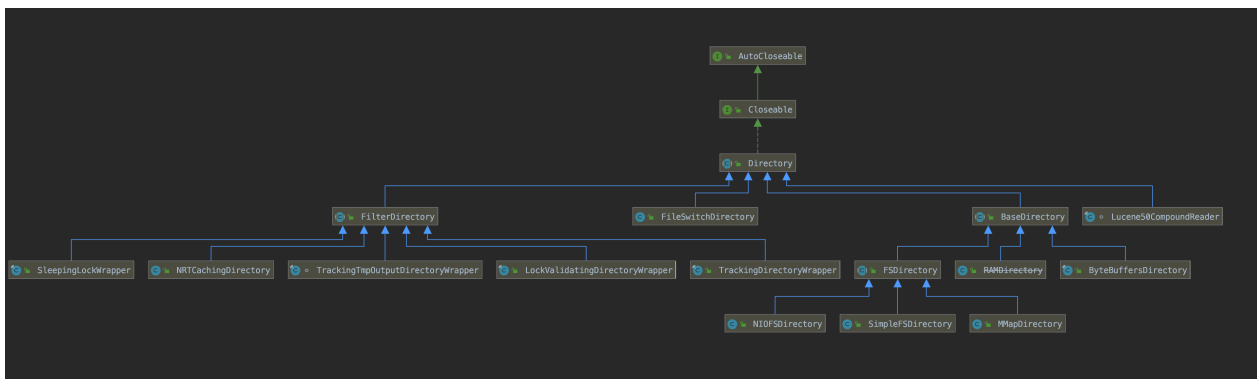


Directory_(下)

在[Directory_\(上\)](#)中，介绍了BaseDirectory类，它作为Directory的子类，该类及其子类实现了维护索引文件的所有操作，即创建、打开、删除、读取、重命名、同步(持久化索引文件至磁盘)、校验和(checksum computing)等等，而Directory的其他子类，不具备上述的维护索引文件的操作，而是封装了上述Directory类，提供更多高级功能。

图1：

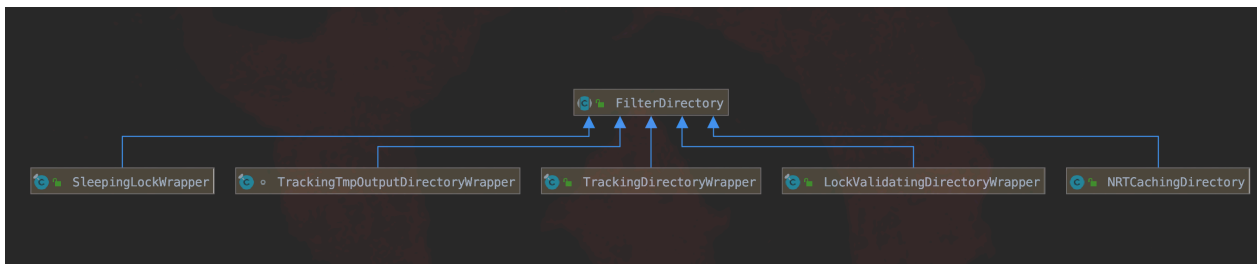


强调是，图1中只列出了Lucene7.5.0的core模块中的类图。

FilterDirectory

FilterDirectory类作为一个抽象类，它的子类对封装的Directory类增加了不同的限制(limitation)来实现高级功能。

图2：



SleepingLockWrapper

在文章[索引文件锁LockFactory](#)中，索引目录同一时间只允许一个IndexWriter对象进行操作，此时另一个IndexWriter对象(不同的引用)操作该目录时会抛出LockObtainFailedException异常：

图3：

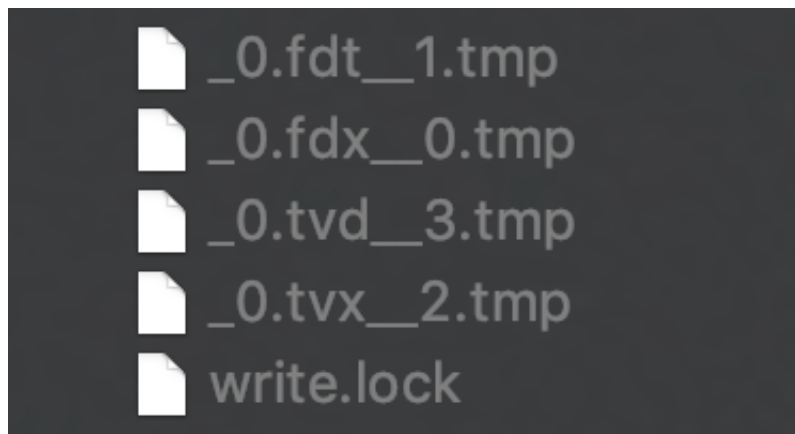
```
Exception in thread "main" org.apache.lucene.store.LockObtainFailedException:
```

在使用了SleepingLockWrapper后，会捕获LockObtainFailedException异常，同时等待1秒（默认为1秒）后重试，如果在重试次数期间仍无法获得索引文件锁，那么抛出LockObtainFailedException异常。

TrackingTmpOutputDirectoryWrapper

该类用来记录新创建临时索引文件，即带有.tmp后缀的文件。在[两阶段生成索引文件之第一阶段](#)中可以知道，IndexWriter在调用addDocument()的方法时，flush()或者commit()前，就会生成.fdx、.fdt以及.tvd、.tvx索引文件，而如果IndexWriter配置IndexSort，那么在上述期间内就只会生成临时的索引文件，TrackingTmpOutputDirectoryWrapper会记录这些临时索引文件，在后面介绍IndexWriter时会展开介绍：

图4：



TrackingDirectoryWrapper

该类用来记录新生成的索引文件名，不会记录从已有的索引目录中读取的索引文件名，比如在初始化Directory对象阶段会先读取索引目录中的索引文件。

LockValidatingDirectoryWrapper

该类使得在执行 **创建**、**删除**、**重命名**、**同步** (持久化索引文件至磁盘)的操作前都会先检查索引文件锁的状态是否有效的，比如说如果用户手动的把write.lock文件删除，那么会导致索引文件锁失效。

IndexWriter中使用了该类来维护索引文件。

NRTCachingDirectory

该类维护了一个RAMDirectory，并封装了另一个Directory类，使用该类需要定义两个重要参数：

- maxMergeSizeBytes：允许的段合并生成的索引文件大小最大值
- maxCachedBytes：RAMDirectory允许存储的索引文件大小总和最大值

NRTCachingDirectory的处理逻辑就是根据下面的条件来选择使用RAMDirectory或者使用封装的Directory来存储索引条件

```
boolean doCache = (bytes <= maxMergeSizeBytes) && (bytes +
cache.ramBytesUsed()) <= maxCachedBytes
```

其中bytes为索引文件的大小，cache.ramBytesUsed()为RAMDirectory已经存储的所有索引文件大小总和，当doCache为真，则继续使用RAMDirectory存储该索引文件，否则使用封装的Directory。

FileSwitchDirectory

在前面的介绍中我们知道，索引文件都存放同一个目录中，使用一个Directory对象来维护，而FileSwitchDirectory则使用两个Directory对象，即primaryDir跟secondaryDir，用户可以将索引文件分别写到primaryDir或者secondaryDir，使用primaryExtensions的Set对象来指定哪些后缀的索引文件使用primaryDir维护，否则使用secondaryDir维护，另外primaryDir或者secondaryDir可以使用同一个目录。

图5:

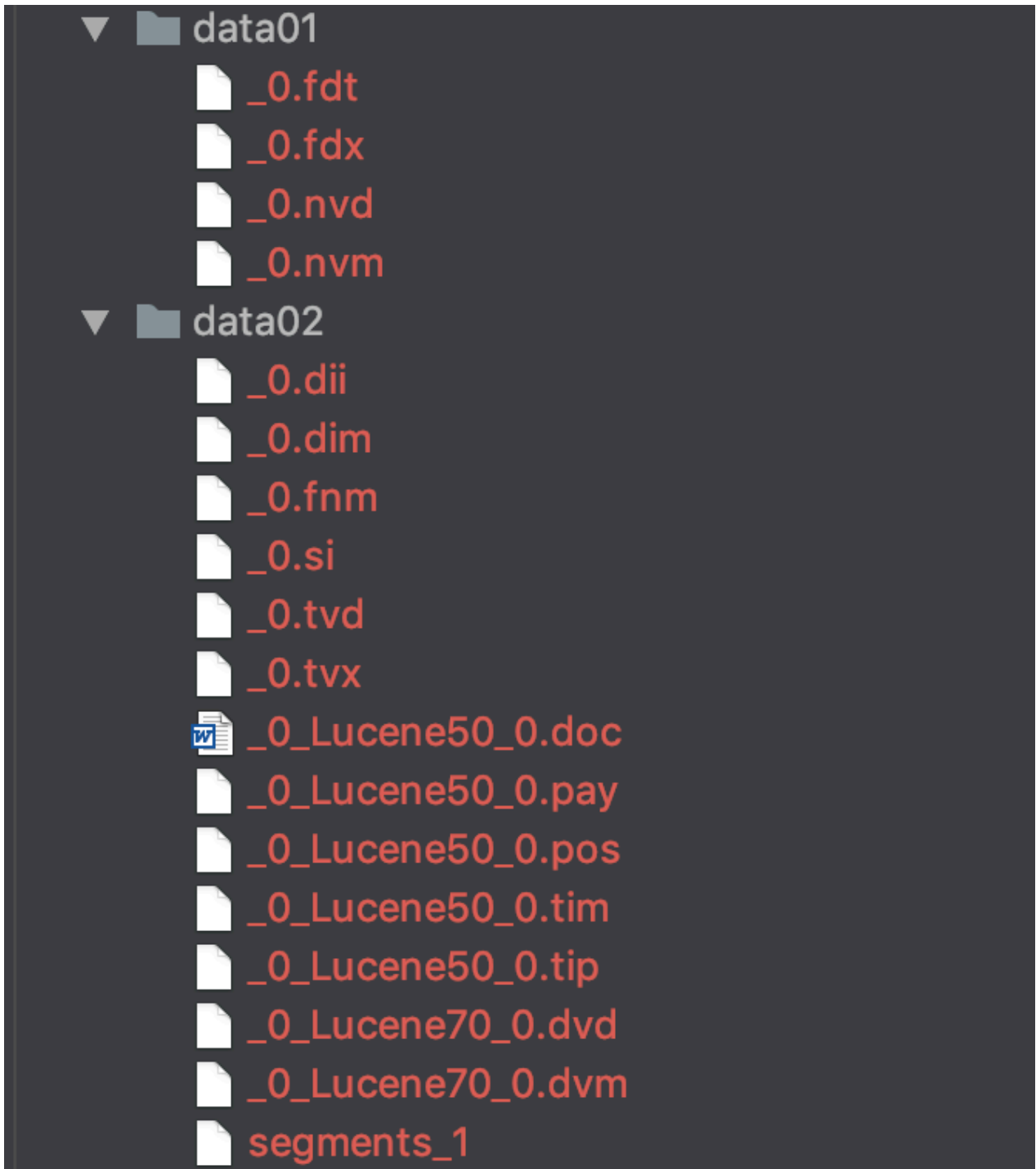
```
Directory primaryDir = FSDirectory.open(Paths.get( first: "./data01"));
Directory secondaryDir = FSDirectory.open(Paths.get( first: "./data02"));

Set<String> primaryExtensions = new HashSet<>();
primaryExtensions.add("fdx");
primaryExtensions.add("fdt");
primaryExtensions.add("nvd");
primaryExtensions.add("nvm");

Directory directory = new FileSwitchDirectory(primaryExtensions, primaryDir, secondaryDir, doClose: true);
```

图5中，primaryExtensions对象指定后缀为fdx、fdt、nvd、nvm的索引文件由primaryDir维护，即存放到 data01目录中，其他的索引文件存放到data02中，由secondaryDir维护，执行一次添加文档操作后，索引目录如下图：

图6:



Lucene50CompoundReader

该类仅用来读取复合文件(Compound File)，所以它仅支持 `打开`、`读取`。比如当我们在初始化 `IndexWriter` 时，需要读取旧的索引文件，如果该索引文件使用了复合文件，那么就会调用 `Lucene50CompoundReader` 类中的方法来读取旧索引信息。

[点击下载](#) Markdown 文件