

liv

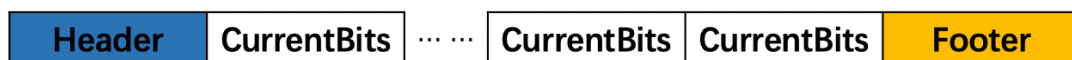
索引文件.liv只有在一个segment中包含被删除的文档时才会生成，它记录了当前段中没有被删除的文档号。这里不会讨论一个segment是如何获得被删除的文档号，在后面的文章中，介绍IndexWriter.flush()时会详细介绍，本篇文章只介绍那些被删除的文档号生成的索引文件的数据结构。

预备知识

介绍.liv文件的数据结构前，大家必须得了解Lucene的一个工具类FixedBitSet，这个类在源码中有大量的应用，是必须熟悉的一个工具。

数据结构

图1：



CurrentBits

CurrentBits占固定8个字节，即写入的是一个long类型的值，每一个CurrentBits分别表示了FixedBitSet对象中的bits[]数组的元素。

例子

图2:

```

41 Document doc ;
42 // 0
43 doc = new Document();
44 doc.add(new TextField( name: "content", value: "h", Field.Store.YES));
45 indexWriter.addDocument(doc);
46 // 1
47 doc = new Document();
48 doc.add(new TextField( name: "content", value: "b", Field.Store.YES));
49 indexWriter.addDocument(doc);
50 // 2
51 doc = new Document();
52 doc.add(new TextField( name: "content", value: "a c", Field.Store.YES));
53 indexWriter.addDocument(doc);
54 // 3
55 doc = new Document();
56 doc.add(new TextField( name: "content", value: "a c e", Field.Store.YES));
57 indexWriter.addDocument(doc);
58 // 4
59 doc = new Document();
60 doc.add(new TextField( name: "content", value: "h", Field.Store.YES));
61 indexWriter.addDocument(doc);
62 // 5
63 doc = new Document();
64 doc.add(new TextField( name: "content", value: "i", Field.Store.YES));
65 indexWriter.addDocument(doc);
66 // 6
67 doc = new Document();
68 doc.add(new TextField( name: "content", value: "c a e", Field.Store.YES));
69 indexWriter.addDocument(doc);
70 // 7
71 doc = new Document();
72 doc.add(new TextField( name: "content", value: "f", Field.Store.YES));
73 indexWriter.addDocument(doc);
74 // 8
75 doc = new Document();
76 doc.add(new TextField( name: "content", value: "b c d e c e", Field.Store.YES));
77 indexWriter.addDocument(doc);
78 // 9
79 doc = new Document();
80 doc.add(new TextField( name: "content", value: "a c e a b c", Field.Store.YES));
81 indexWriter.addDocument(doc);
82 // 删除文档
83 indexWriter.deleteDocuments(new Term( fld: "content", text: "h"));
84 indexWriter.deleteDocuments(new Term( fld: "content", text: "f"));
85 indexWriter.commit();
86

```

上图中添加了10篇文档，对应文档号0~9，然后在第84、85行执行了删除操作，即满足域名为"content"，域值为"h"或者"f"的文档都会被删除，即文档号0、4、7会被删除。在删除操作以后，在查询阶段实际可以获得的文档号只有1、2、3、5、6、8、9。由于一共只有10篇文档，所以只要一个long类型的值就可以表示这些文档号，即FixedBitSet对象中的long bit[]数组只有一个元素，数组下标代表了文档号。图3:

bit[]数组



结语

.liv索引文件非常的简单，只要熟悉FixedBitSet的用法，相信其数据结构也一目了然。

[点击下载Markdown文件](#)