| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/java/nio/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/nio/channels/spi/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package java.nio.channels

Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**ByteChannel**](http://docs.google.com/java/nio/channels/ByteChannel.html) | A channel that can read and write bytes. |
| [**Channel**](http://docs.google.com/java/nio/channels/Channel.html) | A nexus for I/O operations. |
| [**GatheringByteChannel**](http://docs.google.com/java/nio/channels/GatheringByteChannel.html) | A channel that can write bytes from a sequence of buffers. |
| [**InterruptibleChannel**](http://docs.google.com/java/nio/channels/InterruptibleChannel.html) | A channel that can be asynchronously closed and interrupted. |
| [**ReadableByteChannel**](http://docs.google.com/java/nio/channels/ReadableByteChannel.html) | A channel that can read bytes. |
| [**ScatteringByteChannel**](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html) | A channel that can read bytes into a sequence of buffers. |
| [**WritableByteChannel**](http://docs.google.com/java/nio/channels/WritableByteChannel.html) | A channel that can write bytes. |

| **Class Summary** | |
| --- | --- |
| [**Channels**](http://docs.google.com/java/nio/channels/Channels.html) | Utility methods for channels and streams. |
| [**DatagramChannel**](http://docs.google.com/java/nio/channels/DatagramChannel.html) | A selectable channel for datagram-oriented sockets. |
| [**FileChannel**](http://docs.google.com/java/nio/channels/FileChannel.html) | A channel for reading, writing, mapping, and manipulating a file. |
| [**FileChannel.MapMode**](http://docs.google.com/java/nio/channels/FileChannel.MapMode.html) | A typesafe enumeration for file-mapping modes. |
| [**FileLock**](http://docs.google.com/java/nio/channels/FileLock.html) | A token representing a lock on a region of a file. |
| [**Pipe**](http://docs.google.com/java/nio/channels/Pipe.html) | A pair of channels that implements a unidirectional pipe. |
| [**Pipe.SinkChannel**](http://docs.google.com/java/nio/channels/Pipe.SinkChannel.html) | A channel representing the writable end of a [Pipe](http://docs.google.com/java/nio/channels/Pipe.html). |
| [**Pipe.SourceChannel**](http://docs.google.com/java/nio/channels/Pipe.SourceChannel.html) | A channel representing the readable end of a [Pipe](http://docs.google.com/java/nio/channels/Pipe.html). |
| [**SelectableChannel**](http://docs.google.com/java/nio/channels/SelectableChannel.html) | A channel that can be multiplexed via a [Selector](http://docs.google.com/java/nio/channels/Selector.html). |
| [**SelectionKey**](http://docs.google.com/java/nio/channels/SelectionKey.html) | A token representing the registration of a [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) with a [Selector](http://docs.google.com/java/nio/channels/Selector.html). |
| [**Selector**](http://docs.google.com/java/nio/channels/Selector.html) | A multiplexor of [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) objects. |
| [**ServerSocketChannel**](http://docs.google.com/java/nio/channels/ServerSocketChannel.html) | A selectable channel for stream-oriented listening sockets. |
| [**SocketChannel**](http://docs.google.com/java/nio/channels/SocketChannel.html) | A selectable channel for stream-oriented connecting sockets. |

| **Exception Summary** | |
| --- | --- |
| [**AlreadyConnectedException**](http://docs.google.com/java/nio/channels/AlreadyConnectedException.html) | Unchecked exception thrown when an attempt is made to connect a [SocketChannel](http://docs.google.com/java/nio/channels/SocketChannel.html) that is already connected. |
| [**AsynchronousCloseException**](http://docs.google.com/java/nio/channels/AsynchronousCloseException.html) | Checked exception received by a thread when another thread closes the channel or the part of the channel upon which it is blocked in an I/O operation. |
| [**CancelledKeyException**](http://docs.google.com/java/nio/channels/CancelledKeyException.html) | Unchecked exception thrown when an attempt is made to use a selection key that is no longer valid. |
| [**ClosedByInterruptException**](http://docs.google.com/java/nio/channels/ClosedByInterruptException.html) | Checked exception received by a thread when another thread interrupts it while it is blocked in an I/O operation upon a channel. |
| [**ClosedChannelException**](http://docs.google.com/java/nio/channels/ClosedChannelException.html) | Checked exception thrown when an attempt is made to invoke or complete an I/O operation upon channel that is closed, or at least closed to that operation. |
| [**ClosedSelectorException**](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) | Unchecked exception thrown when an attempt is made to invoke an I/O operation upon a closed selector. |
| [**ConnectionPendingException**](http://docs.google.com/java/nio/channels/ConnectionPendingException.html) | Unchecked exception thrown when an attempt is made to connect a [SocketChannel](http://docs.google.com/java/nio/channels/SocketChannel.html) for which a non-blocking connection operation is already in progress. |
| [**FileLockInterruptionException**](http://docs.google.com/java/nio/channels/FileLockInterruptionException.html) | Checked exception received by a thread when another thread interrupts it while it is waiting to acquire a file lock. |
| [**IllegalBlockingModeException**](http://docs.google.com/java/nio/channels/IllegalBlockingModeException.html) | Unchecked exception thrown when a blocking-mode-specific operation is invoked upon a channel in the incorrect blocking mode. |
| [**IllegalSelectorException**](http://docs.google.com/java/nio/channels/IllegalSelectorException.html) | Unchecked exception thrown when an attempt is made to register a channel with a selector that was not created by the provider that created the channel. |
| [**NoConnectionPendingException**](http://docs.google.com/java/nio/channels/NoConnectionPendingException.html) | Unchecked exception thrown when the [finishConnect](http://docs.google.com/java/nio/channels/SocketChannel.html#finishConnect()) method of a [SocketChannel](http://docs.google.com/java/nio/channels/SocketChannel.html) is invoked without first successfully invoking its [connect](http://docs.google.com/java/nio/channels/SocketChannel.html#connect(java.net.SocketAddress)) method. |
| [**NonReadableChannelException**](http://docs.google.com/java/nio/channels/NonReadableChannelException.html) | Unchecked exception thrown when an attempt is made to read from a channel that was not originally opened for reading. |
| [**NonWritableChannelException**](http://docs.google.com/java/nio/channels/NonWritableChannelException.html) | Unchecked exception thrown when an attempt is made to write to a channel that was not originally opened for writing. |
| [**NotYetBoundException**](http://docs.google.com/java/nio/channels/NotYetBoundException.html) | Unchecked exception thrown when an attempt is made to invoke an I/O operation upon a server socket channel that is not yet bound. |
| [**NotYetConnectedException**](http://docs.google.com/java/nio/channels/NotYetConnectedException.html) | Unchecked exception thrown when an attempt is made to invoke an I/O operation upon a socket channel that is not yet connected. |
| [**OverlappingFileLockException**](http://docs.google.com/java/nio/channels/OverlappingFileLockException.html) | Unchecked exception thrown when an attempt is made to acquire a lock on a region of a file that overlaps a region already locked by the same Java virtual machine, or when another thread is already waiting to lock an overlapping region of the same file. |
| [**UnresolvedAddressException**](http://docs.google.com/java/nio/channels/UnresolvedAddressException.html) | Unchecked exception thrown when an attempt is made to invoke a network operation upon an unresolved socket address. |
| [**UnsupportedAddressTypeException**](http://docs.google.com/java/nio/channels/UnsupportedAddressTypeException.html) | Unchecked exception thrown when an attempt is made to bind or connect to a socket address of a type that is not supported. |

## Package java.nio.channels Description

Defines channels, which represent connections to entities that are capable of performing I/O operations, such as files and sockets; defines selectors, for multiplexed, non-blocking I/O operations.

| Channels | Description |
| --- | --- |
| [*Channel*](http://docs.google.com/java/nio/channels/Channel.html) | A nexus for I/O operations |
| [*ReadableByteChannel*](http://docs.google.com/java/nio/channels/ReadableByteChannel.html) | Can read into a buffer |
| [*ScatteringByteChannel*](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html) | Can read into a sequence of buffers |
| [*WritableByteChannel*](http://docs.google.com/java/nio/channels/WritableByteChannel.html) | Can write from a buffer |
| [*GatheringByteChannel*](http://docs.google.com/java/nio/channels/GatheringByteChannel.html) | Can write from a sequence of buffers |
| [*ByteChannel*](http://docs.google.com/java/nio/channels/ByteChannel.html) | Can read/write to/from a buffer |
| [Channels](http://docs.google.com/java/nio/channels/Channels.html) | Utility methods for channel/stream interoperation |

A *channel* represents an open connection to an entity such as a hardware device, a file, a network socket, or a program component that is capable of performing one or more distinct I/O operations, for example reading or writing. As specified in the [Channel](http://docs.google.com/java/nio/channels/Channel.html) interface, channels are either open or closed, and they are both *asynchronously closeable* and *interruptible*.

The [Channel](http://docs.google.com/java/nio/channels/Channel.html) interface is extended by several other interfaces, each of which specifies a new I/O operation.

The [ReadableByteChannel](http://docs.google.com/java/nio/channels/ReadableByteChannel.html) interface specifies a [read](http://docs.google.com/java/nio/channels/ReadableByteChannel.html#read(java.nio.ByteBuffer)) method that reads bytes from the channel into a buffer; similarly, the [WritableByteChannel](http://docs.google.com/java/nio/channels/WritableByteChannel.html) interface specifies a [write](http://docs.google.com/java/nio/channels/WritableByteChannel.html#write(java.nio.ByteBuffer)) method that writes bytes from a buffer to the channel. The [ByteChannel](http://docs.google.com/java/nio/channels/ByteChannel.html) interface unifies these two interfaces for the common case of channels that can both read and write bytes.

The [ScatteringByteChannel](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html) and [GatheringByteChannel](http://docs.google.com/java/nio/channels/GatheringByteChannel.html) interfaces extend the [ReadableByteChannel](http://docs.google.com/java/nio/channels/ReadableByteChannel.html) and [WritableByteChannel](http://docs.google.com/java/nio/channels/WritableByteChannel.html) interfaces, respectively, adding [read](http://docs.google.com/java/nio/channels/ScatteringByteChannel.html#read(java.nio.ByteBuffer%5B%5D,%20int,%20int)) and [write](http://docs.google.com/java/nio/channels/GatheringByteChannel.html#write(java.nio.ByteBuffer%5B%5D,%20int,%20int)) methods that take a sequence of buffers rather than a single buffer.

The [Channels](http://docs.google.com/java/nio/channels/Channels.html) utility class defines static methods that support the interoperation of the stream classes of the [java.io](http://docs.google.com/java/io/package-summary.html) package with the channel classes of this package. An appropriate channel can be constructed from an [InputStream](http://docs.google.com/java/io/InputStream.html) or an [OutputStream](http://docs.google.com/java/io/OutputStream.html), and conversely an [InputStream](http://docs.google.com/java/io/InputStream.html) or an [OutputStream](http://docs.google.com/java/io/OutputStream.html) can be constructed from a channel. A [Reader](http://docs.google.com/java/io/Reader.html) can be constructed that uses a given charset to decode bytes from a given readable byte channel, and conversely a [Writer](http://docs.google.com/java/io/Writer.html) can be constructed that uses a given charset to encode characters into bytes and write them to a given writable byte channel.

| File channels | Description |
| --- | --- |
| [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) | Reads, writes, maps, and manipulates files |
| [FileLock](http://docs.google.com/java/nio/channels/FileLock.html) | A lock on a (region of a) file |
| [MappedByteBuffer](http://docs.google.com/java/nio/MappedByteBuffer.html) | A direct byte buffer mapped to a region of a file |

The [FileChannel](http://docs.google.com/java/nio/channels/FileChannel.html) class supports the usual operations of reading bytes from, and writing bytes to, a channel connected to a file, as well as those of querying and modifying the current file position and truncating the file to a specific size. It defines methods for acquiring locks on the whole file or on a specific region of a file; these methods return instances of the [FileLock](http://docs.google.com/java/nio/channels/FileLock.html) class. Finally, it defines methods for forcing updates to the file to be written to the storage device that contains it, for efficiently transferring bytes between the file and other channels, and for mapping a region of the file directly into memory. This last operation creates an instance of the [MappedByteBuffer](http://docs.google.com/java/nio/MappedByteBuffer.html) class, which extends the [ByteBuffer](http://docs.google.com/java/nio/ByteBuffer.html) class with several file-related operations.

A getChannel method has been added to each of the [FileInputStream](http://docs.google.com/java/io/FileInputStream.html#getChannel()), [FileOutputStream](http://docs.google.com/java/io/FileOutputStream.html#getChannel()), and [RandomAccessFile](http://docs.google.com/java/io/RandomAccessFile.html#getChannel()) classes of the [java.io](http://docs.google.com/java/io/package-summary.html) package. Invoking this method upon an instance of one of these classes will return a file channel connected to the underlying file.

| Multiplexed, non-blocking I/O | Description |
| --- | --- |
| [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) | A channel that can be multiplexed |
| [DatagramChannel](http://docs.google.com/java/nio/channels/DatagramChannel.html) | A channel for a [java.net.DatagramSocket](http://docs.google.com/java/net/DatagramSocket.html) |
| [Pipe.SinkChannel](http://docs.google.com/java/nio/channels/Pipe.SinkChannel.html) | The write end of a pipe |
| [Pipe.SourceChannel](http://docs.google.com/java/nio/channels/Pipe.SourceChannel.html) | The read end of a pipe |
| [ServerSocketChannel](http://docs.google.com/java/nio/channels/ServerSocketChannel.html) | A channel for a [java.net.ServerSocket](http://docs.google.com/java/net/ServerSocket.html) |
| [SocketChannel](http://docs.google.com/java/nio/channels/SocketChannel.html) | A channel for a [java.net.Socket](http://docs.google.com/java/net/Socket.html) |
| [Selector](http://docs.google.com/java/nio/channels/Selector.html) | A multiplexor of selectable channels |
| [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) | A token representing the registration  of a channel with a selector |
| [Pipe](http://docs.google.com/java/nio/channels/Pipe.html) | Two channels that form a unidirectional pipe |

Multiplexed, non-blocking I/O, which is much more scalable than thread-oriented, blocking I/O, is provided by *selectors*, *selectable channels*, and *selection keys*.

A [*selector*](http://docs.google.com/Selector.html) is a multiplexor of [*selectable channels*](http://docs.google.com/SelectableChannel.html), which in turn are a special type of channel that can be put into [*non-blocking mode*](http://docs.google.com/SelectableChannel.html#bm). To perform multiplexed I/O operations, one or more selectable channels are first created, put into non-blocking mode, and [*registered*](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int,%20java.lang.Object)) with a selector. Registering a channel specifies the set of I/O operations that will be tested for readiness by the selector, and returns a [*selection key*](http://docs.google.com/SelectionKey.html) that represents the registration.

Once some channels have been registered with a selector, a [*selection operation*](http://docs.google.com/Selector.html#selop) can be performed in order to discover which channels, if any, have become ready to perform one or more of the operations in which interest was previously declared. If a channel is ready then the key returned when it was registered will be added to the selector's *selected-key set*. The key set, and the keys within it, can be examined in order to determine the operations for which each channel is ready. From each key one can retrieve the corresponding channel in order to perform whatever I/O operations are required.

That a selection key indicates that its channel is ready for some operation is a hint, but not a guarantee, that such an operation can be performed by a thread without causing the thread to block. It is imperative that code that performs multiplexed I/O be written so as to ignore these hints when they prove to be incorrect.

This package defines selectable-channel classes corresponding to the [DatagramSocket](http://docs.google.com/java/net/DatagramSocket.html), [ServerSocket](http://docs.google.com/java/net/ServerSocket.html), and [Socket](http://docs.google.com/java/net/Socket.html) classes defined in the [java.net](http://docs.google.com/java/net/package-summary.html) package. Minor changes to these classes have been made in order to support sockets that are associated with channels. This package also defines a simple class that implements unidirectional pipes. In all cases, a new selectable channel is created by invoking the static open method of the corresponding class. If a channel needs an associated socket then a socket will be created as a side effect of this operation.

The implementation of selectors, selectable channels, and selection keys can be replaced by "plugging in" an alternative definition or instance of the [SelectorProvider](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html) class defined in the [java.nio.channels.spi](http://docs.google.com/java/nio/channels/spi/package-summary.html) package. It is not expected that many developers will actually make use of this facility; it is provided primarily so that sophisticated users can take advantage of operating-system-specific I/O-multiplexing mechanisms when very high performance is required.

Much of the bookkeeping and synchronization required to implement the multiplexed-I/O abstractions is performed by the [AbstractInterruptibleChannel](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html), [AbstractSelectableChannel](http://docs.google.com/java/nio/channels/spi/AbstractSelectableChannel.html), [AbstractSelectionKey](http://docs.google.com/java/nio/channels/spi/AbstractSelectionKey.html), and [AbstractSelector](http://docs.google.com/java/nio/channels/spi/AbstractSelector.html) classes in the [java.nio.channels.spi](http://docs.google.com/java/nio/channels/spi/package-summary.html) package. When defining a custom selector provider, only the [AbstractSelector](http://docs.google.com/java/nio/channels/spi/AbstractSelector.html) and [AbstractSelectionKey](http://docs.google.com/java/nio/channels/spi/AbstractSelectionKey.html) classes should be subclassed directly; custom channel classes should extend the appropriate [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) subclasses defined in this package.

Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) to be thrown.

**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/java/nio/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/nio/channels/spi/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).