

# **pv(1) - Linux man page**

---

## **Name**

pv - monitor the progress of data through a pipe

## **Synopsis**

**pv** [*OPTION*] [*FILE*]...

**pv** [-h|-V]

## **Description**

**pv** allows a user to see the progress of data through a pipeline, by giving information such as time elapsed, percentage completed (with progress bar), current throughput rate, total data transferred, and ETA.

To use it, insert it in a pipeline between two processes, with the appropriate options. Its standard input will be passed through to its standard output and progress will be shown on standard error.

**pv** will copy each supplied **FILE** in turn to standard output (- means standard input), or if no **FILES** are specified just standard input is copied. This is the same behaviour as **cat**(1).

A simple example to watch how quickly a file is transferred using **nc**(1):

**pv file | nc -w 1 somewhere.com 3000**

A similar example, transferring a file from another process and passing the expected size to **pv**:

```
cat file | pv -s 12345 | nc -w 1  
somewhere.com 3000
```

A more complicated example using numeric output to feed into the ***dialog***(1) program for a full-screen progress display:

```
(tar cf - . \  
| pv -n -s 'du -sb . | awk '{print $1}'" \  
| gzip -9 > out.tgz) 2>&1 \  
| dialog --gauge 'Progress' 7 70
```

Frequent use of this third form is not recommended as it may cause the programmer to overheat.

## Options

**pv** takes many options, which are divided into display switches, output modifiers, and general options.

## Display Switches

If no display switches are specified, **pv** behaves as if **-p**, **-t**, **-e**, **-r**, and **-b** had been given (i.e. everything is switched on). Otherwise, only those display types that are explicitly switched on will be shown.

### **-p, --progress**

Turn the progress bar on. If standard input is not a file and no size was given (with the **-s** modifier), the progress bar cannot indicate how close to completion the transfer is, so it will just move left and right to indicate that data is moving.

### **-t, --timer**

Turn the timer on. This will display the total elapsed time that **pv** has been running for.

### **-e, --eta**

Turn the ETA timer on. This will attempt to guess, based on previous transfer rates and the total data size, how long it will be before completion. This option will have no effect if the total data size cannot be determined.

**-r, --rate**

Turn the rate counter on. This will display the current rate of data transfer.

**-b, --bytes**

Turn the total byte counter on. This will display the total amount of data transferred so far.

**-n, --numeric**

Numeric output. Instead of giving a visual indication of progress, **pv** will give an integer percentage, one per line, on standard error, suitable for piping (via convoluted redirection) into ***dialog**(1)*. Note that **-f** is not required if **-n** is being used.

**-q, --quiet**

No output. Useful if the **-L** option is being used on its own to just limit the transfer rate of a pipe.

## Output Modifiers

**-W, --wait**

Wait until the first byte has been transferred before showing any progress information or calculating any ETAs. Useful if the program you are piping to or from requires extra information before it starts, eg piping data into ***gpg**(1)* or ***mcrypt**(1)* which require a passphrase before data can be processed.

**-s SIZE, --size SIZE**

Assume the total amount of data to be transferred is **SIZE** bytes when calculating percentages and ETAs. The same suffixes of "k", "m" etc can be used as with **-L**.

**-l, --line-mode**

Instead of counting bytes, count lines (newline characters). The progress bar will only move when a new line is found, and the value passed to the **-s** option will be interpreted as a line count.

**-i SEC, --interval SEC**

Wait **SEC** seconds between updates. The default is to update every second. Note that this can be a decimal such as 0.1.

**-w WIDTH, --width WIDTH**

Assume the terminal is **WIDTH** characters wide, instead of trying to work it out (or assuming 80 if it cannot be guessed).

**-H HEIGHT, --height HEIGHT**

Assume the terminal is **HEIGHT** rows high, instead of trying to work it out (or assuming 25 if it cannot be guessed).

**-N NAME, --name NAME**

Prefix the output information with **NAME**. Useful in conjunction with **-c** if you have a complicated pipeline and you want to be able to tell different parts of it apart.

**-f, --force**

Force output. Normally, **pv** will not output any visual display if standard error is not a terminal. This option forces it to do so.

**-c, --cursor**

Use cursor positioning escape sequences instead of just using carriage returns. This is useful in conjunction with **-N** (name) if you are using multiple **pv** invocations in a single, long, pipeline.

## Data Transfer Modifiers

**-L RATE, --rate-limit RATE**

Limit the transfer to a maximum of **RATE** bytes per second. A suffix of "k", "m", "g", or "t" can be added to denote kilobytes (\*1024), megabytes, and so on.

### **-B BYTES, --buffer-size BYTES**

Use a transfer buffer size of **BYTES** bytes. A suffix of "k", "m", "g", or "t" can be added to denote kilobytes (\*1024), megabytes, and so on. The default buffer size is the block size of the input file's filesystem multiplied by 32 (512kb max), or 400kb if the block size cannot be determined.

### **-R PID, --remote PID**

If **PID** is an instance of **pv** that is already running, **-R PID** will cause that instance to act as though it had been given this instance's command line instead. For example, if **pv -L 123k** is running with process ID 9876, then running **pv -R 9876 -L 321k** will cause it to start using a rate limit of 321k instead of 123k. Note that some options cannot be changed while running, such as **-c**, **-l**, and **-f**.

## **General Options**

### **-h, --help**

Print a usage message on standard output and exit successfully.

### **-V, --version**

Print version information on standard output and exit successfully.