

**NAME**

**ptyopen** – Runs a command in a pseudo-terminal and redirects it to stdin/stdout

**SYNOPSIS**

**ptyopen** [-v] [--verbose] [-V] [--version] [-u] [--unsecure] [-w] [--write] [-r *size*] [ --ringsize *size*] [-g *geometry*] [--geometry *geometry*] [-h] [--help] *command* [*arguments...*]

**DESCRIPTION**

**ptyopen** creates a pseudo-terminal and runs the specified *command* with its *arguments* in it. All tty output of the slave *command* is redirected to **ptyopen's** stdout. All data fed into **ptyopen's** stdin is presented to the slave *command* as terminal input.

**ptyopen** is useful to control programmatically commands that use the terminal directly (typically by opening /dev/tty), like **dump(1)** or **telnet(1)** for example.

**OPTIONS**

-v, --verbose

Runs verbosely. Namely, **ptyopen** will report "benign" errors while handling terminal capabilities and window sizing. **ptyopen** will also report verbosely the exit code of *command*.

-V, --version

Prints **ptyopen's** version.

-u, --unsecure

If running under a system which requires **ptyopen** to be setuid root, the unsecure option allows **ptyopen** to run without changing the permissions and ownership of the pseudo-terminal (and indeed makes **ptyopen** usable without making it setuid, at the expense of security). This option is not available on systems not requiring **ptyopen** to be setuid root (Linux >= 2.2 with glibc >= 2.1, Solaris, ...)

-w, --write

Allows write permissions to the pseudo-terminal. This is probably useless, unless you're running some interactive shell under **ptyopen**.

-r, --ringsize <*size*>

Takes an argument, the number of bytes of **ptyopen's** input and output buffers. When writing to **ptyopen's** stdin, **ptyopen** buffers some amount of data. Similarly, when the *program* writes to the terminal, **ptyopen** buffers the output before sending it to stdout. One might want to change the default buffer size for obscure reasons.

-g, --geometry <*geometry*>

Makes *program* believe it runs in a terminal whose size is given by <*geometry*>. For example 80x25 or 132x50.

-h, --help

Prints a summary of these options.

**NOTES****SIGNALS**

**ptyopen** traps almost all common deadly signals so that the pseudo-terminal permissions are restored correctly.

**ptyopen** also handles SIGTSTP and SIGCONT so that the slave *command* gets stopped and restarted when **ptyopen**'s state changes.

Finally, window size changes get communicated correctly through SIGWINCH, unless the *-g* or *--geometry* (forced geometry) options are used.

## **BUFFERS**

**ptyopen** uses some internal buffers, which can be resized with the *-r* or *--ringsize* options. These buffers are in used in addition to the buffers used by the pseudo-terminal driver, and eventually the pipes and sockets connected to **ptyopen**'s stdin and stdout.

## **END OF FILE**

If **ptyopen**'s stdin is closed, then an EOF character (as determined by the pseudo-terminal's capabilities) is sent to the terminal. Whether or not the slave *command* will take action on this EOF character depends on the *command* itself, and not **ptyopen**.

## **EXIT CODE**

**ptyopen** exits with the exit code of the slave *command* unless it encounters some error. In that case it will exit with 255.

## **SECURITY**

On BSD systems, **ptyopen** needs to be setuid root so that the pseudo-terminal's permissions and ownership can be changed. **pty open** will refuse to run without being setuid root (unless the *-u* or *--unsecure* options are used).

Although great care has been taken to ensure that **ptyopen** does not present a security risk, your mileage may vary.

## **SEE ALSO**

**expect(1)**, **pty(7)**.

## **AUTHOR**

Written and Copyright (C) 1998-2024 Philippe Troin <phil+github-commits@fifi.org>. THIS PROGRAM COMES WITH ABSOLUTELY NO WARRANTY.

## **HISTORY**

This program was written as a way to control **dump(8)** from a backup script.