



Index

Symbols

2PC (two-phase commit) protocol, 456-458

A

abort bit (transactions), 250

abstraction, 10

access checks, 45

enabling, 407-408

access control

defining roles (authorization), 410-413

for message queues, configuring, 333

access control lists (ACLs), 388

access speed of state, 291

accessing. *See also* data access

ASP.NET state from serviced

components, 310-311

serviced components

DCOM, 162-171

SOAP services (.NET remoting),
172-190

Web services, 192-199

**ACID (atomicity, consistency,
isolation, durability), 238**

acknowledge messages, 330

WS-ReliableMessaging, 451

ACLs (access control lists), 388

Activate method, overriding, 66

activation service, 453

**Active Directory, state management,
296-297**

**Active Directory Integration option
(message queuing server
installation), 326**

Active Template Library (ATL), 156

activities, 17, 38, 96-98

selecting synchronization options, 108

services without components, 109

Activity class, 110

example, 112-116

ServiceConfig class, 110

ServiceDomain class, 109

winin serviced components, 116

stateful components, 109

synchronization domain, 97

Activity class, 110

methods, 112

admin components, 436-441

administration

Component Services tool, 23

queues, 329

tools for message queuing, 331-334

ADO.NET

datasets, 222

filling, 224-228

tables in, 224

updating, 228-233

overview, 203-206

affinity of state, 291

- agile objects versus context-bound objects, 33-34**
- aliases, creating, 502**
- apartments, 16, 90-91**
 - MTA (multithreaded apartments), 17, 94-95
 - selecting synchronization options, 108
 - STA (single-threaded apartments), 16, 91-94
 - synchronization, 106
 - TNA (thread-neutral apartments), 17, 95-97
 - versus contexts, 90
- API (application programming interface), 21**
- APPID (application ID), 145**
- application activation, 80-81**
- application classes (CRMs), 275-277**
- application configuration files (publisher policy assembly), 121**
- application development**
 - application layers, 4
 - advantages of, 7-8
 - business services, 6
 - data services, 6
 - presentation services, 5-6
 - cluster services, 9
 - logical tiers versus physical systems, 8
 - service-oriented architecture (SOA), 9-11
 - smart clients, 11-13
 - two-tier system, 2-4
- application domains, 34-35**
- application layers, 4**
 - advantages of, 7-8
 - business services, 6
 - data services, 6
 - presentation services, 5-6
- application pooling, 29, 81**
- application programming interface (API), 21**
- application proxies, 163-164**
 - creating, 164-167
 - installing, 167-168
 - .NET remoting, 191
- application recycling, 29, 81**
- application servers, 18**
- application services**
 - automatic transactions, 24-26
 - COM+ 1.5 new features, 28-30
 - history of, 13
 - COM versus .NET, 13-18
 - COM+ Services, 20-21
 - Microsoft Transaction Server (MTS), 18-20
 - loosely coupled events (LCEs), 27
 - queued components (QCs), 26
 - role-based security, 27-28
- application state (ASP.NET), 304**
- applications, 21-22**
 - APPID (application ID), 145
- architecture**
 - CRMs, 273-275
 - LCEs, 357
 - event classes, 358
 - filters, 359-360
 - publishers, 358
 - subscribers, 358
- array lists, datagrid mapping, 497**
- ASP.NET**
 - implementing Web services, 194-198
 - state management, 304
 - accessing from serviced components, 310-311
 - cache, 308-310
 - characteristics of, 311
 - session state, 305-308
 - transaction support, 264
- ASP.NET Web applications, security, 393**
 - authentication, 396-398
 - authorization, 398
 - course management sample application, 503-508
 - identity, 394-395

**ASP.NET Web services,
authentication, 399**

SOAP headers, 399-401
WSE (Web Services Enhancement),
401-404

assemblies

entity classes, 207-211
global assembly cache, creating serviced
components, 47-48
installing in global assembly cache, 433
loading at runtime, 147
COM registry entries, 148-149
private versus shared assemblies, 147
multiple assembly versions, 120-121
private assemblies, 47
separating, 254
shared assemblies, 47
versioning, 121
signing, 46
versioning, 143-144

assembly attributes

creating queued components, 335-336
creating serviced components, 44-45

asynchronous programming, 321**ATL (Active Template Library), 156****atomic transactions versus business
activities, 460****atomicity (transactions), 238****attributes, 434**

[AutoComplete], 258-259
list of, 434-436
[SecurityRole], 412
ServicedComponent class, 51-55
[SoapHeader], 399
[Transaction], 247-249

authentication, 386

ASP.NET security, 396-398
ASP.NET Web services
SOAP headers, 399-401
WSE (Web Services Enhancement),
401-404
IIS (Internet Information Server),
391-392

SQL Server, 388-389
Windows security, 388

**authentication levels (Enterprise
Services security), 414****AuthenticationOption values, 415-416****authorization, 386**

ASP.NET security, 398
Enterprise Services security, 407
checking roles programmatically,
413-414
defining role access, 410-413
defining roles, 408-409
enabling access checks, 407-408

[AutoComplete] attribute, 258-259**[AutoComplete] methods, registration
message, 69****automatic deployment, 426****automatic failover, message
queuing, 340****automatic transactions, 24-26, 246**

attributes, 247-249
[AutoComplete] attribute, 258-259
example, 253-257
outcomes, 250-251
setting transactional vote, 258
transaction streams, 250

automatic transactions (MTS), 20**B****basic authentication (IIS), 392****blittable data types, 122****boundary, 10**

for activities, 99

boxing and unboxing, 214**business activities, 458-460**

versus atomic transactions, 460

business components, 4

versus datasets, 234

**business services (application
layers), 6**

C

C++ clients, accessing serviced components, 151-154

C++ components, calling COM objects with serviced components, 156-157

cache (ASP.NET state management), 304, 308-310

call contexts (.NET remoting), client state management, 315-318

caller identity, Enterprise Services security, 419-420

calling COM objects with serviced components, 155-159

case study. *See* course management sample application

causality, 97

causality-based locks, 107

CCWs (COM callable wrappers), 122-123, 171

certificates, server certificates (IIS security), 392

challenge/response protocol, 388

channel sinks (.NET remoting), 173

channels
Indigo connector, 465
.NET remoting, 174

class attributes, ServicedComponent class, 51-55

class ID (CLSID), 145-146

class library projects, creating, 43

classes

ContextUtil

methods, 252

transaction outcomes, 251

CRM application classes, 275-277

differentiating, 197

entity classes, 207-211

course management sample application, 475-482

versus datasets, 234

event classes, 358

proxies, 378

as queued components, 380

sample LCE application, 360-362

generics, 212

helper classes (CRM sample application), 280-282

object-based collection classes, 210

sink classes, 369

synchronization in, 88

type-safe collection classes, 210-211

CLB (component load balancing), 9

Clerk class, 275-276

clerks (CRM), 274-275

monitoring, 287

ClickOnce deployment, 5

client applications

authentication levels, 414

C++ clients, accessing serviced components, 151-154

calling COM objects with serviced components, 158-159

COM interoperability, 121-122

COM callable wrappers (CCWs), 122-123

default COM configuration, 123-127

runtime callable wrappers (RCWs), 122-123

type libraries, 127-131

CRM sample application, 285-287

DCOM clients, creating, 169-171

message queuing clients, 327

with multiple threads, 76-77

.NET clients, accessing serviced components, 120-121

.NET remoting clients, creating, 186-190

queued components, 341-345

scripting clients, accessing serviced components, 154-155

serviced components, creating with, 59-61

- services without components, 113
- smart clients, 11-13
- for SPM (Shared Property Manager), 302-303
- STA, synchronization, 106
- Visual Basic 6 clients, accessing serviced components, 150-151
- Windows Forms client application (course management sample application), 495-499
- client state management, 312**
 - .NET remoting call contexts, 315-318
 - SOAP requests, 312-315
- client-activated objects (.NET remoting), 174**
- CLR (Common Language Runtime), 32, 143**
- CLS (Common Language Specification), 14**
- CLSID (class ID), 145-146**
- cluster services, 9**
- COM**
 - interface types, 134
 - ComVisible attribute, 142-143
 - custom interfaces, 134-135
 - dispatch interfaces, 135-139
 - duel interfaces, 139-142
 - which to use, 142
 - and SPM, 297
 - versus .NET, 13
 - configuration, 18
 - error handling, 18
 - inheritance, 15
 - language support, 13-14
 - late binding, 16
 - metadata, 15
 - object lifetime management, 14-15
 - threading, 16-18
- COM callable wrappers (CCWs), 122-123, 171**
- COM clients, accessing serviced components. *See also* client applications**
 - C++ clients, 151-154
 - scripting clients, 154-155
 - Visual Basic 6 clients, 150-151
- COM components**
 - apartments, 90-91
 - MTA (multithreaded apartments), 94-95
 - STA (single-threaded apartments), 91-94
 - TNA (thread-neutral apartments), 95-97
 - calling with serviced components, 155-159
 - interoperability with serviced components, 121-122
 - COM callable wrappers (CCWs), 122-123
 - default COM configuration, 123-127
 - runtimecallable wrappers (RCWs), 122-123
 - type libraries, 127-131
 - versioning, 144-145
 - APPID (application ID), 145
 - CLSID (class ID), 145-146
 - COMCompatibleVersion attribute, 147
 - IID (interface ID), 146-147
- COM contexts, mixing with .NET contexts, 41**
 - COM object calling .NET component, 42-43
 - .NET component calling COM object, 41
- COM events, 355-356**
- COM registry entries, 148-149**
- COM+ 1.5 new features, 28-30**
- COM+ routing, 378**
- COM+ Services, 13, 20-21**

- COMCompatibleVersion attribute, 147**
- Command Language Specification (CLS), 14**
- committed transactions, 238**
- Common Language Runtime (CLR), 32, 143**
- compensating resource managers. *See* CRMs**
- Compensator class, 277**
- compensators (CRM), 274-275**
 - CRM sample application, 284-285
- completion protocol, 456**
- component load balancing (CLB), 9**
- Component Services tool, 23**
 - creating serviced components, 58-59
 - refreshing components, 145
- components, 21-22. *See also* serviced components**
 - required transactions, 249
 - supported transactions, 248-249
- ComVisible attribute, 142-143**
- concurrency versus consistency, 266. *See also* synchronization**
- confidentiality, 386. *See also* secure data transfer**
- configured components, 23**
- configuring**
 - access control for message queues, 333
 - admin components, 436-441
 - attributes. *See* attributes
 - COM versus .NET, 18
 - creating serviced components (Component Services Explorer tool), 58-59
 - CRM sample application, 279-280
 - default COM configuration, 123-127
 - IIS (Internet Information Server) for .NET remoting, 181-183
 - message queues, 331-332
 - .NET remoting, 184-186
 - server applications for .NET remoting, 180
 - synchronization, 99-100
 - as Windows service, 441-442
- connection pools, 215-216**
 - and user accounts, 389
- connector (Indigo), 465-466**
- consistency (transactions), 238-239**
 - versus concurrency, 266
- consistent bit (transactions), 250**
- construction strings, object activation, 64-65**
- context-bound objects versus agile objects, 33-34**
- contexts, 16, 31-33**
 - mixing .NET and COM contexts, 41
 - COM object calling .NET component, 42-43
 - .NET component calling COM object, 41
 - .NET remoting, 34-37
 - serviced components, 37-38
 - accessing information, 39-40
 - method call interception, 39
 - object activation interception, 38
 - services without components example, 115-116
 - synchronized access, 35-37
 - versus apartments, 90
- ContextUtil class**
 - methods, 252
 - static properties, 39-40
 - transaction outcomes, 251
- contracts, 10**
 - interfaces as, 48
- cookies, session state (ASP.NET), 306**
- coordination contexts, 453**
- coordination protocols, 453**
- coordination services, 453**

course management sample application

data access, 484-489
 databases, 483-484
 entity classes, 475-482
 projects in, 473-475
 queued components, 509-510
 serviced components, 489-495
 use cases, 472-473
 Web applications, 503-508
 Web services, 499-503
 Windows Forms client application, 495-499

CourseControl class (creating queued components), 338-339**CourseManagement database (course management sample application), 483-484****CourseManagement.Control component library (course management sample application), 474, 489-495****CourseManagement.Data component library (course management sample application), 474, 484-489****CourseManagement.Entities component library (course management sample application), 474-482****CourseManager application (course management sample application), 475, 495-499****CourseRegistration.Control application (course management sample application), 475, 509-510****CourseRegistrationService Web service (course management sample application), 475, 499-503****CourseRegistrationWeb application (course management sample application), 475, 503-508****CRMs (compensating resource managers), 273**

application classes, 275-277
 architecture, 273-275
 monitoring clerks, 287
 sample application, 276-279
 client applications, 285-287
 compensator components, 284-285
 configuration, 279-280
 helper classes, 280-282
 worker components, 282-283

custom installers, creating, 426-431**custom interfaces, 134-135****D****data access**

accessing the database, 212-213
 inserts, 217-218
 queries, 213-217
 stored procedures, 220-221
 updates, 219-220
 ADO.NET overview, 203-206
 course management sample application, 484-489
 datasets, 222
 filling, 224-228
 tables in, 224
 updating, 228-233
 entity classes, 207-211
 sample database, 206

Data Access Application Block, 489**data formats (.NET remoting), 174****data providers, 204-206****data services (application layers), 6****databases. *See also* data access**

accessing, 212-213
 inserts, 217-218
 queries, 213-217
 stored procedures, 220-221
 updates, 219-220

- course management sample application, 483-484
- sample database, 206
- state management, 293-294
 - Active Directory, 296-297
 - isolated storage, 294-296
- tables
 - identifiers, 207
 - timestamps, 207
- datagrid mapping to array lists, 497**
- datasets, 204-205, 222**
 - filling, 224-228
 - tables in, 224
 - updating, 228-233
 - versus entity classes, 234
- DCOM, 162**
 - application proxies, 163-164
 - creating, 164-167
 - installing, 167-168
 - client applications, creating, 169-171
 - transport protocols, 162-163
 - when to use, 171
- Deactivate method, overriding, 66**
- dead-letter queues, 328**
- deadlocks, 88**
- debugging**
 - library applications, 62
 - server applications, 62
- declarative programming, 23**
- default COM configuration, 123-127**
- default constructors, object activation, 64-65**
- delegates, 356**
- delivery modes of messages, 330**
- dependencies, cache dependencies (ASP.NET state management), 309**
- dependent clients (message queuing), 327**
- deployment, 425. See also installing**
 - automatic deployment, 426
 - custom installer, creating, 426-431
 - installation package, creating, 432-433
 - manual deployment, 426
- dialog boxes and serviced components, 127**
- dialog channel, 465**
- differentiating classes, 197**
- digest authentication (IIS), 392**
- dirty reads, 265**
- dispatch interfaces, 135-139**
- dispatchers (.NET remoting), 173**
- Dispose method, JITA (just-in-time activation), 70**
- dispose pattern, 74-75**
- disposing of pooled objects, 78**
- distributed applications, future of. See future of distributed applications**
- distributed transaction coordinator (DTC), 25, 240, 250**
- distributed transactions, 239-241, 260**
 - ASP.NET support, 264
- DLL (dynamic link library), 19**
- dllhost.exe process, 62**
- document style SOAP messages, 445**
- done bit (transactions), 250, 253**
- doomed bit (transactions), 250**
- Downlevel Client Support option (message queuing server installation), 326**
- DTC (distributed transaction coordinator), 25, 240, 250**
 - monitoring, 268
- duel interfaces, 139-142**
- durability**
 - state, 291
 - transactions, 239
- durable 2PC protocol, 458**
- dynamic link library (DLL), 19**
- dynamic queries (database access) versus stored procedures, 221**
- dynamic registration, creating serviced components, 56**

E**early binding, 16, 136****enabling access checks, 407-408****encapsulation, 10****encoded style SOAP messages, 445****encryption, storing username and password, 390****Enterprise Services**application configuration. *See*

configuring

deployment. *See* deploymentfuture of. *See* future of distributed

applications

migrating to Indigo, 467-469

runtime, loading assemblies, 147

COM registry entries, 148-149

private versus shared assemblies, 147

security, 405

authentication levels, 414

authorization, 407-414

identity, 405-406, 419-420

impersonation, 416-419

.NET remoting, 420-421

entity classes, 207-211course management sample application,
475-482

versus datasets, 234

error components, 346-348**error handling**

COM versus .NET, 18

and event log, 431

queued components, 346-349

event classes, 358

proxies, 378

as queued components, 380

sample LCE application, 360-362

event log

CRM, 279

and error handling, 431

writing messages to, 431

event monitoring in serviced components, 50**EventClassAttribute properties, 363****events, 27**

COM events, 355-356

LCEs. *See* LCEs

.NET events, 356

example application. *See* course**management sample application****exception handling, [AutoComplete]****attribute, 259****exceptions, 18****express delivery mode (messages), 330****extensibility of SOAP, 447-448****F****factory pattern, 482****files, state management, 293-294**

Active Directory, 296-297

isolated storage, 294-296

filters (LCEs), 359-360

publisher filters, 375-376

subscriber filters, 373-375

formatters (.NET remoting), 172**Forms authentication (ASP.NET), 396****freeing resources, dispose pattern,
74-75****functional decomposition, 10****future of distributed applications**

Indigo, 463

connector, 465-466

hosting environments, 463-464

messaging services, 466-467

migrating from Enterprise Services,
467-469

service model, 466

system services, 464-465

Web services specifications, 444

messaging, 444-450

reliable messaging, 450-452

security, 460-463

transactions, 452-460

G**GAC (global assembly cache), 120**

- creating serviced components, 47-48
- installing assemblies in, 433

generics, 212**groups**

- assigning to roles, 412
- naming, 300

GUIDs (global unique identifiers), 144-145

- creating, 300
- as identifiers, 207

H**helper classes (CRM sample application), 280-282****history of application services, 13**

- COM versus .NET, 13-18
- COM+ Services, 20-21
- Microsoft Transaction Server (MTS), 18-20

hosting environments, Indigo, 463-464**HTTP channel (.NET remoting), 191****HTTPS protocol, 392****I****identifiers in database tables, 207****identity**

- ASP.NET security, 394-395
- configuring server applications, 81
- Enterprise Services security, 405-406
 - caller identity, 419-420
 - library applications, 405

IID (interface ID), 130, 146-147**IIS (Internet Information Server)**

- configuring for .NET remoting, 181-183
- security, 391
 - authentication, 391-392
 - secure data transfer, 392-393

IMDB project, cancellation of, 294**immutable state, 290****impersonation, Enterprise Services security, 416-419****ImpersonationLevelOption values, 418****independent clients (message queuing), 327****Indigo, 463**

- connector, 465-466
- declaring transactions, 271
- hosting environments, 463-464
- messaging services, 466-467
- migrating from Enterprise Services, 467-469
- service model, 466
- system services, 464-465

inheritance, 10

- COM versus .NET, 15

initialization code, where to place, 74**inserts (database access), 217-218****installation packages, creating, 432-433****InstallationFlags values, 428-429****installing. *See also* deployment**

- application proxies, 167-168
- assemblies in global assembly cache, 433
- message queuing servers, 325

integrated Windows authentication (IIS), 392**interactive user accounts, identity, 407****interception**

- method call interception, 39
- object activation interception, 38
- proxies, 38

interface ID (IID), 130, 146-147**interfaces, 22**

- COM interface types, 134
 - ComVisible attribute, 142-143
 - custom interfaces, 134-135
 - dispatch interfaces, 135-139
 - duel interfaces, 139-142
 - which to use, 142

- as contracts, 48
- defining, creating queued components, 336-338
- implementation, creating serviced components, 48
- queuable interfaces, parameters, 338

Interlocked class, 89**Internet Information Server. See IIS interoperability, COM and serviced components, 121-122**

- COM callable wrappers (CCWs), 122-123
- default COM configuration, 123-127
- runtimecallable wrappers (RCWs), 122-123
- type libraries, 127-131

isolated storage (state management), 294-296**isolation (code), 31-32****isolation (transactions), 239, 264**

- levels of, 29, 265-268
- problems with, 265

J-K**JITA (just-in-time activation), 65-70, 179**

- advantages of, 69
- transactions, 257

JITA objects, synchronization, 100**journal queues, 328****L****LAN (local area network), 3****language support, COM versus .NET, 13-14****late binding, 136**

- COM versus .NET, 16

layers, application layers, 4

- advantages of, 7-8
- business services, 6
- data services, 6
- presentation services, 5-6

LCE routing, 376-377**LCEs (loosely coupled events), 27, 355**

- architecture, 357
- event classes, 358
- filters, 359-360
- publishers, 358
- subscribers, 358
- filters
 - publisher filters, 375-376
 - subscriber filters, 373-375
- network distribution, 376-379
 - event class proxies, 378
 - LCE routing, 376-377
- overview, 355-358
- with queued components, 379-380
 - event classes, 380
 - subscribers, 380-382
- returning data, 353
- sample application, 360
 - event classes, 360-362
 - persistent subscribers, 364-367
 - publishers, 364
 - transient subscribers, 367-373
- versus WS-Eventing specification, 450

library applications, 44

- advantages of, 80
- application activation, 80-81
- debugging, 62
- identity, 405

lifetime of state, 291**listings**

- accessing ASP.NET cache from the serviced component, 310
- accessing call context in serviced component, 316
- accessing the SoapContext, 402
- AddCourse method with programmatic transactions, 243
- AddCourse method without programmatic transactions, 254
- AddCourse using the [AutoComplete] attribute, 259

- AddCourse using the property
 - MyTransactionVote, 258
- adding course date, 498
- adding new data with CourseData class, 218
- application attribute for
 - CourseRegistration.Control application, 510
- application configuration file, 371
- application configuration file to define connection string, 245
- applying roles to components, 412
- assembly attributes, 44, 336, 494
- assembly attributes (synchronization example), 101
- assembly attributes for .NET remoting access, 317
- AssemblyInfo.cs—application configuration, 280
- AssemblyInfo.cs—.NET remoting configuration, 175
- AttendeeInfo structure, 301
- [AutoComplete] attribute with exception handling, 259
- C++ client using custom interface, 152
- C++ client using dispatch interface, 153
- C++ COM object method—returning context ID, 156
- cache, using, 310
- calling stateless component, 190
- calling stored procedures, 220
- client applications, 60, 285, 302, 401
 - accessing synchronized objects, 103
 - instantiating recorder, 342
 - passing a custom SOAP header, 315
 - using cookies for the session, 306
 - using multiple threads, 76
- client test application, 244
- CommandHelper class, 485
- compensator component, 284
- configuration of custom token manager, 404
- configuration of WSE in Web.config, 402
- configuring .NET remoting for the client, 188
- construction strings, 64
- coordination context example, 455
- Course class, 479
- Course class for Web service, 195
- Course entity class, 208
- CourseComponent class, 49
- CourseControl class, 339
- CourseData class, 213, 487
- CourseList.aspx—OnRegisterCourse, 506
- CourseList.aspx—OnSelectCourse, 506
- CourseList.aspx—Page_Load, 506
- CourseManagement component, 126
- CourseManagerForm constructor, 496
- CourseRegistrationControl, 509
- create command objects for dataset updates, 229
- creating shared property groups, 300
- creating transient subscriptions, 372
- creating Windows service, 442
- custom collection class
 - CourseDateCollection, 211
- custom interface, 135
- custom SOAP header class
 - AuthenticationInfo, 399
- custom username token manager, 404
- CustomerControl component, 127
- declaring transactions with Indigo Services, 271
- defined types for Web service, 500
- defining custom SOAP header, 313
- defining roles, 409
- dispatch interface, 137
- dispatch interface with class attribute, 138
- displaying application properties, 438
- dual interface, 139
- dual interface with class attribute, 140
- entity class, 477
- EntityCreator class, 482

- event class, 361
- [ExceptionClass] attribute, using, 348
- extracting the UsernameToken, 403
- get courses, 497
- GetCourses method, 194
- helper class CacheControl, 505
- helper classes, 281
- helper methods to change UI element values, 369
- ICourseOrder interface, 48
- implementation of error component, 347
- implementation of interface
 - IServiceCall, 112
- implementation of Web method, 197
- Indigo service class, 468
- Indigo service contracts, 468
- Indigo with compatibility mode, using, 467
- iterating components in application collection, 439
- key-file attributes, 46
- methods for CoursesComponent class, 54
- .NET client application, 159, 170
- .NET remoting client, 187
- .NET remoting client using configuration file, 188
- .NET remoting configuration file, 184
- object deactivation, 68
- passing a username in the SOAP header, 462
- passing call context with client application, 318
- programmatically transactions, 242
- programming transactions with .NET 2.0, 270
- publisher component firing events, 364
- QueryCourseControl class, 493
- querying data with CoursesData class, 214
- querying data with parameter, 216
- queueable interface
 - IQueueableCourseRegistration, 338
- race conditions, 87
- read data from isolated storage, 295
- reading event information from configuration file, 371
- RegisterCourse.aspx—OnRegister method, 507
- RegistrationHelper class, using, 430
- returning context ID from .NET component, 158
- returning datasets, 224
- returning filenames of isolated storage, 296
- returning typed datasets, 226
- saving courses and dates, 499
- selection of course, 498
- serializable call context data, 315
- serializable classes, 124
- server-side configuration file allows passing objects to server, 189
- serviced component application to use with .NET remoting, 177
- serviced component class
 - CourseUpdateComponent, 255
- serviced component class using object pooling, 72
- serviced component class with just-in-time activation, 67
- ServiceDomain class, 115
- session configuration, 307
- session start and end events, 308
- shared properties, using, 302
- sink class for transient subscriber application, 370
- [SoapHeader] attribute, using, 400
- SOAP message with document literal style, 447
- SOAP message with RPC encoding style, 446
- SOAP request with SOAP header, 314, 400

- starting subscriptions, 373
- stateless serviced component for .NET
 - remoting access, 179
- stored procedure
 - GetActiveCoursesByCategory, 484
 - stored procedure InsertAttendee, 484
 - stored procedure UpdateCourseInfo, 220, 228
- subscriber component, 365
- synchronization using .NET contexts, 35
- synchronized class A, 102
- synchronized class B, 102
- synchronized method using the lock
 - statement, 87
- transactions with services without
 - components, 262
- typed datasets, 228
- untyped datasets, 225
- UpdateCourseControl class, 491
- updating data with CoursesData
 - class, 219
- updating datasets, 232
- updating datasets with multiple
 - tables, 233
- VBScript client, 155
- Visual Basic 6 client, 150
- Web service method GetCourses, 501
- Web service method RegisterCourse, 503
- Web service method with sessions, 305
- Web.config file to allow passing the call
 - context, 317
- worker component, 282
- wrapping COMAPI calls, 418
- write data to isolated storage, 295
- WS-Addressing example, 448
- WS-Eventing subscription example, 449
- WS-ReliableMessaging sequence
 - example, 452
- WSDL.exe-generated proxy class, 198
- XML CourseOrder example, 51
- literal style SOAP messages, 445**

- loading assemblies at runtime, 147**
 - COM registry entries, 148-149
 - private versus shared assemblies, 147
- local area network (LAN), 3**
- local service accounts, identity, 406**
- local system accounts, identity, 406**
- locking mechanisms, synchronized

 - access, 37**
- locks, 87-88**
 - causality-based locks, 107
- logical threads, 98**
- logical tiers versus physical systems, 8**
- LogRecord class, 276**
- loosely coupled events. *See* LCEs**

M

- manual deployment, 426**
- manual registration, creating serviced

 - components, 56-58**
- mapping, datagrid mapping to array

 - lists, 497**
- marshaling (.NET remoting), 174**
- message numbers,

 - WS-ReliableMessaging, 451**
- message queuing. *See also* queued

 - components
 - administration tools, 331-334
 - advantages, 325
 - characteristics of messages, 330
 - clients, 327
 - configuring queues, 331-332
 - message management, 333
 - overview, 322-324
 - product versions, 325
 - queue name syntax, 329
 - retries, 340
 - servers, 325-326
 - types of queues, 328-329
 - workgroup mode, 327
 - and WS-ReliableMessaging, 452**

messages, 11

- Indigo connector, 465
- .NET remoting, 172
- writing to event log, 431

messaging services, 321

- Indigo, 466-467
- Web services specifications, 444
 - SOAP, 444-448
 - WS-Addressing specification, 448
 - WS-Eventing specification, 449-450
 - WS-ReliableMessaging specification, 450-452

metabase, 23**metadata**

- COM versus .NET, 15
- .NET reflection, 132-133

method call interception, 39**methods, 22**

- Activity class, 112
- invoking dynamically, 132-133
- ServicedComponent class, 50-51

Microsoft Application Center, 9**Microsoft Data Engine (MSDE), 12, 294****Microsoft Transaction Server (MTS), 13, 18-20****Mixed Mode authentication (SQL Server), 388****monikers, 341**

- parameters, 343-345
- queued component client applications, 341-345

monitoring

- clerks (CRM), 287
- DTC, 268
- events in serviced components, 50
- object pooling, 78

MSDE (Microsoft Data Engine), 12, 294**MSMQ HTTPSupport option (message queuing server installation), 326****MTA (multithreaded apartments), 17, 94-95****MTS (Microsoft Transaction Server), 13, 18-20****multi-tier application development.**

See application layers

multiple assembly versions, 120-121**multiple tables in datasets, updating, 232****multiple threads for client applications, 76-77. *See also* synchronization****multithreaded apartments (MTA), 17, 94-95****multithreaded applications, synchronization issues, 88****N****NA (neutral apartments). *See* TNA (thread-neutral apartments)****namespaces, creating aliases, 502****naming**

- groups, 300
- message queues, 329

.NET

- assemblies. *See* assemblies
- clients, accessing serviced components, 120-121. *See also* client applications
- contexts, mixing with COM contexts, 41
- COM object calling .NET component, 42-43
- .NET component calling COM object, 41
- creating publisher filters, 376
- data provider, 204-206
- events, 356
- versus COM, 13
 - configuration, 18
 - error handling, 18
 - inheritance, 15
 - language support, 13-14

- late binding, 16
- metadata, 15
- object lifetime management, 14-15
- threading, 16-18
- .NET 2.0 transactions, 269-271**
- .NET components**
 - referencing, 157-158
 - synchronization, 97
 - activities, 97-98
 - configuring, 99-100
 - example, 100-106
 - reentrancy, 107-108
 - selecting options, 108
- .NET Enterprise Services**
 - automatic transactions, 24-26
 - COM+ 1.5 new features, 28-30
 - Component Services tool, 23
 - declarative programming, 23
 - history of, 13
 - COM versus .NET, 13-18
 - COM+ Services, 20-21
 - Microsoft Transaction Server (MTS), 18-20
 - loosely coupled events (LCEs), 27
 - metabase, 23
 - queued components (QCs), 26
 - role-based security, 27-28
 - scalability, 9
 - terminology, 21-22
- .NET Framework classes, synchronization, 88**
- .NET Passport authentication (IIS), 392**
- .NET reflection, 132-133**
- .NET remoting, 34-37, 172**
 - application proxies, 191
 - call contexts, client state management, 315-318
 - configuring, 184-186
 - Internet Information Server (IIS), 181-183
 - server application, 180
 - creating
 - client application, 186-190
 - serviced components, 175
 - disadvantages of, 190-191
 - Enterprise Services security, 420-421
 - operational overview, 172-175
 - performance, 191
 - security, 176, 181
 - serviced component class example, 176-179
- .NET runtime serialization, 339**
- network communication. *See* messaging**
- network distribution (LCEs), 376-379**
 - event class proxies, 378
 - LCE routing, 376-377
- network load balancing (NLB), 9**
- network service accounts, identity, 406**
- network transports (.NET remoting), 174**
- networking. *See* DCOM; .NET remoting; Web services**
- neutral apartments (NA). *See* TNA (thread-neutral apartments)**
- neutral threaded apartments (NTA). *See* TNA (thread-neutral apartments)**
- new features in COM+ 1.5, 28-30**
- NLB (network load balancing), 9**
- no-touch deployment. *See* ClickOnce deployment**
- nonrepeatable reads, 265**
- normal messages, 330**
- normalized data, 291**
- NTA (neutral threaded apartments). *See* TNA (thread-neutral apartments)**

O**object activation, 64**

- construction strings, 64-65
- JITA (just-in-time activation), 65-70
- object pooling, 71-76
 - advantages of, 79
 - client applications with multiple threads, 76-77
 - disposing of pooled objects, 78
 - monitoring, 78
 - private components, 79-80

object activation interception, 38**object lifetime management, COM versus .NET, 14-15****object pooling, 71-76, 179**

- advantages of, 79
- client applications with multiple threads, 76-77
- disposing of pooled objects, 78
- monitoring, 78

object-based collection classes, 210**object-oriented programming, 10****ObjectPooling attribute, properties, 71****OLE/COM Object Viewer, viewing type libraries, 127-131****outcomes (transactions), 250-251****overriding**

- Activate method, 66
- Deactivate method, 66

P**packages, creating installation****packages, 432-433****parameters, building SQL strings, 217****partitions, 29****passwords, storing encrypted, 390****performance**

- .NET remoting, 191
- SPM (Shared Property Manager), 303

persistent subscribers, 358

- sample LCE application, 364-367

phantoms, 265**physical systems versus logical tiers, 8****pooled objects, disposing of, 78. See****also object pooling****Populate method, 439****ports, Indigo connector, 465****presentation services (application layers), 5-6****priorities of messages, 330****private assemblies, 47**

- loading at runtime, 147

private components, 29, 79-80**private queues, 328****processes, 31**

- application domains, 34-35

product versions, message**queuing, 325****progid, 146****programmatic transactions, 241-246****projects in course management****sample application, 473-475****properties**

- ContextUtil class, 39-40
- EventClassAttribute properties, 363
- ObjectPooling attribute, 71
- ServiceConfig class, 110-111
- shared properties, 301-302

property groups, shared property**group manager, 298**

- creating shared property groups, 299-300
- property lock mode, 298
- property release mode, 299
- shared property groups, 301

property lock mode, 298**property release mode, 299****protected methods,****ServiceComponent class, 50-51**

proxies, 32-33

- application proxies, 163-164
 - creating, 164-167
 - installing, 167-168
 - .NET remoting, 191
- event class proxies, 378
- interception, 38
- .NET remoting, 172
- ServiceComponentProxy class, 41

proxy classes, creating, 198-199**public queues, 328****publisher filters, 360, 375-376****publisher policy assembly, 121****publishers, 358**

- application types, 364
- sample LCE application, 364

Q**queries (database access), 213-217**

- dynamic queries versus stored procedures, 221

queuable interfaces, parameters, 338**queued components (QCs), 26. See****also message queuing**

- client application, 341-345
- course management sample application, 509-510
- creating, 334
 - assembly attributes, 335-336
 - interface definition, 336-338
 - registration, 340
 - sample application, 338-339
- error handling, 346-349
- with LCEs, 379-380
 - event classes, 380
 - subscribers, 380-382
- overview, 322-324
- returning data, 350-352
- transactional queues, 349-350
- and WS-ReliableMessaging, 452

R**race conditions, 86****RCW (runtime callable wrapper), 122-123, 171**

- calling COM objects with serviced components, 157

read committed isolation level, 266**read uncommitted isolation level, 266****recorders, queued component client applications, 341-342****recoverable messages, 330****reentrancy, 107-108****referencing**

- .NET components, 157-158
- strongly named assemblies, 46

refreshing components, 145**registration**

- creating queued components, 340
- creating serviced components, 55
- dynamic registration, 56
- manual registration, 56-58
- serviced components with COM+ runtime, 158

registration message, [AutoComplete] methods, 69**registration service, 453****RegistrationConfig class, properties, 428****RegistrationHelper class, 429****registry entries, COM components, 148-149****regsvcs utility, 426**

- options, 57, 427

reliable messaging, Web services**specifications, 450-452****remote objects (.NET remoting), 173****remoting hosts (.NET remoting), 173****repeatable read isolation level, 266****report messages, 330****report queues, 329****required transactions, 249**

resource managers (distributed transactions), 239. See also CRMs

response queues, 329

retries, message queuing, 340

returning data
with loosely coupled events, 353
from queued components, 350-352

role-based security, 27-28

roles (authorization), 407
advantages of, 407
checking programmatically, 413-414
defining, 408-409
defining access, 410-413

rollback (transactions), 238

root objects, 250

routing
COM+ routing, 378
LCE routing, 376-377

Routing Support option (message queuing server installation), 326

RPC style SOAP messages, 445

runtime, loading assemblies, 147
COM registry entries, 148-149
private versus shared assemblies, 147

runtime callable wrapper (RCW), 122-123, 171
calling COM objects with serviced components, 157

runtime environment, 31. See also contexts

S

sample application. See course management sample application

sample database (data access), 206

SaveChanges method, 439

scalability of .NET Enterprise Services, 9

scale out systems, 8

scale up systems, 8

scripting clients, accessing serviced components, 154-155

secrets, storing, 390

secure data transfer (IIS), 392-393

security
ASP.NET Web applications, 393
authentication, 396-398
authorization, 398
identity, 394-395
ASP.NET Web services, 399
authentication with SOAP headers, 399-401
authentication with WSE (Web Services Enhancement), 401-404
authentication, 386
authorization, 386
confidentiality, 386
Enterprise Services, 405
authentication levels, 414
authorization, 407-414
identity, 405-406, 419-420
impersonation, 416-419
.NET remoting, 420-421
IIS (Internet Information Server), 391
authentication, 391-392
secure data transfer, 392-393
.NET remoting, 176, 181
overview, 386-388
role-based security, 27-28
SQL Server, 388-389
storing username and password, 390
Web services specifications, 460-463
Windows, 388

SecurityCallContext class, 419-420

[SecurityRole] attribute, 412

separating assemblies, 254

sequences, WS-ReliableMessaging, 451

serializable classes, listing, 124

serializable isolation level, 266

serialization
.NET runtime serialization, 339
XML serialization, 339

SerializationException, accessing serviced components with DCOM, 171

server applications, 44

- advantages of, 81
- application activation, 80-81
- configuring for .NET remoting, 180
- debugging, 62

server certificates, IIS security, 392**server-activated objects (.NET remoting), 174****servers**

- authentication levels, 414
- message queuing servers, 325-326

service model (Indigo), 466**service-oriented architecture (SOA), 9-11****ServiceConfig class, 110**

- properties, 110-111

serviced components

- accessing
 - ASP.NET state, 310-311
 - with C++ clients, 151-154
 - with DCOM, 162-171
 - with .NET clients, 120-121
 - with scripting clients, 154-155
 - with SOAP services (.NET remoting), 172-190
 - with Visual Basic 6 clients, 150-151
 - with Web services, 192-199
- calling COM objects, 155
 - C++ components, 156-157
 - client application, 158-159
 - RCW (runtime callable wrapper), 157
 - referencing .NET components, 157-158
 - registration, 158
- client application creation, 59-61
- COM interoperability, 121-122
 - COM callable wrappers (CCWs), 122-123
 - default COM configuration, 123-127
 - runtime callable wrappers (RCWs), 122-123
 - type libraries, 127-131

ComVisible attribute, 143

contexts, 37-38

- accessing information, 39-40
- method call interception, 39
- object activation interception, 38
- course management sample application, 489-495
- creating, 43
 - assembly attributes, 44-45
 - configuration, 58-59
 - global assembly cache, 47-48
 - interface implementation, 48
 - with .NET remoting, 175
 - registration, 55-58
 - ServiceComponent class, 49-55
 - strongly named assemblies, 46
- and dialog boxes, 127
- object activation, 64
 - construction strings, 64-65
 - JITA (just-in-time activation), 65-70
 - object pooling, 71-79
 - private components, 79-80
- services without components
 - within, 116
- stateful versus stateless, 292-293
- transactions without, 260-263. *See also* transactions

ServiceComponent class, creating serviced components, 49-50

- class attributes, 51-55
- methods, 50-51

ServiceComponentProxy class, 41**ServiceDomain class, 109**

- example, 115

services, Indigo connector, 465**services without components, 29, 109**

- Activity class, 110
- example, 112-116
- ServiceConfig class, 110
- ServiceDomain class, 109
- within serviced components, 116

session state (ASP.NET), 304-308

- shared assemblies, 47**
 - loading at runtime, 147
 - versioning, 121
- shared properties, 297, 301-302**
- shared property group manager, 297-298**
 - creating shared property groups, 299-300
 - property lock mode, 298
 - property release mode, 299
- shared property groups, 297, 301**
 - creating, 299-300
- Shared Property Manager. *See* SPM**
- SharedPropertyGroup class, 301**
- SharedPropertyGroupManager class, 298**
- signing assemblies, 46**
- single-threaded apartments (STA), 16, 91-94**
 - synchronization, 106
 - when to use, 97
- sink classes, 369**
- sink objects (.NET remoting), 172**
- size of state, 291**
- smart clients, 11, 13**
- smart pointers, 151**
- SOA (service-oriented architecture), 9-11**
- SOAP, 444-445. *See also* .NET remoting**
 - extensibility, 447-448
 - format, 11
 - headers, ASP.NET Web services authentication, 399-401
 - message styles, 445-447
 - requests, client state management, 312-315
 - services, 28
 - standard, 193
- [SoapHeader] attribute, 399**
- specifications. *See* Web services specifications**
- splitting updates (datasets), 232**
- SPM (Shared Property Manager), 297-298**
 - characteristics of, 303
 - client application, 302-303 and COM, 297
 - performance issues, 303
 - shared properties, 301-302
 - shared property group manager, 298-300
 - shared property groups, 301
- SQL injection, avoiding, 217**
- SQL Server**
 - data provider, 206
 - security, 388-389
- SQL strings, building, 217**
- STA (single-threaded apartments), 16, 91-94**
 - synchronization, 106
 - when to use, 97
- stable data, 291**
- state**
 - characteristics of, 290-292
 - maintaining with JITA components, 69
- state management**
 - ASP.NET, 304
 - accessing from serviced components, 310-311
 - cache, 308-310
 - characteristics of, 311
 - session state, 305-308
 - client state management, 312
 - .NET remoting call contexts, 315-318
 - SOAP requests, 312-315
 - with databases and files, 293-294
 - Active Directory, 296-297
 - isolated storage, 294-296
 - SPM (Shared Property Manager), 297-298
 - characteristics of, 303
 - client application, 302-303
 - performance issues, 303

- shared properties, 301-302
 - shared property group manager, 298-300
 - shared property groups, 301
 - stateful components, 290**
 - activities, 109
 - .NET remoting, 174
 - versus stateless components, 292-293
 - stateless components, 290**
 - .NET remoting, 174
 - versus stateful components, 292-293
 - static properties of ContextUtil class, 39-40**
 - storage (state management)**
 - Active Directory, 296-297
 - isolated storage, 294-296
 - stored procedures (database access), 220-221**
 - course management sample application, 483-484
 - versus dynamic queries, 221
 - storing secrets, 390**
 - streams (transactions), 250**
 - strongly named assemblies, creating serviced components, 46**
 - subscriber filters, 359, 373-375**
 - subscribers, 358**
 - persistent subscribers (sample LCE application), 364-367
 - as queued components, 380-382
 - transient subscribers (sample LCE application), 367-373
 - supported transactions, 248-249**
 - synchronization, 97. See also threading**
 - activities, 97-98
 - apartments, 90-91
 - MTA (multithreaded apartments), 94-95
 - STA (single-threaded apartments), 91-94
 - TNA (thread-neutral apartments), 95-97
 - configuring, 99-100
 - example, 100-106
 - Interlocked class, 89
 - locks, 87-88
 - in .NET Framework classes, 88
 - overview, 86-87
 - reentrancy, 107-108
 - selecting options, 108
 - [Synchronization] attribute, 89
 - and transactions, 257
 - [Synchronization] attribute, 89**
 - synchronization domain, 97**
 - SynchronizationAttribute class, 34, 99-100**
 - SynchronizationOption option, 99-100**
 - synchronized access, 35-37**
 - synchronous communication, 321**
 - system queues, 329**
 - system services (Indigo), 464-465**
- T**
- tables in databases, identifiers, 207**
 - timestamps, 207**
 - in datasets, 224
 - updating multiple, 232
 - TCP channel (.NET remoting), 191**
 - thread pools**
 - MTA (multithreaded apartments), 95
 - STA (single-threaded apartments), 92-93
 - thread-affinity, 97**
 - thread-neutral apartments (TNA), 17, 95-97**
 - when to use, 97
 - threads**
 - COM versus .NET, 16-18
 - multiple threads for client applications, 76-77
 - synchronization, 97
 - activities, 97-98
 - apartments, 90-97
 - configuring, 99-100

- example, 100-106
 - Interlocked class, 89
 - locks, 87-88
 - in .NET Framework classes, 88
 - overview, 86-87
 - reentrancy, 107-108
 - selecting options, 108
 - [Synchronization] attribute, 89
 - and Windows controls, 368-369
 - tightly connected events, 27**
 - timestamps in database tables, 207**
 - TNA (thread-neutral apartments), 17, 95-97**
 - when to use, 97
 - [Transaction] attribute, 247-249**
 - transaction isolation level, 29**
 - transaction managers (distributed transactions), 240**
 - transactional dead-letter queues, 328**
 - transactional objects, synchronization, 100**
 - transactional queues, 349-350**
 - transactional vote, setting, 258**
 - TransactionIsolationLevel values, 268**
 - TransactionOption values, 247**
 - behavior of, 248
 - transactions**
 - ASP.NET support, 264
 - atomicity, 238
 - automatic transactions, 24-26, 246
 - attributes, 247-249
 - [AutoComplete] attribute, 258-259
 - example, 253-257
 - MTS, 20
 - outcomes, 250-251
 - setting transactional vote, 258
 - transaction streams, 250
 - consistency, 238-239
 - distributed transactions, 239-241, 260
 - durability, 239
 - isolation, 239, 264
 - levels of, 265-268
 - problems with, 265
 - monitoring DTC, 268
 - with .NET 2.0, 269-271
 - overview, 237-238
 - programmatic transactions, 241-246
 - resource managers. *See* CRMs
 - Web services specifications, 452
 - WS-AtomicTransaction specification, 456-458
 - WS-BusinessActivity specification, 458-460
 - WS-Coordination specification, 453-456
 - without serviced components, 260-263
 - TransactionStatus values, 263**
 - transient subscribers, 358**
 - sample LCE application, 367-373
 - transport protocols, DCOM support, 162-163**
 - Triggers option (message queuing server installation), 326**
 - two-phase commit (2PC) protocol, 456-458**
 - two-phase commit process (distributed transactions), 239**
 - two-tier application development, 2-4**
 - type library (COM), 15**
 - viewing, 127-131
 - type-safe collection classes, 210-211**
 - typed datasets, 226-227**
 - versus untyped datasets, 234
- U**
- UDDI (Universal Description, Discovery and Integration), 193**
 - UI (user interface), 1**
 - unboxing and boxing, 214**
 - untyped datasets, 225**
 - versus typed datasets, 234
 - updates (database access), 219-220, 228-233**
 - use cases (course management sample application), 472-473**

user accounts and connection pools, 389
user identity. *See* identity
user interface (UI), 1
username, storing encrypted, 390
users, assigning to roles, 412
using directive, creating aliases, 502

V

versioning
 assemblies, 143-144
 COM components, 144-145
 APPID (application ID), 145
 CLSID (class ID), 145-146
 COMCompatibleVersion attribute, 147
 IID (interface ID), 146-147
 message queuing products, 325
 shared assemblies, 121
Visual Basic 6 clients, accessing serviced components, 150-151
volatile 2PC protocol, 458
volatility of state, 291
vtable, 134

W

Web applications (ASP.NET). *See* ASP.NET Web applications
Web service farms, session state in, 307
Web services, 192. *See also* ASP.NET Web services
 advantages of, 199
 course management sample application, 499-503
 creating proxy classes, 198-199
 implementing with ASP.NET, 194-198
 operational overview, 193
Web Services Description Language (WSDL), 181-182, 193

Web Services Enhancement (WSE), ASP.NET Web services authentication, 401-404
Web services specifications, 444
 messaging, 444
 SOAP, 444-448
 WS-Addressing specification, 448
 WS-Eventing specification, 449-450
 reliable messaging, 450-452
 security, 460-463
 transactions, 452
 WS-AtomicTransaction specification, 456-458
 WS-BusinessActivity specification, 458-460
 WS-Coordination specification, 453-456
well-known SingleCall objects (.NET remoting), 174
Windows authentication
 ASP.NET, 396
 SQL Server, 388
Windows controls and threads, 368-369
Windows Forms
 client applications (course management sample application), 495-499
 presentation services, 5
Windows security, 388
Windows Server 2003
 distributed transactions, 260
 SOAP configuration, 181
Windows services, 29
 configuring applications as, 441-442
 configuring server applications, 81
workers (CRM), 274-275
 CRM sample application, 282-283
workgroup mode, message queuing, 327
writing event log messages, 431
WS-Addressing specification, 448

WS-AtomicTransaction specification,
456-458

WS-BusinessActivity specification,
458-460

WS-Coordination specification,
453-456

WS-Eventing specification, 449-450

WS-Federation specification, 462

WS-ReliableMessaging specification,
450-452

WS-SecureConversation
specification, 462

WS-Security specification, 401

WS-Trust specification, 462

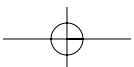
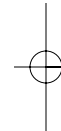
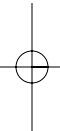
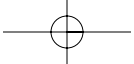
WSDL (Web Services Description
Language), 181-182, 193

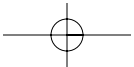
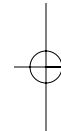
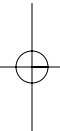
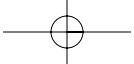
WSE (Web Services Enhancement),
ASP.NET Web services
authentication, 401-404

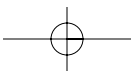
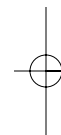
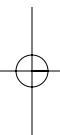
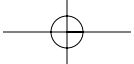
X-Z

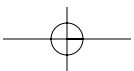
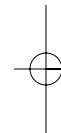
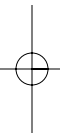
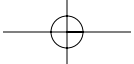
XML

dataset support for, 204
information set, 448
serialization, 339











THIS BOOK IS SAFARI ENABLED

INCLUDES FREE 45-DAY ACCESS TO THE ONLINE EDITION

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days.

Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

TO GAIN 45-DAY SAFARI ENABLED ACCESS TO THIS BOOK:

- Go to <http://www.awprofessional.com/safarienabled>
- Complete the brief registration form
- Enter the coupon code found in the front of this book on the "Copyright" page

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail customer-service@safaribooksonline.com.



Register Your Book

at www.awprofessional.com/register

You may be eligible to receive:

- Advance notice of forthcoming editions of the book
- Related book recommendations
- Chapter excerpts and supplements of forthcoming titles
- Information about special contests and promotions throughout the year
- Notices and reminders about author appearances, tradeshows, and online chats with special guests

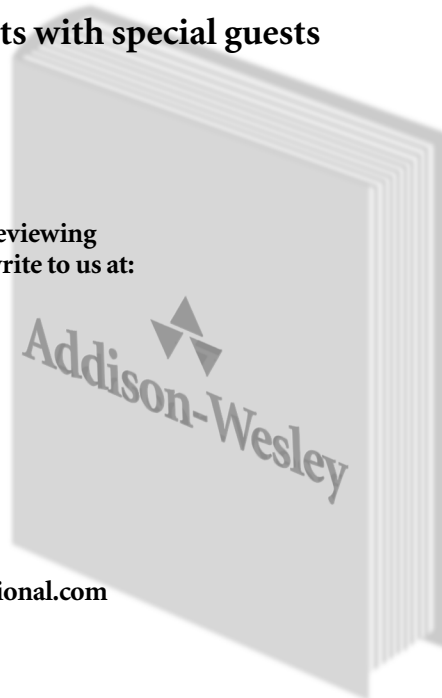


Contact us

If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

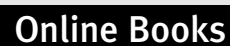
Editorial Department
Addison-Wesley Professional
75 Arlington Street, Suite 300
Boston, MA 02116 USA
Email: AWPro@aw.com

Visit us on the Web: <http://www.awprofessional.com>



The logo for InformIT, with 'inform' in a bold, lowercase sans-serif font and 'IT' in a larger, uppercase sans-serif font, all in black.The website address 'www.informit.com' in a white, lowercase sans-serif font, centered on a black horizontal bar.The main title 'YOUR GUIDE TO IT REFERENCE' in a bold, uppercase sans-serif font, centered on a light gray background.The section header 'Articles' in a white, bold, uppercase sans-serif font, centered on a black horizontal bar. To the left is a small image of a computer keyboard.

Keep your edge with thousands of free articles, in-depth features, interviews, and IT reference recommendations – all written by experts you know and trust.

The section header 'Online Books' in a white, bold, uppercase sans-serif font, centered on a black horizontal bar. To the left is a small image of a book cover.

Answers in an instant from **InformIT Online Book's** 600+ fully searchable on line books. For a limited time, you can get your first 14 days **free**.

The logo for Safari Tech Books Online, featuring the word 'Safari' in a large, bold, serif font with a small 'POWERED BY' above it, and 'TECH BOOKS ONLINE' in a smaller, bold, uppercase sans-serif font below it.The section header 'Catalog' in a white, bold, uppercase sans-serif font, centered on a black horizontal bar. To the left is a small image of a book catalog.

Review online sample chapters, author biographies and customer rankings and choose exactly the right book from a selection of over 5,000 titles.