences and real-time interaction.

The core of XMPP is the exchange of small, structured chunks of information. Like HTTP, XMPP is a client-server protocol, but it differs from HTTP by allowing either side to send data to the other asynchronously. XMPP connections are long lived, and data is pushed instead of pulled.

Because of XMPP's differences, it provides an excellent companion protocol to HTTP. XMPP-powered web applications are to AJAX what AJAX was to the static web site; they are the next level of interactivity and dynamism. Where JavaScript and dynamic HTML have brought desktop application features to the web browser, XMPP brings new communications possibilities to the Web.

XMPP has many common social web features built in, due to its instant messaging heritage. Contact lists and subscriptions create social graphs, presence updates help users keep track of who is doing what, and private messaging makes communication among users trivial. XMPP also has nearly 300 extensions, providing a broad and useful range of tools on which to build sophisticated applications. With only a handful of these, along with the core protocol, amazing things can be built

This book teaches you to harness the promise of XMPP in your own applications, enabling you to build applications that are social, collaborative, real time, or all of the above. You will develop a series of increasingly sophisticated XMPP applications, starting from "Hello, World!" and finishing with a collaborative text editor, a shared sketch pad, and a real-time, multi-player game. By the end, you will have all the tools you need to build the next generation of applications using XMPP or to add new real-time, push, or social features to your current applications.

## WHO THIS BOOK IS FOR

This book is written for developers interested in making XMPP applications. You need not have any previous experience with XMPP, although it will certainly be helpful if you do. The book starts from the assumption that you've heard great things about XMPP and are looking to dive right in.

The JavaScript language is used to develop all the applications in the book because it is an easy language to understand, is familiar to a large number of programmers, and comes on every computer with a web browser. Even though this book uses JavaScript, all the concepts and applications could be developed in any language; most of the “hard parts” are not related to the programming language, the libraries used, or the web browser. You do not need to be a JavaScript expert to understand and work with the code in this book.

It is assumed that you understand the basic front-end web technologies, CSS and HTML. If you’ve ever written a little HTML from scratch and changed a few CSS styling properties, you should be fine.

This book also makes use of two libraries, jQuery and Strophe. It is helpful if you have used jQuery before, but if you haven’t, a short primer is included in Appendix A. The Strophe library is explained fully as the applications are developed.

## WHAT THIS BOOK COVERS

The XMPP protocol and its extensions cover a lot of ground. This book focuses on the pieces of XMPP in wide use. The following topics receive much attention:

- XMPP’s instant messaging features like rosters, presence and subscriptions, and private chats
- XMPP stanzas, stanza errors, and client protocol syntax and semantics
- Extending XMPP stanzas
- Service discovery (XEP-0030)
- Data Forms (XEP-0004)
- Multi-User Chat (XEP-0045)
- Publish-Subscribe (XEP-0060)

Although these topics are all approached from the client side, almost all of it is equally applicable to XMPP bots or server components and plug-ins.

The book also covers XMPP programming related topics such as application design, event handling, and combining simple protocol elements into a greater whole. Along the way, a few web programming topics are also discussed such as the Canvas API.

XMPP is now more than 10 years old and quite mature. This book covers the 1.0 version of the core protocol. The XMPP protocol parts of this book should work unchanged in future versions of the protocol, just as HTTP 1.0 clients can easily communicate with HTTP 1.1 servers.

XMPP has many extensions and several of these are also covered. For the most part, the book concentrates on extensions that are in a stable, mature state. For each extension used, the document number is always given, and if in doubt, you can always check the latest version of the extension to see if it has been changed or superseded.

The book was written with the 1.3 series versions of jQuery and the 1.7 series versions of jQuery UI. These libraries generally remain backward compatible to a large degree. Version 1.0 of the Strophe library is used, but future 1.X versions should also work fine.

## HOW THIS BOOK IS STRUCTURED

This book is primarily organized as a walkthrough tutorial of a series of example XMPP applications. Each application increases in difficulty and teaches you one or more useful parts of the XMPP protocol and its extensions. These applications are stripped down for clarity, but they are examples of the kinds of applications XMPP developers create every day.

This book is divided into three parts.

The first part is an introduction to the XMPP protocol, its uses, and XMPP application design. Chapter 1 covers the use cases for XMPP, the history of the protocol, and its component parts. Chapter 2 explains when XMPP is a good choice for the job and goes into detail about how XMPP applications work, particularly for the Web.

The second part is the meat of the book and contains nine XMPP applications that solve a variety of problems. Each application is more complex than the last and builds on the concepts of the previous ones. Chapter 3 starts with a simple “Hello, World!” type example, and by Chapter 11 you build a real-time, multi-player game.

The last part covers a few advanced but important topics. Chapter 12 discusses attached sessions, a useful trick for security, optimization, and persistence. Chapter 13 goes into detail about how best to deploy and scale XMPP-based applications. Chapter 14 explains how to use Strophe’s plug-in system and how to create your own plug-ins.

## WHAT YOU NEED TO USE THIS BOOK

This book makes use of web technologies and therefore requires almost no special tools. You can use, build, and run the applications in this book on virtually any platform. The libraries needed for the applications are explained in Chapter 3, and most can be used without downloading any code.

You will need some way to serve web pages such as a local web server or a hosting account somewhere. If you don’t have these readily available, you can use the Tape program to serve the files; Tape is a simple web server and is explained in Appendix B. It is an unfortunate requirement of browser security policy that you can’t easily run these applications directly from your local file system.

You will need an XMPP account (or multiple accounts in some cases if you want to test the code by yourself) to run the applications. You can avail yourself of any of the public XMPP servers for this purpose, although you will need to ensure that the server has support for publish-subscribe and multi-user chat; most do. You can also download and run your own XMPP server instead, although this is not covered in the book.

Chapter 12 requires some server-side assistance. The example uses the Python programming language along with the Django framework to provide this. This chapter is an advanced topic and is not needed for the normal applications in the book.

## CONVENTIONS

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.



*Boxes like this one hold important, not-to-be forgotten information that is directly relevant to the surrounding text.*



*Notes, tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.*

As for styles in the text:

- We *highlight* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show file names, URLs, and code within the text like so: `persistence.properties`.
- We present code in two different ways:

We use a monofont type with no highlighting for most code examples.

**We use boldface highlighting to emphasize code that is of particularly importance in the present context.**

## SOURCE CODE

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at <http://www.wrox.com>. Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.





*Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-54071-8.*

Once you download the code, just decompress it with your favorite compression tool. Alternatively, you can go to the main Wrox code download page at <http://www.wrox.com/dynamic/books/download.aspx> to see the code available for this book and all other Wrox books.

## ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you may save another reader hours of frustration and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to <http://www.wrox.com> and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at [www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml).

If you don't spot "your" error on the Book Errata page, go to [www.wrox.com/contact/techsupport.shtml](http://www.wrox.com/contact/techsupport.shtml) and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

## P2P.WROX.COM

For author and peer discussion, join the P2P forums at [p2p.wrox.com](http://p2p.wrox.com). The forums are a web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At <http://p2p.wrox.com> you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to [p2p.wrox.com](http://p2p.wrox.com) and click the Register link.
2. Read the terms of use and click Agree.

3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.



*You can read messages in the forums without joining P2P but in order to post your own messages, you must join.*

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

# PART I

## XMPP Protocol and Architecture

---

- ▶ **CHAPTER 1:** Getting to Know XMPP
- ▶ **CHAPTER 2:** Designing XMPP Applications





# 1

## Getting to Know XMPP

### **WHAT'S IN THIS CHAPTER?**

---

- ▶ The history of XMPP
- ▶ XMPP networks and connections
- ▶ XMPP's three building block stanzas

The eXtensible Messaging and Presence Protocol (XMPP) is, at its most basic level, a protocol for moving small, structured pieces of data between two places. From this humble basis, it has been used to build large-scale instant messaging systems, Internet gaming platforms, search engines, collaboration spaces, and voice and video conferencing systems. More unique applications appear every day, further demonstrating how versatile and powerful XMPP can be.

XMPP is made of a few small building blocks, and on top of these primitives many larger constructions have been made. Within XMPP are systems for building publish-subscribe services, multi-user chat, form retrieval and processing, service discovery, real-time data transfer, privacy control, and remote procedure calls. Often, XMPP programmers create their own, unique constructions that are fitted exactly for the problem at hand.

Most social media constructs that have propelled web sites like Facebook, MySpace, and Twitter into the forefront are also baked into XMPP. Within XMPP, you'll find rosters full of contacts that create a social graph with directed or undirected edges. Presence notifications are sent automatically when contacts come online and go offline, and private and public messages are the bread and butter application of XMPP systems. Developers will sometimes choose XMPP as the underlying technology layer simply because it gives them many social features for free, leaving them to concentrate on the unique pieces of their application.

The possibilities are vast, but before you can begin, you need to know about XMPP's different pieces and how they fit together into a cohesive whole.

## WHAT IS XMPP?

XMPP, like all protocols, defines a format for moving data between two or more communicating entities. In XMPP's case, the entities are normally a client and a server, although it also allows for peer-to-peer communication between two servers or two clients. Many XMPP servers exist on the Internet, accessible to all, and form a federated network of interconnected systems.

Data exchanged over XMPP is in XML, giving the communication a rich, extensible structure. Many modern protocols forgo the bandwidth savings of a binary encoding for the more practical feature of being human readable and therefore easily debugged. XMPP's choice to piggyback on XML means that it can take advantage of the large amount of knowledge and supporting software for dealing with XML.

One major feature XMPP gets by using XML is XML's extensibility. It is extremely easy to add new features to the protocol that are both backward and forward compatible. This extensibility is put to great use in the more than 200 protocol extensions registered with the XMPP Standards Foundation and has provided developers with a rich and practically unlimited set of tools.

XML is known primarily as a document format, but in XMPP, XML data is organized as a pair of streams, one stream for each direction of communication. Each XML stream consists of an opening element, followed by XMPP stanzas and other top-level elements, and then a closing element. Each XMPP stanza is a first-level child element of the stream with all its descendent elements and attributes. At the end of an XMPP connection, the two streams form a pair of valid XML documents.

XMPP stanzas make up the core part of the protocol, and XMPP applications are concerned with sending and responding to various kinds of stanzas. Stanzas may contain information about other entities' availability on the network, personal messages similar to e-mail, or structured communication intended for computer processing. An example stanza is shown here:

```
<message to='elizabeth@longbourn.lit'
        from='darcy@pemberley.lit/dance'
        type='chat'>
  <body>What think you of books?</body>
</message>
```

In a typical client-server XMPP session, a stanza such as this one from Elizabeth to Mr. Darcy will travel from Elizabeth's client to her server. Her server will notice that it is addressed to an entity on a remote server and will establish an XMPP connection with the remote server and forward the message there. This communication between servers resembles the e-mail network, but unlike e-mail servers, XMPP servers always communicate directly with each other and not through intermediate servers.

This direct communication eliminates some common vectors for spam and unauthorized messages. This is just one of the many ways in which XMPP is designed for security. It also supports encrypted communications between endpoints through use of Transport Layer Security (TLS) and strong authentication mechanisms via Simple Authentication and Security Layers (SASL).

XMPP is designed for the exchange of small bits of information, not large blobs of binary data. XMPP can, however, be used to negotiate and set up out-of-band or in-band transports, which can move large blocks from point to point. For these kinds of transfers, XMPP functions as a signaling layer.