| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SelectableChannel.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/ScatteringByteChannel.html)   [**NEXT CLASS**](http://docs.google.com/java/nio/channels/SelectionKey.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/SelectableChannel.html)    [**NO FRAMES**](http://docs.google.com/SelectableChannel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

## **java.nio.channels**

Class SelectableChannel

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.nio.channels.spi.AbstractInterruptibleChannel](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html)  
 **java.nio.channels.SelectableChannel**

**All Implemented Interfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Channel](http://docs.google.com/java/nio/channels/Channel.html), [InterruptibleChannel](http://docs.google.com/java/nio/channels/InterruptibleChannel.html) **Direct Known Subclasses:** [AbstractSelectableChannel](http://docs.google.com/java/nio/channels/spi/AbstractSelectableChannel.html)

public abstract class **SelectableChannel**extends [AbstractInterruptibleChannel](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html)implements [Channel](http://docs.google.com/java/nio/channels/Channel.html)

A channel that can be multiplexed via a [Selector](http://docs.google.com/java/nio/channels/Selector.html).

In order to be used with a selector, an instance of this class must first be *registered* via the [register](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int,%20java.lang.Object)) method. This method returns a new [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) object that represents the channel's registration with the selector.

Once registered with a selector, a channel remains registered until it is *deregistered*. This involves deallocating whatever resources were allocated to the channel by the selector.

A channel cannot be deregistered directly; instead, the key representing its registration must be *cancelled*. Cancelling a key requests that the channel be deregistered during the selector's next selection operation. A key may be cancelled explicitly by invoking its [cancel](http://docs.google.com/java/nio/channels/SelectionKey.html#cancel()) method. All of a channel's keys are cancelled implicitly when the channel is closed, whether by invoking its [close](http://docs.google.com/java/nio/channels/Channel.html#close()) method or by interrupting a thread blocked in an I/O operation upon the channel.

If the selector itself is closed then the channel will be deregistered, and the key representing its registration will be invalidated, without further delay.

A channel may be registered at most once with any particular selector.

Whether or not a channel is registered with one or more selectors may be determined by invoking the [isRegistered](http://docs.google.com/java/nio/channels/SelectableChannel.html#isRegistered()) method.

Selectable channels are safe for use by multiple concurrent threads.

#### Blocking mode

A selectable channel is either in *blocking* mode or in *non-blocking* mode. In blocking mode, every I/O operation invoked upon the channel will block until it completes. In non-blocking mode an I/O operation will never block and may transfer fewer bytes than were requested or possibly no bytes at all. The blocking mode of a selectable channel may be determined by invoking its [isBlocking](http://docs.google.com/java/nio/channels/SelectableChannel.html#isBlocking()) method.

Newly-created selectable channels are always in blocking mode. Non-blocking mode is most useful in conjunction with selector-based multiplexing. A channel must be placed into non-blocking mode before being registered with a selector, and may not be returned to blocking mode until it has been deregistered.

**Since:** 1.4 **See Also:**[SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html), [Selector](http://docs.google.com/java/nio/channels/Selector.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**SelectableChannel**](http://docs.google.com/java/nio/channels/SelectableChannel.html#SelectableChannel())()            Initializes a new instance of this class. |

| **Method Summary** | |
| --- | --- |
| abstract  [Object](http://docs.google.com/java/lang/Object.html) | [**blockingLock**](http://docs.google.com/java/nio/channels/SelectableChannel.html#blockingLock())()            Retrieves the object upon which the [configureBlocking](http://docs.google.com/java/nio/channels/SelectableChannel.html#configureBlocking(boolean)) and [register](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int,%20java.lang.Object)) methods synchronize. |
| abstract  [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) | [**configureBlocking**](http://docs.google.com/java/nio/channels/SelectableChannel.html#configureBlocking(boolean))(boolean block)            Adjusts this channel's blocking mode. |
| abstract  boolean | [**isBlocking**](http://docs.google.com/java/nio/channels/SelectableChannel.html#isBlocking())()            Tells whether or not every I/O operation on this channel will block until it completes. |
| abstract  boolean | [**isRegistered**](http://docs.google.com/java/nio/channels/SelectableChannel.html#isRegistered())()            Tells whether or not this channel is currently registered with any selectors. |
| abstract  [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) | [**keyFor**](http://docs.google.com/java/nio/channels/SelectableChannel.html#keyFor(java.nio.channels.Selector))([Selector](http://docs.google.com/java/nio/channels/Selector.html) sel)            Retrieves the key representing the channel's registration with the given selector. |
| abstract  [SelectorProvider](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html) | [**provider**](http://docs.google.com/java/nio/channels/SelectableChannel.html#provider())()            Returns the provider that created this channel. |
| [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) | [**register**](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int))([Selector](http://docs.google.com/java/nio/channels/Selector.html) sel, int ops)            Registers this channel with the given selector, returning a selection key. |
| abstract  [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) | [**register**](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int,%20java.lang.Object))([Selector](http://docs.google.com/java/nio/channels/Selector.html) sel, int ops, [Object](http://docs.google.com/java/lang/Object.html) att)            Registers this channel with the given selector, returning a selection key. |
| abstract  int | [**validOps**](http://docs.google.com/java/nio/channels/SelectableChannel.html#validOps())()            Returns an [operation set](http://docs.google.com/SelectionKey.html#opsets) identifying this channel's supported operations. |

| **Methods inherited from class java.nio.channels.spi.**[**AbstractInterruptibleChannel**](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html) |
| --- |
| [begin](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html#begin()), [close](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html#close()), [end](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html#end(boolean)), [implCloseChannel](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html#implCloseChannel()), [isOpen](http://docs.google.com/java/nio/channels/spi/AbstractInterruptibleChannel.html#isOpen()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **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()) |

| **Constructor Detail** |
| --- |

### SelectableChannel

protected **SelectableChannel**()

Initializes a new instance of this class.

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

### provider

public abstract [SelectorProvider](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html) **provider**()

Returns the provider that created this channel.

**Returns:**The provider that created this channel

### validOps

public abstract int **validOps**()

Returns an [operation set](http://docs.google.com/SelectionKey.html#opsets) identifying this channel's supported operations. The bits that are set in this integer value denote exactly the operations that are valid for this channel. This method always returns the same value for a given concrete channel class.

**Returns:**The valid-operation set

### isRegistered

public abstract boolean **isRegistered**()

Tells whether or not this channel is currently registered with any selectors. A newly-created channel is not registered.

Due to the inherent delay between key cancellation and channel deregistration, a channel may remain registered for some time after all of its keys have been cancelled. A channel may also remain registered for some time after it is closed.

**Returns:**true if, and only if, this channel is registered

### keyFor

public abstract [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) **keyFor**([Selector](http://docs.google.com/java/nio/channels/Selector.html) sel)

Retrieves the key representing the channel's registration with the given selector.

**Returns:**The key returned when this channel was last registered with the given selector, or null if this channel is not currently registered with that selector

### register

public abstract [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) **register**([Selector](http://docs.google.com/java/nio/channels/Selector.html) sel,  
 int ops,  
 [Object](http://docs.google.com/java/lang/Object.html) att)  
 throws [ClosedChannelException](http://docs.google.com/java/nio/channels/ClosedChannelException.html)

Registers this channel with the given selector, returning a selection key.

If this channel is currently registered with the given selector then the selection key representing that registration is returned. The key's interest set will have been changed to ops, as if by invoking the [interestOps(int)](http://docs.google.com/java/nio/channels/SelectionKey.html#interestOps(int)) method. If the att argument is not null then the key's attachment will have been set to that value. A [CancelledKeyException](http://docs.google.com/java/nio/channels/CancelledKeyException.html) will be thrown if the key has already been cancelled.

Otherwise this channel has not yet been registered with the given selector, so it is registered and the resulting new key is returned. The key's initial interest set will be ops and its attachment will be att.

This method may be invoked at any time. If this method is invoked while another invocation of this method or of the [configureBlocking](http://docs.google.com/java/nio/channels/SelectableChannel.html#configureBlocking(boolean)) method is in progress then it will first block until the other operation is complete. This method will then synchronize on the selector's key set and therefore may block if invoked concurrently with another registration or selection operation involving the same selector.

If this channel is closed while this operation is in progress then the key returned by this method will have been cancelled and will therefore be invalid.

**Parameters:**sel - The selector with which this channel is to be registeredops - The interest set for the resulting keyatt - The attachment for the resulting key; may be null **Returns:**A key representing the registration of this channel with the given selector **Throws:** [ClosedChannelException](http://docs.google.com/java/nio/channels/ClosedChannelException.html) - If this channel is closed [IllegalBlockingModeException](http://docs.google.com/java/nio/channels/IllegalBlockingModeException.html) - If this channel is in blocking mode [IllegalSelectorException](http://docs.google.com/java/nio/channels/IllegalSelectorException.html) - If this channel was not created by the same provider as the given selector [CancelledKeyException](http://docs.google.com/java/nio/channels/CancelledKeyException.html) - If this channel is currently registered with the given selector but the corresponding key has already been cancelled [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If a bit in the ops set does not correspond to an operation that is supported by this channel, that is, if set & ~validOps() != 0

### register

public final [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) **register**([Selector](http://docs.google.com/java/nio/channels/Selector.html) sel,  
 int ops)  
 throws [ClosedChannelException](http://docs.google.com/java/nio/channels/ClosedChannelException.html)

Registers this channel with the given selector, returning a selection key.

An invocation of this convenience method of the form

sc.register(sel, ops)behaves in exactly the same way as the invocationsc.[register](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int,%20java.lang.Object))(sel, ops, null)

**Parameters:**sel - The selector with which this channel is to be registeredops - The interest set for the resulting key **Returns:**A key representing the registration of this channel with the given selector **Throws:** [ClosedChannelException](http://docs.google.com/java/nio/channels/ClosedChannelException.html) - If this channel is closed [IllegalBlockingModeException](http://docs.google.com/java/nio/channels/IllegalBlockingModeException.html) - If this channel is in blocking mode [IllegalSelectorException](http://docs.google.com/java/nio/channels/IllegalSelectorException.html) - If this channel was not created by the same provider as the given selector [CancelledKeyException](http://docs.google.com/java/nio/channels/CancelledKeyException.html) - If this channel is currently registered with the given selector but the corresponding key has already been cancelled [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If a bit in ops does not correspond to an operation that is supported by this channel, that is, if set & ~validOps() != 0

### configureBlocking

public abstract [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) **configureBlocking**(boolean block)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Adjusts this channel's blocking mode.

If this channel is registered with one or more selectors then an attempt to place it into blocking mode will cause an [IllegalBlockingModeException](http://docs.google.com/java/nio/channels/IllegalBlockingModeException.html) to be thrown.

This method may be invoked at any time. The new blocking mode will only affect I/O operations that are initiated after this method returns. For some implementations this may require blocking until all pending I/O operations are complete.

If this method is invoked while another invocation of this method or of the [register](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int)) method is in progress then it will first block until the other operation is complete.

**Parameters:**block - If true then this channel will be placed in blocking mode; if false then it will be placed non-blocking mode **Returns:**This selectable channel **Throws:** [ClosedChannelException](http://docs.google.com/java/nio/channels/ClosedChannelException.html) - If this channel is closed [IllegalBlockingModeException](http://docs.google.com/java/nio/channels/IllegalBlockingModeException.html) - If block is true and this channel is registered with one or more selectors [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### isBlocking

public abstract boolean **isBlocking**()

Tells whether or not every I/O operation on this channel will block until it completes. A newly-created channel is always in blocking mode.

If this channel is closed then the value returned by this method is not specified.

**Returns:**true if, and only if, this channel is in blocking mode

### blockingLock

public abstract [Object](http://docs.google.com/java/lang/Object.html) **blockingLock**()

Retrieves the object upon which the [configureBlocking](http://docs.google.com/java/nio/channels/SelectableChannel.html#configureBlocking(boolean)) and [register](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int,%20java.lang.Object)) methods synchronize. This is often useful in the implementation of adaptors that require a specific blocking mode to be maintained for a short period of time.

**Returns:**The blocking-mode lock object

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SelectableChannel.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/ScatteringByteChannel.html)   [**NEXT CLASS**](http://docs.google.com/java/nio/channels/SelectionKey.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/SelectableChannel.html)    [**NO FRAMES**](http://docs.google.com/SelectableChannel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

[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).