

ImportNew

- [首页](#)
- [所有文章](#)
- [资讯](#)
- [Web](#)
- [架构](#)
- [基础技术](#)
- [书籍](#)
- [教程](#)
- [Java小组](#)
- [工具资源](#)

- 导航条 - ▼

Java NIO系列教程（10）：Java NIO DatagramChannel

2016/04/06 | 分类：[教程](#) | [0 条评论](#) | 标签：[DatagramChannel](#), [Java NIO](#)

分享到：

¹² 译文出处：[郑玉婷](#) 原文出处：[Jakob Jenkov](#)



Java NIO中的DatagramChannel是一个能收发UDP包的通道。因为UDP是无连接的网络协议，所以不能像其它通道那样读取和写入。它发送和接收的是数据包。

打开 DatagramChannel

下面是 DatagramChannel 的打开方式：

```
1 | DatagramChannel channel = DatagramChannel.open();  
2 | channel.socket().bind(new InetSocketAddress(9999));
```

这个例子打开的 DatagramChannel可以在UDP端口9999上接收数据包。

接收数据

通过receive()方法从DatagramChannel接收数据，如：

```
1 | ByteBuffer buf = ByteBuffer.allocate(48);  
2 | buf.clear();  
3 | channel.receive(buf);
```

receive()方法会将接收到的数据包内容复制到指定的Buffer。如果Buffer容不下收到的数据，多出的数据将被丢弃。

发送数据

通过send()方法从DatagramChannel发送数据，如：

```
1 String newData = "New String to write to file..." + System.currentTimeMillis();
2
3 ByteBuffer buf = ByteBuffer.allocate(48);
4 buf.clear();
5 buf.put(newData.getBytes());
6 buf.flip();
7
8 int bytesSent = channel.send(buf, new InetSocketAddress("jenkov.com", 80));
```

这个例子发送一串字符到“jenkov.com”服务器的UDP端口80。因为服务端并没有监控这个端口，所以什么也不会发生。也不会通知你发出的数据包是否已收到，因为UDP在数据传送方面没有任何保证。

连接到特定的地址

可以将DatagramChannel“连接”到网络中的特定地址的。由于UDP是无连接的，连接到特定地址并不会像TCP通道那样创建一个真正的连接。而是锁住DatagramChannel，让其只能从特定地址收发数据。

这里有个例子：

```
1 channel.connect(new InetSocketAddress("jenkov.com", 80));
```

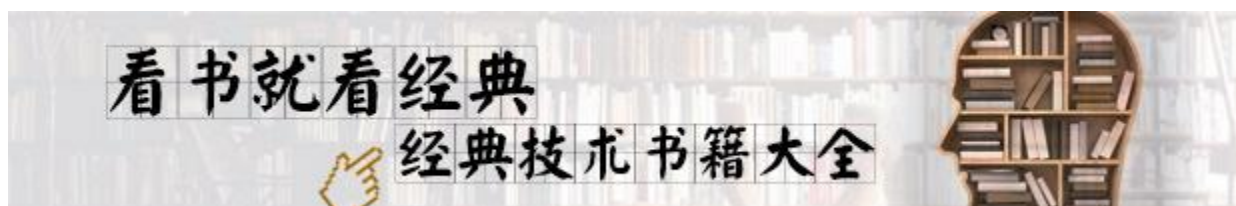
当连接后，也可以使用read()和write()方法，就像在用传统的通道一样。只是在数据传送方面没有任何保证。这里有几个例子：

```
1 int bytesRead = channel.read(buf);
2 int bytesWritten = channel.write(buf);
```

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](#)
- [Java NIO系列教程（6）：Selector](#)
- [Java NIO系列教程（7）：FileChannel](#)
- [Java NIO系列教程（8）：SocketChannel](#)
- [Java NIO系列教程（9）：ServerSocketChannel](#)
- [Java NIO系列教程（10）：Java NIO DatagramChannel](#)



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（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（5）：通道之间的数据传输](#)
- [Java NIO系列教程（4）：Scatter/Gather](#)
- [Java NIO系列教程（3）：Buffer](#)

发表评论

Comment form

Name*

邮箱*

网站 (请以 http://开头)

评论内容*



(*) 表示必填项

[提交评论](#)

还没有评论。

[« Java NIO系列教程（9）：ServerSocketChannel](#)

[Java NIO系列教程（11）：Pipe »](#)

Search for:



- [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](#)



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: [Spring+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

© 2017 ImportNew

