

Index

A

Accelerators, 48–50, 198
 add() method, 11, 167–168
 add() (to region), 422
 addDisposeListener(), 21
 addFilter(), 93–94
 addImageLoaderListener(), 492
 addListener(), 18, 91
 Advanced controls, 209, 257
 classes TabFolder/TabItem *see* TabFolder/TabItem
 classes Tree/TreelItem *see* Tree/TreelItem
 alpha (ImageData field), 469
 alphaData (ImageData field), 469–470
 append(), 157
 Application data, 122–123
 named/unnamed, 25
 see also Display
 Applications, xx, 511–512
 see also FileExplorer; Minesweeper
 asyncExec(), 101–102
 from user interface thread, 106, 107c, 108
 queue code for user interface thread example, 103–104
 TableItems creation in an “idle handler”
 code example, 245c–246c
 AWT/Swing, xxvii–xxix

B

Basic controls, 135, 179
 class button *see* Button

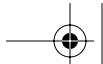
 class combo *see* Combo
 class label *see* Label
 class list *see* List
 class text *see* Text

beep(), 127
 BIDI *see* Bidirectional languages
 Bidirectional languages (BIDI), support, 458–459
 Bounds, 65–66, 65f, 112–113
 Button, 132
 class button
 alignment, 142
 arrow buttons, 145–146
 button events, 140–141, 146–147
 button styles, 140
 check buttons, 142–143
 example, 140
 hierarchy, 140
 push buttons, 142
 radio buttons, 144–145, 146–147
 text and images, 141
 toggle buttons, 145
 default button, 290
 bytesPerLine field (image), 464

C

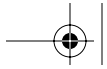
Canvas, 132, 302
 tab group traversal implementation example, 45–46
 see also Class Canvas





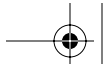
- Caret, 132, 148, 296–297, 302
 - see also* Class Caret
- Character, 33
 - Convenience Constants, 33t
 - generated control characters, 34
- Character encoding, 439
- check()**, 525
- checkItems()** method, 219
- checkPath()** method, 219
- checkSubclass()**, 15–16
- Class Canvas
 - events, 296
 - example, 295
 - hierarchy, 295
 - painting in a Canvas, 296
 - scrolling, 297–298
 - styles, 296
 - see also* Caret
- Class Caret
 - bounds/size/location, 298–299
 - events, 298
 - example, 298
 - hierarchy, 298
 - Input Method Editor font, 300
 - styles, 298
 - visibility, 299–300
 - writing single-line text editor code example, 300c–302c
- Class Composite
 - events, 278
 - example, 277
 - getting children, 278
 - hierarchy, 277
 - and Layout, 279
 - styles, 277
 - see also* Tab traversal
- Class Control
 - context menus, 275
 - cursors, 276–277
 - events, 274
 - example, 273
 - foreground/background/font, 275–276
 - hierarchy, 273
 - parent/shell/monitor, 275
 - styles, 274
 - tool tips, 276
- Class Group
 - events, 279–280
 - example, 279
 - hierarchy, 279
 - styles, 279
 - text setting, 280
- Class Shell
 - closing a Shell, 285
 - creation/no trim coding example, 287, 287c–288c
 - creation/nonstandard trim, 286–287
 - creation/standard trim, 286
 - default button, 290
 - events, 282, 292–294
 - example, 280
 - hierarchy, 281
 - menu bar setting, 285
 - minimizing/maximizing/restoring, 289–290
 - modality, 289
 - nonrectangular, 290–291
 - coding example, 291c–292c
 - opening a Shell, 283–284
 - styles, 281
 - text and image methods, 282–283
- Class SWT, 10
- Classes ScrollBar and Slider
 - configuring options, 268–269
 - events, 265–266, 269–272
 - example, 265
 - hierarchy, 265
 - historical background, 266
 - increment/page increment, 268
 - range operations, 267
 - selection and thumb, 267–268
 - styles, 265
- clearSelection()**, 155
- Client area, 66–67, 67f, 113
- close()**, 93, 285
- Color()**, 430
- Colors, 427
 - Class Color, 430–433
 - coding example allocating random colors in clipping region, 432c–433





- preallocation issues, 468
 - System Colors, 433–434
 - constants, 433t–434t, 435t, 436
 - GRADIENT versions, 435*see also* RGB/Class RGB
 - Combo, 132
 - box, 174
 - class combo, 175
 - drop-down combo resizing, 177
 - events, 174–175, 177–178
 - example, 173
 - hierarchy, 174
 - list methods, 176–177
 - styles, 174
 - text methods, 175–176
 - Common Widgets, xxvi–xxvii
 - computeSize()**, 77, 307
 - Constructors
 - convenience constructors, 11
 - use of operating system resources, 8
 - Consuming* a key, 28
 - coding example, 38–39
 - Containment, 273
 - Containment* hierarchy, 14
 - contains()**, 408, 423
 - Context menu request, 59–60
 - Contextual forms, 458–459
 - Control*, 16, 130, 294
 - see also* Advanced controls; Basic controls; Class Composite; Class Control; Class Group; Class Shell; Draggable controls; Menu/MenuItem; Range-based controls; ToolBar/ToolItem
 - Control** (Drawable interface), 410, 478
 - Control fundamentals, 65, 88
 - enabling/disabling, 74–75
 - preferred size, 76
 - computing, 77
 - packing, 76–77
 - visibility, 72–73
 - z-order, 74
 - see also* Bounds; Client area; Moving/Resizing; Painting
 - Control.setMenu()**, 14
 - CoolBar/CoolItem, 132
 - Class CoolBar/CoolItem
 - and clipped content code example, 309c–311c
 - CoolBar events, 304
 - CoolBar style, 303
 - CoolItem configuring, 305–307
 - CoolItem events, 304, 312–313
 - CoolItem styles, 304
 - example, 303
 - hierarchies, 303
 - locking CoolBar, 312
 - saving/restoring CoolBar layout, 311–312
 - searching operations, 312
 - CoolBar resize and control repositioning
 - coding example, 308c–309c
 - create cool bar with four tool bars coding example, 307c–308c
 - “item children” (cool items) and “gripper,” 304
 - rebar controls (cool bars), 304
 - “shell-like” behavior, 305
- copy()**, 157
 - copyArea()**, 477
 - createDragSource()**, 523
 - createDropTarget()**, 523
 - Cursor()**, 498, 501, 502
 - Cursor, 497
 - color, 506
 - converting an ImageData to indexed color
 - coding example, 506, 507c–508c
 - custom cursors, 501–503
 - black and white, 503
 - coding example converting ImageData to black/white/transparent, 503c–505c
 - size limits, 505
 - hotspot, 497
 - inherited, 277
 - multiple, 277
 - resizing cursors, 499
 - constants, 499t–500t
 - setting, 276
 - shape, 497
 - system cursors, 498, 500
 - constants, 498t–499t
 - cut()**, 157



**D**Data *see* Application data*data* field (image), 464–465

Decorations, class, 131

delayTime (ImageData fields), 472*depth* field (image), 464*deselect()*, 171, 240–241*deselectAll()*, 171, 221, 241*detail* field issues, and traversal codes, 43–44, 45t

Device (Drawable interface), 410

Dialog(), 323

Dialogs, 323, 341

ColorDialog, 337, 338f

creation, 337

opening, 337–338

setting RGB, 337

creation, 323–324

dialog window, 323

DirectoryDialog, 334, 336f

creation, 334

directory path setting, 335

message setting, 334–335

opening, 336

FileDialog, 328, 331f, 334f

 coding example for image file using file
 extensions, 334c–333c

creation, 328–329

filter extensions setting, 329–330

filter path setting, 329

multiselection capability, 332

opening, 332

set/get file name, 331–332

FontDialog, 338–339, 341f

creation, 339

FontData List setting, 339

opening (coding example), 340c

RGB setting, 339

MessageBox, 325

buttons, 326–327

creation, 325

icons, 326

opening, 327–328

setting message, 327

opening, 324

title setting, 324–325

Display(), 91

Display, 16–17, 89, 128

application data, 122–123

dismissal alignment, 123–124

application naming, 90

beep, 127

cursor control/location, 116

getting cursor control, 116–118

setting/getting cursor location, 118

depth and DPI, 118–119

system colors, 120

system font, 120

system information, 119

double-click time, 127

events/listeners, 91–93

life cycle, 90–91

querying, 26

updating, 121–122

see also Event filters; Event loop; Monitors;
 Runnables; Shells; Threading; Timers*disposalMethod* (ImageData fields), 472–473**dispose()**, 12

vs. close, 285

(display destruction), 91

and system objects, 119

disposeExec(), 95

DisposeListener interface, 22

see also Widget events*doit* field issues

input validation, 39

and traversal implementation/stop, 44, 45t

Drag-and-drop request, 60–61

in application example, 522–523

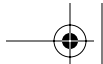
Draggable controls, 303, 321

 Class CoolBar/CoolItem *see* CoolBar/
 CoolItem Class Sash *see* Sash Class Tracker *see* Tracker

Drawable interface, 410

and controls, 478

drawArc(), 417**drawFocus()**, 416–417**drawImage()**, 475–476**drawLine()**, 412**drawOval()**, 417**drawPolygon()**, 412



drawPolyline(), 412
drawRectangle(), 416
drawRoundedRectangle(), 416
drawString(), 448, 451
drawText(), 449

E

Eclipse Project, xix
 EvenOdd rule, 419
 Event filters, 93–94
 Event loop, 96, 111

- event queue flushing, 97
- inactivity detection using timer, 109–110
- reading/dispatching, 96–96
- sleeping/waking, 97–100

 Events, 17–18

- Class event (Public Fields), 19t
- filters, 18
- Focus events, 30, 30t, 31f
- Key events, 32–33, 32t
- Move, 68–69, 68t
- Resize, 69–70, 70t
- Selection events, 61–62, 62t
- Text events, 159–164
- Typed, 22t–23t
- Untyped, 20t–21t

see also Keyboard support; Listeners;
 Mouse events; Traversal events;
 Widget events

F

FileExplorer (sample application), 519

- application launching, 523–524
- drag-and-drop, 522–523
- features, 519–521, 526
- in-place editing, 525–526
- layout flexibility, 521–522
- multithreading, 524–525

fillArc(), 419
FillLayout(), 347
fillOval(), 419
fillPolygon(), 419
fillRectangle(), 418
fillRoundedRectangle(), 418

fillTable() method, 243

- coding example, 244c

 Filtering process, 329

- filter path, 329

 Filters *see* Event filters
 Flicker reduction, 84–85, 86–87
Flooding, 104
 Focus control (keyboard events), 28, 116

- forcing, 29–30
- setting, 29

 Focus events, 30–31
Font(), 443, 444
FontData(), 437, 441
 Fonts, 437

- Class Font, 442–443
 - coding rules for platform independent fonts, 447–448
 - using more than one font, 444–447
- Class FontData, 437–438
 - character encoding, 439
 - locale, 439–441
 - scalable vs. bitmapped fonts, 438–439
 - tracking font choices, 441–442
- Class FontMetrics, 451–452, 453
 - height measurements, 452
 - invalid data issues, 452
 - methods, 453–454
 - TwoTextSizes coding example, 455c–456c
 - width measurements, 456–458

 drawing text, 448–449

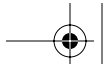
- coding example, 450c, 451f
- drawText() constants, 449t
- rules for platform-independent code, 451

forceActive(), 284
forceFocus(), 29–30
FormAttachment(), 379
FormData(), 378
 FormLayout code example (cool bar and shell
 resize), 308c–309c
Fragile superclass problem, 14–15

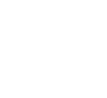
G

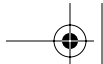
GC *see* Graphics/class Graphics Context (GC)
 GC(), 409





- getAccelerator(), 49
- getActiveShell(), 115
- getAdvanceWidth(), 458
- getAlignment(), 139, 235
- getAlpha/s(), 470
- getAscent(), 454
- getAverageCharWidth(), 457
- getBackground(), 220, 238, 276, 481
- getBlue(), 431
- getBounds(), 66, 112–113, 189, 242, 299, 307, 423
- getCharWidth(), 458
- getChecked(), 217, 237
- getChildren(), 278
- getClientArea(), 67, 113
 - centering Shell on primary monitor example, 113c–114c
- getClipping(), 424–425
- getControl(), 187, 255, 305
- getCursorControl(), 117–118
- getCursorLocation(), 118
- getCursorSizes(), 505
- getData(), 25
- getDefaultButton(), 290
- getDescent(), 454
- getDisabledImage(), 190
- getDismissalAlignment(), 123–124
- getDisplay(), 26
- getDoubleClickTime(), 127
- getEchoChar(), 151
- getEnabled(), 75, 190, 197
- getExpanded(), 214
- getFileName(), 331–332
- getFilterExtensions(), 329
- getFilterNames(), 330
- getFilterPath(), 329, 335
- getFocusControl(), 116
- getFont(), 220, 238, 276, 300, 443
- getFontList(), 339, 438, 446
- getFontMetrics(), 453
- getForeground(), 220, 237, 276
- getGrayed(), 217, 237
- getGreen(), 431
- getHeaderHeight(), 231
- getHeaderVisible(), 231
- getHeight(), 453
- getHorizonBar(), 267
- getHotImage(), 190–191
- getIconDepth(), 119
- getImage(), 137, 141, 185, 197, 234, 236, 254, 283
- getImageBounds(), 242
- getIncrement(), 264, 268
- getItemCount(), 215, 256, 312
- getItemHeight(), 167, 231–232
- getItemOrder(), 311
- getItems()/getItem()/getItemCount(), 166–167, 193, 200, 215, 230, 242, 256, 312
- getItemSizes(), 311
- getLayout(), 346
- getLeading(), 454
- getLineCount(), 151
- getLineDelimiter(), 153
- getLineHeight(), 151
- getLineStyle(), 416
- getLinesVisible(), 231
- getLineWidth(), 416
- getLocale(), 441
- getLocation(), 66, 299
- getMaximum(), 261, 263, 267
- getMenu(), 198, 275
- getMenuBar(), 285
- getMessage(), 327, 335
- getMinimum(), 261, 263, 267
- getMinimumSize(), 306
- getMonitors(), 112, 275
- getPageIncrement(), 264, 268
- getParent(), 9–10, 275
- getParentItem(), 215
- getPixel/s(), 465, 467
- getPreferredSize(), 305–306
- getPrimaryMonitor(), 112
- getRectangles(), 317
- getRed(), 431
- getRGB(), 337, 431, 464, 467
- getRowCount(), 191
- getSelection(), 143, 155, 170, 185, 221, 240, 255, 261, 264, 268
- getSelectionCount(), 221
- getSelectionIndices(), 170, 255
- getShells(), 115–116, 275
- getSize(), 66, 299, 307
- getStippled(), 317





getStyle(), 10–11
 getSystemColor(), 433
 getSystemFont(), 443
 getTabList(), 42
 getText(), 137, 141, 150, 184, 197, 234, 236, 254, 280, 283
 getTextLimit(), 159
 getThread(), 100
 getThumb(), 268
 getToolTipText(), 189, 254, 276
 getTopIndex(), 158, 241
 getTopItem(), 221
 getTransparencyMask(), 472
 getTransparencyType(), 472
 getVericalBar(), 267
 getVisible(), 73, 300
 getWidth(), 186, 234
 getWrapIndices(), 311
 getXORMode(), 416
Grab control, 54
 Graphical Editing Framework (GEF), 403
 Graphics, xviii, 401–403
 3D, 402
 advanced, 2D, 402
 class Graphics Context (GC), 409–410
 Class Point, 406–407
 points in hashed collections, 407
 Class Rectangle, 407–409
 clipping, 421, 424–426
 line and figure drawing, 412–413, 414f
 coding example (rectangle/oval/arc and XOR model), 417c–418c, 418f
 controlling line appearance, 414–415
 filling, 418–419, 419c–420c, 420f
 Moiré pattern, 412
 and platform graphics routines, 415–417
 points and rectangles, 405
 regions, 422–424
 and threads, 410–412
 video and animation, 402
 Grid *see* Layout
 Graduate(), 365
 Griddled(), 360–361
 Group, 132
 GK. widget toolkit, 27
 GUI (graphical user interface), xix

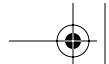
H

height field (image), 464
 “Hit test” operations, 193, 242
 HTML, and user interfaces, 1

I

I-beam *see* Caret
Idle handlers, 106–108
 coding example creating TableItems, 245–246
 Image(), 474–475
 Image (Drawable interface), 410
 ImageData(), 462
 imageDataLoad(), 493
 ImageLoader(), 487 T
 Images, 461
 animation, 481
 algorithm, 481–482, 483c–486c
 double buffering, 482
 Class Image, 473–475
 image copying, 477–478
 coding example, 478c–479c
 image drawing, 475–476
 coding “magnifying image viewer,” 476c–477c
 removing transparency, 479–481
 Class ImageData, 461–462
 data field, 464–466
 fields affecting transparency, 469–470
 fields for animation support, 472–473
 image attributes fields, 464
 indexed color display images, 467–468
 isDirect field, 466–467
 palette field, 466
 public fields, 463f
 transparency coding example, 470c–471c, 471f
 Class ImageLoader, 486–487
 coding example, 493c–494
 multiple images, 488–489
 saving images, 489–490
 Class PaletteData, 466–467
 disabled, 190
 hot, 190–191





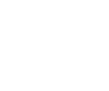
loading, 486
 background thread, 490–492
 incremental display, 492–495
 normal, 190
indexOf(), 193, 200, 242, 256, 312
Inheritance hierarchy, 14
Input Method Editor (IM/IME), 27, 300
 Input validation, *doit* field issues, 39
insert(), 156
 Interlacing, 492
intersection(), 408
intersects(), 408, 409, 423
isEnabled(), 75, 190, 197
isFocusControl(), 29
isSelected(), 170, 240
isVisible(), 73, 300
 Item, 130, 166
 see also List/class list

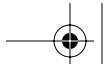
K

Keyboard shortcut (menu accelerator), 48–50, 198
 Keyboard support, 28, 51
 accelerators, 48
 specification, 49–50
 text, 50
 characters/key codes/state masks, 33–39
 “consuming” a key, 28
 focus control (keyboard events), 28, 116
 forcing, 29–30
 setting, 29
 focus events, 30–31
 inactivity detection using timer, 110–111
 key events, 32–33
 traversal events, 42–48
 traversal keys, 39
 mnemonic, 39–40
 tab, 41–42
 window system keys, 50–51
keyCode field, 34–35

L

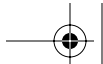
Label, 132
 class label, 135f
 alignment, 138–139
 label events, 136–137
 label hierarchy, 136
 label styles, 136
 separators, 139
 text/images, 137–138
 wrapping, 138
launch(), 523
layout(), 393
 Layout, 343–344, 399
 advantages, 343
 assigning width/height hints, 390–391
 computing height hint
 (Table/Tree/List/Text), 391–392
 computing width hint
 (Table/Tree/List/Text), 391
 Class FillLayout, 347
 fill layout with multiple children coding
 example, 348c–349c
 fill layout using all fields coding exam-
 ple, 349c–350c
 shell creation with single push button
 coding example, 347c–348c
 Class FormLayout, 377–378
 class FormAttachment, 378–380
 class FormData, 378
 edge attached at offset, 382–383
 edge attached as percentage, 381–382
 edge attached to another control, 384
 edge attached to opposite edge of con-
 trol, 384–386
 implementing a find dialog, 386–387
 implementing a SashPane, 388, 389c, 390f
 overconstraining a form, 380–381
 Class Griddled, 360–361
 coding example *numColumns* field used,
 362c–363c
 Class Layout, 345
 setting, 345–346
 Classes RowLayout/RowData, 350–352
 RowData to configure controls, 358–360





- RowLayout and column controls, 353–354
 - RowLayout and filling, 355–356
 - RowLayout and packing/justifying, 357
 - RowLayout and wrapping, 354–344
 - defining grid, 363–364
 - “convenience constants” (Graduate), 373–375
 - Graduate for aligning Controls, 368c–369c
 - Graduate for Controls configuration, 364–367
 - Graduate to grab excess space coding example, 369c–371c
 - Graduate to span rows/columns coding example, 371c–372c
 - Griddled to implement a Find Dialog coding example, 375–376, 376c–377c
 - hidden controls usage coding example, 372c
 - sizing controls coding example, 367c
 - FillLayout/RowLayout coding examples, 352c–353
 - forcing, 392–394
 - forcing controls to wrap code example, 394c–395c, 395f, 396
 - margin/spacing issues, 352
 - selection criteria, 392
 - usage, 344
 - composite resized, 344
 - layout() method of class Composite called, 344
 - width hints and SWT.Resize to wrap label coding example, 396c–398c
 - width hints and SWT.Resize to wrap tool bar coding example, 398c–399c
 - and z-order, 345
 - Leaf items, 214
 - Ligature, 458
 - List, 133
 - class list
 - adding items, 167–168
 - deselecting/selecting items, 170–171
 - example, 164
 - item height, 167
 - list events, 165–166, 172–173
 - list hierarchy, 165
 - list styles, 165
 - removing items, 168
 - scrolling, 171–172
 - searching, 172
 - selection setting methods, 169–170
 - setting items, 166–167
 - single-/multiselect lists, 166
 - Listeners, 17, 23–24
 - typed, 21–23
 - untyped, 18–21
 - see also* Events
 - load(), 488
 - Locale, 439–441
- M**
- map() method forms, 124
 - Mapping coordinates, 124
 - maskData (ImageData field), 469
 - maskPad (ImageData field), 469
 - Measuring operation *see* Fonts/Class Font-Metrics
 - Menu accelerator (keyboard shortcut), 48–50
 - Menu bar setting, 285
 - Menu/MenuItem, 133
 - casca delsubmenus*, 196
 - class Menu/MenuItem, 206–207
 - buttonlike MenuItem, 196–197
 - example, 194
 - hierarchy, 194
 - index-based operations, 200
 - main menus, 198, 198c–199c
 - Menu events, 195, 201–203
 - Menu styles, 194
 - MenuItem events, 195–196, 203–207
 - MenuItem styles, 195
 - MenuItems with accelerators, 198
 - MenuItems that show others, 198
 - pop-up menu creation, 199, 200c
 - separator-like MenuItem, 197





drop-down menus, 196
pop-up/context menus, 196
MenuItem.setMenu(), 13
 MessageBox *see* Dialogs
 Minesweeper (sample application), 513

- Class Board, 517–518
- Class Counter, 516
- Class Minesweeper, 515–516
- Class Timer, 516–517
- implementation notes, 515
- overview, 513–515

 Mirroring, 124–125

- coding example of Shell alignment, 125c–126c, 127f

 Mnemonic traversal, 39–40

- mnemonic strings, 40t

 Modality, 282, 289
 Monitors, 112, 112f

- Shell positioning coding example, 114c
- see also* Bounds; Client area

 Mouse, 53–54

- intercept (coding example), 75c–76c
- wiggly mouse problem*, 98–100

 Mouse events, 54, 56t–57t, 63

- buttons/coordinates, 54–56
- context menu request, 59
- displaying example, 58c–60c
- double-click events, 57–58, 127
- drag-and-drop request, 60–61
- selection* operation on widget, 61–62
- state masks, 55–56, 55t

moveAbove(), 74
moveBelow(), 74
 Moving/Resizing, 68

N
 Native widgets, xvii, 129–130

- class hierarchy, 130, 131f

 Nesting *see* Containment
 Notebooks *see* TabFolder
notifyListener(), 19
numColumns, 361, 363

- coding example, 362c–363c

O

open(), 284, 316
org.eclipse.swt.events, typed listeners and event classes, 21

P

pack(), 76–77, 234
 Painting, 77–78

- background color control, 86
 - and flicker reduction, 86–87
- in a canvas, 296
- deferred update strategy, 78
- events, 78–79, 78t, 80f
- forcing an update, 81
- redraw, 80–81
 - turning off, 82
- repainting/resizing, 82–85

palette field (image), 464
 Parent parameter, 9–10
paste(), 158
Platform, xxi
Point(), 406
 Point class, 66
 Pointing device *see* Mouse
 Points vs. *int* parameters, 410
 Position parameter, 11
 ProgressBar, 132

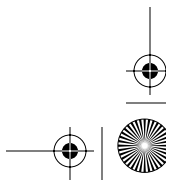
- Class ProgressBar
 - example, 259
 - hierarchy, 259
 - indeterminate ProgressBar, 262
 - ProgressBar events, 260
 - ProgressBar styles, 260
 - range operations, 261
 - selection of “fill rectangle” extent, 261
 - creating shell with progress bar coding example, 260c–261c

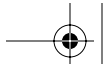
 Progressive JPEG, 492

R

R3.0, Table performance improvements, 244
 Range-based controls, 259, 272

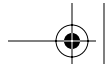
- Class ProgressBar *see* ProgressBar
- Class Scale *see* Scale
- see also* Classes ScrollBar and Slider





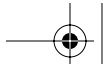
- readAndDispatch(), 96
 - Rebar controls, 304
 - Rectangle(), 407
 - redraw(), 80
 - Region(), 422
 - remove(), 168, 239
 - removeAll(), 168–169, 222, 238–239
 - removeDisposeListener(), 22
 - removeFilter(), 94
 - removeImageLoaderListener(), 492
 - removeListener(), 18, 19, 91
 - Resizing *see* Moving/Resizing
 - RGB(), 427
 - RGB
 - Class RGB, 427–428
 - direct color, 428–429
 - indexed color, 429–430
 - see also* Dialogs/ColorDialog
 - RowData(), 358
 - RowLayout(), 351
 - Rubber banding, 380
 - see also* Tracker
 - Runnables (runnable execs), 95
- S**
- Sash, 132
 - Class Sash; example, 313
 - events, 313–314
 - hierarchy, 313
 - styles, 313
 - save(), 489
 - Scale, 132
 - Class Scale
 - events, 262–263, 264
 - example, 262
 - hierarchy, 262
 - increment/page increment, 264
 - range operations, 263
 - selection, 264
 - styles, 262
 - thumb, 263
 - and selection, 264
 - ticks, 263
 - trough, 263
 - scanLinePad field (image), 464
 - scroll(), 297–298
 - Scrollable, 130
 - ScrollBar, 133
 - creation, 266–267
 - and tables, 228
 - and trees, 212
 - see also* Classes ScrollBar and Slider
 - select(), 171, 241
 - selectAll(), 154, 171, 221, 241
 - Separators, 139
 - setAccelerator(), 49
 - setActive(), 284
 - setAlignment(), 139, 235
 - setAlpha/s(), 470
 - setAppName(), 90
 - setBackground(), 220, 238, 276, 480
 - setBounds(), 68, 299
 - setCaret(), 296–297
 - setChecked(), 217, 237
 - setClipping(), 421
 - setControl(), 187, 255, 305
 - setCursor(), 276, 317
 - setCursorLocation(), 118
 - setData(), 25
 - setDefaultButton(), 290
 - setDisabledImage(), 190
 - setEchoChar(), 151
 - setEnabled(), 75, 190, 197
 - intercept mouse events example, 75c–76c
 - setExpanded(), 214
 - and SWT.Selection, 215
 - setFileName(), 331
 - setFilterExtensions(), 329
 - setFilterNames(), 330
 - setFilterPath(), 329, 335
 - setFocus(), 18, 29
 - setFont(), 220, 238, 276, 300, 443
 - setFontList(), 339, 446
 - setForeground(), 220, 237, 275
 - setGrayed(), 217, 237
 - setHeaderVisible(), 231
 - setHotImage(), 190
 - setImage(), 137, 141, 184, 197, 234, 236, 237, 254, 283
 - setIncrement(), 264, 268
 - setItemLayout(), 312





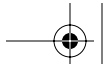
- setItems()/setItem(), 166
 - and Table, 231
- setLayout(), 345
- setLineStyle(), 414
- setLinesVisible(), 231
- setLineWidth(), 414
- setLocale(), 440
- setLocation(), 68, 199, 299
- setMaximum(), 261, 263, 267
- setMenu(), 198, 198c–199c, 275
- setMenuBar(), 285
- setMessage(), 327, 335
- setMinimum(), 261, 263, 267
- setMinimumSize(), 306
- setPageIncrement(), 264, 268
- setPixel/s(), 465–466
- setPreferredSize(), 305–306
- setRectangles(), 317
- setRedraw(), 82
 - creating Table items coding example, 244–245
- setRegion(), 290–291
- setRGB(), 337
- setSelection(), 143, 154, 169–170, 185, 221, 239–240, 255–256
 - with ScrollBar and Slider, 267–268
 - ScrollBar and Slider applications, 267–268
 - and string selection, 155c–156c
 - and SWT.Selection event, 143
 - within ProgressBar, 261
 - within Scale, 264
- setSize(), 68, 299, 306
 - and Control configuration, 358
- setStippled(), 317
- setTabList(), 42
- setText(), 137, 141, 149, 184, 196, 234, 236, 254, 280, 283
- setTextLimit(), 159
- setThumb(), 268
- setToolTipText(), 189, 254, 276
- setTopIndex(), 158, 171, 241
- setTopItem(), 221
- setValues(), 269
- setVisible(), 73, 199, 299
- setWidth(), 186, 234
- setXORMode(), 414
- Shell(), 286
- Shells, 16–17, 132
 - active state, 283–284
 - dialog, 282
 - and Display, 115
 - getting the Active Shell, 115
 - getting Focus Control, 116
 - getting list of Shells, 115–116
 - top-level, 282
 - see also Class Shell
- showItem(), 222, 242
- showSelection(), 158, 172, 221–222, 241
- sleep(), 96
- Slider, 133
 - see also Classes ScrollBar and Slider
- Smalltalk, xxv
 - influence on SWT, xxv–xxvii
- Splitters, 314
- stateMask field, 35–37, 35t, 36t
- Stippled effect, 317
- stringExtent(), 456
- Style parameter, 10–11
- StyledText, 149
- subtract() (from region), 424
- SWT (*Standard Widget Toolkit*), xix, 1
 - acquisition of, xxxv–xxxvi
 - available drops/platforms, xxxvi–xxxvii
 - full source via Eclipse, xxxl–xlii
 - command line arguments, xxxix
 - Eclipse plug-ins development, xliii
 - exception/error checking, 9, 12
 - “HelloWorld” (sample program), xxviii–xxix, 3–4
 - indexing and range operations, 150
 - installation as an extension, xxxviii–xxxix
 - overview, xx–xxi
 - efficiencies, xxii–xxiii
 - history, xxv–xxviii
 - implementations, xxi
 - “look and feel,” xxiii–xxiv
 - packages, xxx–xxxii
 - and memory constraints, xxxii–xxxiv
 - run components, xxxiv–xxxv
 - stand-alone and Mac OS X, xxxix–xl
 - see also Class SWT; Events; Keyboard support; Listeners; Native widgets;





- Widget hierarchy; Widget life cycle
- SWT.Activate event, 294
- SWT.Arm (ArmEvent), 204
 - updating status line example, 204, 205c, 206f
- SWT.ARROW, 146
- SWT.BAR, 198
- SWT.BUTTON_MASK, 55–56
- SWT.CASCADE, 198
- SWT.CHECK, 142–143, 185, 196
- SWT.Close event, 92, 92c–93c
 - for Shell, 292–293
 - preventing a Shell from closing code example, 293c–294c
- SWT.Collapse (TreeEvent), 225–226
- SWT.Deactivate event, 294
- SWT.DefaultSelection (SelectionEvent), 159, 172, 177, 178, 222–223
 - “recently typed” list coding example, 177c–178c
 - and Table events, 247
 - and TableColumn events, 250
- SWT.Deiconify event, 294
- SWT.Dispose event, 92, 92c–93c
- SWT.DragDetect event, 60–61
- SWT.DROP_DOWN(), 187–189, 196
 - code example with CoolItems, 309c–311c
- SWT.Expand (TreeEvent), 223
 - coding example (fill tree lazily), 223–225
- SWT.FLAT, 191–192
- SWT.FocusIn/SWT.FocusOut case examples, 30–31
- SWT.FULL_SELECTION, 229
- SWT.Help (HelpEvent), 203, 206
- SWT.Hide (MenuEvent), 202, 204
- SWT.HIDE_SELECTION, 229
- SWT.HORIZONTAL/VERTICAL, 183
- SWT.H_SCROLL, 266–267
- SWT.Iconify event, 294
- SWT.INDETERMINATE, 262
- SWT.KeyDown()/SWT.KeyUp(), 32–33
 - coding examples, 37–39
 - doit* field issues, 39
- SWT.LEFT/CENTER/RIGHT styles
 - and buttons, 142
 - and labels, 138–139
- SWT.MenuDetect event, 59–60
- SWT.Modify (ModifyEvent), 159, 179
 - invalid input warning to user, 161c–162c
 - track modified state of text control example, 160c–161c
- SWT.Move (ControlEvent)
 - Shell “follows the parent” example, 69c
 - TableColumn usage, 251
 - track a parent Shell example, 70c–71c
 - for Tracker, 318
 - coding example with tear-off controls, 318c–320c
- SWT.NO_BACKGROUND, and flicker reduction, 86–87
- SWT.NO_REDRAW_RESIZE, 83–84
 - and flicker reduction, 84–85
- SWT.Paint event, 78, 78t
 - draw in a canvas coding example, 79, 79c, 80f
- SWT.PASSWORD style, 150–151
- SWT.POP_UP style, 196, 199
- SWT.PUSH, 142, 185
- SWT.RADIO style, 144–145, 185, 196
- SWT.Resize
 - coding example with width hints to wrap label, 396c–397c
 - coding example with width hints to wrap tool bar, 398c–399c
 - move/resize children example, 71c–72c
 - TableColumn usage, 251
- SWT.Selection event, 143, 146, 172, 177, 178, 193–194, 203
 - CoolItem events, 312–313
 - implementing a drop-down ToolItem, 194
 - print current selection example, 172c–173c
 - with radio buttons, 146–147
 - for Sash events, 314
 - Scale changes, 264
 - for ScrollBar and Slider events, 269
 - scroll a list of strings case example, 269–270, 270c–272c
 - and `setExpanded()`, 215
- TabFolder event, 256
 - manual paging coding example, 256c–257c





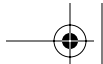
- and Table events, 247
 - check box table coding example, 248c
- and TableColumn events, 248
 - sort items in a Table coding example, 248c–250c
- and TreeItem, 211, 222
- SWT.SEPARATOR, 186, 192, 197
- SWT.SHADOW_OUT, 192
- SWT.Show (MenuEvent), 201–202
- SWT.SINGLE/MULTI, 149
- SWT.Size, track a parent Shell example, 70c–71c
- SWT.TOGGLE, 145
- SWT.Traverse() event *see* Traversal events
- SWT.Verify (VerifyEvent), 162c–164c
- SWT.V_SCROLL, 266–267
- SWT.WRAP style, 138, 157, 191
 - and TableItem, 235
- syncExec(), 101–102
 - in fillTable() implementation coding example, 246c–247c
 - running code in user interface thread example, 102, 102c–103c
- System font, 120
- System objects, 119

T

- Tab traversal, 41, 278
 - tab group, 41
 - tab item, 41
 - tab ordering specification, 42
- TabFolder/TabItem, 132
 - classes TabFolder/TabItem
 - example, 251
 - hierarchies, 251
 - searching operations, 256
 - TabFolder events, 251, 256–257
 - TabFolder styles, 251
 - TabFolders paging modes, 252–253
 - TabFolders selection methods, 255–256
 - TabItem events, 252
 - TabItem page controls, 254–255
 - TabItem styles, 252
 - TabItem text and images support, 253–254
 - TabItem tool tips, 254

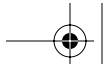
- Table/TableItem/TableColumn, 132–133
 - classes Table/TableItem/TableColumn
 - checked/grayed table items, 237
 - deselecting/selecting items, 240–241
 - example, 226
 - full row selection, 229
 - get items, 230
 - headers/grid lines, 231
 - hiding selection, 229
 - hierarchies, 226
 - hit testing operations, 242–243
 - item height, 231–232
 - “list mode,” 230
 - multiple-column tables, 231–234
 - program examples large tables, 244c–247c
 - removing table items, 238–239
 - scrolling, 241–242
 - selection setting methods, 239–240
 - table column alignment, 235
 - table column width, 234–235
 - Table events, 227, 247
 - Table styles, 227
 - TableColumn events, 228–229, 248–251
 - TableColumn styles, 227
 - TableItem events, 227
 - TableItem foreground/background/font, 237–238
 - TableItem styles, 227
 - text and images in table column, 234
 - text and images in table item, 235–237
 - performance improvements, 244
 - program example multiple table columns, 233c, 234f
 - program example one table column, 232c
 - setItems() method issue, 231
 - virtual tables, 243
 - see also* asyncExec(); setRedraw()
- tableDefaultSelected(), 524
- Tearing off control, 317
 - coding example, 318c–320c
- Text, 133
 - button support, 141
 - class text
 - clipboard operations, 157–158
 - echo character, 151





- example, 147
- hierarchy, 148
- inserting/appending, 156–157
- line delimiters, 152–153
- lines/line height, 151–152
- passwords, 150–151
- scrolling, 158
- single-line/multiline controls, 149
- string operations, 149–150
- substring selection, 153–156
- text events, 148–149, 159–164
- text limits, 158–159
- text styles, 148
- wrapping, 157
- label support, 137
- textExtent()**, 456–457
- Threading
 - apartment threading, 100–101
 - and creating table items coding example, 246c–247c
 - and *flooding*, 104
 - free threading, 101
 - multithreading in application example, 524–525
 - user interface thread, 100
 - syncExec() vs. asyncExec() issues, 104–106
 - see also* **asyncExec()**; Graphics/and threads; **syncExec()**
- Thumb, 263, 264
- Ticks, 263
- timerExec()**, 108
- Timers, 108–109
 - detecting event loop inactivity, 109–110
 - detecting keyboard inactivity, 110–111
- ToolBar/ToolItem, 129, 133
 - classes ToolBar/ToolItem, 106–207
 - bounds (tool item), 189
 - button-like ToolItems, 184–186
 - drop-down feature, 187–189
 - enabling/disabling tool items, 189–190
 - example, 181
 - flat tool bar, 191–192
 - hierarchies, 181
 - image supports, 190
 - searching/”hit test” operations, 193
 - separator above tool bar, 192
 - separator-like ToolItems, 187
 - tips (tool item), 189
 - ToolBar events, 182
 - ToolBar style, 182
 - ToolItem events, 182–183, 193–194
 - ToolItem styles, 182
 - wrapping tool bar, 191
 - Tool Tips, 184
 - ToolItems creation coding example, 185c–186
- Tracker, 133
- Tracker()**, 316
- Tracker
 - Class Tracker
 - creation, 316
 - events, 315, 318–320
 - example, 315
 - hierarchy, 315
 - opening, 316–317
 - setting rectangles, 317
 - stippling/cursor use, 317
 - styles, 315
 - and Dialog, 316
 - rubber banding, 316
- transparentPixel* (ImageData field), 469
- Traversal events, 42–43, 43t
 - SWT.Traverse() event, 43
 - implementing tab group traversal, 45–46
 - implementing text traversal, 46–47
 - intercept traversal, 47–48
 - keyboard traversal codes, 44t
 - managing traversals, 44, 45t
 - public fields, 43t
- Tree/TreeItem, 133
 - classes Tree/TreeItem
 - checked/grayed tree items, 217–218, 218c–219c, 220f
 - example, 210
 - expanding/collapsing, 214–215
 - hierarchical operations, 215–217
 - hierarchies, 210
 - hierarchy creation, 212, 213c
 - removing items, 222
 - scrolling, 221–222
 - selection, 220–221
 - and SWT.Selection, 211–211





- texts/images, 213–214
- Tree events, 210, 222–226
- Tree styles, 210
- Treeltem events, 211
- Treeltem foreground/background/font, 220
- Treeltem styles, 211
- Trimings, 282
- Trough, 263
- type* field (image), 464

U

- `union()`, 409
- `update()` method, 121–122
 - forcing a paint event, 81

V

- Visibility, 72–73

W

- `wake()`, 98
 - coding example, 99c–100c
- Widget, xvii–xviii, 1, 7, 26, 130
 - and application data, 25–26
 - constructor use of operating system resources, 8
 - custom, xxiv–xxv
 - heavyweight*, 130
 - lightweight*, 130

- selection*, 61–62
 - see also* Common Widgets; Native widgets
- Widget events, `SWT.Dispose()`, 24–25
- Widget hierarchy
 - containment* hierarchy, 14
 - controls/composites/Shells/Display, 16–17
 - importance of, 9
 - inheritance* hierarchy, 14
 - non-control widgets, 17
 - subclassing, 14–16
 - “final” tag issue, 15–16
- Widget life cycle, 7
 - creation overview, 8
 - disposal of, 12
 - rules, 13–14
 - parameters, 9
 - parent parameter, 9–10
 - position parameter, 11
 - style parameter, 10–11
 - standard constructors, 8–9
- width* field (image), 464

X

- X/Motif implementations, 444–445
- x/y* (ImageData fields), 472
- XML, and user interfaces, 1

Z

- Z-order, 74
 - and layout, 345

