

Index

A

- AbstractController **class**, 82
- AbstractEntityManager, 259
- access control**
 - admin and, 152
 - server-side forward and, 753
- acknowledgement strategy, JMS**, 403–404
- action classes, servlet-centric (Struts) architecture and**, 49
- action pages, JSP**, 47
- active cookie persistence**, 748
- Active Directory**, 466–467
- Active Server Pages (ASP)**, 7
- admin**
 - access control and, 152
 - server states, 542
- admin channels**
 - configuring, 572–575
 - overview of, 546
- admin ports**
 - configuring, 572–575
 - default, 529
- admin servers**
 - communication with node manager, 562
 - configuring ports and channels, 572–575
 - managing servers in a domain, 538–541
 - migrating, 640–641
 - overview of, 520
 - setting up new domains, 549–550
- administration, WebLogic Server**
 - application configuration, 575
 - application deployment, 590–592
 - bigrez.com Data Source configuration, 583
 - cluster configuration, 553–555
 - creating JMS connection factories, 588
 - creating JMS distributed destinations, 586–588
 - creating JMS servers and persistent stores, 584–586
 - cross-cluster replication, 555–557
 - database resource configuration, 575
 - domain configuration, 547–549
 - JavaMail session configuration, 588–589
 - JDBC DataSource configuration, 577–580
 - JDBC driver configuration, 576
 - JMS resource configuration, 583–584
 - JVM configuration, 567
 - locating data sources in Java EE applications, 582–583
 - modifying application configuration using deployment plans, 592–595
 - network channel configuration, 557–560
 - network channels, 545–547
 - operating system configuration, 566–567
 - overview of, 542
 - port and channel configuration, 572–575
 - security, 447–449
 - self-health monitoring, 544–545
 - server configuration, 551–553
 - server states, 542–543
 - setting up new domain, 549–551
 - setting up node manager, 560–566
 - transaction options for JDBC DataSource, 580–582
 - web server plugin configuration, 567–572
 - work manager, request classes, and constraint configuration, 589–590
- administration components, web applications**, 105
- administration controller form view overrides**, 122–123
- administration site**
 - bigrez.com, 58
 - display and source components, 74
 - weblogic.xml descriptor file, 133–137
 - web.xml descriptor file, 128–133
- administrative privileges, for running WLST commands**, 597
- advanced queuing (AQ)**, 438–440
- aggregation, cascading operations and**, 176
- AIX, tuning**, 649
- alarms, WLDF**, 627
- annotation-based programming, in EJB 3.0**, 156
- annotations**
 - cascading operations and, 176
 - dependency injections and, 223–225
 - vs. deployment descriptors, 221–222
 - referencing EJBs, 197–198
 - simple packaging and, 266
 - SLSB (stateless session beans) and, 159
- Ant build tool**
 - from Apache Software Foundation, 125
 - bigrez.com projects, 291–292
 - clientgen task, 319
 - creating web application archive file, 144–145
 - jwsc task, 311
 - persistent class enhancement, 173
 - wlappc and, 137
 - wldeploy task, 150
 - wSDL task, 321
 - wSDLc task, 314–316
- AOP (Aspect Oriented Programming)**
 - AspectJ, 623
 - Java SE and, 334
- Apache**
 - Ant build tool from, 125
 - Commons Logging API, 620–621
 - JMeter, 725

Apache web servers

Apache web servers

- common parameters, 570
- debugging parameters, 571–572
- plug-in configuration, 567–570
- response time parameters, 571

appc

- compiler utility, 137–139
- dynamic compilation feature and, 192–193
- persistent class enhancement, 173
- production builds with, 293–294

APP-INF directory, 285

application components, web applications, 73–75

application container, WebLogic Server, 529–530

application design

- application requirements, 55–56
- business domain models, 56–57
- business-tiers interfaces, 67–68
- controllers for calling business services, 68–69
- controllers for populating forms for JSPs, 69–70
- designing MDB-based applications, 431–432
- external view assembly using Tiles framework, 66–67
- master page assembly approach, 64–66
- overview of, 55
- presentation approach, 60–61
- presentation requirements, 57–59
- relationships in presentation components, 71–72
- review, 72
- self-assembly approach, 61–64
- updates requiring explicit service calls, 71
- Web application architecture and, 59

application design, JMS. See JMS applications, designing

application hotspots, 644

Application Library Directory, Java EE, 285

application management, WebLogic Server

- Commons Logging API, 620–621
- database failure management, 630–632
- exporting diagnostic data, 626
- failure management, 630
- harvesting metric information, 625–626
- instrumenting bigrez.com, 622–625
- logging services, 617–618
- message catalog logging, 618–619
- migrating admin server, 640–641
- migrating custom singleton services, 637–638
- migrating JMS services, 633–636
- migrating JTA service, 636–637
- migrating whole server, 638–640
- non-catalog logging, 620
- overview of, 615
- server failure management, 632
- troubleshooting, 615–617
- versioning applications, 628–630
- watches and notifications, 627–628
- WLDF (WebLogic Diagnostic Framework) and, 621–622

application monitoring, WebLogic Server

- command-line tool for, 600–601
- JMX programmatic monitoring, 603–606
- overview of, 595
- SNMP for, 607–614
- WebLogic Console for, 601–603
- WLST for, 595–600

application security, WebLogic Server

- managing EJB security, 493–495
- managing users and groups, 496
- models, 487
- overview of, 487, 496
- securing enterprise applications, 496
- securing J2EE CA resource adapters, 495
- securing resources, 496–497
- securing web applications, 488–493
- security models for, 454
- working with roles and policies, 497–500

application servers, troubleshooting

- identifying and correcting bottlenecks, 679
- throughput, 682–683

application servers, tuning, 661–665

- configuring connection-related parameters, 661–662
- optimizing thread management, 662–664
- overview of, 661
- resource pools and cache sizes, 664–665
- using native I/O muxer, 662
- verifying/adjusting settings, 665

application tiers, Java EE, 31–32

applications

- building EJB applications. *See* building EJB applications
- deploying Web Logic Server applications, 590–592
- enterprise. *See* enterprise applications
- system requirements when designing, 55–56
- web. *See* web applications

applications, WebLogic Server

- authentication with SSL, 486
- configuring, 575
- deployment, 590–592
- modifying application configuration using deployment plans, 592–595
- tuning, 645–646
- types of modules in, 287
- web application clustering, 531–536

application.xml descriptor file, 265, 278–280

application.xml file, 136

AQ (advanced queuing), 438–440

architecture

- business layer. *See* business layer architecture
- presentation-tier, 67–71
- web application. *See* web application architecture
- web services containers, 303–304

architecture, WebLogic Server

- deployment. *See* deployment architecture, WebLogic Server
- server. *See* server architecture, WebLogic Server

archive files

- creating EJB archive file, 265–266
- web application archive. *See* .war (web application archive) file

archives, managing diagnostic archive, 627

arguments, remote start attributes for WebLogic Servers, 563

ASP (Active Server Pages), 7

Aspect Oriented Programming (AOP)

- AspectJ, 623
- Java SE and, 334

AspectJ, 334, 623

assertions, SAML, 502

associations

building web applications

- bi-directional vs. unidirectional, 248
 - mapping, 231–233
 - asynchronous web services**
 - calling, 343–344
 - creating one-way web services, 344
 - returning multiple asynchronous web services, 344–346
 - Atlassian's Clover**, 716
 - attachments, SOAP**, 338–341
 - attribute statements, SAML**, 502
 - audit trails, in business layer requirements**, 235
 - AuditFilter, 130
 - auditing**
 - using WebLogic Server log file, 756
 - in WebLogic security, 459–460
 - authentication**
 - application authentication, 486
 - Basic Authentication, 302
 - choosing components for web application, 105–108
 - configuring web service security and, 359
 - custom authentication providers, 516
 - managing external LDAP authentication, 465–467
 - managing RDBMS authentication, 467–469
 - node manager and, 562
 - node manager requiring, 562
 - presenting role-based views of data, 36
 - providers, 455–457
 - servlet authentication filters, 517
 - transport-level security and, 350–351
 - WS-Security, 352
 - authorization**
 - choosing components for web application, 105–108
 - creating own authorization framework, 36
 - security providers, 458–459
 - AUTO_ACKNOWLEDGE, **JMS**, 366
 - autodeploy **directory**, 146
 - automatic deployment**
 - of admin server, 539
 - bigrez.com application, 297
 - disabling, 540
 - of web applications, 145–148
 - automatic failover, JMS servers**, 380
 - availability**
 - display for reservation process, 94–101
 - HA (high availability) in failure management, 630
 - maintenance pages, 117–122
 - simple WebLogic Server clusters and, 732
 - AvailabilityAndRates **objects**, 98, 100–101
- B**
- backups, replication and**, 205
 - bandwidth, clustering and**, 741
 - bare document style, SOAP**, 324–325
 - basic authentication**
 - securing web applications, 488
 - transport-level security and, 350–351
 - WS-Security and, 302
 - basic navigation**, 42
 - batch updates, databases**, 675
 - BEA Home, remote start attributes for WebLogic Servers**, 563
 - BEA WebLogic Server Tools project**, 705
 - bearer, SAML Token Profile**, 507
 - bi-directional relationships**
 - associations, 248
 - managing in OpenJPA, 219–220
 - mapping and managing, 175
 - mapping associations, 231
 - BigRez channel**, 557
 - bigrez.com
 - adding web services to, 359–361
 - altering to use shared library, 292–293
 - build targets for, 702
 - compiling production builds with appc, 293–294
 - components, 291
 - configuring Eclipse as IDE for, 705–710
 - configuring JDeveloper as IDE for bigrez.com, 711–714
 - Data Source configuration, 583
 - deploying, 297–298
 - importing into Eclipse, 706–707
 - instrumenting, 622–625
 - logical database design, 56–57
 - resources, 296
 - running from Eclipse, 707–708
 - bigrez.com **implementation**, 244–245
 - database schema, 245–247
 - domain model, 247–249
 - email integration, 255–258
 - generating JPA classes from database schema, 249–250
 - generic JPA functionality, 250–252
 - logging interceptor, 258
 - optimistic locking, 259–260
 - service implementation, 253–255
 - services, 252
 - session façade service, 252–253
 - TopLink and. *See* TopLink
 - unit tests, 258–259
 - binary files, sending as SOAP attachments**, 338
 - black box testing**, 719
 - bookmarks, presentation-tier controls**, 44–45
 - booting WebLog Server**, 500–501
 - bottlenecks, identifying and correcting**, 678–680
 - buffering JSP responses**, 11–12
 - build process**, 701–702
 - building EJB applications**
 - business layer architecture. *See* business layer architecture
 - business layer requirements. *See* business layer requirements
 - converting application for use with TopLink. *See* TopLink
 - implementing bigrez.com example. *See* bigrez.com implementation
 - overview of, 227
 - review, 264
 - building web applications**, 71
 - administration components, 105
 - administration controller form view overrides, 122–123
 - application components, 73–75
 - authentication/authorization components, 105–108
 - availability display and room-type selection page, 94–101
 - availability maintenance pages, 117–122
 - choosing basic controller components, 82–84
 - choosing form-processing controller components, 84–85

building web applications (continued)

building web applications (continued)

- choosing handler and resolver components, 85–88
- choosing Spring MVC components, 81
- constructing, 75–76
- constructing application skeleton, 75–76
- constructing user site components, 76
- core reservation process components, 79
- date selection page, 91–94
- defining navigation paths, 79–81
- Java servlets. *See* Java servlets
- JSP pages. *See* JSP (JavaServer Pages)
- overview of, 1, 73
- property main form, 109–114
- property maintenance components, 108–109
- property search/selection pages, 88–91
- rate maintenance pages, 114–117
- reservation creation process, 101–103
- reservation information components, 76–79
- review, 30, 123
- targeted offers components, 103–105
- build.xml file**
 - creating exploded file, 140–142
 - creating web application archive file, 144–145
 - establishing a build process, 701–702
 - precompiling JSP components, 137
- bulk updates, JPQL**, 179–180
- bundled libraries**
 - enterprise application directory structure, 277
 - overview of, 284–286
- business domain models**, 56–57
- business keys**, 170
- business layer architecture**
 - overview of, 235–236
 - SLSBs and session façade requirements, 236–237
 - SLSBs with CMP entity beans, 238, 240–242
 - SLSBs with JDBC, 237–239
 - SLSBs with JPA, 242–245
- business layer interface**, 228–229
- business layer requirements**
 - additional requirements, 235
 - business logic, 228–230
 - data access, 234–235
 - ORM, 230–234
 - overview of, 227–228
 - table of, 236
- business logic**
 - controllers and, 83
 - domain model and DTOs, 229–230
 - encapsulating in session beans, 98
 - encapsulation and, 228–229
 - overview of, 228
 - transaction management and security and, 229
 - validation and, 230
- business processes, SLSB (stateless session beans) and**, 159
- business services**, bigrez.com
 - email integration, 255–258
 - implementing, 253–255
 - logging interceptor, 258
 - overview of, 252
 - session façade, 252–253
 - unit tests, 258–259

business-tier

- controllers for calling business services, 68–69
- controllers for populating forms for JSPs, 69–70
- interacting with presentation tier for updating model objects, 41–42
- interfaces with presentation-tier, 67–68
- Java EE application tiers, 32
- updates requiring explicit service calls, 71
- BytesMessage object, JMS**, 370, 401–402

C

- C Client, JMS clients**, 381
- CA (certificate authority)**, 470–472, 574–575
- CacheFilter, **servlet filtering**, 24–26
- Cache-Flush **policy, writing to persistent stores**, 390
- caching. *See also* data caches**
 - cache size, 664–665
 - dynamic content caching of JSPs, 669
 - GemFire product for, 217
 - HTTPResponse caching, 24–26
 - idle-timeout-seconds property and, 203–204
 - increasing performance and throughput with, 645
 - LRU (least recently used), 203, 579
 - NRU (not recently used), 203
 - prepared statement caching, 220–221, 673–674
 - SFSB (stateful session bean), 202–204, 271
- Cactus, Jakarta**, 719
- call router objects**, 536
- Canoo WebTest**, 722
- capacity constraints, work managers**, 524
- cascading operations, JPA**, 176–177
- certificate authority (CA)**, 470–472, 574–575
- Certificate Revocation Lists (CRLs)**, 476
- certificates, WS-Security and**, 355–357
- certification path, for X.509 certificates**, 460–461
- CGI (Common Gateway Interface)**, 2
- change detection/propagation, in WebLogic Server clusters**, 535–536
- channels**
 - configuring admin channel, 572–575
 - configuring network channels, 557–560
 - default, 529
 - network channels, 545–547
- Checkstyle**, 704
- Class D addresses, for multicast messages**, 419
- classes**
 - customizing class loading, 289–291
 - generating JPA classes from database schema, 249–250
 - Java classes as servlets, 2–3
 - mapping simple classes, 230–231
 - Spring MVC controller classes, 51
- classloaders**
 - customizing, 289–291
 - stand-alone deployment and, 275–276
 - system classloader, 530
- classpath, remote start attributes for WebLogic Servers**, 563
- Class-Path header, **bundled libraries and**, 284–286
- client identifiers, for JMS subscribers**, 379
- Client Reconnect attributes, JMS connection factories**, 386

Container Managed Persistence (CMP)

- client tiers, Java EE**, 32
- `CLIENT_ACKNOWLEDGE`, 366
- client-cert**, 489
- `clientgen` **task, Ant build tool**, 319
- clients**
 - associating service handlers with service clients, 338
 - developing web service clients, 318–322
 - fat and thin, 749
 - JMS. *See* JMS clients
 - security-aware Java clients. *See* Java clients, security-aware
 - WebLogic Test Client, 313
- client-side validation**, 38–39
- Clover code coverage tool, Atlassian's**, 716
- cluster-aware stubs**, 536
- clustering, JMS. *See* JMS clustering**
- clustering, WebLogic Server**
 - configuring clusters, 553–555
 - cross-cluster replication, 555–557
 - design options, 738
 - designing multiple-site WebLogic clusters, 738–741
 - detecting and propagating changes, 535–536
 - implementing clusters that span multiple sites, 741–742
 - implementing complex cluster for single-site deployment, 734–737
 - implementing one cluster per site, 743–745
 - implementing simple cluster for single-site deployment, 732–734
 - increasing performance and throughput with, 645
 - monitoring with WebLogic Console, 602
 - overview of, 520, 530–531
 - for RMI-based applications, 536–538
 - selecting secondary server, 534–535
 - session-based routing, 532–534
 - for web applications, 531
- CLV (certificate lookup and validation) framework**, 460–461
- CMP (Container Managed Persistence)**
 - entity beans, 238, 240–242
 - in JPA history, 163–164
- Cobertura code coverage tool**, 716
- code, co-developing unit tests with**, 716–717
- code coverage tools**, 716
- code inspection tools**, 704
- code-first web services**
 - developing, 307–314
 - overview of, 306
- Coherence, Oracle**
 - overview of, 217
 - using, 218–219
- Cold Fusion**, 7
- columns, retrieving database columns explicitly**, 673
- command classes, Spring MVC**, 51
- command pattern**, 667
- command-line tool, for monitoring WebLogic Server applications**, 600–601
- comments, JSP vs. HTML**, 755
- commit protocols**, 581–582
- commitment control level, database access best practices**, 675
- Common Gateway Interface (CGI)**, 2
- Common Secure Interoperability Version 2 (CSIV2)**, 445
- Commons Logging, WebLogic Server logging services**, 620–621
- community names, SNMP**, 608–609
- compiling production builds, with appc**, 293–294
- composite view pattern, for bizgrez.com example**, 60–61
- composition**
 - cascading operations and, 176
 - mapping associations and, 232–233
- compression, JMS**, 402
- concurrency**
 - data access requirements, 235
 - locking options and, 181–182
 - MDBs (message-driven beans) and, 427–428
 - optimistic locking, 182–186
- concurrency testing, unit tests**, 724–725
- concurrent collectors, garbage collection**, 654, 659
- confidentiality, WS-Security**, 352
- Config MBeans, WLST**, 598
- configuration files, controlling navigation with**, 49
- `configWss.py`, 359
- connection factories, JMS**
 - attributes of, 385–386
 - creating, 588
 - overview of, 365
- connection filters, 751–753**
 - built-in, 751–752
 - custom, 752–753
 - overview of, 751
- connection pools, database**
 - contention, 662
 - database access best practices, 676
 - identifying and correcting bottlenecks, 680
- connection routing, in JMS clustering**, 371–372
- connection-related parameters, tuning application servers**, 661–662
- connections**
 - controlling access to WebLogic Server, 751
 - JMS, 365
 - monitoring with WebLogic Console, 602
- Connector Architecture, J2EE**, 495
- `ConnectRetrySecs`, **web server plug-in parameters**, 570
- `ConnectTimeoutSecs`, **web server plug-in parameters**, 570
- Consensus-based leasing**, 634
- console extension component, of WLDF**, 622
- constraints**
 - configuring, 589–590
 - defining, 525–526
 - specifying security constraints, 462
 - work managers, 524–525
- consumers, JMS**
 - asynchronous, 368
 - balancing with producers, 664
 - consuming messages from a distributed queue, 375–377
 - consuming messages from a distributed topic, 378–380
 - overview of, 364, 367–369
 - zero consumer queues, 375
- Container Managed Persistence (CMP)**
 - entity beans, 238, 240–242
 - in JPA history, 163–164

containers

containers

- component technologies and, 155
- EJB container. *See* EJB container, WebLogic Server
- web services container architecture, 303–304
- context request class, WebLogic Server**, 523
- context root**, `application.xml` file and, 136
- continuous integration, in development process**, 702–703
- control flags, authentication providers**, 455–456
- controller classes, Spring MVC**, 51
- controllers**
 - administration controller form view overrides, 122–123
 - for calling business services, 68–69
 - choosing components for web application, 82–84
 - choosing form-processing controller components, 84–85
 - for populating forms for JSPs, 69–70
 - `PropertyAvailabilityController`, 119–121
 - `PropertyMainController`, 109–113
 - `PropertyRatesController`, 114–117
 - `PropertySearchController`, 88–90
 - `ReviewReservationController`, 101–102
 - `SelectDatesController`, 92
 - `SelectRoomTypeController`, 96–97
 - Spring MVC, 51
- cookies**
 - persistence, 748
 - session ID cookie safety, 753–754
 - session tracking and, 5
- cookies, session tracking and**, 342–343
- Copying collector, garbage collection**, 654
- core dumps**
 - defined, 685
 - reading, 686–687
- core library, JSLT, 19
- core reservation process components. *See* reservation process**
- correlation IDs, request/reply messaging and**, 422–423
- CPUs**
 - JVM-to-processor ratio, 730
 - troubleshooting CPU utilization, 682–683
 - vertical scaling, 730–731
- create, read, update, delete (CRUD)**
 - data access requirements, 234
 - encapsulating repetitive CRUD operations, 255
 - `createNamedQuery()`, `EntityManager`, 178
 - `createQuery()`, `EntityManager`, 178
 - `createQueue()`, 400
 - `createReservation` service, 103
 - `createTopic()`, 400
- credential mapping, in WebLogic security**, 460
- CRLs (Certificate Revocation Lists)**, 476
- cross domain security, setting up**, 510–511
- cross domain user**, 511
- cross-cluster replication**, 555–557
- CRUD (create, read, update, delete)**
 - data access requirements, 234
 - encapsulating repetitive CRUD operations, 255
- Cruise Control**, 703
- CSA (customer services agents), configuring network channels**, 557–559
- CSlv2 (Common Secure Interoperability Version 2)**, 445
- CSR (Certificate Signing Request)**, 470–471

- custom authentication providers**, 516

custom tags

- calling in JSP pages, 16
- components of, 20
- development and use, 21–22
- directory structure and, 126
- sources for, 22
- “tag files” differentiated from, 20–21
- unique capabilities of JSP pages, 12
- using JSTL tags to reduce scriptlet code, 15
- customer services agents (CSA), configuring network channels**, 557–559
- customizing class loading, packaging enterprise applications**, 289–291

D

- Dali utility (Eclipse IDE)**, 249
- DAOs (data access objects)**, 254–255
- data access, business layer requirements**, 234–235
- data access objects (DAOs)**, 254–255
- data caches**
 - configuring for multiple JVMs, 218
 - overview of, 217–219
 - tuning, 217–218
 - using Oracle Coherence, 218–219
- data flow, in application architecture**, 749
- data replication requirements, in multiple site deployment strategy**, 740–741
- data sources**
 - configuring for `bigrez.com`, 583
 - configuring JDBC data source for use with WebLogic Server, 577–580
 - locating in Java EE applications, 582–583
 - monitoring with WebLogic Console, 602
 - Multi Data Source support, 631–632
 - transaction options for JDBC data source, 580–582
- data tier**, 32
- database access best practices**
 - batch updates, 675
 - caching prepared statements, 673–674
 - commitment control level, 675
 - connection pools, 676
 - overview of, 672
 - principles of, 673
 - releasing JDBC resources, 676–677
 - retrieving columns explicitly, 673
 - transaction model, 674
- database schema**
 - `bigrez.com`, 245–247
 - embedded entity classes and, 248
 - generating JPA classes from, 249–250
 - making changes to BigRez schema, 252
- database servers**
 - development environment hardware and software requirements, 692
 - identifying and correcting bottlenecks, 680
 - sharing between developers, 693–694
- database-backed replication, for cross cluster replication**, 739
- database-based leasing**, 634

development environment

databases

- bulk updates, 179–180
- configuring for use with WebLogic Server, 575
- database services in unit test context, 721
- failure management, 630–632
- locking, 181–182
- mapping inheritance in, 233–234
- transaction isolation levels, 199–201

date selection page, 91–94**DCR (dynamic class redefinition)**, 700Debug **parameter, web server plug-in**, 571–572DebugConfigInfo **parameter, web server plug-in**, 571–572**debugging**

- with IDEs, 714–715
- SAML, 510
- security framework debug scope, 464
- SSL (Secure Sockets Layer), 477–478

deep fetches, ORM, 232**default channels**, 546**Default Message Delivery attributes, JMS connection factories**, 385**default network configuration**, 529DefaultFileName, **web server plug-ins**, 570**delivery overrides, JMS messages**, 392–393**demilitarized zones (DMZ)**, 750–751**denial-of-service, server configuration and**, 551**Denial-of-Service attack**, 755**dependency injection**

- annotations, descriptors, and plans and, 223–225
- in EJB 3.0, 157
- unit tests and, 721

Deployer utility, WebLogic, 148–150**deploying enterprise applications**

- automatic deployment, 297
- creating required services, 295–296
- overview of, 294
- review, 298–299
- WebLogic Console deployment, 298
- weblogic.Deployer, 297–298

deploying web applications

- automatic deployment, 145–148
- creating required users and groups, 152–153
- overview of, 145
- review, 154
- WebLogic Console deployment, 150–151
- WebLogic Deployer utility, 148–150

deploying Web Logic Server applications, 590–592**deployment architecture, WebLogic Server**

- admin server, 538–541
- application container, 529–530
- clustering for RMI-based applications, 536–538
- clustering for web applications, 531–536
- domain architecture, 519–521
- execute queues, execute threads, and work managers, 521–527
- listen ports and listen threads, 529
- node manager, 541–542
- overview of, 519
- server architecture, 521
- server clustering architecture, 530–531
- socket muxer, 527–529

deployment descriptors

- vs. annotations, 221–222
- binding to logical JMS message destination, 268–270
- dependency injections and, 223–225
- free pool settings, 271
- JNDI bindings and references to EJBs and other resources, 267–268
- in Kodo, 209–210
- MDB settings, 272
- overview of, 266–267
- security settings, 270
- SFSB cache settings, 271
- transaction settings, 270–271
- tuning and optimization settings, 272–273

deployment plans

- dependency injections and, 223–225
- EJB, 222–223
- modifying application configuration with, 592–595

deployment strategies

- complex cluster for single-site deployment, 734–737
- designing multiple-site WebLogic clusters, 738–741
- evaluating, 728–729
- implementing clusters that span multiple sites, 741–742
- implementing one cluster per site, 743–745
- multiple site deployment strategies, 737–738
- overview of, 727–728
- server deployment, 729–731
- simple cluster for single-site deployment, 732–734
- single-site deployment, 731–732

deployment/redeployment feature, EJB container, 192**DER (Distinguished Encoding Rules), for certificates**, 472**descriptor files, enterprise applications**

- application.xml descriptor file, 278–280
- overview of, 278
- weblogic-application.xml descriptor file, 280–282

descriptor files, web applications

- overview of, 128
- weblogic.xml, 133–137
- web.xml, 128–133

design patterns

- designing for performance, 665–666
- Java EE, 33
- MVC (model-view-controller), 32–33

designing Java EE application. See application design**destinations, JMS**

- choosing destination type, 398
- connecting to distributed destinations, 430–431
- consumers/producers exchanging messages at, 365–366
- creating distributed destinations, 586–588
- destination keys, 387
- distributed, 373–374
- error destinations, 409
- first-in-first-out ordering or last-in-last-out ordering, 387
- locating, 399–400
- reply to destinations, 421
- terminal destinations, 416

developer workstations, 692**development environment**

- build process, 701–702

development environment (*continued*)

development environment (*continued*)

- code inspection and reporting tools, 704
- continuous integration, 702–703
- creating/configuring WebLogic server domain, 696–697
- FastSwap, 700–701
- hardware and software requirements, 692–693
- IDEs. *See* IDEs (integrated development environments)
- installing WebLogic Server software, 695
- overview of, 691
- project structure, 697–698
- review, 725
- separating from production environment, 754
- sharing database server, 693–694
- split directory for, 698–700
- streamlining development cycle, 698
- unit tests. *See* unit tests
- development servers**, 692
- diagnostic archive component, WLDF**
 - managing diagnostic archive, 627
 - overview of, 621
- diagnostics. *See also* WLDF (WebLogic Diagnostic Framework)**
 - diagnosing application errors, 617
 - exporting diagnostic data, 626
 - managing diagnostic archive, 627
- diagnostics image capture component, WLDF**, 622
- directories**
 - autodeploy directory, 146
 - enterprise applications and, 277
 - mapping information, 136
 - packaging and deploying web applications and, 126–128
 - split directory for development environment, 698–700
- Direct-Write **policy, writing to persistent stores**, 390
- Disabled **policy, writing to persistent stores**, 390
- disaster recovery**, 555
- Discard, **JMS message expiration**, 408
- disconnected objects, updates requirements and**, 71
- disk I/O**, 684
- Dispatch **APIs (JAX-WS)**
 - creating dynamic web services, 328–332
 - when to use `Provider/Dispatch` approach, 333–334
- dispatch policy, work managers**, 136
- DispatcherServlet, **startup and mapping information in**, 131
- display requirements, presentation-tier**
 - displaying list of model objects, 35–36
 - displaying model data, 35
 - flexibility and maintainability of display, 37–38
 - internationalization and, 36–37
 - overview of, 34
 - presenting role-based views of data, 36
- Distinguished Encoding Rules (DER), for certificates**, 472
- distributed applications, EJB supporting**, 156
- distributed destinations, JMS**
 - clustering and, 373–374
 - connecting to, 430–431
 - creating, 586–588
- DMZ (demilitarized zones)**, 750–751
- DNS**, 749
- document, **SOAP styles**, 323–324
- domain model**

- `bigrez.com`, 247–249

- implementing business logic with, 229–230
- superclasses in `bigrez.com` domain model, 252

domain objects, 74

- DomainEntity
 - generic JPA functionality, 250–252
 - inheriting from, 259

domains, WebLogic Server

- administering security at domain level, 447–448
- architecture of, 519–521
- configuring, 547–549
- creating/configuring for development environment, 696–697
- cross domain security, 510–511
- managing servers in, 538–541
- overview of, 520
- setting up new, 549–551

DOM-based API

- for dynamic web services, 331–332
- `getAttachments()`, 339

Don't Repeat Yourself principle, 175, 255

drivers, JDBC, 576

DTOs, domain model and, 229–230

`DUPS_OK_ACKNOWLEDGE`, **JMS message acknowledgement**, 366

duration subscriptions

- dealing with, 429–430
- JMS messages, 364, 368
- message persistence and, 392

dynamic compilation

- EJB container, 192–193
- simple packaging and, 266

dynamic content caching, JSP, 669

dynamic queries, JPQL, 178

Dynamic Shared Objects, Apache support for, 568

dynamic web services, creating, 327–334

E

eager fetching

- in JPA, 180–181
- in Kodo, 212

.ear (enterprise application) files

- `bigrez.com` application, 297
- exploded deployment and, 282
- packaging applications and, 265
- packaging enterprise applications, 274
- passing components by reference, 199
- referring to EJB components in same application, 195–197
- referring to external EJB components, 198

Eclipse

- configuring as IDE for `bigrez.com`, 705–710
- development tools, 704
- importing `bigrez.com`, 706–707
- publishing (deploying to server), 708
- running `bigrez.com` from, 707–708
- target runtime in, 706
- Virtual EAR, 709–710

Eclipse IDE (Dali utility), 249

Eclipse Persistence Platform (EclipseLink), 164, 290

Eclipse Web Tools Platform Project, 249

EntityManager

- Edit MBean, WLST**, 599
- EJB (Enterprise JavaBeans)**
 - application programming in JMS, 424–427
 - archive file, 265–266
 - bindings and references to, 267–268
 - building EJB applications. *See* building EJB applications
 - clustering EJB applications, 536–538
 - component types, 157
 - dependency injections, 223–225
 - deployment descriptors vs. annotations, 221–222
 - deployment plans, 222–223
 - `ejb-local-ref` elements, 132–133
 - interceptors, 162–163
 - managing EJB entities. *See* JPA (Java Persistence API)
 - MDB (message-driven bean) components, 161–162
 - monitoring EJB instances, 602–603
 - optimizing EJB 3.0 entities, 671
 - out-of-container testing, 718–719
 - overview of, 155
 - QL (query language), 177
 - review, 225
 - security management, 493–495
 - service components, 74
 - SFSB (stateful session bean) components, 159–161
 - SLSB (stateless session bean) components, 158–159
 - The Spring Framework, 157
 - technology of, 155–156
 - version 3.0, 156–157
 - web service standards, 302
- `@EJB` **annotation**, 197
- EJB container best practices**
 - JNDI lookup strategies, 670–671
 - optimizing EJB 3.0 entities, 671
 - overview of, 670
 - pass-by semantics, 671–672
- EJB container, WebLogic Server**
 - calling EJB components by reference, 198–199
 - controlling flush behavior in OpenJPA, 219
 - deployment descriptors in Kodo, 209–210
 - deployment/redeployment feature, 192
 - dynamic compilation feature, 192–193
 - eager fetching in Kodo, 212
 - EJB lifecycle in, 190–192
 - fetch groups in OpenJPA, 210–212
 - handles to EJBs, 207
 - idempotent methods, 207
 - inheritance strategies, 220
 - JDBC transaction isolation levels, 199–201
 - large result set proxies in OpenJPA, 216
 - lock groups in Kodo, 214–215
 - managing inverses in OpenJPA, 219–220
 - MDB (message-driven bean) features, 208–209
 - OpenJPA and Kodo features, 209
 - overview of, 189–190
 - prepared statement caching in OpenJPA, 220–221
 - queries with large result sets in OpenJPA, 215–216
 - reference to external EJB, 198
 - references between EJBs in same application, 195–197
 - remote business interfaces and JNDI, 193–195
 - second level data caches in Kodo, 217–219
 - SFSB cache management, 202–204
 - SFSB in-memory replication, 204–207
 - SLSB pooling, 201–202
 - state comparison version strategy in OpenJPA, 213–214
 - support for version field types in OpenJPA, 212–213
- EJB deployment descriptors**
 - binding to logical JMS message destination, 268–270
 - free pool settings, 271
 - JNDI bindings and references to EJBs and other resources, 267–268
 - MDB settings, 272
 - overview of, 266–267
 - security settings, 270
 - SFSB cache settings, 271
 - transaction settings, 270–271
 - tuning and optimization settings, 272–273
- EJBGen tool**, 164
- `ejb-jar.xml`
 - referring to EJB components in same application, 195–196
 - referring to external EJB components, 198
- `ejb-link`, 196–197
- `ejb-local-ref`
 - referring to EJB components in same application, 195–196
 - resource references and, 136
- `ejb-ref`
 - referring to EJB components in same application, 195
 - referring to external EJB components, 198
- `ejb-reference-description`, 198
- EL (Expression language), used in JSTL calls**, 16–18
- email integration, in bigrez.com**, 255–258
- embedded entity classes**
 - fine-grained object model with, 248
 - JPA, 180
- embedded LDAP server**, 450–451
- EMMA code coverage tool**, 716
- Emulate Two-Phase Commit, **transaction commit protocols**, 581
- encapsulation, business layer interface and**, 228–229
- encoded, **SOAP and**, 324
- encryption**
 - message-level security and, 357
 - web service security and, 359
- enhancer tool, Kodo**
 - disabling when switching to TopLink, 261
 - OpenJPA and Kodo and, 173
- enterprise applications**
 - deploying. *See* deploying enterprise applications
 - files. *See* `.ear` (enterprise application) files
 - packaging. *See* packaging enterprise applications
 - securing, 496
 - types of modules in, 287
- Enterprise Service Bus (ESB)**, 348
- entities**
 - embedded entity classes in JPA, 180
 - JPA entities in Java SE environment, 171–173
- EntityManager
 - `createNamedQuery()` and `createQuery()`, 178
 - creating, 168
 - `find()`, 177
 - `getEntityManager()`, 254

EntityManager (continued)

EntityManager (continued)

- injecting into SLSB, 169
- Java SE obtaining, 172
- JPA (Java Persistence API), 165–166
- lock(), 186
- merge(), 181, 187
- EntityManagerFactory, 168, 172
- error destinations, JMS**, 409, 411–412
- error handling**
 - data access requirements, 234
 - date selection and, 94
 - error pages in web.xml, 132
 - MDBs (message-driven beans) and, 258
 - presentation-tier mechanism for displaying errors, 40–41
 - web applications, 13–15
- error-page **element**, web.xml, 15
- ErrorMessage **parameter, web server plug-in**, 570
- ESB (Enterprise Service Bus)**, 348
- Excel files, creating with servlets and JSP pages**, 26–29
- exceptions. See error handling**
- exclusive concurrency strategy**, 183
- execute queues**, 521–527
- execute threads**, 521–527
- expiration strategy, for JMS messages**, 407–409
- explicit invalidation, session management and**, 668
- explicit locking, JPA**, 186
- exploded deployment**
 - overview of, 282–284
 - packaging enterprise applications, 274
 - split directory development and, 698
- exploded web application**, 139–144
- exporting diagnostic data**, 626
- Expression language (EL), used in JSTL calls**, 16–18
- external LDAP servers**, 465–467
- external providers, JMS. See JMS (Java Message Service), external providers**
- external view assembly, using Tiles framework**, 66–67
- Extreme Programming (XP)**, 716

F

- failed server states**, 542–543
- failover**
 - automatic failover of JMS servers, 380
 - in JMS clustering, 371–372
 - Multi Data Source support and, 632
 - replication and, 205
 - simple WebLogic Server clusters and, 733
 - WebLogic Server clustering and, 530
- failure management, WebLogic Server**
 - database failures, 630–632
 - migrating admin server, 640–641
 - migrating custom singleton services, 637–638
 - migrating JMS services, 633–636
 - migrating JTA service, 636–637
 - migrating whole server, 638–640
 - overview of, 630
 - server failures, 632
- fair share request class, WebLogic Server**, 523
- FastSwap**
 - classloaders compared to, 290
 - development environment and, 137
 - exploded deployment and, 283
 - lib classloader and, 276
 - redeployment with, 700–701
- fat clients**, 749
- federation services, SAML**, 504–506
- fetch groups, OpenJPA**, 210–212
- fetch joins, fixing incompatibilities between Kodo and TopLink**, 262
- field types, version strategies and**, 212–213
- FIFO (first-in-first-out), JMS**, 387, 395
- file-based stores, JMS persistent stores**, 389–390
- files**
 - archive files. *See* archive files
 - descriptor files. *See* descriptor files
 - enterprise application files. *See* .ear (enterprise application) files
- filters**
 - AuditFilter, 130
 - authentication filters, 517
 - classloader, 290
 - web service handlers compared with servlet filters, 334
- filters, servlet**, 22–26
 - CacheFilter, 24–26
 - overview of, 22–23
 - SnoopFilter, 23–24
- find(), of EntityManager, 177
- FindBugs**, 704
- firewalls**
 - identifying and correcting bottlenecks, 679
 - layout of, 750–751
 - network channels supporting, 546
- first-in-first-out (FIFO)**, 387, 395
- flexibility, of display**, 37–38
- flow control, JMS messages**, 396–398
- flush behavior, in OpenJPA**, 219
- fmt **library, JSLT**, 19
- forced stop scripts, in WebLogic Server clusters**, 734–737
- Foreign Server, JMS**, 437–438
- form beans, in servlet-centric (Struts) architecture**, 49
- form-based authentication**, 488
- form-processing controller**, 84–85
- forms**
 - administration controller form view overrides, 122–123
 - controllers for populating forms for JSPs, 69–70
 - displaying model data in, 35
 - extracting form data, 39–40
 - session tracking and hidden fields, 5
- form/update requirements, presentation-tier**
 - client-side validation, 38–39
 - displaying errors, 40–41
 - extracting form data, 39–40
 - interacting with business tier for updating model objects, 41–42
 - overview of, 38
 - server-side validation, 40
- free pool settings, EJB deployment descriptors**, 271
- functions **library, JSLT**, 19

IDEs (integrated development environments)

G

GC (garbage collection)

JRockit models, 659
 JVM-to-processor ratio and, 730
 troubleshooting, 616–617
 tuning JVMs, 652–654

GemFire, as caching product, 217

generated values, *Person id* field, 170–171

generational collectors, garbage collection, 654, 659

`getAttachments()`, in **DOM-based API**, 339

`getSession()` method, of `HttpServletRequest`, 6

GLI (globalization, localization, and internationalization), 36–37

global and local traffic management

overview of, 745
 using global load balancers with WebLogic Server, 748–749
 using load balancers, 745–746
 using load balancers with WebLogic Server, 747–748

global load balancers

traffic management with, 747
 using with WebLogic Server, 748–749

global reservation systems (GRS), configuring network channels, 557

global transactions, data sources supporting, 580–581

global trust, 510–511

globalization, localization, and internationalization (GLI), 36–37

The Grinder, 725

groups

creating when deploying web applications, 152–153
 managing, 496

GRS (global reservation systems), configuring network channels, 557

H

HA (high availability), 630

handler mapping components, for web application, 85–86

handlers, web services, 334–338

associating with service clients, 338
 compared with filters (servlets) and interceptors (EJB), 334–335
 logical handlers, 336–337

protocol handlers, 337–338

handles, to EJBs, 207

happy path, through reservation process, 80

hard-coded navigation, avoiding, 44

hardware acceleration, SSL, 756

hardware requirements, for development environment, 692–693

harvester component, WLDF

harvesting metric information, 625–626
 overview of, 621

hash-based routing, for JMS messages, 416–417

headers, HTTP, 325–326

health monitoring scripts, in WebLogic Server clusters, 734–737

heap size, HotSpot JVM, 654–658

Hibernate

JBoss, 164

JPA derived from, 189

hidden fields

session tracking and, 5
 storing version information in, 187

high availability (HA), 630

HOK (holder-of-key), SAML Token Profile, 507

horizontal partitioning, options for mapping inheritance, 233–234

horizontal scaling

in server deployment, 731
 simple WebLogic Server clusters as example of, 734

hostname verification, in SSL, 486

hot deployment, of applications, 529–530

HotSpot JVM, 654–658

HP-UX, tuning, 649–650

HTML form data, extracting, 39–40

HTTP (HyperText Transfer Protocol)

basic authentication, 350
 headers, 325–326
 redirection, 87–88
 request/response model, 2
 session ID cookie safety, 753–754
 SOAP over HTTP, 303
 tracking HTTP sessions, 342–343
 transport-level security, 348
 writing Java clients that use SSL, 484–485

HTTPS, 348, 753–754

`HttpServletRequest` class, 2–3

`HttpServletRequest` parameter

accessing servlet request data, 5–6

extracting form data, 39–40

servlet capabilities available to JSP pages, 30

`HttpServletResponse` parameter, 9

`HttpSession`, 5–7

changes to session object, 535

replication and, 205–206

session sharing, 127

SFSB component, 205

storing reservation information in, 77–78

storing web service state in, 342

HttpUnit, 722

Hudson, 703

hyperlinks

directory structure and, 126

navigating web applications, 43

HyperText Transfer Protocol. See HTTP (HyperText Transfer Protocol)

I

I18N (internationalization), 36–37

IDEA, IntelliJ, 704

idempotent methods, 207

Idempotent parameter, web server plug-in, 570

identity assertion, in WebLogic security, 457–458

identity provider partners, in SAML, 508–509

IDEs (integrated development environments)

configuring Eclipse for `bigrez.com`, 705–710

IDEs (integrated development environments) *(continued)*

IDEs (integrated development environments) *(continued)*

- configuring JDeveloper for `bigrez.com`, 711–714
- continuous integration and, 703
- debugging with, 714–715
- overview of, 704–705
- prerequisites, 705
- `idle-timeout-seconds` **property, caching and**, 203–204
- include directives, directory structure and**, 126
- in-container testing**, 718
- Incremental collector, garbage collection**, 654
- indexed subscribers, JMS**, 406
- inheritance**
 - JPA, 188
 - Kodo, 220
 - mapping, 233–234
- `init()` **method, servlet class**, 4
- initialization components**, 131–132
- `InitializationListener`, **event listener in** `web.xml`, 131
- in-memory replication**
 - selecting secondary server in clustering for, 534
 - session management and, 667
 - session persistence and, 531
 - SFSB and, 204–207
 - for single cluster and cross cluster replication, 739
 - WAN replication, 556–557
- in-place redeployment**
 - versioning applications and, 628
 - WebLogic Server models for redeploying applications, 530
- instances, WebLogic Server**
 - assigning to machines, 534
 - configuring, 551
 - managing, 541–542
 - overview of, 519
- instrumentation component, WLDF**
 - instrumenting `bigrez.com`, 622
 - overview of, 621
- integration strategy, for external JMS providers**, 440–441
- integrity, WS-Security**, 352
- IntelliJ IDEA**, 704
- interceptors**
 - logging interceptor, 258
 - MDB classes, 162–163
 - web service handlers compared with EJB interceptors, 334
- interfaces, business-tier with presentation-tier**, 67–68
- internationalization (i18n)**, 36–37
- `InverseManager`, 219–220
- inverses (bi-directional relationships). See bi-directional relationships**
- I/O, troubleshooting disk I/O**, 684
- I/O muxer**, 662
- IP addresses, for multicast messages**, 419
- isolation levels, JDBC transactions**, 199–201

J

- J2EE CA (Connector Architecture) resource adapters**, 495
- JAAS (Java Authentication and Authorization Service)**
 - credential mapping, 460
 - overview of, 445
 - writing security aware Java clients, 479–482

- JACC (Java Authorization Contract for Containers)**, 479
- Jakarta Cactus**, 719
- .jar file**
 - bundled libraries, 284–286
 - exploded deployment and, 282
- `jar` **command, creating web application archive file**, 144–145
- Java**
 - comparing Java programming to EJB programming, 167
 - customizing mappings to XML, 346–347
 - Java classes as servlets, 2–3
 - mapping to XML, 306
 - object-oriented design and, 233–234
 - RMI, 159
- Java API for XML Binding. See JAXB (Java API for XML Binding)**
- Java API for XML-based Web Services. See JAX-WS (Java API for XML-based Web Services)**
- Java Authorization Contract for Containers (JACC)**, 479
- Java clients, security-aware**
 - JAAS and, 479–482
 - overview of, 478–479
 - SSL and, 482–487
- Java Cryptography Extension (JCE)**, 756
- Java Data Objects (JDO)**, 164
- Java EE (Enterprise Edition)**
 - application design. *See* application design
 - Application Library Directory, 285
 - application security, 105
 - application tiers, 31–32
 - design patterns, 33
 - EJB 3.0 and, 156
 - increasing performance and throughput, 645
 - locating data sources in, 582–583
 - managing EJB security, 493–495
 - mapping EJBs to JNDI, 193–195
 - ORM products and, 164
 - securing enterprise applications, 496
 - securing web applications, 488–493
 - servlets. *See* Java servlets
 - setting up application security, 488
 - shared libraries and optional packages, 286–287
 - The Spring Framework and, 157
 - web applications, 1
- Java Full Client, JMS**, 381
- Java Home, remote start attributes for servers**, 563
- Java Install Client, JMS**, 381
- Java Management Extensions (JMX)**
 - call router making calls to, 536
 - programmatic monitoring with, 603–606
- Java Naming and Directory Interface. See JNDI (Java Naming and Directory Interface)**
- Java Persistence API. See JPA (Java Persistence API)**
- Java Persistence Query Language. See JPQL (Java Persistence Query Language)**
- Java Platform Debugging Architecture (JPDA)**, 714
- Java SE (Standard Edition)**
 - AOP (Aspect Oriented Programming) and, 334
 - JPA entities in, 171–173
 - persistent class enhancement, 173
 - SSL package, 352
- Java Secure Socket Extension (JSSE)**, 483

JMS (Java Message Service)

Java servlets

- accessing request data, 5
- capabilities available to JSP pages, 9–11
- characteristics of, 1
- converting JSP pages to, 8
- creating Excel files, 26–29
- filtering common behaviors, 22–26
- lifecycle of, 3–4
- multiple parallel request processing, 4–5
- as pure Java classes, 2–3
- reference books for, 1
- registering in applications, 3
- request/response model, 2
- session tracking by, 5–7
- Spring MVC and, 50–53
- Struts, 48–50
- viewing servlet code generated for JSP page, 30

Java stack traces

- overview of, 684
- reading, 687–688

Java Thin Client, JMS, 381

Java thread dumps, 686

Java Transaction API. *See* JTA (Java Transaction API)

Java Vendor, remote start attributes for WebLogic Servers, 563

Java Virtual Machines (JVMs)

- configuring, 567
- data caches for multiple, 218

Java Web Service class. *See* JWS (Java Web Service) class

Java Web Service Compiler. *See* JWSC (Java Web Service Compiler)

javadoc, 704

JavaMail API, 588–589

JavaServer Pages Standard Tag Library. *See* JSTL (JavaServer Pages Standard Tag Library)

JAXB (Java API for XML Binding)

- data POJO class generated by, 320
- mapping Java to XML, 306, 346–347
- overview of, 302
- web services container architecture, 304

JAX-RPC

- support for SOAP style/use combinations, 324–325
- web services container architecture, 304
- working with, 322–323

JAX-WS (Java API for XML-based Web Services)

- BindingProvider class, 350
- classes and annotations in Java EE and Java SE, 309
- creating dynamic web services, 328–332
- creating one-way web services, 344
- developing web service clients, 321–322
- developing web services for WebLogic Server, 304–306
- exception handling, 310
- handlers, 334
- implementing stateful web services, 342
- JAX-RPC incompatible with, 322
- mapping Java to XML, 346
- overview of, 302
- packaging web services, 306
- Provider and Dispatch APIs for dynamic web services, 328–332

- SOAP attachments and, 339

- support for SOAP style/use combinations, 324–325

- web services container architecture, 303–304

- when to use Provider//Dispatch approach, 333–334

JBoss, 164

JCE (Java Cryptography Extension), 756

JDBC

- configuring data sources for use with WebLogic Server, 577–580

- connection pools, 676

- JMS persistent stores and, 389–390

- modules, 287–289

- persistence, 667

- releasing JDBC resources, 676–677

- selecting/configuring drivers for use with WebLogic Server, 576

- session persistence, 738

- SLSBs and, 237–239

- transaction isolation levels, 199–201

- transaction options for data sources, 580–582

JDBC Data Source Configuration Wizard, 577

JDepend, 704

JDeveloper, 711–714

- configuring server instances, 712

- creating first project, 711–712

- creating remaining projects, 712–713

- development tools, 704

- generating web service, 713–714

- installing, 711

- overview of, 711

- running/deploying projects from, 714

JDO (Java Data Objects), 164

JDOM, 333

JMS (Java Message Service)

- balancing producers and consumers, 664

- binding to logical message destination, 268–270

- clients, 380–383

- connection factories, 588

- connections, 365

- destinations, 365–366

- distributed destinations, 586–588

- MDBs and, 161–162, 208

- message producers and consumers, 367–369

- messages, 369–370

- messaging models, 363–364

- migrating JMS services, 633–636

- modules, 287–289, 383–384

- monitoring with WebLogic Console, 603

- overview of, 363

- persistent stores, 584–586

- queue for retrying failed email attempts, 256–257

- resource configuration, 583–584

- review, 441

- reviewing JMS API, 364

- servers, 370–371, 584–586

- sessions, 366–367

- SOAP over, 303

JMS (Java Message Service), configuring

JMS (Java Message Service), configuring

- connection factories, 385–386
- delivery overrides, 392–393
- destination keys, 387
- flow control, 396–398
- message paging, 395–396
- modules, 383–384
- persistent stores, 388–392
- quotas, 393–395
- templates, 387
- time-to-deliver extension, 388

JMS (Java Message Service), external providers

- choosing integration strategy, 440–441
- integrating AQ (advanced queuing), 438–440
- mapping external JMS objects to WebLogic JNDI, 437–438

- MDBs (message-driven beans), 436
- messaging bridge and, 433
- overview of, 432–433
- store-and-forward service, 434–436

JMS applications, designing

- choosing destination type, 398
- choosing message expiration strategy, 407–409
- choosing message type, 400–402
- compressing large messages, 402
- designing message selectors, 404–407
- handling message ordering, 412–413
- handling poison messages, 409–411
- handling request/reply style message exchange, 420–424
- locating destinations, 399–400
- multicast session, 419–420
- overview of, 398
- Path Service and hash-based routing, 416–417
- Redelivery Delay, Redelivery Limit, and Error Destination, 411–412
- selecting message acknowledgement strategy, 403–404
- transactions, 417–419
- unit-of-order messages, 413–415
- unit-of-work messages, 415–416

JMS applications, programming

- consuming asynchronous messages on server, 427
- designing MDB-based applications, 431–432
- MDBs (message-driven beans), 427–431
- overview of, 424
- servlets and EJBs, 424–427

JMS clients, 380–383

- overview of, 380–381
- types of, 381
- WebLogic Server client URLs, 382–383

JMS clustering

- connection routing, load balancing and failover, 371–372
- consuming messages from a distributed queue, 375–377
- consuming messages from a distributed topic, 378–380
- distributed destinations, 373–374
- location transparency, 371
- overview of, 371
- producing messages to a distributed queue, 374–375
- producing messages to a distributed topic, 377–378

JMS servers

- automatic failover, 380

- consuming asynchronous messages on server, 427
- overview of, 370–371

JMX (Java Management Extensions)

- call router making calls to, 536
- programmatic monitoring with, 603–606

JNDI (Java Naming and Directory Interface)

- bindings and references to EJBs and other resources, 267–268
- deployment plans and, 222–223
- EJBs referencing each other by global JNDI names, 196
- JMS applications looking up destinations from, 365
- JndiObjectFactoryBean utility, 133
- lookup strategies, 670–671
- mapping external JMS objects to WebLogic JNDI, 437–438
- remote business interfaces and, 193–195

joined tables, inheritance strategies, 188

JPA (Java Persistence API)

- architecture requirements analysis, 245
- cascading operations, 176–177
- concepts, 165–166
- eager and lazy fetching, 180–181
- embedded entity classes, 180
- entities in Java SE environment, 171–173
- explicit locking, 186
- generated values, 170–171
- generating JPA classes from database schema, 249–250
- generic functionality, 250–252
- history of, 163–164
- importing project into Eclipse and, 707
- inheritance, 188
- JPQL bulk updates, 179–180
- JPQL queries, 177–179
- OpenJPA and Kodo components, 165
- optimistic locking and version fields, 181–185
- overview of, 163
- packaging JPA persistent units, 273–274
- persistence management, 67
- persistent class enhancement, 173
- relationship mapping and management, 174–176
- review, 189
- sample, 166–170
- SLSBs and, 242–244
- writing web layer suited to optimistic locking, 187–188

JPDA (Java Platform Debugging Architecture), 714

JPQL (Java Persistence Query Language)

- bulk updates, 179–180
- controlling flush behavior, 219
- large result sets and, 216
- queries, 177–179

JRA (JRockit Runtime Analyzer), 661

JRockit

- monitoring server performance, 602
- using, 658–661

JRockit Latency Analyzer, 661

JRockit Leak Detector, 661

JRockit Management Console, 661

JRockit Mission Control, 661

JSP (JavaServer Pages)

- buffering JSP responses, 11–12
- controllers populating forms for, 69–70
- converting JSP pages to servlets, 8

literal, SOAP uses

creating Excel files, 26–29
 display requirements and, 34
 display tasks in servlet-centric architecture, 49
 display tasks in Spring MVC, 52
 implicit objects, 18
 JSP-centric architecture, 46–48
 overview of, 7
 precompiling JSP components, 137–139
 precompiling pages in production system, 754–755
 reference books for, 1
 as scripting technology, 7
 servlet capabilities available to, 9–11
 tags and scripting elements available with, 8–9
 unique capabilities not present in servlets, 12
 viewing servlet code generated for JSP page, 30
 web container best practices, 669
JspBase class, 8
JSPClassServlet, 138–139
jsp-descriptor section, of weblogic.xml, 136
.jspx file extension, 139
JSPServlet, 138–139
JSSE (Java Secure Socket Extension), 483
JSTL (JavaServer Pages Standard Tag Library)
 accessing data stored in JSP implicit objects, 10
 Expression language (EL) used in JSTL calls, 16–18
 tag libraries, 19
 using JSTL tags to reduce scriptlet code, 15
JTA (Java Transaction API)
 database transaction semantics, 575
 global transactions, 580–581
 JTA-aware transactions in JMS, 418
 local and distributed transactions supported by WebLogic Server, 674
 migrating JTA service, 636–637
JUnit
 overview of, 717–718
 packaging *bigrez.com*, 291
 testing *bigrez.com*, 258–259
JVMs (Java Virtual Machines)
 configuring, 567
 data caches for multiple, 218
 JVM-to-processor ratio, 730
JVMs (Java Virtual Machines), tuning
 garbage collection, 652–654
 overview of, 652
 tuning HotSpot JVM heap size, 654–658
 using JRockit JVM, 658–661
JVM-to-processor ratio, in server deployment, 730
JWebUnit, 722–723
JWS (Java Web Service) class
 developing code-first web services, 307
 developing web services for WebLogic Server, 304–305
 developing WSDL-first web services, 316–317
JWSC (Java Web Service Compiler)
 deciding which web service operation
 to invoke, 325–326
 developing code-first web services, 307–312
 developing WSDL-first web services, 317
jwsc task, Ant build tool, 311
Jython scripting language, 596

K

KDC (Kerberos Distribution Center), 511–513
key stores
 configuring one-way SSL, 472–473
 configuring two-way SSL, 474–477
keytool, 471–472
Kodo
 converting to TopLink. *See* TopLink, converting from Kodo to
 deployment descriptors in, 209–210
 eager fetching in, 212
 features not in open source version, 165
 features of EJB container, 209
 JPA and, 164, 189
 lock groups in, 214–215
 mixing inheritance strategies, 220
 persistent class enhancement, 173
 runtime enhancer, 172
 second level data caches in, 217–219
 vs. TopLink, 260
kodoc tool, for persistent class enhancement, 173

L

L10N (localization), 36–37
L2 (second level) data caches, in Kodo. *See* data caches
large result sets
 data access requirements, 234
 proxies in OpenJPA, 216
 queries in OpenJPA, 215–216
Last Logging Resource, transaction commit protocols,
 581–582
layout, display flexibility and maintainability, 37
lazy fetching, JPA, 180–181
lazy instantiation, 232
lazy loading, OpenJPA and Kodo, 173
LDAP (lightweight directory access protocol)
 embedded LDAP server, 450–451
 managing external LDAP authentication, 465–467
leases, service migration and, 633–634
lib classloader, 276
libraries
 bundled, 277, 284–286
 custom tags, 22
 Java EE Application Library Directory, 285
 JavaServer Pages Standard Tag Library. *See* JSTL (JavaServer
 Pages Standard Tag Library)
 shared, 286–287, 292–293
libraries directory, 277
lifecycle, EJB, 190–192
LIFO (last-in-last-out), 387
**lightweight directory access protocol. *See* LDAP (lightweight
 directory access protocol)**
Linux, tuning, 650
listen ports
 deployment architecture, WebLogic Server, 529
 network channels for defining, 545–547
listen threads, 529, 545–547, 688
lists, displaying list of model objects, 35–36
literal, SOAP uses, 324

load balancing

load balancing

- identifying and correcting bottlenecks, 679
- improving SSL performance, 756
- in JMS clustering, 371–372
- Multi Data Source support and, 631–632
- overview of, 745–746
- round-robin algorithm, 536
- routing mechanisms and, 532–533
- simple WebLogic Server clusters and, 733
- using global load balancers with WebLogic Server, 748–749
- using local load balancers with WebLogic Server, 747–748
- WebLogic Server clustering, 530

Load Balancing Enabled, JMS, 386

load testing, 725

local load balancers

- overview of, 746
- using with WebLogic Server, 747–748

localization (L10N), 36–37

location transparency

- distributed applications and, 156
- EJB and, 671
- in JMS clustering, 371

lock(), in EntityManager, 186

lock groups, optimistic locking strategy, 214–215

locking down web applications, 753–754

Log, JMS message expiration, 408

logging

- activity on administrative site, 130
- Apache Commons Logging API, 620–621
- business layer requirements, 235
- configuring, 136
- configuring for WebLogic Server, 550–551
- interceptor, 258
- message catalog logging, 618–619
- non-catalog logging, 620
- overview of, 617–618
- using WebLogic Server log file, 756

logical handlers, JAX-WS, 334, 336–337

Login Timeout, 552

login-config element, 106

Login.jsp page, 107–108

LRU (least recently used) algorithm, in caching, 203, 217, 579

M

machines

- defining, 534
- in WebLogic architecture, 519

maintainability, of display, 37–38

maintenance pages, availability and, 117–122

MAN (metropolitan-area network) replication

- configuring, 556
- implementing one cluster per site, 744
- overview of, 535

managed server independence (MSI), 540

managed servers

- overview of, 520
- setting up new domain and, 549–551

man-in-the-middle attacks, 755

many-to-many associations, 231–233

many-to-one associations, 231–233

MapMessage object, JMS, 370, 401

mapped superclasses, JPA, 188

mapping

- EJBs to JNDI, 193–195
- external JMS objects to WebLogic JNDI, 437–438
- Java to XML, 306, 346–347
- reverse mapping tool for generating JPA classes, 249–250

mapping, ORM

- associations, 231–233
- inheritance and, 233–234
- simple classes, 230–231

Mark-and-compact collector, garbage collection, 654

Mark-and-Sweep collector, garbage collection, 653

master page assembly approach, 64–66

max-beans-in-cache, 202–203

Maximum Messages per Session, JMS connection factories, 385

MDBs (message-driven beans), 427–431

- components, 161–162
- concurrency and, 427–428
- connecting to distributed destinations, 430–431
- dealing with durable subscriptions, 429–430
- EJB deployment descriptors, 272
- email management with, 257
- error handling and, 258
- features of EJB container, 208–209
- interceptor classes, 162–163
- overview of, 427
- transactions, 428–429
- tuning and optimization settings, 273
- using with WebLogic Server, 436
- merge(), of EntityManager, 181, 187

merge(), of EntityManager, 181, 187

message acknowledgement, JMS, 366

message catalog logging, WebLogic Server, 618–619

message forwarding agents, 432

message selectors, JMS, 369

Message Transmission Optimization Mechanism (MTOM), 341

message-driven beans. *See* MDBs (message-driven beans)

message-level security

- overview of, 348–349
- web services, 352–358

message-passing model, 333

messages, JMS. *See also* JMS (Java Message Service)

- overview of, 369–370
- types of message objects, 370

messaging bridge

- as message forwarding technology, 432
- overview of, 433

messaging intermediaries, Provider//Dispatch approach, 333

messaging models, JMS, 363–364

metadata, providing deployment metadata, 221

META-INF directory, 277–278

metrics, harvesting metric information with WLDF, 625–626

Microsoft Excel files, 26–29

migration, 338

- admin server, 640–641
- custom singleton services, 637–638
- JMS services, 633–636

OEPE (Oracle Enterprise Pack for Eclipse)

- JTA service, 636–637
 - service level migration, 633
 - whole server migration, 638–640
- mocking frameworks, for unit tests**, 719–720
- Mockito**
 - library, 259
 - as mocking framework for unit tests, 719–720
- model data, displaying**, 35
- model objects**, 35–36
- models**. *See also* **MVC (model-view-controller)**
 - application security, 487
 - displaying list of model objects, 35–36
 - displaying model data, 35
 - domain model. *See* domain model
 - JMS messaging models, 363–364, 392
 - message-passing model, 333
 - redployment model, 530
 - request/response model. *See* request/response model
 - security model, 36
 - updating model objects, 41–42
- model-view-controller**. *See* **MVC (model-view-controller)**
- modular approach, to project structure**, 697–698
- modularity, display flexibility and maintainability**, 37
- modules**
 - defined, 265
 - descriptor files, 266
 - EJB, 265–266
 - exploded deployment and, 282–284
 - identifying when not descriptor provided, 279
 - JMS, 383–384
 - other types in enterprise applications, 287–289
 - security settings, 270
 - stand-alone, 274
- monitor locks (mutex)**, 685
- monitoring**
 - self-health monitoring, 544–545
 - SNMP models for, 607
 - WebLogic Server applications. *See* application monitoring, WebLogic Server
- MSI (managed server independence)**, 540
- MTOM (Message Transmission Optimization Mechanism)**, 341
- Multi Data Source support, in WebLogic Server**, 631–632
- multicast messages, UDP**, 741
- multicast sessions, in JMS**, 419–420
- `MULTICAST_NO_ACKNOWLEDGE`, 403–404
- multicast-based clustering, UDP**, 553, 555
- multiple site deployment strategies**
 - designing multiple-site WebLogic clusters, 738–741
 - implementing clusters that span multiple sites, 741–742
 - implementing one cluster per site, 743–745
 - overview of, 737–738
- mutex (monitor locks)**, 685
- muxers (socket multiplexing)**
 - managing server sockets, 527–529
 - using native I/O muxer, 662
- MVC (model-view-controller)**
 - controllers for calling business services, 68–69
 - controllers for populating forms for, 69–70
 - Java EE components mapped to, 32–33
 - servlet-centric (Spring MVC) architecture and, 50–53
- N**
 - named queries, JPQL**, 178
 - naming conventions, for components**, 79
 - NAT (network address translation)**, 546, 750
 - native I/O muxer**, 662
 - navigation**
 - controlling with configuration files, 49
 - defining navigation paths for web application, 79–81
 - navigation requirements, presentation-tier**
 - basic navigation, 42
 - outcome-based navigation, 43–44
 - overview of, 42
 - submission/bookmark controls, 44–45
 - .NET Client**, 381
 - NetBeans, Sun**, 704
 - network address translation (NAT)**, 546, 750
 - network channels**
 - configuring, 557–560
 - for defining listen ports and listen threads, 545–547
 - network latency, clustering and**, 741
 - networks, tuning**, 651–652
 - NIO (non-blocking I/O), in JDK 1.4**, 528
 - `NO_ACKNOWLEDGE`, 403
 - no-args constructor**, 306, 320
 - node manager**, 560–566
 - authentication required by, 562
 - configuring in WebLogic Console, 562–563
 - installing and configuring on Unix OSs, 561
 - managing server instances with, 541–542
 - overview of, 520, 560–561
 - remote start attributes for servers, 563–564
 - scripts for starting, 564–566
 - X.509 certificates, 565
 - non-blocking I/O (NIO), in JDK 1.4**, 528
 - non-catalog logging, WebLogic Server**, 620
 - nonce, passwords and**, 352
 - nondurable subscriptions, JMS messages**, 364
 - non-persistent messages, JMS messages**, 363
 - notifications, WLDF**, 621, 627–628
 - NRU (not recently used) algorithm, caching and**, 203
- O**
 - object graphs, ORM and**, 232
 - Object Relational Mapping**. *See* **ORM (Object Relational Mapping)**
 - ObjectMessage object, JMS**, 370, 401
 - object-oriented design**
 - inheritance and, 188
 - mapping inheritance and, 233–234
 - OCSP (Online Certificate Status Protocol)**, 476
 - OEPE (Oracle Enterprise Pack for Eclipse)**
 - application deployment plan, 593
 - importing `bigrez.com`, 706–707
 - overview of, 705–706
 - publishing (deploying to server), 708
 - running `bigrez.com` from, 707–708
 - target runtime in, 706
 - Virtual EAR feature, 698, 709–710

Offers.jsp page

Offers.jsp **page**, 104–105

One-Phase Commit, **transaction commit protocols**, 581

one-to-many associations, 231–233

one-to-one associations, 231

one-way SSL

- configuring, 472–473
- overview of, 469

one-way web services, 344

Online Certificate Status Protocol (OCSP), 476

OpenJPA

- controlling flush behavior in, 219
- features beyond those required by JPA, 165
- features of EJB container, 209
- fetch groups, 210–212
- Kodo source code donated to, 164
- managing inverses (bi-directional relationships) in, 219–220
- persistent class enhancement, 173
- prepared statement caching in, 220–221
- reverse mapping tool for generating JPA classes, 249–250
- state comparison version strategy, 213–214, 259
- support for version field types in, 212–213

operating systems (OSs), configuring for use with WebLogic Server, 566–567

operations, web services, 325–327

optimistic locking

- adding to bigrez.com, 259–260
- bulk updates and, 180
- Kodo strategies, 214–215
- locking options and, 182–186
- OpenJPA strategies, 212–214
- writing web layer suited to, 187–188

optimization

- deployment descriptor settings, 272–273
- when to use Provider//Dispatch approach, 333–334

Oracle

- AQ (advanced queuing), 438–440
- Coherence, 217–219
- databases, 199–201
- JDeveloper. *See* JDeveloper
- JRockit, 602, 658–661
- TopLink and Kodo, 164

Oracle Enterprise Pack for Eclipse. *See* OEPE (Oracle Enterprise Pack for Eclipse)

ORM (Object Relational Mapping)

- business layer requirements, 230
- JPA and, 163
- mapping and managing relationships, 174–176
- mapping associations, 231–233
- mapping inheritance, 233–234
- mapping simple classes, 230–231

orm.xml **file**, 172

OSs (operating systems), 645–646

- configuring for use with WebLogic Server, 566–567
- tuning, 645–646
- tuning HP-UX, 649–650
- tuning JVMs. *See* JVMs (Java Virtual Machines), tuning
- tuning Linux, 650
- tuning network, 651–652
- tuning Solaris, 646–649
- tuning Windows, 650
- using file system security mechanisms, 755

outcome-based navigation, 43–44

out-of-container testing, 718–719

P

packaging bigrez.com

- altering bigrez.com to use shared library, 292–293
- compiling production builds with appc, 293–294
- components, 291
- projects, 291–292

packaging enterprise applications

- bundled libraries, 284–286
- customizing class loading, 289–291
- descriptor files, 278
- directory structure, 277
- module types and, 287–289
- optional packages, 286–287
- overview of, 274–276
- review, 298–299
- shared Java EE libraries, 286–287
- standard application.xml descriptor file, 278–280
- weblogic-application.xml descriptor file, 280–282

packaging JPA persistent units, 273–274

packaging web applications

- creating archive file, 144–145
- creating exploded file, 139–144
- directory structure and, 126–128
- internal components in WEB-INF directory, 128
- overview of, 125–126
- precompiling JSP components, 137–139
- review, 153
- weblogic.xml descriptor file, 133–137
- web.xml descriptor file, 128–133

pagging, JMS messages, 395–396

parallel collectors, garbage collection, 654, 659

ParameterizableViewController **class**, 82, 84

partial pages, precompiling JSP components, 139

partial redeployment, 283, 530

pass-by semantics

- calling components by reference, 198
- EJB container best practices, 671–672

pass-by-reference, 671–672

pass-by-value, 671

passive cookie persistence, 748

password digests

- message-level security, 352
- using in SOAP messages, 358

passwords

- booting WebLogic Server, 500–501
- exporting, 463
- remote start attributes for WebLogic Servers, 563
- SNMP community names, 608–609
- user-password lockouts, 755

Path Service, for JMS messages, 416–417

PathPrepend **parameter, web server plug-ins**, 570

PathTrim **parameter, web server plug-ins**, 570

PEM (Privacy-Enhanced Mail), for certificates, 471–472

performance

- business layer requirements, 235
- unit tests for, 724–725

producers, JMS

- when to use `Provider//Dispatch` approach, 333
- performance best practices**
 - command pattern, 667
 - designing for performance, 665–666
 - overview of, 665
 - session façade, 666
 - value objects, 666
- performance optimization, WebLogic Server**
 - database access best practices. *See* database access best practices
 - EJB container best practices. *See* EJB container best practices
 - performance best practices. *See* performance best practices
 - system performance. *See* system performance
 - troubleshooting. *See* troubleshooting performance problems
 - tuning application servers. *See* application servers, tuning
 - tuning JVMs. *See* JVMs (Java Virtual Machines), tuning
 - web container best practices. *See* web container best practices
- persistence context, in `EntityManager`**, 166, 169
- persistence provider, converting to `TopLink` and**, 261
- persistence unit, descriptors defining**, 168–169
- `persistence-configuration.xml` file**, 209–210
- `persistence.xml` file**
 - Kodo deployment descriptor, 209–210
 - packaging JPA persistent units, 273–274
 - specifying EclipseLink JPA provider, 261
- persistent class enhancement, JPA**, 173
- persistent messages, JMS**, 363, 392
- persistent stores, JMS**
 - configuring, 389–391
 - creating, 584–586
 - destinations and, 374
 - overview of, 388–392
- persistent units, JPA**, 273–274
- Person entity**
 - `id` field, 169–170
 - JPA entity representing, 166–167
 - mapping and managing relationships to, 174–176
 - optimistic locking and, 183
 - SLSB creating, 168
- pessimistic locking**, 182
- PING command**, 600, 651
- PKI**
 - PKI Credential Mapper, 357–358
 - SAML and, 502
- Plain Old Java Objects. *See* POJOs (Plain Old Java Objects)**
- POC (Proof of Concept), testing**, 729
- pointcut expression**, 623
- point-to-point messaging model, JMS**, 363–364
- poison messages**
 - handling in JMS, 409–411
 - handling in MDBs, 258
- POJOs (Plain Old Java Objects)**
 - data POJOs, 306
 - data POJOs compared to JavaBean classes, 310
 - developing code-first web services, 307
 - EJBs as, 156–157
- policies**
 - defining web service security policies, 349–350
- remote start attributes for WebLogic Servers, 563
- service-level migration, 634
- working with, 497–500
- pools**
 - free pool settings, 271
 - MDBs, 208
 - SLSBs, 201–202
- ports**
 - configuring admin port, 572–575
 - listen ports, 529
- ports (innovation points), web services**, 320
- precompiling JSP components**, 137–139
- Preemptive values, JMS message quotas and**, 395
- prepared statement caching**
 - database access best practices, 673–674
 - in OpenJPA, 220–221
- presentation approaches, for `bizgrez.com`**
 - composite view pattern, 60–61
 - external view assembly using Tiles framework, 66–67
 - master page assembly, 64–66
 - self-assembly, 61–64
- presentation requirements, for `bizgrez.com`**, 57–59
- presentation-tier**
 - basic navigation, 42
 - building, 45–46
 - client-side validation, 38–39
 - controllers for calling business services, 68–69
 - controllers for populating forms for JSPs, 69–70
 - display requirements, 34
 - displaying errors, 40–41
 - displaying list of model objects, 35–36
 - displaying model data, 35
 - extracting form data, 39–40
 - flexibility and maintainability of display, 37–38
 - form/update requirements, 38
 - interacting with business tier for updating model objects, 41–42
 - internationalization and, 36–37
 - Java EE application tiers, 32
 - JSP-centric architecture, 46–48
 - navigation requirements, 42
 - outcome-based navigation, 43–44
 - overview of, 33, 67–68
 - presenting role-based views of data, 36
 - requirements, 34
 - server-side validation, 40
 - servlet-centric (Spring MVC) architecture, 50–53
 - servlet-centric (Struts) architecture, 48–50
 - submission/bookmark controls, 44–45
 - updates requiring explicit service calls, 71
- primary keys**
 - entities and, 170
 - searching by, 177
- principle of least exposure, encapsulation and**, 228
- Privacy-Enhanced Mail (PEM), for certificates**, 471–472
- problem resolution process**, 680–682
- producers, JMS**
 - balancing with consumers, 664
 - producing messages generally, 363, 367–369

producers, JMS (*continued*)

producers, JMS (*continued*)

- producing messages to a distributed queue, 374–375
- producing messages to a distributed topic, 377–378

production environment

- cluster design options, 738
- complex cluster for single-site deployment, 734–737
- connection filters, 751–753
- data replication requirements, 740–741
- deployment strategies, 727–728
- designing multiple-site WebLogic clusters, 738
- evaluating deployment strategies, 728–729
- examining other security options, 754–756
- firewall layout in, 750–751
- global and local traffic management, 745
- implementing clusters that span multiple sites, 741–742
- implementing one cluster per site, 743–744
- load balancing, 745–746
- locking down web applications, 753–754
- MAN replication, 744
- multiple site deployment strategies, 737–738
- overview of, 727
- review, 757
- security strategies, 749
- separating from development environment, 754
- server deployment strategies, 729–731
- session replication, 738–740
- simple cluster for single-site deployment, 732–734
- single-site deployment strategies, 731–732
- SSL hardware acceleration, 756
- transaction collocation requirements in multiple site deployment strategies, 740
- understanding application data flow, 749
- using global load balancers with WebLogic Server, 748–749
- using load balancers with WebLogic Server, 747–748
- WAN replication, 744–745

production redeployment

- versioning applications and, 628–630
- WebLogic Server models for redeploying applications, 530

projects

- bigrez.com, 291–292
- creating project structure in development environment, 697–698

Proof of Concept (POC), testing, 729

properties

- availability maintenance pages, 117–122
- main form, 109–114
- rate maintenance pages, 114–117
- search pages, 88–90
- updating and maintaining, 108–109

PropertyAvailabilityController, 119–122

PropertyAvail.jsp, 117–122

PropertyList.jsp, 89–91, 108–109

PropertyMainController, 109–113

PropertyMain.jsp, 109–114

PropertyRatesController, 114–117

PropertyRates.jsp **page**, 114–117

PropertySearchController, 88–90

PropertySearch.jsp, 88–89

protocol handlers

- JAX-WS, 334, 337–338
- SOAP, 339–340

Provider APIs (JAX-WS)

- creating dynamic web services, 328–332
- when to use Provider/Dispatch approach, 333–334

provider management level, WebLogic Server security administration, 449

providers, JMS

- clusters. *See* JMS clustering
- external. *See* JMS (Java Message Service), external providers overview of, 370
- Web servers, 370–371

providers, WebLogic security. *See* security providers

proxies

- large result set proxies, 216
- web service clients using stub Java class to invoke web services, 318

public/private keys, in message-level security, 357

publish-and-subscribe messaging model, JMS, 363–364, 392

publishing, deployment in Eclipse, 708

Q

QL (query language), 177

queries

- JPQL, 177–179
- large result sets and, 215–216

query language (QL), 177

queues, JMS

- AQ (advanced queuing), 438–440
- choosing destination type and, 398
- consuming messages from a distributed queue, 375–376
- overview of, 363
- producing messages to a distributed queue, 374
- zero consumer queues, 375

quotas, JMS messages, 393–395

R

.rar file, 282

rate maintenance pages, web applications, 114–117

RDBMS

- authentication, 467–469
- security store, 451–452

read only concurrency strategy, 183

READ_COMMITTED, JDBC isolation levels, 200–201

realms

- administering security at realm level, 448–449
- security models and, 455

receive windows, TCP, 648

receivers, JMS messages. *See* consumers, JMS

Redelivery Delay, handling poison messages in JMS, 411–412

Redelivery Limit, handling poison messages in JMS, 411–412

REDEPLOY file, 147–148

Redirect, JMS message expiration, 408

redirection, HTTP, 87–88

references between EJBs

- calling components by reference, 198–199
- to external EJB, 198
- in same application, 195–197

- relational database, for storing domain objects**, 56
- relationships**
 - mapping and managing in JPA, 174–176
 - mapping associations, 231–233
 - in presentation components, 71–72
- reliability, clustering and**, 741
- remote business interfaces**
 - handles to EJBs, 207
 - JNDI and, 193–195
 - tuning and optimization settings, 273
- remote procedure calls (RPCs)**
 - JAX-RPC, 304, 322–323
 - SOAP and, 301
- remote services, EJBs providing**, 159
- replica-aware stubs**, 536
- replication**
 - cross-cluster replication, 555–557
 - data replication requirements in multiple site deployment, 740–741
 - in-memory replication, 204–207, 531
 - MAN and WAN replication, 535
 - session replication in production environment, 738–740
 - specifying replication groups, 534
- reporting tools, for development environment**, 704
- request classes**
 - configuring, 589–590
 - defining, 525–526
 - WebLogic Server, 523, 525
- request/reply style, for exchanging JMS messages**, 420–424
- request/response model, HTTP**
 - accessing request data, 5
 - buffering JSP responses, 11–12
 - Java servlets, 2
 - JSP pages and, 11
 - multiple parallel request processing, 4–5
 - response caching, 24–26
- request/response model, SFSB**, 159
- requests, prioritizing in thread management**, 664
- reservation creation process, in building web application**, 101–103
- reservation information component, building web applications**, 76–79
- reservation process**
 - availability display and room-type selection page, 94–101
 - bigrez.com, 58
 - building web applications, 79
 - choosing basic controller components, 82–84
 - choosing form-processing controller components, 84–85
 - choosing handler and resolver components, 85–88
 - choosing Spring MVC components, 81
 - date selection page, 91–94
 - defining navigation paths, 79–81
 - property search/selection pages, 88–91
 - reservation creation process, 101–103
 - ReservationInfo **object**, 77–78
 - resolver, view**, 86–88
 - resource contention, troubleshooting**, 616
 - resource pools**, 664–665
 - <resource-description> **element**, 267–268
 - resources**
 - bigrez.com requiring in environment, 296
 - bindings and references to EJB and other resources, 267–268
 - control over resource loading, 290–291
 - ejb-local-ref elements and, 136
 - securing, 496–497
 - response times**
 - response time request class, 523
 - troubleshooting, 683–684
 - responses. See request/response model, HTTP**
 - result sets, large. See large result sets**
 - reverse mapping, for generating JPA classes**, 249–250
 - ReviewReservationController, 101–102
 - ReviewReservation.jsp **page**, 101–103
 - RMI**
 - clustering for RMI-based applications, 536–538
 - EJBs providing remote services via Java RMI, 159
 - remote business interfaces and JNDI and, 193–195
 - tuning and optimization settings, 273
 - writing Java clients that use SSL, 482
 - role mapping**
 - EJBs supporting, 494
 - security providers, 458–459
 - role-based views of data**, 36
 - roles, working with**, 497–500
 - room-type selection page, in reservation process**, 94–101
 - Root directory, remote start attributes for WebLogic Servers**, 563
 - round-robin algorithm, for load balancing**, 536
 - round-robin DNS policies**, 749
 - routing, session-based routing in WebLogic**, 532–534
 - rpc, **SOAP styles**, 323–324
 - RPCs (remote procedure calls)**
 - JAX-RPC, 304, 322–323
 - SOAP and, 301
 - runnable state, threads**, 685
 - Runtime MBeans, WLST**, 598

S

 - SAF (store-and-forward)**
 - agent configuration, 435
 - MDBs and, 209
 - as message forwarding technology, 432
 - overview of, 434–436
 - SAML (Security Assertion Markup Language)**
 - configuring federation services, 504–506
 - controlling SAML partner exports, 463
 - debugging, 510
 - identity provider partners, 508–509
 - managing SSO partners, 506–507
 - managing web services partners, 507–508
 - overview of, 501–502
 - security providers, 502–504
 - service provider partners, 509
 - WebLogic Server support, 302
 - SAML Token Profile**, 507
 - scalability**
 - business layer requirements, 235
 - techniques, 644

SCM (source code management)

SCM (source code management), 692

scripts. See also JSP (JavaServer Pages)

for complex WebLogic Server clusters, 736

JSP scripting elements, 8–9

scripting technology, 7

using JSTL tags to reduce scriptlet code, 15

search pages, property search, 88–90

second level (L2) data caches, in Kodo. See data caches

Secure Sockets Layer. See SSL (Secure Sockets Layer)

security

admin and, 152

authentication/authorization components, 105–108

business logic requirements and, 229

EJB deployment descriptors, 270

presenting role-based views of data, 36

threat models, 749

security, web service

configuring, 358–359

defining security policies, 349–350

message-level security, 352–358

overview of, 348–349

transport-level security, 350–352

security, WebLogic

administration of, 447–448

application security models, 487

auditing, 459–460

authentication, 455–457

authorization, role mapping, and adjudication, 458–459

booting WebLogic Server and, 500–501

certification path, 460–461

configuring one-way SSL, 472–473

configuring two-way SSL, 474–477

credential mapping, 460

cross domain security, 510–511

custom authentication providers, 516

debugging, 464

debugging SSL problems, 477–478

domain level, 447–448

embedded LDAP server for, 450–451

external security stores, 465

framework for, 449

identity assertion, 457–458

managing EJB security, 493–495

managing external LDAP authentication, 465–467

managing RDBMS authentication, 467–469

managing security providers, 461–464

managing users and groups, 496

obtaining X.509 certificates, 470–472

overview of, 443–446

provider management level, 449

RDBMS security store, 451–452

realm level, 448–449

realms and providers, 453–455

resources and, 496–497

review, 517

SAML. *See* SAML (Security Assertion Markup Language)

securing enterprise applications, 496

securing J2EE CA resource adapters, 495

securing web applications, 488–493

server level, 447–448

servlet authentication filters, 517

SSL and X.509 certificates, 469–470

SSO (single sign-on), 501, 511–516

working with roles and policies, 497–500

writing security aware Java clients, 478–479

writing security-aware Java clients that use JAAS, 479–482

writing security-aware Java clients that use SSL, 482–487

Security Assertion Markup Language. See SAML (Security Assertion Markup Language)

security models, 454

security policies

defining web service security policies, 349–350

remote start attributes for WebLogic Servers, 563

security providers

administering security at provider management level, 449

auditing, 459–460

authentication providers, 455–457

authorization, role mapping, and adjudication, 458–459

credential mapping, 460

custom authentication providers, 516

managing, 461–464

SAML, 502–504

types and features, 453–455

Security Realms, 152–153

security stores

external, 465

RDBMS, 451–452

security strategies, in production environment

application data flow and, 749

connection filters, 751–753

examining other security options, 754–756

firewall layout, 750–751

locking down web applications, 753–754

overview of, 749

SSL hardware acceleration, 756

security-aware Java clients. See Java clients, security-aware

SEI approach, to implementing web services, 327

SelectDatesController, 92

SelectDates.jsp, 91–94

selection pages, 90–91

selectors, designing JMS message selectors, 404–407

SelectRoomTypeController, 96–97

SelectRoomType.jsp, 94–101

self-assembly approach, 61–64

self-health monitoring, 544–545

send windows, TCP, 648

sender -voucher, SAML Token Profile, 507

senders, JMS messages. See producers, JMS

separate directories, vs. separate web applications, 127

SERIALIZABLE, JDBC isolation levels, 200

server affinity, JMS, 375

JMS connection factories, 386

server architecture, WebLogic Server, 521–530

application container, 529–530

execute queues, execute threads, and work managers,

521–527

listen ports and listen threads, 529

overview of, 521

socket muxer, 527–529

server deadlocks, 663

server deployment, 729–731

horizontal scaling, 731

SNMP (Simple Network Management Protocol)

- JVM-to-processor ratio, 730
- overview of, 729
- vertical scaling, 730–731
- server overload**, 663–664
- servers**
 - administering security at server level, 447–448
 - configuring WebLogic Server instances, 551–553
 - creating JMS servers, 584–586
 - managing server failure, 632
 - migrating JMS servers, 584–585
 - monitoring performance of, 601–603
 - overview of JMS servers, 370–371
 - remote start attributes for, 563
 - selecting secondary server in clusters, 534–535
 - server states in WebLogic Server, 542–543
 - troubleshooting throughput, 682–683
 - tuning application servers, 661–665
 - in WebLogic architecture, 519
- server-side forward**, 753
- server-side validation**, 40
- service components, EJB**, 74
- service consumers, RPCs and**, 301
- Service Level Agreements (SLAs)**, 632
- `service()` **method, of servlet class**, 4
- service provider partners, in SAML**, 509
- service providers, RPCs**, 301
- service-level migration**
 - migrating custom singleton services, 637–638
 - migrating JMS servers, 584–585
 - migrating JMS services, 633–636
 - migrating JTA service, 636–637
- services, creating required services for enterprise applications**, 295–296
- Servlet Context Event listener classes**, 131–132
- servlets. *See also* Java servlets**
 - application programming in JMS, 424–427
 - authentication filters, 517
 - configuring for work managers, 527
 - web container best practices, 669–670
- session beans**
 - cache management, 202–204
 - handles to EJBs, 207
 - idempotent methods, 207
 - in-memory replication, 204–207
 - pooling, 201–202
 - remote business interfaces and JNDI, 193–195
 - security settings, 270
 - stateful. *See* SFSB (stateful session bean)
 - stateless. *See* SLSB (stateless session bean)
- session façade**
 - business service in `bigrez.com`, 252–253
 - design pattern, 33
 - DTOs and, 229–230
 - implementing with session EJBs, 228–229
 - performance best practices and, 666
 - SLSB (stateless session beans) and, 159
 - transaction management and security and, 229
- session IDs, cookie safety**, 753–754
- session persistence**
 - changes to `HttpSession` object and, 535
 - in-memory replication, 531
- JDBC, 738
- session timeouts**, 668
- session tracking**
 - Java servlets, 5–7
 - JSP, 10–11
- session-based routing, WebLogic Server clusters**, 532–534
- session-descriptor **section, of** `weblogic.xml`, 136
- sessions**
 - JSP, 669
 - managing, 667–668
 - replication, 738–740
- sessions, HTTP**
 - storing view entities in, 187
 - tracking, 342–343
- sessions, JMS**
 - multicast sessions, 419–420
 - overview of, 366–367
- sessions, sharing**, 127
- `setAttribute()` **method, HttpSession**, 7
- SFSB (stateful session bean)**
 - cache management, 202–204, 271
 - components, 159–161
 - in-memory replication, 204–207
 - lifecycle of, 191–192
- shallow fetches, ORM**, 232
- shared libraries**
 - altering `bigrez.com` to use, 292–293
 - Java EE (Enterprise Edition), 286–287
- shutdown, server states**, 542
- shutdown triggers, for handling stuck threads**, 544–545
- Simple Network Management Protocol. *See* SNMP (Simple Network Management Protocol)**
- Simple Object Access Protocol. *See* SOAP (Simple Object Access Protocol)**
- Simple Protected GSSAPI Negotiation (SPNEGO)**, 510
- Single Sign On. *See* SSO (Single Sign On)**
- single table, inheritance strategies**, 188
- single-site deployment strategies**
 - complex WebLogic Server clusters, 734–737
 - overview of, 731–732
 - simple WebLogic Server clusters, 732–734
- SLAs (Service Level Agreements)**, 632
- SLSB (stateless session bean)**
 - business-tiers interfaces, 67
 - with CMP entity beans, 238, 240–242
 - components, 158–159
 - controllers for calling business services, 68–69
 - with JDBC, 237–239
 - with JPA, 242–245
 - lifecycle of, 190–191
 - pooling, 201–202
 - session façade requirements, 236–237
- Smalltalk**, 32
- snippets, precompiling JSP components and**, 139
- SNMP (Simple Network Management Protocol)**, 607–614
 - communication with WebLogic SNMP agent, 609
 - community names, 608–609
 - creating SNMP agents for servers, 607
 - monitoring models, 607
 - specifying UDP port, 607–608

SNMP (Simple Network Management Protocol) (continued)

SNMP (Simple Network Management Protocol) (continued)

- traps, 609–612
- WebNMS MIB Browser and Trap Viewer as manager for
 - WebLogic Server, 612–614
- `SnoopFilter`, **servlet filtering**, 23–24
- SOAP (Simple Object Access Protocol)**
 - attachments, 338–341
 - configuring web service security and, 358–359
 - developing WSDL-first web services, 317–318
 - overview of, 301
 - style/use attributes, 323–325
 - transport-level security, 348
 - WebLogic Server support, 304
 - wrapped convention, 315
- SOAP Messages with Attachments specification**, 338, 341
- SOAP over HTTP**
 - deciding which web service operation to invoke, 325–326
 - developing WSDL-first web services, 315
 - invoking web services via, 303
 - ports (innovation points), 320
- SOAP over JMS**
 - invoking web services via, 303
 - JAX-RPC support for, 323
 - ports (innovation points), 320
- SOAP UI utility**, 313–314
- `soapUI`, 724
- socket multiplexing (muxers)**
 - managing server sockets, 527–529
 - using native I/O muxer, 662
- socket muxers**
 - for managing server sockets, 527–529
 - using native I/O muxer, 662
- software**
 - installing WebLogic Server software in development environment, 695
 - requirements for development environment, 692–693
- Solaris, tuning**, 646–649
- Solarmetric**, 164
- source code management (SCM)**, 692
- split directory development**, 698–700
- SPNEGO (Simple Protected GSSAPI Negotiation)**, 510
- spreadsheets, creating with servlets and JSP pages**, 26–29
- The Spring Framework**, 157
- Spring MVC. See also MVC (model-view-controller)**
 - architecture for `bigrez.com` example, 59
 - choosing components for web application, 81
 - `DispatcherServlet`, 131
 - servlet-centric architecture, 50–53
 - web service handler, 334
- SQL injection attacks**, 178
- `sql` library, **JSLT**, 19
- SQL statements**
 - batch updates, 675
 - data access requirements, 234
 - JPA supporting, 178
- SSL (Secure Sockets Layer)**
 - application authentication, hostname verification, and trust managers, 486–487
 - configuring one-way SSL, 472–473
 - configuring two-way SSL, 474–477
 - debugging, 477–478
 - hardware acceleration, 756
 - obtaining X.509 certificates, 470–472
 - one-way SSL, 350–351
 - overview of, 469–470
 - SSL/TLS compared with, 755
 - transport-level security and, 303, 348
 - two-way SSL, 351–352
 - writing security aware Java clients, 482–487
 - WS-Security and, 302
- SSL persistence**, 756
- SSL/TLS**
 - setting up, 469
 - SSL compared with, 755
- SSO (Single Sign On)**
 - managing SSO partners in SAML, 506–507
 - overview of, 501
 - from Windows OS desktops, 511–516
- stack traces**
 - Java, 684, 687–688
 - WebLogic Server, 688–689
- stand-alone deployment, of enterprise applications**, 274–275
- stand-alone modules, vs. system modules**, 288
- Standard Trust Keystore, for CA certificates**, 574–575
- standards, web service**, 301–302
- standards-defined descriptors, vs. vendor-specific descriptors**, 221
- standby state, servers**, 542
- start scripts, in WebLogic Server clusters**, 734–737
- state comparison, optimistic locking strategy**, 213–214
- state comparison version strategy, OpenJPA**, 259
- stateful session bean. See SFSB (stateful session bean)**
- stateful web services**, 341–343
- stateless session beans. See SLSB (stateless session bean)**
- static classes, advantages of SLSB components over**, 159
- static persist, global load balancers**, 743, 749
- sticky load balancing algorithm**, 743
- store-and-forward. See SAF (store-and-forward)**
- StreamMessage object, JMS**, 370
- stop scripts, in WebLogic Server clusters**, 734–737
- Structs, in servlet-centric architecture**, 48–50
- `struts-config.xml`, 49
- stubs**
 - invoking web services and, 321
 - replica-aware, 536
 - web service clients using stub Java class to invoke web services, 318
 - web services, 306
- style attributes, SOAP**, 323–325
- style sheets**, 37
- subdeployments, in JMS modules**, 384
- submission, presentation-tier controls**, 44–45
- subscribers, JMS**
 - client identifiers for, 379
 - durable, 368
 - indexed subscribers, 406
 - overview of, 364
- Sun NetBeans**, 704
- superclasses**
 - `bigrez.com` domain model, 252
 - mapped superclasses supported in JPA, 188
- superusers**, 648

transport-level security

synchronization logic, parallel request processing an, 5

system classloader, 530

system modules, 288

system performance

core principles, 644–645

overview of, 644

tuning AIX, 649

tuning application servers. *See* application servers, tuning

tuning applications, 645–646

tuning HP-UX, 649–650

tuning JVMs. *See* JVMs (Java Virtual Machines), tuning

tuning Linux, 650

tuning network, 651–652

tuning operating systems, 645–646

tuning Solaris, 646–649

tuning Windows, 650

system requirements, 55–56

system security, 755

system-level process dumps, 684–685

T

table per class, inheritance strategies, 188

Tag Extra Information (TEI) class, 20

tag files, custom tags vs., 20–21

Tag Handler class, 20

Tag Library Descriptor (TLD), 20

tags, JSP

calling custom, 16

JSTL tag libraries, 19

list of available, 8–9

using for selected behaviors, 19–22

using JSTL tags to reduce scriptlet code, 15

target runtime, configuring in Eclipse, 706

targeted offers, web application, 103–105

TCP

time wait interval, 566

tuning, 648

unicast-based clustering, 553

TCPMON, SOAP UI utility, 313–314

TEI (Tag Extra Information) class, 20

templates, JMS, 387

terminal destinations, JMS, 416

Testing tab, WebLogic Console, 312

TextMessage object, JMS, 370, 400

thin clients, 749

thread safety

EJB components and, 159

JSP pages and, 11

servlets and, 4–5

threads

checking for stuck threads, 527

generating Java thread dumps, 686

handling stuck, 544–545

listen threads, 529

optimizing thread management, 662–664

understanding thread states, 684–685

work manager constraints, 524

threat models, 749

throughput, troubleshooting, 682–683

Tiles framework

external view assembly using, 66–67

resolving view names as Tiles definitions, 86–88

reviewreservation definitions, 101

selectdates, 91

selectroomtype definition, 98

tiles-config.xml, 104

viewproperty, 91

tiles-config.xml, 80, 104

time variables, in unit tests, 722

time wait interval

TCP, 566

tuning operating systems, 646

time-to-deliver extension, JMS, 388

time-to-live (TTL)

configuring clusters and, 553

JMS multicast messages and, 419

TLD (Tag Library Descriptor), 20

TLS (Transport Layer Security), 469

topics, JMS

choosing destination type, 398

consuming messages from a distributed topic, 378–380

producing messages to a distributed topic, 377–378

unit-of-order used with, 415

TopLink

enhancing, 261

JPA derived from, 189

vs. Kodo, 260

Oracle, 164

TopLink, converting from Kodo to

changing persistence provider, 261

enhancing TopLink, 261

fixing incompatibilities, 262–263

overview of, 260

running unit tests, 261–262

summary, 263–264

testing integrated application, 263

traffic management. See global and local traffic management

transacted sessions, JMS, 366

transaction affinity, JMS, 374

transaction batching, MDBs and, 208–209

transaction isolation

JDBC, 199–201

referencing EJBs and, 197–198

SLSBs, 159

transaction model, database access best practices, 674

transactions

business logic requirements and, 229

collocation requirements in multiple site deployment

strategies, 740

commit protocols, 581

debugging transaction timeouts, 715

EJB deployment descriptors, 270–271

MDBs and, 428–429

options for selecting for JDBC data sources, 580–582

transactions, JMS

asynchronous, 368

using, 417–419

Transport Layer Security (TLS), 469. *See also* SSL/TLS

transport-level security

overview of, 348–349

transport-level security (*continued*)

transport-level security (*continued*)

- SSL and, 303
- web services, 350–352
- Trap Viewer**, 612–614
- traps, SNMP**, 609–612
- troubleshooting performance problems**
 - bottleneck identification and correction, 678–680
 - CPU utilization and application server throughput, 682–683
 - disk I/O, 684
 - generating Java thread dumps, 686
 - generating system-level process dumps, 684–685
 - Java stack traces and, 684
 - overview of, 677
 - preparation for, 678
 - problem resolution, 680–682
 - reading core dumps, 686–687
 - reading Java stack traces, 687–688
 - response times, 683–684
 - understanding thread states, 684–685
 - understanding WebLogic Server stack traces, 688–689
- trust, between domains**, 510–511
- trust managers, in SSL**, 486–487
- TTL (time-to-live)**
 - configuring clusters and, 553
 - JMS multicast messages and, 419
- tuning settings, deployment descriptors**, 272–273
- Two-Phase Commit, **transaction commit protocols**, 582
- two-way SSL**
 - configuring, 474–477
 - overview of, 469
 - preventing man-in-the-middle attacks, 755
- typed partitioning, options for mapping inheritance**, 233–234

U

- UDDI (Universal Description, Discovery, and Integration)**, 302
- UDP**
 - multicast-based clustering, 553
 - unicast and multicast messages, 741
- unicast messages, UDP**, 741
- unicast-based clustering, TCP**, 553
- unidirectional relationships**
 - associations, 248
 - mapping associations, 231
- Uniform Distributed Destinations, WebLogic Server 9.0**, 373
- Unit of Order, MDBs and**, 209
- unit tests**
 - bigrez.com, 258–259
 - co-developing with code, 716–717
 - database services in unit test context, 721
 - dependency injection and, 721
 - for entity classes, 189
 - fixing incompatibilities between Kodo and TopLink, 262–263
 - importance of, 715–716
 - JUnit, 717–718
 - mocking frameworks, 719–720
 - out-of-container testing, 718–719
 - overview of, 715
 - performance and concurrency testing, 724–725
 - running in TopLink, 261–262

- testing TopLink integrated application, 263
- testing web interfaces, 722–724
- testing web services, 724
- time and other variables, 722
- unit-of-order, JMS**, 413–415
- unit-of-work, JMS**, 415–416
- Universal Description, Discovery, and Integration (UDDI)**, 302
- Unix/Linux OSs, installing node manager on**, 561
- updates**
 - JPQL bulk updates, 179–180
 - requiring explicit service calls, 71
- url-pattern descriptor**, 127
- URL-rewriting**, 5
- URLs**
 - callback URL in one-way web services, 345
 - determining URL of deployed web service, 312
 - handler mapping components and, 85–86
 - Java servlets and, 3
 - JMS clients and, 382–383
- use attributes, SOAP**, 323–325
- user configuration file, credentials stored in**, 566
- user names, remote start attributes for WebLogic Servers**, 563
- user site components**
 - constructing, 76
 - core reservation process components. *See* reservation process
 - display and source components, 74
 - reservation information components, 76–79
 - targeted offers, 103–105
- userapp-servlet.xml**, 80–81
- usernames, booting WebLogic Server**, 500–501
- user-password lockouts**, 755
- users**
 - creating required users when deploying web applications, 152–153
 - cross domain user, 511
 - managing, 496
- Users and Groups tab, WebLogic Console**, 152–153
- UTMLUnit**, 722

V

- validation**
 - client-side validation, 38–39
 - date selection validator, 92–93
 - of objects and data elements in business layer, 230
 - server-side validation, 40
- value objects**
 - design pattern, 33
 - generic JPA functionality, 250–252
 - performance best practices and, 666
- VCS (Veritas Cluster Server)**, 735–736
- vendor-specific descriptors, vs. standards-defined descriptors**, 221
- Veritas Cluster Server (VCS)**, 735–736
- version fields, optimistic locking and**, 181–185
- version strategies, OpenJPA**
 - field types, 212–213
 - state comparison, 213–214

WebLogic Deployer utility (`weblogic.Deployer`)

versioning management, WebLogic Server applications,
628–630
in-place redeployment and, 628
production redeployment and, 628–630

versions, storing version information in hidden fields, 187

vertical partitioning, options for mapping inheritance,
233–234

vertical scaling, in server deployment, 730–731

view resolver, web application component, 86–88

viewproperty, Tiles definitions, 91

Virtual EAR feature
OEPE (Oracle Enterprise Pack for Eclipse), 698
overview of, 709–710

W

WAN (wide-area network) replication
configuring, 556–557
implementing one cluster per site, 744–745
overview of, 535

.war (web application archive) file
creating, 144–145
deploying web applications as, 147
developing code-first web services, 307–312
developing WSDL-first web services, 317
exploded deployment and, 282

watches, WLDF, 621, 627–628

web application architecture
for `bigrez.com` example, 59
Java EE application tiers, 31–32
Java EE design patterns, 33
MVC (model-view-controller), 32–33
overview of, 31
presentation tier. *See* presentation-tier

web application archive file. *See* .war (web application archive) file

web applications
building. *See* building web applications
calling custom tags in JSP pages, 16
custom tags used for selected behaviors, 19–22
deploying. *See* deploying web applications
error handling, 13–15
Excel files using servlets and JSP pages, 26–29
Expression language (EL) used in JSTL calls, 16–18
Java EE technologies for, 31
JSTL tag libraries, 19
locking down, 753–754
overview of, 13
packaging. *See* packaging web applications
securing, 488–493
servlet filtering used for common behaviors, 22–26
using JSTL tags to reduce scriptlet code, 15
viewing servlet code generated for JSP page, 30

web container best practices
JSP (JavaServer Pages), 669
overview of, 667
servlets, 669–670
session management, 667–668

web interfaces, unit testing, 722–724

web server plug-ins

common parameters, 570
configuring, 567–570
debugging parameters, 571–572
response time parameters, 571

web services
adding to `bigrez.com`, 359–361
calling asynchronously, 343–344
configuring security for, 358–359
container architecture, 303–304
creating dynamic, 327–334
creating one-way, 344
customizing mappings between Java and XML, 346–347
deciding which operation to invoke, 325–327
defining security policies, 349–350
determining URL of, 312
developing code-first, 307–312
developing for WebLogic Server, 304–307
developing web service clients, 318–322
developing WSDL-first, 314–318
handlers, 334–338
identifying and correcting bottlenecks, 679
implementing stateful, 341–343
JAX-RPC standard, 322–323
managing web services partners in SAML, 507–508
message-level security, 352–358
overview of, 301
returning multiple asynchronous, 344–346
review, 361
security features, 348–349
SOAP attachments, 338–341
SOAP style/use attributes, 323–325
standards, 301–302
testing, 313–314
transport-level security, 350–352
types of application components, 74
unit tests, 724
writing security aware Java clients, 486

Web Services Description Language. *See* WSDL (Web Services Description Language)

Web Services Description Language Compiler (WSDLC), 314–316

WEB-INF directory
bundled libraries and, 285
deploying web applications and, 147–148
internal components in, 128
precompiling JSP components and, 138
web application directory structure, 126–128

WebLogic Configuration Wizard, 501, 549

WebLogic Console
changing WebLogic configuration, 539
creating users and groups when deploying web applications, 152–153
defining PKI credentials, 357–358
deploying enterprise applications, 298
deploying web applications, 150–151
determining URL of deployed web service, 312
monitoring server performance, 601–603
node manager, 541–542
WLST script, 359

WebLogic Deployer utility (`weblogic.Deployer`)
deploying `bigrez.com` application, 297–298

WebLogic Deployer utility (`weblogic.Deployer`) (continued)

WebLogic Deployer utility (`weblogic.Deployer`) (continued)

deploying web applications, 148–150

WebLogic Diagnostic Framework. *See* WLDF (WebLogic Diagnostic Framework)

WebLogic Scripting Tool. *See* WLST (WebLogic Scripting Tool)

WebLogic Security Framework, 449–465

auditing, 459–460
 authentication, 455–457
 authorization, role mapping, and adjudication, 458–459
 certification path, 460–461
 credential mapping, 460
 debugging, 464
 embedded LDAP server for, 450–451
 external security stores, 465
 identity assertion, 457–458
 managing security providers, 461–464
 RDBMS security store, 451–452
 realms and providers, 453–455

WebLogic Server

administration. *See* administration, WebLogic Server
 application management. *See* application management, WebLogic Server
 application monitoring. *See* application monitoring, WebLogic Server
 best practices for web applications. *See* web applications
 booting, 500–501
 defining users and groups, 105
 deployment architecture. *See* deployment architecture, WebLogic Server
 developing web services for, 304–307
 EJB container. *See* EJB container, WebLogic Server
 installing on development machines, 695
 integrating AQ with older server releases, 440
 Java servlets. *See* Java servlets
 JMS. *See* JMS (Java Message Service)
 JMS clients. *See* JMS clients
 JMS Foreign Server, 437
 JNDI provider, 437
 JSP pages. *See* JSP (JavaServer Pages)
 MDBs (message-driven beans) used with, 436
 Messaging Bridge, 433
 packaging applications. *See* packaging enterprise applications
 PKI Credential Mapper, 357
 registering servlets, 3
 SAF (store-and-forward), 434–436
 sample HTTP servlets, 3
 security model, 36
 steps in packing and deployment process, 125
 Uniform Distributed Destinations in version 9.0, 373
 XML Catalog, 321

WebLogic Test Client, 313

`weblogic.Admin`, 600–601
`weblogic.appc`, 137–139
`weblogic-application.xml` **descriptor file**, 280–282
`weblogic.Deployer` (**WebLogic Deployer utility**), 148–150, 297–298
`weblogic-ejb-jar.xml`, 272–273
`weblogic-version` **section, of** `weblogic.xml`, 136
`weblogic.xml`, 133–137

WebNMS MIB Browser, as SNMP manager for WebLogic Server, 612–614

WebServiceContext **object**, JAX-WS, 342

`web.xml`

defining filter in, 24
 descriptor file, 128–133
 error-page element, 15
 web application security and, 105–106

white box testing, 719

whole server migration. *See* WSM (whole server migration)

wide-area network (WAN) replication, 535, 556–557

Windows OSs

SSO (single sign-on) from Windows desktop, 511–516
 tuning, 650

Windows OSs, installing node manager on, 561

`wlappc`, **Ant build tool**, 137

`wldeploy`, **Ant build tool**, 150

WLDF (WebLogic Diagnostic Framework)

exporting diagnostic data, 626
 harvesting metric information, 625–626
 instrumenting `bigrez.com`, 622–625
 managing diagnostic archive, 627–628
 modules, 287–288
 overview of, 621–622
 watches and notifications, 627–628

`WLIOTimeoutSecs`, **web server plug-ins**, 570

`WLLogFile`, **web server plug-ins**, 571–572

`@WLServlet` **annotation, for registering servlets**, 3, 28

WLST (WebLogic Scripting Tool)

for changing WebLogic configuration, 539
 commands, 597–600
 modes of execution and operational states, 596
 overview of, 595–596
 start script for starting node managers, 565–566
 starting admin server, 596–597
 WebLogic Console and, 359

work managers, 521–527

components of, 523
 configuring, 589–590
 configuring servlets or JSP to use, 527
 constraints, 524
 defining, 525–526
 dispatch policy and, 136

workstations, for developers, 692

wrapped document style, SOAP, 315, 324–325

wrapper exceptions, JAX-WS, 310

WS Reliable Messaging. *See* WS-RM (WS Reliable Messaging)

WS-Addressing

deciding which web service operation to invoke, 326–327
 supporting callback URL, 345
 WebLogic Server support, 302

WSDL (Web Services Description Language)

Ant `wsdl` task for accessing, 321
 deciding which web service operation to invoke, 325–327
 defining web service security policies, 349
 developing WSDL-first web services, 314–318
 overview of, 302
 WebLogic Server support, 304
 WSDL-first web services, 306, 318

`wsdl` **task, Ant build tool**, 321

WSDL (Web Services Description Language Compiler), 314–316

zero consumer queues, JMS

wsdlc task, Ant build tool, 314–316

WS-I (WS-Interoperability)

- overview of, 302
- support for SOAP style/use combinations, 324–325
- supporting HTTP cookies, 343
- web services container architecture, 303
- WSI Analyzer checking, 313–314

WSI Analyzer, SOAP UI utility, 313–314

WS-Interoperability. See WS-I (WS-Interoperability)

WSM (whole server migration)

- migrating JMS servers and, 585
- overview of, 638–639
- requirements for using, 639
- setting up, 639–640
- WebLogic Server, 564

WS-Policy, defining web service security policies, 349

WS-RM (WS Reliable Messaging)

- JAX-RPC support for, 323
- SAF support for, 434
- WebLogic Server support, 302

WS-SecureConversation 1.3, 323

WS-Security

- defining web service security policies, 349
- message-level security, 348, 352–358
- overview of, 302
- SAML Token Profile, 507
- Web Service Identity Providers, 508
- web services container architecture, 303

WS-Trust 1.3 (clients only), 323

X

X.509 certificates, 351, 355, 565

- certification path, 460–461
- certification path for, 460–461
- identity assertion and, 457
- obtaining, 470–472
- overview of, 469–470
- SAML and, 502

XA Connection Factory Enabled, JMS, 386

XML

- APIs for dynamic web services, 333
- customizing mappings between Java and XML, 346–347
- embedding XML tags within JSP pages, 12, 16
- mapping to Java, 306
- Provider//Dispatch approach supporting XML processing tools, 333
- SOAP using, 301–302

XML Catalog, WebLogic Server, 321

XML library, JSLT, 19

XMLMessage, JMS, 400–401

XP (Extreme Programming), 716

XPath, 333

XQuery, 333

XSLT, 333

Z

zero consumer queues, JMS, 375