

**NAME**

**units** - conversion calculator

**SYNOPSIS**

**units** [-ehqtUVv] [-f *unitsfile*] [-o *format*] [*from to*]

**DESCRIPTION**

The **units** program converts quantities expressed in various scales to their equivalents in other scales. It can only handle multiplicative or affine scale changes.

**units** can work interactively by prompting the user for input (see *EXAMPLES*) or non-interactively, providing a conversion for given arguments *from* and *to*.

The following options are available:

**-e, --exponential**

Same as **-o %6e** (see the description of the **-o** flag).

**-f unitsfile, --file unitsfile**

Specify the name of the units data file to load. This option may be specified multiple times.

**-H historyfile, --history historyfile**

Ignored, for compatibility with GNU units.

**-h, --help**

Show an overview of options.

**-o format, --output-format format**

Select the output format string by which numbers are printed. Defaults to "%.8g".

**-q, --quiet**

Suppress prompting of the user for units and the display of statistics about the number of units loaded.

**-t, --terse**

Only print the result. This is used when calling **units** from other programs for easy to parse results.

**-U, --unitsfile**

Print the location of the default unit file if it exists. Otherwise, print "Units data file not found".

**-V, --version**

Print the version number (which is fixed at "FreeBSD units"), the path to the units data file and exit.

**-v, --verbose**

Print the units in the conversion output. Be more verbose in general.

*from to* Allow a single unit conversion to be done directly from the command line. The program will not print prompts. It will print out the result of the single specified conversion. Both arguments, i.e., *from* and *to*, can be just a unit (e.g., "**cm**"), a quantity (e.g., "**42**"), or a quantity with a unit (e.g., "**42 cm**")

**Mathematical operators**

- Powers of units can be specified using the "^" character as shown in the example, or by simple concatenation: "**cm3**" is equivalent to "**cm^3**". See the *BUGS* section for details on the limitations of exponent values.
- Multiplication of units can be specified by using spaces (" "), a dash ("-") or an asterisk ("\*").
- Division of units is indicated by the slash ("/").
- Division of numbers must be indicated using the vertical bar ("|").

Note that multiplication has a higher precedence than division, so "**m/s/s**" is the same as "**m/s^2**" or "**m/s s**".

**Units**

The conversion information is read from a units data file. The default file includes definitions for most familiar units, abbreviations and metric prefixes. Some constants of nature included are:

pi	ratio of circumference to diameter
c	speed of light
e	charge on an electron
g	acceleration of gravity
force	same as g
mole	Avogadro's number
water	pressure per unit height of water
mercury	pressure per unit height of mercury
au	astronomical unit

The unit "**pound**" is a unit of mass. Compound names are run together so "**pound force**" is a unit of force. The unit "**ounce**" is also a unit of mass. The fluid ounce is "**floz**". British units that differ from their US counterparts are prefixed with "br", and currency is prefixed with its country name:

**"belgiumfranc", "britainpound"**. When searching for a unit, if the specified string does not appear exactly as a unit name, then **units** will try to remove a trailing "s" or a trailing "es" and check again for a match.

### Units file format

To find out what units are available read the standard units file. If you want to add your own units you can supply your own file. A unit is specified on a single line by giving its name and an equivalence. Be careful to define new units in terms of old ones so that a reduction leads to the primitive units which are marked with "!" characters. The **units** program will not detect infinite loops that could be caused by careless unit definitions. Comments in the unit definition file begin with a "#" or "/" character at the beginning of a line.

Prefixes are defined in the same way as standard units, but with a trailing dash ("-") at the end of the prefix name. If a unit is not found even after removing trailing "s" or "es", then it will be checked against the list of prefixes. Prefixes will be removed until a legal base unit is identified.

## ENVIRONMENT

**PATH** The colon-separated list of root directories at which **units** tries to find */usr/share/misc/definitions.units*.

For example if **PATH** is set to *"/tmp:/usr/local"*, no **-f** flags are provided, and */usr/share/misc/definitions.units* is missing then **units** tries to open the following files as the default units file: */tmp/usr/share/misc/definitions.units*, */usr/share/misc/definitions.units*, and */usr/local/usr/share/misc/definitions.units*.

## FILES

*/usr/share/misc/definitions.units* The standard units file.

## EXIT STATUS

The **units** utility exits 0 on success, and >0 if an error occurs.

## EXAMPLES

**Example 1:** Simple conversion of units

This example shows how to do simple conversions, for example from gigabytes to bytes:

```
$ units -o %0.f -t '4 gigabytes' bytes  
4294967296
```

The **-o %0.f** part of the command is required to print the result in a non-scientific notation (e.g,

4294967296 instead of 4.29497e+09).

### Example 2: Interactive usage

Here is an example of an interactive session where the user is prompted for units:

You have: **meters**

You want: **feet**

\* 3.2808399

/ 0.3048

You have: **cm^3**

You want: **gallons**

\* 0.00026417205

/ 3785.4118

You have: **meters/s**

You want: **furlongs/fortnight**

\* 6012.8848

/ 0.00016630952

You have: **1|2 inch**

You want: **cm**

\* 1.27

/ 0.78740157

You have: **85 degF**

You want: **degC**

29.444444

### Example 3: Difference between "|" and "/" division

The following command shows how to convert half a meter to centimeters.

```
$ units '1|2 meter' cm
```

```
* 50
```

```
/ 0.02
```

**units** prints the expected result because the division operator for numbers ("|") was used.

Using the division operator for units ("/") would result in an error:

```
$ units '1/2 meter' cm
conformability error
0.5 / m
0.01 m
```

It is because **units** interprets "**1/2 meter**" as "**0.5/meter**", which is not conformable to "**cm**".

#### Example 4: Simple units file

Here is an example of a short units file that defines some basic units:

```
m      !a!
sec     !b!
micro-  1e-6
minute  60 sec
hour    60 min
inch    0.0254 m
ft      12 inches
mile    5280 ft
```

#### Example 5: Viewing units and conversions of the default units file

The following shell one-liner allows the user to view the contents of the default units file:

```
$ less "$(units -U)"
```

## DIAGNOSTICS

**can't find units file '%s'** The default units file is not in its default location (see *FILES*) and it is not present in any file tree starting with their roots at directories from *PATH* (see *ENVIRONMENT*).

**cap\_rights\_limit() failed** See capsicum(4).

**conformability error** It is not possible to reduce the given units to one common unit: they are not conformable. Instead of a conversion, **units** will display the reduced form for each provided unit:

```
You have: ergs/hour
You want: fathoms kg^2 / day
conformability error
2.7777778e-11 kg m^2 / sec^3
2.1166667e-05 kg^2 m / sec
```

**Could not initialize history** See editline(3).

**dupstr** strdup(3) failed.

**memory for prefixes exceeded in line %d** Over 100 prefixes were defined.

**memory for units exceeded in line %d** Over 1000 prefixes were defined.

**memory overflow in unit reduction** The requested conversion involves too many units (see *BUGS*).

**redefinition of prefix '%s' on line %d ignored**

**redefinition of unit '%s' on line %d ignored**

**unexpected end of prefix on line %d**

**unexpected end of unit on line %d**

**Units data file not found** The default units file is missing.

**unable to enter capability mode** See capsicum(4).

**unable to open units file '%s'** One of the user-specified units files cannot be opened.

**unit reduces to zero**

**unknown unit '%s'** The provided unit cannot be found in the units file.

**WARNING: conversion of non-proportional quantities.** **units** may fail to convert *from* to *to* because the units are not proportional. The warning is printed when a quantity is a part of the *to* argument. It can be illustrated on an example of conversion from Fahrenheit to Celsius:

```
$ units "degF" "degC"
```

```
(-> x*0.55555556g -17.777778g)
```

```
(<- y*1.8g 32g)
```

```
$ units "degF" "1 degC"
```

```
WARNING: conversion of non-proportional quantities.
```

```
(-> x*0.55555556g -17.777778g)
```

```
(<- y*1.8g 32g)
```

```
$ units "1 degF" "1 degC"
```

WARNING: conversion of non-proportional quantities.

-17.222222

## SEE ALSO

bc(1)

## HISTORY

The **units** first appeared in NetBSD and was ported to FreeBSD 2.2.0.

The manual page was significantly rewritten in FreeBSD 13.0 by Mateusz Piotrowski <Omp@FreeBSD.org>.

## AUTHORS

Adrian Mariano <adrian@cam.cornell.edu>

## BUGS

The effect of including a "/" in a prefix is surprising.

Exponents entered by the user can be only one digit. You can work around this by multiplying several terms.

The user must use "|" to indicate division of numbers and "/" to indicate division of symbols. This distinction should not be necessary.

The program contains various arbitrary limits on the length of the units converted and on the length of the data file.

The program should use a hash table to store units so that it does not take so long to load the units list and check for duplication.

It is not possible to convert a negative value.

The **units** program does not handle reductions of long lists of units very well:

```
$ units "$ (yes m | head -n 154)" "$ (yes cm | head -n 154)"
    * 1e+308
    / 1e-308
$ units "$ (yes m | head -n 333)" "$ (yes cm | head -n 333)"
    * inf
    / 0
```

```
$ units "$(yes m | head -n 500)" "$(yes cm | head -n 500)"
```

units: memory overflow in unit reduction

conformability error

1 m^500

1 centi cm^499

```
$ units "$(yes m | head -n 501)" "$(yes cm | head -n 501)"
```

units: memory overflow in unit reduction

units: memory overflow in unit reduction

units: memory overflow in unit reduction

conformability error

1 m^500

1 centi cm^499