

Index

Numbers/Symbols

- `{bean.value}` expression, 385, 391, 392
- `*`. (asterisk and period combination), 44
- `*` (asterisk)
 - for multimapping, 47–49
 - as the role-name, 118
- `**` wildcard, 656, 672
- `"/"` value
 - for a cookie, 27
 - sending cookies to all URLs, 26
- `/` (slash), in `url-pattern` element, 44
- `.` (period), setting the `basedir` attribute, 649
- 401 (Unauthorized) response code, 189
- 401 status code, 78
- 403 Forbidden HTTP error code, 114
- 404 error messages, 73, 74

A

- absolute URLs**
 - with `c:redirect`, 435
 - modifications required by, 133
- abstract roles**
 - in BASIC authentication, 145, 146
 - determining, 108
 - for form-based authentication, 108, 118–119

- Accept-Language request header, 554
- access restriction, 80–82, 105, 179
- access time filter, 223–226
- Account Status link, 132
- account status page, linking to, 129, 132
- action(s)**
 - advantages of subdirectories, 524
 - defining in `struts-config.xml`, 459–461
 - understanding, 458–462
 - URL design strategies for, 523–524
 - writing, 461–462
- action attribute**
 - of the `html:form` tag, 509
 - with `html:rewrite`, 569
- Action class**
 - calling, 569
 - complete listing for, 467
 - defining, 457
 - defining to handle requests, 454, 485, 496–497, 516–517, 533
 - extending, 462, 467
 - manually validating input fields, 599
 - mapping incoming `.do` addresses, 459–460, 465, 471, 491, 507, 511–512, 529, 546
 - requirements for using in Struts, 458
 - validation in, 593, 594–607

- action declarations, updating, 487, 492**
- action elements**
 - attributes of, 459–460
 - defining in `struts-config.xml`, 465
- action entries**
 - adding to `action-mappings`, 459
 - mapping incoming `.do` addresses, 455, 507
 - in `struts-config.xml`, 454
- action example, with an input attribute, 610**
- action mapping, 450, 452, 521, 535**
- Action objects**
 - creating to handle requests, 473–474
 - creating to requests, 466–467
 - execute method of, 450
 - invoking the execute method, 482
 - processing requests with, 458–481
 - sharing results beans, 453
 - in Web application, 457
- action URLs**
 - designing with subdirectories, 523–524
 - designing without subdirectories, 523
 - similar path for all, 524
- ActionErrors object**
 - placing messages in, 611
 - use of, 610
 - in the `validate` method, 610
- ActionForm argument, of `execute`, 462**
- ActionForm bean**
 - creating, 493
 - validating input fields in, 607
 - validation in, 609–612, 618, 619
- ActionForm class**
 - extending, 487
 - validating input fields in, 612
 - writing, 487–488
- ActionMapping argument, of `execute`, 462**
- ActionMessages**
 - adding messages to, 598
 - adding parametric arguments to, 620
 - adding to the `ActionErrors` collection, 613
 - constructing, 610
 - placing in request scope, 596, 597, 599
 - referring to a message in the properties file, 611
- ActionServlet, 452**
- add method, of `ActionErrors`, 610**
- addresses with subdirectories for URLs, 523**
- admin directory, protecting URLs in, 138–139**
- aliases**
 - defining for roles, 181
 - for a form bean, 487
 - for a security role name, 83
- allowed access, in declarative security, 180**
- Ant, 645**
 - benefits of, 646
 - building a Web application, 668–669
 - creating a WAR file, 675–678
 - dependencies, 669
 - directory structure for `boats` Web application, 661, 663
 - extended by `Maven`, 16
 - installing and setting up, 646–647
 - integrating an IDE with, 17
 - overhead of learning, 16
 - using for deployment, 16–17
 - written in Java programming language, 16
- Ant 1.7.0, bundled as a ZIP file, 646**
- Ant implementation, downloading, 646**
- Ant library, 651**
- Ant manual, bookmarking, 647**
- Ant projects**
 - building, 670
 - creating, 648–652
 - defining, 648–650
 - writing a simple, 661–668
- Ant targets, running, 651–652**
- Ant Tasks menu option, 647**
- Ant tasks, reviewing common, 652–661**
- ANT_HOME variable, 647**
- Apache Ant. *See* Ant**
- Apache Foundation, Ant developed by, 16**
- Apache Maven, 16–17, 646**
- Apache Struts framework. *See* Struts**
- apparent execution flow, through the Struts framework, 450**
- app-blank Web application**
 - copying and renaming, 14
 - downloading, 5, 10
 - `web.xml` file included with, 37
 - `web.xml` of, 7
- app-blank.zip, downloading, 18**
- append attribute, with `<echo>`, 653**
- application event listeners, 93–94, 268**
- application scope, 453, 487**
- application variable, in a JSP page, 26**
- applications**
 - internationalizing, 554
 - moving complex with a minimum of effort, 3
- application-wide data, creating, 270**
- arg element, 631**
- arg0 attribute, 545**

argN attribute, providing, 553**arrays**

- forEach custom tag iterating over, 399
- LoopTest, 400

asterisk and period combination (*.), 44**attackers, stealing network data in transit, 105, 179****attribute(s)**

- assigning to tags, 357–359
- cleaning up, 297
- of the forward element in struts-config.xml, 461

attribute changes, 322–323**attribute directive, 374****attribute element, nested elements, 358****attribute names**

- case sensitivity of, 359
- comparing to stored names, 278, 306

<attribute> subelement, 678**attributeAdded method, 277, 322, 323**

- implementing, 306, 307
- triggering, 277

attributeRemoved method, 322

- of HttpSessionAttributeListener, 297–298
- implementing, 306, 307
- triggering, 277

attributeReplaced method, 322

- implementing, 279, 306, 307
- obtaining name and value of modified attribute, 279
- triggering, 277

auth-constraint subelement, of security-constraint, 38, 80, 81–82, 108, 115, 117–118, 146**authenticated users**

- allowing all to access a resource, 118
- protecting from directly accessing the login page, 112

authentication, 105, 179

- example of form-based, 122–143
- form-based, 106–121, 143–144
- mechanisms, 78
- users failing, 338

authentication method, designating, 78–79**auth-method subelement**

- of FORM, 107
- of login-config, 78–79
- value of FORM for, 110

authorization attempt

- result of a failed, 191, 194
- result of successful, 191, 194

Authorization header

- populating, 78
- remembering authenticated users, 144

auto-deploy directory. *See also* hot-deploy directory

- of Tomcat, 23
- in WebLogic, 13

automatic validation, 594, 624

- compared to manual, 624
- example of, 633–643
- steps in using, 627–633

automatic validator, turning on, 628**B****bad-address condition, 473, 477, 478, 501****bad-address3.jsp page, 498, 499****bad-address.jsp page, 475****bad-password condition, 473, 477, 478**

- execute method returning, 501
- response for, 503

bad-password.jsp page, 475, 498, 499**banned sites, obtaining a list of, 227****BannedAccessFilter class, 227–228, 230–232****base64 encoding, 189, 190****BASE64Decoder class, 189, 190****basedir attribute**

- of <jar>, 676
- in a <project> element, 649

BASIC authentication

- for the auth-method subelement, 145
- disadvantages of, 144
- easier to set up, 144
- example, 147–156
- mechanism, 78
- steps involved for, 189
- telling the server you are using, 144, 145–146

BASIC security, 144–145**BEA WebLogic server, 13****bean(s)**

- accessing input fields from, 596
- associating with forms, 450, 509
- populating, 443
- representing incoming form data, 490
- system finding or instantiating, 518

bean properties

- displaying in JSP pages, 488–489
- for each incoming request parameter, 484
- names staying in sync with request parameters, 504
- outputting, 507

bean properties (*continued*)

- updating and creating, 427
- used for the form fields, 518
- value of null or an empty string, 510

bean Struts tags, 562**bean tag library**

- importing, 490–491, 498
- loading, 544

bean: tags, in Struts, 443**bean:message tag**

- bundle attribute, 543
- common attributes for, 545
- displaying title, head, and registration labels, 551
- not filtering for special HTML characters, 553
- outputting messages, 544–545
- providing a value for the key attribute, 552
- retrieving parameterized messages, 553
- from the Struts tag library, 539
- taking messages from the properties file, 550

BeanRegisterAction, complete code for, 497**bean:write tag**

- common attributes of, 489
- compared to bean:message tag, 545
- displaying bean properties, 457, 488–489, 498
- displaying each message, 597
- displaying registration information, 550–551
- filtering special HTML characters, 489
- in an output page, 600
- outputting bean properties, 455, 490, 519, 522

bin directory, 647**boats Web application, files and directory structure for, 661, 662****boats1.xml build file, 664, 667–668****boats2.xml build file, 670, 673–675****boats3.xml build file, 679, 681–682****Body section, of a Tiles layout, 570, 571****body sections, for a Web site, 572, 574–576****body-content element, value of scriptless for, 363****body-content subelement, of tag, 350–351****body-index.jsp section, 572****bookmarking Struts documentation, 447****browser(s)**

- cookies maintained by, 26
- setting language preferences in, 554–555
- supporting content encoding, 245

browser-based deployment wizard, in WebLogic, 13**build directory, 664****build directory structure, 671****build files**

- in Ant projects, 649
- example, 645
- general-purpose, 645
- specifying in Ant, 651–652

build target, in boats2.xml, 670, 672–675**Built-On date, 654****bundle attribute**

- for bean:message tag, 545
- specifying in the bean:message tag, 543

business logic, handling exceptions, 437**business logic class, 494, 495–496****business plan protected page, available only to executives, 154–156****ByteArrayOutputStream, 245–246****C****"c", as the prefix for JSTL core tag library, 421****CA, signing a server certificate as, 167****CA certificate**

- exporting, 170–174
- importing as a Trusted Root Certificate, 170

case sensitivity, of XML elements, 36**catch tag, 437–438****catch.jsp page, 437–438****categories of users, storing the HttpSession object, 188****Caucho Resin server, 13–14****c:catch tag, 437–438****c:choose tag, 425–427****centralized file-based configuration, of Struts, 442****centralized updates, 540****ceo role, declaring, 279****Certificate Import Wizard, 161, 162–163, 171–173****certificate information, viewing, 159, 161****certificate signing request. See CSR****certificate store, 172****certificate warning dialog box, 159****Certificates dialog box, 174****certtool script, in JSTK, 167****c:forEach tag, 422–424****c:forTokens tag, 422–424****ChangeCompanyName servlet, code listing, 283–284****ChangeDailySpecial servlet, 334, 336–337****ChangedCompanyNameListener class, 279, 280****ChangedDailySpecialListener, 334–335**

- changes, centralized with Tiles Definitions, 583**
- character encoding**
 - configuring, 93
 - used by JSP pages, 88
- choose tag, 425–427**
- c:if tag, 424–425**
- c:import tag, 430–433, 434**
- class files**
 - compiling source code to, 658
 - destination directory for, 659
 - placing in WEB-INF/classes, 6
 - reverse engineering, 677
- ClassCastException, 412**
- classic tag API, 348**
- classic version, of Java-based custom tags, 371–372**
- CLASSPATH**
 - automatically used by Ant, 659
 - of the properties file, 543
 - providing explicitly in an Ant project, 661
 - on UNIX/Linux compatible systems, 21
 - updating, 447
 - used by Ant for the operating environment, 658
 - in Windows, 20
- classpath attribute, of <javac>, 658**
- classpath property, referencing, 660**
- <classpath> element, 660–661, 665**
- clean target**
 - in boats1.xml, 665
 - in boats2.xml, 670
 - removing all existing .class files, 666
- client certificates**
 - security based on, 144
 - users authenticating, 120
- client dependency, on the container, 101**
- CLIENT-CERT authentication mechanism, 79**
- CLIENT-CERT value, 120**
- clients, served by different machines, 95**
- client-side validation, 624–625**
- closing tag, with the body-content of empty, 356**
- clustered environment, developing for, 95–97**
- coin flips, simulating a random number of, 387**
- CoinBean.java class, 387**
- colors**
 - collecting as an input form, 612
 - user selecting from an input page, 599
- colorUtil.randomColor method, 299, 304**
- commented-out examples, removing from struts-config.xml, 456**
- commonly used data**
 - initializing, 271–277
 - monitoring changes to, 278–288
- company home page, 290, 293, 296**
- company information page**
 - with new custom tags, 290, 293–294
 - reworking to use custom tags, 296
- company name**
 - changing during execution of a Web application, 278
 - updating the former, 278
- company name listeners, packaging, 290–296**
- CompanyNameTag code listing, 290**
- compile target**
 - in boats1.xml, 665
 - in boats2.xml, 670
 - regenerating .class files, 666
- compress attributes, of <jar>, 676**
- compression filter, 245–251**
- CompressionFilter class, 245, 247–249**
- conditional operations, 425**
- CONFIDENTIAL value**
 - mandating the use of SSL, 82
 - for transport-guarantee, 119
- configuration files**
 - making changes in, 540
 - used in Tiles Definitions, 583
- confirmation page, for registration, 522**
- confirm-registration.jsp page, 475, 498, 500, 550–551**
- ContactFormBean, 513–515**
 - Action class using, 511
 - code listing, 530–532
- container-managed security, 82**
- containers. See server(s)**
- content, 430**
- Content-Encoding response header, 245**
- Context element, 12**
- Context entry, adding to tomcat_dir/conf/server.xml, 11**
- context sharing, enabling on Tomcat, 27**
- contextDestroyed method, 270, 271**
- contextInitialized method**
 - implementing, 270, 271
 - of ServletContextListener, 96
- context-param element, 38, 63–64, 269, 271**
- ContextReporter listener, 94**
- control flow. See flow of control**
- controller section, supporting an existing application, 628**

cookies, 26

- accessing in Struts, 472
- disabling, 305
- drawback to data sharing with, 26
- sharing information among Web applications, 25
- tracking by reserved session-level, 78

<copy> target, of boats2.xml, 670, 671–672

<copy> task, in Ant, 656–658

core library of JSTL, 419

core tasks

- assigning to Ant targets, 648
- online documentation for, 652
- for targets provided by the Ant library, 651

c:otherwise tag, 425–427

count attribute, of a for loop tag, 387

counters, incrementing, 298

c:out tag, 421–422

- advantages of, 421
- as an alternative to JSP EL, 421
- displaying bean properties, 488

c:param tag, 433–435

c:redirect tag, 435–436

credit card number

- protecting in transit, 137
- validating, 632, 636

creditCard validation rule, 632, 636

c:remove tag, 428–430

crossContext attribute, in Tomcat, 27

cross-site scripting (CSS) attacks, 421

- information on, 545
- removing the risk of, 620

csajsp-taglib-adv.tld file, 381, 383, 387, 390, 393, 397, 400, 402–403, 407, 408–409, 413, 416

csajsp-taglib.tld, 359, 360, 364, 365–366, 369

c:set tag, 427–430

CSR (certificate signing request)

- generating, 168–169
- submitting to commercial CAs, 167

c:url tag, 433–435

currentSessionCount, 298

custom error messages, parameterizing, 620

custom table-formatting tag, listing records, 393

custom tags. *See also* tag(s)

- conditionally outputting tag bodies, 368
- creating complex presentation logic, 348
- creating using tag files, 371
- handling nested, 410–412

custom URLs, 44–50

c:when tag, 425–427

D**daily special**

- order count, resetting, 334–339
- tracking orders for, 326

daily special servlet, 229

DailySpecialRegistrar listener, 326–328

DailySpecialWatcher listener, 326, 328–330

data

- initializing commonly used, 271–277
- monitoring changes to commonly used, 278–288
- sharing among Web applications, 25–32
- storing in an easily accessible location, 270

database

- closing connections, 270
- cookie value as a key into, 26

DDE.jsp, code listing, 76

debug attribute, of <javac>, 659

debug tag

- improved, 407–410
- limitation to, 404

debug.jsp page, 368, 369–371, 407

DebugTag class, 368–369, 407

declarative security, 106

- advantages to the developer, 188
- levels of access for each resource, 180

decodeBuffer method, 189

default attribute, in an Ant <project> element, 649

default files, accessing in WEB-INF/web.xml file, 6

default properties file, 554

default target, in the <project> element in Ant, 652

default URLs

- initialization parameters not available, 57
- turning off, 52

defaultexcludes attribute, of <delete>, 655

definition attribute, in tiles:insert, 586

definition element, in tiles-def.xml, 585

definition file, 582

Delete Account page, accessible only to administrator role, 138

<delete> task

- <fileset> clarifying which files to delete, 655
- in Ant, 654–655
- more information on, 656

delims attribute, of c:forTokens, 422

denied access, in declarative security, 180

dependencies, among targets, 669

depends attribute, of <target>, 669

- deployers**
 - customizing filters, 222
 - customizing the behavior of servlets or JSP pages, 221
 - deployment descriptor**
 - purpose of, 35–36
 - for a Web application, 123–127, 149–151
 - deployment descriptor file.** *See* web.xml file
 - derived layout definitions, specifying, 588**
 - derived layouts, creating, 585–586**
 - description attribute, in the <target> element, 650**
 - description element, 39, 85, 97**
 - description subelement**
 - of filter, 207
 - of tag, 350
 - <description> subelement, in an Ant <project> element, 649**
 - deserialization, of session objects, 268**
 - design strategies, for action URLs, 523–524**
 - destdir attribute, of <javac>, 659**
 - destfile attribute, of <jar>, 676**
 - developers, customizing servlets or JSP pages, 221**
 - development directory, keeping separate from deployment, 9**
 - development directory structure, 10**
 - development environments**
 - explicit support for servlets and JSP, 84
 - servlet- and JSP-savvy, 16
 - DIGEST authentication, 78, 189**
 - DIGEST security, 144**
 - dir attribute**
 - of <delete>, 654
 - of <fileset>, 655, 676
 - of <mkdir>, 654
 - direct access, restricting to the login page, 112–114**
 - directories**
 - copying in Ant, 656–658
 - creating manually, 10
 - directory mappings, preferred over extension mappings, 50**
 - directory structure**
 - for the compile and copy targets, 670
 - creating for Web applications, 14
 - dispatcher deployment descriptor element, 251**
 - dispatcher elements**
 - specifying, 254
 - values of, 252–253
 - dispatcher subelement, of filter-mapping, 208**
 - display-name element, 39, 84–85**
 - display-name subelement**
 - of filter, 207
 - of security-constraint, 80, 82, 115, 118, 146
 - dist directory, 664**
 - dist/boats.war file, 680**
 - distributable element, 39, 97**
 - distribution directory, creating for a WAR file, 679**
 - .do addresses, mapping incoming, 455, 459–460, 465, 471, 491, 507, 511–512, 529, 546**
 - .do extension, 452**
 - .do suffix, 509**
 - docBase attribute, of Context, 12**
 - DOCTYPE declaration, 37**
 - Document Type Definition (DTD), 37**
 - documentation**
 - improving by dividing tasks, 669
 - JSTL tag library, 420
 - online for Apache Ant, 645
 - of Struts, 444–445
 - doFilter method**
 - of an associated filter, 68–69
 - executing, 205
 - filtering behavior in, 204, 206, 210, 218, 223, 228, 254
 - passing response wrapper to, 234
 - doGet method, 316**
 - Domain Name System (DNS) servers, preventing attackers from hacking, 157**
 - domain names, banning from your site, 227**
 - doTag method, 349**
 - of DebugTag.java, 368
 - generating <TABLE> and start tags, 364
 - placing an attribute of tag body scope, 399
 - DSTAMP property, in Ant, 653**
 - DTD (Document Type Definition), 37**
 - DumbDeveloperException class, 75–76**
 - dynamic keys, 552**
 - dynamic values, assigning to tag attributes, 385–387**
- ## E
- e-boats application, 570–581**
 - file structure of, 578
 - main pages for, 577–578

- e-boats application** (*continued*)
 - results for, 579–582
 - reworking with Tiles Definitions, 586–590
 - <echo> task**, in Ant, 652–653
 - EchoServer program**, 165
 - editor**, setting CLASSPATH in, 447
 - EJB reference**, 98
 - ejb-link element**, path name specified, 98–99
 - ejb-local-ref element**, 40, 99
 - ejb-ref element**, 40, 98–99
 - EL**. *See* JSP EL
 - el-ignored subelement**, of `jsp-property-group`, 87–88
 - else tag**, 412
 - ElseTag class**, 412, 414–415
 - e-mail address**, registration form with an illegal, 502
 - email validation rule**
 - validating against, 636
 - in `validation-rules.xml`, 632
 - empty auth-constraint element**, 118
 - empty value**, for `body-content`, 350
 - encodeURL method**, 433
 - encoding subelement**, of locale-encoding-mapping elements, 93
 - encryption algorithm**, 196
 - encryption key**, 196
 - end users**, customizing servlets or JSP pages, 221
 - English**, internationalizing for, 555–557
 - enterprise bean**, local home interface of, 99
 - enterprise-level Web applications**, clustered environment for, 95
 - env-entry element**, 39, 98
 - env-entry-type element**, 98
 - environment entry**, declaring a Web application's, 98
 - environment variables**, setting for Apache Ant, 647
 - error(s)**, designating pages to handle, 72–77
 - error code**, mapping back to the input form, 596
 - error conditions**, mapping, 525, 527
 - error messages**
 - adding prompts or messages to substitute into, 628
 - adding to ActionMessages, 596–597
 - creating and displaying, 527–528
 - customizing, 519
 - determining arguments for, 631
 - determining names of, 631
 - passing to JSP pages, 516
 - storing, 521
 - storing and displaying, 525
 - ERROR value**, of dispatcher, 252–253
 - error-code element**, 73–74
 - error-code subelement**, 72
 - error-page element**, 38, 72, 138, 152
 - escapeXml attribute**, of `c:out`, 421
 - event listener class**, 39
 - event listeners**
 - declaring, 269
 - packaging with tag libraries, 288–289
 - registering, 93
 - responding to Web application life-cycle events, 268
 - steps for using, 268–269
 - events deployment descriptor**, 339–345
 - exact-match patterns**, 45–47
 - Examine Certificate button**, in Firefox, 159
 - /examples**, not using in Tomcat, 12
 - exception class**, 75
 - exceptions**
 - catching explicitly in code, 75
 - handling with business logic, 437
 - exception-type element**, 75–77
 - exception-type subelement**, of error-page, 72
 - <exclude> element**, in `fileset`, 672
 - <exclude> subelement**, of `<fileset>`, 656
 - excludes attribute**, of `<jar>`, 676
 - execute method**
 - of the Action object, 453–454, 482
 - forwarding response back to the input page, 525
 - overriding, 462, 467
 - of `SignupAction2`, 535
 - using form beans to access incoming request parameters, 472
 - execution flow**, through the Struts framework, 450
 - executive role**, logging in as a user in, 154–155
 - expression language functions**, 404–406
 - extends attribute**, in the definition element, 585
 - extension mappings**, overriding standard, 85
 - extension subelement**, of mime-mapping, 85
 - external resource**, used with a resource factory, 98
- F**
- false**, specifying for the message attribute, 611
 - fields**, defining in `validation.xml`, 630–632
 - file attribute**
 - of `<copy>`, 657

- of <delete>, 654
- of <fileset>, 655
- file types, locations for various, 5–8**
- file-based configuration, of Struts, 442**
- files, copying in Ant, 656–658**
- <fileset> element, selecting files or directories to copy, 658**
- <fileset> subelement, specifying multiple files or directories, 654**
- <fileset> type, 655–656**
 - clarifying files to copy, 671
 - explicitly selecting files and directories, 677
 - information on, 656
 - selecting all .class files, 665
 - using with <copy> task, 657–658
- filter(s), 203**
 - accessing servlet context from, 217
 - creating basic, 204
 - declaring, 68–71
 - intercepting requests, 69
 - options for, 203
 - registering with servlets and JSP pages, 204, 207–208
 - wrapping response object, 69
- filter attribute, of bean:write, 489**
- filter element, 38, 70**
 - registering a filtering object, 207
 - subelements of, 207
- filter initialization parameters, 221–222**
- Filter interface**
 - class implementing, 210, 218, 223, 227
 - creating a class implementing, 204, 205–206
 - doFilter method of, 206
- FilterChain object**
 - calling doFilter method of, 204, 206, 210, 218, 223, 226
 - conditional call to the doFilter method of, 228
 - of doFilter, 205
- filter-class subelement, of filter, 207**
- FilterConfig class**
 - getInitParameter method of, 222
 - getServletContext method, 217
- FilterConfig object**
 - storing, 217
 - using init to store, 205–206
- filterhtml custom tag, 381**
- filtering, applying to actions, 524**
- filter-mapping element, 38, 70**
 - associating the filter with the daily special servlet, 210
 - placing the dispatcher element inside, 251
 - specifying URLs for filtering, 207
 - subelements of, 208
 - url-pattern, 223
 - using a servlet-name of LongServlet, 246
- filter-name subelement**
 - of filter, 207
 - of filter-mapping, 208
- financial plan protected page, 152–154**
- findAncestorWithClass method, 411, 413**
- Firefox**
 - certificate information presented on, 160
 - changing the language setting, 555
 - disabling cookies, 305
 - disabling JavaScript in, 641
 - new certificate page for, 201
- flatten attribute, of <copy>, 657**
- flexibility, increased with Tiles, 558**
- flow of control**
 - implementing for Struts, 450–454
 - simplified Struts, 463, 464
 - in the Struts framework, 454–458, 504, 525
 - Struts updates for bean use, 482–483
- font sizes, collecting on an input form, 612**
- fonts, user selecting, 599**
- footer, specifying for an entire group of JSP pages, 89**
- footer attribute, of html:messages, 598**
- Footer section, of a Tiles layout, 570, 571**
- for loop**
 - c:forEach as a counter-based, 422
 - iterating through an array of strings, 398
- for loop tag, 387**
- forbidden.jsp page, 138, 139**
- forchoose.jsp page, 426–427**
- forEach tag, 399, 422–424**
- ForEachTag class, 400, 401–402**
- forif.jsp page, 424–425**
- for.jsp page, 422–424**
- fork attribute value, 659**
- form(s)**
 - associating beans with, 509
 - declaring using html:form, 509
 - example of prepopulating, 511–522
 - improving filtering and security settings for, 524
 - prepopulating, 510–511
 - redisplaying, 443, 525–528
 - redisplaying example, 528–537
 - simplifying organization and editing of, 524
 - submitting data to a URL, 452

form and results beans, example using,
490–503

FORM authentication mechanism, 78

form bean(s), 481

- automatically populated with request data, 490
- critical role in the Struts framework, 486
- declaring, 455, 461, 484, 486–487, 491, 507, 512–513, 529–530
- defining, 454, 456, 472, 484, 486–487, 493–494, 508, 513–515, 530–532, 548–549
- extending ActionForm class, 490
- extending ValidatorForm, 627, 632, 638–639
- handling request parameters with, 481–503
- messages held in, 528
- populating, 443
- representing input data, 485
- understanding, 486–488
- validating in, 593, 607–623
- warning message stored inside, 533
- writing, 487–488

form bean template, supporting error messages, 527, 528

form element, in validation.xml, 629–630

form field validation, in Struts, 443–444

form field values, derived from bean properties, 534

form fields

- bean properties used for, 518
- validating, 593

form submissions, protecting, 116

<format> subelement, 653

FormatFormBean

- code listing, 599, 601–602, 613–615
- modified for parameterized messages, 621–622
- with a validate method, 613

form-based authentication, 106–121, 143–144

- creating your own, 188
- disadvantages, 144
- example of, 122–143
- not guaranteed for session tracking, 107

form-based security, 107–108

form-bean entry

- adding to the form-beans element, 486
- declaring with name and type attributes, 547
- in the form-beans section, 484
- in struts-config.xml, 454, 491

form-beans element, 486

form-beans section, of struts-config.xml, 506

FormerCompanyNameTag code listing, 291

form-login-config subelement, 78, 79, 110

formset element, in validation.xml, 629

form-validation element, in validation.xml, 629

ForTag.java class, 387, 388

forTokens tag, 422–424

forward attributes, common, 461

forward elements

- attributes of, 461
- defining in struts-config.xml, 596
- mapping return conditions, 453
- in struts-config.xml, 604

forward entries

- adding to the action element in struts-config.xml, 460
- corresponding to missing data, 527
- mapping return conditions to JSP pages, 455, 507
- replacing with a single entry in global-forwards, 479
- in struts-config.xml, 454, 458, 465, 471

forward method, of the RequestDispatcher class, 435

FORWARD value, of dispatcher, 252

fragmentation, in the Struts community, 445

French, internationalizing for, 555–557

function-class element, 406

function-signature element, 406

G

GET request

- never using with password fields, 111
- in WebClient, 164

getAttributeName method, 358

getInitParameter method

- of FilterConfig, 206, 207
- of ServletConfig, 63
- of ServletContext, 63

getJspBody() .invoke(null) method, 362, 368, 380

getName method, of the

- ServletRequestAttributeEvent, 323

getOutputStream method, overriding, 234, 235

getParent method, 411, 413

getRecentRecords method, 400

getRemoteHost method, of ServletRequest, 228

getRemoteUser method

- in HttpServletRequest, 181
- returning username of the currently logged-in user, 113

getRequestRecords method, 316

- getScheme method, of ServletRequest interface, 195**
- getServletContext method**
 - of a servlet, 26
 - of ServletRequestEvent class, 314
- getServletRequest method, 314**
- gettable bean properties, 488, 493**
- getter methods**
 - calling on UserFormBean, 496
 - for results beans, 494
- getUserPrincipal method, in HttpServletRequest, 181**
- getWriter, overriding, 234**
- global properties, declaring, 649–650**
- global-forwards section, 479, 512**
- GZIPOutputStream, 246**
- gzipped content, browsers handling, 245**

H

- handler element, 101**
- handler-class element, 101**
- handler-name element, 101**
- hardware load balancer, 95**
- header attribute, of html:messages, 598**
- Header section, of a Tiles layout, 570, 571**
- heading element, defining a tag for, 364**
- heading tag, using tag files, 376–377**
- heading-1.jsp, 364, 366–367**
- headings-2.jsp file, 376–377**
- HeadingTag, 364–365**
- heading.tag file, 376**
- home page, for a Web application, 122–123, 147–148**
- hot-deploy directory, 10**
- hot-deployment, 10**
- hot-dot-com.com, 122**
- HTML**
 - generated from invoking/ustm, 92
 - importing a snippet of, 431
 - in template files, 561
- HTML code, custom tag filtering out, 381**
- HTML forms**
 - creating for user input, 455
 - declarative security using, 106
- HTML header elements, not repeating in layout pieces, 562**
- HTML page**
 - forwarding, 253
 - requesting a JSP page, 299, 303
- HTML snippets, importing, 431**

- html Struts tags, 562**
- html: tags, building the Struts input form, 450**
- HTML tags, provided by Struts, 443**
- HtmlFilterTag tag handler class, 381–382**
- html:form element**
 - associating a form with a bean, 455
 - building a FORM element, 521, 535
 - building an HTML form, 534
 - declaring a form, 508, 509
 - for the main form, 510
 - results yielded by, 509
- html:form tag, 508, 518**
 - adding an onsubmit attribute, 633
 - building form elements, 599
 - creating a form, 452
- html:javascript tag, adding to JSP page, 633**
- html:messages tag**
 - common attributes of, 597–598
 - displaying error messages, 599, 641
 - in input form, 609, 611–612
 - on the input page, 627
 - outputting error messages in input form, 596, 597–598
- html:rewrite tag, 569, 572**
- html:tags**
 - as key to the redisplay process, 525
 - prepopulating forms, 510
 - using, 508–510
- html:text element**
 - building an HTML form, 534
 - declaring input fields of a form, 508
 - declaring the input fields of a form, 510
 - gathering user data, 455
- html:text tag, 508**
 - building form elements, 599
 - building input elements, 518
 - creating a form, 452
 - for text fields, 511
- HTTP authentication**
 - native, 189
 - standard, 78
- HTTP BASIC security, 144–145**
- HTTP error codes, 72**
- HTTP request headers, accessing in Struts, 472**
- HTTP version, selecting for WebClient, 165**
- http-method element, 80**
- HTTPS (HTTP over SSL), servers using, 79**
- https connections, 120**

HttpServletRequest object

hybrid security methods provided in, 181
 special-purpose attributes of, 72
 in Struts, 472

HttpServletRequestWrapper class, 234**HttpSession object(s)**

invalidating, 112
 objects stored in implementing Serializable, 96
 session tracking data stored in per-user, 297
 session tracking provided, 433
 sharing among machines in a cluster, 95
 storing minimal information in, 96

HttpSessionAttributeListener interface, 297, 306, 307, 326**HttpSessionBindingEvent class**

as an argument, 306
 methods of, 306

HttpSessionEvent, getSession method, 297**HttpSessionListener interface, 297–298****https://localhost/, accessing, 159****I****I18N application, creating with Struts, 554****icon element, 39, 84****icon subelement, of filter, 207****id attribute, of html:messages, 597, 598****IDE**

setting CLASSPATH in, 447
 using in combination with Ant, 17

IDE-specific deployment features, 16**if statement, getJspBody().invoke(null) invocation, 368****if tag, 412, 424–425****IfTag class, 412, 413****if-test.jsp file, 413, 416–417****if-then-else tag, 412–417****illegal password, registration form with, 503****IllegalArgumentException, container throwing, 96****images, using on multiple pages, 570****/images directory, 570****import statements, adding Struts-specific, 462, 466–467****import tag, 430–433, 434****import.jsp page, 431, 432, 433****include directive**

at the beginning of each JSP page, 89
 using static inclusion, 430

INCLUDE value, of dispatcher, 252**<include> element, in fileset, 672****<include> subelement, of <fileset>, 656****include-coda subelement, of jsp-property-group, 89****includeEmptyDirs attribute**

of <copy>, 657
 of <delete>, 655

include-prepare subelement, of jsp-property-group, 89**includes attribute, of <jar>, 676****incoming user data, form beans representing, 486****index attribute, of <jar>, 677****inheritance, Tiles Definitions as a loose form of, 583****init method**

executing for filters, 205–206
 illegal in a JSP declaration, 61

initial registration page, 521**InitialCompanyNameListener, 271, 272–274**

augmented, 290, 294–295

initialization, in JSP, 57**initialization parameters**

assigning to servlets, 56–60
 common gotchas, 57
 defining, 222
 missing result, 222
 parsing, 222
 placing slow-loading, 68
 providing for commonly-used data, 272
 providing for event listeners, 269, 271
 reading, 222
 in servlets accessed by default URLs, 57
 supplying application-wide, 63–64

InitPage.jsp, 61, 62**init-param element, 56, 101****init-param subelement**

defining startTime and endTime, 223
 of filter, 207, 222, 228

InitServlet

accessing with a custom URL, 60
 accessing with the default URL, 60
 using init method, 58–59

input attribute

of the action element, 460
 of action in struts-config.xml, 609, 612
 adding to the action element in
 struts-config.xml, 613

input data, validation of, 593, 625

input elements, declaring, 511**input fields**

- declaring using `html:text`, 510
- prepopulating, 511
- setting up validation rules, 630–632
- specifying validation rules for, 636
- targeting messages for particular, 598
- validating in `ActionForm`, 612

input form(s)

- advantages of subdirectories, 524
- code listing for, 599, 600–601
- collecting font sizes and colors, 599
- creating, 457, 474–475, 507
- `html:messages` in, 611–612
- listing of complete, 518–519
- listing the address of, 627
- mapping error condition back to, 527
- placing, 468
- prepopulating and redisplaying, 504–537
- redisplaying with correctly entered values and missing values flagged, 528
- in Struts, 450
- using the form bean, 513

input page

- for the amazing Un-Divet tool, 639
- for `chooseFormat2`, 613, 616–617
- redisplayed, 525
- redisplayed with error messages, 535, 536, 537
- for a sample resume, 605, 606
- for the sample resume builder, 618
- specifying in `struts-config.xml`, 610–611

input values, automatically checking, 594**Install Certificate button, 159, 161****instance variables, avoiding for shared data, 95–96****Integer.parseInt call, enclosing within a try/catch block, 222****INTEGRAL value**

- mandating the use of SSL, 82
- for transport-guarantee, 119

interfaces

- implementing for event listeners, 268
- layout template as, 561

internationalization (I18N), 540**Internet Explorer**

- Certificate Import Wizard, 162–163, 171–173
- certificate information presented by, 161
- certificate warning dialog box, 159, 161, 163
- changing the language setting, 555

Content tab for Internet Options in, 170

disabling cookies, 305

disabling JavaScript in, 641

invalidate, calling, 112**investing directory, of hotdotcom Web application, 129–132****invoke method, 380****invoker servlet**

- disabling, 52–56, 58, 140, 204, 209, 210, 218
- providing a default URL for servlets, 121
- Tomcat 5 turning off by default, 55
- turning off for form-based authentication, 108, 120–121
- turning off in BASIC authentication, 145

isELIgnored attribute, of the page directive, 87–88**isMissing method, of FormatFormBean, 599****isSecure method, of ServletRequest interface, 195****isUserInRole method, of HttpServletRequest, 83, 181****is-xml subelement, of jsp-property-group, 88–89****iteration, provided by c:forEach, 422****J****j_security_check, as a “magic” name, 111****J2EE elements, 97–102****Jakarta Taglibs project, 420****jar command, creating a WAR file, 17****JAR file(s)**

- copying from `struts-blank`, 449
- giving a `.war` file name extension, 676
- location of, 4
- moving to a common directory in Ant, 660
- placing in `WEB-INF/lib`, 7, 14
- running `WebClient` as, 164
- tag files bundled in, 8
- with a `.war` extension, 17

jar options, with WAR files, 18**jar utility**

- creating `MANIFEST.MF`, 8
- listing the contents of a WAR file, 680

<jar> task

- common attributes of, 676
- creating a distribution directory, 680
- creating a JAR file, 676
- creating WAR files, 675
- simple example of, 677

- Java, Ant built from, 646**
- Java class files**
 - location of, 4
 - placing into WEB-INF/classes, 14
- Java code**
 - avoiding inside JSP pages, 379
 - in JSP pages, 67, 347
- Java compilation process, 35**
- Java Developer's Kit (JDK), 167**
- .java files, compiling, 665**
- Java Naming and Directory Interface name.**
 - See JNDI name
- Java Security Took Kit. See JSTK**
- Java system properties, accessing in Ant, 650**
- Java Virtual Machines, 95**
- JAVA_HOME variable, 647**
- Java-based custom tags, 371, 380**
- javac program, command-line options**
 - available in, 658
- <javac> task, 658–661, 665**
- JavaScript**
 - client-side validation with, 625, 633
 - disabling in the browser, 641
 - enabling validation, 627, 639
- JavaServer Faces (JSF), compared to Struts, 445**
- javax.servlet.Filter interface, 38, 205–206**
- javax.servlet.http.*, importing, 462**
- javax.servlet.jsp.tagext package, 348**
- javax.xml.rpc.Service, 100**
- JAX-RPC interface name, specifying the fully qualified, 100**
- jaxrpc-mapping-file element, 100**
- JBoss server, 13**
- JDK, on a Windows platform, 167**
- JNDI name, 97**
 - looking up the Web Service, 100
 - of a message destination reference, 101
- job-openings.html**
 - accessing, 254
 - located in the secure directory, 258
- JSP, automatically populating a JavaBean, 443**
- JSP 2.0 EL (Expression Language)**
 - displaying bean properties, 488
 - in JSP pages, 441
- JSP actions, 419**
- JSP custom tag libraries, Struts as, 442**
- JSP Document, 88**
- JSP EL (Expression Language)**
 - attribute value as, 358
 - functions, 405–406
 - implementing complex presentation logic, 419
 - invoking a method, 406
 - turning off and treating as regular text, 87
- JSP files**
 - adding attribute support, 359
 - complex dynamic attribute values, 392
 - dynamic attribute values, 386–387
 - making use of a tag, 352
 - supporting tag bodies, 363
- JSP initialization parameters, 60–63**
- JSP page(s)**
 - accessing before LoadInitServlet, 67
 - accessing in myWebApp, 12
 - bean properties, displaying in, 488–489
 - blocking access to, 205
 - business presentation logic, presenting output of, 379
 - companyName, retrieving a previously stored, 64, 65, 66
 - configuring, 68, 86–92
 - custom URL pattern, assigning the original URL as, 61
 - daily specials, displaying current, 334, 335
 - dependencies, 188
 - direct access, ensuring no, 80
 - dynamic content, generating inside, 347
 - filter, associated with, 70
 - form displaying the current company name, 279, 283–284
 - htmlfilter tag, using, 381, 384–385
 - initialization behavior, customizing, 221
 - initialization parameters, reading, 51
 - initializing and preloading, 56–68
 - jspInit, using, 57
 - layout, creating to populate, 560, 563, 567–568, 577–578
 - layout pieces, creating to define, 560, 562, 565–566, 572–576, 587
 - layout pieces, defining, 584
 - layouts, creating to use, 584, 586, 589–590
 - locating, 5–6
 - messages, displaying in, 542
 - messages, outputting in, 544–545
 - modifying the output of, 205
 - my WebApp, placing in, 11
 - names, assigning to, 42–44
 - names, declaring for, 50–51
 - naming, 50–52
 - never accessing directly, 117

- order count, summarizing current, 326, 332
 - parametric replacements, providing in, 553
 - process of initializing, 61–63
 - registered custom URL pattern, results of
 - mapping to, 63
 - results, displaying in, 455, 457–458, 468–469, 475–477, 485–486, 498–503, 508, 519–522, 550–551
 - return conditions, mapping to, 460–461, 465–466, 471–472, 491, 512, 529, 546
 - security, managing, 83
 - servlets, compared to, 371
 - Struts forwarding requests to appropriate, 450, 482
 - tag file, creating to use, 372
 - tag handler as a, 371
 - WEB-INF directory, placing inside, 117
 - JSP page properties, configuring, 87–92**
 - JSP results pages, Struts invoking, 454**
 - JSP scripting elements, 488**
 - JSP scripting expression, 358**
 - JSP Standard Tag Library. *See* JSTL**
 - JSP tag libraries. *See* tag library**
 - JSP tags. *See also* tag(s)**
 - custom provided by Struts for beans, 443
 - defining separate components to use custom, 348–352
 - JSP value, for body-content, 351**
 - JSP-based custom tags, 371, 372**
 - JSP-based tag file, 372**
 - jsp-config element, 39, 86**
 - not available in version 2.3, 41
 - <jsp:doBody/> tag, 376**
 - jsp-file element**
 - required for JSP pages, 57
 - substituting for the servlet-class element, 50
 - using instead of servlet-class, 60–61
 - jsp:forward tag, accessing pages, 80**
 - jsp:getProperty tag, displaying bean properties, 488**
 - jsp:include standard action, 431**
 - compared to Tiles approach, 558
 - using dynamic inclusion, 430
 - jspInit method**
 - JSP pages using, 51
 - using in a JSP declaration, 61
 - jsp-property-group elements, 39, 87**
 - jsp-property-group subelement, of jsp-config element, 39, 86**
 - jsp:setProperty, compared to form beans, 481**
 - JspTag instance, 411**
 - JspTagException, 412**
 - jsp:useBean tags**
 - compared to bean:write, 489
 - displaying bean properties, 488
 - JspWriter object**
 - directing the output of the tag body to, 380
 - referencing, 353
 - JSTK (Java Security Tool Kit), 167**
 - JSTK Test CA, signing the CSR, 169**
 - JSTL (JSP Standard Tag Library), 419**
 - downloading, 420
 - installation of, 420–421
 - jstl.jar, copying into WEB-INF/lib, 420**
 - JstWriter class, obtaining an instance of, 349**
 - JVMs, clustered Web applications running in multiple, 95**
- ## K
- key. *See* encryption key**
 - key attribute**
 - for bean:message tag, 545
 - identifying each additional properties file, 543
 - of message-resources, 544
 - referring to a substitution value, 631
 - key-message pairs, in properties files, 539, 542, 552, 611**
 - keystore**
 - for a CA, 167–168
 - selecting for WebClient, 165
 - specifying a different, 175
 - used as default by JRE, 175
 - keystore argument, 158**
 - .keystore file, 158**
 - keytool**
 - certificates from, 157
 - details on, 158
 - keytool command, 168, 169**
- ## L
- LAN (local area network), 95**
 - language preferences, setting in browsers, 554–555**
 - large-icon element, 84**
 - LateAccessFilter class, 223, 224–225**
 - layout(s)**
 - creating JSP pages to populate, 563, 567–568, 577–578
 - defining in WEB-INF/tiles-defs.xml, 584, 588–589

- layout(s) (continued)**
 - making a template file representing, 571–572
 - sketching desired, 560–561, 563–564, 570–571, 586–587
 - sketching using Tiles Definitions, 583
 - layout pieces**
 - JSP pages defining, 562, 565–566, 572–576
 - not accessible by users, 562
 - learning curve, required for Struts, 444**
 - lib directory, 664**
 - life-cycle events, listener classes associated with, 93**
 - listener element, 39, 93, 270**
 - listener-class element, 93, 270**
 - listeners**
 - counting sessions, 298–305
 - packaging with tag libraries, 271
 - using multiple cooperating, 325–339
 - listfiles attribute, of <javac>, 659**
 - load balancer, 95**
 - LoadInitServlet**
 - code listing, 64–65
 - init method storing initialization, 64
 - loading into memory at startup, 67
 - load-on-startup element, 67, 68, 96**
 - local area network (LAN), 95**
 - locale**
 - character encoding used for a particular, 93
 - multiple properties files corresponding to, 540
 - setting explicitly, 554
 - locale files, containing translated messages, 555**
 - locale subelement, 93**
 - locale-encoding-mapping elements, 93**
 - locale-encoding-mapping-list element, 39**
 - locale-encoding-mapping-list subelement, 93**
 - locale-specific properties file, 554**
 - localhost, 12**
 - location element, designating a URL, 72**
 - log file entries, 313**
 - log methods, of the servlet API, 217**
 - LogFilter class, 218, 219**
 - logging filter, 218–221**
 - logging in, as a user with additional privileges, 154**
 - logging out, page for, 112**
 - logical tags, 381**
 - login and login-failure pages, for form-based authentication, 107, 110–111**
 - login attempts**
 - creating a page to report failed, 114
 - result of unsuccessful, 129, 131
 - login page**
 - creating for form-based authentication, 108, 111–114
 - creating to report failed login attempts, 114
 - restricting direct access to, 112–114
 - for a Web application, 128–129
 - login-config element, 38, 111**
 - controlling authentication method, 145
 - specifying form-based authentication, 279
 - specifying user authentication, 78, 254
 - of web.xml, 107, 110
 - login-error.jsp page, 128, 129**
 - login-failure page, 128–129**
 - login.jsp page, 128–129**
 - LongServlet, 246, 247, 249–250**
 - looping index variable, i, 398**
 - looping structure, 399**
 - looping tags, 387–390**
 - creating, 398–399
 - nesting, 400
 - LoopTest servlet, 400**
 - loop-test.jsp page, 400, 401**
 - Luhn checksum, 632**
- ## M
- main connector entry, changing in Tomcat, 158**
 - main-layout.jsp file, 571**
 - main-layout.jsp template file, 588**
 - maintenance, ease with Tiles, 558**
 - make utility, 16**
 - makeTable tag, displaying records in a table, 393, 395**
 - MakeTableTag class, 393, 395–396**
 - makeWarning method**
 - calling, 535
 - modifying, 533
 - manifest attribute, of <jar>, 676**
 - <manifest> task, 678**
 - MANIFEST.MF file, 8**
 - manual validation, 624**
 - of input fields in the Action class, 594
 - performing in input fields in an Action, 596
 - map values, updating and creating, 427**
 - mapping.findForward, 462**
 - marty-with-snake.html, 431, 432**
 - mask, specifying for an input field, 636**
 - mask validation rule, 632**
 - Maven, 16–17, 646**

- message(s)**
 - outputting in JSP pages, 542, 544–545
 - parameterized, 553
 - targeting for a particular input field, 598
 - message attribute, of `html:messages`, 597, 598**
 - message destination**
 - specifying the type and use of, 101
 - unique name for, 102
 - MessageBean, 516, 519**
 - message-destination element, 40, 102**
 - message-destination-name element, 102**
 - message-destination-ref element, 40, 101–102**
 - message-resources element**
 - adding to `struts-config.xml` file, 613
 - common attributes of, 544
 - listing the properties file, 628
 - in `struts-config.xml`, 554
 - message-resources entry**
 - after the controller entry, 628
 - in `struts-config.xml`, 543
 - MessageResources_es.properties file, 556**
 - MessageResources_fr.properties file, 556**
 - MessageResources.properties file, 547**
 - META-INF subdirectory, 8**
 - METHOD, of POST with password fields, 111**
 - methods, implementing for event listeners, 268–269**
 - MIME types, associating files with, 85**
 - mime-mapping element, 39, 85**
 - mime-type subelement, of mime-mapping, 85**
 - missing, 516**
 - missing-data condition**
 - returning, 605
 - in `struts-config.xml`, 604
 - missing-data forward condition, 599**
 - missing-value condition**
 - of `execute`, 511
 - mapped to the original input page, 535
 - returning, 516
 - specifying the location of the original input form, 529
 - missing-value value, 535**
 - missing-value.jsp page, 519, 520, 521**
 - <mkdir> task**
 - in Ant, 654
 - creating directories, 671
 - defining to create a destination directory, 665
 - Model-View-Controller architecture. *See* MVC architecture**
 - modification filter**
 - generic, 237–239
 - specific, 239–244
 - ModificationFilter class, 239**
 - ModificationFilter servlet, 238–239**
 - multimapping patterns, 47–50**
 - MVC (Model-View-Controller) architecture, 441**
 - consistent implementation of, 444
 - loading initialization parameters, 60
 - rigid approach to, 445
 - security constraints not applying to resources accessed through, 253
 - url-pattern not applying to, 80
 - MVC framework, Struts as, 442**
- N**
- {n} notation, in messages, 620**
 - name attribute, 512**
 - in a `<project>` element in Ant, 649
 - of the action element, 460
 - adding to the action element, 487, 492
 - in an Ant target, 650
 - of `bean:write`, 489
 - definition element with, 585
 - of `forward`, 461
 - name element, of attribute, 358**
 - name subelement, of tag, 350**
 - names, assigning to servlets on JSP pages, 42–44**
 - namespace URLs, in the XML Schema file, 37**
 - name-value pairs, in manifest entries, 678**
 - nested custom tags**
 - handling, 410–412
 - usage of, 405
 - no-argument constructor**
 - instantiating the tag handler class, 349
 - provided by the Java compiler, 493
 - NoInvoker servlet, in `hotdotcom`, 140–141**
 - NoInvokerServlet, 53, 54, 156, 210, 216**
 - NONE value, 119**
 - nonexistent values, for initialization parameters, 57**
 - non-SSL requests, redirecting, 195–196**
 - NotFound.jsp, 74**
 - NotFound.jsp error page, 252**
 - null, checking for, 57**
 - null attribute, in message-resources, 543, 544**

O

- object creation, as an expensive operation, 315**
- one result mapping, 463–470**
- onsubmit attribute**
 - adding to html:form tag, 633
 - adding to the beginning html:form tag, 639–640
- order count, resetting, 334–339**
- order form, submitting with missing fields, 642**
- order-confirmation.jsp page, 641, 643**
- OrderFormBean, 638–639**
- order-form.jsp, 639, 640**
- OrderHandlingServlet, 310**
- ordering form, 641**
- orders**
 - HTML form for sending, 307, 311, 312
 - result of cancelling and ordering, 313
- order-tracking system, 326–333**
- organization, facilitated by Web applications, 4**
- org.apache.struts.action.* , importing, 462**
- otherwise tag, 425–427**
- out tag. *See* c:out tag**
- outer tags, communicating with inner, 411–412**
- out.jsp page, 421–422**
- overlapping patterns, resolving matching for URLs, 50**
- overwrite attribute, of <copy>, 657**

P

- packages**
 - placing Action objects in, 466
 - placing all classes in, 461
- packaging company name listeners, 290–296**
- page(s)**
 - creating to report failed login attempts, 108, 114
 - laying out with files, 558–582
 - for logging out, 112
 - reusing repeated sections of, 558
 - simplifying the creation of similar, 558
 - unprotected in a Web application, 141–143
- page layout**
 - sketching out desired, 560
 - template file representing, 564–565
- pageEncoding attribute, of the page directive, 88**
- page-encoding subelement, of jsp-property-group, 88**
- param tag, 433–435**
- parameter attribute, in message-resources, 543, 544**
- parameterized error messages, 620–623**
- parameterized messages, 553**
- parameterized properties file, 621, 623**
- parametric arguments, adding to ActionMessages, 620**
- param-name subelement, 56, 222**
- param-value subelement, 56, 222**
- password(s)**
 - checking in BASIC authentication, 189
 - encrypting, 109
 - setting up for form-based security, 107, 108–110
 - setting with Tomcat, 109–110
 - storing unencrypted, 109
- password file**
 - in form-based authentication, 127
 - Tomcat-specific, 279, 286–287
 - used by Tomcat for a Web application, 151
- password-protected resources, 146**
- Paste Shortcut, 15**
- path attribute**
 - of the action element, 460
 - of Context, 12
 - of forward, 461
- <path> element, 660**
- patterns**
 - exact-match for URLs, 45–47
 - multimapping for URLs, 47–50
 - resolving matching overlapping for URLs, 50
- physical style tags, 381**
- PKI (public key infrastructure), 167**
- placeholders, Struts substituting text into, 631**
- plug-in element, not entering by hand, 628**
- port component, handler for, 101**
- portability, of Web applications, 4, 9, 157**
- port-component-link element, 101**
- port-component-ref element, 101**
- port-name element, 101**
- POST data, sending to the server, 165**
- POST METHOD**
 - as default, 509
 - as default method for html:form, 510
- POST request, in WebClient, 164**
- prefix attribute, of taglib, 352**
- prepare target, in boats2.xml, 670–671**
- prepopulating, forms, 504–537**
- presentation code, separating high-level access decisions from, 203–204**
- preservelastmodified attribute, of <copy>, 657**
- prime number, outputting a random 50-digit, 353**

- prime tag, with variable length, 359–361, 374–375**
 - prime2.tag, declaring an optional attribute, 374**
 - Primes class, 353–355**
 - primes-1.jsp, 359, 361**
 - primes-2.jsp page, 374, 375**
 - PrimeTag class, 359, 360**
 - Principal object, 181**
 - print method**
 - of the JspWriter class, 349
 - on the JspWriter object reference, 353
 - PrintWriter, returning, 234**
 - process control, guaranteeing, 668**
 - ProcessRequestStats servlet, 316, 318–319, 323**
 - programmatically security, 106, 180**
 - combining with container-managed, 180–183
 - example combining container-managed and, 183–187
 - with SSL, 195–201
 - prohibited-site filter, 227–233**
 - project directory structure, placing JAR files in, 660**
 - <project> root element, in an Ant build file, 649**
 - projecthelp option**
 - requesting in Ant, 650–651
 - running Ant on boats1.xml, 664–665
 - properties, defining in Ant, 649**
 - properties file(s), 539, 554**
 - adding parametric notation to messages in, 553, 620
 - advantages of, 540
 - for chooseFormat2, 613
 - containing key-message pairs, 552
 - copying from Struts, 449
 - creating, 542, 609, 611
 - declaring, 543, 547
 - displaying messages in an application, 546
 - editing, 627, 628–629, 635
 - flow of control updates for, 540–541
 - listing in struts-config.xml, 628
 - loading in struts-config.xml, 543–544
 - loading locale-specific, 554
 - parameterized, 621, 623
 - steps for using, 542–545
 - unique identifier for, 544
 - property attribute**
 - of bean:write, 489
 - of c:set, 428
 - in a field element, 630
 - of html:messages, 598
 - indicating messages to extract, 612
 - protected pages, at a Web application, 152–156**
 - proxy headers, proxy servers using, 166**
 - proxy servers, for WebClient, 165–166**
 - pseudo-directory, for each action, 523**
 - public key certificate, creating a self-signed, 157–158, 168**
 - public key infrastructure (PKI), 167**
 - public static method, creating a class with, 405**
 - public void destroy() method, 206**
 - put element, 585**
- Q**
- QName, of the Simple Object Access Protocol (SOAP) header, 101**
 - quiet attribute, of <delete>, 654**
- R**
- random coin flip, 387**
 - realm, designating, 145–146**
 - realm name, designating, 144**
 - realm-name subelement, of login-config, 78, 79**
 - recordRecord method, 316**
 - redirect attribute, of forward, 461**
 - redirect tag, 435–436**
 - redirect1.jsp page, 435–436**
 - redirect2.jsp page, 435–436**
 - RegisterAction2 class, 473–474**
 - registering, listeners, 93**
 - register.jsp page, 550**
 - registration**
 - with the daily special servlet, 228
 - of a filter, 210
 - output pages for incomplete, 522
 - result of a successful, 537
 - as server specific, 9
 - registration page, 521, 536**
 - RegistrationAction.java, 549**
 - RegistrationFormBean.java, 548–549**
 - relative URLs**
 - with c:redirect, 435
 - handling with Tags, 568–569
 - referring to images or style sheets, 141
 - referring to servlets, 133
 - relogin servlet, writing, 144**
 - remove tag, 428–430**

- repetition, avoiding with Tiles Definitions, 583
- replacement filter, 237–244
- ReplaceSiteNameFilter class, implementing
 - the Filter interface, 239, 241–243
- ReportFilter class, 210, 211
 - code listing, 69
 - in the coreservlets package, 70
 - updating to place messages in the log file, 218
- reporting filter, 210–216
- req-stats.jsp page, 316, 319
- request(s)
 - creating Action objects to handle, 473–474
 - defining an Action class to handle, 485, 496–497, 516–517
 - processing with Action objects, 458–481
- request attribute listeners, 268
- request frequency collection, 323–325
- request headers, specifying for WebClient, 165
- request listeners, 268
- request objects, determining number created, 315
- request parameter names, in sync with bean property names, 504
- request parameters
 - extracting from the form bean, 516
 - handling with form beans, 481–503
- request scope, 487
 - beans in, 498
 - storing result beans, 453
- request scope attributes, listener monitoring, 322
- REQUEST value, of dispatcher, 252
- RequestCounter class, 316, 317–318
- requestDestroyed method, 314, 315
- RequestDispatcher
 - accessing pages, 80
 - configuring filters to work with, 251–253
 - handling forwarding of requests, 605, 606
 - handling the request flow, 618
 - in servlets, 441
- RequestDispatcher.forward method, 255
- request.getContextPath, prefixing a URL, 568
- request.getParameter
 - calling, 472
 - making repeated calls, 481
- requestInitialized method
 - calling, 315, 316
 - implementing, 314
- Requests for Comments (RFCs), 190, 197
- RequestStatsBean, 317, 320
- required attribute, of the attribute directive, 374
- required element, of attribute, 358
- required validation rule
 - validating against, 636
 - in validation-rules.xml, 632
- reset method
 - of ActionForm, 487
 - of form beans, 481
- resource factory, 98
- resource-env-ref element, 40, 97
- resource-env-ref-name element, 97
- resource-env-type element, 97
- resource-ref element, 40, 98
- resources
 - accidentally providing unrestricted access to, 253
 - organizing using Web applications, 4
 - releasing of, 270
 - restricting access to, 80–82
- response(s)
 - to authentication failure, 189
 - blocking the normal, 226–227
 - modifying with filters, 234
- response headers, setting, 246
- response wrapper
 - creating, 234
 - reusable, 235–237
- result data, 485
- result mappings, 470–479
- result pages, placing, 465
- results, displaying in JSP pages, 519–522
- results beans
 - creating, 454, 456, 485, 494–496, 506, 508, 515–516
 - creating and storing, 490
- results pages
 - placing, 472
 - preventing direct access to, 458
- resume
 - application, 620
 - choosing colors and font sizes for, 612–619
 - output page for, 600, 605
 - producing an online, 599
- return conditions
 - in execute method, 473
 - mapping to JSP pages, 460–461, 471–472, 491, 512, 529, 546
- return values
 - of initialization parameters, 57
 - mapped by struts-config.xml, 521

RFCs (Requests for Comments), 190, 197**RiskyPage.jsp**

- code listing, 76
- lucky results of, 77
- unlucky results of, 77

role(s)

- accessing a JSP page from different, 183–187
- setting up for form-based security, 107, 108–110
- for users, 107

role element, rolename attribute, 109**role names**

- aliases for existing, 182
- assigning, 83
- in web.xml, 81

role-name elements

- in auth-constraint, 81, 117
- in security-role elements, 108

role-name subelements, of security-role-ref element, 38, 83**root installation directory, in Tomcat, 15****rtexprvalue element**

- of attribute, 358
- as true, 386, 391

runtime values, 620**S****scalability, of J2EE applications, 97****scope (request) attribute, 512****scope attribute**

- of the action element, 460
- adding to the action element, 487, 492
- of bean:write, 489
- of c:import, 430, 431
- of c:set, 428
- of c:url, 434

scoped attribute, setting, 427**scripting elements, not legal in the tag body, 351****scripting-invalid subelement, of jsp-property-group, 88****scriptless value**

- for body-content, 351
- specifying body-content element, 380

Search menu section, of a Tiles layout, 570, 571**secure directory, safeguarding, 254****Secure Sockets Layer. See SSL****security**

- approaches within the Web application framework, 106
- handling programmatically, 188–194

providing, 78–83

strategies for implementing, 179

security hole, plugging, 253–260**security role aliases, 182–183****security settings, applying to actions, 524****security-constraint element, 38, 80**

- example of the use of, 115
- restricting resource access, 254
- specifying restricted URLs, 146
- stipulating authenticated users in the ceo role, 279
- user-data-constraint subelement of, 108
- of web.xml, 108, 115

SecurityHolePlugFilter class

- implementing the Filter interface, 253, 255–256
- servlet testing, 254

SecurityHoleServlet servlet, 254, 256**SecurityInfo servlet, 198–200, 201****security-role element, 38–39**

- containing a required role-name element, 145
- declaring abstract roles, 118
- declaring the executive role, 254
- directly under web-app, 81

security-role-ref element, 38–39

- defining aliases, 181
- as a subelement of servlet, 83

security-role-ref subelement, of servlet, 182**self-signed certificates**

- resulting in warnings, 136
- for testing purposes, 157

sendRedirect method

- calling, 254
- servers using, 112

sendRedirectURL method, 435**sensitive data, preventing unauthorized access of, 105, 179****separation, of Web applications, 4****Serializable interface, HttpSession objects implementing, 96****serialization, of session objects, 268****server(s)**

- assigning passwords and role membership, 109
- handling security, 106
- loading lower numbered servlets or JSP pages before higher numbered, 68
- prohibiting access to ServletContext, 27
- registering Web applications with, 9–14

server certificate

- first and last name for, 157
- signing, 167–169

- server request load, 315–321
- server-side validation, 625, 641
- server-specific component, 188
- server.xml, backing up before editing, 11
- Service Endpoint Interface, 101
- service-endpoint-interface element, 101
- service-interface element, 100
- service-qname element, 101
- service-ref element, 40, 99–101
- service-ref-name element, 100
- servlet(s)
 - accessing, 5–6, 12
 - assigning names to, 42–44
 - associating a filter with, 70
 - blocking access to, 205
 - configuring to load at server startup, 68
 - customizing initialization behavior, 221
 - dependencies, 188
 - explicitly declaring, 11
 - initializing and preloading, 56–68
 - loading when the server starts, 64–68
 - managing security, 83
 - modifying the output of, 205
 - placing in WEB-INF/classes, 6
 - protecting, 116
 - providing initialization parameters, 56
 - referring to by registered name, 43
 - security-role-ref subelement, 182
 - specifying with the same load-on-startup number, 68
- /servlet/ * URL mapping entry, 55
- servlet application programming interface (API), 20
- servlet context
 - accessing from filters, 217
 - monitoring creation and destruction of, 270–271
 - obtaining a reference to, 270, 272
 - using, 270, 272
 - of a Web application, 26
- servlet context attribute listeners, 268
 - detecting changes in attributes, 334
 - steps in using, 277–278
- servlet context listeners, 268
- servlet contexts, drawbacks to sharing, 27
- servlet element, 38, 42
 - in web.xml, 11, 209
- servlet log file, writing reports into, 217
- servlet name, associating the filter with a specific, 70
- servlet request, identifying initialization and destruction, 314–315
- servlet specification, versions of, 37
- servlet URL, default, 108
- /servlet/ URL pattern, remapping, 53–55
- /servlet/* pattern, remapping, 52, 121, 209
- servlet-api.jar file, in Tomcat, 20
- servlet-class subelement, of servlet, 42–43
- <servlet-class> element, 21
- ServletContext
 - accessing from the doFilter method, 217
 - not storing data in, 96
 - obtaining a reference to, 316
- ServletContext objects
 - associated with a specific URL, 26
 - sharing minimal information among Web applications, 25
- ServletContextAttributeEvent class, methods of, 278
- ServletContextAttributeListener interface, 277, 279, 326
- ServletContextEvent class, getServletContext method, 270
- ServletContextListener class, 270–271, 326
- servlet-mapping element, 38
 - assigning a custom URL, 44
 - url-pattern subelement, 87
 - in web.xml, 6, 11
- servlet-mapping web.xml element, 120, 209
- servlet-name subelement
 - of filter-mapping, 208
 - of servlet, 42–43
 - of servlet mapping, 44
- ServletOutputStream
 - as an abstract class, 235
 - returning, 234
- ServletRequest argument, of doFilter, 205
- ServletRequest interface, 195
- ServletRequest object
 - in doFilter, 204
 - filters having full access to, 69
- ServletRequestAttributeEvent class, methods of, 322
- ServletRequestAttributeListener interface, 322
 - adding to version 2.4 of the servlet specification, 269
 - implementing, 322, 323, 324
- ServletRequestEvent class, getServletRequest method, 314

- ServletRequestListener interface, 314**
 - adding to version 2.4 of the servlet specification, 269
 - implementing, 314, 316
 - steps involved, 314
- ServletResponse argument, of doFilter, 205**
- ServletUtilities class, 382–383**
- session(s), listeners counting, 298–305**
- session attribute listeners, 268**
- session attributes**
 - names of all, 27
 - watching for changes in, 306
- session counts**
 - JSP page displaying, 299, 302
 - placing, 298–299
- session creation and destruction, 297–298**
- session listeners, 268**
- session objects, 39**
- session scope, 453, 487**
- session timeouts**
 - choosing low, 299
 - controlling, 83–84
- session tracking**
 - automatic, 433
 - increasing the server's memory load, 298
 - remembering validated users, 107
- session-config element, 39, 83–84**
- session-config entry, creating in web.xml, 299**
- SessionCounter class, 298, 300–301**
- sessionCreated method, 297, 298**
- sessionDestroyed method, 297, 298**
- session-timeout element, 83–84**
- set tag, 427–430**
- setAttribute method, calling on the session object, 297**
- setAttributeName method, 358**
- setCharacterEncoding method, 93**
- setContentype method, 93**
- setCount method, of ForTag, 387**
- set.jsp page, 428–430**
- setLength method, of PrimeTag, 359**
- setLocale method, 93**
- setMaxInactiveInterval method, of HttpSession, 39, 83, 84**
- setPath method, of the Cookie class, 26**
- SetSharedInfo servlet, 27–28**
- settable bean properties, 488, 493**
- setter method**
 - providing for an attribute, 385–386
 - for results beans, 494
- shareability, connections obtained from a resource, 98**
- shared condition (forward) mappings, combining, 479–481**
- shortcuts, simplifying in deployment, 15**
- ShowRecords servlet, 393**
- show-records.jsp page, 393, 395**
- ShowSampleAction Action class, 599, 603–604**
- ShowSampleAction class, modified, 613, 617**
- ShowSharedInfo servlet, 27, 28–29**
 - shareTest results, 31, 32
 - shareTest2 results, 31, 32
- signup2.jsp, 534**
- SignupAction1.java, 517**
- SignupAction2.java, 533**
- signup-confirmation.jsp page, 519, 520, 521**
- simple addresses (without subdirectories), for URLs, 523**
- simple authentication, 105**
- simple messages example, 546–552**
- simple prime tag, using tag files, 372–373**
- simple Tiles example, 563–568**
- SimpleLoopTest.java class, 387, 388**
- simple-loop-test.jsp page, 387, 389**
- simplePrime2.tag, 372**
- simple-primes-1.jsp page, 356–357**
- simple-primes-2.jsp, 372, 373**
- SimplePrimeTag tag handler class, 353, 355**
- SimpleTag API**
 - custom tags utilizing, 348
 - illegal to include scripting elements, 351
 - letting inner tag get hold of an outer tag, 411
- SimpleTag interface, 348, 357**
- SimpleTag model, 349**
- SimpleTagSupport class**
 - extending, 353, 381
 - in javax.servlet.jsp.tagext package, 348
- slash (/), beginning the value of url-pattern element, 44**
- small-icon element, 84**
- snake.html, importing, 431, 432**
- soap-header element, 101**
- soap-role element, 101**
- software-based load balancers, 95**
- source code**
 - archive for this book, 559
 - copying files to the deployment server, 15
 - protecting from potential hackers, 677
- source code management system, 664**
- source files, recompiling, 666**

- SPAN element, one-cell table enclosing**, 364
- Spanish, internationalizing for**, 555–557
- SRC attribute**, 431
- src directory**, 664
- srcdir attribute, of <javac>**, 658, 659
- src/main/java folder**, 16
- SSL (Secure Sockets Layer)**
 - with Apache Tomcat, 174
 - configuring Tomcat to use, 156–164
 - determining if in use, 195
 - encrypting traffic, 105, 119
 - need for, 190
 - programmatically security and, 197–201
 - programmatically security with, 195–197
 - redirection to the URL using, 107
 - with relative URLs, 133
 - specifying URLs available only with, 108, 119–120
 - in WebClient, 165
- SSL communication, with Tomcat through WebClient**, 175, 176
- SSL connector entry, editing in Tomcat**, 158
- ssl directory**, 132–137
- SSLHandshakeException “No trusted certificate found.” exception**, 175
- SSLSocketFactory, obtaining a socket from**, 165
- standard header, providing in multiple pages of an application**, 89
- standard.jar**, 420
- startup file, setting CLASSPATH in**, 447
- static content, associating a filter with**, 70
- static data, avoiding for shared data**, 95–96
- static method, calling inside a JSP page**, 405
- stock purchase page, accessing with SSL**, 137
- StockTip servlet**, 191–193
- StopRequestCounter class**, 323
- StringOutputStream method**, 235, 237
- strings, defining in properties files**, 542
- String.split method, of BannedAccessFilter**, 227–228
- StringWrapper class**, 235–236
- Struts**
 - Action class manual validation of input data, 594–595
 - ActionForm bean updates for validation, 607–608
 - adding to an existing Web application, 449
 - advantages of, 442–444
 - automated validation, flow of control with updates for, 625–626
 - automatic validation, configuring for, 627
 - basic steps in using, 484–486, 506–508
 - compared to the standard MVC approach, 442–445
 - different views of, 441–442
 - disadvantages of, 444–445
 - downloading, 446–447
 - flow of control, 450–454, 525
 - html:tags and form beans, flow of control, 504–506
 - installing, 446–447
 - populating the bean with input data, 599
 - processing flow of a request in, 490
 - processing flow through, 469
 - properties files, flow of control with updates for, 540
 - setting up, 446–449
 - steps implementing control flow, 454–458
 - testing, 448
 - understanding, 441–445
- Struts 1 framework**, 445
- Struts 1.3**, 446
- Struts 2 framework**, 445
- Struts applications**, 448–449
- Struts configuration file. *See* struts-config.xml file**
- Struts documentation, bookmarking**, 446, 447
- Struts html: tag library, importing**, 569
- Struts html:tags. *See* html:tags**
- Struts servlet definition**, 558–559
- Struts servlet entry**, 559–560
- Struts Shale**, 445
- Struts Tiles**, 539. *See also* Tiles
- Struts Tiles tag library**, 563, 586
- Struts Validation framework**, 593
- Struts-based Web applications**, 446
- struts-beans/register.jsp form**, 498
- struts-blank Web application**, 446, 448
- struts-config.xml file**, 465–466
 - changing, 455–456
 - checking a properties file, 609
 - for chooseFormat1, 600, 604
 - for chooseFormat2, 613, 616
 - complete, 472
 - complete listing, 492, 512–513, 547
 - configuring for automatic validation, 627–628, 634
 - copying, 449
 - defining Actions in, 459
 - defining form beans, 486

- deleting examples and comments from, 456
 - with a forward definition, 453
 - forward entry, 463
 - listing of, 530
 - loading the properties file in, 542, 543–544
 - mapping error code back to the input form, 599
 - modifying, 454, 455–456, 463, 465–466, 471–472, 484, 491–492, 507, 511–513, 529, 546
 - reading only when Web applications are first loaded, 456
 - specifying an input page, 609, 610
 - with two actions, 479–481
 - struts-message Web application, 552**
 - stubbed-out layout, HTML fragments for, 560**
 - stubs (tiles:tags), establishing, 561**
 - stylesheet information, retrieving from a bean, 600, 605**
 - stylesheets, 570**
 - subdirectories**
 - advantages of using, 524
 - designing action URLs without and with, 523–524
 - disadvantages of using, 524
 - grouping all when grouping files in a fileset, 656
 - success condition, 477, 479**
 - of execute, 511
 - execute method returning, 501
 - mapped by struts-config.xml, 535
 - mapping to a single results page, 546
 - success return condition, 473**
 - successful login, result of, 129, 132**
 - SuggestionBean class, 494, 495, 497**
 - SuggestionUtils class, 494, 495–496**
 - Sun Certified Web Component Developer (SCWCD) certification, 419**
 - sun.misc.BASE64Decoder class, 190**
 - swimming world records, 393**
 - switch-case-default statements, 425**
 - symbolic links, 15**
 - system setting, setting CLASSPATH through, 447**
 - system time, recordRecord method adding, 316**
 - system-wide initialization parameters, 63**
- T**
- table-formatting tag, listing records, 393**
 - tag(s). *See also* custom tags; JSP tags**
 - assigning attributes to, 357–359
 - tag attributes**
 - assigning complex objects as values, 391–392
 - assigning dynamic values to, 385–387
 - declaring inside a tag element, 358
 - tag body**
 - controlling a small part of, 398
 - including in tag output, 362–363
 - manipulating, 380–381
 - outputting inside a tag file, 376
 - sending the JSP content of, 363
 - tag body scope attribute, 399**
 - tag element, subelements of, 350–351**
 - .tag extension, 372**
 - tag files**
 - attributes declared inside, 374
 - creating, 371–372
 - heading tag using, 376–377
 - placing in META-INF/tags directory in JAR files, 8
 - placing in WEB-INF/tags directory, 8
 - prime tag with variable length using, 374–375
 - simple prime tag using, 372–373
 - tag handler class, 348–349**
 - adding attribute support, 357–358
 - complex dynamic attribute values, 391
 - dynamic attribute values, 385–386
 - supporting tag bodies, 362–363
 - tag library. *See also* bean tag library**
 - declaring, 406, 508
 - packaging event listeners in, 288–289
 - Tag Library Descriptor files. *See* TLD files**
 - tag library descriptors, locating, 86**
 - tag output, including tag body in, 362–363**
 - tag-based code, inside JSP pages, 379**
 - tag-class subelement, of tag, 350**
 - tagdependent value, for body-content, 351**
 - tagdir attribute, declaring tag files, 8**
 - taglib directive, 352, 386, 392**
 - assigning a prefix for a JSP page, 406
 - declaring tag files, 8
 - importing the JSTL tag library, 420
 - JSP pages accessing TLD files, 7–8
 - tagdir attribute, 372
 - uri attribute, 86
 - taglib element, 86**
 - taglib subelement, of jsp-config element, 39, 86**
 - taglib-location subelement, of taglib element, 86**
 - taglib-uri subelement, of taglib element, 86**
 - target attribute, of c:set, 428**
 - target dependencies, benefits of, 668–669**

<target>, with a depends attribute, 669**targets**

- in an Ant project, 648
- assigning tasks to in Ant, 651
- creating dependencies between in the build file, 668
- logically bundling together all, 672
- writing in Ant projects, 650–651

.tar.gz format file, 420**tasks, assigning to targets in Ant, 651****template, for the TLD, 351****template file(s)**

- creating, 561–562
- making to represent a layout, 571–572
- not allowing user access to, 562
- placing in WEB-INF, 560, 562
- representing a layout, 560, 561–562, 564–565, 583

test attribute

- of c:if, 424
- of c:when, 425

testApp

- accessing with URLs, 23–25
- unzipping app-blank.zip to, 18

testApp directory, copying to tomcat_dir/webapps, 23**test.html**

- adding to the testApp directory, 19
- downloading, 19
- URL for retrieving, 23, 24

test.jsp

- adding to the testApp directory, 19
- downloading, 19
- URL for retrieving, 23, 24, 25

TestPage.jsp

- listing of, 51
- results of invoking, 51, 52

TestServlet1, result of invoking, 44–45**TestServlet1 code, 43****TestServlet.class**

- declaring, 21
- URL for retrieving, 23, 25

TestServlet.java

- compiling, 20
- downloading, 19

text fields

- html:text for, 511
- prepopulating with initial values, 504
- redisplaying values in, 504

text file, placing messages in, 539**text value, substituting into a message, 553****ThenTag class, 412, 414****tile:insert tag, declaring the template file, 567****Tiles**

- laying out pages with, 558–582
- page layout facility, 442
- prerequisites for, 558–560
- reasons for using, 558
- steps in using, 560–563

Tiles Definitions, 582–590

- reasons for using, 583
- steps in using, 583–586

Tiles JAR file, 559**Tiles plug-in**

- adding to struts-config.xml, 559
- using in Struts, 449

Tiles tag library, declaring, 561**tiles-def.xml, 582**

- defining layouts in, 584–586
- format for, 584–585
- specifying the top-level layout definition, 588

tiles:getAsString tag, 560, 564, 565, 583

- stubbing out locations for String substitutions, 561–562

- for the title section, 571

tiles:insert tag, 560, 564, 565, 583

- applying appropriate layout definitions, 589
- referring to the definition layout, 586
- referring to the layout template file, 563
- stubbing out sections, 561, 571

tiles:put elements, on main Web pages, 577**tiles:put tag, 560**

- specifying content, 567
- specifying file fragments, 577
- specifying layout pieces, 563

tiles:tags, 539, 558

- acting as stubs, 560
- placing layout pieces, 561

time properties, 653**timeout, setting, 83****timestamp property, for the manifest file, 679–680****Title section, of a Tiles layout, 570, 571****TLD (Tag Library Descriptor) file(s), 349–351**

- adding attribute support, 358
- changing the location of, 39
- for company-name-taglib, 292
- complex dynamic attribute values, 391–392
- declaring a tag file and its path, 8
- declaring an EL function method inside, 405–406
- dynamic attribute values, 386

- example, 349
 - giving the real location of, 86
 - placing in WEB-INF directory, 7–8, 289, 351
 - placing inside META-INF directory, 8
 - putting listener declarations in, 288
 - referring to from multiple JSP pages, 352
 - tag bodies and, 363
 - template for, 289
 - in XML format, 349
 - TLD csajsp-taglib.tld, 355–356**
 - .tld extensions, system searching for, 288**
 - TLDDoc, for JSTL, 420**
 - tds directory, within WEB-INF, 7**
 - toBeMatchedClass type, 411**
 - TODAY property, in Ant, 653**
 - TodaySpecialServlet, 215**
 - todir attribute, of <copy>, 657, 658**
 - tofile attribute, of <copy>, 657**
 - Tomcat**
 - authentication used by, 81
 - auto-deployment directory, 23
 - configuring to use SSL, 156–164
 - enabling SSL support in, 157–164
 - global web.xml configuration settings, 7
 - globally disabling the invoker, 55–56
 - information on setting up, 55
 - localhost log on, 220–221
 - locating JAR files, 7
 - password file used by, 127
 - registering Web applications with, 10–12
 - setting passwords with, 109–110
 - setting up to use SSL directly, 120
 - setup and configuration guide, 9
 - software-based load balancer, 95
 - starting, 23
 - Tomcat 5.5, placing the .war file, 675**
 - tomcat_dir, 15**
 - tomcat_dir/conf/server.xml, Context element in, 11**
 - tomcat_dir/webapps, copying testApp to, 23**
 - tomcat-users element, 109**
 - top-level directory**
 - location varying from server to server, 9
 - of a Web application, 5
 - top-level home page, for a Web application, 122, 147–148**
 - top-level layout, defining, 582, 585**
 - totalSessionCount, 298**
 - transparency, less with Struts applications, 445**
 - transport-guarantee element, 133**
 - transport-guarantee subelement, of user-data-constraint, 82, 119**
 - trusted certificate, 169**
 - Trusted Root Certification Authorities tab, 171**
 - try/catch block, enclosing Integer.parseInt call, 222**
 - try/catch construct, 437**
 - <tstamp> task, in Ant, 653–654**
 - type attribute**
 - of the action element, 460
 - designating a fully qualified class name, 465
- ## U
- unauthenticated users**
 - dialog box for, 153
 - directly accessing the login page, 112
 - invoking another protected resource, 114
 - requests by, 338
 - unauthorized users, preventing from accessing sensitive data, 105, 179**
 - Un-Divet tool, Web site for ordering, 633**
 - uniform resource locators. See URL(s)**
 - UNIX/Linux, setting the CLASSPATH, 21**
 - unprotected pages, in a Web application, 141–143**
 - unsuccessful login page, 153**
 - update attribute, of <jar>, 676**
 - uri attribute, of taglib, 352**
 - URL(s) (uniform resource locators)**
 - accessing testApps, 23–25
 - assigning multiple to invoke the same servlet, 47
 - beginning with slashes, 568, 569
 - custom, 44–50
 - default, 52, 57
 - designating for protection, 80
 - for each Web application, 4
 - encoding, 433
 - ending with a certain extension, 49
 - filter registration with all, 218
 - general options for, 523
 - handling relative with Tags, 568–569
 - mapping to match JSP pages, 87
 - password protected for form-based authentication, 108, 115–118
 - password protected in BASIC authentication, 145, 146
 - redirecting in JSP pages, 435
 - for a servlet as a security hole, 5–6
 - specifying SSL availability only, 108, 119–120, 145, 147

- url attribute, of `c:redirect`, 435**
- URL pattern, matching exactly, 45**
- URL prefix, registering for a Web application, 5**
- URL rewriting**
 - as the basis of session tracking, 107
 - session tracking through, 433
- url tag, 433–435**
- url.jsp page, 434–435**
- url-pattern element, 80**
 - applying to clients accessing resources directly, 116–117
 - associating original URL with registered servlet name, 61
 - directing matching requests to a servlet, 53
 - in `web.xml`, 11
- url-pattern subelement**
 - of filter-mapping, 208
 - of `jsp-property-group`, 87
 - of servlet mapping, 44
- user(s)**
 - accessing protected resources, 81
 - defining for a Web application, 151
 - defining in Tomcat, 109, 110
 - designating for resource access, 117
 - returning name of current, 181
- user authentication. *See* authentication**
- user elements, attributes of, 109**
- user ID, storing the `HttpSession` object, 188**
- user-data-constraint, 80, 82, 115, 119, 145, 146**
- `user.dir` property, with `<echo>`, 653**
- `UserFormBean`, 493–494**
- username/password string, getting the encoded, 189**
- usernames**
 - checking in BASIC authentication, 189
 - setting up for form-based security, 107, 108–110
- user-specific data, monitoring changes to, 297**
- `USTMBody.jsp`, 89, 91**
- `USTMFooter.jsp`, 89, 91**
- `USTMHeader.jsp`, 89, 91**
- Util class, defining, 407, 408**
- utilities, Struts as a collection of, 442**
- V**
- validate attribute, specifying as optional, 610–611**
- validate method**
 - of `ActionForm`, 487, 593, 618
 - checking input fields for errors, 607
 - creating in `ActionForm`, 609–610
 - of form beans, 481
- validation**
 - in the Action, 593
 - automatic, 594
 - client-side versus server-side, 624–625
 - performing in Action, 596–598
 - preventing, 611
- validation logic, problems in, 475**
- validation rules**
 - available from `validation-rule.xml`, 630, 632
 - putting in `validation.xml`, 629–632
 - setting up for input fields, 630–632
 - specifying dependency of, 630
 - in the Struts Validation framework, 630
- validation.xml file**
 - putting validation rules in, 627, 629–632, 636–637
 - structure of, 629–630
- validator messages, editing in the properties file, 628**
- Validator plug-in, 449**
- `ValidatorForm`, extending, 627, 638–639**
- value attribute**
 - of `c:set`, 427–428
 - of `c:url`, 433
- `value.required` parameterized message, 620**
- var attribute**
 - of `c:import`, 430
 - of `c:set`, 428
 - of `c:url`, 434
- vendor lock-in, avoiding, 4**
- verbose attribute, of `<javac>`, 659**
- `-verbose` command-line option, 666**
- version 2.3 of `web.xml`, ordering of web-app elements with, 40–42**
- version 2.4 of `web.xml`, web-app subelements declared with, 38–40**
- W**
- WAR (Web ARchive) files**
 - bundling Web applications into, 17–18
 - copying into `tomcat_dir/webapps`, 23
 - creating using Ant, 675–678
 - for deployment of boats Web application, 679
 - listing the contents of, 680
 - target directory for building, 664
 - zipping `testApp` into, 23
- WAR file name, creating a local property for, 679**
- `.war` file name extension, 676**
- war target, for the build file, 679–682**

warning property, of ContactFormBean, 530

warnings, stored in the form bean, 533

Web application(s), 3, 267

- adding Struts application to existing, 449
- advantages of, 3–4
- aspects to securing, 105
- auto-deploying through a WAR file, 675
- building, 18–25, 670–675
- bundling into WAR files, 17–18
- creating a directory structure, 10, 14
- deploying, 14–17, 36
- developing, 14, 95–97
- development and deployment, 14–17
- documenting, 84–85
- home page for, 147–148
- keeping data and functionality separate, 25
- organizing into subdirectories, 664
- prefixes prepended automatically, 509
- registering, 5, 9–14
- separation of, 4
- sharing data among, 25–32
- standard location for each type of resource, 4
- structure of, 5–8
- Tomcat password files not specific to, 151
- using Ant to build, 668–669

Web application objects, obtaining access for event listeners, 269

Web application security, fundamental aspects of, 179

Web application WAR file, creating, 679–682

Web archive files. *See* WAR files

Web console application, WebLogic's, 13

web directory, 664

Web pages. *See* JSP page(s)

Web resources. *See* resources

Web servers

- choosing the vendor of, 4
- talking to interactively, 164–166

Web Service, declaring a reference to, 99–100

Web Services Description Language file. *See* WSDL file

Web sites

- designing using Tiles Definitions, 582
- representative pages for, 563–564

web-app

- correctly ordering version 2.3 elements, 40
- as top-level (root) element, for the deployment descriptor, 36

Web-based applications, debugging, 164

WebClient

- options, 164
- representative conversation of, 166
- talking to Web servers interactively, 164–166
- with Tomcat and SSL, 175–176

WEB-INF directory, 4

WEB-INF/classes directory, 4, 542

WEB-INF/lib

- directory, 4
- JAR files in, 7

WEB-INF/tiles-defs.xml

- defining layouts in, 584–586
- for e-boats2, 588–589
- specifying the top-level layout definition, 588

WebLogic server, registering Web applications with, 13

web-resource-collection, 80–81

- element, 108
- subelement of security-constraint, 108, 115, 116–117, 146

web-resource-name element, 80

web.xml file

- abstract roles, listing all possible, 118
- for access time filter, 226
- from the app-blank template Web app, 30–31
- for the application events examples, 339–345
- company name listener, changed, 281–283
- for compression filter, 250–251
- configuration properties to JSP pages, configuring, 87
- creating, 10
- custom URLs, declaring names and mapping, 42
- declaring with version 2.3, 37
- default servlet URLs, blocking, 209
- default servlet URLs, redirecting requests from, 121
- default URLs, showing how to disable, 53
- designating form-based authentication, 110, 111
- distributable element, 97
- downloading a blank, 37
- elements of, 37
- error pages for exceptions, designating, 77
- error pages for HTTP error codes, designating, 73
- exact matching, showing, 45–46
- filter declaration in, 70
- filter examples, complete version for, 260–265
- filter usage, showing, 71
- formal specification of, 35
- global on a few servers, 7

web.xml file (continued)

- initialization parameters, illustrating, 59
- J2EE environment, elements in, 97
- JSP pages, illustrating the naming of, 51
- jsp-property-group, showing, 89, 90
- lines, adding to, 21
- listener, declaring, 94
- location of, 4
- for logging filter, 220
- login-config, showing, 79
- modifying, 11
- multimapping, showing, 47
- for prohibited-site filter, 232–233
- protected URLs, specifying, 115
- purpose of, 35–36
- registered name for a JSP page, assigning, 61, 62–63
- for reporting filter, 211–212
- for RequestCounter listener, 321
- for resetting order counts, 337
- security, elements relating to, 78–83
- security role aliases, illustrating, 182–183
- from securityInfo Web app, 200
- server supporting nonstandard, 110
- server-specific global, 55
- servlet name, showing, 44
- for session counting listener, 299, 301
- for site name replacement filter, 243
- for StopRequestCounter listener, 324–325
- testApp, complete code of, 21–23
- text editor, opening in, 21
- tracking daily special orders, 331
- URL pattern, assigning a custom, 58, 59
- using instead of servlet-class, 62–63
- version 2.3, ordering of web-app elements with, 40–42
- visual development environment, elements designed for, 84
- way of catching exceptions, 75
- for a Web application, 149–151
- Web applications, complete version for, 124–127

- Web applications, customizing behavior of, 36
- WEB-INF subdirectory, placing in, 7
- welcome-file-list entry in, 6
- yacht-watching listener, 309

web.xml version 2.4, web-app subelements declared with, 38–40**welcome pages, specifying, 71–72****welcome-file element, 71, 72****welcome-file-list element, 38, 71–72****welcome-file-list entry**

- making an explicit, 6
- in web.xml, 122, 141, 147

when tag, 425–427**Windows, setting the CLASSPATH, 20****WorldRecords class, 393, 394****WSDL (Web Services Description Language) file, 100****WSDL Service element, 101****wsdl-file element, 100, 101****WWW-Authentication header, 78****X****X.509 certificates, accessing client, 197****XML**

- Ant projects written in, 648
- Apache Ant based on, 646
- case sensitivity of elements, 36

XML header, of the deployment descriptor, 36**XML Schema, 36, 289****XML type characters, escaping, 421****Y****yacht orders, monitoring, 307–313****YachtWatcher class, 307, 308–309****Z****zero-argument constructor, 488****zip format, 420****<zipfileset> type, 677**