Index

A
Abstract classes, 114, 150
Abstract syntax tree (AST), 319–323
Accessibility, 261–262
Actions
  bookmark, 351–352
  Command pattern, JFace, 342–343
  context menus, creating, 121–123
  contributing, 127
  ExcludeTestAction, 224–227
  filtering, 128, 354
  global, 216
  invoking, 26
  menu items to run tests, 39–44
  repository, 346–347
  run, 347
  set apart associations, 228
  sets, 23–24
  Shared ActionContributor, 220
  submenus, 57
Active help, 255
Adapt To IResource Rule, 353
AdapterFactories, 289–294
Adapters
  factories, 288–294
  IAdaptable. See IAdaptable
  pluggable, JFace, 336–340
  resources to Java elements, 308–309
  SelectionAdapter, 212
  workbench-provided, 353–354
Add CVS Repository, 346–347
Anchors, 250–251

APIs
  abstract-based vs. interface-based, 114
  compatibility, preserving, 143
  Debug Model API, 357
  Defensive API Rule, 266
  Explicit API Rule, 266
  ground rules, 272
  plug-ins, publishing, 263–264
  Program to API Contract Rule, 117
  subclassing, 329
Architecture
  contributions in, 15–16
  overview of, 5–7
Arguments, passing to builders, 159
Arrays, configuring builders, 155
assertEditor() helper methods, 211
AST (Abstract syntax tree), 319–323
ASTVisitor, 321–323
asyncExec() method, 209
Auto-test properties
  controls, adding, 165–168
  dialog implementors, finding, 166
  forward pointers, 170
  overview, 161, 169
  property pages, 161–164
  run-time workbench, 168–169
AutoTestBuilder, 179, 375–377
Auto-testing
  builders, installing, 154–159
  editors, excluding tests with. See Editors, excluding tests
Auto-testing, continued
help, 250–251
natures, 154–159
AutoTestPropertyPage, 173

B
Basic widgets, 326, 328. See also Standard Widget Toolkit (SWT)
Batching resource changes, 302–305
Beeping plug-in, 277–278
Binary projects, 11–14
Bindings, AST, 318–319
Black-box reuse, 342
Bookmarks
externalizing strings from code, 258
IResourceAdapter, 351–352
view, 194–195
Bridge file-system resources, 296–299, 310
Browsing, Eclipse source, 12–14
Bruce Anderson’s Architecture Handbook workshop, 3
Bugzilla, 359
build.properties, 20
BuildCommand, 153
Builders
configuring using natures, 154–159
forward pointers, 159
marker resolution, 186–187
order dependency, 152–154
overview, 151
passing arguments to, 159
resource listeners v. builders, 152
results, collecting, 317–319
 Bundling Plug-Ins into features, 83–86

C
Caches, Java element, 311–312
Change Method Signature, 135–136
changeColor() helper method, 209–210
Circle One
closing, 89–90
defined, 4
overview, 33–34
Circle Three
defined, 5
overview, 281–282
Circle Two
closing, 275–280
defined, 4
introducing as new feature, 104–106
overview, 93–94
test-driven plug-in development. See Test-driven plug-in development
Circle Zero, 4
Class creation wizard, 110, 117
Class loaders, 54–55, 265
Class path files, 307
Classes
action invocation, 26
contributor implementation, 224–227
exporting, 264–265
IAdaptable, 286
loading, 16
plug-in, 40
Classpath, 26–27
Code assist engine, 318–319
Collector views, 221
Colors
changing, 113–118
listening to testing progress, 111–113
resetting, 235–238
testing, 231–232
view contribution, 107–111
Commands
BuildCommand, 153
JFace actions as, 342–343
Committers, 2
“Common State,” 131
Compact maps, 132
Compare component, 358
Compile errors, 129–130
Composite
Eclipse components using, 355
Java element trees, 313–314
Standard Widget Toolkit (SWT), 326–330
workspace, 300–301
Compound widgets, 326, 328. See also Standard Widget Toolkit (SWT)
Index

Configurers, 2
Conformance Rule, 26, 48
Content providers
  ContentProvider, 232–233
  outlines, 228
  TableViewer, 195
  TestReportContentProvider, 204–210
Context menus
  creating, 119–124
  extending, 124–128
  forward pointers, 128
  targeting, 126
Context-sensitive help, 251–255
Contributed actions, UI workbench, 346–347
Contributed classes, 276
Contribution Circles
  Circle One, 4, 33–34, 89–90
  Circle Three, 5, 281–282
  Circle Two, 4, 93–94, 275–280
  Circle Zero, 4
    closing, 355–360
    completing, 9, 33
    defined, 3–5
Contribution Rule, 15, 247
Contributions
  actions, 57, 127
  context menu extensions, 124–128
  contributors, 224–228
  declaration/implementation split, 16
  Eclipse architecture, 15–16
  editors, 219–223
  extending, 2–3
  Hello button, 15–16
  invitation to make, 359–360
  JUnit plug-ins, testing, 100–103
  Lazy Loading Rule, 16, 44
  menu. See Menu contributions
  objects, 40, 125, 353
  overcoming inertia in making, 9
  plug-ins, publishing, 86–88
  precision, 45
  Relevance Rule, 43
  RunTestAction, 254–255
  submenus, 57
  views, 107–111
Core runtime, 283–287
Core v. user interface (UI) functionality, 268
Core workspace
  accessing file-system resources, 296–299
  batching changes, 305–306
  composites, 300–301
  Execute Around Method, 302–305
  overview, 295
  resource changes, tracking, 302–305
  resource trees, traversing, 301–302
CoreException, 172–173, 188–190
createImage() helper method, 226
createPartControl() view method, 200
CVS Repository, 346–347, 358

D
Debugging
  components, 356–357
  options, 179–183
  perspectives, 241
Declarations
  Auto-test properties, 162–164
  Declaration Definition (manifest), 17
  extension points, 69–71
  implementation/declaration split, 16
Defensive API Rule, 266
Deltas, resource
  listeners v. builders, 152
  resetting colors, 235–238
  tracking Java element changes, 316–317
  tracking resource changes, 304–305
Der Plan, 35
Design patterns. See Patterns, design
dev.eclipse.org, 359
Dialog implementors, 166
DirtyListener, 234
Diversity Rule, 67
Doc.zip files, 255
Domain objects, 207
Dynamic enablement, menu item behavior, 57
Dynamic registration, listeners, 263

E
Eclipse
  architecture, 5–7
Eclipse, continued
   basics, 3
   design principles, 4
   feature structure, 84
   host, 94–95
   Java development tools (JDT), 5–7
   JUnit. See JUnit
   learning about, 3–4
   levels of applying, 1–3
   plug-in activation, 75
   plug-in architecture, 5
   Plug-In Development Environment (PDE), 5–7
   Run-Time, 94–95, 100
   startup times, 89
   test-driven plug-in development. See
      Test-driven plug-in development
   user programming, enabling, 1
   viewing test results. See Viewing test results
Editors
   Rerun Marker Resolution, 187
   test cases, 210–212
   user interface (UI) workbench, 350–351
   workbench, saving, 187, 192
Editors, excluding tests
   contributing contributors, 224–228
   contributing editors, 219–223
   forward pointers, 228–229
   overview, 217–219
   User Arbitration Rule, 222
   views vs. editors, 221
Element trees
   composite, 313–314
   Extension Point, 270
   Grammar, 270
   scalability issues, 310
   storing resource information, 299
Elements, Java
   (virtual) proxies, 309–313
   Adapters, 308–309
   filtering, 341–342
   resources, 308–309
   ResultView changes, listening to, 232–233
   tracking changes using Observer, 316–317
   trees. See Element trees
Enablers, defined, 2
Error dialog, 173–175
Events, JFace, 210–215
Exception handling
   CoreException, 172–173
   error dialogs, 173–175
   IStatus, 172–173
   logging errors, 175–177
   overview, 171–172
ExcludeTestAction, 224–227
Excluding tests. See Editors, excluding tests
Execute Around Method, 302–305
Explicit API Rule, 266
Explicit Extension Rule, 66–67
Extenders, 2
Extensible Markup Language (XML), 16–17
Extension Interface, 285–287, 294
“Extension Interface,” 285
Extension Object, 285–287, 294
Extension point schemas, 197–199, 268–272
Extension points, 74–75
   declaring, 69–71, 355
   defined, 5
   Diversity Rule, 67
   explicit declaration, 355
   Explicit Extension Rule, 66–67
   Fair Play Rule, 66
   Invitation Rule, 65–66
   listeners, 68–69, 71–74
   loading, 71–74
   markerImageProvider, 142
   overview, 65
   property pages, 162
   publishing, 268–272
   searching for, 40–42
   team component, 358
Extensions
   context menus, 124–128
   contributions, 16
   defined, 5
   IExecutableExtension, 75
   instantiation, 74
   notifying, 77–78
   perspective, 241–244
   test report view, 197
Externalize String Wizard, 259
Factories
- adapters, 288–294
- IElementFactory, 350
Failed tests, 129–130. See also Markers
Fair Play Rule, 66, 120, 122
Feature Plug-Ins, 88
feature.properties file, 280
Features
- bundling plug-ins, 83–86
- publishing plug-ins, 86–87
File-system resources, 296–299
FileWithProperties, 289
FillLayout algorithm, 332
Filters
- actions, 128, 354
- elements, 341
- IActionFilter, 354
- Resource Navigator, 341–342
test result view, 215
tests, 218–219
Finding tests, 147–150
Fixtures, creating with TestProject, 98–100
Flexibility v. performance, 355
FormLayout algorithm, 332
Forward pointers
- accessibility, 262
- adding button to toolbar, 31
- auto-test properties, 170
- builders, 159
- context menus, 128
- editors, excluding tests with, 228–229
- extension points, defining, 74–75
- final, 354–359
- help, 255
- instrumenting, 179–183
- Internationalization, 262
- linked u. copied contents, 14
- marker resolution, 192
- menu contributions, 128
- menu item behavior, implementing, 57
- menu items for running tests, contributing, 44–45
- natures, 159
- perspectives, 244–245
- publishing plug-ins, 88, 272–273
- ResultView, 238
- Test Report View, 215–216
test results, displaying, 63
- tracing, 179–183
- Fragments, 260
- FTP Plug-Ins, 88

getAdapter() method, 287–288, 292
Global actions, 216
Good Fences Rule, 77–78
Grammar, element, 270–271
Granularity, 152
GridLayout
- algorithm, 332
- auto-testing property pages, 167
defining SWT layout strategy, 331

Handles
- resources, 296–299
- virtual proxies, 309–310
Handling events. See Events, JFace
Heartbeat, 152
Hello button
- action implementation, 25–26
- action sets, 23–24
- Conformance Rule, 26
- default implementations, 27
- forward pointers, 31
- Lazy Loading Rule, 29–30
- manifest file, 21
- overview, 15–16
- plug-in project, creating, 18–20
- proxy actions, 29–30
- run-time workbench, 22–24
- Sharing Rule, 23–24
- Spider Navigator, 28–29
HelloAction.java, 30–31
Help
- context-sensitive, 251–255
- forward pointers, 255
- integrated, 250–251
- overview, 247
top-level, 247–250
Hierarchies, type. See Type hierarchies
Host Eclipse, 94–95, 100
Host workbench, 21

I

I*2 Interfaces, 143. See also Interfaces
IActionBars, 214, 216
IActionFilter, 354
IAdaptable
  interfaces, adding to existing types, 289–294
  interfaces, surfacing, 288
  overview, 283–285
UI workbench, 349–354
IBM WebSphere Application Developer, 15
 ICommand, 153
IContainers, 300–301
IContentOutlinePage, 354
IContentProvider, 337–340
IDocuments, 227–228
IDs
  action sets, 23
  manifests, 21
IEditorInput, 211
IExecutableExtension, 75
IFile
  AdapterFactories, 289
  Composite pattern, 300
  proxy and bridge patterns, 297
IJavaElement, 308–310
IJavaProject, deleting markers, 140
ILabelProvider, 201–204, 337–340
Images, 140–141, 159
IMarker
  creating markers, 137
  marker resolution, 142–145
  testing for markers, 132–134
IMemento, 348–351
Implementation
  actions, 25–26
  extensions, 27–29
  interfaces, 27
  Lazy Loading Rule, 29–30
  menu items for running tests, 39
Implementation/declaration split, 16–17
Implementation patterns. See Patterns, implementation.
Incremental builds
  AutoTestBuilder, 159
  marker resolution, 187, 192
Independence, resource listeners v. builders, 152
Infopops
  context-sensitive help, 252
  help, 255
  links, 247
  test results, 253
Information/Detail views, 221
Installing PDE JUnit, 93
Instantiation, extensions, 74
Integrated help, 250–251
Integration in Eclipse/JUnit, 37
Integration Rule
  defined, 129–130
  views in perspective, 240, 252
Interfaces
  I*2, 143
  adding, 289–294
  extension, 285–287, 294
  interface-based APIs, 114
  surfacing using IAdaptable, 288
Internationalization
  forward pointers, 262
  fragments, 260
  overview, 257
  strings, externalizing from code, 258–260
  strings, externalizing from manifest files, 257–258
InterruptedException, 188–190
Invitation Rule, 65–66, 119
InvocationTargetException, 188–190
IParent interface, 313–314, 315
IPersistentElement, 354
IPerspectiveFactory, 244
IPreferenceMonitor, 147–150, 188–191
IProject, 158–159, 164
IPreferenceSource, 288, 353–354
IResource
  Adapters, 308–309
  getParent() method, 301
  proxy and bridge patterns, 296–298
IResource Adapter, 351–352
IResourceProxy, 302
IResourceVisitor, 301
Index

IRunnableContext, 188–191
ISafeRunnable, 75, 77–78
IShowInTarget, 354
IStatus, 172–173
IStructuredContentProvider, 204
ITaskListResourceAdapter, 354
ITestRunListener
  accessibility, 263–264
  Change Method Signature, 136
  creating markers, 136–137
  published API, 268
ITreeContentProvider, 337–340
IType
  defined, 42
  finding tests, 147–150
  menu item behavior, 55
ITypeHierarchy
  avoiding performance penalties, 149
  objectifying associations, 315
  tracking element changes, 317
IViewPart, 110
IWorkbenchAdapter, 351
IWorkspaceRunnable
  batching changes, 305–306
  configuring builders, 157
  deleting markers, 140–141
  error handling, 191–192

J
Java Archive (JAR) files, 16–17
Java core
  abstract syntax tree analysis, 319–323
  Adapters, 308–309
  Builder, 317–319
  element changes, 316–317
  element trees, composite, 313–314
  objectifying associations, 314–315
  Observer, 316–317
  overview, 307
  resources to elements, 308–309
  traversing Java model, 315
  type hierarchies, 314
  Visitor, 319–323
Java development tools (JDT), 5–7, 15. See also Plug-In Development Environment (PDE)
Java editor, 359
Java elements
  (virtual) proxies, 309–313
  Adapters, 308–309
  filtering, 341–342
  resources, 308–309
  ResultView changes, 232–233
  tracking changes, 316–317
  trees. See Element trees
Java model structure, 308–309
JavaUI, 216
JDT (Java development tools), 5–7, 15.
  See also Plug-In Development Environment (PDE)
JDT Debug UI, 357
JFace
  Command pattern, 342–343
  event handling, 210–215
  label decorations, 338–339
  ListenerList, 63
  pluggable Adapter, 336–340
  source documents, 358–359
  Test Report View, 193–195
  TestReportContentProvider, 204–210
  TestReportLabelProvider, 201–204
  TestResult, 195–196
  tests, writing, 196–197
  TextEditor, 229
  user interface framework, 335–336
  viewers, customizing, 341–342
  views, creating, 196–197
JUnit
  Contributed JUnit plug-ins, testing, 100–103
  example, 35–37
  integration, 37
  object invocation, 48
  test results, displaying, 59–60
  test-driven plug-in development, 92–97
  TestProject, creating fixtures with, 98–100
  TestRunner, 62–63, 100
JUnitCore, 264, 268
JUnitPlugin, 134

K
Keyboard access, 261
Index

L
Label providers
  contributing contributors, 227
  TableViewer, 195
  TestReportLabelProvider, 201–204
Labels
  changes, 57
  decorations, 338–339
  defined, 115
  focus, 117
Layout definition strategy, Standard Widget Toolkit (SWT), 330–333
Lazy Loading Rule
  defined, 16, 25
  menu item behavior, implementing, 47, 57
  menu items for running test, 44
  toolbar buttons, adding, 29–30
  UI workbench, 346–347
  verification, 52–53
Libraries
  classes, 264–265
  plug-ins, 96–97
License Rule, 85
Linked contents, 14
Listeners
  BeepingListener, 278
  context menus, 120–122
  Contributed JUnit plug-ins, 100–103
  dynamic registration, 263
  extension point schemas, 270
  extension points, defining, 68–70
  extension points, loading, 71–74
  ListenerList, 63
  progress testing, 111–113
  registering with widgets, 333–334
  resource listeners v. builders, 152–154
  ResultView changes, 232–235, 238
  SelectionListener, 212
  test results, 60–63
Log String technique, 102
Logging errors, 175–177
Long-running operations, invoking. See Marker resolution

M
Manifest files
  Declaration Definition (manifest), 17
  exploring, 21
  extension points, 41
  plugin.xml, 20
  shadow creation, 49
  strings, externalizing, 257–258
Marker resolution
  build and rerun, 186–187
  forward pointers, 192
  overview, 185–186
  progress, 188–191
  rerunning tests, 191–192
  testing, 186
Markers
  configuring builders, 157
  creating, 136–139
  defined, 130
  deleting, 140–141
  images, 141–143
  logging errors, 175
  passing projects, 134–136
  resolution, 142–145
  test failures as, 131
  testing for, 131–134
Memento, 347–351
Menu contributions
  creating context menus, 119–124
  extending context menus, 124–128
  overview, 119
Menu item behavior
  build class path, updating, 51
  forward pointers, 57
  interface implementation, 48–49
  Lazy Loading Rule, 47, 52–53, 57
  overview, 47
  plug-in classes, 54–57
  selection storage, 50–51
  shadow creation, 49
  stub for running tests, 56–57
Menu items for running tests
  forward pointers, 44–45
  Monkey See/Monkey Do Rule, 39–40
Object Contributions, 40–45
overview, 39
Relevance Rule, 43–44
Safe Platform Rule, 44

Menus
actions, 127
context, 127
editors, 229
Run As, 93–95, 97
Show View, 242–244
TaskList, 121
MessageDialog, 173
Metadata, 218
Milestone 3 of Eclipse 3.0 development stream, 93
Mnemonics, 261–262
Monkey See/Monkey Do Rule, 39–40
MultiPageEditorPart, 228

N
Naming Plug-Ins, 21
Natures
auto-test property pages, 168
builders, 151, 154–159
Navigation history, workbench editors, 228
Navigator
IMemento, 349
UI workbench, 354
views, 221
Navigator, Spider. See Spider Navigator
New Dialog, 229
New file wizard, 219
New Repository action, 346–347
NotificationManager, 302–303
Notifying extensions, 77–78
NullProgressMonitor, 148

O
Object Contributions, 40–45, 125
Objectifying associations, type hierarchies, 314–315
Objects, extension, 285–287, 294
Observer
SWT events, 333–334
test results, 59–60
tracking element changes, 316–317
tracking resource changes, 302–305
OOPSLA-93, 3
Open Type dialog, 142–145, 166
Open-in-editor function, 213–214
Option path, tracing, 180–183
org.junit, 96–97
Other Rule, 242
Outline view, 221

P
Package Explorer
auto-test property pages, 164
test points, 41
IResource Adapter, 351–352
Java model structure, 308–309
object contributions, 353
PackageExploreContentProvider, 232–233
perspective, 241
ResultView, 232–233
views, 221
Packages, published v. internal, 266
Packaging Plug-Ins, 79–82
Parents
class loaders, 54
handles, 298
interfaces, 313–314
widgets, 329
Pattern stories
contributing, 281–282
core runtime, 283–287
core workspace. See Core workspace
IAdaptable. See IAdaptable
Java core. See Java core
JFace. See JFace
Standard Widget Toolkit (SWT). See
Standard Widget Toolkit (SWT)
Patterns. See Patterns, design; Patterns, implementation
Patterns, design
Adapter, 337
Bridge, 296, 299
Builder, 317, 318
Command, 342
Composite, 300, 327, 328, 355
Decorator, 339
Extension Object, 285. See also Extension Interface
Façade, 109
Factory Method, 74
Interpreter, 318
Memento, 313, 348
Observer, 59, 60, 303
Proxy, 47, 296, 350. See also virtual proxies
Singleton, 56
Special Case, 148
Strategy, 330, 341
Value Object, 297
Visitor, 301, 319, 320
Patterns, implementation
Collecting Parameter, 148
Common State, 131
Complete Constructor, 329
Execute Around Method, 306
Safe Copy, 303
Self Delegation, 347
Variable State, 131, 330
PDE (Plug-In Development Environment). See Plug-In Development Environment (PDE)
Persistent attribute, 139
Perspectives
Debug, 357
forward pointers, 244–245
Other Rule, 242
overview, 239
Show View menus, 242–244
views, 239–242
Platform
Eclipse architecture, 5–7
Platform UI.getWorkbench, 187, 192
plug-in shadows, 49
Plug-In Development Environment (PDE)
build.properties, 20
debug options, 180
Eclipse architecture, 5–7
Eclipse setup, 11–14
extension point schema files, 74–75
extension points, publishing, 269–272
JUnit, 92–97
site creation wizard, 86–87
TestProject, 98–100
user documentation, 248
workspace, 11–12
plugin.xml
editors, excluding tests, 220
extension points, 41–42
menu items, contributing, 125
plug-ins, redeploying, 278
refactoring and updating, 267
strings, externalizing, 257–258
test report view, 197–198
toolbar buttons, 31
Plug-Ins
accessibility, 257, 261–262
activation, 75
architecture, 5
auto-test properties. See Auto-test properties
beeping, 277–278
builders. See Builders
class, 40
compile errors, 129–130
diagnosing, 69
downloading, 359
Eclipse source, 12–14
editors for excluding tests. See Editors, excluding tests
exception handling. See Exception handling
exporting imported, 273
extension notification, 77–78
extension points. See Extension points
fragments, 260
help. See Help
instrumenting, 179–183
Internationalization. See Internationalization
JFace. See JFace
JUnit, 35–37
library, 96–97
marker resolution. See Marker resolution
markers. See Markers
menu contributions. See Menu contributions
menu item behavior. See Menu item behavior
menu items for running tests. See Menu items for running tests
metadata, 218
natures, 151, 154–159
overview, 33–34
packaging, 79–82
perspectives. See Perspectives
publishing. See Publishing plug-ins
redeploying, 278–280
ResultView. See ResultView
servicing, reducing costs, 177
shadow creation, 49
structure, 16
Test Report View. See Test Report View
test results, 59–63
test-driven. See Test-driven plug-in
development
tests, creating, 98
tests, failed, 129–130
tests, finding, 147–150
toolbar buttons, 18–20
tracing, 179–183
Pop-up menus, 40–43
Preferences
   editors for excluding tests, 217
   pages, 170
Primary features, 88
Program to API Contract Rule, 117
ProgressMonitorDialog, 189
Project order dependency, 152–154, 157
ProjectDescription, 153
Properties
   auto-testing pages, 162–167
   core runtime views, 284–285
   editors for excluding tests, 217–218
   workbench-provided adapters, 353–354
Proxies, virtual. See Virtual proxies
Proxy actions, 29–30
Proxy file-system resources, 296–299
Publishers, 2
Publishing plug-ins
   API, 263–264
   bundling plug-ins into features, 83–86
   contributing, 86–88
   Defensive API Rule, 266
   Explicit API Rule, 266
   exporting classes, 264–265
   extension points, 268–272
   forward pointers, 88, 272–273
   License Rule, 85
   overview, 79, 263
   packaging plug-ins, 79–82
   plugin.xml, refactoring and updating, 267
   separating core and refactoring, 268
   separating published and internal packages, 266
   Stability Rule, 266
   Pull model, 303
   Push model, 303

Q
QuickFix, 114, 142

R
Redeployment, plug-ins, 278–280
References, 379
refresh() method, 206
Relevance Rule, 43–44, 163
Repositories, 346–347
Rerun Marker Resolution. See also Marker
   resolution
   invoking long-running operations, 185–192
   marker generator, 144–145
RerunTest method, 124
Resolution, marker. See Marker resolution
Resources
   batching changes, 302–305
   defined, 130
   deltas. See Deltas, resource
   Execute Around Method, 302–305
   IResourceAdapter, 351–352
   Java elements, 308–309
   proxy and bridge access patterns, 296–299
   resource listeners v. builders, 152–154
   resource trees, traversing, 301–302
   ResultView changes, 232–233
   Responsibility Rule, 171–172
   Result/Output views, 221
   ResultView
      forward pointers, 221
      help, 247
ResultView continued
  observing changes, 231, 232–235
  reacting to changes, 235–238
  testing colors, 231–232
RowLayout algorithm, 332
Rules
  Adapt To IResource, 353
  Conformance, 26, 48
  Contribution, 15, 247
  Defensive API, 266
  Diversity, 67
  Explicit API, 266
  Explicit Extension, 66–67
  Fair Play, 66, 120, 122
  Good Fences, 77–78
  Integration, 129–130, 240, 243
  Invitation, 65–66, 119
  Lazy Loading, 16, 57, 346–347
  License, 85
  Monkey See/Monkey Do, 40
  Other, 242
  Program to API Contract, 117
  Relevance, 43–44, 163
  Responsibility, 171–172
  Safe Platform, 44
  Sharing, 24
  Stability, 266
  Strata, 356
  User Arbitration, 222, 339
  User Continuity, 348
Run action, 347
RunTestAction, 254–255
Runtime
  component, 6
  core, 283–287
Run-Time Eclipse, 100
Run-time kernel, 16, 17
Run-time workbench
  hello action, 26–30
  Plug-In Development Environment (PDE), 21–22
  Run-Time Workbench launching option, 94–95
Run-time workspace
  run-time Eclipse started from, 94–96
  test project fixtures, 98–100
S
  Safe copies, 303
  Safe Platform Rule, 44
  Scope, resource listeners v. builders, 152
  Screen readers, 261
  Searching
    Eclipse source, 12–14
    extension points, 40–42
    org.eclipse.search plug-in, 358
    TestSearcher, 147–149
    traversing Java model, 315
  Selection linking, 215
  SelectionAdapter, 212
  SelectionListener, 212, 333–334
  Self Delegation, 347
  Separators, 122
  Shadow world, 49
  Shared images, test result view, 215
  SharedActionContributor, 220
  Sharing Rule, 24
  Sharing, resource listeners v. builders, 152
  Shortcuts
    editor actions, 228
    perspective contributions, 244
  Sites
    creation wizard, 86–87
    workbench parts, 123
  Smalltalk, defined, 1
  SocketTestRunner details, 368–370
  Sorters, 215
  Sorting viewers, 341
  Source, browsing and searching, 12–14
  Spider Navigator
    Bookmarks view, 194–195
    click function, 120
    mapping between domain objects and SWT widgets, 340
    providing, 28–29
  Stability Rule, 266
  Standard Widget Toolkit (SWT)
    API definition, 114
    beeping plug-in, 277–278
    Composite, 326–330
    example codes, 359
    JFace. See JFace
    layout definition strategy, 330–333
opening test cases in editor, 210–212
overview, 325–326
Plug-In Development Environment
(PDE), 6
posting events, 209
responding to events, 333–334
subclassing, 329
test report view, 200, 207–208
Startup time, 89
Storage, Memento, 348–351
Strata Rule, 356
String literals, updating references, 267
Strings, externalizing
from code, 258–260
from manifest files, 257–258
Structured views, 215
Style bits idiom, 329
Subclassing
Eclipse API, 329
viewer customizing, 341–342
SubProgressMonitors, 192
Supertypes, 149
Surfacing interfaces using IAdaptable, 288
SWT (Standard Widget Toolkit). See Standard
Widget Toolkit (SWT)
syncExec() method, 209

T
Table of contents (toc) trees, 247, 251
Table widget
event handling, 210
opening test cases in editors, 212
TestReportView, 200–201
TableViewer
Bookmarks, 194–195
test report view, 200–201
Target run-time, 180
Targeting context menus, 126
Task List, 121, 135
Team component, 358
Test Report View
creating views, 197–201
event handling, 210–215
forward pointers, 215–216
overview, 193–195
TestReportContentProvider, 204–210
TestReportLabelProvider, 201–204
TestReportView, 199–201
TestResult, 195–196
writing tests, 196–197
Test-driven plug-in development
Contributed JUnit plug-ins, testing,
100–103
overview, 91–92
PDE JUnit, 92–97
Run-Time Workspace, 100
Test-Driven Development (TDD) defined,
91–92
TestProject, 98–100
TestRunner, 103
TestProject
creating fixtures, 98–100
source code, 371–374
test report view, 196–197
TestReportContentProvider, 204–210
TestReportLabelProvider, 201–204
TestReportView, 247
TestResult, 195–196
TestRunner
contributed JUnit plug-ins, 103
details, 363–368
SocketTestRunner details, 368–370
test failures as markers, 135
test project fixtures, 100
test results, 60, 62–63
Tests
excluding. See Editors, excluding tests
failures as markers. See Markers
finding, 147–150
perspectives, 240
plug-ins, 98
TextEditor, 229
Timing operations, tracing, 183
Toolbars
actions, 127
contributing to editors, 229
open-in-editor function, 213–214
Top-level help, 247–250
Tracing
forward pointers, 179–183
Java element, 312–313
Tracing continued
plug-ins, 179–183
ResultView, 237–238

Trees
element, 299, 310
Element Grammar, 270
Extension Point Element, 270
resource deltas, 152
table of contents (toc), 247, 251
traversing resources, 301–302
widget, 327, 330

TreeView, 336–340

Type hierarchies
finding tests, 149
objectifying associations, 314–315
views, 221

Type Hierarchy. See also ITypeHierarchy
IAdaptable, 286–287
IResource, 296
Java element changes, tracking, 316–317
Typed listeners, 333–334

U
UI (user interface). See User interface (UI)
Untyped listeners, 333–334

Updating
build class path, 51
plugin.xml, 267
sites, 86–88
viewers, 206–208
views, 215

User Arbitration Rule, 222, 339
User Continuity Rule, 348

User interface (UI)
Debug UI, 357
Eclipse architecture, 6–7
Lazy Loading Rule, 346–347
object contributions, 353
publishing plug-ins, 268
restoring editors, 350–351
UI v. core functionality, 268
virtual proxies, 346–347

User interface (UI) workbench
IAdaptable, 350–354
object contributions, 353
overview, 345
persisting UI state, 347–351
restoring editors, 350–351
restoring views, 350–351
virtual proxies, 346–347
workbench-provided adapters, 353–354

User programming
Eclipse supporting, 1–3
Smalltalk, 1

V
Value Objects, 297
Variable State, 131
ViewColorTest, 231–232

Viewers
customizing without subclassing, 341–342
object contributions, 125
pluggable Adapter, 336–340
updating, test report view, 206–208

Viewing test results
abstract-based v. interface-based APIs, 114
changing colors, 113–118
contributing views, 107–111
overview, 107
Program to API Contract Rule, 117
testing progress, 111–113

ViewPart, 110

Views
Bookmarks, 194–195
contributing, 107–111
editors v., 221
open-in-editor function on toolbar, 213–214
perspectives, 239–242
Show View menus, 242–244
structured, 215
Test Report. See Test report view
UI workbench, 350–351

Virtual proxies
IResourceProxy, 302
Java elements, 309–313
UI workbench, 346–347

Visitor
abstract syntax tree analysis, 319–323
traversing resource trees, 301–302
W
WebDav Plug-Ins, 88
Widgets. See also Standard Widget Toolkit (SWT)
  accessibility, 207–208
  auto-test property pages, 166–167
  JFace actions, 343
  layout definition strategy, 330
  registering listeners, 333–334
  table, 200–201, 210, 212
  trees, 327, 330
Wizards
  class creation, 110
  creating, 228
  Externalize String, 259
  new file, 219
  Plug-In Project, 18–20
  schemas used in, 269
  showing progress embedded in wizard dialog, 189
  test report view, 197
Workbench
  host v. run-time, 21
  Spider Navigator, 28–29
  page providing access to editor and viewer preferences, 350–351
  Plug-In Development Environment (PDE), 6–7
  Rerun Marker Resolution, 187
  Run-Time, 94–95
  UI. See User interface (UI) workbench views in perspective, 239–242
  WorkbenchContentProvider, 232–233
  Working with v. working on, 11
Workspace
  browsing and searching Eclipse source, 12–14
  core. See Core workspace
  linked contents, 14
  Plug-In Development Environment (PDE), 6
  root, 133
  Run-Time, 94–95
  setting up, 11–12
Wrappers, property-sheet, 289
www.dev.eclipse.org, 93
www.eclipse.org
  accessibility requirements, 262
  builders, 159
  downloading plug-in examples, 359
  Eclipse API ground rules, 272
  Eclipse developer articles, 359
  evolving Java APIs, 272
  perspectives, 245
  preference pages, 170
  resource deltas, 238
  shared images, 215
  SWT design principles, 325
  SWT layout support, 332
  testing internationalized plug-ins, 262
  UI guidelines, 7
www.section508gov/, 262

X
XML (Extensible Markup Language), 16–17

Z
Zip files, help, 255