

Index

Symbols

- \$wnd object, JSNI, 216
- & (ampersand), in GET and POST parameters, 112–113
- { } (curly braces), JSON, 123
- ? (question mark), GET requests, 112

A

- Abstract Factory pattern, 258–259
- Abstract methods, 332
- Abstract Window Toolkit (AWT), Java, 31
- AbstractImagePrototype object, 245
- Abstraction, DAOs and, 486
- AbstractMessengerService
 - Comet, 474
 - Jetty Continuations, 477
- action attribute, HTML form tag, 507
- Action-based web applications
 - overview of, 116
 - PHP scripts for building, 523
- ActionObjectDAO class, 527–530
- Actions, server integration with, 507–508
- ActionScript, 6
- ActiveX, 7
- Add Import command
 - creating classes in Eclipse, 152
 - writing Java code using Eclipse Java editor, 154–155
- addContact method
 - ContactList class, 428
 - ContactListView class, 441
- addHistoryListener method, Hyperlink widget, 46
- addItem method, MenuBar widget, 68–69
- addLoadListener method, Image widget, 44
- addMessage method, ChatWindowView class, 444–445
- addSearchResult method
 - JUnit test case, 175
 - SearchResultsView class, 329
- addSearchView method, MultiSearchView class, 327
- addStyleName method, connecting GWT widgets to CSS, 201
- addToken method, handling back button, 199
- addTreeListener method, Tree widget, 67
- Adobe Flash and Flex, 6–7
- Aggregator pattern
 - defined, 34
 - Multi-Search application and, 319–321
 - sample application, 35
- Aggregators, 320
- Ajax (Asynchronous JavaScript and XML)
 - alternatives to, 6–8
 - application development and, 14–16
 - building web applications and, 479
 - emergence of, 3–5
 - Google Gears for storage, 306–309
 - Same Origin policy and, 335
 - success and limitations of, 5–6
 - usefulness of GWT to developers, 13
 - word processor, 362
- Ajax (Asynchronous JavaScript and XML), software engineering for, 139–190
 - adding projects to Eclipse, 145–149

- Ajax (Asynchronous JavaScript and XML), *continued*
 - automating deployment with Ant, 187–190
 - building and sharing modules, 180–183
 - debugging in Eclipse, 158–164
 - deploying servlet to servlet container, 184–186
 - deploying to web server, 184
 - installing Eclipse, 144–145
 - installing GWT, 140–144
 - installing JDK, 140
 - organizing application structure, 164–168
 - overview of, 8–11
 - testing applications. *See* Testing applications
 - using modules, 180–181
 - writing Java code in Eclipse, 149–157
 - Amazon Search, 354–358
 - Amazon Web Services (AWS), 354–356
 - Anonymous classes, 194–195
 - Ant script, 187–190
 - Apache
 - caching on, 246–247
 - compression on, 249
 - API key, widget based on Google Map, 224
 - Application patterns, 34–35
 - applicationCreator script
 - creating module with, 182
 - installing GWT, 141–142
 - organizing application structure, 164–168
 - Applications. *See also* Web applications
 - development skills, 14–16
 - internationalizing. *See* Internationalizing applications
 - sharing compiled, 183
 - APR connector, 475
 - AsyncCallback interface, 134–135, 454–456
 - asyncGet method, HTTPRequest, 28
 - Asynchronous HTTP requests
 - server integration techniques, 106–109
 - using, 27–29
 - Asynchronous interface, GWT-RPC using, 454–456
 - Asynchronous JavaScript and XML. *See* Ajax (Asynchronous JavaScript and XML)
 - Asynchronous programming, 191–196
 - blocking operations and, 192
 - features in GWT, 193–196
 - IncrementalCommand class, 193
 - sequential vs., 191–192
 - Atom
 - aggregators and, 320
 - GData protocol based on, 344
 - RSS feeds and, 402
 - AtomEnabled Alliance, 402
 - Authentication, 222–226
 - AuthSub interface, 404–410
 - AWS (Amazon Web Services), 354–356
 - AWT (Abstract Window Toolkit), Java, 31
- ## B
- Back button, handling in Ajax, 197–200
 - BaseObject class, 484
 - Benchmarking, for testing applications, 176–179
 - Binding, browser interoperability and, 302–303
 - Block elements
 - CSS properties, 207–209
 - defined, 71
 - Blocking operations, in asynchronous programming, 192
 - Blog class, 365–367
 - Blog Editor application, 361–422
 - Atom Publishing Protocol and GData and, 402
 - BlogEditorView class, 370–372
 - BlogEntryView class, 376–378
 - BloggerService class, 402–403
 - BlogView class, 372–376
 - bypassing HTTP request limitations, 108
 - controller for, 390–392
 - creating new widgets, 213
 - cross-domain RequestBuilder for proxy servlet, 394–397
 - delete requests for blog entries, 419–420
 - designing, 365
 - EditEntryView class, 378–381
 - Google account sign-in, 403–410
 - HTTP Proxy servlet, building, 392–394
 - ImageBundle, internationalization, and RichTextToolBar, 383–386
 - integration with, 401
 - integration with stateless servers, 130–131
 - LoadingPanel widget, 386–388
 - model for, 365–367
 - multiple document view for, 367–369
 - overview of, 361
 - page relationships in, 368
 - reading and writing XML, 120–121
 - rich text editing added to, 381–382
 - RichTextArea widget, 382–383
 - summary, 421–422
 - third-party server integration, 128
 - TitleCommandBar widget, 388–390
 - web services client, building, 363–365
 - Workspace pattern, 361–363
 - writing proxy servlet, 397–401
 - XML for creating and saving blog entries, 416–419
 - XML listing of blog entries, 413–415
 - XML listing of blogs, 410–413

- BlogEditorView class, 370–372
- BlogEntry class, 365–367, 416–419
- BlogEntryView class, 376–378
- BloggerService class, 402–403
- Blogs
 - defined, 361
 - improvements in usability, 4
- BlogView class, 372–376
- BlogViewListener interface, 371
- Borders, CSS properties, 207–208
- Breakpoints, in Eclipse
 - adding exception, 161–163
 - debugging Eclipse, 158–161
- broadcastEvent method, instant messenger server, 468–469
- Browsers
 - back button and history and, 197–200
 - CookieStorage class and, 300–302
 - Google Gears for storage, 306–309
 - GWT support across, 218–221
 - handling differences between, 302–303
 - importing structured data formats, 334
 - Same Origin policy and, 334–336
- Button states, Button widget, 50–52
- Button widget
 - button states, 50–52
 - creating dynamic interface, 25
 - overview of, 47–49
- Buttons
 - Button widget, 47–49
 - CheckBox widget, 53
 - Gadget Desktop, 280–282
 - PushButton widget, 49–52
 - RadioButton widget, 54
 - ToggleButton widget, 49–52
- C**
- Cache-control header
 - caching on Apache, 247
 - caching on servlet container, 247–248
- Caching
 - on Apache, 246–247
 - on servlet container, 247–248
- Callbacks
 - Blogger services, 414–415
 - third-party server integration using JSONP, 127
- Calling stack, debugging in Eclipse, 161
- Cascading Style Sheets. *See* CSS (Cascading Style Sheets)
- Challenge, defined, 222
- ChangeTableDialog class, DialogBox widget, 93
- Chat windows
 - building Instant Messenger interface, 430–431
 - MessengerView class handling, 432–436
- ChatWindowView class
 - building Instant Messenger interface, 442–447
 - ContactListView class, 441–442
 - defined, 431
 - MessengerView class, 435–436
- CheckBox widget
 - overview of, 53
 - RadioButton widget vs., 54
- class attribute, GWT-RPC, 133
- Classes
 - anonymous, 194–195
 - Eclipse, 149–153
 - Eclipse Java editor for writing Java code, 153–157
 - implementing code generator, 239
 - Instant Messenger applications, 426–428, 449
 - local, 194
 - RPC interfaces and, 451
- ClassSourceFileComposwerFactory, 239
- cleanString method, instant messenger server, 468
- Click events, Image widgets, 43
- ClickListener
 - creating dynamic interface, 25
 - EditEntryView class, 378–381
 - Hyperlink widget, 441
 - MultiSearchView class, 327
- ClickListenerCollection, 98
- Clients
 - web services client, 363–365
 - writing Action data access layer for, 526–530
- closeChatWindow method, 435, 444
- Code generators, 235–242
 - automatic serialization to JSON, 521–522
 - automatic serialization to XML, 521
 - generate method, 239–242
 - generic GWT code generator, 512–521
 - scaffolding and, 530
 - working with, 235–237
 - writing, 237–239
- Code management, in Ajax applications, 16
- Code syntax coloring, in Java, 153–154
- Collaborator pattern
 - applied to instant messaging, 423–426
 - defined, 34
- Color
 - creating interfaces with CSS, 203–204
 - hex values for standard, 203–204
 - in Java code syntax, 153–154
 - RGB notation, 292

- Column layout, Gadget Desktop, 271–273
- Columned container interface, Gadget Desktop, 264–266
- Comet, using on Tomcat, 472–475
- `com.google.gwt.core.ext`, 33, 237–239
- `com.google.gwt.http.client`, 32
- `com.google.gwt.il8n.client`, 33
- `com.google.gwt.json.client`, 33
- `com.google.gwt.junit.client`, 33
- `com.google.gwt.user.client`, 31
- `com.google.gwt.user.client.ui`, 31
- `com.google.gwt.user.User` module, 166
- `com.google.gwt.xml.client`, 32–33
- `com.google.user.client.rpc`, 32
- `com.google.user.server.rpc`, 32
- Command interface, `MenuBar` widget using, 68
- `CommandBar` widget, 495
- Compilation
 - Ajax application development, 11
 - Ant for application deployment, 189–190
 - deploying applications to web server with, 184
 - generating code during setup. *See* Code generators
 - installing GWT, 142–144
 - organizing application structure, 168
 - sharing compiled applications, 183
- Complex container panels. *See* Panels, complex container
- Complex layout panels, 80–84
- Complex widgets
 - defined, 37
 - `MenuBar`, 67–69
 - `Tree`, 63–67
- Composite widget
 - `BlogEditorView` class extending, 370
 - `BlogView` class extending, 373
 - `DatabaseEditorView` class extending, 489
 - integrating in GWT application, 225–226
 - `MessengerView` class extending, 432
 - `MultiSearchView` class extending, 326
 - overview of, 84–85
 - `SignInView` class extending, 438
 - `SimplePanel` widget vs., 85, 441
 - `TitleCommandBar` widget extending, 389
- Compression
 - on Apache, 249
 - on servlet container, 249–250
- Connections, HTTP request limitations, 108–109
- Constant interface
 - `Dictionary` class vs., 235
 - internationalizing applications with, 227–228
 - managing locales, 232–234
- `Contact` class, `Instant Messenger`, 426–428
- `Contact List` view, `DeckPanel` widget, 77
- `ContactClickListener` class, `Hyperlink` widget, 441
- `ContactList` class
 - `Instant Messenger`, 427–428
 - server connections, 457
- `ContactListView` class
 - chat windows, 436
 - defined, 431
 - `Instant Messenger`, 439–442
 - `MessengerView` class displaying, 432, 433
- Container panels
 - complex. *See* Panels, complex container
 - simple, 84–86
- Container pattern
 - defined, 34
 - Gadget Desktop application, 256–257
 - sample application, 35
- Containers
 - building columned container interface for Gadget Desktop, 264–266
 - CSS styles added to Gadget Desktop containers, 297–298
 - dockable windows and, 283
 - for gadgets, 276–279
- Continuations, using on Jetty, 475–478
- Controller
 - `Blog Editor` application, 390–392
 - `Instant Messenger` application, 447–450
 - `Multi-Search` application, 331–333
- `Cookie` class, 300
- `CookieConstants` class, 302–303
- Cookies
 - client-side storage with, 300–302
 - loading and saving, 303–305
- `CookieStorage` class, 177, 300–302
- create command
 - creating objects, 502–507
 - `Database Editor` application, 495–497
- create method
 - creating `Constants` interface, 227
 - creating `ImageBundle`, 236
 - creating `Messages` interface, 230
 - `DisclosurePanel` widget, 91
- `createCallbackFunction`, JSNI, 217
- `createMessages` flag, localization, 230–231
- Creating, reading, updating, and deleting (CRUD)
 - application processes and, 480
 - HTTP methods, 364
- Cross-browser support, 218–221
- CRUD (creating, reading, updating, and deleting)
 - application processes and, 480
 - HTTP methods, 364

- CSS (Cascading Style Sheets)
 - ChatWindowView class, 447
 - CheckBox widget, 53
 - DialogBox widget, 95
 - FlexTable and Grid widgets, 74–76
 - GadgetContainerView styles, 297–298
 - GWT and, 291–292
 - menu styles, 296
 - MenuBar widget, 68
 - rounding corners of gadgets, 298–299
 - StackPanel widget, 81
 - StackPanel widget, 82–83
 - as standard used for GWT, 14
 - SuggestBox widget, 57
 - tab styles, 293–295
 - ToggleButton/PushButton widgets, 52
 - Tree widget, 66
 - user interfaces on Gadget Desktop, 290–291
 - CSS (Cascading Style Sheets), building elegant interfaces, 200–212
 - connecting GWT widgets to CSS, 200–203
 - overview of, 200
 - sample application, 208–210
 - specifying colors, 203
 - specifying units, 203–205
 - style rules, 201–203
 - using cursor property, 210–212
 - using font properties, 205
 - using margin, border and padding properties, 207–208
 - using text properties, 205–206
 - Ctrl+Click feature, Eclipse Java editor, 155–156
 - cursor property, CSS, 210–212
 - CustomButton class, ToggleButton/PushButton widgets, 49–52
- ## D
- DAOs (Data Access Objects)
 - Actions version for server interface, 523–530
 - asynchronous, 485–488
 - function of, 486
 - mapping to PHP, 522–523
 - overview of, 480
 - REST version for server interface, 538–542
 - Data format libraries, 117–125
 - reading and writing JSON, 122–125
 - reading and writing XML, 117–122
 - Data formats category, GWT library, 32–33
 - Data types, JSON, 123
 - Database Editor application, 479–554
 - Action API built with PHP, 523
 - Actions for server integration, 507–508
 - asynchronous data access objects, 485–488
 - automatic serialization to JSON, 521–522
 - automatic serialization to XML, 521
 - bypassing HTTP request limitations, 108
 - code generator for, 238
 - code generator, writing, 512–521
 - DAO layer, writing Action version, 523–530
 - DAO layer, writing REST version, 538–542
 - dialogs for editing and creating objects, 502–507
 - editor interface, 488
 - extending/loading tree items, 490–494
 - handling back button, 198
 - Hibernate tool for storing the model, 548–553
 - integrating with action-based PHP scripts, 522–523
 - integrating with RPC servlet, 542–543
 - integrating with Ruby on Rails, 530–531
 - Manager pattern, 480–482
 - model for, 482–485
 - overview of, 479–480
 - REST methods for server integration, 508–509
 - RPC for server integration, 509–510
 - RPC service, writing, 543–548
 - Ruby on Rails for building REST API, 531–538
 - serialization of classes, 510–512
 - server integration, 507
 - summary, 554
 - Tree widget and SplitPanel widgets, 489–490
 - workspace views, 494–502
 - DatabaseEditorView class, 489–490, 494–502
 - Debug command. *See also* Debug mode, Eclipse
 - launching application from Eclipse using, 148–149
 - running JUnit test case from Eclipse, 173
 - Debug mode, Eclipse, 158–164
 - adding exception breakpoints, 161–163
 - fixing code while debugging, 163–164
 - setting breakpoints, 158–161
 - testing applications. *see* Testing applications
 - DeckPanel widget
 - MessengerView class implementing, 432–433
 - overview of, 77–78
 - StackPanel widget using, 82
 - DeferredCommand, 243
 - delayTestFinish method, JUnit test case, 175–176
 - delete command, ObjectView class, 499, 501
 - DELETE method
 - REST and, 419–420, 508
 - Ruby on Rails and, 541–542
 - Deploying applications, 183–190
 - automating with Ant, 187–190
 - servlet to servlet container, 184–186
 - to web server, 184

- Design patterns, 256. *See also* individual pattern
 - Design Patterns: Elements of Reusable Object-Oriented Software* (Gamma, Helm, Johnson, Vlissides), 256
 - Desktop application developers, 13–14
 - Desktop applications
 - Ajax applications vs., 5–6
 - GWT widgets modeled after, 15
 - DesktopConstants dictionary, 235
 - DesktopView class
 - defining view for Gadget Desktop, 266
 - HorizontalPanel and FlowPanel used for column layout, 271–273
 - HorizontalPanel widget used for menus, 267–269
 - insertGadget method, 278–279
 - TabPanel widget used for page selection, 269–271
 - VerticalPanel widget used for main layout, 267
 - Development background, evaluating for GWT, 11–14
 - Ajax developers and, 13
 - desktop application developers and, 13–14
 - web application developers and, 13
 - web designers and, 12
 - web site developers and, 12
 - DialogBox widget, 93–95
 - Dialogs, for editing and creating objects, 502–507
 - Dictionary class, 234–235
 - Directory structure, HelloWorld application, 164–168
 - DisclosurePanel widget, 89–91, 329
 - div tag
 - creating dynamic interface, 23–24
 - creating widget based on Google Maps, 225
 - SimplePanel vs. Composite widget, 441
 - Dockable gadgets, 283–284
 - Dockable windows, 283
 - DockableWidget
 - DeferredCommand, 243
 - docking widgets between panels, 284–285
 - dragging widgets using mouse events, 285–288
 - DockPanel widget, 78–79
 - Document class, 119–122
 - DOM (document object model)
 - creating external link widget using, 47
 - creating new widgets with, 213–214
 - user interface package, 31
 - XML implementation of, 521
 - doPost method, Jetty Continuations, 476
 - doSearch method, Button widget, 48
 - doSearchTest method, JUnit test case, 174–176
 - Drag-and-drop gadgets, 283–284
 - Dragging widgets, 285–288
 - Dropping widgets, 288–290
 - Dugg, defined, 8
- ## E
- Eclipse
 - adding projects to, 145–149
 - Ajax application development using, 11
 - creating classes when writing Java code, 149–153
 - Eclipse Java editor for writing Java code, 153–157
 - installing, 144–145
 - internationalizing applications using localization, 230–232
 - Eclipse, debugging in, 158–164
 - adding exception breakpoints, 161–163
 - fixing code while debugging, 163–164
 - setting breakpoints, 158–161
 - Eclipse Editor, 151–152
 - Eclipse Java editor, 153–157
 - edit command
 - editing objects, 502–507
 - ObjectView class, 499, 501
 - EditEntryView class, 378–381
 - Element class, 119–120
 - E-mail, polling protocols, 458–460
 - EmptyCallback class, 457–458
 - Encoded parameter types, Java, 216
 - encoding attribute, HTML form tag, 508
 - EntryPoint interface, 22, 450
 - entry-point tag, organizing application structure, 166
 - Errors
 - creating classes in Eclipse, 151–153
 - internationalizing applications with constants, 228
 - writing Java code using Eclipse Java editor, 154–155
 - escapeHTML method, HTML widget, 42
 - eval function
 - reading and writing JSON, 122
 - using JSNI to extend GWT, 217
 - Event-based protocols, RPC events, 460–462
 - ExampleFields interface, 241–242
 - ExampleGenerator, 241–242
 - Exception breakpoints, adding, 161–163
 - execute method
 - IncrementalCommand, 243
 - MenuBar widget, 68

Extensible Markup Language. *See* XML (eXtensible Markup Language)
Extensible Stylesheet Language Transformations (XSLT), 354–356
External resources, creating link to, 47

F

Feed aggregators, 320. *See also* Aggregator pattern
Feeds, types of, 343
FileUpload widget, 62
Filter interface, 247–248
findDropContainer method, 289
Find/Replace command, in Eclipse Java Editor, 155–156
finishTest method, JUnit test case, 175
fireClick method, ClickListenerCollection, 98
Flash, 6–7
FlashPanel widget, cross-browser support, 219–221
Flex, 6–7
FlexTable widget, 73–76, 441
Flickr, 348–353
FlowPanel widget
 column layout for Gadget Desktop application, 271–273
 HorizontalPanel widget vs., 71
 overview of, 70–71
FocusPanel widget, 86
Fonts, CSS properties, 205–206
Form encoding, 113–114
form tag, HTML, 87
Form widgets, 47–62
 Button, 47–49
 Checkbox, 53
 defined, 37
 FileUpload, 62
 Hidden, 62
 ListBox, 54–55
 overview of, 47
 PasswordTextBox, 59–60
 PushButton, 49–52
 RadioButton, 54
 RichTextArea, 61–62
 SuggestBox, 56–58
 TextArea, 60
 TextBox, 58–59
 ToggleButton, 49–52
Formats
 asynchronous HTTP requests, 107
 FlexTable widget, 75

 number and value, creating interfaces with CSS, 203–205
 reading and writing JSON, 122–125
 reading and writing XML, 117–122
 RichTextArea widget, 62
FormPanel class, 62
FormPanel widget
 integrating with traditional server-side technologies, 114–116
 overview of, 87–89

G

Gadget class, 261–262
Gadget Desktop application, 255–318
 Abstract Factory pattern, 258–259
 adding gadgets to view, 274–275
 benchmarking cookie test, 177–179
 browser interoperability and, 302–303
 buttons, 280–282
 column layout, 271–273
 columned container interface for, 264–266
 Container Application pattern, 256–257
 container for gadgets, 276–279
 cookies, 303–305
 CookieStorage class, 300–302
 CSS for cleaning up user interfaces, 290–291
 CSS styles added to menu, 296, 297–298
 CSS styles added to tabs, 293–295
 CSS styles for rounding corners of gadgets, 298–299
 CSS with GWT, 291–292
 docking widgets between panels, 284–285
 drag-and-drop gadgets, 283–284
 dragging widgets using mouse events, 285–288
 dropping widgets, 288–290
 Gadget class, 261–262
 GadgetClass class, 259–260
 Google Ajax API integrated with, 311–313
 Google Gears for storage, 306–309
 Google Maps API integrated with, 309–311
 ImageBundle, 236, 245
 JavaScript API and, 126
 JavaScript libraries for adding functionality to, 309
 main layout, 267
 menu, 267–269
 model for, 257–258
 overview of, 255–256
 page selection, 269–271
 persistence, 300
 RSS news gadget, 316–317

- Gadget Desktop application, *continued*
 - Strategy design pattern in, 219
 - summary, 318
 - user preferences, 262–264
 - view for, 266
 - weather gadget, 313–316
- GadgetClass class, 259–260
- GadgetContainer widget, 274–275
- GadgetContainerView class
 - buttons for Gadget Desktop application, 280–282
 - creating gadget containers, 276–279
 - CSS styles added to Gadget Desktop containers, 297–298
 - dragging widgets using mouse events, 285–288
- GData
 - Atom as basis of, 402
 - Blogger API and, 401–402
 - Google account sign-in and, 403
 - Google Base service and, 344
- GearsStorage class, 306
- generate method
 - implementing code generator, 237–239
 - internationalizing applications using localization, 230–232
 - SerializationGenerator class, 515–517
 - writing, 239–242
- Generated code
 - generate method and, 239–242
 - using, 235–237
 - writing, 237–239
- GeneratorContext, 238
- GET method
 - FormPanel widget, 88
 - HTTP request limitations, 108
 - as REST method, 417
 - server integration, 508
 - server-side technologies and, 112–114
 - in traditional web applications, 15
- getBasicFormatter method, RichTextArea widget, 62
- getCellFormatter method, FlexTable widget, 75
- getChatWindowView method, 435–436
- getContact method, ContactList class, 428
- getContactCount method, ContactList class, 428
- getCurrentUser method, Instant Messenger, 465–466
- getElementDocument method, XML, 118
- getElementText method, XML, 118
- getEvents method, Hanging RPC call, 462
- getEvents method, Instant Messenger, 463–468
- getExtendedFormatter method, RichTextArea widget, 62
- getFlexCellFormatter method, FlexTable widget, 75
- getHTML method
 - HTML widget, 41
 - RichTextArea widget, 61
- getItemText method, ListBox widget, 55
- getModuleName method, JUnit, 172
- getSelectedIndex method, ListBox widget, 55
- getText method, RichTextArea widget, 61
- getValuesAsString method, 303
- getView method, JUnit test case, 174
- Google
 - account sign-in, 403–410
 - downloading Google APIs, 305
 - purchase of Pyra Labs, 401
 - Google Ajax API, 311–313
 - Google Base, 344–348
 - Google Data API. *See* GData
 - Google Gears, 306–309
 - Google Maps API
 - with Gadget Desktop, 309–311
 - using JavaScript APIs with, 126
 - Google Web Toolkit. *See* GWT (Google Web Toolkit)
 - Google Web Toolkit SDK, 17
 - GoogleBaseSearch class, 347–348
 - GoogleFeedRequest class
 - RSS news gadget, 316–317
 - weather gadget, 313–316
 - GoogleMapsGadget class, 311
 - Grid widget, 73–76
 - guess method, creating dynamic interface, 25
 - GWT (Google Web Toolkit), first steps, 3–36
 - emergence of Ajax, 3–5
 - evaluating your background, 11–14
 - GWT applications, 34–36
 - importance of application development skills, 14–16
 - installation, 140–144
 - Quick Tutorial. *See* Quick Tutorial
 - rethinking web applications, 5–8
 - software engineering for Ajax, 8–11
 - toolkit packages, 30–34
 - GWT (Google Web Toolkit), using effectively
 - building new widgets, 212–215
 - creating elegant interfaces. *See* CSS (Cascading Style Sheets), building elegant interfaces
 - generating code, 235–242
 - handling back button, 197–200
 - improving performance, 242–250
 - internationalizing applications. *See* Internationalizing applications
 - providing cross-browser support, 218–221
 - using asynchronous programming, 191–196

- using JavaScript Native Interface, 215–218
 - using other JavaScript libraries, 221–226
 - gwt-HTML, 41
 - GWT.create, 136
 - gwt-Label, 39
 - GWT-RPC
 - deploying servlet to servlet container, 184–186
 - problems using Continuations with GWT-RPC, 475
 - working with, 131–137
 - GWT-RPC, Instant Messenger application
 - overview of, 450–452
 - RemoteService interface, 452–453
 - RemoteServiceServlet class, 453
 - using asynchronous interface, 454–456
 - gwt-servlet.jar file
 - deploying servlet to servlet container, 186
 - installing GWT, 141
 - using GWT-RPC, 134
 - GWTTestCase class, running JUnit test case, 170
 - gwt.typeArgs Javadoc annotation, 463
 - gwt-user.jar file
 - installing GWT, 141
 - overview of, 30–31
 - using modules, 180
 - .gz extension, for compression on servlet container, 250
 - gzip
 - compressing JavaScript files, 249
 - compression on servlet container, 249–250
- ## H
- Hand pointer cursor, CSS, 211–212
 - handleEvent method, Instant Messenger application, 464–465
 - Hanging RPC call, 462–465
 - Hangman application, Ajax. *See* Quick Tutorial
 - HasFocus interface, Button widget, 48
 - Hash values, caching on Apache, 246–247
 - HashMap, 371
 - HasHorizontalAlignment interface, 40
 - HasHTML interface, 41
 - HasName interface, 87
 - HasWordWrap interface, 40
 - Hello World gadgets
 - adding to Gadget Desktop, 279
 - creating, 274–275
 - HelloWorld application
 - adding projects to Eclipse, 145–149
 - creating while installing GWT, 142
 - organizing structure of, 164–168
 - HelloWorld class, 274–275
 - HelloWorldView class, 153
 - Hibernate tool, storing Database Editor model, 548–553
 - Hidden widget, 62
 - Hierarchical structure
 - of modules, 182
 - using Tree widget to show, 63
 - History feature, browser back button and, 197–200
 - HistoryListener interface
 - back button and, 199
 - Hyperlink widget implementing, 46
 - HorizontalPanel widget
 - column layout for Gadget Desktop, 271–273
 - Eclipse Java editor used with, 154–155
 - menu layout for Gadget Desktop, 267–269
 - MultiSearchView, 324
 - overview of, 71–72
 - TitleCommandBar widget and, 390
 - HorizontalSplitPanel widget
 - DatabaseEditorView class, 489–490
 - overview of, 72–73
 - Hosted mode, GWT, 19–20
 - HTML (HyperText Markup Language)
 - action attribute for HTML forms, 507
 - building interfaces. *See* CSS (Cascading Style Sheets), building elegant interfaces
 - creating dynamic interface, 22–27
 - CSS styles and, 292
 - custom button settings, 52
 - customizing generated GWT application, 20–22
 - forms. *See* Form widgets
 - RichTextArea widget and, 383
 - server integration with plain text and, 109–111
 - XML vs., 117
 - HTML anchor, 197–200
 - HTML widget
 - Google Maps API extending, 310
 - HTMLPanel widget vs., 79–80
 - overview of, 41–42
 - HTMLPanel widget, 79–80
 - HTMLTable superclass, 73–74
 - HTTP (HyperText Transfer Protocol)
 - REST as extension of, 364
 - RPC technologies used over, 509
 - in traditional web applications, 15
 - web browsers and, 334
 - HTTP asynchronous requests
 - basic server integration with, 106–109
 - Hangman tutorial example, 27–29
 - RequestBuilder class, 393–394
 - HTTP proxy
 - cross-domain RequestBuilder for proxy servlet, 394–397
 - overview of, 392–394

- HTTP proxy, *continued*
 - third-party server integration using, 128
 - writing proxy servlet, 397–401
 - HTTPProxy class, 397–401
 - HTTPProxyRequestBuilder class, 394–397
 - HTTPRequest class, 28
 - HttpServlet, 476
 - URLConnection class, 400
 - Hyperlink widget
 - building, 329–330
 - ContactListView class, 441
 - handling back button, 198–199
 - overview of, 44–47
 - HyperText Markup Language. *See* HTML (HyperText Markup Language)
 - HyperText Transfer Protocol. *See* HTTP (HyperText Transfer Protocol)
- I**
- IDE (Integrated Development Environment), Flex using, 6
 - il8nCreator script
 - installing GWT, 141
 - internationalizing applications using, 230–232
 - Image widget
 - creating dynamic interface, 24–25
 - overview of, 42–44
 - using image bundles, 246
 - ImageBundle
 - decreasing application load time with, 11
 - generated code, 235–237
 - images as toolbar buttons, 383–386
 - improving performance using, 244–246
 - including resources with reusable modules using, 183
 - supplying images to title bar of Gadget Desktop application, 280–282
 - Tree widget using, 65
 - writing code generator and, 237–239
 - Images
 - adding to PushButton widget, 236–237
 - setting for custom buttons, 52
 - for title bar of Gadget Desktop application, 280–282
 - Tree widget using, 65–66
 - using image bundles to improve performance, 244–246
 - Importing
 - creating classes in Eclipse, 152
 - GWT project into Eclipse, 146–147
 - writing Java code using Eclipse Java editor, 154–155
 - IncrementalCommand class, 193
 - IncrementalCommand interface, 243–244
 - Inheritance, 182
 - initMap method, Google Map, 225
 - initWidget method, Composite widget, 85
 - initWidget method, Composite widget, 225–226
 - Inline elements, 71
 - Inline tags, CSS, 207
 - input tag, HTML
 - CheckBox widget wrapping, 53
 - FileUpload widget encapsulating, 62
 - TextBox widget wrapping, 58
 - InputListener class, ChatWindowView class, 444–445
 - InsertGadget method, DesktopView class, 278–279
 - Instant Messenger application, 423–478
 - adding RPC events, 458–465
 - ChatWindowView class, 442–447
 - Collaborator Application pattern, 423–424
 - connecting to server, 456–458
 - ContactListView class, 439–442
 - controller, 447–450
 - deploying servlet to servlet container, 186
 - design, 425–426
 - GWT-RPC and, 132, 134–137, 450–456
 - MessengerView class, 432–436
 - model for, 426–429
 - overview of, 429–431
 - server for, 465–470
 - SignInView class, 436–438
 - structure of, 167
 - Instant Messenger application, advanced server IO, 470–478
 - Comet on Tomcat, 472–475
 - Continuations on Jetty, 475–478
 - overview of, 470–472
 - Integrated Development Environment (IDE), Flex using, 6
 - Interfaces. *See also* User interface library
 - building columned container interface for Gadget Desktop, 264–266
 - cleaning up, 290–291
 - creating dynamic, 22–27
 - listeners, 325
 - relationships between RPC classes and, 451
 - two-paned interface for Database Editor, 488
 - Interfaces, CSS for elegant, 200–212
 - color specification, 203
 - connecting GWT widgets to CSS, 200–203
 - cursor property, 210–212
 - font properties, 205
 - margin, border and padding properties, 207–208
 - overview of, 200

- sample application, 208–210
 - text properties, 205–206
 - unit specification, 203–205
 - Internationalizing applications, 226–235
 - declaring constants, 227–228
 - declaring messages, 229–230
 - defined, 226
 - dictionary for, 234–235
 - extension package for, 33
 - localization tools, 230–232
 - managing locales, 232–234
 - overview of, 226–227
 - RichTextToolBar, 383–386
 - utility class package for, 33
 - IntRange constructor, for benchmarking code, 178
 - IsSerializable interface
 - GWT-RPC using, 135–136
 - Instant Messenger application and, 427
 - RemoteService interface, 452
- J**
- J2EE (Java 2 Enterprise Edition), 7
 - JAR files
 - GWT modules distributed as, 180–181
 - overview of, 30–31
 - Java
 - Ajax application development, 11
 - creating classes when writing Java code, 149–153
 - CSS detaching from Java code, 292
 - Eclipse Java editor for writing Java code, 153–157
 - Hibernate tool, 548
 - signature encodings for types in, 216
 - Java 2 Enterprise Edition (J2EE), 7
 - Java Development Kit (JDK)
 - as GWT requirement, 17, 139
 - installing, 140
 - Java FX, 8
 - Java Runtime Environment (JRE)
 - installing, 140
 - removed from Internet Explorer, 7
 - javac-version, GWT projects, 17
 - java.lang, 33
 - JavaScript
 - code management problem in, 15–16
 - emergence of Ajax, 4–5
 - Same Origin policy and, 335
 - third-party server integration using JavaScript API, 126
 - JavaScript libraries
 - adding functionality to Gadget Desktop, 309
 - Google Ajax API, 311–313
 - Google Maps API, 309–311
 - integrating other libraries into existing code, 221–226
 - RSS news gadget, 316–317
 - weather gadget, 313–316
 - JavaScript Native Interface. *See* JSNI (JavaScript Native Interface)
 - JavaScript Object Notation. *See* JSON (JavaScript Object Notation)
 - java.util, 33
 - JClassType instance, 239, 241
 - JDK (Java Development Kit)
 - as GWT requirement, 17, 139
 - installing, 140
 - Jetty, using Continuations on, 475–478
 - Join tables, 485
 - JRE (Java Runtime Environment)
 - installing, 140
 - removed from Internet Explorer, 7
 - JRE emulation category, GWT library, 33
 - JSNI (JavaScript Native Interface)
 - accessing JavaScript methods, 310
 - defined, 10
 - integrating other libraries with, 221–226
 - working with, 215–218
 - JSON (JavaScript Object Notation)
 - automatic serialization to, 521–522
 - browser importing structured data formats, 334
 - Google Ajax API for JSON feeds, 311
 - JSONP for loading JSON feeds, 336–339, 354–358
 - libraries, 10
 - reading and writing, 122–125
 - transforming XML to, 354–356
 - JSONP (JSON with Padding)
 - loading JSON feeds, 336–339
 - third-party server integration using, 127–128
 - JSONRequest class, 127
 - JSONRequestHandler Java interface, 218
 - JUnit
 - Ajax application development using, 11
 - benchmarking code through, 176–179
 - testing applications using, 170–176
 - utility class package for, 33
 - junitCreator script
 - benchmarking code through JUnit, 176–179
 - installing GWT, 141
 - running JUnit test case, 170–173

K

- KeyboardListener interface
 - ChatWindowView class, 444–445
 - MultiSearchView class, 327
 - SignInView class, 438
- KeyboardListenerAdapter
 - implementing in GWT, 97
 - InputListener extending, 445

L

- Label widget
 - creating dynamic interface, 24
 - creating titles, 388
 - HTML widget extending, 41
 - hyperlinks and, 330
 - overview of, 38–41
- Latency, e-mail, 459
- Layout, Gadget Desktop, 267
- Layout panels
 - complex, 80–84
 - simple. *See* Panels, simple layout
- Length units, CSS, 205
- Letters widget, 24
- Library, GWT, 30–31
- Link widget
 - BlogView class, 373
 - building, 329–330
- Links
 - handling back button, 198
 - using Hyperlink widget, 44–47
- ListBox widget, 54–55
- Listeners, 325
- Listview class, 495–497
- Load events, Image widget, 43–44
- load method, cookies, 304–305
- LoadingPanel widget, Blog Editor application, 386–388
- LoadListener interface, Image widget, 44
- Local class, asynchronous programming and, 194
- Locales, managing, 232–234
- Localization tools, 230–232
- Logical bugs, in JavaScript, 170

M

- Manager pattern
 - Database Editor application, 480–482
 - defined, 35
 - sample application, 35
 - two-paneled interface for Database Editor, 488
- Many-to-many relationships, 485
- Margins, CSS properties, 207–208
- Mashups, 183
- MD5 hash function, 222–226
- me attribute, ContactList class, 427
- MenuItem class, 67–68
- Menus
 - CSS styles added to, 296
 - Gadget Desktop, 267–269
- message attribute, 427
- Message class, Instant Messenger application, 426–427, 429
- Messages interface
 - Constants interface vs., 229–230
 - internationalizing applications with, 229–230
 - localization and, 230–231
 - managing locales, 232–234
- Messenger class, Instant Messenger application, 449–450
- MessengerService interface
 - asynchronous interface, 454–456
 - Comet on Tomcat and, 474
 - GWT-RPC and, 451
 - implementing instant messenger server, 465–470
 - implementing RPC events, 463–465
 - RemoteService interface, 452–453
 - RemoteServiceservlet class, 453
- MessengerServiceAsync interface
 - asynchronous interface, 454–456
 - GWT-RPC and, 451–452
 - implementing RPC events, 463–465
- MessengerServiceClientImpl class
 - GWT-RPC and, 451
 - implementing RPC events, 463–465
 - Instant Messenger application, 450
 - overview of, 456
- MessengerServiceImpl class
 - GWT-RPC and, 452
 - implementing instant messenger server, 465–470
- MessengerView class
 - building Instant Messenger interface, 432–436
 - ChatWindowView class, 444–447
 - defined, 431
- MessengerViewListener interface, 433
- method attribute, HTML form tag, 508
- Microsoft Silverlight, 7
- Mock objects, JUnit test case, 175
- mod_header module, caching on Apache, 247

- Modeling. *See* MVC (model-view-controller) architecture
- Model-view-controller architecture. *See* MVC (model-view-controller) architecture
- Modules, GWT
- creating reusable, 181–182
 - managing locales, 234
 - sharing compiled application (mashups), 183
 - using, 180–181
- Mouse events
- dragging widgets, 285–288
 - dropping widgets, 288–290
 - Image widget supporting, 43
- Move cursor, CSS, 210–211
- movies.txt file, 28
- Multimedia eXtensible Markup Language (MXML), 6
- Multiple document view, Blog Editor application, 367–369
- Multi-Search application, 319–359
- Aggregator pattern, 319–321
 - Amazon Search integration with, 354–358
 - controller for, 331–333
 - designing multi-search capability, 321–322
 - Eclipse running JUnit test case, 170–176
 - Flickr integration with, 348–353
 - Google integration with, 344–348
 - handling back button, 198
 - JSNI extending GWT, 217
 - JSON and, 125
 - JSONP for loading JSON feeds, 336–339
 - JSONP for server integration, 127
 - Link widget for, 329–330
 - modeling, 167, 322–323
 - MultiSearchView class, 325–328
 - overview of, 319
 - Same Origin policy and, 334–336
 - SearchResultsView class, 328–329
 - structured data formats and, 334
 - summary, 358–359
 - view for, 324
 - Yahoo! Search integration with, 340–344
- MultiSearchView class, 324–328
- MultiSearchViewListener, 325
- MultiWordSuggestOracle, SuggestBox widget, 56–57
- MVC (model-view-controller) architecture
- Blog Editor application, 365–367, 369
 - Database Editor application, 482–485
 - Gadget Desktop application, 257–258
 - Instant Messenger application, 426–429
 - Multi-Search application, 167, 322
 - organizing application structure, 167
- MXML (Multimedia eXtensible Markup Language), 6
- mysql_query method, SuggestBox widget, 58
- ## N
- name attribute, Contact class, 427
- Name-value pairs, constants for internationalization, 227–228
- New Class icon, Eclipse, 149
- new entry command, 373
- New Java Class dialog, Eclipse, 149–152
- NIO connector, 475
- no-arg constructor, Serializable interface, 427
- notify method, observer pattern, 96
- notifyAll method, instant messenger server, 467–469
- Number formats, CSS, 203–205
- ## O
- ObjectDAO interface, 486, 538–539
- ObjectFactory, 487, 493
- Objects
- BaseObject class, 484
 - dialogs for editing and creating, 502–507
- ObjectView class, 495, 499, 501
- Observer pattern
- event-based protocols benefiting from, 460–462
 - implementing user interface events with, 95–100
 - making server calls using HTTP library, 107–108
- onBrowseEvent method, creating new widgets, 215
- onClick method
- CheckBox widget, 53
 - ClickListenerCollection, 98
 - creating dynamic interface, 25
- onCompletion method, asynchronous HTTP and, 28
- onError method, Image widget, 44
- onFailure method
- asynchronous interface, 454, 456
 - GWT-RPC, 135
- onHistoryChanged method
- handling back button, 199
 - Hyperlink widget, 46
- onKeyPress method, ChatWindowView class, 445
- onLoad method, Image widget, 44
- onModuleLoad method
- asynchronous HTTP and, 28

- onModuleLoad method, *continued*
 - creating dynamic interface, 24
 - customizing generated application, 22
 - Instant Messenger application controller, 450
 - onMouseMove method, 243, 288–289
 - onResponseReceived method, asynchronous
 - HTTP requests, 107–108
 - onSendMessage method
 - connecting to server, 457
 - MessengerView class, 433
 - onSignIn method
 - connecting to server, 457
 - MessengerView class, 433
 - SignInView class, 438
 - onSignOut method
 - connecting to server, 457
 - MessengerView class, 433
 - SignInView class, 434
 - onSuccess method
 - connecting to server, 457
 - GWT-RPC, 135
 - GWT-RPC using asynchronous interface, 454–455
 - onTreeItemSelected method, Tree widget, 67
 - onTreeItemStateChanged method, Tree widget, 67
 - onWindowClosed method, MessengerView class, 434
 - onWindowClosing method, MessengerView class, 434
- P**
- Padding, CSS properties, 207–208
 - Pages, tab-based selection of gadget pages, 269–271
 - Panels
 - docking widgets between, 284–285
 - functionality of, 37–38
 - overview of, 9
 - Panels, complex container, 87–95
 - DialogBox widget, 93–95
 - DisclosurePanel widget, 89–91
 - FormPanel widget, 87–89
 - PopupPanel widget, 91–92
 - Panels, complex layout, 80–84
 - Panels, simple container, 84–86
 - Panels, simple layout, 70–80
 - DeckPanel widget, 77–78
 - DockPanel widget, 78–79
 - FlexTable widget, 73–76
 - FlowPanel widget, 70–71
 - Grid widget, 73–76
 - HorizontalPanel widget vs., 71–72
 - HorizontalSplitPanel widget, 72–73
 - HTMLPanel widget, 79–80
 - VerticalPanel widget, 71–72
 - VerticalSplitPanel widget, 72–73
 - Parsing data
 - JSON data, 123
 - with XML library, 117–118
 - Passwords, for secure hash function, 222–226
 - PasswordTextBox widget, 59–60
 - path attribute, GWT-RPC, 133
 - PATH variable
 - adding GWT installation path to, 17–18
 - installing GWT on, 141
 - PendingRequest class, Comet on Tomcat, 473–474
 - Percent format, 203–205
 - Performance improvement, 242–250
 - caching on Apache, 246–247
 - caching on servlet container, 247–248
 - compression on Apache, 249
 - compression on servlet container, 249–250
 - handling long processing, 242–244
 - ImageBundle for, 244–246
 - Persistency
 - browser interoperability and, 302–303
 - cookies, loading and saving, 303–305
 - CookieStorage class, 300–302
 - Google Gears for storage, 306–309
 - maintaining state, 300
 - PHP scripts, 522–523
 - Action API built with, 523
 - integrating with traditional server-side technologies, 112–116
 - SuggestBox widget, 57–58
 - Pixel format, 203–205
 - Plain text, server integration with HTML and, 109–111
 - Polling protocols, RPC events, 458–460
 - PopupPanel widget
 - DialogBox widget extending, 93–95
 - overview of, 91–92
 - POST method
 - FormPanel widget, 88
 - HTTP proxy and, 397
 - HTTP request limitations, 108
 - as REST method, 417
 - server integration and, 508
 - server-side technologies and, 112–115
 - prefetch method, Image widget, 44
 - PrintWriter, 238
 - Progress cursor, CSS, 211–212
 - projectCreator script, 141, 145–149

- Proxy servlet
 - cross-domain `RequestBuilder` for, 394–397
 - overview of, 392–394
 - writing, 397–401
 - PushButton widget
 - adding image to, 236–237
 - buttons for Gadget Desktop application, 280–282
 - `ImageBundle` and, 246
 - overview of, 49–52
 - PUT method
 - as REST method, 417
 - Ruby on Rails and, 541–542
 - server integration and, 508
 - Pyra Labs, 401
- Q**
- Quick Tutorial, 16–28
 - creating dynamic interface, 22–27
 - customizing generated application, 20–22
 - starting GWT project, 17–20
 - using asynchronous HTTP, 27–29
- R**
- RadioButton widget, 54
 - READ events, Comet on Tomcat, 472
 - Refactor menu, 156–157
 - `register.php` script, 114–116
 - Relational databases, mapping Java classes and relationships to, 548
 - Remote procedure calls. *See* RPC (remote procedure calls)
 - `RemoteService` interface, 452–453
 - `RemoteServiceServlet` class
 - GWT-RPC, 133, 451
 - overview of, 453
 - `removeStyleName` method, connecting GWT widgets to CSS, 201
 - `replace-with` element, cross-browser support, 218
 - Representational state transfer. *See* REST (representational state transfer)
 - `RequestBuilder` class
 - HTTP, 393
 - for proxy servlet, 394–397
 - `RequestCallback` interface
 - asynchronous HTTP requests, 107–108
 - Blogger services, 411–412, 414–415
 - `requestSuggestions` method, `SuggestBox` widget, 57–58
 - `resetChatWindowPositions` method, `MessengerView` class, 434–436
 - `ResponseTextHandler` interface, asynchronous HTTP, 28
 - REST (representational state transfer)
 - Amazon Search and, 354–356
 - Atom and, 402
 - building REST API with Ruby on Rails, 531–538
 - as extension to HTTP, 364
 - Flickr searches and, 348
 - Google Base service and, 344
 - overview of, 130–131
 - sample applications, 35–36
 - server integration with, 508–509
 - web services and, 419
 - XML listing of account blogs, 410–413
 - Yahoo! Search and, 340–341
 - `RESTObjectFactory` class, 540
 - Resume button, debugging in Eclipse, 159–160
 - RGB notation, CSS, 203, 292
 - RIAs (Rich Internet Applications), 6
 - Rich text editing, adding to Blog Editor application, 381–382
 - `RichTextArea` widget
 - Blog Editor application, 382–383
 - overview of, 61–62
 - `RichTextToolbar`, Blog Editor application, 383–386
 - `RootPanel` class, Database Editor application, 489
 - `RoundedPanel` class, for rounding widget corners, 298–299
 - RPC (remote procedure calls). *See also* GWT-RPC
 - adding events to Instant Messenger application, 458–465
 - based on stateful servers, 129
 - blog services and, 402
 - Database Editor application integration with, 542–543
 - Instant Messenger application controller and, 449–450
 - server calls library package, 32
 - server integration with, 509–510
 - web services and, 363
 - writing RPC service for Database Editor, 543–548
 - XML-RPC, 129–130
 - RSS feeds
 - aggregators and, 320
 - building RSS news gadget, 316–317
 - user preferences, 262–264
 - Ruby on Rails
 - building REST API, 531–538
 - integrating Database Editor with, 530–531

Rules, CSS, 201–203

Run command, launching applications from
Eclipse, 148, 173

S

Safari Browser, 108

Same Origin policy

 bypassing, 334–336

 overview of, 108

 third-party server integration and, 125

save method, cookies, 304

Scaffolding, Ruby on Rails providing, 530

script tag, JavaScript

 Google Ajax API and, 312

 including JavaScript in module, 182

 integrating other libraries into existing code, 222

 JavaScript APIs and, 126

 server integration using JSONP, 127

 sharing compiled applications, 183

ScrollPane widget

 ChatWindowView class, 444–445

 overview of, 85–86

Search engines. *See also* Multi-Search application

 Amazon Search integration with Multi-Search,
 354–358

 Flickr integration with Multi-Search, 348–353

 Google Base integration with Multi-Search,
 344–348

 Yahoo! Search integration with Multi-Search,
 340–344

Searcher classes

 GoogleBaseSearch class, 347–348

 YahooSearcher class, 342–343

Searches, in Eclipse Java Editor, 155–157

SearchResultsView class, 328–329

Security

 FormPanel widget, 89

 preventing server calls to third-party servers, 108

selectTab method, StackPanel widget, 84

sendMessage method

 instant messenger server, 467–468

 RemoteService interface, 452

SendMessageEvent, Instant Messenger, 464–465,
468

sendRequest method, asynchronous HTTP
 requests, 107

sendResponse method, Comet on Tomcat,
473–474

Sequential programming, 191–192

Serializable interface

 BaseObject class, 484

 extending, 517

 GWT-RPC using, 135–136, 452, 543–548

 Instant Messenger application model, 427

Serialization of classes

 automatic serialization to JSON, 521–522

 automatic serialization to XML, 521

 overview of, 510–512

 writing generic GWT code generator, 512–521

SerializationGenerator class, 514

Server calls category, GWT library, 32

Server integration techniques, 128–137

 Actions, 507–508, 526–530

 basic, 106–111

 data format libraries, 117–124

 GWT-RPC, 131–137

 overview of, 507

 PHP scripts, 112–116

 REST methods, 508–509, 538–542

 RPC, 509–510

 stateful servers and, 128–129

 stateless servers and, 130

 third-party, 125–128

Servers

 connecting Instant Messenger to, 456–458

 deploying Ajax to Web server, 184

 instant messenger functionality in, 465–470

 stateful, 128–129

 stateless, 130

 writing Action data access layer for server inter-
 face, 526–530

 writing REST data access layer for server inter-
 face, 538–542

Servers, advanced IO, 470–478

 Comet on Tomcat, 472–475

 Continuations on Jetty, 475–478

 overview of, 470–472

Servers, third-party

 restrictions preventing server calls to, 108

 Same Origin policy and, 125–128

 using HTTP proxy, 128

 using JavaScript APIs, 126

 using JSONP, 127–128

Server-side script technologies, 111

ServiceDefTarget interface, GWT-RPC, 136,
451–452, 455

Servlet container

 caching on, 247–248

 compression on, 249–250

 deploying servlet to, 184–186

Session tokens, AuthSub interface, 404–410

Sessions

 defined, 129

 overview of, 466

- setCenter function, JSNI, 225
- setCenterImpl method, JSNI, 225
- setCharacterWidth method, TextArea widget, 60
- setChecked method, CheckBox widget, 53
- setHorizontalAlignment method, Label widget, 40
- setHTML method
 - FlexTable and Grid widgets, 74
 - HTML widget, 41
 - RichTextArea widget, 61
- setLeftWidget method, SplitPanel widgets, 73
- setMethod, FormPanel widget, 88
- setPopupPosition method, PopupPanel widget, 91
- setPosition method
 - ChatWindowView class, 446–447
 - MessengerView class implementing, 435
- setRightWidget method, SplitPanel widgets, 73
- setStyleName method, connecting GWT widgets to CSS, 201
- setText method
 - FlexTable and Grid widgets, 74
 - RichTextArea widget, 61
- setValuesFromString method, 303
- setVisible method, chat window, 435
- setVisibleItemCount method, ListBox widget, 55
- setVisibleLines method, TextArea widget, 60
- setWidth method
 - FlexTable and Grid widgets, 74
 - SimplePanel widget, 85
- setWidth method, widgets, 205
- setWordWrap method, Label widget, 40
- Sharing compiled applications, 183
- show method, PopupPanel widget, 91
- Sign In view, DeckPanel widget, 77
- Signature encodings, for Java types, 216
- signIn method
 - Comet on Tomcat, 475
 - RemoteService interface, 452
 - SignInView class, 438
- SignInCallback class, Instant Messenger application, 457, 463–464
- SignInEvent, instant messenger server, 469
- SignInView class
 - building Instant Messenger interface, 436–438
 - defined, 431–432
 - handling chat windows, 436
 - implementing WindowCloseListener, 434
- SignOffEvent
 - Instant Messenger application, 464–465
 - instant messenger server, 468
- SignOnEvent
 - Instant Messenger application, 464–465
 - instant messenger server, 468–469
- signOut method, RemoteService interface, 452
- SignOutEvent, instant messenger server, 469
- Simple container panels, 84–86
- Simple layout panels. *See* Panels, simple layout
- SimplePanel widget
 - ContactListView class implementing, 441
 - FocusPanel vs., 86
 - LoadingPanel widget and, 388
 - overview of, 85
 - PopupPanel widget vs., 91
 - ScrollPanel widget extending, 85–86
- sinkEvents method, when creating new widgets, 213
- Slashdotted, defined, 9
- SOAP
 - replacing XML-RPC, 130
 - server integration and, 509
 - web services and, 363–364
- Sockets, event-based protocols and, 461
- Software engineering
 - for Ajax. *See* Ajax (Asynchronous JavaScript and XML), software engineering for
 - for desktop applications, 5
- SourceClickEvents interface, Button widget, 48
- SourcesChangeEvent, TextBox widget, 58
- SourcesClickEvents interface
 - Image widget, 43
 - TextBox widget, 58
- SourcesFocusEvents, Button widget, 48
- SourcesLoadEvents interface, Image widget, 43
- SourcesMouseEvents interface, Image widget, 43
- SourceWriter class
 - code generation, 516–520
 - generate method, 239
- Split panels, 72
- SplitPanel widgets, Database Editor application, 489–490
- src/com/gwt examples directory, 164
- StackPanel widget
 - overview of, 80–81
 - VerticalPanel widget vs., 81–82
- Standard Widget Toolkit (SWT), Java, 7
- Standards
 - benefits of, 364–365
 - GWT extensively using CSS, 14
 - GWT interoperation through, 10
 - lack of ActiveX, 7
 - lack of Flash, 7

- startGame method
 - asynchronous HTTP and, 29
 - dynamic interface, 24–25
 - State, maintaining. *See* Persistency
 - Stateful server applications, 128–129
 - Stateless server applications, 130
 - States, button
 - CheckBox widget, 53
 - CustomButton widget, 50–52
 - overview of, 49–50
 - Static widgets, 38–47
 - defined, 37
 - HTML, 41–42
 - Hyperlink, 44–47
 - Image, 42–44
 - Label, 38–41
 - Step Into button, debugging in Eclipse, 160
 - Step Over button, debugging in Eclipse, 160
 - Step Return button, debugging in Eclipse, 160
 - Storage class, Google Gears for storage, 306–309
 - Strategy pattern, cross-browser support using, 218–219
 - String.split() method, asynchronous HTTP, 29
 - Structured data formats, Multi-Search application and, 334
 - SuggestBox widget, 56–58
 - SuggestOracle class, 56–57
 - SWF files, in Adobe Flash and Flex, 6
 - SWT (Standard Widget Toolkit), Java, 7
- T**
- Tab index, FocusPanel widget, 86
 - TabBar widget, StackPanel widget, 82–83
 - table, HTML tags, 73
 - TableListener interface, Grid widget, 75–76
 - TabPanel class, 77
 - TabPanel widget
 - CSS styles added to Gadget Desktop tabs, 293–295
 - overview of, 81–84
 - RoundedPanel class for rounding widget corners, 298–299
 - StackPanel widget using, 83–84
 - tab-based selection of gadget pages, 269–271
 - Tabs, CSS styles added to, 293–295
 - tar command, extracting GWT download with, 140
 - Targets, automating deployment with Ant, 188–189
 - TCP/IP, event-based protocols with, 461
 - Terminate button, debugging in Eclipse, 159–160
 - TestCase class, running JUnit test case from Eclipse, 170
 - Testing applications, 168–179
 - benchmarking, 176–179
 - overview of, 168–170
 - using JUnit, 170–176
 - testSimpleString method, benchmarking and, 178
 - Text
 - aligning in Label widget, 40
 - CSS font properties, 205–206
 - server integration with HTML and plain text, 109–111
 - setting for custom buttons, 52
 - Text class, accessing XML data with, 119–120
 - TextArea widget
 - Blog Editor application, 382
 - overview of, 60
 - TextBox widget
 - ChatWindowView class, 444
 - overview of, 58–59
 - PasswordTextBox widget vs., 58–59
 - Third-party servers
 - restrictions preventing server calls to, 108
 - Same Origin policy and, 125–128
 - using HTTP proxy, 128
 - using JavaScript APIs, 126
 - using JSONP, 127–128
 - TitleCommandBar class, 389
 - TitleCommandBar widget
 - BlogView class, 373
 - ChatWindowView class, 444
 - DatabaseEditorView class, 495
 - EditEntryView class, 380
 - overview of, 388–390
 - ToggleButton widget
 - Gadget Desktop application, 280–282
 - overview of, 49–52
 - Tomcat
 - Comet on, 472–475
 - compression option, 249–250
 - deploying servlet to servlet container, 185–186
 - GWT-RPC with, 133–134
 - proxy servlet running in, 400
 - Toolkit packages, 30–34
 - Tooltips, displaying errors in Java, 152
 - toString method
 - reading and writing JSON, 124
 - searching in Eclipse, 156
 - Tree widget
 - Database Editor application, 489–490
 - extending/loading tree items, 490–494
 - MenuBar widget vs., 67
 - TreeItem class, 64–65, 490–494
 - TreeListener interface, 66–67

Tutorial, GWT
 creating dynamic interface, 22–27
 customizing generated application, 20–22
 starting, 17–20
 using asynchronous HTTP, 27–29
 Type_x_Y.properties, managing locales, 233
 TypeOracle, generate method and, 239, 241

U

UML diagrams, 331. *See also* MVC (model-view-controller) architecture
 URL.encode method, 113
 URL.encodeComponent method, 113
 URLs
 handling back button, 198–199
 linking to, 330
 specifying locale using URL parameters, 234
 using Hyperlink widget, 44–47
 using Image widget, 42–43
 User interface library, 37–103. *See also* Interfaces
 complex widgets, 63–69
 defining, 31
 event interfaces, 95–100
 feature interfaces, 100–103
 form widgets. *See* Form widgets
 overview of, 37–38
 panels, complex container. *See* Panels, complex container
 panels, complex layout, 80–84
 panels, simple container, 84–86
 panels, simple layout. *See* Panels, simple layout
 static widgets. *See* Static widgets
 User preferences, gadgets, 262–264
 UserInfo class, instant messenger server, 466
 UserInfo instance, instant messenger server, 468–469
 Utility classes category, GWT library, 33

V

Value formats, CSS, 203–205
 VerticalPanel widget
 BlogView class, 373
 ChatWindowView class, 444
 as child widget of ScrollPanel, 85–86
 for main layout of Gadget Desktop, 267
 overview of, 71–72
 SignInView class setting, 438
 StackPanel widget vs., 80–81

VerticalSplitPanel widget, 72–73

Views

adding gadgets to view, 274–275
 Gadget Desktop application, 266
 Multi-Search application, 324

W

wait method, instant messenger server, 467
 Weather gadget, building, 313–316
 Web application developers, 13
 Web applications
 adopting Ajax techniques to build, 5
 Database Editor. *See* Database Editor application
 rethinking, 5–8
 traditional vs. GWT, 15
 Web browsers. *See* Browsers
 Web designers, 12
 Web servers, deploying applications to, 184
 Web services. *See also* Blog Editor application
 GWT interoperation through, 10
 interacting with, 393
 Web services client, Blog Editor application, 363–365
 Web site developers, 12
 Web support
 Adobe Flex, 7
 Ant download, 187
 Eclipse download, 144
 Google Web Toolkit SDK download, 17, 140
 hash functions, 222
 hex values for standard colors, 203–204
 Instant Messenger application, 423
 Java resources, 18–19
 JDK download, 17, 140
 Microsoft Silverlight, 7
 ScrollPanel widget methods, 86
 for this book, xxi
 XML data access, 119
 Web technologies, GWT and, 14–15
 WEB-INF directory, 134, 186
 Widgets
 connecting to CSS, 200–203
 creating dynamic interface using, 24–27
 creating new, 212–215
 docking between panels, 284–285
 dragging using mouse events, 285–288
 event interfaces, 95–100
 feature interfaces, 100–103
 Google Map, integrating in GWT application, 224–226

Widgets, *continued*

- margin, border and padding properties, 207–209
- modeled after desktop applications, 15
- overview of, 9–10
- user interface package and, 31

Widgets, types of

- complex, 63–69
- complex container panels. *See* Panels, complex container
- complex layout panels, 80–84
- form. *See* Form widgets
- simple container panels, 84–86
- simple layout panels. *See* Panels, simple layout
- static. *See* Static widgets

WindowCloseListener interface,

- MessengerView class, 432, 434

Window.getClientHeight method,

- ChatWindowView class, 446

Window.getClientWidth method,

- ChatWindowView class, 446

WindowResizeListener interface,

- MessengerView class, 432, 434

Windows, dockable, 283**Word processor, Ajax, 362****Word wrap, Label widget, 40****Workspace pattern**

- Blog Editor application based on, 361–363
- defined, 34
- sample application, 35

Workspaces

- loading in Eclipse, 145–147
- workspace views in Database Editor application, 494–502

Writely, 362**X****XML (eXtensible Markup Language)**

- Ant accepting XML build files, 187
- Atom syndication format for, 402
- automatic serialization to XML, 521
- blog entry listing with, 413–415
- blog listing with, 410–413
- browsers importing structured data formats and, 334
- creating and saving blog entries, 416–419
- Google Ajax API for XML feeds, 311
- GWT interoperation through XML libraries, 10
- JSON vs., 122–125
- reading and writing, 117–122
- server calls package, 32
- transforming XML to JSON, 354–356
- web services and, 363

XMLHttpRequest object

- event-based protocols and, 461–462
- server calls library package, 32
- toolkit packages using, 32

XML-RPC

- overview of, 129
- server integration and, 509
- SOAP replacing, 130

XSLT (Extensible Stylesheet Language Transformations), 354–356**Y****Yahoo! Search, 340–344****YahooSearcher class, 342–343**