

INDEX

A

- absolute positioning, 59
- adding EGL Rich UI code, 78–80, **78**, **79**
- adding widgets to an app, 54–57, **55**, **56**, **57**, 62–67
- administration console, 118, **118**
- Ajax, 3, 13, 37
- aligning text, 71–72, **72**
- Apache, 13
- APIs, 11–12
- architecture of multi-tiered applications, 7–16
 - 3270 as first, 8–9, **8**
 - 5250 as early prototype of, 9, **9**
 - 5250 emulation and, 10, **10**
 - business logic tier in, 187–216. *See also* business logic tier
 - communication between tiers in, 32–34
 - debugging and, 234–241
 - defining tiers in, 81–82, **81**, **82**
 - JavaServer Pages (JSP) 2.0 and, 13–14, **14**
 - model-view-controller (MVC) and, 13–14, 51
 - non-EGL design steps for, 25–28, **25**
 - PCs in, 10, **10**
 - placeholders and, 87–89
 - project for Tier 2 in, 83–84, **84**

- records to include in, 82–83
- rich client in, 14–15, **14**
- rich web client messaging in, 15–16
- screen scrapers and, 10–11, **10**
- servers and, 187–189, **188**
- simple design prototyping for, 82–83
- simple, two-tiered application in, 7–8, **7**
- thick client in, 11–12, **11**, 18
- thin client (browsers) in, 12–13, **12**, 18, 34–35, **35**, 97–115. *See also* thin client tiers in, deciding on number of, 18–19
- WSIWYG designer and, 14
- asynchronous processing, 134–184
- auto-completion, 90, **90**

B

- behaviors, 157–160, **158**, **159**
- bidirectional parameters, 33–34, 89, 115
- boxes, 57–59, **57**, **58**. *See also* widgets
 - EGL Rich UI Application and, 149–150
 - hidden, 150
- breakpoints 229, **229**, 234–235, **235**, 241, **241**, 246–250, **246**, **249**. *See also* debugging

browsers, 21
 multi-tiered application architecture and,
 12–13, **12**

Build Parts Editor, 199–207, **199**

Build Path dialog, 141–142, **141**

business logic tier, 187–216
 Build Parts Editor in, 199–207, **199**
 calls to, 209
 configuring the project to talk to the RPG
 program, 198–207, **199**
 content-assist and, 210
 EGL and, 187
 EGL code added to, 207–211
 error handling in, 209
 JAR file for, 205–206, **206**
 keywords and, 193, 194
 linkage part added to, 200–205, **200, 201,**
 203, 204, 205
 messages in, 189–192, **191**
 ORDINQR program in RPG for, 192–198
 results of application using, 211–215,
 211–215
 servers and, 187–189, **188**
 user IDs and passwords in, 191–192, 208

buttons, 24, 59, 72. *See also* widgets,

C

C/C++ languages, 19, 20

callback routines, 136

callbacks, 138, 157–160

calls, function, 112, **112, 144, 144, 209**

Cascading Style Sheets (CSS), 3, 147, 153,
153, 215

centering text, 71–72, **72**

CGI, 22, 34

check box enabling, 168–170, **169**. *See also*
 event handlers; widgets

classes, 161–162, **162**

cloud computing, 7

COBOL, 7, 17, 19, 21, 50

color, 153

color codes and drop zones, 55, **55, 63**

composite records, 32–34

concatenation, 93

Configure Data Controls wizard, 106, **107**

 configuring the project to talk to the RPG
 program, 198–207, **199**

Construction function, 110

content-assist, 110, **111, 210**

copying
 code/libraries, 139–140, **140**
 packages, 142–143, **143**

CPU power, 2

cross-hatch drop zones, 63–64, **64**

CRUD servers, 188–189, **189**

CSS. *See* Cascading Style Sheets

D

data initialization, 110–112

DB2, 20

Debug perspective, 231–232, **231, 232**

debugging, 4, 93–95, **94, 97, 227–251**. *See also*
 error handling
 breakpoints in, 229, **229, 234–235, 235, 241,**
 241, 246–250, 246, 249
 browser selection for, 230, **230**
 Editor view in, 232, **233**
 EGL Rich UI Application and, 144, **144, 145**
 fixing the error in, 250–251, **250, 251**
 generated code in, 237, **237**
 host program, 241–251
 Interface Mapping error in, 231, **231**
 JAR file in, 238–240, **239, 240**
 JavaScript, 228–234
 launching Debug for, 229, **229**
 listeners and, 233
 logging and, 228, **228**
 middle tier, 234–241
 perspective for, 231–232, **231, 232**
 Preferences dialog setting for, 236, **236**
 Remote System Explorer and, 242–244,
 242, 243
 Resume tool in, 247–248, **248**
 Run To Location option in, 249–250, **249**
 service entry point (SEP) in, 242, 244, **245,**
 246, 246, 247, 247
 step into setting, 236–238, **238**
 user ID and passwords in, 244, **245**
 Variables view in, 232, **233**

- defining
 - functions, 225
 - tiers, 81–82, **81**, **82**
 - the web service, 38
 - delegates, 135–136, 171–174, **172**. *See also* listeners
 - Design mode, 103, **103**, 103
 - designing the UI, 22–23, **23**
 - development environment, 26
 - Dijit toolkit, 256. *See also* Dojo
 - distributed services, 7
 - Dojo, 26, 167, 255–260, **259**
 - dotted line areas in WYSIWYG editor, 56, **56**
 - drag symbol, **65**
 - drop zone, 55, **55**, 65, 69. *See also* widgets; WYSIWYG editor
 - cross hatching in, 63–64, **64**
- E**
- editable fields, 68
 - Editor view, for debugging, 232, **233**
 - EGL, 13, 17–39
 - adding EGL Rich UI code, 78–80, **78**, **79**
 - assumptions for proceeding with this book and, 21–25
 - bidirectional parameters in, 33–34, 89, 115
 - browsers and, 21
 - business logic tier and, 187
 - composite records and, 32–34
 - ease of use with, 19–21
 - extensibility of, 4, 254–255, **254**
 - future of, 253–262
 - initializing data in, 89–91
 - inter-tier communication in, 32–34
 - interface design using, 23, **23**
 - interface for, 22
 - JavaServer Faces (JSF), 20
 - JSON and, 21
 - library functions in, 20–21, **20**, 117–118, **117**
 - listeners in, 37–38
 - page handler in, 20
 - parts in, 20
 - programming language and, 21
 - RDİ-SOA and, 19, 21
 - record definition using, 29–32, **29**, **30**, **31**
 - REST and, 21
 - reuse concepts in, 115
 - servers and, 19, 21
 - SOAP and, 21
 - technology-centric design of, 18
 - thick client and, 19
 - thin client and, 19–21, **20**, 34–35, **35**
 - UI design using, 22–23
 - web services and, 35–36, **36**
 - widgets and, 24–25
 - EGL Free, 4–5, 260–261
 - EGL library functions, 20–21, **20**, 117–118, **117**
 - EGL Rich UI application, 4, 133–185
 - asynchronous processing in, 134–184
 - attaching to the service with, 175–181, **175–177**
 - behaviors and, 157–160, **158**, **159**
 - boxes added to, 149–150
 - Build Path dialog in, 141–142, **141**
 - callback routines and, 136, 157–160
 - calling functions in, 144, **144**
 - Cascading Style Sheet (CSS) and, 147, 153, **153**
 - classes and, 161–162, **162**
 - delegates in, 135–136, 171–174. *See also* listeners
 - error handlers, 140–141, **141**, **143**
 - event-driven processing in, 135
 - finished example design for, **146**
 - GoogleMaps application for, 24–25, **25**, 146, **146**, 163–171, **165**, **170**, **171**
 - Grid widget in, 149
 - hidden boxes and, 150
 - Infobus and, 137, 181–184, **184**
 - integrating additional technologies in, 162–171
 - listeners in, 135–136
 - naming conventions for, 148
 - OpenAjaxHub technology and, 137
 - ports for, 178
 - publish-and-subscribe concept in, 136–137, **136**
 - rapid application design in, 145–146
 - synchronous processing in, 134–135, **134**, 137–145
 - testing, 180–181, **180**
 - testing/debugging, 144, **144**, **145**
 - UI separated from business logic in, 146–147
 - emulators, 9

Enterprise Archive (EAR) projects, 84
error handling, 4, 91–92, 140–141, **141**, **143**,
217–251. *See also* debugging
architecture for, 217
business logic tier and, 209
coding for, 219–227
debugging and, 227–251. *See also* debugging
error messages in, 218–219, **218**
Error record in, 219–227
logging in, 217, 222–224, 228, **228**
sample code for, 218–219, **218**
timestamp for, 224–227, **227**
visual indicators for, 220–222
event handlers, 168–170, **159**
adding action to widgets, 74–77, **75**, **76**, **77**
event-driven processing, 135
Excel, 23
Executive Dashboard, 254–255, **254**, **255**
extensibility of EGL, 4, 254–255, **254**
extensions, 86
external applications for web pages, 24
GoogleMap, 24–25, **25**, 146, **146**, 163–171,
165, **170**, **171**
timestamp for errors, 224–227, **227**
external references, 122–123, **123**
External Type, 163–171

F

5250 as early prototype of multi-tiered
application, 9, **9**
5250 emulation, 10, **10**
fetch function, 88
folders, 98–99, **98**, **99**
adding, 163–164, **164**
creating, 52, **52**
fonts, 155
format of the book, 5
formatting, 152–162
Cascading Style Sheet (CSS) and, 153, **153**
color, 153
EGL Rich UI Application and, 144, **144**, **145**
fonts, 155
HTML and, 153
justification, 161, **161**
padding and spacing, 156, **156**

width adjustment in, 156–157, **157**, 158–162
framework for application, 81–96
function calls, 112, **112**
functions
defining, 225
in example application, 92
future of EGL Rich UI, 253–262

G

getter functions, 88
GoogleMaps application, 24–25, **25**, 146, **146**,
163–171, **165**, **170**, **171**
graphical UIs, 2–3
green screen vs. Rich UI applications, 2
grid computing, 7
Grid widget, 24, 149. *See also* formatting;
widgets
formatting, 152–162
text added to, 151–152
updating, 152

H

Hibernate, 26
hidden boxes, 150
hierarchy popup, 65–66, **66**, 69
host program debugging, 241–251
host tables, 243
HTML, 3, 13, 15, 23, 100, 153
HTTP servers, 13

I

IBM i, 2, 5, 21
IBM Rational, 17–39
ILE, 2, 25, 26
Infobus, 134, 137, 181–184, **184**, 253, 254
initializing data in EGL, 89–91, 110–112
Installation Manager, 42–44, **42**, **43**, **44**, 261
integrating additional technologies, 162–171
inter-tier communication, 32–34
interface design, 23, **23**
Interface Mapping error, 231, **231**
ISAM, 2
iSeries Access, 9

J

JAR file
 business logic tier and, 205–206, **206**
 debugging and, 238–240, **239**, **240**
 Java, 3, 9, 18, 19, 25, 26, 49, 167, 253
 Java Toolbox, 26
 JavaScript, 3, 15, 25, 26, 28, 97, 167, 253
 debugging and, 228–234
 JavaScript Object Notation. *See* JSON
 JavaServer Faces (JSF), 20
 JavaServer Pages (JSP), 50, 100, 101, 189, 215
 multi-tiered application architecture and,
 13–14, **14**
 JAX, 25, 26
 jQuery, 255
 JSON, 3, 4, 15, 16, 21, 28, 37
 justification/alignment, 161, **161**

K

keywords, 90–91, 193, 194

L

labels, 24
 Laffra, Chris, 158
 libraries
 copying, 139–140, **140**
 proxy, 86, 124
 library functions, EGL, 20–21, **20**, 117–118, **117**
 Lines array, 106–108, **107**
 linkage parts, 200–205, **200**, **201**, **203**, **204**, **205**
 Linux, 9
 listeners, 37–38, 135–136. *See also* delegates
 debugging and, 233
 logging, 217, 222–224, 228, **228**. *See also*
 debugging; error handling

M

maximizing the view, 75, **75**, 109
 messaging
 business logic tier and, 189–192, **191**
 Infobus and, 181–184, **184**
 record definition and, 29–32, **29**, **30**, **31**
 rich web client, 15–16

model-view-controller (MVC), 13–14, 51
 modes in designer, 102, **102**
 multi-tiered applications, architecture of. *See*
 architecture of multi-tiered applications
 MyEclipse, 19, 26

N

naming conventions, 148
 EGL Rich UI Application and, 148
 packages, 85–87, 86*t*
 servers, 192
 widgets, 56, **57**, 60, **60**
 nested boxes, 58, **58**, 60, **60**
 Net, 260
 .NET, 18, 22
 NetBeans, 19
 New EGL Data Variable wizard, 105–110, **105**
 new project creation, 47–59, **47**
 defining tiers in, 81–82, **81**, **82**
 folder creation in, 52, **52**
 folder organization and, 98–99, **98**, **99**
 for Tier 2, 83–84, **84**
 framework for, 81–96
 package creation in, 52
 page handlers and, 50–51, **51**
 perspectives in, 48–50, **49**
 placeholders and, 87–89
 records to include in, 82–83, 83*t*
 rich UI application creation in, 50–54
 Rich UI handler in, 50–51, **51**, 52–54, **53**
 simple design prototyping for, 82–83
 type of project in, 47–48, **48**
 non-EGL design of multi-tiered application,
 25–28, **25**
 null values, 124, 130

O

onConstruction function, 110
 onPreRender function, 110
 OpenAjax, 137, 215
 OpenAjaxHub, 4
 ORDINQR program in RPG, 192–198
 output of example application, **95**

P

packages, 42–43, 84–87
 copying, 142–143, **143**
 creating, 52
 naming, 85–87, 86*t*
 placeholders and, 87–89
 web services and, 119, **119**
padding and spacing, 156, **156**
Page Data view, 103–104, **104**
page handlers, 20, 50–51, **51**, 104, 110, **111**
Palette, 55, **55**
panning, 56
parameters, bidirectional, 33–34, 89, 115
parts, in EGL, 20
 web services and, 120, **120**, **121**
passwords, 191–192, 208, 244, **245**
PCs in multi-tiered applications, 10, **10**
Perl, 18, 34
perspectives, 46, 48–50, **49**
PHP, 18, 19, 34, 260
placeholders, 87–89, 138–140, **140**
port numbers for services, 118–119, **119**, 126, 178
Prerender function, 110
programming languages, 21
Project Explorer, 47
Project Interchange Files (PIFs), 5
Properties dialogs, 61–62, **61**, **62**, 67–68, **67**, **68**,
 71, **71**. *See also* formatting
proxy libraries, 86, 124
publish-and-subscribe concept, 136–137, **136**
Python, 18

Q

query servers, 188

R

radio buttons, 24. *See also* widgets
Rational, 26
Rational Business Developer, 3, 42, 260–261
Rational Developer for the i, 3, 42
Rational Team Concert for i (RTCi), 44
RD-i-SOA, 3, 5, 13, 19, 21, 97
 installation for, 41–44, **42–44**
 splash screen for, 46, **46**

records, 82–83, 83*t*
 composite, 32–34
 definition for, 29–32, **29**, **30**, **31**
Remote System Explorer, 242–244, **242**, **243**
Representational State Transfer. *See* REST
REST, 4, 15, 16, 21, 25, 28, 117–118
 EGL library functions and, 117–118, **117**
 web services and, 36, 120, 124
Resume tool, in debugging, 247–248, **248**
reuse concepts, 3–4, 115
 copying code and libraries for, 139–140, **140**
rich client architecture, 14–15, **14**
Rich Internet Application (RIA), 3
Rich UI, 2, 133–134. *See also* EGL Rich UI
 Application
 Rich UI application creation, 50–54
 Rich UI editor, 54, **54**
 Rich UI handler, 50–51, **51**, 52–54, **53**
 rich web application, **1**
 rich web client messaging, 15–16
RPG, 2, 7, 13, 17, 19, 21, 22, 34, 50
 business logic tier and, configuring the
 project to talk to, 198–207, **199**
Run To Location option, in debugging,
 249–250, **249**
runtime serves, 113–114, **114**

S

screen scrapers, multi-tiered application
 architecture and, 10–11, **10**
servers, 19, 21, 118
 business logic tier and 187–189, **188**
 CRUD, 188–189, **189**
 HTTP, 13
 IBM i, 2, 5
 naming, 192
 port numbers and, 118–119, **119**, 126
 query, 188
 runtime, 113–114, **114**
service entry point (SEP), 242, 244, **245**, 246,
 246, 247, **247**
services. *See* web services
Simple Object Access Protocol. *See* SOAP
SOA, 4

SOAP, 15–16, 21, 25, 28, 117–118
 EGL library functions and, 117–118, **117**
 web services and, 36, 120, 124
 spacing text, 156, **156**
 spanning boxes, 58, **58**
 splash screen for, in RDI-SOA, 46, **46**
 split screen mode, 102, **102**
 Spring, 26
 SQL, 19, 20
 step into debugging, 236–238, **238**
 synchronous processing, 134–135, **134**, 137–145

T

3270 as first multi-tiered application, 8–9, **8**
 Target Runtime, 84, **85**
 technology-centric design of EGL, 18
 templates, 100
 service part, 120, **121**
 testing, 93–95, **94**. *See also* debugging
 EGL Rich UI Application and, 144, **144**, **145**,
 180–181, **180**
 thin client web page, 113–115, **113**, **115**
 web services and, 128–130, **128**, **129**, **130**, **131**
 TextLabel, 59, 64–66, **64**
 thick client, 11–12, **11**, 18
 EGL and, 19
 thin client, 18, 34–35, **35**, 97–115
 design of, 97–98
 EGL and, **20**
 folders to organization project for, 98–99, **98**, **99**
 function calls in, 112, **112**
 initializing data for, 110–112
 multi-tiered application architecture and,
 12–13, **12**
 page handlers in, 104, 110, **111**
 test page run for, 113–115, **113**, **115**
 variables for, 103–110
 web page creation for, 100–103, **100–103**
 tiers, number of, 18–19
 timestamp for errors, 224–227, **227**
 TIOBE, 19
 TN5250J, 9
 Tomcat, 13, 18, 19
 two-tiered application, 7–8, **7**
 type of project, 47–48, **48**

U

UI design, 22–23
 Universal Description Discovery and integration
 (UDDI), 15
 updates/saves in WYSIWYG editor, 62
 updating the widget, 152
 URI definition for web service, 38
 user IDs and passwords in, 191–192, 191, 208,
 244, **245**

V

variables, 103–110
 Configure Data Controls wizard for, 106, **107**
 Lines array and, 106–108, **107**
 New EGL Data Variable wizard for,
 105–110, **105**
 Variables view, in debugging, 232, **233**
 views, 46
 Visual Ajax Studio, 26
 Visual Basic, 18

W

Web 2.0, 3, 13
 web page creation, 100–103, **100–103**
 Web perspective, 103
 web service, 35–36, **36**, 117–132
 creating, 120–125
 defining, 38
 EGL library functions and, 117–118, **117**
 EGL Rich UI Application and attachment to,
 175–181, **175–177**
 external references in, 122–123, **123**
 function definitions for, 122, **122**
 listeners for, 37–38
 null values in, 124
 package for, 119, **119**
 parts for, 120, **120**, **121**
 port numbers for, 118–119, **119**, 126, 178
 proxy libraries for, 124
 REST, 36
 sharing, 125–128
 SOAP vs. REST in, 36, 120, 124
 template for, 120, **121**
 testing, 128–130, **128**, **129**, **130**, **131**
 URI definition for, 38

- web service, *continued*
 - Web Services Definition Language (WSDL)
 - and, 28, 125–128, **125**, **126**, **127**, **128**
 - Web Services Explorer and, 129–131, **129**
 - Web Services Definition Language (WSDL), 28, 125–128, **125**, **126**, **127**, **128**
 - Web Services Explorer, 129–131, **129**
 - WebSphere, 13, 18, 19
 - WebSphere Test Environment, 42, 44
 - widgets, 24–25
 - adding to app, using WYSIWYG editor, 54–57, **55**, **56**, **57**, 62–67
 - aligning/centering text in, 71–72, **72**
 - boxes and, 57–59, **57**, **58**
 - designer actions in, 69–70
 - drop zones for, 55, **55**, 63–65, **64**, 69
 - field editing in, 68–69, **69**
 - formatting of, 152–162 **153**, **155**, **156**
 - hierarchy popup for, 65–66, **66**, 69
 - naming, 56, **57**, 60, **60**
 - text added to, 66–67
 - updating, 152
 - width adjustment in, 72–73, **72**, **73**, 156–157, **157**
 - width adjustment, 72–73, **72**, **73**, 156–158, **157**
 - Windows, 9
 - Word, 23
 - Workbench, 45–47, **45–47**
 - workspaces, 45, **45**
 - WSDL. *See* Web Services Definition Language
 - WYSIWYG designer, 4, 14, 23, 26, 41–80, 253, 261
 - absolute positioning in, 59
 - adding EGL Rich UI code using, 78–80, **78**, **79**
 - aligning/centering text in, 71–72, **72**
 - box properties in, 61–62, **61**, **62**
 - boxes in, 57–59, **57**, **58**
 - button in, 59
 - color codes and drop zones, in, 55, **55**, 63–64, **64**
 - Design mode in, 103, **103**
 - dotted line areas in, 56, **56**
 - drag symbol in, **65**
 - drop zones in, 55, **55**, 63–64, **64**, 65, 69
 - editable fields and, 68
 - EGL Rich UI Application and, 145–153
 - events and, adding action to widgets in, 74–77, **75**, **76**, **77**
 - fields and, 68–69, **69**
 - folder creation in, 52, **52**
 - hierarchy popup in, 65–66, **66**, 69
 - maximizing the view in, 75, **75**, 109
 - modes in, 102, **102**
 - nesting boxes in, 58, **58**, 60, **60**
 - new project creation in, 47–59, **47**
 - package creation in, 52
 - page handlers and, 50–51, **51**
 - Palette, 55, **55**
 - panning in, 56
 - perspectives in, 46, 48–50, **49**
 - Properties dialogs in, 61–62, **61**, **62**, 67–68, **67**, **68**, 71, **71**
 - RDI-SOA installation for, 41–44, **42–44**
 - Rich UI application creation with, 50–54
 - Rich UI editor in, 54, **54**
 - Rich UI handler in, 50–51, **51**, 52–54, **53**
 - spanning boxes in, 58, **58**
 - splash screen for, in RDI-SOA, 46, **46**
 - split screen mode in, 102, **102**
 - text added to widget in, 66–67
 - TextLabel in, 59, 64–66, **64**
 - type of project in, 47–48, **48**
 - updates/saves in, 62
 - views in, 46
 - widgets added to app using, 54–57, **55**, **56**, **57**, 62–67
 - width adjustment in, 72–73, **72**, **73**
 - Workbench for, starting up, 45–47, **45–47**
 - workspaces in, 45, **45**
- X**
- XML, 15, 16, 28
 - X–Windows, 2
- Y**
- Yahoo IU (YUI), 255
- Z**
- Zend Studio, 19