

# Index

## A

- abstract attribute (XSD), `MustInherit` class, 27, 28, 56
- Abstract base types, 56–57
- Advanced Program to Program Communication (APPC), 2
- `<all>` element (XSD), 34
- Anonymous types (XSD), 38–39
- `anyType` type (XSD), 27, 28
- Apache Axis, SOAP implementation testing, 494
- Apache SOAP for Java Clients
  - alternate Java-client tools, 494
  - messages
    - document/literal, 500–505
    - RPC/encoded, 496–500
    - Web Services Toolkit (IBM), 494–496
- APPC (Advanced Program to Program Communication), 2
- Applications (distributed)
  - and DCOM, 1–2
  - and HTTP, 2
  - interoperability of applications, 2
  - Web services definition, 2–3
- Arrays, serializing with SOAP encoding, 116
- `.asmx` file extensions, .NET Web services, 10
- ASP listeners, SOAP Toolkit Web services
  - exposing, 190, 195
  - invoking, 203–205
- ASP.NET framework
  - basics, 247
  - caching
    - data, 251–253
    - output, 250–251
  - distributed transactions, 253–254
  - sessions, 248–250
  - Web services
    - simple example, 3
    - using UDDI, 468–474
- `<attribute>` element (XSD), values, 35
- `attributeFormDefault` attribute, `<schema>` element (XSD), 43
- Attributes (XSD)
  - complex types
    - with complex content, 35–36
    - with simple content, 35–36
  - declaring, 35
  - naming collision prevention with XML namespaces
    - basics, 39–42
    - in instance XML documents, 42–43
    - URLs, 40–41
    - URLs, 40, 42
    - URNs, 41
    - XSD types, 43–46
- Authentication. See Authorization and authentication
- `<authInfo>` header (SOAP), user credentials, 120–121
- Authorization and authentication, SOAP extensions
  - client example, 396–397
  - database schemas, 380–381
  - methods
    - `CheckPermission`, 385–387
    - `LogOn` and `LogOff`, 381–385
  - security architecture, 381
  - with SOAP headers, 387
  - Web services, 540–542
  - WS-Security recommendations, 380

## B

- Base64 encoding/decoding, 121–122
- B2B applications, integrating Web services, 17–18

- Binary data handling (SOAP)
  - attachments, 122–124
  - Base64 encoding/decoding, 121–122
  - sending URLs where clients retrieve data, 124–126
- Bindings (WSDL), 132–134
  - HTTP binding, 179–183
  - MIME binding, 183–184
  - SOAP binding
    - document/literal, 143, 144–149
    - extension elements, 141–143
    - extension elements, error handling, 156–158
    - extension elements, headers, 153–155
    - RPC/encoded, 143, 149–153
- bindingTemplate UDDI data structures, 420, 421, 427
  - example, 435–439
- businessEntity UDDI data structures, 420, 421, 423–426
  - categorization, 439–441
  - identifiers, 441
- businessService UDDI data structures, 420, 421, 426
  - example, 435–439

## C

- Categorization (UDDI), 439–441
- categoryBag (UDDI), 439–441
- Child elements. *See also* Facets
  - complex types
    - mixed content, 36–38
    - simple content, 35–36
  - empty elements, 38
  - simple types, 30
- <choice> element (XSD), 34
- Classification (UDDI). *See* Categorization (UDDI)
- CLR (Common Language Runtime) attributes, SOAP
  - extensions, 370–371
- COBOL, interoperability of distributed applications, 2
- Collisions in naming elements and attributes, XSD
  - and XML namespaces
    - basics, 39–42
    - URLs, 40–41
    - URLs, 40, 42
    - URNs, 41
    - in XML documents, 42–43
    - XSD types, 43–46

- COM (Component Object Model)
  - class libraries, 474
  - disadvantages, 3
- Common Language Runtime (CLR) attributes, SOAP
  - extensions, 370–371
- Complex types (XSD), 26
  - attributes
    - with complex content, 35–36
    - with simple content, 35–36
  - defining, 33–34
  - DHTML Web Service Behavior, 509–512
- <complexType> element (XSD), 33–34, 36
- Component Object Model (COM)
  - class libraries, 474
  - disadvantages, 3
- Compression SOAP extension, 398–406
- Constraints, uniqueness, XSD type system, 57–61
- Content models, complex types, 33
- Content-Type header (SOAP), 96

## D

- Data structures (UDDI)
  - basics, 420–423
  - bindingTemplate, 420, 421, 427
    - example, 435–439
  - businessEntity, 420, 421, 423–426
    - categorization, 439–441
    - example, 435–439
    - identifiers, 441
  - businessService, 420, 421, 426
  - publisherAssertion, 421, 433–435
  - tModel, 420, 421, 428
    - categorization, 439–441, 445–446
    - identifiers, 441
      - as meta data, 430–432
      - as service types, 428–429, 443
  - tModelInstanceInfo, 421, 432–433
- DataSets
  - .NET Web services and relational databases
    - basics, 324–328
    - round tripping, 337–341
    - typed DataSets, 328–335
    - typed DataSets, nested elements, 335–337
  - .NET Web services and XML schemas, 341–348
    - schema annotations, 348–350
    - typed DataSets, 350–354
- Datatype systems. *See* Type systems

- DCOM (Distributed Component Object Model)
  - disadvantages, 3
  - and distributed applications, 1–2
- default attribute, `<attribute>` element (XSD), 35
- Derived types (XSD), 27
  - by extension, 27
  - by list, 27
  - by restriction, 27–29
  - by substitution, 53–57
- Deserialization/serialization
  - .NET Web services, 323–324
  - SOAP Toolkit high level API
    - alternative with XML as native data format, 212–215
    - basics, 205–206
    - custom type mappers, 212
    - generic type mappers, 206–210
    - UDT type mappers, 210–211
  - XSD objects
    - basics, 61–63
    - inheritance, 80–81
    - lack of support for interface serialization, 81
    - overriding serialization attributes, 78–80
    - process, 63–65
    - support by SOAP Section 5 encoding, 113
    - XML serialization, 65–78
- `<detail>` element (SOAP), error handling, 117, 118–119
- DHTML Web Service Behavior. *See also* Apache SOAP; PocketSOAP; SOAP
  - basics, 505
  - behavior setup, 505–506
  - handling complex types, 509–512
  - invoking Web services, 506–509
- Distributed applications
  - and DCOM, 1–2
  - and HTTP, 2
  - interoperability of applications, 2
  - Web services definition, 2–3
- Distributed Component Object Model (DCOM)
  - disadvantages, 3
  - and distributed applications, 1–2
- Document/encoded SOAP messages, 91–93
  - with WSDL, 143, 144–149
  - .NET Web services, 259–261
- Document/literal SOAP messages, 91–93
  - Apache SOAP for Java clients, 500–505

- basic process, 93–95
- PocketSOAP, 491–494
- with WSDL, .NET Web services, 259–261

Documentation, WSDL, 173

## E

- `elementFormDefault` attribute, `<schema>` element (XSD), 43
- Elements (XSD)
  - attributes with mixed content, 36–38
  - declaring, 29–30
  - empty content, 38
  - global, 34
  - with mixed content, 36–38
  - XSD and XML namespaces, naming collision
    - prevention
      - basics, 39–42
      - in instance XML documents, 42–43
    - URIs, 40–41
    - URLs, 40, 42
    - URNs, 41
    - XSD types, 43–46
- Empty content, elements (XSD), 38
- `enumeration` facet, `<restriction>` element (XSD), 31, 32
- Error handling, SOAP
  - basics, 116–117
  - `<detail>` element (SOAP), 117, 118–119
  - `<faultCode>` element (SOAP), 117, 118–119
  - `<faultString>` element (SOAP), 117
  - high level APIs, 201–202
- `<example>` element (XSD)
  - mixed content, 36–37
  - simple content, 36
- Extensible Markup Language. *See* XML

## F

- Facets, 32–33
  - resource information, 33
  - `<restriction>` element (XSD), 30
- `<faultCode>` element (SOAP), 117, 118–119
- `<faultString>` element (SOAP), 117
- `<field>` element (XSD), 58
- Foreign keys in relational databases, *versus* XSD
  - key-based relationships, 59–61

`fractionDigits facet, <restriction>`  
 element (XSD), 32

## G

Global elements, complex types, 34  
 GLUE (The Mind Electric), SOAP implementation  
 testing, 494

## H

Handling errors, SOAP  
 basics, 116–117  
   `<detail>` element (SOAP), 117, 118–119  
   `<faultCode>` element (SOAP), 117,  
     118–119  
   `<faultString>` element (SOAP), 117  
 high level APIs, 201–202  
 Handling SOAP binary data  
 attachments, 122–124  
 Base64 encoding/decoding, 121–122  
 sending URLs where clients retrieve data,  
 124–126  
 Headers (SOAP)  
 authorization extensions, 387  
 extending SOAP, 119–121  
   bindings with WSDL, 153–155  
 IHeaderHandler interface  
   exposing services, 215, 216–218  
   invoking services, 218–221  
 .NET Web services  
   defining classes, 273–274  
   unknown headers, accessing, 279–282  
   unknown headers, using `DidUnderstand`  
     property, 282–284  
   using in services, 274–276  
   using proxy classes, 276–279  
 High level APIs (SOAP Toolkit), 185–186  
 from classic ASP pages, 203–204  
 error handling, 201–202  
 exposing Web services, 188–199  
 invoking Web services, 199–201  
 serialization/deserialization  
   alternative with XML as native data format,  
     212–215  
   basics, 205–206  
   custom type mappers, 212  
   generic type mappers, 206–210

UDT type mappers, 210–211  
 troubleshooting with Trace Utility, 202–203  
 HTTP  
   binding and WSDL, 179–183  
   and distributed applications, 2  
   interoperability problems, 483  
   invoking .NET Web services with proxies,  
     244–247  
   and SOAP, 85–87  
   SOAP messaging, 95–96  
 Hypertext Transfer Protocol. See HTTP

## I

IBM's APPC (Advanced Program to Program  
 Communication), 2  
 identifierBag schema (UDDI), 441  
 Identifiers (UDDI), identifierBag schema, 441  
 IHeaderHandler interface (SOAP)  
   Web services  
     exposing, 215, 216–218  
     invoking, 218–221  
 IIS (Internet Information Server), role in Web  
 services, 6  
`<import>` element (WSDL), 173, 176–177  
 Importing WSDL elements, 173  
   bindings, 174–175  
   messages, 174–175  
   portTypes, 174–175  
   separating elements, 174  
 Internet-based Web service interfaces  
   changing namespaces, 312–315  
   defining, 294–301  
   implementing, 301–306  
     multiple interfaces, 306–309  
     multiple interfaces, steps, 309–312  
     at runtime, 320–321  
   programming against interfaces by generating  
     proxies, 315–319  
 Internet Information Server (IIS), role in Web  
 services, 6  
 Internet protocol stack, 86  
 Interop  
   definition, 481–482  
   distributed applications, 2  
   efforts  
     SOAP-Builders forum, 485–486  
     testing techniques, 486–487

**Interop** (*continued*)

- problems
  - basics, 482–483
  - HTTP, 483
  - SOAP, 484–485
  - WSDL, 484

Interoperability. *See* Interop

Invocation patterns (UDDI), 415, 416

`<invoice>` element (XSD), 58, 59  
 overriding serializing, 78

ISAPI listeners, SOAP Toolkit Web services, 190  
 advanced settings, 232  
 basics, 229–231

`<item>` element (XSD), 38

IXMLDOMNodeList, result of invoking  
 complexTypeServer, 212–215

**J–K**

Java, Apache SOAP for Java clients and alternate  
 tools, 494

Key-based relationships, *versus* foreign-key and  
 primary-key in relational databases, 57,  
 59–61

`<key>` element (XSD), 59

keyedReference elements (UDDI), 439–441

`<keyref>` element (XSD), 59, 61

**L**

length facet, `<restriction>` element (XSD),  
 32

Low level APIs (SOAP Toolkit), 186, 221

Web services

- exposing, 221–225
- invoking, 226–229

**M**

maxExclusive facet, `<restriction>` element  
 (XSD), 32

maxInclusive facet, `<restriction>` element  
 (XSD), 32

maxLength facet, `<restriction>` element  
 (XSD), 30, 32

maxOccurs attribute, `<quantity>` element  
 (XSD), 29–30

restrictions on global element declarations, 34

**Messaging** (SOAP), 6–7

capturing messages, 96–97

contents, 88–90

document/encoded, 91–93

document/literal, 91–93

basics, 93–95

message part (WSDL), 132–133

.NET Web services support, 6–7

RPC/encoded, 91–93

RPC/literal, 91–93

*versus* RPC, 6–7

SOAP over HTTP, 95–96

Web services

client-side, 102–103

example, 97–101

Metering (usage accounting), SOAP extensions,  
 542–543

basics, 391–395

client example, 396–397

ThreadPool object, 395–396

Microsoft SOAP Toolkit. *See* SOAP Toolkit

MIME binding, WSDL, 183–184

minExclusive facet, `<restriction>` element  
 (XSD), 32

minInclusive facet, `<restriction>` element  
 (XSD), 32

minLength facet, `<restriction>` element  
 (XSD), 32

minOccurs attribute, `<quantity>` element  
 (XSD), 29–30

restrictions on global element declarations, 34

Model groups, complex types, 34

MSXML 4.0 parser, 49–50, 51, 97, 102

**N**

NAICS (North American Industry Classification  
 System), 440

Namespaces (XML)

default changes in .NET Web services, 255–257

.NET SDK and UDDI, 474–475

and XSD

basics, 39–42

in instance XML documents, 42–43

XSD types, 43–46

Naming collisions, XSD and XML namespaces

basics, 39–42

in instance XML documents, 42–43

Naming collisions, XSD and XML namespaces  
*(continued)*  
 URIs, 40–41  
 URLs, 40, 42  
 URNs, 41  
 XSD types, 43–46

.NET Web services  
 class libraries, 474  
 clients and SOAP interoperability efforts,  
 515–516  
   solving other interoperability issues,  
 520–524  
   solving poor WSDL code, 516–520  
 creating, 8, 10–15  
 creating with VB .NET  
   basics, 235–237  
   process, 237–241  
 customizing with WSDL  
   changing default names and namespaces,  
 255–257  
   client complications, 264–270  
   disabling protocols, 254–255  
   ignoring/specifying SOAP actions, 262–264  
   SOAP message encoding styles, changing,  
 257–259  
   SOAP message encoding styles, changing  
   parameters, 259–261  
   WSDL one-way operations, 261–262  
 data handling, 323  
 error reporting with `<soap:Fault>` element,  
 285–290  
 invoking, 241–244  
   with HTTP proxies, 244–247  
 RPC and messaging support, 6–7  
 serialization of objects, 323–324  
 SOAP headers  
   defining classes, 273–274  
   unknown headers, accessing, 279–282  
   unknown headers, using `DidUnderstand`  
   property, 282–284  
   using in services, 274–276  
   using proxy classes, 276–279  
 starting with relational databases, 324–328  
   DataSets, round tripping, 337–341  
   typed DataSets, 328–335  
   typed DataSets, nested elements,  
 335–337

starting with XML schemas  
   DataSets, 341–348  
   DataSets, schema annotations, 348–350  
   typed DataSets, 350–354  
 support of SOAP Base64 encoding/decoding,  
 10–15, 121  
 type system, similarity to XSD type system, 27  
 XML document handling, 354–360  
 XML object array handling, 360–363

NMTOKENS type (XSD), 27–29

Non-relational databases, interoperability of  
 distributed applications, 2

`noNamespaceSchemaLocation` attribute, 47

North American Industry Classification System  
 (NAICS), 440

Notification WSDL operations, 179

`<Numbers>` element (XSD), 37

## O

Object serialization/deserialization (XSD)  
 basics, 61–63  
 inheritance, 80–81  
 lack of support for interface serialization, 81  
 overriding serialization attributes, 78–80  
 process, 63–65  
 support by SOAP Section 5 encoding, 113  
 XML serialization, 65–78

Operations (WSDL), 131–134  
 notification, 179  
 one-way, 179, 261–262  
 request-response, 178, 261  
 solicit-response, 179

Operators (UDDI), 411–412

`optional value, <attribute>` element (XSD),  
 35

## P–Q

Parsers, validating XML documents with XSD, 46

`<partNum>` element (XSD), 61

Parts, messages (WSDL), 132–133

`Pattern` facet, `<restriction>` element (XSD),  
 31, 33

PocketSOAP, 481. *See also* Apache SOAP; DHTML  
 Web Service Behavior; SOAP  
 basics, 487–488  
 messages

PocketSOAP (*continued*)  
 messages (*continued*)  
   document/literal, 491–494  
   RPC/encoded, 489–491

Ports (WSDL), 133, 134

portType operations (WSDL), 131–134

Primary-key in relational databases, *versus* XSD key-based relationships, 57–61

Primitive types (XSD), 27

<product> element (XSD), 60

Prohibited value, <attribute> element (XSD), 35

Publish/subscribe patterns (UDDI), 418–420

publisherAssertion UDDI data structures, 421, 433–435

Publishing Web services with UDDI, 465–467  
   businesses, 463–465  
   interfaces, 461–463  
   security, 460–461

<quantity> element (XSD), 29–30

## R

Registrars (UDDI), 412  
   registering services, 417–418

Relational databases  
   interoperability of distributed applications, 2  
   .NET Web services, 324–328  
     DataSets, round tripping, 337–341  
     typed DataSets, 328–335  
     typed DataSets, nested elements, 335–337  
   primary-key and foreign-key relationships, *versus* XSD key-based relationships, 57, 59–61

Remote Procedure Calls. *See* RPCs

Request-response WSDL operations, 178

required value, <attribute> element (XSD), 35

<restriction> element (XSD), 30  
   facets, 30–33

RPC/encoded messages  
   Apache SOAP for Java clients, 496–500  
   PocketSOAP, 489–491

RPC/encoded SOAP messages, 91–93  
   with WSDL, 143, 150–153  
   .NET Web services, 257–261

RPC/literal SOAP messages, 91–93  
   .NET Web services, 259–261

RPCs  
   *versus* messaging, 6–7  
   .NET Web services support, 6–7  
   SOAP support, 6–7

## S

Schema Designer, 528–533

<schema> element (XSD), 34, 43

schemaLocation attribute, 47

Section 5, SOAP specification, 104, 113–116

Section 7, SOAP specification, 104

Secure Socket Layers (SSL), 380

Security  
   SOAP extensions, authorization  
     CheckPermission method, 385–387  
     client example, 396–397  
     database schemas, 380–381  
     LogOn and LogOff methods, 381–385  
     security architecture, 381  
     with SOAP headers, 387  
     SoapSecurity method, 387–389  
     SoapSecurity method, example with SoapHeader attribute, 390–391  
     WS-Security recommendations, 380  
   Web services implementation, 538–539

<selector> element (XSD), 58

<sequence> element (XSD), 33–34

Serialization/deserialization  
   .NET Web services, 323–324  
   SOAP Toolkit high level API  
     alternative with XML as native data format, 212–215  
     basics, 205–206  
     custom type mappers, 212  
     generic type mappers, 206–210  
     UDT type mappers, 210–211

XSD objects  
   basics, 61–63  
   inheritance, 80–81  
   lack of support for interface serialization, 81  
   overriding serialization attributes, 78–80  
   process, 63–65  
   support by SOAP Section 5 encoding, 113  
   XML serialization, 65–78

Simple Object Access Protocol. *See* SOAP

- Simple types (XSD), 26
  - defining, 30–33
  - facets, 30–33
- `<simpleContent>` element (XSD), 36
- SOAP. *See also* Apache SOAP; DHTML Web Service Behavior; PocketSOAP
  - architecture, 88–90
  - basics, 85–87
  - binary data handling
    - attachments, 122–124
    - Base64 encoding/decoding, 121–122
    - sending URLs where clients retrieve data, 124–126
  - definition, 4–5
  - error handling
    - basics, 116–117
    - `<detail>` element (SOAP), 117, 118–119
    - `<faultCode>` element (SOAP), 117, 118–119
    - `<faultString>` element (SOAP), 117
  - extensions, authorization
    - CheckPermission method, 385–387
    - client example, 396–397
    - database schemas, 380–381
    - LogOn and LogOff methods, 381–385
    - security architecture, 381
    - with SOAP headers, 387
    - SoapSecurity method, 387–389
    - SoapSecurity method, example with SoapHeader attribute, 390–391
    - WS-Security recommendations, 380
  - extensions, compression, 398–406
  - extensions, user accounting (metering)
    - basics, 391–395
    - client example, 396–397
    - ThreadPool object, 395–396
  - extensions, and Web service request
    - processing, 367–368
    - applying, 369–370
    - applying with CLR attributes, 370–371
    - client-side, 379–380
    - initialization, per-instance, 368–369
    - initialization, per-Web method, 374–376
    - initialization, per-Web service, 372–374
    - lifetime summary, 378–379
    - modifying message streams, 376–378
  - forums, SOAP-Builders, 485–486
- headers
  - authorization extensions, 387
  - extending SOAP, 119–121
  - extending SOAP binding with WSDL, 153–155
  - IHeaderHandler interface, exposing services, 215, 216–218
  - IHeaderHandler interface, invoking services, 218–221
- headers and .NET Web services
  - defining classes, 273–274
  - unknown headers, accessing, 279–282
  - unknown headers, using `DidUnderstand` property, 282–284
  - using in services, 274–276
  - using proxy classes, 276–279
- interoperability problems, 484–485
- messaging
  - basics, 93–95
  - capturing messages, 96–97
  - contents, 88–90
  - document/encoded, 91–93, 259–261
  - document/literal, 91–93, 259–261
  - RPC/encoded, 91–93, 257–261
  - RPC/literal, 91–93
  - SOAP over HTTP, 95–96
  - Web services, client-side, 102–103
  - Web services, example, 97–101
- .NET Web services, error reporting with
  - `<soap:Fault>` element, 285–290
- relationship to HTTP, 85–87
- RPC and messaging support, 6–7
- SOAP binding with WSDL
  - document/literal, 143, 144–149
  - extensions with error handling, 156–158
  - extensions with headers, 153–155
  - RPC/encoded, 143, 149–153
- SOAP for RPC
  - basics, 103–106
  - client-used proxy class invoking Web services, 110–113
  - COM components as Web services, 106–108
  - DCOM applications, converting to SOAP, 104
  - request/response encoded messages, 108–110
  - Section 5 encoding, 104, 105, 108–116
  - Section 7 encoding, 104
- with TCP and IP, 85–87
- and XML, 87



SOAP-Builders forum, 485–486

SOAP Toolkit

- definition, 4–5
- exposing Web services, 186–188
- high level APIs, 185–186
  - from classic ASP pages, 203–204
  - error handling, 201–202
  - exposing Web services, 188–199
  - invoking Web services, 199–201
  - troubleshooting with Trace Utility, 202–203
- high level APIs, serialization/deserialization
  - alternative with XML as native data format, 212–215
  - basics, 205–206
  - custom type mappers, 212
  - generic type mappers, 206–210
  - UDT type mappers, 210–211
- IHeaderHandler interface
  - exposing Web services, 215, 216–218
  - invoking Web services, 218–221
- low level APIs, 186, 221
  - exposing Web services, 221–225
  - invoking Web services, 226–229
- support of Base64 encoding/decoding, 121
- Trace Utility, 97
- Web services, creating, 8–10
- WSDL Generator, 135

SOAPAction header (SOAP), 96

<soap:address> element (SOAP), 141, 143

<soap:binding> element (SOAP), 140, 141

<soap:body> element (SOAP), 141

SoapClient COM component, 8–10

<soap:detail> element (SOAP), 156–158

<soapenv:Body> element (SOAP), 91

- messaging basics, 93–95

<soapenv:Header> element (SOAP), 91

- user credentials, 120

<soap:fault> element (SOAP), 141, 156–158

<soap:header> element (SOAP), 141, 153–155

<soap:headerfault> element (SOAP), 141

<soap:operation> element (SOAP), 141, 142

SoapSecurity, 387–389

SoapSecurity, example with SoapHeader attribute, 390–391

SOAPSender utility, 102–103

SoapServer COM component

- Web services, creating with SOAP Toolkit, 8–10

Solicit-response WSDL operations, 179

SSL (Secure Socket Layers), 380

System.Object type system root (.NET), 27

## T

tModel UDDI data structures, 420, 421, 428

- categorization, 439–441, 445–446
- identifiers, 441
- as meta data, 430–432
- as service types, 428–429, 443

tModelInstanceInfo UDDI data structures, 421, 432–433

totalDigits facet, <restriction> element (XSD), 33

Trace Utility (SOAP Toolkit), 97

- high level APIs, 202–203

type attribute, <element> element (XSD), 39

Type mapping (SOAP), 205

- custom mapping, 212
- generic mapping, 206–210
- UDT mapping, 210–211

Type systems (XSD)

- abstract base types, 56–57
- anonymous types, 38–39
- attribute declaration, 35
- basic concepts, 26–27
- complex types, 26, 33–34
  - attributes with complex content, 35–36
  - attributes with simple content, 35–36
- components, 24–26
- defining, 30–33
- derived types, 27
  - by extension, 27
  - by list, 27
  - by restriction, 27–29
  - by substitution, 53–57
- elements declaring, 29–30
  - empty content, 38
  - mixed content, 36–38
- nil values, 57
- object serialization/deserialization
  - basics, 61–63
  - inheritance, 80–81
  - lack of support for interface serialization, 81
  - overriding serialization attributes, 78–80
  - process, 63–65
  - XML serialization, 65–78
- primitive types, 27

- Type systems (*continued*)
    - reasons for needing, 23–24
    - similarities
      - to .NET type system, 27
      - to programming languages, 25
    - simple types, 26, 30–33
    - support by Section 5, SOAP encoding, 113
    - uniqueness constraints and key-based relationships, 57–61
  - Typed DataSets, .NET Web services
    - and relational databases, 328–335
      - nested elements, 335–337
    - and XML schemas, 350–354
- U**
- UDDI
    - API example, 449
    - API queries, 449–450
      - businesses by categories, 452–453
      - businesses by name, 450–452
      - businesses by Web services interfaces, 455–457
      - Web services by name, 457
      - Web services details, 458–460
      - Web services interfaces, 453–455
    - ASP.NET Web service example, 468–474
    - categorization, 439–441
    - components
      - operators, 411–412
      - private implementations, 413, 477
      - registrars, 412
      - specifications, 411
    - data structures
      - basics, 420–423
      - bindingTemplate, 420, 421, 427
      - bindingTemplate, example, 435–439
      - businessEntity, 420, 421, 423–426
      - businessEntity, categorization, 439–441
      - businessEntity, example, 435–439
      - businessEntity, identifiers, 441
      - businessService, 420, 421, 426
      - publisherAssertion, 421, 433–435
      - tModel, 420, 421, 428
      - tModel, as meta data, 430–432
      - tModel, as service types, 428–429, 443
      - tModel, categorization, 439–441, 445–446
      - tModel, identifiers, 441
      - tModelInstanceInfo, 421, 432–433
    - definition, 409–411
    - identifiers, 441
    - message types, 448
    - programming with UDDI SDK, 447–448
      - basics, 474
      - invoking UDDI APIs, 475–476
      - namespaces, 474–475
    - publishing Web services, 465–467
      - businesses, 463–465
      - interfaces, 461–463
      - security, 460–461
      - specifying business relations, 467–468
    - registering WSDL-based Web services, 442–443
    - UDDI Services for Windows .NET, 477
    - uses
      - finding closest businesses, 413–414
      - finding trading partners's services, 415–417
      - invocation patterns, 415, 416
      - publish/subscribe patterns, 418–420
      - registering services, 417–418, 435–439
  - VS .NET Web services
    - Add Web Reference dialog, 444
    - registering services with UDDI from start page, 447
    - searching UDDI from start page, 444–446
    - VB Web Service Search Categorization, 445–446
    - Web services, 550–554
  - UDT (User-Defined Type) mappers, 210–211
  - Uniform Resource Identifiers. See URIs
  - Uniform Resource Locators. See URLs
  - Uniform Resource Names. See URNs
  - <unique> element (XSD), 57–58
  - Uniqueness constraints, XSD type system, 57–61
  - Universal Description Discovery and Integration. See UDDI
  - URIs, XSD and XML namespaces, 40–41
  - URLs, XSD and XML namespaces, 40, 42
  - URNs, XSD and XML namespaces, 41
  - User accounting (metering), 542–543
    - basics, 391–395
    - client example, 396–397
    - ThreadPool object, 395–396
  - User-Defined Type (UDT) mappers, 210–211

**V**

- Valid XML documents, 26
- Validating XML documents with XSD
  - basics, 46–48
  - parsers, 46
  - programmatically, 48–52
  - XML/XSD editors, 46
- VB 6
  - clients and SOAP interoperability efforts, 512–515
  - testing Web services on clients, 544–550
  - Web services
    - creating with SOAP Toolkit, 8–10
- VB .NET. *See* .NET
- Visual Basic 6. *See* VB 6
- Visual Basic .NET. *See* .NET
- Visual Studio .NET. *See* VS .NET
- VS .NET Web services
  - creating
    - basics, 235–237
    - process, 237–241
  - invoking, 241–244
    - with HTTP proxies, 244–247
  - UDDI
    - registering services with UDDI from start page, 447
    - searching UDDI from start page, 444–446
  - VB Web Service Search Categorization, 445–446
  - VS's Add Web Reference dialog, 444

**W**

- WASP Developer (Systinet), SOAP implementation
  - testing, 494
- W3C, XML Schema Definition
  - built-in datatype hierarchy, 28
  - origin, 23–24
- Web Service behaviors (DHTML)
  - basics, 505
  - behavior setup, 505–506
  - handling complex types, 509–512
  - invoking Web services, 506–509
- Web Service Description Language. *See* WSDL
- Web service request processing, 365–366
  - SOAP extensions, 367–368
  - applying, 369–370
    - applying with CLR attributes, 370–371
    - client-side, 379–380
    - initialization, per-instance, 368–369
    - initialization, per-Web method, 374–376
    - initialization, per-Web service, 372–374
    - lifetime summary, 378–379
    - modifying message streams, 376–378
- Web services
  - architecture, 5–6
  - definition, 2–3
  - designing messages with Schema Designer, 528–533
  - documentation pages, 543–544
  - implementing services
    - basics, 533–534
    - getting information, 537–538
    - RequestElement-Bare problems, 539–540
    - security, 538–539
    - writing code, 535–537
  - .NET services, 10–15
  - platform features, 3–4
    - SOAP, 4–5
    - WSDL, 5
    - XML, 4
    - XSD, 4
  - process steps, 5–6
  - situations advantageous for using
    - integrating applications, 16–17
    - integrating B2B applications, 17–18
    - reusing software, 18–19
  - situations disadvantageous for using, 19
    - homogeneous applications on LANs, 20–21
    - single machine applications, 20
  - testing on VB 6 clients, 544–550
  - UDDI, 550–554
  - user accounting (metering), 542–543
  - WSL (Web Services Layer), 187–188
- Web Services Toolkit (IBM), Apache SOAP for Java
  - clients, 494–496
- whiteSpace facet, <restriction> element (XSD), 33
- Windows .NET Server, UDDI version, 477
- World Wide Web Consortium. *See* W3C
- WS-Security recommendations, 380
- WSDL
  - customizing .NET Web services
    - changing default names and namespaces, 255–257

WSDL (*continued*)

- customizing .NET Web services (*continued*)
  - client complications, 264–270
  - disabling protocols, 254–255
  - ignoring/specifying SOAP actions, 262–264
  - SOAP message encoding styles, changing, 257–259
  - SOAP message encoding styles, changing parameters, 259–261
  - WSDL one-way operations, 261–262
- definition, 5
- documentation, 173
- extensibility elements, 177–178
- HTTP binding, 179–183
- importing elements, 173
  - bindings, 174–175
    - with `<import>` element, 176–177
    - messages, 174–175
    - portTypes, 174–175
    - separating elements, 174
  - interoperability problems, 484
- MIME binding, 183–184
- operations
  - notification, 179
  - one-way, 179, 261–262
  - request-response, 178, 261
  - solicit-response, 179
- registering Web services with UDDI, 442–443
- SOAP binding
  - document/literal, 143, 144–149
  - extension elements, 141–143
  - extension elements, error handling, 156–158
  - extension elements, headers, 153–155
  - RPC/encoded, 143, 149–153
- and Web services
  - examples, created by WSDL Generator, 134–140
  - examples, Product Checker, 159–163
  - examples, Quote of the Day, 168–172
  - examples, Weather Service, 164–168
  - overview, 131–134
  - relationship, 129–131
- WSDL Generator (SOAP Toolkit), 135
  - example, 134–140
  - exposing Web services, 188–199
- WSDL Wizard (SOAP Toolkit)
  - creating Web services, 8–10
  - exposing Web services, 188–199

- wsdl.exe .NET tool, 14
- WSL (Web Services Layer), 187–188
- WSTK (Web Services Toolkit, IBM), Apache SOAP for Java Clients, 494–496

## X–Z

XML

- definition, 4
- .NET Web services
  - DataSets, 341–348
  - DataSets, schema annotations, 348–350
  - typed DataSets, 350–354
- and SOAP, 87
- validating documents with XML editors, 46
- XML documents
  - inheritance, 80–81
  - lack of support for interface serialization, 81
  - .NET Web services
    - handling documents, 354–360
    - handling object arrays, 360–363
  - overriding serialization attributes, 78–80
- valid, definition, 26
- validating with XSD
  - basics, 46–48
  - parsers, 46
  - programmatically, 48–52
  - XML/XSD editors, 46
- XML serialization, 65–78
- XSD and namespaces, 42–43
- XSD object serialization/deserialization
  - basics, 61–63
  - process, 63–65
- XML namespaces and XSD
  - basics, 39–42
  - in instance XML documents, 42–43
  - URLs, 40–41
  - URLs, 40, 42
  - URNs, 41
  - XSD types, 43–46
- XML Schema Definition (W3C). See XSD
- XML Signature, 380
- `XmlArrayItem` attribute (XSD), 76
- `XmlAttributeAttribute` attribute (.NET), 64
- `XmlElementAttribute` attribute (.NET), 64
- `XmlIgnore` attribute (XSD), 75
- `XmlRoot` attribute (XSD), 75
- `XMLSchemaCollection` class, 63, 64, 65–78

- XMLSerializer class, 63, 64, 65–78
- XMLValidatingReader class, 63, 64, 65–78
- XmlValidatingReader parser, 49, 50–52
- XSD
  - components, 25
  - definition, 4
  - origin through W3C, 23–24
  - validating XML documents
    - basics, 46–48
    - parsers, 46
    - programmatically, 48–52
    - XML/XSD editors, 46
  - validating XML documents with XSD editors, 46
- XML namespaces
  - basics, 39–42
  - in instance XML documents, 42–43
  - URIs, 40–41
  - URLs, 40, 42
  - URNs, 41
  - XSD types, 43–46
- XSD type system
  - abstract base types, 56–57
  - anonymous types, 38–39
  - attributes, 35
  - basic concepts, 26–27
  - complex types, 26
    - attributes with complex content, 35–36
    - attributes with simple content, 35–36
  - components, 24–26
  - defining complex types, 33–34
  - defining simple types, 30–33
  - derived types, 27
    - by extension, 27
    - by list, 27
    - by restriction, 27–29
    - by substitution, 53–57
  - elements, 29–30
    - empty content, 38
    - mixed content, 36–38
  - nil values, 57
  - object serialization/deserialization
    - basics, 61–63
    - inheritance, 80–81
    - lack of support for interface serialization, 81
    - overriding serialization attributes, 78–80
    - process, 63–65
    - XML serialization, 65–78
  - primitive types, 27
  - reasons for needing, 23–24
  - similarities
    - to .NET type system, 27
    - to programming languages, 25
  - simple types, 26
  - support by Section 5, SOAP encoding, 113
  - uniqueness constraints and key-based relationships, 57–61
  - XML namespaces, 43–46
- xsi:type attribute (XSD), 57