| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ReadableByteChannel.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/nio/channels/Pipe.SourceChannel.html)   [**NEXT CLASS**](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/ReadableByteChannel.html)    [**NO FRAMES**](http://docs.google.com/ReadableByteChannel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.nio.channels**

Interface ReadableByteChannel

**All Superinterfaces:** [Channel](http://docs.google.com/java/nio/channels/Channel.html), [Closeable](http://docs.google.com/java/io/Closeable.html) **All Known Subinterfaces:** [ByteChannel](http://docs.google.com/java/nio/channels/ByteChannel.html), [ScatteringByteChannel](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html) **All Known Implementing Classes:** [DatagramChannel](http://docs.google.com/java/nio/channels/DatagramChannel.html), [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html), [Pipe.SourceChannel](http://docs.google.com/java/nio/channels/Pipe.SourceChannel.html), [SocketChannel](http://docs.google.com/java/nio/channels/SocketChannel.html)

public interface **ReadableByteChannel**extends [Channel](http://docs.google.com/java/nio/channels/Channel.html)

A channel that can read bytes.

Only one read operation upon a readable channel may be in progress at any given time. If one thread initiates a read operation upon a channel then any other thread that attempts to initiate another read operation will block until the first operation is complete. Whether or not other kinds of I/O operations may proceed concurrently with a read operation depends upon the type of the channel.

**Since:** 1.4

| **Method Summary** | |
| --- | --- |
| int | [**read**](http://docs.google.com/java/nio/channels/ReadableByteChannel.html#read(java.nio.ByteBuffer))([ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) dst)            Reads a sequence of bytes from this channel into the given buffer. |

| **Methods inherited from interface java.nio.channels.**[**Channel**](http://docs.google.com/java/nio/channels/Channel.html) |
| --- |
| [close](http://docs.google.com/java/nio/channels/Channel.html#close()), [isOpen](http://docs.google.com/java/nio/channels/Channel.html#isOpen()) |

| **Method Detail** |
| --- |

### read

int **read**([ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) dst)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Reads a sequence of bytes from this channel into the given buffer.

An attempt is made to read up to *r* bytes from the channel, where *r* is the number of bytes remaining in the buffer, that is, dst.remaining(), at the moment this method is invoked.

Suppose that a byte sequence of length *n* is read, where 0 <= *n* <= *r*. This byte sequence will be transferred into the buffer so that the first byte in the sequence is at index *p* and the last byte is at index *p* + *n* - 1, where *p* is the buffer's position at the moment this method is invoked. Upon return the buffer's position will be equal to *p* + *n*; its limit will not have changed.

A read operation might not fill the buffer, and in fact it might not read any bytes at all. Whether or not it does so depends upon the nature and state of the channel. A socket channel in non-blocking mode, for example, cannot read any more bytes than are immediately available from the socket's input buffer; similarly, a file channel cannot read any more bytes than remain in the file. It is guaranteed, however, that if a channel is in blocking mode and there is at least one byte remaining in the buffer then this method will block until at least one byte is read.

This method may be invoked at any time. If another thread has already initiated a read operation upon this channel, however, then an invocation of this method will block until the first operation is complete.

**Parameters:**dst - The buffer into which bytes are to be transferred **Returns:**The number of bytes read, possibly zero, or -1 if the channel has reached end-of-stream **Throws:** [NonReadableChannelException](http://docs.google.com/java/nio/channels/NonReadableChannelException.html) - If this channel was not opened for reading [ClosedChannelException](http://docs.google.com/java/nio/channels/ClosedChannelException.html) - If this channel is closed [AsynchronousCloseException](http://docs.google.com/java/nio/channels/AsynchronousCloseException.html) - If another thread closes this channel while the read operation is in progress [ClosedByInterruptException](http://docs.google.com/java/nio/channels/ClosedByInterruptException.html) - If another thread interrupts the current thread while the read operation is in progress, thereby closing the channel and setting the current thread's interrupt status [IOException](http://docs.google.com/java/io/IOException.html) - If some other I/O error occurs

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ReadableByteChannel.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/nio/channels/Pipe.SourceChannel.html)   [**NEXT CLASS**](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/ReadableByteChannel.html)    [**NO FRAMES**](http://docs.google.com/ReadableByteChannel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).