



"Learn about Extensible Markup Language (XML) - what it is, how it works and how to use it in your data exchange applications - for the web, e-commerce and more!"

BEGINNING
XML



David Hunter,

with contributions by Paul D. Boal, Gillian, Niklas, G. J. To, Brian, and Paul Springer

Table of Contents

Introduction	1
Who Is this Book For?	1
What's Covered in this Book?	2
What You Need to Use this Book	4
Conventions	4
Downloading the Source Code	6
Tell Us What You Think	6
Errata & Updates	6
Chapter 1: What Is XML?	9
Of Data, Files, and Text	9
Binary Files	10
Text Files	11
A Brief History of Markup	12
So What Is XML?	13
What Does XML Buy Us?	15
How Else Would We Describe Our Data?	15
XML Parsers	16
Why "Extensible"?	17
HTML and XML: Apples and Red Delicious Apples	18
Hierarchies of Information	18
Hierarchies in HTML	19
Hierarchies in XML	19
What's a Document Type?	21
No, Really – What's a Document Type?	21
Who Is the World Wide Web Consortium?	22
What are the Pieces that Make Up XML?	23
Where is XML Used, and Where Can It be Used?	23
Reducing Server Load	24
Web Site Content	24
Remote Procedure Calls	24
E-Commerce	24
Summary	25

Tags & Text & Elements, Oh My!	28
Rules for Elements	31
Element Names	33
Case Sensitivity	34
White Space in PCDATA	35
Attributes	38
Why Use Attributes?	41
Comments	42
Empty Elements	46
XML Declaration	47
Encoding	47
Unicode	48
Specifying Character Encoding for XML	48
Standalone	49
Processing Instructions	51
Is the XML Declaration a Processing Instruction?	51
Illegal PCDATA Characters	53
Escaping Characters	53
CDATA Sections	54
Parsing XML	57
Microsoft Internet Explorer Parser	57
James Clark's Expat	57
Vivid Creations ActiveDOM	57
DataChannel XJ Parser	57
IBM xml4j	58
Apache Xerces	58
Errors in XML	58
Summary	59

Chapter 3: XML in the Browser: Cascading Style Sheets **61**

Why Do We Need Style Sheets?	62
The History of Styling HTML	63
The Origins of Markup	63

Style Sheets and HTML	65
Creating Styles in HTML	66
The style Attribute	67
The <STYLE> Element	67
Cascading Styles	68
Overriding Styles with Style Elements	70
Comments in Style Sheets	71
A Class Act in HTML	71
Cascading Class Selectors	75
Pseudo-Classes	78
External Styles in HTML	81
<LINK>	81
The @import Directive	82
Cascading Style Sheets and XML	83
External Style Sheets in XML	83
More on External Style Sheets	85
Internal XML Cascading Style Sheets	87
The Medium is the Message	87
The @media Directive	88
Media Groups	89
The Box Model	90
The display Property	91
Other display Values	91
The Difference Between display and visibility	92
Setting Colors	94
Hex Color Codes	95
Building Borders	96
Background Images	99
Limitations When Working with Images	100
Positioning and Repeating Background Images	101
Specifying Multiple Background Properties	102
Creating Backgrounds	102
Understanding Positioning Schemes	104
Static Position	104
Relative Position	104
Absolute Position	104
Fixed Position	105
Inherited Position	105
Setting Positions and Units	105
Width and Height	107
Overflow	108

Clipping	108
Floats	109
Lists and Tables	111
Working with Lists in XML	113
Supporting Tables in CSS	114
Text and Font Manipulation	116
font-size	116
font-style and font-weight	117
font-family and @font-face	117
font-variant, text-decoration, text-transform	119
Indentation, Padding and Margins	119
text-indent	119
Margins and Padding	120
Kerning and Spacing	121
letter-spacing	121
line-height	121
Alignment	122
text-align	122
vertical-align	122
Ruby Characters	122
Printing	122
Units – Pixels versus Points	123
The @page Directive	124
Page Breaks	124
Printing Using Scripting	125
Aural Style Sheets	125
Summary	128
Chapter 4: XSLT and XPath	131
Running the Examples	132
XT	132
MSXML	132
What is XSL?	133
So How Do XSLT Style Sheets Work?	133
Associating Style Sheets Using Processing Instructions	134
Why is XSLT So Important for E-Commerce?	134

Imperative versus Declarative Programming	137
Imperative Programming	137
Declarative Programming	138
XSLT Has No Side Effects	140
XPath	141
order.xml	141
Node? What's a Node?	142
Location Paths	142
Building Location Paths	143
But Why Doesn't XPath Use XML Syntax?	144
More Specific Location Paths	144
XPath Functions	146
Node Functions	146
Positional Functions	148
Numeric Functions	149
Boolean Functions	150
String Functions	152
Axis Names	156
Summary	161

Chapter 5: XSLT – The Gory Details **163**

Templates	163
XSLT's Order of Operations	164
Default Templates	165
How Do Templates Affect the Context Node?	165
XSLT Elements and their Use	166
<xsl:stylesheet>	166
<xsl:template>	167
Template Priority	167
<xsl:apply-templates>	168
<xsl:value-of>	171
<xsl:output>	173
<xsl:element>	175
<xsl:attribute> and <xsl:attribute-set>	178
Related Groups of Attributes	179
<xsl:text>	181
Conditional Processing with <xsl:if> and <xsl:choose>	182
<xsl:for-each>	187

<xsl:copy-of>	190
<xsl:copy>	192
<xsl:sort>	195
Modes	197
Variables, Constants, and Named Templates	201
<xsl:variable>	202
Named Templates	204
Parameters	205
Using XSLT with CSS	208
CSS and XSL-Formatting Objects	213
XSLT in the Real World	213
Summary	214
<hr/>	
Chapter 6: The Document Object Model (DOM)	217
What Is the DOM?	218
XML as an Object Model	218
The XML DOM	219
DOMString	219
DOM Interfaces	220
So What are Interfaces?	221
Implementing Interfaces	221
DOM Implementations	223
The DOM Core	224
Extending Interfaces	225
The DOM Core in Detail	226
Exceptions	226
DOMException	227
Node	227
Getting Node Information	227
Traversing the Tree	230
Adding and Removing Nodes	232
Document	235
DOMImplementation	238
DocumentFragment	239
NodeList	240
Element	242
NamedNodeMap	245

Attr	247
CharacterData and Text	249
Handling Complete Strings	249
Handling Sub-Strings	250
Modifying Strings	250
Splitting Text	251
Comment and CDATASection	254
ProcessingInstruction	256
Summary	259
Chapter 7: The Simple API for XML (SAX)	261
<hr/>	
What Is SAX, and Why Was It Invented?	261
The Fabulous Lost Treasure of the Xenics	262
So What Does that Have to Do with SAX?	262
Where to Get SAX	263
Who is this Megginson Guy Anyway?	264
Using the SAX Interfaces	264
Preparing the Ground	264
How to Receive SAX Events	268
Extracting Character Data	273
Extracting Attributes	276
Error Handling	281
More About Errors – Using the Locator Object	284
Even More About Errors – Catching Parsing Errors	286
Other Methods in DocumentHandler	288
ignorableWhitespace	288
processingInstruction	289
Good SAX and Bad SAX	290
SAX 2.0	291
Summary	291
Chapter 8: Namespaces	293
<hr/>	
Why Do We Need Namespaces?	293
Using Prefixes	294
So Why Doesn't XML Just Use These Prefixes?	296

How XML Namespaces Work	297
Default Namespaces	300
Namespaces of Descendents	301
Canceling Default Namespaces	303
Do Different Notations Make Any Difference?	303
Namespaces and Attributes	305
XSLT from a Namespace Point of View	308
What Exactly are URIs?	312
URLs	312
URNs	313
Why Use URLs for Namespaces, Not URNs?	314
What Do Namespace URIs Really Mean?	315
When Should I Use Namespaces?	316
Summary	317
Chapter 9: Basic Valid XML: DTDs	319
Why Do We Need DTDs?	320
XML Application Requirements	320
Document Type Definitions	321
Valid XML	321
Validating Parsers	322
Sharing DTDs	324
A Simple Example	324
Why Use XML to Do This?	324
The Basic BookCatalog Data Model	325
DTDs	329
Internal versus External DTDs	330
When to Use Internal and External Subsets	330
Associating a DTD with an XML Document	331
The XML Declaration (A Refresher)	331
The Document Type (DOCTYPE) Declaration	332
Basic DTD Markup	337
Element Type (ELEMENT) Declarations	338
Element Content Categories	338
Content Models	340

Attribute (ATTLIST) Declarations	349
Attribute Types	351
Attribute Defaults	356
Extending the BookCatalog Data Model	358
Limitations of DTDs	373
Summary	376
Chapter 10: Valid XML: Schemas	379
Schemas versus DTDs	380
Syntax	381
Content Models	381
Data Typing	382
Extensibility	382
Dynamic Schemas	383
Basic Schema Principles	383
XML Parsers with Schema Support	384
Structures and Datatypes, or Chickens and Eggs	385
Datatypes	385
Primitive Datatypes	386
Primitive Types for XML Schema	386
Derived Datatypes	387
Built-in Derived Types for XML Schema	387
Atomic and List Datatypes	388
Atomic Datatypes	388
List Datatypes	388
Aspects of Datatypes	389
Value Spaces	390
Lexical Spaces	390
Facets	390
Fundamental Facets	390
Constraining Facets	392
Associating Schemas with XML Documents	395
XML Schema Preamble – The <schema> Element	395
Specifying the Namespaces	396
Specifying the Version	396
XDR Preamble – The <Schema> Element	396

Structures	397
Datatype Definitions	397
Simple Type Definitions	398
Complex Type Definitions	400
Content Models	406
Element Content Categories	407
Cardinality – minOccurs and maxOccurs	407
Choice and Sequence Groups – <choice> and <sequence>	408
Mixed Content (Unsequenced Group) – <all>	409
Content From Another Schema – <any>	410
Model Groups – <group>	411
Element Declarations	411
Element Constraints – <element>	412
Attribute Declarations	414
Attribute Constraints – <attribute>	414
Attribute Groups – <attributeGroup>	416
Annotations	418
Summary	419
Chapter 11: Advanced Valid XML	421
<hr/>	
Entities	422
Entity References	423
Character References	424
General Entities	424
Parsed Entities	425
Unparsed Entities	429
Parameter Entities	430
ISO and Other Standard Entity Sets	432
Non-XML Data (Notations)	434
The <!NOTATION> Declaration	435
The NOTATION Attribute Type	435
The ENTITY and ENTITIES Attribute Types	435
Internal versus External DTD Subsets	438
The standalone Attribute	439
Conditional Sections	439
Nesting Conditional Sections	443
Summary	443

HTML Linking	446
XML Linking	447
XLink	448
XLink Attributes	449
The type Attribute	450
The href Attribute	451
Semantic Attributes (role and title)	451
Behavior Attributes (actuate and show)	452
The from and to Attributes	453
Link Types	454
Simple Links	454
Extended Links	457
Locator-type Elements	458
Arcs	459
Resource-type Elements	461
Title-type Elements	463
Defaulting XLink Attributes	463
Pointing to Document Fragments with XPointer	464
Appending XPointer Expressions to URIs	465
Using Multiple XPointer Expressions	465
XPointer Schemes	467
XPointer Shorthand Syntaxes	468
Full Syntax	468
Bare Names Syntax	469
Child Sequence Syntax	469
Locations, Points and Ranges	470
Points	470
Ranges	471
How Do We Select Ranges?	472
XPointer Function Extensions to XPath	474
Querying	475
Summary	475

Databases, Yesterday and Today	478
SQL	479
Retrieving Data from Multiple Tables	480
Normalization	481
Making Use of What You've Got	482
<i>N-Tier Architecture</i>	482
Using XML in an N-tier Application	484
Returning XML from a Data Object	486
Integrating XML into the Database Itself	487
Storing XML in a Database	487
Using the Database	488
An Even Simpler Data Object	489
Database Vendors and XML	490
Microsoft's XML Technologies	490
MSXML	490
Visual Basic Code Generator	491
SQL Server	491
Putting SQL Server to Work	492
Oracle's XML Technologies	494
XML Parsers	494
Code Generators	494
XML SQL Utility for Java	494
XSQL Servlet	495
Putting Oracle to Work	496
Who Needs a Database Anyway?	496
Summary	498

Chapter 14: Other Uses for XML **501**

Serializing Object Models	502
Suggestions for a Good Object Model	508
Include Pretty-Printing Characters	508
Allow Each Object to Serialize Itself	510
Commit/Rollback Functionality	517
Don't Return Anything from an Empty Object	517
Keep Track of Historical Information, Including Errors	518

XML-RPC	519
SOAP	521
The Poor Man's RPC	521
Stateless Objects	526
Resource Description Framework (RDF)	528
How Does It Work?	529
Schematron	530
Summary	534
<hr/> Case Study 1: Lydia's Lugnuts Web Store	<hr/> 537
Complexity	538
Maintenance	540
Creating Order from Chaos	540
Designing the XML Structure	540
Creating XML Files	542
Grouping Products Together	544
Transforming and Displaying XML with XSL	545
The Existing Web Page	547
The XSLT Style Sheet	549
Incorporating Content from Multiple Sources	557
Product Listings	558
Nuts, Nuts? Nuts! Templates	561
The Nut Haus Templates	562
The New HTML Web Pages	563
The Future of Lydia's Lugnuts	565
<hr/> Case Study 2: Discussion Group System	<hr/> 567
Designing the Architecture	567
An XHTML Framework System	568
A Note About the XMLPipes Framework	569
What Does the XHTML Framework Do?	570
Defining the Server	572
Defining Membership Directories	573

Designing the Message Boards	576
Retrieving the Message Child Tree	580
Retrieving the Message Ancestor Tree	580
Retrieving the Message Sibling Tree – Next	581
Retrieving the Message Sibling Tree – Previous	582
Creating Messages	583
Using Namespaces	583
Editing and Deleting Messages	590
Deleting a Message	592
Viewing Lists of Messages	593
A Few Thoughts on Online Editors	599
Editing and Viewing Individual Messages	600
Summary	608
<hr/> Case Study 3: XML for a Business-to-Business Application	<hr/> 611
An E-Commerce Problem and Our Solution	612
E-Commerce Requirements	613
Why Doesn't EDI Achieve This?	614
The eBIS-XML Concept	615
So What (exactly) is eBIS-XML?	617
An Existing eBIS-XML Application	618
Decisions, Decisions	619
DTD versus Schema	620
Style Sheets and XSL	620
Tools for Developing Schemas	621
eBIS-XML Schemas	624
Developing the v2.4 eBIS-XML Schemas	624
Moving On Up – Version 3	627
The Final Schema	632
Summary of Schema Development Tips	635
A Reprise on the Tools Used	635
eBIS-XML Style Sheets	639
A Simple Invoice	639
Push and Pull Models	643
More About the Invoice Style Sheet	644
Doing it the Easy Way?	654
Summary of the Style Sheet	655

Have We Met Arthur's Requirements?	655
Next Steps	657
Simple Routing	657
The BizTalk Header	658
Choreography	659
More Futures	660
And If We Were to Start Now?	660
Summary	661
Appendix A: The XML Document Object Model	663
Fundamental Interfaces	663
DOMException	663
Node	664
Document	666
DOMImplementation	668
DocumentFragment	668
NodeList	668
Element	669
NamedNodeMap	670
Attr	671
CharacterData	671
Text	672
Comment	672
Extended Interfaces	672
CDATASection	672
ProcessingInstruction	673
DocumentType	673
Notation	673
Entity	674
EntityReference	674
Appendix B: SAX 1.0: The Simple API for XML	677
Class Hierarchy	678
Interface org.xml.sax.AttributeList	678
Interface org.xml.sax.DocumentHandler	681
Interface org.xml.sax.DTDHandler	685

Interface org.xml.sax.EntityResolver	687
Interface org.xml.sax.ErrorHandler	689
Class org.xml.sax.HandlerBase	691
Class org.xml.sax.InputSource	694
Interface org.xml.sax.Locator	698
Interface org.xml.sax.Parser	699
Class org.xml.sax.SAXException	702
Class org.xml.sax.SAXParseException	703

Appendix C: ASP Quick Start Tutorial **707**

The Anatomy of the HTTP Protocol **708**

Overview	708
The HTTP Server	708
Protocol Basics	709
The Connection	709
The Request	709
The Response	712
Disconnecting	712

Introducing Active Server Pages **713**

How the Server Recognizes ASPs	713
ASP Basics	713
The Tags of ASP	714
<SCRIPT> Blocks	714
The Default Scripting Language	714
Mixing HTML and ASP	715
Commenting Your ASP Code	716

The Active Server Pages Object Model **717**

Collections	717
Iterating the Contents Collection	718
Removing an Item from the Contents Collection	719
The Request Object	719
The Request Object's Collections	720
Request Object Property and Method	724
The Response Object	724
The Response Object's Collection	724
The Response Object's Methods	725
The Response Object's Properties	727

The Application and Session Objects	728
Scope Springs Eternal	728
The global.asa File	729
The Application Object	729
The Session Object	730
The Server Object	732
Properties	732
Methods	732
TheObjectContext Object	733
SetAbort	734
SetComplete	734
Using Active Server Pages Effectively	734
Designing the Site	734
Creating the global.asa File	735
Creating our Main Page	735
The ASP/VBScript Section	736
The HTML Section	738
Summary	740
Appendix D: JavaScript Quick Start Tutorial	743
<hr/>	
JavaScript and HTML	744
JavaScript and Java	744
Running the Examples in this Appendix	745
Statements	746
Comments	747
Single Line Comments	747
Multi-line Comments	747
Variables	748
Data Types	749
Variable Names	750
Operators	751
Mathematical	751
Assignment	752
Logical (Boolean)	752
Precedence of Operators	753

Functions	754
Creating Functions	754
Returning Information from a Function	754
Passing Information to a Function	755
Built-in Functions	755
Program Flow	756
Conditional Code	756
The if Statement	756
The switch Statement	758
Loops	759
The for Loop	760
while and do ... while Loops	760
The break and continue Statements	761
Objects	762
String	763
The toLowerCase and toUpperCase Methods	763
Methods to Return Strings and Characters	763
Methods to Search Strings	764
Joining Strings	764
Number	764
Math	765
Date	766
Arrays	766
Summary	767

Appendix E: Support, Errata, and P2P.Wrox.Com	769
--	------------

The Online Forums at P2P.Wrox.Com	769
How to Enroll for Online Support	770
Why this System Offers the Best Support	771
Support and Errata	771
Wrox Developer's Membership	771
Finding an Erratum on the Web Site	772
Add an Erratum: E-mail Support	773
Customer Support	774
Editorial	774
The Authors	774
What We Can't Answer	774
How to Tell Us Exactly What You Think	774

Appendix G: XML Schema Datatypes

Primitive Types

783

string	784
boolean	784
decimal	784
float	785
double	785
timeDuration	785
recurringDuration	787
binary	787
uriReference	788
ID	788
IDREF	788
ENTITY	788
NOTATION	789
QName	789
Constraining Facets for Primitive Types	789

Built-In Derived Types

790

language	791
name, NCName	792
integer, negativeInteger, positiveInteger, nonNegativeInteger, nonPositiveInteger	792
byte, short, int, long	792
unsignedByte, unsignedShort, unsignedInt, unsignedLong	792
century, year, month, date	792
recurringDate, recurringDay	793
time, timeInstant, timePeriod	793
Constraining Facets for Derived Types	794