

Index

- A**
- AbstractFactory pattern, 274–275
- Action pane, Web Services Explorer, 452
- action property, DocBook validator, 590
- Actions
 - adding to WSDL editor design view, 627–635
 - defined, 124–125
- Add and Remove Projects. *See* Run On Server wizard
- addTestSuite method, 519–520
- adminas tool, 561, 569
- Adobe Flash, 116
- Agents, in REST architectural style, 424
- “Ajax: A New Approach to Web Applications” (Garrett), 206
- AJAX. *See* Asynchronous JavaScript and XML (AJAX)
- AJAX Toolkit Framework (ATF), 92, 206, 679
- All-in-one zips, WTP installation, 98–100
- AllTests.java listing, 519
- Annotations
 - adding new content to XSD using, 607
 - developing JPA with WTP using, 415–418
 - in EJB 3.0, 326–327
 - in J2EE development, 35
- Ant tasks, Web service, 444
- antlr.jar, 166
- AP (Attachments Profile), 471
- Apache
 - Axis2, 680
 - Beehive project, 129
 - Cactus. *See* Cactus
 - Derby. *See* Derby
 - Jakarta project, 262
 - Struts. *See* Struts
 - W3C WSDL 2.0, 680
- “Apache Derby Fortune Server Tutorial” (Anderson), 78
- API. *See* Application Programming Interfaces (APIs)
- Applets, Java, 44
- Application controller, MVC, 123–127
- Application infrastructures, 113
- Application logic
 - designing Java Web applications, 119
 - domain model vs., 298–299
 - handling in Model View Controller, 123
 - layered Web application design, 121
 - services as mix of business and, 132
- Application Programming Interfaces (APIs)
 - creating transparency for persistence layer, 370
 - defined, 683
 - in future versions of WTP, 584
 - JDBC, mapping objects to database using, 371–372
 - JST support for J2EE, 35
 - server, 553

Web Tools Platform, 3
 in WTP architecture, 28
 WTP goal of providing platform, 24
Application-scoped objects, 279
Application servers, deploying application to, 352–355
application.xml. *See* **Deployment descriptors**
Architectural Styles and the Design of Network-based Software Architectures (Fielding), 425
Architectures
 building business tier with
 component, 297
 component, 297
 WTP, 27–28
article.dockbook listing, 583–584
Articles, writing WTP, 39
Artifacts, J2EE development, 44–45
 “Asynchronous Java Script Technology and XML (AJAX) With Java 2 Platform, Enterprise Edition” (Murray), 206
Asynchronous JavaScript and XML (AJAX), 205–206
AsyncLeagueFacadeBean.java listing, 362–363
ATF (AJAX Toolkit Framework), 92, 206, 679
Attachments Profile (AP), 471
Audio, server preference, 107, 264
Autobuilds, disabling for validation, 250–251
Automated testing, 511–512
Axis
 Axis2, 680
 deploying Top-Down Web services, 441, 444–445
 WTP including, 100
Axis Emitter Web services preference, 108
AxisServlet, 445–446, 463

B

Backups, 105
Base time, sorting by, 544–545
Basic Profile (BP), 471
BEA Workshop, 15, 670

Bean-managed persistence (BMP), 372, 392
Bean-managed transaction demarcation, 343–344
Beehive project, Apache, 129
BindingGenerationAction, 630
Birds-of-a-Feather (BOF), WTP, 15
Blackbox testing, 510, 683
BMP (bean-managed persistence), 372, 392
Body, SOAP, 423
Bottom up Java Bean Web service, 84–85, 460
Bottom-Up Web services. *See* **Web services**, **Bottom-Up development**
BP (Basic Profile), 471
Breakpoints
 debugging in JSP with, 61–63
 debugging servlets, 69–70
 setting in JSP source files, 280
Bugs. *See also* **Debugging**
 fixing unassigned WTP, 39
 reporting to Bugzilla, 38
Bugzilla, 38
Build section, Maven, 186
Build types, WTP, 92–95, 98–100
Builders, 51, 140
 “Building Web Services the REST Way” (Costello), 426
Business logic tier, 297–367
 common design for, 300–301
 as core of system, 199
 defined, 297
 domain model and. *See* **Domain models**
 implementing business logic, 121–122
 Java Web applications, 120, 128
 Message-Driven Beans and. *See* **Message-Driven Beans (MDBs)**
 overview of, 115–116, 297–300
 presentation logic vs., 199–200
 session EJBs and. *See* **Enterprise JavaBeans (EJBs)**, **session**
Business objects
 as business logic code, 121–122
 defined, 127
 providing services. *See* **Service-Oriented Architecture (SOA)**

C

Cache preference, Internet, 107

Cactus, 520–528

- creating integration test, 523–527
- instance variables accessed by, 521–522
- overview of, 520–521

Callisto Discovery Site, 43

Cascading Style Sheets (CSS)

- features of, 203–204
- J2EE Web pages containing, 45
- specifying styles with, 230–234

Cascading Style Sheets, level 2, CSS2 Specification [CSS2], 230

Catalog extension listing, 652

Change control, 161

Chemical Markup Language (CML), 421

Class structure page, New Servlet wizard, 479

Classpath management

- with Maven and WTP, 184
- Web applications, 175–176

cleanup method, DocBook validator, 595

Client-side cookies, 289

Client-side processing, 204–205

Client tier, 205

Closed content model, XML, 607

CML (Chemical Markup Language), 421

CMP. *See* Container-managed persistence (CMP) entity beans

Code generators, 140

CodeGear JBuilder, 670

Commands

- Maven, 188
- WST tools for, 34

Committer, to WTP, 39

commons*.jar libraries, 166, 186

Compilation, JSP, 279

Components

- building business tier with, 297
- vendor cooperation on common, 16

config instance variable, 522

Configuration, 276

Configuration and release management, 161

confirmation.html listing, 243

Connections, database, 72–75

Consuming services, 132

Container-managed persistence (CMP)

entity beans

- adding.ejbCreate and finder methods, 401–408
- adding Ice Hockey CMP data access object, 408–410
- adding to project, 396–400
- defined, 372, 392
- preparing Derby for, 393–394
- preparing JBoss for, 394–395, 397
- preparing XDoclet for, 394–396
- testing, 410–414

Container-managed transaction

demarcation, 343–344

Containers

- EJB, 148, 326
- J2EE, 117

Content assist

- CSS, 232–233
- HTML, 214–215
- JSP, 284–286
- source editors, 219
- templates, 223–225

Content negotiation, 425, 426

Content types

- declaring DocBook, 601–604
- using SSE for mixed, 604

Context URLs, Cactus tests, 527

Cookies, 289

Cookies (Laurent), 289

Core J2EE Patterns (Alur et al.), 326

Create Enterprise JavaBean wizard, 359–363

create method, entity beans, 392

Create Servlet wizard, 270–273

Create Web service command, 83–86, 460

Create XML File wizard, 248–250

CreateLeagueAction.java listing, 364–365

createtables.sqlpage listing, 382

CSS. *See* Cascading Style Sheets (CSS)

Customer support, 20–21

D

Dai, Naci, xxv

Dali incubator project

defined, 327

developing EJB 3.0 and JPA support in, 414

overview of, 679

Data access layer, domain models, 315–320

Data access tools, WST, 33

Data entry

JavaScript form validation, 236–247

presentation tier, 204–205

Data layer

implementing, 386–391

Java application frameworks and, 128

overview of, 115–116

Data Output view, 73, 77

Data tools

disconnecting Derby when finished using, 78

JDBC used by, 71

setting Data preferences for, 106–107
WST, 33, 374

Data Tools Platform (DTP), 33, 678

Database access, 71–82

adding to Web application, 78–82

connecting to database, 72–75

with Derby, 376–378

executing SQL statements, 75–78

overview of, 71–72

when not connected, 378–379

Database, creating, 375–385

adding sample data to tables, 382–385

adding tables, 379–381

creating connection to Derby, 375–378

Database Explorer

accessing database when not connected, 378–379

adding data to tables, 384–385

connecting to database, 73–75

connecting to database with Derby, 376–378

defined, 33

filtering items shown by, 378

opening SQL Scrapbook Editor in, 76

Database.java listing, 78–79

.dbk files, 602–604

De jure standards, 26

Debugging

JavaScript, 245–247

JSP, 61–64, 280, 287–288

plug-ins, 573–580

servlets, 69–70

unassigned WTP bugs, 39

WTP supporting JSP, 28

Default character set, 602

Define a New Server page. *See* Run On Server wizard

Definition files, generic server, 566–570

Dependencies

building Web client, 349–350

creating stateless session bean, 344–349

implementing DocBook validator, 587

implementing Maven, 186–187

Dependent modules, 165

Deployment descriptors

in Cactus tests, 526

creating EAR project, 157, 159

creating EAR project with multiple modules, 170

creating servlets, 67, 478–479

as J2EE development artifact, 44–45

overview of, 649

specifying runtime policies in, 326

Derby

creating database connection to, 78–82, 376–378

developing CMPs with, 393–394

disconnecting from, 77–78

downloading and installing, 71–72

in multiple Web applications, 78

overview of, 375

single-user embedded vs. shared mode, 390

Derby Database configuration listing, 394

Design Patterns (Gamma et al.), 274

Design tab, XML editor, 250, 252

Design view, WSDL editor

adding custom actions to, 627–635

customizing extensibility elements in, 614–617

overview of, 613

- Designing Interfaces* (Tidwell), 203
 - “Develop HTML widgets with Dojo” (Kusakov), 206
 - Developer view, 161
 - “Developing the WTP with Eclipse” (Hutchinson), 39
 - Development artifacts, J2EE, 44–45
 - Development expenses, reducing, 16–18
 - Development project organization, 137–197
 - advanced Web projects, 160–165
 - dividing Web module into multiple projects, 171–180
 - enterprise application example, 165–171
 - Maven for Web applications.
 - See Maven
 - Web projects and J2EE applications.
 - See Web projects
 - Directives, JSP markup, 279
 - Disconnect command, 77–78
 - Discovery, 32, 495
 - dispose method, LabelProvider, 616
 - Distributed applications, 325
 - Distributions, for revenue, 19
 - DocBook, 583–604
 - creating custom marker type, 598–600
 - creating extension plug-in, 585
 - declaring content type, 600–604
 - overview of, 583–584
 - SSE framework and, 604
 - DocBook validator, 585–598
 - best practices, 598
 - implementing, 587–598
 - overview of, 585–586
 - WTP validation framework and, 586–587
 - DocBook validator contribution to plugin.xml listing, 591–592
 - DocbookValidator class, 589
 - DockBookValidator.java listing, 592–594
 - Document/literal wrapped, 462
 - Document Object Model (DOM), 31, 613–614
 - Document style, SOAP binding, 461
 - Document Type Definitions (DTDs), 257–261
 - adding catalog of resources to XML catalog, 654–657
 - adding single resource to XML catalog, 653–654
 - contributing resources to XML catalog, 648–650
 - registering for resource resolution in XML catalog, 650–651
 - WST tools for, 30–32
 - doGet method, HelloServlet.java, 68–69
 - Dojo, 206
 - DOM (Document Object Model), 31, 613–614
 - Domain, GlassFish, 554
 - Domain models
 - application logic vs., 298–299
 - building with POJO, 298
 - object models as, 298
 - Domain models, building, 301–324
 - data access layer, 315–320
 - J2EE Utility Projects, 301–304
 - object model, 304–310
 - service layer, 310–314
 - testing, 320–324
 - Drag effect, 20
 - DTDs. *See* Document Type Definitions (DTDs)
 - DTP (Data Tools Platform), 33, 678
 - Dynamic invocation, of Web services, 464
 - Dynamic Web project wizard, 142–148, 268
 - Dynamic Web projects
 - creating, 52–56, 142–148
 - default layout for, 141–142
 - defined, 141
 - Maven, 183–185
 - overview of, 51, 267–269
 - servers, 261–267
 - servlets, 270–279
- ## E
- E-mail address obfuscation, 234–239
 - Echo.wsdl listing, 610–611
 - Eclipse
 - Bugzilla system, xxi
 - installing for WTP, 92
 - PHP Development Tools project, 674, 680–681
 - Pollinate project, 126
 - simple projects, 139

- SOA Tools platform project, 681
- Tools layer, 28
- WTP architecture and, 28
- Eclipse Foundation, 14**
- Eclipse Management Organization (EMO), 14–15, 22, 683**
- Eclipse Modeling Framework: A Developer's Guide* (Budinsky), 614
- Eclipse Modeling Framework (EMF)**
 - adding custom elements to, 614
 - installing for WTP, 92
 - JST support for J2EE and, 35
 - in WTP architecture, 28, 44
- EclipseCon, 14–15**
- Economics, WTP, 15–22**
 - generating revenue, 19–22
 - Open Source, 15–16
 - reducing development expenses, 16–18
- Edge servers, 207**
- Edit Profiling Options wizard, 543–545**
- Editing**
 - EARs, 159–160
 - HTML page with source editor, 218–230
 - JSPs, 56–57
- EJB. *See* Enterprise JavaBean (EJB)**
- EJB Design Patterns* (Marinescu), 326
- ejbCreate method, GameBean, 401–408**
- ejbdoctlet, 109, 334**
- EMF. *See* Eclipse Modeling Framework (EMF)**
- EMO (Eclipse Management Organization), 14–15, 22, 683**
- employee.xsd, 658–659**
- enabled property, DocBook validator, 589**
- Encoded use, SOAP binding, 461**
- encodeURL method, of**
 - HttpServletResponse class, 289
- Encoding and Line Delimiter preferences, 108**
- Encoding, SOAP, 424**
- Enterprise Application Client (JAR) module, 164, 176–177**
- Enterprise Application wizard, 155–159**
- Enterprise Applications (EARs), 154–160**
 - adding existing Web and EJB modules to, 155–159
 - creating new Web or EJB module in, 154–155
 - creating project with multiple modules, 165–171
 - editing, 159–160
 - EJB project combined with other modules in, 148
 - overview of, 154
 - running session EJB application, 353
 - structural model for, 163–164
- Enterprise Information System (EIS) tier, 205, 300**
- Enterprise JavaBean (EJB). *See also* Entity beans**
 - 3.0 specification, 326–327
 - adding project to EAR, 168–169
 - alternatives to, 326–327
 - Client Projects, 152
 - containers, 117–118, 148
 - creating applications, 148–154
 - JARs, 164
 - JST tools for, 36
 - projects, 335–339
 - Tools component of JST, 374
- Enterprise JavaBean (EJB) client module**
 - adding to J2EE dependencies in Web project, 170–171
 - creating, 168–169
 - creating EJB projects, 337
- Enterprise JavaBeans (EJBs), session, 325–358**
 - adding JBoss to workspace, 329–332
 - alternatives to EJBs, 326–327
 - building Web client, 349–352
 - creating EJB projects, 335–339
 - creating session beans, 339–349
 - defined, 36
 - developing 3.0 with WTP, 356–358
 - entity beans vs., 392
 - generating code with XDoclet, 332–335
 - Message-Driven Beans on server-side as, 358
 - overview of, 325–329
 - running application, 352–355
 - as type of EJB, 326

Enterprise JavaBeans (Monson-Haefel), 325

Enterprise Service Bus (ESB), 681

Entity beans, 392–418

adding CMP, 396–401

adding ejbCreate and finder methods,
401–408

adding Ice Hockey CMP data access
object, 408–410

defined, 326

developing JPA with WTP, 414–418

mapping objects to database using,
372–373

preparing JBoss, Derby and XDoclet,
393–396

testing CMP implementation,
410–414

EPIC, 674

Errors

404, 69, 287

DocBook validator, 596, 598

EJB, 152

JavaScript, 245–247

reporting, 38

requirements vs., 38

source editor highlighting, 219

WSDL validation and naming,
643–644

Essential XML Quick Reference

(Skonnard), 429

Exadel Studio, 670

Expenses, reducing development, 16–18

Exporting, workspace preferences, 110

Extensible HTML (XHTML), 45

Extensible Markup Language (XML)

built on SSE, 604

describing structured/semi-structured
data, 248–252

Design and Source tabs, 250–252

DocBook validation using, 595–596

DTD using, 257–261

HTTP on Internet using, 421–422

J2EE Web applications and, 45

resource resolution in, 646

source editors for, 31

SQL scrapbook pages in, 76

transforming into HTML with XSLT,
252–257

Web service interoperability using,
471

WST tools for, 31–32

*Extensible Markup Language (XML) 1.0
(Third Edition)* [XML10], 248

Extensible Stylesheet Language

Transformations (XSLT)

features of, 204, 248

J2EE Web applications and, 45

transforming XML data into HTML,
252–257

Extensions. *See also* Web Service

Description Language (WSDL), creating
extensions

content type, 601–602

customized DocBook markers, 599

DocBook extension plug-ins, 585

DocBook validator, 587–592

future versions of WTP, 584

installing for WTP, 92

new runtime target handlers, 562

new server runtimes, 559

wizard fragments and icons, 565–566

Extreme programming (XP), 511

F

Facade layer, WSDL editor, 628–629

Façades, service, 311

Facets. *See also* Project Facets page

defined, 551–552

and runtime components, 563–565

Feature Details button, 103

Feedback, about this book, xxii

File Associations page, HTML, 212–213,
218

File types, supporting new. *See* DocBook

Filters

database connection, 378–379

DocBook validator, 590

WSDL editor, 623–624

find method, 392

Find page, Web Services Explorer,
497–498

finder methods, entity beans, 401–408

Firefox, 245–247

Flexible projects, 35, 161

FolderUriResolver.java listing, 664
 Fonts, conventions used in this book, xxii
 Formatting, source editor, 219
 form.html listing, 366
 forward method, JSP, 280
 404 errors, handling, 69, 287
 fullBuild property, DocBook validator, 590
 Functional requirements tests
 overview of, 510
 running with HttpUnit, 521

G

Game form, simple, 413–414
 Game with JPA annotations, 416–418
 GameBean ejbCreate method listing, 402
 GameBean.java listing, 403–408
 GameDetail.java listing, 456
 GameException.java listing, 457
 GameFormat.java listing, 489–490
 Game.java listing, 308–310
 GameServlet.java listing, 412–413
 GEF. *See* Graphical Editing Framework (GEF)
 Generate WSIL command, 504
 GenerateBindingAction.java listing, 632–634
 Generic servers, 554, 566–570
 Geronimo, 148
 GET method, 280, 425
 GetEndpoint method, 467, 472–473
 GetGameDetail operation, update Web service, 455–456, 458
 GetGameDetails method, JSP test client, 467–470
 getImage method, LabelProvider, 616
 getLogs method, 522
 getParameter method, 69
 getSchedule operation, 448, 452–454, 491
 getSchedulingRule method, DocBook validator, 594
 getServletConfig method, 522
 getSession method, JSP, 280
 getUser method, JSP, 280
 GlassFish
 adding runtime target handler, 562
 adding server runtime support, 558–561

 creating server adapter plug-in, 556–558
 defining runtime server types, 561–562
 extending Server tools UI, 565–566
 facets and runtime components, 563–565
 generic server framework and, 566–570
 server runtime, 554–556
 testing server adapter, 573–580

GlassFish Runtime wizard, 574–576

glassfish.serverdef listing, 567–570

Global preferences, 110

Goals, Maven, 188

Google

 developing servlet that applies XSLT, 270–279
 generating interest in AJAX, 206
 non-support of XML files, 270

Grammar

 defining for XML documents with DTDs, 257
 in well-formed documents, 257
 in XML content models, 607–608

Graphic design, 203–204

Graphical Editing Framework (GEF)

 installing for WTP, 92
 as prerequisite for WTP, 44
 in WTP architecture, 28

H

Header, SOAP, 423

“Hello, world”. *See* Web Tools Platform (WTP), Quick Tour

hello-world.jsp listing, 61, 81

HelloServlet.java listing, 68, 80–81

helper extension, DocBook validator, 590

Hibernate

 developing JPA with WTP by using, 415
 mapping objects to database using, 373–374
 overview of, 130

Homesteading the Noosphere (Raymond), 23

HTML. *See* Hypertext Markup Language (HTML)

HTML 4.01 Specification (HTML401), 212

The HTML Sourcebook: A Complete Guide to HTML (Graham), 212

HTTP. *See* Hypertext Transfer Protocol (HTTP)

HTTP Recording wizard, 536–538

HttpServletRequest object, 119

HttpUnit, 528–533

creating system test with, 529–533

downloading, 529

overview of, 527–528

running functional tests with, 521

Hyperlinking, in Web architecture, 425

Hypertext Markup Language (HTML)

adding styles with CSS, 230–234

creating files, 211–218

creating source editors for, 31

data entry form validation, 237–247

editing with source editor, 218–230

J2EE Web pages containing, 45

screen scraping of, 421

transforming XML data into, 252–257

XHTML, 45

Hypertext Transfer Protocol (HTTP), 289–295

modifying and resending messages, 293–295

overview of, 289

recording test, 536–538

TCP/IP monitor and, 290

using with XML on Internet, 421–422

viewing with TCP/IP monitor, 291–293

I

I-builds (integration builds), 94

IBM

1.5 SDK requirement for Cactus, 522

development of WTP, 14–15

PHP Tools project of, 680–681

proposing WTP, 13

Rational Application Developer (RAD) for WebSphere Software, 670–671

IceHockey Jdbc DAOI Implementation listing, 389–390

IceHockeyCMPDAOImpl.java listing, 408–410

IceHockeyDAOImpl.java listing, 316–320

IceHockeyFacade.java listing, 316

IceHockeyJdbcDAOImpl.java listing, 387–389

IceHockeyWebTest project, 514–515

Icons, extending Server tools UI, 565–566

IDE (integrated development environment), 14, 16–18, 583

Idempotence, 425

IEC (International Electrotechnical Commission), 26

IETF (Internet Engineering Task Force), 26

Image handles, 616–617

Importing, exported workspaces, 110

Incremental approach, integration testing, 520–521

Incremental property, DocBook validator, 589

The Inmates Are Running the Asylum (Cooper), 200

Input

designing Java Web applications, 119

JSP test client, 468, 472

layered Web application design, 121

Input controller, MVC

defined, 123–124

overview of, 124

Inquiry interface, 495

INRIA (Institut National de Recherche en Informatique), 15

inspection.wsil, 501, 504–505, 506

Install New Server dialog, 102–103

Install/Update wizard, 95–98

Installed runtimes

JBoss, 332

server preferences for, 107, 264–265

Instance variables, accessed by Cactus, 521–522

Institut National de Recherche en Informatique (INRIA), 15

- Integrated development environment (IDE), 14, 16–18, 583
 - “Integrating Cloudscape and Tomcat” (Bader), 78
 - Integration builds (I-builds), 94
 - Integration tests
 - with Cactus, 520–527
 - as functional requirements tests, 510–511
 - overview of, 520–521
 - Interaction design, 200–202
 - Interfaces
 - service, 312–314
 - stateless session bean, 341–342
 - International Electrotechnical Commission (IEC), 26
 - International Organization for Standardization (ISO), 26
 - International Telecommunication Union (ITU), 26
 - Internet
 - developing standards for, 26
 - preferences, 106, 107
 - WST tools for, 30
 - Internet Engineering Task Force (IETF), 26
 - Interoperability, Web services, 470–477
 - Introduction, 3–11
 - contents of this book, 4–5
 - League Planet Web site, 9–10
 - organization of this book, 5–8
 - overview of, 3–4
 - source code examples, 8–9
 - Invoke a WSDL Operation page, Web Services Explorer, 501
 - Invoke button, 86
 - ISO (International Organization for Standardization), 26
 - ITU (International Telecommunication Union), 26
 - IWSDLValidator.java listing, 642
- J**
- J2EE. *See* Java 2 Enterprise Edition (J2EE)
 - J2EE Module Dependencies
 - adding dependency to utility project, 175–176
 - adding EJB module to, 170–171
 - adding/removing modules to EARs, 159–160
 - implementing Web services, 446–447
 - J2EE Standard Tools (JST)
 - creating, 25–27
 - defined, 685
 - EJB Tools component of, 374
 - installing for WTP, 92
 - server tools and APIs as components of, 553
 - subproject, 34–37
 - in WTP architecture, 28
 - J2EE Utility Projects
 - adding dependency to, 175–176
 - adding JARs to build path of, 176
 - creating, 173–175
 - creating for domain models, 301–304
 - Java Build Path page, 176–177
 - J2SE Development Kit (JDK)
 - developing JPA, 414
 - obtaining and installing, 42
 - required for developing JSP, 49–50, 92
 - setup of, 103
 - J2SE (Java 2 Standard Edition), 44–45
 - J2SE Runtime Environment (JRE), 44, 49–50
 - Jakarta project, 262
 - jakarta*.jar, 166
 - JAR (Java archive) files, 148
 - Java 2 Enterprise Edition (J2EE)
 - creating new Web projects, 141–148
 - defined, 684
 - Enterprise Applications. *See* Enterprise Applications (EARs)
 - generating revenue by developing WTP, 20–22
 - history of IDEs, 14
 - IDEs available in 2003, 14
 - for Java Web applications, 116–117
 - JST tools for, 35
 - modules, 140–141
 - value proposition for tool vendors of, 16–18
 - Java 2 Enterprise Edition (J2EE), creating Web applications, 44–60

- adding server runtime environment, 46–50
 - creating and editing JSP file, 56–57
 - creating dynamic Web project, 51–56
 - as iteration 1 of “Hello, world”, 41
 - overview of, 44–45
 - running JSP on server, 57–60
 - summary, 57–60
- Java 3 Standard Edition (J2SE), 44–45**
- Java API for XML Processing (JAXP)**
interface, 595
- The Java API for XML Processing (JAXP)*
[JSR63], 270
- Java applets, 206**
- Java application frameworks, 128–130**
- Java Architecture for XML Binding*
[JAXB], 257
- Java archive (JAR) files, 148**
- Java Bean Identity, Web Service wizard, 461**
- Java Build Path page, Utility Project, 176–177**
- Java client proxy**
generating, 464–468
generating for Query Web service client, 477–478
testing Web service with, 468–470
- Java Community Process (JCP), 26, 684**
- Java Database Connectivity (JDBC)**
Data tools accessing databases with, 71
mapping classes to database with, 386–389
mapping objects to database with, 371–372
replacing persistence implementation with, 389–390
testing data layer code, 390
- Java Development Kit. *See* J2SE Development Kit (JDK)**
- Java Development Tools (JDT), 28, 57–60**
- Java Edit Model (JEM)**
defined, 92
prerequisite for downloading WTP, 44
in WTP architecture, 28
- Java EE 5 (Java Enterprise Edition 5), 679**
- Java Enterprise Edition 5 (Java EE 5), 679**
- Java Management Extension (JMX), 35, 263**
- Java Message Service (JMS), 358–367**
adding queue to JBoss, 363
client, 358
as client-side of Message-Driven Beans, 358–359
creating Message-Driven Bean, 359–363
creating Web client, 363–366
overview of, 358–359
- Java Persistence Architecture (JPA)**
developing with WTP, 414–418
vs. entity beans, 372
mapping objects to database using, 373–374
overview of, 373
Tools project (Dali), 92, 679
- Java Profiling, 543–545**
- Java projects, 139**
- Java service implementation, 455–459**
- Java Servlet Development Kit, 262**
- Java Specification Request (JSR), 36, 684**
- Java Utility Projects. *See* J2EE Utility Projects**
- Java-version command, 42**
- Java Web Application Archive (WAR) files, 147–148**
- Java Web applications, 116–120**
- Java Web tools, 669–673**
- JavaBeans, 44**
- Javadoc listings, 4**
- Javadoc tags, 35**
- JavaScript, 234–247**
data entry form validation, 236–247
debugger, 245–247
e-mail address obfuscation, 234–239
J2EE Web pages containing, 45
overview of, 234
- JavaScript Object Notation (JSON), 424**
- JavaScript: The Definitive Guide, Fourth Edition* (Flanagan), 234
- JavaServer Faces (JSFs)**
application controller design, 125–126
defined, 26, 129
Oracle development of, 15
project, 36

- Tools project, 678
 - view mechanisms, 126–127
 - WTP features for, 92
 - JavaServer Pages (JSPs)**
 - adding Java scriptlet to, 60–61
 - creating and editing, 56–57
 - debugging, 61–64
 - dynamically generating Web pages with, 280–289
 - generating test client, 464–468
 - generating Web pages, 45
 - Java Web applications using, 118–120
 - JDK required for development of, 49
 - JST tools for, 35–36
 - markup, 279–280
 - running on server, 57–60
 - test clients, 472
 - testing Web service with, 468–470
 - WTP architecture for debugging, 28
 - JavaServer Pages (Whitehead)*, 280
 - JAXB. See Java Architecture for XML Binding [JAXB]**
 - JAX-RPC Web service standard**, 36, 481–482
 - JAX-WS**, 679
 - JAXP (Java API for XML Processing) interface**, 595
 - JBoss**
 - adding queue to, 363
 - CMP development with, 394–395, 397
 - creating EJB projects with, 336
 - defining for running application, 352–353
 - EJB development with, 148, 329–332
 - IDE for Eclipse, 671
 - WTP development with, 15
 - JBoss Server wizard**, 330–331
 - JCP (Java Community Process)**, 26, 684
 - JDBC. See Java Database Connectivity (JDBC)**
 - JDK. See J2SE Development Kit (JDK)**
 - JDT (Java Development Tools)**, 28, 57–60
 - JEM. See Java Edit Model (JEM)**
 - JMS. See Java Message Service (JMS)**
 - JMX (Java Management Extension)**, 35, 263
 - JOFFAD**, 181
 - JOOnAS**, 148
 - JPA. See Java Persistence Architecture (JPA)**
 - JRE (J2SE Runtime Environment)**, 44, 49–50
 - JSFs. See JavaServer Faces (JSFs)**
 - JSON (JavaScript Object Notation)**, 424
 - JSP editor**, 56–57
 - JSP: JavaServer Pages (Burd)*, 280
 - JSPs. See JavaServer Pages (JSPs)**
 - JSR 31: XML Data Binding Specification [JSR31]*, 257
 - JSR 45: Debugging Support for Other Languages [JSR45]*, 280
 - JSR (Java Specification Request)**, 36, 684
 - JST. See J2EE Standard Tools (JST)**
 - JUnit JAR**, 191
 - JUnit test case wizard**, 190–192
 - “JUnit Test Infected” (Beck), 509
 - JUnit tests**, 512–520
 - configuring for Cactus, 526–527
 - contributing, 39
 - creating HTTP recording test, 537–538
 - creating new project for, 513–515
 - creating system test with HttpUnit and, 529–533
 - creating test case, 515–518
 - of domain models, 320–324
 - limitations of, 520–521
 - with Maven, 190–193
 - test suites, 518–520
- L**
- Label Decorations preference, Data tools**, 107
 - LabelProvider**, 616
 - LAMP**, 206, 685
 - Launching preferences, servers**, 107
 - Layers**
 - service, 130–132
 - Web application, 115–116
 - Web application design, 120–122
 - League DAO interface listing**, 386
 - League Planet**
 - case study, 133–134
 - description of, 9–10
 - inspiring examples in book, 5

LeagueDAO interface listing, 315
 LeagueFacade EJB 3.0 stateless session
 bean listing, 357
 LeagueFacadeBean, 340, 344, 346–348
 LeagueFacadeBean.java listing, 347–349
 LeagueFacade.java listing, 312–313
 LeagueFacadelmpl.java listing, 313–314
 LeagueFacadeTest.java listing, 322–323
 League.java listing, 306–307
Libraries
 getting HttpUnit, 530
 importing Struts into EAR project,
 166, 168
Licenses
 Dynamic Web project, 54–55
 EJB Project validation, 337–338
 installing WTP via Update Manager,
 98
 verifying WTP installation, 103–104
 WTP policy on redistributing 3PC,
 100
 List All WSDL Services page, Web Services
 Explorer, 504–505, 506–507
 Literal use, SOAP binding, 461
 LocalizedMessage object, 595–596
 log4j*.jar, 166
 Logical layers, physically distributing, 325
 logIn method, JUnit tests, 517
 login-user.jsp listing, 65
 login.jsp listing, 285–286
 LoginServlet.java, 282–283, 522–527
 LoginServletTest.java listing, 525
 logOut method, JUnit tests, 517
 logout.jsp listing, 286–287
 Lomboz, 15
 lookupFullname method, 78, 86–88

M

M-builds, 94
The Magic Cauldron (Raymond), 16
 main method, 44
 Maintenance, 509
 Maintenance builds (M-builds), 94
 Managed backing beans, JSF, 125, 127
 Management webContent folder, 177–178
 Mandel, Lawrence, xxv

Margins, source editor indicators in, 219
 markerIDs, 599
 Markers, creating custom, 598
 Markup, JSP, 279–280
Mastering JavaBeans (Roman et al.), 326
 Mathematical Markup Language
 (MathML), 421
 Maven, 180–196
 console output, 188–189
 creating project with, 183–185
 defined, 685
 defining Maven POM, 185–190
 description of conventions for, 182
 maintaining project dependencies to
 JARs, 175–176
 overview of, 180–183
 project information and reports with,
 192–196
 unit tests with, 190–192
Maven: A Developer's Notebook
 (Massol/O'Brien), 181
 MBeans, 363
 MDBs. *See* Message-Driven Beans (MDBs)
 Meat and Poultry Markup Language
 (mpXML), 421
 Mentoring services, 22
 Message-Driven Beans (MDBs), 358–367
 adding queue to JBoss, 363
 creating, 359–363
 creating JMS Web client, 363–367
 defined, 326
 entity beans vs., 392
 overview of, 358–359
 Message-log validator, 471, 474–476
 Message-oriented middleware (MOM), 358
 messages, SOAP
 formatting and processing Web
 services, 83
 monitoring, 87–88
 using JSP test client, 86–87
 Method execution statistics view, TPTP,
 544–545
 Method stubs, Create Servlet wizard,
 271–272
Methods
 checking messages for WS-I
 compliance, 472

DocBookValidator, 594
 REST architectural style, 425
Methods pane, JSP test client, 468
Mevenide, 188
Microsoft, Windows Communication Framework (WCF), 680
Middle tier, defined, 205
Model 1 architecture, 118–120
Model 2 architecture, 118
Model View Controller (MVC)
 application controller, 125–127
 input controller, 124
 Model 2 architecture as adaptation of, 118
 overview of, 123–124
 for Web, 123–127
Modules
 building Web client, 349–350
 creating EAR project with multiple, 165–171
 creating EJB projects, 337
 dependent, 165
 editing EAR, 159–160
 Maven Web, 183
 structural model for, 162–164
 types of J2EE, 164–165
 Web Application, 154–159
Modules, EJB
 adding to Enterprise Applications, 155–159
 creating EJB applications, 151
 creating in Enterprise Applications, 154–155
 defined, 148
MOM (message-oriented middleware), 358
Monitoring
 HTTP sessions, 289–295
 ports, with TCP/IP Monitor, 291–292
 SOAP messages, 87–88
 WTP newsgroup, 37
Monospace fonts, xxii
Mozilla Thunderbird, 37
mpXML (Meat and Poultry Markup Language), 421
Multi-tab editors, WST XML tool, 31
Multi-tiered distributed applications, 205

Multitier Model of Designing Enterprise Applications with the J2EE Platform, Second Edition (Singh), 205
MVC. See Model View Controller (MVC)
MyEclipse, 672

N

Naming conventions
 future coordinated Eclipse releases, 96
 servlets, 66
 validation for, 640–644
 Web service client proxies, 465
 XDoclet beans, 340
NamingConventionValidator.java listing, 642–643
Navigation pane, Web Services Explorer, 451–452
.NET programming, vs. J2EE, 14
New Cactus Test Case wizard, 523–527
New Class wizard, 305–306
New Connection wizard, 73–75, 376–378
New CSS File wizard, 231–233
New Dynamic Web Project wizard
 building Web client, 349–352
 creating Dynamic Web project, 268–269
 creating JUnit test project, 514–515
 testing server adapters, 576–577
New EJB Project wizard, 149–150, 336–339
New Enterprise Application Project wizard, 155–159
New Extension wizard, 559
New File wizard, 597
New HTML File wizard, 237–247
New HTML Page wizard, 212–213
New Java Class wizard, 588, 631
New Java Project wizard, 320–321, 536
New Java Server Page wizard, 350–352
New Java Utility Project wizard, 173–174
New JavaScript File wizard, 241–242
New JavaServer Page wizard, 56
New JSP wizard, 65–66, 283–284
New JUnit Test Case wizard, 320–324, 515–520
New Modules button, 157

- New Plug-In Project wizard
 - DocBook extension plug-in, 585
 - resource resolution extension plug-in, 647–648
 - server adapter plug-in, 556–558
 - WSDL extension plug-in, 612
 - New Project wizard
 - DocBook validator, 597
 - Static Web project, 208–211
 - New Report wizard, 538–539
 - New Server Runtime wizard
 - adding JBoss to workspace, 330–331
 - adding server runtime environment, 48–49
 - configuring servers, 264–265
 - testing server adapter, 573–575
 - New Servlet wizard, 66–67, 411–414, 478–480
 - New SQP Scrapbook Page wizard, 379
 - New Static Web Project wizard, 208–211
 - New Utility Project wizard, 302–304
 - New Web Project wizard, 142
 - New wizard, 523–527
 - nextCatalog extension listing, 656
 - Nightly builds (N-builds), 94
 - The Non-Designer's Web Book* (Williams and Tollett), 203
 - Nonfunctional requirements tests, 510–511.
 - See also* Test and Performance Tools Platform (TPTP)
 - Noosphere, 23
- O**
- O/R (Object-Relational) design, 371, 373–374
 - O/R (Object-Relational) mapping problem, 373
 - OASIS (Organization for the Advancement of Structured Information Standards), 26, 654–657
 - Obfuscation, e-mail address, 234–239
 - Object models, 298, 304–310
 - Object-Oriented (OO) programming, 373
 - Object Relational Mapping (ORM) framework, 130
 - Object-Relational (O/R) design, 371, 373–374
 - Object-Relational (O/R) mapping problem, 373
 - ObjectWeb, 15
 - ObjectWeb Lomboz, 672
 - Online Help, 4
 - OO (Object-Oriented) programming, 373
 - Open content model, 607–608, 609
 - Open Registry page, Web Services Explorer, 496–497
 - Open Services Gateway Initiative (OSGi), 128–129, 327
 - Open Source movement, 3, 16–18, 19–22
 - Oracle, WTP development, 15
 - Organization for the Advancement of Structured Information Standards (OASIS), 26, 654–657
 - ORM (Object Relational Mapping) framework, 130
 - OSGi Alliance, 128–129
 - OSGi. *See* Open Services Gateway Initiative (OSGi)
 - Output
 - Data tools preference, 107
 - designing Java Web applications, 120
 - Maven console, 188–189
 - Maven test, 191–192
- P**
- Page flow, 126
 - Page IDs, 533
 - Passwords, JSP, 281, 287–288
 - PDE (Plug-in Development Environment), 573–580
 - Performance testing
 - as nonfunctional requirement test, 510–511
 - overview of, 533–540
 - profiling with TPTP, 540–545
 - with TPTP. *See* Test and Performance Tools Platform (TPTP)
 - Perl Web tools, 674
 - Persistence layer
 - business tier interface to, 300
 - creating APIs for transparency of, 370

- defined, 369
- designing, 122, 370–374
- Persistence tier, 369–420**
 - creating database, 375–385
 - data layer, 386–391
 - designing, 370–374
 - entity beans and. *See* Entity beans
 - iterations overview, 374–375
 - overview of, 369–370
- Personas, 200–202**
- Perspectives, dynamic Web project, 54–56**
- PHP Tools project, 680–681**
- PHP Web tools, 674**
- PHPEclipse, 674**
- PicoContainer, 130, 326–327**
- Plain Old Java Objects (POJOs)**
 - building domain models with, 298
 - business tier design and, 300
 - creating EJB 3.0 bean with, 356–358
 - defined, 686
 - Java Persistence Architecture based on, 414
 - testing domain models, 320–324
- Plain Old XML (POX), 686**
- Platform APIs, 28**
- Plug-in Development Environment (PDE), 573–580**
- Plug-ins**
 - developing WTP, 39
 - DocBook, 585
 - generating revenue with WTP, 19–20
 - resource resolution, 647–648
 - server adapter, 556–558
 - testing and debugging, 573–580
 - WSDL, 612
 - in WTP architecture, 27–28
- plugin.xml file, 559–560**
- PMC. *See* Project Management Committee (PMC)**
- Point-to-point (PTP) messaging, JMS, 358–359**
- POJOs. *See* Plain Old Java Objects (POJOs)**
- POM. *See* Project Object Model (POM)**
- pom.xml listing, 193–194**
- Popup Dialog Selection, Web services, 109**
- Portability, 117**
- Portal servers, 207**
- Portlets, 207**
- POST method, JSP, 281**
- POX (Plain Old XML), 686**
- Preferences, 105–110**
 - checking messages for WS-I compliance, 472–474
 - content types, 603
 - Data tools, 107
 - EJBDoclet, 334
 - Internet, 107
 - overview of, 106
 - server, 263–265
 - Server tools, 107
 - sharing settings, 110
 - snippet, 224
 - source editor, 219–221
 - template, 222
 - validation, 107–108, 251
 - Web and XML, 108
 - Web services, 108–109
 - XDoclet, 109
 - XML catalog, 656
- Presentation layer, 115–116, 122, 128**
- Presentation logic, 199–200**
- Presentation tier, 199–295**
 - adding HTML page, 211–218
 - adding interactivity. *See* JavaScript
 - adding server, 261–267
 - creating Dynamic Web projects, 267–269
 - creating static Web project, 208–211
 - CSS, 230–234
 - defined, 200
 - DTD, 257–261
 - editing HTML page with source editor, 218–230
 - graphic design, 203–204
 - interaction design, 200–202
 - introduction to, 199–200
 - JSP, 279–289
 - monitoring HTTP sessions, 289–295
 - servlets, 270–279
 - snippets, 224–230
 - structure of, 204–207
 - templates, 221–224

- XML, 248–252
 - XSLT stylesheets, 252–257
 - Problems view, WTP validation**
 - displaying DocBook validator errors in, 598
 - filtering DocBook markers from, 599–600
 - service element naming error message in, 643
 - Profile and Logging perspective, 544**
 - Profile Compliance and Validation, Web services, 109, 472**
 - Profiling with TPTP, 540–545**
 - Project deliverables, 160**
 - Project Facets page**
 - EAR, 157–158
 - EJB applications, 149–151, 336
 - new Dynamic Web project, 52–53, 144
 - New Java Utility Project, 173–174, 303–304
 - Static Web project, 208, 210
 - testing server adapter, 575–577
 - Project Management Committee (PMC)**
 - defined, 685
 - in structure of WTP, 23
 - of WTP development, 15
 - Project Navigator, 35**
 - Project Object Model (POM)**
 - content of, 185
 - creating for Web application, 185–190
 - defined, 686
 - Maven project directory structure in, 182
 - project information with, 193–196
 - Project Topology preference, Web services, 109**
 - Projects**
 - defined, 51
 - organizing. *See* Development project organization
 - Properties**
 - DocBook validator, 589–591
 - generic server definition, 566–567
 - stateless session bean, 341
 - style rule, 230
 - Properties view, 617–627**
 - Proxy Settings, Internet, 107**
 - PTP (point-to-point) messaging, JMS, 358–359**
 - Public identifiers (publicId), 649, 653–654**
 - Publications, Web service, 495**
 - Publish-and-subscribe (pubsub) messages, 358–359**
 - Publishing**
 - adding new servers, 570–573
 - defined, 32
 - dividing Web module into multiple projects, 180
 - dynamic Web project, 147–148
 - EAR, 154–155
 - EJB, 152–153
 - flexibility of DocBook, 584
 - Web services. *See* Web services, discovering and publishing
 - Publishing interface, 495**
 - pubsub (publish-and-subscribe) messages, 358–359**
 - PyDev, 675**
 - Python Web tools, 675**
- Q**
- Query parameters**
 - adding Java scriptlet to JSPs, 61
 - creating servlets, 69
 - debugging, 63
 - Query Results page, Web Services Explorer, 498–499**
 - Query Web service client, 477–478**
 - Query.java, 449–451**
 - QuerySOAPImpl.java listing, 448**
 - Query.wsdl listing, 437–438**
 - Queues, JMS, 359, 363**
 - Quick tour. *See* Web Tools Platform (WTP), quick tour**
- R**
- R-builds, 93–94**
 - RadRails, 675**
 - RAR (Resource Adapter for J2CA), 164**
 - Rational Application Developer (RAD), 15, 670–671**

Recommendations, W3C, 422

Recording test, HTTP, 536–538

Redo, with source editor, 219

References

- AJAX, 205–206
- Apache Derby, 78
- Cloudscape and Tomcat integration, 78
- cookies, 289
- CSS, 230
- Dojo development, 206
- DTDs, 257
- EJBs, 325, 326
- EMF, 614
- graphic design, 205
- HTML, 212
- interaction design, 200
- Java standard for XML binding, 257
- JavaScript, 234
- JSP markup, 280
- Maven, 181
- multi-tiered design, 205
- noospheres, 23
- REST architectural style, 425
- REST for Web services, 426
- SQL, 379
- structured source editors, 219
- on testing, 509
- WTP development with Eclipse, 39
- XML, 248
- XSD, 429
- XSLT, 248

References, online

- Adobe Flash, 116
- AJAX toolkit, Dojo, 206
- Cactus, 521, 522
- Derby, 71–72
- DocBook, 583
- EJB containers, 148
- example code, 7, 11
- feedback about this book, xxii
- finding particular references, 7
- Firefox, 245
- GlassFish, 554
- graphic design, 203
- HttpUnit, 529
- Java Web tools, 669–673
- JBoss, 330

JDK, 42

JOFFAD, 181

Maven, 182, 184

Perl Web tools, 674

PHP Web tools, 674

Python Web tools, 675

reporting bugs to Eclipse, xxi

Ruby Web tools, 675

SOAP schema, 620

Tomcat, 46–47, 265

TPTP, 535

Web site for this book, xxi–xxii

WST subproject, 29

WTP 2.0, 677–681

WTP bug tracking system, 38

WTP downloads page, 98

WTP installation, 43

WTP newsgroup, 37

WTP wizards and project structure,
148

XDoclet, 333

XSLT, 248

Refresh command, 384

**Registry Details page, Web Services
Explorer, 496–497**

Release builds (R-builds), 93–94

**Remote Method Invocation (RMI),
369**

remove method, entity beans, 392

**Replacing persistence implementation
with JDBC, 389–390**

Reports

- Maven project, 192–196
- for performance test using TPTP,
538–539

**Representation State Transfer (REST),
424–427**

request instance variable, 521

Requirements

- errors vs., 38
- suggesting WTP improvements, 38

Resource Adapter for J2CA (RAR), 164

Resource Management, 109

Resource resolution, customizing, 645–666
contributing resources to XML
catalog. *See* XML catalog
creating extension plug-in, 647–648

- creating folder URI resolver, 661–664
 - defined, 646
 - implementing, 657–659
 - overview of, 645–647
 - URI resolution framework, 659–661
 - Resource resolution strategy, 657–665**
 - creating folder URI resolver, 661–665
 - defined, 646
 - overview of, 657–659
 - URI resolution framework, 659–661
 - Resources**
 - defined, 646
 - overview of, 164
 - REST architectural style, 424
 - response instance variable, 522**
 - REST. *See* Representation State Transfer (REST)**
 - Result pane, JSP test client, 468, 472**
 - Revenue, generating, 19–22**
 - Rich desktop applications, 117**
 - RMI (Remote Method Invocation), 369**
 - RPC style, SOAP binding, 461**
 - Ruby Web tools, 675**
 - Run As command, 30, 57–58**
 - run Extension, 588–589**
 - run method, 631**
 - Run On Server wizard**
 - configuring servlets, 275–278
 - developing session EJBs, 352–355
 - running JSP on server, 57–59
 - testing server adapters, 577–580
 - Run SQL, 382**
 - Run wizard, Cactus, 526**
 - Runtime. *See* Server runtime environments (runtime)**
 - Ryman, Arthur, xxvi**
- S**
- S-builds (stable builds), 94**
 - Sans serif fonts, xxii**
 - SAP NetWeaver Developer Studio, 672**
 - Scenario Defaults, Web services, 109**
 - schedule-css.html listing, 230–231**
 - schedule-dtd.xml listing, 259–260**
 - schedule-js.html listing, 235**
 - schedule.css listing, 232–233, 490**
 - schedule.dtd listing, 258–259**
 - schedule.html listing, 215–218**
 - schedule.jsp listing, 351–352, 488–489**
 - ScheduleServlet.java listing, 273–274, 486–487**
 - schedule.xml listing, 250, 255–256, 430–431**
 - schedule.xsd listing, 432–434, 448**
 - schedule.xml listing, 253–255**
 - Schemas**
 - adding catalog of resources to XML catalog, 654–657
 - adding single resource to XML catalog, 651–654
 - contributing resources to XML catalog, 648–650
 - registering for resource resolution in XML catalog, 650–651
 - Scopes**
 - defining WTP, 23–24
 - defining WTP subproject, 25–26
 - JSP documents and, 279
 - score-confirmation.jsp listing, 485–486**
 - score-form.html listing, 237–238**
 - score-form.jsp listing, 483–484**
 - score-validator.js listing, 242–243, 491**
 - ScoreException.java listing, 457–458**
 - Screen scraping, 421**
 - Scriptlets**
 - adding to JSP, 60–61
 - debugging JSP, 61–64
 - markup for JSP with, 279
 - SDOs (Standards Development Organizations), 26**
 - selectionChanged method, 631**
 - Selector style rule, 230**
 - Semantic gap, in OO programming, 373**
 - Serialization, Java, 369**
 - Server adapters**
 - adding to WTP, 554
 - defined, 552
 - JST, 34–35
 - plug-ins, 556–558
 - supporting server runtime environments, 29, 262–263, 265
 - testing and debugging plug-ins, 573–580

- Server and Runtime preference, Web services, 109
- Server runtime environments (runtime)
 - adding to workspace, 29, 46–51, 262–263
 - business layer not requiring, 300
 - Cactus test warning, 522
 - defined, 551–552
 - developing JPA with WTP, 415
 - installing third-party content, 101–102
 - libraries and, 143
 - obtaining and installing, 262
 - structural model for, 162–164
 - supporting EJBs, 329–332
- Server Startup page, Web Service Client wizard, 465–466
- Server Startup page, Web Service wizard, 442–443
- Server tools
 - extending UI for new server, 565–566
 - functions of, 553
 - JST, 34–35
 - preferences, 106–107
 - WST, 29–30
- Servers
 - adding dynamic Web projects to, 263
 - creating dynamic Web projects, 261–267
 - defined, 551–552
 - delays during runtime, 355
 - deploying application to, 352–355
 - portal vs. edge, 207
 - running JSP on, 57–60
- Servers, adding new, 551–581
 - extending server tools UI, 565–566
 - facets and runtime components, 563–565
 - generic server definition, 566–570
 - GlassFish server runtime, 554–556
 - new runtime target handler, 562
 - new type for runtime, 561–562
 - overview of, 551–554
 - publishers, 570–573
 - server adapter plug-ins, 556–558
 - support for runtime, 558–561
 - testing server adapter, 573–580
 - writing generic server adapter, 554
- Servers view
 - displaying new Server in, 277–278
 - J2EE Module Dependencies, 446–447
 - running session EJB application, 355
 - viewing HTTP sessions with TCP/IP Monitor, 291
- Service definition, 441
- Service Details page, Web Services Explorer, 498–499
- Service infrastructures, 113
- Service layer, 300, 310–314
- Service-Oriented Architecture (SOA)
 - for component architecture problems, 297
 - overview of, 130–132
 - SOA Tools platform project, 681
- Services
 - adding service layer to application, 130–132
 - consuming, 132
 - defining, 130
 - providing service layer in League Planet case study, 133–134
- Servlet engine, 262
- Servlets, 64–70
 - adding dynamic content with, 270–279
 - creating, 65–69
 - debugging, 69–70
 - developing user interface based on, 478–480
 - development of, 262
 - integrating testing. *See* Cactus
 - Java Web applications using, 118–120
 - JST tools for, 35
 - overview of, 64–65
- Session beans. *See* Enterprise JavaBeans (EJBs), session
- Session Façade, EJB, 327–328
- session instance variable, 522
- Session-scoped objects, 279
- Sessions, HTTP. *See* Hypertext Transfer Protocol (HTTP)
- setActivePart method, 631
- setEndpoint method, 467, 472–473

- setInitParameter method, 522
- setMethods method, CMP, 401
- setUp method, Cactus, 525
- setUp method, JUnit, 515, 517–518
- Sharing settings, preferences, 110
- Simple Object Access Protocol (SOAP)
 - 1.2 platform, 680
 - adding extensibility elements to
 - WSDL editor, 617–627
 - binding in WSDL 1.1, 609–610
 - customizing extensibility elements in
 - WSDL editor design view, 614–617
 - defined, 686
 - deploying Java class as Web service, 460–462
 - interoperability of Web services and, 471
 - overview of, 423–424
 - REST style Web services and, 426–427
 - WSDL 1.1 binding for 1.2, 610
- Simple SOAP Binding Profile (SSBP), 471
- site goals, Maven, 188, 191, 196
- sjsas.xml listing, 570–573
- Sliders, Web Service wizard, 83–85
- Snippets, 224–230
- Snippets View, WST, 33
- SOA. *See* Service-Oriented Architecture (SOA)
- SOA Tools platform (STP) project, 681
- SOAP. *See* Simple Object Access Protocol (SOAP)
- SOAP engine, 83–85
- SOAPExtensibilityElementFilter.java listing, 625–626
- SOAPLabelProvider.java listing, 616, 622–623, 625
- Sorting, by base time, 544–545
- Source code examples, 8–9
- Source editors
 - creating with templates, 221–224
 - editing HTML page with, 218–230
 - HTML, 214–215
 - overview of, 218–219
 - SSE for developing, 31
 - user preferences controlling, 219–221
 - using snippets, 224–229
- Source preferences, 108
- Source tab, XML editor, 250
- Source view, WSDL editor, 613
- Spammers, e-mail, 234
- Specify Service Project Settings dialog, 441–442
- Spring Framework, 129, 326–327
- SQL. *See* Structured Query Language (SQL)
- SQL Scrapbook Editor, 33, 75–78
- src directory, 182–183
- SSBP (Simple SOAP Binding Profile), 471
- SSE. *See* Structured source editing (SSE) framework
- Stable builds (S-builds), 94
- Stand-alone components, EJB projects, 148
- Stand-alone Web resources, 646
- Standards
 - developing J2EE, 23–24
 - Internet and Web, 26
 - J2EE Web application, 141–142
 - role of SDOs in, 26
- Standards Development Organizations (SDOs), 26
- Stateless session beans, 339–349
- States, REST architectural style, 425
- Static Web projects
 - adding HTML page to, 208–211
 - creating, 208–211
 - defined, 141
 - editing HTML page with source editor, 211–218
 - overview of, 208
- Status pane, Web Services Explorer, 452, 501–502
- STP (SOA Tools platform) project, 681
- Structural model, Web project
 - defined, 161
 - Maven-style, 184
 - overview of, 162–164
- Structure, presentation tier, 204–207
- Structured Query Language (SQL)
 - accessing relational databases, 45
 - adding tables to databases, 379–383
 - mapping objects to databases, 371–372
 - using SQL Scrapbook Editor, 75–78

Structured source editing (SSE) framework
 overview of, 604
 WSDL editor extending, 613–614
 WST tools for, 31

Structured source editors
 JSP, 280
 overview of, 218–219
 templates, 221–224
 user preferences controlling, 218–219
 using snippets, 224–229

Structured Text Editors preferences, 219–221

Struts
 application controller design,
 125–126
 defined, 129
 importing into EAR project, 166, 168

struts*.jar libraries, 166, 186

Styles
 preferences, 108
 rules for, 230
 SOAP, 461
 specifying with CSS, 230

Stylesheets
 containing list of style rules, 230
 CSS. *See* Cascading Style Sheets (CSS)
 separating Web page from content
 with, 203
 XSLT. *See* Extensible Stylesheet
 Language Transformations (XSLT)

Subprojects, WTP
 J2EE Standard Tools, 34–37
 overview of, 24–27
 Web Standard Tools, 29–34

Syntax highlighting, 219
System identifiers (systemId), 649, 653–654
System testing, with HttpUnit, 528–533
System tests, 510–511, 528–533
**systemId (system identifiers), 649,
 653–654**

T

Table editor, 384–385

Tables, 379–385

Tag libraries (taglibs), 126–127

Tags, JSP markup, 279

TCP/IP Monitor

configuring JSP test client for,
 472–474
 modifying and resending messages,
 293–295
 session tracking, 290–291
 viewing HTTP sessions, 291–293

TCP/IP Monitor View, 87–88

Teams, development, 161, 510

tearDown method, Cactus test, 525

tearDown method, JUnit, 515, 517–518

Templates

for HTML page, 214
 for JSP, 284–285
 overview of, 221–224
 preferences, 108
 snippets vs., 226–229

**Test and Performance Tools Platform
 (TPTP), 533–540**

creating HTTP recording test,
 536–538
 creating performance test project, 536
 generating report, 538–539
 installing, 535–536
 overview of, 533–534
 profiling with, 540–545

Test case, JUnit, 513, 515–520

Test client, JSP, 86

Test Connection button, 74–75

Test Facility Defaults, Web services, 109

**Test Methods page, New Cactus Test Case
 wizard, 524**

Test suite, JUnit, 513, 518–520

testGetUser method, Cactus, 525

Testing, 509–547

advantages of early user, 16
 automated, 511–512
 for bugs, 39
 CMP implementation, 410–414
 data layer code, 390–391
 database connection to Derby, 378
 development considerations, 161
 domain models, 320–324
 for integration, with Cactus, 520–528
 with JUnit, 512–520
 naming convention validator, 643
 overview of, 509–511

- performance with TPTP, 533–540
 - profiling with TPTP, 540–545
 - session EJBs, 355
 - system with HttpUnit, 528–533
 - Top-Down Web services, 451–454
 - Web services for interoperability, 470–477
 - Text Editor preferences, 219–221
 - Thin clients, 116, 117
 - Third-Party Content (3PC), 92, 100–103
 - Tier-0, 205
 - Tier-1, 205
 - Tier-2, 205
 - Tier-3, 205
 - Tomcat
 - adding TCP/IP Monitor to, 291
 - adding to server runtime environment, 46–50
 - adding to workspace, 262–267
 - deploying Top-Down Web services, 440, 445
 - history of, 46, 262
 - launching in JVM process, 60
 - Tools, 669–675
 - developing set of J2EE, 24
 - generating revenue with WTP, 22
 - Java Web, 669–673
 - Perl Web, 674
 - PHP Web, 674
 - Python Web, 675
 - Ruby Web, 675
 - using URI resolution framework in, 660–661
 - Top-down development, 85. *See also* Web services, Top-Down development
 - Topics, JMS, 359, 363
 - TOPLink, 373–374, 415
 - TPTP. *See* Test and Performance Tools Platform (TPTP)
 - TPTP URL (JUnit) Test Definition Code Generation wizard, 537–538
 - Training, WTP, 21–22
 - Transactions, EJB bean-managed vs. container-managed, 343–344
 - Transformation API for XML (TrAX), 270, 274–275
 - Tutorials, writing WTP, 39
- ## U
- UDDI. *See* Universal Description, Discovery, and Integration (UDDI)
 - UDDI4J, 495–496
 - Understanding the New SQL: A Complete Guide* (Melton), 379
 - Undo, with source editor, 219
 - Uniform interfaces, REST, 425
 - Uniform Resource Identifiers (URIs), REST architectural style, 424
 - Uniform Resource Locator (URL)
 - creating HTTP recording test, 537–538
 - defined, 687
 - not revealing implementation technology in, 478
 - session tracking by rewriting, 289
 - specifying for Cactus tests, 527
 - WSL Internet tools for, 30
 - Uniform Resource Locator (URL) mapping
 - Create Servlet wizard, 271–272
 - creating login servlet, 282–283
 - creating servlet, 66
 - Unit tests
 - with Cactus, 521
 - as functional requirements tests, 510–511
 - with JUnit. *See* JUnit tests
 - with Maven, 190–192
 - Universal Description, Discovery, and Integration (UDDI)
 - accessing registries, 495
 - cloud, 496
 - defined, 687
 - searching registry for Web services, 495–501
 - Web Service Explorer support for, 32
 - Unknown league unknown schedule page, 491
 - Update Manager
 - installing WTP via, 95–98
 - overview of, 92–93
 - updating WTP with, 104–105
 - Update Web service, 455–456, 464–468
 - Update.java listing, 458–459

Updates

- installing WTP via Update Manager, 96–97
- WTP, 104–105
- UpdateScore method, 467, 468–469
- UpdateScore operation, update Web service, 455–457, 458
- UpdateScoreServlet.java listing, 481–482
- UpdateScoreTest.java listing, 530–532
- Update.wsdl, 462–463
- URI resolution framework, 647, 659–665
- URIResolverExtension.java listing, 663
- URIs (Uniform Resource Identifiers), REST architectural style, 424
- URL. *See* Uniform Resource Locator (URL); Uniform Resource Locator (URL) mapping
- User community
 - additional Open Source costs, 18
 - becoming part of WTP, 37
 - early testing by, 16
 - growing WTP, 39
- User interface (UI). *See also* Presentation tier
 - extending Server tools, 565–566
 - presentation logic as, 199–200
 - testing Web services in Web applications, 491–494
- User interface (UI), importing code, 480–491
 - GameFormat.java, 489–490
 - overview of, 489
 - schedule.css, 490
 - schedule.jsp, 488–489
 - ScheduleServlet.java, 486–487
 - score-confirmation.jsp, 485–486
 - score-form.jsp, 483–484
 - score-validator.js, 491
 - UpdateScoreServlet.java, 481–482
 - validator.css, 490
- userId, 281, 287–288
- User.java listing, 281–282
- UserTest.java, 516–518
- Using SQL (Groff), 379
- Utility class, XDoclet, 352
- Utility components, WST, 33–34
- Utility Modules, JARs, 164
- Utility Projects. *See* J2EE Utility Projects

V

- Validate command, 251
- Validate menu command, 257, 261
- validate method, DocBook validator, 594
- Validate WS-I Message Log File wizard, 474–476
- validateInJob method, DocBook validator, 594
- Validation
 - client-side, 204–205
 - DocBook. *See* DocBook validator
 - EJB Project, 337–338
 - JavaScript, data entry form, 236–247
 - preferences, 106, 107–108
 - with WSDL validator. *See* WSDL validator
 - WTP framework for, 586–587
 - XML, 250–251, 257–261, 656
- Validation Framework, WST, 34
- validator.css listing, 241, 490
- Validators, Web project, 140
- Variables, denoting template, 222
- Vendors
 - .NET programming and, 14
 - generating revenue, 19–22
 - history of J2EE IDEs, 13–14
 - reducing development expense, 16–18
 - supporting J2EE standards, 24
 - supporting WTP, 15
 - WTP policy on redistributing 3PC, 100–101
- Verbs (methods), REST architectural style, 425
- Verification, WTP installation, 103–104
- Views
 - database, 73
 - Model View Controller, 127
 - types of XSD editor graphical, 434–435
- Visual Studio .NET, 14

W

- W3C. *See* World Wide Web Consortium (W3C)
- W3C Note, 422, 429
- W4T Eclipse, 672–673

- WAR (Web Archive), 35
- Warnings
 - disabling when using DocBook validator, 596
 - removing for WTP SOAP namespace, 611
- WCF (Windows Communication Framework), 680
- Web
 - App Libraries category, 184
 - development of standards for, 26
 - evolving landscape of, 113–115
 - and XML preferences, 106, 108
- Web application architecture and design, 113–135
 - case study: League Planet, 133–134
 - Java application frameworks, 128–130
 - Java Web applications, 116
 - layered, 120–122
 - Model View Controller for, 123–127
 - Service-Oriented Architecture, 130–132
 - tiers or layers of, 115–116
 - Web landscape, 113–115
- Web application servers
 - defined, 29
 - vs. Web servers, 261–262
- Web applications, J2EE, 44–60
 - adding server runtime environment, 46–50
 - creating and editing JSP file, 56–57
 - creating dynamic Web project, 51–56
 - as iteration 1 of “Hello, world”, 41
 - overview of, 44–45
 - running JSP on server, 57–60
 - as scope of WTP, 23
 - summary, 57–60
- Web applications (modules)
 - creating, 141–148
 - defined, 551–552
 - libraries, 166
 - managing classpath, 175–176
 - using Web services in. *See* Web services, in Web applications
- Web Archive (WAR), 35
- Web browsers, 216–218, 255–256
- Web containers, 262
- Web content folder, 210–211
- Web Design in a Nutshell: A Desktop Quick Reference* (Niederst), 212, 230, 234
- WEB-INF/lib directory, 78, 166, 184
- web1db.sqlpage listing, 77
- Web module
 - adding to workspace, 51–56
 - creating stateless session bean, 349–352
 - dividing into multiple projects, 171–180
 - Maven, 183
- Web Module page, Dynamic Web project, 52, 54, 144–145
- Web pages
 - Web services vs., 82
 - WST tools for, 31
- Web Pages That Suck* (Flanders and Willis), 203
- Web project types, 138–160
 - creating EJB applications, 148–154
 - creating enterprise applications, 154–160
 - creating Web applications, 141–148
 - J2EE modules, 140
 - overview of, 138
 - Web projects, 139–140
- Web projects
 - adding to EAR, 167
 - creating new, 142–148
 - defined, 161
 - dynamic, 141
 - overview of, 139–140
 - static, 141
- Web projects, advanced, 160–165
 - dependent modules, 165
 - modeling developer view, 162–164
 - modules, 164–165
 - overview of, 160–161
 - resources, 164
- Web Service Client test page, 467
- Web Service Description Language (WSDL)
 - 2.0 platform, 680
 - defined, 687
 - editor for, 32

- interoperability of Web services, 471
- invoking WS-I message-log validator, 474–476
- overview of, 422–423
- REST style Web services, 426–427
- searching UDDI registry for Web services, 498–501
- top-down Web services, 435–439
- Web Service wizard generating, 86
- in Web services development, 45
- Web Service Description Language (WSDL), creating extensions, 607–644**
 - adding custom actions to editor’s design view, 627–635
 - adding extensibility elements to editor, 617–627
 - customizing editor’s design view, 614–617
 - overview of, 607–611
 - plug-in, 611
 - validation, 635–644
 - for WSDL editor, 611–614
- Web Service Explorer, 32, 100**
- Web Service Inspection Language (WSIL)**
 - overview of, 501–502
 - publishing Web services information, 502–508
 - Web Service Explorer supporting, 32
- Web Service Proxy page, Web Service Client wizard, 465–466**
- Web Service Validation Tools (WSVT), 26**
- Web service wizard**
 - defined, 32, 34
 - deploying Bottom-Up Web services, 460–462
 - deploying Top-Down Web services, 440–446
 - deploying Web services, 83–86
 - generating Java client proxy and JSP test client, 464–468
 - using JSP test client, 468–470
- Web services, 83–89, 421–508**
 - Ant tasks, 444
 - deploying, 83–86
 - discovery, 495
 - generating client proxies, 464–470
 - history of, 421–422
 - JST tools for, 36–37
 - monitoring SOAP messages, 87–88
 - overview of, 82–83
 - preferences, 106, 108–109
 - publication, 495
 - REST style, 424–427
 - REST style Web services, 426–427
 - testing for interoperability, 470–477
 - using test client, 86–87
 - WSDL, 422–423
 - WST tools for, 32–33
- Web services, Bottom-Up development, 454–464**
 - deploying, 460–464
 - developing Java service implementation, 455–460
 - overview of, 454–455
- Web services, discovering and publishing, 494–508**
 - overview of, 494–495
 - UDDI, 495–501
 - WSIL, 501–508
- Web Services Explorer**
 - defined, 496
 - publishing Web services with WSIL, 504–508
 - testing Top-Down Web services with, 451–454
 - XMethods UDDI registry in, 496–501
- Web services, in Web applications, 477–494**
 - creating servlets, 478–480
 - generating Query Web Service client, 477–478
 - importing user interface code, 480–491
 - overview of, 477
 - testing user interface, 491–494
- Web Services Interoperability Organization (WS-I)**
 - checking messages for compliance with, 471–477
 - defined, 26
 - Test Tools, 32–33
 - validator, 611
- Web services, Top-Down development, 428–454**
 - deploying, 440–446

- implementing, 446–451
 - overview of, 428–429
 - testing with Web Services Explorer, 451–454
 - WSDL, 435–439
 - XSD, 429–435
- Web Services Validation Tools (WSVT)**
Technology project, 471
- Web Standard Tools (WST)**
creating, 25–27
- Data Tools component of, 374
 - defined, 687
 - installing for WTP, 92
 - server tools and APIs as components of, 553
 - in WTP architecture, 28
- Web Standard Tools (WST), subproject, 29–34**
- Data tools, 33
 - Internet tools, 30
 - Server Tools, 29–30
 - Structured Source Editor framework, 31
 - utility components, 33–34
 - Web page tools, 31
 - Web Service tools, 32–33
 - XML tools, 31–32
- Web-Tier Application Framework Design of Designing Enterprise Applications with the J2EE™ Platform, Second Edition* (Singh), 118
- Web Tools Platform (WTP), 13–40**
- 2.0 platform, 327, 677–681
 - architecture of, 27–28
 - contributing to, 37–40
 - defined, 3
 - EJB 3.0 development with, 356–358
 - generating revenue, 19–22
 - history of, 13–15
 - JPA development with, 414–418
 - JST subproject in, 34–37
 - maintenance releases of, xix
 - reducing development expenses, 16–18
 - scope, 23–24
 - subprojects, 24–27, 29–34
 - validation framework, 586–587
- Web Tools Platform (WTP), installing, 91–105**
- build types, 92–95
 - configuring, 105–110
 - installable components, 91–92
 - JDK, 103
 - third-party content, 100–103
 - verifying installation, 103–104
 - via Update Manager, 95–98
 - via zip files, 98–100
- Web Tools Platform (WTP), quick tour, 41–89**
- installing Eclipse and WTP, 43–44
 - installing JDK, 42
 - installing WTP, 43–44
 - Iteration 1. *See* Java 2 Enterprise Edition (J2EE), creating Web applications
 - Iteration 2. *See* Scriptlets; Servlets
 - Iteration 3. *See* Database access
 - Iteration 4. *See* Web services
 - iterations of, 41–42
- webContent folder**
- creating new dynamic Web project, 145–147
 - creating new JSP, 283–284
 - dividing Web module into multiple projects, 176, 178
 - link to Management, 177–178
- webdoclet, 109**
- WebLogic application server, 18, 129, 552, 554**
- WebSphere application server, 17, 129, 670**
- WebSphere Studio Application Developer, 13–15**
- Web.xml listing, 364, 526, 645**
- Well-formed documents, 257**
- Wikipedia, 7**
- Windows Communication Framework (WCF), 680**
- Wizards, online resources for, 148**
- WORA (“Write once, run anywhere”) principle, 18**
- Workspace. *See also* Web Tools Platform (WTP), installing**
- adding Web module to, 51–56
 - configuring WTP, 105–110

- defined, 51
 - updating WTP, 104–105
 - World Wide Web Consortium (W3C)**
 - defined, 26
 - WSDL, 422
 - WSDL 1.1 binding for SOAP 1.2, 610
 - XSD, 429
 - “Write once, run anywhere” (WORA)**
 - principle, 18
 - WS-I. *See* Web Services Interoperability Organization (WS-I)**
 - WS-* specifications, 427, 680**
 - WSDL. *See* Web Service Description Language (WSDL)**
 - WSDL Binding Details page, Web Services Explorer, 498, 500**
 - WSDL validator, 635–644**
 - checking messages for WS-I compliance, 471
 - contributing custom validation rules, 640–644
 - contributing to WSDL 1.1 validation, 636–640
 - overview of, 635
 - WSIL. *See* Web Service Inspection Language (WSIL)**
 - WSIL Service Details page, Web Services Explorer, 508**
 - WST. *See* Web Standard Tools (WST)**
 - WSVT (Web Service Validation Tools), 26, 471**
 - WTP SOAP binding generation extension in plugin.xml listing, 630**
 - wtpsoap.xsd listing, 620**
 - WYSIWYG (“What You See Is What You Get”) editing, 211**
- X**
- XDoclet**
 - creating EJB applications, 150–151, 332–335
 - developing CMPs with, 394–396
 - EJB utility class, 352
 - preferences, 106, 109
 - processing Javadoc tags, 35
 - unchecking option when creating servlet, 66
 - XDoclet EJB wizard, 339–343**
 - XHTML (extensible HTML), 45**
 - XHTML™ 1.0 *The Extensible HyperText Markup Language (Second Edition)* [XHTML], 212**
 - XMethods**
 - Home page, 502–503
 - Programmatic interfaces to, 502–503
 - UDDI registry, 496–501
 - viewing WSIL document published at, 502–503
 - XML. *See* Extensible Markup Language (XML)**
 - XML: A *Primer* (St. Laurent), 248, 257**
 - XML catalog**
 - adding catalog of resources to, 654–657
 - adding single resource to, 651–654
 - contributing resources to, 648–650
 - overview of, 650–651
 - preferences, 108
 - XML Schema Description (XSD)**
 - adding new content using annotations, 607
 - developing Web services with, 45, 429–435
 - overview of, 429
 - WSL tools for Internet, 30
 - WST tools for XML, 31–32
 - XML Schema Part 0:Primer [XSD10-Part0], 429**
 - XP (extreme programming), 617**
 - XSD. *See* XML Schema Description (XSD)**
 - XSL Transformations (XSLT) *Version 1.0* [XSLT10], 248**
 - XSLT. *See* Extensible Stylesheet Language Transformations (XSLT)**
 - XSLT Reference (Nic), 248**
 - XSLT (Tidwell), 248**
- Z**
- Zend, 681**
 - Zip files, 92–93, 98–100**