

Symbols

.NET 7–8, 14, 412
 services 324
.Net 222
@ declarative 253

Numerics

3D visualizations, user
 experience 459

A

absolute
 positioning 372
 sizing 78
abstract 405
abstract class 187, 328
abstracted methods 363
abstraction 186, 388
 PureMVC, lack of 415
acceptDragDrop()
 method 513, 515
Accordion 270, 272
 creating 270
 populating 271–272
 user interaction 272
accuracy 141
Action Message Format.
 See AMF
ActionScript 11–12, 14, 19, 38,
 43, 150
 chart classes 531
 components 375
 conditional statement 51

 developing custom
 components 362
 with effects 490
 embedded in MXML 65
 files 65
 formatter 153
 loops. *See* looping
 strict data typing 46
 tidbits 58
 usage 375
 validation 134
 validation. *See* validator
 variables. *See* variable
 working with 44
Actor, MVCS Actor class 438
adaptor 420
addChild() method 377, 379
 component lifecycle 377
 reparent 309
addedEffect property 472
addElement() method 379
 component lifecycle 377
Adobe Consulting 415
Adobe developer
 connection 464
Adobe Integrated Runtime.
 See AIR
Adobe MAX 389
Adobe Open Source
 website 334
Adobe Pixel Bender
 Toolkit 493
advanced components,
 building 375–386
advanced data visualizations
 with 3D 459
advanced item renderer 211

AdvancedDataGrid 172, 195,
 212
 events 210
 item renderer 211
 rendering data 211
AdvancedDataGridColumn
 212
AdvancedDataGridRenderer-
 Provider 212–213
AdvancedListBase 156
 mx 179
agile programming, best
 practices 462
AIR 13
AJAX 10, 12, 270
 limitations 10
Alert 61, 63–64, 66, 269
 advanced, example 291
 appearance, modifying 291
 button 290
 click 292
 default 292
 labels 291
 order 290
 buttonLabel 292
 buttonWidth 292
 class 288
 closing 290
 control 447
 curly braces optional 52
 icon 291
 resizing 291
 show, arguments 289
 static properties 290–291
 transparency, changing 292
 working with 293

alert 52, 226
 advanced, example 289
 confirmation 289
 creating 288–289
 position, changing 292
 algorithm, Luhn mod10 125
 allowMultipleSelection
 property 509
 allowNegative 122
 AMF 7, 10, 320, 324
 open source 321
 performance 322
 stack trace 447
 web service protocol 317
 AMF.NET 323
 AMFPHP 161, 317, 322
 analysis, component code 393
 angleXFrom 460
 angleXTo 460
 Animate 482
 AnimateFilter
 bitmapFilter 475
 Blur 475
 Convolution 475
 DropShadow 475
 example 490
 Glow 475
 property
 customFilter 493
 repeatBehavior 475
 using 490
 AnimateProperty 482–483
 animation 11
 pointless 460
 Ant 25
 anthropology 444
 anti-patterns 361
 Apache 25
 API
 effects 459
 language reference 139
 API Reference 36, 43, 113
 Flash Builder 25, 136
 Flex 35, 102, 113, 553
 OOP 35–37
 ApplicationController.getInstance()
 410
 Application
 container 84
 application
 creating release build of 584
 deployment 579
 development workflow 457
 execution 46
 external parameters 580

 interface 446
 organization 361
 performance 461
 roll your own
 architecture 405
 scaling 465
 server 161, 324
 services 316
 state 363
 Application container 84–86,
 89
 application controller class,
 facets of 411
 application framework micro-
 architecture,
 difference 413
 Application server type,
 coldfusion 332
 ApplicationControlBar 89
 See also containers
 ApplicationModel, application
 model 412
 applyChangesPostLayout 489
 architect 406
 architectural
 conventions 408
 decisions 406
 design pattern 359
 policies, establishing 413
 standards, creating 415
 area chart 540–541
 example 536
 stacking 539
 area series, area stroke 553
 AreaChart 532
 areaFill 554
 AreaSeries 540
 argument 348
 Array 347–348
 Fill 554
 array 48, 56
 associative 56–57
 dot notation 58
 explanation 54
 functions 55
 index 49
 initialization 56
 looping 56
 splice 55
 types of 54
 ArrayCollection 162, 193, 217,
 246, 438, 533
 creating a variable 107
 data typing 330
 dataProvider 181

 definition 159
 feeding a dataProvider 258
 initializing a collection 160
 myAC variable 197
 nested, in MXML 246
 ASP 222
 assets 31
 Atlassian Crucible 462
 Atlassian Fisheye 462
 attaching a skin using CSS
 398
 attribute, binding in
 MXML 383
 audio 11
 autoCenterTransform,
 usage 460
 automatic
 layout 80
 mediation 429
 auto-suggest 348
 autoTransformCenter 489
 axis
 category 534
 chart, renderers 531
 labels 534

B

background color 161
 bar chart 541, 543, 554
 example 537
 stacking 539
 BarChart 532
 base class 139
 component 366
 object typing 395
 BasicLayout 73, 81, 88, 372
 behavior, defined 473
 behavioral logic 360, 367, 457
 best judgment with design
 patterns 408
 best practices 372, 385, 443
 Flex in the enterprise 359
 usage, improving user
 experience 461
 BevelFilter 493
 bevels 445
 binary distribution,
 BlazeDS 334
 bind 365
 MXML binding
 attributes 383
 Bindable 176
 bindable 159, 431
 setting a variable 386

- binding
 - ActionScript 231
 - approaches, reference 231
 - Bindable 176
 - components listen 159
 - defining facets of Flex 407
 - effects 460
 - model to view 335
 - MXML 229
 - simple 66–67
 - two way 69
 - variables to component
 - properties 383
 - BindingUtils 231
 - BitmapFilterQuality 490
 - BitmapImage, Spark 200
 - BlazeDS 18, 317, 587
 - data-centric
 - development 334
 - introduction 13–14
 - setting up 334
 - support 322
 - blogs
 - Dan Orlando 452
 - Joe Johnston 458
 - Peter Dehaan 459
 - Ryan Campbell 458
 - Sujit Reddy G 334
 - BlurFilter effect
 - BitmapImage 490
 - property list 490
 - boiler-plate code, reducing 418
 - boolean 362
 - value 392
 - Bootstrapping 427
 - BorderContainer 202, 372
 - braces 58
 - brown bag lunches 462
 - browser, compatibility 245
 - bubble 224
 - chart 544, 546
 - BubbleChart 532
 - BubbleSeries 544
 - bug fixes 465
 - build application. *See* compile
 - built-in
 - help 35, 38
 - validator 118
 - business
 - delegates 416
 - development 461
 - intelligence 544
 - layer 411, 457
 - logic 150, 209
 - optimization 461
 - requirements 408, 413
 - Button 108–109, 430
 - control 105
 - drop-in item renderer 203, 205
 - Hello! 74
 - styling 453
 - button
 - Display It 63
 - id 136
 - layout 77
 - Update my variable 69
 - ButtonBar 105–106, 180–181, 193, 267
 - Spark 180
 - byte code 10
- C**
-
- C# 47, 342
 - C#.NET 406
 - C++ 46, 342
 - Cage, John, composer (1912–1992) 414
 - Cairngorm 415–416
 - calendar 101–102
 - CandleStick 532
 - chart 546, 548, 553
 - CandlestickChart 547
 - CandlestickSeries 547
 - Canvas 371
 - layout 78
 - See also* container
 - Cartesian class 531
 - Cascading Style Sheet. *See* CSS
 - case sensitive 48
 - case. *See* switch statement
 - cast 145
 - catalog.xml 450
 - dissecting a SWC file 400
 - Catalyst 12
 - category, axis 534
 - CDATA 63, 115
 - cell, editable 207
 - Census RIA Benchmark 320
 - centerPopUp 277
 - CentralDispatcher 412
 - CFBuilder 322, 332
 - CFC 34, 47
 - change 173
 - ChangeWatcher 230
 - isWatching 231
 - unwatch 231
 - weakReference 231
 - chart
 - changing type 536
 - creating 533, 539
 - customizing 536, 553, 555
 - filterData 538
 - filterDataValues 538
 - filterFunction 538
 - filtering data 538
 - interpolateValues 543
 - invoking 534
 - legend 535
 - parts 531
 - series 535
 - single-color series fill 554
 - stacked 539
 - transparency 536
 - type property 539
 - types 532, 540, 553
 - charting
 - introduction 530, 532
 - series 532–533
 - check box 259
 - CheckBox 109, 111–112
 - drop-in item renderer 205
 - checkpoint 209
 - child, node 172
 - class 28, 37
 - base 343
 - creating 349–352
 - creation 341
 - declaration 391
 - effect. *See* effect
 - enforced Singleton 422
 - extending 344
 - file 349
 - interface 351
 - introspection 328
 - methods 47, 347
 - modifiers 350
 - naming 350
 - path 400
 - properties 347, 352–355
 - public 350
 - selector 456
 - superclass 351
 - theory 342
 - top level 47
 - usage 61
 - class hierarchy, Spark list
 - components 185–186
 - classes 81
 - clean code 466–468
 - click 173
 - event 107–108
 - client side 332
 - applications 407
 - client-server,
 - communication 317

- clipboard 13
- close button 276
- CloseEvent 291
 - detail 290, 292
- closeField 547
- code
 - analysis 393
 - chaos 465
 - generating features 340
 - hinting 179, 240
 - reusability 389, 396
 - OOP 361
 - reuse 225, 389
 - runtime shared libraries 400
- code review
 - Flash Builder 338
 - review tool 462
- cognitive science 444
- ColdFusion 317, 322
 - component. *See* CFC
 - data services 322
 - development lifecycle 16
 - and FlashBuilder 28
 - with Flex 7, 332
 - formatters 140, 142
 - non-OO language 35
 - services 324, 332
 - synchronous requests 222
- ColdSpring 417
- collapse 172
- collections 158, 182
 - class 160
 - component 160
 - filtering 216
 - initializing 160–161
 - populating 161
- color
 - background 161
 - scheme 445
- ColorMatrixFilter 493
- ColorPicker 110, 115
- colspan 214
- column
 - advanced renderers 211
 - chart 541
 - stacking 539
 - header 161
 - property 212–213
 - referencing 213
 - repositioning 213
 - resize 167
 - sortable 155
 - sorting 167
 - spanning 214
 - width 165
- ColumnChart 532
- columnIndex 213–214
 - property 212
- columnSpan 214
 - property 212
- columnWidth property 165
- ComboBox 110, 195, 211, 365
 - extending 363
- Command
 - classes 423
 - PureMVC design
 - pattern 415
- command line, applying
 - themes 449
- CommandMap 420, 423
- commandMap.mapEvent 423
- commands
 - example 435
 - purpose 436
- comments 45
- commitment pass 71
- commitProperties()
 - method 363, 376
 - override 379
 - validation 378
- communication
 - between components 359
 - changes in state 384
 - component
 - communication 381
 - data 340
- compile 33, 45–46
 - environment 22
 - script 24
- compiler 28
 - argument 450
 - usage 449
- compiling MXML 362
- errors 401
- compiling 400
- component class 360
 - declaring skins 368
- component implementation
 - skins 370
- component skin states
 - identification 391
- component skins, analysis 393
- components 359
 - architecture 185
 - charts 26
 - communication 381–386
 - container 70
 - core 22
 - custom 28, 85
 - custom, base view state 295
- DataGrid 166
- default 359
- definition 34
- diagram 377
- DropDownList 183
- editable 206
- hasEventListener 233
- item renderer 199
- Label 66
- layout 81
- lifecycle 376–377
- model 389
 - position, changing with Move
 - effect 484
 - positioning 84
- Text 66, 69
- TextInput 66
- types 361
- ViewStack 181
- visual 22
- wrapper 370
- composite components 370
 - grouping 361, 395
 - parts of 370
 - with MXML 362
- composite effect 478, 481
 - composite 471
- composites 374
- computer science 444
- concat 347
- concrete implementation 426
- conditional 361, 393
 - statement 51
- conditions 362
- configuration file, compiling a
 - theme swc 450
- configure, send and return
 - types 329
- configuring
 - data types 330
 - server 324
- Connect to database button 328
- Console tab 30
- constant, defined 512
- constraint 80
 - basic 74, 76
 - content-sized 76, 79
 - enhanced 79
 - fixed 76
 - offset 79
 - relative 76
 - See also* layout
- ConstraintColumn 76–78, 80
- ConstraintRow 76, 78

- constructor 160, 175, 412
 - defined 352
 - function 376
 - ContactsView 430
 - containers 70, 72, 79, 370
 - Application 81
 - ApplicationControlBar 89
 - Canvas 86
 - component type 363
 - DividedBox 92–93
 - Form 93–94
 - Grid 94–95
 - Group based 86
 - host 83
 - layout 72
 - Panel 88
 - sizing 83
 - Skinnable 87
 - SkinnableContainer 86
 - Spark types 84
 - types 84
 - content
 - interactive 193
 - revealing 389
 - Context
 - bootstrapping with 427
 - event dispatcher 435
 - Robotlegs MVCS
 - implementation 429
 - context, role of UXD 445
 - continuing education 463
 - contract, component-skin 391, 395
 - control
 - buttons 105
 - date 101–102
 - definition 34
 - getting values 112, 115
 - numeric 102
 - real time 104
 - user input 96, 134
 - ControlBar 90
 - Controller 335
 - controller
 - initializing 410
 - MVC 407
 - package 434
 - controls, component type 359, 363
 - convenience methods 412
 - conventions 451
 - convert 145
 - ConvolutionFilter 493
 - coordinates layout 72
 - counter-intuitive micro-architecture, usage 416
 - createChildren 376
 - usage 377
 - CreateContactCancel-Command 435
 - CreateContactCommand 435
 - creating
 - address submission form 364
 - MXML composite
 - components 373
 - rectangle, programmatic drawing 394
 - creating components, best practices 190
 - creationComplete 109
 - creationCompleteEffect
 - property 472
 - Creative Suite 26
 - credit card 125
 - CreditCardValidator 125–126
 - criteria, matches value 135
 - CrossFade effect 493
 - cross-platform issues 9
 - CSS 450, 453
 - best practices 451
 - changing drag-proxy
 - icon 526
 - migration 367
 - namespaces 398
 - prefix 398
 - psuedo selectors 455
 - purposes of 452
 - selection by component
 - state 455
 - styling 451
 - CSS 3.0 specification 452
 - Ctrl key 508
 - curly braces 59, 160
 - currency 142
 - CurrencyFormatter.
 - See* formatter
 - CurrencyValidator 127
 - See also* validator
 - currentCSSState 296
 - currentState 296, 299, 368
 - null 312
 - currentStateChange 296
 - currentTarget 112, 175
 - property 175
 - custom component development
 - in Flex 4 389
 - key facets 376
 - custom components 358
 - advanced 375
 - composite 361, 370
 - defined 363
 - developing 187
 - organizing 190
 - simple 361, 363
 - structure 376
 - types 361
 - custom event 234–235
 - custom item renderers,
 - creating 201
 - custom logic 151
 - custom properties, adding 376
 - customizing lists 192–217
- ## D
-
- dashboard 93, 104, 216
 - components 211
 - data
 - binding 66, 331
 - collection 365
 - communication 316, 322
 - enterprise 316
 - controlling a subentry 216
 - display 163
 - displaying 335
 - driven control 158
 - entry 211
 - flat 214
 - hierarchical 214
 - loading 235
 - loading, reference 224
 - model 335
 - object binding 383
 - object types 330
 - property 202
 - representing 438
 - sending to window 286
 - service automation 322
 - services 317
 - tree structure 245
 - trending 549
 - updating 206
 - visualizing 335
 - data return types, auto-detection 330
 - Data Transfer Object. *See* DTO
 - data transfer time,
 - comparing 321
 - data types
 - compatibility 207
 - lists 180
 - primitive 47
 - special 48
 - Data/Services panel 328

- database 154
 - configuration 328
 - schema 327
 - working with 327
- data-centric 323
- JavaEE 334
- data-centric development
 - features 323
 - with Flash Builder 323, 334–340
- dataChange 159, 194, 379
- data-driven interfaces, performance concerns 447
- data-driven, development 323
- dataField 170, 206
 - property 212
- DataGrid 167–168, 170, 195, 197, 336, 431
 - component 166
 - creation 167
 - editing 209
 - inserting rows
 - automatically 211
 - interaction 173–174
 - properties 167
- DataGridColumn 169, 197, 205
- DataGroup 90, 187, 372
 - container 84
- dataProvider 160–162
 - ButtonBar 107
 - component 365
 - DataGrid 168, 197
 - refreshing display 171
 - SkinnableDataContainer 180
 - Spark list control 181
 - with collections 158, 160
- date 144
 - comparison 198
 - control 101–102
 - conversion 198
 - field 196
- Date object 144–145
- DateChooser 101–102
- DateField 101–102, 195
 - drop-in item renderer 205
- DateFormatter 145
 - See also* formatter
- DateValidator 123–124
 - See also* validator
- day 123
- debug, application version 32
- debugging 366, 457
- decimal 141
 - point precision 127
 - separator 127
- Declarations 182
- declarations, MXML tag 365
- declarative graphics 456
 - Degrafa 458
- decouple 114, 225, 234
 - display 374
 - layout 372
- decreased dependencies 415
- deep linking 580
- default state, component 390
- deferred instantiation 263
- defined dependencies, injecting 425
- defining appearance of a state 368
- Degrafa 458
 - integrating with Flex 4
 - projects 459
 - web address 458
- Dehaan, Peter 459
- DeleteContactCommand 435
- dependencies 362, 391, 408, 465
 - decreased 415
 - injecting defined 425
- dependency injection 417–418
 - SwiftSuspenders 419
- dependency-driven code 466
- deployment 15, 584–585, 587
 - model 4, 6
 - path 403
- descendant
 - example 454
 - usage 454
- design 7, 12, 20
 - elements 448
 - mode 34–35
 - view 363
 - workflow 452
- design patterns 217
 - guidelines 413
 - Inversion of Control 417
 - mistakes 407
 - object communication 381
 - second-generation micro-architectures 416
- developing websites 372
- development tool, ColdFusion 332
- DHTML 9, 245
- digitally signed 402
- direction property 165
- dispatch
 - custom event dispatching 384
 - dispatching events 385
- dispatchEvent 433
- DisplacementMapFilter 493
- display
 - content with skins 397
 - dashboard 93
 - list 363
 - logic, separation 366, 390
 - object 382
- displayName 535
- DividedBox. *See* container
- DLL 400
- documentation
 - BlazeDS 334
 - Flex 4 371
- doDrag() method 513, 519
- domain 124
 - model 363, 412
- dot notation 58, 400
- doubleClick 173
- drag image, changing 526
- drag initiator 503
- drag proxy 503
 - image, changing 526
- drag-and-drop 13, 531
 - accepting or denying 513
 - adding to non-List components 518
 - components that support 505
 - custom 516, 518
 - customizing icons 526–527
 - data binding 331
 - enabling on List-based components 506
 - events 504
 - multiple items 509, 511
 - process 503–504
 - user-controlled sorting 508, 512
- dragComplete 504–505
 - event 518
- dragDrop 504–505
 - event 516, 518
- dragEnabled 508
 - property 506, 511
- dragEnter 504–505
 - event 513, 518
- dragExit 504
- DragManager
 - COPY constant 512
 - LINK constant 512
 - MOVE constant 512
 - NONE constant 512
- dragMoveEnabled 508
 - property 507, 511

dragOver 504–505
 dragStart 503, 505
 drawing and styling with
 Spark 394
 drawing API 364
 Dreamweaver 329
 See also IDE
 drop target 503–504
 drop-down and reveal 390
 DropDownList 430
 enhancing usability 184
 Spark 183
 dropEnabled 508
 property 506, 511
 DropShadowFilter effect 492
 drop-shadows 445
 DTO 319, 412
 duration property 460
 dynamic
 class modifier 350
 mediation 418
 SWC linkage 402
 type checking. *See* type
 checking
 dynamic HTML. *See* DHTML

E

easer
 Bounce 497
 Elastic 497
 Linear 497
 Power 497
 Sine 497
 easing, customizing 496
 Eclipse 14, 25, 42, 332
 configuration matrix 324
 plug-in 26
 See also IDE
 ECMAScript 12, 43, 46, 69, 219
 ecosystem 12, 20
 editable 207
 EditContactCancelCommand 435
 EditContactCommand 435
 editing 209
 editor 28
 as renderer, using 211
 See also IDE
 editorDataField 209
 effect
 AnimateFilter 475
 BlurFilter 490
 composite 471, 478, 480
 custom easer 497
 Declaration block 473
 definition 469
 easer 497
 easing functions 496
 easing, theory behind it 496
 example
 Animate 483
 Fade 484
 maintaining font 499
 Move 486
 Pause 481
 photo album 472
 Scale 488
 state transition 477
 GlowFilter 475
 maintaining font style 498
 out of the box types 470
 parallel 479
 programmatic 477
 property list 482
 Resize 472
 rollOutEffect 473
 rollOverEffect 473
 Scale 479
 sequential 478
 sound effect 500
 state transition 477
 suspendBackground-
 Processing 481
 trigger, event 471–473
 using 471
 using ActionScript 474, 476
 Effect object 477
 EffectiveUI 458
 effects 38, 448
 interactive experience 456
 on multiple targets 460
 EJB 322
 EJB 3 417
 element, chart 531
 else statement 52
 email 124, 135
 validation 124
 verifying 118
 EmailValidator 125
 See also validator
 Embed 452
 emotional state,
 identifying 448
 encapsulates 370, 405
 encapsulation 344, 363, 388
 endpoint URI 326
 enforce conventions 413
 enforcement 118
 enhanced performance 415
 Enter key 102
 enterprise application
 architecture 406
 structure 359
 enterprise Flex application,
 development of 406
 enterprise Flex
 development 340
 Enterprise JavaBean. *See* EJB
 environment
 compile 22
 setup 22, 24
 variable 23
 error
 message 141, 148, 150
 property 147, 153
 Event
 handler 42
 JavaScript 40
 overview 16–17
 event 16, 38, 40
 AdvancedDataGridEvent 210
 applicationComplete 228,
 230
 approach, MXML vs
 ActionScript 227
 asynchronous 222
 attribute, priority 242
 attributes, reference 227
 available 136
 binding 68
 bubble 224
 bubbling phase 227
 bus 412, 423
 capture phase 227, 233
 change 102, 132, 158
 click 107–108, 133, 226, 228
 clone 237–238
 constant, naming 237
 creationComplete 202, 410
 custom 225, 237, 241
 data binding 229
 dataChange 158, 202
 DataGridEvent 210
 defining facets of Flex 407
 definition 221
 delivery system 224
 dispatch 222
 dispatcher 223–224
 drag-and-drop 503, 513
 flow 224, 243
 handler 17, 112, 114, 129,
 158
 inheritance 361

- event (*continued*)
 - inline MXML 226
 - ITEM_CLICK 184
 - itemClick 107–108, 158, 173
 - itemDoubleClick 158
 - itemEditBeginning 209
 - item-editing 209
 - itemFocusChanged 185
 - itemRollOut 158
 - itemRollOver 158
 - JavaScript 41
 - List events 173
 - listener 222, 227
 - functions 234
 - removing 232
 - useCapture 233
 - listening 227
 - ListEvent 210
 - metadata 238, 240
 - model 43
 - mouseOver 136
 - naming 235
 - notification 224
 - object 41, 43, 173, 175, 210
 - passing data 384
 - phases, reference 225
 - propagation, stopping 225, 242
 - properties 210
 - render 376
 - ResultEvent 319
 - selectionChanging 185
 - sending and receiving 225, 233
 - synchronous, definition
 - of 223
 - system 222–223, 225
 - trigger 17, 40, 222
 - triggerEvent 133, 136
 - usage 383
 - valueCommit 102, 121, 133
 - event listener 159, 227, 230, 283, 380, 393
 - adding 227
 - Event Listening 433
 - eventDispatcher 433
 - event-driven
 - design 407
 - framework 218
 - language 407
 - EventMap 432–433
 - eventPhase 226
 - execute() 423
 - expand 172
 - expected use 444
 - exporting 457
 - expression, RegEx 128
 - extending
 - classes in the Flex framework 361
 - components 360
 - sub-classing 361
 - extensibility 464
 - definition 465
 - External API 12
 - external asset 585
- F**
-
- FABridge 12
 - Façade, PureMVC design
 - pattern 415
 - Facebook feed 446
 - Factory Method Design
 - pattern 417
 - Fade effect 396, 479, 483
 - example 484
 - property list 484
 - fault handling 319
 - feature implementation 465
 - features, Spark 178
 - feedback 447
 - real time 102
 - file
 - client side 585
 - size, with embedded fonts 499
 - wrapper 24, 33
 - FileSystemDataGrid 195
 - filter 102, 218
 - function 216–217
 - filterFunction 217
 - filtering 158
 - final class 350
 - Firefox 23
 - Fireworks 7, 456
 - importing from 457
 - first generation micro-architecture 414
 - PureMVC, Cairngorm 415
 - fixed sizing 83
 - fixing bugs 384
 - Flash 19
 - editor 14
 - intros 460
 - Flash Builder 25, 179
 - beta releases 371
 - design mode 34
 - features 323
 - html template 580–581
 - importing from Flash Catalyst CS5 457
 - introduction 25, 29
 - layout 30
 - plug-in 26
 - setting constraints 75
 - standalone 26
 - versions 26
 - views 31
 - working with RSLs 403
 - Flash Builder 4
 - data-centric
 - development 340
 - profiler enhancements 463
 - Flash Catalyst CS5 456
 - optimizing workflow 457
 - round-tripping 457
 - Flash Media Server 317
 - Flash platform 469
 - community 389
 - Flash Player 7, 11, 16, 33, 46, 72, 580
 - framework caching 402
 - Flash Remoting 317, 332
 - setting up 332
 - Flash Text Engine 498
 - Flash Virtual Machine 322, 459
 - Flash XML Graphics 367
 - flashvar 580, 583–584
 - accessing in application 584
 - creation of 583
 - Flex 22
 - application development, using SWC libraries 400
 - architecture 405
 - becoming acquainted 11
 - component lifecycle 363
 - frameworks 359
 - how it works 14
 - profiler 463
 - properties 34
 - taking advantage of Flash 11
 - what's new 18–19
 - working with JavaScript 12
 - Flex 3 CSS limitations 453
 - Flex 4
 - components, advanced, creating 374
 - CSS namespace support 453
 - features 398
 - Flex and PHP
 - communication 322
 - Flex Builder 14, 16, 24, 136
 - design mode 34–35
 - Flex Library Build Path 401
 - Flex-AJAX Bridge. *See* FABridge
 - Flex-driven design patterns 405

FlexGlobals
 accessing parameters
 object 584
 topLevelApplication 584
 flexibility, component
 flexibility 395
 FlexLib 344
 FlexUnit 4 385, 463
 FlexUnit.org 463
 flow of control 407
 FLOW3, PHP 5 417
 focus 206
 change 120
 focusInEffect property 472
 focusOutEffect property 472
 fonts
 compiling 450
 embedding 498–499
 system 498
 forEach 347
 Form 98, 108, 271
 component 96
 container 93
 form 99
 container 115
 item editor. *See* item editors
 structure 271
 validation 119, 133
 See also validators
 format 154
 function 144, 151, 154
 formatString 143, 146, 148,
 151–152
 formatted 197
 result 150
 formatters 138–139, 150–151,
 154, 196, 217
 base class 139
 class 139, 151
 component 150
 custom 149
 types of 139–150
 usage 138
 formatting 192
 errors 153
 real-time 138, 150
 scripted 138, 150, 153
 FormHeader 93
 FormItem 93
 foundation building 386
 Fowler, Martin 417, 461
 frameworks 7, 14, 22, 156
 Degrafa 458
 runtime shared libraries,
 RSLs 400
 FTE 498

functions 36
 calling 64
 creation 62
 declaration 62
 getter 354
 inheritance 361
 overriding 376
 parameters 62, 348
 passing data to 175
 return value 348
 scope 62
 setter 354, 357
 usage 61
 FXG 367, 452, 456
 FXP 457

G

garbage collection 380, 393,
 433
 Generate Details Form 337
 generated code 340
 geocentric 364
 get 354
 getAllItems() 330, 336
 getCurrentSkinState() 376
 overriding 393
 usage 380
 getInstance() 411
 getObjectByID() 337
 getter 353
 getters and setters, using 381
 global selector
 by namespace 452
 CSS 398, 453
 Glow 475
 GlowFilter effect 474
 example, on Button 492
 property list 491
 gradient, used for a chart
 series 535
 GradientBevelFilter 493
 GradientGlowFilter 493
 graphic design 444
 graphical summation 322
 graphics API 394
 Greenfield, Ely 389
 Grid container. *See* containers
 GridItem 94
 GridRow 94
 Group 186, 202, 371, 431
 container 84
 container, usage 395
 subclassing 363
 UIComponent
 relationship 187

GroupBase 371–372
 GroupingCollection,
 definition 159
 GUI 105, 324
 guidelines 76

H

Halo 97
 compared to Spark 71, 371
 container 84
 positioning and sizing 71
 slider 104
 happy path 446, 463
 Hatcher, Erik 25
 HBox 371
 HDividedBox 371
 See also containers
 Heider, Jun 464
 hexadecimal response 447
 HGroup 88, 202, 371
 Hibernate 322
 hideEffect property 472
 highField 547
 history 585
 HLOC chart 546
 HLOCChart 532
 HLOCSeries 547
 Hooks, Joel 385
 Robotlegs 417
 horizontalAxis 543
 horizontalCenter 83
 HorizontalLayout 81, 371–372
 HorizontalList 163–164, 195,
 200
 HostComponent
 metadata 368
 usage 370
 HSlider 103, 217
 HTML 40
 character codes 362
 CheckBox 109
 control 96
 creating wrapper files 24
 drop-down menus 245
 Form 97
 form 93, 101, 111
 frames 92
 layout 73, 80, 82
 page 294
 table 94, 166, 212, 214
 validation 117
 viewing page in browser 223
 wrapper deployment 585

html
 template 581
 wrapper 581
 HTTP 10, 317, 319, 324
 HTTPSService 283, 318
 declaring in MXML 318
 invoking 319
 setting up 332
 using the 318
 human factors and
 ergonomics 444

I

IBitmapFilter 493
 ICollectionView 181
 icon 172
 iconField 157
 IContactService 437
 IContainer 371
 id
 property 97–98
 referencing by 213
 selector CSS, usage 454
 IDE 7, 14, 42, 323, 332, 340
 Aptana Studio 24
 extensions 24–25
 no cost option 21
 identifying emotional state
 448
 IDropInListItemRenderer 202
 IEaser 482, 496
 IEventDispatcher 412, 433
 if statement 51–52
 IFlexDisplayObject 277
 IGraphicElement 187
 IInterpolator 483
 IItemRenderer 202
 IListItemRenderer 202
 Illustrator 456
 importing from 457
 Image
 as an item renderer 204
 drop-in item renderer 205
 images 11
 implementation
 component 358, 369
 implementations of MVC 407
 implements 38, 437
 import 357
 improve user experience with
 animation 456
 includeIn property, usage 395
 InDesign 456
 Index, referencing by 213

IndexChangedEvent 185, 267–
 268
 properties 267
 indexed array 54
 industrial design 444
 inherit 360
 inheritance 37, 185, 360, 365
 concepts 343–344
 initialize() 412
 inject IoC 418
 injection points 420
 Injector 420
 injector.mapSingleton 422
 inline
 editing 217
 item renderer 204
 instantiation controller 410
 int 49, 353
 integer 122
 integrated development envi-
 ronment. *See* IDE
 integrating component
 reuse 367
 interaction events 218
 interface, definition 351
 internal scope 350
 Internet Explorer 23
 introspection service 327
 Invalid
 format 141, 153
 value 141, 153
 invalidateDisplayList()
 method 202, 379
 invalidateProperties()
 method 202, 392
 invalidateSizeFlag 379
 invalidateSkinState() 377, 392
 invalidation 376
 component lifecycle 377
 methods, order of 377
 phase 377
 routines, making calls to 377
 Inversion of Control 417
 Martin Fowler 418
 IRA design decisions 317
 item editing, enabling 206
 item editors 211, 217
 creating 207
 customizing 206
 declaring 208
 default 207
 initializing 209
 item renderers 158, 179, 198,
 217
 advanced 211
 best practices 190

component type 363
 default 364
 drop-in 204
 and editor 211
 implementation 199
 inline 203
 MXML 199
 spanning columns 211
 Spark 180
 types of 198
 itemClick 107–108, 173
 event 176
 itemClickEvent 113
 itemEditBegin 209
 itemEditBeginning 209
 itemEditEnd 210
 itemFocusChanged,
 IndexChangedEvent 184
 itemRenderer 181
 HorizontalList 201
 ItemRenderer, using 198
 Iteration II 415
 iteration. *See* looping
 UIComponent 187

J

J2EE 31
 Cairngorm 416
 services 324
 setting up 334
 Java 2 Platform, Enterprise
 Edition. *See* J2EE
 Java and Flex, integrating 322
 Java Runtime Environment.
See JRE
 Java Spring 417–418
 JavaEE and BlazeDS, data-
 centric development 334
 JavaFX by Sun 8
 JavaScript 10, 16, 46
 alerts 288
 asynchronously updating
 page 223
 and date controls 101
 embedded 65
 event 41
 Flash Player version
 checking 33
 and Flex 12
 getElementByID 97
 interpreted language 10
 validation 125
 JBoss 322
 JMS 14

job security 468
 JRE 22
 JSON 318

K

key
 arrow 133
 Ctrl 510
 Enter 133
 Tab 133
 keyboard stroke 118
 keyword, bindable 68

L

Label 395
 subclass, `TextBase` 395
 label 163
 chart 531
 control 99
 formatting 194
 function 198
 multicolumn 195, 197
 parameters 197
 single column 195
 specifying 163
 label function
 multicolumn 196
 single column 195
 labelDisplay 392
 labelField 163, 181, 193
 `SkinnableDataContainer`
 180
 using 193
 labelFunction 157, 170, 181,
 197
 `SkinnableDataContainer`
 180
 using 193–198
 Lafferty, Joan 456
 large applications,
 managing 395
 large-scale enterprise
 environment 323
 large-scale Flex applications,
 working with 407
 Law of Demeter 362, 466–467
 layers 11
 of separation 408
 layout 70, 96
 absolute 72–73, 76
 automatic 80
 component 367
 constraint 74

 defining 372
 horizontal 92
 offset 79
 pass 71
 relative 75
 rules 395
 Spark usage 372
 types of 72
 usage 371
 vertical 92
 layout manager 71
 LayoutManager 379
 LCDS 13, 18, 161, 317, 323
 setting up 334
 legend 535
 length
 property 347
 leveraging existing code 388
 lib folder 426
 library project, creating 401
 library.swf, dissecting a SWC
 file 400
 lifecycle application 412
 line chart 543–544
 example 532
 line series 534
 LineChart 532
 link type 403
 LinkBar 105–106
 LinkButton 105
 styling 454
 linking, deep 580
 Linux 24
 List 167
 component 156, 161, 163
 events 158
 genealogy 156, 158
 interaction 177
 Spark 179, 187
 list
 components skinning 190
 customization
 techniques 198
 events handling 210
 ListBase 156, 158, 186, 191
 MX 179
 Spark 179
 sub-classing 185
 list-based components 155,
 217, 364
 architecture 185
 creating 185–186
 drag-drop data binding 335
 Spark 185
 list-based controls, Spark 180

list-based events, handling 187
 ListCollectionView 158
 listener 68, 119
 function 202
 listItems 182
 ListSkin 190
 LiveCycle 8, 13
 LiveCycle Data Services.
 See LSDS
 load-config 450
 LoadContactsCommand 435
 lockedColumnCount 157
 lockedRowCount 157
 logical
 operators 59
 reasoning 406
 login form 295
 look and feel design 448
 looping
 array 56
 do while 50
 for 48, 56
 for each..in 50, 57
 for..in 49, 57
 while 50
 XML 49
 loose coupling 345, 381
 micro-architecture
 advantages 415
 Robotlegs 418
 loosely coupled 391, 413, 418
 model 367
 Loughran, Steve 25
 lowField 547
 Luhn mod10 125

M

main business classes,
 instantiating 412
 maintenance, maintaining
 code 388
 manifest.xml 450
 mapClass 421–422
 mapping
 components 425
 utilities 427
 mapSingleton 421
 mapSingletonOf 421
 usage 437
 mapValue 421–422
 mapView 423
 Martin, Robert C. 468
 master detail form,
 generating 337

- Mate 416
 - mathematical operators 60
 - measure() 376, 379
 - measurement pass 71
 - media files, compiling 450
 - mediator 407, 412
 - Mediator design pattern, delivering and receiving messages 429
 - MediatorMap 420, 423
 - Mediators 423
 - PureMVC design pattern 415
 - relationship 423
 - Robotlegs MVCS implementation 430
 - memory, storing 218
 - Menu 108
 - creating 252–253
 - customizing 254
 - item attributes 254
 - positioning 254
 - showing 253
 - user interaction 256
 - menu
 - array, children 246
 - BarButton 180
 - drop-down 245
 - U.S. states 364
 - MenuBar 258, 262
 - creating 258
 - customizing 259–260
 - events 260
 - positioning 258
 - user interaction 260
 - MenuEvent 256
 - properties 260
 - merged to code. *See* RSL
 - metadata
 - binding 368
 - Effect 472
 - event, for components 240
 - SwiftSuspenders 419
 - method 38, 347
 - argument 348
 - getter and setter functions 355
 - overrides 392
 - parameters 348
 - See also* functions
 - micro-architecture
 - already-existing projects 416
 - definition 413
 - framework, difference 413
 - Mate 416
 - Parsley 416
 - Robotlegs 416
 - Swiz 416
 - why 413
 - Microsoft 444
 - MIDI 445
 - minField 540
 - modal 276
 - placement 280
 - Model 247
 - advantages 247
 - property, source 248
 - model
 - application state 363
 - data 324, 335, 363
 - MVC 407
 - view 363
 - Model-View-Controller 324, 359, 407
 - design pattern
 - breakdown 335, 362
 - foundation of Flex 413
 - visualizing 407
 - Model-View-Controller-Service 425
 - modifying existing code 465
 - modular applications
 - building 427
 - module 404
 - month 123
 - mouse click 133
 - mouseDown 503, 505
 - mouseDownEffect
 - property 472
 - mouseMove 503, 505
 - event 513, 518
 - mouse-over 118
 - mouseUpEffect property 472
 - Move effect 484
 - move vs. copy 507
 - moveEffect property 472
 - multicolumn 195
 - multiple class selectors 456
 - Musical Instrument Digital Interface. *See* MIDI
 - MVCS 425
 - MX component 71
 - MX containers, relationship to Spark 371
 - MXML 16, 33–34, 151, 207
 - ActionScript embedded 65
 - ActionScript overview 38, 42
 - charts 531
 - component 97, 149
 - defining layout 399
 - mapping Mate 418
 - skin implementation 390
 - structure 38
 - syntax 394
 - the Declarations tag 382
 - XML-compliant 63
 - MXMLC 24
 - See also* compiler
 - myClassInstance 422
-
- ## N
-
- named injections,
 - configuring 419
 - namespaces 179
 - usage, runtime shared libraries 400
 - naming convention 467
 - NaN 48
 - native support 322
 - NavigatorContent 263
 - Needle for Ruby 417
 - nested array 245
 - Network Monitor 336
 - new features in Flex 4 452
 - new operator 346
 - node 172
 - Notepad 24
 - notify event listeners 384
 - null 48
 - NumberFormatter 141
 - additional properties 140
 - See also* formatters
 - NumberInterpolator 483
 - NumberValidator. *See* validators
 - numeric controls 102, 104
 - NumericStepper 102, 377
-
- ## O
-
- object 36–37, 57
 - binding 407
 - children 70
 - class definition 342
 - classes, relationship 342
 - primitive 346
 - property 347
 - scope 343
 - theory 342
 - working with 346, 348
 - object oriented
 - concepts 341, 343–344
 - function 342
 - inheritance 343

- object oriented (*continued*)
 - methods 342
 - properties 342
 - theory 342, 345
 - Object property looping 49
 - objective, staying objective
 - about code 384
 - object-mapping 331
 - object-oriented language.
 - See* OOP
 - object-oriented programming.
 - See* OOP
 - OLAPDataGrid 195
 - on-demand loading classes 400
 - OOP 35, 43–44, 388, 405, 417
 - fundamental
 - characteristics 361
 - open source framework,
 - Degrafa 458
 - openField 547
 - operations
 - calling service 328
 - invoking 335
 - tag 320
 - operator
 - comparative 60
 - logical 59
 - mathematical 60
 - optimizing dependencies,
 - Robotlegs 418
 - overlap layout 73
 - override 359, 393
 - overriding
 - methods of
 - UIComponent 363
 - properties 360
- P**
-
- package 37, 61, 328
 - creation 341, 350
 - declaration 376
 - import 47, 63
 - top level 47
 - page layout 372
 - pagination 446
 - pane, resize 92
 - Panel 99, 371
 - component 34
 - container 84
 - extending 188
 - See also* containers
 - Parallel 479
 - parameters, passing 413
 - parent class 343, 376
 - Parsley, micro-architecture 416
 - partAdded() method 376, 380
 - intercepting 380
 - overriding 393
 - partRemoved() method 376, 393
 - intercepting 380
 - passing data 318
 - internally 381
 - Path environment variable 23
 - pattern
 - mask 143
 - matching 129
 - patterns and practices,
 - defining 362
 - Pause effect 481
 - performance
 - data-driven interfaces 447
 - layout 72
 - list optimization 194
 - PERL 222
 - perspective 29
 - customization 30
 - debug 29
 - development 29
 - profiling 29
 - phone number 146, 150
 - North American 128
 - validator 128
 - verifying 117
 - PhoneFormatter 146–147, 152
 - PhoneNumberValidator.
 - See* validators
 - Photoshop 7, 12, 456
 - importing from 457
 - pixel-shaders 496
 - PHP 16, 28, 31, 47, 222
 - services 324
 - setting up 327
 - using 326
 - picklist controls 109, 112
 - pie chart 549, 551, 554
 - changing colors of 554
 - PieChart 532
 - Pixel Bender Shaders 493, 496
 - pixel-shaders 493
 - plot chart 551
 - PlotChart 532
 - POJOs 322
 - PolarChart class 531
 - polling 377
 - polymorphism 388, 422
 - pop-up 288
 - ActionScript tags 274
 - centering 277
 - closing 276
 - communication 280
 - creating 274, 277
 - data, retrieving 283
 - icon 292
 - initialization 274
 - MXML tags 274
 - parent 276
 - placement 278–279
 - positioning 277, 280
 - PopUpButton 105, 108–109
 - PopUpManager 275–277
 - PopUpMenuButton 105, 107–109
 - position 72
 - positioning layout 73
 - postal code 147
 - Canada 132
 - See also* ZIP code
 - precision 140
 - predefined layout 373
 - predicting user paths 446
 - preloader 85
 - Presentation Model 362–363
 - design pattern 407
 - primitive
 - data type 47
 - object 346
 - PrintDataGrid 195
 - private 62
 - scope 343
 - procedural scripting
 - language 417
 - product development
 - lifecycle 456
 - production build
 - creating 584
 - deploying to 586
 - profiler 29
 - profiling 194, 457
 - applications 463
 - programmatic drawing
 - creating a rectangle 394
 - fills and strokes 394
 - programmatic skin classes 448, 450
 - programmer's curse 465
 - progress bar, preloader 85
 - project 33
 - assets 27
 - creation 31
 - output directory 31
 - setup wizard 332
 - structure 391
 - wizard 31
 - prompt property 183

proof of concept 204, 407
 building a 408
 properties 38, 352
 getters and setters 381
 height 83
 inheritance 361
 injection 419
 of a class 360
 proxy 382
 width 83
 Property control mapping 337
 protected 62
 scope 343
 protocol 320
 prototype application
 development 408
 prototyping 34, 457
 proxies, PureMVC, data
 management 415
 psuedo selectors 452, 455
 CSS 398
 psychology 444
 public 62
 properties 374, 399
 scope 350, 352
 PureMVC, first generation 414
 Python 222

Q

Quality Assurance 448

R

radio button 259
 RadioButton 110, 112
 RadioButtonGroup 111, 113
 range, filtering values 217
 raw data, formatting 197
 RDA 13
 real world scenario, component
 development 389
 real-time formatting 154
 refactor 389, 408
 refactoring 462
 reformat 141
 RegEx 128
 RegExpValidator 129–130
 registration point 460
 regressions 408
 causing 464
 regular expression 128–130
 relationship
 components and skins 369
 defining 423

relative
 layout 74
 sizing 78
 release
 build 584
 schedule 465
 remote object 532, 586
 RemoteObject 161, 283
 remotng 587
 removedEffect property 472
 renderer
 editing 206
 property 212
 rendererIsEditor 211
 rendererProviders 212
 reparent 309
 RepeatBehavior 475, 482
 request-response 318
 required property 370
 requiredFieldError 119
 requiresSelection,
 SkinnableDataContainer
 180
 reserved words 48
 Resize 472
 effects, usage 396
 property list 486
 resize 74
 real time 102
 resizeEffect property 472
 resource consumption, valida-
 tion routines 377
 resource overconsumption 376
 response handling 319
 responsiveness and
 performance 445
 REST 317
 result
 server 319
 sets 327
 return on investment. *See* ROI
 return type 62
 configure 330
 setting 336
 return type configuration,
 invoking methods 336
 reusability 152, 173, 270
 accepting input
 parameters 61
 handling menus 257
 limited 114–115
 mechanisms of 388
 package. *See* package
 power of 387

reusable 381
 components, building 388–
 400
 components, making 374
 Reveal, component 389, 391,
 399
 reverse domain notation 350
 reverse-engineer, SWF files 362
 revision-control software 462
 RGBInterpolator 483
 RIA 4–5, 19, 586
 best of both worlds 5–6
 contenders 7–8
 evolution of 185
 and Flash 11–12
 how they do it 6
 making applications 448
 solution 5
 Rich Desktop Applications.
See RDA
 Rich Internet Applications.
See RIA
 RichEditableText 98–99
 RichText 98–99
 RichTextEditor 98
 Robotlegs 406, 416
 Actor 426
 architectural structure 426
 building an application
 with 427
 Command 426
 component wiring 418
 Context 426
 facilitating
 communication 426
 framework, usage 418
 library, accessing online 426
 loose coupling 418
 Mapping Utilities 420
 Mediator 426
 MVCS, creating an applica-
 tion with 426–440
 support forum online
 address 440
 robust 413
 ROI 10
 rollOutEffect property 472
 rollOverEffect property 472
 Rotate effect 488
 property list 489
 Rotate3D 459
 declaring 460
 rotation 460
 rounding 140–141

- row
 - header 167
 - selected, accessing
 - directly 176
 - spanning 214, 216
 - summary 214
 - rowCount 157
 - rowHeight 157
 - DataGrid 205
 - rowSpan property 212
 - RPC 317, 324
 - RSL
 - cache 403
 - framework 403
 - importing 403
 - merged to code 403
 - standard 401
 - types of 400
 - RTMP 7, 317
 - RubyAMF 323
 - runtime shared libraries
 - 400–404
 - usage 400
 - RWA 12
 - the last stand 9
- S**
-
- SaveContactCommand 435
 - scalability 178, 217, 413, 465
 - Cairngorm 416
 - scalability wall 461
 - avoiding 462
 - definition 461
 - user experience effects 461
 - scalable 359, 409, 436
 - Scale 488
 - schema, reading from Flash
 - Builder 327
 - scope 62
 - internal 343
 - public 343, 354
 - purpose of 343
 - static 343
 - Script 562
 - scrollbar 164
 - ScrollControlBase 156, 179
 - scrolling 166
 - SDK 14, 16, 21–22, 24, 42
 - SDLC 16
 - with Flex 16
 - second generation micro-architectures, design patterns 416
 - select and apply styles
 - to a component 398
 - selectable 157
 - SelectContactCommand 435
 - selectedIndex 157, 175, 267
 - SkinnableDataContainer 180
 - selectedIndices 157
 - selectedItem 157, 175
 - SkinnableDataContainer 180
 - selectedItems 157
 - selectedValue 112
 - selectionChanged,
 - IndexChangedEvent 184
 - selectionChanging
 - event 188
 - IndexChangedEvent 184
 - selectors, setting 398
 - send and return types,
 - configuring 329
 - sendHTTPRequest() 319
 - separation
 - of concerns 372, 407
 - programming practices 191
 - Sequence 478
 - series
 - chart 531
 - fills 554
 - server 222
 - configuration dialog 326
 - connect from existing
 - project 324
 - connect using wizard 324
 - connecting 324
 - error 447
 - server side 316
 - development 332
 - platform 319, 323
 - technology 318, 321, 323
 - service stub, generating 327
 - services 412
 - call 318
 - class 328, 413, 436
 - folder 328
 - generating with Flash
 - Builder 327
 - generator window 328
 - types 325
 - using 324
 - set 354
 - setter 353
 - ShaderFilter 493
 - sharing code during
 - runtime 400
 - shell 23
 - shortcuts 58
 - showFeedback() method 513
 - Silverlight 8, 17
 - simple components,
 - grouping 361
 - Simple Object Access Protocol.
 - See* SOAP
 - simple Spark components 365
 - SimpleMotionPath 475
 - simulated 3D rendering 459
 - Sine 482
 - singleton 408
 - class 411
 - singleton design pattern
 - PureMVC 415
 - purpose 411
 - size, layout. *See* layout
 - sizing
 - fixed 83–84
 - variable. *See* layout
 - skin class 380
 - skin parts
 - bindings, usage of 370
 - declaring 368
 - defining appearance 368
 - identification 391
 - object typing 395
 - Skinnable container 84
 - SkinnableComponent 191, 202
 - extending 186
 - UIComponent
 - relationship 187
 - SkinnableContainer 87, 202, 372, 390, 399
 - SkinnableContainerBase 186, 191
 - SkinnableDataContainer 179, 186–187, 191
 - properties 180
 - sub-classing 185
 - skinning 366
 - SkinPart
 - metadata 368
 - usage 369
 - skins 359, 443, 448
 - Spark 366
 - Spark architecture 185
 - SkinState
 - binding 368
 - defining 368
 - metadata 368
 - slide open and reveal 390
 - slide range 217
 - slider 104, 115

- SmallTalk 407
 - smoke and mirrors 408
 - snapInterval 103
 - SOAP 10, 317, 319
 - packet parsing 320
 - performance issues 321
 - Social Security Number. *See* SSN
 - SocialSecurityValidator 131
 - See also* validators
 - SolidColor 554
 - sortable property 168
 - sortableColumns 168
 - sorting 158
 - with drag-and-drop 508
 - SoundEffect effect 500
 - example 501
 - panning 500
 - property list 500
 - source 119
 - BlazeDS 334
 - code 31
 - Spacer 82
 - See also* layout
 - span 100
 - Spark 97, 178
 - architecture, key facets 359
 - as compared to Halo 71
 - compatibility 179
 - component architecture 360
 - container 84
 - containers, list of 371
 - drawing and styling 394
 - list-based components, interacting with 184
 - position and sizing 71
 - Reusability theory and concepts in 389
 - skinning essentials 370
 - slider 104
 - spark.layouts package 372
 - Spinner 102
 - splice 55
 - split 348
 - Spring Services 322
 - Spring.NET 417
 - Spry 270
 - src folder 328
 - SSN 129–130
 - pattern 131
 - rule 130
 - stage 76
 - staging environment, deploying to 586
 - stakeholders 465
 - standards and best practices, progressing toward 412
 - standards, UML 446
 - state management, Spark 185
 - state related items, managing 440
 - stateful tabbed navigation 446
 - stateGroups 303
 - stateless page 294
 - states 34, 294
 - default 295
 - defining 391
 - property 295–296
 - setting custom 392
 - using Flash Catalyst CS5 457
 - static
 - page 294
 - property 383
 - SWC linkage 402
 - type checking. *See* type checking
 - vs. dynamic type checking 46
 - stock price, charting 547
 - stopImmediatePropagation 242
 - stopPropagation 242
 - strict data typing 46
 - String 65, 144, 162
 - style properties, Flex 4, using 451
 - Style tag 398
 - styleName, property 456
 - styles 38
 - inheritance 361
 - stylesheet 38, 586
 - styling 398
 - subclass 38, 344, 360
 - subcomponents, instantiating 377
 - super calling 376
 - superclass. *See* class
 - support for namespaces, CSS 452
 - SWC 400
 - accessing classes inside 402
 - advantages 449
 - disadvantages 449
 - SWC files
 - converting classes to 401
 - deploying 403
 - dissecting 400
 - SWC libraries
 - centralizing 402
 - distribution 400
 - swfobject 580, 582–585
 - SwiftSuspenders 419
 - switch statement 51, 53
 - SwitchSymbolFormatter 149–150
 - See also* formatters
 - Swiz 416
 - SWZ file 402
 - Systems Development Life Cycle. *See* SDLC
- ## T
-
- tab 269
 - tabbed navigation 446
 - TabNavigator 181, 268, 270
 - creating 268
 - user interaction 269
 - target 460
 - targeted user
 - demographic 444
 - TDD 385, 461
 - test driven development. *See* TDD
 - testable code 466
 - writing 395
 - text property 120
 - Text, control 98–99
 - TextArea 98
 - TextBase 395
 - TextInput 98, 123, 135
 - data copied to string variable 69
 - prompting a user 344
 - themes 400, 443, 448
 - applying 449
 - SWC 449
 - usage 448
 - theory 341
 - thumb 103
 - tick 103, 546
 - tight coupling 234, 264, 465
 - preventing 461
 - Tile 165
 - component 166
 - TileGroup 372
 - TileLayout 81, 372
 - TileList 195
 - component 166
 - direction 164
 - performance 166
 - time field 196
 - time to market
 - opportunity 465
 - timeline 11
 - TitleWindow 274, 371
 - close button 276
 - ToggleButton 395

- ToggleButtonBar 105–106
 - Tomcat 322
 - trace 48, 56
 - log 28
 - transfer speed 324
 - Transition 493
 - definition 494
 - reversing 396
 - usage 396
 - transition effects
 - declaring with skins 399
 - using 397
 - transitions
 - shader 496
 - using Flash Catalyst CS5 457
 - transparency 29
 - chart series 535
 - Tree 156, 171–172, 195, 381
 - creation 171
 - TreeSelector 382
 - attributes 382
 - trigger 119, 132, 136
 - validation 121
 - triggerEvent 119–120
 - troubleshoot 384, 447
 - turnkey, BlazeDS 334
 - type checking
 - dynamic 46
 - static 46
 - types of errors 444
 - typing 331
- U**
-
- UED 458
 - UI 98
 - control 96
 - interaction 74
 - UI mockup
 - creating 445
 - UIComponent 156, 179, 185, 351–352, 366, 387
 - default effects 486
 - granddaddy of visual Flex components 359
 - transitions property 478
 - uint 353
 - UML 446
 - undefined 48
 - underscore 353
 - understanding code reuse 389
 - unique identifier 337
 - unit test 412
 - coverage 436
 - effectively 467
 - usage example 385
 - unit testing 385, 461
 - relationship to user experience 463
 - untestable code 362
 - Universal Mind 417, 458
 - unmapListeners 433
 - unwatch 232
 - updateDisplayList() 202, 363, 376, 379
 - updateProperties 202
 - updating records 340
 - usability 4, 9, 117, 133, 155, 364
 - data entry 211
 - interface 183
 - testing 448
 - use case scenarios 444
 - code reuse 396
 - use of 3D space in Flex 4 459
 - user
 - acceptance 584
 - group 587
 - user experience 443
 - background 443
 - common mistakes 445
 - considering context 445
 - example 446
 - interactivity 445
 - metaphorical references 445
 - navigation 445
 - principles 444
 - user demands 5
 - user feedback 445, 460
 - obtaining 447
 - user input 99
 - capturing 96
 - user interactions 460
 - using Flash Catalyst CS5 457
 - user interface designs, importing 457
 - user interface development 359
 - user response 446
 - user stories
 - definition 444
 - fictional character 445
 - useVirtualLayout, SkinnableDataContainer 180
 - UXD 457
 - best practices 447
 - process 444
- V**
-
- validate 154
 - ZIP Code 132
 - Validate Configuration 326
 - validating server
 - connection 326
 - validation 210
 - committed value 118, 133
 - overview 118
 - pass-through 118, 133
 - phase diagram 378
 - real time 118, 132–133
 - scripted 118, 134
 - tidbits 135
 - validator 122, 124, 135, 138, 154
 - built-in 118
 - class 118
 - types of 118–132
 - value
 - axis 534
 - matches criteria 135
 - object 319, 331, 412
 - variable
 - data typing 46
 - global 85
 - naming 46
 - public 207
 - sizing 83
 - special data type 48
 - strong typing 369
 - types of 45, 48
 - variableRowHeight 157
 - Datagrid property 205
 - VBox 371
 - vs. VGroup 398
 - VDividedBox 371
 - See also* containers
 - vertical direction 165
 - verticalCenter 83
 - VerticalLayout 81, 372
 - VGroup 200, 202, 372, 398
 - VIBE 448
 - Model 443, 448
 - video 11
 - view 294
 - base view state 295
 - MVC 407
 - stack 268
 - view component mediator 432
 - view states 80, 380
 - base 295
 - basedOn 302
 - change, event handler 300, 302
 - change, event handler syntax 302
 - change, property 297, 300

view states (*continued*)
 change, property syntax 298
 creation policy 306
 default 295, 300
 destruction policy 306
 events 296, 311, 313
 events,
 currentStateChange 312
 events, enterState 312
 events, exitState 312
 example
 change property 297–298
 excludeFrom 306
 includeIn 305
 itemDestructionPolicy 306
 real-world sample 313
 reparent 309
 excludeFrom 305
 groups 302–303, 305
 groups, definition 303
 groups, difference from state 303
 groups, example 302
 groups, syntax 302
 include and exclude 305, 309
 includeIn 305
 includeIn vs
 includeInLayout 306
 introduction 295
 itemCreationPolicy,
 details 309
 itemCreationPolicy,
 immediate 308
 itemDestructionPolicy,
 details 309
 reparent 309, 311
 uses 295
 ViewMap 420, 425
 views 367
 design 29
 and perspectives 29, 31
 states, declaring 368
 ViewStack 181, 262
 adding navigation 264, 266
 creating 262, 264
 user interaction 267
 using Spark components 263

visible 170
 component property 202
 property 396
 visual and interaction
 design 445
 visual components 366, 387
 visual stimulus 446
 VO 412
 void 48
 volume, SoundEffect effect 500
 VSlider 103

W

watch 231
 watermark 530
 web application 218
 drawback 4
 prolific nature of 3–4
 web design, evolution of 185
 Web Service Description Language. *See* WSDL
 web services 7, 28, 532, 586
 consuming 319
 infrastructure 317
 integration of 319
 SOAP, RPC 319
 WebLogic 322
 WebOrb 323
 WebService 161, 283, 319
 setting up 332
 WebService.getOperation 320
 websites, developing 372
 WebSphere 322
 white space 39
 window pop-up 274
 WinZip 400
 Wipe effect 495
 definition 495
 example 495
 wired 418
 wireframing 457
 wizards 28
 data-centric
 development 323
 project 31
 word wrapping 155
 wordWrap 157

workflow 323
 designer-developer 457
 faster 387
 speeding up 340
 working with data
 services 323
 wrapper 370
 component 370
 customizing 584
 files 580
 WSDL 317, 319

X

XML 10, 16, 38, 145, 219, 235, 248
 benefits 248
 communication between client and server 320
 data 161
 data source 181
 external 251
 for menus 249
 namespace 264
 parsing 235
 root node rule 249
 rules 63
 structured data 171
 XMLContactService 437
 XMLHttpRequest 9–10
 XMLList 49, 249
 XMLListCollection 193, 250
 definition 159
 property, source 250

Z

Zend 317
 framework 322
 Zend_AMF 322
 ZIP Code 147
 format 131
 validate 132
 zip file 400
 ZipCodeFormatter 148
 See also formatters
 ZipCodeValidator 131–132