Indexing Hands-on

Search Engines

Master Degree in Computer Engineering
Master Degree in Data Science
Academic Year 2023/2024

Nicola Ferro

Intelligent Interactive Information Access (IIIA) Hub
Department of Information Engineering
University of Padua

Outline

- Indexing Options in Lucene
- Inside Lucene Indexes
- Lucene Codecs
- Extracting Information from a Lucene Index

Indexing Options in Lucene

Inverted Index: Just Documents

- It allows for retrieving documents containing any or all query terms
  - small marsupial → **intersection** if searching for all the terms; **union** if searching for any terms
- No ranking possible
- No proximity match/ phrase match possible

Dictionary	Posting List
a	1 3
about	1
and	3
approach	3
are	2
australia	2 3
cat	1
closely	3
commonly	3
domestic	1
exists	3
fear	3
genus	1
have	3
humans	3
in	3
is	1
island	3
legged	2
little	3
marsupial	1
marsupials	2
member	1
muscular	2
native	2
of	1 3
on	3
only	1
particularly	3
people	3
population	3
prevalent	3
quadrupedal	2
quokka	1
quokkas	3
rottnest	3
setonix	1
short	2
size	1
small	1 2
that	2
the	1
to	2
where	3
wombats	2

Inverted Index: Just Documents

Luke: Lucene Toolbox Project - v8.8.1

File Tools Help

Overview | Documents | Search | Analysis | Commits | Logs

Index Path: /Users/ferro/Documents/progetti/software/search-engines/se-unipd/hello-ir/experiment/index
Number of Fields: 2
Number of Documents: 3
Number of Terms: 48
Has deletions? / Optimized?: No / Yes
Index Version: 8
Index Format: Lucene 8.6 or later
Directory implementation: org.apache.lucene.store.MMapDirectory
Currently opened commit point: segments_2 (generation=2, segs=1)
Current commit user data: {}

Number of documents containing the term

Select a field from the list below, and press button to view top terms in the field.

Available fields and term counts per field:

Name	Term count	%
body	45	93.75 %
id	3	6.25 %

Selected field: body

Show top terms >

Num of terms: 68

Top ranking terms: (Double-click for more options.)

Rank	Freq	Text
1	2	small
2	2	of
3	2	australia
4	2	a
5	1	wombats
6	1	where
7	1	to
8	1	the
9	1	that
10	1	size
11	1	short
12	1	setonix
13	1	rottnest
14	1	quokkas
15	1	quokka
16	1	quadrupedal
17	1	prevalent
18	1	population
19	1	people
20	1	particularly
21	1	only
22	1	on
23	1	native
24	1	muscular
25	1	member
26	1	marsupials
27	1	marsupial
28	1	little
29	1	legged
30	1	island
31	1	is
32	1	in
33	1	humans
34	1	have
35	1	genus
36	1	fear
37	1	exists
38	1	domestic
39	1	commonly
40	1	closely
41	1	cat
42	1	are
43	1	approach
44	1	and
45	1	about

Luke: Lucene Toolbox Project - v8.8.1

File Tools Help

Overview | Documents | Search | Analysis | Commits | Logs

Browse terms in field: body

Browse documents by term: small

« First Term | small | Next

« First Doc | 1 | Next | in 2 docs

Hint:
Edit the text field above and press Enter to seek to arbitrary terms.

Position	Offsets	Payload
11		

Browse documents by Doc # 0 in 3 docs

Copy values | More like this | Add document

(Select a row and double-click for more options.)

(To copy all or arbitrary field value(s), unselect all rows or select row(s), and click 'Copy values' button.)

Field	Flags Help	Norm	Value
id	Idfp-N-S----------	1	d1
body	Idfp-N-S----------	20	The quokka, the only member of the genus Setonix, is a small marsupial a...

Luke: Lucene Toolbox Project - v8.8.1

File Tools Help

Overview | Documents | Search | Analysis | Commits | Logs

Browse terms in field: body

Browse documents by term: small

« First Term | small | Next

« First Doc | 2 | Next | in 2 docs

Hint:
Edit the text field above and press Enter to seek to arbitrary terms.

Position	Offsets	Payload
2		

Browse documents by Doc # 1 in 3 docs

Copy values | More like this | Add document

(Select a row and double-click for more options.)

(To copy all or arbitrary field value(s), unselect all rows or select row(s), and click 'Copy values' button.)

Field	Flags Help	Norm	Value
id	Idfp-N-S----------	1	d2
body	Idfp-N-S----------	13	Wombats are small, short-legged, muscular quadrupedal marsupials that ar...

Inverted Index: Documents and Positions

- It allows for ranking documents by their relevance
- Proximity match/ phrase match possible
  - “small marsupial”
  - find `small` within 5 words of `marsupials`

Dictionary	Posting List
a	1:10,17 3:18
about	1:13
and	3:6
approach	3:8
are	2:1,9
australia	2:12 3:16
cat	1:19
closely	3:10
commonly	3:7
domestic	1:18
exists	3:21
fear	3:3
genus	1:7
have	3:1
humans	3:5
in	3:15
is	1:9
island	3:14
legged	2:4
little	3:2
marsupial	1:**12**
marsupials	2:**7**
member	1:4
muscular	2:5
native	2:10
of	1:5,16 3:4
on	3:12
only	1:3
particularly	3:11
people	3:9
population	3:20
prevalent	3:19
quadrupedal	2:6
quokka	1:1
quokkas	3:0
rottnest	3:13
setonix	1:8
short	2:3
size	1:15
small	1:**11** 2:**2**
that	2:8
the	1:0,2,6,14
to	2:11
where	3:17
wombats	2:0

Inverted Index: Documents and Positions

Luke: Lucene Toolbox Project - v8.8.1

File Tools Help

Overview | Documents | Search | Analysis | Commits | Logs

Browse terms in field:

body

« First Term | the | Next

Hint:
Edit the text field above and press Enter to seek to arbitrary terms.

Browse documents by term:

the

« First Doc | 1 | Next | in 1 docs

Position	Offsets	Payload
0		
2		
6		
14		

Browse documents by Doc # 0 in 3 docs

Copy values | More like this | Add document

(Select a row and double-click for more options.)

(To copy all or arbitrary field value(s), unselect all rows or select row(s), and click 'Copy values' button.)

Field	Flags Help	Norm	Value
id	Idfp-N-S----------	1	d1
body	Idfp-N-S----------	20	The quokka, the only member of the genus Setonix, is a small marsupial about the size of a domestic cat

Inverted Index: Options in Lucene

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | ENUM CONSTANTS | FIELD | METHOD DETAIL: ENUM CONSTANTS | FIELD | METHOD

org.apache.lucene.index

Enum IndexOptions

java.lang.Object
  java.lang.Enum<IndexOptions>
    org.apache.lucene.index.IndexOptions

All Implemented Interfaces:

`Serializable, Comparable<IndexOptions>`

```
public enum IndexOptions
extends Enum<IndexOptions>
```

Controls how much information is stored in the postings lists.

WARNING: This API is experimental and might change in incompatible ways in the next release.

Enum Constant Summary

Enum Constants

Enum Constant and Description
DOCS Only documents are indexed: term frequencies and positions are omitted.
DOCS_AND_FREQS Only documents and term frequencies are indexed: positions are omitted.
DOCS_AND_FREQS_AND_POSITIONS Indexes documents, frequencies and positions.
DOCS_AND_FREQS_AND_POSITIONS_AND_OFFSETS Indexes documents, frequencies, positions and offsets.
NONE Not indexed

Method Summary

All Methods | **Static Methods** | **Concrete Methods**

Modifier and Type	Method and Description

a che carattere in doc appare token

⇓

es. usarlo per estrarre snippet

```java
/**
 * Represents a {@link Field} for containing the body of a document.
 * <p>
 * It is a tokenized field, not stored, keeping only document ids and term frequencies (see {@link
 * IndexOptions#DOCS_AND_FREQS} in order to minimize the space occupation.
 *
 * @author Nicola Ferro (ferro@dei.unipd.it)
 * @version 1.00
 * @since 1.00
 */
public class BodyField extends Field {

    /**
     * The type of the document body field
     */
    private static final FieldType BODY_TYPE = new FieldType();

    static {
        BODY_TYPE.setIndexOptions(IndexOptions.DOCS_AND_FREQS);
        BODY_TYPE.setTokenized(true);
        BODY_TYPE.setStored(false);
    }


    /**
     * Create a new field for the body of a document.
     *
     * @param value the contents of the body of a document.
     */
    public BodyField(final Reader value) { super(ParsedDocument.FIELDS.BODY, value, BODY_TYPE); }

    /**
     * Create a new field for the body of a document.
     *
     * @param value the contents of the body of a document.
     */
    public BodyField(final String value) { super(ParsedDocument.FIELDS.BODY, value, BODY_TYPE); }

}
```

Some Indexing Statistics

No Stop No Stem	Size	Time
Documents	151 MByte	92 seconds
Documents and Frequencies	209 MByte	101 seconds
Documents and Positions	555 Mbyte	105 seconds
Documents and Positions and Offsets	999 MByte	136 seconds

Inside Lucene Indexes

Lucene Indexes

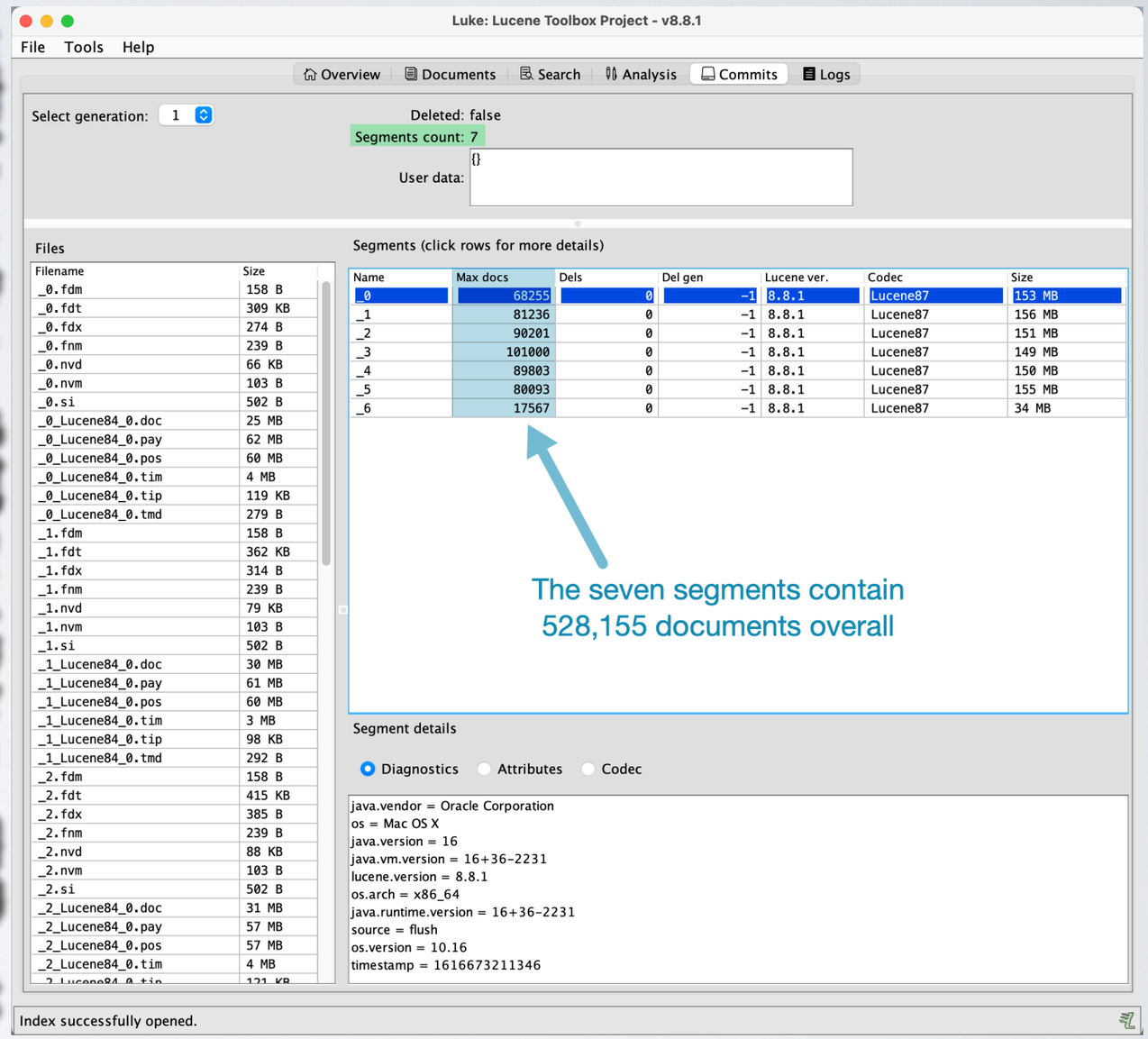
- Lucene indexes may be composed of multiple sub-indexes, or **segments**
  - Each segment is a fully independent index, which could be searched separately
- Indexes evolve by
  - Creating new segments for newly added documents
  - Merging existing segments
- Segments allow for
  - **incremental indexing**
  - **distributed indexing**
- Searches may involve multiple segments and/or multiple indexes, each index potentially composed of a set of segments
- Changes to the content of an index are made **visible** only after the writer **commits** by writing a new segments file (`segments_N`)
  - This point in time, when the action of writing of a new segments file to the directory is completed, is an `index commit`

What's in a Lucene Segment?

- **Segment info**: metadata about a segment, such as the number of documents, what files it uses, and information about how the segment is sorted
- **Field names**: the set of field names used in the index
- Inverted index
  - **Term dictionary**: a dictionary containing all of the terms used in all of the indexed fields of all of the documents. The dictionary also contains the number of documents which contain the term, and pointers to the term's frequency and proximity data.
  - **Term Frequency data**: for each term in the dictionary, the numbers of all the documents that contain that term, and the frequency of the term in that document
  - **Term Proximity data**: for each term in the dictionary, the positions that the term occurs in each document
  - **Stored Field values**: for each document, a list of attribute-value pairs, where the attributes are field names. These are used to store auxiliary information about the document, such as its title
  - **Normalization factors**: for each field in each document, a value is stored that is multiplied into the score for hits on that field
  - **Per-document values**: like stored values, these are also keyed by document number, but are generally intended to be loaded into main memory for fast access. Whereas stored values are generally intended for summary results from searches, per-document values are useful for things like scoring factors, faceted browsing, etc.
- Direct index
  - **Term Vectors**: for each field in each document, the term vector (sometimes called document vector) may be stored. A term vector consists of term text and term frequency
- Other
  - **Point values**: Optional pair of files, recording dimensionally indexed fields, to enable fast numeric range filtering and large numeric values like BigInteger and BigDecimal (1D) and geographic shape intersection (2D, 3D)
  - **Live documents**: an optional file indicating which documents are live.

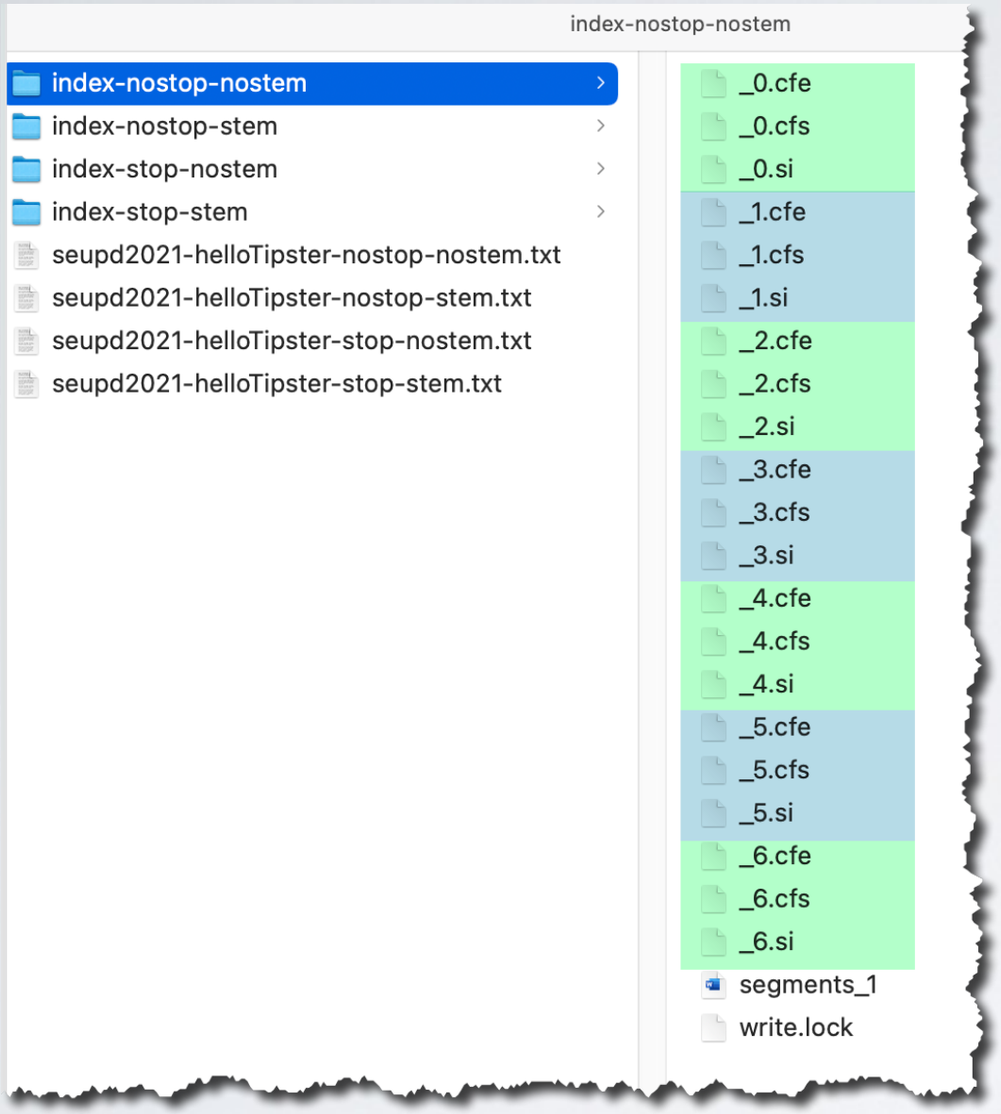
Example of TIPSTER Index

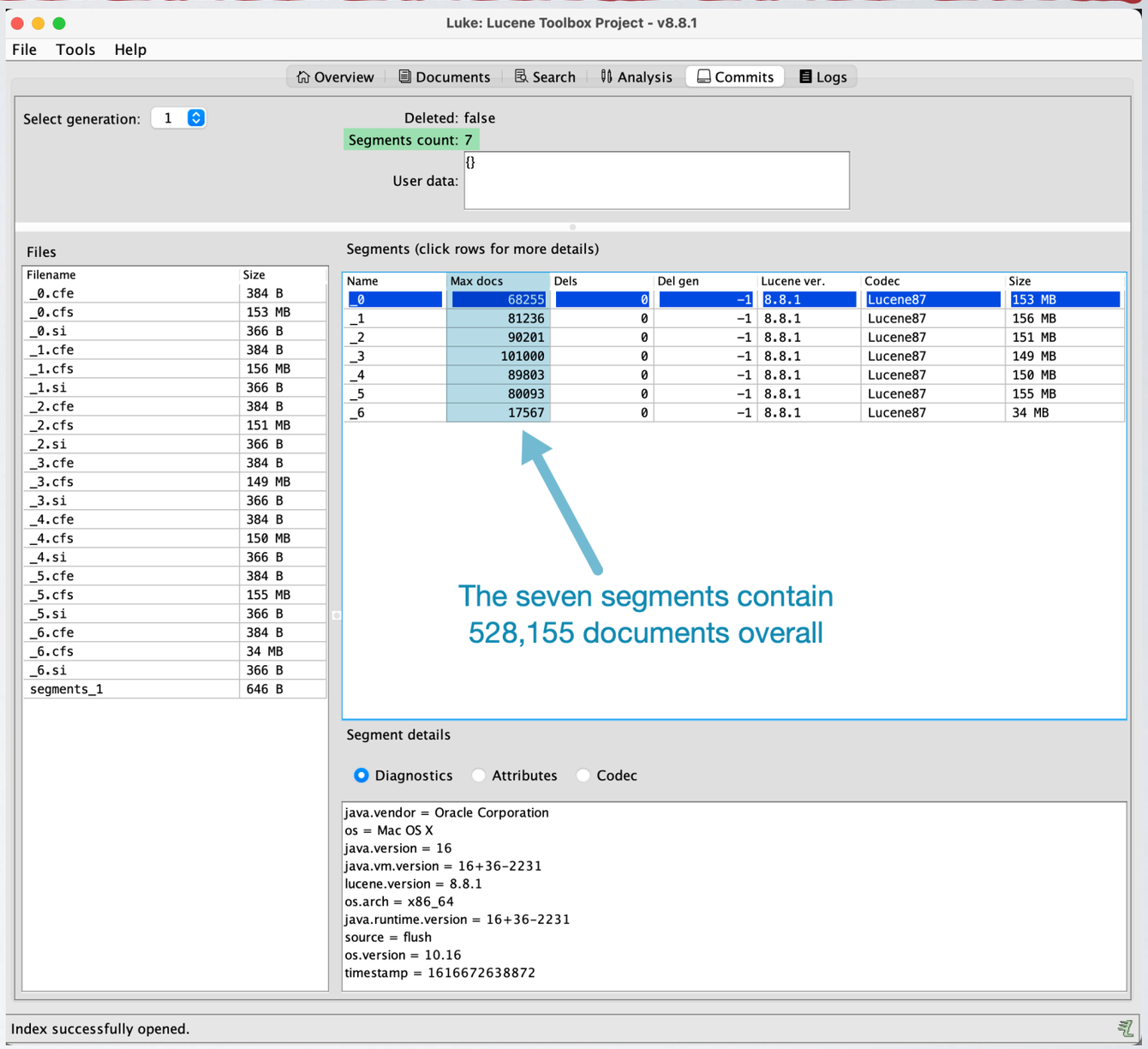
```
[√ index-nostop-nostem % ls
_0.fdm                  _1.fnm                  _2.si
_0.fdt                  _1.nvd                  _2_Lucene84_0.doc
_0.fdx                  _1.nvm                  _2_Lucene84_0.pay
_0.fnm                  _1.si                   _2_Lucene84_0.pos
_0.nvd                  _1_Lucene84_0.doc       _2_Lucene84_0.tim
_0.nvm                  _1_Lucene84_0.pay       _2_Lucene84_0.tip
_0.si                   _1_Lucene84_0.pos       _2_Lucene84_0.tmd
_0_Lucene84_0.doc       _1_Lucene84_0.tim       _3.fdm
_0_Lucene84_0.pay       _1_Lucene84_0.tip       _3.fdt
_0_Lucene84_0.pos       _1_Lucene84_0.tmd       _3.fdx
_0_Lucene84_0.tim       _2.fdm                  _3.fnm
_0_Lucene84_0.tip       _2.fdt                  _3.nvd
_0_Lucene84_0.tmd       _2.fdx                  _3.nvm
_1.fdm                  _2.fnm                  _3.si
_1.fdt                  _2.nvd                  _3_Lucene84_0.doc
_1.fdx                  _2.nvm                  _3_Lucene84_0.pay

_3_Lucene84_0.pos       _4_Lucene84_0.tmd       _6.fdx
_3_Lucene84_0.tim       _5.fdm                  _6.fnm
_3_Lucene84_0.tip       _5.fdt                  _6.nvd
_3_Lucene84_0.tmd       _5.fdx                  _6.nvm
_4.fdm                  _5.fnm                  _6.si
_4.fdt                  _5.nvd                  _6_Lucene84_0.doc
_4.fdx                  _5.nvm                  _6_Lucene84_0.pay
_4.fnm                  _5.si                   _6_Lucene84_0.pos
_4.nvd                  _5_Lucene84_0.doc       _6_Lucene84_0.tim
_4.nvm                  _5_Lucene84_0.pay       _6_Lucene84_0.tip
_4.si                   _5_Lucene84_0.pos       _6_Lucene84_0.tmd
_4_Lucene84_0.doc       _5_Lucene84_0.tim       segments_1
_4_Lucene84_0.pay       _5_Lucene84_0.tip       write.lock
_4_Lucene84_0.pos       _5_Lucene84_0.tmd
_4_Lucene84_0.tim       _6.fdm
_4_Lucene84_0.tip       _6.fdt
```



- No stop, no stemming
- Documents, frequencies, positions, and offsets

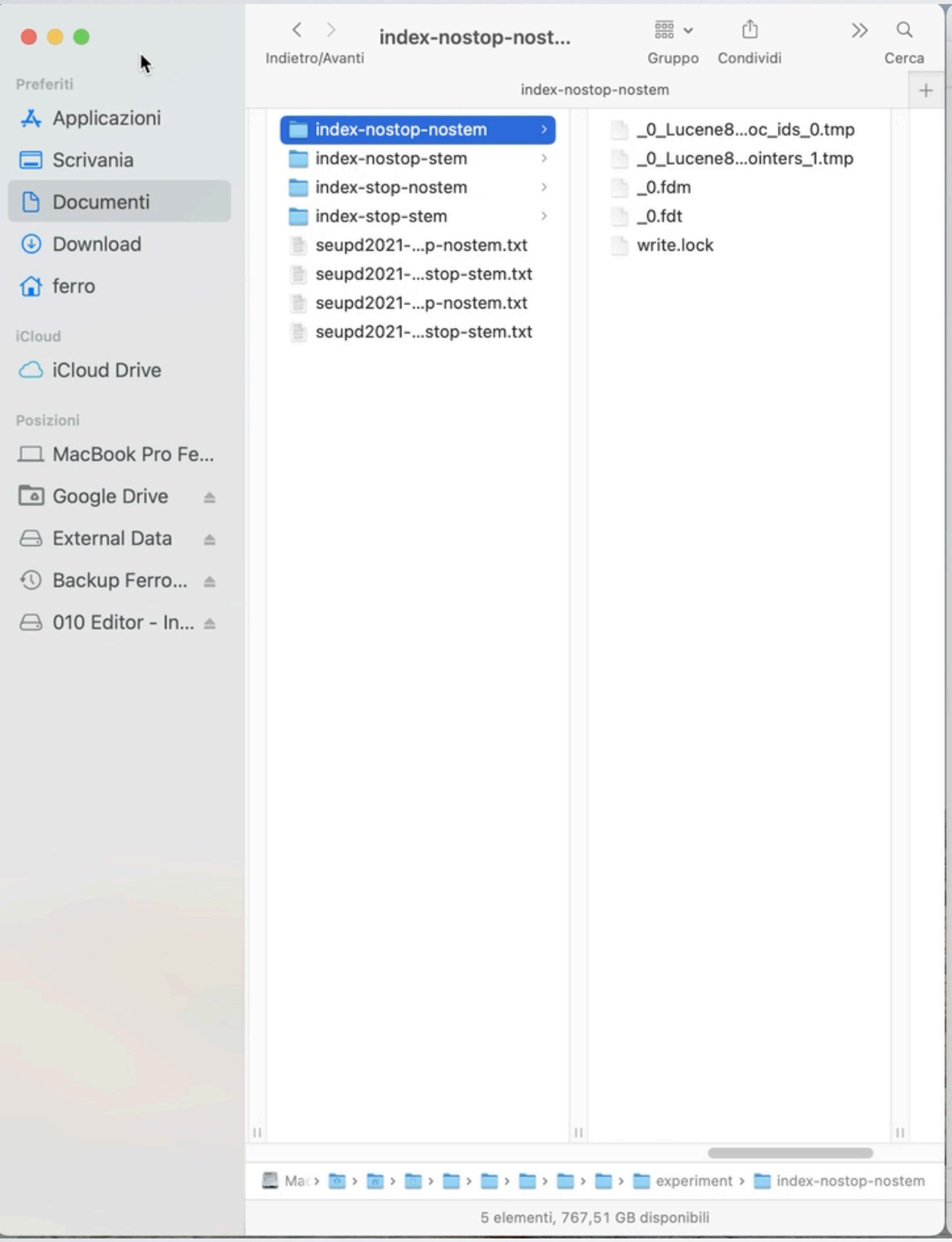
Example of TIPSTER Index: Compound File Format

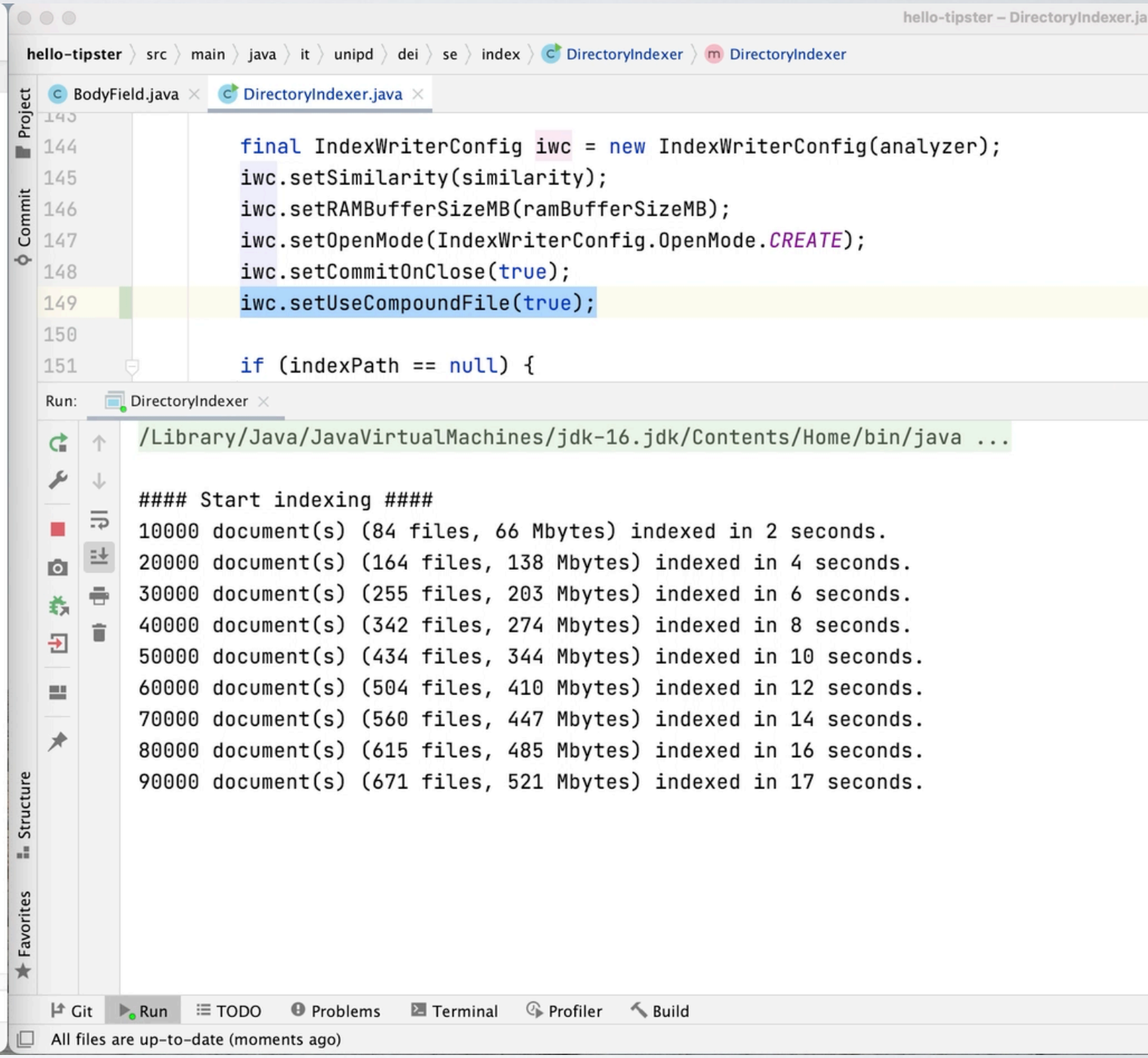




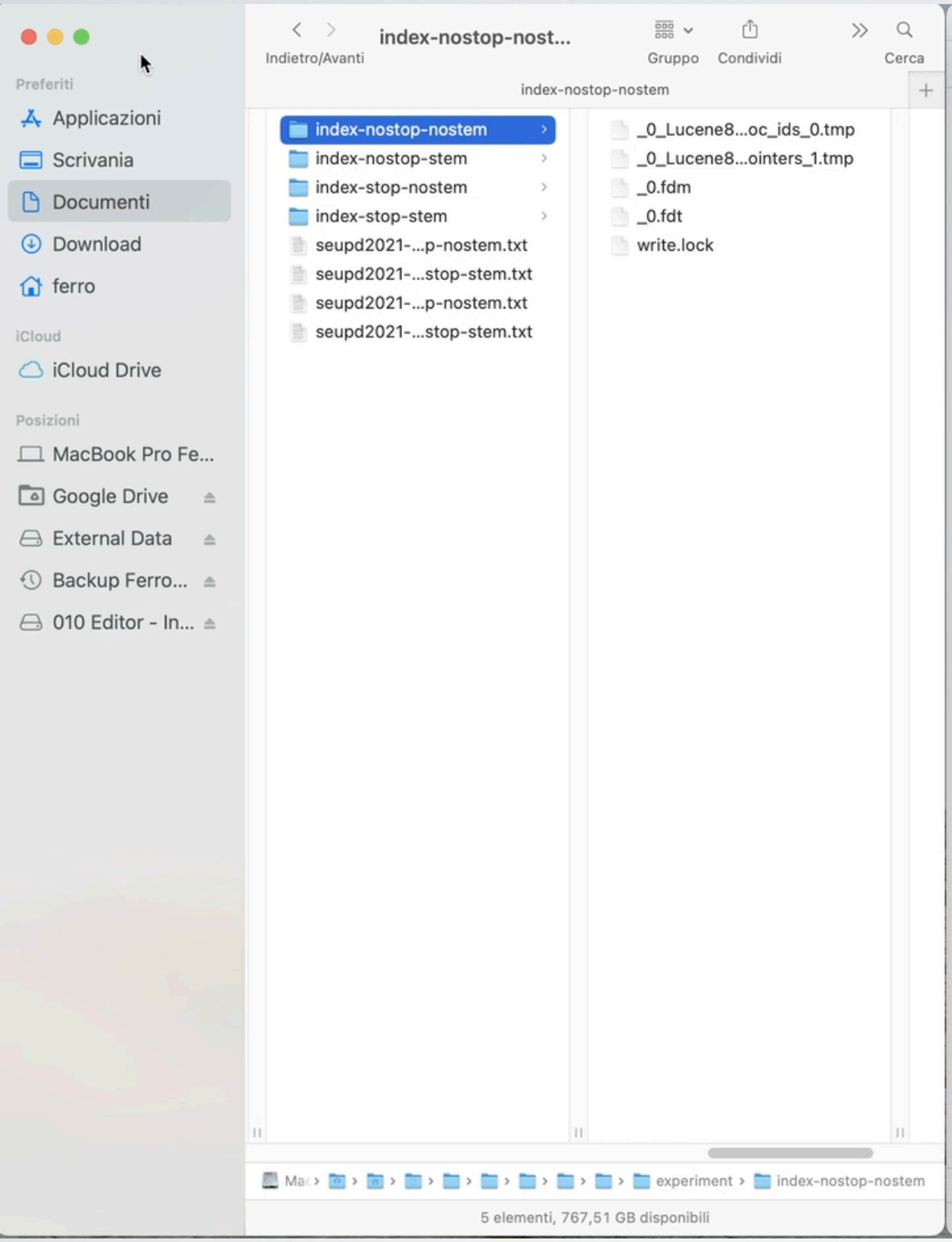
- For each segment
  - the **Compound File Format (.cfs)** merges all the index files into a single file
  - The **Compound Entry Table (.cfe)** holds pointers to all entries, i.e. the index files in the corresponding `.cfs` file
- The CFS is useful for systems which often run out of handles, due to the many files accessed during indexing
  - It is the default for small size segments
- The CFS slightly slows down index creation due to the usual creation of index files that are than packed together into the `.cfs` one

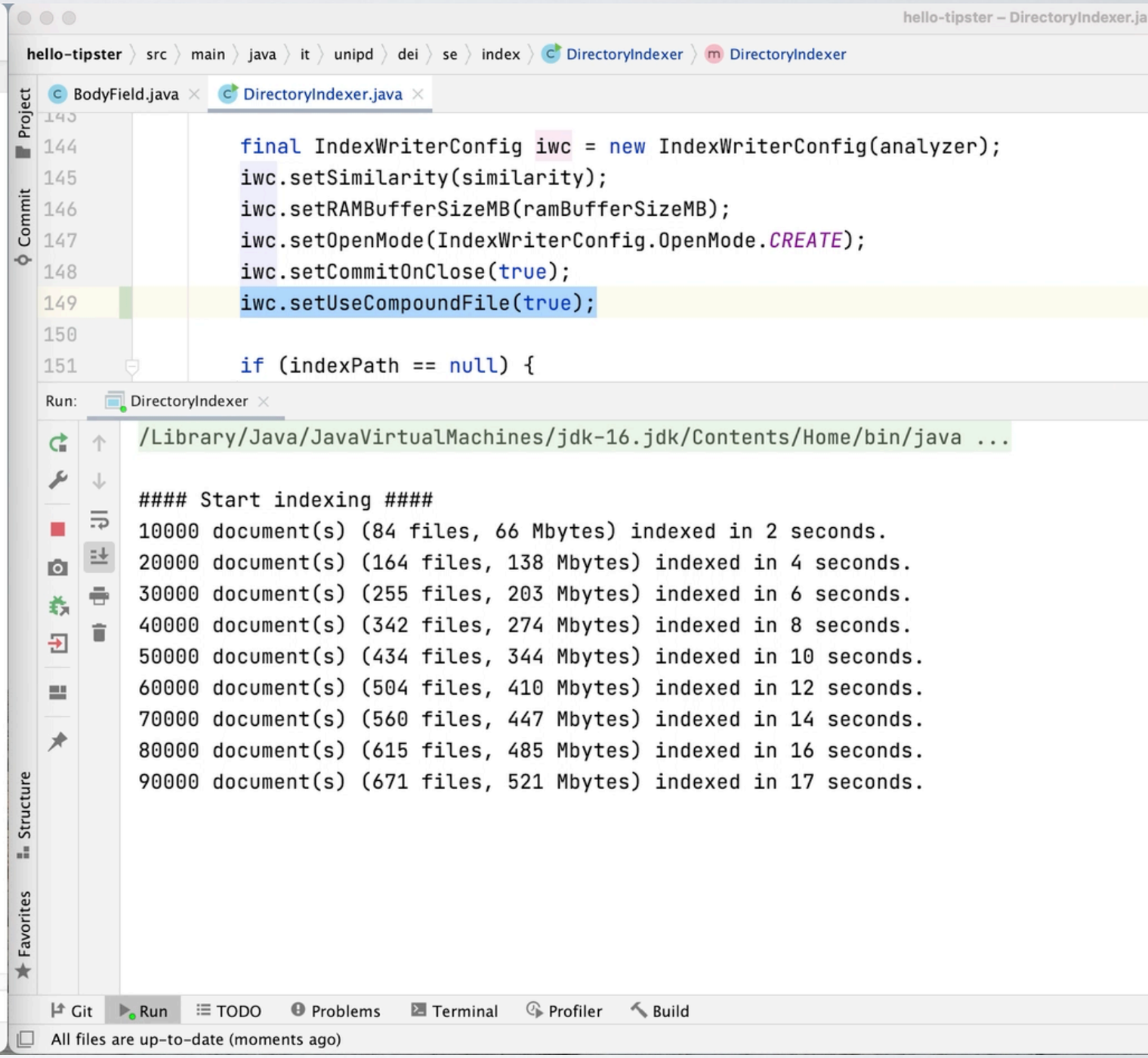
Example of TIPSTER Index: Live Compound File Format





Example of TIPSTER Index: Live Compound File Format



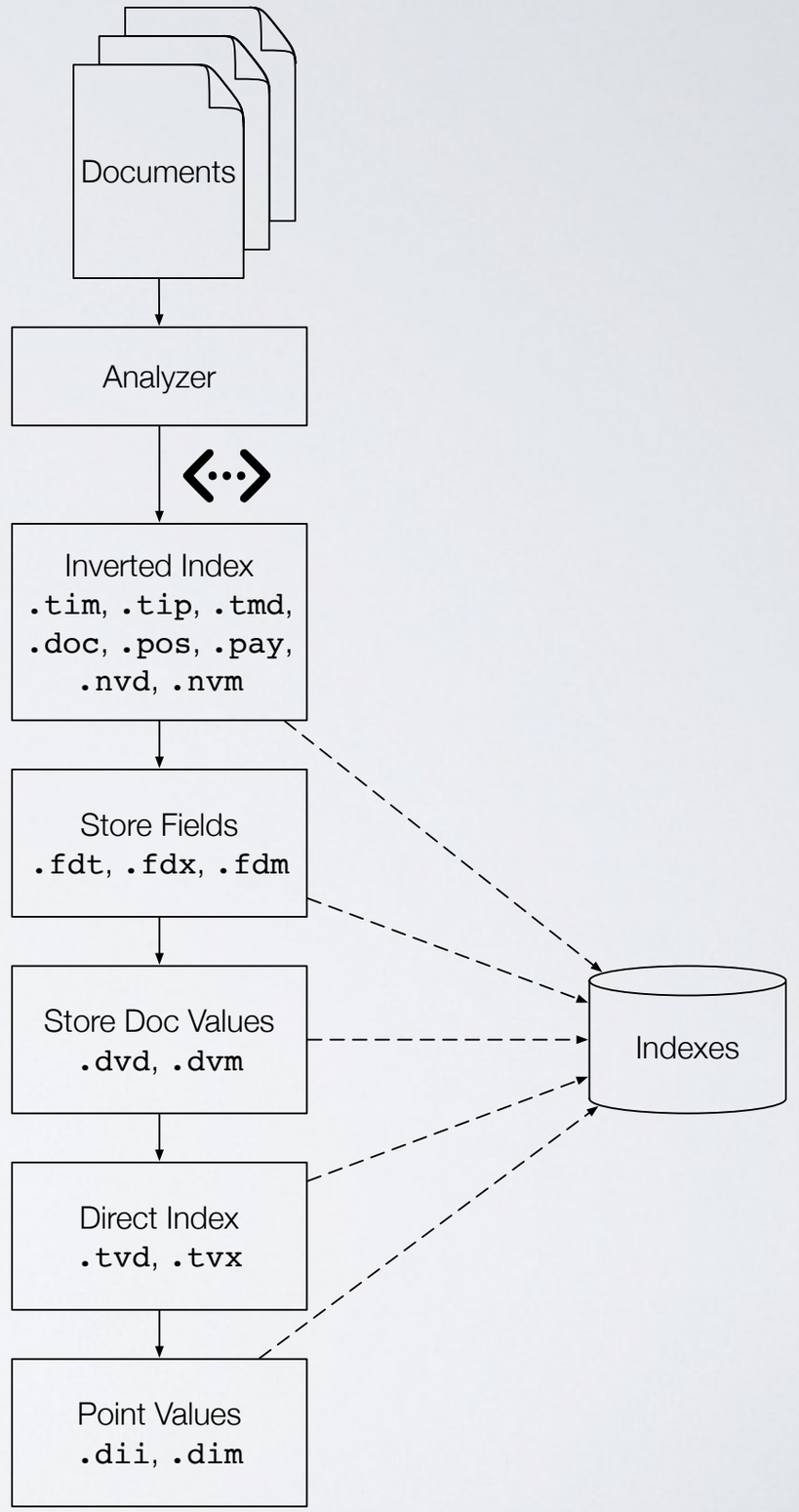


Lucene Codec File Names

- All files belonging to a segment have the same name with varying extensions
  - File names are never re-used
  - The generation is a sequential long integer represented in alpha-numeric (base 36) form
- Overall
  - **Segments File (segments_N)**: information about a commit point
  - **Lock File (write.lock)**: prevents multiple IndexWriters from writing to the same file
- Segment
  - **Segment Info (.si)**: metadata about a segment
- Fields
  - **Fields (.fnm)**: information about the fields
  - **Field Data (.fdt)**: the stored fields (if any) for documents in compressed (LZ4) blocks of 16KB or more
    `FieldType.setStored(true)`
  - **Field Index (.fdx)**: pointers to field data, i.e. first document ID of each block and its offset on disk
  - **Fields Metadata (.fdm)**: metadata about the pointers
- Terms (the inverted index)
  - **Term Dictionary (.tim)**: the term dictionary, contains the list of terms in each field along with per-term statistics (such as docfreq) and pointers to the frequencies, positions, payload and skip data in the .doc, .pos, and .pay files
  - **Term Index (.tip)**: the index into the Term Dictionary, so that it can be accessed randomly
  - **Term Metadata (.tdm)**: metadata about the pointers
  - **Frequencies (.doc)**: the list of documents which contain each term along with frequency
    `FieldType.setIndexOptions(IndexOptions.DOCS_AND_FREQS)`
  - **Positions (.pos)**: position information about where a term occurs in the index
    `FieldType.setIndexOptions(IndexOptions.DOCS_AND_FREQS_AND_POSITIONS)`
  - **Payloads (.pay)**: additional per-position metadata information such as character offsets and user payloads
    `FieldType.setIndexOptions(IndexOptions.DOCS_AND_FREQS_AND_POSITIONS_AND_OFFSET)`
  - **Norms (.nvd, .nvm)**: encodes length and boost factors for documents and fields
    `FieldType.setOmitNorms(false)`
  - **Per-Document Values (.dvd, .dvm)**: encodes additional scoring factors or other per-document information
    `FieldType.setDocValuesType(…)`
- Direct Index
  - **Term Vector Data (.tvd)**: term vector data
    `FieldType.setStoreTermVectors(true)`
  - **Term Vector Index (.tvx)**: offset into the term vector data file
- Other
  - **Point values (.dii, .dim)**: indexed points, if any
    `FieldType.setDimensions(…)`
  - **Live Documents (.liv)**: information about what documents are live

Inside Lucene Indexing

- Lucene provides different types of indexes
  - inverted indexes
  - store field
  - doc values (not for textual fields)
  - direct indexes
- Within `IndexWriter`, the order of indexing on the indexing chain is
  - inverted index
  - store fields
  - doc values
  - direct index
  - point values



The HelloIndex (toy) example

```java
*/
public class BodyField extends Field {

    /**
     * The type of the document body field
     */
    private static final FieldType BODY_TYPE = new FieldType();

    static {
        BODY_TYPE.setIndexOptions(IndexOptions.DOCS_AND_FREQS);
        BODY_TYPE.setTokenized(true);
        BODY_TYPE.setStored(true);
        BODY_TYPE.setStoreTermVectors(true);
    }


    /**
     * Create a new field for the body of a document.
     *
     * @param value the contents of the body of a document.
     */
    public BodyField(final Reader value) { super(HelloIndex.BODY,
```

```java
 */
public void index(final List<Document> docs, final Analyzer analyzer) throws IOException {

    System.out.printf("%n------------ INDEXING DOCUMENTS ------------%n");

    // Open the directory in Lucene
    final Directory directory = FSDirectory.open(indexPath);

    // Utility class for holding all the required configuration for the indexer
    final IndexWriterConfig config = new IndexWriterConfig(analyzer);

    // force to re-create the index if it already exists
    config.setOpenMode(IndexWriterConfig.OpenMode.CREATE);

    // set the similarity. BM25 is already the default one
    config.setSimilarity(new BM25Similarity());

    // set the text-based codec
    //config.setCodec(new Lucene87Codec(Lucene87Codec.Mode.BEST_SPEED));
    config.setCodec(new SimpleTextCodec());

    // prevent using the compound file format
    config.setUseCompoundFile(false);

    // The actual indexer
    final IndexWriter writer = new IndexWriter(directory, config);
```

Inside Lucene Segments

segments_6 ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	08	73	65	67	6D	65	6E	74	73	00	00	00	?×l.segments...
0010h:	0A	B6	67	F0	7F	E1	80	0E	FD	59	C2	AA	2A	0C	48	DB	.¶gð.á€.ýYÂª*.HÛ
0020h:	EC	01	36	08	08	01	08	00	00	00	00	00	00	00	18	06	ì.6..............
0030h:	00	00	00	01	08	08	01	02	5F	35	B6	67	F0	7F	E1	80	........_5¶gð.á€
0040h:	0E	FD	59	C2	AA	2A	0C	48	DB	E9	08	4C	75	63	65	6E	.ýYÂª*.HÛé.Lucen
0050h:	65	38	37	FF	FF	FF	FF	FF	FF	FF	FF	00	00	00	00	FF	e87ÿÿÿÿÿÿÿÿ....ÿ
0060h:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ.
0070h:	00	00	00	01	B6	67	F0	7F	E1	80	0E	FD	59	C2	AA	2A	....¶gð.á€.ýYÂª*
0080h:	0C	48	DB	EB	00	00	00	00	00	00	C0	28	93	E8	00	00	.HÛë......À(“è..
0090h:	00	00	00	00	00	00	97	23	D6	C1							......—#ÖÁ

org.apache.lucene.index

SegmentsInfo

segments_N

Magic Number 0x3fd76c17	segments	Version	…	Lucene Version	…	Seg Count	…	Seg Name	…	Seg Codec	…	Magic Number 0xc02893e8	0	Checksum

header — for each segment — footer

org.apache.lucene.codec

SegmentInfoFormat

.si

Magic Number 0x3fd76c17	Lucene86 SegmentInfo	Version	…	Diagnostic	Files	…	IsCoumpound File	…	Magic Number 0xc02893e8	0	Checksum

_5.si ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	13	4C	75	63	65	6E	65	38	36	53	65	67	?×l..Lucene86Seg
0010h:	6D	65	6E	74	49	6E	66	6F	00	00	00	00	B6	67	F0	7F	mentInfo....¶gð.
0020h:	E1	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	00	00	00	00	á€.ýYÂª*.HÛé....
0030h:	08	00	00	00	08	00	00	00	01	01	00	00	00	08	00	00	....
0040h:	00	08	00	00	00	01	00	00	00	03	FF	0A	02	6F	73	08	..........ÿ..os.
0050h:	4D	61	63	20	4F	53	20	58	0B	6A	61	76	61	2E	76	65	Mac OS X.java.ve
0060h:	6E	64	6F	72	12	4F	72	61	63	6C	65	20	43	6F	72	70	ndor.Oracle Corp
0070h:	6F	72	61	74	69	6F	6E	0C	6A	61	76	61	2E	76	65	72	oration.java.ver
0080h:	73	69	6F	6E	02	31	36	0F	6A	61	76	61	2E	76	6D	2E	sion.16.java.vm.
0090h:	76	65	72	73	69	6F	6E	0A	31	36	2B	33	36	2D	32	32	version.16+36-22
00A0h:	33	31	0E	6C	75	63	65	6E	65	2E	76	65	72	73	69	6F	31.lucene.versio
00B0h:	6E	05	38	2E	38	2E	31	07	6F	73	2E	61	72	63	68	06	n.8.8.1.os.arch.
00C0h:	78	38	36	5F	36	34	14	6A	61	76	61	2E	72	75	6E	74	x86_64.java.runt
00D0h:	69	6D	65	2E	76	65	72	73	69	6F	6E	0A	31	36	2B	33	ime.version.16+3
00E0h:	36	2D	32	32	33	31	06	73	6F	75	72	63	65	05	66	6C	6-2231.source.fl
00F0h:	75	73	68	0A	6F	73	2E	76	65	72	73	69	6F	6E	05	31	ush.os.version.1
0100h:	30	2E	31	36	09	74	69	6D	65	73	74	61	6D	70	0D	31	0.16.timestamp.1
0110h:	36	31	36	36	39	34	33	31	31	38	36	30	0E	06	5F	35	616694311860.._5
0120h:	2E	66	64	74	06	5F	35	2E	74	76	78	06	5F	35	2E	66	.fdt._5.tvx._5.f
0130h:	64	6D	06	5F	35	2E	74	76	6D	11	5F	35	5F	4C	75	63	dm._5.tvm._5_Luc
0140h:	65	6E	65	38	34	5F	30	2E	74	6D	64	06	5F	35	2E	6E	ene84_0.tmd._5.n
0150h:	76	6D	06	5F	35	2E	66	6E	6D	06	5F	35	2E	74	76	64	vm._5.fnm._5.tvd
0160h:	06	5F	35	2E	6E	76	64	11	5F	35	5F	4C	75	63	65	6E	._5.nvd._5_Lucen
0170h:	65	38	34	5F	30	2E	64	6F	63	11	5F	35	5F	4C	75	63	e84_0.doc._5_Luc
0180h:	65	6E	65	38	34	5F	30	2E	74	69	6D	05	5F	35	2E	73	ene84_0.tim._5.s
0190h:	69	11	5F	35	5F	4C	75	63	65	6E	65	38	34	5F	30	2E	i._5_Lucene84_0.
01A0h:	74	69	70	06	5F	35	2E	66	64	78	01	1F	4C	75	63	65	tip._5.fdx..Luce
01B0h:	6E	65	38	37	53	74	6F	72	65	64	46	69	65	6C	64	73	ne87StoredFields
01C0h:	46	6F	72	6D	61	74	2E	6D	6F	64	65	0A	42	45	53	54	Format.mode.BEST
01D0h:	5F	53	50	45	45	44	00	C0	28	93	E8	00	00	00	00	00	_SPEED.À(“è.....
01E0h:	00	00	00	64	0D	7E	9A										...d.~š

Inside Lucene Fields

_5.fnm ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	12	4C	75	63	65	6E	65	36	30	46	69	65	?×l..Lucene60Fie
0010h:	6C	64	49	6E	66	6F	73	00	00	00	02	B6	67	F0	7F	E1	ldInfos....¶gð.á
0020h:	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	00	02	02	69	64	€.ýYÂª*.HÛé....id
0030h:	00	02	01	00	FF	FF	FF	FF	FF	FF	FF	FF	02	1D	50	65	....ÿÿÿÿÿÿÿÿ..Pe
0040h:	72	46	69	65	6C	64	50	6F	73	74	69	6E	67	73	46	6F	rFieldPostingsFo
0050h:	72	6D	61	74	2E	66	6F	72	6D	61	74	08	4C	75	63	65	rmat.format.Luce
0060h:	6E	65	38	34	1D	50	65	72	46	69	65	6C	64	50	6F	73	ne84.PerFieldPos
0070h:	74	69	6E	67	73	46	6F	72	6D	61	74	2E	73	75	66	66	tingsFormat.suff
0080h:	69	78	01	30	00	04	62	6F	64	79	01	01	02	00	FF	FF	ix.0..body....ÿÿ
0090h:	FF	FF	FF	FF	FF	FF	02	1D	50	65	72	46	69	65	6C	64	ÿÿÿÿÿÿ..PerField
00A0h:	50	6F	73	74	69	6E	67	73	46	6F	72	6D	61	74	2E	66	PostingsFormat.f
00B0h:	6F	72	6D	61	74	08	4C	75	63	65	6E	65	38	34	1D	50	ormat.Lucene84.P
00C0h:	65	72	46	69	65	6C	64	50	6F	73	74	69	6E	67	73	46	erFieldPostingsF
00D0h:	6F	72	6D	61	74	2E	73	75	66	66	69	78	01	30	00	C0	ormat.suffix.0.À
00E0h:	28	93	E8	00	00	00	00	00	00	00	00	DE	F2	B8	2D		(“è........Þò¸-

org.apache.lucene.codec
FieldInfosFormat

.fnm

Magic Number `0x3fd76c17`	`Lucene60 FieldInfos`	Version	Fields Count	Field Name	Field Number	Term Vectors	Omit Norms	Payloads	Docs Only	Docs Freqs	Docs Freqs Pos	Docs Freqs Pos Offsets	Doc Values	…	Attributes	…	Magic Number `0xc02893e8`	0	Checksum

for each field

org.apache.lucene.codec
StoredFieldsFormat

Only when you actually store the contents of a field `FieldType.setStored(true)`

.fdt

_5.fdt ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	1C	4C	75	63	65	6E	65	38	37	53	74	6F	?×l..Lucene87Sto
0010h:	72	65	64	46	69	65	6C	64	73	46	61	73	74	44	61	74	redFieldsFastDat
0020h:	61	00	00	00	03	B6	67	F0	7F	E1	80	0E	FD	59	C2	AA	a....¶gð.á€.ýYÂª
0030h:	2A	0C	48	DB	E9	00	00	06	00	02	08	6D	63	A1	02	25	*.HÛé.....mc¡.%
0040h:	03	27	27	27	27	27	27	27	27	27	24	20	00	02	F0	16	.''''''''''$ ..ð.
0050h:	64	31	08	67	54	68	65	20	71	75	6F	6B	6B	61	2C	20	d1.gThe quokka,
0060h:	74	68	65	20	6F	6E	6C	79	20	6D	65	6D	62	65	72	20	the only member
0070h:	6F	66	20	74	68	F0	16	65	20	67	65	6E	75	73	20	53	of thð.e genus S
0080h:	65	74	6F	6E	69	78	2C	20	69	73	20	61	20	73	6D	61	etonix, is a sma
0090h:	6C	6C	20	6D	61	72	73	75	70	69	61	6C	F0	16	20	61	ll marsupialð. a
00A0h:	62	6F	75	74	20	74	68	65	20	73	69	7A	65	20	6F	66	bout the size of
00B0h:	20	61	20	64	6F	6D	65	73	74	69	63	20	63	61	74	00	a domestic cat.
00C0h:	02	64	32	F0	16	08	5D	57	6F	6D	62	61	74	73	20	61	.d2ð..]Wombats a
00D0h:	72	65	20	73	6D	61	6C	6C	2C	20	73	68	6F	72	74	2D	re small, short-
00E0h:	6C	65	67	67	65	64	2C	20	6D	75	F0	16	73	63	75	6C	legged, muð.scul
00F0h:	61	72	20	71	75	61	64	72	75	70	65	64	61	6C	20	6D	ar quadrupedal m
0100h:	61	72	73	75	70	69	61	6C	73	20	74	68	61	74	20	61	arsupials that a
0110h:	72	F0	16	65	20	6E	61	74	69	76	65	20	74	6F	20	41	rð.e native to A
0120h:	75	73	74	72	61	6C	69	61	00	02	64	33	08	9A	01	51	ustralia..d3.š.Q
0130h:	75	6F	6B	6B	61	73	20	68	F0	16	61	76	65	20	6C	69	uokkas hð.ave li
0140h:	74	74	6C	65	20	66	65	61	72	20	6F	66	20	68	75	6D	ttle fear of hum
0150h:	61	6E	73	20	61	6E	64	20	63	6F	6D	6D	6F	6E	6C	F0	ans and commonlð
0160h:	16	79	20	61	70	70	72	6F	61	63	68	20	70	65	6F	70	.y approach peop
0170h:	6C	65	20	63	6C	6F	73	65	6C	79	2C	20	70	61	72	74	le closely, part
0180h:	69	63	75	6C	61	72	F0	16	6C	79	20	6F	6E	20	52	6F	icularð.ly on Ro
0190h:	74	74	6E	65	73	74	20	49	73	6C	61	6E	64	20	69	6E	ttnest Island in
01A0h:	20	41	75	73	74	72	61	6C	69	61	2C	20	77	F0	13	68	Australia, wð.h
01B0h:	65	72	65	20	61	20	70	72	65	76	61	6C	65	6E	74	20	ere a prevalent
01C0h:	70	6F	70	75	6C	61	74	69	6F	6E	20	65	78	69	73	74	population exist
01D0h:	73	C0	28	93	E8	00	00	00	00	00	00	00	00	A1	F5	0D	sÀ(“è........¡õ.
01E0h:	77																w

.fdx

_5.fdx ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	16	4C	75	63	65	6E	65	38	35	46	69	65	?×l..Lucene85Fie
0010h:	6C	64	73	49	6E	64	65	78	49	64	78	00	00	00	00	B6	ldsIndexIdx....¶
0020h:	67	F0	7F	E1	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	00	gð.á€.ýYÂª*.HÛé.
0030h:	C0	28	93	E8	00	00	00	00	00	00	00	00	32	19	E3	7C	À(“è........2.ã|
0040h:																	

.fdm

_5.fdm ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	17	4C	75	63	65	6E	65	38	35	46	69	65	?×l..Lucene85Fie
0010h:	6C	64	73	49	6E	64	65	78	4D	65	74	61	00	00	00	03	ldsIndexMeta....
0020h:	B6	67	F0	7F	E1	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	¶gð.á€.ýYÂª*.HÛé
0030h:	00	80	C0	25	02	00	00	00	03	00	00	00	0A	00	00	00	.€À%............
0040h:	02	00	00	00	00	00	00	00	30	00	00	00	00	00	00	00	........0.......
0050h:	00	40	40	00	00	00	00	00	00	00	00	00	00	00	00	00	.@@.............
0060h:	00	00	00	00	00	30	00	00	00	00	00	00	00	36	43	CD	.....0.......6CÍ
0070h:	80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	€...............
0080h:	00	00	30	00	00	00	00	00	00	01	D1	01	FD	07	C0	28	..0.......Ñ.ý.À(
0090h:	93	E8	00	00	00	00	00	00	00	00	92	36	EC	46			“è........’6ìF

Inside Lucene Inverted Index

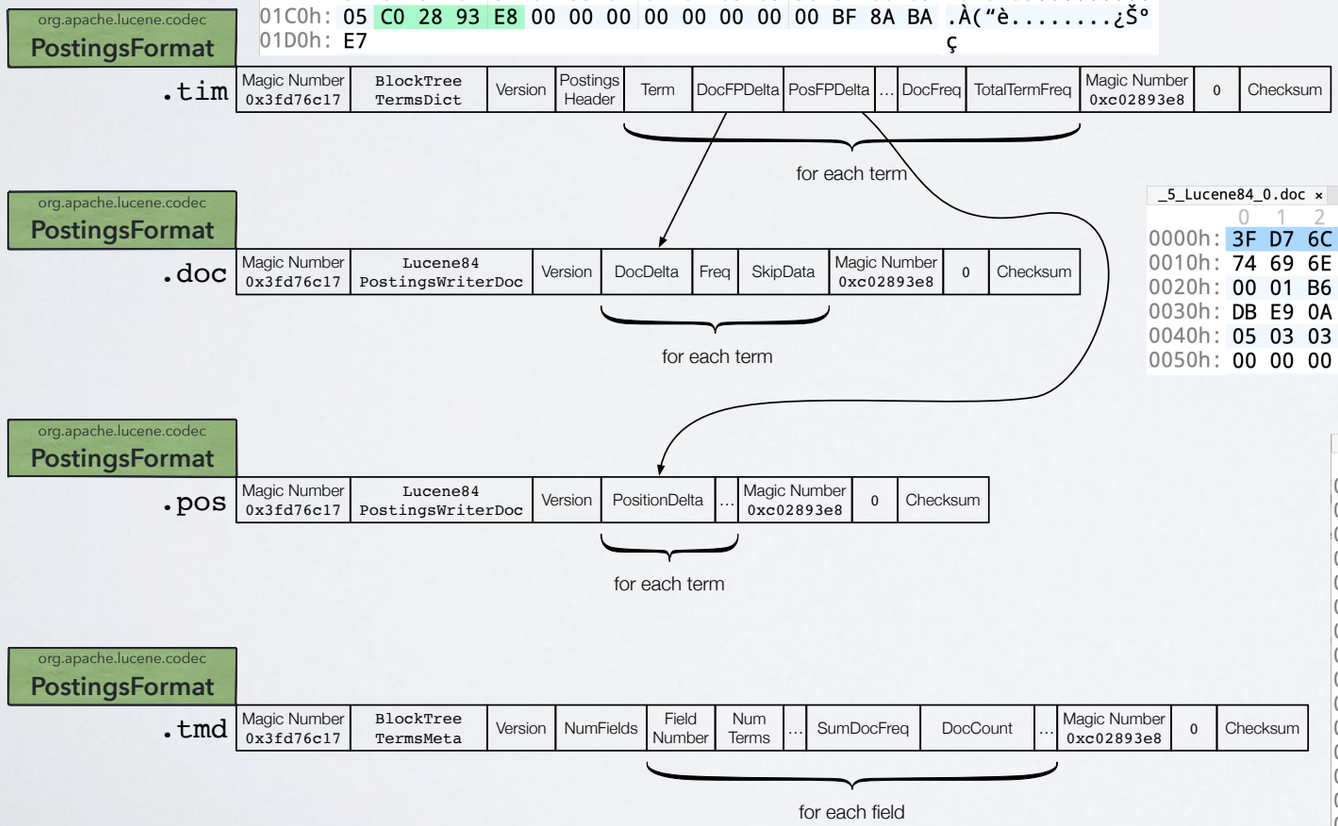
_5_Lucene84_0.tim

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	12	42	6C	6F	63	6B	54	72	65	65	54	65	?×l..BlockTreeTe
0010h:	72	6D	73	44	69	63	74	00	00	00	06	B6	67	F0	7F	E1	rmsDict....¶gð.á
0020h:	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	0A	4C	75	63	65	€.ýYÂª*.HÛé.Luce
0030h:	6E	65	38	34	5F	30	5B	A4	10	61	61	62	6F	75	74	61	ne84_0[¤.aabouta
0040h:	6E	64	61	70	70	72	6F	61	63	68	61	72	65	61	75	73	ndapproachareaus
0050h:	74	72	61	6C	69	61	63	61	74	63	6C	6F	73	65	6C	79	traliacatclosely
0060h:	63	6F	6D	6D	6F	6E	6C	79	64	6F	6D	65	73	74	69	63	commonlydomestic
0070h:	65	78	69	73	74	73	66	65	61	72	67	65	6E	75	73	68	existsfeargenush
0080h:	61	76	65	68	75	6D	61	6E	73	69	6E	69	73	69	73	6C	avehumansinisisl
0090h:	61	6E	64	6C	65	67	67	65	64	6C	69	74	74	6C	65	6D	andleggedlittlem
00A0h:	61	72	73	75	70	69	61	6C	6D	61	72	73	75	70	69	61	arsupialmarsupia
00B0h:	6C	73	6D	65	6D	62	65	72	6D	75	73	63	75	6C	61	72	lsmembermuscular
00C0h:	6E	61	74	69	76	65	6F	66	6F	6E	6F	6E	6C	79	70	61	nativeofononlypa
00D0h:	72	74	69	63	75	6C	61	72	6C	79	70	65	6F	70	6C	65	rticularlypeople
00E0h:	70	6F	70	75	6C	61	74	69	6F	6E	70	72	65	76	61	6C	populationpreval
00F0h:	65	6E	74	71	75	61	64	72	75	70	65	64	61	6C	71	75	entquadrupedalqu
0100h:	6F	6B	6B	61	71	75	6F	6B	6B	61	73	72	6F	74	74	6E	okkaquokkasrottn
0110h:	65	73	74	73	65	74	6F	6E	69	78	73	68	6F	72	74	73	estsetonixshorts
0120h:	69	7A	65	73	6D	61	6C	6C	74	68	61	74	74	68	65	74	izesmallthatthet
0130h:	6F	77	68	65	72	65	77	6F	6D	62	61	74	73	5A	01	05	owherewombatsZ..
0140h:	03	08	03	09	03	07	08	08	06	04	05	04	06	02	02	06	....
0150h:	06	06	09	0A	06	08	06	02	02	04	0C	06	0A	09	0B	06	....
0160h:	07	08	07	05	04	05	04	03	02	05	07	11	04	01	05	02	....
0170h:	01	04	00	25	04	01	19	04	00	01	02	03	05	31	7C	06	...%.........1|.
0180h:	00	09	01	03	00	04	00	09	01	07	09	01	07	09	01	01	....
0190h:	07	09	03	05	07	05	03	05	01	00	06	02	07	09	01	01	....
01A0h:	01	03	03	09	01	07	05	03	00	04	01	03	05	05	03	07	....
01B0h:	34	64	31	64	32	64	33	07	02	01	05	05	90	01	00	05	4d1d2d3.........
01C0h:	05	C0	28	93	E8	00	00	00	00	00	00	00	BF	8A	BA		.À(“è........¿Šº
01D0h:	E7																ç

.tip

_5_Lucene84_0.tip

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	13	42	6C	6F	63	6B	54	72	65	65	54	65	?×l..BlockTreeTe
0010h:	72	6D	73	49	6E	64	65	78	00	00	00	06	B6	67	F0	7F	rmsIndex....¶gð.
0020h:	E1	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	0A	4C	75	63	á€.ýYÂª*.HÛé.Luc
0030h:	65	6E	65	38	34	5F	30	00	00	C0	28	93	E8	00	00	00	ene84_0..À(“è...
0040h:	00	00	00	00	00	57	B3	2C	73								.....W³,s



_5_Lucene84_0.doc

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	19	4C	75	63	65	6E	65	38	34	50	6F	73	?×l..Lucene84Pos
0010h:	74	69	6E	67	73	57	72	69	74	65	72	44	6F	63	00	00	tingsWriterDoc..
0020h:	00	01	B6	67	F0	7F	E1	80	0E	FD	59	C2	AA	2A	0C	48	..¶gð.á€.ýYÂª*.H
0030h:	DB	E9	0A	4C	75	63	65	6E	65	38	34	5F	30	4C	00	02	Ûé.Lucene84_0L..
0040h:	05	03	03	00	02	05	01	03	C0	28	93	E8	00	00	00	00	........À(“è....
0050h:	00	00	00	00	FB	A4	07	41									....û¤.A

_5_Lucene84_0.tmd

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	12	42	6C	6F	63	6B	54	72	65	65	54	65	?×l..BlockTreeTe
0010h:	72	6D	73	4D	65	74	61	00	00	00	06	B6	67	F0	7F	E1	rmsMeta....¶gð.á
0020h:	80	0E	FD	59	C2	AA	2A	0C	48	DB	E9	0A	4C	75	63	65	€.ýYÂª*.HÛé.Luce
0030h:	6E	65	38	34	5F	30	3F	D7	6C	17	1B	4C	75	63	65	6E	ne84_0?×l..Lucen
0040h:	65	38	34	50	6F	73	74	69	6E	67	73	57	72	69	74	65	e84PostingsWrite
0050h:	72	54	65	72	6D	73	00	00	00	01	B6	67	F0	7F	E1	80	rTerms....¶gð.á€
0060h:	0E	FD	59	C2	AA	2A	0C	48	DB	E9	0A	4C	75	63	65	6E	.ýYÂª*.HÛé.Lucen
0070h:	65	38	34	5F	30	80	01	02	01	2D	02	DA	01	37	31	03	e84_0€...-.Ú.71.
0080h:	01	61	07	77	6F	6D	62	61	74	73	37	3F	D7	6C	17	03	.a.wombats7?×l..
0090h:	46	53	54	00	00	00	07	01	03	01	DA	02	00	00	01	00	FST.....Ú.....
00A0h:	03	02	BE	0D	03	03	02	64	31	02	64	33	38	3F	D7	6C	..¾....d1.d38?×l
00B0h:	17	03	46	53	54	00	00	00	07	01	03	0D	BE	02	00	00	..FST.....¾...
00C0h:	01	00	00	00	00	00	00	00	49	00	00	00	00	00	00	01	........I.......
00D0h:	D1	C0	28	93	E8	00	00	00	00	00	00	00	00	52	8A	28	ÑÀ(“è.....RŠ(
00E0h:	FB																û

Lucene Index Summary

Index Commit (`segments_N`)		
Seg Name	Seg Codec	…
_5	Lucene87	…

Segment Info (`_5.si`)	
Files	…
_5.fnm	…
_5.fdt	…
_5.fdx	…
_5.fdm	…
_5_Lucene84_0.tim	…
_5_Lucene84_0.doc	…
_5_Lucene84_0.pos	…
_5_Lucene84_0.tmd	…
_5.tvd	…
_5.tvx	…

Field Info (`_5.fnm`)

Field Name	Field Number	Term Vectors?	Omit Norms?	Payloads?	Docs Only	Docs Freqs	Docs Freqs Pos	Docs Freqs Pos Offsets	Docs Values	…
id	0	false	false	false	true	false	false	false	NONE	…
body	1	true	false	false	false	true	false	false	NONE	…

Postings (`_5_Lucene84_0.tim`)

Term	PosFPDelta	DocFreq	TotalTermFreq	…	DocFPDelta
a		2	3	…	
about		1	1	…	
and		1	1	…	
approach		1	1	…	
are		1	2	…	
australia		2	2	…	
…		…	…	…	

Frequency (`_5_Lucene84_0.doc`)

DocDelta	Freq	…
0	2	…
1	1	…
0	1	…
2	1	…
2	1	…
1	2	…
1	1	…
2	1	…
…	…	…

Lucene Codecs

What's in a Codec?

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

org.apache.lucene.codecs

Class Codec

java.lang.Object
org.apache.lucene.codecs.Codec

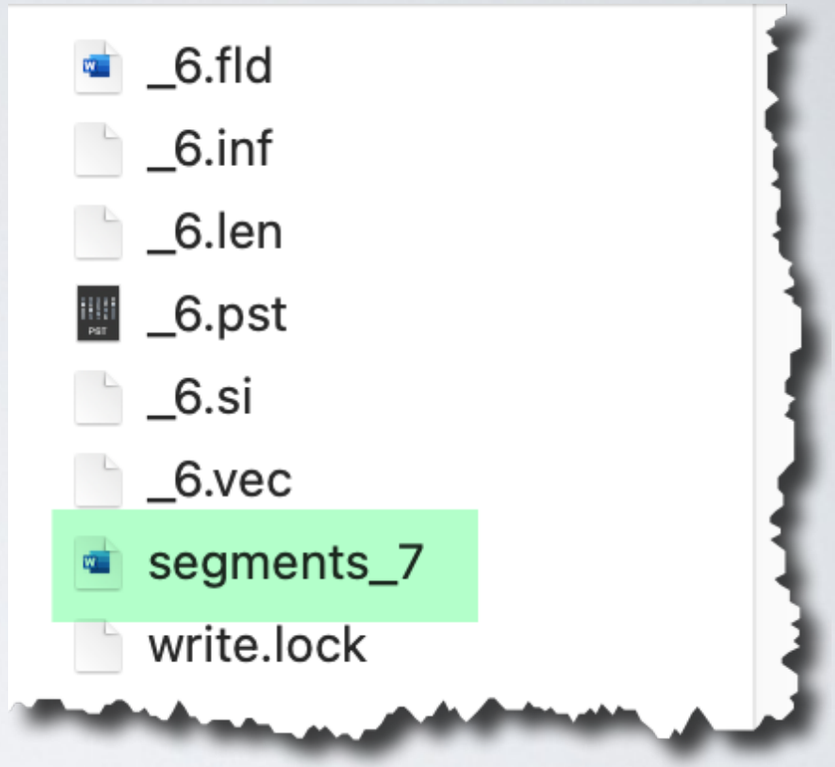
Method Summary

All Methods | Static Methods | Instance Methods | Abstract Methods | Concrete Methods

Modifier and Type	Method and Description
static Set<String>	availableCodecs() returns a list of all available codec names
abstract CompoundFormat	compoundFormat() Encodes/decodes compound files
abstract DocValuesFormat	docValuesFormat() Encodes/decodes docvalues
abstract FieldInfosFormat	fieldInfosFormat() Encodes/decodes field infos file
static Codec	forName(String name) looks up a codec by name
static Codec	getDefault() expert: returns the default codec used for newly created IndexWriterConfigs.
String	getName() Returns this codec's name
abstract LiveDocsFormat	liveDocsFormat() Encodes/decodes live docs
abstract NormsFormat	normsFormat() Encodes/decodes document normalization values
abstract PointsFormat	pointsFormat() Encodes/decodes points index
abstract PostingsFormat	postingsFormat() Encodes/decodes postings
static void	reloadCodecs(ClassLoader classloader) Reloads the codec list from the given ClassLoader.
abstract SegmentInfoFormat	segmentInfoFormat() Encodes/decodes segment info file
static void	setDefault(Codec codec) expert: sets the default codec used for newly created IndexWriterConfigs.
abstract StoredFieldsFormat	storedFieldsFormat() Encodes/decodes stored fields
abstract TermVectorsFormat	termVectorsFormat() Encodes/decodes term vectors

The SimpleText Codec

- It encodes an index in a textual format
  - Not to be used in production but useful for learning
- Note that file name and extensions are different from those of the default Lucene codec
  - The `segments_N` file has always the same format because it is written by the **IndexWriter** and not by a `Codec`



segments_7 ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	3F	D7	6C	17	08	73	65	67	6D	65	6E	74	73	00	00	00	?×l..segments...
0010h:	0A	AA	71	85	D6	CC	5C	88	53	E1	D4	C5	CB	F3	F1	F4	.ªq…ÖÌ\ˆSáÔÅËóñô
0020h:	18	01	37	08	08	01	08	00	00	00	00	00	00	00	1C	07	..7..............
0030h:	00	00	00	01	08	08	01	02	5F	36	AA	71	85	D6	CC	5C	........_6ªq…ÖÌ\
0040h:	88	53	E1	D4	C5	CB	F3	F1	F4	15	0A	53	69	6D	70	6C	ˆSáÔÅËóñô..Simpl
0050h:	65	54	65	78	74	FF	FF	FF	FF	FF	FF	FF	FF	00	00	00	eTextÿÿÿÿÿÿÿÿÿ...
0060h:	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0070h:	FF	00	00	00	00	01	AA	71	85	D6	CC	5C	88	53	E1	D4	ÿ.....ªq…ÖÌ\ˆSáÔ
0080h:	C5	CB	F3	F1	F4	17	00	00	00	00	00	00	C0	28	93	E8	ÅËóñô.......À(“è
0090h:	00	00	00	00	00	00	00	00	BA	AF	F2	D9					........º¯òÙ

SimpleText Codec: Segment Info

~/Documents/progetti/software/search-engines/se-unipd/hello-index/experiment/index/_6.si

```
    version 8.8.1
    min version 8.8.1
    number of documents 3
    uses compound file false
    diagnostics 10
      key os
      value Mac OS X
      key java.vendor
      value Oracle Corporation
      key java.version
      value 16
      key java.vm.version
      value 16+36-2231
      key lucene.version
      value 8.8.1
      key os.arch
      value x86_64
      key java.runtime.version
      value 16+36-2231
      key source
      value flush
      key os.version
      value 10.16
      key timestamp
      value 1616749553902
    attributes 0
    files 6
      file _6.vec
      file _6.pst
      file _6.inf
      file _6.len
      file _6.fld
      file _6.si
    id ™qÖ÷Ã\\àS·‘≈ÀÛÒÙ¿
    sort 0
checksum 0000000000197842539
```

SimpleText Codec: Fields

~/Documents/progetti/software/search-engines/se-unipd/hello-index/experiment/index/_6.inf

```
number of fields 2
  name id
  number 0
  index options DOCS
  term vectors false
  payloads false
  norms false
  doc values NONE
  doc values gen -1
  attributes 0
  data dimensional count 0
  index dimensional count 0
  dimensional num bytes 0
  soft-deletes false
  name body
  number 1
  index options DOCS_AND_FREQS
  term vectors true
  payloads false
  norms true
  doc values NONE
  doc values gen -1
  attributes 0
  data dimensional count 0
  index dimensional count 0
  dimensional num bytes 0
  soft-deletes false
checksum 00000000004031967692

```

~/Documents/progetti/software/search-engines/se-unipd/hello-index/experiment/index/_6.fld master

```
doc 0
  field 0
    name id
    type string
    value d1
  field 1
    name body
    type string
    value The quokka, the only member of the genus Setonix, is a small marsupial about the size of a domestic cat
doc 1
  field 0
    name id
    type string
    value d2
  field 1
    name body
    type string
    value Wombats are small, short-legged, muscular quadrupedal marsupials that are native to Australia
doc 2
  field 0
    name id
    type string
    value d3
  field 1
    name body
    type string
    value Quokkas have little fear of humans and commonly approach people closely, particularly on Rottnest Island in Au
END
checksum 00000000000156156961

```

SimpleText Codec: Inverted Index

~/Documents/progetti/software/search-engines/se-unipd/hello-index/experiment/index/_6.pst master

```
field body
  term a
    doc 0
      freq 2
    doc 2
      freq 1
  term about
    doc 0
      freq 1
  term and
    doc 2
      freq 1
  term approach
    doc 2
      freq 1
  term are
    doc 1
      freq 2
  term australia
    doc 1
      freq 1
    doc 2
      freq 1
  term cat
    doc 0
      freq 1
  term closely
    doc 2
      freq 1
  term commonly
    doc 2
```

SimpleText Codec: Direct Index

~/Documents/progetti/software/search-engines/se-unipd/hello-index/experiment/index/_6.vec

```
doc 0
  numfields 1
  field 1
    name body
    positions false
    offsets   false
    payloads  false
    numterms 15
    term a
      freq 2
    term about
      freq 1
    term cat
      freq 1
    term domestic
      freq 1
    term genus
      freq 1
    term is
      freq 1
    term marsupial
      freq 1
    term member
      freq 1
    term of
      freq 2
    term only
      freq 1
    term quokka
      freq 1
    term setonix
      freq 1
    term size
      freq 1
    term small
      freq 1
    term the
      freq 4
doc 1
  numfields 1
  field 1
    name body
    positions false
    offsets   false
    payloads  false
    numterms 12
    term are
      freq 2
    term australia
      freq 1
    term tagged
```

IndexWriterConfig: Full Indexing Configuration

org.apache.lucene.index

Class IndexWriterConfig

java.lang.Object
  org.apache.lucene.index.LiveIndexWriterConfig
    org.apache.lucene.index.IndexWriterConfig

```
public final class IndexWriterConfig
extends LiveIndexWriterConfig
```

Holds all the configuration that is used to create an `IndexWriter`. Once `IndexWriter` has been created with this object, changes to this object will not affect the `IndexWriter` instance. For that, use `LiveIndexWriterConfig` that is returned from `IndexWriter.getConfig()`.

All setter methods return `IndexWriterConfig` to allow chaining settings conveniently, for example:

```
IndexWriterConfig conf = new IndexWriterConfig(analyzer);
conf.setter1().setter2();
```

Since:

3.1

See Also:

`IndexWriter.getConfig()`

Nested Class Summary

Nested Classes

Modifier and Type	Class and Description
static class	**IndexWriterConfig.OpenMode** Specifies the open mode for **IndexWriter**.

Field Summary

Fields

Modifier and Type	Field and Description
static boolean	**DEFAULT_COMMIT_ON_CLOSE** Default value for whether calls to **IndexWriter.close()** include a commit.
static int	**DEFAULT_MAX_BUFFERED_DELETE_TERMS** Disabled by default (because IndexWriter flushes by RAM usage by default).
static int	**DEFAULT_MAX_BUFFERED_DOCS** Disabled by default (because IndexWriter flushes by RAM usage by default).
static long	**DEFAULT_MAX_FULL_FLUSH_MERGE_WAIT_MILLIS** Default value for time to wait for merges on commit or getReader (when using a **MergePolicy** that implements **MergePolicy.findFullFlushMerges(org.apache.lucene.index.MergeTrigger, org.apache.lucene.index.SegmentInfos, org.apache.lucene.index.MergePolicy.MergeContext)**).
static double	**DEFAULT_RAM_BUFFER_SIZE_MB** Default value is 16 MB (which means flush when buffered docs consume approximately 16 MB RAM).

IndexWriterConfig: Further Indexing Options

- **OpenMode**: defines how the index is opened (create, append, create or append)
- **Analyzer**: defines how the document pre-processing (tokenization, lexical analysis, …) is performed
- **Similarity**: defines the adopted scoring algorithm. By default, Lucene uses the BM25 algorithm
- **Codec**: defines Lucene's internal encoders and decoders for all types of indexes
- **UseCompoundFile**: whether newly written segments should be packed in a compound file
- **IndexDeletionPolicy**: enables the management of commit points to implement such functionality as snapshots. By default DeletionPolicy only retains the last commit point
- **CommitOnClose**: determines whether a commit is performed before closing an index
- **MergePolicy**: determines how segments are merged.
- **MergeScheduler**: determines when segments are merged.
- **FlushPolicy**: determines when in-memory buffers are flushed. By default the timing will depend on the size of RAM and number of documents.
- **MaxBufferedDoc**: the minimal number of documents required before the buffered in-memory documents are flushed
- **RAMBufferSizeMB**: the amount of RAM that may be used for buffering added documents and deletions before they are flushed. Generally for faster indexing performance it's best to flush by RAM usage instead of document count and use as large a RAM buffer as you can

Extracting Information from a Lucene Index

Printing the Inverted Index

```java
 */
public void printInvertedIndex() throws IOException {

    System.out.printf("%n------------- PRINTING THE INVERTED INDEX -------------%n");

    // Open the directory in Lucene
    final Directory dir = FSDirectory.open(indexPath);

    // Open the index
    final IndexReader index = DirectoryReader.open(dir);

    // The inverted index. Keys are terms; values are lists of (docID, term frequency) pairs
    final TreeMap<String, List<Map.Entry<String, Long>>> invertedIndex = new TreeMap<>();

    // Iterate over each document in the index
    for (int i = 0, docs = index.numDocs(); i < docs; i++) {

        // Read the document and get its identifier
        String docID = index.document(i).get(ID);

        // Get the vector of terms for that document
        Terms terms = index.getTermVector(i, BODY);

        // Get an iterator over the vector of terms
        TermsEnum termsEnum = terms.iterator();

        // Iterate until there are terms
        for (BytesRef term = termsEnum.next(); term != null; term = termsEnum.next()) {

            // Get the text string of the term
            String termstr = term.utf8ToString();

            // Get the total frequency of the term
            long freq = termsEnum.totalTermFreq();

            // Create a new (docID, term frequency) pair
            Map.Entry<String, Long> entry = new AbstractMap.SimpleEntry<>(docID, freq);
```

The inverted index is re-created in-memory as a map of terms -> postings list

Iterate over each document in the index and get the terms of vector for that document

Iterate over each term in the document and retrieve its frequency, to create an entry for the map

Printing the Inverted Index

```
            // update the inverted index with the new entry
            invertedIndex.compute(termstr, (k, v) -> {

                // if the term is not already in the index, create a new list for its pairs
                if (v==null) {
                    v = new ArrayList<>();
                }

                // add the pair to the list
                v.add(entry);

                // return the updated list
                return v;
            });
        }
    }

    // close the index and the directory
    index.close();
    dir.close();

    // Print the inverted index to the console
    invertedIndex.forEach( (k, v) -> {

        System.out.printf("+ %s ->", k);

        v.forEach((e) -> {
            System.out.printf(" %s:%d", e.getKey(), e.getValue());
        });

        System.out.printf("%n");

    });

    System.out.printf("------------------------------------------------------%n");
}
```

Add the entry to the map, creating a new postings list if the term is not already present

Print each element of the map to the console

Printing Vocabulary Statistics

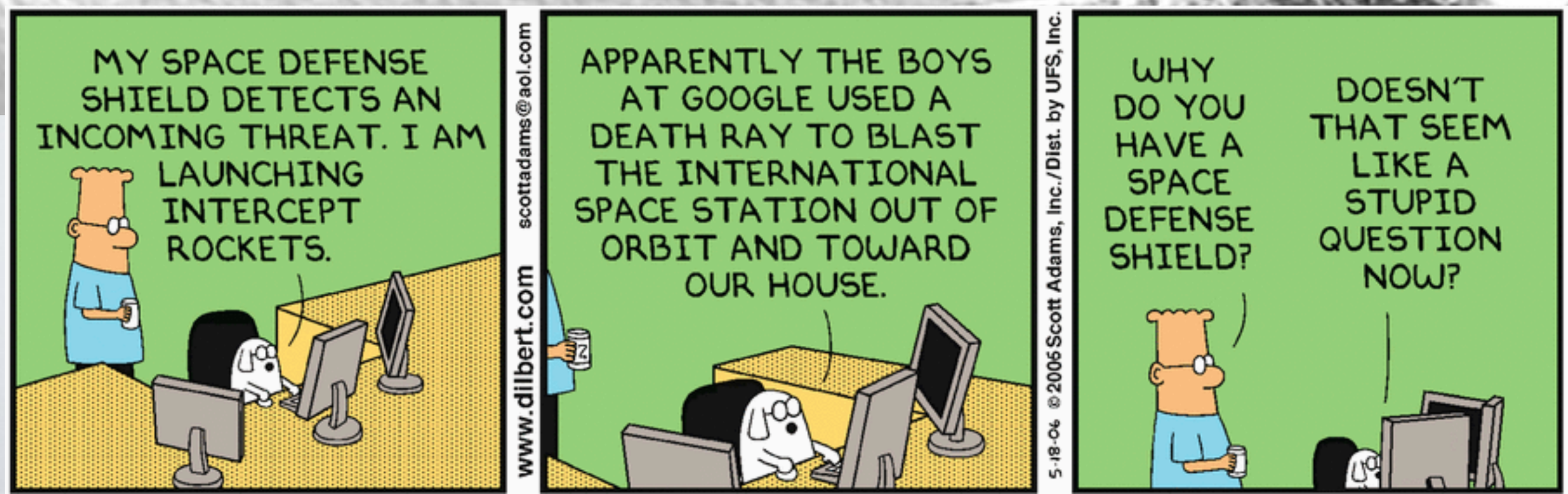
```java
public void printVocabularyStatistics() throws IOException {

    System.out.printf("%n------------- PRINTING VOCABULARY STATISTICS -------------%n");

    // Open the directory in Lucene
    final Directory dir = FSDirectory.open(indexPath);

    // Open the index - we need to get a LeafReader to be able to directly access terms
    final LeafReader index = DirectoryReader.open(dir).leaves().get(0).reader();

    // Total number of documents in the collection
    System.out.printf("+ Total number of documents: %d%n", index.numDocs());

    // Get the vocabulary of the index.
    final Terms voc = index.terms(BODY);

    // Total number of unique terms in the collection
    System.out.printf("+ Total number of unique terms: %d%n", voc.size());

    // Total number of terms in the collection
    System.out.printf("+ Total number of terms: %d%n", voc.getSumTotalTermFreq());

    // Get an iterator over the vector of terms in the vocabulary
    final TermsEnum termsEnum = voc.iterator();

    // Iterate until there are terms
    System.out.printf("+ Vocabulary:%n");
    System.out.printf("  - %-20s%-5s%-5s%n", "TERM", "DF", "FREQ");
    for (BytesRef term = termsEnum.next(); term != null; term = termsEnum.next()) {

        // Get the text string of the term
        String termstr = term.utf8ToString();

        // Get the document frequency (DF) of the term
        int df = termsEnum.docFreq();

        // Get the total frequency of the term
        long freq = termsEnum.totalTermFreq();
        System.out.printf("  - %-20s%-5d%-5d%n", termstr, df, freq);
    }
```

We need a `LeafReader` to be able to directly read a vector of all the terms from the index

Print overall statistics about the vocabulary

Iterate over terms in the index and retrieve their statistics to print them

Questions?


MY SPACE DEFENSE SHIELD DETECTS AN INCOMING THREAT. I AM LAUNCHING INTERCEPT ROCKETS.
www.dilbert.com
scottadams@aol.com
APPARENTLY THE BOYS AT GOOGLE USED A DEATH RAY TO BLAST THE INTERNATIONAL SPACE STATION OUT OF ORBIT AND TOWARD OUR HOUSE.
5-18-06 ©2006 Scott Adams, Inc./Dist. by UFS, Inc.
WHY DO YOU HAVE A SPACE DEFENSE SHIELD?
DOESN'T THAT SEEM LIKE A STUPID QUESTION NOW?