

NAME

numfmt – Convert numbers from/to human-readable strings

SYNOPSIS

numfmt [*OPTION*]... [*NUMBER*]...

DESCRIPTION

Reformat *NUMBER*(s), or the numbers from standard input if none are specified.

Mandatory arguments to long options are mandatory for short options too.

--debug

print warnings about invalid input

-d, --delimiter=*X*

use *X* instead of whitespace for field delimiter

--field=*FIELDS*

replace the numbers in these input fields (default=1) see *FIELDS* below

--format=*FORMAT*

use printf style floating–point *FORMAT*; see *FORMAT* below for details

--from=*UNIT*

auto–scale input numbers to *UNIT*s; default is 'none'; see *UNIT* below

--from–unit=*N*

specify the input unit size (instead of the default 1)

--grouping

use locale–defined grouping of digits, e.g. 1,000,000 (which means it has no effect in the C/POSIX locale)

--header[=*N*]

print (without converting) the first *N* header lines; *N* defaults to 1 if not specified

--invalid=*MODE*

failure mode for invalid numbers: *MODE* can be: abort (default), fail, warn, ignore

--padding=*N*

pad the output to *N* characters; positive *N* will right–align; negative *N* will left–align; padding is ignored if the output is wider than *N*; the default is to automatically pad if a whitespace is found

--round=*METHOD*

use *METHOD* for rounding when scaling; *METHOD* can be: up, down, from–zero (default), towards–zero, nearest

--suffix=*SUFFIX*

add *SUFFIX* to output numbers, and accept optional *SUFFIX* in input numbers

--to=*UNIT*

auto–scale output numbers to *UNIT*s; see *UNIT* below

--to–unit=*N*

the output unit size (instead of the default 1)

-z, --zero–terminated

line delimiter is NUL, not newline

--help display this help and exit**--version**

output version information and exit

UNIT options:

none no auto-scaling is done; suffixes will trigger an error
 auto accept optional single/two letter suffix:
 1K = 1000, 1Ki = 1024, 1M = 1000000, 1Mi = 1048576,
 si accept optional single letter suffix:
 1K = 1000, 1M = 1000000, ...
 iec accept optional single letter suffix:
 1K = 1024, 1M = 1048576, ...
 iec-i accept optional two-letter suffix:
 1Ki = 1024, 1Mi = 1048576, ...

FIELDS supports **cut(1)** style field ranges:

N N'th field, counted from 1
 N- from N'th field, to end of line
 N-M from N'th to M'th field (inclusive)
 -M from first to M'th field (inclusive)
 - all fields

Multiple fields/ranges can be separated with commas

FORMAT must be suitable for printing one floating-point argument '%f'. Optional quote (%'f) will enable **--grouping** (if supported by current locale). Optional width value (%10f) will pad output. Optional zero (%010f) width will zero pad the number. Optional negative values (%-10f) will left align. Optional precision (%.1f) will override the input determined precision.

Exit status is 0 if all input numbers were successfully converted. By default, numfmt will stop at the first conversion error with exit status 2. With **--invalid='fail'** a warning is printed for each conversion error and the exit status is 2. With **--invalid='warn'** each conversion error is diagnosed, but the exit status is 0. With **--invalid='ignore'** conversion errors are not diagnosed and the exit status is 0.

EXAMPLES

```

$ numfmt --to=si 1000
-> "1.0K"

$ numfmt --to=iec 2048
-> "2.0K"

$ numfmt --to=iec-i 4096
-> "4.0Ki"

$ echo 1K | numfmt --from=si
-> "1000"

$ echo 1K | numfmt --from=iec
-> "1024"

$ df -B1 | numfmt --header --field 2-4 --to=si
$ ls -l | numfmt --header --field 5 --to=iec
$ ls -lh | numfmt --header --field 5 --from=iec --padding=10
$ ls -lh | numfmt --header --field 5 --from=iec --format %10f
  
```

AUTHOR

Written by Assaf Gordon.

REPORTING BUGS

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>

Report numfmt translation bugs to <<https://translationproject.org/team/>>

COPYRIGHT

Copyright © 2018 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

Full documentation at: <<https://www.gnu.org/software/coreutils/numfmt>>
or available locally via: info '(coreutils) numfmt invocation'