ImportNew

Java NIO系列教程（5）：通道之间的数据传输

2016/04/01 | 分类： 教程 | 3 条评论 | 标签： Java NIO, 通道

 译文出处： 郭蕾 原文出处：Jakob Jenkov

在Java NIO中，如果两个通道中有一个是FileChannel，那你可以直接将数据从一个channel（译者注：channel中文常译作通道）传输到另外一个channel。

transferFrom()

FileChannel的transferFrom()方法可以将数据从源通道传输到FileChannel中（译者注：这个方法在JDK文档中的解释为将字节从给定的可读取字节通道传输到此通道的文件中）。下面是一个简单的例子：

```
RandomAccessFile fromFile = new RandomAccessFile("fromFile.txt", "rw");
FileChannel      fromChannel = fromFile.getChannel();

RandomAccessFile toFile = new RandomAccessFile("toFile.txt", "rw");
FileChannel      toChannel = toFile.getChannel();

long position = 0;
long count = fromChannel.size();

toChannel.transferFrom(position, count, fromChannel);
```

方法的输入参数position表示从position处开始向目标文件写入数据，count表示最多传输的字节数。如果源通道的剩余空间小于 count 个字节，则所传输的字节数要小于请求的字节数。
此外要注意，在SoketChannel的实现中，SocketChannel只会传输此刻准备好的数据（可能不足count字节）。因此，SocketChannel可能不会将请求的所有数据(count个字节)全部传输到FileChannel中。

transferTo()

transferTo()方法将数据从FileChannel传输到其他的channel中。下面是一个简单的例子：

```
RandomAccessFile fromFile = new RandomAccessFile("fromFile.txt", "rw");
FileChannel      fromChannel = fromFile.getChannel();

```

```
4  RandomAccessFile toFile = new RandomAccessFile("toFile.txt", "rw");
5  FileChannel      toChannel = toFile.getChannel();
6
7  long position = 0;
8  long count = fromChannel.size();
9
10 fromChannel.transferTo(position, count, toChannel);
```

是不是发现这个例子和前面那个例子特别相似？除了调用方法的FileChannel对象不一样外，其他的都一样。
上面所说的关于SocketChannel的问题在transferTo()方法中同样存在。SocketChannel会一直传输数据直到目标buffer被填满。

本系列：

- Java NIO系列教程（1）：Java NIO 概述
- Java NIO系列教程（2）：Channel
- Java NIO系列教程（3）：Buffer
- Java NIO系列教程（4）：Scatter/Gather
- Java NIO系列教程（5）：通道之间的数据传输

相关文章

- 攻破JAVA NIO技术壁垒
- Java NIO系列教程（12）：Java NIO与IO
- Java NIO系列教程（11）：Pipe
- Java NIO系列教程（10）：Java NIO DatagramChannel
- Java NIO系列教程（9）：ServerSocketChannel
- Java NIO系列教程（8）：SocketChannel
- Java NIO系列教程（7）：FileChannel
- Java NIO系列教程（6）：Selector
- Java NIO系列教程（4）：Scatter/Gather
- Java NIO系列教程（3）：Buffer

发表评论

Comment form

Name*

姓名

邮箱*

请填写邮箱

网站 (请以 http://开头)

请填写网站地址

评论内容*

请填写评论内容

(*) 表示必填项

提交评论

3 条评论

1. *Alifather* 说道：
   2016/04/12 下午 3:42

   transferFrom写错了
   transferFrom(ReadableByteChannel src, long position, long count)

   0 0

   回复
2. *加速奔跑的蜗牛* 说道：
   2017/08/08 上午 10:19

   fromChannel.transferTo(position, count, toChannel);这方法写错了吧，麻烦楼主看下api

   0 0

   回复
   - *breeze* 说道：
     2017/08/31 上午 9:51

     没有写错啊

     0 0

     回复

« Java NIO系列教程（4）：Scatter/Gather
Java NIO系列教程（6）：Selector »

Search for:

Search

Search

- 本周热门文章
- 本月热门
- 热门标签

0 记一次集群内无可用 http 服务问题...

1 Java 技术之垃圾回收机制

2 公司编程竞赛之最长路径问题

3 Java 中的十个"单行代码编程"（O...

4 Java 中 9 个处理 Exception ...

5 HttpClient 以及 Json 传递的...

6 浅析 Spring 中的事件驱动机制

7 浅析分布式下的事件驱动机制（ PubS...

8 探索各种随机函数（Java 环境...

9 Java 守护线程概述

0 Java ArrayList 踩坑记录

1 Spring4 + Spring MVC + MyB...

2 在 Java 中提升函数以更好地“函...

3 Dagger2 神器入门（二）

4 关于 Java 中的 double check lo...

5 Dagger2 神器入门（一）

6 Java 正则表达式 StackOverflowEr...

7 Java synchronized 中的 while 和 no...

8 编辑从字节码和 JVM 的角度解析 Jav...

9 深入了解 Java 之虚拟机内存

android23days Android开发 AOP ArrayList ConcurrentHashMap Eclipse GC Guava Hadoop HashMap HashSet HBase Hibernate IntelliJ io Java java8 java 8 Java9 Java NIO Java乱码 Java编程入门 JDBC JDK JMX JPA Jsoup JUnit JVM Lambda log4j MapReduce maven Mybatis Netty nio oracle ORM RabbitMQ redis RESTful Scala Servlet Socket solr Spring Spring4 spring boot springboot SpringMVC Spring MVC Spring Security String synchronized ThreadLocal Tomcat volatile Web Service Zookeeper 事务 内存管理 分布式 动态代理 参数太多怎么办 反射 垃圾回收 基础技术 多线程 字符串 字节码 安全 工具 并发 并发编程 序列化 异常 异常处理 性能 性能优化 性能调优 教程 数据结构 日志 架构 泛型 注解 测试 游戏 源码分析 算法 线程 线程池 缓存 自动化测试 虚拟机 设计模式 资讯 集合 面试 面试题

最新评论

- Re: 攻破JAVA NIO技术壁垒
  Hi，请到伯乐在线的小组发帖提问，支持微信登录。链接是： http://group.jobbole.... 唐尤华
- Re: 攻破JAVA NIO技术壁垒
  TCP服务端的NIO写法 服务端怎么发送呢。原谅小白 菜鸟
- Re: 关于 Java 中的 double check ...
  volatile 可以避免指令重排啊。所以double check还是可以用的。 hipilee
- Re: Spring4 + Spring MVC + M...
  Hi，请到伯乐在线的小组发帖提问，支持微信登录。链接是： http://group.jobbole.... 唐尤华
- Re: Spring4 + Spring MVC + M...
  我的一直不太明白，spring的bean容器和springmvc的bean容器之间的关系。 hw_绝影
- Re: Spirng+SpringMVC+Maven+Myba...
  很好，按照步骤，已经成功。 莫凡
- Re: Spring中@Transactional事务...
  声明式事务可以用aop来实现,分别是jdk代理和cglib代理,基于接口和普通类.在同一个类中一个方... chengjiliang

- Re: 关于 Java 中的 double check ...
  在JDK1.5之后，用volatile关键字修饰_INSTANCE属性 就能避免因指令重排导致的对象...
  Byron

关于ImportNew

ImportNew 专注于 Java 技术分享。于2012年11月11日 11:11正式上线。是的，这是一个很特别的时刻 :)

ImportNew 由两个 Java 关键字 import 和 new 组成，意指：Java 开发者学习新知识的网站。import 可认为是学习和吸收， new 则可认为是新知识、新技术圈子和新朋友……

联系我们

Email：ImportNew.com@gmail.com
新浪微博：@ImportNew
推荐微信号

反馈建议：ImportNew.com@gmail.com
广告与商务合作QQ：2302462408

推荐关注

小组 – 好的话题、有启发的回复、值得信赖的圈子
头条 – 写了文章？看干货？去头条！
相亲 – 为IT单身男女服务的征婚传播平台
资源 – 优秀的工具资源导航
翻译 – 活跃 & 专业的翻译小组
博客 – 国内外的精选博客文章
设计 – UI,网页，交互和用户体验
前端 – JavaScript, HTML5, CSS
安卓 – 专注Android技术分享
iOS – 专注iOS技术分享
Java – 专注Java技术分享
Python – 专注Python技术分享