

索引文件.fnm用来描述域信息 (FieldInfo)

## 例子

为了便于介绍.fnm中的各个字段，给出下面的例子

图1:

```
FieldType type = new FieldType();
type.setStored(true);
type.setStoreTermVectors(true);
type.setStoreTermVectorPositions(true);
type.setStoreTermVectorPayloads(true);
type.setStoreTermVectorOffsets(true);
type.setTokenized(true);
type.setIndexOptions(IndexOptions.DOCS_AND_FREQS_AND_POSITIONS_AND_OFFSETS);

// 文档0
doc = new Document();
doc.add(new SortedDocValuesField( name: "author", new BytesRef( text: "Chris")));
doc.add(new Field( name: "content", value: "a", type));
doc.add(new Field( name: "content", value: "e", type));
doc.add(new IntPoint( name: "abc", ...point: 3, 5, 9));
indexWriter.addDocument(doc);
```

## fnm文件的数据结构

图2:

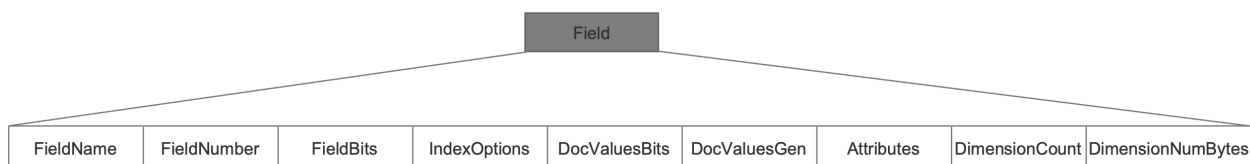


### FieldsCount

FieldsCount描述的是.fnm中域的种类。

### Field

图3:



## FieldName

该字段描述的是域名，例如图1中的"author"、"content"、"abc"都是FieldName。

## FieldNumber

域的编号，根据处理域的先后顺序，每个域都会获得一个从0开始递增的域的编号。

## FieldBits

该字段是一个组合值，它用来描述当前域是否有以下的属性：

- 是否存储词向量(termVector): 0x1, 词向量的介绍可以看这里[索引文件之tvx&&tvd](#)
- 是否忽略域的norm值: 0x2, 用于域的打分的norm值的介绍可以看这里[索引文件之nvd&&nvm](#)
- 是否带有payload: 0x4, payload的介绍可以看这里[索引文件之pos&&pay](#)
- 该域是否为软删除域(soft delete field): 0x8, 这个概念在后面的文档中会介绍

在图1中，域"content"的FieldBits的值为 (0x1 | 0x2 | 0x4) = 0x0111。

## IndexOptions

该字段描述了当前域的索引选项(IndexOptions), IndexOptions有以下值，每个选项的含义在[两阶段生成索引文件之第一阶段](#)已介绍，不赘述：

- 0: NONE
- 1: DOCS
- 2: DOCS\_AND\_FREQS
- 3: DOCS\_AND\_FREQS\_AND\_POSITIONS
- 4: DOCS\_AND\_FREQS\_AND\_POSITIONS\_AND\_OFFSETS

## DocValuesBits

该字段占用一个字节，其中高4个bit用来描述是否记录norm，低4个bit用来描述DocValues类型，DocValues的类型包括以下类型，在[DocValues](#)对每一种都已介绍，不赘述：

- 0: NONE
- 1: NUMERIC
- 2: BINARY
- 3: SORTED
- 4: SORTED\_SET
- 5: SORTED\_NUMERIC

# DocValuesGen

该字段描述了DocValues类型的域的更新状态，比如我们调用IndexWriter.updateDocValues(...)方法后，那么DocValuesGen的值会变更，这里不展开介绍，在介绍IndexWriter时会详细介绍。

# Attributes

该字段描述了存储当前域的索引文件的格式(format)，比如说当前是一个DocValues的域，那么Attributes的字段会有下面的值：

- PerFieldDocValuesFormat.format: Lucene70

表示使用[Lucene70](#)这种格式来生成[索引文件.dvd、.dvm](#)。

# DimensionCount

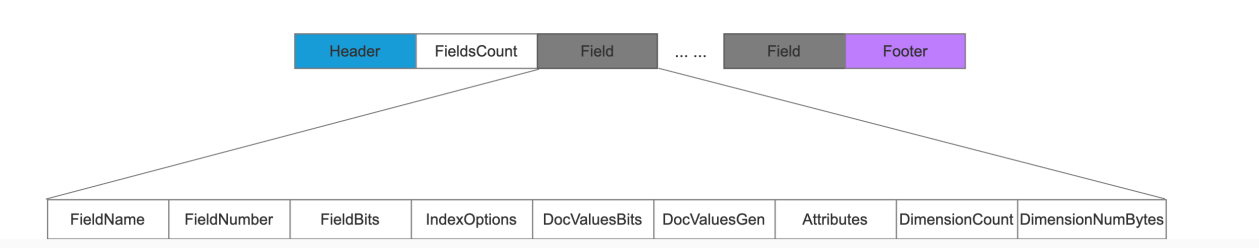
该字段描述的是如果域为点数据类型，那么DimensionCount的值为点数据的维度，点数据以及维度的概念在[Bkd-Tree](#)以及[索引文件之dim&&dii](#)介绍不赘述，在图1中，IntPoint域即为点数据域，DimensionCount的值为3，因为有3，5，9共三个值。

# DimensionNumBytes

该字段描述的是每一个维度占用的字节个数（数值类型被编码为多个字节），同样已经在前面的文章中介绍了。

# fnm文件的总数据结构

图4：



[点击下载](#)Markdown文档