

Index

COPYRIGHTED MATERIAL

Index

SYMBOLS

& (ampersand) symbol, 8
' (apostrophe) symbol, 8
@ (at) expression, 160
// (backslash, double) expression, 160
/ (backslash) expression, 160
[] (brackets) expression, 160
> (greater than) symbol, 8
< (less than) symbol, 8
.. (period, double) expression, 160
. (period) expression, 160
| (pipe) expression, 160
" (quotation) symbol, 8

A

Actor property, 427, 431
Add method, 108
AddExtensionObject method, 190
AddParam method, 190
administration and management, 35–37
ADO.NET
 DataColumn objects, controlling the rendering
 behavior of, 225
 DataSet schemas, 218–222
 DataTable, 243–245
 DiffGram, 223–225
 executing FOR XML queries from, 290–297
 loading XML into a DataSet, 214–218
 modifying table and column names, 225–227

Multiple Active Result Sets (MARS) in,
 323–326
 nesting XML output from a DataSet,
 228–230
 .NET Framework, 42, 56
 overview, 214
 ReadXml method, 214
 string from a DataSet, getting XML as,
 227–228
 transforming DataSet to XML, 222–230
 XML data type columns and, 303–316
 XmlDataDocument class, 235–241
 XmlReadMode enumeration, 215
 XmlWriteMode enumeration, 222
 XPathNavigator class and XmlData
 Document relationship between, 242–243
advanced operations, 207–209
advanced serialization
 collections, serializing, 392–394
 namespaces used to generate qualified names,
 389–392
 overview, 384–385
 XmlAttributeOverrides object, 387–389
 XmlAttributes class, 385–387
 XmlSerializerNamespaces object,
 389–392
 xsd namespace declaration, removing,
 391–392
 xsi namespace declaration, removing,
 391–392

airline reservation system (case study)

- architecture of system, 336
- ASP.NET 2.0 script callback, 367–369
- business processes, 336–337
- confirmation page implementation, 374
- data access layer methods, implementation of, 343–349
- database design, 337–342
- HTML, transforming XML to, 371–373
- limitations, 337
- login process, 352–355
- logout process, 360–362
- new user registration process, 356–360
- overview, 335–337
- search flight process, 362–367
- testing, 374
- Web site implementation, 349–374
- XML formatted search, 369–370
- XSD validation, 370–371

ALTER XML SCHEMA **statement**, 317

ambiguity and namespaces, 11

ampersand (&) symbol, 8

annotations used with typed DataSets, 234–235

`<anonymousIdentification>` **configuration section**, 468

`AnonymousIdentificationSection` **class**, 469

App_Code subdirectory, 25

`AppendChild` **method**, 137, 164

`AppendChildElement` **method**, 164

application settings, 473–474

applying a style sheet to an XML document, 179–186

apostrophe (') symbol, 8

`/appsettingbaseurl: [url]` **command line switch**, 418

`AppSettings` **property**, 472

`/appsettingurlkey: [key]` **command line switch**, 418

architecture

- of airline reservation system (case study), 336
- of Atlas, 280–281
- for configuration, 57
- of ShoppingAssistant (case study), 498–499

`<asp:AccessDataSource>`, 26

`<asp:DetailsView>`, 27

`<asp:FormView>`, 27

`<asp:GridView>`, 27

`<asp:Login>`, 28

`<asp:LoginName>`, 28

`<asp:LoginStatus>`, 28

`<asp:LoginView>`, 28

`<asp:Menu>`, 27

ASP.NET 2.0. See also airline reservation system (case study); ASP.NET configuration

- administration and management, 35–37

- callback feature of, 272–279

- developer productivity, 23–35

- Master Pages, 24–25, 514

- new controls, 26–29

- overview, 23

- precompilation, 36–37

- security features, 517

- speed and performance, 37–39

- themes, 29–30

- Web Parts Framework and, 30–33

ASP.NET configuration

- `anonymousIdentification` configuration section, 468

- `AnonymousIdentificationSection` class, 469

- ASP.NET 1.x way of accessing configuration sections, 467

- `AuthenticationSection` class, 469

- `AuthorizationSection` class, 469

- built-in configuration management tools, 491–495

- `CacheSection` class, 469

- classes, configuration sections, 468–471

- `CompilationSection` class, 469

- `connectionStrings` configuration section, 468

- custom configuration section, 487–491

- `CustomErrorsSection` class, 469

- decryption, 478–481

- encryption, 478–481

- enumerating configuration sections, 482

- expression builders, 476

- `GlobalizationSection` class, 469

- `healthMonitoring` configuration section, 468

- HealthMonitoringSection class, 469
 - hierarchical configuration model, 466–467
 - HostingEnvironmentSection class, 469
 - HttpCookiesSection class, 469
 - HttpHandlersSection class, 470
 - HttpModulesSection class, 470
 - HttpRuntimeSection class, 470
 - IdentitySection class, 470
 - inheritance, 466–467
 - MachineKeySection class, 470
 - membership configuration section, 468
 - MembershipSection class, 470
 - new configuration sections in ASP.NET 2.0, 468–471
 - overview, 466
 - PagesSection class, 470
 - ProcessModelSection class, 470
 - profile configuration section, 468
 - ProfileSection class, 470
 - raw XML used to retrieve configuration settings, 485–486
 - reading configuration sections, 483–486
 - roleManager configuration section, 468
 - SessionPageStateSection class, 470
 - SessionStateSection class, 471
 - siteMap configuration section, 468
 - SiteMapSection class, 471
 - strongly typed API used to retrieve configuration settings, 484–485
 - TraceSection class, 471
 - TrustSection class, 471
 - urlMappings configuration section, 468
 - UrlMappingsSection class, 471
 - WebConfigurationManager class, 471–478
 - webParts configuration section, 468
 - WebPartsSection class, 471
 - XhtmlConformanceSection class, 471
 - ASP.NET MMC snapin, 492**
 - ASP.NET page, implementation of, 204–207**
 - ASP.NET 1.x way of accessing configuration sections, 467**
 - ASPNET_REGIIS tool, 495**
 - ASPNET_REGSQL tool, 495**
 - `<asp:ObjectDataSource>`, 26
 - `<asp>PasswordRecovery>`, 28
 - `<asp:SiteMapDataSource>`, 26
 - `<asp:SqlDataSource>`, 26
 - `<asp:TreeView>`, 27
 - `<asp:XmlDataSource>`, 26
 - async pages, 446–447**
 - asynchronous invocation of Web service**
 - async pages, 446–447
 - from a browser using IE Web service behavior, 448–453
 - callService method, 448
 - from a client application, 443–447
 - createUseOptions method, 448
 - how it works, 448–450, 455–457
 - methods, 454–457
 - .NET client application, asynchronous programming in, 444–446
 - overview, 443–444, 448, 454
 - ShoppingAssistant (case study), 530
 - useService method, 448
 - asynchronous invocation of Windows service, 526–531**
 - at (@) expression, 160**
 - Atlas**
 - architecture of, 280–281
 - features of, 281
 - overview, 280
 - retrieving data from a Web service using, 281–284
 - AttributeCount property, 64**
 - attributes**
 - appending, 151
 - elements compared, 6
 - overview, 6–7
 - reading, 69, 73–76
 - writing, 86
 - Attributes property, 136**
 - AuthenticationSection class, 469**
 - AuthorizationSection class, 469**
 - AUTO mode, 289**
- ## B
- `// (backslash, double) expression, 160`
 - `/ (backslash) expression, 160`

benefits of configuration system used by ASP.NET, 58

binary serialization, 51

brackets ([]) expression, 160

`BufferResponse` **property, 416**

building XML Web service, 412–416

built-in configuration management tools

ASP.NET MMC snapin, 492

ASP.NET_REGIIS tool, 495

ASP.NET_REGSQL tool, 495

command line tools, 495

GUI tools, 491–494

overview, 491

Web Site Administration Tool, 493–494

business processes

airline reservation system (case study), 336–337

ShoppingAssistant (case study), 499–500

C

`CacheDuration` **property, 252, 416**

`CacheExpirationPolicy` **property, 252**

`CacheSection` **class, 469**

caching

with `DataSource` controls, 38

overview, 37–38, 262

page output caching, 262–263

Substitution control, using, 39

using SQL cache invalidation, 38

`XmlDataSource` control used to implement, 263–265

`callService` **method, 448**

`CanDeserialize` **method, 379**

Cascading Style Sheets (CSS), 17

categories listing process in ShoppingAssistant (case study), 520–521

CDATA references, 7–8

`ChainStream` **method, 436**

changes, persisting, 151–152

`CheckCharacters` **property, 80, 90**

`CheckValidity` **method, 164**

`ChildNodes` **property, 136**

classes, configuration sections, 468–471

`Clear` **method, 190**

client

SOAP extensions, 443

using XML schema on, 317–323

XML data type columns and, 307–311

client callback method, implementing, 278–279

client-side script, 277–278

client-side XML

callback feature of ASP.NET 2.0, 272–279

overview, 272

`Clone` **method, 137, 164**

`Close` **method, 65, 84**

CLR integration with SQL Server 2005, 304

`Code` **property, 427**

code refactoring, 34

collection classes, 136

collection serialization

custom collections, serializing, 392–394

overview, 392

command line tools, 495

communication between client and XML Web service, 54–55

`CompilationSection` **class, 469**

`Compile` **method, 108, 164**

complex data types

custom object returned from Web service, 424–426

data binding and, 423–424, 425–426

`DataSet` object returned from Web service, 420–424

exception handling, 427–430

returning, 420–430

`SoapException` class, 427–430

components of XML document

attributes, 6–7

elements, 5–6

overview, 4–5

UTF-8 characters, 4

UTF-16 characters, 4

conditions, evaluating, 183

configuration. See also ASP.NET configuration

architecture for, 57

benefits of configuration system used by ASP.NET, 58

`machine.config` file, 57

overview, 57
 settings, support for accessing, 58
 Web.config file, 57
 of XmlReader object, 79–81
ConformanceLevel property, 80
connection strings, 474–476
 <connectionStrings> **configuration section, 468**
 ConnectionStrings **property, 472**
 Contains **method, 108**
content, reading, 69–70
controls
 data controls, 26–28
 new ASP.NET 2.0 controls, 26–29
 security controls, 28
 validation groups, 29
count expression, 161
 Count **property, 108**
 Create **method, 65, 84**
 CreateAttribute **method, 137, 164**
 CreateAttributes **method, 164**
 CreateCDATASection **method, 137**
 CreateComment **method, 138**
 CreateDocumentFragment **method, 138**
 CreateDocumentType **method, 138**
 CreateElement **method, 138**
 CreateEntityReference **method, 138**
 CreateNode **method, 138**
 CreateProcessingInstruction **method, 138**
 CreateReader **method, 309**
 CreateSignificantWhitespace **method, 138**
 CreateTextNode **method, 138**
 CreateUseOptions **method, 448**
 CreateWhitespace **method, 138**
 CreateXmlDeclaration **method, 138**
CSS (Cascading Style Sheets), 17
 current **function, 179**
custom attributes, 439
custom collections, serializing, 392–394
custom configuration section, 487–491
custom object returned from Web service, 424–426
Custom Web Part, 31
 CustomErrorsSection **class, 469**
customization-themes, 29

D

data binding, 423–426, 454–255
data controls
 <asp:AccessDataSource>, 26
 <asp:DetailsView>, 27
 <asp:FormView>, 27
 <asp:GridView>, 27
 <asp:Menu>, 27
 <asp:ObjectDataSource>, 26
 <asp:SiteMapDataSource>, 26
 <asp:SqlDataSource>, 26
 <asp:TreeView>, 27
 <asp:XmlDataSource>, 26
 overview, 26
 Data **property, 252**
data, writing, 87
database design
 airline reservation system (case study), 337–342
 ShoppingAssistant (case study), 501–503
 DataColumn **objects, controlling the rendering behavior of, 225**
 DataFile **property, 252**
 DataReader, **245**
 DataSet **class, 56**
 DataSet **object, 236, 326–329, 420–424**
 DataSet **property, 235**
 DataSet **schemas**
 FillSchema method, 220–221
 inference rules, 218–219
 InferXmlSchema method, 222
 overview, 218
 ReadXmlSchema method, 221–222
 supplied schemas, 219–222
 typed DataSets, 230–235
 writing, 230
DataSource controls, 38
 DataTable
 DataReader, 245
 methods, 243
 overview, 243
 ReadXml method, 243
 ReadXmlSchema method, 243
 serializing, 244
 WriteXml method, 243
 WriteXmlSchema method, 243

datatypes, 16

declaring namespaces, 10–11

decompression in Web service client, enabling, 463

decryption, 478–481

default namespace, 10

defining namespaces, 10

DeleteSelf **method, 164**

Depth **property, 64**

description component of XML Web service, 53

Description **property, 416**

deserialization used to map SQL Server data, 399–403

Deserialize **method, 379, 394–403**

design goals for XML support in .NET Framework, 41–42

Detail **property, 427**

developer productivity

Master Pages, 24–25, 514

.NET Framework and, 42

new ASP.NET 2.0 controls and, 26–29

overview, 23–24

personalization framework and, 33

reusable components, creating and sharing, 25
themes and, 29–30

Visual Studio 2005 improvements and, 33–35

Web Parts Framework and, 30–33

DidUnderstand **property, 431**

DiffGram, **223–225**

directories, 53

discovery component of XML Web service, 53

doctypes, 3, 12–13

document classes, 135–136

document **function, 179**

document, reading a, 68

Document Type Definition (DTD), 3, 12–13

DocumentContent **property, 265**

DocumentElement **property, 136**

DocumentSource **property, 265**

DocumentType **property, 136**

DOM object model

attributes, appending, 151

changes, persisting, 151–152

collection classes, 136

creating an XML document from scratch,
152–156

described, 18

document classes, 135–136

events raised by XmlDocument class, handling,
157–158

node data, changing, 157

nodes, creating and appending, 149–151

nodes, deleting, 157

overview, 131–132

processing, 132

programmatically creating XML documents,
149–158

programming with, 134

tree, XML document loaded into DOM, 132–134

validating XmlDocument object, 171

validation, 110–111

XmlDocument class, 136–148

XmlDocumentFragment class, 159

XPath support, 159–170

XPathNavigator object, 168–170

DOM subtree, selecting, 148

/domain: [domain] **command line switch, 418**

DTD (Document Type Definition), 3, 12–13

DTD validation

described, 100

overview, 115

syntax for, 115–117

using, 119

validating an XML document against a, 117–119

when to use, 119

E

element-available **function, 179**

elements

attributes compared, 6

overview, 5–6

reading, 68

writing, 86

of XSLT, 176–179

embedding scripts inside style sheet, 196–197

/enableDataBinding **command line
switch, 418**

- EnableSession property, 416**
- EncodeMustUnderstand12 property, 431**
- Encoding property, 90**
- encryption, 478–481**
- ending a document, 85–86**
- entity references, 8**
- enumerating configuration sections, 482**
- EOF property, 64**
- event handling, 157–158, 396–399**
- exception handling, 73, 106, 408–409, 427–430**
- exist method, 302**
- EXPLICIT mode, 289**
- expression builders, 476**
- Extensible Hypertext Markup Language (XHTML), 8**
- Extensible Markup Language (XML)**
 - advantages of, 20–21
 - CDATA references, 7–8
 - components of XML document, 4–7
 - entity references, 8
 - namespaces, 8–11
 - overview, 1–2
 - PCDATA references, 7–8
 - self-describing data, 2–3
 - terminology, 3–4
 - valid documents, 3–4
 - well-formed documents, 3
- Extensible Stylesheet Language Transformations (XSLT)**
 - advanced operations, 207–209
 - applying a style sheet to an XML document, 179–186
 - ASP.NET page, implementation of, 204–207
 - conditions, evaluating, 183
 - current function, 179
 - document function, 179
 - element-available function, 179
 - elements of, 176–179
 - embedding scripts inside style sheet, 196–197
 - example, complete, 199–207
 - extension objects, 194–196
 - format-number function, 179
 - function-available function, 179
 - functions, 179
 - generate-id function, 179
 - how it works, 174–175
 - key function, 179
 - multiple conditions, evaluating, 184
 - need for, 175–176
 - .NET classes used for XSL transformations, 186–192
 - nodeset as parameter to style sheet, passing a, 207–208
 - overview, 9, 17, 174
 - parameters, 186, 190–192
 - passing parameters to a style sheet, 190–192
 - sorting an XML file using a style sheet, 182–183
 - statically applying a style sheet, 180–186
 - system-property function, 179
 - unparsed-identity-uri function, 179
 - used to transform XML data, 49–50
 - user-defined functions, 193–197
 - variables, 185–186
 - XmlDataDocument class, 187
 - XmlDocument class, 187
 - XmlPathDocument class, 187
 - XmlReader class, 187
 - XmlResolver class, 208–209
 - XmlWriter class, 187
 - xsl:apply-imports element, 177
 - xsl:apply-templates element, 177, 185
 - xsl:attribute element, 177
 - xsl:attribute-set element, 177
 - xsl:call-template element, 177
 - xsl:choose element, 177
 - xsl:comment element, 177
 - XslCompiledTransform class, 187
 - XslCompiledTransform object, 188
 - xsl:copy element, 177
 - xsl:copy-of element, 177
 - xsl:decimal-format element, 177
 - xsl:element element, 177
 - xsl:fallback element, 177
 - xsl:for-each element, 177
 - xsl:if element, 177
 - xsl:import element, 177
 - xsl:include element, 177
 - xsl:key element, 178

Extensible Stylesheet Language

Transformations (XSLT) (continued)

xsl:message element, 178
xsl:namespace-alias element, 178
xsl:number element, 178
xsl:otherwise element, 178
xsl:output element, 178
xsl:param element, 178
xsl:preserve-space element, 178
xsl:processing-instruction element, 178
xsl:sort element, 178
xsl:strip-space element, 178
xsl:stylesheet element, 178
XsltArgumentList class, 187, 190
XsltCompileException class, 187
xsl:template element, 178
XsltException class, 187
xsl:text element, 178
xsl:transform element, 178
XsltSettings class, 187, 198–199
xsl:value-of element, 178
xsl:variable element, 178
xsl:when element, 178
xsl:with-param element, 178

extension objects, 194–196

extracting XML elements from DataSet,
238–240

F

/fields **command line switch, 418**

file system support, 35

FillSchema **method, 220–221**

FirstChild **property, 136**

FOR XML **clause**

ADO.NET, executing FOR XML queries from,
290–297

asynchronous execution of, 293–297

AUTO mode, 289

EXPLICIT mode, 289

overview, 289

RAW mode, 289

results, assigning, 290

XML data type, integration with, 289–290

format-number **function, 179**

forward-only access, 44–45

FromMappings **method, 379**

FromTypes **method, 379**

FTP support, 35

functionality, 538

function-available **function, 179**

functions, 179

G

generate-id **function, 179**

GenerateSerializer **method, 379**

Generic Web Part, 31

generics

collections, serializing generics, 406

overview, 403–406

XML serialization and, 403–406

GetAttribute **method, 65, 164**

GetCallbackEventReference **method,
277–278**

GetElementById **method, 138, 146**

GetElementFromRow **method, 235**

GetElementsByTagName **method, 138, 146**

GetEnumerator **method, 138**

GetExtensionObject **method, 190**

GetInitializer **method, 436**

GetNamespace **method, 164**

GetNamespaceIncScope **method, 164**

GetParam **method, 190**

GetRowFromElement **method, 235**

GetWebApplicationSection **method, 472**

GetXmlSerializerAssemblyName
method, 379

GetXsdType **method, 309**

GlobalAttributes **property, 108**

GlobalElements **property, 108**

GlobalizationSection **class, 469**

GlobalTypes **property, 108**

greater than (>) **symbol, 8**

GridView **control used for data binding,
255–257**

GUI tools, 491–494

H

HasAttributes property, 64
HasChildNodes property, 136
HasValue property, 64
 <healthMonitoring> configuration section, 468
HealthMonitoringSection class, 469
hierarchical configuration model, 466–467
hierarchical data controls
 overview, 248
 site navigation and, 248–251
 SiteMapDataSource control, 249–251
 XmlDataSource control, 251–261
HostingEnvironmentSection class, 469
HTML (Hypertext Markup Language)
 overview, 1
 source preservation, 34
HTTP (HyperText Transfer Protocol), 1
HttpCookiesSection class, 469
HttpHandlersSection class, 470
HttpModulesSection class, 470
HttpRuntimeSection class, 470

I

IdentitySection class, 470
IgnoreComments property, 80
IgnoreProcessingInstructions property, 80
IgnoreWhitespace property, 80
images, writing, 95–96
ImportNode method, 138
Indent property, 90
IndentChars property, 90
indexing XML columns, 302–303
inference rules, 218–219
InferXmlSchema method, 222
inheritance, 466–467
Initialize method, 436
inline XML data, 257–259
inline XSD schemas, 112–115
InnerText property, 136
InnerXml property, 136
in-place precompilation, 37
in-process access to XML data type columns, 304–306

InsertAfter method, 138, 164
InsertBefore method, 138, 164
InsertElementAfter method, 164
InsertElementBefore method, 164
inserting data into an XML column, 301
invoking Web service with, 450–453
IsCompiled property, 108
IsDescendant method, 164
IsEmptyElement property, 64
IsNull property, 309
IsReadOnly property, 137
IsStartElement method, 65
Item property, 137
IXmlSerializable interface, 457–459
IXPathNavigable interface, 47

K

key function, 179

L

/language: [CS|VB|JS] **command line switch**, 417
LastChild property, 137
less than (<) symbol, 8
Load method, 138, 235
loading a DataSet through XmlDataDocument class, 236–238
loading XML documents, 140–142
loading XML into a DataSet, 214–218
LoadXml method, 138
local IIS support, 35
LocalName property, 64, 137
login process
 airline reservation system (case study), 352–355
 ShoppingAssistant (case study), 515–519
logout process
 airline reservation system (case study), 360–362
 ShoppingAssistant (case study), 519–520

M

machine.config file, 57, 462–463
MachineKeySection class, 470
managed code, 307

MARS (Multiple Active Result Sets), 323–326

Master Pages, 24–25, 514

Mathematical Markup Language (MathML), 8

<membership> configuration section, 468

MembershipSection **class, 470**

Message **property, 427**

MessageName **property, 416**

methods

DataTable, 243

WebConfigurationManager class, 472

XML data type, 302

XmlSerializer class, 379

Microsoft XML Core Services (MSXML), 45–46

modify method, 302

modifying table and column names, 225–227

monitoring capabilities, 536–538

MoveToAttribute **method, 65, 164**

MoveToChild **method, 164**

MoveToContent **method, 65**

MoveToElement **method, 65**

MoveToFirst **method, 164**

MoveToFirstAttribute **method, 65, 164**

MoveToFirstChild **method, 164**

MoveToFirstNamespace **method, 165**

MoveToFollowing **method, 165**

MoveToId **method, 165**

MoveToNamespace **method, 165**

MoveToNext **method, 165**

MoveToNextAttribute **method, 65, 165**

MoveToNextNamespace **method, 165**

MoveToParent **method, 165**

MoveToPrevious **method, 165**

MoveToRoot **method, 165**

MSXML (Microsoft XML Core Services), 45–46

Multiple Active Result Sets (MARS), 323–326

multiple conditions, evaluating, 184

multiple namespaces, 11

MustUnderstand **property, 431**

N

Name **property, 64, 137**

/namespace: [namespace] **command line switch, 418**

namespaces

ambiguity and, 11

declaring, 10–11

default namespace, 10

defining, 10

multiple namespaces, 11

overview, 8–10

prefixed namespace, 10

reuse and, 11

role of, 11

support, 92–95

unique identifier, 10

used to generate qualified names, 389–392

for XML Web service, 55

xmlns, 10

NamespaceURI **property, 65, 137**

nested DataList controls, 260–261

nesting XML output from a DataSet, 228–230

.NET classes used for XSL transformations, 186–192

.NET client application, asynchronous programming in, 444–446

.NET Framework

ADO.NET and, 42, 56

design goals for XML support in, 41–42

developer productivity and, 42

overview, 41

parsing and, 43–46

performance of, 42

Schema Object Model (SOM), 47–49

serialization and, 51–52

standards compliance, 42

usability, 42

validation and, 49

writing XML and, 46

XML namespaces, 42–43

XML Web service and, 52–55

XPath support, 46–47

XSLT used to transform XML data, 49–50

.NET Framework 2.0, 463–464

new configuration sections in ASP.NET 2.0, 468–471

new controls, 26–29

new features in SQL Server 2005, 288

NewLineChars **property, 90**
 NewLineOnAttributes **property, 90**
 NextSibling **property, 137**
node data, changing, 157
 Node **property, 427**
 NodeChanged **event, 157**
 NodeChanging **event, 157**
 NodeInserted **event, 157**
 NodeInserting **event, 157**
 NodeRemoved **event, 157**
 NodeRemoving **event, 157**

nodes

creating and appending, 149–151
 deleting, 157
 finding, 145–148

nodes **method, 302**

nodeset as parameter to style sheet, 207–208

NodeType **property, 65, 67, 137**
 /nologo **command line switch, 417**

non-validating parsers, 44

Null **property, 309**

NVARCHAR, **312–313**

O

object graphs, 382–384

objects, serializing, 380–382

OmitXmlDeclaration **property, 90**

OpenMachineConfiguration **method, 472**

OpenMappedMachineConfiguration
method, 472

OpenMappedWebConfiguration **method, 472**

OpenWebConfiguration **method, 472**

OPENXML **function, 329–331**

/order **command line switch, 418**

OuterXml **property, 137**

/out: [filename] **command line switch, 418**

output, formatting, 90–92

P

page output caching, 262–263

PagesSection **class, 470**

parameters

passed to XML data type columns, 311–316
 style sheet, passing parameters to, 190–192,
 207–208

XSLT, 186

ParentNode **property, 137**

/parsableerrors **command line switch, 418**

parsing

an XML document, 143–145

forward-only access, 44–45

MSXML 6.0, 45–46

.NET Framework, 43–46

non-validating parsers, 44

overview, 43–44

random access via DOM, 45

reader, choosing the right XML, 45

SAX and, 45

validating parsers, 44

/password: [password] **command line
 switch, 418**

PCDATA references, 7–8

performance of .NET Framework, 42

period, double (..) expression, 160

period (.) expression, 160

personalization framework, 33

pipe (|) expression, 160

position expression, 161

precompilation, 36–37

**predefined sections, retrieving configuration from,
 473–478**

Prefix **property, 65, 137**

prefixed namespace, 10

pregenerating serialization assemblies, 407–408

PrependChild **method, 138, 165**

PrependChildElement **method, 165**

PreserveWhitespace **property, 137**

PreviousSibling **property, 137**

primary XML index, 303

processing data in XML file, 76–79

ProcessMessage **method, 436**

ProcessModelSection **class, 470**

**product details listing process in ShoppingAssis-
 tant (case study), 523–525**

products listing process in ShoppingAssistant (case study), 521–523

<profile> **configuration section**, 468

ProfileSection **class**, 470

programmatically creating a schema, 125–129

programmatically creating XML documents, 149–158

ProhibitDtd **property**, 80, 103

/protocol: [SOAP |HttpPost |HttpGet] **command line switch**, 418

proxy class

creating, 416–420

Visual Studio, using Add Web Reference option in, 419–420

WSDL utility used to generate proxy code, 416–419

/proxy: [url] **command line switch**, 418

/proxydomain: [domain] **command line switch**, 418

/proxypassword: [password] **command line switch**, 418

/proxyusername: [username] **command line switch**, 418

Q

query **method**, 302

quotation (“) **symbol**, 8

R

random access via DOM, 45

RAW **mode**, 289

raw XML used to retrieve configuration settings, 485–486

Read **method**, 65

ReadContentAs **method**, 65

ReadElementContentAs **method**, 66

ReadEndElement **method**, 66

reader, choosing the right XML, 45

reader classes

overview, 62–63

XmlNodeReader class, 63

XmlReader class, 62, 63–82

XmlTextReader class, 62

XmlValidatingReader class, 62

reading a schema from a file, 123–125

reading configuration sections, 483–486

ReadInnerXml **method**, 66

ReadNode **method**, 138

ReadOuterXml **method**, 66

ReadState **property**, 65

ReadSubTree **method**, 165

ReadToDescendant **method**, 66

ReadToFollowing **method**, 66

ReadToNextSibling **method**, 66

ReadValueChunk **method**, 66

ReadXml **method**, 214, 243

ReadXmlSchema **method**, 221–222, 243

Relay **property**, 431

Remove **method**, 108

RemoveAll **method**, 138

RemoveChild **method**, 138

RemoveExtensionObject **method**, 190

RemoveParam **method**, 190

ReplaceChild **method**, 138

ReplaceSelf **method**, 165

reporting data collection in ShoppingAssistant (case study), 532–538

Reprocess **method**, 108

results, assigning, 290

reusable components, creating and sharing, 25

reuse and namespaces, 11

role of namespaces, 11

Role **property**, 431

<roleManager> **configuration section**, 468

RSS 2.0 XML file, 269–270

S

Save **method**, 139

SAX (Simple API for XML)

overview, 19

and parsing, 45

XML Reader, 82

Scalable Vector Graphics (SVG), 8

schema importer extensions

creating, 461–462

machine.config file, 462–463

overview, 460

SchemaImporterExtension class, 460–463

- SchemaImporterExtensionCollection
 - class, 460
 - using, 460–463
- Schema Object Model (SOM)**
 - .NET Framework, 47–49
 - overview, 122–123
 - programmatically creating a schema, 125–129
 - reading a schema from a file, 123–125
 - XML file, programmatically inferring XSD schema from, 129–130
- SchemaImporterExtension **class, 460–463**
- SchemaImporterExtensionCollection
 - class, 460**
 - schemas. See also DataSet schemas; XSD schemas**
 - datatypes, 16
 - described, 14
 - example of, 15–16
 - inline XSD schemas, 112–115
 - usage scenarios, 16–17
- Schemas **property, 80, 103, 137**
- Script Callback**
 - client callback method, implementing, 278–279
 - client-side script for, generating, 277–278
 - GetCallbackEventReference method, 277–278
 - overview, 272–273
 - retrieving XML data dynamically using, 273–276
 - server-side event for, implementing, 276–277
- secondary XML index, 303**
- security controls, 28**
- security features, 517**
- Select **method, 165**
- SelectAncestors **method, 165**
- SelectChildren **method, 165**
- SelectDescendants **method, 165**
- SelectNodes **method, 139, 146**
- SelectSingleNode **method, 139, 147, 165**
- self-describing data, 2–3**
- serialization, 51–52, 244**
- Serialize **method, 379**
- /serverinterface **command line switch, 418**
- server-side event, **276–277**
- SessionPageStateSection **class, 470**
- SessionStateSection **class, 471**
- SET SHOWPLAN_XML **command, 332–333**
- Settings **property, 65, 83**
- settings, support for accessing configuration, 58**
- SetTypedValue **method, 165**
- SetValue **method, 165**
- SGML (Standard Generalized Markup Language), 2**
- /sharetypes **command line switch, 417**
- ShoppingAssistant (case study)**
 - architecture of, 498–499
 - asynchronous invocation of Web service, 530
 - asynchronous invocation of Windows service, 526–531
 - business processes, 499–500
 - categories listing process, 520–521
 - database design, 501–503
 - functionality of, 538
 - login process, 515–519
 - logout process, 519–520
 - monitoring capabilities, adding, 536–538
 - overview, 497–498
 - product details listing process, 523–525
 - products listing process, 521–523
 - reporting data collection, 532–538
 - testing the application, 525
 - Visual Studio 2005 and, 511–513
 - Web application implementation, 513–525
 - Web service implementation, 503–511
 - XML files, modifying Web pages to use, 531–532
- Simple API for XML (SAX)**
 - overview, 19
 - and parsing, 45
 - XML Reader, 82
- Simple Object Access Protocol (SOAP), 3, 54–55**
- site navigation**
 - hierarchical data controls, 248–251
 - overview, 248
 - SiteMapDataSource control, 249–251
 - siteMapNode element, 248–249
 - web.sitemap file, 249–250
- <siteMap> **configuration section, 468**
- SiteMapDataSource **control, 249–251**
- siteMapNode **element, 248–249**
- SiteMapSection **class, 471**

Smart Task, 34

SMIL (Synchronized Multimedia Integration Language), 8

SOAP extensions

- on client, 443
- creating SoapExtension class, 436–439
- custom attributes, creating, 439
- overview, 436
- using, 436–443
- Web method, applying to, 440–443

SOAP headers

- Actor property, 431
- DidUnderstand property, 431
- EncodeMustUnderstand12 property, 431
- implementing, 432–433
- MustUnderstand property, 431
- overview, 431–432
- Relay property, 431
- Role property, 431
- using, 431–436
- Web service, processing SOAP header from, 433–436

SOAP (Simple Object Access Protocol), 3, 54–55

SoapException class

- Actor property, 427
- advantages of, 430
- Code property, 427
- Detail property, 427
- Message property, 427
- Node property, 427
- overview, 427–430
- SubCode property, 427

SOM (Schema Object Model)

- .NET Framework, 47–49
- overview, 122–123
- programmatically creating a schema, 125–129
- reading a schema from a file, 123–125
- XML file, programmatically inferring XSD schema from, 129–130

source code editing, 33

speed and performance, 37–39

SQL cache invalidation, 38

SQL Server 2005. See also airline reservation system (case study)

- client, using XML schema on, 317–323
- CLR integration, 304
- FOR XML clause, 289–297
- new features in, 288
- OPENXML function, 329–331
- validation using XSD schemas, performing client side, 320–323
- XML data type, 288, 298–303
- XSD schemas retrieved from client application, 318–320

SqlContext class, 305

SqlPipe class, 305

SqlXml object

- passing, 313–316
- retrieval as, 309–311

square brackets ([]) expression, 160

Standard Generalized Markup Language (SGML), 2

standards compliance, 42

starting a document, 85

starts-with expression, 160

statically applying a style sheet, 180–186

string

- from a DataSet, getting XML as, 227–228
- retrieval as a, 307–308

strongly typed API used to retrieve configuration settings, 484–485

style sheets

- applying a style sheet to an XML document, 179–186
- debugging, 209–211
- embedding scripts inside style sheet, 196–197
- implementation of, 200–203
- nodesets and, 207–208
- passing parameters to a style sheet, 190–192
- sorting an XML file using a style sheet, 182–183
- statically applying a style sheet, 180–186

style sheet-themes, 29

SubCode property, 427

Substitution control, 39

supplied schemas, 219–222

Supports **method**, 139
SVG (Scalable Vector Graphics), 8
Synchronized Multimedia Integration Language (SMIL), 8
synchronous command execution, 293–294
syntax for DTD validation, 115–117
system-property **function**, 179
System.Web.Services **namespace**, 55
System.Web.Services.Configuration **namespace**, 55
System.Web.Services.Description **namespace**, 55
System.Web.Services.Discovery **namespace**, 55
System.Web.Services.Protocols **namespace**, 55
System.Xml **namespace**, 43
System.Xml.Schema **namespace**, 43
System.Xml.Serialization **namespace**, 43
System.Xml.XPath **namespace**, 43
System.Xml.Xsl **namespace**, 43

T

Tag Navigator, 34
testing the application, 525
themes
characteristics of, 30
creating, 30
customization-themes, 29
overview, 29
style sheet-themes, 29
TraceSection **class**, 471
TransactionOption **property**, 416
Transact-SQL, 307
Transform **property**, 252, 265
TransformArgumentList **property**, 252, 265
TransformFile **property**, 252
transforming DataSet to XML, 222–230
TransformSource **property**, 265
tree, XML document loaded into DOM, 132–134
TrustSection **class**, 471
type sharing across proxies, 463–464

typed DataSets
annotations used with, 234–235
generating, 231–233
overview, 230–231
using, 233–234
typed XML column, 298–299

U

Uniform Resource Numbers (URNs), 9
unique identifier, 10
Universal Resource Indicators (URIs), 9
Universal Resource Locators (URLs), 9
UnknownAttribute **event**, 396
UnknownElement **event**, 396
UnknownNode **event**, 396
unparsed-identity-uri **function**, 179
UnreferencedObject **event**, 396
untyped XML column, 298–299
URIs (Universal Resource Indicators), 9
<url or path> **command line switch**, 417
<urlMappings> **configuration section**, 468
UrlMappingsSection **class**, 471
URLs (Universal Resource Locators), 9
URNs (Uniform Resource Numbers), 9
usability, 42
user-defined functions, 193–197
/username: [username] **command line switch**, 418
useService **method**, 448
UTF-8 characters, 4
UTF-16 characters, 4

V

valid documents, 3–4
Validate **method**, 139
validating parsers, 44
validating XmlDocument object, 171
validation
DTD validation, 100, 115–119
inline XSD schemas, 112–115
and .NET Framework, 49, 100–101
overview, 100

validation (continued)

validation (continued)

- Schema Object Model (SOM), 122–130
- Visual Studio 2005 and, 119–122
- XDR, 100
- XML DOM validation, 110–111
- XSD schemas, 101–106, 320–323
- validation groups, 29**
- ValidationEventHandler **event, 103**
- ValidationEventHandler **property, 103**
- ValidationFlags **property, 80, 103**
- ValidationType **property, 80, 103**
- value **method, 302**
- Value **property, 65, 137, 309**
- ValueAs **method, 165**
- ValueType **property, 65**
- variables, 185–186**
- /verbose **command line switch, 418**
- Visual Studio 2005**
 - code refactoring, 34
 - file system support, 35
 - FTP support, 35
 - HTML source preservation, 34
 - improvements in, 33–35
 - local IIS support, 35
 - overview, 33
 - ShoppingAssistant (case study), 511–513
 - Smart Task, 34
 - source code editing, 33
 - Tag Navigator, 34
 - targeting specific browsers and HTML
 - validation, 34
 - using Add Web Reference option in, 419–420
 - and validation, 119–122
 - Web projects, creating, 34–35

W

- W3C SOAP 1.2, support for, 464**
- Web application implementation, 513–525**
- Web method, 440–443**
- Web Parts Framework**
 - creating a simple Web Part, 31–33
 - Custom Web Part, 31

- Generic Web Part, 31
- overview, 30–31
- Web projects, creating, 34–35**
- Web Site Administration Tool, 493–494**
- Web.config **file, 57, 477–478**
- WebConfigurationManager **class**
 - application settings, 473–474
 - AppSettings property, 472
 - connection strings, 474–476
 - ConnectionStrings property, 472
 - GetWebApplicationSection method, 472
 - methods, 472
 - OpenMachineConfiguration method, 472
 - OpenMappedMachineConfiguration
 - method, 472
 - OpenMappedWebConfiguration method, 472
 - OpenWebConfiguration method, 472
 - overview, 471–473
 - predefined sections, retrieving configuration
 - from, 473–478
 - properties, 472
 - web.config file, 477–478
- WebMethod **attribute, 415, 415–416**
- <webParts> **configuration section, 468**
- WebPartsSection **class, 471**
- WebService **attribute, 414–415**
- WebService **class, 412–414**
- WebServiceBinding **attribute, 415**
- web.sitemap **file, 249–250**
- well-formed documents, 3**
- wire formats, 53–54**
- WriteAttributes **method, 84**
- WriteAttributesString **method, 84**
- WriteBase64 **method, 84**
- WriteCData **method, 84, 87**
- WriteCharEntity **method, 84**
- WriteChars **method, 84**
- WriteComment **method, 84, 87**
- WriteContentTo **method, 139**
- WriteDocType **method, 84**
- WriteElementString **method, 84**
- WriteEndAttribute **method, 84**
- WriteEndDocument **method, 84**

WriteEndElement **method, 84**
 WriteEntityRef **method, 84**
 WriteFullEndElement **method, 84**
 WriteName **method, 84**
 WriteNode **method, 84**
 WriteProcessingInstruction **method, 84**
 WriteQualifiedName **method, 85**

writer classes

overview, 62–63
 XmlTextWriter class, 63
 XmlWriter class, 63, 83–96

WriteRaw **method, 85, 87**
 WriteStartAttribute **method, 85**
 WriteStartDocument **method, 85**
 WriteStartElement **method, 85**
 WriteState **property, 83, 87**
 WriteString **method, 85, 87**
 WriteSubTree **method, 165**
 WriteTo **method, 139**
 WriteValue **method, 85**
 WriteWhitespace **method, 85**
 WriteXml **method, 243**
 WriteXmlSchema **method, 243**

writing

attributes, 86
 data, 87
 DataSet schemas, 230
 elements, 86
 images, 95–96
 XML file, 46, 87–90

WSDL utility used to generate proxy code, 416–419

X

XDR (XML Data Reduced), 13, 100

XForms, 8

XHTML (Extensible Hypertext Markup Language), 8

XhtmlConformanceSection **class, 471**

XLink, 8, 20

Xml **control**

DocumentContent property, 265
 DocumentSource property, 265

overview, 265–269
 programmatically transforming data using, 269–271
 RSS 2.0 XML file, 269–270
 Transform property, 265
 TransformArgumentList property, 265
 TransformSource property, 265

XML Data Reduced (XDR), 13, 100

XML data type

and DataSet object, 326–329
 exist method, 302
 indexing XML columns, 302–303
 inserting data into an XML column, 301
 integration, 289–290
 methods, 302
 modify method, 302
 nodes method, 302
 overview, 298
 primary XML index, 303
 query method, 302
 secondary XML index, 303
 typed XML column, 298–299
 untyped XML column, 298–299
 value method, 302
 XML schema collections, 299–301

XML data type columns

ADO.NET and, 303–316
 client, retrieval from, 307–311
 in-process access to, 304–306
 NVARCHAR as parameter type, passing, 312–313
 parameters passed to, 311–316
 SqlXml object, passing, 313–316
 SqlXml object, retrieval as a, 309–311
 string, retrieval as a, 307–308

XML (Extensible Markup Language)

advantages of, 20–21
 CDATA references, 7–8
 components of XML document, 4–7
 entity references, 8
 namespaces, 8–11
 overview, 1–2
 PCDATA references, 7–8
 self-describing data, 2–3

XML (Extensible Markup Language) (continued)

XML (Extensible Markup Language) (continued)

- terminology, 3–4
- valid documents, 3–4
- well-formed documents, 3

XML files

- created from XSD schemas, 121–122
- programmatically inferring XSD schema from, 129–130
- reading, 70–76
- ShoppingAssistant (case study), 531–532
- writing, 87–90

XML namespaces

- in .NET Framework, 42–43
- overview, 42
- `System.Xml` namespace, 43
- `System.Xml.Schema` namespace, 43
- `System.Xml.Serialization` namespace, 43
- `System.Xml.XPath` namespace, 43
- `System.Xml.Xsl` namespace, 43

XML schema collections, 299–301

XML Schema Definition Tool, 384

XML Schema Definition (XSD), 3, 14–15, 100

XML Schema Working Group, 17

XML serialization

- advanced serialization, 384–394
- deserialization used to map SQL Server data, 399–403
- `Deserialize` method, 394–403
- exception handling, 408–409
- generics and, 403–406
- overview, 51–52, 377–378
- pregenerating serialization assemblies, 407–408
- XML Serializer Generator, 407–408
- `XmlSerializer` class, 379–384, 396–399

XML Serializer Generator, 407–408

XML technologies, 12–20

XML Web service

- asynchronous invocation of Web services from a browser using IE Web service behavior, 448–453
- asynchronous invocation of Web services from a client application, 443–447
- asynchronous Web service methods, 454–457
- `BufferResponse` property, 416

- building, 412–416
- `CacheDuration` property, 416
- communication between client and, 54–55
- complex data types, returning, 420–430
- components of, 53–54
- decompression in Web service client, enabling, 463
- description component of, 53
- `Description` property, 416
- directories, 53
- discovery component of, 53
- `EnableSession` property, 416
- implementation, 503–511
- `IXmlSerializable` interface, 457–459
- `MessageName` property, 416
- namespaces for, 55
- in .NET Framework, 52–55, 463–464
- overview, 52–53, 412
- processing SOAP header from, 433–436
- proxy class, creating, 416–420
- schema importer extensions, using, 460–463
- SOAP extensions, using, 436–443
- SOAP headers, using, 431–436
- SOAP protocol and, 54–55
- `TransactionOption` property, 416
- type sharing across proxies, 463–464
- W3C SOAP 1.2, support for, 464
- `WebMethod` attribute, 415, 415–416
- `WebService` attribute, 414–415
- `WebService` class, 412–414
- `WebServiceBinding` attribute, 415
- wire formats, 53–54
- `XmlAnyAttribute` **property**, 385
- `XmlAnyElements` **property**, 385
- `XmlArray` **property**, 385
- `XmlArrayItems` **property**, 385
- `XmlAttribute` **property**, 385
- `XmlAttributeOverrides` **object**, 387–389
- `XmlAttributes` **class**
- `XmlChoiceIdentifier` property, 385
- `XmlDefaultValue` property, 385
- `XmlElements` property, 385
- `XmlEnum` property, 386
- `XmlIgnore` property, 386

- Xmlns property, 386
- XmlRoot property, 386
- XmlText property, 386
- XmlType property, 386
- XmlDataDocument **class**
 - DataSet object and, 236–238
 - DataSet property, 235
 - extracting XML elements from DataSet, 238–240
 - GetElementFromRow method, 235
 - GetRowFromElement method, 235
 - Load method, 235
 - merging data with, 240–241
 - overview, 235–236
- XmlDataSource **control**
 - CacheDuration property, 252
 - CacheExpirationPolicy property, 252
 - caching, used to implement, 263–265
 - data binding with, 254–255
 - Data property, 252
 - DataFile property, 252
 - GridView control used for data binding, 255–257
 - inline XML data as input for, 257–259
 - nested DataList controls with, 260–261
 - overview, 251–254
 - properties of, 252
 - Transform property, 252
 - TransformArgumentList property, 252
 - TransformFile property, 252
 - XPath property, 252
 - XSL transformations with, 259
- XmlDefaultValue **property, 385**
- XmlDocument **class**
 - AppendChild method, 137
 - Attributes property, 136
 - ChildNodes property, 136
 - Clone method, 137
 - CreateAttribute method, 137
 - CreateCDATASection method, 137
 - CreateComment method, 138
 - CreateDocumentFragment method, 138
 - CreateDocumentType method, 138
 - CreateElement method, 138
 - CreateEntityReference method, 138
 - CreateNode method, 138
 - CreateProcessingInstruction method, 138
 - CreateSignificantWhitespace method, 138
 - CreateTextNode method, 138
 - CreateWhitespace method, 138
 - CreateXmlDeclaration method, 138
 - creating an XmlDocument, 139–140
 - DocumentElement property, 136
 - DocumentType property, 136
 - DOM subtree, selecting, 148
 - FirstChild property, 136
 - GetElementById method, 138, 146
 - GetElementsByTagName method, 138, 146
 - GetEnumerator method, 138
 - HasChildNodes property, 136
 - ImportNode method, 138
 - InnerText property, 136
 - InnerXml property, 136
 - InsertAfter method, 138
 - InsertBefore method, 138
 - IsReadOnly property, 137
 - Item property, 137
 - LastChild property, 137
 - Load method, 138
 - loading XML documents, 140–142
 - LoadXml method, 138
 - LocalName property, 137
 - Name property, 137
 - NamespaceURI property, 137
 - NextSibling property, 137
 - nodes, finding, 145–148
 - NodeType property, 137
 - OuterXml property, 137
 - overview, 136–139
 - ParentNode property, 137
 - parsing an XML document, 143–145
 - Prefix property, 137
 - PrependChild method, 138
 - PreserveWhitespace property, 137
 - PreviousSibling property, 137
 - ReadNode method, 138

XmlDocument class (continued)

XmlDocument **class (continued)**

- RemoveAll method, 138
- RemoveChild method, 138
- ReplaceChild method, 138
- Save method, 139
- Schemas property, 137
- SelectNodes method, 139, 146
- SelectSingleNode method, 139, 147
- Supports method, 139
- Validate method, 139
- Value property, 137
- WriteContentTo method, 139
- WriteTo method, 139
- XmlResolver property, 137

XmlDocumentFragment class, 159

XmlElement property, 385

XmlAttribute property, 386

XmlIgnore property, 386

XmlLang property, 83

XmlNodeReader class, 63

XmlNodeType enumeration, 67

Xmlns property, 386

XmlPathDocument class, 187

XmlReader class

- AttributeCount property, 64
- attributes, reading, 69, 73–76
- Close method, 65
- configuration of XmlReader object, 79–81
- content, reading, 69–70
- Create method, 65
- Depth property, 64
- described, 62
- document, reading a, 68
- elements, reading, 68
- EOF property, 64
- exceptions, handling, 73
- GetAttribute method, 65
- HasAttributes property, 64
- HasValue property, 64
- IsEmptyElement property, 64
- IsStartElement method, 65
- LocalName property, 64
- MoveToAttribute method, 65
- MoveToContent method, 65

- MoveToElement method, 65
- MoveToFirstAttribute method, 65
- MoveToNextAttribute method, 65
- Name property, 64
- NamespaceURI property, 65
- NodeType property, 65, 67
- overview, 63–64
- Prefix property, 65
- processing data in XML file, 76–79
- Read method, 65
- ReadContentAs method, 65
- ReadElementContentAs method, 66
- ReadEndElement method, 66
- ReadInnerXml method, 66
- ReadOuterXml method, 66
- ReadState property, 65
- ReadToDescendant method, 66
- ReadToFollowing method, 66
- ReadToNextSibling method, 66
- ReadValueChunk method, 66
- Settings property, 65
- using, 64–67
- Value property, 65
- ValueType property, 65
- XML file, reading, 70–76
- XmlNodeType enumeration, 67

XmlReaderSettings class, 80, 103

XmlReadMode enumeration, 215

XmlResolver class, 208–209

XmlResolver property, 80, 137

XmlRoot property, 386

XmlSchema class, 107

XmlSchemaSet class, 107–109

XmlSerializer class

- CanDeserialize method, 379
- Deserialize method, 379
- event handling by, 396–399
- FromMappings method, 379
- FromTypes method, 379
- GenerateSerializer method, 379
- GetXmlSerializerAssemblyName method, 379
- methods, 379
- object graphs and, 382–384

- objects, serializing, 380–382
- overview, 379–380
- Serialize method, 379
- UnknownAttribute event, 396
- UnknownElement event, 396
- UnknownNode event, 396
- UnreferencedObject event, 396
- XML Schema Definition Tool, 384
- XmlSerializerNamespaces **object, 389–392**
- XmlSpace **property, 83**
- XmlText **property, 386**
- XmlTextReader **class, 62**
- XmlTextWriter **class, 63**
- XmlType **property, 386**
- XmlValidatingReader **class, 62**
- XmlWriteMode **enumeration, 222**
- XmlWriter **class**
 - attributes, writing, 86
 - Close method, 84
 - Create method, 84
 - data, writing, 87
 - elements, writing, 86
 - ending a document, 85–86
 - images, writing, 95–96
 - namespace support, 92–95
 - output, formatting, 90–92
 - overview, 83–85
 - Settings property, 83
 - starting a document, 85
 - WriteAttributes method, 84
 - WriteAttributesString method, 84
 - WriteBase64 method, 84
 - WriteCData method, 84, 87
 - WriteCharEntity method, 84
 - WriteChars method, 84
 - WriteComment method, 84, 87
 - WriteDocType method, 84
 - WriteElementString method, 84
 - WriteEndAttribute method, 84
 - WriteEndDocument method, 84
 - WriteEndElement method, 84
 - WriteEntityRef method, 84
 - WriteFullEndElement method, 84
 - WriteName method, 84
 - WriteNode method, 84
 - WriteProcessingInstruction method, 84
 - WriteQualifiedName method, 85
 - WriteRaw method, 85, 87
 - WriteStartAttribute method, 85
 - WriteStartDocument method, 85
 - WriteStartElement method, 85
 - WriteState property, 83, 87
 - WriteString method, 85, 87
 - WriteValue method, 85
 - WriteWhitespace method, 85
 - XML file, writing, 87–90
 - XmlLang property, 83
 - XmlSpace property, 83
- XmlWriterSettings **class, 90**
- XPath**
 - at (@) expression, 160
 - backslash, double (//) expression, 160
 - backslash (/) expression, 160
 - brackets ([]) expression, 160
 - count expression, 161
 - overview, 18–19
 - period, double (..) expression, 160
 - period (.) expression, 160
 - pipe (|) expression, 160
 - position expression, 161
 - starts-with expression, 160
 - support, 159–170
 - XPathNavigator class, 163–167
- XPath **property, 252**
- XpathDocument **class, 46**
- XpathException **class, 46**
- XpathExpression **class, 46**
- XpathItem **class, 47**
- XPathNavigator **class**
 - AppendChild method, 164
 - AppendChildElement method, 164
 - CheckValidity method, 164
 - Clone method, 164
 - Compile method, 164
 - CreateAttribute method, 164
 - CreateAttributes method, 164
 - DeleteSelf method, 164
 - described, 47

XPathNavigator class (continued)

XPathNavigator **class (continued)**

GetAttribute method, 164
GetNamespace method, 164
GetNamespaceIncScope method, 164
InsertAfter method, 164
InsertBefore method, 164
InsertElementAfter method, 164
InsertElementBefore method, 164
IsDescendant method, 164
MoveToAttribute method, 164
MoveToChild method, 164
MoveToFirst method, 164
MoveToFirstAttribute method, 164
MoveToFirstChild method, 164
MoveToFirstNamespace method, 165
MoveToFollowing method, 165
MoveToId method, 165
MoveToNamespace method, 165
MoveToNext method, 165
MoveToNextAttribute method, 165
MoveToNextNamespace method, 165
MoveToParent method, 165
MoveToPrevious method, 165
MoveToRoot method, 165
PrependChild method, 165
PrependChildElement method, 165
ReadSubTree method, 165
ReplaceSelf method, 165
Select method, 165
SelectAncestors method, 165
SelectChildren method, 165
SelectDescendants method, 165
SelectSingleNode method, 165
SetTypedValue method, 165
SetValue method, 165
ValueAs method, 165
WriteSubTree method, 165
and XmlDocument relationship between, 242–243
XPathNavigator **object, 168–170**
XPathNodeIterator **class, 47**
XPointer, 20
XQuery, 20, 302

xsd namespace declaration, removing, 391–392

XSD schemas

described, 100
example, 103–106
exceptions, handling, 106
overview, 101–102
retrieved from client application, 318–320
steps for validating an XML document, 102–103
ValidationEventHandler event, 103
XML file, created from, 121–122
XmlReaderSettings class, 103

XSD (XML Schema Definition), 3, 14–15, 100

xsi namespace declaration, removing, 391–392
xsl:apply-imports **element, 177**
xsl:apply-templates **element, 177, 185**
xsl:attribute **element, 177**
xsl:attribute-set **element, 177**
xsl:call-template **element, 177**
xsl:choose **element, 177**
xsl:comment **element, 177**
XslCompiledTransform **class, 50, 187**
XslCompiledTransform **object, 188**
xsl:copy **element, 177**
xsl:copy-of **element, 177**
xsl:decimal-format **element, 177**
xsl:element **element, 177**
xsl:fallback **element, 177**
xsl:for-each **element, 177**
xsl:if **element, 177**
xsl:import **element, 177**
xsl:include **element, 177**
xsl:key **element, 178**
xsl:message **element, 178**
xsl:namespace-alias **element, 178**
xsl:number **element, 178**
xsl:otherwise **element, 178**
xsl:output **element, 178**
xsl:param **element, 178**
xsl:preserve-space **element, 178**
xsl:processing-instruction **element, 178**
xsl:sort **element, 178**
xsl:strip-space **element, 178**
xsl:stylesheet **element, 178**

XSLT (Extensible Stylesheet Language Transformations)

- advanced operations, 207–209
- applying a style sheet to an XML document, 179–186
- ASP.NET page, implementation of, 204–207
- conditions, evaluating, 183
- current function, 179
- document function, 179
- element-available function, 179
- elements of, 176–179
- embedding scripts inside style sheet, 196–197
- example, complete, 199–207
- extension objects, 194–196
- format-number function, 179
- function-available function, 179
- functions, 179
- generate-id function, 179
- how it works, 174–175
- key function, 179
- multiple conditions, evaluating, 184
- need for, 175–176
- .NET classes used for XSL transformations, 186–192
- nodeset as parameter to style sheet, passing a, 207–208
- overview, 9, 17, 174
- parameters, 186, 190–192
- passing parameters to a style sheet, 190–192
- sorting an XML file using a style sheet, 182–183
- statically applying a style sheet, 180–186
- system-property function, 179
- unparsed-identity-uri function, 179
- used to transform XML data, 49–50
- user-defined functions, 193–197
- variables, 185–186
- XmlDataDocument class, 187
- XmlDocument class, 187
- XmlPathDocument class, 187
- XmlReader class, 187
- XmlResolver class, 208–209
- XmlWriter class, 187
- xsl:apply-imports element, 177
- xsl:apply-templates element, 177, 185
- xsl:attribute element, 177
- xsl:attribute-set element, 177
- xsl:call-template element, 177
- xsl:choose element, 177
- xsl:comment element, 177
- XslCompiledTransform class, 187
- XslCompiledTransform object, 188
- xsl:copy element, 177
- xsl:copy-of element, 177
- xsl:decimal-format element, 177
- xsl:element element, 177
- xsl:fallback element, 177
- xsl:for-each element, 177
- xsl:if element, 177
- xsl:import element, 177
- xsl:include element, 177
- xsl:key element, 178
- xsl:message element, 178
- xsl:namespace-alias element, 178
- xsl:number element, 178
- xsl:otherwise element, 178
- xsl:output element, 178
- xsl:param element, 178
- xsl:preserve-space element, 178
- xsl:processing-instruction element, 178
- xsl:sort element, 178
- xsl:strip-space element, 178
- xsl:stylesheet element, 178
- XsltArgumentList class, 187, 190
- XsltCompileException class, 187
- xsl:template element, 178
- XsltException class, 187
- xsl:text element, 178
- xsl:transform element, 178
- XsltSettings class, 187, 198–199
- xsl:value-of element, 178
- xsl:variable element, 178
- xsl:when element, 178
- xsl:with-param element, 178
- XsltArgumentList **class, 50, 187, 190**
- XsltCompileException **class, 187**
- xsl:template **element, 178**

XsltException class

XsltException **class, 50, 187**

xsl:text **element, 178**

xsl:transform **element, 178**

XsltSettings **class, 187, 198–199**

xsl:value-of **element, 178**

xsl:variable **element, 178**

xsl:when **element, 178**

xsl:with-param **element, 178**

