siunitx – A comprehensive (SI) units package

Joseph Wright†

Released 2023-10-07

Contents

I  siunitx-angle – Formatting angles 1
   1  Formatting angles 1
      1.1  Key–value options ......................................... 1

II  siunitx-compound – Compound numbers and quantities 3

III  siunitx-locale – Localisation 7

IV  siunitx-number – Parsing and formatting numbers 8
   1  Formatting numbers 8
      1.1  Key–value options ......................................... 10

V  siunitx-print – Printing material with font control 16
   1  Printing quantities 16
      1.1  Key–value options ......................................... 17

VI  siunitx-quantity – Quantities 20

VII  siunitx-symbol – Symbol-related settings 21

VIII  siunitx-table – Formatting numbers in tables 22

*This file describes v3.3.7, last revised 2023-10-07.
†E-mail: joseph.wright@morningstar2.co.uk
Part I

\textbf{siunitx-angle} – Formatting angles

1 Formatting angles

\texttt{\textbackslash siunitx\_angle:n} \hspace{1em} \texttt{\textbackslash siunitx\_angle:x}
\texttt{\textbackslash siunitx\_angle:nnn} \hspace{1em} \texttt{\textbackslash siunitx\_angle:xxx}

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

1.1 Key–value options

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

\texttt{\textbackslash angle\_mode} \hspace{1em} \texttt{\textbackslash angle\_mode = (\texttt{choice})}

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{\textbackslash siunitx\_angle:n} and \texttt{\textbackslash siunitx\_angle:nnn}) is maintained. The standard setting is \texttt{input}.

\texttt{\textbackslash angle\_symbol\_degree} \hspace{1em} \texttt{\textbackslash angle\_symbol\_degree = (\texttt{symbol})}
\texttt{\textbackslash angle\_symbol\_minute} \hspace{1em} \texttt{\textbackslash angle\_symbol\_minute}
\texttt{\textbackslash angle\_symbol\_second} \hspace{1em} \texttt{\textbackslash angle\_symbol\_second}

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

\texttt{\textbackslash angle\_symbol\_over\_decimal} \hspace{1em} \texttt{\textbackslash angle\_symbol\_over\_decimal = true|false}

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

\texttt{\textbackslash arc\_separator} \hspace{1em} \texttt{\textbackslash arc\_separator = (\texttt{separator})}

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

\texttt{\textbackslash fill\_arc\_degrees} \hspace{1em} \texttt{\textbackslash fill\_arc\_degrees = true|false}

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

\texttt{\textbackslash fill\_arc\_minutes} \hspace{1em} \texttt{\textbackslash fill\_arc\_minutes = true|false}

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

**fill-arc-seconds**

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

**number-angle-product**

Inserted between the value of an angle and the unit (degree, minute or second component). The standard setting is `,`.
Part II
siunitx-compound – Compound numbers and quantities

\siunitx_compound_number:n \siunitx_compound_number:n \{\{entries\}\}

Prints a set of numbers in the \{\{entries\}\}, each of which should be given as a \{balanced text\}. Unlike \siunitx_number_list:nn, this function may semantically take any form.

\siunitx_compound_quantity:nn \siunitx_compound_quantity:nn \{\{entries\}\} \{\{unit\}\}

Prints a set of quantities in the \{\{entries\}\}, each of which should be given as a \{balanced text\}. Unlike \siunitx_quantity_list:nn, this function may semantically take any form.

\siunitx_number_list:nn \siunitx_number_list:nn \{\{entries\}\}

Prints the list of numbers in the \{\{entries\}\}, each of which should be given as a \{balanced text\}.

\siunitx_quantity_list:nn \siunitx_quantity_list:nn \{\{entries\}\} \{\{unit\}\}

Prints the list of quantities in the \{\{entries\}\}, each of which should be given as a \{balanced text\}.

\siunitx_number_product:n \siunitx_number_product:n \{\{entries\}\}

Prints the series of numbers in the \{\{entries\}\}, each of which should be given as a \{balanced text\}.

\siunitx_quantity_product:nn \siunitx_number_product:n \{\{entries\}\} \{\{unit\}\}

Prints the series of quantities in the \{\{entries\}\}, each of which should be given as a \{balanced text\}.

\siunitx_number_range:nn \siunitx_number_range:nn \{\{start\}\} \{\{end\}\}

Prints the range of numbers from the \{\{start\}\} to the \{\{end\}\}.

\siunitx_quantity_range:nn \siunitx_number_range:nn \{\{start\}\} \{\{end\}\} \{\{unit\}\}

Prints the range of quantities from the \{\{start\}\} to the \{\{end\}\}.

\l_siunitx_list_separator_pair_tl \l_siunitx_list_separator_tl \l_siunitx_list_separator_final_tl

Separators for lists of numbers and quantities.
\l\_siunitx\_range\_phrase\_tl

Phrase (or similar) used between limits of a range.

\begin{verbatim}
\compoundboundarymode \compoundboundarymode = \text{number|text}
\end{verbatim}

Choice which determines whether the material at the start and end of a compound quantity are typeset a \textbf{number} or as \textbf{text}; the latter is the standard setting.

\begin{verbatim}
\compoundcloseboundary \compoundcloseboundary = (tokens)
\compoundopenboundary \compoundopenboundary = (tokens)
\end{verbatim}

Literals which are inserted at the opening and closing boundary of a compound quantity; they are not used when the number of items is one. The standard settings set these empty.

\begin{verbatim}
\compoundclosebracket \compoundclosebracket = (token)
\compoundopenbracket \compoundopenbracket = (token)
\end{verbatim}

Literals containing the tokens inserted at the start and end of a compound value when \texttt{compounds-units} is set to \texttt{bracket}. The standard settings are ( and ).

\begin{verbatim}
\compoundexponents \compoundexponents = \text{combine|combine-bracket|individual}
\end{verbatim}

\begin{verbatim}
\compoundfinalseparator \compoundfinalseparator = (text)
\end{verbatim}

\begin{verbatim}
\compoundindependenmprefix \compoundindependenmprefix = \text{true|false}
\end{verbatim}

Switch which determines whether unit prefixes are calculated independently when units are repeated. The standard setting is \textbf{false}.

\begin{verbatim}
\compoundpairseparator \compoundpairseparator = (text)
\compoundseparator \compoundseparator = (text)
\end{verbatim}

\begin{verbatim}
\compoundseparatormode \compoundseparatormode = \text{number|text}
\end{verbatim}

Choice which determines whether the separators between components of compound quantity are typeset a \textbf{number} or as \textbf{text}; the latter is the standard setting.

\begin{verbatim}
\compoundunits \compoundunits = \text{bracket|repeat|single}
\end{verbatim}

\begin{verbatim}
\listclosebracket \listclosebracket = (token)
\listopenbracket \listopenbracket = (token)
\end{verbatim}

Literals containing the tokens inserted at the start and end of a list when \texttt{list-units} is set to \texttt{bracket}. The standard settings are ( and ).

\begin{verbatim}
\listexponents \listexponents = \text{combine|combine-bracket|individual}
\end{verbatim}
list-final-separator  list-final-separator = (text)

list-independent-prefix  list-independent-prefix = true|false
Switch which determines whether unit prefixes are calculated independently when units are repeated. The standard setting is false.

list-pair-separator  list-pair-separator = (text)

list-separator  list-separator = (text)

list-units  list-units = bracket|repeat|single

product-close-bracket  product-close-bracket = ⟨token⟩
product-open-bracket  product-open-bracket = ⟨token⟩
Literals containing the tokens inserted at the start and end of a product when product-units is set to bracket. The standard settings are ( and ).

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

product-independent-prefix  product-independent-prefix = true|false
Switch which determines whether unit prefixes are calculated independently when units are repeated. The standard setting is false.

product-mode  product-mode = phrase|choice

product-phrase  product-phrase = ⟨text⟩

product-symbol  product-symbol = ⟨symbol⟩

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

range-close-bracket  range-close-bracket = ⟨token⟩
range-open-bracket  range-open-bracket = ⟨token⟩
Literals containing the tokens inserted at the start and end of a range when range-units is set to bracket. The standard settings are ( and ).

range-independent-prefix  range-independent-prefix = true|false
Switch which determines whether unit prefixes are calculated independently when units are repeated. The standard setting is false.
range-open-phrase  range-open-phrase = (text)

Literal containing the material to be inserted at the start of a range. The standard setting is empty.

range-phrase  range-phrase = (text)

Literal containing the material to be inserted between the start and end of a range. The standard setting contains the word to inside the \text command, along with appropriate spacing commands to allow this material to work in both math and text typesetting modes.

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

\textbf{siunitx-locale} – Localisation

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

\begin{verbatim}
locale locale = \langle locale \rangle
\end{verbatim}

Selects the \texttt{(locale)} used to apply standard settings for other keys, principally \texttt{exponent-product}, \texttt{inter-unit-product} and \texttt{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 \(\mp\), \(\pm\), \(\ll\), \(\le\), \(\gg\) and \(\ge\) are used to replace two-character input; \(\pm\) is also required for the output of uncertainties. The standard settings require \(\times\). For the display of colored negative numbers, the command \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 \siunitx_number_output:NN). Such a color will be the first part of the result, meaning that a test for an initial \color and following brace group may be used to detect/remove/adjust this part.

1 Formatting numbers

\begin{verbatim}
\siunitx_number_parse:nN ⟨number⟩ ⟨tl var⟩
\end{verbatim}

Parses the number and stores the resulting internal representation in the \(⟨tl\ var⟩\). The parsing is influenced by the various key–value settings for numerical input. The \(⟨number⟩\) 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 \(⟨tl\ var⟩\).

The structure of a valid number is:

\{(comparator)\} \{(sign)\} \{(integer)\} \{(decimal)\} \{(uncertainty)\} \{(exponent sign)\} \{(exponent)\}

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 either the \(⟨integer⟩\) and \(⟨exponent⟩\) parts. If the number is invalid, or if number parsing is disabled, the result will be an entirely empty \(⟨tl\ var⟩\).

The structure of a valid number is:

\{(comparator)\} \{(sign)\} \{(integer)\} \{(decimal)\} \{(uncertainty)\} \{(exponent sign)\} \{(exponent)\}

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 either the \(⟨integer⟩\) and \(⟨exponent⟩\) parts. If the number is invalid, or if number parsing is disabled, the result will be an entirely empty \(⟨tl\ var⟩\).

The structure of a valid number is:

\{(comparator)\} \{(sign)\} \{(integer)\} \{(decimal)\} \{(uncertainty)\} \{(exponent sign)\} \{(exponent)\}

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 either the \(⟨integer⟩\) and \(⟨exponent⟩\) parts. If the number is invalid, or if number parsing is disabled, the result will be an entirely empty \(⟨tl\ var⟩\).

\begin{itemize}
  \item \(\mathbb{S}\) A single symmetrical uncertainty (e.g. a statistical standard uncertainty). The data item here is a single value representing the uncertainty in the least-significant digits.
  \item \(\mathbb{A}\) A single unsymmetrical uncertainty. The data item here contains two brace groups, each using the same least-significant digit approach as the \(\mathbb{S}\) type. The positive component is given first and the negative second, and neither has a sign.
  \item A combination of \(\mathbb{S}\) and \(\mathbb{A}\) 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 \(⟨decimal⟩\) part should contain \empty.
Applies a set of number processing operations to the \textit{internal number} stored in \texttt{tl var1}, \textit{viz.} in order

\begin{enumerate}
\item Dropping uncertainty
\item Converting to scientific mode (or similar)
\item Rounding
\item Dropping zero decimal part
\item Forcing a minimum number of digits
\end{enumerate}

with the result stored in \texttt{tl var2}.

\begin{Verbatim}
\texttt{\\textsf{\textbackslash siunitx\_number\_format:nN}} \{\texttt{\texttt{\textbackslash number}}\}\ \texttt{\texttt{\textbackslash tl \ texttt{\ var}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:n}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:cc}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:NN}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:nN}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:cc}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:c}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:n}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\texttt{\texttt{\textbackslash siunitx\_number\_format:cc}} \texttt{\{\texttt{\textbackslash number}\}\ \texttt{\texttt{\textbackslash marker}}}  \\
\end{Verbatim}

Formats the \textit{(number)} (in the \texttt{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 \textit{formatting} does not apply any manipulation (processing) to the number. This function is usable in an \texttt{e-} or \texttt{x-type expansion}, and further uncontrolled expansion is prevented by appropriate use of \texttt{\exp\_not:n} internally.

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

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

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

\begin{Verbatim}
\texttt{\texttt{\textbackslash siunitx\_number\_format:nN}} \{\texttt{\texttt{\textbackslash number}}\}\ \texttt{\texttt{\textbackslash tl \ texttt{\ var}}}  \\
\end{Verbatim}

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 \texttt{x-type expansion} to place the result in the \texttt{\texttt{\textbackslash tl \ texttt{\ var}}}.

If \texttt{\textbackslash l\_siunitx\_number\_parse\_bool if false}, the input is simply stored inside the \texttt{\texttt{\textbackslash tl \ texttt{\ var}}} inside \texttt{\texttt{\textbackslash ensuremath}}.
Adjusts the exponent of the \textit{number} (in internal format) by the \textit{fp expr} and leaves the result in the input stream.

Replaces all multi-token signs and comparators in the \textit{tl var} with their single-token equivalents. Replaces any active hyphen tokens with non-active versions.

Determines if the \textit{tokens} form a valid number which can be fully parsed by \texttt{siunitx}.

Determines if the \textit{token} 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.

\subsection*{1.1 Key-value options}

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

\texttt{bracket-ambiguous-numbers} \hspace{1em} \texttt{bracket-ambiguous-numbers = true|false}

\texttt{bracket-negative-numbers} \hspace{1em} \texttt{bracket-negative-numbers = true|false}

\texttt{drop-exponent} \hspace{1em} \texttt{drop-exponent = true|false}
drop-uncertainty = true|false

drop-zero-decimal = true|false

evaluate-expression = true|false

exponent-base = \langle base \rangle

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

exponent-product = \langle symbol \rangle

expression = \langle expression \rangle

fixed-exponent = \langle exponent \rangle

digit-group-number = \langle integer \rangle

\text{digit-group-size}
\text{digit-group-first-size}
\text{digit-group-other-size}

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

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

`group-separator = ⟨symbol⟩`

Sets the symbol inserted between groups of digits. The standard setting is a thin space (\,).

**input-close-uncertainty**

`input-close-uncertainty = ⟨tokens⟩`

**input-comparators**

`input-comparators = ⟨tokens⟩`

**input-close-uncertainty**

`input-close-uncertainty = ⟨tokens⟩`

**input-decimal-markers**

`input-decimal-markers = ⟨tokens⟩`

**input-digits**

`input-digits = ⟨tokens⟩`

**input-exponent-markers**

`input-exponent-markers = ⟨tokens⟩`

**input-open-uncertainty**

`input-open-uncertainty = ⟨tokens⟩`

**input-signs**

`input-signs = ⟨tokens⟩`

**input-uncertainty-signs**

`input-uncertainty-signs = ⟨tokens⟩`

**input-uncertainty-divider**

`input-uncertainty-divider = ⟨tokens⟩`

**minimum-decimal-digits**

`minimum-decimal-digits = ⟨min⟩`

**minimum-integer-digits**

`minimum-integer-digits = ⟨min⟩`

**negative-color**

`negative-color = ⟨color⟩`

**output-close-uncertainty**

`output-close-uncertainty = ⟨symbol⟩`

**output-decimal-marker**

`output-decimal-marker = ⟨symbol⟩`

**output-open-uncertainty**

`output-open-uncertainty = ⟨symbol⟩`
parse-numbers  parse-numbers = true|false

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

print-zero-exponent  print-zero-exponent = true|false

print-zero-integer  print-zero-integer = true|false

retain-explicit-plus  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  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  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  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-direction  round-direction = down|nearest|up

Choice which determines how values values are rounded. The setting up means that the value is always rounded away from zero, whereas the setting down means that the value will be rounded toward zero. The setting nearest means that the value will be rounded to the nearest (either up or down), taking account of the setting of round-half. The standard setting is nearest.

round-half  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  
`round-minimum = \langle \text{\texttt{min}} \rangle`  
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  
`round-mode = \langle \text{\texttt{figures|none|places|uncertainty}} \rangle`  
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 = \langle \text{\texttt{true|false}} \rangle`  
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 = \langle \text{\texttt{precision}} \rangle`  
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 = \langle \text{\texttt{true|false}} \rangle`  
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 = \langle \text{\texttt{true|false}} \rangle`  

uncertainty-descriptor-mode  
`uncertainty-descriptor = \langle \text{\texttt{brace|brace-separator|separator|subscript}} \rangle`  
Selects how uncertainty descriptors are formatted: a choice from the options `brace`, `text` and `subscript`. The option `brace` wraps the descriptor in parenthesis, `brace-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 `brace-separator`.

uncertainty-descriptor-separator  
`uncertainty-descriptor-separator = \langle \text{\texttt{separator}} \rangle`  
Separator inserted between the uncertainty and descriptor when one is required by `uncertainty-descriptor-mode`. The standard setting is $\backslash$. 

uncertainty-descriptors  
`uncertainty-descriptors = \langle \text{\texttt{clist}} \rangle`  
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-round-direction  uncertainty-round-direction = down|nearest|up

Choice which determines how uncertainty values are rounded. The setting up means that the uncertainty is always rounded away from zero, whereas the setting down means that the uncertainty will be rounded toward zero. The setting nearest means that the uncertainty will be rounded to the nearest (either up or down), taking account of the setting of round-half. The standard setting is nearest.

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.
Part V
siunitx-print – Printing material with font control

1 Printing quantities

This submodule is focussed 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 auxiliary font selection commands are available, in particular

- \bfseries,
- \mathrm,
- \mathversion,
- \fontfamily,
- \fontseries,
- \fontshape,
- \familydefault,
- \seriesdefault,
- \shapedefault and
- \selectfont.

It also requires the standard \LaTeX auxiliary kernel commands

- \ensuremath,
- \mbox,
- \textsubscript and
- \textsuperscript

for printing in text mode. The following packages are also required to provide the functionality detailed.

- color: support for color using \textcolor
- textcomp: \textminus, \textpm, \texttimes and \textcenteredperiod for printing in text mode
- amstext: the \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 \textit{required} to exist.

\verbatim
\siunitx_print_number:n \{\texttt{\textbackslash{material}}} \}
\siunitx_print_unit:n \{\texttt{\textbackslash{material}}} \}
\siunitx_print_match:n \{\texttt{\textbackslash{material}}} \}
\siunitx_print_math:n \{\texttt{\textbackslash{material}}} \}
\siunitx_print_text:n \{\texttt{\textbackslash{material}}} \}
\verbatim

Prints the \texttt{\textbackslash{material}} according the the prevailing settings for the submodule as applicable to the \texttt{\textbackslash{type}} of content \texttt{(number or unit)}. The \texttt{\textbackslash{material}} 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 	exttt{^} and 	exttt{\_} will typically be used to indicate super- and subscripts, respectively. These elements will be correctly handled when printing for example using \texttt{\mathsf} in math mode, or using only text fonts. No printing takes place if the \texttt{\textbackslash{material}} is entirely empty after a single expansion.

\verbatim
\siunitx_print_match:n \{\texttt{\textbackslash{material}}} \}
\siunitx_print_math:n \{\texttt{\textbackslash{material}}} \}
\siunitx_print_text:n \{\texttt{\textbackslash{material}}} \}
\verbatim

Prints the \texttt{\textbackslash{material}} as described for \texttt{\textbackslash{siunitx_print_...:n}} but with a fixed text or math mode output. The printing does \textit{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. No printing takes place if the \texttt{\textbackslash{material}} is entirely empty after a single expansion.

\subsection{1.1 Key–value options}

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

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

\verbatim
\verbatim
\verbatim

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

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 = ⟨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 = 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 = ⟨cmd⟩
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-\text{print_...n is called. The math and text options choose the relevant \TeX mode for printing. The standard setting is math.}

**series-version-mapping**

series-version-mapping / ⟨weight⟩ = ⟨version⟩

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

- ul
- el
- l
- sl
- m
- sb
- b
- eb
- ub

As standard, the m weight maps to normal math version whilst all of the b weights map to bold and all of the l weights map to light.
Part VI

\textbf{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.

\begin{verbatim}
\siunitx_quantity:nn \siunitx_quantity:nn \langle number \rangle \{ \langle unit \rangle \}
\end{verbatim}

Parses the \langle number \rangle and the \langle unit \rangle as detailed for \texttt{siunitx_number_parse:nN} and \texttt{siunitx_unit_format:nN}, the prints the results using \texttt{siunitx_print_unit:n}.

\begin{verbatim}
\siunitx_quantity_print:nn \siunitx_quantity_print:nn \langle number \rangle \{ \langle unit \rangle \}
\end{verbatim}

\begin{verbatim}
\siunitx_quantity_print:nn \siunitx_quantity_print:nn \langle number \rangle \{ \langle unit \rangle \}
\end{verbatim}

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{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 \texttt{combine-exponent}, \texttt{extract-exponent} and \texttt{input}. The option \texttt{combine-exponent} combines any exponent from the number with the prefix of the first unit, and prints the updated prefix. The option \texttt{extract-exponent} removes all prefixes from the unit, and combines them with the exponent of number. The option \texttt{input} prints prefixes and exponent as given in the source. The standard setting is \texttt{input}.

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

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

separate-uncertainty-units separate-uncertainty-units = bracketrepeat|single

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

\texttt{siunitx-symbol} – Symbol-related settings

This sub-module exposes no API.
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\tablular cell construct featuring \ignorespaces. Therefore, if used outside of a \LaTeX\tablular, it is necessary to provide this token.

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

Collects the \{preamble\} and \{content\} 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 \keys siunitx tree.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>table-align-comparator</td>
<td>true/false</td>
<td>Switch which determines whether alignment of comparators is attempted within table cells. The standard setting is true.</td>
</tr>
<tr>
<td>table-align-exponent</td>
<td>true/false</td>
<td>Switch which determines whether alignment of exponents is attempted within table cells. The standard setting is true.</td>
</tr>
<tr>
<td>table-align-text-after</td>
<td>true/false</td>
<td>Switch which determines whether alignment of text falling after a number is attempted within table cells. The standard setting is true.</td>
</tr>
<tr>
<td>table-align-text-before</td>
<td>true/false</td>
<td>Switch which determines whether alignment of text falling before a number is attempted within table cells. The standard setting is true.</td>
</tr>
<tr>
<td>table-align-uncertainty</td>
<td>true/false</td>
<td>Switch which determines whether alignment of separated uncertainty values is attempted within table cells. The standard setting is true.</td>
</tr>
</tbody>
</table>
**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-model-setup**  
`table-model-setup = ⟨commands⟩`

Additional commands to be inserted when using the `table-format` to create a model for alignment of cells. Typically this will be used to handle variable-width fonts in columns. The standard setting is empty.

**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`. 

---

23
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

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

\section{Formatting units}

\begin{verbatim}
\texttt{\textbackslash siunitx\_unit\_format:nN \{\langle units\rangle\} \{tl var\}}
\end{verbatim}

This function converts the input \texttt{\langle units\rangle} into a processed \texttt{\{tl var\}} 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}
\texttt{\textbackslash siunitx\_unit\_format:nN \{\kilo \textbackslash metre \textbackslash per \textbackslash second\} \l_tmpa_tl}
\end{verbatim}

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

\begin{verbatim}
\texttt{\textbackslash mathrm\{km\}\textbackslash ,\textbackslash mathrm\{s\}^{-1}}
\end{verbatim}
This function formats the \( \langle \text{units} \rangle \) in the same way as described for \texttt{\siunitx_unit_format:nN}. 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 \( \langle \text{fp var} \rangle \).

For example,

\begin{verbatim}
\siunitx_unit_format_extract_prefixes:nNN { \kilo \metre \per \second }
\end{verbatim}

will, with standard settings, result in \( \l_tmpa_tl \) being set to

\begin{verbatim}
\texttt{\siunitx_unit_format_extract_prefixes:nNN { \kilo \metre \per \second } \l_tmpa_tl \l_tmpa_fp}
\end{verbatim}

\[ \texttt{\l_tmpa_tl} \] will, with standard settings, result in \( \text{m} \, \text{s}^{-1} \) with \( \texttt{\l_tmpa_fp} \) 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 \( \langle \text{units} \rangle \) in the same way as described for \texttt{\siunitx_unit_format:nN}. The \( \langle \text{exponent} \rangle \) is combined with any prefix for the first unit of the \( \langle \text{units} \rangle \), and an updated prefix is introduced.

For example,

\begin{verbatim}
\siunitx_unit_format_combine_exponent:nnN { \metre \per \second } { 3 } \l_tmpa_tl
\end{verbatim}

will, with standard settings, result in \( \l_tmpa_tl \) being set to

\begin{verbatim}
\texttt{\siunitx_unit_format_combine_exponent:nnN { \metre \per \second } { 3 } \l_tmpa_tl}
\end{verbatim}

\[ \texttt{\l_tmpa_tl} \] will, with standard settings, result in \( \text{km} \, \text{s}^{-1} \)

These function formats the \( \langle \text{units} \rangle \) in the same way as described for \texttt{\siunitx_unit_format:nN}. The units are multiplied by the \( \langle \text{factor} \rangle \), and further processing takes place as previously described.

For example,

\begin{verbatim}
\siunitx_unit_format_multiply:nnN { \metre \per \second } { 3 } \l_tmpa_tl
\end{verbatim}

will, with standard settings, result in \( \l_tmpa_tl \) being set to

\begin{verbatim}
\texttt{\siunitx_unit_format_multiply:nnN { \metre \per \second } { 3 } \l_tmpa_tl}
\end{verbatim}

\[ \texttt{\l_tmpa_tl} \] will, with standard settings, result in \( \text{km} \, \text{s}^{-3} \)
2 Defining symbolic units

\siunitx_declare_prefix:Nnn \siunitx_declare_prefix:Nnn (prefix) {⟨power⟩} {⟨symbol⟩}
\siunitx_declare_prefix:Nx

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

\siunitx_declare_prefix:Nn
\siunitx_declare_power:NNn
\siunitx_declare_qualifier:Nn
\siunitx_declare_unit:Nn
\siunitx_declare_unit:Nnn
\siunitx_declare_unit:Nxn

The font function which is applied to the text of units when constructing formatted units: set by font-command.

\l_siunitx_unit_font_tl
\_l\_siunitx\_unit\_fraction\_tl

The fraction function which is applied when constructing fractional units: set by `fraction-command`.

\_l\_siunitx\_unit\_symbolic\_seq

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.

\_l\_siunitx\_unit\_seq

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 \langle unit(s) \rangle`

Applies any unit-specific options set up using `\siunitx\_declare\_unit:Nnn`. This allows their 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:`

`\siunitx\_unit\_pdfstring\_context:`
\langle Expansion context \rangle \langle units \rangle
`\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 `meter` is defined as an alias for `metre` as the former spelling is common in the US (although the latter is the official spelling).

The base unit `kilogram` is defined using an SI prefix: as such the (derived) unit `gram` is required by the module to correctly produce output for the `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 `kilo` prefix is required to define the base `kilogram` unit. Also note the two spellings available for `deca`/`deka`.

\begin{itemize}
\item \texttt{gram}
\item \texttt{deca}
\item \texttt{deka}
\item \texttt{hecto}
\item \texttt{kilo}
\item \texttt{giga}
\item \texttt{tera}
\item \texttt{peta}
\item \texttt{exa}
\item \texttt{zetta}
\item \texttt{yotta}
\item \texttt{ronna}
\item \texttt{quetta}
\end{itemize}
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 \textit{degreeCelsius}, and that this unit name includes “degree”.

\begin{itemize}
\item \texttt{becquerel}
\item \texttt{degreeCelsius}
\item \texttt{coulomb}
\item \texttt{farad}
\item \texttt{gray}
\item \texttt{hertz}
\item \texttt{henry}
\item \texttt{joule}
\item \texttt{katal}
\item \texttt{lumen}
\item \texttt{lux}
\item \texttt{newton}
\item \texttt{ohm}
\item \texttt{pascal}
\item \texttt{radian}
\item \texttt{siemens}
\item \texttt{sievert}
\item \texttt{steradian}
\item \texttt{tesla}
\item \texttt{volt}
\item \texttt{watt}
\item \texttt{weber}
\end{itemize}

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 l and 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”).

\begin{itemize}
\item \texttt{astronomicalunit}
\item \texttt{bel}
\item \texttt{dalton}
\item \texttt{day}
\item \texttt{decibel}
\item \texttt{electronvolt}
\item \texttt{hectare}
\item \texttt{hour}
\item \texttt{litre}
\item \texttt{liter}
\item \texttt{neper}
\item \texttt{minute}
\item \texttt{tonne}
\end{itemize}

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.

\begin{itemize}
\item \texttt{percent}
\item \texttt{square} \texttt{(prefix)} \texttt{(unit)}
\item \texttt{cubic} \texttt{(prefix)} \texttt{(unit)}
\end{itemize}

The mathematical concept of percent, usable with the SI as detailed in Section 5.4.7 of the SI Brochure.

Pre-defined unit powers which apply to the next \texttt{(prefix)/\texttt{(unit)}} combination.
\squared \ (prefix) \ (unit) \squared
\cubed \ (prefix) \ (unit) \cubed

Pre-defined unit powers which apply to the preceding \(\text{(prefix)}/\text{(unit)}\) combination.

\per \ (prefix) \ (unit) \ (power)

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

\cancel \ (prefix) \ (unit) \ (power)

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

\highlight \ (color) \ (prefix) \ (unit) \ (power)

Indicates that the next \(\text{(prefix)}/\text{(unit)}/\text{(power)}\) combination should be highlighted in the specified \texttt{(color)}. In the parsed output, the entire unit combination will be given as the argument to a function \texttt{\textcolor}, which is assumed to be available at a higher level. In literal mode, the same higher-level \texttt{\textcolor} will be applied to the next token. It is the responsibility of the calling code to provide an appropriate definition for \texttt{\textcolor} outside of the scope of the unit parser.

\of \ (prefix) \ (unit) \ (power) \ (qualifier)

Indicates that the \texttt{(qualifier)} applies to the current \(\text{(prefix)}/\text{(unit)}/\text{(power)}\) combination. In parsed mode, the display of the result will depend upon module options. In literal mode, the \texttt{(qualifier)} will be printed in parentheses following the preceding \texttt{(unit)} and a full-width space.

\raiseto \ (power) \ (prefix) \ (unit)
\tothe \ (prefix) \ (unit) \ (power)

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

5.1 Key–value options

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

\texttt{\bracket-unit-denominator}
\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 \texttt{\per-mode} as \texttt{repeated-symbol} or \texttt{symbol}). The standard setting is \texttt{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 \!.
power-half-as-sqrt

\[ \text{power-half-as-sqrt} = \text{true} | \text{false} \]

Used to determine whether a power of exactly half is converted to $\sqrt{\text{...}}$ in the output. The standard setting is $\text{false}$.

qualifier-mode

\[ \text{qualifier-mode} = \text{bracket} | \text{combine} | \text{phrase} | \text{subscript} \]

Selects how qualifiers are formatted: a choice from the options $\text{bracket}$, $\text{combine}$, $\text{phrase}$ and $\text{subscript}$. The option $\text{bracket}$ wraps the qualifier in parenthesis, $\text{combine}$ joins the qualifier with the unit directly, $\text{phrase}$ joins the material using $\text{qualifier-phrase}$ as a link, and $\text{subscript}$ formats the qualifier as a subscript. The standard setting is $\text{subscript}$.

qualifier-phrase

\[ \text{qualifier-phrase} = \langle \text{phrase} \rangle \]

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

sticky-per

\[ \text{sticky-per} = \text{true} | \text{false} \]

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

unit-font-command

\[ \text{unit-font-command} = \langle \text{command} \rangle \]

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 $\text{\LaTeX} \text{\backslash mathrm}$.

References


Part X

siunitx-abbreviations – Abbreviations

\textbf{Abbreviations for currents.}

\textbf{Abbreviations for masses.}

\textbf{Abbreviations for temperature.}

\textbf{Abbreviations for lengths.}

\textbf{Abbreviations for times.}

\textbf{Abbreviations for frequencies.}
<table>
<thead>
<tr>
<th>Abbreviations for moles.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\mol</td>
<td></td>
</tr>
<tr>
<td>\fmol</td>
<td></td>
</tr>
<tr>
<td>\pmol</td>
<td></td>
</tr>
<tr>
<td>\nmol</td>
<td></td>
</tr>
<tr>
<td>\umol</td>
<td></td>
</tr>
<tr>
<td>\mmol</td>
<td></td>
</tr>
<tr>
<td>\kmol</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviations for potentials.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\V</td>
<td></td>
</tr>
<tr>
<td>\pV</td>
<td></td>
</tr>
<tr>
<td>\nV</td>
<td></td>
</tr>
<tr>
<td>\uV</td>
<td></td>
</tr>
<tr>
<td>\mV</td>
<td></td>
</tr>
<tr>
<td>\kV</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviations for volumes.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\hl</td>
<td></td>
</tr>
<tr>
<td>\l</td>
<td></td>
</tr>
<tr>
<td>\ml</td>
<td></td>
</tr>
<tr>
<td>\ul</td>
<td></td>
</tr>
<tr>
<td>\hlL</td>
<td></td>
</tr>
<tr>
<td>\lL</td>
<td></td>
</tr>
<tr>
<td>\mlL</td>
<td></td>
</tr>
<tr>
<td>\ulL</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviations for powers.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\W</td>
<td></td>
</tr>
<tr>
<td>\nW</td>
<td></td>
</tr>
<tr>
<td>\uW</td>
<td></td>
</tr>
<tr>
<td>\mW</td>
<td></td>
</tr>
<tr>
<td>\kW</td>
<td></td>
</tr>
<tr>
<td>\MW</td>
<td></td>
</tr>
<tr>
<td>\GW</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviations for energies.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\kJ</td>
<td></td>
</tr>
<tr>
<td>\J</td>
<td></td>
</tr>
<tr>
<td>\mJ</td>
<td></td>
</tr>
<tr>
<td>\uJ</td>
<td></td>
</tr>
<tr>
<td>\eV</td>
<td></td>
</tr>
<tr>
<td>\meV</td>
<td></td>
</tr>
<tr>
<td>\keV</td>
<td></td>
</tr>
<tr>
<td>\MeV</td>
<td></td>
</tr>
<tr>
<td>\GeV</td>
<td></td>
</tr>
<tr>
<td>\TeV</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abbreviations for forces.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\N</td>
<td></td>
</tr>
<tr>
<td>\mN</td>
<td></td>
</tr>
<tr>
<td>\kN</td>
<td></td>
</tr>
<tr>
<td>\MN</td>
<td></td>
</tr>
</tbody>
</table>
Abbreviations for pressures.
\(\text{Pa}\) \(\text{kPa}\) \(\text{MPa}\) \(\text{GPa}\)

Abbreviations for resistance.
\(\text{mohm}\) \(\text{kohm}\) \(\text{Mohm}\)

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.
\(\text{C}\) \(\text{nC}\) \(\text{uC}\) \(\text{mC}\)

Abbreviations for magnetic field.
\(\text{T}\) \(\text{nT}\) \(\text{uT}\)

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.

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

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

\bit \byte

Units for bits and bytes.
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

\siunitx_command_create: \siunitx_command_create:
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>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>free-standing-units</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>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>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>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>Switch to determine whether free standing document commands use the xparse package to insert space after the command names. The standard setting is false. When set true, the xparse package will be loaded at the start of the document if not already available.</td>
</tr>
</tbody>
</table>

38
Part XIII

\texttt{siunitx-emulation} – Emulation

This sub-module exposes no API.
# 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>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( )</td>
<td>12, 20, 25</td>
</tr>
<tr>
<td>( \mathbf{A} )</td>
<td>34</td>
</tr>
<tr>
<td>allow-quantity-breaks</td>
<td>20</td>
</tr>
<tr>
<td>( \text{ampere} )</td>
<td>29</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>( \text{arcminute} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{arcsecond} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{as} )</td>
<td>34</td>
</tr>
<tr>
<td>( \text{astronomicalunit} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{at} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{becquerel} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{bel} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{bfseries} )</td>
<td>16</td>
</tr>
<tr>
<td>( \text{bit} )</td>
<td>37</td>
</tr>
<tr>
<td>( \text{boldmath} )</td>
<td>18</td>
</tr>
<tr>
<td>bracket-ambiguous-numbers</td>
<td>10</td>
</tr>
<tr>
<td>bracket-negative-numbers</td>
<td>10</td>
</tr>
<tr>
<td>bracket-unit-denominator</td>
<td>31</td>
</tr>
<tr>
<td>( \text{byte} )</td>
<td>37</td>
</tr>
<tr>
<td>( \mathbf{C} )</td>
<td>36</td>
</tr>
<tr>
<td>( \text{cancel} )</td>
<td>25, 31</td>
</tr>
<tr>
<td>( \text{candela} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{centi} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{cm} )</td>
<td>34</td>
</tr>
<tr>
<td>( \text{color} )</td>
<td>8</td>
</tr>
<tr>
<td>color</td>
<td>17</td>
</tr>
<tr>
<td>compound-boundary-mode</td>
<td>4</td>
</tr>
<tr>
<td>compound-close-boundary</td>
<td>4</td>
</tr>
<tr>
<td>compound-close-bracket</td>
<td>4</td>
</tr>
<tr>
<td>compound-exponents</td>
<td>4</td>
</tr>
<tr>
<td>compound-final-separator</td>
<td>4</td>
</tr>
<tr>
<td>compound-independent-prefix</td>
<td>4</td>
</tr>
<tr>
<td>compound-open-boundary</td>
<td>4</td>
</tr>
<tr>
<td>compound-open-bracket</td>
<td>4</td>
</tr>
<tr>
<td>compound-pair-separator</td>
<td>4</td>
</tr>
<tr>
<td>compound-separator</td>
<td>4</td>
</tr>
<tr>
<td>compound-separator-mode</td>
<td>4</td>
</tr>
<tr>
<td>compound-units</td>
<td>4</td>
</tr>
<tr>
<td>( \text{coulomb} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{cubed} )</td>
<td>31</td>
</tr>
<tr>
<td>( \text{cubic} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{Dalton} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{day} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{dB} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{deci} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{decibel} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{degree} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{degree} \text{Celsius} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{deka} )</td>
<td>29</td>
</tr>
<tr>
<td>digit-group-first-size</td>
<td>11</td>
</tr>
<tr>
<td>digit-group-other-size</td>
<td>11</td>
</tr>
<tr>
<td>digit-group-size</td>
<td>11</td>
</tr>
<tr>
<td>display-per-mode</td>
<td>32</td>
</tr>
<tr>
<td>( \text{dm} )</td>
<td>34</td>
</tr>
<tr>
<td>drop-exponent</td>
<td>10</td>
</tr>
<tr>
<td>drop-uncertainty</td>
<td>11</td>
</tr>
<tr>
<td>drop-zero-decimal</td>
<td>11</td>
</tr>
<tr>
<td>( \text{electronvolt} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{empty} )</td>
<td>8</td>
</tr>
<tr>
<td>\text{ensuremath}</td>
<td>9, 16</td>
</tr>
<tr>
<td>( \text{eV} )</td>
<td>35</td>
</tr>
<tr>
<td>evaluate-expression</td>
<td>11</td>
</tr>
<tr>
<td>( \text{exa} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{exbi} )</td>
<td>37</td>
</tr>
<tr>
<td>exponent-base</td>
<td>11</td>
</tr>
<tr>
<td>exponent-mode</td>
<td>11</td>
</tr>
<tr>
<td>exponent-product</td>
<td>11</td>
</tr>
<tr>
<td>expression</td>
<td>11</td>
</tr>
<tr>
<td>extract-mass-in-kilograms</td>
<td>32</td>
</tr>
<tr>
<td>( \mathbf{F} )</td>
<td>36</td>
</tr>
<tr>
<td>( \text{familydefault} )</td>
<td>16</td>
</tr>
<tr>
<td>( \text{farad} )</td>
<td>30</td>
</tr>
<tr>
<td>( \text{femto} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{femto} )</td>
<td>29</td>
</tr>
<tr>
<td>( \text{fEm} )</td>
<td>36</td>
</tr>
<tr>
<td>( \text{fG} )</td>
<td>34</td>
</tr>
<tr>
<td>( \text{fH} )</td>
<td>36</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>
</tbody>
</table>

40
\zepto \hfill 29 \hspace{1cm} \text{zero-symbol} \hfill 15
\hspace{1cm} \text{zero-decimal-as-symbol} \hfill 15 \hspace{1cm} \zettsymbol \hfill 29