

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



FIFTH EDITION



Beginning

# **XML**

Joe Fawcett, Liam R. E. Quin, Danny Ayers



# BEGINNING XML

---

|   |       |
|---|-------|
| INTRODUCTION .....                                  | xxvii |
| <b>▶ PART I INTRODUCING XML</b>                     |       |
| CHAPTER 1 What Is XML? .....                        | 3     |
| CHAPTER 2 Well-Formed XML .....                     | 25    |
| CHAPTER 3 XML Namespaces .....                      | 53    |
| <b>▶ PART II VALIDATION</b>                         |       |
| CHAPTER 4 Document Type Definitions .....           | 83    |
| CHAPTER 5 XML Schemas .....                         | 117   |
| CHAPTER 6 RELAX NG and Schematron .....             | 175   |
| <b>▶ PART III PROCESSING</b>                        |       |
| CHAPTER 7 Extracting Data from XML .....            | 211   |
| CHAPTER 8 XSLT .....                                | 239   |
| <b>▶ PART IV DATABASES</b>                          |       |
| CHAPTER 9 XQuery .....                              | 307   |
| CHAPTER 10 XML and Databases .....                  | 341   |
| <b>▶ PART V PROGRAMMING</b>                         |       |
| CHAPTER 11 Event-Driven Programming .....           | 403   |
| CHAPTER 12 LINQ to XML .....                        | 451   |
| <b>▶ PART VI COMMUNICATION</b>                      |       |
| CHAPTER 13 RSS, Atom, and Content Syndication ..... | 485   |
| CHAPTER 14 Web Services .....                       | 539   |
| CHAPTER 15 SOAP and WSDL .....                      | 573   |
| CHAPTER 16 AJAX .....                               | 615   |

► **PART VII DISPLAY**

**CHAPTER 17** XHTML and HTML 5 ..... 649

**CHAPTER 18** Scalable Vector Graphics (SVG) ..... 689

► **PART VIII CASE STUDY**

**CHAPTER 19** Case Study: XML in Publishing ..... 727

**APPENDIX A** Answers to Exercises ..... 749

**APPENDIX B** XPath Functions ..... 773

**APPENDIX C** XML Schema Data Types ..... 797

**INDEX** ..... 811

BEGINNING

**XML**



BEGINNING  
**XML**

Joe Fawcett  
Liam R.E. Quin  
Danny Ayers



WILEY

John Wiley & Sons, Inc.

## Beginning XML

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

Copyright © 2012 by Joe Fawcett, Liam R.E. Quin, and Danny Ayers  
Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-16213-2  
ISBN: 978-1-118-22612-4 (ebk)  
ISBN: 978-1-118-23948-3 (ebk)  
ISBN: 978-1-118-26409-6 (ebk)

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 publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2012937910

**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. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

*I'd like to dedicate this book to my parents, especially to my mother Sheila who, unfortunately, will never be able to read this. I love you both.*

—JOE FAWCETT

*Dedicated to Yuri Rubinsky, without whom there would be no XML.*

—LIAM QUIN

*Dedicated to my mother, Mary (because this will amuse her no end).*

—DANNY AYERS



## ABOUT THE AUTHORS



**JOE FAWCETT** (<http://joe.fawcett.name>) has been writing software, on and off, for forty years. He was one of the first people to be awarded the accolade of Most Valuable Professional in XML by Microsoft. Joe is head of software development for Kaplan Financial UK in London, which specializes in training people in business and accountancy and has one of the leading accountancy e-learning systems in the UK. This is the third title for Wrox that he has written in addition to the previous editions of this book.



**LIAM QUIN** (<http://www.holoweb.net/~liam>) is in charge of the XML work at the World Wide Web Consortium (W3C). He has been involved with markup languages and text since the early 1980s, and was involved with XML from its inception. He has a background in computer science and digital typography, and also maintains a website dedicated to the love of books and illustrations at [www.fromoldbooks.org](http://www.fromoldbooks.org). He lives on an old farm near Milford, in rural Ontario, Canada.



**DANNY AYERS** (<http://dannayayers.com>) is an independent researcher and developer of Web technologies, primarily those related to linked data. He has been an XML enthusiast since its early days. His background is in electronic music, although this interest has taken a back seat since the inception of the Web. Offline, he's also an amateur woodcarver. Originally from the UK, he now lives in rural Tuscany with two dogs and two cats.

## ABOUT THE TECHNICAL EDITOR

**KAREN TEGTMEYER** is an independent consultant and software developer with more than 10 years of experience. She has worked in a variety of roles, including design, development, training, and architecture. She also is an Adjunct Computer Science Instructor at Des Moines Area Community College.



# CREDITS

**EXECUTIVE EDITOR**

Carol Long

**PROJECT EDITOR**

Victoria Swider

**TECHNICAL EDITOR**

Karen Tegtmeyer

**PRODUCTION EDITOR**

Kathleen Wisor

**COPY EDITOR**

Kim Cofer

**EDITORIAL MANAGER**

Mary Beth Wakefield

**FREELANCER EDITORIAL MANAGER**

Rosemarie Graham

**ASSOCIATE DIRECTOR OF MARKETING**

David Mayhew

**MARKETING MANAGER**

Ashley Zurcher

**BUSINESS MANAGER**

Amy Knies

**PRODUCTION MANAGER**

Tim Tate

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

Richard Swadley

**VICE PRESIDENT AND EXECUTIVE PUBLISHER**

Neil Edde

**ASSOCIATE PUBLISHER**

Jim Minatel

**PROJECT COORDINATOR, COVER**

Katie Crocker

**PROOFREADERS**

James Saturnio, Word One

Sara Eddleman-Clute, Word One

**INDEXER**

Johnna VanHoose Dinse

**COVER DESIGNER**

Ryan Sneed

**COVER IMAGE**

© Marcello Bortolino



# ACKNOWLEDGMENTS

**I'D LIKE TO HEARTILY ACKNOWLEDGE** the help of the editor Victoria Swider and the acquisitions editor Carol Long, who kept the project going when it looked as if it would never get finished. I'd like to thank the authors of the previous edition, especially Jeff Rafter and David Hunter, who let us build on their work when necessary. I'd also like to thank my wife Gillian and my children Persephone and Xavier for putting up with my absences and ill humor over the last year; I'll make it up to you, I promise.

—JOE FAWCETT

**THANKS** are due to my partner and to the pets for tolerating long and erratic hours, and of course to Alexander Chalmers, for creating the Dictionary of Biography in 1810.

—LIAM QUIN

**MANY THANKS** to Victoria, Carol, and the team for making everything work. Thanks too to Joe for providing the momentum behind this project and to Liam for keeping it going.

—DANNY AYERS



# CONTENTS

INTRODUCTION

XXVII

## PART I: INTRODUCING XML

### CHAPTER 1: WHAT IS XML? 3

---

#### Steps Leading up to XML: Data Representation and Markups 4

Binary Files 4

Text Files 5

A Brief History of Markup 6

#### The Birth of XML 6

#### More Advantages of XML 10

XML Rules 10

Hierarchical Data Representation 11

Interoperability 12

#### XML in Practice 13

Data Versus Document 13

XML Scenarios 14

XML Technologies 16

#### Summary 23

### CHAPTER 2: WELL-FORMED XML 25

---

#### What Does Well-Formed Mean? 26

#### Creating XML in a Text Editor 26

Forbidden Characters 26

XML Prolog 27

Creating Elements 30

Attributes 33

Element and Attribute Content 34

Processing Instructions 38

CDATA Sections 39

#### Advanced XML Parsing 40

XML Equivalence 41

Whitespace Handling 42

Error Handling 44

#### The XML Infoset 47

The Document Information Item 48

|  |           |
|--|-----------|
| Element Information Items                      | 48        |
| Attribute Information Items                    | 48        |
| Processing Instruction Information Items       | 48        |
| Character Information Item                     | 49        |
| Comment Information Item                       | 49        |
| Namespace Information Item                     | 49        |
| The Document Type Declaration Information Item | 49        |
| Unexpanded Entity Reference Information Item   | 49        |
| Unparsed Entity Information Item               | 49        |
| Notation Information Item                      | 49        |
| <b>Summary</b>                                 | <b>50</b> |

---

**CHAPTER 3: XML NAMESPACES** **53**

---

|   |           |
|---|-----------|
| <b>Defining Namespaces</b>                | <b>54</b> |
| <b>Why Do You Need Namespaces?</b>        | <b>54</b> |
| <b>How Do You Choose a Namespace?</b>     | <b>56</b> |
| URLs, URIs, and URNs                      | 56        |
| Creating Your First Namespace             | 57        |
| <b>How to Declare a Namespace</b>         | <b>58</b> |
| How Exactly Does Scope Work?              | 62        |
| Declaring More Than One Namespace         | 62        |
| Changing a Namespace Declaration          | 64        |
| <b>Namespace Usage in the Real World</b>  | <b>68</b> |
| XML Schema                                | 68        |
| Documents with Multiple Namespaces        | 68        |
| <b>When to Use and Not Use Namespaces</b> | <b>72</b> |
| When Namespaces are Needed                | 72        |
| When Namespaces Are Not Needed            | 73        |
| Versioning and Namespaces                 | 74        |
| <b>Common Namespaces</b>                  | <b>75</b> |
| The XML Namespace                         | 75        |
| The XMLNS Namespace                       | 76        |
| The XML Schema Namespace                  | 76        |
| The XSLT Namespace                        | 76        |
| The SOAP Namespaces                       | 77        |
| The WSDL Namespace                        | 77        |
| The Atom Namespace                        | 77        |
| The MathML Namespace                      | 77        |
| The Docbook Namespace                     | 78        |
| <b>Summary</b>                            | <b>78</b> |

**PART II: VALIDATION**

|  |            |
|--|------------|
| <b>CHAPTER 4: DOCUMENT TYPE DEFINITIONS</b>      | <b>83</b>  |
| <b>What Are Document Type Definitions?</b>       | <b>83</b>  |
| Working with DTDs                                | 84         |
| Using jEdit                                      | 84         |
| The Document Type Declaration in Detail          | 88         |
| Sharing DTDs                                     | 92         |
| <b>Anatomy of a DTD</b>                          | <b>93</b>  |
| Element Declarations                             | 93         |
| Attribute Declarations                           | 103        |
| Entity Declarations                              | 109        |
| <b>DTD Limitations</b>                           | <b>114</b> |
| <b>Summary</b>                                   | <b>114</b> |
| <b>CHAPTER 5: XML SCHEMAS</b>                    | <b>117</b> |
| <b>Benefits of XML Schemas</b>                   | <b>118</b> |
| XML Schemas Use XML Syntax                       | 118        |
| XML Schema Namespace Support                     | 118        |
| XML Schema Data Types                            | 119        |
| XML Schema Content Models                        | 119        |
| XML Schema Specifications                        | 119        |
| <b>XML Schemas in Practice</b>                   | <b>120</b> |
| <b>Defining XML Schemas</b>                      | <b>124</b> |
| <schema> Declarations                            | 124        |
| <element> Declarations                           | 128        |
| Mixed Content                                    | 138        |
| <group> Declarations                             | 139        |
| <attribute> Declarations                         | 141        |
| An XML Schema for Contacts                       | 141        |
| Data Types                                       | 148        |
| <simpleType> Declarations                        | 155        |
| <b>Creating a Schema from Multiple Documents</b> | <b>161</b> |
| <import> Declarations                            | 161        |
| <include> Declarations                           | 165        |
| <b>Documenting XML Schemas</b>                   | <b>167</b> |
| <b>XML Schema 1.1</b>                            | <b>170</b> |
| Relaxed Rules                                    | 170        |
| <assert>   | 171        |
| <b>Summary</b>                                   | <b>171</b> |

---

|   |            |
|---|------------|
| <b>CHAPTER 6: RELAX NG AND SCHEMATRON</b>           | <b>175</b> |
| <b>Why Do You Need More Ways of Validating XML?</b> | <b>176</b> |
| <b>Setting Up Your Environment</b>                  | <b>176</b> |
| <b>Using RELAX NG</b>                               | <b>177</b> |
| Understanding the Basics of RELAX NG                | 177        |
| Understanding RELAX NG's Compact Syntax             | 183        |
| Converting Between the Two RELAX NG Formats         | 185        |
| Constraining Content                                | 186        |
| Reusing Code in RELAX NG Schema                     | 189        |
| <b>Using Schematron</b>                             | <b>193</b> |
| Understanding the Basics of Schematron              | 193        |
| Choosing a Version of Schematron                    | 194        |
| Understanding the Basic Process                     | 194        |
| Writing Basic Rules in Schematron                   | 195        |
| Creating a Schematron Document                      | 196        |
| Adding More Information to Messages                 | 197        |
| Constraining Values in Schematron                   | 200        |
| Handling Co-Constraints in Schematron               | 202        |
| Using Schematron from Within XML Schema             | 203        |
| <b>Summary</b>                                      | <b>207</b> |

### **PART III: PROCESSING**

---

|  |            |
|--|------------|
| <b>CHAPTER 7: EXTRACTING DATA FROM XML</b>         | <b>211</b> |
| <b>Document Models: Representing XML in Memory</b> | <b>211</b> |
| Meet the Models: DOM, XDM, and PSVI                | 212        |
| A Sample DOM Tree                                  | 212        |
| DOM Node Types                                     | 213        |
| DOM Node Lists                                     | 214        |
| The Limitations of DOM                             | 215        |
| <b>The XPath Language</b>                          | <b>216</b> |
| XPath Basics                                       | 216        |
| XPath Predicates: The Full Story                   | 218        |
| XPath Steps and Axes                               | 220        |
| XPath Expressions                                  | 222        |
| Variables in XPath Expressions                     | 226        |
| New Expressions in XPath 2                         | 227        |
| XPath Functions                                    | 231        |

---

|  |            |
|--|------------|
| XPath Set Operations                                       | 234        |
| XPath and Namespaces                                       | 234        |
| <b>Summary</b>   | <b>236</b> |
| <b>CHAPTER 8: XSLT</b>                                     | <b>239</b> |
| <hr/>  |            |
| <b>What XSLT Is Used For</b>                               | <b>240</b> |
| XPath as a Declarative Language                            | 240        |
| How Is XSLT a Functional Language?                         | 242        |
| <b>Setting Up Your XSLT Development Environment</b>        | <b>242</b> |
| Setting Up Saxon for .NET                                  | 243        |
| Setting Up Saxon for Java                                  | 244        |
| <b>Foundational XSLT Elements</b>                          | <b>245</b> |
| The <xsl:stylesheet> Element                               | 246        |
| The <xsl:template> Element                                 | 247        |
| The <xsl:apply-templates> Element                          | 251        |
| The <xsl:value-of> Element                                 | 251        |
| The <xsl:for-each> Element                                 | 253        |
| Push-Processing versus Pull-Processing                     | 254        |
| The Role of XPath in XSLT                                  | 254        |
| Using Named Templates                                      | 256        |
| The <xsl:call-template> Element                            | 259        |
| The document() Function in XSLT                            | 260        |
| Conditional Logic  | 266        |
| The <xsl:param> element                                    | 270        |
| The <xsl:sort> Element                                     | 271        |
| <xsl:copy> and <xsl:copy-of> Elements                      | 273        |
| <b>Reusing Code in XSLT</b>                                | <b>276</b> |
| The <xsl:include> Element                                  | 276        |
| The <xsl:import> Element                                   | 279        |
| The <xsl:template> Mode Attribute                          | 280        |
| <b>Understanding Built-In Templates and Built-In Rules</b> | <b>282</b> |
| <b>Using XSLT 2.0</b>                                      | <b>284</b> |
| Understanding Data Types in XSLT 2.0                       | 285        |
| Creating User-Defined Functions                            | 285        |
| Creating Multiple Output Documents                         | 290        |
| Using the collection() Function                            | 291        |
| Grouping in XSLT 2.0                                       | 292        |
| Handling Non-XML Input with XSLT 2.0                       | 295        |
| <b>XSLT and XPath 3.0: What's Coming Next?</b>             | <b>302</b> |
| <b>Summary</b>   | <b>303</b> |

**PART IV: DATABASES**

|  |            |
|--|------------|
| <b>CHAPTER 9: XQUERY</b>                                     | <b>307</b> |
| <b>XQuery, XPath, and XSLT</b>                               | <b>308</b> |
| XQuery and XSLT  | 308        |
| XQuery and XPath   | 308        |
| <b>XQuery in Practice</b>                                    | <b>309</b> |
| Standalone XQuery Applications                               | 309        |
| Part of SQL  | 309        |
| Callable from Java or Other Languages                        | 309        |
| A Native-XML Server  | 310        |
| XQuery Anywhere  | 310        |
| <b>Building Blocks of XQuery</b>                             | <b>313</b> |
| FLWOR Expressions, Modules, and Functions                    | 313        |
| XQuery Expressions Do Not Have a Default Context             | 317        |
| <b>The Anatomy of a Query Expression</b>                     | <b>318</b> |
| The Version Declaration                                      | 318        |
| The Query Prolog   | 319        |
| The Query Body   | 325        |
| <b>Some Optional XQuery Features</b>                         | <b>332</b> |
| XQuery and XPath Full Text                                   | 332        |
| The XQuery Update Facility                                   | 333        |
| XQuery Scripting Extension                                   | 333        |
| <b>Coming in XQuery 3.0</b>                                  | <b>333</b> |
| Grouping and Windowing                                       | 334        |
| The count Clause   | 335        |
| Try and Catch  | 335        |
| switch Expressions   | 336        |
| Function Items and Higher Order Functions                    | 337        |
| JSON Features  | 338        |
| XQuery, Linked Data, and the Semantic Web                    | 338        |
| <b>Summary</b>   | <b>338</b> |
| <b>CHAPTER 10: XML AND DATABASES</b>                         | <b>341</b> |
| <b>Understanding Why Databases Need to Handle XML</b>        | <b>341</b> |
| <b>Analyzing which XML Features are Needed in a Database</b> | <b>343</b> |
| Retrieving Documents   | 343        |
| Retrieving Data from Documents                               | 343        |
| Updating XML Documents                                       | 344        |
| Displaying Relational Data as XML                            | 344        |
| Presenting XML as Relational Data                            | 344        |

|   |            |
|---|------------|
| <b>Using MySQL with XML</b>             | <b>345</b> |
| Installing MySQL                        | 345        |
| Adding Information in MySQL             | 345        |
| Querying MySQL                          | 348        |
| Updating XML in MySQL                   | 353        |
| Usability of XML in MySQL               | 353        |
| Client-Side XML Support                 | 354        |
| <b>Using SQL Server with XML</b>        | <b>355</b> |
| Installing SQL Server                   | 355        |
| Presenting Relational Data as XML       | 356        |
| Understanding the xml Data Type         | 371        |
| Creating Indexes with the xml Data Type | 374        |
| W3C XML Schema in SQL Server            | 384        |
| Dealing with Namespaced Documents       | 385        |
| <b>Using eXist with XML</b>             | <b>386</b> |
| Downloading and Installing eXist        | 387        |
| Interacting with eXist                  | 389        |
| <b>Summary</b>                          | <b>399</b> |

## PART V: PROGRAMMING

|   |            |
|---|------------|
| <b>CHAPTER 11: EVENT-DRIVEN PROGRAMMING</b> | <b>403</b> |
| <b>Understanding Sequential Processing</b>  | <b>404</b> |
| <b>Using SAX in Sequential Processing</b>   | <b>404</b> |
| Preparing to Run the Examples               | 405        |
| Receiving SAX Events                        | 406        |
| Handling Invalid Content                    | 417        |
| Using the DTDHandler Interface              | 427        |
| EntityResolver Interface                    | 428        |
| Understanding Features and Properties       | 429        |
| <b>Using XmlReader</b>                      | <b>434</b> |
| Using XmlReaderSettings                     | 441        |
| Controlling External Resources              | 447        |
| <b>Summary</b>                              | <b>448</b> |
| <b>CHAPTER 12: LINQ TO XML</b>              | <b>451</b> |
| <b>What Is LINQ?</b>                        | <b>451</b> |
| Why You Need LINQ to XML                    | 454        |
| Using LINQ to XML                           | 454        |

|   |            |
|---|------------|
| <b>Creating Documents</b>                             | <b>457</b> |
| Creating Documents with Namespaces                    | 459        |
| Creating Documents with Prefixed Namespaces           | 460        |
| <b>Extracting Data from an XML Document</b>           | <b>461</b> |
| <b>Modifying Documents</b>                            | <b>468</b> |
| Adding Content to a Document                          | 468        |
| Removing Content from a Document                      | 470        |
| Updating and Replacing Existing Content in a Document | 470        |
| <b>Transforming Documents</b>                         | <b>472</b> |
| <b>Using VB.NET XML Features</b>                      | <b>474</b> |
| Using VB.NET XML Literals                             | 474        |
| Understanding Axis Properties in VB.NET               | 478        |
| Managing Namespaces in VB.NET                         | 480        |
| <b>Summary</b>  | <b>480</b> |

## PART VI: COMMUNICATION

|   |            |
|---|------------|
| <b>CHAPTER 13: RSS, ATOM, AND CONTENT SYNDICATION</b> | <b>485</b> |
| <b>Syndication</b>                                    | <b>485</b> |
| XML Syndication                                       | 486        |
| Syndication Systems                                   | 488        |
| Format Anatomy  | 491        |
| <b>Working with News Feeds</b>                        | <b>501</b> |
| Newsreaders   | 501        |
| Data Quality  | 501        |
| <b>A Simple Aggregator</b>                            | <b>502</b> |
| Modeling Feeds  | 502        |
| Program Flow  | 505        |
| Implementation  | 505        |
| Extending the Aggregator                              | 521        |
| <b>Transforming RSS with XSLT</b>                     | <b>524</b> |
| Generating a Feed from Existing Data                  | 524        |
| Processing Feed Data for Display                      | 528        |
| Browser Processing                                    | 531        |
| Preprocessing Feed Data                               | 532        |
| Reviewing the Different Formats                       | 532        |
| <b>Useful Resources</b>                               | <b>534</b> |
| <b>Summary</b>  | <b>535</b> |

---

|   |            |
|---|------------|
| <b>CHAPTER 14: WEB SERVICES</b>           | <b>539</b> |
| <b>What Is an RPC?</b>                    | <b>539</b> |
| <b>RPC Protocols</b>                      | <b>541</b> |
| COM and DCOM                              | 541        |
| COBRA and IIOP                            | 542        |
| Java RMI                                  | 543        |
| <b>The New RPC Protocol: Web Services</b> | <b>543</b> |
| The Same Origin Policy                    | 544        |
| Understanding XML-RPC                     | 546        |
| Choosing a Network Transport              | 548        |
| Understanding REST Services               | 559        |
| <b>The Web Services Stack</b>             | <b>564</b> |
| SOAP                                      | 565        |
| WSDL                                      | 566        |
| UDDI                                      | 566        |
| Surrounding Specifications                | 567        |
| <b>Summary</b>                            | <b>569</b> |
| <b>CHAPTER 15: SOAP AND WSDL</b>          | <b>573</b> |
| <b>Laying the Groundwork</b>              | <b>574</b> |
| <b>The New RPC Protocol: SOAP</b>         | <b>574</b> |
| Comparing SOAP to REST                    | 579        |
| Basic SOAP Messages                       | 579        |
| More Complex SOAP Interactions            | 587        |
| <b>Defining Web Services: WSDL</b>        | <b>600</b> |
| <definitions>                             | 601        |
| <types>                                   | 601        |
| <messages>                                | 602        |
| <portTypes>                               | 602        |
| <binding>                                 | 603        |
| <service>                                 | 605        |
| Other Bindings                            | 609        |
| <b>Summary</b>                            | <b>612</b> |
| <b>CHAPTER 16: AJAX</b>                   | <b>615</b> |
| <b>AJAX Overview</b>                      | <b>615</b> |
| AJAX Provides Feedback                    | 616        |
| Loading Incomplete Data With AJAX         | 616        |
| AJAX Performs Asynchronous Operations     | 617        |

|   |            |
|---|------------|
| <b>Introduction to JavaScript</b>           | <b>617</b> |
| The Web Browser Console                     | 618        |
| Values, Expressions, and Variables          | 619        |
| Control Flow Statements                     | 621        |
| Properties, Objects, Functions and Classes  | 622        |
| <b>The XMLHttpRequest Function</b>          | <b>623</b> |
| <b>Using HTTP Methods with AJAX</b>         | <b>628</b> |
| <b>Accessibility Considerations</b>         | <b>629</b> |
| <b>The jQuery Library</b>                   | <b>630</b> |
| Learning jQuery                             | 631        |
| The Domain-Specific Language (DSL) Approach | 631        |
| jQuery Plug-ins and Add-On Libraries        | 632        |
| <b>JSON and AJAX</b>                        | <b>635</b> |
| JSON Example                                | 635        |
| JSON Syntax                                 | 636        |
| JSON and jQuery                             | 636        |
| JSONP and CORS                              | 637        |
| <b>The Web Server Back End</b>              | <b>637</b> |
| Sending Images and Other Non-Textual Data   | 638        |
| Performance                                 | 638        |
| The Server Logs Are Your Friend             | 639        |
| <b>A Larger Example</b>                     | <b>639</b> |
| <b>Summary</b>                              | <b>644</b> |

## **PART VII: DISPLAY**

|   |            |
|---|------------|
| <b>CHAPTER 17: XHTML AND HTML 5</b>             | <b>649</b> |
| <b>Background of SGML</b>                       | <b>650</b> |
| HTML and SGML                                   | 650        |
| XML and SGML                                    | 651        |
| <b>The Open Web Platform</b>                    | <b>651</b> |
| <b>Introduction to XHTML</b>                    | <b>652</b> |
| The XHTML <html> Element                        | 653        |
| The XHTML <head> Element                        | 654        |
| The XHTML <body> Element                        | 656        |
| More Advanced HTML Topics                       | 668        |
| <b>XHTML and HTML: Problems and Workarounds</b> | <b>669</b> |
| <b>Cascading Style Sheets (CSS)</b>             | <b>670</b> |
| CSS Levels and Versions                         | 670        |
| CSS at a Glance                                 | 671        |
| CSS Selectors                                   | 673        |
| CSS Properties                                  | 675        |

---

|   |            |
|---|------------|
| CSS Special Rules                                   | 680        |
| CSS and XML   | 681        |
| Separating Style and Markup: Unobtrusive CSS        | 682        |
| <b>Unobtrusive JavaScript</b>                       | <b>682</b> |
| <b>HTML 5</b>                                       | <b>683</b> |
| Benefits of HTML 5                                  | 683        |
| Caveats of HTML 5                                   | 683        |
| New Elements in HTML 5                              | 684        |
| <b>Summary</b>                                      | <b>686</b> |
| <b>CHAPTER 18: SCALABLE VECTOR GRAPHICS (SVG)</b>   | <b>689</b> |
| <hr/>   |            |
| <b>Scalable Vector Graphics and Bitmaps</b>         | <b>689</b> |
| Procedural Graphics                                 | 690        |
| Declarative Graphics                                | 690        |
| Bitmap Graphics                                     | 691        |
| Vector Images                                       | 692        |
| SVG Images  | 692        |
| <b>The SVG Graphics Model</b>                       | <b>694</b> |
| <b>SVG and CSS</b>                                  | <b>696</b> |
| <b>SVG Tools</b>                                    | <b>698</b> |
| <b>SVG Basic Built-in Shapes</b>                    | <b>700</b> |
| Rectangles  | 701        |
| Circles   | 702        |
| Ellipses  | 702        |
| Straight Lines                                      | 703        |
| Polylines and Polygons                              | 704        |
| SVG Paths   | 705        |
| <b>SVG Transforms and Groups</b>                    | <b>708</b> |
| Transforms  | 708        |
| Groups  | 709        |
| <b>SVG Definitions and Metadata</b>                 | <b>709</b> |
| The SVG <title> and <desc> Elements                 | 709        |
| The SVG <metadata> Element                          | 710        |
| The SVG <defs> Element and Reusable Content         | 711        |
| <b>Viewports and Coordinates</b>                    | <b>712</b> |
| <b>SVG Colors and Gradients</b>                     | <b>713</b> |
| <b>Including Bitmap Images in SVG</b>               | <b>715</b> |
| <b>SVG Text and Fonts</b>                           | <b>716</b> |
| <b>SVG Animation Four Ways</b>                      | <b>717</b> |
| Synchronized Multimedia Integration Language (SMIL) | 718        |
| Scripted Animation                                  | 719        |

|                                       |            |
|---------------------------------------|------------|
| CSS Animation                         | 719        |
| External Libraries                    | 720        |
| <b>SVG and HTML 5</b>                 | <b>720</b> |
| <b>SVG and Web Apps</b>               | <b>721</b> |
| <b>Making SVG with XQuery or XSLT</b> | <b>722</b> |
| <b>Resources</b>                      | <b>722</b> |
| <b>Summary</b>                        | <b>723</b> |

## **PART VIII: CASE STUDY**

### **CHAPTER 19: CASE STUDY: XML IN PUBLISHING** **727**

|   |            |
|---|------------|
| <b>Background</b>                             | <b>727</b> |
| <b>Project Introduction: Current Workflow</b> | <b>728</b> |
| <b>Introducing a New XML-Based Workflow</b>   | <b>728</b> |
| Consultations                                 | 729        |
| Documenting the Project                       | 729        |
| Prototyping                                   | 729        |
| <b>Creating a New Process</b>                 | <b>730</b> |
| Challenging Criteria                          | 730        |
| The New Workflow                              | 731        |
| Document Conversion and Technologies          | 731        |
| Costs and Benefits Analysis                   | 732        |
| Deployment                                    | 733        |
| <b>Some Technical Aspects</b>                 | <b>733</b> |
| XQuery and Modules                            | 734        |
| XInclude                                      | 734        |
| Equations and MathML                          | 734        |
| XProc: An XML Pipelining Language             | 737        |
| XForms, REST, and XQuery                      | 738        |
| Formatting to PDF with XSL-FO                 | 738        |
| XML Markup for Documentation                  | 741        |
| Markup for the Humanities: TEI                | 741        |
| <b>The Hoy Books Website</b>                  | <b>741</b> |
| <b>Summary</b>                                | <b>746</b> |

### **APPENDIX A: ANSWERS TO EXERCISES** **749**

### **APPENDIX B: XPATH FUNCTIONS** **773**

### **APPENDIX C: XML SCHEMA DATA TYPES** **797**

### **INDEX** **811**

# INTRODUCTION

**THIS IS THE FIFTH EDITION OF A BOOK** that has proven popular with professional developers and academic institutions. It strives to impart knowledge on a subject that at first was seen by some as just another fad, but that instead has come to maturity and is now often just taken for granted. Almost six years have passed since the previous edition — a veritable lifetime in IT terms. In reviewing the fourth edition for what should be kept, what should be updated, and what new material was needed, the current authors found that about three-quarters of the material was substantially out of date. XML has far more uses than five years ago, and there is also much more reliance on it under the covers. It is now no longer essential to be able to handcraft esoteric configuration files to get a web service up and running. It has also been found that, in some places, XML is not always the best fit. These situations and others, along with a complete overhaul of the content, form the basis for this newer version.

So, what is XML? XML stands for eXtensible Markup Language, which is a language that can be used to describe data in a meaningful way. Virtually anywhere there is a need to store data, especially where it may need to be consumed by more than one application, XML is a good place to start. It has gained a reputation for being a candidate where interoperability is important, either between two applications in different businesses or simply those within a company. Hundreds of standardized XML formats now exist, known as *schemas*, which have been agreed on by businesses to represent different types of data, from medical records to financial transactions to GPS coordinates representing a journey.

## WHO THIS BOOK IS FOR

This book aims to suit a fairly wide range of readers. Most developers have heard of XML but may have been a bit afraid of it. XML has a habit nowadays of being used behind the scenes, and it's only when things don't work as expected or when developers want to do something a little different, that users start to realize that they must *open the hood*. To those people we say: *fear no longer*. It should also suit the developer experienced in other fields who has never had a formal grounding in the subject. Finally, it can be used as reference when you need to try something out for the first time. Nearly all the technologies in the book have a Try It Out section associated with them that first gets you up and running with a simple example and then explains how to progress from there.

What you don't need for this book is any knowledge of markup languages in general. This is all covered in the first few chapters. It is expected that most of the readership will have some knowledge of and experience with web programming, but we've tried to spread our examples so that knowledge could include using the Microsoft stack, Java, or one of the other open source frameworks, such as PHP or Python.

And just in case you are worried about the *Beginning* part of the title, that's a Wrox conceit that applies more to the style of the book than to your level of experience. Many of the concepts covered, especially in later chapters, are from the real world and are far from the *Hello World* genre.

## WHAT THIS BOOK COVERS

This book aims to teach you all you need to know about XML — what it is, how it works, what technologies accompany it, and how you can make it work for you, from simple data transfer to a way to provide multi-channeled content. The book sets out to answer these fundamental questions:

- What is XML?
- How do you use XML?
- How does it work?
- What can you use it for?

The basic concepts of XML have remained unchanged since their launch, but the surrounding technologies have changed dramatically. This book gives a basic overview of each technology and how it arose, but the majority of the examples use the latest version available. The examples are also drawn from more than one platform, with Java and .NET sharing most of the stage. XML products have also evolved; at one time there were many free and commercial Extensible Stylesheet Language Transformation (XSLT) processors; for example, XSLT is used to manipulate XML, changing it from one structure to another, and is covered in Chapter 8, but since version 2 appeared the number has reduced considerably as the work needed to develop and maintain the software has risen.

## HOW THIS BOOK IS STRUCTURED

We've tried to arrange the subjects covered in this book to lead you along the path of novice to expert in as logical a manner as possible. The sections each cover a different area of expertise. Unless you're fairly knowledgeable about the basics, we suggest you read the introductory chapters in Part 1, although skimming through may well be enough for the savvier user. The other sections can then be read in order or can be targeted directly if they cover an area that you are particularly interested in. For example, when your boss suddenly tells you that your next release must offer an XQuery add-in, you can head straight to Chapter 9. A brief overview of the book is as follows:

- You begin by learning exactly what XML is and why people felt it was needed.
- We then take you through how to create XML and what rules need to be followed.
- Once you've mastered that, you move on to what a valid XML document is and how you can be sure that yours is one of them.
- Then you'll look at how you can manipulate XML documents to extract data and to transform them into other formats.
- Next you deal with storing XML in databases — the advantages and disadvantages and how to query them when they're there.
- You then look at other ways to extract data, especially those suitable to dealing with large documents.