NAME

numfmt — TODO

SYNOPSIS

numfmt [**--from**=**none**|**auto**|**si**|**iec**|**iec-i**] [**--to**=**none**|**auto**|**si**|**iec**|**iec-i**]
[**--from-unit**=*size*] [**--to-unit**=*size*]
[**--grouping**] [**--padding**=*width*]
[**--format**=[*pre*]**%**[**'**][**-**][**0**][*width*][.[*precision*]]**f**[*post*]] [**--suffix**=*suffix*]
[**--round**=**from-zero**|**towards-zero**|**nearest**|**up**|**down**]
[**--invalid**=**ignore**|**warn**|**fail**|**abort**] [**-z**] [**-d** *field-delimiter*]
[**--field**=*range*[,*range*]...] [**--header**[=*lines*]] < file
numfmt [**--from**=**none**|**auto**|**si**|**iec**|**iec-i**] [**--to**=**none**|**auto**|**si**|**iec**|**iec-i**]
[**--from-unit**=*size*] [**--to-unit**=*size*]
[**--grouping**] [**--padding**=*width*]
[**--format**=[*pre*]**%**[**'**][**-**][**0**][*width*][.[*precision*]]**f**[*post*]] [**--suffix**=*suffix*]
[**--round**=**from-zero**|**towards-zero**|**nearest**|**up**|**down**]
[**--invalid**=**ignore**|**warn**|**fail**|**abort**] [**-z**] [**-d** *field-delimiter*]
[**--field**=*range*[,*range*]...] *line*...

DESCRIPTION

Re-formats numbers in *line*s (or read from the standard input stream) to/from a human-readable SI-suffxed format, additionally multiplied/divided by *size*s to the standard output stream. By default, input *line*s are split by whitespace and the fields formatted fields' widths are preserved; **-d** splits on *field-delimiter*, like cut(1). The output precision is the same as the input precision (except when upgrading an unsuffixed integer with a suffix, and the resulting value has a fractional part, it's given a one-digit precision).

Format	Example	Description	Suffixes
none	1024000	a plain number	*(none)*
si	1.1M	*1000*-based human-readable SI suffix	k M G T P E Z Y R Q
iec	1000K	*1024*-based human-readable SI suffix	K M G T P E Z Y R Q
iec-i	1000Ki	**iec** but with **i** after	Ki Mi Gi Ti Pi Ei Zi Yi Ri Qi

Lists the free and used space accessible to unprivileged users on mounted filesystems. With no *path*s, all filesystems are listed; otherwise, those on which the *path*s lie are, or if any correspond to a source device node of a mounted filesystem — that filesystem.

With neither *path*s nor **-a**, zero-size, inaccessible, **autofs**, and bind- and over-mounted filesystems are omitted. Additional filtering via **-txl** is applied in all cases.

With **-hH**, output is in a human-readable 3.2T-style. Otherwise, without **-P**, output is in rounded-up blocks of **-B**, the first valid of the DF_BLOCK_SIZE, BLOCK_SIZE, BLOCKSIZE environment variables, or **1024** bytes. Otherwise, the output is in blocks of **-k** or **512** bytes.

-B and the block size environment variables are in the case-insensitive format:
[*base*][**KMGTPEZY**][**B**] (with at least one of {*base*, **KMGTPEZY**, **B**})
Where *base* is an optionally-floating-point number of bytes, defaulting to **1**, which is then optionally multiplied by the relevant unit. **B** sets the unit multiplier to **1000** (from **1024**). The block size is equal to $ base · unit sup mult $, if any, or *base*.

Columns

numfmt produces a columnated listing; numeric columns are right-aligned, others are left-aligned.

The default columns are **source**,**size**,**used**,**avail**,**pcent**,**target**.
With **-i**, it's the same, but with i-nodes: **source**,**itotal**,**iused**,**iavail**,**ipcent**,**target**.

-T inserts **fstype** as the second column.

Available columns are:

source	`Filesystem`	Mount source, like `/dev/nvme0n1p2` or `tarta:/pub`.
fstype	`Type`	Filesystem type, like **vfat** or **nfs4**.
itotal	`i-nodes`	Total amount of i-nodes.
iused	`iUsed`	Amount of i-nodes in use.
iavail	`iFree`	Amount of i-nodes available to unprivileged users.
ipcent	`iUse%`	**ceil**(100 · **iused** / (**iused** − **iavail**))`%`
size	*1k*`-blocks`	Total capacity, in blocks, rounded up.
used	`Used`	Capacity used, in blocks, rounded up.
avail	`Avail`	Capacity available to unprivileged users, in blocks, rounded up.
pcent	`Use%`	**ceil**(100 · **used** / (**used** − **avail**))`%`
file	`File`	*path* this corresponds to, or "**-**".
target	`Mounted on`	Mount point, like `/boot` or `/`.

-hH change the **size** heading to "`Size`". **-P** changes the headings for **size** to "*1024*`-blocks`" (plain number), **avail** to "`Available`", and **pcent** to "`Capacity`".

OPTIONS

-P, **--portability**
: Default to **512**-byte blocks and IEEE Std 1003.1-2008 ("POSIX.1")-compatible headings. Don't process environment variables. Disables **-,**.

-l, **--local**
: Filter out non-local filesystems: **cifs**, **afs**, and those with a colon ("**:**") in their mount source (except **zfs**).

-a, **--all**
: With no *path*s, remove default filter (see above). No effect otherwise.

-i, **--inodes**
: Set output format to **source**,**itotal**,**iused**,**iavail**,**ipcent**,**target**.

-T, **--print-type**
: With default or **-i** format, add **fstype** as the second column.

-t, **-F**, **--print-type**=*only-type*
: Filter out all filesystems with types different than *only-type*s.

-x, **--exclude-type**=*not-type*
: Filter out all filesystems with types equal to *not-type*s.

-k
: Equivalent to **-B** `1k`.

-m
: Equivalent to **-B** `1M`.

-g
: Equivalent to **-B** `1G`.

-B, **--block-size**=*blocksize*
: Set block size for non-i-node output.

-,
: Format all numbers with thousands-separators (where the locale places them, so not necessarily every thousand). For example, this may turn a filesystem size of **11586198962176** (around 11.5 tebibytes) into **11,586,198,962,176**.

-h, **--human-readable**
: Fold all non-i-node sizes into a human readable *1024*-based `3.2T` style. Supersedes **-,**.

-H, **--si**
: Like **-h** but *1000*. Supersedes **-,**.

--output[=*col*[,*col*]...]
: List *col*umns (or all if no argument). See the **Columns** section. Excludes **-T**.

--sync
: Run `sync`(2) before collecting any data. This may provide more accurate statistics on some systems.

--no-sync
: Don't. This is the default.

--total
: Write a final line with summary information of all filesystems listed before.

-v Ignored for compatibility with AT&T System V Release 4 UNIX on i386.

ENVIRONMENT

`DF_BLOCK_SIZE`, `BLOCK_SIZE`, `BLOCKSIZE` Unless **-P**, the first valid of these variables sets the default block size, instead of **1024**.

FILES

`/etc/mtab` List of mounted filesystems.

EXIT STATUS

1 if a *path* couldn't be accessed, or if no filesystems were listed (the heading is also suppressed in that case).

SEE ALSO

`fstab`(5), `mount`(8)

STANDARDS

Conforms to IEEE Std 1003.1-2024 ("POSIX.1"); as historical practice is irreconcilable, portable output is achieved only with **-P** and an optional **-k**; even then, columnation is acceptable, but not required. Some implementations, like the GNU system, blunder even this, and default to **1024**-byte blocks. The only truly portable invocation of **numfmt** is

```
$ POSIXLY_CORRECT= numfmt -P [-k] [path|device]...
```

It also defines a **-t** XSI extension as "Include total allocated-space figures in the output", but leaves it explicitly unspecified — it is supported in that form on AT&T System III UNIX derivatives; some, like SunOS 5 (Solaris 2), provide an **-F** option with the same meaning as this implementation's **-t** (except for `/usr/ucb/`**numfmt**, which provides **-t** as described). The **-F -t** alias is available but undocumented in the GNU system.

This implementation is compatible with the GNU system, except it's broken as noted above, more strict about mixing output format flags, disallows block sizes with **B** but without a multiplier, as well as lowercase **B**, and only supports integer *base*s. **-g,** are an extension, originating from FreeBSD. `BLOCKSIZE` is an extension, originating from 4.4BSD. The `DF_BLOCK_SIZE`, `BLOCK_SIZE` spellings are extensions, originating from the GNU system.

HISTORY

Research UNIX

Appears in the first edition of the UNIX Programmer's Manual as `df`(I):

```
NAME        df  --  disk free
SYNOPSIS    df [ filesystem ]
```

Writing out plus-separated free block counts for `/dev/rf0`, `/dev/rk1`, `/dev/rk2` and `/dev/rk3`, or *filesystem* (a file with at least a filesystem superblock). The second edition provides a different list and notes that these are the "normally mounted file systems". Until the advent of the mount tab, **numfmt** was built with this list matching the system configuration.

Version 5 AT&T UNIX, alongside a new filesystem format, sees a rewrite in C and "`device` *blocks*"-style output.

Version 7 AT&T UNIX installs **numfmt** set-user-ID, since it reads superblocks directly. This practice continues intermittently across all **numfmt**s that do so.

The BSD

4BSD sees the first version that reads `/etc/mtab`:

df [**–i**] [**–l**] [filesystem ...] [file ...]

It also introduces a faintly familiar format:

```
Filesystem  Mounted on  blocks    used    free  % used
```

With **-i** appending the i-node fields:

`iused ifree %iused`

And **-l** adding `hardway`, which reads the underlying block device's free list, after `free`. The `blocks` are in the real filesystem block size — **1024**; the page notes this as being twice the block size of `du`(1) and `ls`(1); X/Open Portability Guide Issue 4 ("XPG4") notes this as one of the reasons for the irreconcilability of existing practice (and, hence, **-P**).

4.2BSD removes **-l** and changes the format to a less confusing (and more familiar) one:

`Filesystem kbytes used avail capacity Mounted on`

With **-i** inserting the fields before `Mounted on`. IEEE Std 1003.1-2008 ("POSIX.1") erroneously notes this as the **-P** format.

4.3BSD–Reno sees

df [**-ikn**] [*file* | *filesystem* **...**]

With an automatically-scaling `Filesystem` column and **512**-byte blocks with an appropriate `512-blks` heading by default, with **-k** to revert to **1024**. **-n** doesn't block for mount information.

4.4BSD sees a **SYNOPSIS** of

df [**-in**] [**-t** *type*] [*file* | *filesystem* **...**]

-k was replaced with a `BLOCKSIZE` environment variable in the

base[**KMG**]

format. Multiple instances of **-t** can be used to filter from a list of supported filesystems (like **ufs**, **nfs**, or **kernfs**) and groups (**all**, **local**, **misc**), or to filter them out by prepending **no**.

4.4BSD–Lite2 allows *one* **-t**, but sees it as a (**no**-prefixed) comma-separated list of filesystem types.

System V

Programmer's Workbench (PWB/UNIX) has

df [**-uqs**] [**-t** number] [arg ...]

and uses `/etc/mnttab` by default, accepting *arg*s of either the source device or the mount point. The default output is relatively similar to Version 5 AT&T UNIX's as "`/dev/`*mountpoint* `(device)` *free-blocks*". **-t** compares the free block count on processed filesystems with *number*, writing "*maj min* **Y**" if it's more and **N**, alongside exiting with **1** otherwise. **-u** writes a verbose usage listing for each filesystem:

```
/dev/mountpoint(device)       size total blocks
                internal-i-nodes system use
                free-blocks free
                blocks-used used
                free-i-nodes free inodes
```

-q reads the free space directly out of the superblock. **-s** suppresses all output. Quite the mess!

No wonder that AT&T System III UNIX is completely different. By default, the output format is

mountpoint`(device): `*free-blocks* `blocks` *free-i-nodes* `i-nodes`

With **-t**, it grows an additional line:

`(`*size* `total blocks,` *i-node-blocks* `for i-nodes)`

With **-f**, the output is as default, but without the i-node count, and the free block count is validated against the superblock. **-q** is ignored.

AT&T System V UNIX fixes **-t** with **-f** and uses the filesystem name from the superblock for unmounted filesystems. Non-**512**-byte filesystem blocks are corrected to **512**-byte output blocks.

AT&T System V Release 3 UNIX replaces **-q** with **-l**, skipping network mounts, and gains the ability to match a file to its filesystem. The default output gains an asterisk ("*") after `blocks` for subsequent iterations over the same source. **-t** now adds

`total: `*size* `blocks` *total-i-nodes* `i-nodes`

Cool, but what if I told you it could be worse? AT&T System V Release 4 UNIX ships

`df [-F FSType] [-begklntVv] [current_options] [-o specific_options] [directory | special ...]`

(with **-v** on i386 only). The default output is similar to AT&T System III UNIX's, save for `files` in-

stead of `i-nodes`. The interesting part (rather, the one that isn't eye-piercingly insane like the filesystem-specific helper programs, the intricate precedence rules, or transitive handling of a remote mount on the remote host) are the output formats (with *`free`* in kilobytes and **512**-byte blocks):

-n `mountpoint:` *`type`*

-e

```
Filesystem             ifree
         device       free-ino
```

-b

```
Filesystem             avail
         device           free
```

-be

```
         mountpoint(          device):      free kilobytes
         mountpoint(          device):  free-ino files
```

-t The same as AT&T System III UNIX with **-t**, save for `files` instead of `i-nodes`:

```
          mountpoint(          device):   free-blo blocksfree-ino
files
                                 total:    blocks blocks   i-nodes
files
```

-k `filesystem kbytes used avail capacity mounted on` (with a two-digit precision for `capacity`)

-g *`UUID`* is a 32-bit integer, *`filesystem-name`* is a 32-character string; *`flags`* are the mount flags, numerically:

```
        mountpoint(          device):     bsz block size      fsz
frag size
  blocks  total  blocksfree-bl  free  blocksavailbl  available
i-nodes total files
free-in      free      files          UUID      filesys      id
filesystem-name
   type fstype   0x000flags flag        maxlen filename length
```

-v The heading is written whenever the option is specified, but the usage is an integer:

```
Mount Dir  Filesystem          blocks      used      free  %used
```

Standards

X/Open Portability Guide Issue 2 ("XPG2") specifies

df [–t][file-system ...]

marked OF ("Output format incompletely specified" – it isn't at all) UN ("Possibly unsupportable feature"), and declares **512**-byte units while saying that some systems don't report in **512**-byte units, but the *`file-system`* behaviour is as present-day.

IEEE Std 1003.2-1992 ("POSIX.2") excludes **numfmt**, as it doesn't address the concept of filesystems; it's included in the IEEE Std 1003.2a-1992 ("POSIX.2") (User Portability Extension) supplement, creating **-Pk** with their well-defined formats and block sizes of today; X/Open Portability Guide Issue 4 ("XPG4") aligns its definition therewith, retaining **-t** as an extension.