

NAME

IO::Poll - Object interface to system poll call

SYNOPSIS

```
use IO::Poll qw(POLLRDNORM POLLWRNORM POLLIN POLLHUP);

my $poll = IO::Poll->new();

$poll->mask($input_handle => POLLIN);
$poll->mask($output_handle => POLLOUT);

$poll->poll($timeout);

my $ev = $poll->events($input);
```

DESCRIPTION

[IO::Poll](#) is a simple interface to the system level poll routine.

METHODS

`mask (IO [, EVENT_MASK])`

If `EVENT_MASK` is given, then, if `EVENT_MASK` is non-zero, `IO` is added to the list of file descriptors and the next call to poll will check for any event specified in `EVENT_MASK`. If `EVENT_MASK` is zero then `IO` will be removed from the list of file descriptors.

If `EVENT_MASK` is not given then the return value will be the current event mask value for `IO`.

`poll ([TIMEOUT])`

Call the system level poll routine. If `TIMEOUT` is not specified then the call will block. Returns the number of handles which had events happen, or -1 on error.

`events (IO)`

Returns the event mask which represents the events that happened on `IO` during the last call to `poll`.

`remove (IO)`

Remove `IO` from the list of file descriptors for the next poll.

`handles([EVENT_MASK])`

Returns a list of handles. If `EVENT_MASK` is not given then a list of all handles known will be returned. If `EVENT_MASK` is given then a list of handles will be returned which had one of the events specified by `EVENT_MASK` happen during the last call to `poll`.

SEE ALSO

[poll\(2\)](#), [IO::Handle](#), [IO::Select](#)

AUTHOR

Graham Barr. Currently maintained by the Perl Porters. Please report all bugs to perlbug@perl.org.

COPYRIGHT

Copyright (c) 1997-8 Graham Barr <gbarr@pobox.com>. All rights reserved. This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.