Beginning XML
4th Edition

David Hunter,
Jeff Rafter,
Joe Fawcett,
Eric van der Vlist,
Danny Ayers,
Jon Duckett,
Andrew Watt, and
Linda McKinnon
Beginning XML
4th Edition

David Hunter,
Jeff Rafter,
Joe Fawcett,
Eric van der Vlist,
Danny Ayers,
Jon Duckett,
Andrew Watt, and
Linda McKinnon

Wiley Publishing, Inc.
I would like to thank God, for continuing to give me opportunities to do what I love; my church family, for giving me more support than I deserve; and Andrea, for giving me more support than anyone deserves.
I would also like to thank the editors, for their constant help.
Their dedication to the quality of this book was a major factor in its success.
—David

To Ali and Jude, for their loving patience.
—Jeff

To my two brothers, Peter and Stephen, who have both helped me in my life and career in their own ways, many thanks.
—Joe

To my wife, Catherine, and children, Deborah, David, Samuel, and Sarah, for their patience and support while I am busy writing books.
—Eric

To my late grandmother, Mona Cartledge, who once gave me a Commodore Pet.
—Danny
About the Authors

David Hunter is a Senior Technical Consultant for CGI, a full-service IT and business process services partner. Providing technical leadership and guidance for solving his clients’ business problems, he is a jack-of-all-trades and master of some. With a career that has included design, development, support, training, writing, and other roles, he has had extensive experience building scalable, reliable, enterprise-class applications. David loves to peek under the hood at any new technology that comes his way, and when one catches his fancy, he really gets his hands dirty. He loves nothing more than sharing these technologies with others.

Jeff Rafter is an independent consultant based in Redlands, California. His focus is on emerging technology and web standards, including XML and validation. He currently works with Baobab Health Partnership with a focus on improving world health.

Joe Fawcett (http://joe.fawcett.name) started programming in the 1970s and worked briefly in IT when leaving full-time education. He then pursued a more checkered career before returning to software development in 1994. In 2003 he was awarded the title of Microsoft Most Valuable Professional in XML for community contributions and technical expertise; he has subsequently been re-awarded every year since. Joe currently works in London and is head of software development for FTC Kaplan Ltd., a leading international provider of accountancy and business training.

Eric van der Vlist is an independent consultant and trainer. His domains of expertise include web development and XML technologies. He is the creator and main editor of XMLfr.org, the main site dedicated to XML technologies in French, the lead author of Professional Web 2.0 Programming, the author of the O’Reilly animal books XML Schema and RELAX NG and a member of the ISO DSDL (http://dsdl.org) working group focused on XML schema languages. He is based in Paris and can be reached at vdv@dyomedea.com, or meet him at one of the many conferences where he presents his projects.

Danny Ayers is a freelance developer and consultant specializing in cutting-edge web technologies. His blog (http://dannyayers.com) tends to feature material relating to the Semantic Web and/or cat photos.

Linda McKinnon has more than 10 years of experience as a successful trainer and network engineer, assisting both private and public enterprises in network architecture design, implementation, system administration, and RFP procurement. She is a renowned mentor and has published numerous Linux study guides for Wiley Press and Gearhead Press.
Acknowledgments

This book would not have been possible without the work of the many developers dedicated to improving the Web through standards. We would also like to thank the countless contributors to mailing lists, IRC channels, forums, and friends that have helped us through the difficult corners of the specifications and technologies presented in this book.

Thanks to Nicholas C. Zakas for his ideas and assistance in implementing the AutoSuggest Control. Many thanks to Phillip Pearson, who runs TopicExchange.com. He provided much-needed technical support that otherwise would have meant rewriting most of Chapter 14. We would also like to thank Jim Ley and Doug Schepers for their assistance on the case study and Chapter 19. Special thanks to our lead editor, Sara Shlaer, for her gentle and not so gentle persuasive powers and attention to detail; to editor Lisa Thibault, for her thoughtful assistance; and to Phred Menyhert, for a rigorous technical edit. Many thanks to our acquisitions editor, Jim Minatel, who has shepherded this book through many incarnations.
# Contents

<table>
<thead>
<tr>
<th>Acknowledgments</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>xxvii</td>
</tr>
</tbody>
</table>

## Part I: Introduction

### Chapter 1: What Is XML?

- Of Data, Files, and Text
  - Binary Files
  - Text Files
  - A Brief History of Markup
- So What Is XML?
  - What Does XML Buy Us?
  - HTML and XML: Apples and Red Delicious Apples
  - Hierarchies of Information
  - What’s a Document Type?
  - No, Really — What’s a Document Type?
- Origin of the XML Standards
  - What Is the World Wide Web Consortium?
  - Components of XML
- Where XML Can Be Used, and What You Can Use It For
  - Reducing Server Load
  - Website Content
  - Distributed Computing
  - e-Commerce
- Summary
- Exercise Questions
  - Question 1
  - Question 2

### Chapter 2: Well-Formed XML

- Parsing XML
- Tags and Text and Elements, Oh My!
  - Rules for Elements
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>39</td>
</tr>
<tr>
<td>When to Use Attributes</td>
<td>43</td>
</tr>
<tr>
<td>Comments</td>
<td>45</td>
</tr>
<tr>
<td>Empty Elements</td>
<td>49</td>
</tr>
<tr>
<td>XML Declarations</td>
<td>50</td>
</tr>
<tr>
<td>Version</td>
<td>51</td>
</tr>
<tr>
<td>Encoding</td>
<td>51</td>
</tr>
<tr>
<td>Standalone</td>
<td>53</td>
</tr>
<tr>
<td>Processing Instructions</td>
<td>56</td>
</tr>
<tr>
<td>Illegal PCDATA Characters</td>
<td>59</td>
</tr>
<tr>
<td>Escaping Characters</td>
<td>60</td>
</tr>
<tr>
<td>CDATA Sections</td>
<td>61</td>
</tr>
<tr>
<td>Errors in XML</td>
<td>64</td>
</tr>
<tr>
<td>Summary</td>
<td>64</td>
</tr>
<tr>
<td>Exercise Questions</td>
<td>65</td>
</tr>
<tr>
<td>Question 1</td>
<td>65</td>
</tr>
<tr>
<td>Question 2</td>
<td>65</td>
</tr>
<tr>
<td>Chapter 3: XML Namespaces</td>
<td>67</td>
</tr>
<tr>
<td>Why We Need Namespaces</td>
<td>67</td>
</tr>
<tr>
<td>Using Prefixes</td>
<td>69</td>
</tr>
<tr>
<td>Why Doesn’t XML Just Use These Prefixes?</td>
<td>70</td>
</tr>
<tr>
<td>How XML Namespaces Work</td>
<td>72</td>
</tr>
<tr>
<td>Default Namespaces</td>
<td>75</td>
</tr>
<tr>
<td>Do Different Notations Make Any Difference?</td>
<td>81</td>
</tr>
<tr>
<td>Namespaces and Attributes</td>
<td>83</td>
</tr>
<tr>
<td>Understanding URIs</td>
<td>86</td>
</tr>
<tr>
<td>URLs</td>
<td>86</td>
</tr>
<tr>
<td>URNs</td>
<td>87</td>
</tr>
<tr>
<td>Why Use URLs for Namespaces, Not URNs?</td>
<td>87</td>
</tr>
<tr>
<td>What Do Namespace URIs Really Mean?</td>
<td>88</td>
</tr>
<tr>
<td>RDDL</td>
<td>89</td>
</tr>
<tr>
<td>When to Use Namespaces</td>
<td>89</td>
</tr>
<tr>
<td>Summary</td>
<td>90</td>
</tr>
<tr>
<td>Exercise Questions</td>
<td>91</td>
</tr>
<tr>
<td>Question 1</td>
<td>91</td>
</tr>
<tr>
<td>Question 2</td>
<td>91</td>
</tr>
<tr>
<td>Question 3</td>
<td>91</td>
</tr>
</tbody>
</table>
Part II: Validation  

Chapter 4: Document Type Definitions  

Running the Samples  
Preparation of the Ground  
The Document Type Declaration  

Sharing Vocabularies  
Anatomy of a DTD  
Element Declarations  
Attribute Declarations  
Entities  

Developing DTDs  

DTD Limitations  

XML Namespaces  
Data Typing  
Limited Content Model Descriptions  

Summary  

Exercise Questions  

Question 1  
Question 2  
Question 3  

Chapter 5: XML Schemas  

Benefits of XML Schemas  
XML Schemas Use XML Syntax  
XML Schema Namespace Support  
XML Schema Data Types  
XML Schema Content Models  

Do We Still Need DTDs?  

XML Schemas  
The XML Schema Document  
Running the Samples  
<schema> Declarations  
<element> Declarations  
<complexType> Declarations  
<group> Declarations  
Content Models  
<attribute> Declarations  
<attributeGroup> Declarations
Contents

Creating Elements with Simple Content and Attributes 185
Datatypes 186
<simpleType> Declarations 193

Creating a Schema from Multiple Documents 200
<import> Declarations 200
<include> Declarations 204

Documenting XML Schemas 206
Comments 206
Attributes from Other Namespaces 207
Annotations 208

Summary 209
Exercise Questions 210

Chapter 6: RELAX NG 211

XML and Compact Syntaxes 212
Running the Samples 212

RELAX NG Patterns 213
Element, Attribute, and Text Patterns 213

Combining and Reusing Patterns and Grammars 227
Named Patterns 227
Combining Named Pattern Definitions 230
Schema Modularization Using the include Directive 231
Redefining Included Named Patterns 232
Removing Patterns with the notAllowed Pattern 233
Extensions and Restrictions 234
Nested Grammars 235

Additional RELAX NG Features 236
Namespaces 236
Name-Classes 237
Datatypes 241
List Patterns 243
Comments and Divisions 244

Useful Resources 245
Summary 245
Exercise Questions 245

Question 1 246
Question 2 246
# Contents

## Part III: Processing

### Chapter 7: XPath

- **Ways of Looking at an XML Document**
- **Modeling XML Documents**
- **Visualizing XPath**
  - Understanding Context
  - What Is a Node?
  - XPath 1.0 Types
- **Abbreviated and Unabbreviated Syntax**
- **XPath 1.0 Axes**
  - Child Axis
  - attribute Axis
  - ancestor Axis
  - ancestor-or-self Axis
  - descendant Axis
  - descendant-or-self Axis
  - following Axis
  - following-sibling Axis
  - namespace Axis
  - parent Axis
  - preceding Axis
  - preceding-sibling Axis
  - self Axis
- **XPath 1.0 Functions**
  - Boolean Functions
  - Node-Set Functions
  - Numeric Functions
  - String Functions
- **Predicates**
- **Structure of XPath Expressions**
- **XPath 2.0**
  - Revised XPath Data Model
  - W3C XML Schema Data Types
  - Additional XPath 2.0 Functions
  - XPath 2.0 Features
- **Summary**
- **Exercise Questions**
  - Question 1
  - Question 2
## Chapter 8: XSLT

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Is XSLT?</td>
<td>287</td>
</tr>
<tr>
<td>Restructuring XML</td>
<td>288</td>
</tr>
<tr>
<td>Presenting XML Content</td>
<td>288</td>
</tr>
<tr>
<td>How an XSLT Processor Works</td>
<td>288</td>
</tr>
<tr>
<td>Running the Examples</td>
<td>289</td>
</tr>
<tr>
<td>Introducing the Saxon XSLT Processor</td>
<td>289</td>
</tr>
<tr>
<td>Installing the Saxon XSLT Processor</td>
<td>290</td>
</tr>
<tr>
<td>Procedural versus Declarative Programming</td>
<td>292</td>
</tr>
<tr>
<td>Procedural Programming</td>
<td>292</td>
</tr>
<tr>
<td>Declarative Programming</td>
<td>292</td>
</tr>
<tr>
<td>Foundational XSLT Elements</td>
<td>293</td>
</tr>
<tr>
<td>The <a href="">xsl:stylesheet</a> Element</td>
<td>295</td>
</tr>
<tr>
<td>The <a href="">xsl:template</a> Element</td>
<td>296</td>
</tr>
<tr>
<td>The <a href="">xsl:apply-templates</a> Element</td>
<td>296</td>
</tr>
<tr>
<td>Getting Information from the Source Tree</td>
<td>297</td>
</tr>
<tr>
<td>The <a href="">xsl:value-of</a> Element</td>
<td>297</td>
</tr>
<tr>
<td>The <a href="">xsl:copy</a> Element</td>
<td>299</td>
</tr>
<tr>
<td>The <a href="">xsl:copy-of</a> Element</td>
<td>303</td>
</tr>
<tr>
<td>Influencing the Output with the <a href="">xsl:output</a> Element</td>
<td>306</td>
</tr>
<tr>
<td>Conditional Processing</td>
<td>306</td>
</tr>
<tr>
<td>The <a href="">xsl:if</a> Element</td>
<td>306</td>
</tr>
<tr>
<td>The <a href="">xsl:choose</a> Element</td>
<td>308</td>
</tr>
<tr>
<td>The <a href="">xsl:for-each</a> Element</td>
<td>311</td>
</tr>
<tr>
<td>The <a href="">xsl:sort</a> Element</td>
<td>312</td>
</tr>
<tr>
<td>XSLT Modes</td>
<td>314</td>
</tr>
<tr>
<td>XSLT Variables and Parameters</td>
<td>320</td>
</tr>
<tr>
<td>Named Templates and the <a href="">xsl:call-template</a> Element</td>
<td>322</td>
</tr>
<tr>
<td>XSLT Functions</td>
<td>323</td>
</tr>
<tr>
<td>XSLT 2.0</td>
<td>323</td>
</tr>
<tr>
<td>Grouping in Version 2.0</td>
<td>324</td>
</tr>
<tr>
<td>Non-XML Input and String Handling</td>
<td>327</td>
</tr>
<tr>
<td>Multiple Outputs</td>
<td>330</td>
</tr>
<tr>
<td>User-Defined Functions</td>
<td>332</td>
</tr>
<tr>
<td>xsl:value-of changes</td>
<td>334</td>
</tr>
<tr>
<td>Summary</td>
<td>335</td>
</tr>
<tr>
<td>Exercise Questions</td>
<td>335</td>
</tr>
<tr>
<td>Question 1</td>
<td>335</td>
</tr>
<tr>
<td>Question 2</td>
<td>335</td>
</tr>
</tbody>
</table>
**Part IV: Databases**  337  

**Chapter 9: XQuery, the XML Query Language**  339  

**Why XQuery?**  340  
- Historical Factors  340  
- Technical Factors  340  
- Current Status  341  

**XQuery Tools**  343  
- Saxon  343  
- X-Hive.com Online  345  
- X-Hive Database  346  
- Tamino Database  346  
- Microsoft SQL Server 2005  346  
- Oracle  346  

**Some XQuery Examples**  346  
- Input Functions  346  
- Retrieving Nodes  348  
- Element Constructors  351  
- The XQuery Prolog  355  
- Computed Constructors  358  
- Syntax  359  

**The XQuery Data Model**  360  
- Shared Data Model with XPath 2.0 and XSLT 2.0  360  
- Node Kinds  361  
- Sequences of Node-Sets  361  
- Document Order  361  
- Comparing Items and Nodes  361  
- Types in XQuery  361  
- Axes in XQuery  361  

**XQuery Expressions**  362  
- FLWOR Expressions  362  

**XQuery Functions**  368  
- The concat() Function  369  
- The count() Function  369  

**Using Parameters with XQuery**  370  

**User-Defined Functions**  371  

**Looking Ahead**  372  
- Update Functionality  372  
- Full-Text Search  372  

**Summary**  372
Chapter 10: XML and Databases

The Need for Efficient XML Data Stores
The Increasing Amount of XML
Comparing XML-Based Data and Relational Data

Approaches to Storing XML
Storing XML on File Systems
Using XML With Conventional Databases
Native XML Databases

Using Native XML Databases
Obtaining and Installing eXist
Interacting with eXist

XML in Commercial RDBMSs
XML Functionality in SQL Server 2000
Web Service Support

XML in Open Source RDBMS
Installing MySQL
Adding Information in MySQL
Querying MySQL
Updating XML in MySQL
Usability of XML in MySQL
Client-Side XML Support

Choosing a Database to Store XML
Looking Ahead
Summary
Exercise Questions

Part V: Programming

Chapter 11: The XML Document Object Model (DOM)
Contents

Two Ways to View DOM Nodes 448
  Overview of the XML DOM 448
Tools to Run the Examples 450
  Browser Differences 450
The Node Object 458
  Properties of the Node Object 458
  Methods of the Node Object 461
  Loading an XML Document 462
  The Effect of Text Nodes 468
  The NamedNodeMap Object 471
  The NodeList Object 475
  The DOMException Object 476
The Document Interface 478
How the XML DOM Is Used in InfoPath 2007 481
Summary 482
Exercise Questions 482
  Question 1 482
  Question 2 482

Chapter 12: Simple API for XML (SAX) 483

What Is SAX and Why Was It Invented? 483
  A Brief History of SAX 484
  Where to Get SAX 485
  Setting Up SAX 486
Receiving SAX Events 486
  ContentHandler Interface 487
  ErrorHandler Interface 504
  DTDHandler Interface 509
  EntityResolver Interface 510
  Features and Properties 510
  Extension Interfaces 514
Good SAX and Bad SAX 515
Consumers, Producers, and Filters 516
Other Languages 516
Summary 517
Exercise Questions 518
  Question 1 518
  Question 2 518
## Part VI: Communication 519

### Chapter 13: RSS, Atom, and Content Syndication 521

**Syndication and Meta Data** 521  
Syndication Systems 522  
The Origin of RSS Species 525  
RSS-DEV and RSS 1.0 529  
UserLand and RSS 2.0 531  
Atom 533  

**Working with News Feeds** 536  
Newsreaders 536  
Data Quality 536  

**A Simple Aggregator** 537  
Modeling Feeds 537  
Program Flow 540  
Implementation 540  
Transforming RSS with XSLT 557  

**Useful Resources** 567  

**Summary** 568  

**Exercise Questions** 568  
Question 1 569  
Question 2 569  

### Chapter 14: Web Services 571

**What Is an RPC?** 571  

**RPC Protocols** 573  
DCOM 573  
IIOP 574  
Java RMI 575  

**The New RPC Protocol: Web Services** 575  
XML-RPC 576  
The Network Transport 579  

**Taking a REST** 596  

**The Web Services Stack** 600  
SOAP 600  
WSDL 601  
UDDI 602  
Surrounding Specifications 602  

**Summary** 604
### Exercise Questions
- Question 1 605
- Question 2 605

### Chapter 15: SOAP and WSDL

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laying the Groundwork</td>
<td>608</td>
</tr>
<tr>
<td>Running Examples in Windows 2003, XP, and 2000</td>
<td>608</td>
</tr>
<tr>
<td>The New RPC Protocol: SOAP</td>
<td>608</td>
</tr>
<tr>
<td>Just RESTing</td>
<td>612</td>
</tr>
<tr>
<td>Basic SOAP Messages</td>
<td>613</td>
</tr>
<tr>
<td>More Complex SOAP Interactions</td>
<td>620</td>
</tr>
<tr>
<td>Defining Web Services: WSDL</td>
<td>632</td>
</tr>
<tr>
<td>&lt;definitions&gt;</td>
<td>633</td>
</tr>
<tr>
<td>&lt;types&gt;</td>
<td>633</td>
</tr>
<tr>
<td>&lt;messages&gt;</td>
<td>634</td>
</tr>
<tr>
<td>&lt;portTypes&gt;</td>
<td>635</td>
</tr>
<tr>
<td>&lt;binding&gt;</td>
<td>635</td>
</tr>
<tr>
<td><a href="">soap:body</a></td>
<td>637</td>
</tr>
<tr>
<td>&lt;service&gt;</td>
<td>638</td>
</tr>
<tr>
<td>Other Bindings</td>
<td>641</td>
</tr>
<tr>
<td>Summary</td>
<td>644</td>
</tr>
<tr>
<td>Exercise Questions</td>
<td>644</td>
</tr>
<tr>
<td>Question 1</td>
<td>644</td>
</tr>
<tr>
<td>Question 2</td>
<td>644</td>
</tr>
</tbody>
</table>

### Chapter 16: Ajax

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early Attempts at Asynchronous Updates</td>
<td>645</td>
</tr>
<tr>
<td>Microsoft versus Mozilla</td>
<td>647</td>
</tr>
<tr>
<td>Cross-Browser Solutions</td>
<td>647</td>
</tr>
<tr>
<td>Basic Posting Techniques</td>
<td>649</td>
</tr>
<tr>
<td>Transport and Processing on the Server</td>
<td>652</td>
</tr>
<tr>
<td>JSON</td>
<td>652</td>
</tr>
<tr>
<td>Payment Card Validator</td>
<td>653</td>
</tr>
<tr>
<td>The AutoSuggest Box</td>
<td>658</td>
</tr>
<tr>
<td>Server-Side Proxies</td>
<td>681</td>
</tr>
<tr>
<td>The Currency Converter Proxy</td>
<td>682</td>
</tr>
<tr>
<td>Summary</td>
<td>686</td>
</tr>
<tr>
<td>Exercise Questions</td>
<td>687</td>
</tr>
<tr>
<td>Question 1</td>
<td>687</td>
</tr>
<tr>
<td>Question 2</td>
<td>687</td>
</tr>
<tr>
<td>Question 3</td>
<td>687</td>
</tr>
</tbody>
</table>
Part VII: Display

Chapter 17: Cascading Style Sheets (CSS)

Why Stylesheets? 692
Introducing CSS 693
  CSS Properties 694
  Inheritance 695
  Attaching the Stylesheet to an XML Document 699
  Selectors 700
Using CSS for Layout of XML Documents 701
  Understanding the Box Model 702
  Positioning in CSS 706
Laying Out Tabular Data 719
Links in XML Documents 721
  XLink Support in Firefox 721
  Forcing Links Using the XHTML Namespace 725
Images in XML Documents 725
Using CSS to Add Content to Documents 726
  Attribute Content 729
    Attribute Selectors 729
    Using Attribute Values in Documents 729
Summary 732
Exercise Questions 732
  Question 1 732
  Question 2 733
  Question 3 733
  Question 4 733

Chapter 18: XHTML 735

Separating Style from Content 736
Learning XHTML 1.x 738
  Document Type Definitions for XHTML 738
  Basic Changes in Writing XHTML 740
  Styling XHTML Documents 751
  Stricter Documents Make Faster and Lighter Processors 753
  XHTML Tools 753
  Validating XHTML Documents 754
  Validation Pitfalls 756
  Mime Types Pitfalls 757
## Contents

<table>
<thead>
<tr>
<th>Chapter 19: Scalable Vector Graphics (SVG)</th>
<th>767</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What Is SVG?</strong></td>
<td></td>
</tr>
<tr>
<td>Scalable, Vector, Graphics</td>
<td>767</td>
</tr>
<tr>
<td>Putting SVG to Work</td>
<td>768</td>
</tr>
<tr>
<td>An SVG Toolkit</td>
<td>769</td>
</tr>
<tr>
<td><strong>Getting Started</strong></td>
<td>771</td>
</tr>
<tr>
<td>Views and Units</td>
<td>774</td>
</tr>
<tr>
<td>The Painter’s Model</td>
<td>774</td>
</tr>
<tr>
<td>Grouping</td>
<td>776</td>
</tr>
<tr>
<td>Transformations</td>
<td>776</td>
</tr>
<tr>
<td>Paths</td>
<td>777</td>
</tr>
<tr>
<td>Images</td>
<td>780</td>
</tr>
<tr>
<td>Text</td>
<td>781</td>
</tr>
<tr>
<td>Comments, Annotation, and Metadata</td>
<td>782</td>
</tr>
<tr>
<td>Scripting</td>
<td>784</td>
</tr>
<tr>
<td>SVG on Your Website</td>
<td>785</td>
</tr>
<tr>
<td><strong>Tangram: A Simple Application</strong></td>
<td>786</td>
</tr>
<tr>
<td>XHTML Wrapper</td>
<td>787</td>
</tr>
<tr>
<td>SVG Shapes</td>
<td>788</td>
</tr>
<tr>
<td>Tangram Script</td>
<td>792</td>
</tr>
<tr>
<td><strong>Useful Resources</strong></td>
<td>799</td>
</tr>
<tr>
<td><strong>Summary</strong></td>
<td>800</td>
</tr>
<tr>
<td><strong>Exercise Questions</strong></td>
<td>800</td>
</tr>
<tr>
<td>Question 1</td>
<td>800</td>
</tr>
<tr>
<td>Question 2</td>
<td>801</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 20: XForms</th>
<th>803</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>How XForms Improves on HTML Forms</strong></td>
<td>804</td>
</tr>
<tr>
<td>XForms Tools</td>
<td>804</td>
</tr>
<tr>
<td>An Illustrative XForms Example</td>
<td>810</td>
</tr>
</tbody>
</table>
## Contents

### XForms Form Controls
- The xforms:input Element 817
- The xforms:secret Element 818
- The xforms:textarea Element 818
- The xforms:output Element 818
- The xforms:upload Element 819
- The xforms:range Element 819
- The xforms:trigger Element 820
- The xforms:submit Element 820
- The xforms:select Element 821
- The xforms:select1 Element 822

### Constraining XForms Instances
- The xforms:bind Element 828
- W3C XML Schema in XForms 833
- Schema or Bind Elements: Which One to Choose? 834

### XForms Events
- The XForms Action Module 835
  - Developing and Debugging XForms 836

### Alternatives to XForms
- Microsoft InfoPath 836
- Adobe LiveCycle 837
- HTML Forms 838

### Summary

### Exercise Questions
- Question 1 838
- Question 2 838

### Part VIII: Case Study

#### Chapter 21: Case Study: Payment Calculator

- Mortgage Calculations 841
- What You’ll Need 842
- Online Loan Calculator 842
  - Integrating the Calculation Web Service 849
  - Enhancing the Display with SVG 865
  - Adding the Frame to the Main Page 868
- Summary 872

#### Chapter 22: Case Study: Payment Calculator — Ruby on Rails

xxiv
Welcome to Beginning XML, Fourth Edition, the book I wish I’d had when I was first learning the language!

When we wrote the first edition of this book, XML was a relatively new language but already gaining ground fast and becoming more and more widely used in a vast range of applications. By the time we started the second edition, XML had already proven itself to be more than a passing fad, and was in fact being used throughout the industry for an incredibly wide range of uses. As we began the third edition, it was clear that XML was a mature technology, but more important, it became evident that the XML landscape was dividing into several areas of expertise. In this edition, we needed to categorize the increasing number of specifications surrounding XML, which either use XML or provide functionality in addition to the XML core specification.

So what is XML? It’s a markup language, used to describe the structure of data in meaningful ways. Anywhere that data is input/output, stored, or transmitted from one place to another, is a potential fit for XML’s capabilities. Perhaps the most well-known applications are web-related (especially with the latest developments in handheld web access—for which some of the technology is XML-based). However, there are many other non-web-based applications for which XML is useful—for example, as a replacement for (or to complement) traditional databases, or for the transfer of financial information between businesses. News organizations, along with individuals, have also been using XML to distribute syndicated news stories and blog entries.

This book aims to teach you all you need to know about XML—what it is, how it works, what technologies surround it, and how it can best be used in a variety of situations, from simple data transfer to using XML in your web pages. It answers the fundamental questions:

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

**Who Is This Book For?**

This book is for people who know that it would be a pretty good idea to learn XML but aren’t 100 percent sure why. You’ve heard the hype but haven’t seen enough substance to figure out what XML is and what it can do. You may be using development tools that try to hide the XML behind user interfaces and scripts, but you want to know what is really happening behind the scenes. You may already be somehow involved in web development and probably even know the basics of HTML, although neither of these qualifications is absolutely necessary for this book.
Introduction

What you don’t need is knowledge of markup languages in general. This book assumes that you’re new to the concept of markup languages, and we have structured it in a way that should make sense to the beginner and yet quickly bring you to XML expert status.

The word “Beginning” in the title refers to the style of the book, rather than the reader’s experience level. There are two types of beginner for whom this book is ideal:

- Programmers who are already familiar with some web programming or data exchange techniques. Programmers in this category will already understand some of the concepts discussed here, but you will learn how you can incorporate XML technologies to enhance those solutions you currently develop.
- Those working in a programming environment but with no substantial knowledge or experience of web development or data exchange applications. In addition to learning how XML technologies can be applied to such applications, you will be introduced to some new concepts to help you understand how such systems work.

How This Book Is Organized

We’ve arranged the subjects covered in this book to take you from novice to expert in as logical a manner as we could. In this Fourth Edition, we have structured the book in sections that are based on various areas of XML expertise. Unless you are already using XML, you should start by reading the introduction to XML in Part I. From there, you can quickly jump into specific areas of expertise, or, if you prefer, you can read through the book in order. Keep in mind that there is quite a lot of overlap in XML, and that some of the sections make use of techniques described elsewhere in the book.

- We begin by explaining what exactly XML is and why the industry felt that a language like this was needed.
- After covering the why, the next logical step is the how, so we show you how to create well-formed XML.
- Once you understand the whys and hows of XML, you’ll go on to some more advanced things you can do when creating your XML documents, to make them not only well formed, but valid. (And you’ll learn what “valid” really means.)
- After you’re comfortable with XML and have seen it in action, we unleash the programmer within and look at an XML-based programming language that you can use to transform XML documents from one format to another.
- Eventually, you will need to store and retrieve XML information from databases. At this point, you will learn not only the state of the art for XML and databases, but also how to query XML information using an SQL-like syntax called XQuery.
- XML wouldn’t really be useful unless you could write programs to read the data in XML documents and create new XML documents, so we’ll get back to programming and look at a couple of ways that you can do that.
- Understanding how to program and use XML within your own business is one thing, but sending that information to a business partner or publishing it to the Internet is another. You’ll learn about technologies that use XML that enable you to send messages across the Internet, publish information, and discover services that provide information.