unitsdef – Typesetting units with $\LaTeX\,2_\varepsilon$[*]

PATRICK HAPPEL[†]

2005/01/04

Abstract

There are a lot of packages for typesetting units in $\LaTeX\,2_\varepsilon$. Some define macros to typeset a lot of units but do not suit to the actual font settings, some make the characters needed available but do not predefine any unit.

This package tries to comply with both requirements. It predefines common units, defines an easy to use interface to define new units and changes the output concerning to the surrounding font settings.

Contents

---

[*] Version v0.2

[†] patrick.happel@rub.de

Changes

v0.1

General: First public release 1

v0.11

General: Added electronvolt 1

v0.12

General: Fixed some bugs relating to gensymb. 1

v0.13

General: Some bugfixes and some units added. 1

v0.13a

General: More bugfixes. `\utimes` is now called `\unittimes` 1

v0.2

`\newnosepunit`: Added `\newnosepunit` and `\renewnosepunit` for units without any space between value and unitsign . . . 19

`\newunit`: `\newunit` now uses `\DeclareRobustCommand` instead of `\newcommand` 18

Added code for the use of xspace 18

Now unitsdef uses `\unitvaluesep` inistead of `\,`, thereby the interface is more flexible and allows the users to commit changes. 19

`\renewunit`: Added code for the use of xspace. `\DeclareRobustCommand` is used instead of `\renewcommand`. 19

`\SI`: Added `\SI` to typeset units and values obeying the SI-rules. 24

`\unitSIdef`: Added `\unitSIdef` to define units typeset by `\SI`. 24

`\unitsignonly`: Now defined by `\DeclareRobustCommand` . . . 18

Thanks

I have to thank some people who helped me by delivering suggestions, bug-reports, feature request, providing general ideas or anything else: Frank Heimann, Stephan Mann, Frank Küster, Stefan Salewski, Stephan Hennig, Klaus Wacker, Christoph Löser

1 Packages loaded by unitsdef

unitsdef loads (and so needs) the following packages with the given options:

- [T1]fontenc
- textcomp
- amsmath
- units
- xspace (as long as `noxspace`, see section 3.2, isn't specified)

The first two packages are needed for different characters in the encodings set by these packages, amsmath provides the very useful `\text` macro and units is used for typesetting the units. xpsace is included to obtain the facility to write `5\mm and some text` instead of `5\mm\ and some text`.

2 General usage

To use unitsdef simply specify

`\usepackage[`⟨*options*⟩`]{unitsdef}`

in the preamble of your document.

3 Options

3.1 Options related to gensymb

unitsdef knows three options to avoid conflicts with the gensymb-package that also predefines macros for typesetting »µ«, »Ω« and »°C«. Some macros are definied both by gensymb and unitsdef. To solve this problem three different options exist:

OHM makes unitsdef use uppercase macros (`\Ohm`, `\Celsius`, `\Micro` and `\Degree`). The predefined units with prefixes still appear lowercase (`\kiloohm`).

ohm is the opposite to `OHM` and is the default setting.

redef-gensymb This option copies `\ohm`, `\celsius`, `\micro` and `\degree` from gensymb to `\gensymbohm`, `\gensymbcelsius`, `\gensymbmicro` and `\gensymbdegree`. unitsdef then redefines this four commands so that they have the same meaning as without gensymb. **Important:** To use this option gensymb has to be loaded *before* unitsdef!

3.2 More options

unitsdef knows some other options to change its behaviour:

LITER Makes unitsdef use the *uppercase* L as the sign produced by `\liter`. This is default behaviour.

liter Makes unitsdef use the *lowercase* l as the sign produced by `\liter`.

noxspace Maybe some problems occur due to the xspace functionality of unitsdef. This option forces unitsdef to do without xspace.

noconfig Tells unitsdef not to look for a local unitsdef configuration file `unitsdef.cfg` (see section 6 for details).

noabbrv Is an abbreviation for *no abbreviations.* unitsdef does not define any short commands for a unit. This might be helpfull if another package defines a command with the same name and LaTeX exits with an error.

ugly Applies the option `ugly` to units, thus all units containing fraction will be typeset like m/s in text mode and $\frac{\mathrm{m}}{\mathrm{s}}$ in mathmode.

skipping only particular abbreviations Usually only one macro causes conflict between a definition made by unitsdef and another package. There are some options to avoid loading only particular abbreviations to avoid the definition that causes the conflict:

noamperageabbr Avoids defining the abbreviations for units of amperage (`\pA`, `\nA`, `\micA`, `\mA`, `\kA`).

nofrequncyabbr Avoids defining `\kHz`, `\MHz` and `\GHz`.

nomolabbr Avoids defining `\fmol`, `\pmol`, `\nmol`, `\micmol` and `\mmol`.

novoltageabbr Avoids defining `\kV` and `\mv`.

novolumeabbr Avoids defining `\fl`, `\pl`, `\nl`, `\micl`, `\ml`, `\cl` and `\dl`.

noweightabbr Avoids defining `\kg`, `\fg`, `\pg`, `\nanog`, `\micg` and `\mg`.

noenergyabbr Avoids defining `\kJ`, `\eV`, `\meV`, `\keV`, `\MeV`, `\GeV` and `\TeV`.

nolengthabbr Avoids defining `\picom`, `\nm`, `\micm`, `\mm`, `\cm`, `\dm` and `\km`.

notimeabbr Avoids defining `\sek`, `\fs`, `\ps`, `\ns`, `\mics` and `\ms`.

4 Typesetting units

unitsdef includes a large set of predefined units. They are listed in section 7. These units, as well as units defined by you with `\newunit` (see below), can be used as follows:

\⟨*unitmacro*⟩[⟨value⟩]

As you see, the value is an optional argument. Thus you can write `5\kg` (instead of `\kg[5]`), this may simplify the writing process. The distance in both cases is `\unitvaluesep`.

If the optional argument is not used you still can write `5\kg and some text` due to the xspace functionality of unitsdef.

`\unitvaluesep` If you want another distance between value and unit you simply have to redefine `\unitvaluesep`:

`\renewcommand{\unitvaluesep}{\hspace*{`⟨*distance*⟩`}}`

`\unitsignonly` If you want to typeset the sign of a unit without any value (for introducing a new unit-sign f.e.) `\unitvaluesep` must not be set. This is provided by the macro `\unitsignonly{\<unitmacro>}`.

Important! If you want to specify a unit with a prefix, never use something like `\milli\hour`. This will lead to problems. Always define the units first using `\newunit` or use the command `\ilu` to typeset an unit without defining it first!

4.1 Inline units

`\ilu` Some packages for typesetting units provide the possibility to write things like `\micro\meter` to obtain µm. With unitsdef this will cause some problems due to the toggling of some internal switches. If you want to use prefices without defining a new unit you have to use the command `\ilu`. It has one optional and one mandatory argument. The first contains the value to be typeset, the latter contains the unit:

```
A force of
\ilu[17]{\micro\newton}
is applied.
$F=17\ilu{\micro\newton}$
\textsf{17\ilu{\micro\newton}}
```

A force of 17 µN is applied. $F = 17\,\mu\mathrm{N}$ 17 µN

4.2 Typesetting angles and arcs

If you type `5\degree10\arcmin12\arcsec` you will get the following result: »5° 10′ 12″ «. There are full spaces between the single values and the preceding unit-symbol. This is due to the xspace functionality of unitsdef. The correct output you will obtain by using the macro `\arc`[1]. Using it in this way `\arc{5;10;12}` will lead to »5° 10′ 12″«. There are some more features of this macro, best understood if you look at this example:

```
\begin{enumerate}
  \item \arc{1}      % 1
  \item \arc{1;2}    % 2
  \item \arc{1;2;3}  % 3
  \item \arc{;2;3}   % 4
  \item \arc{;;3}    % 5
  \item \arc{1;;2}   % 6
  \item \arc{1;;}    % 7
  \item \arc{;;;}    % 8
\end{enumerate}
```

1. 1°
2. 1° 2′
3. 1° 2′ 3″
4. 0° 2′ 3″
5. 0° 0′ 3″
6. 1° 0′ 2″
7. 1°
8.

4.3 Logical markup for units

One advantage of unitsdef is that it typesets the units suiting to the surrounding font settings. But some people (magazines, universities, ...) want units always to be typeset in a special way. There is a checklist available at `http://physics.nist.gov/cuu/Units/checklist.html`. unitsdef does not fit to all these requirements by default. But there are some macros for figuring out the output. First, there is the command `\SI` which typesets (by default) all units in upright shape, but doesn't change anything else according to the font settings. The command `\SI` has two mandatory arguments, the

`\SI`

---

[1] It is expected to have (nearly?) the same usage and results as `\ang` from the SIstyle package.

first has to contain the value, the latter contains the unit:

```
\textbf{\textit{A force of
 \SI{17}{\newton}
 was applied.}}
```

A force of **17 N** ***was applied.***

If you want to customize the output of `\SI`, you can redefine the macro `\unitSIdef`. By default it contains `\upright`, but if you want units to be typeset upright in serif family, you can achieve this by redefining it to `\upshape\rmfamily`:

```
\renewcommand{\unitSIdef}
  {\upshape\rmfamily}
\textsf{\textbf{\textit{%
 A force of
 \SI{17}{\newton}
 was applied.}}}
```

A force of **17 N** ***was applied.***

5 The interface

5.1 Defining new units

`\newunit` Besides the predefined unit-macros (see section 7) unitsdef offers the possibility to define new units. This is done with the macro `\newunit`:

`\newunit{\`⟨*unitmacro*⟩`}{`⟨*unitsign*⟩`}`

To typeset a unit *Newton* (represented by `N`) you have to define[2] it as follows:

```
\newunit{\newton}{N}
```

After this you can use the macro `\newton`:

```
5\newton plus \newton[3] is 8\newton.
```

This will give the following output:

5 N plus 3 N is 8 N.

To define a unit *millinewton* (`mN`) simply define:

```
\newunit{\millinewton}{\milli\newton}
```

`\renewunit` For redefining units, the macro `\renewunit` can be used with the same syntax as `\newunit` (on the lines of `\newcommand` and `\renewcommand`).

`\newnosepunit` As an additional facility to define new units the macro `\newnosepunit` exists with the same syntax as `\newunit`. It allows you to define a unit without any space between the value and the unit-symbol. The only unit I know typeset without leading space is *degree*. You typeset 5° instead of 5 °. But maybe there are other units that must not have leading space, so I included this command in the user interface.

`\renewnosepunit` If you want to redifine a unit to be typeset without leading space, use `\renewnosepunit`.

`\per` To typeset units containing a fraction the macro `\per` is defined. The usage is:

`\per{`⟨*numerator*⟩`}{`⟨*denominator*⟩`}`

To define *newton per squaremeter* you have to say:

```
\newunit{\newtonpersmeter}{\per{\newton}{\squaremeter}}
```

`\unittimes` To typeset a multiplication sign between two units the macro `\unittimes` is used:

```
\newunit{\newtonmeter}{\newton\unittimes\meter}
\newunit{\newtonmeterpersec}{\per{\newton\unittimes\meter}{\second}}
```

`\unitsep` To typeset additional space (instead of `\unittimes`) between two units you can use `\unitsep` (do NOT use `\,`):

```
\newunit{\newtonmeter}{\newton\unitsep\meter}
```

The default is `\,` but if you want different amount of space you can simply `\renewcommand{\unitsep}{`⟨*distance*⟩`}`.

`\unitsuperscript` To typeset superscripts use `\unitsuperscript`:

```
\newunit{\squaremeter}{\meter\unitsuperscript{2}}
```

5.2 Typesetting µ, Ω, ° and ℃

`\setTextOmega`
`\setMathOmega`
`\setTextmu`
`\setMathmu`
`\setTextCelsius`
`\setMathCelsius`
`\setMathDegree`
`\setTextDegree`

Typesetting units would be quite easy without *Ohm* and *micro* (*degree* and *de-*

---

[2] You do not really have to define *newton*, it is already defined by unitsdef. This is just an example.

gree Celsius as well). The representing characters µ and Ω have to exist in three variants: One for use in math-mode (suiting to `\mathrm`) the latter two for use in `\rmfamily` and `\sffamily`. Most fonts have a suiting µ but very few fonts own a Ω. The fonts from BITSTREAM (delivered with some versions of COREL DRAW) have an italic non-suiting µ, so you have to take care of a µ when using such fonts, too. Also the µ in Latin Modern fonts isn't very good looking yet (but LM is still in beta-state). Because of nearly infinite combinations of different fonts it is impossible to choose the correct characters for µ, Ω and ℃ automatically. To define these characters use the macros `\setTextOmega`, `\setMathOmega`, `\setTextmu`, `\setMathmu`, `\setTextCelsius`, `\setMathCelsius`, `\setMathDegree` and `\setTextDegree`:

`\setTextOmega{`⟨*serif-definition*⟩`}{`⟨*sans-serif-definition*⟩`}`
`\setMathOmega{`⟨*definition*⟩`}`

`\setTextmu{`⟨*serif-definition*⟩`}{`⟨*sans-serif-definition*⟩`}`
`\setMathmu{`⟨*definition*⟩`}`

`\setTextCelsius{`⟨*serif-definition*⟩`}{`⟨*sans-serif-definition*⟩`}`
`\setMathCelsius{`⟨*definition*⟩`}`

`\setTextDegree{`⟨*serif-definition*⟩`}{`⟨*sans-serif-definition*⟩`}`
`\setMathDegree{`⟨*definition*⟩`}`

Important! The macros concerning typesetting in math-mode use amsamth's `\text`-macro. So you have to switch to math-mode to use a character out of the math alphabet. The reason for this is units. units uses `\mathrm` to typeset in math-mode. But only few mathfonts have a suitable Ω, for example:

```
\documentclass{article}

\usepackage[T1]{fontenc}
\usepackage{textcomp}
\usepackage{mathpazo}

\begin{document}
  $\Omega\qquad\mathrm{\Omega}$
\end{document}
```

unitsdef sets the defaults in a way suitable to the CM-family:

```
\setMathOmega{$\mathrm{\Omega}$}
\setMathmu{\textmu}
\setTextOmega{\textohm}{\textohm}
```

```
\setTextmu{\textmu}{\textmu}
\setMathCelsius{\textcelsius}
\setTextCelsius{\textcelsius}{\textcelsius}
\setMathDegree{\textdegree}
\setTextDegree{\textdegree}{\textdegree}
```

To find a suitable Ω for different fonts use WALTER SCHMIDT's gensymb-package[3] and read the documentation of this package.

6 Local configuration file

Since version 0.2 unitsdef by default looks for a file `unitsdef.cfg` and inputs its contents. This is useful if you frequently use a similar set of units in your documents. You put your definitions in `unitsdef.cfg` and they are included automaticly in your document[4]

A configuration file may look like this:

```
\ProvidesFile{unitsdef.cfg}%
  [2004/12/09 v1.0 some definitions for unitsdef]

\newunit{\molar}{\per{\mole}{\liter}}
\newunit{\millimolar}{\per{\millimole}{\liter}}

\endinput
```

If, by some reason, you don't want your local config file to be loaded specify the option `noconfig` in the `\usepackage` command.

7 Predefined units and prefices

Table 1: predefined preficies

name	*prefix*	*macro*	*value*	*name*	*prefix*	*macro*	*value*
yocto	y	`\yocto`	10^{-24}	zepto	z	`\zepto`	10^{-21}

---

[3] `CTAN:macros/latex/contrib/was/`

[4] Keep in mind that TeX has to find the file, so follow the instructions given by your TeX-distribution. On `*nix`-like systems and a TDS compliant distribution it might be possible to put `unitsdef.cfg` in `~/texmf/tex/latex/unitsdef/config/` and update the filename database.

Table 1: predefined preficies

name	*prefix*	*macro*	*value*	*name*	*prefix*	*macro*	*value*
atto	a	`\atto`	10^{-18}	femto	f	`\femto`	10^{-15}
pico	p	`\pico`	10^{-12}	nano	n	`\nano`	10^{-9}
micro	µ	`\micro`	10^{-6}	milli	m	`\milli`	10^{-3}
centi	c	`\centi`	10^{-2}	deci	d	`\deci`	10^{-1}
deca	da	`\deca`	10^{+1}	hecto	h	`\hecto`	10^{+2}
kilo	k	`\kilo`	10^{+3}	mega	M	`\mega`	10^{+6}
giga	G	`\giga`	10^{+9}	tera	T	`\tera`	10^{+12}
peta	P	`\peta`	10^{+15}	exa	E	`\exa`	10^{+18}
zetta	Z	`\zetta`	10^{+21}	yotta	Y	`\yotta`	10^{+24}

Table 2: predefined units

name	*sign*	*macro*	*alias*
base units			
meter	m	`\meter`	
kilogram	kg	`\kilogram`	`\kg`
mole	mol	`\mole`	
second	s	`\second`	`\sek`
ampere	A	`\ampere`	
kelvin	K	`\kelvin`	
candela	cd	`\candela`	
units of length			
picometer	pm	`\picometer`	`\picom`
nanometer	nm	`\nanometer`	`\nm`
micrometer	µm	`\micrometer`	`\micm`
millimeter	mm	`\millimeter`	`\mm`
centimeter	cm	`\centimeter`	`\cm`
decimeter	dm	`\decimeter`	`\dm`
kilometer	km	`\kilometer`	`\km`
units of weight			
gram	g	`\gram`	
femtogram	fg	`\femtogram`	`\fg`
picogram	pg	`\picoram`	`\pg`

Table 2: predefined units

name	*sign*	*macro*	*alias*
nanogram	ng	`\nanogram`	`\nanog`
microgram	µg	`\microgram`	`\micg`
milligram	mg	`\milligram`	`\mg`
units of amount of substance			
femtomole	fmol	`\femtomole`	`\fmol`
picomole	pmol	`\picomole`	`\pmol`
nanomole	nmol	`\nanomole`	`\nmol`
micromole	µmol	`\micromole`	`\micmol`
millimole	mmol	`\millimole`	`\mmol`
units of time			
attosecond	as	`\attosecond`	
femtosecond	fs	`\femtosecond`	`\fs`
pikosecond	ps	`\picosecond`	`\ps`
nanosecond	ns	`\nanosecond`	`\ns`
microsecond	µs	`\microsecond`	`\mics`
millisecond	ms	`\millisecond`	`\ms`
units of amperage			
picoampere	pA	`\picoampere`	`\pA`
nanoampere	nA	`\nanoampere`	`\nA`
microampere	µA	`\microampere`	`\micA`
milliampere	mA	`\milliampere`	`\mA`
kiloampere	kA	`\kiloampere`	`\kA`
units of volume			
liter	L	`\liter`	
femtoliter	fL	`\femtoliter`	`\fl`
picoliter	pL	`\picoliter`	`\pl`
nanoliter	nL	`\nanoliter`	`\nl`
microliter	µL	`\microliter`	`\micl`
milliliter	mL	`\milliliter`	`\ml`
centiliter	cL	`\centiliter`	`\cl`
deciliter	dL	`\deciliter`	`\dl`
hectoliter	hL	`\hectoliter`	`\hl`
cubicmeter	m^3	`\cubicmeter`	
cubicmicrometer	$\mathrm{\mu m}^3$	`\cubicmicrometer`	
cubicmillimeter	mm^3	`\cubicmillimeter`	

Table 2: predefined units

name	*sign*	*macro*	*alias*
units of area			
squaremeter	m^2	`\squaremeter`	
ar	a	`\ar`	
Hektar	ha	`\hektar`	
squarecentimeter	cm^2	`\squarecentimeter`	
squaremillimeter	mm^2	`\squaremillimeter`	
squarekilometer	km^2	`\squarekilometer`	
more units of weight			
ton	t	`\ton`	
derived units of electricity			
volt	V	`\volt`	
millivolt	mV	`\millivolt`	`\mV`
kilovolt	kV	`\kilovolt`	`\kv`
Watt	W	`\watt`	
milliwatt	mW	`\milliwatt`	
kilowatt	kW	`\kilowatt`	
megawatt	MW	`\megawatt`	
coulomb	C	`\coulomb`	
ohm	Ω	`\ohm` or `\Ohm`	
kiloohm	kΩ	`\kiloohm`	
megaohm	MΩ	`\megaohm`	
gigaohm	GΩ	`\gigaohm`	
siemens	S	`\siemens`	
millisiemens	mS	`\millisiemens`	
farad	F	`\farad`	
femtofarad	fF	`\femtofarad`	
picofarad	pF	`\picofarad`	
nanofarad	nF	`\nanofarad`	
microfarad	µF	`\microfarad`	
millifarad	mF	`\millifarad`	
units of energy			
joule	J	`\joule`	
millijoule	mJ	`\millijoule`	
kilojoule	kJ	`\kilojoule`	`\kJ`
megajoule	MJ	`\megajoule`	

Table 2: predefined units

name	*sign*	*macro*	*alias*
calorie	cal	`\calory`	
kilocalorie	kcal	`\kilocalory`	
electronvolt	eV	`\electronvolt`	`\eV`
millielectronvolt	meV	`\millielectronvolt`	`\meV`
kiloelectronvolt	keV	`\kiloelectronvolt`	`\keV`
megaelectronvolt	MeV	`\megaelectronvolt`	`\MeV`
gigaelectronvolt	GeV	`\gigaelectronvolt`	`\GeV`
teraelectronvolt	TeV	`\teraelectronvolt`	`\TeV`
more units of time			
minute	min	`\minute`	
hour	h	`\hour`	
days	d	`\days`	
units of temperature			
degree Celsius	℃	`\celsius` or `\Celsius`	
units of angle			
radian	rad	`\radian`	
steradian	sr	`\steradian`	
degree	$^\circ$	`\degree` or `\Degree`	
arc minute	$'$	`\arcmin`	
arc second	$''$	`\arcsec`	
units of frequencies			
hertz	Hz	`\hertz`	
kilohertz	kHz	`\kilohertz`	`\kHz`
megahertz	MHz	`\megahertz`	`\MHz`
gigahertz	GHz	`\gigahertz`	`\GHz`
units of force			
newton	N	`\newton`	
millinewton	mN	`\millinewton`	
kilonewton	kN	`\kilonewton`	
units of pressure			
pascal	Pa	`\pascal`	
hectopascal	hPa	`\hectopascal`	
bar	bar	`\uBar`	
millibar	mbar	`\millibar`	

Table 2: predefined units

name	*sign*	*macro*	*alias*
units of magnetism			
weber	Wb	`\weber`	
tesla	T	`\tesla`	
henry	H	`\henry`	
units of light			
lumen	lm	`\lumen`	
lux	lx	`\lux`	
units of radioactivity			
becquerel	Bq	`\becquerel`	
megabecquerel	MBq	`\megabecquerel`	
curie	Cu	`\curie`	
sievert	Sv	`\sievert`	
millisievert	mSv	`\millisievert`	
percent			
percent	%	`\percent`	

8 To Do

There are a lot of things to be done. Some are mentioned here:

- Add some more macros to figure out the ouptut (something similar to `\SI`). Provide the possibility to typeset all units in math mode.
- Improve the documentation.
- Adding interfaces units for Å, ‰ and °F

9 Code

```
⟨*package⟩
```

9.1 switches

`\if@@setunitsep` This switch is used internally to decide, whether a distance is to be typesetted or not.

```
\newif\if@@setunitsep
\@@setunitseptrue
```

`\ifunit@@Ohm` This switch decides whether `\ohm` or `\Ohm` and `\celsius` or `\Celsius` is defined (according to gensymb).

```
\newif\ifunit@@Ohm%
\unit@@Ohmfalse
```

`\ifunit@@redefgensymb` This switch makes gensymb's macros `\ohm` and `\celsius` available as `\gensymbohm` and `\gensymbcelsius`. unitsdef uses `\ohm` and `\celsius`.

```
\newif\ifunit@@redefgensymb
\unit@@redefgensymbfalse
```

`\ifunit@@liter` This switch decides whether the output of the unit `\iter` is typeset as `l` or `L`

```
\newif\ifunit@@liter
\unit@@literfalse
```

`\ifunit@@xspace` This switch decides whether xspace is used.

```
\newif\ifunit@@xspace
\unit@@xspacetrue
```

`\ifunit@@xspace` This switch decides whether abbreviattions are loaded.

```
\newif\ifunit@@useabbrv
\unit@@useabbrvtrue
```

The follwing switches decide whether to load a `.cfg` file containing some abbreviations or not.

```
\newif\ifunit@@useampabbrv
\unit@@useampabbrvtrue
\newif\ifunit@@usefreqabbrv
\unit@@usefreqabbrvtrue
\newif\ifunit@@usemolabbrv
\unit@@usemolabbrvtrue
\newif\ifunit@@usevoltabbrv
\unit@@usevoltabbrvtrue
\newif\ifunit@@usevolabbrv
\unit@@usevolabbrvtrue
\newif\ifunit@@useweightabbrv
\unit@@useweightabbrvtrue
\newif\ifunit@@useenergyabbrv
\unit@@useenergyabbrvtrue
\newif\ifunit@@uselengthabbrv
\unit@@uselengthabbrvtrue
\newif\ifunit@@usetimeabbrv
\unit@@usetimeabbrvtrue
```

`\ifunit@@useconfigfile` This switch stores whether the option `noconfig` is given.

```
\newif\ifunit@@useconfigfile
\unit@@useconfigfiletrue
```

9.2 Options

```
\DeclareOption{OHM}{\unit@@Ohmtrue}
\DeclareOption{ohm}{\unit@@Ohmfalse}
\DeclareOption{redef-gensymb}{%
  \@ifpackageloaded{gensymb}{%
    \unit@@redefgensymbtrue%
    \AtBeginDocument{%
      \let\gensymbohm\ohm%
      \let\gensymbcelsius\celsius%
      \let\gensymbmicro\micro%
      \renewunit{\ohm}{\unitOmega}%
      \renewunit{\celsius}{\unitCelsius}%
      \renewcommand{\micro}{\unitmu\@@setunitsepfalse}%
    }
  }{%
    \PackageError{unitsdef}{
      You requestet me to save some macros from the\MessageBreak
      gensymb-package. This package is  not loaded.\MessageBreak
      If you load it later, tell me to use  uppercase\MessageBreak
      macronames where conflicts appear by giving me\MessageBreak
      the option OHM.
    }
  }
}
\DeclareOption{liter}{\unit@@litertrue}
\DeclareOption{LITER}{\unit@@literfalse}
\DeclareOption{noxspace}{\unit@@xspacefalse}
\DeclareOption{noabbrv}{\unit@@useabbrvfalse}
\DeclareOption{ugly}{\PassOptionsToPackage{ugly}{units}}
\DeclareOption{noamperageabbr}{\unit@@useampabbrvfalse}
\DeclareOption{nofrequncyabbr}{\unit@@usefreqabbrvfalse}
\DeclareOption{nomolabbr}{\unit@@usemolabbrvfalse}
\DeclareOption{novoltageabbr}{\unit@@usevoltabbrvfalse}
\DeclareOption{novolumeabbr}{\unit@@usevolabbrvfalse}
\DeclareOption{noweightabbr}{\unit@@useweightabbrvfalse}
\DeclareOption{noenergyabbr}{\unit@@useenergyabbrvfalse}
\DeclareOption{nolengthabbr}{\unit@@uselengthabbrvfalse}
\DeclareOption{notimeabbr}{\unit@@usetimeabbrvfalse}
\DeclareOption{noconfig}{\unit@@useconfigfilefalse}
```

```
\ProcessOptions
```

9.3 Loading packages

```
\RequirePackage[T1]{fontenc}
\RequirePackage{amsmath}
\RequirePackage{textcomp}
\RequirePackage{units}
\ifunit@@xspace
  \RequirePackage{xspace}%
  \let\unit@@xspace\xspace%
\else
  \let\unit@@xspace\relax
\fi
```

9.4 The interface

`\unitsignonly` This macro is used to typeset a unit without leading spacing. To achieve this `\@@setunitsep` is set to `false`. `\unitsignonly` has an mandatory argument containig the unit to typeset: `\unitsignonly{`⟨*unit*⟩`}`.

```
\DeclareRobustCommand{\unitsignonly}[1]{%
  \@@setunitsepfalse%
  \begingroup%
  \let\unit@@xspace\relax%
  #1%
  \endgroup}
```

`\unitvaluesep` This Macro specifies the distance between value and unit, default value is `\,`.

```
\newcommand{\unitvaluesep}{}
\let\unitvaluesep\,%
```

`\newunit` This macro is the interface to define new units. Usage is:

> `\newunit{`\⟨*unitmacro*⟩`}{`⟨*unitsign*⟩`}`.

```
\newcommand{\newunit}[2]{%
```

First there is a check whether the macro already exists:

```
    \newcommand{#1}{}%
```

then its definition is performed.

```
    \DeclareRobustCommand{#1}[1][]{##1%
```

The redefinition of `\unitvaluesep` has to stay local:

```
      \begingroup%
```

`\xpsace` must not do anything when invoked inside a unit. So it is set to `\relax` inside this group.

```
       \let\unit@@xspace\relax%
       \if@@setunitsep%
```

```
         \unitvaluesep%
```

Once a spacing is typeset there must no further spacing be typeset. To avoid typesetting more spacings I redefine `\unitvaluesep` to `\relax`.

```
         \let\unitvaluesep\relax%
       \fi%
       \unit{#2}\global\@@setunitseptrue%
      \endgroup%
```

Now `\xspace` has to be invoked, as long as `noxspace` is not set.

```
      \unit@@xspace%
    }%
}
```

`\renewunit`

This macro is to redifine existing units.

```
\newcommand{\renewunit}[2]{%
    \renewcommand{#1}{}%
    \DeclareRobustCommand{#1}[1][]{##1%
      \begingroup%
       \let\unit@@xspace\relax%
       \if@@setunitsep%
         \unitvaluesep%
         \let\,\relax%
       \fi%
       \unit{#2}\global\@@setunitseptrue%
      \endgroup%
      \unit@@xspace%
    }%
}

```

`\newnosepunit`

```
\newcommand{\newnosepunit}[2]{%
    \newcommand{#1}{}%
    \DeclareRobustCommand{#1}[1][]{##1%
      \begingroup%
       \let\xspace\relax%
       \if@@setunitsep%
         \let\,\relax%
       \fi%
       \unit{#2}\global\@@setunitseptrue%
      \endgroup%
      \unit@@xspace%
    }%
}
\newcommand{\renewnosepunit}[2]{%
    \renewcommand{#1}{}%
    \DeclareRobustCommand{#1}[1][]{##1%
      \begingroup%
       \let\unit@@xspace\relax%
       \if@@setunitsep%
         \let\,\relax%
       \fi%
       \unit{#2}\global\@@setunitseptrue%
      \endgroup%
      \unit@@xspace%
    }%
}
```

`\per`

```
\newcommand\per[2]{%
  \@@setunitsepfalse%
  \unitfrac{#1}{#2}%
}
```

`\ilu` The command `\ilu` provides the possibility to typeset inline-units that are not defined by a previous `\newunit` command.

```
\newcommand{\ilu}[2][]{%
  \begingroup%
    \@@setunitsepfalse%
    \let\unit@@xspace\relax%
    #1\,\unit{#2}%
  \endgroup%
}
```

`\unittimes`

```
\newcommand{\unittimes}{\@@setunitsepfalse\ensuremath{\cdot}}
```

`\unitsep`

```
\let\unitsep\,
```

`\unitsuperscript`

```
\newcommand{\unitsuperscript}[1]{%
  \ifmmode^{#1}\else\textsuperscript{#1}\fi%
}
```

\unitMathOmega

```
\newcommand{\unitMathOmega}{}
```

\unitTextOmega

```
\newcommand{\unitTextOmega}{}
```

\unittextmu

```
\newcommand{\unitTextmu}{}
```

\unitmathmu

```
\newcommand{\unitMathmu}{}
```

\unitMathCelsius

```
\newcommand{\unitMathCelsius}{}
```

\unitTextCelsius

```
\newcommand{\unitTextCelsius}{}
```

\unitTextDegree

```
\newcommand{\unitTextDegree}{}
```

\unitMathDegree

```
\newcommand{\unitMathDegree}{}
```

\unitCelsius

```
\newcommand{\unitCelsius}{%
  \ifmmode\unitMathCelsius\else\unitTextCelsius\fi%
}
```

\unitmu

```
\newcommand{\unitmu}{%
  \ifmmode\unitMathmu\else\unitTextmu\fi%
}
```

\unitOmega

```
\newcommand{\unitOmega}{%
  \ifmmode\unitMathOmega\else\unitTextOmega\fi%
}
```

\unitDegree

```
\newcommand{\unitDegree}{%
  \ifmmode\unitMathDegree\else\unitTextDegree\fi%
}
```

`\setMathOmega`

```
\newcommand{\setMathOmega}[1]{%
  \renewcommand{\unitMathOmega}{\text{#1}}%
}
\setMathOmega{$\mathrm{\Omega}$}
```

`\setMathmu`

```
\newcommand{\setMathmu}[1]{%
  \renewcommand{\unitMathmu}{\text{#1}}%
}
\setMathmu{\textmu}
```

`\setMathCelsius`

```
\newcommand{\setMathCelsius}[1]{%
  \renewcommand{\unitMathCelsius}{\text{#1}}%
}
\setMathCelsius{\textcelsius}
```

`\setMathDegree`

```
\newcommand{\setMathDegree}[1]{%
  \renewcommand{\unitMathDegree}{\text{#1}}%
}
\setMathDegree{\textdegree}
```

`\setTextOmega` This macro is to define the »Ω« that is used in text mode. The first argument is used when `\rmfamily` is active, the latter when `\sffamily` is active.

```
\newcommand{\setTextOmega}[2]{%
  \renewcommand{\unitTextOmega}{%
    \begingroup%
      \edef\@tempa{\sfdefault}%
      \ifx\f@family\@tempa%
        #2%
      \else%
        #1%
      \fi%
    \endgroup%
  }%
}
\setTextOmega{\textohm}{\textohm}
```

`\setTextmu` This macro is to define the »µ« that is used in text mode. The first argument is used when `\rmfamily` is active, the latter when `\sffamily` is active.

```
\newcommand{\setTextmu}[2]{%
```

```
  \renewcommand{\unitTextmu}{%
    \begingroup%
      \edef\@tempa{\sfdefault}%
      \ifx\f@family\@tempa%
        #2%
      \else%
        #1%
      \fi%
    \endgroup%
  }%
}
\setTextmu{\textmu}{\textmu}
```

`\setTextCelsius` This macro is to define the »℃« that is used in text mode. The first argument is used when `\rmfamily` is active, the latter when `\sffamily` is active.

```
\newcommand{\setTextCelsius}[2]{%
  \renewcommand{\unitTextCelsius}{%
    \begingroup%
      \edef\@tempa{\sfdefault}%
      \ifx\f@family\@tempa%
        #2%
      \else%
        #1%
      \fi%
    \endgroup%
  }%
}
\setTextCelsius{\textcelsius}{\textcelsius}
```

`\setTextDegree` This macro is to define the »°« that is used in text mode. The first argument is used when `\rmfamily` is active, the latter when `\sffamily` is active.

```
\newcommand{\setTextDegree}[2]{%
  \renewcommand{\unitTextDegree}{%
    \begingroup%
      \edef\@tempa{\sfdefault}%
      \ifx\f@family\@tempa%
        #2%
      \else%
        #1%
      \fi%
    \endgroup%
  }%
}
\setTextDegree{\textdegree}{\textdegree}
```

\unitSIdef

```
\newcommand\unitSIdef{\upshape}
```

\SI

```
\newcommand{\SI}{}
\DeclareRobustCommand{\SI}[2]{%
  \begingroup%
    \let\unit@@xspace\relax%
    \unitSIdef\selectfont%
      #1#2%
  \endgroup%
}
```

9.5 Definition of prefices

```
\newcommand{\yocto}{y\@@setunitsepfalse}            % -24
\newcommand{\zepto}{z\@@setunitsepfalse}            % -21
\newcommand{\atto}{a\@@setunitsepfalse}             % -18
\newcommand{\femto}{f\@@setunitsepfalse}            % -15
\newcommand{\pico}{p\@@setunitsepfalse}             % -12
\newcommand{\nano}{n\@@setunitsepfalse}             %  -9
\ifunit@@Ohm
  \newcommand{\Micro}{\unitmu\@@setunitsepfalse}
  \let\@unit@micro\Micro
\else
  \ifunit@@redefgensymb\else
    \newcommand{\micro}{\unitmu\@@setunitsepfalse}
  \let\@unit@micro\micro
  \fi
\fi
\newcommand{\milli}{m\@@setunitsepfalse}            %  -3
\newcommand{\centi}{c\@@setunitsepfalse}            %  -2
\newcommand{\deci}{d\@@setunitsepfalse}             %  -1

\newcommand{\deca}{da\@@setunitsepfalse}            %  +1
\newcommand{\hecto}{h\@@setunitsepfalse}            %  +2

\newcommand{\kilo}{k\@@setunitsepfalse}             %  +3
\newcommand{\mega}{M\@@setunitsepfalse}             %  +6
\newcommand{\giga}{G\@@setunitsepfalse}             %  +9
\newcommand{\tera}{T\@@setunitsepfalse}             % +12
\newcommand{\peta}{P\@@setunitsepfalse}             % +15
\newcommand{\exa}{E\@@setunitsepfalse}              % +18
\newcommand{\zetta}{Z\@@setunitsepfalse}            % +21
```

```
\newcommand{\yotta}{Y\@@setunitsepfalse}            % +24

```

9.6 Definitions of units

9.6.1 base units

```
\newunit{\meter}{m}
    \newunit{\gram}{g}
\newunit{\kilogram}{\kilo\gram}
\newunit{\mole}{mol}
\newunit{\second}{s}
\newunit{\ampere}{A}
\newunit{\kelvin}{K}
\newunit{\candela}{cd}

```

9.6.2 Units of length

```

\newunit{\picometer}{\pico\meter}
\newunit{\nanometer}{\nano\meter}
\newunit{\micrometer}{\@unit@micro\meter}
\newunit{\millimeter}{\milli\meter}
\newunit{\centimeter}{\centi\meter}
\newunit{\decimeter}{\deci\meter}
\newunit{\kilometer}{\kilo\meter}

```

9.6.3 Units of weight

```

\newunit{\femtogram}{\femto\gram}
\newunit{\picogram}{\pico\gram}
\newunit{\nanogram}{\nano\gram}
\newunit{\microgram}{\@unit@micro\gram}
\newunit{\milligram}{\milli\gram}

```

9.6.4 Units of quantity

```

\newunit{\femtomole}{\femto\mole}
\newunit{\picomole}{\pico\mole}
\newunit{\nanomole}{\nano\mole}
\newunit{\micromole}{\@unit@micro\mole}
\newunit{\millimole}{\milli\mole}

```

9.6.5 Units of time

```
\newunit{\attosecond}{\atto\sek}
\newunit{\femtosecond}{\femto\sek}
\newunit{\picosecond}{\pico\sek}
\newunit{\nanosecond}{\nano\sek}
\newunit{\microsecond}{\@unit@micro\sek}
\newunit{\millisecond}{\milli\sek}
```

9.6.6 amperage

```
\newunit{\picoampere}{\pico\ampere}
\newunit{\nanoampere}{\nano\ampere}
\newunit{\microampere}{\@unit@micro\ampere}
\newunit{\milliampere}{\milli\ampere}
\newunit{\kiloampere}{\kilo\ampere}
```

9.6.7 Percent

```
\newunit{\percent}{\%}
```

9.6.8 Volumes

```
\ifunit@@liter
  \newunit{\liter}{l}
\else
  \newunit{\liter}{L}
\fi

\newunit{\femtoliter}{\femto\liter}
\newunit{\picoliter}{\pico\liter}
\newunit{\nanoliter}{\nano\liter}
\newunit{\microliter}{\@unit@micro\liter}
\newunit{\milliliter}{\milli\liter}
\newunit{\centiliter}{\centi\liter}
\newunit{\deciliter}{\deci\liter}
\newunit{\hectoliter}{\hecto\liter}
```

```
\newunit{\cubicmeter}{\meter\unitsuperscript{3}}
\newunit{\cubicmicrometer}{\micrometer\unitsuperscript{3}}
\newunit{\cubicmillimeter}{\millimeter\unitsuperscript{3}}


```

9.6.9 Areas

```

\newunit{\squaremeter}{\meter\unitsuperscript{2}}

\newunit{\ar}{a}
\newunit{\hektar}{\hecto\ar}

\newunit{\squarecentimeter}{\centimeter\unitsuperscript{2}}
\newunit{\squaremillimeter}{\millimeter\unitsuperscript{2}}
\newunit{\squarekilometer}{\kilometer\unitsuperscript{2}}
```

9.6.10 more units of weight

```

\newunit{\ton}{t}

```

9.6.11 Derived electrical units

```

\newunit{\volt}{V}
\newunit{\millivolt}{\milli\volt}

\newunit{\kilovolt}{\kilo\volt}

\newunit{\watt}{W}
\newunit{\milliwatt}{\milli\watt}
\newunit{\kilowatt}{\kilo\watt}
\newunit{\megawatt}{\mega\watt}

\newunit{\coulomb}{C}
```

Don’t forget the options when typesetting Ω!

```
\ifunit@@Ohm
  \newunit{\Ohm}{\unitOmega}
  \newunit{\kiloohm}{\kilo\Ohm}
  \newunit{\megaohm}{\mega\Ohm}
  \newunit{\gigaohm}{\giga\Ohm}
\else
  \ifunit@@redefgensymb\else
    \newunit{\ohm}{\unitOmega}
  \fi
  \newunit{\kiloohm}{\kilo\ohm}
  \newunit{\megaohm}{\mega\ohm}
  \newunit{\gigaohm}{\giga\ohm}
\fi
\newunit{\siemens}{S}
\newunit{\millisiemens}{\milli\siemens}
\newunit{\farad}{F}
\newunit{\femtofarad}{\femto\farad}
\newunit{\picofarad}{\pico\farad}
\newunit{\nanofarad}{\nano\farad}
\newunit{\microfarad}{\@unit@micro\farad}
\newunit{\millifarad}{\milli\farad}
```

9.6.12 Units of energy

```
\newunit{\joule}{J}
\newunit{\millijoule}{\milli\joule}
\newunit{\kilojoule}{\kilo\joule}
\newunit{\megajoule}{\mega\joule}

\newunit{\calory}{cal}
\newunit{\kilocalory}{\kilo\calory}
\newunit{\electronvolt}{eV}
\newunit{\millielectronvolt}{\milli\eV}
\newunit{\kiloelectronvolt}{\kilo\eV}
\newunit{\megaelectronvolt}{\mega\eV}
\newunit{\gigaelectronvolt}{\giga\eV}
\newunit{\teraelectronvolt}{\tera\eV}

```

9.6.13 more units of time

```
\newunit{\minute}{min}
\newunit{\hour}{h}
\newunit{\days}{d}
```

9.6.14 more units of temperature

Don't forget `gensymb` when typesetting degree Celsius.

```
\ifunit@@Ohm
  \newunit{\Celsius}{\unitCelsius}
\else
  \ifunit@@redefgensymb\else
    \newunit{\celsius}{\unitCelsius}
  \fi
\fi
```

9.6.15 Angles and arcs

```
\newunit{\radian}{rad}
\newunit{\steradian}{sr}
\ifunit@@Ohm
  \newnosepunit{\Degree}{\unitDegree}
\else
  \ifunit@@redefgensymb\else
    \newnosepunit{\degree}{\unitDegree}
  \fi
\fi

\newunit{\arcmin}{\ensuremath{{}^\prime}}
\newunit{\arcsec}{\ensuremath{{}^{\prime\prime}}}
```

Some TEX to realize the syntax using `;` as separator.

```
\DeclareRobustCommand{\arc}[1]{\expandafter\unit@arc#1;;!}

\def\unit@arc#1;#2;#3!{%
   \ifx\\#1\\\def\unit@arcdegreevalue{0}%
   \else\def\unit@arcdegreevalue{#1}\fi%
   \ifx\\#2\\\def\unit@arcminvalue{0}%
   \else\def\unit@arcminvalue{#2}\fi%
   \ifx\\#3\\\def\unit@arcsecvalue{0}%
   \else\edef\unit@arcsecvalue{\expandafter\unit@strip#3;!}\fi%
   \begingroup%
     \let\unit@@xspace\relax%
     \ifnum\unit@arcsecvalue=0\relax%
       \ifnum\unit@arcminvalue=0\relax%
         \ifnum\unit@arcdegreevalue=0\relax\else%
           \unit@arcdegreevalue\degree%
         \fi%
       \else%
         \unit@arcdegreevalue\degree%
         \unitvaluesep%
         \unit@arcminvalue\arcmin%
       \fi%
     \else%
       \unit@arcdegreevalue\degree%
       \unitvaluesep%
       \unit@arcminvalue\arcmin%
       \unitvaluesep%
       \unit@arcsecvalue\arcsec%
     \fi%
   \endgroup%
}

\def\unit@strip#1;#2!{%
  \ifx\\#1\\0\else#1\fi%
}

```

9.6.16 Frequencies

```
\newunit{\hertz}{Hz}
  \newunit{\kilohertz}{\kilo\hertz}
  \newunit{\megahertz}{\mega\hertz}
  \newunit{\gigahertz}{\giga\hertz}

```

9.6.17 Force

```
\newunit{\newton}{N}
  \newunit{\millinewton}{\milli\newton}
  \newunit{\kilonewton}{\kilo\newton}
```

9.6.18 Pressure

```
\newunit{\pascal}{Pa}
  \newunit{\hectopascal}{\hecto\pascal}
\newunit{\uBar}{bar}
  \newunit{\millibar}{\milli\uBar}
```

9.6.19 magnetic field strength

```
\newunit{\weber}{Wb}
```

9.6.20 magnetic flux density

```
\newunit{\tesla}{T}
```

9.6.21 Induction

```
\newunit{\henry}{H}
```

9.6.22 Lumen

```
\newunit{\lumen}{lm}
```

9.6.23 Illuminance

```
\newunit{\lux}{lx}
```

9.6.24 Radioactivity

```
\newunit{\becquerel}{Bq}
  \newunit{\megabecquerel}{\mega\becquerel}
\newunit{\curie}{Cu}
```

9.6.25 Sievert

```
\newunit{\sievert}{Sv}
  \newunit{\millisievert}{\milli\sievert}
```

9.7 Loading abbreviations

Now the abbreviations are loaded if no option is specified to withhold some abbreviations.

```
\ifunit@@useabbrv
  \ifunit@@useampabbrv
     \InputIfFileExists{ampabbrv.cfg}%
       {\PackageInfo{unitsdef}{Abbreviations for units of amperage loaded.}}%
       {\PackageWarning{unitsdef}{ampabbrv.cfg not found!}}%
  \fi
  \ifunit@@usefreqabbrv
     \InputIfFileExists{freqabbr.cfg}%
       {\PackageInfo{unitsdef}{Abbreviations for units of frequency loaded.}}%
       {\PackageWarning{unitsdef}{freqabbr.cfg not found!}}%
  \fi
  \ifunit@@usemolabbrv
     \InputIfFileExists{molabbrv.cfg}%
       {\PackageInfo{unitsdef}{Abbreviations for units of amount of substances loaded.}}%
       {\PackageWarning{unitsdef}{molabbrv.cfg not found!}}%
  \fi
  \ifunit@@usevoltabbrv
    \InputIfFileExists{voltabbr.cfg}%
      {\PackageInfo{unitsdef}{Abbreviations for units of voltage loaded.}}%
      {\PackageWarning{unitsdef}{voltabbr.cfg not found!}}%
  \fi
  \ifunit@@usevolabbrv
    \InputIfFileExists{volabbrv.cfg}%
      {\PackageInfo{unitsdef}{Abbreviations for units of volume loaded.}}%
      {\PackageWarning{unitsdef}{volabbrv.cfg not found!}}%
  \fi
  \ifunit@@useweightabbrv
    \InputIfFileExists{weigabbr.cfg}%
      {\PackageInfo{unitsdef}{Abbreviations for units of weight loaded.}}%
      {\PackageWarning{unitsdef}{weigabbr.cfg not found!}}%
  \fi
  \ifunit@@useenergyabbrv
    \InputIfFileExists{enerabbr.cfg}%
      {\PackageInfo{unitsdef}{Abbreviations for units of energy loaded.}}%
      {\PackageWarning{unitsdef}{enerabbr.cfg not found!}}%
  \fi
  \ifunit@@uselengthabbrv
    \InputIfFileExists{lengabbr.cfg}%
      {\PackageInfo{unitsdef}{Abbreviations for units of length loaded.}}%
      {\PackageWarning{unitsdef}{lengabbr.cfg not found!}}%
  \fi
  \ifunit@@usetimeabbrv
    \InputIfFileExists{timeabbr.cfg}%
      {\PackageInfo{unitsdef}{Abbreviations for units of time loaded.}}%
      {\PackageWarning{unitsdef}{timeabbr.cfg not found!}}%
  \fi
\fi
\ifunit@@useconfigfile
  \InputIfFileExists{unitsdef.cfg}%
    {\PackageInfo{unitsdef}{Local config file loaded.}}%
    {\PackageInfo{unitsdef}{No local config file found.}}%
\else
  \PackageInfo{unitsdef}{Local config file not loaded.}
\fi
%
⟨/package⟩
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.