

Index

Symbols and Numbers

- _** (underscore symbol), MXML naming conventions and, 128
- { }** (curly braces), data binding and, 118, 231–235
- ==** (equality), 590
- ===** (strict equality), 590
- ?:** (conditional operator), **ActionScript**, 46
- .** (dot) notation
 - ActionScript**, 42
 - ColdFusion**, 977
- ()** (parentheses), in **ActionScript** syntax, 42
- 3D** effects, **Flex 4**, 458
- 3D** graphics API, in **Flash Player 10**, 79–81
- 40KB** message limit, working around in **AIR**, 793–794

A

AAC/AAC+

- Flash Player supporting, 539
- usable media formats, 514

absolute positioning. *See also* coordinate positioning

- CSS styles and, 415
- defined, 112

abstract classes, in OOP languages, 1361–1362

abstraction, benefits of MVC frameworks, 1080

AbstractTarget class, in logging framework, 1342

access modifiers

- ActionScript**, 61
- comparing OOP languages, 1362

Accordion container, as navigation

component, 265

acknowledgement events, LCDS messages and, 984–985

Acrobat Reader, 25

Acrobat.com, 6

Action Message Format. *See* AMF (Action Message Format)

ActionScript

- advanced constraints in, 278
- application development and, 104
- AppSettings** class, 284–285
- benefits of **Flex** and, 12
- BindingUtils** class, 780
- BrowserManager** API, 733
- calling embedded JavaScript from, 726–728
- calling from JavaScript, 728–729
- calling Web Services from, 924–925
- class naming conventions, 1143–1144
- code assist and, 171
- creating **ActionScript** project in **Flex Builder**, 120–123
- data binding and, 232–234, 380–381
- data type conversion to/from JavaScript, 1019
- dynamic classes, 589
- event listeners, 240
- exception handling, 860
- as **Flex Builder** project type, 106
- as **Flex** source file, 3, 218
- formatter classes and, 226
- forms created with, 281
- history of **Flex** and, 18
- HTML DOM controlled from, 729
- invoking from JavaScript, 708–709
- as language in **Flex 3** ecosystem, 27–28
- LCDS classes for, 955
- libraries, 397
- loading SWF files, 506
- manually invoking effects, 448
- mapping ID3 tags to, 526
- mapping to Java types, 1065

ActionScript (continued)

- mapping to XML schema types, 911
- metadata added to, 372–373
- parameter binding vs. explicit parameter passing in RPC, 838–839
- passing complex objects with Flex, 713–716
- passing parameters with Flex, 709–711
- passing return values with Flex, 712
- pop-ups created with, 288
- regular expressions, 607
- RPC (remote procedure call) and, 838
- sound embedded in, 515–517
- states in, 391–393
- WebService component in, 844–845

ActionScript, comparing OOP languages

- access modifiers, 1362
- classes, filenames, and packages, 1360–1361
- console output and, 1357
- data types and, 1357–1359
- declaring variables, 1359–1360
- dynamic classes, 1362
- global variable notation, 1357
- inheritance, 1362
- memory management, 1356
- overview of, 1355
- polymorphism, 1361–1362
- regular expressions, 1360
- runtime environments, 1355–1356
- statement termination, 1360
- summary, 1363
- type casting, 1359
- variable hoisting, 1357

ActionScript, custom components

- adding child components, 325–326
- base class, 330–331
- company logo example, 320
- `CompanyLogo` class, 321–323
- component measurement, 327–328
- component structure, 323
- extending `UIComponent` and `IUIComponent`, 320–321
- invalidate functions, 324–325
- `invalidateProperties()`, for applying properties, 326–327
- layout, 328–330
- MXML components compared with, 343–344

- overview of, 319–320
- properties and life cycle and, 323–324
- rocker switch example, 333–340
- sizing child components, 332–333
- summary, 341

ActionScript, fundamentals

- access modifiers, 61
- arrays, 50–52
- casting, 54
- class attributes, 59
- class members, 59
- conditional statements, 44–46
- constants, 63
- control structures, 44
- getters and setters, 63
- interfaces and inheritance, 64–65
- introspection, 55–57
- loops, 46–47
- namespaces, 61–62
- numeric types, 48
- `Object` class, 58
- object introspection, 55
- objects, 53
- OOP (object-oriented programming) and, 57
- overview of, 41
- packages and classes, 58–59
- properties and methods, 59–61
- reserved keywords, 43
- static properties and methods, 63
- strings, 49–50
- summary, 65
- syntax, 42
- value types, 47–48
- variables and data types, 44
- vectors, 52–53

ActionScript, in MXML

- functions, 145–146
- inline ActionScript, 141–143
- linking external ActionScript, 146–147
- overview of, 141
- rules of usage for ActionScript, 147
- `Script` tag, 144–145
- when to use ActionScript vs. MXML, 147–148

ActionScript, working with

- 3D graphics and pixel bender in Flash Player 10, 79–81

- AIR APIs, 78–79
- Display API, 74–78
- display architecture, 68–69
- event model, 69–72
- Flash platform and, 67–68
- FTE (Flash Text Engine) in Flash Player 10, 81
- media APIs in Flash Player 10, 82–83
- new APIs in Flash Player 10, 79
- summary, 83
- system integration APIs in Flash Player 10, 82
- ActionScript, XML and**
 - accessing XML objects using E4X operators and expressions, 579
 - accessing XML objects using XML class
 - methods, 578–579
 - before E4X, 573–576
 - miscellaneous debris, 581
 - modifying XML, 580
 - Namespace class, 577
 - QName class, 577
 - using E4X, 576
 - XML class, 576–577
 - XMLList class, 577
- ActionScript Editor, 483–484**
- ActionScript object adapter, for DMS, 1014**
- ActionScriptAdapter, BlazeDS integration points, 1055**
 - .actionscriptProperties, 150
- adapters**
 - BlazeDS, 1055
 - DMS, 1014–1015
 - LCDS, 959
- adaptive polling, in LCDS, 1037**
- administrator screens, ColdFusion, 973–975**
- Adobe AIR Update Framework, 31**
- Adobe Bug and Issue Management System**
 - bug tracking and, 214
 - overview of, 1262
 - registering/logging in, 1262
 - reporting bugs, 1262–1264
 - watching existing bug report, 1264–1265
- Adobe Buzzword, 1259**
- Adobe Creative Suite. See CS3 suite**
- Adobe Data Services Stress-Testing Framework, 31**
- Adobe Flash Collaboration Service (AFCS), 38–39**
- Adobe Flex Software Development Kit (SDK)**
 - ASDoc packaged in, 1203
 - development enhancements in Flex Builder 3, 99
 - Flex 3 ecosystem, 29
- Adobe Illustrator, 99**
- Adobe Image Foundation (AIF). See Pixel Bender**
- Adobe Integrated Runtime. See AIR (Adobe Integrated Runtime)**
- Adobe LiveDocs, 977**
- Adobe Media Rights Management Server, 514**
- Adobe Photoshop, 99**
- Adobe Spry Framework, 31**
- Adobe Wave, 39**
- ADO.NET Entity Framework, 916**
- adt utility, for code signing from command line, 1222
- advanced components, development enhancements in Flex Builder 3, 99**
- advanced constraints**
 - in ActionScript, 278
 - constraint columns, 276–277
 - constraint rows, 277–278
 - constraint sizes, 278
 - development enhancements in Flex Builder 3, 99
 - overview of, 275–276
- AdvancedDataGrid control, 637–639
 - grouping data in, 638–639
 - hierarchical data in, 639
 - overview of, 637
- AES-CBC 128-bit encryption, 831**
- AFCS (Adobe Flash Collaboration Service), 38–39**
- aggregations, applying to OLAPDataGrid control, 644**
- AIF (Adobe Image Foundation). See Pixel Bender**
- AIR (Adobe Integrated Runtime)**
 - 40KB message limit, 793–794
 - APIs, 30–31
 - application configuration tags, 306–307
 - applications that can be built with, 300–301
 - benefits of Flex and, 15
 - building applications with, 302–303
 - capabilities and Flash runtime, 786–787
 - configuration file, 306
 - deploying applications with. See application deployment, with AIR

AIR (continued)

desktop environment provided by, 217

drag and drop. See drag and drop, in AIR

file management. See file management, with AIR

Flash Player APIs for, 78–79

Flex 3 ecosystem and, 24–27

as Flex Builder project type, 106

Flex deployment and, 3

Garbage Collection and, 591

globalization (g11n), 699–700

HTML mashups in. See HTML mashups, in AIR

initial window configuration tags, 307–308

Install Badge for one-click install from web pages, 1223–1225

Install Badge Launch Now feature, 1225–1226

inter-SWF communication, see
LocalConnection class

IntroToAir-app.xml descriptor file, 304–306, 314–316

IntroToAir.mxml file, 303–304

MainWindow.mxml file, 313–314

modifying IntroToAir.mxml window, 311–313

native desktop support. See native desktop support, with AIR

offline data access. See SQLite

other configuration tags, 309–311

overview of, 299–300

reasons for using, 300

requirements for, 301–302

summary, 316

system capabilities, 780–781

Alchemy, 39

Alert control, **289–291**

alerts

arguments of Alert class, 289–290

overview of, 289

PopUpManager class for displaying, 291–293

algorithms, optimizing with performance profiler, 1296–1300

alignment, positioning by, 274–275

Allocation Trace View, memory profiling, 1281–1283

allow-access-from, in **Flash Player security model, 1164–1165**

allowBrowserInvocation, **AIR configuration tags, 310**

allowFullScreen **option, Stage class, 757–758**

allow-http-request-headers-from, in **Flash Player security model, 1165–1166**

AMF (Action Message Format)

Charles for monitoring and debugging messages, 1257

Flash data formats, 33

LCDS and, 962–963

LocalConnection class and LSO support for custom classes, 793

RPC (remote procedure call), 850–851

streaming to NIO end points, 1036

support for Zend PHP framework, 881

AMFPhp, third party solutions for RPC services, 851

AnimateProperty **effect, built-in effects, 440–441**

animation

animated cursors, 297

effects and, 437

examples. See Punkhead and Ticklehead examples

Flash and, 217

managing progression of animation with easing functions, 449–450

annotations, customizing, 212

Ant

as AIR tool, 301

continuous integration with, 1322–1323

removing logging and trace with, 1341–1342

anti-patterns, Cairngorm

Chains of Dependency anti-pattern, 1129–1130

Fat Controller anti-pattern, 1127–1128

Model Locator Landfill anti-pattern, 1128–1129

overview of, 1127

Apache servers

configuring for RESTful Web Services, 883

Tomcat, 964

Zend framework support, 884–885

APIs

AIR extensions to Flex framework, 299

asynchronous. See asynchronous APIs, AIR

BrowserManager API. See BrowserManager API

collections. See Collections API

display. See Display APIs, ActionScript

drawing. See Drawing API, Flash Player

Flash Player, 73
 Flex 3 ecosystem, 30–32
 HistoryManager API, 749
 ListBase class, 614–616
 new in Flash Player 10. See Flash Player APIs
 synchronous. See synchronous API, AIR
 YouTube Chromeless Player API, 722

appearance

vs. behavior in extending Flex components, 361
 display appearance methods, 76–77
 skins. See skins
 themes. See themes

application configuration tags, AIR configuration file, 306–307

Application control, as layout container, 260

application deployment, with AIR. See also dual deployment (Flex and AIR)

certificates for code signing, 1216–1218
 code signing from command line, 1222
 code signing in Flex Builder, 1218–1220
 code signing with developer certificates, 1220–1221
 distribution to end users, 1222–1223
 launching installed applications, 1225–1226
 one-click install from web pages, 1223–1225
 overview of, 1215
 silent installations, 1226–1227
 summary, 1228
 updates, 1227–1228

application development, 1141

ActionScript and, 104
 clarity in, 1142
 class names, 1143–1144
 clean code in, 1142
 code behind, 1158–1160
 code syntax, 1145
 consistency of coding, 1141
 constant and static members, 1147
 dictionaries and associative arrays, 1148–1149
 encapsulation, 1149–1150
 getter and setter methods, 1153–1157
 inheritance, 1151–1152
 modular. See modular application development
 namespaces, 1143
 naming conventions, 1142
 OOP and, 1149

overriding functions, 1152–1153
 overview of, 1141
 polymorphism and interfaces, 1157–1158
 proper indentation of code, 1145–1146
 summary, 1160
 ternary operators, 1146
 typed vs. dynamic objects, 1148
 variable and function names, 1144–1145
 variable scope, 1150–1151

application performance

associative arrays and dictionaries for
 improving, 1195–1197
 bitmap caching, 1198–1199
 collection events and, 1192–1193
 event listeners and, 1194–1195
 graphics frame rates, 1198
 impact of multiple service invocations on,
 1191–1192
 lazy loading of data and, 1190–1191
 managing external SWF content, 1199
 object caching and recycling and, 1199–1200
 object creation policies and, 1189–1190
 overview of, 1189
 structuring data for, 1190
 summary, 1201

application programming interfaces. See APIs
application server, configuring for LCDS, 966–967
application state

examining in debugging process, 1252
 Expressions View, 1254–1255
 Variables View, 1253–1254

ApplicationControlBar container, 264

application-level menus, 780

ApplicationPersistence.as file, 1239

ApplicationPersistenceFactory, 1235–1238

applications

creating with Flex Builder, 108
 debugging. See debugging
 Flex settings, 150–151

applications, AIR

building, 302–303
 types of applications that can be built with AIR,
 300–301

applications, FMS

setting up, 875
 setting up instances of, 867–868

applications, LCDS Message Service

- chat room application, 990–993
- overview of, 990
- PDF generator application, 997–1004
- stock price ticker application, 993–997

applications list, Flex Builder, 151

application-specific debuggers, 1259

`AppSettings` class, **ActionScript 3.0, 284–285**

Aptana, 886

architecture

- ActionScript display architecture, 68–69
- Flex architecture for deep linking, 738–739
- LCDS clients, 956–958
- LCDS server architecture, 958–959
- MVC approach to component architecture, 1082

archives. See libraries

area charts, 672–673

`ArrayCollection` class

- charts and, 671
- Collections API and, 586
- `dataProvider` for `DataGrid` control, 628

arrays

- ActionScript, 50–52
- associative. See associative arrays

as **operator, casting and, 54**

as3corelib, 31

ASCII characters, encoding/decoding URL components and, 737

ASDoc

- classes, 1206–1207
- comments and formatting, 1205
- custom templates, 1213
- excluding classes from generated output, 1213
- functions, 1207–1208
- getters/setters, 1209–1210
- HTML formatting, 1211–1212
- logging, 1327
- metadata, 1210–1211
- overview of, 1203
- properties, 1208–1209
- referencing outside elements, 1212
- summary, 1214
- tags, 1206
- using, 1204–1205

ASP.NET, enabling Flex to receive ASP.NET SOAP faults, 920–921

assemblers, DMS

- custom, 1022
- overview of, 1015
- parcel assembler, 1023–1024
- vehicle assembler, 1022–1023

assemblers, LCDS, 959

assertion functions, FlexUnit, 1311

assets

- embedded, 180, 423–424
- loading external. See loading external assets
- using library assets, 407

associative arrays

- application performance and, 1195–1197
- code syntax, 1148–1149
- dynamic classes, 589
- equality operators, 590
- iteration, 591
- objects as keys, 590
- overview of, 589
- strings as keys, 589
- weak keys, 591–592

asynchronous APIs, AIR

- file operations, 816–818
- opening databases, 931–932
- SQLite database operations, 930–931

asynchronous commands, Cairngorm, 1109–1110, 1132–1133

asynchronous communication, RIAs and, 4

asynchronous error handling

- Debug Player and, 862
- important events, 861–862
- overview of, 860

`AsyncToken`, **890**

attributes

- class attributes in ActionScript, 59
- property attributes in MXML, 136–137
- XML attributes vs. child nodes, 572

audio. See sound

automation framework

- automating custom components, 1353–1354
- compiling for automation, 1348–1349
- component identification, 1346
- creating testable application, 1347–1348
- installing automation tool, 1347
- libraries, 1345
- overview of, 1343

- recording and playback, 1344–1345
- recording test scripts, 1349–1352
- replaying test scripts, 1352–1353
- runtime vs. compile time automation, 1345–1346
- simple application example, 1346–1347
- stopping recording, 1352
- summary, 1354

AVMs (ActionScript Virtual Machines)

- Flash Player and, 20–21
- memory management in OOP languages, 1356

axes, chart, 680–681

- customizing axis label, 681
- overview of, 680
- types of, 681

B

backing variable, getter/setter properties and, 324

bandwidth check system, FMS (Flash Media Server), 869–872

bar charts, 674–675

base classes

- ActionScript 3.0, 68
- `CairngormEvent` base class, 1085
- events, 369
- formatters, 596
- inheritance and, 64
- lists, *see* `ListBase` class
- selecting for ActionScript component, 330–331
- `TileBase` class, 621–622
- `UIComponent`, *see* `UIComponent` base class

base states, 383

baselines, profiling and, 1269

BDD (Behavior-Driven Development), 1324

behavior vs. appearance, extending Flex components and, 361

Behavior-Driven Development (BDD), 1324

bindable properties, `DataService` class, 1013

[Bindable] metadata tag, 235–236, 377–379

binding controls, to validators, 604

binding data. *See* data binding

binding expressions, for overcoming item-pending errors, 1032–1033

binding keyboard shortcuts, 188

`BindingManager` class, 1260–1261

`BindingUtils` class (ActionScript), 380

bitmaps

- caching, application performance, 1198–1199
- fills in Drawing API, 469–470
- saving as JPEG file, 827–828

BlazeDS (Blaze Data Services), 35

- channels of communication, 1055
- coding remote Java objects, 1064
- configuring Proxy Service destinations, 1068–1069
- configuring Remoting Service, 1063–1064
- connecting Flex clients via Proxy Service, 1069–1071
- creating combined Flex/Java project, 1057–1058
- creating `RemoteObject` instance, 1060
- deploying project to Tomcat, 1059
- enabling logging, 1059–1060
- features, 1054
- `HTTPService`, 1067–1068
- integration points (adapters), 1055
- LCDS service solutions, 956
- object mapping in Remoting Service, 1065–1067
- overview of, 1053
- projects, 1056
- remote methods calls, 1060–1063
- requesting data from Proxy Service, 1071
- services and client components, 1054
- summary, 1071
- turnkey distribution, 1056–1057

BlazeDS Developer Guide, 1065

blending modes, in Drawing API, 478

block-level scoping, comparing OOP languages, 1356–1357

Blur effect, built-in effects, 441

bottlenecks, performance optimization and, 1268

breakpoints, 1246–1249

- conditional and counting, 1248
- managing at runtime, 1248–1249
- in MXML, 1248
- placing, 1246–1248

Browser Servers

- configuring/starting for stress testing, 1046–1047
- overview of, 1040
- parameterizing test applications, 1049

BrowserManager API, 734–737

- ActionScript 3.0 and, 733

BrowserManager (*continued*)

- examining URL fragment ID, 734–735
- example applying, 735–737
- initializing `IBrowserManager` interface, 734

browsers. See web browsers

bubble charts, 675–676

bubbling phase

- event flow and, 72
- overview of, 238

buffer control, FMS and, 873–874

Bug and Issue Management System. See Adobe Bug and Issue Management System

bugs. See also debugging

- preventing by testing, 1304
- Subclipse for tracking, 214

build path

- Flex libraries, 406–407
- Flex settings, 151
- LCDS project, 965–966

build tools, Flex Builder 3, 86

building components

- Flash skin for custom Flex component, 494–498
- Flash/Flex integration, 492–494
- modular application development and, 1173

built-in effects

- `AnimateProperty` effect, 440–441
- Blur effect, 441
- Dissolve effect, 441–442
- Fade effect, 442
- Glow effect, 442–443
- Iris effect, 443
- Move effect, 443
- overview of, 440
- Pause effect, 444
- Resize effect, 444
- Rotate effect, 444–445
- `SoundEffect` effect, 444–445
- `WipeLeft/WipeRight/WipeUp/WipeDown` effects, 445–446
- Zoom effect, 446

business delegates, Cairngorm, 1110–1113

- creating, 1112
- overview of, 1087–1088, 1110–1111
- stubbing the service layer, 1112–1113
- using in a command, 1111

business delegates, stub implementations of, 1135

busy cursor, 295–296

Button control

- extending, 358–359
- as input control, 249

ButtonBar control, as navigation component, 269

buttons, applying visual effects, 438, 446

Buzzword, Adobe, 1259

C

C++, comparing OOP languages

- access modifiers, 1362
- block-level scoping, 1356–1357
- classes, filenames, and packages, 1360–1361
- console output and, 1357
- data types and, 1357–1359
- declaring variables, 1359–1360
- dynamic classes, 1362
- global variable notation, 1357
- inheritance, 1362
- memory management, 1356
- overview of, 1355
- polymorphism, 1361–1362
- regular expressions, 1360
- runtime environments, 1355–1356
- statement termination, 1360
- summary, 1363
- type casting, 1359

CA (certificate authority), 1216

caching

- bitmap caching, 1198–1199
- objects, 1199–1200

Cairngorm

- Adobe frameworks for development of Flex or AIR applications, 31
- binding view to model, 1099–1100
- business delegates, 1087–1088, 1110–1113
- command execution, 1086–1087, 1108–1110
- event classes, 1105
- event types, 1105–1106
- events, 1085–1086
- events as parameter objects, 1106–1107
- front controllers, 1102–1104
- getting started with, 1095–1096

- minimizing dependencies, 1100–1102
- model locator, 1090–1092, 1096
- modeling the application, 1098–1099
- overview of, 1083, 1095
- RESTful services and, 1115
- self-dispatching events, 1107
- self-documenting code, 1098
- service locator, 1089–1090, 1113–1115
- single point of access to model locator, 1097–1098
- stages of Cairngorm interactions, 1083–1084
- summary, 1092–1094, 1115–1118
- user gestures, 1084–1085
- value objects, 1115–1118
- view rejecting changes, 1092–1094
- Cairngorm, advanced**
 - anti-patterns, 1127
 - Chains of Dependency anti-pattern, 1129–1130
 - command callbacks for locating presentation model, 1123–1124
 - customization, 1131
 - decentralizing controllers, 1136–1137
 - disposable controllers, 1137–1138
 - domain models, 1124–1125
 - Fat Controller anti-pattern, 1127–1128
 - injecting dependencies into commands, 1133–1135
 - logging Cairngorm events and commands, 1131–1132
 - Model Locator Landfill anti-pattern, 1128–1129
 - organizing large projects, 1126–1127
 - organizing small projects, 1125–1126
 - overview of, 1119
 - presentation model dispatching Cairngorm events, 1123
 - Presentation Model pattern, 1119–1120
 - simplicity of code in presentation model views, 1122–1123
 - stubbing the service layer, 1135–1136
 - summary, 1138
 - timing asynchronous commands, 1132–1133
 - using Cairngorm in conjunction with presentation models, 1120–1121
- CairngormEvent **base class, 1085**
- calculating values, with getter/setter methods, 1157**
- camel naming conventions**
 - classes using, 1143–1144
 - variables and functions using, 1144–1145
- Camera **class, Flash, 538–539**
- candlestick charts, 676**
- Canvas **container**
 - absolute positioning and, 415
 - changing dimensions of, 385
 - as layout container, 256–257
 - optimization by wrapping common state change elements in, 394–395
- capture phase**
 - event flow and, 72
 - overview of, 236–237
- Cascading Style Sheets. See CSS (Cascading Style Sheets)**
- case sensitivity**
 - ActionScript, 42
 - ColdFusion, 977
- casting**
 - ActionScript, 54
 - comparing OOP languages, 1359
- catch **block, for error handling, 855–856**
- CDATA tags, 143**
- cells, customizing**
 - custom component item renderers, 634–635
 - drop-in item renderers, 632–633
 - inline item renderers, 633
 - overview of, 632
- certificate authority (CA), 1216**
- certificates**
 - acquiring, 1216
 - creating/signing AIR package in Flex Builder, 1218–1220
 - developer certificates, 1220–1221
 - exporting Thawte certificate from Firefox, 1217–1218
 - overview of, 1216–1218
 - purchasing from Thawte, 1216–1217
- CFC (ColdFusion components), 976**
- Chains of Dependency anti-pattern, 1129–1130**
- changeCSSStyle() **method, 432–433**

ChangeWatcher, data binding and, 234–235, 380–381

channels

BlazeDS communication, 1055
LCDS communication, 958, 980

character encoding, 691–692

Eclipse and, 691–692
overview of, 691

Charles, web debugging proxy, 1257–1258

chart series

overview of, 681
series types, 682

charting

area charts, 672–673
assigning data to charts, 671–672
axes, 680–681
bar charts, 674–675
bubble charts, 675–676
candlestick charts, 676
chart series, 681–682
column charts, 677
custom renderers, 685–686
data tips, 682–683
effects, 686–687
HLOC (HighLowOpenClose) charts, 677
legends, 683
line charts, 677–678
overview of, 671
pie charts, 678–679
plot charts, 680
programmatic interaction, 684–685
styling, 686
summary, 687
types of charts, 672
user interaction, 684

chat room application, LCDS Message Service, 990–993

chat destinations, 991–992
ChatRoom class, 992
overview of, 990–991
producers and consumers, 992–993
summary, 993

ChatRoom **class, 992**

CheckBox **control, 251**

checkpoints, RIATest, 1350–1352

child components, ActionScript

adding, 325–326
sizing, 332–333
steps in writing components, 319

child components, Flex, 365

child components, MXML, 350–351

chrome, overriding layoutChrome(), 346–348

Chromeless Player API, YouTube, 722

clarity, as coding standard, 1142

class attributes, ActionScript, 59

class inheritance, CSS styles and, 418–419

class instances, ActionScript, 58

class members, ActionScript, 59

class outlines, Outline View in Flex Builder, 174–175

class packages

container classes, 224–225
formatters, validators, and collection classes, 226–228
list classes, 225–226
overview of, 222
UIComponent base class, 222–224

Class View, Outline View options, 116

classes

ActionScript, 58–59
bringing Flex classes into Flash, 545–546
camel naming conventions, 1143–1144
comparing OOP languages, 1360–1361
documenting with ASDoc, 1206–1207
dynamic, 589
excluding from ASDoc generated output, 1214
Flash, 222
importing custom Flex classes into Flash, 546
importing native Flex classes into Flash, 547
instantiation and, 58

-clean argument, workbench optimization in Eclipse, 183

clean code, coding standards, 1142

clear() method, for clearing contents of graphics objects, 462

clients

BlazeDS components, 1054
detecting system capabilities, 751–755
pushing LCDS messages to, 989–990
remote logging, 1334–1335
RIAs residing on, 4

software requirements for RESTful Web Services, 886
 software requirements for Routes Web Service, with Java, 902
 software requirements for Routes Web Service, with .NET, 918

clients, Flex

architecture of LCDS clients, 956–958
 connecting via Proxy Service, 1069–1071
 creating messages on, 983–984
 PDF generation client, 1002–1003

client-side

BlazeDS logging, 1059–1060
 code, 806–807
 LCDS logging, 968

client-side DMS

client/server collaboration, 1015
 coding for DMS Fleet tracker, 1024
 components, 1010–1011
 programming, 1012–1014

clipboard

adding data to, 658–659
 AIR for cross platform support, 300
 data behind image on, 662–663
 reading data from, 659
 reading/writing custom data from clipboard, 660

clustering, LCDS and, 1037**CMS, Flex/Flash integration with, 9****code**

Flex Builder preferences, 173
 independence, 405
 libraries. *See* libraries
 stepping through, 1250–1252
 strengthening in test suites, 1314–1315

code behind, 1158–1160**code editing, with Flex Builder**

ActionScript code assist, 171
 code introspection, 173
 code preferences, 173
 Mark Occurences button, 172–173
 MXML code assist, 172
 overview of, 170
 syntax highlighting, 171

code introspection, Flex Builder, 173. *See also* Outline View, Flex Builder**code repositories**

creating project from Subclipse repository, 203–210
 customizing Flex Builder 3, 190
 establishing repository for Subclipse, 195–196
 importing projects from, 157
 Local History, 191–192
 open source Flex SDX, 196
 overview of, 190
 Professional Flex 3 book code, 196
 project management for multiple Subclipse repositories, 210
 sharing projects in Subclipse repository, 197–202

code reuse, 545**code signing**

certificates for, 1216–1218
 from command line, 1222
 with developer certificates, 1220–1221
 in Flex Builder, 1218–1220

code syntax

ActionScript, 42
 constant and static members, 1147
 dictionaries and associative arrays, 1148–1149
 MXML, 128–129
 overview of, 1145
 proper indentation of code, 1145–1146
 syntax highlighting in Flex Builder, 171
 ternary operators, 1146
 typed vs. dynamic objects, 1148

CODECs, Flash video formats, 33**code-first development, in WSDL, 900, 916****coding by interface**

as dual deployment option, 1229–1231
 pros/cons of coding by interface, 1231
 referencing main application
 (MainAppEntryPoint.xml), 1233–1234
 setting up Flex Builder projects, 1232
 working with
 ApplicationPersistenceFactory,
 1235–1238
 working with IApplicationPersistence
 interface, 1235

coding standards, 1141–1149

clarity, 1142
 class names, 1143–1144

coding standards (continued)

- clean code, 1142
- code syntax, 1145
- consistency, 1141
- constant and static members, 1147
- dictionaries and associative arrays, 1148–1149
- namespaces, 1143
- naming conventions, 1142
- overview of, 1141
- proper indentation of code, 1145–1146
- ternary operators, 1146
- typed vs. dynamic objects, 1148
- variable and function names, 1144–1145

ColdFusion

- administrator screens, 973–975
- Adobe servers used by Flex applications, 35
- event gateways, 974–976
- installation screens, 971–973
- LCDS integration with, 971
- managing LCDS data in, 976–977
- server-side functionality for file uploads, 803

ColdFusion components (CFC), 976

- `collectionChange` events, 1193

- `CollectionEvent` class, 587–588

collections

- classes/subclasses of `Collection` class, 227–228
- data binding for updating contents, 379
- filling collections with parcels and vehicles, 1024–1025
- filling in DMS demo, 1009
- list controls and, 270
- as preferred way for handling Flex applications, 1192–1193

Collections API

- `ArrayCollection` class, 586
- `CollectionEvent` class, 587–588
- `dataProvider` property for data binding, 587
- filtering, 583–584
- `ICollectionView`, 582
- `IList`, 582
- `IViewCursor`, 584–585
- `ListCollectionView`, 586
- overview of, 582
- sorting, 582–583
- `XMLListCollection` class, 586–587

color palette, input controls, 256

- `ColorPicker` control, input controls, 256

column charts, 677

- columns, `DataGrid` control working with, 629–631

ComboBox control

- Flex filesystem classes, 829
- input controls, 252
- pop-ups and, 288

command-line

- AIR tools, 301
- ASDoc as command-line tool, 1204
- code signing from, 1222
- COMPC options, 403
- debugger, 1255
- startup options in Flex Builder, 182–186

commands, Cairngorm, 1108–1110

- command callbacks for locating presentation model, 1123–1124
- executing asynchronous commands, 1109–1110
- executing synchronous commands, 1108–1109
- injecting dependencies into, 1133–1135
- logging, 1131–1132
- overview of command execution, 1086–1087
- stub implementations of, 1135

comments

- ActionScript, 42
- documenting with ASDoc, 1205
- `commit()`, `DataService`, 1010
- `commitProperties()`, `UIComponent` class, 323
- communication, see `LocalConnection` class
 - BlazeDS channels for, 1055
 - cross-module, 1177–1182
 - DSS Test administrator, 1044–1045
 - Flex RIA process flow, 37–38
 - LCDS communication channels, 961–962
 - protocols in Flex 3 ecosystem, 32–33
 - with remote servers, 836

company logo

- ActionScript custom component example, 320
- `CompanyLogo` class, 321–323

COMPC, 402–404

- adding assets to library, 403
- command-line options, 403
- overview of, 402–403
- specifying manifest file, 403–404

compilation

- automation framework and, 1348–1349
- comparing OOP languages, 1355
- compiling application in Flex Builder Design mode, 119
- configuring Flex compiler, 1240
- development enhancements in Flex Builder 3, 99
- Flex Compiler settings, 151–153
- process in Flex Builder, 105
- slow downs, 182

compile time, vs. runtime automation, 1345–1346**compiletime shared libraries (CSLs), 485****complex fills, Drawing API, 470–471****complex objects, passing between JavaScript or
ActionScript and Flex, 713–716****complex types**

- ActionScript, 48
- comparing OOP languages, 1358

component styles

- ActionScript, 413–415
- MXML, 139–140

component tags, MXML, 133**components**

- ActionScript custom components. See ActionScript, custom components
- advanced, 99
- building. See child components

components

- composite. See composite components
- creating Flex component in Flash, 548–549
- extending standard, 360–361
- layout. See layout
- life cycle. See life cycle, Flex components
- lists. See list components
- MVC approach to component architecture, 1082
- MXML custom components. See MXML, custom components
- MXML-only components, 134
- navigation. See navigation components
- nesting in Flex framework, 220–221
- removing in state management, 389–390
- reuse, benefits of MVC frameworks, 1080
- skinning custom Flex component, 498–500
- third party frameworks and components for Flex, 32

Components views, Flex Builder, 112–113**composite components**

- appearance vs. behavior, 361
- creating, 354–355
- customizing, 365–366
- events, 356–357
- example, 353–354
- extending Button control, 358–359
- extending TextInput control, 359–360
- extending UIComponent vs. standard components, 360–361
- layout, 357
- overview of, 353
- properties for passing data to, 355–356

composite effects

- nested effects, 453–454
- overview of, 450
- Parallel effects, 451–452
- Sequence effects, 452–453

**concurrency property, MXML RPC classes
and, 838****conditional breakpoints, 1248****conditional compilation**

- ApplicationPersistence.as file, 1239
- configuring Flex compiler, 1240
- as dual deployment option, 1238–1239
- pros/cons of, 1238

conditional operator (?), ActionScript, 46**conditional statements, 44–46**

- code syntax, 1145–1146
- conditional operator (?), 46
- if and else if, 44–45
- switch, 45–46

configuration file, AIR

- application configuration tags, 306–307
- initial window configuration tags, 307–308
- other configuration tags, 309–311
- overview of, 306

conflict resolution, DMS and, 1037**connection name, LocalConnection class, 792****connectivity**

- detecting for LCDS Message Service, 988
- monitoring network connections with AIR, 785–786
- occasional connectivity as reason for using DMS, 1008
- RIAs and, 6
- Subclipse settings, 212

ConnectNow server, Adobe servers used by Flex applications, 34–35

consistency, coding standards and, 1141

console output, comparing OOP languages, 1357

constants

ActionScript, 63

code syntax, 1147

comparing OOP languages, 1358

constraint columns, layout strategies, 276–277

constraint rows, layout strategies, 277–278

constraints

composite components and, 357

constraint columns, 276–277

constraint rows, 277–278

constraint sizes, 278

constraints in ActionScript, 278

CSS styles and, 415–418

layout constraints in Flex Builder, 113–114

overview of, 272–273

constraints, advanced

in ActionScript, 278

constraint columns, 276–277

constraint rows, 277–278

constraint sizes, 278

development enhancements in Flex Builder 3, 99

overview of, 275–276

constructor methods, ActionScript, 60–61

consumers, LCDS

chat room application, 992–993

declaring in MXML, 986–987

detecting connectivity, 988

Message Service, LCDS, 980

receiving messages and, 985

contact-first development, WSDL, 900, 916

container classes, 224–225

classes/subclasses of, 224

extending `Container` class, 362

Flash and Flex class packages, 224–225

overview of, 224

containers

adding effects to data containers, 448

creating custom containers using MXML components, 346

layout. See layout containers

MXML components for extending container classes, 343

navigation components. See navigation components

optimization by wrapping common state change elements in container, 394–395

overview of, 256

containers, custom

applying Flex component life cycle to, 362–363

overview of, 361–362

`TagCloud` component, 363–365

content tag, AIR initial window configuration, 307

context editing, editing component text, 112–113

context menus

adding/removing options, 760–763

creating native menus, 774–776

invoking actions from, 763–765

limitations of custom, 766

contextual assistance, Flex Builder, 90

continuous integration, with Ant and Maven

scripts, 1322–1323

control classes, Cairngorm, 1084

control structures, ActionScript, 44

`ControlBar` container, 263

controllers, Cairngorm

decentralizing, 1136–1137

disposable, 1137–1138

front controllers. See front controllers, Cairngorm

controllers, in MVC framework, 1078–1079

controlling sound, 522–525

controlling video, 535–536

`VideoDisplay` component, 537–538

controls

advanced data controls. See data controls, advanced

binding to validators, 604

`dataProvider` property for data binding, 587

input. See input controls

layout containers. See layout containers

list, 268–269

navigation components. See navigation components

overview of, 245

state management, 389–390

static. See static controls

coordinate positioning

CSS styles and, 415

Drawing API and, 463

- flexibility and, 357
- layout strategies, 271
- copylocale.exe , 694
- copyright **tag, AIR application configuration, 307**
- counting breakpoints, 1248**
- create, read, update, delete (CRUD). See CRUD (create, read, update, and delete)**
- Create Application from Database Wizard, new in Flex Builder 3, 98**
- create operations. See CRUD (create, read, update, and delete)**
- CREATE TABLE, **SQL statement, 945–947**
- createChildren(), **adding child components, 325–326**
- createPopUp(), 292
- creation policies, objects, 1189–1190**
- creationPolicy **property, navigation components and, 264**
- Creative Suite. See CS3 suite**
- credit card validators, 602**
- cross-domain security policies**
 - allow-access-from, 1164–1165
 - allow-http-request-headers-from, 1165–1166
 - crossdomain.xml, 1162–1164
 - site-control entry and meta-policies, 1166–1167
- crossdomain.xml, **1162–1164**
- cross-module communication, 1177–1182**
- cross-module data access, 1182–1187**
- cross-platform runtimes. See AIR (Adobe Integrated Runtime)**
- CRUD (create, read, update, and delete)**
 - creating SQLite data, 936–938
 - deleting SQLite data, 940–941
 - DMS API for, 1006
 - RESTful Web Services and, 881–882, 895–896
 - retrieving SQLite data, 938–939
 - updating SQLite data, 939–940
- CS3 suite**
 - Flex Builder integration with, 99–100
 - Flex integration with, 14
- CSLs (compiletime shared libraries), 485**
- CSS (Cascading Style Sheets)**
 - applying skins with, 472
 - changing skin styles at runtime, 430–433
 - class inheritance, 418–419
 - constraint-based layout, 415–418
 - creating themes and, 435
 - defining skins, 426–428
 - development of Flex applications and, 104
 - inline styles vs., 411–413
 - languages in Flex 3 ecosystem, 28
 - overview of, 411
 - subcomponents, 419–421
- CSS (Cascading Style Sheets), in MXML**
 - CSSStyle tag, 139–140
 - inline CSS, 139–140
 - linking external CSS, 140–141
 - overview of, 139
- CSS Design View, Flex Builder 3, 98**
- CSS Outline View, Flex Builder 3, 99**
- CSS View, Outline View options, 116**
- CSSStyle **tag, MXML and CSS, 139–140**
- cue points, video, 538**
- curl **utility, for HTTP request, 886**
- curly braces ({}), data binding syntax, 118, 231–235**
- currency**
 - formatters, 600
 - internationalization and localization and, 690
 - validators, 602
- CurrencyFormatter, **600**
- CursorManager **class, 295**
- cursors**
 - busy cursor, 295–296
 - CursorManager class, 295
 - custom, 296–297
 - hand cursor, 296
 - IViewCursor, 584–585
 - overview of, 295
- curves, drawing, 464–465**
- custom cells**
 - custom component item renderers, 634–635
 - drop-in item renderers, 632–633
 - inline item renderers, 633
 - overview of, 632
- custom composite components, 365–366**
- custom containers. See containers, custom**
- custom context menus, 766**
- custom cursors, 296–297**
- custom data binding. See data binding, custom**

custom data types, 799–801

custom error types, 858–859

custom events

[Event] metadata and, 242–243

creating, 241–242

extending standard events, 371

one vs. many rule when creating, 371–372

vs. standard events, 369–371

strategies for using, 243–244

custom renderers, 632–637

charts and, 685–686

custom component item renderers, 634–635

customizing cells, 632

drop-in item renderers, 632–633

headers, 636–637

inline item renderers, 633

item editors, 635–636

custom server-side errors

Routes Web Services, with Java, 909

Routes Web Services, with .NET, 925–926

custom shortcuts, Flex Builder 3, 188–189

custom skins. See skins

custom templates, ASDoc applying to documentation, 1213

custom themes. See themes

custom tooltips, 286–288

`customUpdateUI` tag, AIR configuration, 310

CVS, version control systems, 192

D

data

assigning data to charts, 671–672

assigning to `Tree` control, 624–625

associative arrays and. See associative arrays

Collections API. See Collections API

creating/updating in RESTful Web Services, 894–896

cross-module data access, 1182–1187

deleting in RESTful Web Services, 896

effects, 448–449

formats in Flex 3 ecosystem, 33–34

formatters. See formatters

grouping in `AdvancedDataGrid` control, 638–639

handling data input errors, 600–601

hierarchical order in `AdvancedDataGrid` control, 639

incorporating into media streams, 877–879

lazy loading, 1190–1191

`LocalConnection` class and, 793

managing in ColdFusion, 976–977

reading/writing. See reading/writing data

regular expressions. See regular expressions

`Repeaters` class. See `Repeaters` class

retrieving from RESTful Web Services, 889

saving/loading profiling data, 1289

storing in/retrieving from LSOs, 795

structuring to improvement application performance, 1190

summary, 594

validators. See validators

working with SQLite data, 935

XML. See XML (Extensible Markup Language)

data binding

[Bindable] metadata, 235–236

`dataProvider` property for, 587

debugging and, 1260–1262

defined, 118

Flex Builder and, 117–118

impact on application performance, 1192

inline ActionScript and, 141

MXML and, 344

overview of, 231

state management and, 391

summary, 244

syntax, 231–235

data binding, custom

in ActionScript, 380–381

[Bindable] metadata tag, 377–379

`BindingUtils` class (ActionScript), 380

`ChangeWatcher`, 380–381

overview of, 377

data controls, advanced

`AdvancedDataGrid` control, 637–639

custom renderers, 632–637

`DataGrid` control, 628–632

`OLAPDataGrid` control, 640–644

overview of, 623

summary, 644

`Tree` control, 624–628

Data Definition Language (DDL), 933–935

Data Management Service. See **DMS (Data Management Service)**

Data Management Service, LCDS, 960–961

data model, MVC frameworks, 1077–1078

data paging, DMS and, 1037

data persistence, 4. See also LSOs (local shared objects)

data services

BlazeDS. See **BlazeDS (Blaze Data Services)**

LCDS. See **LCDS (LiveCycle Data Service)**

Data Services Stress Testing Framework

adding communication channel for test results, 1044–1045

adding destination for result data, 1045

adding remote object for logging, 1046

compiling Test Administrator, 1047

configuring default Proxy Service, 1046

configuring/starting Browser Servers, 1046–1047

counting successes and failures, 1043

creating customized test reports, 1050–1051

overview of, 1039–1040

parameterizing test applications, 1049

preconfiguring Test Administrator, 1049–1050

prerequisites for stress testing, 1041

running stress test, 1047–1049

services required in LCDS application, 1044

starting test scenario, 1042–1043

steps in stress testing LCDS applications, 1041

stopping test, 1043–1044

summary, 1051

writing test application, 1041–1042

data sets, paging large, 1007

data sources, binding to destinations, 118

data tips

charts and, 682–683

ListBase class providing, 616

Data Transfer Objects (DTO), 919

data types

comparing OOP languages, 1357–1359

as Flash classes, 222

LocalConnection class and LSO support for custom classes, 799–801

maintaining data type information while dragging and dropping, 656

MXML, 135

data types, ActionScript

arrays, 50–52

complex, 48

mapping to Java types, 1065

numbers, 48

overview of, 44

primitive, 47

special, 48

strings, 49

vectors, 52–53

databases, SQLite. See SQLite

data-checking, with getter/setter methods, 1156

DataGrid control, 628–632

AdvancedDataGrid control. See

AdvancedDataGrid control

creating composite components, 355–356

Flex filesystem classes, 829

handling events, 631–632

OLAPDataGrid control. See OLAPDataGrid control

overview of, 628

populating, 628–629

providing managed data to in DMS demo, 1009

working with columns, 629–631

DataManagement gateway, ColdFusion event gateways, 976

dataProvider property, for data binding, 587

DataService class

DMS demo and, 1009–1010

exceptions, 1014

managed data, 1011

methods and properties, 1012–1013

DataServicesMessaging gateway, ColdFusion event gateways, 976

DateChooser control, 250–251

DateField control

input controls, 250–251

pop-ups and, 288

DateFormatter, 598–599

dates

formatters, 598–599

input controls for, 250–251

internationalization and localization and, 690

validators, 602

DDL (Data Definition Language), 933–935

debug()

`debug()`, **log levels, 1329–1330**

debug builds, creating with Flex Builder, 164–165

Debug mode, Flex Builder, 1249–1250

Debug Player. See Flash Debug Player

debug tracing, RPC, 846

debugging. See also Flex Builder Profiler

Adobe Bug and Issue Management System,
1262–1265

application-specific, 1259

bindings and, 1260–1262

breakpoints for, 1246–1249

from command-line, 1255

context menu option, 760

debugging perspective in Flex Builder, 1245–1246

examining application state, 1252

Expressions View, 1254–1255

Flash Player and, 1245–1246

Flex Builder 3 and, 86

Flex Builder 3 views for, 96

Java server-side code and, 1255–1257

launching application in Debug mode, 1249–1250

logging, 1259–1260

monitoring Internet traffic, 1257–1258

overview of, 1245

server-side debugging in LCDS, 969

stepping through code, 1250–1252

summary, 1265

Variables View, 1253–1254

debugging perspective, in Eclipse, 1256

debugging perspective, in Flex Builder

breakpoints in MXML, 1248

conditional and counting breakpoints, 1248

managing breakpoints at runtime, 1248–1249

overview of, 1246

placing breakpoint, 1246–1248

debugging RPC

debug tracing, 846

logging calls to web services, 848–850

logging HTTPService requests, 846–847

logging WebService initialization, 847–848

network packet sniffing, 846

overview of, 845

Declarative Graphics Framework (Degrafa)

Drawing API and, 478

third party frameworks and components for
Flex, 32

declaring variables, OOP and, 1359–1360

deep linking

BrowserManager API, 734–737

development enhancements in Flex Builder 3, 99

encoding/decoding URL components, 737–738

Flex 3 support for, 733

Flex applications and, 732–733

Flex architecture for, 738–739

fragment identifiers in URLs, 733

HistoryManager API, 749

image browser examples, 739–740, 743–747

navigational states and, 740–743

overview of, 731

search engine indexing and optimization and,
747–748

summary, 749

SWFAddress open source project, 748

UIKit, 748–749

default themes, Flex 3, 433

**defaultButton, for Enter or Return key on
forms, 286**

deferred creation policy, 1189

Degrafa (Declarative Graphics Framework)

Drawing API and, 478

third party frameworks and components for
Flex, 32

delegates, business. See business delegates

DELETE, SQLite data transactions, 941

**delete operations. See also CRUD (create, read,
update, and delete)**

elements in XML objects, 580–581

files and directories, 819

files recovered from Local History, 191

LSOs and, 797–798

memory management and, 1356

dependencies

Cairngorm and, 1119

Chains of Dependency anti-pattern, 1129–1130

injecting into Cairngorm commands, 1133–1135

mock objects for overcoming, 1318–1319

dependency injection

into Cairngorm commands, 1133–1135

defined, 1305

deployment

application deployment. See application
deployment, with AIR

- dual deployment. See dual deployment (Flex and AIR)
- descendants, List class, 619**
- description tag, **AIR application configuration, 307**
- design**
 - Flex RIA process flow and, 36–37
 - testing as means of improving, 1304
- Design Area, options in Flex Builder, 114–115**
- design editors, Flex Builder 3, 85**
- Design mode, Flex Builder**
 - compiling application in, 119
 - Components and Properties views, 112–113
 - data binding and, 117–118
 - enhancements in Flex Builder 3, 98
 - Fast Views, 116–117
 - layout constraints, 113–114
 - navigating design area, 114–115
 - overview of, 111–112
 - triggers and effects, 118–119
 - viewing display hierarchy with Outline View, 116
- design patterns, 1119–1120**
- Design View, Flex Builder**
 - building forms with, 285–286
 - creating states, 384–387
 - Property Inspector, 286
- desktop**
 - deployment. See application deployment, with AIR
 - flash.desktop package, 82
 - native desktop support with AIR. See native desktop support, with AIR
 - RIAs can be desktop-enabled, 5
- destinations**
 - binding data source to, 118
 - configuring Proxy Service destinations, 1068–1069
- destinations, in DMS Fleet tracker example**
 - configuring, 1019
 - parcel destination, 1021–1022
 - vehicle destination, 1019–1021
- destinations, LCDS**
 - chat room application, 991–992
 - configuring, 981
 - overview of, 959
 - PDF generator application, 999
 - routing messages to, 989
 - stock feed application, 995–996
- developer certificates, code signing with, 1220–1221**
- developers**
 - AIR developer requirements, 301
 - benefits of testing to, 1305
 - ease in learning Flex, 14
- development**
 - best practices in Flex Builder, 179–181
 - distributed development with MVC, 1080–1081
 - enhancements in Flex Builder 3, 99
 - Flex development community, 10–11
 - Flex RIA process flow, 36–37
 - for FMS in Flex, 868–869
 - modular. See modular application development
 - process in Flex Builder, 104
 - tools in Flash Platform, 483–484
 - views in Flex Builder 3, 95
 - workflows in Flash Platform, 482–483
- development environment, setting up for RESTful Web Services, 883**
- diagnostics. See logging**
- Dictionary class**
 - application performance and, 1197
 - code syntax, 1148–1149
 - dynamic classes in ActionScript, 590
 - iteration or looping through, 591
 - weak keys, 591–592
- Diff tool, Local History and, 191**
- directories**
 - checking existence of, 819
 - deleting, 819
 - enumerating, 820
 - importing projects from, 156
 - LSO directory locations, 798–799
 - moving, 819–820
 - native browsing for, 821–822
 - referencing, 813–814
 - sharing source directories, 404–405
 - temporary, 820–821
- disk storage settings, LSOs, 797**
- display**
 - architecture in ActionScript 3.0, 68–69
 - hierarchy in Flex Builder, 116
 - hierarchy in MXML, 129
- Display APIs, ActionScript, 74–78**
 - core display classes, 74–75

Display APIs (continued)

methods for display appearance and spatial properties, 76–77

methods for finding display objects, 77

methods for handling display events, 77–78

methods for manipulating display lists, 75

overview of, 74–78

display lists

ActionScript, 68

methods for manipulating, 75

disposable controllers, Cairngorm, 1137–1138

Dissolve effect, built-in effects, 441–442

distributed development, benefits of MVC frameworks, 1080–1081

distributing packages, to end users

one-click install from web pages, 1223–1225

overview of, 1222–1223

silent installations, 1226–1227

DMS (Data Management Service)

adapters, 1014–1015

assemblers, 1015

client and server components of, 1010–1011

client programming, 1012–1014

client/server collaboration, 1015

connectivity and, 1008

CRUD (create, read, update, delete) operations and, 1006

destination for stress test results, 1045

domain modeling and, 1006

managed data, 1011

managed relationships, 1011–1012

overview of, 1005

paging large data sets, 1007

server integration, 1014

sharing data, 1006–1007

six-lines of code demonstrating, 1008–1010

stress testing, 1041

summary, 1028

DMS (Data Management Service), advanced

additional resources for, 1036–1037

creating `MessageDetails` items, 1034

creating/delivering user messages, 1034–1035

instant messaging system, 1033–1034

managing shared and user-specific data, 1033

overcoming item-pending errors, 1031–1032

overview of, 1029

repeated attacks and, 1032–1033

replacing lazy relationships with `getItem()` or `fill()`, 1030–1031

streaming AMF to NIO end points, 1036

summary, 1037–1038

summary-to-details modeling, 1030

DMS Fleet tracker example

client coding, 1024

configuring destinations, 1019

creating new parcel, 1026–1027

custom assemblers, 1022

domain modeling and, 1016–1017

filling collections with parcels and vehicles, 1024–1025

filtering parcel list, 1026

overview of, 1015–1016

parcel assembler, 1023–1024

`Parcel` class, 1018–1019

parcel destination, 1021–1022

vehicle assembler, 1022–1023

`Vehicle` class, 1017–1018

vehicle destination, 1019–1021

visualizing vehicle movements, 1025–1026

do while statements, ActionScript loops, 46–47

Dock (OSX), AIR support for interaction with, 782–783

docking views, 189

Document Object Model. See DOM (Document Object Model)

documentation. See also ASDoc

benefits of tests, 1305

Flex Builder, 87, 90

DOM (Document Object Model)

ActionScript event model based on DOM Level 3, 69

controlling HTML DOM from ActionScript, 729

interaction with XML and, 573–574

domain modeling

Cairngorm, 1124–1125

DMS Fleet tracker example, 1016–1017

`Parcel` class, 1018–1019

reasons for using DMS, 1006

`Vehicle` class, 1017–1018

domains

cross domain communication between SWFs, 792

- LocalConnection class allowing, 792–793
 - dot (.) notation**
 - ActionScript, 42
 - ColdFusion, 977
 - downloading files**
 - canceling, 808
 - with FileReference, 808
 - drag and drop APIs, ListBase class, 616**
 - drag and drop, in AIR**
 - adding data to clipboard, 658–659
 - cross platform support, 299
 - dragging data from AIR application, 662–666
 - dragging data to AIR application, 666–669
 - events in, 661–662
 - NativeDragManager class, 660
 - overview of, 657
 - phases of, 661
 - reading data from clipboard, 659
 - reading/writing custom data from clipboard, 660
 - summary, 669
 - drag and drop, in Flex**
 - adding to nonenabled components, 652–653
 - custom drag proxies, 655–656
 - custom drag source formats, 655
 - enabling List for, 650
 - events, 651–652
 - maintaining data type information, 656
 - moving an item, 650–651
 - overview of, 647
 - preventing duplicates, 651
 - setting up drag initiator, 653–654
 - setting up drop target, 654–655
 - summary, 656
 - working with drag and drop-enabled components, 648–650
 - drag proxies, 655–656**
 - dragEnabled **property, 648**
 - DragManager **class, 652–653**
 - Drawing API, Flash Player**
 - bitmap fills, 469–470
 - blending modes, 478
 - complex fills, 470–471
 - curves, 464–465
 - Degrafa and, 478
 - fills, 467
 - filters, 477–478
 - gradient fills, 468–469
 - lines, 463–464
 - masks, 476–477
 - overview of, 461–462
 - programmatically drawing, 462–463
 - for programmatically skins, 424
 - programmatically skins, 471–475
 - shapes, 465–467
 - solid fills, 467–468
 - summary and, 479
 - dropEnabled **property, 648**
 - drop-in item renderers, 632–633**
 - DTO (Data Transfer Objects), 919**
 - dual deployment (Flex and AIR)**
 - ApplicationPersistence.as file, 1239
 - coding by interface, 1229–1231
 - conditional compilation, 1238–1239
 - configuring Flex compiler, 1240
 - overview of, 1229
 - pros/cons of coding by interface, 1231
 - pros/cons of conditional compilation, 1238
 - referencing main application
 - (MainAppEntryPoint.xml), 1233–1234
 - setting up Flex Builder projects, 1232
 - summary, 1241
 - working with
 - ApplicationPersistenceFactory, 1235–1238
 - working with IApplicationPersistence interface, 1235
 - duplication, preventing duplicates when dragging and dropping, 651**
 - dynamic classes**
 - ActionScript 3.0, 589
 - comparing OOP languages, 1362
 - Dynamic Help View, Flex Builder, 90**
 - dynamic objects, code syntax, 1148**
 - dynamic skinning. See programmatically skins**
- ## E
- ### E4X
- accessing XML objects using E4X operators and expressions, 579
 - ActionScript and XML before E4X, 573–576
 - ActionScript and XML using E4X, 576

eager relationships, DMS support, 1012

easing functions, animation and, 449–450

EasyMVC, 32

Eclipse

- character encoding and, 691–692
- command-line startup options, 182–183
- compilation slow down and, 182
- debugging features, 1255–1257
- Flex Builder and, 86–87, 149
- Groovy plug-in for, 902
- Heap View, 181–182
- Java development tools and, 902
- memory tuning, 184
- plug-in and third-party tools, 86
- workbench optimizations, 183–184

editors

- ActionScript, 483–484
- Flex Builder 3, 93
- item editors, 635–636
- Link With Editor button, 164
- RichTextEditor control, 254, 353–354, 365
- text editors, 85

effects. See also visual effects

- charts and, 686–687
- Flex Builder and, 118–119

else if statements, ActionScript conditional statements, 44–45

email validators, 602

@Embed directives, defining skins per object instance, 426

embedded assets

- development best practices and, 180
- skins and, 423–424

embeddedPlayer.html, 725–726

embedding

- HTML in AIR applications, 721–722
- log viewer, 1335–1337
- sound in ActionScript, 515–517
- sound in MXML, 517–519

encapsulation, OOP and, 1149–1150

encoding/decoding, URL components, 737–738

encrypting local data

- deleting encrypted data, 832
- overview of, 831
- reading encrypted data, 831–832
- writing encrypted data, 831

end points

- LCDS communication channels, 962
- in LCDS server architecture, 958
- streaming AMF to NIO end points, 1036

enterprise development, Flex and, 11

enumeration

- comparing OOP languages, 1358
- directories, 820

equality operators, 590

error(), log levels, 1329–1330

Error class, 858–859

error handling

- asynchronous events, 860
- asynchronous events in Debug Player, 862
- custom error types, 858–859
- events, 861–862
- exception bubbling, 856–858
- Flash Debug Player vs. Release Player, 853–855
- in functions, 860
- overview of, 853
- RESTful Web Services and, 893–894
- summary, 863
- synchronous code for, 853
- throw, 859–860
- try/catch/finally statements, 855–856

error property, 596

ErrorEvent class, 861

errors

- changing default error messages, 605–606
- custom server-side errors, 909
- FileReference, 805
- handling data input errors, 600–601
- handling event runtime errors, 373
- item-pending errors, 1031–1033
- LocalConnection class, 794

event classes

- ActionScript, 69–70
- Event base class, 369

event flow

- ActionScript, 71–72
- event priority, 373–374
- forwarding events, 374–376
- preventing/changing default behaviors, 374

event gateways, ColdFusion

- configuring, 976–977
- types of, 975–976

event handlers

- changing, 390–391
- for changing state, 387
- DataGrid control, 631–632
- event methods and, 71
- handleSubmit(), 373
- LCDS message events, 985–986
- methods for handling display events, 77–78
- overview of, 236
- preventing/changing default behaviors, 374
- Tree control, 627–628
- triggers compared with, 118

event listeners

- adding, 357
- application performance and, 1194–1195
- creating, 70
- data gathered by, 370
- impact on application performance, 1192
- MXML and ActionScript, 240
- MXML files and, 344
- overview of, 239

event model, ActionScript

- event classes and types, 69–70
- event flow, 71–72
- event process, 70–71
- EventDispatcher class, 72
- overview of, 69

event process, ActionScript, 70–71**event triggers. See triggers**

[Event] metadata, 242–243

EventDispatcher class, ActionScript, 72

EventDispatcher interface, 763

events, 854

- asynchronous events, for error handling, 860–862
- bubbling phase, 238
- Cairngorm events compared with Flash events, 1085
- capture phase, 236–237
- collection events for improving application performance, 1192–1193
- CollectionEvent class, 587–588
- composite components, 356–357
- creating custom, 241–242
- cross-module communication and, 1177–1182
- drag and drop, in AIR, 661–662

- drag and drop, in Flex, 651–652
- effect lifecycle, 457–458
- [Event] metadata, 242–243
- EventDispatcher interface, 763
- flash.events package, 80
- Flex framework, 228–229
- handling FileReference events, 804–805
- handling LCDS message events, 985–986
- inline event scripts, 141–143
- list events, 618
- listeners, 239–240
- native menus, 776–777
- overview of, 236
- setting validation trigger events, 606
- stopping, 238–239
- strategies for using custom, 243–244
- summary, 244
- target phase, 237
- test cases and, 1323
- types of, 241

events, advanced programming

- adding metadata to ActionScript and MXML files, 372–373
- custom data binding. See data binding, custom
- custom events vs. standard events, 369–371
- event priority, 373–374
- extending standard events, 371
- forwarding events, 374–376
- handling event runtime errors, 373
- one vs. many rule when creating custom events, 371–372
- overview of, 369
- preventing/changing default behaviors, 374
- summary, 381

events, Cairngorm

- dispatching, 1085–1086
- event classes, 1105
- event types, 1105–1106
- logging, 1131–1132
- overview of, 1104
- as parameter objects, 1106–1107
- presentation model dispatching, 1123
- self-dispatching events, 1107

events package, 369

exception bubbling, 856–858

exceptions. See *also* error handling; errors

- `DataService` class, 1014
- in Flash Player, 854
- execution, Flex RIA process flow, 37**
- expiration, LCDS messages, 982**
- explicit casting, ActionScript, 54**
- explicit parameter passing vs. parameter binding, 838–839**
- export releases, creating with Flex Builder, 165–167**
- exporting projects, with Flex Builder, 154**
- Expressions View, 1254–1255**
- extending Flex components**
 - composite components. See composite components
 - custom containers. See containers, custom summary, 367
- extending MXML components, 351–352**
- extends **keyword**
 - ActionScript 3.0, 344
 - defining subclasses and, 64
 - extending MXML components, 352
- Extensible Markup Language. See XML (Extensible Markup Language)**
- external applications, profiling, 1274–1276**
- external assets, loading**
 - images, 503–506
 - `Loader` class, 507–511
 - overview of, 503
 - RSL (Runtime Shared Libraries), 511
 - summary, 511
 - `SWFLoader` class, 506–507
- external references, ASDoc and, 1212**
- ExternalInterface class**
 - disabling browser integration, 718–719
 - Flex interaction with JavaScript via, 706–707
 - invoking ActionScript from JavaScript, 708–709
 - invoking JavaScript from Flex, 707
 - passing complex objects between JavaScript or ActionScript and Flex, 713–716
 - passing parameters between JavaScript or ActionScript and Flex, 709–711
 - passing return values between JavaScript or ActionScript and Flex, 712

F

- FABridge (Flex-Ajax Bridge), 717–718**
- Fade effect, built-in effects, 442**
- failure messages, FlexUnit, 1311–1312**
- Fast Views, Flex Builder, 116–117**
- Fat Controller anti-pattern, 1127–1128**
- `fatal()`, **log levels, 1329–1330**
- fault events, LCDS messages and, 984–985**
- fault handling**
 - `HTTPService` class, 893–894
 - remote methods calls, 1062–1063
 - Routes Web Services, with Java, 905
 - Routes Web Services, with .NET, 920–921
- feedback. See interactivity/feedback**
- file dialogs, native**
 - browsing for Open dialog, 822–824
 - browsing for Open Multiple dialog, 824–825
 - browsing for Save dialog, 825–826
 - directory browser, 821–822
 - overview of, 821
- file formats, 513–514**
- file management**
 - canceling file upload or download, 808
 - client-side code and, 806–807
 - downloading files with `FileReference`, 808
 - `FileReference` class, 804
 - handling `FileReference` events, 804–805
 - overview of, 803
 - prerequisites, 803–804
 - server-side code and, 807
 - server-side considerations, 810
 - in Subclipse, 211
 - summary, 811
 - uploading files, 805
 - working with multiple files, 808–810
- file management, with AIR**
 - checking existence of files and directories, 819
 - copying files asynchronously, 817–818
 - copying files synchronously, 817
 - creating temporary files and directories, 820–821
 - deleting encrypted data, 832
 - deleting files and directories, 819
 - encrypting local data, 831
 - enumerating directories, 820
 - `File` operations, 813

- filesystem controls, 828–831
- moving files and directories, 819–820
- native browsing for directory, 821–822
- native browsing for Open dialog, 822–824
- native browsing for Open Multiple dialog, 824–825
- native browsing for Save dialog, 825–826
- navigating paths, 814–816
- overview of, 813
- reading encrypted data, 831–832
- reading XML data from a file, 828
- reading/writing file data, 826
- referencing directories, 813–814
- referencing file, 814
- saving bitmap data as JPEG file, 827–828
- summary, 832
- synchronous vs. asynchronous API, 816–817
- working with native file dialogs, 821
- writing encrypted data, 831
- File object**
 - navigating paths, 814–816
 - overview of, 813
 - referencing directories, 813–814
 - referencing files, 814
- FileFilter class, 810**
- filename tag, AIR application configuration, 307**
- filenames, comparing OOP languages, 1360–1361**
- FileReference class**
 - canceling file upload or download, 808
 - downloading files with, 808
 - handling FileReference events, 804–805
 - overview of, 804
 - server-side considerations, 810
- FileReferenceList class, 808–810**
- files**
 - checking existence of, 819
 - comparing file sizes in Flex Builder, 123
 - creating file links with Flex Builder, 159–160
 - deleting, 819
 - impact of Flex on file size, 9
 - moving, 819–820
 - recovering deleted files from Local History, 191
 - referencing, 814
 - temporary, 820–821
- FileStream**
 - reading XML data from a file, 828
 - saving bitmap data as JPEG file, 827–828
 - working with, 826
- FileSystem Component Explorer, 829**
- filesystems**
 - controls, 828–831
 - Flex classes, 829
 - Flex enumeration modes, 830
 - local sandboxes, 1167–1168
- fileTypes tag, AIR configuration, 310**
- fill(), replacing lazy relationships, 1030–1031**
- fills**
 - bitmap fills, 469–470
 - complex fills, 470–471
 - Drawing API, 467
 - gradient fills, 468–469
 - solid fills, 467–468
- filters/filtering**
 - data, 583–584
 - flash.filters package, 80
 - in Flex Profiler, 1287–1288
 - LCDS messages, 987–988
 - log statements, 1330
 - Navigator filters, 163–164
 - overview of, 477–478
 - parcel list in DMS Fleet tracker, 1026
 - PixelBender toolkit, 458–459
- finally block, error handling an, 855–856**
- Firebug**
 - debugging PHP scripts, 886
 - network packet sniffing, 846
 - ThunderBolt logger for, 1339–1341
- Firefox, exporting Thawte certificate from, 1217–1218**
- Fireworks, Flex Builder integration with CS3 suite, 99–100**
- Flash Catalyst**
 - Flash-Flex workflow and, 485
 - future additions to Flex, 38
- Flash CS4**
 - Flash-Flex workflow and, 485
 - future additions to Flex, 38
- Flash Debug Player**
 - asynchronous error handling and, 862
 - debugging perspective in Flex Builder, 1245–1246
 - vs. Release Player, 853–855

Flash Media Development Server, 865

Flash Media Encoder (FME), 874

Flash Media Interactive Server (FMIS), 865

Flash Media Server. See FMS (Flash Media Server)

Flash Media Streaming Server (FMSS), 865

Flash platform

benefits of Flex and, 12

classes, 222

Debug Player vs. Release Player, 853–855

development tools in, 483–484

development workflows, 482–483

Flex Builder integration with CS3 suite, 99

history of Flex and, 17–18

IDE, 483–484

integration, 481

integration with Flex. See Flash/Flex integration overview of, 67–68

RIAs (Rich Internet Applications) and, 37

role in Flex applications, 217–218

Flash Player

AIR technologies and, 25

asynchronous events in Debug Player, 862

audio and video capabilities of, 513

AVMs and, 20–21

blending modes, 478

Debug Player vs. Release Player, 853–855

detecting system capabilities, 751

Drawing API. See Drawing API, Flash Player as event driven system, 369

exception handling in, 854

filters, 477–478

Flex 3 ecosystem and, 19–20

Flex 4 effects, 458–459

Flex deployment and, 3

full-screen mode, 757–760

full-screen mode limitations, 765–766

Garbage Collection and, 591

history of Flex and, 18–19

ID3 information and, 525–526

libraries, 398

loading SWF files, 506

LocalConnection class, see LocalConnection class

LSOs (local shared objects). See LSOs (local shared objects)

market penetration, 12

masks, 476–477

media formats, 513–514

MovieStar update, 514, 539, 873–874

myths regarding, 7–11

scale-9 grids, 429–430

security model. See security model, Flash Player SWFs files and, 7–8

version 10 releases, 22–23

version 9 releases, 21–22

working around 40KB limit for messages, 793–794

Flash Player APIs

3D graphics and pixel bender in Flash Player 10, 79–81

AIR APIs, 78–79

Display API, 74–78

FTE (Flash Text Engine) in Flash Player 10, 81

media APIs in Flash Player 10, 82–83

new in Flash Player 10, 79

overview of, 73

system integration APIs in Flash Player 10, 82

Flash Player Settings Manager, 797

flash prefix, indicating Flash classes, 222

Flash Remoting, enabling, 973

Flash runtime, AIR capabilities and, 786–787

Flash Text Engine (FTE), 81

`flash.desktop` package, 82

`flash.display` package, 80

`flash.events` package

Event base class, 369

new APIs in Flash Player 10, 82

overview of, 80

`flash.filters` package, 80

Flash/Flex integration

building custom Flex component, 492–494

building Flash skin for custom Flex component, 494–498

Flash Platform development tools, 483–484

Flash Platform development workflows, 482–483

Flash-to-Flex and Flex-to-Flash workflows, 483

Flex Skin Design Extension for Flash CS3, 486–490

future of, 485

integration kit bugs, 491

overview of, 481

planning custom Flex component, 490–492

- skinning custom Flex component, 490, 498–500
- summary, 501–502
- testing custom Flex component, 501
- workflow best practices, 494
- workflow integration options, 483–485
- Flash/Flex integration, advanced**
 - adding custom states, 557–558
 - adding transitions, 558
 - bringing Flash into Flex, 547
 - bringing Flex classes into Flash, 545–546
 - custom states and transitions in Flex, 556–557
 - examining `FLVPlayback` compilation process, 555–556
 - Flash shortcut for compiling in Flex Builder, 566
 - Flex Component Kit for Flash CS3, 547–551
 - `FLVPlayback` (Flash) vs. `VideoDisplay` (Flex), 551–552
 - importing custom Flex classes into Flash, 546
 - importing Flash components into Flex, 551
 - importing native Flex classes into Flash, 547
 - launching Flash documents from Flex Builder, 560
 - overview of, 545
 - porting `FLVPlayback` to Flex, 552
 - Punkhead and Ticklehead animation examples, 559–567
 - summary, 567
 - using `FLVPlayback` in Flex, 552–555
- `flash.geom` **package, 79–80**
- `flash.media` **package, 82**
- `flash.net` **package, 83**
- `flash.system` **package, 83**
- `flash.system.Capabilities` **class, 751–752**
- `flash.text.engine` **package, 81**
- `flash.ui` **package, 82**
- `flashVars` **property, for passing data into Flex applications, 704–706**
- Flex, introduction**
 - benefits of, 12–15
 - competition for, 6–7
 - history of, 17–19
 - integration with Flash. See Flash/Flex integration
 - myths regarding, 7–11
 - RIAs and, 4–6
 - what it is, 3
- Flex 3 ecosystem**
 - ActionScript 3.0, 27–28
 - AIR (Adobe Integrated Runtime), 24
 - communication protocols, 32–33
 - data formats, 33–34
 - Flash platform and, 67–68
 - Flash Player, 19–24
 - Flex Builder 3, 29
 - Flex RIA process flow and, 35–38
 - Flex SDK, 29
 - frameworks and APIs, 30–32
 - future additions to, 38–39
 - history of Flex and, 17–19
 - languages, 27
 - MXML and CSS, 28
 - Pixel Bender, 29–30
 - related technologies in, 17
 - runtimes, 19
 - servers, 34–35
 - summary, 39–40
- Flex 4 (Gumbo)**
 - effects, 458–459
 - future additions to Flex, 38
 - MVC approach to component architecture, 1082
- Flex Builder**
 - ActionScript code assist and, 171
 - ActionScript projects created with, 120–123
 - class outlines in Outline View, 174–175
 - code editing with, 170
 - code introspection, 173
 - code preferences, 173
 - code signing in, 1218–1220
 - coding by interface and, 1232
 - comparing file sizes, 123
 - compilation process, 105
 - compiling application in Design mode, 119
 - Components and Properties views, 112–113
 - CS3 suite integration, 99–100
 - data binding and, 117–118
 - Debug mode, 1249–1250
 - debug releases, 164–165
 - debugging perspective in, 1246
 - design mode enhancements, 98
 - Design View. See Design View, Flex Builder
 - development enhancements, 99
 - development process, 104
 - Eclipse and, 86–87
 - editors, 93

Flex Builder (continued)

- export releases, 165–167
- exporting projects, 154
- Fast Views, 116–117
- file linkages, creating, 159–160
- Flash documents launched from, 560
- Flash shortcut for compiling in, 566
- Flash-Flex workflow and, 483–484
- Flex Application settings, 150–151
- Flex applications created with, 108
- Flex Build Path settings, 151
- Flex Compiler settings, 151–153
- folder linkages, creating, 158–159
- folder structure, creating, 157–158
- getting started with, 85–86
- “Go into” browsing option, 160–161
- Help system, 86
- Import Web Service (WSDL) tool, 922–924
- importing projects, 154–157
- keyboard shortcuts, 176–177
- language intelligence and, 170
- languages in Flex 3 ecosystem, 29
- layout constraints, 113–114
- LCDS project created in, 963
- learning Flex and getting help, 87–90
- library creation, 399–400
- Link With Editor button, 164
- Mark Occurrences button, 172–173
- MXML code assist, 172
- navigating design area, 114–115
- navigating projects, 157
- Navigator filters, 163–164
- new wizards in version 3, 98
- other outline views, 175–176
- other project properties, 153–154
- overview of, 85, 103, 149
- perspectives, 94–98
- profiling. *See* Flex Builder Profiler
- project creation with, 107–111
- project management with, 149
- Project Navigator, 399
- project settings files, 150
- project types, 106–107
- publishing process, 106
- self-signed certificates with, 1221
- source view option, enabling, 167–170

- stepping through code, 1250–1252
- summary, 100, 123–125, 176–177
- syntax highlighting, 171
- themes, 433–434
- triggers and effects, 118–119
- undocumented [SWF] metadata tags for
 - configuring compiler settings, 123–125
- viewing display hierarchy with Outline View, 116
- views, 93–94
- workbench, 90
- working sets, 161–163
- workspaces, 91–93

Flex Builder, customizing

- code repositories, 190
- command-line startup options, 182–183
- compilation slow down and, 182
- custom shortcuts, 188–189
- Heap Status indicator, 181–182
- layout options for increasing productivity, 189–190
- Local History options, 191–192
- memory tuning, 184–185
- overview of, 179
- project development best practices, 179–181
- Subclipse for version control. *See* Subclipse
- summary, 214
- tweaking Eclipse settings, 181
- workbench optimizations, 183–184
- workspace preferences, 190
- workspaces list, editing, 186–187

Flex Builder Profiler, 1264–1265

- clearing accumulated performance data, 1287
- creating own profiler, 1300–1301
- detecting memory leaks, 1292–1296
- development enhancements in Flex Builder 3, 99
- examining object references, 1279–1281
- filtering in, 1287–1288
- in Flex 3, 1271
- Garbage Collection, 1290
- inspecting cause of object instantiation, 1281–1283
- launching Flex 3 profiler, 1272–1276
- Memory Usage graph, 1276–1277
- navigating between view, 1291
- optimizing algorithm with performance profiler, 1296–1300
- preferences, 1290

profiling perspective in Flex 3, 1271–1272
 saving/loading profiling data, 1289
 summary, 1301
 taking memory snapshots, 1278–1279
 taking performance snapshots, 1285–1286
 using memory profiler, 1276
 using performance profiler, 1284–1285
 viewing impact of live objects on memory usage, 1277
 viewing method statistics, 1286–1287
 viewing source code, 1291
 viewing Zero Time Methods, 1290–1291
 views requiring two memory snapshots, 1283–1284

Flex compiler. See MXMLC compiler

Flex Component Kit for Flash CS3, 547–551

creating custom components, 549–550
 creating Flex component in Flash, 548–549
 custom states and transitions in Flex, 556–557
 installing, 548
 overview of, 547–548
 porting Flash native components into Flex, 551
 setting up component bounding box, 551
 using custom components, 550

Flex framework

container classes, 224–225
 Flash and Flex class packages, 222
 formatters, validators, and collection classes, 226–228
 frame-by-frame execution, 218–220
 functions and events, 228–229
 life cycle, 220–222
 list classes, 225–226
 overview of, 30, 218
 summary, 229
 UIComponent base class, 222–224

Flex Integration screen, ColdFusion Administrator, 973

Flex Language Reference, 1327

Flex Navigator

filters, 163–164
 “Go into” menu options, 160
 Link With Editor button, 164
 working sets and, 161–163

Flex RIA process flow, 35–38

communication, 37–38

design and development, 36–37
 execution, 37
 infrastructure, 38
 overview of, 35–36

Flex Skin Design Extension, for Flash CS3, 485, 486–490

Flex-Ajax Bridge (FABridge), 717–718

FlexLib, third party frameworks and components for Flex, 31

`.flexProperties`, **project settings files, 150**

FlexUnit, 31

assembling test suite, 1308–1309
 assertion functions, 1311
 BDD (Behavior-Driven Development), 1324
 continuous integration with Ant and Maven scripts, 1322–1323
 creating test case, 1308
 downloading, 1307
 eventful test cases, 1323
 failure messages, 1311–1312
 fleshing out test case, 1312–1313
 happy and unhappy paths, 1313
 library, 1305
 mock objects and dependencies and, 1318–1322
 organizing test suites, 1318
 overview of, 1303
 preparing for unit testing, 1307
 running tests, 1306
 setting up/ tearing down test cases, 1315–1317
 stages of unit testing, 1310
 strengthening tests, 1314–1315
 summary, 1324–1325
 TDD (Test-Driven Development), 1307
 test coverage, 1323–1324
 test failure, 1309–1310
 unit tests, test cases, and test suites, 1306

Flow container, extending, 363–365

FLV files

Flash video formats, 33
 playing VOD (video on demand), 866

FLVPlayback (Flash)

examining FLVPlayback compilation process, 555–556
 porting to Flex, 552
 using in Flex, 552–555
 vs. VideoDisplay (Flex), 551–552

FME (Flash Media Encoder), 874

FMIS (Flash Media Interactive Server), 865

FMS (Flash Media Server)

- Adobe servers used by Flex applications, 34
- bandwidth check system, 869–872
- buffer control, 873–874
- developing for FMS in Flex, 868–869
- exercise playing video on demand, 866–867
- incorporating data into streams, 877–879
- `NetStream` class and, 872–873
- overview of, 865
- publishing and recording live streams, 876–877
- publishing video, 874–875
- setting up application instances, 867–868
- setting up FMS applications, 875
- streaming preexisting video, 875–876
- streaming video with, 865–866
- summary, 880

FMSS (Flash Media Streaming Server), 865

folders

- AIR configuration tags and, 309
- creating folder links with Flex Builder, 158–159
- creating folder structure with Flex Builder, 157–158
- `libs` folder, 405–406

- for `each..in` operator, introspection in **ActionScript, 56–57**

- for **statements, ActionScript, 46**

- for `..in` operator, introspection in **ActionScript, 56**

Form container

- layout containers, 262–263
- tags, 283

- `format()` method, **596**

formats

- ASDoc formats, 1205
- image formats, 246
- media formats, 513–514

- `Formatter` classes, **596**

formatters, 226–227

- base class (`Formatter`), 596
- centralizing, 609–610
- classes/subclasses of, 226–227
- `CurrencyFormatter`, 600
- `DateFormatter`, 598–599
- handling data input errors, 600–601

- how to utilize, 597

- `NumberFormatter`, 599–600

- overview of, 595

- `PhoneFormatter`, 598

- standard, 596–597

- summary, 612

- `ZipCodeFormatter`, 597–598

forms

- creating, 281–283

- `defaultButton` on, 286

- Flex Builder Design View for building, 285–286

- submitting form content, 284–285

fragment identifiers, in URLs, 733

frame rates, application performance and, 1198

frame-by-frame execution, Flex framework, 218–220

frameworks

- Adobe frameworks for development of Flex or AIR applications, 31
- Flex. See Flex framework
- Flex 3 ecosystem, 30–32
- Grails framework. See Grails framework
- third party frameworks and components for Flex, 31

front controllers, Cairngorm, 1102–1104

- command execution, 1086–1087

- decentralizing, 1136–1137

- declaring, 1104

- disposable, 1137–1138

- internal classes, 1102

- overview of, 1102

- `ShopController`, 1102–1104

FTE (Flash Text Engine), 81

full-page content, Flex applications as, 703

full-screen display

- enabling, 757–760

- enabling with AIR, 787

- limitations of, 765–766

- video and, 540–542

- `Function` tag, **ActionScript in MXML and, 145–146**

functions

- ActionScript in MXML and, 145–146

- declaring, 59–60

- documenting with ASDoc, 1207–1208

- error handling in, 860

Flex framework, 228–229
 getter/setter methods acting as, 1155–1156
 naming conventions, 1144–1145
 overriding, 1152–1153

FXG, future additions to Flex, 38

G

Garbage Collection

comparing OOP languages, 1356
 freeing up memory, 182
 memory management and, 591
 running manually, 1290

GET, HTTP route request, 889

`getItem()`, **1030–1031**

`getLogger()`, **1328–1329**

getters/setters, 1153–1157

acting as functions, 1155–1156
 ActionScript, 63
 backing variable and, 324
 calculating values with, 1157
 data-checking with, 1156
 documenting with ASDoc, 1209–1210
 overriding, 1157
 overview of, 1153–1154
 public variables compared with, 1154
 read/write access control and, 1154–1155

Git, version control system, 192

global variable notation, comparing OOP languages, 1357

globalization (g11n). See also internationalization (i18n); localization (L10n)

of AIR applications, 699–700
 internationalization and localization and, 690
 loading resources at runtime, 698
 summary, 700

Glow effect, built-in effects, 442–443

“Go into” browsing option, 160–161

Google, `swfobject.js`, 726

gradient fills, Drawing API, 468–469

Grails framework

configuring for Routes Web Service, 900–901
 overview of, 899, 902–903
`Route.groovy` domain class, 903
`RouteService.groovy` class, 904

Granite Data Services, third party solutions for RPC services, 851

graphic filters. See filters/filtering

graphics. See also visual effects

bitmap caching, 1198–1199
 Degrafa (Declarative Graphics Framework), 478
 frame rates, 1198
 new APIs in Flash Player 10, 79–81

Graphics class

curves, 464
 Flash and, 462
 lines, 463–464
 shapes, 465–467

Grid container, layout containers, 260–261

Groovy Java language

configuring for Routes Web Service, 900–901
 Eclipse plug-in, 902
 Grails based on, 899
`Route.groovy` domain class, 903
`RouteService.groovy` class, 904

GroupingCollection, 638–639

H

H.264

Flash Player supporting, 539
 streaming video, 865–866, 872
 usable media formats, 514

Halo theme, 433

hand cursor, 296

happy and unhappy paths, FlexUnit, 1313

hashes/hash maps. See associative arrays

Hbox container, layout containers, 257–258

HD (high definition) media format, 514

HDividedBox container, layout containers, 258–259

headers

data controls, 636–637
 grouping columns under common, 638

Heap Status indicator, 181–182

Heap View, memory management and, 181

height tag, AIR initial window configuration, 308

help, Flex Builder Help system, 86–90

helper classes, 226–228

Hibernate assembler, for DMS, 1015

hierarchical data

in `AdvancedDataGrid` control, 639
 DMS and, 1037

HighLowOpenClose (HLOC) charts, 677

- history.js, **JavaScript library**, 739
 - HistoryManager **API**, **deep linking**, 749
 - HLOC (HighLowOpenClose) charts**, 677
 - horizontal lines, control for**, 247–248
 - horizontal sliders, control for**, 253
 - HorizontalList, TileBase **class**, 621–622
 - HRule **control**, **static controls**, 247–248
 - HSlider **control**, **input controls**, 253
 - HTML (Hypertext Markup Language)**
 - ASDoc and, 1205, 1211–1212
 - deep linking and, 731
 - escaping HTML markup, 137
 - flashVars property for passing data into Flex applications, 704–706
 - Flex forms compared with HTML forms, 282
 - SWFs embedded in, 703
 - TileBase class compared with HTML tables, 621
 - HTML mashups, in AIR**
 - calling ActionScript from JavaScript, 728–729
 - calling embedded JavaScript from ActionScript, 726–728
 - controlling HTML DOM from ActionScript, 729
 - displaying HTML in, 722
 - displaying in AIR, 722
 - embedding in AIR applications, 721–722
 - integrating JavaScript/HTML into, 726
 - integrating JavaScript/HTML into AIR, 726
 - loading in-memory HTML content, 724
 - loading in-memory HTML content into AIR, 724
 - loading local file HTML content, 724–726
 - loading local file HTML content into AIR, 724–726
 - loading remote HTML content, 723–724
 - loading remote HTML content into AIR, 723–724
 - reasons for embedding HTML in AIR applications, 721–722
 - summary, 730
 - YouTube Chromeless Player API in example AIR application, 722
 - HTMLLoader **class**, 726–727
 - HTTP verbs**, 881–882, 883
 - HTTPProxyAdapter**, **BlazeDS integration points**, 1055
 - HTTPService **class**
 - AsyncToken, 890
 - BlazeDS and, 1067–1068
 - choosing protocol server for, 850
 - communication with remote servers, 836
 - declaration in RESTful Web Services, 889
 - fault handler, 893–894
 - HTTP verbs and, 882
 - invoking from BlazeDS, 1071
 - LCDS interaction with, 957
 - logging requests, 846–847
 - MXML vs. ActionScript classes, 838
 - RESTful Web Services. *See* RESTful Web Services
 - result handler, 892–893
 - RPC libraries and, 837
 - working with, 839–840
 - hyperlinks, deep linking. See deep linking**
 - Hypertext Markup Language. See HTML (Hypertext Markup Language)**
- █
- IApplicationPersistence **interface**, **coding by interface**, 1235
 - IBitmapDrawable **interface**, 223
 - ICollectionView, **Collections API**
 - filtering data, 583–584
 - IViewCursor, 584–585
 - overview of, 582
 - sorting data, 582–583
 - icon **tag**, **AIR configuration**, 309
 - icons**, ListBase **class supporting**, 616–617
 - Id **tag**, **AIR application configuration**, 306
 - ID3 information**, for Sound **class**, 525–526
 - IDEs (integrated development environments)**
 - application authoring and, 481
 - development workflows and, 482–483
 - Flash, 483–484
 - Flex Builder as IDE for Flex, 108
 - IEventDispatcher **interface**, 223
 - if **statements**, **ActionScript conditional statements**, 44–45
 - IFlexDisplayObject **interface**, 223
 - if-then-else **conditionals**, **ternary operators and**, 1146–1147
 - IList **interface**, 582
 - Illustrator, **Flex Builder integration with CS3 suite**, 99
 - ILOG Elixir, **third party frameworks and components for Flex**, 31

- `ILogger` **interface**, 1328–1329
- image browsers**, 743–747. *See also* **Web browsers**
 - adjusting view state to correspond to
 - navigational state, 742–743, 746–747
 - deep linking architecture and, 742, 745–746
 - multiple views, 745
 - navigational states and, 744
 - using HTML navigation to change states, 747
 - working with navigational states and, 739–740
- `Image` **class**
 - `Loader` class compared with, 507–508
 - loading images and, 503
 - static controls for, 246
- images**
 - dragging to/from AIR applications, 662–663
 - formats, 246
 - loading, 503–506
 - loading image data from metadata, 542–543
 - manipulation features in Flash Player, 475
 - skins and, 423–424
- implicit casting, ActionScript**, 54
- Import Skin Artwork Wizard**
 - importing skins with, 499
 - new in Flex Builder 3, 98
- Import Web Service tool, WSDL**, 922–924
- Import Web Service Wizard, new in Flex Builder 3**, 98
- importing Flash components into Flex**, 551
- importing projects, with Flex Builder**, 154–157
 - multiple projects from directory, 156
 - multiple projects from repository, 157
 - multiple projects from ZIP archive, 155
 - single projects, 154
- importing SQLite data**, 941–943
- importing XML**
 - improving SQLite importer, 948–949
 - into SQLite database, 944–945
- `IMXMLObject` **interface**, 133
- in operator, introspection in ActionScript**, 55
- `includeInLayout` **property**
 - customizing composite components, 365–366
 - layout strategies, for components, 279–280
- indentation, code syntax and**, 1145–1146
- indexing, search engines and**, 747–748
- `info()`, **log levels**, 1329–1330
- infrastructure, Flex RIA process flow and**, 38
- inheritance**
 - ActionScript, 64–65
 - comparing OOP languages, 1362
 - CSS styles and, 418–419
 - OOP and, 1151–1152
 - overriding functions, 1152–1153
 - visual components inheriting from `UIComponent` class, 133
- initial window configuration tags, AIR configuration file**, 307–308
- inline ActionScript**, 141–143
- inline CSS**, 139–140
- inline event scripts**, 141–143
- inline item renderers**, 633
- inline styles, vs. CSS styles**, 411–413
- input controls**
 - `Button` and `LinkButton` control, 249
 - `CheckBox` control, 251
 - `ColorPicker` control, 256
 - `ComboBox` control, 252
 - `DateChooser` and `DateField` controls, 250–251
 - `HSlider` and `VSlider` controls, 253
 - `NumericStepper` control, 250
 - overview of, 248–249
 - `PopUpButton` control, 254–255
 - `PopUpMenuButton` control, 255
 - `RadioButton` and `RadioButtonGroup` controls, 252–253
 - `RichTextEditor` control, 254
 - `TextArea` control, 249–250
- INSERT statements**
 - SQLite data transactions and, 941
 - XML data into SQLite databases, 947–948
- Install Badge, AIR**
 - customizing installation, 1224–1225
 - Launch Now feature for launching installed applications from a web page, 1225–1226
 - for one-click install from web pages, 1223–1224
- installation screens, ColdFusion**, 971–973
- `installFolder` **tag, AIR configuration**, 309
- `instanceof` **operator, introspection in ActionScript**, 55–56
- instant messaging system, creating in DMS**, 1033–1034
- instantiation**
 - creating objects from classes, 58

instantiation (continued)

in life cycle, 228

steps in writing ActionScript components, 319

integrated development environments. See IDEs (integrated development environments)

integration

disabling browser integration, 718–719

Flash/Flex integration. See Flash/Flex integration

Java/Flex integration. See Web Services, with Java of media content, 515

integration kit bugs, Flash/Flex integration, 491

integration points (adapters), BlazeDS, 1055

interactivity/feedback

cursors. See cursors

custom tooltips, 286–288

effects and, 437

forms. See forms

pop-ups. See pop-ups

interfaces

ActionScript, 64–65

comparing OOP languages, 1361–1362

drag and drop and, 647

OOP and, 1157–1158

server-side DMS, 1014

internal access modifier, ActionScript, 61

internal functions, variable scope and, 1151

internationalization (i18n)

@Resource, 695

overview of, 689–690, 694

ResourceBundle, 695–696

ResourceManager, 696–698

Internet

media formats, 513–514

monitoring traffic on, 1257–1258

interoperability, benefits of Flex, 13

interpreted tags, MXML, 134–135

introspection, ActionScript, 55–57

for each..in operator, 56–57

for..in operator, 56

instanceof operator, 55–56

is operator, 55

in operator, 55

overview of, 55

typeof operator, 56

IntroToAir-app.xml descriptor file

building AIR applications, 302

modifying, 314–316

overview of, 304–306

IntroToAir.mxml file

building AIR applications, 302

modifying, 311–313

overview of, 303–304

invalidate functions, calling, 324–325

invalidateDisplayList()

adding properties to ActionScript components, 324–325

adding properties to MXML components, 346

invalidateProperties()

adding properties to MXML components, 326–327

applying properties to ActionScript component, 326–327

inversion of control (IoC)

defined, 1305

presentation models and, 1125

IResponder interface, RPC and, 906–909

Iris effect, built-in effects, 443

IRouteService interface, 918–919

is operator, introspection in ActionScript, 55

issue reporting system, Adobe, 1262–1265

item editors

creating always present editor, 636

overview of, 635–636

item renderers

custom component item renderers, 634–635

design mode support for, 98

drop-in item renderers, 632–633

inline item renderers, 633

overview of, 632

ItemClickEvent, 369, 371

item-editing functions, List class, 618–619

item-pending errors, LCDS

overview of, 1031–1032

repeated attacks for overcoming, 1032–1033

itemRenderer property

for displaying lists, 270

ListBase class, 617

items, dragging and dropping, 650–651

iteration, Dictionary class and, 591

iTunes, usable media formats, 514

IUIComponent interface

ActionScript component extending, 321

UIComponent class implementing, 223

IViewCursor, Collections API and, 584–585

J**J2EE settings, in LCDS project, 964–965****Java**

- coding remote Java objects, 1064
- Data Services Stress Testing Framework and, 1039–1040
- debugging server-side code, 1255–1257
- developer ease in learning Flex, 14
- Eclipse and, 902
- Java/Flex integration. See Web Services, with Java LCDS. See LCDS (LiveCycle Data Service)
- mapping ActionScript types to Java types, 1065
- object adapter for DMS, 1014
- remote logger, 1332–1334
- stock feed, 994–995

Java, comparing OOP languages

- access modifiers, 1362
- block-level scoping, 1356–1357
- classes, filenames, and packages, 1360–1361
- console output and, 1357
- data types and, 1357–1359
- declaring variables, 1359–1360
- dynamic classes, 1362
- global variable notation, 1357
- inheritance, 1362
- memory management, 1356
- overview of, 1355
- polymorphism, 1361–1362
- regular expressions, 1360
- runtime environments, 1355–1356
- statement termination, 1360
- summary, 1363
- type casting, 1359

Java Message Service (JMS)

- configuring JMS queue for PDF generator, 999
- creating JMS queue listener for PDF generator, 1001–1002

Java Open Transaction Manager (JOTM), 967–968**Java virtual Machine (JVM), 1356****JavaAdapter, BlazeDS integration points, 1055****Javadoc, LCDS, 955****JavaScript**

- browser integration via FABridge, 717–718
- browser interaction with, 706–707
- calling ActionScript from, 728–729
- calling embedded JavaScript from ActionScript, 726–728

converting data types to/from ActionScript, 1019

HTML content interacting with, 726

invoking ActionScript from, 708–709

invoking form Flex, 707

passing complex objects between JavaScript and Flex, 713–716

passing parameters between JavaScript and Flex, 709–711

passing return values between JavaScript and Flex, 712

Jetty application server, 900**JMS (Java Message Service)**

- configuring JMS queue for PDF generator, 999
- creating JMS queue listener for PDF generator, 1001–1002

JMSAdapter, BlazeDS integration points, 1055**JOTM (Java Open Transaction Manager), 967–968****JPEG files, saving bitmap data as, 827–828****JVM (Java virtual Machine), 1356****K****keyboard shortcuts**

- Flex Builder, 176–177
- launching profiling from, 1274

keys

- objects as, 590
- strings as, 589
- weak keys, 591–592

keywords, reserved in ActionScript, 43**Komodo, 886****L**

Label control, static controls, 245–246

labels

- axis labels on charts, 681
- customizing, 212
- ListBase class supporting, 616–617

language intelligence, development**enhancements in Flex Builder 3, 99****languages**

- character encoding and, 691–692
- internationalization and, 690
- localization and, 690, 699–700

languages, in Flex 3 ecosystem, 27–30

Launch Now, Install Badge, 1225–1226

layout

- ActionScript components, 328–330
- advanced constraints, 275–276
- composite components, 357
- constraint columns, 276–277
- constraint positioning, 272–273
- constraint rows, 277–278
- constraint sizes, 278
- constraints in ActionScript, 278
- containers. See layout containers
- coordinate positioning, 271
- customizing in Flex Builder 3, 189–190
- `includeInLayout` property, 279–280
- MXML components, 345
- overview of, 271
- performance considerations and, 280
- positioning by alignment, 274–275
- positioning by percentage, 273–274

layout constraints

- absolute layout (positioning) and, 112
- using in Flex Builder, 113–114

layout containers

- `ApplicationControlBar`, 264
- `Canvas`, 256–257
- `ControlBar`, 263
- creating custom containers using MXML components, 346
- `Form`, 262–263
- `Grid`, 260–261
- `HDividedBox` and `VDividedBox`, 258–259
- overview of, 256
- `Panel`, 259–260
- `Tile`, 261–262
- `VBox` and `Hbox`, 257–258
- `layoutChrome()`, **346–348**

lazy loading, data, 1190–1191

lazy relationships

- DMS supported relationships, 1012
- replacing with `getItem()` or `fill()`, 1030–1031

LCDS (LiveCycle Data Service)

- AMF and, 962–963
- BlazeDS compared with, 1053
- client architecture, 956–958
- communicating stress test results and, 1044

- communication channels, 961–962
- configuring application server, 966–967
- creating LCDS project in Flex Builder, 963
- data service solutions, 956
- Data Services Stress Testing Framework. See Data Services Stress Testing Framework
- DMS. See DMS (Data Management Service)
- Message Service. See Message Service, LCDS
- overview of, 953–955
- PDF generation, 961
- project build path, 965–966
- project J2EE settings, 964–965
- project name and server technology settings, 963–964
- Proxy Service, 959–960
- Remoting Service, 959
- resources for, 955
- server architecture, 958–959
- service types, 959
- services required in stress testing, 1044
- steps in stress testing LCDS applications, 1041
- summary, 969, 976–977
- test application and, 1041–1042
- transaction managers for, 967–968
- troubleshooting, 968–969

LCDS (LiveCycle Data Service), ColdFusion and

- administrator screens, 973–975
- data management, 976–977
- event gateways, 975–976
- installation screens, 971–973
- integration, 971

LCDS Developer Guide, 955, 1036

LCDSTestManager class, 1042

legends, chart, 683

libraries

- ActionScript, 397
- adding sources for content, 400
- adding SWCs to, 406–407
- automation framework, 1345
- Cairngorm, 1084
- COMPC for creating, 402–404
- creating SWCs, 400–402
- Flash Player, 398
- Flex, 398
- Flex Builder for creating, 399–400
- FlexUnit, 1305

- history.js (JavaScript library), 739
- libs folder, 405–406
- overview of, 397
- RPC (remote procedure call), 837
- RSL (Runtime Shared Libraries), 511
- sharing source directories and, 404–405
- summary, 408
- SWCs vs. SWFs, 399
- themes and, 435
- types of projects you can create in Flex
 - Builder, 106
 - using assets from, 407
- Library Path, Flex Build Path settings, 151**
- library symbols, mapping Flash library symbols to Flex styles, 500–501**
- libs folder, 405–406
- license agreement screen, ColdFusion, 972**
- life cycle, ActionScript components, 323–324**
- life cycle, Flex components**
 - applying to custom containers, 362–363
 - Flex framework and, 220–222
- line charts, 677–678**
- lines, drawing, 463–464**
- Link With Editor button, Flex Builder, 164**
- LinkBar **control, navigation components, 267**
- LinkButton **control, input controls, 249**
- links**
 - creating file links with Flex Builder, 159–160
 - creating folder links with Flex Builder, 158–159
 - to external ActionScript, 146–147
 - to external CSS, 140–141
 - to external page URL, 706
- list classes**
 - classes/subclasses of, 226
 - enabling for drag and drop, 650
 - Flash and Flex class packages, 225–226
 - Flex filesystem classes, 829
 - Flex framework, 225–226
 - item-editing functions, 618–619
 - List class, 617–618
 - List class descendents (Tree and Menu), 619
 - overview of, 225
- list components**
 - adding effects to data containers, 448
 - controls, 268–270
 - creating menu objects, 619–620
 - data tips of ListBase class, 616
 - drag and drop APIs of ListBase class, 616
 - HorizontalList, 621–622
 - icons of ListBase class, 616–617
 - item-editing functions, 618–619
 - itemRenderer property, 617
 - labels of ListBase class, 616–617
 - list descendents, 619
 - ListBase class, 613–614
 - Menu class, 619
 - overview of, 613
 - selection and update APIs of ListBase class, 614–616
 - summary, 622
 - TileBase class, 621
 - TileList, 621–622
 - working with Menu class, 620
- ListBase **class**
 - data tips, 616
 - drag and drop APIs of, 616
 - icons of, 616–617
 - itemRenderer property, 617
 - labels of, 616–617
 - list classes and subclasses, 226
 - overview of, 613–614
 - selection and update APIs of, 614–616
- ListCollectionView
 - ArrayCollection class, 586
 - overview of, 586
 - XMLListCollection class, 586–587
- Live Object View, Memory Profiler, 1277**
- LiveCycle Data Services, Community Edition, 956**
- LiveCycle Data Services Enterprise Suite, 35, 956**
- LiveDocs, Adobe, 977**
- loadBytes() **method, 508–511**
- Loader **class, 503, 507–511**
- loadImage() **function, 506**
- loading audio, 522**
- loading external assets**
 - images, 503–506
 - Loader class, 507–511
 - overview of, 503
 - RSL (Runtime Shared Libraries), 511
 - summary, 511
 - SWFLoader class, 506–507

loading HTML content

loading HTML content, into AIR

- in-memory content, 724
- local file content, 724–726
- remote content, 723–724

loading image data from metadata, 542–543

loading resources at runtime, 698

loading video, 530–534, 536

local file management. *See* file management, with AIR

Local History options, Flex Builder 3, 191–192

local namespace, MXML, 132

local sandboxes

- local-trusted sandbox, 1169
- local-with-filesystem sandbox, 1167–1168
- local-with-networking sandbox, 1168
- overview of, 1167

local shared objects. *See* LSOs (local shared objects)

LocalConnection class

- allowing domains, 792–793
- building support for custom data types, 799–801
- overview of, 789
- registerClassAlias(), 800–801
- setting up receiver, 790–791
- setting up sender, 791–792
- specifying connection name, 792
- standard errors, 794
- summary, 799–801
- SWF to SWF communication, 790
- working around 40KB limit for messages, 793–794
- working with data, 793

locale chains, 693–694

locales

- copylocale.exe, 694
- locale chains, 693–694
- setting up in Flex projects, 692–693
- switching at runtime, 697–698

localization (L10n)

- of AIR applications, 699–700
- character encoding, 691–692
- copylocale.exe, 694
- Flex project settings for, 692–693
- locale chains, 693–694
- overview of, 689–690
- properties files for, 690–691

log(), ILogger interface, 1329

log levels

- filtering based on, 1330
- overview of, 1329–1330

loggers

- getting right logger, 1328–1329
- Java remote logger, 1332–1334
- stopwatch logger, 1337–1339
- ThunderBolt logger for, 1339–1341

logging

- in BlazeDS, 1059–1060
- Cairngorm events and commands, 1131–1132
- creating client that logs remotely, 1334–1335
- creating logging target, 1328
- creating remote logging target, 1331–1332
- customizing, 1331
- debugging and, 1259–1260
- embedding log viewer, 1335–1337
- filtering log statements, 1330
- getting right logger, 1328–1329
- implementing Java remote logger, 1332–1334
- log levels and, 1329–1330
- objects, 1260
- overhead and, 1330–1331
- overview of, 1327
- removing logging statements from production code, 1341–1342
- stopwatch logging, 1337–1339
- stress testing and, 1046
- summary, 1342
- ThunderBolt logger for Firebug, 1339–1341

logging, in RPC

- calls to web services, 848–850
- HTTPService requests, 846–847
- WebService initialization, 847–848

logging, LCDS

- client-side, 968
- server-side, 969

lookup tables. *See* associative arrays

loops

- code syntax, 1145–1146
- iterating through Dictionary class, 591
- overview of, 46–47
- for statement, 46
- while and do while statement, 46–47

LSOs (local shared objects)

- building support for custom data types, 799–801

creating, 795
 deleting properties and destroying, 797
 directory locations, 798–799
 disk storage settings, 797
 explicitly saving, 795–796
 overview of, 794
 registerClassAlias(), 800–801
 storing/retrieving data, 795
 summary, 799–801

M

Mac OS X. See OSs (operating systems)

Macromedia, 18–19

main class, **ActionScript display lists, 69**

MainAppEntryPoint.mxml, **1233–1234**

maintenance, benefits of MVC frameworks, 1081

MainWindow.mxml file, **313–314**

managed data, DMS and, 1009, 1011

managed relationships, DMS and, 1011–1012

managers

BindingManager class, 1260–1261
 BrowserManager API, 733–737
 CursorManager class, 295
 DragManager class, 652–653
 Flash Player Settings Manager, 797
 HistoryManager API, 749
 LCDSTestManager class, 1042
 NativeDragManager class, 660
 PopUpManager class, 291–293
 ResourceManager class, 696–698
 SchemaTypeManager class, 909–910
 StyleManager and, 286
 SystemManager class, 1181
 transaction managers for LCDS, 967–968

manifest file, specifying with COMPC, 404

manual invocation, of visual effects, 448

many-to-many relationships, DMS, 1012

many-to-one relationships, DMS, 1012

mapping

Flash library symbols to Flex styles, 500–501
 objects in Remoting Service, 1065–1067
 ORM (Object Relational Mapping), 916

maps. See associative arrays

Mark Occurrences button, code editing with Flex Builder, 172–173

masks, Drawing API, 476–477

Mate, Flex MVC frameworks, 1081

Maven, continuous integration with, 1322–1323

maximizable

AIR initial window configuration tags, 308
 properties for rendering windows, 771

maxSize tag, **AIR initial window configuration, 309**

measure()

measuring components, 327–328
 UIComponent class, 323

media

file formats, 513–514
 integrating media content, 515
 new APIs in Flash Player 10, 82–83
 overview of, 513
 progressive loading and streaming, 519
 sound. See sound
 summary, 543
 video. See video

Media Rights Management Server, Adobe, 514

memory management

comparing OOP languages, 1356
 Garbage Collection and, 591
 Heap View and, 181
 List, HorizontalList, and TileList
 classes and, 622

memory profiling

Allocation Trace View, 1281–1283
 detecting memory leaks, 1292–1296
 Live Object View, 1277
 Memory Snapshot View, 1278–1279
 Memory Usage graph, 1276–1277
 Object References View, 1279–1281
 overview of, 1270
 using memory profiler, 1276
 views requiring two memory snapshots,
 1283–1284

Memory Snapshot View, 1278–1279

**memory tuning, customizing Flex Builder 3,
 184–185**

Memory Usage graph, 1276–1277

MenuBar container, **268**

menus

alternatives for defining, 777–779
 context menus, 774–776

menus (continued)

- creating menu objects, 619–620
- Menu class as navigation component, 268–269
- native menu events, 776–777
- native menus, 774
- overview of Menu class, 619
- windows- and application-level menus, 780
- working with Menu class, 620

MesaMySQL, 949

message brokers, in LCDS server architecture, 958

message formats

- BlazeDS communication channels, 1055
- LCDS communication channels, 961

Message Service, BlazeDS, 1054

Message Service, LCDS

- applications, 990
- chat room application, 990–993
- configuring destinations on server, 981
- creating messages on client, 983–984
- declaring producers and consumers in MXML, 986–987
- detecting connectivity, 988
- expiration of messages, 982
- filtering messages, 987–988
- message channels, 980
- message queue processing, 982–983
- overview of, 960, 979
- PDF generator application, 997–1004
- producers and consumers, 980
- publishing from server, 988–990
- publish-subscribe paradigm, 979
- pushing messages from server, 980–981
- receiving messages, 985–986
- sending messages, 984–985
- stock price ticker application, 993–997
- subtopic, 981–982
- summary, 1004

MessageDetails items, 1034

metacharacters, encoding/decoding URL components and, 737

metadata

- adding to ActionScript and MXML files, 372–373
- [Bindable] metadata, 235–236, 377–379
- defaultProperty metadata tag, 350
- documenting with ASDoc, 1210–1211
- [Event] metadata, 242–243

- extending MXML components, 351–352
- ID3-based information for sound, 525–526
- loading image data from, 542–543
- maintaining data type information while dragging and dropping, 656
- setting up for MXML components, 348
- sound, 526–528
- video, 534–535

meta-policies, cross-domain security policies, 1166–1167

method polymorphism, OOP languages and, 1361

Method Statistics View, Performance Profiler, 1286–1287

methods, ActionScript, 59–61

Microphone class, 528

minimizable tag

- AIR initial window configuration tags, 308
- properties for rendering windows, 771

minSize tag, AIR initial window configuration, 309

mock objects

- collaborating class, 1319
- delegate interface, 1320
- overcoming dependencies, 1318–1319
- real vs. mock implementation, 1320
- unit tests and, 1321–1322

modal pop-ups, 289–290

model classes, Cairngorm, 1084

Model Locator Landfill anti-pattern, 1128–1129

Model Locator pattern, Cairngorm

- binding view to model, 1099–1100
- minimizing dependencies, 1100–1102
- modeling the application, 1098–1099
- overview of, 1096
- self-documenting code, 1098
- single point of access to model locator, 1097–1098

Model-Glue Flex, Flex MVC frameworks, 1081

ModelLocator class, 1077

models

- locating and updating in Cairngorm, 1090–1092
- in MVC framework, 1077–1078

modular application development

- building components and, 1173
- cross-module communication, 1177–1182
- cross-module data access, 1182–1187
- development enhancements in Flex Builder 3, 99

- development with Flex Builder and, 104
- Module class and ModuleLoader class, 1173–1177
- overview of, 1171–1172
- reasons for, 1172
- RSL (Runtime Shared Libraries) and, 1173
- summary, 1187
- Module **class, 1173–1177**
- ModuleLoader **class, 1173–1177**
- monitoring**
 - Internet traffic, 1257–1258
 - network connections with AIR, 785–786
- mouse events**
 - composite components and, 356–357
 - event priority and, 373–374
- Move **effect, built-in effects, 443**
- MovieStar Flash Player**
 - smart pause functionality, 873–874
 - updates for Flash Player, 514, 539
- MP3s**
 - accessing sound metadata, 526–528
 - embedding sound in MXML, 517
 - playing VOD (video on demand), 867
 - Sound class and, 520
 - streaming video and, 872
 - as usable media format, 514
- MP4s, 866–867**
- MPEG-4**
 - Flash Player supporting, 539
 - streaming video and, 865, 872
- multiple files, managing, 808–810**
- MVC (Model-View-Controller) framework.**
 - See also Cairngorm**
 - abstraction and, 1080
 - advantages of, 1079
 - as approach to component architecture, 1082
 - component reuse, 1080
 - components, 1077
 - controller, 1078–1079
 - distributed development, 1080–1081
 - maintenance and, 1081
 - models, 1077–1078
 - modular application development contrasted with, 1172
 - overview of, 1076–1077
 - summary, 1082
 - technologies for, 1081
 - views, 1078
- mx **prefix indicating Flex classes, 222**
- <mx:AddChild> **tag, 388–390**
- mx.binding. **See data binding**
- mx.containers. **See containers**
- mx.controls. **See controls**
- mx.effects. **See visual effects**
- mx.logging. **See logging**
- mx.managers. **See managers**
- MXML**
 - ActionScript and, 141, 147–148
 - ActionScript functions declared in, 145–146
 - benefits of Flex and, 12
 - breakpoints in, 1248
 - calling Web Services, 925
 - class naming conventions, 1143–1144
 - code assist and, 172
 - component tags, 133
 - core data type tags, 135
 - CSS and, 139
 - CSSStyle tag, 139–140
 - data binding and, 231–232
 - display hierarchy, 129
 - event listeners, 240
 - Flex application development and, 104
 - Flex based on, 3
 - as Flex source file, 218
 - form creation, 281
 - formatter classes and, 226
 - history of Flex and, 18
 - inline ActionScript, 141–143
 - inline CSS, 139–140
 - interpreted tags, 134–135
 - languages in Flex 3 ecosystem, 28
 - LCDS producers and consumers declared in, 986–987
 - linking external ActionScript, 146–147
 - linking external CSS, 140–141
 - local namespace, 132
 - metadata added to, 372–373
 - MXML tags, 133
 - namespaces, 129–130
 - namespaces and packages, 131–132
 - namespaces and URIs, 130–131
 - naming conventions, 128

MXML (continued)

- null namespace, 132–133
- overview of, 127
- parameter binding vs. explicit parameter passing in RPC, 838–839
- presentation models in, 1125
- properties, 135
- property attributes, 136–137
- property tags, 137–139
- `Repeaters` class and, 592
- RPC (remote procedure call) and, 838
- rules of usage for `ActionScript`, 147
- `Script` tag, 144–145
- sound embedded in, 517–519
- state management with MXML tags, 388–390
- summary, 148
- syntax, 128–129
- uses of XML, 571
- `WebService` component in, 841–844

MXML, custom components

- vs. `ActionScript` components, 343–344
- creating template components, 348–351
- extending, 351–352
- layout, 345
- overriding `layoutChrome()`, 346–348
- overriding `updateDisplayList()`, 345–346
- overview of, 343
- setting up metadata tags, 348
- summary, 352

.mxml file extension, 128

MXML View, Outline View options, 116

MXMLC compiler

- compilation in Flex Builder, 105
- as MXML component, 343
- packaging AIR applications, 1222
- settings, 151–153

MXML-only components, 134

- `<mx:RemoveChild>` tag, 389–390

- `mx.skins`. **See skins**

Mylyn, bug tracking and, 214

MySQL

- configuring for RESTful Web Services, 883, 885
- configuring for Routes Web Service, 900–901
- configuring for Web Service example, 916–917

N

name settings, LCDS project, 963–964

- `Namespace` class, XML, 577

namespaces

- `ActionScript`, 61–62
- naming conventions and, 1143
- XML, 573

namespaces, MXML

- local namespace, 132
- null namespace, 132–133
- overview of, 129–130
- packages and, 131–132
- URLs and, 130–131

naming conventions

- classes, 1143–1144
- events, 447
- MXML, 128
- namespaces and, 1143
- overview of, 1142
- variable and functions, 1144–1145

native desktop support, with AIR

- AIR capabilities and Flash runtime, 786–787
- alternatives for defining menus, 777–779
- context menus, 774–776
- creating native window, 768
- detecting user presence, 784–785
- determining AIR system capabilities, 780–781
- enabling full-screen mode, 787
- interacting with System Tray or Dock, 782–783
- monitoring network connections, 785–786
- native menu events, 776–777
- native menus, 774
- Native Windows Explorer sample application, 773–774
- `NativeWindow` class, 771
- overview of, 767
- properties for rendering windows, 768–771
- sending notifications, 784
- summary, 787
- `Window` class, 772
- windows- and application-level menus, 780
- working with windows, 767

native file dialogs, with AIR

- native browsing for directory, 821–822
- native browsing for Open dialog, 822–824

- native browsing for Open Multiple dialog, 824–825
- native browsing for Save dialog, 825–826
- overview of, 821
- native menus**
 - events, 776–777
 - overview of, 774
- native windows**
 - creating, 768
 - rendering, 768–771
- Native Windows Explorer sample application, 773–774**
- NativeApplication **class**, 780
- NativeDragManager **class**, 660
- NativeWindow **class**
 - checking AIR capabilities, 780
 - using, 771
- navigateToURL **function**, **ActionScript**, 706
- navigating projects, with Flex Builder**
 - file linkages, creating, 159–160
 - folder linkages, creating, 158–159
 - folder structure, creating, 157–158
 - “Go into” browsing option, 160–161
 - Link With Editor button, 164
 - Navigator filters, 163–164
 - overview of, 157
 - working sets, 161–163
- navigation**
 - design area, Flex Builder, 114–115
 - Flex Builder 3 features for, 85
 - paths, in local file management, 814–816
 - profiling views and, 1291
- navigation components**
 - Accordion container, 265
 - BarButton control, 269
 - containers. *See* navigation components
 - creationPolicy property, 264
 - LinkBar control, 267
 - Menu control, 268–269
 - MenuBar container, 268
 - overview of, 264
 - TabBar control, 267–268
 - TabNavigator container, 265–266
 - ToggleButtonBar control, 268–269
 - ViewStack container, 266–267
- navigational states**
 - deep linking and, 740–743
 - encoding, 741
 - Flex applications and, 732
 - handling changes in, 739
 - what it is, 741
- NavigationHistory **class**, **Flex filesystem**, 829
- Navigator filters, Flex Builder, 163–164**
- nesting**
 - components in Flex framework, 220–221
 - container classes, 224
 - effects, 453–454
 - XML and, 573
- .NET. *See* Web Services, with .NET**
- NetConnection **class**
 - creating custom connections to FMS, 868
 - video and, 528–529
- NetStream **class**
 - buffer control and, 873–874
 - FMS and, 872–873
 - incorporating data into media streams, 877–879
 - publishing and recording live streams, 876–877
 - publishing video streams, 874
 - streaming preexisting video, 875–876
 - video and, 529–530
- network packet sniffing, 846**
- networks**
 - local sandboxes and, 1168
 - monitoring network connections with AIR, 785–786
 - RIAs reducing bandwidth consumption, 4
- New Flex Project Wizard, in Flex Builder, 963**
- NIO end point, streaming AMF to, 1036**
- nosplash argument, Eclipse command-line start-up option, 186**
- notifications, sending with AIR, 784**
- null namespace, MXML, 132–133**
- NumberFormatter, **599–600**
- numbers**
 - ActionScript numeric types, 48
 - formatters, 599–600
 - input controls for numeric data, 250
 - internationalization and localization and, 690
 - validators, 602
- NumericStepper **control**
 - custom tooltip used in conjunction with, 286–288
 - input controls, 250

O

Object **class, ActionScript**

dynamic classes in ActionScript, 589
overview of, 58

Object **data type, comparing OOP languages, 1358–1359**

Object References View, Memory Profiler, 1279–1281

Object Relational Mapping (ORM), 916

objects

caching and recycling to improve application performance, 1199–1200
creating menu objects, 619–620
creation, in life cycle, 228
creation policies, 1189–1190
defining skins per object instance, 426
DMS managing object relationships, 1011–1012
instantiation and, 58
as keys, 590
local shared. See LSOs (local shared objects)
logging, 1260
mapping in Remoting Service, 1065–1067
typed vs. dynamic, 1148
as values in MXML, 138–139

objects, ActionScript

introspection, 55
Object class, 58, 589
overview of, 53

objects, XML

adding elements to, 580
deleting elements in, 580–581
modifying elements in, 580

Observe **tag, Adobe Consulting, 1261–1262**

ObserveValue **tag, Adobe Consulting, 1261–1262**

OLAP (online analytical processing), 640

OLAPDataGrid **control, 640–644**

applying aggregations to, 644
creating OLAP query for data, 643–644
creating OLAP schema for data, 642–643
OLAP primer, 640
working with, 640–641

OLTP (online transaction processing), 640

one-to-many relationships, DMS, 1012

one-to-one relationships, DMS, 1012

online analytical processing (OLAP), 640

online data, making available offline, 943–944

online transaction processing (OLTP), 640

OOP (object-oriented programming)

access modifiers, 1362
ActionScript and, 57
block-level scoping, 1356–1357
classes, filenames, and packages, 1360–1361
console output and, 1357
data types and, 1357–1359
declaring variables, 1359–1360
dynamic classes, 1362
encapsulation, 1149–1150
getter and setter methods, 1153–1157
global variable notation, 1357
inheritance, 1151–1152, 1362
interfaces, 1157–1158
memory management, 1356
overriding functions, 1152–1153
overview of, 1355
polymorphism, 1157–1158, 1361–1362
regular expressions, 1360
runtime environments, 1355–1356
statement termination, 1360
summary, 1363
type casting, 1359
variable scope, 1150–1151

Open dialog, native file dialogs, 822–824

Open Multiple dialog, native file dialogs, 824–825

open source

benefits of Flex, 13
Flex SDX, 196

open source project, SWFAddress, 748

OpenFlux, third party frameworks and components for Flex, 32

optimization

Flex 3 workbench and, 183–184
performance profiling and, 1268–1270
search engines, 747–748
state management, 394–395

ORM (Object Relational Mapping), 916

OSs (operating systems)

AIR for cross platform support, 299–300
deleting LSOs and, 798
drag and drop and, 647
native desktop support. See native desktop support, with AIR

native file dialogs. See native file dialogs, with AIR
 native menus, 774, 776–777
 native windows, 768–771

Outline View, Flex Builder

class outlines, 174–175
 other outline views, 175–176
 sort options, 175
 view options in, 116
 viewing display hierarchy with, 116

overhead, logging and, 1330–1331

overrides

defined, 383–384
 getter/setter methods and, 1157
 overriding functions in OOP, 1152–1153

P

.P12 file (PKCS12), 1218

Pacifica, 39

packages

ActionScript, 58–59
 comparing OOP languages, 1360–1361
 deploying. See application deployment, with AIR
 Flash Player APIs, 73
 mx prefix indicating Flex classes, 222
 MXML namespaces and, 131–132

paging, large data sets, 1007

pan

design mode enhancements in Flex Builder 3, 98
 navigating Design Area in Flex Builder, 115
 setting audio volume and pan, 525

Panel container

absolute positioning and, 415
 creating composite components, 355–356
 layout containers, 259–260

Papervision3D, 32

Parallel effects

nested effects, 453–454
 overview of, 451–452

@param tag, ASDoc, 1206

parameters

Cairngorm events as parameter objects, 1106–1107
 parameter binding vs. explicit parameter passing, 838–839
 parameterizing test applications, 1049

passing between JavaScript or ActionScript and Flex, 709–711

Parcel class, DMS Fleet tracker example, 1018–1019

parcels, in DMS Fleet tracker

assemblers, 1023–1024
 creating new parcel, 1026–1027
 destinations, 1021–1022
 filling collections with parcels and vehicles, 1024–1025
 filtering parcel list, 1026

parentheses (), in ActionScript syntax, 42

pattern matching. See regular expressions

Pause effect, built-in effects, 444

pause () method, controlling sound and, 524–525

PDF generator application, LCDS Message Service, 997–1004

configuring destinations, 999
 configuring JMS queue, 999
 creating Flex PDF generation client, 1002–1003
 creating generator, 1000–1001
 creating JMS queue listener, 1001–1002
 defining resource references, 1000
 overview of, 997–998
 preparing project for, 998
 summary, 1003

PDFs, generation with LCDS, 961

percentage, positioning by (layout strategies)

composite components and, 357
 overview of, 273–274

performance

application performance. See application performance
 layout strategies and, 280

performance profiling

clearing accumulated performance data, 1287
 Flex Profiler, 1268–1270
 optimizing algorithm with performance profiler, 1296–1300
 overview of, 1270
 taking performance snapshots, 1285–1286
 using performance profiler, 1284–1285
 viewing method statistics, 1286–1287

persistent framework caching, development enhancements in Flex Builder 3, 99

perspectives

- creating custom, 190
- debugging perspective in Flex Builder, 1246
- Subclipse, 194–195

perspectives, Flex Builder 3, 94–98

- debugging views, 96
- development views, 95
- optional views, 98
- overview of, 94
- profiling views, 97

.PFX files, 1218

phone numbers

- formatters, 598
- validators, 602

PhoneFormatter, 598

Photoshop, Flex Builder integration with CS3 suite, 99

PHP

- code editor for, 886
- Zend framework for PHP development, 881

pie charts, 678–679

Pixel Bender

- languages in Flex 3 ecosystem, 29–30
- new APIs in Flash Player 10, 79–81
- toolkit, 30, 458–459

play() method, controlling sound, 522–523, 525

playback, automation framework and, 1344–1345

plot charts, 680

policies

- cross-domain security policies, 1162–1164
- object creation policies, 1189–1190
- site-control entry and meta-policies, 1166–1167

polymorphism, 1157–1158, 1361–1362

PopUpButton control, 254–255

PopUpManager class

- displaying alerts, 291–293
- overview of, 288

PopUpMenuButton control, 255

pop-ups

- alerts, 289–291
- creating, 288
- custom events and, 371
- displaying more details with, 393
- interacting with, 293–294
- PopUpManager class for displaying alerts, 291–293

preferences

- Flex Profiler, 1290
- Local History, 191
- Subclipse, 213
- workspace, 190

prerequisites

- Data Services Stress Testing Framework, 1041
- file management, 803–804

presentation models

- command callbacks for locating, 1123–1124
- dispatching Cairngorm events with, 1123
- domain models and, 1124–1125
- Presentation Model pattern, 1119–1120
- simplicity of code in model views, 1122–1123
- using Cairngorm in conjunction with, 1120–1121

primitive types, ActionScript, 47

prioritizing events, 373–374

private, access modifiers in ActionScript, 61

private functions, variable scope and, 1151

@private tag, ASDoc, 1206

producers, LCDS

- chat room application, 992–993
- declaring in MXML, 986–987
- in LCDS Message Service, 980
- receiving messages and, 985
- sending messages and, 984–985

production code, removing logging statements from, 1341–1342

Professional Flex 3 book code, 196, 203, 205

profiling

- Flex Builder. See Flex Builder Profiler
- LCDS and, 1037
- memory profiling, 1270
- overview of, 1267–1268
- performance profiling and optimization, 1268–1270
- sampling profiler, 1270–1271

profiling perspective, Flex Builder Profiler, 1271–1272

profiling views, Flex Builder Profiler, 97

programmatically drawing, Drawing API, 462–463

programmatically interaction, charts and, 684–685

programmatically skins

- creating/applying, 472–475
- Drawing API for, 424
- overview of, 471–472

`programMenuFolder` **tag, AIR configuration, 309**

progress bar

- loading images and, 506
- ProgressBar control as static controls, 248

progressive media

- loading sound, 519
- playing streaming video, 866

`.project`, **project settings files, 150**

project management, with Flex Builder

- Flex Application settings, 150–151
- Flex Build Path settings, 151
- Flex Compiler settings, 151–153
- other project properties, 153–154
- overview of, 149
- project settings files, 150

projects

- creating, 108–111
- Flex applications built in context of, 91
- launching profiling from, 1272
- localization (L10n) settings, 692–693

projects, BlazeDS

- creating combined Flex/Java project, 1057–1058
- deploying to Tomcat, 1059
- overview of, 1056

projects, Cairngorm

- large, 1126–1127
- small, 1125–1126

projects, Flex Builder

- configuring project properties, 149
- creating, 107–111
- creating ActionScript projects, 120–123
- development best practices, 179–181
- exporting, 154
- importing, 154–157
- project settings files, 150
- types of projects you can create in Flex Builder, 106

projects, LCDS

- build path, 965–966
- J2EE settings, 964–965
- name and server technology settings, 963–964

projects, Subclipse

- creating from repository, 203–210
- disconnecting local projects, 210–211
- managing multiple repositories, 210
- sharing projects in repository, 197–202

properties

- `DataService` class, 1012–1013
- documenting with ASDoc, 1208–1209
- localization (L10n) and, 690–691
- LSOs (local shared objects) and, 797
- passing data to composite components, 355–356
- project properties in Flex Builder, 149, 153–154
- spatial properties, 76–77
- state management and, 384, 390
- style properties, 424
- [SWF] metadata tags for, 124–125
- system capabilities and, 755–757
- validation properties, 604
- window, 768–771

properties, ActionScript

- applying, 326–327
- overview of, 59–61, 323–324
- static properties, 63
- steps in writing components, 319–320

properties, MXML

- overview of, 135
- property attributes, 136–137
- property tags, 137–139

Properties views, Flex Builder, 112–113

Property Inspector, Design View, 286

property tags, as anonymous ActionScript function, 145

`protected` **access modifiers, in ActionScript, 61**

protected functions

- documenting with ASDoc, 1207
- variable scope and, 1151

protocols

- choosing protocol server, 850
- communication protocols, 32–33
- security protocols, 5
- transfer protocols (BlazeDS), 1055
- transfer protocols (LCDS), 962

Proxy Service, BlazeDS

- configuring destinations, 1068–1069
- connecting Flex clients via, 1069–1071
- overview of, 1054
- requesting data from, 1071

Proxy Service, LCDS

- configuring for stress testing, 1046
- overview of, 959–960

`public` **access modifier, in ActionScript, 61**

public functions

- documenting with ASDoc, 1207
- variable scope and, 1151
- public properties, component class and, 324**
- public variables, getter/setter methods compared with, 1154**
- publishing**
 - process in Flex Builder, 106
 - from servers, LCDS Message Service, 988–990
- publishing, FMS**
 - recording live streams and, 876–877
 - video, 874–875
- publish-subscribe paradigm, in LCDS Message Service, 979**
- Punkhead and Ticklehead examples**
 - adding animation states, 560–561
 - adding transition states, 561–563
 - building Punkhead Flex application, 564–566
 - building Ticklehead Flex application, 566–567
 - creating animations, 559
 - creating integrated components, 563
 - overview of, 559
- PureMVC, 1081**
- PyAMF, third party solutions for RPC services, 851**

Q

- QName class, XML, 577**
- QTP (Quick Test Professional), from HP, 1344**
- queries, creating OLAP query for data, 643–644**
- queues, message queue processing in LCDS, 982–983**
- Quick Test Professional (QTP), from HP, 1344**

R

- RAD (rapid application development), 15**
- RadioButton control, input controls, 252–253**
- RadioButtonGroup control, input controls, 252–253**
- rapid application development (RAD), 15**
- reading/writing data. See also CRUD (create, read, update, and delete)**
 - to clipboard, 658–659
 - from clipboard, 659
 - custom data from clipboard, 660
 - getter/setter methods and, 1154–1155

- overview of, 826
- reading XML data from a file, 828
- saving bitmap data as JPEG file, 827–828
- Real-Time Media Flow Protocol (RTMFP), 33**
- Real-Time Messaging Protocol (RTMP), 32–33**
- receiver, LocalConnection class, 790–791**
- recording, automation framework, 1344–1345**
- recording test scripts**
 - overview of, 1349–1352
 - replaying test scripts, 1352–1353
 - stopping recording, 1352
- recycling objects, 1199–1200**
- Red 5, third party solutions for RPC services, 850**
- redistribution license, 1226**
- refresh argument, workbench optimization in Eclipse, 183–184**
- RegExpValidator, 608**
- registerClassAlias()**
 - serializing custom classes, 799
 - using with LocalConnection class, 800–801
 - using with shared objects, 800–801
- regular expressions**
 - comparing OOP languages, 1360
 - overview of, 595, 607
 - primer for working with, 607–608
 - RegExpValidator, 608
 - summary, 612
 - validators, 602
 - when to use, 607
- relationships, DMS managing object relationships, 1011–1012**
- release builds, 165–167**
- remote method invocation (RMI), ColdFusion and, 974**
- remote methods calls, BlazeDS, 1060–1063**
- remote objects. See also BlazeDS (Blaze Data Services)**
 - choosing protocol server for, 850
 - coding remote Java objects, 1064
 - communication with remote servers, 837
 - connecting to remote data and, 836–837
 - creating instance of in BlazeDS, 1060
 - LCDS interaction with, 957
 - logging stress tests and, 1046
 - RPC libraries and, 837
 - third party solutions for RPC services, 851

remote procedure call. See **RPC (remote procedure call)**

Remoting Service, BlazeDS

- configuring, 1063–1064
- creating `RemoteObject` instance, 1060
- object mapping in, 1065–1067
- overview of, 1054

Remoting Service, LCDS, 959

rendering, cross-rendering capabilities of AIR, 26

repeated attacks, overcoming item-pending errors, 1032–1033

Repeaters class

- after execution, 593
- data. See `Repeaters` class
- during execution, 592–593
- handling interactions, 593–594
- overview of, 592
- when not to use, 594

repositories. See **code repositories**

reserved keywords

- ActionScript, 43
- MXML, 128

resizable tag

- AIR initial window configuration, 308
- properties for rendering windows, 771

Resize effect, built-in effects, 444

@Resource, internationalization (i18n), 695

resource bundles

- development enhancements in Flex Builder 3, 99
- generating programmatically, 699
- internationalization (i18n) and, 695–696

resource modules, 698–699

ResourceManager class, 696–698

- accessing data with, 696–697
- overview of, 696
- switching locales at runtime, 697–698

resources, Flex projects, 91

resources, in internationalization

- @Resource, 695
- loading at runtime, 698
- ResourceBundle, 695–696
- ResourceManager, 696–698
- ways to utilize, 694–695

RESTful Web Services

- Cairngorm service locators and, 1115
- client software requirements, 886

creating and updating data with, 894–896

deleting data from, 896

`HTTPService` declaration, 889

managing errors, 893–894

overview of, 881

processing request results, 891–893

requesting all routes, 889–891

retrieving data from, 889

routes sample, 882–883

server software requirements, 883–885

setting up server and development environment for, 883

summary, 896

what it is, 881–882

Zend and, 886–889

resume() method, controlling sound, 524–525

@return tag, ASDoc, 1206

return values, passing between JavaScript or ActionScript and Flex, 712

reuse

- code reuse, 545
- component reuse, 1080

reverse binding, data binding, 233

revertChanges(), DataService class, 1010

RIAs (Rich Internet Applications)

- creating with Flash IDE, 37
- current RIA technologies competing with Flex, 6–7
- development workflows and, 481
- history of Flex and, 18
- what they are, 4–6

RIATest automation tool

- checkpoints, 1350–1352
- compiling for automation, 1348–1349
- creating RIATest project, 1349
- creating test script and starting recording, 1350
- installing as automation tool, 1347
- replaying test scripts, 1352–1353
- stopping recording, 1352

Rich Internet Applications. See RIAs (Rich Internet Applications)

rich media, 4

RichTextEditor control

- child components, 365
- as example of composite components, 353–354
- input controls, 254

right-click context menus. See context menus

RMI (remote method invocation), ColdFusion and, 974

rocker switch example, ActionScript components, 333–340

`Rotate` **effect, built-in effects, 444–445**

route DTO vs. route persistence, in Web Services, 919

`Route.groovy` **domain class, 903**

routes

example of RESTful Web Services, 882–883
requesting all routes in RESTful Web Services, 889–891

Routes Web Services

Java and. See Web Services, with Java
.NET and. See Web Services, with .NET

`RouteService.as` **class, 906**

`RouteService.asmx` **class, 919–920**

`RouteService.groovy` **class, 904**

RPC (remote procedure call), 911–912

AMF (Action Message Format) and, 850–851
choosing protocol server for, 850
debug tracing, 846
debugging, 845
error handling and, 860
`HTTPService` component, 839–840
`IResponder` interface and, 906–909
logging calls to web services, 848–850
logging `HTTPService` requests, 846–847
logging `WebService` initialization, 847–848
MXML vs. ActionScript classes, 838
`mx.rpc` libraries, 837
network packet sniffing, 846
overview of, 835–837
parameter binding vs. explicit parameter passing, 838–839
process of connecting to remote data, 836
RESTful Web Services. See RESTful Web Services summary, 851
Web Service with Java. See Web Services, with Java
Web Service with .NET. See Web Services, with .NET
`WebService` component, 841
`WebService` in ActionScript, 844–845
`WebService` in MXML, 841–844

RSLs (Runtime Shared Libraries)

Flash/Flex integration and, 545
importing custom Flex classes into Flash, 546
loading external assets, 511
modular application development and, 1173
overview of, 9
publishing in Flex Builder and, 106

RTMFP (Real-Time Media Flow Protocol), 33

RTMP (Real-Time Messaging Protocol), 32–33

Ruby on Rails, Grails compared with, 899

Run menu, launching profiling from, 1272

runtime

AIR runtime. See AIR (Adobe Integrated Runtime)
AIR user runtime requirements, 301
changing skin styles at, 430–433
charts and, 684–685
comparing OOP languages, 1355–1356
vs. compile time automation, 1345–1346
development workflows and, 482
Flash Player runtime. See Flash Player
Flex 3 ecosystem, 19
handling event runtime errors, 373
loading resources at, 698
localization, 99
managing breakpoints at, 1248–1249
switching locales at runtime, 697–698

Runtime Shared Libraries. See RSLs (Runtime Shared Libraries)

S

SaaS (software as a service), 6

sampling profiler, 1270–1271

sandboxes

interacting between, 723–724
working with application sandboxes, 722

sandboxes, in Flash Player security model

`allow-access-from`, 1164–1165
`allow-http-request-headers-from`, 1165–1166
`crossdomain.xml`, 1162–1164
local sandboxes, 1167
`local-trusted` sandbox, 1169
`local-with-filesystem` sandbox, 1167–1168
`local-with-networking` sandbox, 1168

- overview of, 1161–1162
- site-control entry and meta-policies, 1166–1167
- Save dialog, native file dialogs, 825–826**
- scale-9 grids**
 - scaling skins, 429–430
 - tips for, 497
- scaling skins, 428–430**
- SchemaTypeManager, **serialization with, 909–910**
- Script **tag, ActionScript in MXML and, 144–145**
- SDK. See Adobe Flex Software Development Kit (SDK)**
- search engines, indexing and optimization, 747–748**
- security**
 - error handling and, 862
 - Flash-based applications and, 9
 - FMS and, 866
 - LCDS and, 1037
 - protocols, 5
 - sandboxes. See sandboxes
- security model, Flash Player**
 - allow-access-from, 1164–1165
 - allow-http-request-headers-from, 1165–1166
 - crossdomain.xml, 1162–1164
 - local sandboxes, 1167
 - local-trusted sandbox, 1169
 - local-with-filesystem sandbox, 1167–1168
 - local-with-networking sandbox, 1168
 - overview of, 1161–1162
 - sandboxes, 1161–1162
 - site-control entry and meta-policies, 1166–1167
 - summary, 1169
- @see **tag, ASDoc, 1206**
- selection APIs, ListBase class, 614–616**
- selectors, filtering LCDS messages by, 987**
- self-signed certificates, 1220–1221**
- sender**, LocalConnection **class, 791–792**
- Sequence **effects**
 - nested effects, 453–454
 - overview of, 452–453
- serial number screen, ColdFusion, 973**
- serialization, with SchemaTypeManager, 909–910**
- servers, Flex 3 ecosystem, 34–35**
- servers, LCDS**
 - architecture, 958–959
 - configuring destinations on Message Service, 981
 - publishing from Message Service, 988–990
 - pushing messages from Message Service, 980–981
 - technology settings, 963–964
- servers, RESTful Web Services**
 - setting up, 883
 - software requirements, 883–885
- servers, Routes Web Service**
 - calling Web Services from other servers, 912–913, 926–927
 - software requirements for Routes Web Service, with Java, 900–902
 - software requirements for Routes Web Service, with .NET, 916–917
- Servers View, in Web Tools Platform, 1256**
- server-side**
 - BlazeDS logging, 1059–1060
 - file management considerations, 810
 - LCDS logging, 969
- server-side code**
 - debugging Java code, 1255–1257
 - file uploads and, 807
- server-side DMS**
 - client/server collaboration, 1015
 - interfaces, 1014
 - server components, 1010–1011
- service locators, Cairngorm, 1113–1115**
 - creating, 1113–1114
 - locating, 1114–1115
 - overview of, 1113
 - RESTful services and, 1115
- service locators, stub implementations of, 1135**
- service-oriented architecture (SOA), 900**
- services**
 - BlazeDS, 1054
 - impact of multiple invocations on application performance, 1191–1192
 - locating and invoking in Cairngorm, 1089–1090
- services, LCDS**
 - Data Management Service, 960–961
 - Message Service, 960

- overview of, 958–959
- Proxy Service, 959–960
- Remoting Service, 959
- ServiceCapture, third-party debugging tools, 1257**
- setters. See getters/setters**
- shapes, drawing, 465–467**
- sharing data**
 - cross-module data access and, 1182–1187
 - managing in DMS, 1033
 - reasons for using DMS, 1006–1007
- ShockWave Flash. See SWFs**
- shortcuts, Flex Builder**
 - customizing, 188–189
 - overview of, 176–177
- Show Redraw Regions, context menu option, 760**
- Show Zero Time Methods, Flex Profiler, 1290–1291**
- `showBusyCursor` property, **MXML RPC classes and, 838**
- `showCapabilities()` method, **754**
- `-showlocation` argument, **workbench optimization and, 184**
- signed applications. See code signing**
- silent installations, application deployment, 1226–1227**
- simple types, comparing OOP languages, 1357**
- singletons, 1089**
- `site-control` entry, **1166–1167**
- skins, 423–433**
 - applying, 424–425
 - building custom Flex component, 492–494
 - building Flash skin for custom Flex component, 494–498
 - changing styles at runtime, 430–433
 - defining in style sheets, 426–428
 - defining per object, 426
 - embedded assets and, 423–424
 - planning custom Flex component, 490–492
 - programmatic, 424
 - scaling, 428–430
 - skinning custom Flex component, 490, 498–500
 - testing custom Flex component, 501
 - what they are, 423
- smart pause functionality, in FMS and MovieStar Player, 873–874**

- SOA (service-oriented architecture), 900**
- SOAP**
 - enabling Flex to receive SOAP faults, 905, 920–921
 - exchanging SOAP messages with XML, 900
 - serialization with `SchemaTypeManager`, 909–910
 - `strictNullability` property, 911–912
 - uses of XML, 571
 - Web services and. *See* `WebService` class
- SOAPProxyAdapter, BlazeDS integration points, 1055**
- soapUI utility, working with Web Services, 902**
- social security, validators, 602**
- software**
 - modular approach to development, 1171
 - RESTful Web Services servers requirements, 883–885
 - RIAs acting like, 5
 - writing testable, 1304
- software as a service (SaaS), 6**
- solid fills, Drawing API, 467–468**
- sort options, Outline View, 175**
- sorting data, Collections API and, 582–583**
- sound, 515–528**
 - controlling, 522–525
 - embedding in ActionScript, 515–517
 - embedding in MXML, 517–519
 - ID3 information for `Sound` class, 525–526
 - loading audio, 522
 - metadata, 526–528
 - `Microphone` class, 528
 - progressive loading and streaming, 519
 - setting `SoundTransforms`, 525
 - `Sound` class, 519–520
 - `SoundLoaderContext` class, 520–521
 - `SoundMixer` class, 528
- `Sound` class
 - Flash, 519–520
 - ID3 information for, 525–526
- `SoundEffect` **effect, built-in effects, 444–445**
- `SoundLoaderContext` **class, 520–521**
- `SoundMixer` **class, 528**
- `SoundTransforms`
 - `NetStream` class and, 530
 - setting volume and pan, 525

source code, viewing with Flex Profiler, 1291

source directories, sharing library source directories, 404–405

Source Path, Flex Build Path settings, 151

source version control. See Subclipse

Source View, Flex Builder

enabling, 167–170

overview of, 385–386

using custom source view URL, 170

sources, for library content, 400

space management, state management and, 393

spatial properties, display methods, 76–77

special types, ActionScript data types, 48

Sprite class, ActionScript display architecture and, 68–69

Spry Framework, Adobe, 31

SQL assembler, for DMS, 1015

SQLite

AIR and, 25, 299

created table based on XML data, 945–947

creating data, 936–938

creating database, 931

deleting data, 940–941

importing XML into, 944–945

improving XML to SQLite importer, 948–949

inserting XML data into database, 947–948

making online data available offline, 943–944

opening database asynchronously, 931–932

overview of, 929–930

retrieving data, 938–939

summary, 949

synchronous vs. asynchronous, 930–931

third-party management applications for SQLite, 949

transactions for importing data, 941–943

updating data, 939–940

working with data, 935

working with DDL (Data Definition Language), 933–935

XML vs., 930

SQLiteMan, 949

Stage class

ActionScript display lists and, 68

full-screen mode and, 757

standards, coding. See coding standards

startup options, Eclipse command-line

-clean argument, 183

-nosplash argument, 186

-refresh argument, 183–184

-showlocation argument, 184

-vm argument, 186

-vmargs argument, 184–185

startup options, Flex Builder command-line, 182–183

state management

adding components, 388–389

changing event handlers, 390–391

changing states, 387

creating states in ActionScript, 391–393

data binding and, 391

Design View for creating states, 384–387

examining application state, 1252

Expressions View of application state, 1254–1255

MXML tags and, 388

optimization by wrapping common state change

elements in container, 394–395

overriding controls, but not destroying, 389–390

overview of, 383

removing components, 389

setting/changing properties and styles, 390

space management and, 393

States property, 384

states vs. ViewStack, 394

summary, 396

testing application state change, 387–388

transitions, 395–396, 454–457

Variables View of application state, 1253–1254

what states are, 383–384

statement termination, comparing OOP languages, 1360

states

adding animation states, 560–561

adding custom states, 557–558

custom states in Flex, 556–557

States property, 384

States View, 385–386

static controls

HRule and VRule controls, 247–248

Image control, 246

Label and Text controls, 245–246

static controls *(continued)*

- overview of, 245
- ProgressBar control, 248
- SWFLoader control, 247
- static events, cross-module communication and, 1181–1182**
- static members, code syntax, 1147**
- static properties, ActionScript, 63**
- stepping through code, Flex Builder, 1250–1252**
 - overview of, 1250
 - Step In (F5), 1250–1251
 - Step Out (F7), 1252
 - Step Over (F6), 1251–1252
- stickiness, of applications, 437**
- stock price ticker application, LCDS Message Service, 993–997**
 - Java stock feed, 994–995
 - overview of, 993–994
 - stock feed destination, 995–996
 - stock price model, 996–997
 - summary, 997
- stop() method, controlling sound, 524–525**
- stopping events, 238–239**
- stopwatch logger, 1337–1339**
- Stratus, 39**
- streaming media**
 - incorporating data into streams, 877–879
 - preexisting video, 875–876
 - publishing and recording live streams, 876–877
 - sound, 519
 - video, 865–866
- stress testing. See Data Services Stress Testing Framework**
- strictNullability property, Routes Web Service, 911–912**
- strings**
 - ActionScript, 49–50
 - as keys, 589
 - validators, 602
- strongly typed objects, 1148**
- stubbing the service layer**
 - business delegates and, 1112–1113
 - Cairngorm and, 1135–1136
- style properties, applying skins as, 424**
- style sheets. See CSS (Cascading Style Sheets)**
- style tag, CSS in MXML, 140**

- StyleManager, **tooltip styles and, 286**
- styles, 411–422**
 - charts, 686
 - components, 413–415
 - constraint-based layout and, 415–418
 - CSS, 411
 - inheritance and, 418–419
 - inline vs. styles, 411–413
 - mapping Flash library symbols to Flex styles, 500–501
 - setting/changing for state management, 390
 - skins, 430–433
 - subcomponents and, 419–421
 - summary, 422
- subclasses, ActionScript, 64**
- Subclipse**
 - bug tracking and, 214
 - creating project from repository, 203–210
 - disconnecting local projects, 210–211
 - establishing repository for, 195–196
 - installing, 193–194
 - migrating preferences, 213
 - overview of, 192–193
 - perspectives, 194–195
 - project management for multiple repositories, 210
 - settings, 212–213
 - sharing projects in repository, 197–202
 - working with, 211–212
- subcomponents**
 - ColdFusion selection screen and, 972
 - CSS styles and, 419–421
- subtopics, LCDS**
 - filtering messages by, 987–988
 - in Message Service, 981–982
- Subversion, 192. See also Subclipse**
- summary-to-details modeling, DMS**
 - overview of, 1030
 - replacing lazy relationships with `getItem()` or `fill()`, 1030–1031
- superclasses, inheritance and, 64**
- support community, Flex, 13**
- SVN. See Subversion**
- SWCs. See also libraries**
 - ActionScript, 397
 - adding, 406–407
 - adding to `libs` folder, 405–406

creating, 400–402
 development of Flex applications and, 104
 Flash/Flex integration and, 545
 importing custom Flex classes into Flash, 546
 overview of, 399
 RPC libraries distributed as, 837
 skins and, 424
 themes and, 435
 using, 407

[SWF] metadata tags, for configuring compiler settings, 123–125

SWFAddress, open source project, 748

SWFLoader class, 247, 506–507

swfobject.js, 726

SWFs

- comparing file sizes, 123
- control for displaying, 247
- debug releases, 164–165
- Flash and, 7–8
- frame-by-frame execution, 218–220
- integrating media content, 515
- loading, 506–507
- managing external content, 1199
- role of Flash in Flex applications, 218
- skins and, 424
- SWCs compared with, 399
- SWF to SWF communication, 790

switch statements, ActionScript, 45–46

Swiz, Flex MVC frameworks, 1081

synchronous API, AIR

- file operations, 816–817
- opening database synchronously, 933
- SQLite database operations, 930–931

synchronous commands, Cairngorm, 1108–1109

synchronous error handling

- custom error types, 858–859
- exception bubbling, 856–858
- Flash Debug Player vs. Release Player, 853–855
- in functions, 860
- overview of, 853
- throw, 859–860
- try/catch/finally statements, 855–856

syntax. See code syntax

system

- access in AIR, 26
- adding/removing context menu options, 760–763

- detecting client system capabilities, 751–755
- determining AIR system capabilities, 780–781
- Flash Player 10 system integration APIs, 82
- full-screen display and, 757–760
- invoking actions from context menus, 763–765
- limitations of custom context menus, 766
- limitations of full-screen mode, 765–766
- overview of, 751
- summary, 766
- system capabilities properties, 755–757

System Tray (Windows OSs), 782–783

systemChrome tag

- AIR initial window configuration, 308
- properties for rendering windows, 768–769

SystemManager class, 1181

SystemTree class, 829

T

TabBar control, as navigation components, 267–268

tables

- created table based on XML data, 945–947
- DDL (Data Definition Language), 934–935

TabNavigator container, as navigation components, 265–266

tabular data. See DataGrid control

TagCloud component, extending Flow container, 363–365

tags

- ASDoc, 1206
- in Flex Form container, 283

tags, MXML

- component tags, 133
- core data type tags, 135
- interpreted tags, 134–135
- overview of, 133
- property tags, 137–139

Tamarin, 24

target nodes, event flow and, 71

targets

- creating logging target, 1328
- creating remote logging target, 1331–1332
- event flow and, 72, 237

TDD (Test-Driven Development)

- assembling test suite, 1308–1309
- assertion functions, 1311

TDD (continued)

- BDD (Behavior-Driven Development) compared with, 1324
- creating test case, 1308
- failure messages, 1311–1312
- fleshing out test case, 1312–1313
- happy and unhappy paths, 1313
- overview of, 1303, 1307
- preparing for unit testing, 1307
- setting up/ tearing down test cases, 1315–1317
- stages of unit testing, 1310
- strengthening tests, 1314–1315
- test failure, 1309–1310

technologies

- in Flex 3 ecosystem, 17
- MVC, 1081

template components, MXML, 348–351

- creating custom children, 350–351
- defaultProperty metadata tag, 350
- overview of, 348–349

templates, ASDoc, 1213

temporary files and directories, 820–821

ternary operators, code syntax, 1146

Test Administrator, DSS

- communication channel for test results, 1044–1045
- compiling, 1047
- overview of, 1040
- preconfiguring, 1049–1050
- running stress test, 1047–1049
- stopping stress tests, 1043–1044

test applications

- Data Services Stress Testing Framework and, 1040
- writing, 1041–1042

test cases

- creating, 1308
- eventful, 1323
- fleshing out, 1312–1313
- in FlexUnit, 1306
- happy and unhappy paths, 1313
- setting up/ tearing down, 1315–1317

test coverage, 1323–1324

test fixtures, 1315

test scripts

- recording, 1349–1352

- replaying, 1352–1353
- stopping recording, 1352

test suites

- assembling, 1308–1309
- in FlexUnit, 1306
- organizing, 1318
- strengthening code in, 1314–1315

Test-Driven Development. See TDD (Test-Driven Development)

testing

- application state change, 387–388
- automation framework. See automation framework
- Cairngorm and, 1119
- custom Flex component, 501
- preventing bugs by, 1304
- reasons for, 1304–1305
- stress testing. See Data Services Stress Testing Framework

text

- controls for, 245–246
- input controls for, 249–250

Text control, 245–246

text editors

- control for rich text editor, 254
- Flex Builder 3, 85

text properties, MXML, 136

TextArea control

- composite components and, 353
- input controls, 249–250

TextInput control, 359–360

Thawte

- as CA (certificate authority), 1216
- exporting Thawte certificate from Firefox, 1217–1218
- purchasing certificates, 1216–1217

themes

- applying, 433–434
- creating, 435
- default Flex 3, 433
- overview of, 433
- SWC files and, 435

third-party solutions

- debugging tools, 1257–1258
- Eclipse as, 86
- frameworks and components for Flex, 31

- management applications for SQLite, 949
 - for RPC services, 850–851
 - throw **statement, error handling, 859–860**
 - ThunderBolt logger, for Firebug, 1339–1341**
 - Tile container**
 - creating custom containers, 361–362
 - layout containers, 261–262
 - TileBase class**
 - HorizontalList, 621–622
 - overview of, 621
 - TileList, 621–622
 - TileList component**
 - adding effects to data containers, 448
 - overview of, 621–622
 - title tag, AIR initial window configuration, 307**
 - TitleWindow container**
 - layout containers, 260
 - pop-ups and, 291
 - ToggleButtonBar control, as navigation component, 268–269**
 - Tomcat 5.5**
 - deploying BlazeDS project to, 1059
 - J2EE settings, 964
 - JOTM (Java Open Transaction Manager) and, 967–968
 - tooltips**
 - chart data tips, 682–683
 - creating, 286–288
 - overview of, 286
 - top-level classes, ActionScript 3.0, 68**
 - trace**
 - debug tracing in RPC, 846
 - logging compared with, 1327
 - removing trace statements from production code, 1341–1342
 - TraceTarget class, 1328**
 - transaction managers, for LCDS, 967–968**
 - transactions, for importing SQLite data, 941–943**
 - transfer protocols**
 - BlazeDS, 1055
 - LCDS, 962
 - transitions**
 - adding, 558
 - adding transition states, 561–563
 - custom transitions in Flex, 556–557
 - state management and, 395–396
 - visual effects and, 454–457
 - transparency**
 - properties for rendering windows, 770
 - skins and, 423
 - transparent tag, AIR initial window configuration, 308**
 - Tree control, 624–628**
 - assigning data to, 624–625
 - changing look of, 625–626
 - expanding collapsing branches, 626–627
 - handling tree events, 627–628
 - overview of, 624
 - triggers**
 - applying effects with, 437
 - applying visual effects, 447
 - Flex Builder and, 118–119
 - validation, 606–607
 - video cue points, 538
 - troubleshooting LCDS, 968–969**
 - trusted resources, local sandboxes and, 1169**
 - try/catch/finally statements, 855–856**
 - turnkey distribution, BlazeDS, 1056–1057**
 - type casting, comparing OOP languages, 1359**
 - type declaration**
 - ActionScript, 62
 - MXML, 136
 - type property, rendering windows and, 769–770**
 - typeof operator, introspection in ActionScript, 56**
- ## U
- UI controls. See controls**
 - UIComponent class**
 - ActionScript component extending, 320–321
 - class outlines, 174–175
 - classes/subclasses extended from, 223–224
 - extending, 360–361
 - identifying components for automation, 1346
 - interfaces implemented by, 223
 - overview of, 222–223
 - visual components inheriting from, 133
 - visual objects and, 462
 - unbound commands, 188**
 - undeclared types, comparing OOP languages, 1358**
 - underscore symbol (_), MXML naming conventions and, 128**
 - undocumented [SWF] metadata tags, for configuring compiler settings, 123–125**

Unicode characters, 737

Uniform Resource Locators. **See** URLs (Uniform Resource Locators)

unit tests. **See also** FlexUnit

- in FlexUnit, 1306
- overview of, 1303
- preparing for, 1307
- stages of, 1310

Universal Resource Indicator (URIs), 130–131

UPDATE, **SQLite data transactions**, 941

update APIs, `ListBase` class, 614–616

update operations. **See** CRUD (create, read, update, and delete)

- `updateDisplayList()`
 - overriding when creating custom MXML components, 345–346
 - rendering Flash components on-screen, 462
- `UIComponent` class, 323

upgrades, application deployment and, 1227–1228

uploading files

- canceling, 808
- client-side code and, 806–807
- multi-file upload, 809
- overview of, 805
- server-side code and, 807
- server-side considerations, 810

URIs (Universal Resource Indicator), 130–131

UIKit, 748–749

URLs (Uniform Resource Locators)

- deep linking and, 732–733
- encoding/decoding URL components, 737–738
- fragment identifiers, 733
- handling changes in browser's URL, 739
- linking to external page URL, 706
- loading external content and, 507
- using custom source view URL, 170

user gestures

- automation framework and, 1343
- Cairngorm interactions beginning with, 1084–1085

user input. **See** input controls

user interface

- containers. **See** containers
- controls. **See** controls

users

- AIR user runtime requirements, 301

charts and user interaction, 684

creating/delivering user messages, 1034–1035

detecting user presence with AIR, 784–785

managing user-specific data in DMS, 1033

UTF-8, 691

V

validation properties, 604

`Validator` class, 227

`Validator` class, 601

validators

- binding controls to, 604
- centralizing, 610–612
- changing default error messages, 605–606
- classes/subclasses of, 227
- overview of, 595, 601–602
- `RegExpValidator`, 608
- setting validation parameters, 604
- standard, 602–604
- summary, 612
- triggering validation, 606–607

`Validator.validateAll()`, 606–607

value objects, Cairngorm, 1115–1118

- creating, 1116–1118
- overview of, 1115–1116
- sending/receiving, 1118

value types, ActionScript, 47–48

values

- calculating with getter/setter methods, 1157
- data-checking with getters/setters, 1156

variable scope, OOP and, 1150–1151

variables

- ActionScript, 44
- declaring in OOP languages, 1359–1360
- getter/setters allowing overriding in child classes, 1157
- getters/setters treated as, 1155–1156
- global variable notation, 1357
- naming conventions, 1144–1145
- public variables vs. getters/setters, 1154

Variables menu, 1254

Variables View, 1253–1254

`VBox` container, layout containers, 257–258

`VDividedBox` container, layout containers, 258–259

vector graphics. See also Drawing API, Flash Player

- ActionScript, 52–53

- overview of, 461

Vehicle class, DMS Fleet tracker example, 1017–1018**vehicles, in DMS Fleet tracker**

- assemblers, 1022–1023

- destinations, 1019–1021

- filling collections with parcels and vehicles, 1024–1025

- visualizing movements of, 1025–1026

VeriSign, 1216**version control. See Subclipse****version tag, AIR application configuration, 307****vertical lines, control for, 247–248****vertical sliders, 253****video**

- Camera class, 538–539

- classes and components, 528–530

- controlling, 535–536

- cue points, 538

- Flash data formats, 33

- Flash Player MovieStar update, 514, 539

- full-screen display, 540–542

- loading image data from metadata, 542–543

- loading video, 530–534

- metadata, 534–535

- overview of, 528

- playing VOD (video on demand), 866–867

- progressive loading and streaming, 519

- publishing, 874–875

- recording live streams, 876–877

- streaming, 865–866

- undocumented video classes, 539

- VideoDisplay component, 536–538

video on demand (VOD), 866–867**VideoDisplay component, 536–538**

- controlling video, 537–538

- vs. FLVPlayback (Flash), 551–552

- loading video, 536

- overview of, 536

- video cue points, 538

VideoPlayer class, 551–552**view classes, Cairngorm, 1084****view cursors, IViewCursor, 584–585****views**

- Flex Builder 3, 93–94

- in MVC framework, 1078

- rejecting changes in Cairngorm, 1092–1094

- simplicity of code in presentation model views, 1122–1123

ViewStack container

- navigation components, 266–267

- states vs., 394

ViewState, adjusting view to correspond to navigational state, 742**visible tag, AIR initial window configuration, 308****visual effects**

- 3D effects, 458

- AnimateProperty effect, 440–441

- applying, 446

- Blur effect, 441

- built-in, 440

- composite effects, 450

- data effects, 448–449

- Dissolve effect, 441–442

- easing functions, 449–450

- event triggers for applying, 447

- events, 457–458

- Fade effect, 442

- Flex 4 effects, 458–459

- Glow effect, 442–443

- Iris effect, 443

- manually applying, 448

- Move effect, 443

- nested effects, 453–454

- overview of, 437

- Parallel effects, 451–452

- Pause effect, 444

- Resize effect, 444

- Rotate effect, 444–445

- Sequence effects, 452–453

- SoundEffect effect, 444–445

- summary, 459

- transitions and, 454–457

- what they are, 437–439

- WipeLeft/WipeRight/WipeUp/WipeDown effects, 445–446

- Zoom effect, 446

Visual Studio, configuring for Web Service example, 916–917

-vm argument

`-vm` **argument**, Eclipse startup options, 186
`-vmargs` **argument**, Eclipse memory tuning, 184
VOD (video on demand), 866–867
volume, setting audio volume and pan, 525
`VRule` **control**, static controls, 247–248
`VSlider` **control**, input controls, 253

W

`warn()`, log levels, 1329–1330

watch expressions

adding/editing, 1254–1255
overview of, 1254

Wave, Adobe, 39

weak keys, 591–592

weak referenced event listeners, 1194

web browsers. See also image browsers

deep linking and, 732
deep linking architecture and, 738–739
disabling integration, 718–719
FABridge for, 717–718
Flash Player providing browser environment, 217
`flashVars` property for passing data into Flex applications, 704–706
Flex as plug-in, 9
fragment identifiers and, 733
interacting with JavaScript, 706–707
invoking ActionScript from JavaScript, 708–709
invoking JavaScript from Flex, 707
lining to external page URL, 706
overview of, 703
passing complex objects between languages, 713–716
passing parameters between languages, 709–711
passing return values between languages, 712
publishing and, 106
RIAs and, 5
summary, 719
web context and, 703–704

Web Orb by Midnight Coders, 851

Web Services

`HTTPService`. See `HttpService` class
RESTful. See RESTful Web Services
tools, 902
`WebService`. See `WebService` class

Web Services Description Language. See WSDL (Web Services Description Language)

Web Services, with Java

calling Web Services from other servers, 912–913
client software requirements, 902
custom server-side errors, 909
enabling Flex to receive SOAP faults, 905
fault handling, 905
Flex sample application, 906–909
Grails code supporting, 903
mapping ActionScript types to XML schema types, 911
overview of, 899–900
`Route.groovy` domain class, 903
`RouteService.groovy` class, 904
serialization with `SchemaTypeManager`, 909–910
server software requirements, 900–902
`strictNullability` property, 911–912
summary, 912–913

Web Services, with .NET

calling from ActionScript, 924–925
calling from MXML, 925
calling from other servers, 926–927
client software requirements, 918
custom server-side errors, 925–926
fault handling, 920–921
Flex sample application, 921
Import Web Service tool, 922–924
`IRouteService` interface, 918–919
overview of, 915–916
route DTO vs. route persistence, 919
`RouteService.asmx`, 919–920
server software requirements, 916–917
summary, 927

Web Tools Platform. See WTP (Web Tools Platform)

web-based installation

launching installed applications from web page, 1225–1226
one-click install with Install Badge, 1223–1225

WebKit browser, 25

WebService class

BlazeDS and, 1067–1068
calling in ActionScript, 844–845
calling in MXML, 841–844

- choosing protocol server for, 850
 - communication with remote servers, 836
 - invoking from BlazeDS, 1071
 - LCDS interaction with, 957
 - logging calls to, 848–850
 - logging initialization, 847–848
 - MXML vs. ActionScript classes, 838
 - RESTful Web Services. *See* RESTful Web Services
 - RPC libraries and, 837
 - Web service written with Java. *See* Web Services, with Java
 - Web Services with .NET. *See* Web Services, with .NET
 - working with, 841
 - while statements, ActionScript loops, 46–47**
 - widgets, Flex applications as, 703**
 - width tag, AIR initial window configuration, 308**
 - Window **class, 772**
 - windows**
 - creating native window, 768
 - NativeWindow class, 771
 - properties for rendering, 768–771
 - Window class, 772
 - working with, 767
 - Windows OSs. *See* OSs (operating systems)**
 - windows-level menus, 780**
 - WipeLeft/WipeRight/WipeUp/WipeDown **effects, 445–446**
 - WireShark software, 846**
 - wizards, new in Flex Builder 3, 98**
 - workbench, Flex 3**
 - editors, 93
 - optimizations, 183–184
 - overview of, 90
 - perspectives, 94–98
 - views, 93–94
 - workspaces, 91–93
 - workflows**
 - Flash Platform development workflows, 482–483
 - Flash/Flex integration options, 484–485
 - Flash-Flex workflow best practices, 494
 - Flash-to-Flex and Flex-to-Flash, 483
 - working sets, breaking workspace into, 161–163**
 - workspaces**
 - breaking into working sets, 161–163
 - editing workspaces list, 186–187
 - preferences, 190
 - workspaces, Flex Builder 3**
 - configuring, 92–93
 - creating/switching, 92
 - overview of, 91
 - WSDL (Web Services Description Language)**
 - converting WSDL operations to MXML requests, 842–844
 - Flex library supporting Web Services, 915–916
 - Import Web Service tool, 922–924
 - overview of, 899–900
 - Web Service fault handling, 905
 - WTP (Web Tools Platform)**
 - creating combined Flex/Java project, 1057
 - debugging Java server-side components, 1255
 - overview of, 966
- ## X
- x,y coordinates**
 - absolute layout (positioning) and, 112
 - coordinate positioning and, 271
 - x and y tags, AIR initial window configuration, 308**
 - XFire plug-in**
 - configuring for Routes Web Service, 901
 - Java/Flex integration, 899
 - XML (Extensible Markup Language).**
 - See also* MXML**
 - accessing XML objects using E4X operators and expressions, 579
 - accessing XML objects using XML class methods, 578–579
 - ActionScript before E4X, 573–576
 - ActionScript using E4X, 576
 - creating SQLite table based on XML data, 945–947
 - exchanging SOAP messages with, 900
 - inserting XML data into SQLite database, 947–948
 - mapping ActionScript types to XML schema types, 911
 - miscellaneous debris when using with ActionScript, 581

XML (continued)

- modifying, 580
- Namespace class, 577
- overview of, 571
- primer for working with, 571–573
- principles of XML languages, 127
- QName class, 577
- reading XML data from a file, 828
- SQLite vs., 930
- XML class, 576–577
- XMLList class, 577
- XML class**
 - accessing XML objects using XML class methods, 578–579
 - dataProvider for Tree control, 624–625
 - overview of, 576–577
- XMLList **class, 577, 624–625**
- XMLListCollection **class**
 - Collections API and, 586–587
 - dataProvider for DataGrid control, 628
 - dataProvider for Tree control, 624–625
 - Tree control as dataProvider for, 624–625

Y

YouTube

- Chromeless Player API in example AIR application, 722
- embeddedPlayer.html, 725–726

Z

Zend PHP framework

- client software for working with PHP, 886
- configuring, 884
- how it works, 886–889
- RESTful Web Services and, 881
- third party frameworks and components for Flex, 31
- third party solutions for RPC services, 850

ZIP archive, importing projects from, 156

zip codes

- formatters, 597–598
- validators, 602

zoom

- design mode enhancements in Flex Builder 3, 98
- navigating Design Area in Flex Builder, 115

Zoom effect, built-in effects, 446