

# Broker过期文件删除机制

由于RocketMQ操作CommitLog、ConsumeQueue文件是基于内存映射机制并在启动的时候回加载CommitLog、ConsumeQueue目录下的所有文件，为了避免内存与磁盘的浪费，不可能将消息永久存储在消息服务器上，所以要引入一种机制来删除已过期的文件。RocketMQ顺序写CommitLog、ConsumeQueue文件，所有写操作全部落在最后一个CommitLog或者ConsumerQueue文件上，之前的文件在下一个文件创建后将不会再被更新。

RocketMQ清除过期文件的方法时：如果当前文件在在一定时间间隔内没有再次被消费，则认为是过期文件，可以被删除，RocketMQ不会关注这个文件上的消息是否全部被消费。默认每个文件的过期时间为72小时，通过在Broker配置文件中设置fileReservedTime来改变过期时间，单位为小时。

## 默认消息清理的时机

Broker每10秒检查一次文件，扫描是否存在CommitLog满足删除条件。

### 删除文件的条件

第一，默认凌晨4点，删除72小时内没有产生消费的CommitLog文件；

第二，磁盘使用空间超过85%；

第三，手动执行删除。

### 删除CommitLog的执行过程

```
public void run() {
    try {
        this.deleteExpiredFiles();

        this.redeleteHangedFile();
    } catch (Throwable e) {
        defaultMessageStore.log.warn(this.getServiceName() + " service has
exception. ", e);
    }
}
```

第一步：克隆全部的 CommitLog 文件。CommitLog 文件可能随时有数据写入，为了不影响正常写入，所以克隆一份来操作。

第二步：检查每一个CommitLog文件是否过期，如果已过期则立即进行删除。在删除前会做一系列检查：检查文件被引用的次数、清理映射的所有内存数据对象、释放内存。清理完成后，删除物理文件。

第三步：上一步的删除有可能失败，比如有线程引用该过期文件，内存映射清理失败等，都可能导致删除失败。如果文件已经关闭，删除前检查没有通过，则可以通过第二次删除来处理，如果删除失败，间隔120秒后再次尝试执行重试删除直到文件删除。

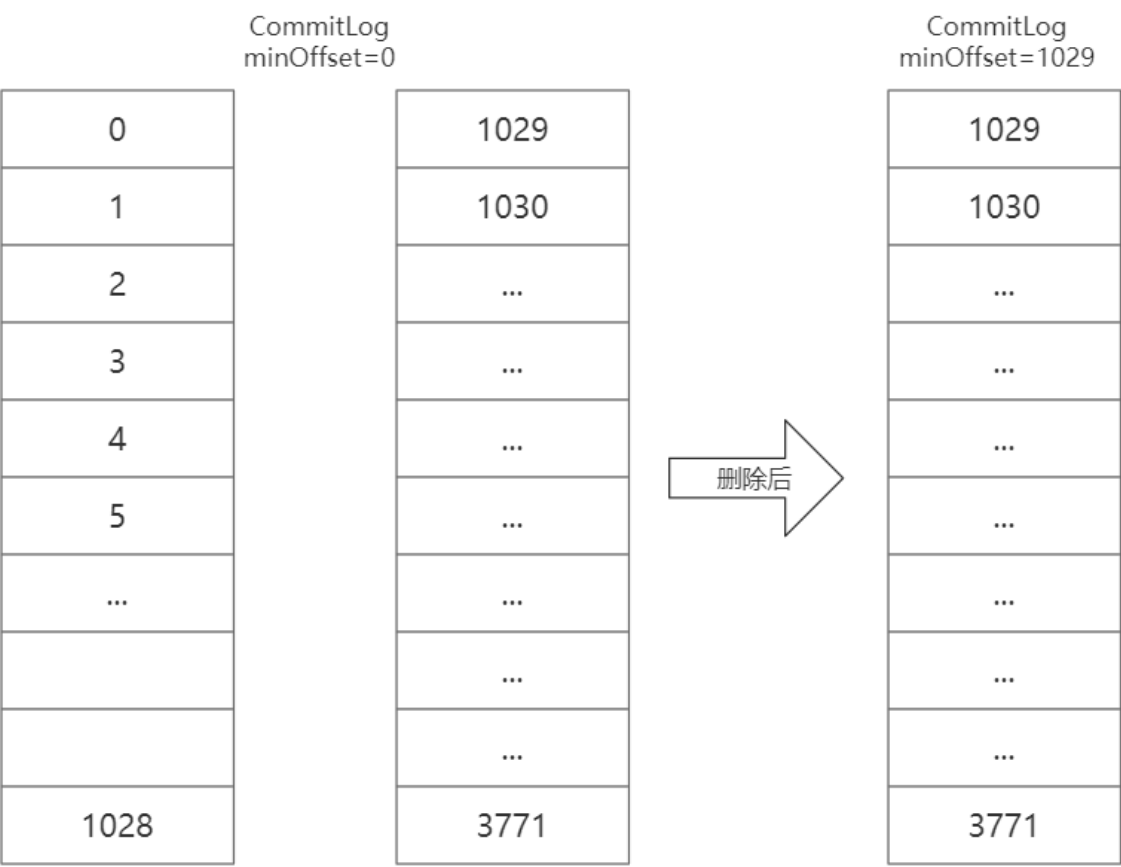
### ConsumeQueue、IndexFile文件的删除过程

Consume Queue和Index File都是索引文件，在CommitLog文件被删除后，对应的索引文件其实没有存在的意义，并且占用磁盘空间，所以这些文件应该被删除。

RocketMQ的删除策略是定时检查，满足删除条件时会删除过期或者无意义的文件。

执行过程如下

- CommitLog.getMinOffset获取当前日志最小偏移量，当前提交日志的最小偏移量对比上一次查询的值发生了变化，也就是说必然是最少一个提交日志文件被删除，那么相应的在消费队列中的过期数据也可以被删除，就执行后面的流程。反之，则意味着不需要执行任何操作，结束方法即可。



- 遍历其中每一个ConsumeQueue对象，删除过期的消费队列文件以及更新消费队列的最小偏移量；
- 如果有删除了ConsumeQueue文件，休眠一小段时间，继续对下一个ConsumeQueue检查执行删除操作。

IndexFile索引文件与ConsumeQueue删除逻辑相同，不做赘述。