

INDEX

The prefix “ON” before a page number indicates that the entry can be found in the online appendices.

A

- Abdera (Apache)
 - Atom Syndication Format and, 247
 - libraries, 73
- accept headers
 - Accept header, 235–237
 - Accept-Language header (HTTP), 231
 - Web scripts and, 182–183
- access (file-based), integrating with Alfresco, 456
- access control
 - extension, 128
 - modifying, 127–128
- Access Control Entries (ACEs)
 - ACLs and, 120
 - ordering and evaluation, 121
- Access Control Lists (ACLs), 120–123
- ACEs. *See* Access Control Entries (ACEs)
- ACLs. *See* Access Control Lists (ACLs)
- Action Wizard, 340
 - actions
 - action list, document library (Share), 446–447
 - document library and (Share), 416–417
 - jBPM, processes and, 137–138
 - repository, 320–321
- Actions API, ON.91
- actions-common.get.head.ftl file, 417
- Activities service, 58, ON.98–99
- Activities Web scripts (RESTful API), ON.9–12
- Advanced Space Wizard, 347, 352
- alf-force-success-response header, 185
- Alfresco, integrating with other
 - applications, 451–511
 - capabilities added, 453
 - content management and, 452–455
 - content management integration patterns. *See* content management integration patterns
 - options for, 455–457
 - reasons for, 451–452
- Alfresco basics, 3–17
 - architecture, 7–8
 - content applications, 10–12
 - ECM and, 5–7
 - meeting specific needs, 12–14
 - open source software, 14–17
 - overview, 3–5, 7–10
- alfresco connector, 308
- alfresco endpoint, 308–309
- Alfresco Java Foundation API, 59
- alfresco/extension directory, 106
- alfrescoNtlm
 - customizing (example), 105–108
 - subsystem, 104
- alfresco-ticket authenticator, 308
- AMP files, defined, 48
- annotated controller example (Surf Framework), 293–294
- annotations (content service mappers), 488–492
- Apache Abdera
 - Atom Syndication Format and, 247
 - libraries, 73
- Is. *See also* JavaScript API; repository FreeMarker template API; scripting API
- Actions API, 91
- CategoryNode Object API, 92–93
- Check in/Check out ScriptNode API, 84–85
- Classification API, 92, ON.69–70
- Content Application Server and, 30–33
- E4X API, 234
- embedded, services and, 59–61

- Form Data API, 244–245
- FreeMarker Template API, ON.43
- JavaScript API, ON.43
- JSON Object API, 249–250
- object-oriented level APIs, 75–76
- People API, 73
- provided by cache root object, 251
- remote APIs, 61, 309
- TemplateNode model API, ON.64–68
- Version History API, ON.68–69
- Workflow API, ON.71–72
- XML Content Processing API, ON.70–71
- app Surf platform root-scoped object, ON.56
- applications. *See also* Knowledge Base applications, foundation for
 - application-related zones, 116–117
 - architecture and, 13
 - basics of, 8
 - configuring, 13
 - content applications, 10–12, 288–290
 - customizing, 13
 - integrating with Alfresco. *See* Alfresco, integrating with other applications
- architecture
 - applications and, 13
 - Surf, 282, 286
- architecture (Alfresco), 19–42
 - basic principles, 19–21
 - basics of, 7–8
 - community and, 21
 - Content Application Server. *See* Content Application Server
 - content modeling, 33–35
 - deployment options, 37–40
 - enterprise infrastructure, 40–41
 - overview, 21–23
 - summary, 41–42
 - Web Application Framework, 35–37
- archive action
 - adding to document library, 423
 - archive action handler, 422
- Archived documents and folder, 350–351
- Archived space, adding to space templates, 348
- Archives, content service mappers, 492–499
- archiving documents, 433–434
- args root object (Web scripts), 199
- args/argsM root objects, 243
- article patterns, content service mappers, 464–467
- articles
 - article approval workflow (repository), 435–437
 - article images, adding Knowledge Base, 426
 - in content repository, creating, 96–99
 - Knowledge Base articles basics, 336–338
 - Knowledge Base articles, editing, 354–355
- aspects
 - assignment aspects (tasks), 151
 - availability of in document library, defining, 441
 - content metamodels, 79, 90–92
 - content models, 315–318
 - document library (Share), 415–416
- assets, 57
- assignment aspects (tasks), 151
- Association for Information and Image Management (AIIM), 5
- associations (content metamodels), 79, 85–87
- Atom
 - Format Reader, 247
 - Publishing Protocol (AtomPub), 68–69, 247
 - subscription feed example, 502–503
 - Syndication Format, 247
- atomic indexing, defined, 85
- attachments (content service mappers), 487–488
- attributes
 - base start task definition, 150
 - base workflow task definitions, 149–150
 - start task, 150–151
 - task attributes, 152
- Audit service, 55
- authentication
 - authentication Web scripts (RESTful API), ON.7–8
 - <authentication> element (Web scripts), 193–194
 - authentication-related zones, 117
 - basics of, 103–104, 114
 - chains. *See* chains (authentication)
 - configuration examples. *See* examples (authentication)
 - integrating with Alfresco and, 457

- Alfresco support for, 455
- basics of, 67–68
- building portable applications and, 10
- CMIS Spaces example, 484
- content repositories and, 22, 24
- core concepts, 69–72
- Liferay portal using (example), 505–507
- relationship to content modeling, 101–102
- requirements, 68–69
- services, 72–73
- Surf and, 286
- using, 73–74
- code
 - downloading from book’s page, 334
 - installing for Knowledge Base applications, 343–345
 - code, for Knowledge Base customizations, 391–412, 434–450
 - repository, configuring, 434–441
 - Share, 441–449
 - Share common library, 403–406
 - Share configuration files, 406–407
 - Share custom dashlet, 391–394
 - Share custom page, 394–401
 - Share custom site preset, 401–402
 - Share message bundle, 407
 - Tomcat ROOT Web application files, 408–412, 449–450
 - code, installing for Knowledge Base
 - customizations, 421–426
 - common library, adding, 383
 - custom dashlet, adding, 380
 - custom message bundles, adding, 384
 - custom site page, adding, 380–382
 - custom site preset, configuring, 383–384
 - document library services, 422–423
 - files, adding to application, 384–385, 425–426
 - message bundle bootstrap component, overriding, 384
 - server, stopping, 380, 422–428
 - Share, customizing, 423–425
 - code, Knowledge Base applications, 356–365
 - Data Dictionary script, 361–362
 - Explorer, 359–361
 - repository, 356–359
 - search Web script, 362–365
 - collaboration services, 27–28, 57–58
 - Commenting service, 58
 - Comments Web scripts (RESTful API), ON.17–18
 - common task models, 149–153
 - component bindings, adding (Surf), 382
 - Component instances (Surf), ON.111–113
 - Component Type (Surf), ON.110–111
 - components
 - Knowledge Base and Share customization, 374
 - message bundle bootstrap component, overriding, 384
 - root-scoped objects and (Surf platform), ON.59
 - Surf Framework, 299–300
 - Surf objects, files for, 395–396
 - Web script component files, 193
 - Web script components, 190–191
 - config Surf platform root-scoped object, ON.48–49
 - config.script root object, 234
 - configuration files, code for (Share), 406–407, 441–443
 - configuration objects (Surf), ON.111–113
 - configuring
 - applications, 13
 - for custom site pages (Share), 376
 - custom site presets, 376–379, 383
 - dashboards (Share custom site presets), 378
 - Explorer for Knowledge Base applications, 344
 - instances for subsystems (authentication), 106–107, 109–110
 - Knowledge Base dashlet, code for, 394
 - properties file for subsystems instances, 109
 - public services, 124
 - repositories, 45
 - repository, 434–441
 - servers for process definition, 144–145
 - services, 61–63
 - Share, for custom site pages, 376
 - Share, forms and, 419–421
 - Share forms for kb:article objects, 424
 - for showing workflow, 423–424
 - site pages for Share, 382
 - task property pages, 162–164
 - Web scripts, 231–235

- Confluence examples, 463–464, 508
- connectors (Surf platform), ON.57–58
- connectors and credentials, 306–312
 - authenticators, 308–309
 - connectors and endpoints, 306–307
 - credentials, 307–308
 - example, 310
 - remote API, 309
- constraints
 - content metamodels, 87–89
 - content modeling, 75–76
 - content models, 315–316
 - Knowledge Base applications, 338–339
- content
 - behavior of, repository and, 316–318
 - content items (Explorer), 323
 - fetching via HTTP, 355–356
 - lifecycle of, 334–335
 - management of, 12
 - Surf and, 282–287
- Content Application Server, 23–33
 - APIs, 30–33
 - basics of, 7, 8, 23–24
 - clustering and, 38–39
 - content repository, 24–26
 - content services, 27–28
 - deployment options, 38
 - Embedded API, 32–33
 - modularity of, 26–27
 - protocols, 28–30
 - Remote API, 30–31
 - Web scripts and, 167
- Content Application Services, 56–58
- content applications (Surf Framework), 288–290
- Content Associations (Surf platform), ON.114–115
- content delivery services (Surf Framework), 287–288
- content management. *See also* Enterprise Content Management (ECM)
 - vs. databases, 453–455
 - vs. file systems, 453–455
 - reasons for integrating into applications, 452–455
 - Surf and WCM, 286–287
 - WCM, 5, 11
- content management integration patterns, 457–507
- content service mappers. *See* content service mappers
 - example integrations, 507–511
- Content Management Interoperability Services. *See* CMIS (Content Management Interoperability Services)
- content metamodels, 78–92
 - aspects, 90–92
 - associations, 85–87
 - basics of, 78–79
 - constraints, 87–89
 - four-layer metadata model, 76–77
 - inheritance, 89–90
 - model headers, 80
 - model namespaces, 81–82
 - properties, 84–85
 - types, 82–83
 - XML schema and, 80
- content modeling, 75–102
 - Alfresco architecture and, 33–35
 - basics of, 75–76
 - content, creating with JavaScript, 96–99
 - content repository data structure, 75–76
 - Content Repository Services, 55
 - deploying, 77–78
 - four-layer metadata model, 76–77
 - metamodels. *See* content metamodels
 - model localization, 99–100
 - out-of-the-box models, 92–96
 - relationship to CMIS, 101–102
- content models
 - defining (Knowledge Base), 356–358
 - Knowledge Base applications, 335–339
 - Knowledge Base, adding, 344
 - repository, 315–316
- content negotiation, 235–238
- content repository
 - basics of, 24–26
 - data structure, 75–76
- Content Repository Services, 54–56
- content rules, adding to Knowledge Base space, 348–350
- content service mappers
 - annotations, 488–492
 - Archives, 492–499

- article patterns, 464–467
- attachments, 487–488
- basics of, 458
- browser views, 472–482
- Catalog views, 484–487
- example of, 459–462
- feeds, 501–503
- Librarian browser, 482–484
- Portlet views, 503–507
- Project Spaces, 499–500
- Property views, 462–464
- Query views, 467–472
- Review and Approval workflow, 500–501
- using, 459
- content services
 - Alfresco architecture and, 27–28
 - Content Application Services, 56
 - tier, 286
- content Surf platform root-scoped object, ON.46–48
- content types (content models), 315–316
- content-centric Web sites, Surf and, 282
- context Surf platform root-scoped object, ON.44–45
- Control services, 27–28, 57
- controller scripts, creating, 197–202
 - basics of, 197–198
 - debugging, 208–209
 - model construction, 202
 - parsing Web script URIs, 198–199
 - response status code, setting, 201
 - services, calling, 199–200
- controllers
 - dashlets Web script controller, code for, 393–394
 - MVC, 289–290, 292
 - scriptable (Knowledge Base), 374
 - scriptable (Surf Framework), 310
 - scriptable, for POST action handler Web script, 439–440
- controls, 420–421
- Counter Service examples, 64–67
- country codes, 231
- Create Content Wizard, 353
- Create Rule Wizard, 349
- Create Site Wizard, 329–330
- Create Space Wizard, 347

- createAssociation method (JavaScript), 99
- createNode method (JavaScript), 98
- create-site dialog drop-down list,
 - overriding, 383–384
- create-site Web script, overriding, 383–384
- creating
 - resource bundles, 159–161
 - task models, 147–148
- credential vaults, 307–308
- credentials, 307–308. *See also* connectors and credentials
- CSS
 - custom, used by document library, 450
 - custom pages, 408–409
 - for Knowledge Base custom page, 408–409
- cURLs, as Web script clients, 176
- current state (Knowledge Base articles), 334
- current status (Knowledge Base articles), 334
- custom constraints, 88
- customization and extension, 313–332
 - Alfresco Explorer. *See* Explorer
 - customization extensions, 47
 - overview of, 313–315
 - repository. *See* repository
 - Share. *See* Share
- customizing. *See also* Knowledge Base, Share advanced customizations; Knowledge Base, Share customizations; process behavior, customizing
 - applications, 13
 - content models, 92–96
 - process behavior. *See* process behavior, customizing

D

- dashboards
 - configuring (Share custom site presets), 378
 - Explorer, 322
 - Share, 328–329, 331–332
- dashlets
 - custom (CSS), 408
 - custom, adding to sites, 387–388
 - default properties bundle, 394
 - Explorer, 322
 - Knowledge Base, configuration (code for), 394

- Share, 332
- Share custom, 370–372
- Share custom, adding, 380
- Share custom, adding dependencies, 385
- Share custom, code for, 391–394
- Web script controller, code for, 392–393
- Web script HEAD markup, 392
- Web script implementation (Knowledge Base), 391
- Web script view, code for, 393–394
- Data Dictionary, adding scripts to, 345–346
- Data Dictionary script, files for copying, 361–362
- data Web scripts, 169
- databases, vs. content management, 453–455
- date (current) in templates, ON.74
- d:content type properties, 98
- debugging
 - client-side JavaScript, 49–50
 - controller scripts, 208–209
 - server-side JavaScript, 49
 - Web scripts, 176
- decision nodes, 135
- DeclarativeWebScript Java class, 264, 273
- default process variables, 155
- definitions
 - process definitions, 143–146
 - workflow definitions, 141–143, 160
- Delete Preferences Web script (RESTful API), ON.33
- delivery environments (Surf Framework), 287
- dependencies
 - browsers (document library), 449–450
 - custom page component, adding, 385
 - document library, adding (Share), 425–426
 - Share dashlets and, 371, 385
 - Share site pages, 374–375
- Dependency Injection (DI), 265–266
- deploying
 - content models, 77–78
 - extensions, 45–48
 - hot-deploying process definitions, 145–146
 - process definitions, 143–144
 - resource bundles, 161–162
 - task models, 153
 - testing, process definitions, 146
- deployment
 - options, 38
 - Surf Framework, 287
- description documents, creating, 191–197
 - basics of, 191–195
 - URI templates, 195–197
- <description> element (Web scripts), 193
- descriptor cache controls, 250–251
- descriptor files, defined, 374
- descriptor options (Web scripts), 254–255
- developers, Alfresco and, 4
- development tools (Surf), 282
- DI (Dependency Injection), 265–266
- Dictionary Web scripts (RESTful API), ON.18–21
- directory servers, integrating Alfresco with, 108
- directory services, integrating with Alfresco and, 457
- Discussion Web scripts (RESTful API), ON.21–24
- dispatchers (MVC), 289–291
- display page, configuring for Alfresco content (Surf example), ON.115
- document library (Share)
 - action list, 446–447
 - actions, 416–417
 - archive action, adding, 423
 - aspects, 415–416
 - availability of aspects in, defining, 441
 - basics of, 414–415
 - browser dependencies, 449–450
 - dependencies, adding, 425–426
 - HEAD override, 444–446
 - initializing, 391
 - pages, 330, 331
 - services, code for customizing, 422, 440–441
 - services, POST action handler for, 438
 - workflow, 417–418
- document objects (CMIS), 70, 71
- documentLibrary subfolder, 391
- documents
 - adding to Knowledge Base, 352–354, 429–430
 - approval for, 431–433
 - Archived, moving to Archived folder, 350–351
 - archiving, 433–434
 - management of, 5, 10–11

- metadata, editing, 430–431
- updating status tags (Knowledge Base space), 350–351
- Documentum, 3, 6
- Down, Tim, 50
- downloads
 - Drupal Content Construction Kit (CCK), 469
 - extending
 - `RuntimeExecutableContentTransformer`, 62
 - Installing and Configuring Alfresco* guide, 43
 - Java API to create new content example, 59
 - JavaScript API to create new content example, 60
 - model headers, 80
 - source files for Knowledge Base applications, 334
 - source files for Knowledge Base customizations, 368
 - Spring Surf releases, 311
- draft status (Knowledge Base articles), 334
- drives, shared drive replacement, 11
- Drupal
 - examples, 469–472, 474–482
 - integrating with Alfresco (example), 507–508
 - interface, 73
- dynamic approach to deploying content models, 77
- dynamic authorities and roles, 115, 117–118

E

- E4X API, 234
- ECM. *See* Enterprise Content Management (ECM)
- ECM Services for IBM Lotus Quickr, 459
- editing Knowledge Base articles, 354–355
- email clients, example of integrating, 511
- embedded Alfresco, 38
- Embedded APIs
 - Content Application Server, 32–33
 - services and, 59–61
- Embedded Java API, tiers and, 58
- endpoints
 - remote API and, 309
 - Surf Framework, 306–307
- end-state nodes, 135–136
- enforced mandatory element (content modeling), 85

- Enterprise Content Management (ECM)
 - Alfresco support of, 20–21
 - basics of, 5–7
 - repository, Web scripts and, 168
- enterprise infrastructure, 40–41
- enterprise portals, 11
- entity tags, 250
- environment independence, 20
- environments, setting up, 43–50
 - Alfresco, installing, 43–45
 - debugging, 48–50
 - extensions, deploying, 45–48
- errors, reporting error status codes (Web scripts), 201
- ETags (entity tags), 250
- event handlers (browser-side), 449–450
- Event Listing Web scripts (RESTful API), ON.7
- events, jBPM, 137–138
- examples. *See also* Knowledge Base Search Web script example
 - ACLs, 121–123
 - annotated controller (Surf Framework), 293–294
 - annotations, 490–492
 - Archives, 494–499
 - article patterns, 466–467
 - Atom subscription feed, 502–503
 - attachments, 488
 - browser views, 472–482
 - Catalog views, 484–487
 - child Page association (Surf), ON.116
 - Chrome for component binding (Surf), ON.109–110
 - Chrome for regions (Surf), ON.108–109
 - CMIS Spaces, 484
 - Confluence, 463–464, 508
 - connectors and credentials (Surf Framework), 310
 - content service mappers, 459–462
 - custom page scope binding (Surf), ON.113
 - display page, configuring for Alfresco content, ON.115
 - Drupal and OpenSearch, 469–472
 - Drupal Integration examples, 474–482, 507–508

- feeds, 502–503
 - Folder Listing Java-backed Web script example, 258–263
 - Folder Listing Web script example, getting started with, 259–263
 - FreeMarker Template processor (Surf), ON.120
 - global scope binding (Surf), ON.113
 - Hello User Web script, 177–181, 185–186
 - Hello World Web script, creating, 171–173
 - image gallery, 486–487
 - integrating email clients, 511
 - Java-backed Web script (Surf Framework), 304–305
 - Joomla, integrating with Alfresco, 507–508
 - jPDL process definition example, 143–144
 - Kofax image scanning, 509
 - Kofax release script, 494–499, 501
 - Librarian, 484
 - Liferay portal, 508
 - Liferay portal using CMIS, 505–507
 - login Page Type with custom property (Surf), ON.117
 - Lotus Quickr, 511
 - Microsoft SharePoint, 509–510
 - page (Surf Framework), 292–293
 - page scope binding (Surf), ON.112
 - Page with authentication, ON.118–119
 - Portlet views, 503–507
 - Property views, 463–464
 - Query views, 469–472
 - resource bundles, deploying, 161
 - Review and Approval workflow, 501
 - sample custom Component Type (Surf), ON.111
 - SAP ArchiveLink, 488, 509
 - site configuration (Surf), ON.114
 - Template Instance with configuration (Surf), ON.119
 - template scope binding (Surf), ON.112–113
 - Theme that overrides login Page Type, ON.121
 - Web script Component Type (Surf), ON.111
 - examples (authentication), 105–114
 - alfrescoNtlm, customizing, 105–108
 - ldap-ad subsystem, 108–111
 - passthru subsystem, 111–114
 - exceptions, Web Script Framework and, 267
 - execute() method, 263
 - executeImpl() template method, 264–268
 - Explorer
 - adding documents to Knowledge Base and, 352–354
 - additional extension points, 328
 - basics of, 8, 21–22, 314–315, 321–324
 - code, for Knowledge Base, 356–358
 - configuring for Knowledge Base applications, 344
 - extensions, Knowledge Base applications, 339–341
 - permissions, 325
 - rules, 325–327
 - signing on to for Knowledge Base applications, 345
 - Explorer Web client, configuring, 159–164
 - resource bundles, creating, 159–161
 - resource bundles, deploying, 161–162
 - task property pages, configuring, 162–164
 - extending services, 61–63
 - extension points
 - additional for Explorer, 328
 - additional for repository, 321
 - for customizing behavior, 153
 - extensions
 - deploying, 45–48
 - Knowledge Base applications, 339–341
 - scriptable, 21
 - external subsystem, 104
- ## F
- family option, 255
 - FeedControl object, ON.99
 - feeds
 - content service mappers and, 501–503
 - integrating with Alfresco and, 456
 - listing children of documents as, 460–462
 - File Folders service, 54
 - file systems, vs. content management, 453–455
 - File Transfer Protocol (FTP), 30
 - File Upload Web scripts (RESTful API), ON.40
 - file-based access, integrating with Alfresco and, 456
 - FileNet, 6

- files
 - adding to Tomcat ROOT Web application, 384–385, 425–426
 - extension files, packaging, 48
 - management of (CMIS), 54
 - script files, 76
 - script for copying to repository, 361–362
 - Search Web script files for copying, 362–364
 - template files, ON.61
 - Folder Listing Java-backed Web script
 - example, 258–263
 - Folder Listing Web script example, 259–263
 - folderpath token, 260
 - folders (CMIS)
 - management of, 54
 - structure of, 71
 - ForEachFork action, 158
 - forks/fork nodes, 135, 136
 - Form Data API, 244–245
 - form processing
 - basics of, 238–244
 - Form Data API, 244–245
 - format objects (Web scripts), 224
 - Format Readers, 247, 248
 - <format> element (Web scripts), 195
 - formats
 - ACLs, 123
 - response templates, 203–204
 - formdata root object, 243, 244
 - formfield root object, 244–245
 - forms
 - form template (sample code), 443–444
 - Forms Web scripts (RESTful API), ON.24–25
 - in Share, 418–421, 424
 - Forms engine, 414, 418–421
 - FreeMarker. *See also* repository FreeMarker
 - template API
 - methods, 206
 - processing XML with, 235
 - response templates and, 190, 202
 - Template processor (Surf example), ON.120
 - templates, customizing Knowledge Base with, 381–382
 - templates for Surf, 294
 - FreeMarker Template API, ON.43
 - FTP (File Transfer Protocol), 30
 - full content services configuration, 286
 - functions
 - for Knowledge Base (code), 403–406
 - subsystems (authentication), 104
- ## G
- Get Associations Web scripts (RESTful API), ON.18–21
 - GET handlers (dashlet Web scripts), 332
 - getDescription() method, 263, 264, 273
 - global scope (Surf Framework), 297, 300, ON.113
 - group objects, ON.102
 - GROUP_ prefix, 122
 - groups
 - defined, 115
 - permission groups, 119
 - root groups, 116
 - security and, 116
- ## H
- hasPermission() check, ON.80
 - HEAD markup, Knowledge Base dashlet (code), 393–394
 - head Surf platform root-scoped object, ON.55–56
 - Hello User Web script, 177–181, 185–186
 - Hello World Web script, creating, 171–173
 - hook points (Drupal integration), 479–482
 - hot backup, defined, 39
 - hot-deploying process definitions, 145–146
 - htmlid Surf platform root-scoped object, ON.55
 - HTTP
 - Accept header, 235
 - Accept-Language header, 231, 235
 - Basic authentication, 179
 - caching and, 250
 - fetching content via, 355–356
 - HTTP Accept header, 182
 - methods, and initiating Web scripts, 193
 - tunneling methods (Web clients), 184–185
 - Web script applications and caching and, 186–187
 - Web script client limitations, 184–186

HTTP GET method, 242
 <http method>, 193
 HTTP POST method, 242

I

I18N (Share)
 custom dashlets and, 371–372
 custom site pages and, 375–376
 custom site presets and, 378–379
 icons for Knowledge Base (Explorer), 341
 images
 article images, adding Knowledge Base, 426
 image gallery example, 486–487
 management of, 5
 IMAP (Internet Message Access Protocol)
 basics of, 30
 IMAP Web scripts (RESTful API), ON.29–30
 importing scripts, ON.76
 index element (content modeling), 85
 indexes (Web scripts), 173–175, 255
 Information Publishing, 11
 inheritance
 content metamodels, 89–90
 content types and aspects and, 315–316
 initiator swimlanes, 157
 Installation Wizard, 44–45
 installing Alfresco, 43–45, 263
Installing and Configuring Alfresco guide,
 14, 43, 45
 installLocation, 44
 instance Surf platform root-scoped object, ON.49
 instances
 configuring for subsystems, 106–107, 109–110
 subsystems (authentication), 105
 internationalization
 custom dashlets and (Share), 371–372
 custom site pages and (Share), 375–376
 custom site presets and (Share), 378–379
 of Web scripts, 227–231
 Internet Message Access Protocol (IMAP)
 basics of, 30
 IMAP Web scripts (RESTful API), ON.29–30
 intranets. *See* enterprise portals
 Invite service, 58
 IT organizations, 4–5

J

Java
 accessing Alfresco core API with, 456–457
 Alfresco basics and, 4, 20–21
 Share and, 10
 Java Content Repository (JCR), 22, 24
 Java Web scripts, 263–268
 calling Alfresco services, 265–266
 fundamentals, 263–264
 model construction, 268
 parsing URIs, 265
 response status code, setting, 266–267
 Java-backed Web scripts, 257–277
 basics of, 257–258
 Folder Listing Java-backed Web script example,
 258–263
 Java approach to Web scripts. *See* Java Web
 scripts
 vs. scripted Web scripts, 257–258
 Spring Framework registration, 268–270
 Surf Framework example, 304–305
 Web scripts, creating new type of, 270–275
 Web scripts, using new type of, 275–277
 JavaScript
 action, 157–158
 articles in content repository, creating, 96–99
 code to define functions for Knowledge Base,
 403–406
 controller scripts and, 190
 custom pages, 408, 410–412
 logging and debugging, 48–50
 JavaScript API
 articles, creating with, 96–99
 counter service example, 64–65
 as embedded API, 59
 FreeMarker API and, ON.43
 tiers and, 58
 JavaScript Debugger, 208–209
 jBoss Business Process Management (jBPM). *See*
 jBPM, processes and
 jBoss jBPM engine, 133–134
 jBoss jBPM Process Designer, 143
 jBPM, processes and, 134–141
 actions and events, 137–138
 adapter, 134

- basics, 134–135
- nodes and transitions, 135–137
- super-states and sub-processes, 139–141
- tasks, swimlanes and timers, 138–139
- jBPM engine, 133–134
- jBPM Process Designer, 143
- JCR (Java Content Repository), 22, 24
- jDPL User Guide*, 143
- Joomla, integrating with Alfresco
 - example, 507–508
- jPDL process definition example, 143–144
- JS reference, customizing document library
 - and, 445–446
- JscriptWorkflowDefinition, ON.104
- JscriptWorkflowInstance, ON.104–105
- JscriptWorkflowNode, ON.105
- JscriptWorkflowPath, ON.105
- JscriptWorkflowTask, ON.106
- JscriptWorkflowTransition, ON.106
- JSON (JavaScript Object Notation)
 - aspects, document library and, 415–416
 - basics of, 180, 194
 - callbacks (Web scripts), 185–186
 - Format Reader, 247
 - Knowledge Base articles and, 357
 - multiple response template support for, 209–210
- JSON Object API, 249–250
- json root object, 248
- JSONArray object type, 249
- JSONObject object type, 249–250
- JSP pages, rendering templates and, ON.74

K

- kb:article aspect, 336–338, 348–350, 423
- kb:article objects, 424
- kb:space, 336
- kb:status property, 418–419
- kerberos subsystem, 104
- key value pairs, content models and, 100
- keywords, Web scripts (RESTful API), ON.34
- Knowledge Base applications, foundation
 - for, 333–365
 - articles, editing, 354–355
 - code (sample). *See* code, Knowledge Base applications

- code, installing, 343–345
- content, fetching, 355–356
- content lifecycle, 334–335
- content model, 335–339
- documents, adding, 352–354
- Explorer, signing on to, 345
- Explorer extensions, 339–341
- Knowledge Base, creating, 351–352
- scripts, adding to Data Dictionary, 345–346
- server, starting, 345
- smart spaces and scripting, 341–342
- space templates, 342–343
- space templates, creating. *See* Knowledge Base
 - space templates, creating
- Knowledge Base Search Web script example
 - creating, 213–218
 - functioning of, 218–219
 - testing, 219–222
- Knowledge Base, Share advanced
 - customizations, 413–450
 - browsing to Knowledge Base, 428
 - code, installing. *See* code, installing for Knowledge Base customizations
 - document approval, requesting, 431–433
 - document library. *See* document library (Share)
 - document metadata, editing, 430–431
 - documents, adding, 429–430
 - documents, archiving, 433–434
 - forms in Share, 418–421
 - Knowledge Base, setting up, 426–428
 - overview of, 413–414
 - sample code. *See* code, for Knowledge Base customizations
 - server, restarting, 426
- Knowledge Base, Share customizations, 367–412
 - basics of, 367–368
 - code (sample). *See* code, for Knowledge Base customizations
 - code, installing. *See* code, installing for Knowledge Base customizations
 - custom dashlet, 370–372
 - custom dashlets, adding to sites, 387–388
 - custom pages, adding to sites, 388–389
 - custom site pages, 373–376

- custom site presets, 376–379
- Document Library, clicking on, 391
- server, starting, 385
- Share site, creating, 386
- sites, 368–369
- sites, creating, 389–391
- Knowledge Base space templates, creating
 - Archived folder, moving documents to, 350–351
 - Archived space, adding, 348
 - content rules, setting up, 348–350
 - status tag, updating, 351
 - template, creating, 347–348
- knowledge management, 10
- knowledgebase-action.js file, 417
- knowledgebase.lib.js file, adding, 383
- Kofax
 - image scanning example, 509
 - release script example, 501

L

- languages
 - scripting, and Web script creation, 189
 - use of Alfresco and, 455
- languages (spoken). *See also* internationalization
 - dialect support and, 372
 - language codes, 231
 - model localization and, 99
 - resource bundles and, 227–231
- ldap subsystem
 - basics of, 104
 - example, 108–111
- Librarian browser (content service mappers), 482–484
- libraries
 - knowledgebase.lib.js file, adding, 383
 - third party, 20
- lifecycle option, 254–255
- lifecycle state (Knowledge Base articles), 334
- Liferay portal examples, 505–507, 508
- locale Surf platform root-scoped object, ON.55
- localization of models, 99–100
- logging
 - client-side JavaScript, 49–50
 - server-side JavaScript, 48–49
- Logging API, ON.91–92

- logic (Web scripts), 205
- login
 - login Page Type with custom property (Surf example), ON.117
 - login Web scripts (RESTful API), ON.7–8
 - Theme that overrides login Page Type, 15
- logout Web scripts (RESTful API), ON.7
- Lotus Quickr example, 511
- Lucene, defined, 8

M

- management. *See* business process management
- mapping content metamodels to CMIS, 101
- message bundles
 - adding custom, 384, 424–425
 - code defining elements for (Share), 407
 - overriding bootstrap component, 384, 425
- metadata
 - document metadata, editing, 430–431
 - Edit form for (Share), 443–444
 - Metadata Web scripts (RESTful API), ON.30
- metadata (content modeling), 75–76
 - basics of, 76
 - four-layer metadata model, 76–77
- Metadata Extraction services, 56
- metamodels. *See* content metamodels
- methods
 - Category node objects and, ON.69–70
 - configuring for security and, 124, 126
 - default model, ON.73–74
 - FreeMarker, 206
 - helper, for accessing CMIS calls, 474–479
 - helper methods, TemplateNode Model API, ON.65–67
 - HTTP, initiating Web scripts and, 193
 - HTTP, tunneling and Web clients and, 184–185
 - method-level security, 124–126
 - working with endpoints using, 309
- Microsoft SharePoint
 - integration example, 509–510
 - protocol, 30
- MIME types, content negotiation and, 235, 236, 238
- model headers
 - basics of, 80

defining content models and, 93
 model objects (Surf platform), ON.56–57
 model root object (controller scripts), 202
 models. *See also* content modeling
 accessing (Web scripts), 204–205
 constructing (controller scripts), 202
 constructing (Java-backed Web scripts), 202
 defined (controller scripts), 202
 defined (Web scripts), 171
 model construction, 268
 template models. *See* template models
 workflow models, repository, 319–320
 Model-View-Controller (MVC)
 basics of, 289–290
 Spring Web MVC, 282, 289–291
 Web scripts and, 170
 modularity
 of Alfresco, 20
 of Content Application Server, 26–27
 msg method, 229
 msg Surf platform root-scoped object, ON.56
 multipart/form-data form submits, 239, 243
 multiple response templates
 basics of, 209–211
 response status code templates, 211–213
 multitenancy, Content Application Server
 and, 39–40
 MVC. *See* Model-View-Controller (MVC); Spring
 Web MVC

N

namespaces (content models), 81–82, 315
 naming
 configuration files, 234
 content models, 81–82, 93
 controller scripts, 198
 permissions, 118
 response template files, 203
 Web script component files, 193
 .NET applications, integrating Alfresco with, 457
 no-compile philosophy, 283
 Node Metadata Retrieval/Storage Service Web
 scripts (RESTful API), ON.30
 nodes
 in content modeling, 75–76, 79

 jBPM, processes and, 135–137
 in repository, 323
 Web scripts and, 270
 workflow nodes, 160
 NodeWebScript kind, 275, 277
 NTLM SSO function, 112
 null values, FreeMarker template engine and,
 ON.63

O

object-oriented level APIs, ON.75–76
 objects. *See also* Surf quick reference
 model objects (Surf platform), ON.56–57
 presentation objects (Surf Framework), 301
 rendering objects (Surf platform), ON.58–59
 root objects reference (Web scripts), 222–225
 Surf Framework, 301
 onActionArchive action, 416–417
 open source software
 Alfresco and, 14–17
 benefits of, 15
 OpenSearch
 example, 469–472
 integrating with Alfresco and, 456
 org.alfresco.web.scripts
 .AbstractWebScript helper Java class,
 263
 org.alfresco.web.scripts
 .DeclarativeWebScript helper Java
 class, 264
 org.alfresco.web.scripts
 .DescriptionExtension interface, 274
 org.alfresco.web.scripts.WebScript Java
 interface, 263
 out-of-the-box content models, 92–96
 OwnableService, 129

P

packages, workflow actions and, 151
 Page Associations (Surf platform),
 ON.115–116
 page component dependencies, adding custom,
 385
 page root-scoped object (Surf platform), ON.48

- page scope (Surf Framework), 298, 304, ON.112, ON.113
- page templates, defining (Knowledge Base), 396–397
- page-centric Web sites, 282
- pages
 - custom (CSS and JavaScript), 408–412
 - custom, adding to Knowledge Base sites, 388–389
 - display page, configuring for Alfresco content, ON.115
 - files required for (Share), 394–401
 - Surf Framework, 295
 - task property pages, configuring, 162–164
- Pages (Surf platform)
 - basics of, ON.117–119
 - Page Types, ON.116–117
 - Page with authentication (example), ON.118–119
- PageViewResolver (Surf Framework), 292–293
- parent elements (content modeling), 89
- parsing
 - Format Reader and, 247
 - URIs (Java-backed Web scripts), 265
 - Web script URIs, 198–199
- passthru subsystem
 - basics, 104
 - examples (authentication), 111–114
- passwords, Active Directory and, 111
- peer associations (content metamodels), 79
- pending approval status, Knowledge Base articles, 334
- people and users, security and, 115–116
- People API, ON.73, ON.88–90
- permissions
 - access control, modifying and, 127
 - basics of, 114–115, 118–120
 - Explorer, 325
 - Permission and Security ScriptNode API, ON.84
 - Permission Service, 55
 - PermissionService, 128–130
- person nodes, 115–116
- Person Web scripts (RESTful API), ON.30–32
- PersonService, 55, 128–129
- Photo Articles, defined, 465
- PHP applications, accessing Alfresco with, 457
- PickerList/PickerItems Web scripts (RESTful API), ON.25
- plug-in for Spring Framework, 289
- portability, 4, 10, 14
- portals, enterprise, 11
- Portlet views (content service mappers), 503–507
- POST action handler Web script, 438–440
- Posting Blogs Web scripts (RESTful API), ON.11–15
- Preference Web scripts (RESTful API), ON.33
- Prefixes, model namespaces and, 81
- presentation content (Surf Framework), 283–284, 302–303
- presentation tier
 - Share and, 314–315
 - Surf Framework, 285–286
- presentation Web scripts, 170
- presets
 - custom site preset, 376–379
 - custom site preset, configuring, 383–384
 - share object, 401–402
- process behavior, customizing, 153–158
 - basics, 153–154
 - ForEachFork action, 158
 - JavaScript action, 157–158
 - process variables, 154–155
 - tasks and swimlanes, assigning, 156–157
- processes
 - jBPM and. *See* jBPM, processes and
 - process definitions, 143–146, 318–319
- production environments (Surf Framework), 287–288
- programming models, 9–10
- Project Spaces (content service mappers), 499–500
- properties
 - for accessing server metadata, 225
 - for accessing Web script metadata, 224–225
 - common object properties (Surf platform), ON.107
 - content metamodels, 79, 84–85
 - for controlling caching, 224
 - default, for subsystems, 107
 - domain models and, 101
 - for interrogating format objects, 224
 - jBPM, and task attributes, 152
 - JSON (document library), 416
 - process definitions, 145
 - status objects, allowing access to, 223

- on `url` objects, 222
- properties files
 - configuring for subsystems instances, 109
 - subsystems, 107
- property sheets (Knowledge Base), 340
- Property view (content service mappers), 462–464, 473, 489–490
- Protocol bindings, defined, 23
- public services, 124–126

Q

- Quercus PHP interpreter, 457
- query engines, 24
- query parameters, URI templates and, 195, 196
- Query views (content service mappers), 467–472

R

- Records Management
 - access control and, 128
 - authorization, 128
 - basics of, 5, 12
 - module, 492–493
- Refresh Web Scripts button, 175
- region scope (Surf Framework), 297, 300
- regions (Surf Framework)
 - basics of, 296–298
 - components and, 299
- registering
 - Java-backed Web scripts, 268–270
 - Web scripts, 207–208, 210, 218
- relaxed mandatory element (content modeling), 85
- Remote APIs
 - basics of, 23
 - Content Application Server, 30–31
 - services and, 61
 - Surf Framework, 309
- remote Surf platform root-scoped object, ON.54
- Remove an Authority from a Group Web scripts (RESTful API), ON.25
- Remove Calendar Subscriptions Web scripts (RESTful API), ON.5
- rendering objects (Surf platform), ON.58–59
- repository
 - actions, 320–321

- additional extension points, 321
- basics of, 313–315
- configuration of (Knowledge Base), 356–359
- configuring, 45
- configuring (sample code), 434–441
- content behavior, 316–318
- content models, 315–316
- Data Dictionary script files for, 361–362
- interaction with Explorer and Share, 314–315
- process definitions, 318–319
- repository tier, Explorer and, 314–315
- Web scripts (RESTful API), ON.7–9
- workflow models, 319–320
- Repository JavaScript API reference
 - object-oriented level APIs, ON.75–76
 - script files, ON.76
 - scripting API. *See* scripting API
 - Services API. *See* Services API
- request processing (Web scripts), 245–250
- `requestbody` root object, 245, 247
- `requestbody.post.json.js` controller script, 249
- requirements (CMIS), 68–69
- residual properties, defined, 85, 98
- resolvers (Surf Framework), 292
- resource bundles
 - content models, 100
 - creating, 159–161
 - deploying, 161–162
 - languages and, 227–231
- response formats (Web scripts), 180–183
- Response object (Surf platform), ON.58
- response status code, setting
 - controller scripts, 201
 - Java-backed Web scripts, 266–267
- response templates, creating, 202–208
 - accessing models, 204–205
 - accessing services, 205–206
 - basics of, 202–204
 - FreeMarker methods, 206
- ResponseStatus object (Surf platform), ON.58
- repository FreeMarker template API, ON.61–74
 - JSP page, ON.74

- template files, ON.61
- template models. *See* template models
- RESTClient, 176
- RESTful API
 - in Alfresco, 31
 - counter service example, 65–67
 - Surf and, 282
 - tiers and, 59, 65–67
 - Web scripts and, 167
- RESTful API reference, ON.3–41
 - Activities Web scripts, ON.9–12
 - Blogs Web scripts, ON.13–17
 - Calendar Web scripts, ON.3–5
 - Comments Web scripts, ON.17–18
 - Dictionary Web scripts, ON.18–21
 - Discussion Web scripts, ON.21–24
 - Forms Web scripts, ON.22–27
 - IMAP Web scripts, ON.27–28
 - Metadata Web scripts, ON.30
 - Person Web scripts, ON.30–32
 - Preference Web scripts, ON.33
 - repository Web scripts, ON.7–9
 - Search Web scripts, ON.33–36
 - Tagging Web scripts, ON.36–37
 - Thumbnail Web scripts, ON.38–39
 - Upload Web scripts, ON.40
 - Version Web scripts, ON.40–41
- retrieving calendar events (RESTful API), ON.3–5
- return types (Surf platform), ON.56–58
- Review and Approval workflow (content service mappers), 500–501
- review-messages.properites (listing), 161
- RFC 2616, 235
- roles
 - dynamic authorities and, 117–118
 - internal, 317
 - permissions and, 323
- root objects (Web scripts)
 - basics of, 199–200
 - reference, 222–225
 - response templates, 205–206
- ROOT Web application files (Tomcat), adding to, 384–385, 425–426
- root-scoped objects, ON.77–78
- root-scoped objects (Surf platform), ON.58–59
 - app, ON.56
 - basics of, ON.44
 - config, ON.48–49
 - content, ON.46–48
 - context, ON.44–45
 - head, ON.55–56
 - htmlid, ON.55
 - instance, ON.49
 - locale, ON.55
 - msg, ON.56
 - page, ON.48
 - remote, ON.54
 - rendering objects, ON.58–60
 - return types and, ON.56–58
 - sitedata, ON.50–54
 - template, ON.48
 - theme, ON.49
 - url, ON.55
 - user, ON.45–46
- RSS feeds
 - content service mappers and, 501–503
 - integrating with Alfresco and, 456
- rules
 - content rules, setting up (Knowledge Base), 348–351
 - defined, 341
 - Explorer, 325–327
 - Rules and Actions (Alfresco), 29
 - rules and actions (Content Application Services), 56
- Run As option, 255
- runtime cache controls, 251–254
- RuntimeExecutableContentTransformer, 62

S

- Sandbox Object API, ON.94–95
- sandboxes, 57
- SAP ArchiveLink integration examples, 488, 509
- Save Calendar Event Web scripts (RESTful API), ON.5
- scalability, 454
- scaling, Alfresco design and, 20
- scopes (Surf Framework), 297
- script files, ON.76
- scriptable controllers
 - Knowledge Base, 374

- for POST action handler Web script, 439–440
- Surf Framework, 310
- script.config root object, 235
- ScriptContent objects, 245
- scripting API
 - Actions API, ON.91
 - CategoryNode Object API, ON.92–93
 - Check in/Check out ScriptNode API, ON.84–85
 - Classification API, ON.92
 - Logging API, ON.91–92
 - overview, ON.76–77
 - People API, ON.88–90
 - Permission and Security ScriptNode API, ON.84
 - root-scoped objects, ON.77–78
 - Sandbox Object API, ON.94–95
 - ScriptAction API, ON.91
 - ScriptContent API, ON.83–84
 - ScriptNode API, ON.78–83
 - Search API, ON.87–88
 - Session API, ON.92
 - Transformation API, ON.86–87
 - utility methods, ON.95–96
 - Versions ScriptNodeAPI, ON.85–86
 - WCM Asset Object API, ON.95
 - WCM web projects, ON.93–96
 - WebProject Object API, ON.93–94
- scripting languages, 189
- scripts. *See also* controller scripts, creating; Web scripts
 - adding to Data Dictionary, 345–346
 - importing, 76
 - Surf and, 282
- Search API, ON.87–88
- Search Engines lists Web scripts (RESTful API), ON.35
- Search service (Content Repository Services), 56
- Search Web scripts
 - Knowledge Base files for, 362–364
 - Restful API, ON.33–36
- searches, for Web scripts, 191
- security. *See also* authentication; permissions
 - access control extension, 128
 - Access Control Lists (ACLs), 120–123
 - access control, modifying, 127–128
 - authorities, 115–118
 - authorization, 114
 - basics, 114
 - method-level security, 124–126
 - permissions, 118–120
 - public services and, 124–126
 - services, 128–131
- semantic content (Surf Framework), 283–285
- server objects (Web scripts), 225
- servers
 - configuring for process definition, 144–145
 - directories, integrating Alfresco with, 108
 - starting, and Knowledge Base applications, 345
 - starting, and Knowledge Base customizations, 385, 426
 - stopping, and Knowledge Base applications, 343
 - stopping, and Knowledge Base customizations, 380, 422–428
- services, 53–74
 - accessing (Web scripts), 205–206
 - available to controller scripts, 200
 - building, 58–59, 63–67
 - calling (Java-backed Web scripts), 265–266
 - CMIS. *See* CMIS (Content Management Interoperability Services)
 - configuring and extending, 61–63
 - Content Application Server, 23
 - Content Application Services, 56–58
 - content management systems and, 454–455
 - Content Repository Services, 54–56
 - content services, 27–28
 - controller script access to, 206
 - controller scripts and, 199–200
 - dependencies, 269–270
 - embedded APIs, 59–61
 - provided by Service Provider Interfaces, 458
 - public services, 124–126
 - remote APIs, 61
 - security, 128–131
 - tiers and, 58–59
- Services API
 - Activities service, ON.98–99
 - Authority service, ON.101–102
 - FeedControl object, ON.99
 - Site service, ON.96–99
 - Tagging service, ON.99–100

- Thumbnail services, ON.100–101
- Workflow services, ON.103–106
- Session API, ON.92
- Set Calendar Color Web scripts (RESTful API), ON.6
- setRepository method, 266
- Share. *See also* Knowledge Base, Share advanced customizations; Knowledge Base, Share customizations
 - basics of, 9, 21–22, 314–315, 328–331
 - collaborative content management and, 12
 - customization extensions, 47
 - dashboards, 331–332
 - dashlets, 332
 - Java and, 10
 - sites. *See* sites (Share)
- Share object, preset, 401–402
- share-config-custom.xml file, 419–420
- <shortname> element (Web scripts), 193
- site pages (Share). *See also* document library (Share)
 - browser dependencies, 373–376
 - configuring, 382
 - configuring Collaboration Site for, 386
 - custom, 373–376, 380–382
 - site functionality and, 369
- site presets
 - custom, 376–379, 383, 401–402
 - defined, 377
 - presets.xml file and, 377
 - using, 389–391
- Site service, ON.96–99
- sitedata Surf platform root-scoped object, ON.50–54
- sites
 - collaboration services and, 57
 - configuration (Surf example), ON.114
- sites (Share)
 - basics of, 368–369
 - creating new, 386, 389–391
 - Share module site creation, 402
 - site spaces as, 368
- smart spaces, 343
- software, open source, 14–17
- SourceSense wiki Property View, 463–464
- space templates (knowledge Base applications), 342–343
- spaces
 - basics of (Explorer), 323–324
 - CMIS Spaces, 484
 - defined, 335, 341
 - Knowledge Base spaces basics, 335–336
 - setting up (Knowledge Base), 426–428
 - smart spaces and scripting (Knowledge Base applications), 341–342
- Spring
 - configuration extensions, 46
 - Spring beans/overrides for Share, 407
 - Spring Roo Add-on for Surf, 289
 - Spring Travel, Spring Pet Clinic Sample Sites, 289
 - SpringSource Tool Suite Plug-in for Surf, 289
- Spring Framework
 - Alfresco and, 26–27
 - basics of, 258
 - plug-in for, 289
 - registering Java-backed Web scripts, 268–270
- Spring Surf. *See* Surf Framework; Surf platform
- Spring Surf Extension project, 310
- Spring Web MVC, 282, 289, 290–291
- Squid, 250
- Stallman, Richard, 14
- standalone constraints, 87–89
- standalone Surf configuration, 285–286
- standards
 - Alfresco and, 21
 - Content Application Server and, 22
- Start Advanced Workflow Wizard, 146
- start task, 150–151
- start task process variables, 156
- start-state nodes, 135, 136
- state nodes, 135, 136
- status
 - Knowledge Base articles, 334
 - status objects (Web scripts), 223–224
 - status root objects, 201
 - status values, custom dashlets, 370
- storage engine, 24
- sub-processes
 - defined, 141
 - jBPM, processes and, 139–141
- subsystems (authentication)
 - authentication chains and, 105
 - basics of, 104, 106

- configuring instances for, 106–107
- defined, 104
- types, 104–105
- super-states, jBPM, 139–141
- support, Enterprise version and, 14
- Surf
 - basics of, 35–37
 - component bindings, adding, 382
 - components, integrating with Web frameworks and, 457
 - object page, file for, 394–395
 - object template instance, file for, 395
 - page, adding to customize Knowledge Base, 381
 - Surf Extension project, 310
 - template instances, adding to customize Knowledge Base, 381
- Surf Framework, 283–311
 - annotated controller example, 293–294
 - basics of, 9, 281–283
 - components, 299–300
 - connectors and credentials. *See* connectors and credentials
 - content, 283–287
 - content applications, 288–290
 - content delivery services, 287–288
 - getting involved, 310–311
 - Java-backed Web script example, 304–305
 - MVC, 289–290
 - objects, 301
 - page example, 292–293
 - pages, 295
 - plug-in for Spring Framework, 289
 - presentation content basics, 302–303
 - Share and, 328
 - Spring Web MVC, 290–291
 - Surf view resolvers, 291–292
 - templates, 302
 - templates and regions, 296–298
 - view composition basics, 294
 - Web scripts, 302–303
- Surf platform, ON.43–59
 - basics of, ON.43
 - rendering objects, ON.58–60
 - return types, ON.56–58
 - root-scoped objects. *See* root-scoped objects (Surf platform)

- Surf quick reference
 - Chrome, ON.108–110
 - common object properties, ON.107
 - Component instances object properties, ON.111–113
 - Component Type, ON.110–111
 - configuration objects, ON.111–113
 - Content Associations, ON.114–115
 - object definitions, ON.107
 - Page Associations, ON.115–116
 - Page Types, ON.116–117
 - Pages, ON.117–119
 - Template Instances, ON.119
 - Template Types, ON.120
 - Themes, ON.121
- swimlanes
 - assigning (process behavior), 156–157
 - defined, 138
 - jBPM, processes and, 138–139
- synchronization
 - basics, 111
 - Synchronization subsystem, 111

T

- Tagging service, 58, ON.99–100
- Tagging Web scripts (RESTful API), ON.36–37
- tags
 - status tags, updating (Knowledge Base), 350–351
 - Surf templates and, 302
 - updating on Knowledge Base content, 348–350
- task models, 146–153
 - basics, 146–147
 - common, 149–153
 - creating, 147–148
 - defined, 141
 - deploying, 153
- task nodes, 135–136
- task property pages, configuring, 162–164
- tasks
 - assigning (process behavior), 156–157
 - jBPM, processes and, 138–139
 - workflow tasks, 160
- Template API, 59

- template instances, 381, 395, ON.119
- template models, ON.62–74
 - advanced `TemplateNode` model API, ON.68
 - Classification API, ON.69–70
 - current date, ON.74
 - default model, ON.62–63
 - default model methods, ON.73–74
 - People API, ON.73
 - `TemplateNode` model API, ON.64–68
 - Version History API, ON.68–69
 - Workflow API, ON.71–72
 - XML Content Processing API, ON.70–71
- template scope (Surf Framework), 297, 300, ON.112–113
- template Surf platform root-scoped object, ON.48
- Template Types (Surf platform), ON.120
- template views (Knowledge Base), 374, 440
- `TemplateNode` model API, ON.64–68
- TemplateNode objects, ON.62, ON.63
- templates. *See also* Knowledge Base space templates,
 - creating; repository FreeMarker template API form template (sample code), 443–444
 - FreeMarker, adding to customize Knowledge Base, 381–382
 - FreeMarker for Surf, 294
 - FreeMarker Template processor (Surf example), ON.120
 - processing, ON.61
 - response status code templates, 211–213
 - response templates. *See* multiple response templates; response templates, creating
 - root-scoped objects and (Surf platform), ON.58–59
 - Share form template, adding, 424
 - space templates, Knowledge Base applications, 342–343
 - Surf Framework, 296–298, 302
 - Template Instance with configuration (Surf example), ON.119
 - URI templates, 195–197
- testing
 - configuration files, 235
 - Knowledge Base Search Web script example, 219–222
 - multipart form processing Web scripts, 244
 - process definition deployment, 146
 - response templates, 230, 231
 - Web scripts, 207–209, 211, 238
- theme Surf platform root-scoped object, ON.49
- Themes (Surf platform), ON.121
- Thumbnail services, 56–57, ON.100–101
- Thumbnail Web scripts (RESTful API), ON.38–39
- tiers
 - Counter Service examples and, 64–67
 - services and, 58–59
- timers, jBPM, 139
- tokenization, defined (content modeling), 85
- tokens, URI templates and, 195, 196, 197
- Tomcat ROOT Web application files, adding to, 384–385, 425–426
- tools
 - development tools (Surf Framework), 282
 - jBPM Process Designer, 143
 - SpringSource Tool Suite Plug-in for Surf, 289
- Torvalds, Linus, 14
- Touch Web script (RESTful API), ON.9
- Transformation API, ON.86–87
- Transformation services, 56
- `transformCommand` property, 63
- transitions
 - jBPM, processes and, 135–137
 - workflow, 160
- triggers, for rules, 325–326
- tunneling HTTP methods (Web clients), 184–185
- types
 - common task models, 149
 - content metamodels, 79, 82–83, 89–90
 - domain models and, 101
 - task model types, 159–160

U

- UI configurations, 141–142
- unstructured information, defined, 5
- Update User Password Web script (RESTful API), ON.30–31
- updating, content models, 78
- Upload Web scripts (RESTful API), ON.40
- URI templates, 195–197
- URIs
 - defining (Web scripts), 194
 - Web script URIs, parsing, 198–199

Web scripts and, 183–184
 url objects (Web scripts), 222–223
 url Surf platform root-scoped object, ON.55
 URLs, for Explorer, 321
 url.templateArgs root object, 199
 user interface configurations
 (UI configurations), 141–142
 user objects, ON.102
 user Surf platform root-scoped object, ON.45–46
 users
 number of and Alfresco, 13
 security and, 115–116
 specifying identity (Web scripts), 179–180
 utility methods, ON.95–96

V

values
 advanced TemplateNode API, ON.68
 family option, 254–255
 lifecycle option, 254–255
 null, FreeMarker template engine and, ON.63
 response templates, 204
 status values, custom dashlets, 370
 variables
 process variables, 154–155
 start-task process variables, 156
 verbose URI argument, 260
 Version History API, ON.68–69
 Version Web scripts (RESTful API), ON.40–41
 Versioning services (CMIS), 54
 Versions ScriptNodeAPI, ON.85–86
 view composition basics (Surf Framework), 294
 view resolvers (Surf Framework), 291–292
 views (Web scripts)
 code for (Knowledge Base), 393–394
 defined, 171

W

warm backup, defined, 39
 WCM (Web Content Management). *See* Web Content Management (WCM)
 WCM Asset Object API, ON.95
 WCM web projects, ON.93–96

the Web, Web scripts and, 168
 Web Application Framework
 basics of, 35–37
 clustering and, 38–39
 Web browsers, invoking Web scripts from, 175–176
 Web Content Management (WCM)
 basics of, 5, 11
 Surf and, 286–287
 Web projects, 57
 Web script descriptor files, defined, 374
 Web Script Framework, 227–255
 advanced descriptor options, 254–255
 basics of, 170–171
 caching, 250–254
 configuration, 231–235
 content negotiation, 235–238
 index of Web scripts in, 173–175
 internationalization, 227–231
 multipart form processing. *See* form processing
 primary design goal of, 227
 request processing, 245–250
 Web scripts, 167–187. *See also* Java-backed Web scripts; RESTful API reference
 adding, to customize site pages, 380–381
 adding Knowledge Base Search Web script, 344–345
 authenticating, 176–180
 basics of, 9–10, 167–169
 binding stock navigation, 396
 binding stock title, 396
 binding to region (Knowledge Base), 395
 caching, 186–187
 clients to invoke from, 175–176
 configuration (Knowledge Base), 397
 defining (Knowledge Base), 400–401
 defining controller (Knowledge Base), 399–400
 defining Knowledge Base view, 398–399
 descriptor (Knowledge Base), 397
 extensions, 46–47
 HEAD markup generator, 398
 HTTP client limitations, 184–186
 integrating with Alfresco and, 456

- JavaScript Debugger and, 208–209
- management of, 191
- override for create site controller, 402
- pre-built, 173–175
- response formats, 180–183
- scripted vs. Java-backed, 257–258
- structure of, 183–184
- Surf Framework, 302–303
- types of, 169–170
- Web script Component Type (Surf example), ON.111
- Web script Framework, 170–171
- Web scripts, creating, 189–225
 - basics of, 189–190
 - controller scripts, creating. *See* controller scripts, creating
 - description documents, creating. *See* description documents, creating
 - Hello World, 171–173
 - Knowledge Base example. *See* Knowledge Base Search Web script example
 - multiple response templates. *See* multiple response templates
 - registering and testing, 207–209
 - response templates, creating. *See* response templates, creating
 - root objects reference, 222–225
 - Web script components, 190–191
- Web sites for downloading. *See* downloads
- Web sites for further information, 184–186
 - Apache Abdera libraries, 73
 - BeanShell, 138
 - CMIS, using with Alfresco, 74
 - Confluence, 463
 - cURLs, 176, 184
 - Dependency Injection (DI), 265
 - Drupal interface, 73
 - E4X, processing with XML, 234
 - HTML form content types, 239
 - HTTP Accept-Language header, 231
 - HTTP status codes, 201
 - IBM services guide, 459
 - jBPM jPDL User Guide*, 153
 - jDPL User Guide*, 143
 - JSON callback mechanism, 185
 - language and country codes, 231
 - namespaces and prefixes, listed, 315
 - processing XML with FreeMarker, 235
 - RESTClient, 176
 - RFC 2616, 182, 235
 - Spring Framework, 258
 - Spring Surf project, 311
 - Squid, 250
 - user guide for jBPM, 135
 - XML, processing E4X with, 234
- Web tier
 - basics of, 9
 - configuration extensions, 47
- WebDAV (Web-based Distributed Authoring and Versioning), 29–30
- WebProject Object API, ON.93–94
- webscript objects, 224–225
- WebScriptViewResolver (Surf Framework), 292
- web.xml file, adding to Web application, 384–385, 425
- wf:articleapproval workflow, 418
- wizards
 - Action Wizard, 340
 - Advanced Space Wizard, 347, 352
 - Create Content Wizard, 353
 - Create Rule Wizard, 349
 - Create Site Wizard, 329–330
 - Create Space Wizard, 347
 - Explorer, for Knowledge Base, 340–341
 - Installation Wizard, 44–45
 - Start Advanced Workflow Wizard, 146
- workflow
 - basics, 133–134
 - customizing (sample code), 435–438
 - document library (Share), 417–418, 422
 - Explorer Web client, configuring. *See* Explorer Web client, configuring
 - JavaScript Workflow API. *See* JavaScript API
 - jBPM engine, processes and. *See* jBPM engine
 - process behavior, customizing. *See* process behavior, customizing
 - process definitions, 143–146
 - processes, jBPM and. *See* jBPM, processes and
 - Review and Approval, 500–501
 - Share, customizing, 448–449

- showing custom, configuration for, 423–424
- task models. *See* task models
- workflow definitions, 141–143
- Workflow API, ON.71–72
- workflow content models, process definitions and, 318, 319
- workflow definitions, 141, 160
- workflow engine, 133–134
- workflow models (repository), 319–320
- workflow nodes, 160
- workflow package actions, 151–152
- workflow packages, 147
- workflow service, 133
- workflow services, ON.103–106
- WorkflowTaskItem API, ON.72
- WorkflowTaskItem objects, ON.71

X

XML

- defining Surf objects and, 301
- description documents and, 190
- presentation content and, 284–285
- processing E4X with, 234
- processing with FreeMarker language, 235
- for Surf pages, 295
- XML files, deploying content models and, 78
- XML Content Processing API, ON.70–71
- XML schema, content metamodels and, 80
- XMLHttpRequest object, 185

Z

- zones, security and, 116–117