

Index

SYMBOLS AND NUMERICS

- { } (braces), errors from omitting, 78
- # (hash sign) before names in CSS, 62, 163
- (minus sign) in Widgets widget, 8
- . (period) before names in CSS, 62
- + (plus sign) in lower-left portion of screen, 4
- " (quotation marks), errors from omitting, 78
- 3D Clock applet, 190–194
- 8-bit Unicode Transformation Format (UTF-8), 26, 72

A

access keys. *See also specific keys*

- for Amazon Album Art widget, 230
- for command-line access, 171–174
- debugging, 78
- for Easy Envelopes widget, 180, 203
- for file system access, 170–171
- for full access, 179–180
- for iPhoto Mini widget, 256–257
- for iTunes Connection Monitor widget, 268
- for Java applet access, 178–179
- for More Widgets widget, 279
- for network access, 174–176
- overview, 169
- for QuickTime plugin, 176, 186
- for SecureCopy widget, 214
- for 3D Clock applet, 191
- for Tile Game widget, 170, 171
- for Timbuktu Quick Connect widget, 245
- for WeatherMaps widget, 174–176
- for WebKit and plugin access, 176
- for Widget plugin access, 177–179

activating the Dashboard, 113–116

Activity Monitor, 117–119, 194

alert() function, 81–83

- AllowFileAccessOutsideOfWidget **access key**
 - for Amazon Album Art widget, 230
 - for iPhoto Mini widget, 256–257
 - overview, 24, 170–171
 - for QuickTime plugin, 186
 - for SecureCopy widget, 214
 - for Tile Game widget, 170, 171

AllowFullAccess **access key, 24, 179–180, 203**

AllowInternetPlugins **access key, 25, 176, 186**

AllowJava **access key, 25, 178–179, 191**

AllowMultipleInstances **property key, 25**

AllowNetworkAccess **access key**

- for Amazon Album Art widget, 230
- debugging connection problems and, 78
- for Easy Envelopes widget, 180
- Info.plist file for, 174–175
- for More Widgets widget, 279
- not required for passing URLs to browser, 176
- overview, 25
- for SecureCopy widget, 214
- for WeatherMaps widget, 174–176

AllowSystem **access key**

- for Amazon Album Art widget, 230
- asynchronous usage and, 173–174
- for command-line access, 171–172
- debugging shell command problems and, 78
- for iTunes Connection Monitor widget, 268
- overview, 25
- synchronous usage and, 173
- for Timbuktu Quick Connect widget, 245
- widget.system() method for, 172, 173–174

Amazon Album Art widget

- access keys, 230
- AppleScripts, 230, 236–237, 238–240
- converting JPEG artwork to PICT format, 239–240
- CSS file, 230, 231–234

Amazon Album Art widget (continued)

- described, 226
- displaying artwork, 234–236
- GetSelectionDetails script, 236–237, 238–240
- HTML file, 230–231
- Info.plist file, 230
- interface, 226–229
- JavaScript functionality, 231, 234–240
- osascript utility used by, 236
- set of files for, 229

Ambrosia Software

- Easy Envelopes widget, 177–178, 179–180, 199–210
- launchSite() function for website, 180

answers to exercises, 291–294

Apple. See also Xcode development environment

- copyright information for widgets, 18, 19–20, 21
- Dashboard Downloads site, 299, 301–302, 304
- Dashboard Widgets site, 9, 299, 302
- Human Interface Guidelines, 160, 295
- Installation Guidelines for Developers, 303–304
- mailing lists, 298
- OS X developers site, 295
- programming information online, 295–298
- tech note on debugging widgets, 78, 296

AppleClasses folder, 95

-apple-dash-board-region property, 125–127

AppleScripts

- for Amazon Album Art widget, 230, 236–237, 238–240
- for Timbuktu Quick Connect widget, 245, 251–252

AppleScrollArea.js file, 146, 147

AppleScrollbar.js file, 146, 147

applets. See Java applets

applications

- AllowFileAccessOutsideOfWidget key and, 170
- bundles for, 33
- extensions for, 32
- viewing contents of, 17
- widgets as, 11
- widgets versus, 13

Aqua Emacs text editor, 36

arrow icons on Widget Bar left and right sides, 7

Automator, 4

B

back side panel. See also preferences for widgets

- adding preference selection, 102–108
- creating, 93–95
- determining size for, 94
- Done button for hiding, 92, 97, 100–102
- flipping the widget to show, 95–102
- info button for showing, 91–92, 97, 99–102

- logos and information on, 92
- pop-up menus for, 103–108
- reading and writing preferences, 108–111

background images. See Default.png file

.backgroundImage style (CSS), 62

backing up development files, 17

BackwardsCompatibleClassLookup property key

- for incorporating applet in a widget, 191
- for iPhoto Mini widget, 256–257
- for More Widgets widget, 279
- overview, 25

Bare Bones Software's TextWrangler text editor, 36

Bart widget, licensing for, 184

BBEdit text editor

- debugging features of, 79
- editing hidden files in, 38, 95
- opening files in Library folders with, 38–39
- saving changes by authenticating, 39
- syntax coloring by, 26, 27
- tuned for website development, 36

Bennet Uk website, 190

blur() function, 123

braces ({ }), errors from omitting, 78

build() function, 287, 288

building

- SimplePlugin in Xcode, 42–47
- WeatherMaps widget, 72–75

bundles, widget, 31–33

C

caching, 13

Calculator widget, 154–155

cancelDefault() method, 164

canceled

- commands run from widgets, 174
- dragging and dropping, 164

canvasDown() function, 205

Cascading Style Sheets. See CSS

case-sensitivity, 30, 78

CFBundleDisplayName property key, 24, 26, 28–29, 30

CFBundleIdentifier property key, 25, 29

CFBundleName property key, 24, 26, 29, 30

CFBundleShortVersionString property key, 26

CFBundleVersion property key, 26, 29, 30

changeCategory() function, 280, 286

charging for widgets, 303

checkboxes in Widgets widget, 8

checkInput() function, 246, 247

class selectors (CSS), 62

clip variable, 155

clipboardData constant, 153**close box, resizing widgets and location of, 145–146**CloseBoxInsetX **property key, 25, 29, 30**CloseBoxInsetY **property key, 25, 29, 30****closing**

commands run from widgets, 174

widgets, 7

command-line access, 171–174**comments, 19–20****connecting to the Internet, 78, 80, 87****console.log file, 52–53, 80–81, 83****Contents folder in application bundles, 33****control regions, 125–127, 140****converting**

images to PNG format, 59

JPEG artwork to PICT format, 239–240

copy command, 222**copy functionality. See pasteboard**copyFrom() **function, 220–221****copying**

adding functionality to WeatherMaps widget, 156–157

files with scp command-line utility, 211–212

widgets to share with other users, 5

widgets to System's Widget folder, avoiding, 6

copyrights

for Apple widgets, 18, 19–20, 21

in CSS files, 19–20

in HTML files, 18, 19

for plugins, 183–184

plugins for protecting, 184–185, 303

for widgets, 185

copyTo() **function, 221****cp command, 222****CPU, widget impact on bandwidth, 117–118****creating a widget. See also WeatherMaps widget**

adding CSS, 61–69

adding HTML, 57–61

adding JavaScript, 63, 69–72

beginning the project, 55–57

building, 72–75

listing features desired, 56

testing, 75

CSS (Cascading Style Sheets). See also ID selectors

for Amazon Album Art widget, 230, 231–234

for applet incorporation in widget, 193–194

body rule, 62

button placement specified by, 97

class selectors, 62

for control regions, 125–127

copyright information in files, 19–20

declaration, 68–69

<div> tags for regions of styles, 63–65

for dragging and dropping, 163–164

for Easy Envelopes widget, 204–206

for flipping a widget, 97–99

for Fortune widget, 18–21

hash sign before names in, 62

hash sign for programmatically changed styles, 163

HTML link to files, 18–19

for iPhoto Mini widget, 260–261

for iTunes Connection Monitor widget, 269–270

mapImage rule, 62

for More Widgets widget, 283–285

period before names in, 62

for pop-up menus, 104–106

positioning text using, 20, 65–69

for Resize widget, 136–139, 142–143

for scrolling, 147–148

for SecureCopy widget, 216–219

selector part, 68

SimplePlugin example, 46–47

testing in Safari, 61

for Timbuktu Quick Connect widget, 249–250

for To Do Tracker widget, 126

for WeatherMaps widget, 61–69, 97–99

widget design provided by, 17

cTempMap() **function, 107****cut functionality. See pasteboard****D****Dashboard**

activation event, 113–116

adding widgets from Widget Bar, 7

cache maintained by, 13

Default.png file for widget interface, 17

described, 4

development mode, 47–48

displaying only widgets used regularly, 8

hiding widgets without removing, 8

installing widgets in, 10

mailing lists, 298

opening with keyboard shortcut, 4

reading and writing preferences, 107–108, 110–111

reference online, 296

reloading widgets in, 13

removing widgets by closing them, 7

removing widgets without installing in, 10, 12

for testing widgets, drawbacks of, 36

tutorial online, 296

websites for, 299

Dashboard Downloads site (Apple), 299, 301–302, 304

Dashboard Lineup website, 299

Dashboard Widgets website (Apple), 9, 299, 302

dataTransfer constant, 153

de Meyer, Jesus (widget creator), 183–184, 253, 275

Debug menu (Safari), 49–50, 51

debugging. See also testing

Apple tech note on, 78, 296

challenges for widgets, 78

code review for, 78–79

fixing bugs, 87–88

JavaScript Console (Safari) for, 50, 51–53, 77

logging exceptions with Safari, 50, 52–53, 80–81

logic errors, 79

overview, 49

printing information with `alert()` function, 81–83

Safari Debug menu for, 49–50

simplifying widget for, 79

stepping through a widget, 83–85

syntax errors, 78

text editor features for, 79

tips and tricks for, 77–83

widget errors, 79

Widgetarium debugger for, 50–51, 83–85

declaration (CSS), 68–69

Default.png file

automatically found by Dashboard, 59

described, 17, 30

finding image size of, 94

for Fortune widget, 19, 30–31

HTML for, 59–60

required for widgets, 30, 79

sizing widget based on dimensions of, 26

for Weather widget, 17, 31, 59

WeatherMaps widget background, 59–61, 63

Widgetarium for creating, 59

deleting. See removing

development environment. See also Widgetarium;

Xcode development environment

BEdit for, 38–39

choosing a text editor, 36

organization features, 35–36

Safari for testing widgets, 36–37

Text Edit for, 36

Dictionary class, 28

directories. See folders

disabling Dashboard development mode, 48

dismissComboBoxes() function, 124

displaying or viewing. See also hiding; opening

application contents, 17

preferences, JavaScript for, 99–100

scrolling pages of widgets on Widget Bar, 7

Widget Bar, 4

widget contents, 15–17

Distorter widget, 178–179, 194

distributing widgets, 301–304

<div> tags

for Amazon Album Art widget, 230–231

for control regions, 126

for creating regions in widgets, 63–65

for Easy Envelopes widget, 204–206

for front and back of widget, 97

ID selectors, 63, 64, 97

for incorporating applet in a widget, 193

for iPhoto Mini widget, 258, 259–261

for iTunes Connection Monitor widget, 268–269

for More Widgets widget, 280–283

for pop-up menus, 104

for preference selection, 103

for QuickTime movie settings, 187

for scrolling, 147

for SecureCopy widget, 214–216

for tiled widget construction, 135

for Timbuktu Quick Connect widget, 246–249, 250

for WeatherMaps widget default map, 123

division markers. See <div> tags

docopy function, 155, 156

Document Object Model (Safari), 297

documentation in application bundles, 33

document.getElementById() method

for Dashboard activation, 116

for pop-up menus, 107–108

in `replaceMap()` function, 110–111

for Resize widget, 140–141, 144

for setting widget focus, 123

for showing and hiding preferences, 99, 100, 101

`widget.system()` method with, 173

docut function, 155

doLoad() function, 115

Done button

CSS for specifying placement, 97, 99

drawn and placed by JavaScript, 97, 99

`hidePrefs()` function for, 100

ID in `<div>` tag for, 97

required for back side panel, 92

testing, 101–102

dopaste function, 155

downloading. See also Internet resources

Dashboard Downloads site, 299, 301–302, 304

Widgetarium, 83

widgets, and installing, 11–13

widgets, websites for, 299

dragdrop function, 164–165, 220

`dragenter` function, 165

dragging and dropping

- adding the `dragover` event, 165–166
- canceling, 164
- CSS file for, 163–164
- drag-and-drop events, 160, 161–162
- Dropper widget example, 160–166
- `event.preventDefault()` function for, 164–165
- `event.stopPropagation()` function for, 164–165
- feedback for user, 160
- from the Finder, 160–161
- HTML file for, 162–163
- Human Interface Guidelines for, 160, 295
- interface extension in OS X, 159
- introduction by Apple, 159
- JavaScript file for, 164–165
- between widgets, 166–167

dragging widgets, 124–125

`dragleave` function, 165

`dragover` function, 165–166

`drawFrontStuff()` function, 207–208

Dropper widget

- adding the `dragover` event, 165–166
- CSS file, 163–164
- HTML file, 162–163
- JavaScript file, 164–165
- programmatically changed CSS styles in, 164
- static CSS styles in, 164
- using, 160–161

E

Easy Envelopes widget (Ambrosia Software)

- access keys, 179–180, 180, 203
- Ambrosia Software website accessed by, 180
- CSS file, 204–206
- described, 199
- EEWPlugin used by, 177–178, 180, 203
- HTML file, 204
- Info.plist file, 177, 203–204
- interface, 199–203
- JavaScript functionality, 177–178, 180, 205, 206–209

EEWPlugin, 177–178, 180, 203

8-bit Unicode Transformation Format (UTF-8), 26, 72

`<embed>` tag for QuickTime plugin, 186

enabling

- Dashboard development mode, 47–48
- debug logging in Safari, 50
- JavaScript Console (Safari), 50
- Safari Debug menu, 49

erasing. *See* removing

errors. *See* debugging

etiquette for using plugins, 183–184

`event.preventDefault()` function, 150, 155, 164–165

events. *See also specific events*

- control regions and, 125–127
- for Dashboard activation, 113–116
- for drag-and-drop, 160, 161–162
- for dragging widgets, 124
- pasteboard, 153–154
- for removing widget preferences, 119–123
- for scrolling, 149–150
- for setting widget focus, 123–124
- system resources and, 117–119
- `event.stopPropagation()` function, 150, 155, 164–165

exercise answers, 291–294

Extensible Markup Language. *See* XML

extensions, 31–32, 43

F

filenames, case-sensitivity of, 30

filesystem

- access keys, 170–171
- domains, 6
- `findAddress()` function, 247
- `findChild()` function, 287, 288

Finder

- dragging and dropping from, 160–161
- installing widgets using, 9–10

finding. *See also* Internet resources

- Default.png image size, 94
- Java applets, 190
- path to command, 173
- plugins on your Mac, 185–186
- URLs for items on pages, 57

Flight Tracker widget, 124, 129–130

`flip()` function, 270, 271

flipping a widget. *See also* back side panel; preferences for widgets

- CSS file for, 97–99
- HTML file for, 95–97
- JavaScript file for, 99–101
- testing, 101–102

`focus()` function, 123

focus, setting for widgets, 123–124

folders. *See also specific folders*

- for developing widgets, 36
- filesystem domains for, 6
- hiding extensions for, 31

folders (continued)

folders (continued)

- installing widgets by dragging into /Library/Widgets, 9
- opening Library files with BBEdit, 38–39
- organizing development files in, 17
- for other users, copying widgets to, 5
- overriding Library folder contents, 6
- for plugins, 185–186
- for property list files, 23
- root-level, 4–5
- System, avoiding copying widgets into, 6
- System, minus sign in Widgets widget and, 8
- users-level, 4, 5
- .wdgt extension for names, 31
- for widget resources, 95
- for widgets displayed in Widget Bar, 4
- for widgets installed by you, 5
- for widgets shipped with Tiger, 4–5
- Xcode preference settings for, 41

Font **property key**, 25

Fortune widget

- building a plugin for, 42–47
- bundling, 31–33
- contents of, 17
- creating Info.plist file for, 28–30
- CSS file, 18–21
- Default.png file, 30–31
- HTML file, 18–19, 47
- JavaScript file, 19, 21–23, 46

Free Java website, 190

functions (JavaScript). See also JavaScript;

specific functions

- calling from HTML, 96
- for Done button, 100
- for info button, 99–100
- for pop-up menus, 107–108
- for printing debug information, 81–83
- <script type="text/javascript"> tag for, 71, 72, 147

G

Gandreas Software's IDE. See Widgetarium

Gas widget, downloading and installing, 11–13

getData **method**, 155

getDetailsFromiTunes() **function**, 236–237

GetSelectionDetails AppleScript, 236–237, 238–240

getside() **function**, 271

Gmail Checker widget preferences, 93

goAmazon() **function**, 234–235

GoogleMail widget, 188

gotoDashboardSite() **function**, 280

grabClipboardAddress() **function**, 177–178

GraphicConverter tool, 94

graphics. See **converting; PNG**

Graphics Converter program, 59

H

hash sign (#) before names in CSS, 62, 163

Height **property key**, 25, 26

Hello World widget, installing, 9–10

hidden files, editing in BBEdit, 38, 95

hidden settings, 93

hidePrefs() **function**, 100

hiding

- extensions for folders, 31
- preferences, JavaScript for, 100
- widgets without removing from Dashboard, 8

HIG (Human Interface Guidelines), 160, 295

HREF **tags (HTML)**, 69

hTempMap() **function**, 107–108

HTML. See also specific tags

- for Amazon Album Art widget, 230–231
- for applet incorporation in widget, 182–183
- calling JavaScript functions from, 96
- comments, 19
- copyright information in files, 18, 19
- for Default.png file, 59–60
- for dragging and dropping, 162–163
- dynamic, XMLHttpRequest object for, 296
- for Easy Envelopes widget, 204
- for flipping a widget, 95–97
- for Fortune widget, 18–19, 47
- for iPhoto Mini widget, 257–259
- for iTunes Connection Monitor widget, 268–269
- links to CSS and JavaScript files, 19
- for More Widgets widget, 280–283
- opening code in Text Edit, 18
- for QuickTime plugin, 187
- for Resize widget, 135, 141–142
- resizing images as they load, 58–59
- for retrieving maps, 57–59
- Safari reference, 297
- for scrolling, 146–147
- for SecureCopy widget, 214–216
- syntax coloring for ease of reading, 36
- testing in Safari, 60–61
- for Timbuktu Quick Connect widget, 246–249
- Unicode format for files, 26
- for WeatherMaps widget, 57–61, 95–97, 103, 123
- widget structure provided by, 15

Human Interface Guidelines (HIG), 160, 295

i icon on widgets, 4, 91, 101**iBook screen space limitations, 129****icon.png file, 17, 30, 73–74****ID selectors**

- for buttons used to flip widget, 97
- for <div> tags, 63, 64
- for dragging and dropping, 163–164
- for incorporating applet in a widget, 193
- for iPhoto Mini widget, 259, 260
- for items to be displayed, 64–65
- for More Widgets widget, 283–285
- overview, 68
- placing images via JavaScript using, 63
- for Resize widget, 136–139, 142–143
- for scrolling, 147–148
- for SecureCopy widget, 216–219
- for tiled widget construction, 136–139
- for Timbuktu Quick Connect widget, 249–250
- using only once per HTML file, 62
- for WeatherMaps widget, 62–63, 64–65

IDE (Integrated Development Environment), 39.

See also specific IDEs

id.rsa file, 222**id.rsa.pub file, 222****images. See converting; PNG**

 tag, **Default.png file linked by, 19**

info button

- CSS for specifying placement, 97, 99
- drawn and placed by JavaScript, 97, 99
- i icon for, 4, 91, 101
- ID in <div> tag for, 97
- required for widget preferences, 92
- showPrefs() function for, 99–100
- testing, 101–102

Info.plist file. See also access keys; property lists

- access keys available for, 169
- AllowFileAccessOutsideOfWidget key usage, 171
- AllowInternetPlugins key usage, 176, 186
- AllowJava key usage, 178
- AllowNetworkAccess key usage, 174–175
- AllowSystem key usage, 171–172
- for Amazon Album Art widget, 230
- configuration information in, 23
- for Easy Envelopes widget, 177, 203–204
- for Fortune widget, 28–30
- for LiveWidget widget, 188–189
- for More Widgets widget, 279
- Plugin key usage, 177
- required for widgets, 32, 79

- for Resize widget, 144–145
- for SecureCopy widget, 214
- for SimplePlugin, 43, 45–46
- syntax errors from omitting access keys, 78
- for Tile Game widget, 170–171
- for Timbuktu Quick Connect widget, 245–246
- for Uptime widget, 171–172
- for Weather widget, 24
- for WeatherMaps widget, 72–73, 174–175
- Xcode creation of, 43

installed (hidden) files, editing, 38, 95**installing widgets**

- Apple guidelines for developers, 303–304
- automatically, improved security for, 9, 10
- changed in Tiger 10.4.2 release, 9
- by double-clicking the bundle folder, 33
- after downloading, 11–13
- by dragging into /Library/Widgets folder, 9
- Finder for, 9–10
- folder for, 5
- removing without installing in Dashboard, 10, 12

Integrated Development Environment (IDE), 39.

See also specific IDEs

interface of widgets. See also back side panel;**Default.png file**

- Amazon Album Art widget, 226–229
- Dashboard, Default.png file for, 17
- Easy Envelopes widget, 199–203
- Human Interface Guidelines for, 160, 295
- iPhoto Mini widget, 254–255
- iTunes Connection Monitor widget, 266–267
- More Widgets widget, 275–278
- preferences, PNG file for, 17
- SecureCopy widget, 212–213
- Timbuktu Quick Connect widget, 242–244

Internet resources

- Apple Installation Guidelines for Developers, 303–304
- Apple OS X developers site, 295
- Bennet Uk site, 190
- Dashboard Downloads site, 299, 301–302, 304
- Dashboard Lineup site, 299
- Dashboard programming information, 295–296
- Dashboard Widgets site, 9, 299, 302
- debugging guide, 296
- Free Java site, 190
- Gmail Checker widget, 93
- Human Interface Guidelines, 295
- Mac OS X Hints site, 299
- MacDevCenter site, 299
- MacTech website, 299
- mailing lists, 298

Internet resources (continued)

- maps for WeatherMaps widget, 58
- QuickTime movie embedding information, 186
- reference for Dashboard, 296
- Safari information, 297
- 3D Clock applet, 190
- tutorial for Dashboard, 296
- Universal Binaries Programming Guidelines, 297–298
- WeatherMaps widget source code, 103
- WebKit information, 298
- widget distribution sites, 301–302
- Widgetarium site, 83
- XMLHttpRequest object information, 296
- Zaptastic widget, 9

`ipAddressAutoComplete()` **function, 246, 247, 252**

iPhoto Mini widget

- access keys, 256–257
- Actions menu, 254
- Albums menu, 254
- benefits of, 253
- CSS file, 260–261
- HTML file, 257–259
- Info.plist file, 256–257
- interface, 254–255
- iPhotoLoader.widgetplugin, 183–184
- JavaScript functionality, 260, 261–264
- set of files for, 255–256

iPhoto program, 253. See also iPhoto Mini widget

iTunes Connection Monitor widget

- access keys, 268
- CSS file, 269–270
- described, 265–266
- free distribution of, 184
- HTML file, 268–269
- Info.plist file, 268
- interface, 266–267
- JavaScript functionality, 270–273
- set of files for, 267
- whichside preference, 270

iTunes music player, 225–226, 236–237, 265

iTunes Music Store, 225

J

Java applets

- access key for, 25, 178–179
- CSS file for, 193–194
- HTML file for, 192–193
- incorporating in widgets, 190–194
- Info.plist file for, 191–192

JavaScript. See also specific functions

- for Amazon Album Art widget, 234–240
- buttons drawn and placed by, 97
- calling functions from HTML, 96
- case-sensitivity of variables, 78
- debugging with Safari, 49–50, 51–53
- debugging with Widgetarium, 50–51, 83–85
- for dragging and dropping, 164–165
- for Easy Envelopes widget, 206–209
- for flipping a widget, 99–101
- for Fortune widget, 19, 21–23, 46
- HTML link to files, 19
- for iPhoto Mini widget, 261–264
- for iTunes Connection Monitor widget, 270–273
- limitations of, 184
- logging exceptions with Safari, 50, 52–53
- for More Widgets widget, 285–289
- placing images using ID selector, 63
- pop-up menu functions, 107–108
- for Resize widget, 140–141, 143–144
- resources in AppleClasses folder, 95
- Safari information online, 297
- `<script type="text/javascript">`
tag for, 71, 72, 147
- for scrolling, 148–151
- for SecureCopy widget, 220–221
- syntax coloring for ease of reading code, 36
- for Timbuktu Quick Connect widget, 250–252
- using plugins with, 184
- for WeatherMaps widget, 69–72, 99–101
- widget functionality provided by, 17
- widget object not available in Safari, 38

JavaScript Console (Safari), 51–53, 77, 80–81

JPEG images, converting to PICT format, 239–240

L

`lastScalarSliderValue` **variable, 149**

`launchSite()` **function, 180**

`launchTB2()` **function, 246, 247, 250, 252**

/Library folders, BBEdit for opening files in, 38–39

/Library/Internet Plug-ins folder, 185–186

/Library/Preferences folder, 23

/Library/Widgets folders, 4–6, 8, 9. See also folders

live resizing. See relative resizing

LiveJournal blog site, 188

LiveWidget widget, 188–190

`loadAlbums()` **function, 261–262**

`load_info()` **function, 272–273**

loading

- calling JavaScript functions during, 96
- reloading keyboard shortcuts, 13

reloading widgets, 13
 resizing images during, 58–59
 widgets by invoking Dashboard, 114
`loadXMLDocument()` **function**, 286–287

logging

`alert()` function results, 83
 JavaScript exceptions with Safari, 50, 52–53, 80–81, 83
 limitations for debugging, 77
 test results, 86–87

logic errors, debugging, 79

M

Mac OS X Hints website, 299

MacBook screen space limitations, 129

MacDevCenter website, 299

MacTech magazine, 298

MacTech website, 299

mailing lists, 298

`MainHTML` **property key**, 24, 25, 29

Marcel Bresink Software-Systeme's TinkerTool, 49

memory

applet use of, 194
 widget impact on, 118

Meyers, Stephan (widget creator), 9

`minimize()` **function**, 206–207

minus sign (-) in Widgets widget, 8

monitoring widgets, 117–119

Monkey Business Labs' To Do Tracker widget, 125–126, 166

More Widgets widget

access key, 279
 CSS file, 283–285
 described, 275
 HTML file, 280–283
 Images folder, 283
 Info.plist file, 279
 interface, 275–278
 JavaScript functionality, 280, 285–289
 set of files for, 278–279
`widget.setPreferenceForKey()` method, 286

`mouseDown()` **function**

for iPhoto Mini widget, 262–263
 for Resize widget, 140, 143
 for scrolling, 149

`mouseMove()` **function**

for iPhoto Mini widget, 263, 264
 for Resize widget, 140, 143–144
 for scrolling, 149

`mouseUp()` **function**

for iPhoto Mini widget, 262, 263–264
 for Resize widget, 141, 144
 for scrolling, 149–150

MyPluginClass.m file, 46–47

N

Netopia

Timbuktu Quick Connect widget, 241–252
 Timbuktu screen-sharing program, 241

network access, 174–176

`nextImage()` **function**, 260

NotePad widget (Widget Machine), 184–185

notepad.widgetplugin, 184–185

O

Objective C language, 183

`onbeforecopy` **event**, 153–154

`onbeforecut` **event**, 153–154

`onbeforepaste` **event**, 153–154

onChange script for pop-up menus, 107

`oncopy` **event**, 153–154

`oncut` **event**, 153–154, 156

`ondrag` **event**, 161–162

`ondragend` **event**, 161–162

`ondragenter` **event**, 162, 166

`ondragleave` **event**, 162, 166

`ondragover` **event**, 162, 166

`ondragstart` **event**, 161–162

`ondragstart()` **function**, 124

`ondrag.start` **handler**, 124

`ondrag.stop` **handler**, 124

`ondrop` **event**, 162, 164

`onhide()` **function for Dashboard activation**, 115, 116

`onHide()` **function (WeatherMaps widget)**, 121

online resources. *See Internet resources*

`onload='setup()'` **for calling JavaScript functions**, 96

`onpaste` **event**, 153–154

`onRemove()` **function**

adding to WeatherMaps widget, 121–123
 in Weather widget, 119–121

`onremove()` **function (iTunes Connection Monitor)**, 270–271

`onshow()` **function**

for Dashboard activation, 115, 116
 for iTunes Connection Monitor widget, 270

`onShow()` **function (WeatherMaps)**, 121, 123

`openInApp()` **function**, 262

opening. *See also displaying or viewing; enabling*

`AllowFileAccessOutsideOfWidget` key and, 170
 closing widgets without, 7
 Dashboard with keyboard shortcut, 4
 files in Library folders with BBEdit, 38–39
 HTML code in Text Edit, 18
 JavaScript Console (Safari), 51

opening (continued)

- Safari from Widgets widget, 9
- Widget Installer without Safari, 13
- Widgets widget, 8
- Xcode for the first time, 41–42

OS X Tiger

- Apple developers site, 295
- automatic widget installation security changes, 9, 10
- cat names for earlier versions, 3
- evolution of, 3
- filesystem domains, 6
- Human Interface Guidelines, 160, 295
- Mac OS X Hints site, 299
- new features, 4
- widget suite shipped with, 4–5

osascript utility, 236, 245

`oTempMap()` function, 107

`outputString` property, 172–173

P

package contents, viewing, 16–17

parameters, debugging and, 79

`parseXMLDocument()` function, 287–288

PasswordPlugin, 189–190

passwords

- for LiveWidget widget, 189–190
- for scp command-line utility, 212
- for SecureCopy widget, eliminating, 221–223

paste functionality. *See* pasteboard

pasteboard

- adding handlers to WeatherMaps widget, 156–157
- Calculator widget implementation, 154–155
- constants, 153
- events, 153–154
- handlers, 154–155

period (.) before names in CSS, 62

PICT format, converting JPEG images to, 239–240

PID number, 80

Plugin access key

- for Easy Envelopes widget, 177–178
- for Fortune widget, 29, 30
- overview, 25
- for SimplePlugin example, 45, 46

plugins

- access key for, 25, 176
- for copyright protection, 184–185, 303
- defined, 183
- errors from failing to include, 79
- etiquette for using, 183–184
- incorporating in widgets, 188–190
- LiveWidget widget example, 188–190

- logic errors from forgetting to call, 79
- Objective C language for, 183
- QuickTime, 186–188
- SimplePlugin example, 42–47
- uses for, 184–185

plus sign (+) in lower-left portion of screen, 4

PNG (Portable Network Graphics). *See also specific files*

- converting other formats to, 59
- for icons of widgets, 17, 30
- for preferences interface of widgets, 17, 95
- use by widgets, 15
- for user interface of widgets, 17, 30
- for Weather widget, 17

pop-up menus

CSS for, 104–106

`<div>` tags for, 104

JavaScript functions for, 107–108

`onChange` script for, 107

for preference selection, 103–108

`<select>` tags for, 103

posting widgets, 303–304

PowerBooks, 4, 129

preferences for widgets

- adding preference selection to widget, 102–108
- creating back side panel for, 93–95
- flipping the widget for, 95–102
- hidden settings for, 93
- i* icon for, 4, 91, 101
- JavaScript for showing and hiding, 99–100
- JavaScript needed for, 17
- need for, 91, 93
- overview, 91–93
- PNG files for, 17
- pop-up menus for, 103–108
- reading and writing, 108–111
- removing, 119–123
- resources for, 95
- setting, 4, 91–92
- writing, access key and, 171

preferences for Xcode, 41–42

`prevImage()` function, 260

printing debug information using `alert()` function, 81–83

private key for SecureCopy widget, 221–223

`processDownloadedImage()` function, 237–240

`processXMLRequest()` function, 286, 287

projects, beginning, 55–57

Property List Editor, 26, 27, 28–30

property lists. *See also access keys; Info.plist file*

- creating, 26–30
- defined, 23
- Dictionary class for root element, 28

editing, 26
 errors from omitting required keys, 79
 folder for files, 23
 Info.plist file for, 23
 property keys in (table), 24–26
 text editors for, 26
 XML header for files, 26–27

public key for SecureCopy widget, 221–223

Q

QuickTime plugin, 176, 186–188

quotation marks ("), errors from omitting, 78

R

radar maps. See WeatherMaps widget

`radMap()` **function, 82, 107**

RDF (resource description framework), 275

refreshing widgets, 113–116

relative resizing

close box location and, 145–146
 CSS file for, 136–139, 142–143
 determining need for, 132
 HTML file for, 135, 141–142
 JavaScript file for, 140–141, 143–144
 making widgets resizable, 132–146
 single panel for, 132, 141–145
 thumb control for, 132, 133, 135, 140–141
 ways of providing, 132–134

reloading widgets or keyboard shortcuts, 13

removing. See also hiding

disabling Dashboard development mode, 48
 preferences for widgets, 119–123
 uninstalling widgets, 8
 widgets from Dashboard, 7
 widgets without installing in Dashboard, 10, 12

`replaceMap()` **function**

getting stored preferences, 110–111
 HTML file for, 70–71
 JavaScript code for, 69–70, 71–72, 110–111
`<script type="text/javascript">` tag for, 71, 72

Resize widget

CSS file, 136–139, 142–143
`<div>` tags for tiled construction, 135
 HTML file, 135, 141–142
 Info.plist file, 144–145
 JavaScript file, 140–141, 143–144
 link to thumb control image, 135
 using single background panel, 141–145
 using tiled construction, 132–141

resizing

close box location and, 145–146
 CSS file for, 136–139, 142–143
 determining need for, 132
 dynamic information and, 130–131
 HTML file for, 135, 141–142
 images as they load, HTML for, 58–59
 images in Widgetarium, 59
 JavaScript file for, 140–141, 143–144
 keeping image proportions when, 59
 making widgets resizable, 132–146
 single panel for, 132, 141–145
 thumb control for, 132, 133, 135, 140–141
 tiled construction for, 132–141
 widget examples illustrating, 129–131

resource description framework (RDF), 275

`restoreUserLogin()` **function, 190**

revisions. See versions of widgets

Rich Site Summary (RSS), 275. See also RSS feeds

Rogers, Nick (widget creator), 241

rollie, 91

RSS feeds, 146, 275. See also More Widgets widget

S

Safari

cache maintained by, 13
 as debugger, 49–50, 51–53
 Document Object Model overview, 297
 interaction with Widget Installer, 11
 JavaScript widget object not available in, 38
 online resources, 297
 opened by Widgets widget More Widgets button, 9
 opening Widget Installer without, 13
 reloading pages in, 13
 testing CSS in, 61
 testing HTML in, 60–61
 testing widgets in, 36–37
 widget download confirmation by, 11
 widget preferences access not supported by, 95
`saveUserLogin()` **function, 189**
scalarSlider selectors, 148
scalarSlider variable, 148, 149
scp command-line utility, 211–212, 220–221. See also SecureCopy widget
scp.css file, 216–219
scp.html file, 214–216
scp.js file, 214
`<script type="text/javascript">` **tag, 71, 72, 147**

scrolling

- CSS file for, 147–148
- determining need for, 146
- HTML file for, 146–147
- JavaScript file for, 148–151
- pages of widgets on Widget Bar, 7
- resources in AppleClasses folder, 95, 146
- tracking slider changes, 150–151

`searching()` **function**, 271

secure shell (SSH) suite, 211

SecureCopy widget

- access keys, 214
- CSS file, 216–219
- eliminating need for password, 221–223
- HTML file, 214–216
- Info.plist file, 214
- interface, 212–213
- JavaScript functionality, 220–221
- scp command-line utility overview, 211–212
- set of files for, 213
- using with public and private keys, 221–223
- `widget.system()` method used by, 212, 220–221

security

- for automatic widget installation, 9, 10
- for `getData` method, 155
- for transferring files, 211

`<select>` **tags for preference selection**, 103

selectors. See ID selectors

`setAlbumArt()` **function**, 231

`setCloseBoxOffset()` **method**, 145–146

`setData` **method**, 155

`setup()` **function**

- for Resize widget, 141
- for scrolling, 150
- for SecureCopy widget, 220
- for WeatherMaps widget, 175
- for WeatherMaps widget copy command, 156, 157

sharing

- music over a network with iTunes, 265
- plugins, etiquette for, 183–184
- Timbuktu screen-sharing program, 241
- widgets with other users on your Mac, 5

`showAddressFontPanel()` **function**, 208, 209

`showDetails()` **function**, 289

showing. See displaying or viewing; hiding

`showPageSetup()` **function**, 205, 208–209

`showPrefs()` **function**, 99–100

`showReturnAddressFontPanel()` **function**, 208, 209

SimplePlugin example

- building in Xcode, 42–47
- CSS file, 46–47

- frameworks for, 43
- linked to widget through Info.plist file, 45–46

SimplePlugin folder, 42

skEdit text editor, 36

Ski Report widget, 83–85

`slideClosed()` **function**, 235

`slideClosedCompleted()` **function**, 236

`slideIt()` **function**, 235–236

`slideOpen()` **function**, 235

slider resources in AppleClasses folder, 95

`sliderChanged()` **function**, 150–151

slogin command-line utility, 211

Sort menu (Widgets widget), 8

sorting widgets in Widgets widget, 8

spelling errors, 78, 99

Spotlight, 4

ssh command-line utility, 211

SSH (secure shell) suite, 211

ssh-keygen -t rsa command, 222

`startTB2Connection()` **function**, 246, 250–252

Stocks widget, 80, 87, 92, 130, 131

stopping commands run from widgets, 174

`sWeatherMap()` **function**, 108, 110

syntax coloring, 26, 27, 36, 79

syntax errors, debugging, 78

sys-keygen command, 221–222

System filesystem domain, 6, 8

system resources, 114, 117–119

T

temperature maps. See WeatherMaps widget

Terminal application, 211–212, 221–222

testing. See also debugging

- allowing feature requests from testers, 87
- beta testers for, 86–87
- Dashboard drawbacks for, 36
- design problems discovered by, 77
- fixing bugs, 87–88
- flipping a widget, 101–102
- HTML file in Safari, 60–61
- logging results, 86–87
- Safari for, 36–37
- trying to break your widget, 75, 86
- using different machines for, 75
- WeatherMaps widget, 75, 101–102
- widget actions not supported by Safari, 95
- by yourself, limitations of, 86

Text Edit, opening HTML code in, 18

text editors. See also specific editors

- choosing, 36
- debugging features of, 79

syntax coloring by, 26, 27, 36, 79
with Unicode or UTF-8 capabilities, 26
XML-specific, 26

TextWrangler text editor (Bare Bones Software), 36

3D Clock applet, 190–194

Tiger. See OS X Tiger

Tile Game widget, 170–171

tiled widget construction. See Resize widget

Timbuktu Quick Connect widget (Netopia)

access keys, 245
AppleScript, 245, 251–252
CSS file, 249–250
described, 241
HTML file, 246–249
Info.plist file, 245–246
interface, 242–244
JavaScript functionality, 246, 247, 250–252
osascript utility used by, 245
set of files for, 244–245

Timbuktu screen-sharing program (Netopia), 241

TinkerTool (Marcel Bresink Software-Systeme), 49

**To Do Tracker widget (Monkey Business Labs),
125–126, 166**

`toggleBarcodes()` **function, 180**

`toggleSlide()` **function, 231, 235**

trashing. See removing

U

Unicode format for property lists, 26

uninstalling widgets, 8

Universal Binaries Programming Guidelines, 297–298

updating widgets, 113–116

`uptime()` **function, 172**

Uptime utility, 172

Uptime widget, 171–173

URLs

finding for items on pages, 57
`widget.openURL()` method for, 174–176

User filesystem domain, 6

UTF-8 (8-bit Unicode Transformation Format), 26, 72

V

validating data entry, 87

version.plist file, 73

versions of widgets

formal numbers versus build numbers, 26
multiple, Dashboard limitations for testing, 36
property keys for numbers, 26
version.plist file for documenting, 73

Victor, Bret (widget creator), 184

viewing. See displaying or viewing; hiding

W

.wdgt extension for folder names, 31

Weather widget

contents of, 15–17
Default.png file, 31
Info.plist file, 24
preferences, 91–92
preferences cleared by, 119–121

`weatherLogo()` **function, 126**

WeatherMaps widget

access keys, 174–176
activating the Dashboard, 116
adding links to other maps, 63–69
`alert()` function in, for debugging, 81–83
beginning the project, 55–57
building the widget, 72–75
control regions, 126–127
copy command for, 156–157
CSS for, 61–63, 65–69, 97–99
`cTempMap()` function, 107
default map, 123
Default.png file, 59–61, 63
<div> tags for creating page regions, 63–65
features list, 56
finding the URLs for maps, 57
flipping, 95–102
`hTempMap()` function, 107–108
HTML for background image, 59–61
HTML for default map, 123
HTML for flipping, 95–97
HTML for preference selection, 103
HTML for retrieving maps, 57–59
HTML incorporating CSS, 62–63
Icon.png file, 73–74
ID selectors for maps, 64–65
Info.plist file, 72–73, 174–175
JavaScript for, 69–72, 99–101
network access, 174–176
`oTempMap()` function, 107
pasteboard handlers for, 156–157
placing the map on the background image, 63
pop-up menus for preferences, 103–108
preference selection for, 102–108
preferences needed for, 91
`radMap()` function, 82, 107
reading and writing preferences, 107–111
removing preferences, 121–123
`replaceMap()` function, 69–72, 110–111
resizing maps as they load, 58–59
`setup()` function, 175
size of maps and design issues, 56–57
source code online for, 103

WeatherMaps widget (continued)

WeatherMaps widget (continued)

`sWeatherMap()` function, 108, 110
testing, 75, 101–102
URLs possibly changed for, 102
version.plist file, 73

Web resources. See Internet resources

WebKit

access key for, 25, 176
default behavior with `ondrop` event, 164
`findChild()` function, 287, 288
information online, 298
plugins on your Mac, 186
XMLHttpRequest object, 286, 296

Welch, Andrew (widget creator), 199

Whitaker, Simon (widget creator), 226

Widget Bar

Icon.png file for, 30
using, 4–5, 7–8

Widget Installer, 10, 11, 13, 33.

See also installing widgets

Widget Machine's NotePad widget, 184–185

Widgetarium (Gandreas Software)

creating Default.png file in, 59
debugger, 50–51, 83–85
overview, 40, 83
saving images in PNG format, 59
website, 83

widget-com.argon18.widget.itcm.plist file, 270

`widget.onhide` property, 114–116, 121

`widget.onreceiverrequest` property, 115

`widget.onremove` property, 115, 121

`widget.onshow` property, 114–116, 121

`widget.openURL()` method, 176

`widget.preferenceForKey()` method

AllowFileAccessOutsideOfWidget function and, 171
in `onshow()` function, 116
in `replaceMap()` function, 110–111
for writing preferences, 108

Widget/Resources folder, 95, 146

Widgets widget

determining widgets you installed, 8
Icon.png file for, 30
overview, 8–9
resizing example, 130, 131
uninstalling widgets, 8

`widget.setPreferenceForKey()` method

AllowFileAccessOutsideOfWidget function and, 171
LiveWidget widget use of, 189

More Widgets widget use of, 286

for pop-up menus, 107–108

in `replaceMap()` function, 110

`widget.system()` method

for AllowSystem access key, 172

asynchronous usage, 173–174

SecureCopy widget use of, 212, 220–221

synchronous usage, 173

Width property key, 25, 26

Wikipedia widget, 166, 167

Wimic widget, 80

`window.onblur` handler, 123

`window.onfocus` handler, 123, 124

`window.resizeTo()` method, 141, 144

X

Xcode development environment (Apple)

building a plugin, 42–47

directory preferences for build files, 41

extension for projects, 43

frameworks, 43

as granddaddy of OS X IDEs, 39

Info.plist file creation by, 43

installing widget from, 9–10

keeping widget copies in, 10

overview, 41

Property List Editor with, 26, 27, 28–30

saving project window states when closing, 41–42

starting for the first time, 41–42

warning for files saved with earlier version, 42

widgets included with, 9, 17

XML (Extensible Markup Language)

dynamic, XMLHttpRequest object for, 296

feeds, 93

for property lists, 23, 26–27

resources in AppleClasses folder, 95

XMLHttpRequest object (WebKit), 286, 296

Y

Yee, Jason (widget creator), 184, 265

Z

Zaptastic widget, obtaining, 9