NAME

vol − split input on or combine output from several volumes

SYNOPSIS

vol [**−rw1**] [**−b** *blocksize*] [**−m** *multiple*] [*size*] *device*

DESCRIPTION

Vol either reads a large input stream from standard input and distributes it over several volumes or combines volumes and sends them to standard output. The size of the volumes is determined automatically if the device supports this, but may be specified before the argument naming the device if automated detection is not possible or if only part of the physical volume is used. The direction of the data is automatically determined by checking whether the input or output of **vol** is a file or pipe. Use the **−r** or **−w** flag if you want to specify the direction explicitly, in shell scripts for instance.

Vol waits for each new volume to be inserted, typing return makes it continue. If no size is explicitely given then the size of the device is determined each time before it is read or written, so it is possible to mix floppies of different sizes. If the size cannot be determined (probably a tape) then the device is assumed to be infinitely big. **Vol** can be used both for block or character devices. It will buffer the data and use a block size appropriate for fixed or variable block sized tapes.

Vol reads or writes 8192 bytes to block devices, usually floppies. Character devices are read or written using a multiple of 512 bytes. This multiple has an upper limit of 32767 bytes (16-bit machine), 64 kb (32-bit), or even 1 Mb (32-bit VM). The last partial write to a character device is padded with zeros to the block size. If a character device is a tape device that responds to the **mtio**(4) status call then the reported tape block size will be used as the smallest unit. If the tape is a variable block length device then it is read or written like a block device, 8192 bytes at the time, with a minimum unit of one byte.

All sizes may be suffixed by the letters **M**, **k**, **b** or **w** to multiply the number by mega, kilo, block (512), or word (2). The volume size by default in kilobytes if there is no suffix.

OPTIONS

−rw Explicitly specify reading or writing. Almost mandatory in scripts.

−1 Just one volume, start immediately.

−b *blocksize*
Specify the device block size.

−m *multiple*
Specify the maximum read or write size of multiple blocks. The **−b** and **−m** options allow one to modify the block size assumptions that are made above. These assumptions are **−b 1 −m 8192** for block devices or variable length tapes, and **−b 512 −m 65536** for character devices (32 bit machine.) These options will not override the tape block size found out with an **mtio**(4) call. The multiple may be larger then the default if **vol** can allocate the memory required.

EXAMPLES

To back up a tree to floppies as a compressed tarfile:

```
tar cf − . | compress | vol /dev/fd0
```

To restore a tree from 720 kb images from possibly bigger floppies:

```
vol 720 /dev/fd0 | uncompress | tar xfp −
```

Read or write a device with 1024 byte blocks:

```
vol −b 1k /dev/rsd15
```

Read or write a variable block length tape using blocking factor 20 as used by default by many **tar**(1) commands:

```
vol −m 20b /dev/rst5
```

Note that **−m** was used in the last example. It sets the size to use to read or write, **−b** sets the basic block size that may be written in multiples.

SEE ALSO

dd(1), **tar**(1), **mt**(1), **mtio**(4).