1 Numbers in tables 20
  1.1 Key–value options ........................................... 20

IX   siunitx-unit – Parsing and formatting units 22
  1 Formatting units 22
  2 Defining symbolic units 24
  3 Per-unit options 25
  4 Units in (PDF) strings 25
  5 Pre-defined symbolic unit components 25
      5.1 Key–value options ........................................... 28

X   siunitx-abbreviations – Abbreviations 31

XI  siunitx-binary – Binary units 34

XII  siunitx-command – Units as document command 35
    1 Creating units as document commands 35
        1.1 Key–value options ........................................... 35

XIII siunitx-emulation – Emulation 36

Index 37
Part I

\textbf{siunitx-angle} – Formatting angles

1 Formatting angles

\begin{verbatim}
\siunitx_angle:n \siunitx_angle:n \{\langle\text{angle}\rangle\}
\siunitx_angle:nnn \{\langle\text{degrees}\rangle\} \{\langle\text{minutes}\rangle\} \{\langle\text{seconds}\rangle\}
\end{verbatim}

Typeset the \langle\text{angle}\rangle\) (which may be given as separate \langle\text{degree}\rangle, \langle\text{minute}\rangle and \langle\text{second}\rangle components). The \langle\text{angle}\rangle\) (or components) may be given as expressions. The \langle\text{angle}\rangle\) should be a number as understood by \texttt{\siunitx\_format\_number:n\_\_\_}, with no uncertainty, exponent or imaginary part. The unit symbols for degrees, minutes and seconds are \texttt{\textdegree}, \texttt{\arcminute} and \texttt{\arcsecond}, respectively.

1.1 Key–value options

The options defined by this submodule are available within the l3keys siunitx tree.

\begin{verbatim}
\texttt{\textbf{angle-mode}} \quad \texttt{\textbf{angle-mode} = \langle\text{choice}\rangle}
\end{verbatim}

Selects how angles are formatted: a choice from the options \texttt{arc}, \texttt{decimal} and \texttt{input}. The option \texttt{arc} means that angles will always be typeset in arc (degree, minute, second) format, whilst \texttt{decimal} means that angles are typeset as a single decimal value. The \texttt{input} setting means that the input format (\textit{i.e.} difference between \texttt{\siunitx\_angle:n} and \texttt{\siunitx\_angle:nnn}) is maintained. The standard setting is \texttt{input}.

\begin{verbatim}
\texttt{\textbf{angle-symbol-degree}} \quad \texttt{\textbf{angle-symbol-degree} = \langle\text{symbol}\rangle}
\texttt{\textbf{angle-symbol-minute}} \quad \texttt{\textbf{angle-symbol-minute}} \quad \texttt{\textbf{angle-symbol-second}} \quad \texttt{\textbf{angle-symbol-second}}
\end{verbatim}

Sets the symbol used for arc degrees, minutes or seconds, respectively.

\begin{verbatim}
\texttt{\textbf{angle-symbol-over-decimal}} \quad \texttt{\textbf{angle-symbol-over-decimal} = true|false}
\end{verbatim}

Determines if the arc separator is printed over the decimal marker, a format used in astronomy. The standard setting is \texttt{false}.

\begin{verbatim}
\texttt{\textbf{arc-separator}} \quad \texttt{\textbf{arc-separator} = \langle\text{separator}\rangle}
\end{verbatim}

Inserted between arc parts (degree, minute and second components). The standard setting is \texttt{\textbackslash,}.

\begin{verbatim}
\texttt{\textbf{fill-angle-degrees}} \quad \texttt{\textbf{fill-arc-degrees} = true|false}
\end{verbatim}

Determines whether a missing degrees part is zero-filled when printing an arc. The standard setting is \texttt{false}.

\begin{verbatim}
\texttt{\textbf{fill-angle-minutes}} \quad \texttt{\textbf{fill-arc-minutes} = true|false}
\end{verbatim}

Determines whether a missing minutes part is zero-filled when printing an arc. The standard setting is \texttt{false}.
fill-arc-seconds  fill-arc-seconds = true|false

Determines whether a missing seconds part is zero-filled when printing an arc. The standard setting is false.

number-angle-product  number-angle-product = ⟨separator⟩

Inserted between the value of an angle and the unit (degree, minute or second component). The standard setting is \,. 
Part II

\texttt{siunitx-compound} – Compound numbers and quantities

\begin{itemize}
\item \texttt{\textbackslash siunitx\_compound\_number:n}
\item \texttt{\textbackslash siunitx\_compound\_quantity:nn}
\item \texttt{\textbackslash siunitx\_number\_list:nn}
\item \texttt{\textbackslash siunitx\_quantity\_list:nn}
\item \texttt{\textbackslash siunitx\_number\_product:n}
\item \texttt{\textbackslash siunitx\_quantity\_product:nn}
\item \texttt{\textbackslash siunitx\_number\_range:nn}
\item \texttt{\textbackslash siunitx\_quantity\_range:nnn}
\end{itemize}

Separators for lists of numbers and quantities.

\texttt{\l\_siunitx\_list\_separator\_pair\_tl}
\texttt{\l\_siunitx\_list\_separator\_tl}
\texttt{\l\_siunitx\_list\_separator\_final\_tl}
Phrase (or similar) used between limits of a range.

\l_siunitx_range_phrase_tl

\l_compound_exponents
compound-exponents = combine|combine-bracket|individual

\l_compound_final_separator
compound-final-separator = (text)

\l_compound_pair_separator
compound-pair-separator = (text)

\l_compound_separator
compound-separator = (text)

\l_compound_separator_mode
compound-separator-mode = number|text

\l_compound_units
compound-units = bracket|repeat|single

\l_list_exponents
list-exponents = combine|combine-bracket|individual

\l_list_final_separator
list-final-separator = (text)

\l_list_pair_separator
list-pair-separator = (text)

\l_list_separator
list-separator = (text)

\l_list_units
list-units = bracket|repeat|single

\l_product_exponents
product-exponents = combine|combine-bracket|individual

\l_product_mode
product-mode = phrase|choice

\l_product_phrase
product-phrase = (text)

\l_product_symbol
product-symbol = (symbol)

\l_range_exponents
range-exponents = combine|combine-bracket|individual

\l_range_phrase
range-phrase = (text)

\l_range_units
range-units = bracket|repeat|single
Part III

siunitx-locale – Localisation

This submodule is concerned with localisation of siunitx output based on the locale. If the translations package is available, this is loaded here and used to provide various fixed strings for output.

\texttt{locale locale = \langle locale\rangle}

Selects the \langle locale\rangle used to apply standard settings for other keys, principally exponent-product, inter-unit-product and output-decimal-marker.
Part IV

siunitx-number – Parsing and formatting numbers

This submodule is dedicated to parsing and formatting numbers. A small number of \LaTeX\ math mode commands are assumed to be available as part of the formatted output. The sign commands \texttt{\textbackslash mp}, \texttt{\textbackslash pm}, \texttt{\textbackslash ll}, \texttt{\textbackslash le}, \texttt{\textbackslash gg} and \texttt{\textbackslash ge} are used to replace two-character input; \texttt{\textbackslash pm} is also required for the output of uncertainties. The standard settings require \texttt{\textbackslash times}. For the display of colored negative numbers, the command \texttt{\textbackslash color} is assumed to be available. Where the latter may apply, numbers should be printed inside a group: note that \LaTeX\ grouping is not added within formatted numbers as they may need to be decomposed into parts (see \texttt{\textbackslash siunitx\_number\_output:NN}). Such a color will be the \textit{first} part of the result, meaning that a test for an initial \texttt{\textbackslash color} and following brace group may be used to detect/remove/adjust this part.

1 Formatting numbers

\begin{verbatim}
\texttt{\textbackslash siunitx\_number\_parse:nN}  \langle number\rangle  \langle tl var\rangle
\texttt{\textbackslash siunitx\_number\_parse:VN}  \langle\texttt{(number)}\rangle  \langle tl var\rangle
\end{verbatim}

Parses the \texttt{number} and stores the resulting internal representation in the \texttt{\langle tl var\rangle}. The parsing is influenced by the various key–value settings for numerical input. The \texttt{\langle number\rangle} should comprise a single real value, possibly with comparator, uncertainty and exponent parts. If the number is invalid, or if number parsing is disabled, the result will be an entirely empty \texttt{\langle tl var\rangle}.

The structure of a valid number is:

\begin{verbatim}
\texttt{\langle comparator\rangle}|\langle sign\rangle|\langle integer\rangle|\langle decimal\rangle|\langle uncertainty\rangle|\langle exponent sign\rangle|\langle exponent\rangle
\end{verbatim}

where the two sign parts must be single tokens if present, and all other components must be given in braces. The number will have at least one digit for both the \texttt{\langle integer\rangle} and \texttt{\langle exponent\rangle} parts: these are required. The \texttt{\langle uncertainty\rangle} part should either be blank or contain an \texttt{\langle identifier\rangle} (as a brace group), followed by one or more data entries. Valid uncertainty \texttt{\langle identifiers\rangle} currently are

\begin{itemize}
  \item \texttt{\textbackslash S} A single symmetrical uncertainty (\textit{e.g.} a statistical standard uncertainty). The data item here is a single value representing the uncertainty in the least-significant digits
  \item \texttt{\textbackslash A} A single unsymmetrical uncertainty. The data item here contains two brace groups, each using the same least-significant digit approach as the \texttt{\textbackslash S} type. The positive component is given first and the negative second, and neither has a sign.
  \item A combination of \texttt{\textbackslash S} and \texttt{\textbackslash A} entries, with one data item per entry. These are then iterated over to be output in order.
\end{itemize}

If a decimal marker should be explicitly recorded as present for a value with no decimal digits, the \texttt{\langle decimal\rangle} part should contain \texttt{\textbackslash empty}. 

6
\texttt{\textbackslash siunitx\_number\_process:NN} \texttt{\textbackslash siunitx\_number\_process:N \langle tl\ var\ 1 \rangle \langle tl\ var\ 2 \rangle}

Applies a set of number processing operations to the \langle \textit{internal number} \rangle stored in the \langle tl\ var\ 1 \rangle, \textit{viz.} in order

1. Dropping uncertainty
2. Converting to scientific mode (or similar)
3. Rounding
4. Dropping zero decimal part
5. Forcing a minimum number of digits

with the result stored in \langle tl\ var\ 2 \rangle.

\texttt{\textbackslash siunitx\_number\_output:N} \texttt{\textbackslash siunitx\_number\_output:N \langle number \rangle}
\texttt{\textbackslash siunitx\_number\_output:n \textbackslash siunitx\_number\_output:n} \texttt{\textbackslash siunitx\_number\_output:nN} \texttt{\textbackslash siunitx\_number\_output:nN}

Formats the \langle \textit{number} \rangle (in the siunitx internal format), producing the result in a form suitable for typesetting in math mode. The details for the formatting are controlled by a number of key–value options. Note that formatting does not apply any manipulation (processing) to the number. This function is usable in an e- or x-type expansion, and further uncontrolled expansion is prevented by appropriate use of \texttt{\exp\ not:n} internally.

In the \texttt{NN} version, the \langle \textit{marker} \rangle token is inserted at each possible alignment position in the output, \textit{viz.}

- Between the comparator and the integer \textit{(before} any sign for the integer)
- Between the sign and the first digit of the integer
- Both sides of the decimal marker
- Both sides of the separated uncertainty sign (\textit{i.e.} after the decimal part and before any integer uncertainty part)
- Both sides of the decimal marker for a separated uncertainty
- Both sides of the multiplication symbol for the exponent part.

The \texttt{n} and \texttt{nN} version take a token list, which should be in the internal siunitx format.

\texttt{\textbackslash siunitx\_number\_format:nN} \texttt{\textbackslash siunitx\_number\_format:nN} \texttt{\langle number \rangle \langle tl\ var \rangle}
\texttt{\textbackslash siunitx\_number\_format:nN \langle number \rangle \langle tl\ var \rangle}
\texttt{\textbackslash siunitx\_number\_format:nN \langle number \rangle \langle marker \rangle}
\texttt{\textbackslash siunitx\_number\_format:nN \langle number \rangle \langle marker \rangle}
\texttt{\textbackslash siunitx\_number\_format:nN \langle number \rangle \langle marker \rangle}

Carries out a combination of \texttt{\textbackslash siunitx\_number\_parse:nN}, \texttt{\textbackslash siunitx\_number\_process:NN} and \texttt{\textbackslash siunitx\_number\_output:N} using x-type expansion to place the result in the \langle tl\ var \rangle. If \texttt{\_\_siunitx\_number\_parse\_bool} if \texttt{false}, the input is simply stored inside the \langle tl\ var \rangle inside \texttt{\ensuremath}.

\texttt{\textbackslash siunitx\_number\_adjust\_exponent:Nn} \texttt{\textbackslash siunitx\_number\_adjust\_exponent:Nn} \texttt{\langle number \rangle \langle fp\ expr \rangle}
\texttt{\textbackslash siunitx\_number\_adjust\_exponent:nn} \texttt{\textbackslash siunitx\_number\_adjust\_exponent:nn} \texttt{\langle number \rangle \langle fp\ expr \rangle}

Adjusts the exponent of the \langle \textit{number} \rangle (in internal format) by the \langle \textit{fp\ expr} \rangle and leaves the result in the input stream.
Replaces all multi-token signs and comparators in the \( tl \) var with their single-token equivalents. Replaces any active hyphen tokens with non-active versions.

Determines if the \( \langle \text{tokens} \rangle \) form a valid number which can be fully parsed by \texttt{siunitx}.

Determines if the \( \langle \text{token} \rangle \) is valid in a number based on those tokens currently set up for detection in a number.

A switch to control whether ambiguous numbers are bracketed: this can also be covered in quantity formatting by a setting there.

A switch to control whether any parsing is attempted for numbers.

The list of possible input comparators, exponent markers and signs.

The list of possible input decimal marker(s), and the output marker.

### 1.1 Key–value options

The options defined by this submodule are available within the \texttt{l3keys siunitx} tree.

\begin{verbatim}
bracket-ambiguous-numbers = true|false

bracket-negative-numbers = true|false

    drop-exponent = true|false

    drop-uncertainty = true|false

    drop-zero-decimal = true|false
\end{verbatim}
evaluate-expression = true|false

exponent-base = \langle base \rangle

exponent-mode = engineering|fixed|input|scientific|threshold
Choice which determines whether numbers are converted to exponent form. The option engineering forces exponent form with an exponent which is the smallest power of three which gives a mantissa with an integer part. The option fixed uses a fixed exponent (set in fixed-exponent). The option input leaves the input unchanged (which will therefore produce an exponent only if the input contained one). The choice scientific gives an exponent with the mantissa \( m \) in the range \( 1 \leq m < 10 \). Finally, the option threshold will apply scientific if the exponent of input is outside of the range stored in exponent-thresholds. The standard setting is input.

exponent-product = \langle symbol \rangle

expression = \langle expression \rangle

fixed-exponent = \langle exponent \rangle

digit-group-number = \langle integer \rangle
Sets the size of the block (the number of digits) used when grouping digits. The option digit-group-size applies to the first grouping, \( i.e. \) immediately next to the decimal marker, while digit-group-other-size applies to all other groups. Both can be set using digit-group-size. The standard setting for both options is 3.

group-digits = all|decimal|integer|none
Choice to specify whether digits in a number are grouped. The option none entirely disables this, while all means that both the integer and decimal parts are grouped. The settings integer and decimal activate grouping for the relevant part only. The standard setting is all.

group-minimum-digits = \langle value \rangle
The number of digits that must be present in a numerical part (integer or decimal) before digit grouping is attempted. The standard setting is 4.

group-separator = \langle symbol \rangle
Sets the symbol inserted between groups of digits. The standard setting is a thin space (\( \, \)).

input-close-uncertainty = \langle tokens \rangle
input-comparators = \{tokens\}

input-close-uncertainty = \{tokens\}

input-decimal-markers = \{tokens\}

input-digits = \{tokens\}

input-exponent-markers = \{tokens\}

input-open-uncertainty = \{tokens\}

input-signs = \{tokens\}

input-uncertainty-signs = \{tokens\}

minimum-decimal-digits = \{min\}

minimum-integer-digits = \{min\}

negative-color = \{color\}

output-close-uncertainty = \{symbol\}

output-decimal-marker = \{symbol\}

output-open-uncertainty = \{symbol\}

parse-numbers = true|false

print-implicit-plus = true|false
print-mantissa-implicit-plus = true|false
print-exponent-implicit-plus = true|false

Controls whether the plus sign implicit in a positive number is printed; this can be controlled at the level of the mantissa or exponent, or can be activated for both.
print-unity-mantissa = true|false

print-zero-exponent = true|false

print-zero-integer = true|false

retain-explicit-plus = true|false
Switch which determines if an explicit + is retained as a sign when parsing. The standard setting is false.

retain-explicit-decimal-marker = true|false
Switch which determines if an explicit decimal marker is retained when parsing a number where there is no decimal part to a number (i.e. whether to differentiate 10 and 10.). The standard setting is false.

retain-negative-zero = true|false
Switch which determines if a negative sign is retained where the value of a parsed number is exactly zero. The standard setting is false.

retain-zero-uncertainty = true|false
Switch which determines if an entirely zero uncertainty part is retained on parsing, or whether this is normalised to remove the uncertainty. The standard setting is false.

round-half = even|up
Choice which determines how values of exactly half are rounded. The setting up means that the value is always rounded away from zero, whereas the setting even means that the value will be rounded to the closes even number. The standard setting is up.

round-minimum = ⟨min⟩
Literal which sets a minimum value below which rounded values will be replaced by this value and a > or <, as appropriate for the sign of the value. The standard setting is empty, i.e. there is no minimum.

round-mode = figures|none|places|uncertainty
Choice which specifies the rounding approach used for numbers. The choice figures means that values are rounding to the number of significant figures specified by round-precision. The setting places rounds to round-precision interpreted as a number of decimal places: this may be negative (rounding to an integer). The setting none disables rounding. The setting uncertainty first rounds the uncertainty to the number of significant figures specified by round-precision, then rounds the main value such that its accuracy is correctly specified by this updated uncertainty. The standard setting is none.
**round-pad**

`round-pad = true|false`

Switch which specifies if values should be padded to the required number length when rounding to a number of decimal places. The standard setting is `true`.

---

**round-precision**

`round-precision = ⟨precision⟩`

Integer specifying the number of digits used as a target when rounding: this may be interpreted as decimal places or significant figures, depending on active `round-mode`. The standard setting is 2.

---

**round-zero-positive**

`round-zero-positive = true|false`

Switch to control whether a value rounded to zero is regarded as a positive number if the input was negative. The standard setting is `true`.

---

**tight-spacing**

`tight-spacing = true|false`

---

**uncertainty-descriptor-mode**

`uncertainty-descriptor = bracket|bracket-separator|separator|subscript`

Selects how uncertainty descriptors are formatted: a choice from the options `bracket`, `text` and `subscript`. The option `bracket` wraps the descriptor in parenthesis, `bracket-separator` does the same but also includes a separator between the uncertainty and opening bracket, `separator` places the descriptor after the uncertainty and a separator, and `subscript` formats the descriptor as a subscript. The standard setting is `bracket-separator`.

---

**uncertainty-descriptor-separator**

`uncertainty-descriptor-separator = ⟨separator⟩`

Separator inserted between the uncertainty and descriptor when one is required by `uncertainty-descriptor-mode`. The standard setting is `\␣.`.

---

**uncertainty-descriptors**

`uncertainty-descriptors = ⟨clist⟩`

Stores the list of descriptors used when there are multiple uncertainty components given. This is not used when there is only a single uncertainty component present. The standard setting is `empty`.

---

**uncertainty-mode**

`uncertainty-mode = compact|compact-marker|full|separate`

Switch to determine how single symmetrical uncertainties are formatted. When this is set to `separate`, the uncertainty is printed as an entirely separate number preceded by `\pm`. Other settings all place the uncertainty in parentheses directly attached to the main value. The standard setting of `compact` prints digits of uncertainty in the least-significant digits. It does not print a decimal marker if the uncertainty crosses the decimal. The setting `full` prints the full value of the uncertainty. The setting `compact-marker` is available to print in the `compact` style except where the uncertainty crosses the decimal, in which case the `full` style is used. The standard setting is `compact`.

---

**uncertainty-separator**

`uncertainty-separator = ⟨separator⟩`

Stores the separator used between the main value and uncertainty when using the `compact` or `compact-marker` style setting for `uncertainty-mode`.

---
zero-decimal-as-symbol  zero-decimal-as-symbol = true|false

Switch to determine if an entirely zero decimal part is replaced by a symbol. Does not apply if the decimal part is marked as entirely absent.

zero-symbol  zero-symbol = ⟨symbol⟩

Material printed when a zero numerical component is replaced by a symbol.
siunitx-print – Printing material with font control

1 Printing quantities

This submodule is focused on providing controlled printing for numbers and units. Key to this is control of font: conventions for printing quantities mean that the exact nature of the output is important. At the same time, this module provides flexibility for the user in terms of which aspects of the font are responsive to the surrounding general text. Printing material may also take place in text or math mode.

The printing routines assume that normal \LaTeX font selection commands are available, in particular \texttt{\textbackslash bfseries, \textbackslash mathrm, \textbackslash mathversion, \textbackslash fontfamily, \textbackslash fontseries and \textbackslash fontshape, \textbackslash familydefault, \textbackslash seriesdefault, \textbackslash shapedefault and \textbackslash selectfont}. It also requires the standard \LaTeX kernel commands \texttt{\textbackslash ensuremath, \texttt{\textbackslash mbox, \texttt{\textbackslash textsubscript and \texttt{\textbackslash textsuperscript} for printing in text mode. The following packages are also required to provide the functionality detailed.

- \texttt{color}: support for color using \texttt{\textbackslash textcolor}
- \texttt{textcomp}: \texttt{\textbackslash textminus, \textbackslash textpm \textbackslash texttimes} and \texttt{\textbackslash textcenteredperiod} for printing in text mode
- \texttt{amstext}: the \texttt{\textbackslash text} command for printing in text mode

For detection of math mode fonts, as well as \texttt{\textbackslash mathrm}, the existence of \texttt{\textbackslash symoperators} is assumed; other math font commands are not required to exist.

\begin{verbatim}
\sisunitx_print_number:n \sisunitx_print_number:n \langle material \rangle \sisunitx_print_unit:n \sisunitx_print_unit:n \langle material \rangle
\end{verbatim}

Prints the \langle material \rangle according the the prevailing settings for the submodule as applicable to the \langle type \rangle of content (\texttt{number} or \texttt{unit}). The \langle material \rangle should comprise normal \LaTeX mark-up for numbers or units. In particular, units will typically use \texttt{\textbackslash mathrm} to indicate material to be printed in the current upright roman font, and ^ and _ will typically be used to indicate super- and subscripts, respectively. These elements will be correctly handled when printing for example using \texttt{\textbackslash mathsbf} in math mode, or using only text fonts.

\begin{verbatim}
\sisunitx_print_match:n \sisunitx_print_match:n \langle material \rangle \sisunitx_print_math:n \sisunitx_print_math:n \langle material \rangle \sisunitx_print_text:n \sisunitx_print_text:n \langle material \rangle
\end{verbatim}

Prints the \langle material \rangle as described for \texttt{\textbackslash siunitx_print_:n} but with a fixed text or math mode output. The printing does not set color (which is managed on a \texttt{unit/number} basis), but otherwise sets the font as described above. The \texttt{match} function uses either the prevailing math or text mode.
1.1 Key–value options

The options defined by this submodule are available within the \l3keys siunitx tree.

\begin{verbatim}
color color = ⟨color⟩

Color to apply to printed output: the latter should be a named color defined for use with \textcolor. The standard setting is empty (no color).

mode mode = match|math|text

Selects which mode (math or text) the output is printed in: a choice from the options match, math or text. The option match matches the mode prevailing at the point \siunitx_print...:n is called. The math and text options choose the relevant \TeX mode for printing. The standard setting is math.

number-color number-color = ⟨color⟩

Color to apply to numbers in output: the latter should be a named color defined for use with \textcolor. The standard setting is empty (no color).

number-mode number-mode = match|math|text

Selects which mode (math or text) the numbers are printed in: a choice from the options match, math or text. The option match matches the mode prevailing at the point \siunitx_print_number:n is called. The math and text options choose the relevant \TeX mode for printing. The standard setting is math.

propagate-math-font propagate-math-font = true|false

Switch to determine if the currently-active math font is applied within printed output. This is relevant only when \siunitx_print...:n is called from within math mode: in text mode there is not active math font. When not active, math mode material will be typeset using standard math mode fonts without any changes being made to the supplied argument. The standard setting is false.

reset-math-version reset-math-version = true|false

Switch to determine whether the active \mathversion is reset to normal when printing in math mode. Note that math version is typically used to select \boldmath, though it is also be used by e.g. sansmath. The standard setting is true.

reset-text-family reset-text-family = true|false

Switch to determine whether the active text family is reset to \rmfamily when printing in text mode. The standard setting is true.

reset-text-series reset-text-series = true|false

Switch to determine whether the active text series is reset to \mdseries when printing in text mode. The standard setting is true.

reset-text-shape reset-text-shape = true|false

Switch to determine whether the active text shape is reset to \upshape when printing in text mode. The standard setting is true.
\end{verbatim}
text-family-to-math text-family-to-math = true|false

Switch to determine if the family of the current text font should be applied (where possible) to printing in math mode. The standard setting is false.

text-font-command text-font-command = (cmd)

Command applied to text during output, inserted after any reset of font set-up. This can therefore be used to apply non-standard font set up when printing in text mode. The standard setting is empty.

text-series-to-math text-series-to-math = true|false

Switch to determine if the weight of the current text font should be applied (where possible) to printing in math mode. This is achieved by setting the \mathversion, and so will override reset-math-version. The mappings between text and math weight are set. The standard setting is false.

text-subscript-command text-subscript-command = (cmd)
text-superscript-command text-superscript-command = (cmd)

Sets the command used when printing material in sub- or superscript positions in text mode. The standard settings are \textsubscript and \textsuperscript, respectively.

unit-color unit-color = (color)

Color to apply to units in output: the latter should be a named color defined for use with \textcolor. The standard setting is empty (no color).

unit-mode unit-mode = match|math|text

Selects which mode (math or text) units are printed in: a choice from the options match, math or text. The option match matches the mode prevailing at the point \siunitx_print...:n is called. The math and text options choose the relevant \TeX mode for printing. The standard setting is math.
Series version mapping \( \text{series-version-mapping} / (\text{weight}) = (\text{version}) \)

Defines how \texttt{siunitx} maps from text font weight to math font version. The pre-defined weights are those used as-standard by \texttt{autoinst}:

- \texttt{ul}
- \texttt{el}
- \texttt{l}
- \texttt{sl}
- \texttt{m}
- \texttt{sb}
- \texttt{b}
- \texttt{eb}
- \texttt{ub}

As standard, the \texttt{m} weight maps to \texttt{normal} math version whilst all of the \texttt{b} weights map to \texttt{bold} and all of the \texttt{l} weights map to \texttt{light}. 

---

17
Part VI

\textbf{\texttt{siunitx-quantity} – Quantities}

This submodule is focussed on providing controlled printing for quantities: the combination of a number and a unit. It largely builds on the submodules \texttt{siunitx-number} and \texttt{siunitx-unit}. A small number of adjustments are made to standard set up in the latter to reflect additional functionality added here.

```latex
\texttt{\textbackslash siunitx\_quantity:nn} \ \texttt{\textbackslash siunitx\_quantity:nn \{\langle number\rangle\} \{\langle unit\rangle\}}
```

Parses the \langle number \rangle and the \langle unit \rangle as detailed for \texttt{\textbackslash siunitx\_number\_parse:nN} and \texttt{\textbackslash siunitx\_unit\_format:nN}, the prints the results using \texttt{\textbackslash siunitx\_print\_unit:n}.

```latex
\texttt{\textbackslash siunitx\_quantity\_print:nn} \ \texttt{\textbackslash siunitx\_quantity\_print:nn \{\langle number\rangle\} \{\langle unit\rangle\}}
\texttt{\textbackslash siunitx\_quantity\_print:nn (nV|VV|xV)}
```

A low-level function which prints the quantity directly: there is no processing applied to either the \langle number \rangle or \langle unit \rangle. The two parts are printed using \texttt{\textbackslash siunitx\_print\_unit:n} and appropriate spacing and break-prevention is applied.

**allow-quantity-breaks**  
allow-quantity-breaks = true|false

Specifies whether breaks are permitted between units. The standard setting is false.

**prefix-mode**  
prefix-mode = combine-exponent|extract-exponent|input

Selects the method used for producing prefixes: a choice from the options combine-exponent, extract-exponent and input. The option combine-exponent combines any exponent from the number with the prefix of the first unit, and prints the updated prefix. The option extract-exponent removes all prefixes from the unit, and combines them with the exponent of number. The option input prints prefixes and exponent as given in the source. The standard setting is input.

**quantity-product**  
quantity-product = \langle tokens \rangle

The product marker used between a number and the unit. The standard setting is \textbackslash ,.

**separate-uncertainty-units**  
separate-uncertainty-units = bracket|repeat|single

Specifies how units are applied when a separated uncertainty is present: a choice from bracket, repeat and single. The option bracket places brackets around the number, with the unit given after these. The option repeat means that the unit it printed with the main value and with the uncertainty. When single is set, the unit is printed only once and no brackets are applied. The standard setting is bracket.
Part VII

\texttt{siunitx-symbol} – Symbol-related settings
Part VIII
siunitx-table – Formatting numbers in tables

1 Numbers in tables

This submodule is concerned with formatting numbers in table cells or similar fixed-width contexts. The main function, `\siunitx_cell_begin:w`, is designed to work with the normal \( \LaTeX \) tabular cell construct featuring `\ignorespaces`. Therefore, if used outside of a \( \LaTeX \) tabular, it is necessary to provide this token.

\begin{verbatim}
\siunitx_cell_begin:w \siunitx_cell_begin:w \langle preamble\rangle \ignorespaces
\langle content\rangle
\siunitx_cell_end:
\siunitx_cell_end:
\end{verbatim}

Collects the `\langle preamble\rangle` and `\langle content\rangle` tokens, and determines if it is text or a number (as parsed by `\siunitx_number_parse:nN`). It produces output of a fixed width suitable for alignment in a table, although it is not required that the code is used within a cell. Note that `\ignorespaces` must occur in the “cell”: it marks the end of the \( \LaTeX \) `\halign` template.

1.1 Key–value options

The options defined by this submodule are available within the \l3keys siunitx tree.

<table>
<thead>
<tr>
<th>Option</th>
<th>Syntax</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table-align-comparator</td>
<td>`table-align-comparator = true</td>
<td>false`</td>
<td>true</td>
</tr>
<tr>
<td>table-align-exponent</td>
<td>`table-align-exponent = true</td>
<td>false`</td>
<td>true</td>
</tr>
<tr>
<td>table-align-text-after</td>
<td>`table-align-text-after = true</td>
<td>false`</td>
<td>true</td>
</tr>
<tr>
<td>table-align-text-before</td>
<td>`table-align-text-before = true</td>
<td>false`</td>
<td>true</td>
</tr>
<tr>
<td>table-align-uncertainty</td>
<td>`table-align-uncertainty = true</td>
<td>false`</td>
<td>true</td>
</tr>
</tbody>
</table>

20
table-alignment  table-alignment = center|left|right
Selects the alignment of all tabular content with the margins of the table cell (or other boundary). See also table-number-alignment and table-text-alignment. The standard setting is center.

table-alignment-mode  table-alignment-mode = format|marker|none
Selects the method used to align numbers with the desired position in the cell (set by table-alignment). When set to format, a dedicated amount of space is calculated from the table-format. When marker is selected, alignment is carried out symmetrically around the decimal marker. Finally, none switches off all alignment: numbers are parsed and formatted but with no attempt at placement within the cell. The standard setting is marker.

table-auto-round  table-auto-round = true|false
Switch which determines whether numbers are rounded to fit within the table-format specification (if possible). The standard setting is false.

table-column-width  table-column-width = ⟨width⟩
Sets the width of the table column used for numbers. This is only used when table-fixed-width is true.

table-fixed-width  table-fixed-width = true|false
Switch which determines whether a fixed-width column is used for numbers in tables. When true, the width is taken from table-column-width. The standard setting is false.

table-format  table-format = ⟨format⟩
Describes the amount of space that should be reserved when table-alignment-mode is set to format. The ⟨format⟩ takes the same general form as input for a table cell, with the numerical parts describing how many digits to reserve space for. For example, 1.2e3 would allow space for one digit in the integer part, two in the decimal part and three in the exponent part. Signs can be allowed for using any valid input sign, so for example +1.2 \pm 1.2 would allow for a sign, a number with one integer and two decimal digits and an uncertainty of the same size.

table-number-alignment  table-number-alignment = center|left|right
Selects the alignment of numerical content with the margins of the table cell (or other boundary). See also table-alignment and table-text-alignment. The standard setting is center.

table-text-alignment  table-text-alignment = center|left|none|right
Selects the alignment of non-numerical content with the margins of the table cell (or other boundary). See also table-alignment and table-number-alignment. Notice the additional support for none here. The standard setting is center.
Part IX
siunitx-unit – Parsing and formatting units

This submodule is dedicated to formatting physical units. The main function, \texttt{\textbackslash siunitx\_unit\_format:nN}, takes user input specifying physical units and converts it into a formatted token list suitable for typesetting in math mode. While the formatter will deal correctly with “literal” user input, the key strength of the module is providing a method to describe physical units in a “symbolic” manner. The output format of these symbolic units can then be controlled by a number of key-value options made available by the module.

A small number of \LaTeX\ math mode commands are assumed to be available as part of the formatted output. The \texttt{\mathchoice} command (normally the \TeX\ primitive) is needed when using different settings for inline and siplay per-mode. The commands \texttt{\frac}, \texttt{\mathrm}, \texttt{\mbox}, \texttt{\_}, and \texttt{\_}, are used by the standard module settings. For the display of colored (highlighted) and cancelled units, the commands \texttt{\textcolor} and \texttt{\cancel} are assumed to be available.

1 Formatting units

\begin{verbatim}
\siunitx_unit_format:nN \siunitx_unit_format:nN \{\langle units\rangle\} \langle tl var\rangle
\end{verbatim}

This function converts the input \texttt{\langle units\rangle} into a processed \texttt{\langle tl var\rangle} which can then be inserted in math mode to typeset the material. Where the \texttt{\langle units\rangle} are given in symbolic form, described elsewhere, this formatting process takes place in two stages: the \texttt{\langle units\rangle} are parsed into a structured form before the generation of the appropriate output form based on the active settings. When the \texttt{\langle units\rangle} are given as literals, processing is minimal: the characters . and - are converted to unit products (boundaries). In both cases, the result is a series of tokens intended to be typeset in math mode with appropriate choice of font for typesetting of the textual parts.

For example,

\begin{verbatim}
\siunitx_unit_format:nN \{ \kilo \metre \per \second \} \l_tmpa_tl
\end{verbatim}

will, with standard settings, result in \texttt{\l_tmpa_tl} being set to

\begin{verbatim}
\mathrm{km}\,, \mathrm{s}^{-1}
\end{verbatim}
This function formats the \(\text{units}\) in the same way as described for \siunitx{\text{unit-format}:nnN}. When the input is given in symbolic form, any decimal unit prefixes will be extracted and the overall power of ten that these represent will be stored in the \(\text{fp var}\).

For example,
\begin{verbatim}
\siunitx{\text{unit-format}:nnN}{\text{kilo \ metre \ per \ second}} \l\_tmpa\_tl \l\_tmpa\_fp
\end{verbatim}
will, with standard settings, result in \l\_tmpa\_tl being set to \text{m s}^{-1} with \l\_tmpa\_fp taking value 3. Note that the latter is a floating point variable: it is possible for non-integer values to be obtained here.

This function formats the \(\text{units}\) in the same way as described for \siunitx{\text{unit-format}:nnN}. The \(\text{exponent}\) is combined with any prefix for the first unit of the \(\text{units}\), and an updated prefix is introduced.

For example,
\begin{verbatim}
\siunitx{\text{unit-format:nnN}}{\text{metre per second}} {3} \l\_tmpa\_tl
\end{verbatim}
will, with standard settings, result in \l\_tmpa\_tl being set to \text{km s}^{-1}.

These function formats the \(\text{units}\) in the same way as described for \siunitx{\text{unit-format}:nnN}. The units are multiplied by the \(\text{factor}\), and further processing takes place as previously described.

For example,
\begin{verbatim}
\siunitx{\text{unit-format:nnN}}{\text{metre per second}} {3} \l\_tmpa\_tl
\end{verbatim}
will, with standard settings, result in \l\_tmpa\_tl being set to \text{km}^{-3}.
2 Defining symbolic units

\siunitx_declare_prefix:Nnn \siunitx_declare_prefix:Nnn \siunitx_declare_prefix:Nnn
\siunitx_declare_prefix:Nnn \siunitx_declare_prefix:Nnn

Defines a symbolic \textit{prefix} (which should be a control sequence such as \texttt{kilo}) to be converted by the parser to the \textit{symbol}. The latter should consist of literal content \textit{(e.g. k)}. In literal mode the \textit{symbol} will be typeset directly. The prefix should represent an integer \textit{power} of 10, and this information may be used to convert from one or more \textit{prefix} symbols to an overall power applying to a unit. See also \siunitx_declare_prefix:Nnn.

\siunitx_declare_power:NNn \siunitx_declare_power:NNn

Defines two symbolic \textit{powers} (which should be control sequences such as \texttt{squared}) to be converted by the parser to the \textit{value}. The latter should be an integer or floating point number in the format defined for \texttt{l3fp}. Powers may precede a unit or be give after it: both forms are declared at once, as indicated by the argument naming. In literal mode, the \textit{value} will be applied as a superscript to either the next token in the input (for the \textit{pre-power}) or appended to the previously-typeset material (for the \textit{post-power}).

\siunitx_declare_qualifier:Nn \siunitx_declare_qualifier:Nn

Defines a symbolic \textit{qualifier} (which should be a control sequence such as \texttt{catalyst}) to be converted by the parser to the \textit{meaning}. The latter should consist of literal content \textit{(e.g. cat)}. In literal mode the \textit{meaning} will be typeset following a space after the unit to which it applies.

\siunitx_declare_unit:Nn \siunitx_declare_unit:Nnn \siunitx_declare_unit:Nnn
\siunitx_declare_unit:Nxn

Defines a symbolic \textit{unit} (which should be a control sequence such as \texttt{kilogram}) to be converted by the parser to the \textit{meaning}. The latter may consist of literal content \textit{(e.g. kg)}, other symbolic unit commands \textit{(e.g. \texttt{kilo\gram}) or a mixture of the two. In literal mode the \textit{meaning} will be typeset directly. The version taking an \textit{options} argument may be used to support per-unit options: these are applied at the top level or using \texttt{\siunitx_unit_options_apply:n}.

\l\_siunitx\_unit\_font\_tl

The font function which is applied to the text of units when constructing formatted units: set by \texttt{font-command}.
The fraction function which is applied when constructing fractional units: set by `fraction-command`.

This sequence contains all of the symbolic names defined: these will be in the form of control sequences such as `\kilogram`. The order of the sequence is unimportant. This includes prefixes and powers as well as units themselves.

This sequence contains all of the symbolic unit names defined: these will be in the form of control sequences such as `\kilogram`. In contrast to \l\_siunitx\_unit\_symbolic\_seq, it only holds units themselves.

3 Per-unit options

`\siunitx\_unit\_options\_apply:n` \siunitx\_unit\_options\_apply:n ⟨unit(s)⟩

Applies any unit-specific options set up using `\siunitx\_declare\_unit:Nnn`. This allows there use outside of unit formatting, for example to influence spacing in quantities. The options are applied only once at a given group level, which allows for user over-ride via `\keys\_set:nn { siunitx } { ... }`.

4 Units in (PDF) strings

`\siunitx\_unit\_pdfstring\_context:` \group\_begin:\n   \siunitx\_unit\_pdfstring\_context:\n   ⟨Expansion context⟩ ⟨units⟩\n   \group\_end:

Sets symbol unit macros to generate text directly. This is needed in expansion contexts where units must be converted to simple text. This function is itself not expandable, so must be using within a surrounding group as show in the example.

5 Pre-defined symbolic unit components

The unit parser is defined to recognise a number of pre-defined units, prefixes and powers, and also interpret a small selection of “generic” symbolic parts.

Broadly, the pre-defined units are those defined by the BIPM in the documentation for the *International System of Units* (SI) [1]. As far as possible, the names given to the command names for units are those used by the BIPM, omitting spaces and using only ASCII characters. The standard symbols are also taken from the same documentation. In the following documentation, the order of the description of units broadly follows the SI Brochure.
The base units as defined in the SI Brochure [2]. Notice that \texttt{meter} is defined as an alias for \texttt{metre} as the former spelling is common in the US (although the latter is the official spelling).

- \texttt{kilogram}
- \texttt{metre}
- \texttt{meter}
- \texttt{mole}
- \texttt{kelvin}
- \texttt{candela}
- \texttt{second}
- \texttt{ampere}

The base unit \texttt{kilogram} is defined using an SI prefix: as such the (derived) unit \texttt{gram} is required by the module to correctly produce output for the \texttt{kilogram}.

Prefixes, all of which are integer powers of 10: the powers are stored internally by the module and can be used for conversion from prefixes to their numerical equivalent. These prefixes are documented in Section 3.1 of the SI Brochure.

Note that the \texttt{kilo} prefix is required to define the base \texttt{kilogram} unit. Also note the two spellings available for \texttt{deca/deka}. 

- \texttt{quecto}
- \texttt{ronto}
- \texttt{yocto}
- \texttt{zepto}
- \texttt{atto}
- \texttt{femto}
- \texttt{pico}
- \texttt{nano}
- \texttt{micro}
- \texttt{milli}
- \texttt{centi}
- \texttt{deci}
- \texttt{deca}
- \texttt{deka}
- \texttt{hecto}
- \texttt{kilo}
- \texttt{mega}
- \texttt{giga}
- \texttt{tera}
- \texttt{peta}
- \texttt{exa}
- \texttt{zetta}
- \texttt{yotta}
- \texttt{ronna}
- \texttt{quetta}
The defined SI units with defined names and symbols, as given in Table 4 of the SI Brochure. Notice that the names of the units are lower case with the exception of \textdegree C, and that this unit name includes “degree”.

\begin{itemize}
\item \texttt{astronomicalunit} \footnotesize Units accepted for use with the SI: here \texttt{minute} is a unit of time not of plane angle. These units are taken from Table 8 of the SI Brochure. For the unit \texttt{litre}, both \texttt{l} and \texttt{L} are listed as acceptable symbols: the latter is the standard setting of the module. The alternative spelling \texttt{liter} is also given for this unit for US users (as with \texttt{metre}, the official spelling is “re”).
\item \texttt{arcminute} \texttt{arcsecond} \texttt{degree} \footnotesize Units for plane angles accepted for use with the SI: to avoid a clash with units for time, here \texttt{arcminute} and \texttt{arcsecond} are used in place of \texttt{minute} and \texttt{second}. These units are taken from Table 8 of the SI Brochure.
\item \texttt{percent} \footnotesize The mathematical concept of percent, usable with the SI as detailed in Section 5.4.7 of the SI Brochure.
\item \texttt{square} \texttt{cubic} \footnotesize Pre-defined unit powers which apply to the next \texttt{(prefix)}\texttt{/}\texttt{(unit)} combination.
\end{itemize}
\[\text{squared}\quad \text{\langle\text{prefix}\rangle\ \text{\langleunit\rangle}\ \text{squared}}\]
\[\text{\cubed}\quad \text{\langle\text{prefix}\rangle\ \text{\langleunit\rangle}\ \text{\cubed}}\]

Pre-defined unit powers which apply to the preceding \langle\text{prefix}/\text{\langleunit\rangle}\ combination.

\[\text{\per}\quad \text{\langle\text{prefix}\rangle\ \text{\langleunit\rangle}\ \langle\text{\power\rangle}}\]

Indicates that the next \langle\text{prefix}/\text{\langleunit\rangle}/\langle\text{\power\rangle}\ combination is reciprocal, i.e. raises it to the power \(-1\). This symbolic representation may be applied in addition to a \text{\power}, and will work correctly if the \text{\power} itself is negative. In literal mode \text{\per} will print a slash (“/”).

\[\text{\cancel}\quad \text{\langle\text{prefix}\rangle\ \text{\langleunit\rangle}\ \langle\text{\power\rangle}}\]

Indicates that the next \langle\text{prefix}/\text{\langleunit\rangle}/\langle\text{\power\rangle}\ combination should be “cancelled out”. In the parsed output, the entire unit combination will be given as the argument to a function \text{\cancel}, which is assumed to be available at a higher level. In literal mode, the same higher-level \text{\cancel} will be applied to the next token. It is the responsibility of the calling code to provide an appropriate definition for \text{\cancel} outside of the scope of the unit parser.

\[\text{\highlight}\quad \text{\langle\langle\text{color}\rangle\ \langle\text{\langle\text{prefix}\rangle\ \text{\langleunit\rangle}\ \langle\text{\power\rangle}\}}\]

Indicates that the \langle\text{\power\rangle} applies to the current \langle\text{\text{\prefix}/\\langle\text{\unit\rangle}/\langle\text{\power\rangle}\ combination. As shown, \text{\raiseto} applies to the next \text{\langle\text{\unit\rangle} whereas \text{\tothe} applies to the preceding unit. In literal mode the \text{\power} will be printed as a superscript attached to the next token (\text{\raiseto}) or preceding token (\text{\tothe}) as appropriate.

5.1 Key–value options

The options defined by this submodule are available within the \text{\l3keys\ siunitx} tree.

\[\text{\bracket-unit-denominator}\quad \text{\bracket-unit-denominator = true|false}\]

Switch to determine whether brackets are added to the denominator part of a unit when printed using inline fractional form (with \text{\per-mode} as \text{\repeated-symbol} or \text{\symbol}). The standard setting is \text{true}.
extract-mass-in-kilograms = true|false
Determines whether prefix extraction treats kilograms as a base unit; when set false, grams are used. The standard setting is true.

forbid-literal-units = true|false
Switch which determines if literal units are allowed when parsing is active; does not apply when parse-units is false.

fraction-command = \{command\}
Command used to create fractional output when per-mode is set to fraction. The standard setting is \frac.

inter-unit-product = \{separator\}
Inserted between unit combinations in parsed mode, and used to replace . and - in literal mode. The standard setting is \\

parse-units = true|false
Determines whether parsing of unit symbols is attempted or literal mode is used directly. The standard setting is true.

per-mode = fraction|power|power-positive-first|repeated-symbol|single-symbol|symbol
Selects how the negative powers (\per) are formatted: a choice from the options fraction, power, power-positive-first, repeated-symbol, single-symbol and symbol. The option fraction generates fractional output when appropriate using the command specified by the fraction-command option. The setting power uses reciprocal powers leaving the units in the order of input, while power-positive-first uses the same display format but sorts units such that the positive powers come before negative ones. The symbol setting uses a symbol (specified by per-symbol) between positive and negative powers, while repeated-symbol uses the same symbol but places it before every unit with a negative power (this is mathematically “wrong” but often seen in real work). The option single-symbol will use a symbol if exactly one is required (i.e. with a single negative power), and will otherwise use powers. The standard setting is power.

The inline-... and display-... settings take the same options and work in exactly the same way, but are restricted in where they apply. The display version only applies in display math contexts, and the inline version applies in all others.

per-symbol = \{symbol\}
Specifies the symbol to be used to denote negative powers when the option per-mode is set to repeated-symbol or symbol. The standard setting is /.

per-symbol-script-correction = \{insert\}
Specifies the tokens used to correct spacing when the symbol set by per-symbol is immediately preceded by a superscript power. The standard setting is \\

29
**power-half-as-sqrt**  
`power-half-as-sqrt = true|false`  
Used to determine whether a power of exactly half is converted to \(\sqrt{\text{ }}\) in the output. The standard setting is `false`.

**qualifier-mode**  
`qualifier-mode = bracket|combine|phrase|subscript`  
Selects how qualifiers are formatted: a choice from the options `bracket`, `combine`, `phrase` and `subscript`. The option `bracket` wraps the qualifier in parenthesis, `combine` joins the qualifier with the unit directly, `phrase` joins the material using `qualifier-phrase` as a link, and `subscript` formats the qualifier as a subscript. The standard setting is `subscript`.

**qualifier-phrase**  
`qualifier-phrase = \langle\text{ phrase}\rangle`  
Defines the `\langle\text{ phrase}\rangle` used when `qualifier-mode` is set to `phrase`.

**sticky-per**  
`sticky-per = true|false`  
Used to determine whether \(\text{ per}\) should be applied one a unit-by-unit basis (when `false`) or should apply to all following units (when `true`). The latter mode is somewhat akin conceptually to the `\LaTeX \over` primitive. The standard setting is `false`.

**unit-font-command**  
`unit-font-command = \langle\text{ command}\rangle`  
Command applied to text during output of units: should be command usable in math mode for font selection. Notice that in a typical unit this does not (necessarily) apply to all output, for example powers or brackets. The standard setting is \(\text{ \texttt{\textbackslash mathrm{}}}\).

**References**

[1] *The International System of Units (SI)*,  

[2] *SI base units*,  
Part X

\textbf{siunitx-abbreviations} – Abbreviations

\( \text{\textbackslash A} \) Abbreviations for currents.
\( \text{\textbackslash pA} \)
\( \text{\textbackslash nA} \)
\( \text{\textbackslash \mu A} \)
\( \text{\textbackslash mA} \)
\( \text{\textbackslash kA} \)

\( \text{\textbackslash fg} \) Abbreviations for masses.
\( \text{\textbackslash pg} \)
\( \text{\textbackslash ng} \)
\( \text{\textbackslash \mu g} \)
\( \text{\textbackslash mg} \)
\( \text{\textbackslash g} \)
\( \text{\textbackslash kg} \)

\( \text{\textbackslash K} \) Abbreviations for temperature.

\( \text{\textbackslash m} \) Abbreviations for lengths.
\( \text{\textbackslash \mu m} \)
\( \text{\textbackslash um} \)
\( \text{\textbackslash \mu m} \)
\( \text{\textbackslash mm} \)
\( \text{\textbackslash cm} \)
\( \text{\textbackslash dm} \)
\( \text{\textbackslash km} \)

\( \text{\textbackslash s} \) Abbreviations for times.
\( \text{\textbackslash as} \)
\( \text{\textbackslash fs} \)
\( \text{\textbackslash ps} \)
\( \text{\textbackslash ns} \)
\( \text{\textbackslash us} \)
\( \text{\textbackslash ms} \)

\( \text{\textbackslash Hz} \) Abbreviations for frequencies.
\( \text{\textbackslash mHz} \)
\( \text{\textbackslash kHz} \)
\( \text{\textbackslash MHz} \)
\( \text{\textbackslash GHz} \)
\( \text{\textbackslash THz} \)
Abbreviations for moles.
\mol
\fmol
\pmol
\nmol
\umol
\mmol
\kmol

Abbreviations for potentials.
\V
\pV
\nV
\uV
\mV
\kV

Abbreviations for volumes.
\hl
\l
\ml
\ul
\hL
\L
\mL
\uL

Abbreviations for powers.
\W
\nW
\uW
\mW
\kW
\MW
\GW

Abbreviations for energies.
\kJ
\J
\mJ
\uJ
\eV
\meV
\keV
\MeV
\GeV
\TeV

Abbreviations for forces.
\N
\mN
\kN
\MN
Abbreviations for pressures.
\( \text{Pa} \)  \( \text{kPa} \)  \( \text{MPa} \)  \( \text{GPa} \)

Abbreviations for resistance.
\( \text{m\,ohm} \)  \( \text{k\,ohm} \)  \( \text{Mohm} \)

Abbreviations for capacitance.
\( \text{F} \)  \( \text{fF} \)  \( \text{pF} \)  \( \text{nF} \)  \( \text{uF} \)  \( \text{mF} \)

Abbreviations for inductance.
\( \text{H} \)  \( \text{fH} \)  \( \text{pH} \)  \( \text{nH} \)  \( \text{uH} \)  \( \text{mH} \)

Abbreviations for charge.
\( \text{C} \)  \( \text{nC} \)  \( \text{\mu C} \)  \( \text{\mu C} \)

Abbreviation for decibel.
\( \text{dB} \)

Abbreviation for kilowatt–hours.
\( \text{kWh} \)
Part XI

siunitx-binary – Binary units

This submodule provides binary units and prefixes. These are not formally part of the SI but are recommended by BIPM as units of information.

Prefixes, all of which are integer powers of 2: the powers are not stored or available for conversion.

\kibi \mebi \gibi \tebi \pebi \zebi \exbi \yobi

Units for bits and bytes.

\bit \byte
Part XII

**siunitx-command – Units as document command**

This submodule provides support for creating free-standing document commands for unit macros.

1 Creating units as document commands

```latex
\sisetup{command}
```

Maps over the list of known unit commands and creates the appropriate document command to support them, as controlled by the options below.

1.1 Key–value options

The options defined by this submodule are available within the \l3keys siunitx tree. These options are all preamble-only.

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>free-standing-units</td>
<td>true/false</td>
<td>Switch to determine whether free standing document commands are created for symbolic units. This will include not only units themselves but also prefixes, etc. The standard setting is false.</td>
</tr>
<tr>
<td>overwrite-commands</td>
<td>true/false</td>
<td>Switch to determine whether when creating free standing document commands, any existing document commands are overwritten. The standard setting is false.</td>
</tr>
<tr>
<td>space-before-unit</td>
<td>true/false</td>
<td>Switch to determine whether a space is inserted before free standing document commands. The standard setting is false.</td>
</tr>
<tr>
<td>unit-optional-argument</td>
<td>true/false</td>
<td>Switch to determine whether free standing document commands take an optional argument (a number). The standard setting is false.</td>
</tr>
<tr>
<td>use-xspace</td>
<td>true/false</td>
<td>Switch to determine whether free standing document commands use the <code>xparse</code> package to insert space after the command names. The standard setting is false. When set true, the <code>xparse</code> package will be loaded at the start of the document if not already available.</td>
</tr>
</tbody>
</table>
Part XIII

\texttt{siunitx-emulation} – Emulation
## Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

### Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>,</code></td>
<td>9, 18, 22</td>
</tr>
<tr>
<td>\A</td>
<td>31</td>
</tr>
<tr>
<td>allow-quantity-breaks</td>
<td>18</td>
</tr>
<tr>
<td>\amperer</td>
<td>26</td>
</tr>
<tr>
<td>angle-mode</td>
<td>1</td>
</tr>
<tr>
<td>angle-symbol-degree</td>
<td>1</td>
</tr>
<tr>
<td>angle-symbol-minute</td>
<td>1</td>
</tr>
<tr>
<td>angle-symbol-over-decimal</td>
<td>1</td>
</tr>
<tr>
<td>angle-symbol-second</td>
<td>1</td>
</tr>
<tr>
<td>arc-separator</td>
<td>1</td>
</tr>
<tr>
<td>\arcminute</td>
<td>27</td>
</tr>
<tr>
<td>\arcsecond</td>
<td>27</td>
</tr>
<tr>
<td>as</td>
<td>31</td>
</tr>
<tr>
<td>astronomical-unit</td>
<td>27</td>
</tr>
<tr>
<td>\atto</td>
<td>26</td>
</tr>
<tr>
<td>\cubic</td>
<td>28</td>
</tr>
<tr>
<td>\cubed</td>
<td>28</td>
</tr>
<tr>
<td>\cubic</td>
<td>27</td>
</tr>
<tr>
<td>\Dalton</td>
<td>27</td>
</tr>
<tr>
<td>\day</td>
<td>27</td>
</tr>
<tr>
<td>\dB</td>
<td>33</td>
</tr>
<tr>
<td>\deca</td>
<td>26</td>
</tr>
<tr>
<td>\deci</td>
<td>26</td>
</tr>
<tr>
<td>\decibel</td>
<td>27</td>
</tr>
<tr>
<td>\degree</td>
<td>27</td>
</tr>
<tr>
<td>\degreeCelsius</td>
<td>27</td>
</tr>
<tr>
<td>\deka</td>
<td>26</td>
</tr>
<tr>
<td>digit-group-first-size</td>
<td>9</td>
</tr>
<tr>
<td>digit-group-other-size</td>
<td>9</td>
</tr>
<tr>
<td>digit-group-size</td>
<td>9</td>
</tr>
<tr>
<td>display-per-mode</td>
<td>29</td>
</tr>
<tr>
<td>\dm</td>
<td>31</td>
</tr>
<tr>
<td>drop-exponent</td>
<td>8</td>
</tr>
<tr>
<td>drop-uncertainty</td>
<td>8</td>
</tr>
<tr>
<td>drop-zero-decimal</td>
<td>8</td>
</tr>
<tr>
<td>\becquerel</td>
<td>27</td>
</tr>
<tr>
<td>\bel</td>
<td>27</td>
</tr>
<tr>
<td>\bfseries</td>
<td>14</td>
</tr>
<tr>
<td>\bit</td>
<td>34</td>
</tr>
<tr>
<td>\boldmath</td>
<td>15</td>
</tr>
<tr>
<td>bracket-ambiguous-numbers</td>
<td>8</td>
</tr>
<tr>
<td>bracket-negative-numbers</td>
<td>8</td>
</tr>
<tr>
<td>bracket-unit-denominator</td>
<td>28</td>
</tr>
<tr>
<td>\byte</td>
<td>34</td>
</tr>
<tr>
<td>\C</td>
<td>33</td>
</tr>
<tr>
<td>\cancel</td>
<td>22, 28</td>
</tr>
<tr>
<td>\candela</td>
<td>26</td>
</tr>
<tr>
<td>\centi</td>
<td>26</td>
</tr>
<tr>
<td>\cm</td>
<td>31</td>
</tr>
<tr>
<td>\color</td>
<td>6</td>
</tr>
<tr>
<td>\color</td>
<td>15</td>
</tr>
<tr>
<td>\exbi</td>
<td>32</td>
</tr>
<tr>
<td>\exbi</td>
<td>34</td>
</tr>
<tr>
<td>\familydefault</td>
<td>14</td>
</tr>
<tr>
<td>\farad</td>
<td>27</td>
</tr>
<tr>
<td>\femto</td>
<td>26</td>
</tr>
<tr>
<td>\fF</td>
<td>33</td>
</tr>
<tr>
<td>\fH</td>
<td>33</td>
</tr>
<tr>
<td>\femto</td>
<td>26</td>
</tr>
<tr>
<td>\fill-angle-degrees</td>
<td>1</td>
</tr>
<tr>
<td>\fill-angle-minutes</td>
<td>1</td>
</tr>
<tr>
<td>\fill-angle-seconds</td>
<td>2</td>
</tr>
<tr>
<td>\fixed-exponent</td>
<td>9</td>
</tr>
<tr>
<td>\empty</td>
<td>6</td>
</tr>
<tr>
<td>\ensuremath</td>
<td>7, 14</td>
</tr>
<tr>
<td>\ev</td>
<td>32</td>
</tr>
<tr>
<td>evaluate-expression</td>
<td>9</td>
</tr>
<tr>
<td>\exa</td>
<td>26</td>
</tr>
<tr>
<td>\exbi</td>
<td>34</td>
</tr>
<tr>
<td>\expONENT</td>
<td>9</td>
</tr>
<tr>
<td>\expONENT-mode</td>
<td>9</td>
</tr>
<tr>
<td>\expression</td>
<td>9</td>
</tr>
<tr>
<td>extract-mass-in-kilograms</td>
<td>29</td>
</tr>
<tr>
<td>\fH</td>
<td>33</td>
</tr>
<tr>
<td>\familydefault</td>
<td>14</td>
</tr>
<tr>
<td>\farad</td>
<td>27</td>
</tr>
<tr>
<td>\femto</td>
<td>26</td>
</tr>
<tr>
<td>\fF</td>
<td>33</td>
</tr>
<tr>
<td>\fH</td>
<td>33</td>
</tr>
<tr>
<td>\femto</td>
<td>26</td>
</tr>
<tr>
<td>\fill-angle-degrees</td>
<td>1</td>
</tr>
<tr>
<td>\fill-angle-minutes</td>
<td>1</td>
</tr>
<tr>
<td>\fill-angle-seconds</td>
<td>2</td>
</tr>
<tr>
<td>\fixed-exponent</td>
<td>9</td>
</tr>
<tr>
<td>\fmol</td>
<td>32</td>
</tr>
<tr>
<td>\fontfamily</td>
<td>14</td>
</tr>
<tr>
<td>\fnsansfamily</td>
<td>14</td>
</tr>
</tbody>
</table>