NAME

IO::Pipe - supply object methods for pipes

SYNOPSIS

```
use IO::Pipe;

$pipe = new IO::Pipe;

if($pid = fork()) { # Parent
    $pipe->reader();

    while(<$pipe>) {
 ...
    }

}
elsif(defined $pid) { # Child
    $pipe->writer();

    print $pipe ...
}

or

$pipe = new IO::Pipe;

$pipe->reader(qw(ls -l));

while(<$pipe>) {
    ...
}
```

DESCRIPTION

`IO::Pipe` provides an interface to creating pipes between processes.

CONSTRUCTOR

new ([READER, WRITER])

Creates an `IO::Pipe`, which is a reference to a newly created symbol (see the `Symbol` package). `IO::Pipe::new` optionally takes two arguments, which should be objects blessed into `IO::Handle`, or a subclass thereof. These two objects will be used for the system call to `pipe`. If no arguments are given then method `handles` is called on the new `IO::Pipe` object.

These two handles are held in the array part of the GLOB until either `reader` or `writer` is called.

METHODS

reader ([ARGS])

The object is re-blessed into a sub-class of `IO::Handle`, and becomes a handle at the reading end of the pipe. If `ARGS` are given then `fork` is called and `ARGS` are passed to exec.

writer ([ARGS])

The object is re-blessed into a sub-class of `IO::Handle`, and becomes a handle at the writing end of the pipe. If `ARGS` are given then `fork` is called and `ARGS` are passed to exec.

handles ()

This method is called during construction by `IO::Pipe::new` on the newly created `IO::Pipe` object. It returns an array of two objects blessed into `IO::Pipe::End`, or a subclass thereof.

SEE ALSO

IO::Handle

AUTHOR

Graham Barr. Currently maintained by the Perl Porters. Please report all bugs to <perl5-porters@perl.org>.

COPYRIGHT

Copyright (c) 1996-8 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.