LUCENE - INDEXWRITER

http://www.tutorialspoint.com/lucene/lucene_indexwriter.htm

Introduction

This class acts as a core component which creates/updates indexes during indexing process.

Class declaration

Following is the declaration for **org.apache.lucene.index.IndexWriter** class:

```
public class IndexWriter
   extends Object
      implements Closeable, TwoPhaseCommit
```

Field

Following are the fields for **org.apache.lucene.index.IndexWriter** class:

- **static int DEFAULT_MAX_BUFFERED_DELETE_TERMS** -- Deprecated. use IndexWriterConfig.DEFAULT_MAX_BUFFERED_DELETE_TERMS instead.
- **static int DEFAULT_MAX_BUFFERED_DOCS** -- Deprecated. Use IndexWriterConfig.DEFAULT_MAX_BUFFERED_DOCS instead.
- **static int DEFAULT_MAX_FIELD_LENGTH** -- Deprecated. See IndexWriterConfig.
- **static double DEFAULT_RAM_BUFFER_SIZE_MB** -- Deprecated. Use IndexWriterConfig.DEFAULT_RAM_BUFFER_SIZE_MB instead.
- **static int DEFAULT_TERM_INDEX_INTERVAL** -- Deprecated. Use IndexWriterConfig.DEFAULT_TERM_INDEX_INTERVAL instead.
- **static int DISABLE_AUTO_FLUSH** -- Deprecated. Use IndexWriterConfig.DISABLE_AUTO_FLUSH instead.
- **static int MAX_TERM_LENGTH** -- Absolute hard maximum length for a term.
- **static String WRITE_LOCK_NAME** -- Name of the write lock in the index.
- **static long WRITE_LOCK_TIMEOUT** -- Deprecated. Use IndexWriterConfig.WRITE_LOCK_TIMEOUT instead.

Class constructors

S.N.	Constructor & Description
1	**IndexWriter** *Directoryd, Analyzera, booleancreate, IndexDeletionPolicydeletionPolicy, IndexWriter. MaxFieldLengthmfl* Deprecated. use IndexWriter*Directory, IndexWriterConfig* instead.
2	**IndexWriter***Directoryd, Analyzera, booleancreate, IndexWriter. MaxFieldLengthmfl* Deprecated. use IndexWriter*Directory, IndexWriterConfig* instead.
3	**IndexWriter***Directoryd, Analyzera, IndexDeletionPolicydeletionPolicy, IndexWriter. MaxFieldLengthmfl* Deprecated. use IndexWriter*Directory, IndexWriterConfig* instead.

4	**IndexWriter** ***Directoryd, Analyzera, IndexDeletionPolicydeletionPolicy, IndexWriter. MaxFieldLengthmfl, IndexCommitcommit*** Deprecated. use IndexWriter*Directory, IndexWriterConfig* instead.
5	**IndexWriter*Directoryd, Analyzera, IndexWriter. MaxFieldLengthmfl*** Deprecated. use IndexWriter*Directory, IndexWriterConfig* instead.
6	**IndexWriter*Directoryd, IndexWriterConfigconf*** Constructs a new IndexWriter per the settings given in conf.

Class methods

S.N.	Method & Description
1	**void addDocument*Documentdoc*** Adds a document to this index.
2	**void addDocument*Documentdoc, Analyzeranalyzer*** Adds a document to this index, using the provided analyzer instead of the value of getAnalyzer.
3	**void addDocuments*Collection < Document > docs*** Atomically adds a block of documents with sequentially assigned document IDs, such that an external reader will see all or none of the documents.
4	**void addDocuments*Collection < Document > docs, Analyzeranalyzer*** Atomically adds a block of documents, analyzed using the provided analyzer, with sequentially assigned document IDs, such that an external reader will see all or none of the documents.
5	**void addIndexes*Directory. . . dirs*** Adds all segments from an array of indexes into this index.
6	**void addIndexes*IndexReader. . . readers*** Merges the provided indexes into this index.
7	**void addIndexesNoOptimize*Directory. . . dirs*** Deprecated.use addIndexes*Directory. . .* instead.

8

void close

Commits all changes to an index and closes all associated files.

9

void close$booleanwaitForMerges$

Closes the index with or without waiting for currently running merges to finish.

10

void commit

Commits all pending changes added & deleted documents, segment merges, added indexes, etc. to the index, and syncs all referenced index files, such that a reader will see the changes and the index updates will survive an OS or machine crash or power loss.

11

void commit$Map < String, String > commitUserData$

Commits all changes to the index, specifying a commitUserData Map $String - > String$.

12

void deleteAll

Delete all documents in the index.

13

void deleteDocuments$Query. . . queries$

Deletes the document$s$ matching any of the provided queries.

14

void deleteDocuments$Queryquery$

Deletes the document$s$ matching the provided query.

15

void deleteDocuments$Term. . . terms$

Deletes the document$s$ containing any of the terms.

16

void deleteDocuments$Termterm$

Deletes the document$s$ containing term.

17

void deleteUnusedFiles

Expert: remove any index files that are no longer used.

18

protected void doAfterFlush

A hook for extending classes to execute operations after pending added and deleted documents have been flushed to the Directory but before the change is committed $newsegments_Nfilewritten$.

19

protected void doBeforeFlush

A hook for extending classes to execute operations before pending added and deleted documents are flushed to the Directory.

20

protected void ensureOpen

21

protected void ensureOpen*booleanincludePendingClose*

Used internally to throw an AlreadyClosedException if this IndexWriter has been closed.

22

void expungeDeletes

Deprecated.

23

void expungeDeletes*booleandoWait*

Deprecated.

24

protected void flush*booleantriggerMerge, booleanapplyAllDeletes*

Flush all in-memory buffered updates *addsanddeletes* to the Directory.

25

protected void flush*booleantriggerMerge, booleanflushDocStores, booleanflushDeletes*

NOTE: flushDocStores is ignored now *hardwiredtotrue*; this method is only here for backwards compatibility

26

void forceMerge*intmaxNumSegments*

Forces merge policy to merge segments until there's <= maxNumSegments.

27

void forceMerge*intmaxNumSegments, booleandoWait*

Just like forceMerge*int*, except you can specify whether the call should block until all merging completes.

28

void forceMergeDeletes

Forces merging of all segments that have deleted documents.

29

void forceMergeDeletes*booleandoWait*

Just like forceMergeDeletes, except you can specify whether the call should block until the operation completes.

30

Analyzer getAnalyzer

Returns the analyzer used by this index.

31

IndexWriterConfig getConfig

Returns the private IndexWriterConfig, cloned from the IndexWriterConfig passed to IndexWriter*Directory, IndexWriterConfig*.

32

static PrintStream getDefaultInfoStream

Returns the current default infoStream for newly instantiated IndexWriters.

33

static long getDefaultWriteLockTimeout

Deprecated.use IndexWriterConfig.getDefaultWriteLockTimeout instead

34

Directory getDirectory

Returns the Directory used by this index.

35

PrintStream getInfoStream

Returns the current infoStream in use by this writer.

36

int getMaxBufferedDeleteTerms

Deprecated.use IndexWriterConfig.getMaxBufferedDeleteTerms instead

37

int getMaxBufferedDocs

Deprecated.use IndexWriterConfig.getMaxBufferedDocs instead.

38

int getMaxFieldLength

Deprecated.use LimitTokenCountAnalyzer to limit number of tokens.

39

int getMaxMergeDocs

Deprecated.use LogMergePolicy.getMaxMergeDocs directly.

40

IndexWriter.IndexReaderWarmer getMergedSegmentWarmer

Deprecated.use IndexWriterConfig.getMergedSegmentWarmer instead.

41

int getMergeFactor

Deprecated.use LogMergePolicy.getMergeFactor directly.

42

MergePolicy getMergePolicy

Deprecated.use IndexWriterConfig.getMergePolicy instead

43

MergeScheduler getMergeScheduler

Deprecated.use IndexWriterConfig.getMergeScheduler instead

44

Collection<SegmentInfo> getMergingSegments

Expert: to be used by a MergePolicy to a void selecting merges for segments already being merged.

45

MergePolicy.OneMerge getNextMerge

Expert: the MergeScheduler calls this method to retrieve the next merge requested by the MergePolicy

46

PayloadProcessorProvider getPayloadProcessorProvider

Returns the PayloadProcessorProvider that is used during segment merges to process payloads.

47

double getRAMBufferSizeMB

Deprecated.use IndexWriterConfig.getRAMBufferSizeMB instead.

48

IndexReader getReader

Deprecated.Please use IndexReader.open*IndexWriter, boolean* instead.

49

IndexReader getReader*inttermInfosIndexDivisor*

Deprecated.Please use IndexReader.open*IndexWriter, boolean* instead. Furthermore, this method cannot guarantee the reader *anditssub − readers* will be opened with the termInfosIndexDivisor setting because some of them may have already been opened according to IndexWriterConfig.setReaderTermsIndexDivisor*int*. You should set the requested termInfosIndexDivisor through IndexWriterConfig.setReaderTermsIndexDivisor *int* and use getReader.

50

int getReaderTermsIndexDivisor

Deprecated.use IndexWriterConfig.getReaderTermsIndexDivisor instead.

51

Similarity getSimilarity

Deprecated.use IndexWriterConfig.getSimilarity instead

52

int getTermIndexInterval

Deprecated.use IndexWriterConfig.getTermIndexInterval

53

boolean getUseCompoundFile

Deprecated.use LogMergePolicy.getUseCompoundFile

54

long getWriteLockTimeout

Deprecated.use IndexWriterConfig.getWriteLockTimeout

55

boolean hasDeletions

56

static boolean isLocked*Directorydirectory*

Returns true iff the index in the named directory is currently locked.

57

int maxDoc

Returns total number of docs in this index, including docs not yet flushed *stillintheRAMbuffer*, not counting deletions.

58

void maybeMerge

Expert: asks the mergePolicy whether any merges are necessary now and if so, runs the requested merges and then iterate *testagainifmergesareneeded* until no more merges are returned by the mergePolicy.

59

void merge*MergePolicy. OneMergemerge*

Merges the indicated segments, replacing them in the stack with a single segment.

60

void message*Stringmessage*

Prints a message to the infoStream $ifnon - null$, prefixed with the identifying information for this writer and the thread that's calling it.

61

int numDeletedDocs*SegmentInfoinfo*

Obtain the number of deleted docs for a pooled reader.

62

int numDocs

Returns total number of docs in this index, including docs not yet flushed *stillintheRAMbuffer*, and including deletions.

63

int numRamDocs

Expert: Return the number of documents currently buffered in RAM.

64

void optimize

Deprecated.

65

void optimize*booleandoWait*

Deprecated.

66

void optimize$\boldsymbol{intmaxNumSegments}$

Deprecated.

67

void prepareCommit

Expert: prepare for commit.

68

void prepareCommit$\boldsymbol{Map < String, String > commitUserData}$

Expert: prepare for commit, specifying commitUserData Map $String- > String$.

69

long ramSizeInBytes

Expert: Return the total size of all index files currently cached in memory.

70

void rollback

Close the IndexWriter without committing any changes that have occurred since the last commit $orsinceitwasopened, ifcommithasn' tbeencalled$.

71

String segString

72

String segString$\boldsymbol{Iterable < SegmentInfo > infos}$

73

String segString$\boldsymbol{SegmentInfoinfo}$

74

static void setDefaultInfoStream$\boldsymbol{PrintStreaminfoStream}$

If non-null, this will be the default infoStream used by a newly instantiated IndexWriter.

75

static void setDefaultWriteLockTimeout$\boldsymbol{longwriteLockTimeout}$

Deprecated.use IndexWriterConfig.setDefaultWriteLockTimeout$long$ instead

76

void setInfoStream$\boldsymbol{PrintStreaminfoStream}$

If non-null, information about merges, deletes and a message when maxFieldLength is reached will be printed to this.

77

void setMaxBufferedDeleteTerms$\boldsymbol{intmaxBufferedDeleteTerms}$

Deprecated.use IndexWriterConfig.setMaxBufferedDeleteTermsint instead.

78

void setMaxBufferedDocs$\boldsymbol{intmaxBufferedDocs}$

Deprecated.use IndexWriterConfig.setMaxBufferedDocs*int* instead.

79

void setMaxFieldLength*intmaxFieldLength*

Deprecated.use LimitTokenCountAnalyzer instead. Note that the behvaior slightly changed - the analyzer limits the number of tokens per token stream created, while this setting limits the total number of tokens to index. This only matters if you index many multi-valued fields though.

80

void setMaxMergeDocs*intmaxMergeDocs*

Deprecated.use LogMergePolicy.setMaxMergeDocs*int* directly.

81

void setMergedSegmentWarmer*IndexWriter. IndexReaderWarmerwarmer*

Deprecated.use IndexWriterConfig.setMergedSegmentWarmer *org. apache. lucene. index. IndexWriter. IndexReaderWarmer* instead.

82

void setMergeFactor*intmergeFactor*

Deprecated.use LogMergePolicy.setMergeFactor*int* directly.

83

void setMergePolicy*MergePolicymp*

Deprecated.use IndexWriterConfig.setMergePolicy*MergePolicy* instead.

84

void setMergeScheduler*MergeSchedulermergeScheduler*

Deprecated.use IndexWriterConfig.setMergeScheduler*MergeScheduler* instead

85

void setPayloadProcessorProvider*PayloadProcessorProviderpcp*

Sets the PayloadProcessorProvider to use when merging payloads.

86

void setRAMBufferSizeMB*doublemb*

Deprecated.use IndexWriterConfig.setRAMBufferSizeMB*double* instead.

87

void setReaderTermsIndexDivisor*intdivisor*

Deprecated.use IndexWriterConfig.setReaderTermsIndexDivisor*int* instead.

88

void setSimilarity*Similaritysimilarity*

Deprecated.use IndexWriterConfig.setSimilarity*Similarity* instead

89

void setTermIndexInterval*intinterval*

Deprecated.use IndexWriterConfig.setTermIndexInterval*int*

90

void setUseCompoundFile*booleanvalue*

Deprecated.use LogMergePolicy.setUseCompoundFile*boolean*.

91

void setWriteLockTimeout*longwriteLockTimeout*

Deprecated.use IndexWriterConfig.setWriteLockTimeout*long* instead

92

static void unlock*Directorydirectory*

Forcibly unlocks the index in the named directory.

93

void updateDocument*Termterm, Documentdoc*

Updates a document by first deleting the document*s* containing term and then adding the new document.

94

void updateDocument*Termterm, Documentdoc, Analyzeranalyzer*

Updates a document by first deleting the document*s* containing term and then adding the new document.

95

void updateDocuments*TermdelTerm, Collection < Document > docs*

Atomically deletes documents matching the provided delTerm and adds a block of documents with sequentially assigned document IDs, such that an external reader will see all or none of the documents.

96

void updateDocuments*TermdelTerm, Collection < Document > docs, Analyzeranalyzer*

Atomically deletes documents matching the provided delTerm and adds a block of documents, analyzed using the provided analyzer, with sequentially assigned document IDs, such that an external reader will see all or none of the documents.

97

boolean verbose

Returns true if verbosing is enabled (i.e., infoStream !

98

void waitForMerges

Wait for any currently outstanding merges to finish.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object