

A

- Abstract Window Toolkit. *See* AWT
- access control list. *See* ACL
- ACL 11
- Activity Monitor process monitor 357
- administration interface 17
- Adobe Flash, extracting text from 235, 237
- Adobe PDF, extracting text from 235, 237
 - See also* PDF
- AFP 59
 - used for indexing 59
- AgoFilterBuilder 304–305
- AllDocCollector 213–214, 297
- AlreadyClosedException 376
- analysis 110
 - bigrams 149
 - by QueryParser 78
 - chain 117
 - character normalization 145–146
 - CJK languages 146
 - creating payloads from token attributes 264
 - custom attributes 123
 - during indexing 113
 - field-specific 140
 - highlighting 124
 - in Nutch 149–151
 - letter ngrams 387
 - multivalued fields 140
 - non-English languages 144–149
 - position gaps 138
 - programming languages 385–386
 - removing common words 127
 - shingles 139, 149
 - splitting source code terms 387
 - stemming example 264
 - substring searching using tokens 386–389
 - token filter 393
 - tokenizer 393
 - tokenizing URIs 400
 - versus parsing 114
 - with QueryParser 114
 - zero position increment 165
- analysis, Snowball, supported languages 265
 - See also* indexing, analysis
- analytics interface 17
 - Google Analytics 18
 - Lucene-specific metrics 17
- analyzer, provided during indexing 38
- AnalyzerDemo 120–121, 147, 261
- Analyzers 115
 - additional 262
 - Arabic 262
 - Brazilian 262
 - Chinese 262
 - CJK 262
 - compound words 262
 - Czech 262
 - Dutch 263
 - field types 113
 - French 263
 - German 262
 - getPositionIncrementGap 140
 - getPreviousTokenStream 115
 - Greek 262
 - miscellaneous 263
 - ngram 263
 - payloads 263
 - Persian 263
 - positions 263
 - Russian 263
 - setPreviousTokenStream 115
 - shingles 263
 - Snowball 262

- Analyzers (*continued*)
 - Thai 263
 - Wikipedia export 263
 - analyzers
 - buffering 133
 - building blocks 118
 - built into Lucene 13
 - built-in 111–115, 127
 - choosing 111, 128
 - creating your own 128
 - definition 110
 - getOffsetGap 140
 - injecting synonyms 131–138, 297
 - language-specific 264
 - multiple tokens at the same position 131
 - ngram 265–266
 - reusable token streams 115
 - shingle 267
 - uses 267
 - ShingleFilter 139
 - Snowball 264
 - TestApp 267
 - StandardAnalyzer 128
 - stemming 138–139
 - StopAnalyzer 127
 - tokens vs. terms 116
 - visualizing 120
 - AnalyzerUtils 121, 130
 - displayTokens 121, 137
 - displayTokensWithFullDetails 122
 - displayTokensWithPositions 137
 - tokensFromAnalysis 125
 - AnalyzingQueryParser 323
 - Another Tool for Language Recognition.
 - See* ANTLR
 - Ant, building Lucene 429
 - See also* Apache Ant
 - ANTLR 108, 386
 - Apache Ant
 - preparing to use 22
 - to build contrib modules 286
 - Apache Commons 129
 - columnar formatting 157
 - Digester, indexing using 250
 - Apache Jakarta 7
 - Apache JMeter 353
 - Apache POI project 247
 - Apache Software Foundation 7
 - Apache Software License 6
 - Apache subversion instance 286
 - Apache Tika. *See* Tika
 - Aperture 12
 - open source project 253
 - Apple Mac OS X, search feature 4
 - AR Archives, extracting text from 237
 - ArabicAnalyzer 262
 - Aroush, George 331
 - ASCIIFoldingFilter 118, 145
 - Asian language analysis 146–148
 - Attribute 123
 - AttributeSource
 - addAttribute 123
 - analysis 123
 - captureState 124
 - restoreState 124
 - audio formats, extracting text from metadata 237
 - AutoDetectParser 239, 245
 - Tika class 240
 - Autonomy 253
 - AWT 147
-
- B**
- backward compatibility. *See* Version
 - BalancedMergePolicy 349
 - BalancedSegmentMergePolicy 322
 - Balmain, David 336–337
 - Beagle 332, 341
 - benchmark, OpenIndex 445
 - Berkeley DB, storing index 292–293
 - BerkeleyDBJESearcher 293
 - Bialecki, Andrzej 256
 - Bobo Browse 407–414
 - beyond simple faceting 413–414
 - integration with Zoie 414
 - Runtime FacetHandlers 414
 - sorting 413
 - BoboBrowser 412
 - BoboIndexReader 408, 413–414
 - BodyContentHandler 239, 246
 - BookLinkCollector 212–213
 - BooksLikeThis 192
 - using MoreLikeThis 283
 - BooleanClause 166
 - Occur
 - MUST 166
 - MUST_NOT 166
 - SHOULD 166
 - BooleanFilter 304
 - BooleanQuery 143, 165, 211, 277, 387, 402, 405
 - combining queries 94
 - from QueryParser 77
 - used with PhraseQuery 163
 - using as a filter 183
 - BooleanQueryBuilder 305
 - BooleanScorer 402
 - boosting 87
 - by recency 187
 - dangers 48
 - documents 48–49
 - fields 49

- BoostingQuery 284
 - negativeQuery 284
 - positiveQuery 284
- boosts 13
- BrazilianAnalyzer 262
- Browsable 412
- Browse Engine, denormalization 34
- BrowseFacet 412
- BrowseHit 412
- BrowseRequest 411–412
 - setFilter 411
- BrowseResult
 - getFacets 412
 - getHits 412
- BrowseSelection 413
- Builder 304–305
- BulletinPayloadsAnalyzer 226
- BulletinPayloadsFilter 226
- BZIP2 files, extracting text from 235, 237

C

- C#, tokenizing 386
- C++, tokenizing 386
- CachedFilter 303
- caching
 - field values 153
 - filter 184
- CachingSpanFilter 178
- CachingTokenFilter 118
 - used during highlighting 271
- CachingWrapperFilter 178, 183–184, 223, 291
- CartesianTierPlotter 310–311
- Cascading Style Sheets. *See* CSS
- Catasta, Michele 392
- catchall field, for searching multiple fields 166
- categorizing documents, using term vectors 191
- CellConjunctionScorer 402
- CellDisjunctionScorer 402
- CellQuery 393
- CellReqExclScorer 402
- CellScorer 402
- ChainedFilter 185
 - combining with AND 291
 - combining with ANDNOT 291
 - combining with OR 290
 - combining with XOR 291
 - security filter example 289
- chaining filters 185
- Chandler 341
 - project 292
- charades 129
- CharFilter 146
- CharReader 146
- CharStream 145
- CharTokenizer 118
- CheckIndex 377, 434
 - tool 376
- Chinese analysis 262
- Chinese, Japanese, and Korean. *See* CJK
- ChineseAnalyzer 146
- ChineseDemo 147
- ChineseTest 146
- CIFS. *See* Samba file system
- CJK, analysis of 146
- CJKAnalyzer 146, 262
- client-server port definition 327
- CloseIndex benchmark task 348
- CLucene 328–331
 - API compatibility 329
 - supported platforms 329
 - Unicode support 331
- Collator, used for sorting String fields 163
- Collector 83, 191, 201
 - acceptsDocsOutOfOrder 211
 - collect 211
 - custom 210–214
 - setNextReader 210
 - setScorer 210
 - using field cache 155
- common errors 375
- Comparable 209
- Compass 19
 - denormalization 34
- ComplexPhraseQueryParser 323
- compound index, creating 436
- CompressionTools 44
- ConcurrentMergeScheduler 55, 72, 349, 356, 363
 - setMaxThreadCount 369
- ConcurrentModificationException 356
- ConstantScoreQuery 184
- content
 - acquiring 11–12
 - dividing into shards 18
 - raw, extracting for documents 12
- ContentHandler 239, 242
- ContentSource 349, 352
- contrib
 - analyzers 262–268
 - benchmark 348
 - AddDoc 444, 456
 - adding custom task 359
 - algorithm output 444
 - analyzer 445
 - ClearStats 452
 - CloseIndex 445, 454
 - CloseReader 454
 - CommitIndex 453
 - compound 448
 - content sources 449

- contrib, benchmark (*continued*)
 - content.source 445
 - content.source.encoding 446, 448
 - content.source.forever 446
 - content.source.log.step 447
 - content.source.verbose 446
 - ContentSource 449
 - control structures 450–452
 - CreateIndex 445, 453
 - creating line doc file 350
 - DeleteByPercent 456
 - DeleteDoc 456
 - deletion.policy 448
 - DirContentSource 449
 - directory 448
 - disabling statistics 451
 - disabling statistics per-task 452
 - doc.body.tokenized.norms 446
 - doc.delete.step 447
 - doc.index.props 447
 - doc.maker 445
 - doc.random.id.limit 446
 - doc.reuse.fields 351, 447
 - doc.store.body.bytes 446
 - doc.stored 446
 - doc.term.vector 446
 - doc.term.vector.offsets 446
 - doc.term.vector.positions 446
 - doc.tokenized 446
 - doc.tokenized.norms 446
 - DocMaker 449
 - docs.dir 446
 - docs.file 446
 - EnwikiContentSource 449
 - EnwikiQueryMaker 450
 - file.query.maker.default.field 447, 450
 - file.query.maker.file 447, 450
 - FileBasedQueryMaker 450
 - FlushReader 454
 - html.parser 446
 - indexing documents from a directory 449
 - indexing documents from a Wikipedia XML export 449
 - indexing documents from lines in a file 449
 - indexing Reuters documents 449
 - indexing TREC documents 449
 - Judge 460
 - line files 457
 - LineDocSource 449
 - log.queries 447
 - log.step 447
 - max.buffered 448
 - max.field.length 448
 - measuring memory usage 444
 - merge.factor 448
 - merge.policy 448
 - merge.scheduler 356, 448
 - naming a sub-task 451
 - NearRealtimeReader 454
 - NewAnalyzer 454
 - NewRound 452
 - OpenIndex 453
 - OpenReader 454
 - OptimizeIndex 453
 - PerfTask 452
 - PrecisionRecall 461
 - QualityQuery 460
 - QualityQueryParser 460
 - QualityStats, average 460
 - query.maker 447, 450
 - ram.flush.mb 351, 448
 - ReadTokens 457
 - RepAll 458
 - repeating a task 451
 - RepSelectByPref 458
 - RepSumByName 458
 - RepSumByNameRound 458
 - RepSumByPref 458
 - RepSumByPrefRound 458
 - ResetInputs 452
 - ResetSystemErase 453
 - ResetSystemSoft 453
 - ReutersContentSource 449
 - ReutersQueryMaker 450
 - RollbackIndex 453
 - running tasks in a background thread 450
 - running tasks in parallel threads 451
 - Search 454
 - SearchTrav 455
 - SearchTravRet 455
 - SearchTravRetHighlight 455
 - SearchTravRetLoadFieldSelector 455
 - SearchTravRetVectorHighlight 455
 - SearchWithSort 455
 - sequence of tasks 450
 - SetProp 455
 - SimpleDocMaker 450
 - SimpleQueryMaker 450
 - SimpleSloppyPhraseQueryMaker 450
 - SingleDocSource 449
 - SortableSingleDocSource 449
 - task.max.depth.log 447
 - TrecContentSource 449
 - TrecJudge 460
 - TrecTopicsReader 460
 - UpdateDoc 456
 - Wait 457
 - Warm 456
 - work.dir 445
 - WriteLineDocTask 457
 - writer.info.stream 447

- contrib (*continued*)
 - benchmark framework 441–462
 - builtin reporting tasks 458
 - builtin tasks 452
 - content source, document maker 448
 - query maker 450
 - running an algorithm 442–445
 - testing search quality 459
 - building 286
 - ChainedFilter 185
 - Highlighter 268–274
 - obtaining analyzers 267
 - QueryParser 320–322
 - Spatial Lucene 308–316
 - field cache usage 314
 - indexing 308
 - Mercator projection 309
 - performance 314
 - projection 309
 - searching 312
 - sinusoidal projection 309
 - Surround query language 306–308
 - wikipedia 351
 - XmlQueryParser 299–305
 - extending 304
 - using 300
 - contrib modules
 - introduction 6
 - spatial search 206
 - coordination, query term 87
 - CorePlusExtensionsParser 301
 - CorruptIndexException 376
 - CPIO Archives, extracting text from 237
 - createLineFile.alg 350
 - CreateSpellCheckerIndex 279
 - CreateThreadedIndexTask 359
 - CSS 268
 - in highlighting 272
 - CustomQueryParser 216
 - CustomScoreQuery 186
 - getCustomScoreProvider 186
 - Cutting, Doug 7, 149
 - relevant work 9
 - CzechAnalyzer 262
- D**
-
- database
 - primary key 40
 - storing index inside Berkeley DB 292
 - DatabaseConfig 292
 - DateField, dateToString 218
 - DateFilter, within ChainedFilter 290
 - DateFormat, SHORT 218
 - DateRecognizerSinkTokenizer 263–264
 - DateTools 218
 - Debian
 - Linux open file limit 366
 - Lucene ports 341
 - debugging queries 102
 - DefaultEncoder 271
 - DefaultSimilarity 88
 - Delbru, Renaud 392
 - DeletionPolicy 381
 - denormalization 34
 - DERI 392
 - Sindice.com search engine 393
 - Dictionary 279
 - Digester
 - addCallMethod 252
 - addObjectCreate 252
 - addSetNext 253
 - addSetProperties 252
 - See also* Apache Commons Digester
 - DigesterXMLDocument 251, 253
 - Digg 393
 - Digital Enterprise Research Institute. *See* DERI
 - DirContentSource 349
 - Directory 38–39, 55, 66, 292, 362, 431–432
 - copying all files 57
 - introduction 26
 - sync 67
 - Directory implementations
 - FileSwitchDirectory 56
 - MMapDirectory 56
 - NIOFSDirectory 56
 - RAMDirectory 56
 - SimpleFSDirectory 56
 - directory in Berkeley DB 292
 - DirectSolrConnection 338
 - DisjunctionMaxQuery 167
 - tie-breaker 168
 - disk usage
 - during backup 365
 - impact of commit frequency 67
 - impact of open readers 365
 - DistanceComparatorSource 207
 - DistanceQueryBuilder 312–313
 - DistanceSortSource 313
 - DocIdBitSet 178
 - DocIdSet 222
 - Document 34, 210, 242, 431, 434
 - editing with Luke 259
 - reuse 351
 - setBoost 48
 - document
 - analyzing 13
 - boosts 13
 - browsing with Luke 257
 - building 12–13

document (*continued*)
 clustering 267
 definition 12
 filters 12
 ID 75
 indexing 14
 introduction 32
 parsing, filtering 242
 vs. Document class 27
 document type definition. *See* DTD
 documentation 427
 documents and fields 32–33
 DOMUtils 305
 Donovan, Aaron 292
 downloading Lucene 426
 Droids 12
 DSight, denormalization 34
 DTD 300
 DuplicateFilter 285
 DutchAnalyzer 263
 dynamic fragmenting vs. highlighting 268

E

EdgeNGramFilter 266
 EdgeNGramTokenizer 279
 Edit distance. *See* Levenshtein distance
 Elastic search, sharding and replication 18
 Elschof, Paul 306
 encoding UTF-8 144
 entitlements, definition 11
 EnvironmentConfig 292
 EnwikiContentSource 349
 Eventful 393
 Excel. *See* Microsoft Excel
 Explanation 88
 Extensible Hypertext Markup Language. *See* XHTML
 Extensible Stylesheet Language. *See* XSL

F

FacetAccessible 412
 faceted search
 Bobo Browse 408
 definition 408
 FacetHandler 408, 413–414
 FacetSpec 411
 setMaxCount 412
 setMinHitCount 412
 setOrderBy 412
 FastVectorHighlighter 275–277
 compared to Highlighter 277
 Ferret 336–339

Field 34, 40, 49, 413, 431
 Index
 NOT_ANALYZED 143, 223
 NOT_ANALYZED_NO_NORMS 142, 186
 introduction 27
 omitTermFreqAndPositions, impact on disk
 usage 364
 reuse 351
 setBoost 49
 setOmitNorms 50
 setOmitTermFreqAndPositions 43
 Store, YES 200, 413
 TermVector, WITH_POSITIONS_OFFSETS 192
 field
 boosting, with catchall field 166
 date and time values 52
 implicit length boost 49
 indexing for sorting 46
 introduction 32
 keyword analysis 142–144
 multivalued 47
 numeric value 51
 NumericField 51
 omitTermFreqAndPositions 43
 options 32–33
 Reader value 45, 47
 reanalysis during searching 195
 TokenStream value 45, 47
 truncation 52–53, 389
 field cache 153–155
 DEFAULT 154
 memory usage 154, 390
 per segment readers 155
 setInfoStream 155
 used by sorting 413
 used for sorting 155
 field options 42–47
 combinations 46
 compressing fields 44
 indexing
 ANALYZED 43
 ANALYZED_NO_NORMS 43
 NO 43
 NOT_ANALYZED 43
 NOT_ANALYZED_NO_NORMS 43
 sorting 46
 storing 44
 NO 44
 YES 44
 term vectors 44
 NO 45
 WITH_OFFSETS 45
 WITH_POSITIONS 45
 WITH_POSITIONS_OFFSETS 45
 YES 45

- FieldCache 354, 372, 414, 422
 - reducing memory usage 372
 - String 372
 - StringIndex 372
 - using from custom Collector 211
 - FieldCacheRangeFilter 177, 179
 - FieldCacheSource 185
 - FieldCacheTermsFilter 177, 180
 - FieldComparator 208
 - FieldComparatorSource 205, 208
 - using field cache 155
 - FieldDocs 209
 - FieldMaskingSpanQuery 169
 - FieldNormModifier 323
 - FieldQuery 277
 - FieldScoreQuery 185
 - FieldSelector 153, 200–201, 355
 - accept 200
 - loading only specified fields 201
 - specify fields by set 201
 - stopping after first field 201
 - time savings 201
 - FieldSelectorResult 200
 - LAZY_LOAD 200
 - LOAD 200
 - LOAD_AND_BREAK 200
 - LOAD_FOR_MERGE 200
 - NO_LOAD 200
 - SIZE 200
 - SIZE_AND_BREAK 200
 - FieldSortedTermVectorMapper 199
 - file descriptors, finding the limit 366
 - FileNotFoundException over remote file
 - systems 60
 - FileSwitchDirectory 58
 - Filter 156, 285, 305
 - custom 221–225
 - turning into Query 224
 - used by Bobo Browse 411
 - using field cache 155
 - filter
 - as a query 185
 - by specific terms 180
 - by term prefix 183
 - cached as bit set 178
 - caching 184
 - ChainedFilter 289–291
 - chaining 185
 - combining multiple 289
 - creating from Query 180
 - creating from SpanQuery 181
 - filtering another filter 184
 - getDocIdSet 222
 - security filter 181
 - using a BooleanQuery 183
 - wrapped as query 184
 - FilteredDocIdSet 178, 184
 - match 184
 - FilteredQuery 185, 224, 291
 - filtering
 - by numeric range 179
 - by term range 178
 - search space 177
 - security filter
 - dynamic 183
 - using field cache 179
 - filtering token. *See* TokenFilter
 - finding similar documents using termvectors 191
 - FlagsAttribute 123
 - Flash. *See* Adobe Flash
 - Formatter 271
 - FragmentsBuilder 277
 - FrenchAnalyzer 263
 - frequency factor formula 88
 - FSDirectory 292, 354, 437
 - open 56
 - open method 26
 - seeing open files 367
 - fsync 69
 - Fuller, Robert 392
 - function queries 185–189
 - boosting by recency 187
 - using field cache 187
 - FuzzyLikeThisQuery 284
 - FuzzyQuery 100, 281, 284, 323, 355
 - formula 101
 - minimumSimilarity 355
 - prohibiting 215
- ## G
-
- GermanAnalyzer 262
 - Glouser, Grant 381
 - Google
 - alternative word suggestions 131
 - as model for basic search features 10
 - definitions 294
 - Google Analytics 18
 - Google Enterprise Connector Manager 12
 - GradientFormatter 271
 - Grails search plugin 19
 - GreekAnalyzer 262
 - Grub 12
 - GZIP compression, extracting text from 237
- ## H
-
- Hadoop, creation of 9
 - Harwood, Mark 268, 283, 299
 - hasDeletions 41
 - Hatcher, Erik 338

- Heritrix 12
- Hibernate Search, denormalization 34
- hierarchical organizational schemes 4
- HighFreqTerms 323
- Highlighter
 - compared to FastVectorHighlighter 277
 - Encoder 271
 - faster alternative 275
 - Formatter 271
 - Fragmenter 269
 - highlighting search results 273
 - Scorer 270
 - setMaxDocCharsToAnalyze 274
 - TokenSources 269
 - using CSS 272
- highlighting
 - query terms 268, 273–274
 - using CSS 268
 - vs. dynamic fragmenting 268
- HighlightIt 272
- Hoschek, Wolfgang 298
- HTML
 - cookie 84
 - extracting text from 235, 237
 - meta tag 144
 - parsing 114
- HtmlParser 239
- HTTP headers, indexing Last-Modified
 - header 52
- HTTP request, content type 144
- HttpServletRequest 219
- Humphrey, Marvin 334, 336

I

- I18N. *See* internationalization
- IDF 87, 183
- images, extracting text from metadata 237
- Index
 - NOT_ANALYZED 160
 - with field cache 154
 - NOT_ANALYZED_NO_NORMS 160
 - with field cache 154
- index
 - accessing over remote file systems 59–60
 - adding input to 35–36
 - commits 67–69
 - ACID transactions 69
 - commitUserData 67
 - custom metadata 67
 - file syncing 67
 - IndexDeletionPolicy 67–68
 - multiple 68
 - rollback 68
 - two-phased 67
 - compound file format 36
 - performance 355
 - creating 22–23
 - definition 11
 - file format 36
 - flexible schema 33
 - inverted, definition 35
 - merging 323
 - norms 50
 - operations 36–42
 - printing high frequency terms 323
 - replicating 18
 - safety after JVM, OS or machine crash 69
 - searching 23–25
 - segments 36, 433
 - segments file 36, 67
 - segments generation 36
 - splitting into subindexes 322
 - transactions 67–69
- index structure, converting 436
- index, inverted 437
- IndexCommit 39
 - getUserData 68
- IndexDeletionPolicy 39, 60, 67–68
 - example usage 68
- Indexer 432
 - limitations 242
- Indexer program 19–23
- IndexFiles 427
- IndexHTML 428
- indexing
 - .del file 440
 - .fdt file 439
 - .fdx file 439
 - .fnm file 437
 - .frq file 439
 - .nrm file 439
 - .prx file 439
 - .tii file 438
 - .tis file 438
 - .tvdf ile 439
 - .tvf file 439
 - .tvf files 439
 - acquiring content 11–12
 - adding documents 37–39
 - advanced topics 64–72
 - analysis 35
 - backing up the search index 373–375
 - batching up deletions 65
 - binary documents 235
 - boosting 47–49
 - browsing tool 256
 - buffering and flushing 66–67
 - building document 12–13
 - catchall field 372

- indexing (*continued*)
 - combining multiple indexes 390
 - components of 11–14
 - compound index 435
 - content, acquiring 11
 - corruption 376
 - causes 376
 - enabling assertions 376
 - repairing the index 377
 - data structures 11
 - dates & times 52
 - debugging 63
 - dedicated memory indices 298
 - definition 11
 - deletes buffered in RAM 40
 - deletions 440
 - directory structure 432
 - disk usage formula 364
 - disk usage over time 364
 - document
 - analyzing 13
 - building 12
 - indexing 14
 - extract text 34
 - extracting text 242
 - file descriptor usage 366
 - file descriptors 436
 - usage while merging 436
 - file view with Luke 261
 - flushing 66
 - by deletion count 66
 - by document count 66
 - by RAM usage 66
 - format 431
 - incremental 434
 - index files 434
 - into RAMDirectory 57
 - locking 60–63
 - LockStressTest 61
 - LockVerifyServer 61
 - NativeFSLockFactory 61
 - NoLockFactory 61
 - SimpleFSLockFactory 61
 - SingleInstanceLockFactory 61
 - testing if an index is locked 62
 - unlocking an index 62
 - VerifyingLockFactory 61
 - logical steps 14
 - logical view 431
 - memory usage 370–373
 - merging 36, 69–72
 - customizing 72
 - formula 70
 - SerialMergeScheduler 72
 - multifile index structure 432
 - norms 372, 439
 - numbers 51–52
 - open files over time 368
 - optimization 54–55
 - background 55
 - partial 54
 - temporary disk usage 55
 - RAM buffer size 369
 - reducing disk usage 363–365
 - removing documents 39–41
 - all documents 40
 - by Query 40
 - by Term 39
 - reclaiming disk space 65
 - using IndexReader 64
 - removing stop words 35
 - replacing documents 41–42
 - resource consumption 363
 - restoring an index from backup 375
 - segments 433–434
 - setInfoStream 63
 - steps 34–36
 - storing in Berkeley DB 292
 - swapping 371
 - term dictionary 438
 - term frequency 439
 - term positions 439
 - term vectors 439
 - for use with Highlighter 274
 - termInfosIndexDivisor 372
 - the Reuters corpus 349
 - the Wikipedia export 349
 - thread and process safety 58
 - tools 256
 - transactional support using BDB 292
 - using threads 356–359
 - XML 247
 - using Apache Commons Digester 250–253
 - using SAX 248–250
- indexing classes 25–28
- IndexMergeTool 323
- IndexReader 155, 210, 283, 298, 365, 411–412, 417
 - acting as a writer 65
 - creation 81
 - deleting documents with 64
 - getTermFreqVector 192, 198
 - listCommits 68
 - multiple readers on same index 58
 - opened in Zoie 418
 - perform deletions 64
 - point in time searching 58, 67
 - point in time view 82
 - read-only 354
 - reload frequently 415
 - reopen 82, 348, 362, 369, 415, 423

- IndexReader (*continued*)
 - reopening after commit 66
 - replace only when required 346
 - retrieving term vectors 192
 - setNorm 50
 - sharing across threads 59
 - thread-safety 58, 356
 - undeleteAll 65
 - verify documents 38
 - with IndexWriter on same index 58
- IndexReaderDecorator 414, 421
- IndexReaderFactory 418
- IndexReaderWarmer 422
- IndexSearcher 38, 301, 316, 362, 431
 - benefit from caching 184
 - close 362
 - compare to MultiSearcher 189
 - compute scores per hit 157
 - creation 81
 - introduction 28
 - opening from a directory 81
 - paging through results 84
 - passing custom Collector 213
 - purpose 75–76
 - reopening 360
 - search 360
 - searching 82
 - setDefaultFieldSortScoring 156
 - thread-safety 356
 - using 80–86
- IndexSplitter 322
- IndexWriter 140
 - addDocument 35, 37, 72, 349, 431
 - addDocument(Document, Analyzer) 37
 - addDocument(Document) 37
 - addIndexes 69, 352
 - addIndexesNoOptimize 57, 352
 - Analyzer instance 113
 - close 40, 66, 358
 - time required 348
 - commit 40, 66–67, 356, 362, 415, 422
 - constructors 39
 - default RAM buffer size 66
 - deleteAll 40
 - deleteDocuments 39, 72
 - deleteDocuments(Query) 40
 - deleteDocuments(Query[]) 40
 - deleteDocuments(Term) 39
 - deleteDocuments(Term[]) 39
 - DISABLE_AUTO_FLUSH 66
 - expungeDeletes 65, 71
 - getMaxFieldLength 53
 - getReader 54, 348
 - getReader for near-real-time search 86, 422
 - handing off to Lucene to index 35
 - introduction 26
 - inverted index 437
 - isLocked 62
 - making changes to an index 335
 - maxBufferedDocs 349
 - maxDoc 41
 - mergeFactor 70, 351
 - multifile index 435
 - numDocs 41
 - optimize 54, 66, 352, 365
 - partial optimize 352, 369
 - per-document analysis 113
 - prepareCommit 68, 356
 - reopening 346
 - rollback 67, 358
 - setIndexReaderWarmer 422
 - setInfoStream 63
 - setInfoStream, field truncation 53
 - setMaxBufferedDocs 351
 - setMaxFieldLength 53
 - setMaxMergeDocs 71
 - setMergedSegmentWarmer 349
 - setMergeFactor 71
 - setRAMBufferSizeMB 351, 369
 - setSimilarity 49
 - setUseCompoundFile 36, 351, 435
 - sharing across threads 59
 - steps during commit 67
 - thread-safety 356
 - unlock 62
 - updateDocument 349
 - updateDocument(Term, Document, analyzer) 41
 - updateDocument(Term, Document) 41
 - waitForMerges 72
- IndexWrtr 365
- information
 - explosion, dealing with 4–6
 - overload 6
 - specific, locating quickly 4
- information retrieval. *See* IR
- InputStream 242
- INSO, filters 253
- Installing Lucene 426–430
- InstantiatedIndex 298
- InstantiatedIndexWriter 298
- intelligent agent, creating 6
- internationalization 144
- InvalidTokenOffsetException 269
- inverse document frequency. *See* IDF
- inverted index. *See* index, inverted
- InvIndexer 335
- IR 6
 - definition 6
 - library vs. search engine 6

ISYS file readers 253
iTunes search feature 5

J

J2ME 129
JaroWinkler, distance metric for spell correction 281
Java
 tokenizing 386
 Unicode escape sequences 146
Java 2 Micro Edition. *See* J2ME
Java C Compiler. *See* JCC
Java class files, extracting text from 237
Java JAR files, extracting text from 237
Java Management Extensions. *See* JMX
Java Native Interface. *See* JNI
Java Runtime Environment. *See* JRE
javac, compile with UTF-8 encoding 146
JavaCC, building Lucene 429
JavaServer Page. *See* JSP
JCC 328
JConsole used by Zoie 420
JEDirectory 292
Jetty 424
JFlex 108, 128
 building Lucene 429
 usage in SIREn 399
JMX 419
 used by Zoie 419
JNI 292
Jones, Tim 205
JRE 56, 256
JRuby 339
 accessing Lucene from 327
JSP 302
JVM
 heap size 371
 seeing garbage collection details 371
 -server switch 346
Jython 341

K

Katta, sharding and replication 18
KeepOnlyLastCommitDeletionPolicy 68, 374
KeyView filters 253
keyword analyzer 143
KeywordAnalyzer 46, 127, 143, 154
KeywordTokenizer 118
KinoSearch 334
 differences vs Lucene 335
Krugle 381–391
 enterprise appliance 383

Krugle.org 381
Krugler, Ken 381
KStem 138

L

language detection 149
Last.fm 393
lemmatization 110
LengthFilter 118
letter ngrams used by spellchecker 278
LetterTokenizer 118, 125, 130
Levenshtein
 distance 100
 distance metric for spell correction 281
LineDocSource 350, 352
LinkedIn 393, 407, 414
LoadFirstFieldSelector 201
local wrapper port, definition 327
LockFactory 376
locking
 during indexing 60–63
 write.lock file 62
LockObtainFailedException 375
LockStressTest 61
LockVerifyServer 61
LogByteSizeMergePolicy 70
 setMaxMergeDocs 71
 setMaxMergeMB 71
 setMergeFactor 71
 setMinMergeMB 71
LogDocMergePolicy 70–71
LowerCaseFilter 35, 118, 125, 399
LowerCaseTokenizer 115, 118, 125, 139
lowercasing, order may matter 126
ls of 366
Lucene
 access from .NET 331
 access from C/C++ 328
 access from Perl 334
 adding search to apps 7
 advantages over other search apps 6
 analyzers, built-in 13
 API, introduction 6
 backing up the search index 373
 backwards compatibility 346
 building from source 429
 capabilities 7
 community 9
 content model 32–34
 demonstration applications 427
 developers 9
 documentation 427
 downloading 426
 fitting into app 18

- Lucene (*continued*)
 - flexible schema 33
 - history 7–9
 - index 11
 - integration of 10
 - introduction 6–9
 - release history 7
 - sample application 19–25
 - sample indexing application 19–23
 - sample searching application 23–25
 - search models 16
 - using from PHP 339
 - using from Python 340
 - using from Ruby 337
 - vs. search engine or web crawler 7
 - website 6
 - wiki 7
 - Lucene ports 325–343
 - Lucene.Net 331–334
 - API compatibility 332
 - index compatibility 334
 - performance 332
 - LUCENE_24 21
 - LUCENE_29 21
 - Lucy 336
 - Luke 51, 64, 256–262, 429
 - Analyzer Tool 261
 - browsing by term 258
 - browsing term vectors 259
 - Custom Similarity 262
 - document browsing 257
 - editing documents 259
 - Hadoop Plugin 262
 - indexing file view 261
 - Overview tab 257
 - scripting with JavaScript 262
 - search explanation 260
 - searching 259
 - searching with QueryParser 260
 - viewing synonyms 296
 - viewing term statistics 262
 - LuSQL, denormalization 34
- M**
-
- Mac OS X
 - open file limit 366
 - search feature 4
 - Spotlight 5
 - Mannix, Jake 407
 - MAP 460
 - MapFieldSelector 201
 - MappingCharFilter 146
 - MatchAllDocsQuery 101, 178
 - used for browsing facets 411
 - Maven 2, used by Tika 240
 - maxDoc vs. numDocs 41
 - MaxFieldLength
 - UNLIMITED 38
 - UNLIMITED or LIMITED 53
 - MD5, reducing field cache memory usage 390
 - mean average precision. *See* MAP
 - mean reciprocal rank. *See* MRR
 - MemoryIndex 298
 - mergeFactor 348, 355, 365, 368
 - performance impact 351
 - MergePolicy 36, 71, 322, 349
 - avoiding large segments 322
 - MergeScheduler 36, 72
 - merging
 - LogByteSizeMergePolicy 70
 - LogDocMergePolicy 70
 - waiting for merges to finish 72
 - Metadata, Tika class 236
 - Metaphone 129
 - Microsoft Excel, extracting text from 235, 237
 - Microsoft Office 2007, extracting text from 237
 - Microsoft Outlook, extracting text from 235, 237
 - Microsoft PowerPoint, extracting text from 235, 237
 - Microsoft Visio, extracting text from 235, 237
 - Microsoft Word
 - extracting text from 235, 237
 - parsing 114
 - MIDI files, extracting text from 237
 - Miller, George and WordNet 294
 - MMapDirectory 56, 354
 - Montezuma 338
 - MoreLikeThis 283
 - MoreLikeThisQuery 195
 - MP3 audio, extracting text from tags 237
 - MRR 460
 - MultiFieldQueryParser 165–166
 - default operator 166
 - interactions with Analyzer 167
 - multifile index, creating 435
 - MultiPassIndexSplitter 322
 - MultiPhraseQuery 136, 153, 163–165
 - QueryParser 165
 - slop 164
 - MultiSearcher 189, 316, 424
 - multithreaded searching. *See* ParallelMulti-Searcher
- N**
-
- native port, definition 326
 - native2ascii, Java tool 146
 - NativeFSLockFactory 61, 375

near-real-time reader 360
 near-real-time search 54, 84, 422
 avoiding commit 67
 introduction 54
 reducing turnaround time 348
 Networked File System. *See* NFS
 newBooleanQuery 214
 newFuzzyQuery 215
 newMatchAllDocsQuery 215
 newMultiPhraseQuery 215
 newPhraseQuery 214
 newPrefixQuery 215
 newRangeQuery 215
 newTermQuery 214
 newWildcardQuery 215
 NFS 59
 sharing index over 68
 used for indexing 59
 NGramTokenizer 279
 NIOFSDirectory 56, 354
 NoLockFactory 61
 non-English language analysis 144
 normalization
 field length 87
 query 87
 norms
 changing after indexing 323
 high memory usage 50
 impact on disk usage 364
 omitting 50
 NullFragmenter 269
 numDocs vs. maxDoc 41
 numeric fields
 filtering during search 179
 in field cache 154
 numeric range queries 216
 NumericField 51–52, 92, 218, 390
 filtering during searching 179
 precisionStep 52
 setDoubleValue 51
 setIntValue 52
 setLongValue 52
 sorting 160
 NumericPayloadTokenFilter 226
 NumericRangeFilter 51, 177, 179
 NumericRangeQuery 51, 92, 179
 created by QueryParser 217
 creation from QueryParser 216
 precisionStep 179
 Nutch 12, 383
 creation of 9
 Explanation 89
 sharding and replication 18
 shingles 139
 NutchDocumentAnalyzer 149

O

O'Leary, Patrick 308
 OfficeParser 239
 OffsetAttribute 123, 140
 endOffset 140
 OLE 34
 Open Office, extracting text from 235
 open source software, judging success 9
 OpenBitSet, used by Filter 223
 OpenDocument files, extracting text from 237
 OpenSolaris, open file limit 366
 optimize 355
 Oracle/Lucene integration, denormalization 34
 OS, I/O cache 354
 Outlook. *See* Microsoft Outlook
 OutOfMemoryError 370
 OutOfMemoryException 375
 OutputStream 239

P

paging through results 84
 ParallelMultiSearcher 191, 316, 390
 Parr, Terr 386
 ParseContext 245
 ParseException 215
 parsing 77, 79
 query expressions. *See* QueryParser
 QueryParser method 79
 versus analysis 114
 ParsingReader 246
 partitioning indexes 316
 PayloadAttribute 123, 226
 PayloadHelper 226
 PayloadNearQuery 230
 payloads 225–230, 398
 access via TermPositions 230
 and SpanQuery 230
 constructors 226
 during analysis 226–227
 during searching 227–230
 example uses 225
 usage in SIREn 392
 used by SIREn 400
 PayloadTermQuery 227, 230
 PDF 237
 See also Adobe PDF
 PDFBox 247
 PDFParser 239
 PerFieldAnalyzerWrapper 127, 143
 performance tuning 345–355
 best practices 346
 increasing indexing throughput 349–353
 issues with WildcardQuery 100

performance tuning (*continued*)

- Java profiler 346
- managing resources 363–373
- reducing disk usage 363–365
- reducing file descriptor usage 366
- reducing index to search delay 348
- reducing memory usage 370
- reducing search latency 353–355
- testing approach 347
- working with threads 355
 - indexing 356
 - searching 360

Perl, tokenizing 386

per-segment searching, field cache 155

PersianAnalyzer 263

PHP Bridge 340

PhraseQuery 43, 165, 277, 388, 400

- contrasted with SpanNearQuery 173
- converting to SpanNearQuery 214
- forcing term order 220
- from QueryParser 99
- multiple terms 98
- scoring 98
- slop 389
- slop factor 96
- with synonyms 135

PipedReader 246

PipedWriter 246

plain text, detecting character set 237

PLucene 335

Porter stemmer. *See* Porter stemming algorithm

Porter stemming algorithm 138

Porter, Dr. Martin 138, 264

PorterStemFilter 35, 118, 138, 197

ports

- acts_as_solr 338, 342
- Beagle 332, 341
- choosing 328
- CLucene 328
- Ferret 337
- JCC 328
- Jython 341
- KinoSearch 334
- Lucene.Net 331
- Lucy 336
- native 326
- other Perl options 337
- other Python options 341
- PHP 339
- PHP Bridge 340
- PLucene 335
- PyLucene 340
 - JCC 340
- Ruby on Rails 338
- SolColdFusion 343

- SolForrest 343
- SolJava 342
- SolJSON 342
- SolPerl 342
- SolPHP 342
- SolPython 341–342
- Solr 342
 - Solr.pm 342
 - SolrJS 343
 - Solrnet 343
 - solr-ruby 337
 - SolrSharp 343
 - SolRuby 342
 - trade-offs 327
 - types of 326
 - types of ports 326–327
 - Zend Framework 339

PositionalPorterStopAnalyzer 138

PositionBasedTermVectorMapper 199

PositionIncrementAttribute 123–124

- setPositionIncrement 124

positionIncrementGap 245

PowerPoint. *See* Microsoft PowerPoint

PrecedenceQueryParser 323

precision, definition 14

PrefixFilter 177, 183

PrefixQuery 93, 323

PrintStream 155, 157

probabilistic model 16

Process Monitor 366

properties file, encoding 147

ps Unix process monitor 357

pure Boolean model 16

PyLucene 292, 340

- API compatibility 341

Python, tokenizing 386

Q

queries, built-in 90

Query 75, 77, 195, 411, 431

- contrib queries 283–285
- creating from Filter 224
- expressing with XML 300
- produced by XmlQueryParser 300
- rewrite 202
 - used for highlighting 270
- starts with 93
- toString 102, 142
- turning into a Filter 180
- types 29
 - See also* QueryParser

query

- building 15
- creating from filter 184

- query (*continued*)
 - flexible, creating 4
 - rewrite 355
 - searching 16
 - toString 355
 - query expression. *See* QueryParser
 - QueryAutoStopWordAnalyzer 263
 - QueryBuilder 320
 - QueryBuilderFactory 305
 - QueryFilter 290
 - as security filter 181
 - querying 75
 - QueryNodeProcessor 320
 - QueryParser
 - analysis 78, 114
 - analyzer choice 114
 - analyzing all text 323
 - and NumericField 104
 - and SpanQuery 177
 - Boolean operators 105
 - combining with another Query 90
 - combining with programmatic queries 109
 - creating FuzzyQuery 106
 - creating MatchAllDocsQuery 107
 - creating PhraseQuery 99, 105
 - creating SpanNearQuery 220
 - creating TermQuery 103
 - creating WildcardQuery 104
 - date parsing locale 219
 - date ranges 218
 - default field name 79
 - embedding wildcard and fuzzy queries inside phrases 323
 - escape characters 102
 - expression examples 80
 - expression syntax 80
 - extending 214–221
 - field selection 107
 - flexible 320
 - getBooleanQuery 215
 - getFieldQuery 215, 220
 - getFuzzyQuery 215
 - getPrefixQuery 215
 - getRangeQuery 215
 - getWildcardQuery 215
 - grouping expressions 107
 - handling numeric ranges 216
 - handling operator precedence 323
 - introduction 15
 - Keyword fields 142
 - newRangeQuery 218
 - one analyzer applied 140
 - overriding Query construction by type 214
 - producing MultiPhraseQuery 165
 - prohibiting expensive queries 215–216
 - purpose 75
 - query on unanalyzed field 141
 - setDateResolution 218
 - setLocale 220
 - setting boost factor 108
 - subclassing 216
 - supporting span queries 306
 - surround contrib module 177
 - term range queries 103
 - testing SynonymAnalyzer 136
 - tradeoffs 108
 - using 79
 - using ParseException for error handling 214
 - version compatibility 79
 - with span queries 177
 - QueryTemplateManager 301
 - QueryTermScorer 270
 - QueryWrapperFilter 177, 180, 182–183, 221
- ## R
-
- RAID array 346
 - RAMDirectory 57, 292, 295, 415, 423, 437
 - RangeFilter 304
 - RDF 292, 392
 - creating the Web of Data 394
 - definition 394
 - triplestores 392
 - ReadTokens task 352
 - recall 459
 - definition 14
 - RecencyBoostingQuery 188
 - RegexFragmenter 270
 - RegexQuery 285
 - regular expressions. *See* WildcardQuery
 - relational database 392
 - relevance 83
 - remote file systems 59–60
 - Remote Method Invocation. *See* RMI
 - remote procedure call. *See* RPC
 - remote searching 316–320
 - RemoteSearchable 316
 - RemoteSearcher 424
 - removing common terms. *See* stop words
 - Representational State Transfer. *See* REST
 - Resource Description Framework. *See* RDF
 - REST 383
 - ReutersContentSource 349
 - reverse native port, definition 327
 - ReverseStringFilter 263–264
 - Rich Text Format. *See* RTF
 - RMI 424
 - searching via 316
 - robocopy, for hot backups of an index 374
 - RPC 424

RSolr 338
 rsync, for hot backups of an index 374
 RTF 235
 extracting text from 235, 237
 Ruby, tokenizing 386
 RussianAnalyzer 263

S

Samba file system, used for indexing 59
 SAX 239
 parsing using 248
 scaling
 index replication 18
 index sharding 18
 schema, flexible 33
 SCM 383
 SCMI 383
 score 75
 ScoreCachingWrapperSource 212
 ScoreDoc 75, 412, 431
 ScoreOrderFragmentsBuilder 277
 Scorer 270, 402
 score 210
 scoring 86
 formula 86
 raw score 87
 scrolling. *See* paging
 search
 administration options 17
 building query 15
 latency 353
 latency vs. throughput 353
 logs, for load testing 354
 pervasiveness of 4
 quality metrics 14
 reopening searcher with threads 360
 resource consumption 363
 results
 boosting 15
 presentation 14
 rendering 16
 searching query 16
 span queries 168–177
 using threads 360–363
 search application
 architecture 10
 baseline requirements 10
 components 9–19
 search engine
 administration interface 17
 analytics interface 17
 components of 14–16
 result presentation 14
 scaling 18

 spell correction 15
 UI 14
 vs. Lucene 7
 search model
 probabilistic 16
 pure Boolean 16
 vector space 16
 search within search, using Filters 177
 Searchable 191, 316, 412
 SearchClient 317
 Searcher program 23–25
 SearcherManager 360
 get 362
 maybeReopen 362
 release 362
 warm 363
 SearchFiles 427
 searching
 advanced user interfaces 299
 API 75
 assigning constant scores 184
 automatically testing index coverage 389
 boosting by sub-query 284
 boosting specific term occurrences 225
 by multiple terms 284
 by regular expression 285
 catchall field 166
 custom filters 221
 custom scoring 185
 custom sorting 205–210
 date ranges 218
 dealing with common terms 384
 dedicated memory indices 298
 definition 14
 example advanced scenario 74
 explanation with Luke 260
 file descriptor usage 366
 filtering 177–185
 overview 177
 for similar documents 192, 283
 handling a search timeout 202
 highlighting 273
 indexes in parallel 191
 multiple indexes 189
 numeric ranges 216
 on multiple fields 166–168
 open files over time 368
 per-segment 155, 187
 remote indexes 316, 320
 removing duplicate documents 285
 sorting search results 155–163
 source code 381–391
 spatial search 308
 stopping a slow search 201
 unanalyzed fields 141

- searching (*continued*)
 - URIs 400
 - using custom Collector 210–214
 - using IndexSearcher 82
 - using XSL to transform requests 302
 - with Luke 259
- searching classes 28–29
- SearchServer 316
- security filtering 181
 - mixed static and dynamic 185
- SegmentReader 362, 422
- SegmentTermEnum, next 405
- Sekiguchi, Koji 275
- Semantic Information Retrieval Engine. *See* SIREn
- semantic web 392
- semistructured data 395
- SerialMergeScheduler 72, 356
- SetBasedFieldSelector 201
- setMaxBufferedDeleteTerms 66
- setRAMBufferSizeMB 66
- shard, definition 18
- shingles
 - no stop words discarded during indexing 267
 - See also* analysis, shingles
 - used by Nutch 149
- Similarity 49, 88, 323
 - improving default relevance 323
 - lengthNorm 49
- similarity between documents. *See* term vectors
- similarity scoring formula 86
- Simple API for XML. *See* SAX
- SimpleAnalyzer 111–112, 115–116, 141
 - discards numbers 51
 - steps taken 127
- SimpleDateFormat 305
- SimpleFragmenter 269
- SimpleFSDirectory 55, 432
- SimpleFSLockFactory 61
- SimpleHTMLEncoder 271
- SimpleHTMLFormatter 271
- SimpleSpanFragmenter 270
- SingleInstanceLockFactory 61
- sinks 120
- SinkTokenizer 118
- SinusoidalProjector 311
- SIREn 392
 - benchmarks 403–405
 - BooleanQuery performance compared to Lucene 405
 - data model 397
 - data preparation 399–400
 - postings format compare to Lucene 404
 - searching entities 400–403
 - semistructured search 392, 403–406
- SirenPayloadFilter 393, 399
- slop 96
 - factor defined 98
 - with MultiPhraseQuery 164
 - with SpanNearQuery 173
- SmartChineseAnalyzer 146, 148, 262
- SnapshotDeletionPolicy
 - limitations 375
 - used for index backups 374
- Snowball stemmer 138
- SnowballAnalyzer 145, 263–264
- solid-state disk. *See* SSD
- SolPerl 337
- SolPHP 339
- SolPython 341
- Solr 12, 342
 - creating analysis chain 128
 - Ruby response format 338
 - sharding and replication 18
 - SIREn integration 393, 403
- Solr.pm 337
- Solr.QParser 403
- Solr.QParserPlugin 403
- Sort 162
 - INDEXORDER 159
 - RELEVANCE 158
- SortedTermVectorMapper 199
- SortField 162, 209
 - types 162
- sorting
 - accessing custom values 209–210
 - by a field 160
 - by field value 156–158
 - by geographic distance 205–206
 - by index order 159
 - by multiple fields 161
 - by relevance 158
 - custom method 205–210
 - field type 163
 - geographic distance formula 207
 - reversing 161
 - search results 155–163
 - specifying locale 163
- SortingExample 156
- Soundex. *See* Metaphone
- source code management interface. *See* SCMI
- source code management. *See* SCM
- span queries 168–177
 - access to payloads 168
 - combining 175
 - dumpSpans method 170
 - excluding matches 174
 - matching near one another 173
 - matching near the field start 172
 - matching single term 170
 - phrase within phrase matching 176

- span queries (*continued*)
 - QueryParser 177
 - turning into a filter 181
 - SpanFirstQuery 169, 172
 - SpanGradientFormatter 271
 - SpanNearQuery 106, 169, 173, 220, 230, 306
 - contrasted with PhraseQuery 173
 - deriving from PhraseQuery 214
 - inOrder flag 173
 - slop 173
 - SpanNotQuery 169, 174–175
 - SpanOrQuery 169, 175, 307
 - SpanQuery 43, 230, 277, 285
 - aggregating 175
 - and QueryParser 177
 - getSpans 230
 - visualization utility 171
 - SpanQueryFilter 177, 181
 - bitSpans 181
 - SpanRegexQuery 285
 - SpanScorer 230, 270
 - SpanTermQuery 168–170, 172, 230, 307
 - SPARQL query language 393
 - SPARQLParser 403
 - SPARQLParserPlugin 403
 - SPARQLQueryAnalyzer 393, 403
 - SpecialsAccessor 223
 - SpecialsFilter 222, 224
 - spell correction 277–282
 - generating suggestions 278–279
 - ideas for improvement 281
 - offering 15
 - picking best candidate 280–281
 - presenting to user 281
 - SpellChecker 279
 - setAccuracy 281
 - suggestSimilar 281
 - Spencer, David 277, 294
 - spider. *See* web crawler
 - Spolsky, Joel 144
 - Spotlight search 5
 - Spring 408
 - Spring-RPC 424
 - SSD 346
 - Stale NFS file handle exception 60
 - StandardAnalyzer 127–128, 351–352
 - common choice 128
 - core analyzers 427
 - example 111
 - steps taken 112
 - with CJK characters 146, 148
 - wrapped with additional filter 133
 - StandardFilter 118, 126
 - StandardQueryParser 320
 - StandardTokenizer 118, 126
 - Stellent document filters. *See* INSO filters
 - stemmers, SnowballAnalyzer family 145
 - stemming analyzer 264
 - stop words 27, 384
 - default 127
 - removing 35
 - StopAnalyzer 51, 111–112, 125, 127, 264
 - StopFilter 35, 118, 125, 127
 - setEnabledPositionIncrements 138
 - StopWordFilter 399
 - Store, YES 160
 - stored fields, custom loading 200
 - String.compareTo, compares by UTF16 code
 - unit 92
 - StringDistance, getDistance 281
 - StringUtils 157
 - Subversion
 - building Lucene from sources 429
 - checkout out contrib sources 286
 - swappiness, controlling swapping on Linux 371
 - SweetSpotSimilarity 323
 - SynLookup 295
 - SynonymAnalyzer 117, 131–138, 297
 - SynonymAnalyzerViewer 297
 - SynonymEngine 297
 - synonyms
 - injecting with MultiPhraseQuery 165
 - tradeoffs for indexing and searching 136
 - with PhraseQuery 136
 - See also* WordNet
 - Syns2Index 295
 - System, nanoTime 346
-
- T**
- Tan, Kelvin 289
 - TAR Archives, extracting text from 235, 237
 - tar, for hot backups of an index 374
 - Task Manager 366
 - measuring page faults 371
 - process monitor 357
 - TeeSinkTokenFilter 120
 - TeeTokenFilter 118
 - Term 198
 - term
 - definition 110
 - frequency 87
 - navigation with Luke 258
 - term vectors 124, 191–199
 - aggregating 196
 - browsing with Luke 259
 - computing angles between 197
 - computing archetype document 195
 - custom loading 198
 - example usage 44
 - formula for angle 197
 - introduction 44

- term vectors (*continued*)
 - regenerating from index 323
 - storing positions and offsets 192
- TermAttribute 123
- TermFreqVector 192
- TermPositions 230
- TermPositionVector 192
- TermQuery 90, 215, 388
 - combining 402
 - contrasted with SpanTermQuery 168
 - in keywords field 182
 - most basic Query type 76
 - with synonyms 135
- TermRangeFilter 177–178, 304
 - includeLower 178
 - includeUpper 178
 - open-ended ranges 179
 - with caching 184
- TermRangeQuery 91, 191, 323, 355
 - created by QueryParser 217
- terms, vs. tokens 116
- TermsFilter 180, 284
 - addTerm 284
- TermVectorAccessor 323
- TermVectorMapper 192, 198, 323, 355
 - isIgnoringOffsets 199
 - isIgnoringPositions 199
 - map 199
 - setDocumentNumber 199
 - setExpectations 199
- ThaiAnalyzer 263
- The Grinder load testing tool 353
- ThreadedIndexWriter 356, 359
- Tika 35
 - alternatives 253
 - built-in text extraction tool 240
 - customizing parser selection 246
 - getFileMetadata 244
 - installing 240
 - introduction 236
 - limitations 246
 - logical design 238
 - metadata extraction 236
 - modular design 240
 - parse 245
 - parser implementations 236
 - using UNIX pipes 241
 - utility class 245
- TikaConfig 246
 - getParsers 246
- TikaException 245
- TikaIndexer 242
- TimeExceededException 201–202
- TimeLimitingCollector 201
 - limitations 202
- Token 123
- TokenFilter 117, 399
 - additional 262
 - importance of order 125
 - shingles 384
 - splitting source code terms 387
- TokenFilters, for creating payloads 226
- tokenization, definition 110
- Tokenizer 117–118
 - additional 262
- TokenOffsetPayloadTokenFilter 226
- TokenRangeSinkTokenizer 263–264
- tokens
 - attributes 123
 - flags 120
 - offset 120
 - offsets 124
 - payload 120
 - positionIncrement 120
 - term 120
 - type 120, 130
 - definition 13, 110
 - endOffset 269
 - introduction 35, 116–117
 - offsets 124
 - offsets used for highlighting 269
 - position increment 117
 - same position 117
 - startOffset 269
 - type 124
 - visualizing positions 137
 - vs. terms 116
- TokenSources 269
 - getAnyTokenStream 274
- TokenStream 115, 117–118
 - architecture 117
 - buffering 119
 - incrementToken 123
 - used for highlighting 269
- TokenTypeSinkTokenizer 263–264
- Tomcat, demo application 428
- tool, Luke 256
- top Unix process monitor 357
- top, measuring page faults 371
- TopDocs 75, 82, 209
- TopFieldCollector 211
- TopFieldDocs 209
- TopScoreDocCollector 202, 211
- Toupikov, Nickolai 392
- triplestore, searching the Web of Data 393
- troubleshooting 430
- truncation. *See* field truncation
- Tummarello, Giovanni 392
- TupleAnalyzer 393, 399
- TupleQuery 393
 - addClause 402
- TupleScorer 402

TupleTokenizer 393, 399
 two-phased commit 67
 TypeAsPayloadTokenFilter 125, 226
 TypeAttribute 123

U

UI novel, creating 6
 unanalyzed fields, searching 141
 Unicode 144
 Unix, deletion of open files 365
 URINormalisationFilter 393, 399
 user interface. *See* UI
 UTF-8 144

V

Vajda, Andi 292, 340–341
 value 45, 47
 ValueSource 185
 ValueSourceQuery 185
 van Klinken, Ben 328
 van Rossum, Guido 340
 Vector Space Model 16, 43
 VerifyingLockFactory 61
 Version 21
 Visio. *See* Microsoft Visio
 vmstat, measuring page faults 371

W

W3C 393
 Wall, Larry 334
 Wang, John 407
 WAVE Audio, extracting text from sampling
 metadata 237
 WeakHashMap
 for filter caching 184
 keyed by IndexReader 154
 Web 3.0 392
 web application
 CSS highlighting 272
 demo 428
 web application server, thread pool 360
 web crawler
 definition 11
 open source 11
 vs. Lucene 7
 Web of Data 393
 Wettin, Karl 298
 WhitespaceAnalyzer 51, 111–112, 127–128
 WhitespaceTokenizer 118
 Wikipedia
 document source 349
 indexing 351

WikipediaTokenizer 264
 WildcardQuery 99, 277, 323
 inefficiency 396
 prohibiting 215
 Windows Explorer 371
 Windows Server 2003, open file limit 366
 Windows, deletion of open files 365
 with payloads 225
 Word. *See* Microsoft Word
 WordNet 294–297
 adding synonyms during analysis 297
 building synonym index 295
 example synonyms 297
 WordNetSynonymEngine 297
 write.lock 376
 WriteLineDoc 349
 Writer 239

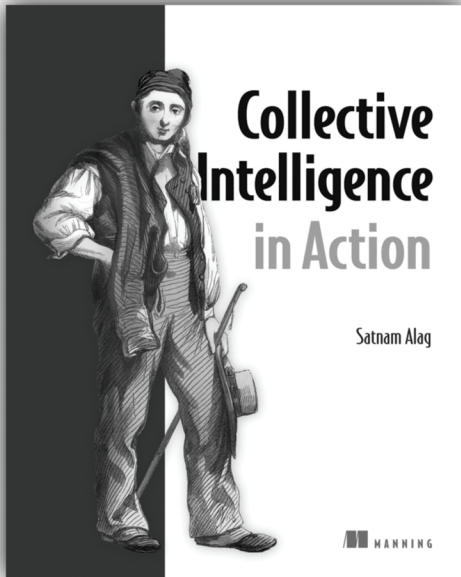
X

XHTML 238
 used by Tika 238
 XML
 encoding 144
 extracting text from 235, 237
 parsing 114
 XmlQueryParser 90, 299
 XSL 300

Z

Zend Framework 339
 ZIP Archives, extracting text from 237
 ZIP files, extracting text from 235
 Zoie 407, 414
 batchDelay 419
 batchSize 419
 compared to Lucene's built-in NRT search 422
 data consumer, definition 416
 data provider, definition 416
 disk index 417
 fault tolerance 416
 indexing requests 419
 mapping Lucene's docID to application UID 415
 near-real-time 419
 RAM indexes 417
 static ranking of people 415
 unique IDs 419
 ZoieIndexable 419
 ZoieIndexableInterpreter 419
 ZoieIndexReader 414, 422
 ZoieSystem 423–424

MORE TITLES FROM MANNING



Collective Intelligence in Action

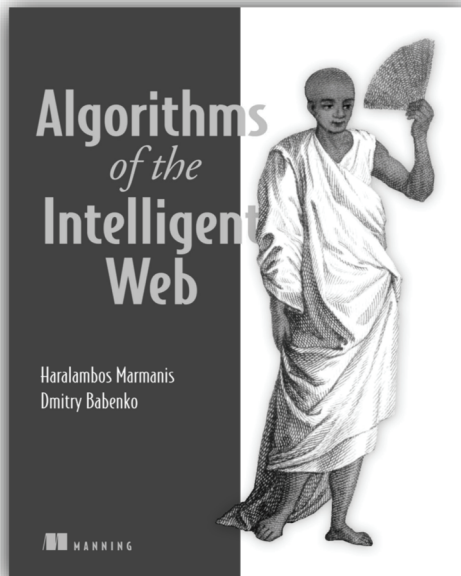
by Satnam Alag

ISBN: 978-1-933988-31-3

424 pages

\$44.99

October 2008



Algorithms of the Intelligent Web

by Haralambos Marmanis
and Dmitry Babenko

ISBN: 978-1-933988-66-5

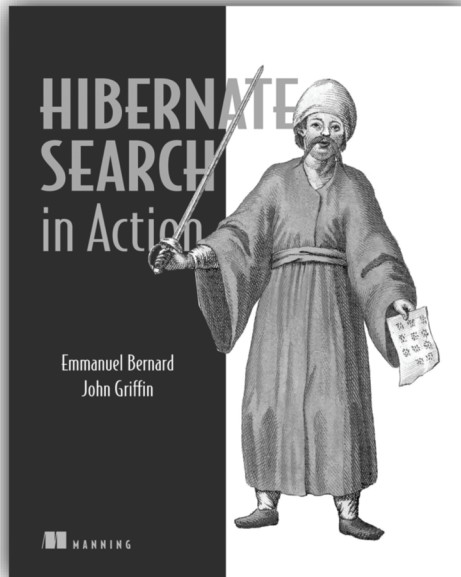
368 pages

\$44.99

May 2009

For ordering information go to www.manning.com

MORE TITLES FROM MANNING



Hibernate Search in Action

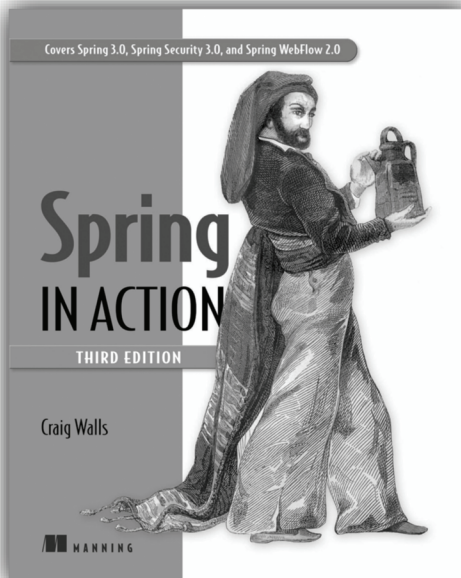
by Emmanuel Bernard and John Griffin

ISBN: 978-1-933988-64-1

488 pages

\$49.99

December 2008



Spring in Action, Third Edition

by Craig Walls

ISBN: 978-1-935182-35-1

700 pages

\$49.99

Fall 2010

For ordering information go to www.manning.com