

:mod:`selectors` --- High-level I/O multiplexing

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 1); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 4)

Unknown directive type "module".

```
.. module:: selectors
   :synopsis: High-level I/O multiplexing.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 7)

Unknown directive type "versionadded".

```
.. versionadded:: 3.4
```

Source code: :source:`Lib/selectors.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 9); [backlink](#)

Unknown interpreted text role "source".

Introduction

This module allows high-level and efficient I/O multiplexing, built upon the :mod:`select` module primitives. Users are encouraged to use this module instead, unless they want precise control over the OS-level primitives used.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 16); [backlink](#)

Unknown interpreted text role "mod".

It defines a :class:`BaseSelector` abstract base class, along with several concrete implementations (:class:`KqueueSelector`, :class:`EpollSelector` ...), that can be used to wait for I/O readiness notification on multiple file objects. In the following, "file object" refers to any object with a :meth:`fileno()` method, or a raw file descriptor. See :term:`file object`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 20); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 20); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 20); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 20); [backlink](#)

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 20); [backlink](#)

Unknown interpreted text role "term".

`:class:`DefaultSelector`` is an alias to the most efficient implementation available on the current platform: this should be the default choice for most users.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 26); [backlink](#)

Unknown interpreted text role "class".

Note

The type of file objects supported depends on the platform: on Windows, sockets are supported, but not pipes, whereas on Unix, both are supported (some other types may be supported as well, such as fifos or special file devices).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 36)

Unknown directive type "seealso".

```
.. seealso::

   :mod:`select`
      Low-level I/O multiplexing module.
```

Classes

Classes hierarchy:

```
BaseSelector
+-- SelectSelector
+-- PollSelector
+-- EpollSelector
+-- DevpollSelector
+-- KqueueSelector
```

In the following, *events* is a bitwise mask indicating which I/O events should be waited for on a given file object. It can be a combination of the modules constants below:

Constant	Meaning
<code>:const:`EVENT_READ`</code> <div>System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 63); backlink Unknown interpreted text role "const".</div>	Available for read

Constant	Meaning
const:'EVENT_WRITE' <div> System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 65); backlink Unknown interpreted text role "const". </div>	Available for write

A `:class:'SelectorKey'` is a `:class:`~collections.namedtuple`` used to associate a file object to its underlying file descriptor, selected event mask and attached data. It is returned by several `:class:'BaseSelector'` methods.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 70); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 70); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 70); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 75)

Unknown directive type "attribute".


```
.. attribute:: fileobj
```


File object registered.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 79)

Unknown directive type "attribute".


```
.. attribute:: fd
```


Underlying file descriptor.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 83)

Unknown directive type "attribute".


```
.. attribute:: events
```


Events that must be waited for on this file object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 87)

Unknown directive type "attribute".

```
.. attribute:: data
```

Optional opaque data associated to this file object: for example, this could be used to store a per-client session ID.

A `:class:`BaseSelector`` is used to wait for I/O event readiness on multiple file objects. It supports file stream registration, unregistration, and a method to wait for I/O events on those streams, with an optional timeout. It's an abstract base class, so cannot be instantiated. Use `:class:`DefaultSelector`` instead, or one of `:class:`SelectSelector``, `:class:`KqueueSelector`` etc. if you want to specifically use an implementation, and your platform supports it. `:class:`BaseSelector`` and its concrete implementations support the `:term:`context manager`` protocol.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 95); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 95); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 95); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 95); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 95); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 95); [backlink](#)

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 105)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: register(fileobj, events, data=None)
```

Register a file object for selection, monitoring it for I/O events.

`*fileobj*` is the file object to monitor. It may either be an integer file descriptor or an object with a `fileno()` method.

`*events*` is a bitwise mask of events to monitor.

`*data*` is an opaque object.

This returns a new `:class:`SelectorKey`` instance, or raises a `:exc:`ValueError`` in case of invalid event mask or file descriptor, or `:exc:`KeyError`` if the file object is already registered.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 118)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: unregister(fileobj)
```

Unregister a file object from selection, removing it from monitoring. A file object shall be unregistered prior to being closed.

`*fileobj*` must be a file object previously registered.

This returns the associated `:class:`SelectorKey`` instance, or raises a `:exc:`KeyError`` if `*fileobj*` is not registered. It will raise `:exc:`ValueError`` if `*fileobj*` is invalid (e.g. it has no `fileno()` method or its `fileno()` method has an invalid return value).

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 130)

Unknown directive type "method".

```
.. method:: modify(fileobj, events, data=None)
```

Change a registered file object's monitored events or attached data.

This is equivalent to `:meth:`BaseSelector.unregister(fileobj)`` followed by `:meth:`BaseSelector.register(fileobj, events, data)``, except that it can be implemented more efficiently.

This returns a new `:class:`SelectorKey`` instance, or raises a `:exc:`ValueError`` in case of invalid event mask or file descriptor, or `:exc:`KeyError`` if the file object is not registered.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 142)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: select(timeout=None)
```

Wait until some registered file objects become ready, or the timeout expires.

If `timeout > 0`, this specifies the maximum wait time, in seconds. If `timeout <= 0`, the call won't block, and will report the currently ready file objects. If `*timeout*` is `None`, the call will block until a monitored file object becomes ready.

This returns a list of `(key, events)` tuples, one for each ready file object.

`*key*` is the `:class:`SelectorKey`` instance corresponding to a ready file object.

`*events*` is a bitmask of events ready on this file object.

```
.. note::
    This method can return before any file object becomes ready or the
    timeout has elapsed if the current process receives a signal: in this
    case, an empty list will be returned.
```

```
.. versionchanged:: 3.5
    The selector is now retried with a recomputed timeout when interrupted
    by a signal if the signal handler did not raise an exception (see
    :pep:`475` for the rationale), instead of returning an empty list
    of events before the timeout.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 171)

Unknown directive type "method".

```
.. method:: close()
```

Close the selector.

This must be called to make sure that any underlying resource is freed. The selector shall not be used once it has been closed.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 178)

Unknown directive type "method".

```
.. method:: get_key(fileobj)
```

Return the key associated with a registered file object.

This returns the :class:`SelectorKey` instance associated to this file object, or raises :exc:`KeyError` if the file object is not registered.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 185)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: get_map()
```

Return a mapping of file objects to selector keys.

This returns a :class:`~collections.abc.Mapping` instance mapping registered file objects to their associated :class:`SelectorKey` instance.

The default selector class, using the most efficient implementation available on the current platform. This should be the default choice for most users.

:func:`select.select`-based selector.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 203); [backlink](#)

Unknown interpreted text role "func".

:func:`select.poll`-based selector.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 208); [backlink](#)

Unknown interpreted text role "func".

:func:`select.epoll`-based selector.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 213); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 215)

Unknown directive type "method".

```
.. method:: fileno()
```

This returns the file descriptor used by the underlying :func:`select.epoll` object.

:func:`select.devpoll`-based selector.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 222); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 224)

Unknown directive type "method".

```
.. method:: fileno()
```

This returns the file descriptor used by the underlying
:func:`select.devpoll` object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 229)

Unknown directive type "versionadded".

```
.. versionadded:: 3.5
```

:func:`select.kqueue`-based selector.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 233); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) selectors.rst, line 235)

Unknown directive type "method".

```
.. method:: fileno()
```

This returns the file descriptor used by the underlying
:func:`select.kqueue` object.

Examples

Here is a simple echo server implementation:

```
import selectors
import socket

sel = selectors.DefaultSelector()

def accept(sock, mask):
    conn, addr = sock.accept() # Should be ready
    print('accepted', conn, 'from', addr)
    conn.setblocking(False)
    sel.register(conn, selectors.EVENT_READ, read)

def read(conn, mask):
    data = conn.recv(1000) # Should be ready
    if data:
        print('echoing', repr(data), 'to', conn)
        conn.send(data) # Hope it won't block
    else:
        print('closing', conn)
        sel.unregister(conn)
        conn.close()

sock = socket.socket()
sock.bind(('localhost', 1234))
sock.listen(100)
sock.setblocking(False)
sel.register(sock, selectors.EVENT_READ, accept)

while True:
    events = sel.select()
    for key, mask in events:
        callback = key.data
        callback(key.fileobj, mask)
```