| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Selector.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/SelectionKey.html)   [**NEXT CLASS**](http://docs.google.com/java/nio/channels/ServerSocketChannel.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/Selector.html)    [**NO FRAMES**](http://docs.google.com/Selector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **java.nio.channels**

Class Selector

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

**Direct Known Subclasses:** [AbstractSelector](http://docs.google.com/java/nio/channels/spi/AbstractSelector.html)

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

A multiplexor of [SelectableChannel](http://docs.google.com/java/nio/channels/SelectableChannel.html) objects.

A selector may be created by invoking the [open](http://docs.google.com/java/nio/channels/Selector.html#open()) method of this class, which will use the system's default [selector provider](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html) to create a new selector. A selector may also be created by invoking the [openSelector](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html#openSelector()) method of a custom selector provider. A selector remains open until it is closed via its [close](http://docs.google.com/java/nio/channels/Selector.html#close()) method.

A selectable channel's registration with a selector is represented by a [SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html) object. A selector maintains three sets of selection keys:

* The *key set* contains the keys representing the current channel registrations of this selector. This set is returned by the [keys](http://docs.google.com/java/nio/channels/Selector.html#keys()) method.
* The *selected-key set* is the set of keys such that each key's channel was detected to be ready for at least one of the operations identified in the key's interest set during a prior selection operation. This set is returned by the [selectedKeys](http://docs.google.com/java/nio/channels/Selector.html#selectedKeys()) method. The selected-key set is always a subset of the key set.
* The *cancelled-key* set is the set of keys that have been cancelled but whose channels have not yet been deregistered. This set is not directly accessible. The cancelled-key set is always a subset of the key set.

All three sets are empty in a newly-created selector.

A key is added to a selector's key set as a side effect of registering a channel via the channel's [register](http://docs.google.com/java/nio/channels/SelectableChannel.html#register(java.nio.channels.Selector,%20int)) method. Cancelled keys are removed from the key set during selection operations. The key set itself is not directly modifiable.

A key is added to its selector's cancelled-key set when it is cancelled, whether by closing its channel or by invoking its [cancel](http://docs.google.com/java/nio/channels/SelectionKey.html#cancel()) method. Cancelling a key will cause its channel to be deregistered during the next selection operation, at which time the key will removed from all of the selector's key sets.

Keys are added to the selected-key set by selection operations. A key may be removed directly from the selected-key set by invoking the set's [remove](http://docs.google.com/java/util/Set.html#remove(java.lang.Object)) method or by invoking the [remove](http://docs.google.com/java/util/Iterator.html#remove()) method of an [iterator](http://docs.google.com/java/util/Iterator.html) obtained from the set. Keys are never removed from the selected-key set in any other way; they are not, in particular, removed as a side effect of selection operations. Keys may not be added directly to the selected-key set.

#### Selection

During each selection operation, keys may be added to and removed from a selector's selected-key set and may be removed from its key and cancelled-key sets. Selection is performed by the [select()](http://docs.google.com/java/nio/channels/Selector.html#select()), [select(long)](http://docs.google.com/java/nio/channels/Selector.html#select(long)), and [selectNow()](http://docs.google.com/java/nio/channels/Selector.html#selectNow()) methods, and involves three steps:

1. Each key in the cancelled-key set is removed from each key set of which it is a member, and its channel is deregistered. This step leaves the cancelled-key set empty.
2. The underlying operating system is queried for an update as to the readiness of each remaining channel to perform any of the operations identified by its key's interest set as of the moment that the selection operation began. For a channel that is ready for at least one such operation, one of the following two actions is performed:
   1. If the channel's key is not already in the selected-key set then it is added to that set and its ready-operation set is modified to identify exactly those operations for which the channel is now reported to be ready. Any readiness information previously recorded in the ready set is discarded.
   2. Otherwise the channel's key is already in the selected-key set, so its ready-operation set is modified to identify any new operations for which the channel is reported to be ready. Any readiness information previously recorded in the ready set is preserved; in other words, the ready set returned by the underlying system is bitwise-disjoined into the key's current ready set.

If all of the keys in the key set at the start of this step have empty interest sets then neither the selected-key set nor any of the keys' ready-operation sets will be updated.

1. If any keys were added to the cancelled-key set while step (2) was in progress then they are processed as in step (1).

Whether or not a selection operation blocks to wait for one or more channels to become ready, and if so for how long, is the only essential difference between the three selection methods.

#### Concurrency

Selectors are themselves safe for use by multiple concurrent threads; their key sets, however, are not.

The selection operations synchronize on the selector itself, on the key set, and on the selected-key set, in that order. They also synchronize on the cancelled-key set during steps (1) and (3) above.

Changes made to the interest sets of a selector's keys while a selection operation is in progress have no effect upon that operation; they will be seen by the next selection operation.

Keys may be cancelled and channels may be closed at any time. Hence the presence of a key in one or more of a selector's key sets does not imply that the key is valid or that its channel is open. Application code should be careful to synchronize and check these conditions as necessary if there is any possibility that another thread will cancel a key or close a channel.

A thread blocked in one of the [select()](http://docs.google.com/java/nio/channels/Selector.html#select()) or [select(long)](http://docs.google.com/java/nio/channels/Selector.html#select(long)) methods may be interrupted by some other thread in one of three ways:

* By invoking the selector's [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method,
* By invoking the selector's [close](http://docs.google.com/java/nio/channels/Selector.html#close()) method, or
* By invoking the blocked thread's [interrupt](http://docs.google.com/java/lang/Thread.html#interrupt()) method, in which case its interrupt status will be set and the selector's [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method will be invoked.

The [close](http://docs.google.com/java/nio/channels/Selector.html#close()) method synchronizes on the selector and all three key sets in the same order as in a selection operation.

A selector's key and selected-key sets are not, in general, safe for use by multiple concurrent threads. If such a thread might modify one of these sets directly then access should be controlled by synchronizing on the set itself. The iterators returned by these sets' [iterator](http://docs.google.com/java/util/Set.html#iterator()) methods are *fail-fast:* If the set is modified after the iterator is created, in any way except by invoking the iterator's own [remove](http://docs.google.com/java/util/Iterator.html#remove()) method, then a [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html) will be thrown.

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

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

| **Method Summary** | |
| --- | --- |
| abstract  void | [**close**](http://docs.google.com/java/nio/channels/Selector.html#close())()            Closes this selector. |
| abstract  boolean | [**isOpen**](http://docs.google.com/java/nio/channels/Selector.html#isOpen())()            Tells whether or not this selector is open. |
| abstract  [Set](http://docs.google.com/java/util/Set.html)<[SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html)> | [**keys**](http://docs.google.com/java/nio/channels/Selector.html#keys())()            Returns this selector's key set. |
| static [Selector](http://docs.google.com/java/nio/channels/Selector.html) | [**open**](http://docs.google.com/java/nio/channels/Selector.html#open())()            Opens a selector. |
| abstract  [SelectorProvider](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html) | [**provider**](http://docs.google.com/java/nio/channels/Selector.html#provider())()            Returns the provider that created this channel. |
| abstract  int | [**select**](http://docs.google.com/java/nio/channels/Selector.html#select())()            Selects a set of keys whose corresponding channels are ready for I/O operations. |
| abstract  int | [**select**](http://docs.google.com/java/nio/channels/Selector.html#select(long))(long timeout)            Selects a set of keys whose corresponding channels are ready for I/O operations. |
| abstract  [Set](http://docs.google.com/java/util/Set.html)<[SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html)> | [**selectedKeys**](http://docs.google.com/java/nio/channels/Selector.html#selectedKeys())()            Returns this selector's selected-key set. |
| abstract  int | [**selectNow**](http://docs.google.com/java/nio/channels/Selector.html#selectNow())()            Selects a set of keys whose corresponding channels are ready for I/O operations. |
| abstract  [Selector](http://docs.google.com/java/nio/channels/Selector.html) | [**wakeup**](http://docs.google.com/java/nio/channels/Selector.html#wakeup())()            Causes the first selection operation that has not yet returned to return immediately. |

| **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)) |

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

### Selector

protected **Selector**()

Initializes a new instance of this class.

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

### open

public static [Selector](http://docs.google.com/java/nio/channels/Selector.html) **open**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Opens a selector.

The new selector is created by invoking the [openSelector](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html#openSelector()) method of the system-wide default [SelectorProvider](http://docs.google.com/java/nio/channels/spi/SelectorProvider.html) object.

**Returns:**A new selector **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs

### isOpen

public abstract boolean **isOpen**()

Tells whether or not this selector is open.

**Returns:**true if, and only if, this selector is open

### 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

### keys

public abstract [Set](http://docs.google.com/java/util/Set.html)<[SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html)> **keys**()

Returns this selector's key set.

The key set is not directly modifiable. A key is removed only after it has been cancelled and its channel has been deregistered. Any attempt to modify the key set will cause an [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) to be thrown.

The key set is [not thread-safe](#3dy6vkm).

**Returns:**This selector's key set **Throws:** [ClosedSelectorException](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) - If this selector is closed

### selectedKeys

public abstract [Set](http://docs.google.com/java/util/Set.html)<[SelectionKey](http://docs.google.com/java/nio/channels/SelectionKey.html)> **selectedKeys**()

Returns this selector's selected-key set.

Keys may be removed from, but not directly added to, the selected-key set. Any attempt to add an object to the key set will cause an [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) to be thrown.

The selected-key set is [not thread-safe](#3dy6vkm).

**Returns:**This selector's selected-key set **Throws:** [ClosedSelectorException](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) - If this selector is closed

### selectNow

public abstract int **selectNow**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Selects a set of keys whose corresponding channels are ready for I/O operations.

This method performs a non-blocking [selection operation](#tyjcwt). If no channels have become selectable since the previous selection operation then this method immediately returns zero.

Invoking this method clears the effect of any previous invocations of the [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method.

**Returns:**The number of keys, possibly zero, whose ready-operation sets were updated by the selection operation **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs [ClosedSelectorException](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) - If this selector is closed

### select

public abstract int **select**(long timeout)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Selects a set of keys whose corresponding channels are ready for I/O operations.

This method performs a blocking [selection operation](#tyjcwt). It returns only after at least one channel is selected, this selector's [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method is invoked, the current thread is interrupted, or the given timeout period expires, whichever comes first.

This method does not offer real-time guarantees: It schedules the timeout as if by invoking the [Object.wait(long)](http://docs.google.com/java/lang/Object.html#wait(long)) method.

**Parameters:**timeout - If positive, block for up to timeout milliseconds, more or less, while waiting for a channel to become ready; if zero, block indefinitely; must not be negative **Returns:**The number of keys, possibly zero, whose ready-operation sets were updated **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs [ClosedSelectorException](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) - If this selector is closed [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If the value of the timeout argument is negative

### select

public abstract int **select**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Selects a set of keys whose corresponding channels are ready for I/O operations.

This method performs a blocking [selection operation](#tyjcwt). It returns only after at least one channel is selected, this selector's [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method is invoked, or the current thread is interrupted, whichever comes first.

**Returns:**The number of keys, possibly zero, whose ready-operation sets were updated **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an I/O error occurs [ClosedSelectorException](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) - If this selector is closed

### wakeup

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

Causes the first selection operation that has not yet returned to return immediately.

If another thread is currently blocked in an invocation of the [select()](http://docs.google.com/java/nio/channels/Selector.html#select()) or [select(long)](http://docs.google.com/java/nio/channels/Selector.html#select(long)) methods then that invocation will return immediately. If no selection operation is currently in progress then the next invocation of one of these methods will return immediately unless the [selectNow()](http://docs.google.com/java/nio/channels/Selector.html#selectNow()) method is invoked in the meantime. In any case the value returned by that invocation may be non-zero. Subsequent invocations of the [select()](http://docs.google.com/java/nio/channels/Selector.html#select()) or [select(long)](http://docs.google.com/java/nio/channels/Selector.html#select(long)) methods will block as usual unless this method is invoked again in the meantime.

Invoking this method more than once between two successive selection operations has the same effect as invoking it just once.

**Returns:**This selector

### close

public abstract void **close**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Closes this selector.

If a thread is currently blocked in one of this selector's selection methods then it is interrupted as if by invoking the selector's [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method.

Any uncancelled keys still associated with this selector are invalidated, their channels are deregistered, and any other resources associated with this selector are released.

If this selector is already closed then invoking this method has no effect.

After a selector is closed, any further attempt to use it, except by invoking this method or the [wakeup](http://docs.google.com/java/nio/channels/Selector.html#wakeup()) method, will cause a [ClosedSelectorException](http://docs.google.com/java/nio/channels/ClosedSelectorException.html) to be thrown.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If an 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/Selector.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/SelectionKey.html)   [**NEXT CLASS**](http://docs.google.com/java/nio/channels/ServerSocketChannel.html) | [**FRAMES**](http://docs.google.com/index.html?java/nio/channels/Selector.html)    [**NO FRAMES**](http://docs.google.com/Selector.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[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).