# Index


## A

Abstract message operations, 205–206
Abstraction, of WSDL, 204, 216, 219
**AcceptSocket** method, 136
**AcceptTcpClient** method, 136
<Ack> element, 286
**action** element, 243
**actor** attribute, 178, 198
  and routing, 239
**AddressType** attribute, 150
AES (Advanced Encryption Standard), 255
<all> element, 168
Apache Axis, 300
Apache SOAP, 7
  interoperability issues, 301
**Application** object, 34
Architecture, 4–5
  Web service, 12–13
Arrays
  partially transmitted and sparse, 183
  wrapper for, 183
  with XML serialization, 101–104
ASP.NET
  extension mechanisms in, 38–39
  procedures for using, 34–38
ASP.NET Web Services, 29, 296
  building, 34
  caching by, 311
  cookies in, 279–280
  and HTTP sessions, 276–277
  interoperability issues, 300
  roadmap to, 33–34
ASPs (Active Server Pages), 276
Asymmetric encryption, 257–258
**AsymmetricAlgorithm** class, 257
Asynchronous methods, 53–54
  caveats regarding, 53
  clients using, 79–81
  implementation of, 54–60
  SOAP headers and, 197–199
**AsyncResult** object, 80
Attachments, DIME and, 247–249
Attributes
  adding description through, 203–204
  namespaces of, 101
  qualifying of, 151
  searching for, 163
  types of, 100
  XML, 100–101
  of XML document element, 149
  in XML schema, 202–203
Audience–centered design, 299–301
Authentication, 177
  components of, 254
  defined, 254
Authentication headers, 55–56
Authorization, defined, 254
**authorizedName**, in **tMode**l, 226
Axis (Apache), 300

## B

Bare parameter style, 42–43
  example of, 44–45
  SOAP request-response for, 43
**BeginInvoke** method, 80

Binary security token, 268, 269
**BinarySecurityToken**, 270
  attributes of, 270–271
  sample of, 271
Binding, 52
  using, 53
<binding> section, of WSDL document, 29, 206
  example of, 211
  policy information in, 216–217
  use of, 210–211
**bindingTemplate** type, 224
**bindingTemplates**, in **businessService**, 225
<Body> element, 23, 176
  and routing, 238–239
Bridges, 178
Browser-based computing, xiv
  importance of, xv
Browsing operations, 227
Business logic, using, 309–310
**businessEntity** type, 224
**businessService** type, 224, 225
**businessServices**, 224

## C

**CacheDuration** property, 310–311
Caching, 310
  by ASP.NET, 311
**CanDeserialize** method, 94–95
Canonicalization, in signature, 265
Capitalism, xiii
**categoryBag** type, 224
  in **businessEntity**, 224
  in **businessService**, 225
  in **tModel**, 226
**ChainStream** method, 121
**CheckUserAgentModule_BeginRequest** method, 128–129
Child tag, 22
<choice> element, 168
Claim, defined, 266
Classes
  creating from schemas, 106–108
  wrapper for, 183
**Client** error code, 181
Clients
  creating using .NET Framework SDK, 65–71
  creating using Visual Studio .NET, 72–74
  document-encoded, 76–77
  document-literal, 74–76
  exception handling on, 81–83
  extending and customizing, 83–85
  manual creation of, 74–81
  one-way, 78–79
  RPC-encoded, 77–78
  that use asynchronous methods, 79–81
**code** element, 242
COM (Component Object Model), 3
**CompleteRequest** method, 129
<ComplexType> element, 168
Compression extension, 122
  configuration for, 123–124
  importer class for, 124
  machine.config entries for, 124–125
Computers, history of, xiii–xiv
Confidentiality
  defined, 254
  encryption and, 254–261
Constructors, and serialization, 89–90
**contacts**, in **businessEntity**, 224
**ContentLength**, 144
Conversation
  defined, 290
  types of, 289
**CookieContainer** object, 277
Cookies, 277, 279
CORBA (Common Object Request Broker Architecture), 3
**Create** method, of **WebRequest** class, 142
**Create(String Url)**, 144
**CryptoStream** object, 255

## D

Data types, 20
  defining, 166–167
  implicitly defining, 167
  interoperability issues, 301–302
  versioning of, 305, 307–309
**DataSet** object, 301
DCOM (Distributed Component Object Model), 3, 295–296
**Decrypt** method, 258
Delphi 6, 7
**description**, in **tModel**, 226

Description formatter, 121–122
**Deserialize** method, 89, 94
Design
  business logic in, 309–310
  caching, 310–311
  for interoperability, 299–305
  for performance, 295–299
  versioning in, 305–309
<detail>, of fault, 180
Dialogues, 289
  defined, 291
  use of, 291–293
**DidUnderstand** property, 58
**Digest** method, 264
Digital signature, 262
  creating, 263
  using with SOAP, 268
  verifying, 263
  with WS-Security, 272
DIME (Direct Internet Message Encapsulation), 247
  anatomy of message in, 248
  and SOAP, 248–250
DISCO (Web Services Discovery Language), 33, 38, 65–67
disco.exe, 65
  using, 66–67
.discomap, 67, 68
Discovery, 221
  ad hoc, 221, 232
  context of, 222
  using UDDI, 222–229
  using WS-Inspection, 229–231
**discoveryURLs**, in **businessEntity**, 224
Distributed application development, 2–3
DNS (domain name system), 12
Document extensions, 203–204
Document passing, 24
Document-encoded
  clients, 76–77
  Web services, 45–47
Document-literal clients, building, 74–76
Document-literal services
  building, 40–45
  programming model for, 41
DOM (Document Object Model)-based programming, 154, 161–163
**DownloadData** method, 142
**DownloadFile** method, 142
DTDs (document type definitions), 153
  disadvantages of, 164

## E

Element
  adding description through, 203–204
  of XML document, 149
Element name, 18
**ElementName** property, 102
Encoded SOAP, 25
  interoperability issues, 303
Encoded XML, 95
  and remote method calls, 181–184
  serialization of, 95–97
**encodingStyle** attribute, 187
**encodingType** attribute, of **BinarySecurityToken**, 270, 271
**Encrypt** method, 257
**EncryptedData** element, 260–261
Encryption
  private key, 254–257
  public key, 254, 257–258
  symmetric, 255
  in WS-Security, 272
  XML, 258–261
**EndInvoke** method, 80
**Envelope** class, 190–194
  using, 196–197
Enveloping, 176–180
EOF (end-of-file), 154
Error handling, 10
  on client, 81–83
  in SOAP, 11, 40, 60–62, 180–181
Eventing, 275
Exceptions, handling by SOAP, 40
Extensibility, of WSDL, 202–204
Extension class, creating, 116
Extension formatters, 122, 123
Extension reflectors, 122, 123

## F

<Fault> element, 11, 180
<faultcode>, 180
<faultfactor>, 180
<faultstring>, 180
Federation, 12
Fell, Simon, 300

Fields, serialization of, 100–101
Firewalls, 178
**found** element, 242
Freenet, 13
Fully qualified name (FQN), 151–152

## G

GET verb, 16, 141
**GetAttribute**, 157
**GetWebRequest** method, 83
  overriding, 126
GLUE (The Mind Electric), 7
Gnutella, 13
GUID (Global Unique Identifier), 30, 223, 276

## H

Hash
  creating, 261–262
  defined, 262
Header, session, 278, 279, 281–283
<Header> element, 177–178
  and routing, 238–239
Headers, 23
  authentication, 55–56
  defined, 17
  SOAP, 11, 54–60
Homing pigeons, 23
**href** attribute, 183
HTCPCP, 23
HTML (Hypertext Markup Language), 1
  convenience of, 213
  distinguished from XML, 148
HTTP (Hypertext Transport Protocol), 1, 13
  interoperability issues, 301
  resources about, 16
  SOAP and, 187–189, 197
  and TCP, 134–135, 140
  uses of, 16
  using, 141–144
HTTP Digest, 254
HTTP modules, 127–129
  drawbacks of, 129
HTTP properties, setting, 125–126
HTTP sessions, 276
  with ASP.NET Web services, 276–277
**HttpApplication** object, 128
HTTP-GET, 37
HTTP-POST, 37
**HttpResponse**, 40
**HttpSoapContext** class, 130
**HttpWebRequest** class, 34, 142, 144
**HttpWebRequest** object, 83

## I

**IAsyncResult** object, 80
IBM, xiii
**id** attribute, 183
**IdBag**, in **tModel**, 226
**identifierBag**, 224
**IHttpModule** interface, 128
IIS, 39
Implementation versioning, 305
  loose coupling and, 306–307
**ImportTypeMapping** method, 95
**Indentation** property, 158
Inheritance, to override proxy class, 126–127
Initialization vector (IV), 254–255
**Initiate** class, 280, 282
Inquiry operations
  browsing operations, 227
  retrieval operations, 227–228
Inspection routines, 67
Integrity
  defined, 254
  implementation of, 261–265
  in WS-Security, 272
Interface versioning, 305, 307–309
Interoperability, 6, 299
  audience-centered design and, 299–301
  issues in, 301–304
  loose coupling and, 7–8
  problems of,301–302
  SOAP platforms and, 300–304
  and WDSL, 304
  Web resources on, 300
  WS-I profiles and, 305
**Invoke** method, 71
IOCompletion ports, 53
IP (Internet Protocol), 133–134
**IsEmptyElement**, 157

## K

Kafka, 7
Kerberos, 254, 266
  and WS-Security, 271
**keyedReference**, 229
**KeyInfo** element, 264

## L

Latency, message size and, 297–298
Literal message, 45
Local name, 18, 151
Logical name
  defined, 236
  functionality of, 236–238
Loose coupling, 7–8
  and implementation versioning, 306–307

## M

**MailAttachment** class, 140
**MailMessage** object, creating, 140
maxOccurs, 169
MEP (Message Exchange Pattern), 197
Message passing, 9
  defined, 22
Message paths
  forward, 239
  reverse, 239, 241–242
<message> section, of WSDL document, 29, 205
  use of, 208–209
Message-level sessions
  header of, 278–279
  initiation of, 280
  termination of, 280
<MessageNum> element, 286
Messages
  API to send, 195–196
  asynchronous, 10, 197–199, 235
  binary data in, 247–250
  as component of operations, 9, 29
  defined, 205
  enveloping of, 176–180
  flow of, 285, 287
  intermediaries and, 178
  literal, 45
  numbering of, 284–286
  paths of, 239–243
  with references, 48–49
  requirements for sending, 201–202
  routing of, 40–41, 238–247
  size of, 297–298
  SOAP and, 189–197
  and Web services, 235–251
  WSDL and, 205–209
Method calls
  encoded XML and, 181–184
  remote, 24–26, 184–187
Methods
  asynchronous, 53–60, 79–81, 197–199
  distinguished from operations, 9
  importance of, 26
**Microsoft.Web.Services.SoapEnvelope**, 190
MIME (Multipurpose Internet Mail Extension), 247
  multipart, 247–248
minOccurs, 169
Mixed content, 152
  alternatives to, 152
  disadvantages of, 152–153
Modularity, 8
  leveraging of, 9
Monologues, 289
  uses of, 290–291
**mustUnderstand** attribute, 57–58, 179–180, 198

## N

Name
  element, 18
  FQN, 151–152
  local, 18, 151
  logical, 236–238
**name**
  in **businessService**, 225
  in **tModel**, 226
**Namespace** property, 98, 99
Namespaces, 17–18
  advanced handling of, 304
  choosing, 278–279
  defined, 18
  programming using, 154–163
  resources about, 21
  in serialized XML, 97–99

Namespaces (*cont.*)
in XML, 150
using, 151
and versioning, 307–308
.NET Framework, 7, 33
and WSDL, 213–216
XML Serialization in, 87–114
SOAP APIs in, 190
.NET Framework SDK
components of, 65
disco.exe, 66–67
wsdl.exe, 68–71
.NET Remoting, 300
Network latency, message size and, 297–298
**NetworkStream** object, 135
**NodeType**, 157
Notification transmission type, 210
Numbering, of messages, 284–286

## O

OASIS group, 30
Object references, 183
One-way clients, building, 78–79
One-way services, 49–50
example of, 50–51
One-way transmission type, 209
OOP (object-oriented programming), 26
Operations
defined, 29
distinguished from methods, 9
in WSDL, 205
**operator**, in **tModel**, 226

## P

Packaging formats, 66
Parsing
stream-based, 94
XML, 148, 155
Partially transmitted arrays, 183
**Password** element, 270
**Path** property, of **SoapContext**, 243
Performance
and design, 295–296
measuring, 296
importance of, 299
writing code for, 296–297
PIs (processing instructions), 153
PocketSOAP, 7, 300
Policy, security, 266–267
Policy information, in WDSL document, 216–217
POP3 (Post Office Protocol), 138
Port 80, uses of, 16
Ports, within WSDL, 212
**PortType** element, 304
<portType> section, of WSDL document, 29, 205
example of, 210
use of, 209–210
POST verb, 16, 142
Prefix matching, 246
Private key encryption, 254
described, 254–255
sample of, 256–257
**ProcessMessage** method, 121
Properties
advantages of, 106
serialization of, 100–101
Protocol, defined, 15, 21
Proxy class, overriding, 126–127
Public key encryption, 254
described, 257
and digital signatures, 261–262
sample of, 258

## Q

Qualified name (QName), 18

## R

**reason** element, 242
**referencedNamespace** attribute, 230
References, in messages, 48–49
Referrals, 237
Reliability
components of, 275
session technology and, 284
Remington Rand, xiii
Remote method calls, 24–26, 184–187
with SOAP, 181–184
<RequestAck> element, 286
Request-response operation, 9–10
defined, 16
in HTTP, 16
Request-response transmission type, 209–210

**RequestSoapContext** property, 130
**ResponseSoapContext** property, 130
Retrieval operations, 227–228
**retryAfter** element, 242
RFC 2324, 23
RFC 2459, 23
Rijndael algorithm, 255
RMI (Remote Method Invocation), 3
Rollover, 13
Routing, 22–23, 40–41, 237–239
  controlling of, 51–52
  defined, 23
  dynamic configuration of, 244–247
  WS-Referral and, 245–247
  WS-Routing and, 239–243
Rpc/Encoded SOAP, 26
  interoperability issues, 302–303
RPC-encoded clients, building, 77–78
RPC-encoded services, building, 47–49
RPCs (remote procedure calls), 175
**RSACryptoServiceProvider** class, 257
Ruby, Sam, xv

## S

Schema
  defined, 18
  example of, 19
  importance of, 26
  programming with, 169–173
  using, 163–169
  using different types in WSDL document, 206–207
Schema tags, 18
Security
  protocols of, 265–272
  technologies and standards of, 253–265
Security token, 265–266
  signed, 266
  types of, 268–269
  unsigned, 266
  using with SOAP, 268
  in WS-Security, 269–272
Semiconductor industry, xiii–xiv
**SendHashed**, 270
<sequence> element, 168
Serialization, 89, 94
  rules of, 182–184
**Serialize** method, 89, 94
**Server** error code, 181
Service element, of WSDL, 212
<service> section, of WSDL document, 206
<Service> tag, 29
**ServiceDescriptionFormatter**, 121, 122
**serviceKey**, in **businessService**, 225
Session
  defined, 276
  HTTP, 276
  initiation of, 280
  message-level, 278–284
  termination of, 280
Session header, 278–279, 281
  acknowledgment request in, 286
  example of, 287–289
  with message numbers, 285
  schema for, 282–284
**Session** object, 34, 276
Signature
  digital, 262–264
  XML, 264–265
**Signature** element, 264
**SignatureMethod**, 264
**SignatureValue** element, 264
**SignData** method, 263
Signed security token, 266
**SignedInfo** element, 264
SMTP (Simple Mail Transport Protocol), 16
  and TCP, 139
  using, 138–140
SOAP (Simple Object Access Protocol), xiv, 5–6, 13
  for ADA, 7
  basics of, 21–23
  basis of, 175–176
  bindings in, 52–53
  described, 3
  DIME and, 248–250
  document passing with, 24
  encoded, 25, 303
  enveloping with, 176–180
  error handling in, 11, 40, 60–62, 180–181
  and HTTP, 187–189, 197
  and interoperability, 300–304
  platforms for, 8, 300
  purpose of, 175
  remote method calls with, 24–26
  routing issues with, 235–236
  security and, 268

SOAP (Simple Object Access Protocol) (*cont.*)
sending messages with, 189–197, 238
and sessions, 279–280
and TCP/IP, 134–137
transport independence of, 133–134, 187–188
versions of, 7, 176
SOAP BEA WebLogic, 7
SOAP extension
described, 115–116
example of, 116–121
SOAP fault, 11
mechanism of, 60–62
SOAP header, 11
modified, 56–57, 58–59
using, 54–60
SOAP routers, dynamic configuration of, 244–247
SOAP Toolkit (Microsoft), 7, 300
SOAP::LITE, 7, 300
**SOAPAction** header, 16, 23
use of, 188–189
**SoapBindingUse**, 110
SOAPBuilders Yahoo! Group, 301
**SoapDocumentMethod** attribute, 71, 74
**SoapDocumentMethodAttribute** attribute, 42, 44
and binding, 53
**OneWay** property of, 50
**Use** property of, 46
**SoapDocumentService** attribute, 52
**SoapDocumentServiceAttribute** attribute, 44
**SoapException**, 40, 60, 81, 82–83
**SoapHeader** class, 57
**SoapHttpClientProtocol**, 71, 75–76
**Invoke** method of, 71
**SoapReflectionImporter** class, 95
**SoapRpcMethod** attribute, 77, 111
**SoapRpcMethodAttribute** attribute, 47, 48
and binding, 53
**OneWay** property of, 50
**SoapRpcService** attribute, 52
**SoapRpcServiceAttribute** attribute, 47, 48
**SoapUnknownHeader** class, 58, 59
SOAPx4, 7
Solicit-response transmission type, 210
Sparse arrays, 183
SSL (Secure Socket Layer), 253
Standards
described, 5–6
significance of, 5
versioning of, 8
Stream-based programming, 154
reading streams, 154–157
sample of, 158–161
writing streams, 157–158
Strings, behavior of,182–183
Structures, wrapper for, 183
Subject, of security token, 266
**SymmetricAlgorithm** class, 255
**System.Net.Sockets.Socket** class, 135
**System.Net.Sockets.TcpClient** class, 135
**System.Net.Sockets.TcpListener** class, 135
**System.Security.Cryptography** namespace, 255
**System.Text.Encoding.ASCII** object, 136
**System.Threading.Thread**, 137
**System.Web.Mail** namespace, 140
**System.Web.Services.Description**, 213
**System.Web.Services.Protocols** namespace, 42
**System.Web.Services.Protocols.SoapHeader** class, 54
**System.Web.Services.Protocols.SoapHttpClientProtocol** base class, 75
**System.Xml**, stream-based parsers in, 94
**System.Xml.Serialization** namespace, 42, 47, 76, 109
**System.Xml.XmlElement**, 59

## T

Tags, 17
child, 22
**targetNamespace** attribute, 304
TCP/IP (Transmission Control Protocol/Internet Protocol), 16
described, 133–134
functions of, 134–135
and HTTP, 134–135, 140
and SMTP, 139
and SOAP, 134–137
**TcpClient** class, 135
**TcpListener** class, 136
**Terminate** class, 280, 282

**tModel** type, 224, 225–226
example of, 229
structure of, 226
**tModelKey**, 226
Token, security, 265–266, 268–272
Transform, in signature, 265
Transmission
notification, 210
one-way, 209
request-response, 209–210
solicit-response, 210
Transport information
exposed HTTP properties, 125–126
Transport protocols
SMTP, 16, 138–140
TCP/IP, 16, 133–137, 139, 140
UDP, 23, 134, 137–138
Tree view, of XML schema, 171–172
Type information, and serialization, 92
Type Model (tModel), 30
**Type** object, creating, 94
<types> section, of WSDL document, 29, 205
use of, 206–207, 212

## U

UDDI (Universal Description, Discovery, and Integration), 13
anatomy of, 223–226
APIs in, 226–228
data format in, 223
design of, 223
functions of, 30, 222
inquiry operations in, 227–228
querying and submitting in, 223
replication API in, 223
sample message in, 30–31
WSDL and, 228–229
and WS-Inspection, 230
UDP (User Datagram Protocol), 23, 134
described, 137
unreliable messages with, 137–138
**UdpClient** class, 137
Unsigned security token, 266
Untyped XML, 104–106
URI (uniform resource identifier), 150
Usenet, 12
**UserAgent** property, 144
setting, 126
Username security token, 268, 269
adding, 269–270
sample, 270
**UsernameToken** object, 270

## V

**ValueType** attribute, of **BinarySecurityToken**, 270–271
Verbs, 16
**VerifyData** method, 263
Versioning, 8
of data types, 305, 307–309
decisions about, 305
of implementation, 305, 306–307
interface, 305, 307–309
namespaces and, 307–309
types of, 305–309
**VersionMismatch**, 181
Virtualization, 245
Visual Studio .NET
adding Web reference in, 108
creating clients using, 72–74

## W

WASP (Systinet), 7, 300
Watson, Thomas, xiii
Web service architecture
baseline specifications of, 13
federation of designs in, 12–13
vision for, 12
Web Service Behavior, 7
Web services, 1–2
advanced messaging services, 275–293
architecture for, 4–8, 12–13
describing, 201–217
designing, 295–311
development of applications, 2–3
discovery of, 221–233
document-encoded, 45–47
document-literal, 40–45
error handling in, 10–11
extensions to, 115–131
interoperability issues, 301–304
issues in, 2
message passing in, 9–10
messaging with, 235–251, 275–293

Web services (*cont.*)
modular design of, 8–9
one-way, 49–51
policy regarding, 217–220
popular technologies for, 7
procedures for creating, 34–38
routing control in, 51–52
RPC-encoded, 47–49
security for, 253–273
standards for, 21–29
structure and function of, 38–52
transport protocols for, 133–145
XML, xiv
XML serialization and, 109–113
Web Services Enhancements for Microsoft .NET (WSE), 236
attachments in, 249
referrals in, 247
routing in, 237–238, 243
security in, 269–272
Web Services Interoperability organization (WS-I), 305
**WebClient** class, 141–142
web.config, using, 159
**WebControlProtocol** method, 126
**WebException**, 82–83
**WebRequest** class, 34, 142
characteristics of, 144
using, 143
**WebRequest** object, modifying use of, 126
**WebService** base class, 125
**WebServiceBindingAttribute** attribute, 53, 71
**Whitespace**, 156
Wire format, changing, 309–310
World Wide Web, xiv
characteristics of, 4–5
Wrapper, 22
**WriteAttribute** method, 157
**WriteAttributeString** method, 157, 158
**WriteEndDocument** method, 157
**WriteEndElement** method, 157
**WriteStartDocument** method, 157, 158
**WriteStartElement** method, 157, 158
**WriteString** method, 157
WS-Inspection, 229
anatomy of, 230–231
characteristics of, 231
UDDI and, 230
WS-Referral, 235
functionality of, 245
issues regarding, 246–247
and prefix matching, 246
sample statement in, 245–246
WS-Routing, 22, 235, 275
building applications, 243–244
described, 23
design in, 237
error handling in, 242
and logical names, 236–237
and message paths, 239–243
sample message in, 240
WS-Security
described, 267–268
integrity and encryption in, 272
and Kerberos, 271
origin of, 265
sample message of, 269
security tokens in, 269–272
signatures with, 272
WSDL (Web Services Description Language), xiv, 13
abstraction in, 204, 216, 219
anatomy of, 205–212
binding in, 52–53
described, 27, 219
describing services with, 26–30
documents in, 29–30
extensibility of, 202–204, 216–217
goals of, 202
importance of, 6
interoperability issues, 304
reading of, 213–216
sample document, 27–29
structure of, 204–205
tool support in, 219–220
transmission types in, 209–210
and UDDI, 228–229
Web service policy and, 217–218
writing of, 212–213
and XML schemas, 218
WSDL First! movement, 218–219
wsdl.exe, 65
creation of classes in, 108
using, 68–71, 213
WSE (Web Services Enhancements for Microsoft .NET), 129–130
W3C (World Wide Web Consortium), 5

# X

x.509 certificates, 266
  using, 271–272
**X509Certificate** object, 271
**X509SecurityToken** object, 271, 272
XML (eXtensible Markup Language), xiv, 1, 13
  characteristics of, 147–149
  described, 17
  encoded, 95–97, 181–184
  encryption of, 258–261
  distinguished from HTML, 148
  interoperability issues, 301
  namespaces in, 17, 150–151, 154–163
  parsers for, 148
  resources about, 21
  rules of, 17
  schemas in, 163–173
  specifications for, 149
  untyped, 104–106
  validity of, 170
  writing and reading, 94–97
XML document
  characteristics of, 149–150
  defined, 148
  sample, 148
XML encryption
  need for special techniques of, 258–259
  sample of, 259–260, 261
  scope of, 260
XML schema documents, 19–20
  data types in, 20
XML schemas, 18
  creating classes from, 106–108
  data types with, 166–168
  example of, 164–166
  manipulating documents in, 171–173
  purpose of, 164
  schemas and, 20
  and WSDL, 218
  to specify XML shape, 168–169
  tree view of, 171–172
XML Serialization, 33–34, 41, 154
  arrays with, 101–104
  customization of, 97–106
  described, 87–94
  of encoded XML, 95–97
  namespaces in, 97–99, 173
  properties and fields in, 99–101
  scope of, 89–94
  of untyped XML, 104–106
  and Web services, 109–113
XML serializer, 39–40
XML signature, 264–265
  sample, 264
XML Web services
  defined, xiv
  standards for, 21–22
**XmlAnyAttribute** attribute, 112
**XmlAnyElement** attribute, 112
**XmlArray** attribute, 102–103
**XmlArrayItem** attribute, 103–104
**XmlAttributeAttribute** attribute class, 42
**XmlDocument** class, 154
  DOM-based programming with, 161–163
**XmlElement** attribute, 104
**XmlElementAttribute** attribute class, 42
**XmlNode[]**, 113
**xmlns** attribute, 151
**XMLSerializer**, 88–89, 109–113
  methods of, 94
**XmlTextReader** class, 154, 156
  properties and methods on, 157
  using, 160, 161
**XmlTextWriter** class, 95, 154, 157–158
  writing changes with, 160–161
**XmlTypeMapping** object, 95
**XmlValidatingReader** class, 170
**XmlWriter**, 95
XSD (XML Schema), 19, 20
  components of, 21
xsd.exe, 106, 108
**Xsi:type** attribute, 302