

Index

SYMBOLS AND NUMERICS

- & (ampersand), avoiding error with, 15**
- ' (apostrophe), avoiding error with, 15**
- * (asterisk) quantifier, DTDs, 157**
- > (greater than sign), avoiding error with, 15, 252**
- < (less than sign), avoiding error with, 15, 252**
- + (plus sign) quantifier, 153–155**
- ? (question mark) quantifier, DTDs, 155–156**
- quotation marks**
 - double (“), avoiding error with, 15
 - single (‘), avoiding error with, 15
- , (sequence concatenation operator), 289**
- | (union concatenation operator), 289**
- 1.1 faults, SOAP**
 - <detail>, 618
 - <faultcode>, 617–618
 - <faultfactor>, 618
 - <faultstring>, 618
 - message, 617
- 1.2 faults, SOAP**
 - <Code>, 619
 - <Detail>, 621
 - message, 619
 - <Node>, 620
 - <Reason>, 620
 - <Role>, 620–621
 - <Subcode>, 620
 - <Text>, 620
 - <Value>, 619–620

- 10g relational database, Oracle**
 - described, 340
 - retrieving XML, 341–343
 - storing XML, 343–345

A

- Access (Microsoft)**
 - importing XML, 792–794
 - saving as XML, 794–798
- actions logic, XForms, 744–745**
- Adaptive Server Enterprise 15 (Sybase), 349**
- addressing WS-* specifications, 657–658**
- aggregate functions (count, avg, max, min, and sum), 312**
- Ajax**
 - client-side code, 423–425
 - contact information file, 418–419
 - cross-browser code, 426–427
 - described, 405
 - DOM
 - described, 415
 - events, 417–418
 - objects, 416–417
 - JavaScript, adding to Web page
 - functions, 406–407
 - problem, 406
 - tag, 406
 - JavaScript data types
 - described, 407–408
 - language features, 409–410
 - JSON data format, 425

Ajax (continued)

- libraries
 - described, 427–428
 - Microsoft, 428–431
 - Prototype, 432–434
- resources, 434
- selected contacts, 420
- server-side query, 420–423
- XMLHttpRequest, 410–415

Altova SemanticWorks

- described, 762
- graph example, 763
- predicate, associating, 766–767
- prefix, adding, 764
- resource, adding, 764–766
- triples, 767–769

Altova XMLSpy 2006

- benefits and disadvantages, 47
- described, 39–44
- schema development, 44–45
- script editor, 46–47
- Spy debugging, 624–629
- XML Schemas, 209–210
- XSLT development, 46

ampersand (&), avoiding error with, 15

annotating schemas, RELAX NG, 231

- <annotation>, **XML Schemas**
- <appInfo>, 206
- <documentation>, 206

ANY value, specifying content with, 148–149

anyURI primitive data type, 185

Apache Xerces, 7

apostrophe ('), avoiding error with, 15

appearance, XForms, 733

- <appInfo>, **XML Schemas, 206**

application

- Microsoft WSE server, configuring, 669–673
- WPF, creating, 815

architecture, SAX

- basic application flow, 380–381
- illustrated, 380

article schema collection

- applying to table, 334
- creating, 333

ASP.NET (Microsoft)

- consuming Web page, building, 589–591
- HTTPModule, 713–714
- Web reference, adding, 587–589

asterisk (*) quantifier, DTDs, 157

asynchronous consumption, Web services

- described, 601–602
- slow service, building, 602–603
- TakeLongTime() WebMethod, 603–605

ate primitive data type, 185

Atom

- dates, 546
- described, 531
- elements, 547–548
- reading RSS
 - described, 548–550
 - with Java, 556–558
 - XmlDocument, 550–552
 - XmlReader, 552–556
- resources, 563
- sample 1.0 feed, 543–546
- writing RSS
 - class libraries, 562
 - with Java, 560–562
 - with .NET, 559–560

atomic simple type, 176–177

attachments, WS-* specifications, 658

Attr interface, DOM, 366–367

attribute declarations, DTDs

- data types, 160
- described, 158–159
- enumerations, using as values, 162–163
- #FIXED keyword, 161–162
- #IMPLIED keyword, 161
- #REQUIRED keyword, 160–161

attribute element, 248

attributeFormDefault attribute, 172–173

attributes

- described, 26
- DOM, 367–368
- empty, 26–27
- extracting, 334–335
- groups, XML Schemas root document, 202–203
- naming, 26
- RELAX NG, 219–220
- uniqueness, 27
- XML Schemas root document
 - attribute groups, 202–203
 - default values, 199–200
 - described, 198–199
 - use attribute, 200
 - values, restricting, 200–202
- xml:lang, 28
- xml:space, 28–29

AUTO query, Microsoft SQL Server 2005, 326, 327–328

axes, XPath, 281–282

Axis Web service

- advanced deployment
 - service implementation, 583–586
 - WSDD file, creating, 586–587
- described, 580
- .java, renaming to .jws, 582–583
- setting up, 580–582

B

background-color property, CSS, 82

base64Binary primitive data type, 185

Beginning Python (Wiley Publishing), 510

Berkeley DB XML, 349

binary attachments, Web services, 650

<binding> WSDL, 643–644

blank space

- tags, 13–14
- XSLT, controlling with `strip-space`, `preserve-space`, and `decimal-format`, 243–244

body, SOAP

- of request, 612–613
- of response, 613

Boolean expressions in predicates, 280

boolean primitive data type, 185

border property, CSS, 82

box layout, CSS, 88–91

box model hack, 90

browser

- Ajax code, 426–427
- CSS compatibility, 88–91
- XPath, 297–298
- XSLT, executing, 262–263

bulk insert, OPENXML, 336

byte derived data type, 186

C

C#

- building Web services
 - described, 571–574
 - Microsoft Web services test page, 575–576
 - protocols, altering, 578–579
 - WebMethod, testing, 576–578
- consuming Web services, 587–591

caching Web services, 599–600

calling XSLT templates, 249–250

cardinality, XML Schemas root document, 192

Cascading Style Sheets (CSS)

- box layout and cross-browser compatibility, 88–91
- external reference with link element, 87–88
- external reference with processing instructions, 87
- levels, active, 81
- multicolumn layout, 92–94
- properties, 82–83
- selectors, 84–85
- style information, adding to page, 85–87
- validating, 94

case-sensitivity, XML, 11–12

CDATA sections, 16–17

CDATASection interface, DOM, 369–370

Central European (windows-1250) character set, 23

CharacterData interface, DOM, 368–369

characters

- DTDs, 163
- enclosing special, 15

child elements, DTDs

- asterisk (*) quantifier, 157
- choice option, 157–158
- described, 150–151
- number of instances, specifying, 151–152
- plus sign (+) quantifier, 153–155
- question mark (?) quantifier, 155–156
- XML elements, reusing, 152–153

child node, DOM, 364

children, XForms control, 731–732

choice option, DTDs, 157–158

choose element, 252

class libraries, RSS, 562

client

- Ajax code, 423–425
- Microsoft WSE, 676–679
- XForms, 755–756
- XML processing, 354

client application

- consuming Web services with Java, 599
- Java Web service, running, 596–597
- running Web services with Java, 599

<Code>, 619

code, executing via XSLT, 259–261

color property, CSS, 82

command line, XSLT, 258–259

comment

- described, 23–25
- DOM interface, 369
- XML Schemas, 205–206
- XPath, 286–287

compact schema, RELAX NG, 236–237

complex types, 180–181

composition

Web services

data, representing and communicating, 568

described, 567–570

discovering, 570–571

XSL-FO, 102–103

Compressed UCS (UTF-16) character set, 23

Compressed Unicode (UTF-8) character set, 23

conditional processing (if element), 251–252

consuming application, Microsoft Trace Utility, 622–623

Contact class, 706–707

contact information file, Ajax, 418–419

Contact Manager class, 707–711

content tags, 12

controlling XSLT, XQuery, and XPath processing

with Java, 491–493

XQJ, 499–500

controls, XForms, 727–730

converting schema types, Oxygen XML Editor 6.2, 58–60

Cooktop editor, 62

coordination, WS-* specifications, 658

copy element, 248

copy-of element, 248

core specifications, WS-*, 659–663

credential exchange, WS-* security, 657

credential verification class, Microsoft WSE server, 667–669

cross-browser

Ajax code, 426–427

CSS compatibility, 88–91

CSS (Cascading Style Sheets)

box layout and cross-browser compatibility, 88–91

external reference with link element, 87–88

external reference with processing instructions, 87

levels, active, 81

multicolumn layout, 92–94

properties, 82–83

selectors, 84–85

style information, adding to page, 85–87

validating, 94

current function, 257

cursor property, CSS, 83

Cyrillic (windows-1251) character set, 23

D

data

DTDs, 160

representing in XML document, 5

Web services, representing and communicating, 568

Data Access Layer class, 711–712

databases

Berkeley DB XML, 349

Mark Logic Server, 349

resources, 350

retrieving data as XML, 323–324

storing XML, 324–325

Sybase Adaptive Server Enterprise 15, 349

dates, Atom, 546

dateTime primitive data type, 186

debugging

Stylus Studio 2006, 54

with trace(), 292–293

with XMLSpy, 46

XSLT, 274–275

decimal primitive data type, 186

decimal-format, 243–244

declaration

DTD attribute

data types, 160

described, 158–159

enumerations, using as values, 162–163

#FIXED keyword, 161–162

#IMPLIED keyword, 161

#REQUIRED keyword, 160–161

encoding attribute, 22–23

standalone attribute, 23

version attribute, 22

declaring elements

atomic simple type, 176–177

complex types, 180–181

list simple type, 177–178

named complex types, reusability and, 181–183

<sequence> and <all> elements, 183–184

type element, 184–188

union simple type, 179

default values, XML Schemas root document, 195–196

DefaultHandler class, SAX, 385–387

<definitions>, WSDL, 638–640

derived data types, XML schema, 186–187

deserializing XML, 444–447

<Detail>, 621

diagnostics, Microsoft WSE server, configuring, 673–674

digital signature, Web services, 651

discovery, 652

display property, CSS, 82

`doc ()` **functions, XQuery, 312**

document

DOM interface, 357–358

editing Excel, 777–782

editing XML in Word, 788–792

saving Word as XML, 782–788

well-formed, 6

WPF

building, 815–820

viewing, 820–823

WSDL structure, defining, 637–638

.xml file, 17–22

XPath, many faces of a, 283

XPS file, as, 825–829

`document` **function, 256**

Document Object Model. See DOM

document type declaration element

described, 144–145

inline and URI, 146–147

SYSTEM and PUBLIC keywords, 145–146

Document Type Definitions. See DTDs

`<documentation>`

WSDL, 647–648

XML Schemas, 206

DOM (Document Object Model)

Ajax

described, 415

events, 417–418

objects, 416–417

Attr interface, 366–367

attributes, creating, 367–368

CDATASection interface, 369–370

CharacterData interface, 368–369

client-side XML processing, 354

Comment interface, 369

described, 353–354

document interface, 357–358

element interface, 360

errors, handling, 370–371

loading document

described, 358–359

`readyState` property, 359–360

NamedNodeMap interface, 366

new element, creating, 361

Node interface

described, 361–363

inserting node before existing node, 364

new child node, appending, 364

new node, creating, 363–364

removing child node, 364

replacing node, 365

text values of elements, accessing, 365

NodeList interface, 365–366

sample file, 354–356

Text interface, 369

transformation using XSL, 371–373

validation, 373–375

`double` **primitive data type, 186**

double quotation marks ("), avoiding error with, 15

DTDs (Document Type Definitions)

attribute declarations

data types, 160

described, 158–159

enumerations, using as values, 162–163

#FIXED keyword, 161–162

#IMPLIED keyword, 161

#REQUIRED keyword, 160–161

building your own, 142–144

child elements

asterisk (*) quantifier, 157

choice option, 157–158

described, 150–151

number of instances, specifying, 151–152

plus sign (+) quantifier, 153–155

question mark (?) quantifier, 155–156

XML elements, reusing, 152–153

described, 135–138

document type declaration element

described, 144–145

inline and URI, 146–147

SYSTEM and PUBLIC keywords, 145–146

element declarations

ANY value, specifying content with, 148–149

described, 147–148

empty values, 150

#PCDATA value, limiting elements with, 149–150

entity declarations

characters, 163

external entities, 165

internal entities, 163–164

external, 140–142

internal, 138–140

notation declarations, 165

validation, 166–167

XML Schemas, problem with, 169–171

XML tools, creating with, 166

`duration` **primitive data type, 186**

dynamic languages, 501. See also Perl; Python; Ruby

E

Eclipse editor, 62, 300

editing

Excel documents, 777–782

XML documents in Word, 788–792

editors

Altova XMLSpy 2006

benefits and disadvantages, 47

described, 39–44

schema development, 44–45

script editor, 46–47

XSLT development, 46

Cooktop, 62

Eclipse, 62

Emacs, 62

Oxygen XML Editor 6.2

basic editing, 55–57

benefits and disadvantages, 62

converting between schema types, 58–60

schema development, 57–58

SVG viewer, 61

XSLT development, 60–61

Stylus Studio 2006

basic editing, 47–50

benefits and disadvantages, 54–55

debugging, 54

schema development, 50–51

XSLT development, 51–53

Vi, 62

Visual Studio .NET 2003

benefits and disadvantages, 35

described, 33–35

schema development, 35

Visual Studio 2005

basic editing, 35–37

schema development, 38

XSLT development, 38–39

XMetaL, 62

`element` **element, 248**

`elementFormDefault` **attribute, 174**

elements

Atom, 547–548

data callback, SAX, 390

described, 7–8

empty, 11

interface, DOM, 360

naming conventions for elements, 8–9

nesting, 10

new DOM, 361

start and end tag, 8

XML Schemas root document restrictions

cardinality, 192

default values, 195–196

fixed values, 196–197

`maxOccurs` attribute, 193–195

`minOccurs` attribute, 192–193

null values, 197–198

Emacs editor, 62

empty attributes, 26–27

empty elements, 11

empty space

tags, 13–14

XSLT, controlling with `strip-space`, `preserve-space`, and `decimal-format`, 243–244

empty values, DTDs, 150

enclosed expressions, XQuery, 306

encoding attribute, 22–23

encoding data types, SOAP, 621

encryption, Web services, 651

`endDocument()` **method, SAX, 387**

`endElement()` **method, SAX, 389–390**

English character set (US-ASCII), 23

ENTITIES derived data type, 186

entity declarations, DTDs

characters, 163

external entities, 165

internal entities, 163–164

ENTITY derived data type, 186

entity references, 14–15

enumeration primitive data type, 201

enumerations, DTDs, 162–163

envelope, SOAP, 611–612

errors

debugging

Stylus Studio 2006, 54

with `trace()`, 292–293

with XMLSpy, 46

XSLT, 274–275

DOM, handling, 370–371

parser, avoiding, 14–15

SAX, 392–393

events

Ajax, 417–418

XForms, 743–744

Excel (Microsoft)

- editing documents, 777–782
- workbooks, saving as XML, 773–777

execQuery(), **invoking**

- XmlCursor, 317–318
- XmlObject, 315–317

exist() **function**, **319****exist()** **method**, **321****existsnode()**, **344****EXPLICIT query**, **326**, **328–329****expression**

- XPath, tuning, 284
- XQuery, structure of, 304–305

Extensible Application Markup Language (XAML)

- described, 807–808
- thin- of thick-client, 808–809
- WPF

- application, creating, 815
- described, 809–812
- document, building, 815–820
- document, viewing, 820–823
- image, adding to document, 823–825
- nesting controls, 812–814
- within Visual Studio 2005, 809–812
- XPS file, saving document as, 825–829

eXtensible HyperText Markup Language. **See XHTML****eXtensible Markup Language**. **See XML****eXtensible Stylesheet Language-Formatting Objects**.

See XSL-FO

extension, SOAP, **655–656****external entities, DTDs**, **165****external reference**

- with link element, CSS, 87–88
- with processing instructions, CSS, 87

extract(), **344****extractvalue()**, **344****F****factory mechanism, JAXP**, **493–495****<faultcode>**, **617–618****<faultfactor>**, **618****faults**

- SOAP 1.1
 - <detail>, 618
 - <faultcode>, 617–618
 - <faultfactor>, 618
 - <faultstring>, 618
 - message, 617

SOAP 1.2

- <Code>, 619
- <Detail>, 621
- message, 619
- <Node>, 620
- <Reason>, 620
- <Role>, 620–621
- <Subcode>, 620
- <Text>, 620
- <Value>, 619–620
- <faultstring>, **618**

feeds, RSS

- described, 531
- reading with Atom
 - described, 548–550
 - with Java, 556–558
 - XmlDocument, 550–552
 - XmlReader, 552–556
- resources, 563
- RSS 1.0, 540–542
- RSS 2.0, 535–540
- sample feeds by version, 532–534
- versions listed, 532
- writing with Atom
 - class libraries, 562
 - with Java, 560–562
 - with .NET, 559–560

Firefox XML parser, **7****#FIXED keyword**, **161–162****fixed values, XML Schemas root document**, **196–197****Flickr Photo Search service**, **696–698****float primitive data type**, **186****float property, CSS**, **83****FLWOR expressions**

- described, 306–307
- for clause, 308–309
- let clause, 309
- order by clause, 310
- return clause, 310
- sample, 307–308
- where clause, 309–310
- XPath expressions versus, 310–311

** element, XHTML**, **81****font property, CSS**, **83****font-style property, CSS**, **83****for clause, FLWOR expressions**, **307**, **308–309****FOR XML AUTO query**, **331–332****for-each element, XSLT**, **252–255****format-number function**, **257**

forms

- Microsoft InfoPath
 - described, 750–754
 - XForms versus, 754–755
- XForms
 - actions logic, 744–745
 - appearance, changing control, 733
 - children, common control, 731–732
 - clients, 755–756
 - controls, 727–730
 - described, 722–727
 - events logic, 743–744
 - grouping controls, 733–737
 - instance data, binding, 737–741
 - Microsoft InfoPath versus, 754–755
 - sample, 745–750
 - submit protocol, 741–743
- XHTML, 721–722
- `fractionDigits` primitive data type, 201
- function calls in XPath expressions, 285–286
- functions
 - exposing as Web services, 336–340
 - JavaScript, adding to, 406–407
 - XQuery
 - aggregate (count, avg, max, min, and sum), 312
 - built-in, described, 311–312
 - `doc()`, 312
 - string, 312–313
 - user-defined, 313–314
 - XSLT built-in, 256–257

G

- `gDay/gMonth/gMonthDay/gYear/gYearMonth` primitive data types, 186
- `generate-id` function, 257
- Geocode service
 - Flickr Photo Search Wrapper class, 699–700
 - `GeographicLocation` class, 689–690
 - parameters, 685–687
 - `PhotoInformation` class, 698–699
 - testing, 690–692
 - testing Term Extraction Wrapper class, 694–696
 - wrapping
 - Flickr Photo Search service, 696–698
 - generally, 687–689
 - Term Extraction service, 692–694
- `GeographicLocation` class, 689–690
- getting data as XML, 323–324

getting XML

- Microsoft SQL Server 2005
 - AUTO query, 326, 327–328
 - EXPLICIT query, 326, 328–329
 - FOR XML AUTO query, 331–332
 - PATH query, 329–331
 - RAW query, 326–327
- Oracle 10g relational database, 341–343
- Xindice database, 345–348
- getting XSLT values
 - calling templates, 249–250
 - described, 248–249
 - multiple templates for single element, 250–251
- grammar, RELAX NG, 223–224
- graph
 - Altova SemanticWorks, 763
 - RDF, 760–762
- greater than sign (>), avoiding error with, 15, 252
- Greek (windows-1253) character set, 23
- `<group>` element, XML Schemas root document, 189–192
- grouping controls, XForms, 733–737
- guaranteed message exchange, Web services, 651

H

- `hCalendar`, 97–98
- `hCard`, 96–97
- header
 - SOAP
 - actor attribute, 615–616
 - consuming messages, 631–633
 - creating messages, 630–631
 - described, 614–615, 629, 655
 - `mustUnderstand` attribute, 616
 - role attribute, 616
 - XSL-FO, 120–122
- height property, CSS, 83
- HelloWorld example
 - Web service, 594–596
 - XSL-FO, 109–110
- `hexBinary` primitive data type, 186
- `hReview`, 98–99
- HTML (HyperText Markup Language), generating XSLT output, 268–272
- HTTP (HyperText Transfer Protocol)
 - handler, 703–704
 - verbs and URLs, 701–703

I

IBM XMLParser for Java, 7

ID **derived data type**, 186

IDREF/IDREFS **derived data types**, 186

if **element**, 251–252

ignorableWhitespace() **method**, SAX, 390–391

image, adding to document, 823–825

#IMPLIED **keyword**, 161

import, 243

<import>

- WSDL, 645–647
- XML Schemas root document, 203–204

include, 243

<include>, **XML Schemas root document**, 204

indent, XML, 10

InfoPath form (Microsoft)

- described, 750–754
- XForms versus, 754–755
- XML, saving, 798

input forms

- Microsoft InfoPath
 - described, 750–754
 - XForms versus, 754–755
- XForms
 - actions logic, 744–745
 - appearance, changing control, 733
 - children, common control, 731–732
 - clients, 755–756
 - controls, 727–730
 - described, 722–727
 - events logic, 743–744
 - grouping controls, 733–737
 - instance data, binding, 737–741
 - Microsoft InfoPath versus, 754–755
 - sample, 745–750
 - submit protocol, 741–743
- XHTML, 721–722

inspection, Web services, 651

installing

- Perl modules, 504
- SAX, 378–379

instances

- data, binding XForms, 737–741
- number, specifying in DTDs, 151–152

int **derived data type**, 187

integer **derived data type**, 187

IntelliJ, 300

IntelliSense editor, 34–35

interface, Node

- described, 361–363
- inserting node before existing node, 364
- new child node, appending, 364
- new node, creating, 363–364
- removing child node, 364
- replacing node, 365
- text values of elements, accessing, 365

internal entities, DTDs, 163–164

Internet Explorer XMLParser (Microsoft), 7, 13

invoice, XSL-FO working example, 119–120

ISO 639 language standard, 28

ISO 3166 language standard, 28

ISO-2022-JP (Japanese) character set, 23

ISO-8859-1 through 4 (Latin) character sets, 23

J**Japanese character set (ISO-2022-JP)**, 23**Java**

- building Web services with, 579
- controlling XSLT, XQuery, and XPath processing
 - described, 491–493
 - JAXP, 493–499
 - XQJ, 499–500
- Java/XML data binding, 486–491
- parsing
 - with SAX, 472–473
 - with StAX, 474–477
- RSS
 - reading, 556–558
 - writing, 560–562
- Web services, consuming with
 - client application, running, 596–597, 599
 - HelloWorld service, 594–596
 - ProductService, consuming, 597–599
- writing XML (serialization)
 - described, 477–478
 - JAXP serializer, 478–479
 - serializing using StAX, 479–480
- with XML, described, 471–472
- XML tree models
 - described, 480–481
 - DOM alternatives, 481
 - DOM4J, 483–484
 - JDOM, 482–483
 - XOM, 484–486
- XPath, 293–295
- XQuery, 314

Java API for XML Processing. See **JAXP**

.java, renaming to .jws, 582–583

Java Virtual Machine (Microsoft), 108

JavaScript

data types

described, 407–408

language features, 409–410

Web page, adding to

functions, 406–407

problem, 406

tag, 406

Java/XML data binding, 486–491

JAXP (Java API for XML Processing)

described, 493

factory mechanism, 493–495

parser API, 495–496

transformation API, 496–498

validation API, 498–499

JSON data format, 425

.jws, renaming .java to, 582–583

K

Kay, Michael (*XPath 2.0 Programmer's Reference*), 300

key function, 256

L

language derived data type, 187

language features, Ajax, 409–410

Latin character sets (ISO-8859-1 through 4), 23

length primitive data type, 201

less than sign (<), avoiding error with, 15, 252

let clause, 307, 309

libraries

described, 427–428

Microsoft, 428–431

Prototype, 432–434

RSS, writing, 562

list

RELAX NG types, 215–218

XSL-FO, 112–114

list simple type, 177–178

list-style property, CSS, 83

loading DOM

described, 358–359

readyState property, 359–360

long derived data type, 187

looping (for-each element), XSLT, 252–255

M

mapping service

Flickr Photo Search Wrapper class, 699–700

GeographicLocation class, 689–690

parameters, 685–687

PhotoInformation class, 698–699

testing, 690–692

testing Term Extraction Wrapper class, 694–696

wrapping

Flickr Photo Search service, 696–698

generally, 687–689

Term Extraction service, 692–694

margin property, CSS, 82

Mark Logic Server, 349

markup

document, 5

XHTML, 67–68

mashups, 700

maxExclusive primitive data type, 201

max-height property, CSS, 83

maxInclusive primitive data type, 202

maxLength primitive data type, 202

maxOccurs attribute, 193–195

max-width property, CSS, 83

message confidentiality, WS-* security, 657

message correlation, Web services, 651

message envelope, Web services, 650

message exchange (routing), Web services, 650

message integrity, WS-* security, 657

message, SOAP, 610–611

<message>, WSDL, 641–642

metadata, 5

microformats, XHTML

described, 94–95

hCalendar, 97–98

hCard, 96–97

hReview, 98–99

RelNoFollow, 95

Rel-Tag, 95

Microsoft Access

importing XML, 792–794

saving as XML, 794–798

Microsoft ASP.NET

consuming Web page, building, 589–591

HTTPModule, 713–714

Web reference, adding, 587–589

Microsoft Excel

editing documents, 777–782

workbooks, saving as XML, 773–777

Microsoft InfoPath form

- described, 750–754
- XForms versus, 754–755
- XML, saving, 798

Microsoft Internet Explorer XMLParser, 7, 13**Microsoft Java Virtual Machine, 108****Microsoft libraries, 428–431****Microsoft .NET**

- casting XML types for compliance to, 459–460
- deserializing XML, 444–447
- RSS, writing, 559–560
- serialization of XML
 - described, 435
 - output of serialized object, changing, 438–444
 - using `XmlSerializer` class, 436–438
- XML in ASP.NET 2.0
 - described, 461
 - `XmlDataSource` server control, 462–466
- `Xml` server control, 467–469
- `XmlDataSource` server control
 - described, 462–466
 - namespace problem, 466–467
- `XmlReader` class
 - casting XML types to .NET-compliant types, 459–460
 - sample, 454–457
 - schema validation, reading with, 457–459
- `XmlWriter`
 - creating XML programmatically, 452–454
 - writing XML with, 450–452
 - `XmlTextWriter`, writing XML using, 447–449
- `XPath`, 295
- `XPathDocument` class, 460–461

Microsoft Office. See also Microsoft Access; Microsoft**Excel; Microsoft Word**

- Open XML format, 799–802
- OpenOffice, 802–805
- resources, 806

Microsoft SQL Server 2005

- retrieving XML
 - AUTO query, 326, 327–328
 - EXPLICIT query, 326, 328–329
 - FOR XML AUTO query, 331–332
 - PATH query, 329–331
 - RAW query, 326–327
 - SELECT queries, 325
- stored procedures or functions, exposing as Web services, 336–340
- storing XML
 - article schema collection, applying to table, 334
 - article schema collection, creating, 333

- bulk insert with OPENXML, 336
 - extracting attributes, 334–335
 - table, creating, 332–333
 - validating column, inserting into, 334
- XQuery, 318–319

Microsoft Trace Utility

- consuming application, modifying, 622–623
- SOAP messages, viewing, 623–624
- using, 622

Microsoft Visual Studio .NET 2003

- benefits and disadvantages, 35
- described, 33–35
- schema development, 35

Microsoft Visual Studio 2005

- basic editing, 35–37
- described, 207–208
- schema development, 38
- WPF, 809–812
- XSLT development, 38–39

Microsoft Web services test page, 575–576**Microsoft Windows character sets, 23****Microsoft Word**

- editing XML documents, 788–792
- saving documents as XML, 782–788

Microsoft WSE (Web Services Enhancements)

- client, 676–679
- described, 665–666
- exchange, result of, 679–681
- functionality, 664–665
- server
 - application, configuring, 669–673
 - building service, 676
 - credential verification class, 667–669
 - described, 666–667
 - diagnostics, configuring, 673–674
 - results, 674–676
 - version 1.0, 664
 - version 2.0, 664
 - version 3.0, 663–664

minExclusive primitive data type, 202**min-height property, CSS, 83****minLength primitive data type, 202****minOccurs attribute, XML Schemas root document, 192–193****min-width property, CSS, 83****mode attribute, 250–251****models, Java tree**

- described, 480–481
- DOM alternatives, 481
- DOM4J, 483–484

models, Java tree (continued)

- JDOM, 482–483
- XOM, 484–486
- modified simple box model hack, 91**
- `modify()` **function, 319**
- `modify` **method, 321–322**
- Mozilla XMLParser, 7**
- multicolumn layout, CSS, 92–94**
- multiple templates for single element, XSLT, 250–251**
- `mustUnderstand` **attribute, 616**

N

name classes, RELAX NG, 229–231

`Name` **derived data type, 187**

named complex types, reusability and, 181–183

NamedNodeMap interface, DOM, 366

namespaces

- RELAX NG, 229–231
- SAX callbacks, 388–389
- syntax, 29–31
- `XmlDataSource` problem, 466–467

naming attributes, 26

`NCName` **derived data type, 187**

`negativeInteger` **derived data type, 187**

nesting

- elements, 10
- WPF controls, 812–814
- XPath comments, 286–287

.NET (Microsoft)

- casting XML types for compliance to, 459–460
- deserializing XML, 444–447
- RSS, writing, 559–560
- serialization of XML
 - described, 435
 - output of serialized object, changing, 438–444
 - using `XmlSerializer` class, 436–438
- XML in ASP.NET 2.0
 - described, 461
 - `XmlDataSource` server control, 462–466
- Xml server control, 467–469
- `XmlDataSource` server control
 - described, 462–466
 - namespace problem, 466–467
- `XmlReader` class
 - casting XML types to .NET-compliant types, 459–460
 - sample, 454–457
 - schema validation, reading with, 457–459

- `XmlWriter`
 - creating XML programmatically, 452–454
 - writing XML with, 450–452
 - `XmlTextWriter`, writing XML using, 447–449
- XPath, 295
- `XPathDocument` class, 460–461

`NMTOKEN/NMTOKENS` **derived data type, 187**

`<Node>`, **620**

Node interface, DOM

- described, 361–363
- inserting node before existing node, 364
- new child node, appending, 364
- new node, creating, 363–364
- removing child node, 364
- replacing node, 365
- text values of elements, accessing, 365

NodeList interface, DOM, 365–366

nodes

- new DOM, 363–364
- XPath, 278

`nodes()` **function, 319**

`nodes` **method, 321**

`nonNegativeInteger` **derived data type, 187**

`nonPositiveInteger` **derived data type, 187**

`normalizedString` **derived data type, 187**

`NOTATION` **primitive data type, 186**

O

objects, Ajax, 416–417

Office (Microsoft). See also Microsoft Access; Microsoft Excel; Microsoft Word

- Open XML format, 799–802
- OpenOffice, 802–805
- resources, 806

1.1 faults, SOAP

- `<detail>`, 618
- `<faultcode>`, 617–618
- `<faultfactor>`, 618
- `<faultstring>`, 618
- message, 617

1.2 faults, SOAP

- `<Code>`, 619
- `<Detail>`, 621
- message, 619
- `<Node>`, 620
- `<Reason>`, 620
- `<Role>`, 620–621
- `<Subcode>`, 620

- <Text>, 620
 - <Value>, 619–620
 - online XPath Sandbox, 296–297**
 - Open XML format, 799–802**
 - OpenOffice, 802–805**
 - OPENXML, bulk insert with, 336**
 - Oracle 10g relational database**
 - described, 340
 - retrieving XML, 341–343
 - storing XML, 343–345
 - Oracle XMLParser, 7**
 - order by clause, **FLWOR, 307, 310**
 - order of elements, **RELAX NG, 220–223**
 - otherwise element, **252**
 - output of serialized object, **changing, 438–444**
 - output, **XSLT, 245–247**
 - overflow property, **CSS, 83**
 - Oxygen XML Editor 6.2**
 - basic editing, 55–57
 - benefits and disadvantages, 62
 - converting between schema types, 58–60
 - RELAX NG, 232–233
 - schema development, 57–58
 - SVG viewer, 61
 - XSLT development, 60–61
- ## P
- packages, SAX, 381**
 - padding property, CSS, 82**
 - pages**
 - adding JavaScript to Web
 - functions, 406–407
 - problem, 406
 - tag, 406
 - CSS style information, adding, 85–87
 - templates, *fo:simple-page-master*, 105–108
 - parameters, XSLT, 255–256**
 - parsing**
 - API, 495–496
 - Java
 - with SAX, 472–473
 - with StAX, 474–477
 - parsers, 6–7
 - whitespace, 13
 - XML data, 377–378
 - path expressions, XPath, 279**
 - PATH query, Microsoft SQL Server 2005, 329–331**
 - pattern primitive data type, **202**
 - #PCDATA value, limiting elements with, 149–150**
 - PDF, XSL-FO, 126–127**
 - Perl**
 - installing modules, 504
 - reading XML, 502–508
 - resources, 527
 - support for other XML formats, 509
 - writing XML, 508–509
 - Photo Search service, Flickr, 696–698**
 - Photoinformation class, 698–699**
 - PHP, XPath, 296**
 - pipe-delimited data representation, 5**
 - plus sign (+) quantifier, 153–155**
 - <portType>, **WSDL, 642–643**
 - position property, CSS, 83**
 - positiveInteger derived data type, 187**
 - predicate**
 - Altova SemanticWorks, 766–767
 - Boolean expressions in, 281
 - XPath, 279–280
 - prefix, Altova SemanticWorks, 764**
 - preserve-space, 243–244**
 - primitive data types**
 - XML schema, 185–186
 - XSD schema, 201–202
 - problems**
 - debugging
 - Stylus Studio 2006, 54
 - with *trace()*, 292–293
 - with XMLSpy, 46
 - XSLT, 274–275
 - DOM, handling, 370–371
 - parser, avoiding, 14–15
 - SAX, 392–393
 - Process Flow Contract Description, 651**
 - processing instructions, 25–26**
 - processingInstruction() method, SAX, 388**
 - ProductService Web services, consuming with Java, 597–599**
 - Professional CSS: Cascading Sheets for Web Design (Wiley Publishing), 81, 82**
 - programmer's directions**
 - described, 23–25
 - DOM interface, 369
 - XML Schemas, 205–206
 - XPath, 286–287
 - Programming Perl (Wiley Publishing), 502**
 - properties, CSS, 82–83**

protocol, SOAP

- body
 - of request, 612–613
 - of response, 613
 - described, 607–609
 - encoding data types, 621
 - envelope, 611–612
 - header
 - actor attribute, 615–616
 - consuming messages, 631–633
 - creating messages, 630–631
 - described, 614–615, 629
 - mustUnderstand attribute, 616
 - role attribute, 616
 - message, 610–611
 - 1.1 faults
 - <detail>, 618
 - <faultcode>, 617–618
 - <faultfactor>, 618
 - <faultstring>, 618
 - message, 617
 - 1.2 faults
 - <Code>, 619
 - <Detail>, 621
 - message, 619
 - <Node>, 620
 - <Reason>, 620
 - <Role>, 620–621
 - <Subcode>, 620
 - <Text>, 620
 - <Value>, 619–620
 - tracing messages
 - described, 621
 - Microsoft Trace Utility, 622–624
 - XMLSpy, 624–629
 - transport protocols, 609–610
 - Web services, extending with
 - described, 653–655
 - extensions, 655–656
 - headers, 655
 - as XML, 609
- protocols, C#, 578–579**
- PUBLIC and SYSTEM keywords, 145–146**
- purpose, XML's, 3–6**
- Python**
- described, 509–510
 - reading XML, 510–514
 - resources, 527
 - support for other formats, 516
 - writing XML, 514–516

Q

- QName **primitive data type, 186**
- query() **function, 319**
- query **method, 320**
- question mark (?) quantifier, DTDs, 155–156**
- quotation marks**
 - double ("), avoiding error with, 15
 - single ('), avoiding error with, 15

R

- RAW **query, Microsoft SQL Server 2005, 326–327**
- RDF (Resource Description Framework)**
 - Altova SemanticWorks
 - described, 762
 - graph example, 763
 - predicate, associating, 766–767
 - prefix, adding, 764
 - resource, adding, 764–766
 - triples, 767–769
 - core structure, 757–760
 - graph model, 760–762
 - XML schema, 769–771
- reading XML**
 - Perl
 - ForceArray parameter, 506
 - hash table, 507–508
 - hasref, converted to, 505–506
 - sample, 503–504
 - stream-based parsing, 506–507
 - XML:Simple library, 502–503, 505
 - Python
 - DOM support, 510–511
 - with SAX, 513–514
 - topretty XML support, 512
 - toxml output, 511–512
 - Ruby
 - output, 519–520
 - REXML prefix, 518–519
 - stream-based method, 517–518, 522–523
 - structure information, getting, 520–521
 - tree-based method, 517–518
- readyState **property, DOM, 359–360**
- Really Simple Syndication (RSS)**
 - described, 531
 - reading with Atom
 - described, 548–550
 - with Java, 556–558
 - XmlDocument, 550–552
 - XmlReader, 552–556

- resources, 563
- RSS 1.0, 540–542
- RSS 2.0, 535–540
- sample feeds by version, 532–534
- versions listed, 532
- writing with Atom
 - class libraries, 562
 - with Java, 560–562
 - with .NET, 559–560
- <Reason>, **620**
- receiving events, SAX**
 - DefaultHandler class, 383
 - described, 382–383
- regular expressions, XPath, 287–288**
- relational databases. See also Microsoft SQL Server 2005**
 - Oracle 10g
 - described, 340
 - retrieving XML, 341–343
 - storing XML, 343–345
 - XML Data Type Query and data modification
 - described, 319–320
 - exist() method, 321
 - modify method, 321–322
 - nodes method, 321
 - query method, 320
 - value method, 320–321
 - XQuery
 - described, 318
 - SQL Server 2005, 318–319
- RELAX NG**
 - annotating schemas, 231
 - attributes, defining schema with, 219–220
 - benefits of using, 211–212
 - compact schema, 236–237
 - grammar, defining, 223–224
 - merging schemas, 226–229
 - namespaces and name classes, 229–231
 - order of elements, 220–223
 - Oxygen editor, 232–233
 - resources, 238
 - schema, defining, 212–214
 - simple elements
 - declaring, 214–215
 - list types, 215–218
 - union types, 218–219
 - Trang multiformat schema converter, 233–236
 - types, reusing, 224–226
- RelNoFollow, **95**
- Rel-Tag, **95**
- rendering table, 123–124**
- replace() function, **288**
- replacing node, DOM, 365**
- REpresentational State Transfer. See REST**
- request, SOAP body, 612–613**
- #REQUIRED keyword, **160–161**
- Resource Description Framework (RDF)**
 - Altova SemanticWorks
 - described, 762
 - graph example, 763
 - predicate, associating, 766–767
 - prefix, adding, 764
 - resource, adding, 764–766
 - triples, 767–769
 - core structure, 757–760
 - graph model, 760–762
 - XML schema, 769–771
- resources**
 - Ajax, 434
 - Altova SemanticWorks, 764–766
 - Atom, 563
 - database, 350
 - Office, 806
 - Perl, 527
 - Python, 527
 - RELAX NG, 238
 - REST, 718
 - Ruby, 527
 - XHTML, 100
- response, SOAP body of, 613**
- REST HTTP module, 714–715**
- REST (REpresentational State Transfer). See also Geocode service**
 - just-enough system (GET and POST URLs)
 - Contact class, 706–707
 - Contact Manager class, 707–711
 - Data Access Layer class, 711–712
 - described, 684, 700
 - HTTP handler, 703–704
 - HTTP verbs and URLs, 701–703
 - resources, defining, 700–701
 - RestHandler, 704–705
 - mashups, 700
 - pure example
 - adding to web.config, 716–717
 - ASP.NET HTTPModule, 713–714
 - REST HTTP module, 714–715
 - URL table, 713
 - pure system, 683–684
 - resources, 718

RestHandler, 704–705

retrieving data as XML, 323–324

retrieving XML

Microsoft SQL Server 2005

AUTO query, 326, 327–328

EXPLICIT query, 326, 328–329

FOR XML AUTO query, 331–332

PATH query, 329–331

RAW query, 326–327

Oracle 10g relational database, 341–343

Xindice database, 345–348

retrieving XSLT values

calling templates, 249–250

described, 248–249

multiple templates for single element, 250–251

return **clause, FLWOR expressions, 307, 310**

reverse axis, XPath, 291–292

Rich Text Format (RTF), 67

<Role>, **620–621**

role **attribute, 616**

root element

attributeFormDefault attribute, 172–173

elementFormDefault attribute, 174

targetNamespace attribute, 174–175

version attribute, 175

xml:lang attribute, 175–176

routing, Web services, 650

RSS 1.0, 540–542

RSS 2.0, 535–540

RSS (Really Simple Syndication)

described, 531

reading with Atom

described, 548–550

with Java, 556–558

XmlDocument, 550–552

XmlReader, 552–556

resources, 563

RSS 1.0, 540–542

RSS 2.0, 535–540

sample feeds by version, 532–534

versions listed, 532

writing with Atom

class libraries, 562

with Java, 560–562

with .NET, 559–560

RTF (Rich Text Format), 67

Ruby

described, 516–517

reading XML, 517–523

resources, 527

support for other XML formats, 526

writing XML, 523–526

S

sale item table, XSL-FO working example, 127–130

saving

Word documents as XML, 782–788

workbooks as XML, 773–777

SAX (Simple API for XML)

advantages and disadvantages, 403

architecture

basic application flow, 380–381

illustrated, 380

DefaultHandler class, 385–387

described, 377, 378

element data callback, 390

errors and warnings, handling, 392–393

ignorableWhitespace() method, 390–391

installing, 378–379

namespace callbacks, 388–389

packages, listed, 381

parsing XML data, 377–378

processingInstruction() method, 388

receiving events

DefaultHandler class, 383

described, 382–383

SAXParser class, 381–382

searching in XML file, 393–396

setDocumentLocator() method, 391

skippedEntity() method, 391

startDocument() and endDocument() methods,
387

startElement() and endElement() methods,
389–390

tree-based APIs, 378

validating XML, 399–402

writing XML contents, 396–399

XMLReader interface, 382, 384–385

SAXParser class, 381–382

schema

development

Altova XMLSpy 2006, 44–45

Oxygen XML Editor 6.2, 57–58

Stylus Studio 2006, 50–51

Visual Studio 2005, 38

RELAX NG

defining, 212–214

merging, 226–229

XmlReader class, reading with validation, 457–459

script editor, Altova XMLSpy 2006, 46–47

scripts, extending XSLT with, 257–258

searching in XML file, SAX, 393–396

security, WS-* specifications

- credential exchange, 657
- message confidentiality, 657
- message integrity, 657

SELECT queries, Microsoft SQL Server 2005, 325

selected contacts, Ajax, 420

selecting XML, XQuery

- XmlCursor, 317–318
- XmlObject, 315–317

selectors, CSS, 84–85

SemanticWorks (Altova)

- described, 762
- graph example, 763
- predicate, associating, 766–767
- prefix, adding, 764
- resource, adding, 764–766
- triples, 767–769

<sequence> and <all> elements, 183–184

sequence concatenation (.) operator, 289

sequences, XPath, 282

serialization

- described, 435
- Java
 - described, 477–478
 - JAXP serializer, 478–479
 - serializing using StAX, 479–480
- output of serialized object, changing, 438–444
- Perl, 508–509
- Python, 514–516
- Ruby, 523–526
- SAX contents, 396–399
- using XmlSerializer class, 436–438
- XmlTextWriter, 447–449
- XmlWriter, 450–452

server

- Ajax query, 420–423
- Microsoft WSE
 - application, configuring, 669–673
 - building service, 676
 - credential verification class, 667–669
 - described, 666–667
 - diagnostics, configuring, 673–674
 - results, 674–676

<service>, 645

service description, 651

service implementation, Axis, 583–586

service, mapping

- Flickr Photo Search Wrapper class, 699–700
- GeographicLocation class, 689–690
- parameters, 685–687
- PhotoInformation class, 698–699
- testing, 690–692
- testing Term Extraction Wrapper class, 694–696
- wrapping
 - Flickr Photo Search service, 696–698
 - generally, 687–689
 - Term Extraction service, 692–694

services, Web

- described, 565–570
- discovering, 570–571
- framework
 - binary attachments, 650
 - digital signature, 651
 - encryption, 651
 - guaranteed message exchange, 651
 - inspection, 651
 - message correlation, 651
 - message envelope and controlled extensibility, 650
 - message exchange (routing), 650
 - Process Flow Contract Description, 651

Java

- building with, 579
- client application, running, 596–597, 599
- HelloWorld service, 594–596
- ProductService, consuming, 597–599

Microsoft WSE

- client, 676–679
- described, 665–666
- exchange, result of, 679–681
- functionality, 664–665
- version 1.0, 664
- version 2.0, 664
- version 3.0, 663–664

Microsoft WSE server

- application, configuring, 669–673
- building service, 676
- credential verification class, 667–669
- described, 666–667
- diagnostics, configuring, 673–674
- results, 674–676

model, 649–650

SOAP, extending with

- described, 653–655
- extensions, 655–656
- headers, 655

services, Web (continued)

- Tomcat
 - described, 580
 - setting up, 580–582
- Windows forms, consuming with, 591–594
- WSDL, defining
 - <binding>, 643–644
 - <definitions>, 638–640
 - described, 633–637
 - document structure, 637–638
 - <documentation>, 647–648
 - <import>, 645–647
 - <message>, 641–642
 - <portType>, 642–643
 - <service>, 645
 - <soap:binding>, 644
 - <soap:body>, 644–645
 - <soap:operation>, 644
 - <types>, 640–641
- setDocumentLocator() **method, 391**
- SGML (Standard Generalized Markup Language), 6, 68**
- short **derived data type, 187**
- Simple API for XML. See SAX**
- simple elements, RELAX NG**
 - declaring, 214–215
 - list types, 215–218
- Simple Object Access Protocol. See SOAP**
- simplified box model hack, 91**
- single quotation marks (‘), avoiding error with, 15**
- skippedEntity() **method, SAX, 391**
- slow service, building, 602–603**
- SOAP (Simple Object Access Protocol)**
 - body
 - of request, 612–613
 - of response, 613
 - described, 607–609
 - encoding data types, 621
 - envelope, 611–612
 - header
 - actor attribute, 615–616
 - consuming messages, 631–633
 - creating messages, 630–631
 - described, 614–615, 629
 - mustUnderstand attribute, 616
 - role attribute, 616
 - message, 610–611
 - 1.1 faults
 - <detail>, 618
 - <faultcode>, 617–618
 - <faultfactor>, 618
 - <faultstring>, 618
 - message, 617
 - 1.2 faults
 - <Code>, 619
 - <Detail>, 621
 - message, 619
 - <Node>, 620
 - <Reason>, 620
 - <Role>, 620–621
 - <Subcode>, 620
 - <Text>, 620
 - <Value>, 619–620
 - tracing messages
 - described, 621
 - Microsoft Trace Utility, 622–624
 - XMLSpy, 624–629
 - transport protocols, 609–610
 - Web services, extending with
 - described, 653–655
 - extensions, 655–656
 - headers, 655
 - as XML, 609
 - <soap:binding>, **644**
 - <soap:body>, **644–645**
 - <soap:operation>, **644**
- spoken languages, tags, 12–13**
- SQL Server 2005 (Microsoft)**
 - retrieving XML
 - AUTO query, 326, 327–328
 - EXPLICIT query, 326, 328–329
 - FOR XML AUTO query, 331–332
 - PATH query, 329–331
 - RAW query, 326–327
 - SELECT queries, 325
 - stored procedures or functions, exposing as Web services, 336–340
 - storing XML
 - article schema collection, applying to table, 334
 - article schema collection, creating, 333
 - bulk insert with OPENXML, 336
 - extracting attributes, 334–335
 - table, creating, 332–333
 - validating column, inserting into, 334
 - XQuery, 318–319
- standalone **attribute, 23**
- Standard Generalized Markup Language (SGML), 6, 68**
- start and end tag, 8**
- startDocument() **method, SAX, 387**

`startElement()` **method, SAX, 389–390**

stored procedures, exposing as Web services, 336–340

storing XML

databases, 324–325

Microsoft SQL Server 2005

 article schema collection, applying to table, 334

 article schema collection, creating, 333

 bulk insert with OPENXML, 336

 described, 332

 extracting attributes, 334–335

 table, creating, 332–333

 validating column, inserting into, 334

Oracle 10g relational database, 343–345

Xindice database, 348–349

streaming feeds, RSS

described, 531

reading with Atom

 described, 548–550

 with Java, 556–558

`XmlDocument`, 550–552

`XmlReader`, 552–556

resources, 563

RSS 1.0, 540–542

RSS 2.0, 535–540

sample feeds by version, 532–534

versions listed, 532

writing with Atom

 class libraries, 562

 with Java, 560–562

 with .NET, 559–560

string functions, XQuery, 312–313

`string` primitive data type, 186

`strip-space`, 243–244

style, CSS

box layout and cross-browser compatibility, 88–91

external reference with link element, 87–88

external reference with processing instructions, 87

levels, active, 81

multicolumn layout, 92–94

properties, 82–83

selectors, 84–85

style information, adding to page, 85–87

validating, 94

style information, CSS, 85–87

stylesheets, XSLT, including with `include` and `import`, 243

Stylus Studio 2006

basic editing, 47–50

benefits and disadvantages, 54–55

debugging, 54

schema development, 50–51

XSLT development, 51–53

`<Subcode>`, 620

submit protocol, XForms, 741–743

`substring()` **function, 288**

summary table, XSL-FO, 130–131

SVG viewer, Oxygen XML Editor 6.2, 61

Sybase Adaptive Server Enterprise 15, 349

syntax

attributes

 described, 26

 empty, 26–27

 naming, 26

 uniqueness, 27

`xml:lang`, 28

`xml:space`, 28–29

comments, 23–25

declaration

 encoding attribute, 22–23

 standalone attribute, 23

 version attribute, 22

document .xml file, 17–22

elements

 described, 7–8

 empty, 11

 naming conventions for elements, 8–9

 nesting, 10

 start and end tag, 8

namespaces, 29–31

parsers, 6–7

processing instructions, 25–26

tags

 CDATA sections, 16–17

 content, 12

 described, 7–8, 11–12

 entity references, 14–15

 spoken languages, 12–13

 text length, 12

 whitespace, 13–14

well-formed documents, 6

SYSTEM and PUBLIC keywords, 145–146

system-property function, 257

T

table

cells, XSL-FO, 122–123

Microsoft SQL Server 2005, creating, 332–333

rendering XSL-FO, 123–124

XSL-FO, 114–116

tags

- CDATA sections, 16–17
- content, 12
- described, 7–8, 11–12
- entity references, 14–15
- JavaScript, adding to, 406
- spoken languages, 12–13
- text length, 12
- whitespace, 13–14
- TakeLongTime() **WebMethod**, 603–605
- targetNamespace **attribute**, 174–175
- templates, XSLT**, 247
- 10g relational database, Oracle**
 - described, 340
 - retrieving XML, 341–343
 - storing XML, 343–345
- Term Extraction service, Geocode service**, 692–694
- Term Extraction Wrapper class, testing**, 694–696
- testing**
 - Geocode service, 690–692
 - Term Extraction Wrapper class, 694–696
 - WebMethod, 576–578
- <Text>, 620
- text **element**, 248
- Text interface, DOM**, 369
- text length tag**, 12
- text values, DOM**, 365
- text-align property, CSS**, 82
- text-decoration property, CSS**, 82
- thin- or thick-client, XAML**, 808–809
- Tidy**
 - resource, 100
 - XHTML application, 73–77
 - XHTML UI application, 78–80
- time **primitive data type**, 186
- token **derived data type**, 187
- Tomcat Web service**
 - described, 580
 - setting up, 580–582
- totalDigits **primitive data type**, 202
- trace() **function**, 288
- Trace Utility (Microsoft)**
 - consuming application, modifying, 622–623
 - SOAP messages, viewing, 623–624
 - using, 622
- tracing SOAP messages**
 - described, 621
 - XMLSpy, 624–629
- Trang multiformat schema converter**, 233–236
- transactions, Web services**, 651

transformation

- DOM, 371–373
- JAXP, 496–498

transport protocols, SOAP, 609–610**tree models, Java**

- described, 480–481
- DOM alternatives, 481
- DOM4J, 483–484
- JDOM, 482–483
- XOM, 484–486

tree structure, XPath, 278–279**tree-based APIs, SAX**, 378**triples, Altova SemanticWorks**, 767–769**type element**, 184–188

<types>, 640–641

types, reusing RELAX NG, 224–226**U****union (|) concatenation operator**, 289**union simple type**, 179**union types, RELAX NG**, 218–219**uniqueness, attribute**, 27unordered() **function, XPath**, 288–289

- unsignedByte/unsignedInt/unsignedLong/
unsignedShort **derived data types**, 187

updateXml(), 344

URL table, 713**US-ASCII (English) character set**, 23

- use **attribute, XML Schemas root document**, 200

use cases, XQuery, 304**user input forms**

- Microsoft InfoPath

- described, 750–754

- XForms versus, 754–755

- XForms

- actions logic, 744–745

- appearance, changing control, 733

- children, common control, 731–732

- clients, 755–756

- controls, 727–730

- described, 722–727

- events logic, 743–744

- grouping controls, 733–737

- instance data, binding, 737–741

- Microsoft InfoPath versus, 754–755

- sample, 745–750

- submit protocol, 741–743

- XHTML, 721–722

user-defined functions, XQuery, 313–314

UTF-8 (Compressed Unicode) character set, 23

UTF-16 (Compressed UCS) character set, 23

V

validation

CSS, 94

DOM, 373–375

DTDs, 166–167

JAXP, 498–499

SAX, 399–402

XHTML, 69–72

<Value>, **619–620**

value() **function, 319**

value **method, 320–321**

value-of **element, 248**

values, restricting in XML Schemas root document, 200–202

variables, XSLT, 255–256

version **attribute, 22, 175**

Vi editor, 62

Visual Studio .NET 2003 (Microsoft)

benefits and disadvantages, 35

described, 33–35

schema development, 35

Visual Studio 2005 (Microsoft)

basic editing, 35–37

described, 207–208

schema development, 38

WPF, 809–812

XSLT development, 38–39

W

Web browser

Ajax code, 426–427

CSS compatibility, 88–91

XPath, 297–298

XSLT, executing, 262–263

Web language

handler, 703–704

verbs and URLs, 701–703

Web page, adding JavaScript to

functions, 406–407

problem, 406

tag, 406

Web reference, ASP.NET, 587–589

Web Service Deployment Descriptor (WSDD) file, creating, 586–587

Web services. *See also* WS-* specifications

asynchronous consumption

described, 601–602

slow service, building, 602–603

TakeLongTime() WebMethod, 603–605

Axis

described, 580

publishing, 582–587

setting up, 580–582

C#

building described, 571–574

consuming with ASP.NET, 587–591

Microsoft Web services test page, 575–576

protocols, altering, 578–579

WebMethod, testing, 576–578

caching, 599–600

composition

data, representing and communicating, 568

described, 567–570

discovering, 570–571

described, 565–570

discovering, 570–571

framework

binary attachments, 650

digital signature, 651

encryption, 651

guaranteed message exchange, 651

inspection, 651

message correlation, 651

message envelope and controlled extensibility, 650

message exchange (routing), 650

Process Flow Contract Description, 651

Java

building with, 579

client application, running, 596–597, 599

HelloWorld service, 594–596

ProductService, consuming, 597–599

Microsoft test page, 575–576

Microsoft WSE

client, 676–679

described, 665–666

exchange, result of, 679–681

functionality, 664–665

version 1.0, 664

version 2.0, 664

version 3.0, 663–664

Microsoft WSE server

application, configuring, 669–673

building service, 676

credential verification class, 667–669

Web services, Microsoft WSE server (continued)

- described, 666–667
- diagnostics, configuring, 673–674
- results, 674–676
- model, 649–650
- SOAP, extending with
 - described, 653–655
 - extensions, 655–656
 - headers, 655
- Tomcat
 - described, 580
 - setting up, 580–582
- Windows forms, consuming with, 591–594
- WSDL, defining
 - <binding>, 643–644
 - <definitions>, 638–640
 - described, 633–637
 - document structure, 637–638
 - <documentation>, 647–648
 - <import>, 645–647
 - <message>, 641–642
 - <portType>, 642–643
 - <service>, 645
 - <soap:binding>, 644
 - <soap:body>, 644–645
 - <soap:operation>, 644
 - <types>, 640–641

Web Services Enhancements. See WSE

web.config, adding REST to, 716–717

WebMethod, testing, 576–578

Western European (windows-1252) character set, 23

when **element**, 252

where **clause**, **FLWOR**, 307, 309–310

whitespace

- tags, 13–14
- XSLT, controlling with `strip-space`, `preserve-space`, and `decimal-format`, 243–244

whitespace **primitive data type**, 202

width property, CSS, 83

Wiley Publishing

- Beginning Python*, 510
- Professional CSS: Cascading Sheets for Web Design*, 81, 82
- Programming Perl*, 502

Windows forms, 591–594

Windows Presentation Foundation. See WPF

windows-1250 (Central European) character set, 23

windows-1251 (Cyrillic) character set, 23

windows-1252 (Western European) character set, 23

windows-1253 (Greek) character set, 23

Word (Microsoft)

- editing XML documents, 788–792
- saving documents as XML, 782–788

word order

- attributes
 - described, 26
 - empty, 26–27
 - naming, 26
 - uniqueness, 27
 - `xml:lang`, 28
 - `xml:space`, 28–29
- comments, 23–25
- declaration
 - encoding attribute, 22–23
 - standalone attribute, 23
 - version attribute, 22
- document .xml file, 17–22
- elements
 - described, 7–8
 - empty, 11
 - naming conventions for elements, 8–9
 - nesting, 10
 - start and end tag, 8
- namespaces, 29–31
- parsers, 6–7
- processing instructions, 25–26
- tags
 - CDATA sections, 16–17
 - content, 12
 - described, 7–8, 11–12
 - entity references, 14–15
 - spoken languages, 12–13
 - text length, 12
 - whitespace, 13–14
 - well-formed documents, 6

workbooks, Excel, saving as XML, 773–777

WPF (Windows Presentation Foundation)

- application, creating, 815
- described, 809–812
- document
 - building, 815–820
 - viewing, 820–823
- image, adding to document, 823–825
- nesting controls, 812–814
- within Visual Studio 2005, 809–812
- XPS file, saving document as, 825–829

wrapping Geocode service, 687–689

writing XML

Java

- described, 477–478
- JAXP serializer, 478–479
- serializing using StAX, 479–480

Perl, 508–509

Python, 514–516

Ruby, 523–526

SAX contents, 396–399

XmlTxtWriter, 447–449

XmlWriter, 450–452

WS-* specifications

- addressing, 657–658
- attachments, 658
- coordination, 658
- core specifications, 659–663
- described, 656
- security
 - credential exchange, 657
 - message confidentiality, 657
 - message integrity, 657
- WS-MetadataExchange, 658

WSD (Web Service Deployment Descriptor) file, creating, 586–587**WSDL Web services, defining**

- <binding>, 643–644
- <definitions>, 638–640
- described, 633–637
- document structure, 637–638
- <documentation>, 647–648
- <import>, 645–647
- <message>, 641–642
- <portType>, 642–643
- <service>, 645
- <soap:binding>, 644
- <soap:body>, 644–645
- <soap:operation>, 644
- <types>, 640–641

WSE (Web Services Enhancements)

- client, 676–679
- described, 665–666
- exchange, result of, 679–681
- functionality, 664–665
- server
 - application, configuring, 669–673
 - building service, 676
 - credential verification class, 667–669
 - described, 666–667
 - diagnostics, configuring, 673–674
 - results, 674–676

- version 1.0, 664
- version 2.0, 664
- version 3.0, 663–664

WS-I.org, 652–653**WS-MetadataExchange, 658****W3C validation service, 72****X****XAML (eXtensible Application Markup Language)**

- described, 807–808
- thin- of thick-client, 808–809
- WPF
 - application, creating, 815
 - described, 809–812
 - document, building, 815–820
 - document, viewing, 820–823
 - image, adding to document, 823–825
 - nesting controls, 812–814
 - within Visual Studio 2005, 809–812
 - XPS file, saving document as, 825–829

Xerces (Apache), 7**XForms**

- actions logic, 744–745
- appearance, changing control, 733
- children, common control, 731–732
- clients, 755–756
- controls, 727–730
- described, 722–727
- events logic, 743–744
- grouping controls, 733–737
- instance data, binding, 737–741
- Microsoft InfoPath versus, 754–755
- sample, 745–750
- submit protocol, 741–743

XHTML (eXtensible HyperText Markup Language)

- element, 81
- forms, 721–722
- markup, evolution of, 67–68
- microformats
 - described, 94–95
 - RelNoFollow, 95
 - Rel-Tag, 95
- microformats, compound
 - hCalendar, 97–98
 - hCard, 96–97
 - hReview, 98–99
- resources, 100
- Tidy application, 73–77
- Tidy UI application, 78–80

XHTML (eXtensible HyperText Markup Language)

(continued)

validating, 69–72
versions, 68
W3C validation service, 72

Xindice database

retrieving XML, 345–348
storing XML, 348–349

XMetal editor, 62

XML (eXtensible Markup Language)

in ASP.NET 2.0
 described, 461
 XmlDataSource server control, 462–466
Data Type Query and data modification
 described, 319–320
 exist() method, 321
 modify method, 321–322
 nodes method, 321
 query method, 320
 value method, 320–321
declaration, XML Schemas root document, 172
elements, reusing in DTDs, 152–153
schemas
 Altova XMLSpy, 209–210
 <annotation>, 206
 commenting
 DTDs, problem with, 169–171
 Microsoft Visual Studio 2005, 207–208
 RDF, 769–771
 standard, 205–206
server control, 467–469
SOAP, 609
syntaxes, converting between, 272–274
tools, creating DTDs with, 166
Xindice database
 retrieving XML, 345–348
 storing XML, 348–349
XPath editor, 299

.xml file, 17–22

XML Schemas root document

declaring elements
 atomic simple type, 176–177
 complex types, 180–181
 list simple type, 177–178
 named complex types, reusability and, 181–183
 <sequence> and <all> elements, 183–184
 type element, 184–188
 union simple type, 179

defining attributes
 attribute groups, 202–203
 default values, 199–200
 described, 198–199
 use attribute, 200
 values, restricting, 200–202
described, 171–172
element restrictions
 cardinality, 192
 default values, 195–196
 fixed values, 196–197
 maxOccurs attribute, 193–195
 minOccurs attribute, 192–193
 null values, 197–198
<group> element, 189–192
<import>, 203–204
<include>, 204
root element
 attributeFormDefault attribute, 172–173
 described, 172
 elementFormDefault attribute, 174
 targetNamespace attribute, 174–175
 version attribute, 175
 xml:lang attribute, 175–176
XML declaration, 172

XML tree models

described, 480–481
DOM alternatives, 481
DOM4J, 483–484
JDOM, 482–483
XOM, 484–486

XmlCursor, invoking execQuery() from, 317–318

XmlDataSource server control

described, 462–466
namespace problem, 466–467

XmlDocument, 550–552

XMLHttpRequest, 410–415

xml:lang attribute, 28, 175–176

XmlObject, invoking execQuery() from, 315–317

XMLParser (Expat), 7

XMLParser for Java (IBM), 7

XMLParser (Mozilla), 7

XMLParser (Oracle), 7

XmlReader

described, 552–556
class
 casting XML types to .NET-compliant types, 459–460
 sample, 454–457
 schema validation, reading with, 457–459
interface, 382, 384–385

XMLReader **interface**, 382, 384–385

xml:sequence(), 344

XmlSerializer **class**, 436–438

xml:space **attribute**, 28–29

XMLSpy 2006 (Altova)

benefits and disadvantages, 47

described, 39–44

schema development, 44–45

script editor, 46–47

Spy debugging, 624–629

XML Schemas, 209–210

XSLT development, 46

XmlTextWriter, writing XML using, 447–449

XmlWriter

creating XML programmatically, 452–454

writing XML with, 450–452

XmlTextWriter, writing XML using, 447–449

XPath

axes, 281–282

in browser, 297–298

comments and nested comments, 286–287

debugging with trace(), 292–293

described, 277

Eclipse, 300

expressions

FLWOR expressions versus, 310–311

tuning, 284

function calls in path expressions, 285–286

//h1[1] different than (//h1)[1], 289–291

IntelliJ, 300

in Java, 293–295

many faces of a document, 283

on .NET, 295

nodes, 278

online XPath Sandbox, 296–297

path expressions, 279

in PHP, 296

predicates, 279–280

regular expressions, 287–288

reverse axis, 291–292

sequences, 282

tree structure, 278–279

union (|) and sequence concatenation (.) operators,
289

unordered() function, 288–289

when A != B is different from not(A = B),
282–283

XML editors, 299

XPath 2.0 Programmer's Reference (Kay), 300

XPathDocument **class**, 460–461

XPS file, saving WPF document as, 825–829

XQJ (XQuery API for Java), 499–500

XQuery

advantages, 304

described, 303–304

enclosed expressions, 306

expression, structure of, 304–305

FLWOR expressions

described, 306–307

for clause, 308–309

let clause, 309

order by clause, 310

return clause, 310

sample, 307–308

where clause, 309–310

XPath expressions versus, 310–311

functions

aggregate (count, avg, max, min, and sum), 312

built-in, described, 311–312

doc(), 312

string, 312–313

user-defined, 313–314

in Java, 314

relational databases

described, 318

SQL Server 2005, 318–319

XML Data Type Query and data modification,
319–322

sample, 305–306

selecting XML

XmlCursor, invoking execQuery() from, 317–318

XmlObject, invoking execQuery() from, 315–317

use cases, 304

XQuery API for Java (XQJ), 499–500

XSL-FO (eXtensible Stylesheet Language-Formatting Objects)

basic formatting

described, 111–112

lists, 112–114

tables, 114–116

composition, 102–103

described, 101

Hello World example, 109–110

Microsoft Java Virtual Machine, 108

overview, 103–105

page templates, fo:simple-page-master,
105–108

XSL-FO (eXtensible Stylesheet Language-Formatting Objects) (continued)

XSL-FO (eXtensible Stylesheet Language-Formatting Objects) (continued)

working example

- described, 117–119
- formatting table cells, 122–123
- header, 120–122
- high-level layout for invoice, 119–120
- PDF, 126–127
- rendering table, 123–124
- sale item table, 127–130
- summary table with total values for invoice, 130–131
- XSLT code, 125–126

XSLT (XSL Transformations)

- changes ahead in version 2.0, 263–268
- code, XSL-FO, 125–126
- conditional processing (*if* element), 251–252
- debugging, 274–275
- development
 - Altova XMLSpy 2006, 46
 - Oxygen XML Editor 6.2, 60–61
 - Stylus Studio 2006, 51–53
 - Visual Studio 2005, 38–39
- executing
 - in browser, 262–263
 - at command line, 258–259
 - via code, 259–261

- extending with script, 257–258
- functions, built-in, 256–257
- looping (*for-each* element), 252–255
- output, generating
 - HTML, 268–272
 - XML syntaxes, converting between, 272–274
- required syntax items
 - output, controlling, 245–247
 - stylesheets, including with *include* and *import*, 243
 - top-level elements, 242–243
 - whitespace and formatting, controlling with
 - strip-space*, *preserve-space*, and *decimal-format*, 243–244
- resources, 276
- retrieving values
 - calling templates, 249–250
 - described, 248–249
 - multiple templates for single element, 250–251
- sample, 241–242
- templates, 247
- variables and parameters, 255–256