

# siunitx-table – Formatting numbers in tables\*

Joseph Wright<sup>†</sup>

Released 2021-05-31

## 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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> tabular cell construct featuring `\ignorespaces`. Therefore, if used outside of a L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> tabular, it is necessary to provide this token.

---

<code>\siunitx_cell_begin:w</code>	<code>\siunitx_cell_begin:w &lt;preamble&gt; \ignorespaces</code>
<code>\siunitx_cell_end:</code>	<code>&lt;content&gt;</code>

---

`\siunitx_cell_end:`  
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 T<sub>E</sub>X `\halign` template.

### 1.1 Key-value options

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

---

<code>table-align-comparator</code>	<code>table-align-comparator = true false</code>
-------------------------------------	--

---

Switch which determines whether alignment of comparators is attempted within table cells. The standard setting is `true`.

---

<code>table-align-exponent</code>	<code>table-align-exponent = true false</code>
-----------------------------------	--

---

Switch which determines whether alignment of exponents is attempted within table cells. The standard setting is `true`.

---

<code>table-align-text-after</code>	<code>table-align-text-after = true false</code>
-------------------------------------	--

---

Switch which determines whether alignment of text falling after a number is attempted within table cells. The standard setting is `true`.

---

\*This file describes v3.0.8, last revised 2021-05-31.

<sup>†</sup>E-mail: [joseph.wright@morningstar2.co.uk](mailto:joseph.wright@morningstar2.co.uk)

<hr/> <hr/> table-align-text-before	<p>table-align-text-before = true false</p> <p>Switch which determines whether alignment of text falling before a number is attempted within table cells. The standard setting is <b>true</b>.</p>
<hr/> <hr/> table-align-uncertainty	<p>table-align-uncertainty = true false</p> <p>Switch which determines whether alignment of separated uncertainty values is attempted within table cells. The standard setting is <b>true</b>.</p>
<hr/> <hr/> table-alignment	<p>table-alignment = center left right</p> <p>Selects the alignment of all tabular content with the margins of the table cell (or other boundary). See also <b>table-number-alignment</b> and <b>table-text-alignment</b>. The standard setting is <b>center</b>.</p>
<hr/> <hr/> table-alignment-mode	<p>table-alignment-mode = format marker none</p> <p>Selects the method used to align numbers with the desired position in the cell (set by <b>table-alignment</b>). When set to <b>format</b>, a dedicated amount of space is calculated from the <b>table-format</b>. When <b>marker</b> is selected, alignment is carried out symmetrically around the decimal marker. Finally, <b>none</b> switches off all alignment: numbers are parsed and formatted but with no attempt at placement within the cell. The standard setting is <b>marker</b>.</p>
<hr/> <hr/> table-auto-round	<p>table-auto-round = true false</p> <p>Switch which determines whether numbers are rounded to fit within the <b>table-format</b> specification (if possible). The standard setting is <b>false</b>.</p>
<hr/> <hr/> table-column-width	<p>table-column-width = <math>\langle width \rangle</math></p> <p>Sets the width of the table column used for numbers. This is only used when <b>table-fixed-width</b> is <b>true</b>.</p>
<hr/> <hr/> table-fixed-width	<p>table-fixed-width = true false</p> <p>Switch which determines whether a fixed-width column is used for numbers in tables. When <b>true</b>, the width is taken from <b>table-column-width</b>. The standard setting is <b>false</b>.</p>
<hr/> <hr/> table-format	<p>table-format = <math>\langle format \rangle</math></p> <p>Describes the amount of space that should be reserved when <b>table-alignment-mode</b> is set to <b>format</b>. The <math>\langle format \rangle</math> 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, <b>1.2e3</b> 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 <b>+1.2 \pm 1.2</b> would allow for a sign, a number with one integer and two decimal digits and a uncertainty of the same size.</p>
<hr/> <hr/> table-number-alignment	<p>table-number-alignment = center left right</p> <p>Selects the alignment of numerical content with the margins of the table cell (or other boundary). See also <b>table-alignment</b> and <b>table-text-alignment</b>. The standard setting is <b>center</b>.</p>

---

**table-text-alignment**

---

`table-text-alignment = center|left|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`. The standard setting is `center`.

## 2 siunitx-table implementation

Start the DocStrip guards.

```
1 <*package>
```

Identify the internal prefix (L<sup>A</sup>T<sub>E</sub>X3 DocStrip convention): only internal material in this *submodule* should be used directly.

```
2 <@@=siunitx_table>
```

```
\l__siunitx_table_tmp_box
```

Scratch space.

```
\l__siunitx_table_tmp_dim
```

```
3 \box_new:N \l__siunitx_table_tmp_box
```

```
\l__siunitx_table_tmp_tl
```

```
4 \dim_new:N \l__siunitx_table_tmp_dim
```

```
5 \tl_new:N \l__siunitx_table_tmp_tl
```

(End definition for `\l__siunitx_table_tmp_box`, `\l__siunitx_table_tmp_dim`, and `\l__siunitx_table_tmp_tl`.)

### 2.1 Interface functions

```
\l__siunitx_table_text_bool
```

Used to track that a cell is purely text.

```
6 \bool_new:N \l__siunitx_table_text_bool
```

(End definition for `\l__siunitx_table_text_bool`.)

```
\siunitx_cell_begin:w
```

The start and end of the cell need to deal with the possibility of a cell containing only text.

```
\siunitx_cell_end:
```

```
7 \cs_new_protected:Npn \siunitx_cell_begin:w
```

```
8 {
```

```
9   \bool_set_false:N \l__siunitx_table_text_bool
```

```
10   \bool_if:NTF \l_siunitx_number_parse_bool
```

```
11     { \__siunitx_table_collect_begin: }
```

```
12     { \__siunitx_table_direct_begin: }
```

```
13 }
```

```
14 \cs_new_protected:Npn \siunitx_cell_end:
```

```
15 {
```

```
16   \bool_if:NF \l__siunitx_table_text_bool
```

```
17   {
```

```
18     \bool_if:NTF \l_siunitx_number_parse_bool
```

```
19       { \__siunitx_table_collect_end: }
```

```
20       { \__siunitx_table_direct_end: }
```

```
21   }
```

```
22 }
```

(End definition for `\siunitx_cell_begin:w` and `\siunitx_cell_end:`. These functions are documented on page [1](#).)

## 2.2 Collecting tokens

`\l__siunitx_table_collect_tl` Space for tokens.

```
23 \tl_new:N \l__siunitx_table_collect_tl
```

(End definition for `\l__siunitx_table_collect_tl`.)

`\__siunitx_table_collect_begin:`  
`\__siunitx_table_collect_begin:w`

Collecting a tabular cell means doing a token-by-token collection. In previous versions of `siunitx` that was done along with picking out the numerical part, but the code flow ends up very tricky. Here, therefore, we just collect up the unchanged tokens first. The definition of `\cr` is used to allow collection of any tokens inserted after the main content when dealing with the last cell of a row: the “group” around it is needed to avoid issues with the underlying `\halign`. (The approach is based on that in `colcell`.) Whilst the group formed by a cell will normally tidy up `\cr`, we add an extra one as the collected material could be a tabular in itself. We use an auxiliary to fish out the `\ignorespaces` from the template: that has to go to avoid issues with the peek-ahead code (everything before the `#` needs to be read *before* the Appendix D trick gets applied). Some packages add additional tokens before the `\ignorespaces`, which are dealt with by the delimited argument.

```
24 \cs_new_protected:Npn \__siunitx_table_collect_begin:
25 {
26   \group_begin:
27   \tl_clear:N \l__siunitx_table_collect_tl
28   \if_false: { \fi:
29     \cs_set_protected:Npn \cr
30     {
31       \__siunitx_table_collect_loop:
32       \tex_cr:D
33     }
34   \if_false: } \fi:
35   \__siunitx_table_collect_begin:w
36 }
37 \cs_new_protected:Npn \__siunitx_table_collect_begin:w #1 \ignorespaces
38 { \__siunitx_table_collect_loop: #1 }
```

(End definition for `\__siunitx_table_collect_begin:` and `\__siunitx_table_collect_begin:w`.)

`\__siunitx_table_collect_loop:`  
`\__siunitx_table_collect_group:n`  
`\__siunitx_table_collect_token:N`  
`\__siunitx_table_collect_token_aux:N`  
`\__siunitx_table_collect_relax:N`  
`\__siunitx_table_collect_search:NnF`  
`\__siunitx_table_collect_search_aux:Nn`

Collecting up the cell content needs a loop: this is done using a `peek` approach as it’s most natural. (A slower approach is possible using something like the `\text_lowercase:n` loop code.) The set of possible tokens is somewhat limited compared to an arbitrary cell (*cf.* the approach in `colcell`): the special cases are pulled out for manual handling. The flexible lookup approach is more-or-less the same idea as in the kernel `case` functions. The `\relax` special case covers the case where `\\` has been expanded in an empty cell. This has to be an explicit token as we can get the same meaning from `\protect`.

```
39 \cs_new_protected:Npn \__siunitx_table_collect_loop:
40 {
41   \peek_catcode_ignore_spaces:NTF \c_group_begin_token
42   { \__siunitx_table_collect_group:n }
43   { \__siunitx_table_collect_token:N }
44 }
45 \cs_new_protected:Npn \__siunitx_table_collect_group:n #1
46 {
47   \tl_put_right:Nn \l__siunitx_table_collect_tl { {#1} }
```

```

48     \__siunitx_table_collect_loop:
49   }
50 \cs_new_protected:Npn \__siunitx_table_collect_token:N #1
51 {
52   \__siunitx_table_collect_search:NnF #1
53   {
54     \unskip          { \__siunitx_table_collect_loop: }
55     \end             { \tabularnewline \end }
56     \relax          { \__siunitx_table_collect_relax:N #1 }
57     \tabularnewline { \tabularnewline }
58     \siunitx_cell_end: { \siunitx_cell_end: }
59   }
60   { \__siunitx_table_collect_token_aux:N #1 }
61 }
62 \cs_new_protected:Npn \__siunitx_table_collect_token_aux:N #1
63 {
64   \tl_put_right:Nn \l__siunitx_table_collect_tl {#1}
65   \__siunitx_table_collect_loop:
66 }
67 \cs_new_protected:Npn \__siunitx_table_collect_relax:N #1
68 {
69   \str_if_eq:nnTF {#1} { } { \relax }
70   { \relax }
71   { \__siunitx_table_collect_token_aux:N #1 }
72 }
73 \AtBeginDocument
74 {
75   \ifpackageloaded { mdwtab }
76   {
77     \cs_set_protected:Npn \__siunitx_table_collect_token:N #1
78     {
79       \__siunitx_table_collect_search:NnF #1
80       {
81         \@maybe@unskip { \__siunitx_table_collect_loop: }
82         \tab@setcr      { \__siunitx_table_collect_loop: }
83         \unskip         { \__siunitx_table_collect_loop: }
84         \end            { \tabularnewline \end }
85         \relax          { \__siunitx_table_collect_relax:N #1 }
86         \tabularnewline { \tabularnewline }
87         \siunitx_cell_end: { \siunitx_cell_end: }
88       }
89       { \__siunitx_table_collect_token_aux:N #1 }
90     }
91   }
92   { }
93 }
94 \cs_new_protected:Npn \__siunitx_table_collect_search:NnF #1#2#3
95 {
96   \__siunitx_table_collect_search_aux:NNn #1
97   #2
98   #1 {#3}
99   \q_stop
100 }
101 \cs_new_protected:Npn \__siunitx_table_collect_search_aux:NNn #1#2#3

```

```

102 {
103   \token_if_eq_meaning:NNTF #1 #2
104   { \use_i_delimit_by_q_stop:nw {#3} }
105   { \__siunitx_table_collect_search_aux:NNn #1 }
106 }

```

(End definition for \\_\_siunitx\_table\_collect\_loop: and others.)

## 2.3 Separating collected material

The input needs to be divided into numerical tokens and those which appear before and after them. This needs a second loop and validation.

Space for tokens.

```

\l__siunitx_table_before_tl
\l__siunitx_table_number_tl
\l__siunitx_table_after_tl
107 \tl_new:N \l__siunitx_table_before_tl
108 \tl_new:N \l__siunitx_table_number_tl
109 \tl_new:N \l__siunitx_table_after_tl

```

(End definition for \l\_\_siunitx\_table\_before\_tl, \l\_\_siunitx\_table\_number\_tl, and \l\_\_siunitx\_table\_after\_tl.)

At the end of the cell, escape the group and check for expansion. We only do that if the entire content is not a brace group: there is more likely to be problematic content in the case of a header.

```

\__siunitx_table_collect_end:
\__siunitx_table_collect_end:n
\__siunitx_table_collect_end_aux:n
\__siunitx_table_collect_end:w
110 \cs_new_protected:Npn \__siunitx_table_collect_end:
111 {
112   \exp_args:NNV \group_end:
113   \__siunitx_table_collect_end:n \l__siunitx_table_collect_tl
114   \exp_args:NV \__siunitx_table_split:nNNN
115   \l__siunitx_table_collect_tl
116   \l__siunitx_table_before_tl
117   \l__siunitx_table_number_tl
118   \l__siunitx_table_after_tl
119   \tl_if_empty:NTF \l__siunitx_table_number_tl
120   { \__siunitx_table_print_text:V \l__siunitx_table_before_tl }
121   {
122     \__siunitx_table_print:VVV
123     \l__siunitx_table_before_tl
124     \l__siunitx_table_number_tl
125     \l__siunitx_table_after_tl
126   }
127 }
128 \cs_new_protected:Npn \__siunitx_table_collect_end:n #1
129 {
130   \str_if_eq:eeTF { \exp_not:n {#1} }
131   { { \__siunitx_table_collect_end_aux:n {#1} } }
132   { \tl_set:Nn }
133   { \protected@edef }
134   \l__siunitx_table_collect_tl {#1}
135 }
136 \cs_new:Npn \__siunitx_table_collect_end_aux:n #1
137 { \exp_after:wN \__siunitx_table_collect_end:w #1 \q_stop }
138 \cs_new:Npn \__siunitx_table_collect_end:w #1 \q_stop
139 { \exp_not:n {#1} }

```

(End definition for `\__siunitx_table_collect_end:` and others.)

```

\__siunitx_table_split:nNNN
  \__siunitx_table_split_loop:NNN
  \__siunitx_table_split_group:NNNn
  \__siunitx_table_split_token:NNNN
140 \cs_new_protected:Npn \__siunitx_table_split:nNNN #1#2#3#4
141 {
142   \tl_clear:N #2
143   \tl_clear:N #3
144   \tl_clear:N #4
145   \__siunitx_table_split_loop:NNN #2#3#4 #1 \q_recursion_tail \q_recursion_stop
146   \__siunitx_table_split_tidy:N #2
147   \__siunitx_table_split_tidy:N #4
148 }
149 \cs_new_protected:Npn \__siunitx_table_split_loop:NNN #1#2#3
150 {
151   \peek_catcode_ignore_spaces:NTF \c_group_begin_token
152   { \__siunitx_table_split_group:NNNn #1#2#3 }
153   { \__siunitx_table_split_token:NNNN #1#2#3 }
154 }
155 \cs_new_protected:Npn \__siunitx_table_split_group:NNNn #1#2#3#4
156 {
157   \tl_if_empty:NTF #2
158   { \tl_put_right:Nn #1 { {#4} } }
159   { \tl_put_right:Nn #3 { {#4} } }
160   \__siunitx_table_split_loop:NNN #1#2#3
161 }
162 \cs_new_protected:Npn \__siunitx_table_split_token:NNNN #1#2#3#4
163 {
164   \quark_if_recursion_tail_stop:N #4
165   \tl_if_empty:NTF \l__siunitx_table_after_tl
166   {
167     \siunitx_if_number_token:NTF #4
168     { \tl_put_right:Nn #2 {#4} }
169     {
170       \tl_if_empty:NTF #2
171       { \tl_put_right:Nn #1 {#4} }
172       { \tl_put_right:Nn #3 {#4} }
173     }
174   }
175   { \tl_put_right:Nn #3 {#4} }
176   \__siunitx_table_split_loop:NNN #1#2#3
177 }

```

(End definition for `\__siunitx_table_split:nNNN` and others.)

`\__siunitx_table_split_tidy:N` A quick test for the entire content being surrounded by a set of braces: rather than look explicitly, use the fact that a string comparison can detect the same thing. The auxiliary is needed to avoid having to go *via* a :D function (for the expansion behaviour).

```

178 \cs_new_protected:Npn \__siunitx_table_split_tidy:N #1
179 {
180   \tl_if_empty:NF #1
181   { \__siunitx_table_split_tidy:NV #1 #1 }
182 }
183 \cs_new_protected:Npn \__siunitx_table_split_tidy:Nn #1#2

```

```

184 {
185   \str_if_eq:onT { \exp_after:wN { \use:n #2 } } {#2}
186   { \tl_set:No #1 { \use:n #2 } }
187 }
188 \cs_generate_variant:Nn \__siunitx_table_split_tidy:Nn { NV }
(End definition for \__siunitx_table_split_tidy:N and \__siunitx_table_split_tidy:Nn.)

```

## 2.4 Printing numbers in cells: spacing

Getting the general alignment correct in tables is made more complex than one would like by the `colortbl` package. In the original L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> definition, cell material is centred by a construction of the (primitive) form

```

\hfil
#
\hfil

```

which only uses `fil` stretch. That is altered by `colortbl` to broadly

```

\hskip 0pt plus 0.5fill
\kern 0pt
#
\hskip 0pt plus 0.5fill

```

which means there is `fill` stretch to worry about and the kern as well.

`\__siunitx_table_skip:n` To prevent combination of skips, a kern is inserted after each one. This is best handled as a short auxiliary.

```

189 \cs_new_protected:Npn \__siunitx_table_skip:n #1
190 {
191   \skip_horizontal:n {#1}
192   \tex_kern:D \c_zero_skip
193 }
(End definition for \__siunitx_table_skip:n.)

```

`\l__siunitx_table_column_width_dim` Settings which apply to aligned columns in general.

```

\l__siunitx_table_fixed_width_bool
194 \keys_define:nn { siunitx }
195 {
196   table-column-width .dim_set:N =
197     \l__siunitx_table_column_width_dim ,
198   table-fixed-width .bool_set:N =
199     \l__siunitx_table_fixed_width_bool
200 }
(End definition for \l__siunitx_table_column_width_dim and \l__siunitx_table_fixed_width_bool.)

```

`\_siunitx_table_align_center:n` The beginning and end of each table cell have to adjust the position of the content using glue. When `colortbl` is loaded the glue is done in two parts: one for our positioning and one to explicitly override that from the package. Using a two-step auxiliary chain avoids needing to repeat any code and the impact of the extra expansion should be trivial.

```

\_siunitx_table_align_left:n
\_siunitx_table_align_right:n
\_siunitx_table_align_auxi:nn
\_siunitx_table_align_auxii:nn
201 \cs_new_protected:Npn \_siunitx_table_align_center:n #1
202 { \_siunitx_table_align_auxi:nn {#1} { 0pt+plus~0.5fill } }

```



```

203 \cs_new_protected:Npn \__siunitx_table_align_left:n #1
204 { \__siunitx_table_align_auxi:nn {#1} { Opt } }
205 \cs_new_protected:Npn \__siunitx_table_align_right:n #1
206 { \__siunitx_table_align_auxi:nn {#1} { Opt~plus~1fill } }
207 \cs_new_protected:Npn \__siunitx_table_align_auxi:nn #1#2
208 {
209   \bool_if:NTF \l__siunitx_table_fixed_width_bool
210   { \hbox_to_wd:nn \l__siunitx_table_column_width_dim }
211   { \use:n }
212   {
213     \__siunitx_table_skip:n {#2}
214     #1
215     \__siunitx_table_skip:n { Opt~plus~1fill - #2 }
216   }
217 }
218 \AtBeginDocument
219 {
220   \@ifpackageloaded { colortbl }
221   {
222     \cs_new_eq:NN
223     \__siunitx_table_align_auxii:nn
224     \__siunitx_table_align_auxi:nn
225     \cs_set_protected:Npn \__siunitx_table_align_auxi:nn #1#2
226     {
227       \__siunitx_table_skip:n{ Opt~plus~-0.5fill }
228       \__siunitx_table_align_auxii:nn {#1} {#2}
229       \__siunitx_table_skip:n { Opt~plus~-0.5fill }
230     }
231   }
232   { }
233 }

```

(End definition for `\__siunitx_table_align_center:n` and others.)

## 2.5 Printing just text

In cases where there is no numerical part, siunitx allows alignment of the “escaped” text independent of the underlying column type.

`\l__siunitx_table_align_text_tl` Alignment is handled using a `tl` as this allows a fast lookup at the point of use.

```

234 \keys_define:nn { siunitx }
235 {
236   table-text-alignment .choices:nn =
237   { center , left , right }
238   { \tl_set:Nn \l__siunitx_table_align_text_tl {#1} } ,
239 }
240 \tl_new:N \l__siunitx_table_align_text_tl

```

(End definition for `\l__siunitx_table_align_text_tl`.)

`\__siunitx_table_print_text:n` Printing escaped text is easy: just place it in correctly in the column.

```

\__siunitx_table_print_text:V
241 \cs_new_protected:Npn \__siunitx_table_print_text:n #1
242 {
243   \bool_set_true:N \l__siunitx_table_text_bool

```

```

244 \use:c { __siunitx_table_align_ \l__siunitx_table_align_text_tl :n } {#1}
245 }
246 \cs_generate_variant:Nn \__siunitx_table_print_text:n { V }
(End definition for \__siunitx_table_print_text:n.)

```

## 2.6 Number alignment: core ideas

```

\l__siunitx_table_integer_box Boxes for the content before and after the decimal marker.
\l__siunitx_table_decimal_box
247 \box_new:N \l__siunitx_table_integer_box
248 \box_new:N \l__siunitx_table_decimal_box
(End definition for \l__siunitx_table_integer_box and \l__siunitx_table_decimal_box.)

\__siunitx_table_fil: Primitives renamed.
\__siunitx_table_fill:
249 \cs_new_eq:NN \__siunitx_table_fil: \tex_hfil:D
250 \cs_new_eq:NN \__siunitx_table_fill: \tex_hfill:D
(End definition for \__siunitx_table_fil: and \__siunitx_table_fill:.)

\__siunitx_table_cleanup_decimal:w To remove the excess marker tokens in a decimal part.
251 \cs_new:Npn \__siunitx_table_cleanup_decimal:w
252   #1 \q_nil #2 \q_nil #3 \q_nil #4 \q_nil #5 \q_nil #6 \q_nil #7 \q_nil
253   { #1#2#3#4#5#6#7 }
(End definition for \__siunitx_table_cleanup_decimal:w.)

\__siunitx_table_color_check:N Handle the fact that splitting a number can leave a negative color dangling.
\__siunitx_table_color_check:w
\__siunitx_table_color_check:Nnw
254 \cs_new_protected:Npn \__siunitx_table_color_check:N #1
255   { \exp_after:wN \__siunitx_table_color_check:w #1 \q_stop }
256 \cs_new_protected:Npn \__siunitx_table_color_check:w #1 \q_nil #2 \q_stop
257   {
258     \tl_if_head_eq_meaning:nNT {#1} \color
259     { \__siunitx_table_color_check:Nnw #1 \q_stop }
260   }
261 \cs_new_protected:Npn \__siunitx_table_color_check:Nnw #1#2#3 \q_stop
262   { \keys_set:nn { siunitx } { number-color = #2 } }
(End definition for \__siunitx_table_color_check:N, \__siunitx_table_color_check:w, and \__-
siunitx_table_color_check:Nnw.)

\__siunitx_table_center_marker: When centering on the decimal marker, the easiest approach is to simply re-box the two
parts. That is needed whether or not we are parsing numbers, so is best as a short
auxiliary. Notice that we need to allow for the width of the decimal marker itself.
263 \cs_new_protected:Npn \__siunitx_table_center_marker:
264   {
265     \hbox_set:Nn \l__siunitx_table_tmp_box
266       { \ensuremath { \mathord { \l__siunitx_number_output_decimal_tl } } }
267     \dim_compare:nNnTF
268       { \box_wd:N \l__siunitx_table_integer_box }
269       >
270       {
271         \box_wd:N \l__siunitx_table_decimal_box
272         - \box_wd:N \l__siunitx_table_tmp_box
273       }

```

```

274 {
275   \hbox_set_to_wd:Nnn \l__siunitx_table_decimal_box
276   {
277     \box_wd:N \l__siunitx_table_integer_box
278     + \box_wd:N \l__siunitx_table_tmp_box
279   }
280   {
281     \hbox_unpack:N \l__siunitx_table_decimal_box
282     \__siunitx_table_fil:
283   }
284 }
285 {
286   \hbox_set_to_wd:Nnn \l__siunitx_table_integer_box
287   {
288     \box_wd:N \l__siunitx_table_decimal_box
289     - \box_wd:N \l__siunitx_table_tmp_box
290   }
291   {
292     \__siunitx_table_fil:
293     \hbox_unpack:N \l__siunitx_table_integer_box
294   }
295 }
296 }

```

(End definition for \\_\_siunitx\_table\_center\_marker:.)

\l\_\_siunitx\_table\_auto\_round\_bool  
\l\_\_siunitx\_table\_align\_mode\_tl  
\l\_\_siunitx\_table\_align\_number\_tl

Options for tables with defined space.

```

297 \keys_define:nn { siunitx }
298 {
299   table-alignment .meta:n =
300     { table-number-alignment = #1 , table-text-alignment = #1 },
301   table-alignment-mode .choices:nn =
302     { none , format , marker }
303     { \tl_set_eq:NN \l__siunitx_table_align_mode_tl \l_keys_choice_tl } ,
304   table-auto-round .bool_set:N =
305     \l__siunitx_table_auto_round_bool ,
306   table-format .code:n =
307     {
308       \__siunitx_table_split:nNNN {#1}
309       \l__siunitx_table_before_model_tl
310       \l__siunitx_table_model_tl
311       \l__siunitx_table_after_model_tl
312       \exp_args:NV \__siunitx_table_generate_model:n \l__siunitx_table_model_tl
313       \tl_set:Nn \l__siunitx_table_align_mode_tl { format }
314     } ,
315   table-number-alignment .choices:nn =
316     { center , left , right }
317     { \tl_set_eq:NN \l__siunitx_table_align_number_tl \l_keys_choice_tl }
318 }
319 \tl_new:N \l__siunitx_table_align_mode_tl
320 \tl_new:N \l__siunitx_table_align_number_tl

```

(End definition for \l\_\_siunitx\_table\_auto\_round\_bool, \l\_\_siunitx\_table\_align\_mode\_tl, and \l\_\_siunitx\_table\_align\_number\_tl.)

\l\_\_siunitx\_table\_format\_tl    The input and output versions of the model entry in a table.

\l\_\_siunitx\_table\_model\_tl    321 \tl\_new:N \l\_\_siunitx\_table\_format\_tl  
322 \tl\_new:N \l\_\_siunitx\_table\_before\_model\_tl  
323 \tl\_new:N \l\_\_siunitx\_table\_model\_tl  
324 \tl\_new:N \l\_\_siunitx\_table\_after\_model\_tl

(End definition for \l\_\_siunitx\_table\_format\_tl and \l\_\_siunitx\_table\_model\_tl.)

\\_\_siunitx\_table\_generate\_model:n    Creating a model for a table at this stage means parsing the format and converting that  
\\_\_siunitx\_table\_generate\_model:nnnnnnn    to an appropriate model. Things are quite straight-forward other than the uncertainty  
\\_\_siunitx\_table\_generate\_model\_S:nnw    part. At this stage there is no point in formatting the model: that has to happen at  
\\_\_siunitx\_table\_generate\_model\_S:nnn    point-of-use. Notice that the uncertainty part needs to allow for the case where we cross  
the decimal.

325 \cs\_new\_protected:Npn \\_\_siunitx\_table\_generate\_model:n #1  
326 {  
327    \group\_begin:  
328    \bool\_set\_true:N \l\_siunitx\_number\_parse\_bool  
329    \keys\_set:nn { siunitx } { retain-explicit-plus = true }  
330    \siunitx\_number\_parse:nN {#1} \l\_\_siunitx\_table\_format\_tl  
331    \exp\_args:NNNV \group\_end:  
332    \tl\_set:Nn \l\_\_siunitx\_table\_format\_tl \l\_\_siunitx\_table\_format\_tl  
333    \tl\_if\_empty:NF \l\_\_siunitx\_table\_format\_tl  
334    {  
335       \exp\_after:wN \\_\_siunitx\_table\_generate\_model:nnnnnnn  
336       \l\_\_siunitx\_table\_format\_tl  
337    }  
338 }  
339 \cs\_new\_protected:Npn \\_\_siunitx\_table\_generate\_model:nnnnnnn #1#2#3#4#5#6#7  
340 {  
341    \tl\_set:Nx \l\_\_siunitx\_table\_model\_tl  
342    {  
343       \exp\_not:n { {#1} {#2} }  
344       { \prg\_replicate:nn {#3} { 8 } }  
345       { \prg\_replicate:nn { 0 #4 } { 8 } }  
346       {  
347          \tl\_if\_blank:NF {#5}  
348          {  
349             \use:c { \_\_siunitx\_table\_generate\_model\_ \tl\_head:n {#5} :nnw }  
350             {#4} #5  
351          }  
352       }  
353       \exp\_not:n { {#6} }  
354       {  
355          \int\_compare:nNnTF {#7} = 0  
356          { 0 }  
357          { \prg\_replicate:nn {#7} { 8 } }  
358       }  
359    }  
360 }  
361 \cs\_new:Npn \\_\_siunitx\_table\_generate\_model\_S:nnw #1#2#3  
362 {  
363    { S }  
364    {  
365       \exp\_args:Nff \\_\_siunitx\_table\_generate\_model\_S:nnn

```

366         { \tl_count:n {#1} } { \tl_count:n {#3} }
367         {#3}
368     }
369 }
370 \cs_new:Npn \__siunitx_table_generate_model_S:nnn #1#2#3
371 {
372     \prg_replicate:nn
373     {
374         \int_compare:nNnTF {#2} > {#1}
375         {
376             \str_range:nnn {#3} { 1 } {#1}
377             +
378             \str_range:nnn {#3} { 1 + #1 } {#2}
379         }
380         {#3}
381     }
382     { 8 }
383 }

```

(End definition for \\_\_siunitx\_table\_generate\_model:n and others.)

## 2.7 Directly printing without collection

Collecting the number allows for various effects but is not as fast as simply aligning on the first token that is a decimal marker. The strategy here is that used by dcolumn.

```

\__siunitx_table_direct_begin: After removing the \ignorespaces at the start of the cell (see comments for \__siunitx_
\__siunitx_table_direct_begin:w table_collect_begin:N), check to see if there is a { and branch as appropriate.
\__siunitx_table_direct_end:
\__siunitx_table_direct_marker: 384 \cs_new_protected:Npn \__siunitx_table_direct_begin:
\__siunitx_table_direct_marker_switch: 385 { \__siunitx_table_direct_begin:w }
\__siunitx_table_direct_marker_end: 386 \cs_new_protected:Npn \__siunitx_table_direct_begin:w #1 \ignorespaces
\__siunitx_table_direct_format: 387 {
\__siunitx_table_direct_format:nnnnnn 388 #1
\__siunitx_table_direct_format:w 389 \peek_catcode_ignore_spaces:NTF \c_group_begin_token
\__siunitx_table_direct_format_switch: 390 { \__siunitx_table_print_text:n }
\__siunitx_table_direct_format_end: 391 {
\__siunitx_table_direct_none: 392 \m@th
\__siunitx_table_direct_none_end: 393 \use:c { __siunitx_table_direct_ \l__siunitx_table_align_mode_tl : }
394 }
395 }
396 \cs_new_protected:Npn \__siunitx_table_direct_end:
397 { \use:c { __siunitx_table_direct_ \l__siunitx_table_align_mode_tl _end: } }

```

When centring the content about a decimal marker, the trick is to collect everything into two boxes and then compare the sizes. As we are always in math mode, we can use a math active token to make the switch. The up-front setting of the decimal box deals with the case where there is no decimal part.

```

398 \cs_new_protected:Npn \__siunitx_table_direct_marker:
399 {
400     \hbox_set:Nn \l__siunitx_table_tmp_box
401     { \ensuremath { \mathord { \l__siunitx_number_output_decimal_tl } } }
402     \hbox_set_to_wd:Nnn \l__siunitx_table_decimal_box
403     { \box_wd:N \l__siunitx_table_tmp_box }
404     { \__siunitx_table_fil: }

```

```

405 \hbox_set:Nw \l__siunitx_table_integer_box
406 \c_math_toggle_token
407 \tl_map_inline:Nn \l_siunitx_number_input_decimal_tl
408 {
409   \char_set_active_eq:NN ##1 \__siunitx_table_direct_marker_switch:
410   \char_set_mathcode:nn { '##1 } { "8000 }
411 }
412 }
413 \cs_new_protected:Npn \__siunitx_table_direct_marker_switch:
414 {
415   \c_math_toggle_token
416   \hbox_set_end:
417   \hbox_set:Nw \l__siunitx_table_decimal_box
418   \c_math_toggle_token
419   \l_siunitx_number_output_decimal_tl
420 }
421 \cs_new_protected:Npn \__siunitx_table_direct_marker_end:
422 {
423   \c_math_toggle_token
424   \hbox_set_end:
425   \__siunitx_table_center_marker:
426   \use:c { __siunitx_table_align_ \l__siunitx_table_align_text_tl :n }
427   {
428     \box_use_drop:N \l__siunitx_table_integer_box
429     \box_use_drop:N \l__siunitx_table_decimal_box
430   }
431 }

```

For the version where there is space reserved, first format and decompose that, then create appropriately-sized boxes.

```

432 \cs_new_protected:Npn \__siunitx_table_direct_format:
433 {
434   \tl_set:Nx \l__siunitx_table_tmp_tl
435   { \siunitx_number_output:NN \l__siunitx_table_model_tl \q_nil }
436   \exp_after:wN \__siunitx_table_direct_format_aux:w
437   \l__siunitx_table_tmp_tl \q_stop
438 }
439 \cs_new_protected:Npn \__siunitx_table_direct_format_aux:w
440 #1 \q_nil #2 \q_nil #3 \q_nil #4 \q_stop
441 {
442   \hbox_set:Nn \l__siunitx_table_tmp_box
443   { \ensuremath { \__siunitx_table_cleanup_decimal:w #4 } }
444   \hbox_set_to_wd:Nnn \l__siunitx_table_decimal_box
445   { \box_wd:N \l__siunitx_table_tmp_box }
446   { \__siunitx_table_fil: }
447   \hbox_set:Nn \l__siunitx_table_tmp_box { \ensuremath { #1#2#3 } }
448   \hbox_set_to_wd:Nnw \l__siunitx_table_integer_box
449   { \box_wd:N \l__siunitx_table_tmp_box }
450   \c_math_toggle_token
451   \tl_map_inline:Nn \l_siunitx_number_input_decimal_tl
452   {
453     \char_set_active_eq:NN ##1 \__siunitx_table_direct_format_switch:
454     \char_set_mathcode:nn { '##1 } { "8000 }
455   }

```

```

456     \_siunitx_table_fill:
457   }
458 \cs_new_protected:Npn \_siunitx_table_direct_format_switch:
459 {
460     \c_math_toggle_token
461   \hbox_set_end:
462   \hbox_set_to_wd:Nnw \l__siunitx_table_decimal_box
463     { \box_wd:N \l__siunitx_table_decimal_box }
464     \c_math_toggle_token
465     \mathord { \l_siunitx_number_output_decimal_tl }
466   }
467 \cs_new_protected:Npn \_siunitx_table_direct_format_end:
468 {
469     \c_math_toggle_token
470     \_siunitx_table_fil:
471   \hbox_set_end:
472   \use:c { __siunitx_table_align_ \l__siunitx_table_align_number_tl :n }
473     {
474       \box_use_drop:N \l__siunitx_table_integer_box
475       \box_use_drop:N \l__siunitx_table_decimal_box
476     }
477   }

```

No parsing and no alignment is easy.

```

478 \cs_new_protected:Npn \_siunitx_table_direct_none: { \c_math_toggle_token }
479 \cs_new_protected:Npn \_siunitx_table_direct_none_end: { \c_math_toggle_token }

```

(End definition for `\_siunitx_table_direct_begin:` and others.)

## 2.8 Printing numbers in cells: main functions

`\l__siunitx_table_before_box` For alignment of text outside of a number.

```

\l__siunitx_table_after_box
480 \box_new:N \l__siunitx_table_before_box
481 \box_new:N \l__siunitx_table_after_box

```

(End definition for `\l__siunitx_table_before_box` and `\l__siunitx_table_after_box`.)

`\l__siunitx_table_carry_dim` Used to “carry forward” the amount of white space which needs to be inserted after the decimal marker.

```

482 \dim_new:N \l__siunitx_table_carry_dim

```

(End definition for `\l__siunitx_table_carry_dim`.)

`\l_siunitx_table_align_comparator_bool` Alignment is handled using a `tl` as this allows a fast lookup at the point of use.

```

\l_siunitx_table_align_exponent_bool
\l_siunitx_table_align_after_bool
\l_siunitx_table_align_before_bool
\l_siunitx_table_align_uncertainty_bool
483 \keys_define:nn { siunitx }
484 {
485   table-align-comparator .bool_set:N =
486     \l__siunitx_table_align_comparator_bool ,
487   table-align-exponent .bool_set:N =
488     \l_siunitx_table_align_exponent_bool ,
489   table-align-text-after .bool_set:N =
490     \l_siunitx_table_align_after_bool ,
491   table-align-text-before .bool_set:N =
492     \l_siunitx_table_align_before_bool ,
493   table-align-uncertainty .bool_set:N =

```

```

494 \l__siunitx_table_align_uncertainty_bool
495 }

```

(End definition for `\l__siunitx_table_align_comparator_bool` and others.)

```

\__siunitx_table_print:nnn
\__siunitx_table_print:VVV
  \__siunitx_table_print_marker:nnn
  \__siunitx_table_print_marker:w
  \__siunitx_table_print_marker_aux:w
  \__siunitx_table_print_format:nnn
  \__siunitx_table_print_marker_auxii:w
  \__siunitx_table_print_marker_auxiii:w
  \__siunitx_table_print_format_after:N
  \__siunitx_table_print_format_box:Nn
  \__siunitx_table_print_none:nnn

```

```

496 \cs_new_protected:Npn \__siunitx_table_print:nnn #1#2#3
497 { \use:c { __siunitx_table_print_ \l__siunitx_table_align_mode_tl :nnn } {#1} {#2} {#3} }
498 \cs_generate_variant:Nn \__siunitx_table_print:nnn { VVV }

```

When centering on the decimal marker, alignment is relatively simple, and close in concept to that used without parsing. First we need to deal with any text before or after the number. For text *before*, there's the case where it has no width and might be a font or color change: that has to be filtered out first. Then we can adjust the size of this material and that after the number such that they are equal. The number itself can then be formatted, splitting at the decimal marker. A bit more size adjustment, then the number itself and any text at the end can be inserted.

```

499 \cs_new_protected:Npn \__siunitx_table_print_marker:nnn #1#2#3
500 {
501   \hbox_set:Nn \l__siunitx_table_before_box {#1}
502   \dim_compare:nNnT { \box_wd:N \l__siunitx_table_before_box } = { Opt }
503   {
504     \box_clear:N \l__siunitx_table_before_box
505     #1
506   }
507   \hbox_set:Nn \l__siunitx_table_after_box {#3}
508   \dim_compare:nNnTF
509   { \box_wd:N \l__siunitx_table_after_box }
510   > { \box_wd:N \l__siunitx_table_before_box }
511   {
512     \hbox_set_to_wd:Nnn \l__siunitx_table_before_box
513     { \box_wd:N \l__siunitx_table_after_box }
514     {
515       \__siunitx_table_fil:
516       \hbox_unpack:N \l__siunitx_table_before_box
517     }
518   }
519   {
520     \hbox_set_to_wd:Nnn \l__siunitx_table_after_box
521     { \box_wd:N \l__siunitx_table_before_box }
522     {
523       \hbox_unpack:N \l__siunitx_table_after_box
524       \__siunitx_table_fil:
525     }
526   }
527   \box_use_drop:N \l__siunitx_table_before_box
528   \siunitx_number_parse:nN {#2} \l__siunitx_table_tmp_tl
529   \siunitx_number_process:NN \l__siunitx_table_tmp_tl \l__siunitx_table_tmp_tl
530   \tl_set:Nx \l__siunitx_table_tmp_tl
531   { \siunitx_number_output:NN \l__siunitx_table_tmp_tl \q_nil }
532   \__siunitx_table_color_check:N \l__siunitx_table_tmp_tl
533   \exp_after:wN \__siunitx_table_print_marker:w
534   \l__siunitx_table_tmp_tl \q_stop
535   \box_use_drop:N \l__siunitx_table_after_box

```



```

536 }
537 \cs_new_protected:Npn \__siunitx_table_print_marker:w
538 #1 \q_nil #2 \q_nil #3 \q_nil #4 \q_stop
539 {
540   \hbox_set:Nn \l__siunitx_table_integer_box
541     { \siunitx_print_number:n { #1#2#3 } }
542   \hbox_set:Nn \l__siunitx_table_decimal_box
543     {
544       \siunitx_print_number:x
545       { \__siunitx_table_print_marker_aux:w #4 }
546     }
547   \__siunitx_table_center_marker:
548   \use:c { __siunitx_table_align_ \l__siunitx_table_align_text_tl :n }
549   {
550     \box_use_drop:N \l__siunitx_table_integer_box
551     \box_use_drop:N \l__siunitx_table_decimal_box
552   }
553 }
554 \cs_new:Npn \__siunitx_table_print_marker_aux:w
555 #1 \q_nil #2 \q_nil #3 \q_nil #4 \q_nil #5 \q_nil #6 \q_nil #7 \q_nil
556 {
557   \exp_not:n {#1#2#3#4#5}
558   \tl_if_blank:nT {#1#2#3#4#5} { { } }
559   \exp_not:n {#6#7}
560 }

```

For positioning based on a format, we have to work part-by-part as there are a number of alignment points to get right. As for the `marker` approach, first we check if the material before the numerical content is of zero width. Next we need to format the model and content numbers, before starting an auxiliary chain to pick out the various parts in order.

```

561 \cs_new_protected:Npn \__siunitx_table_print_format:nnn #1#2#3
562 {
563   \hbox_set:Nn \l__siunitx_table_tmp_box { \l__siunitx_table_before_model_tl }
564   \hbox_set:Nn \l__siunitx_table_before_box {#1}
565   \dim_compare:nNnT { \box_wd:N \l__siunitx_table_before_box } = { 0pt }
566   {
567     \box_clear:N \l__siunitx_table_before_box
568     #1
569   }
570   \bool_if:NF \l__siunitx_table_align_before_bool
571   { \__siunitx_table_print_format_hfuzz:NNnn }
572   \hbox_set_to_wd:Nnn \l__siunitx_table_before_box
573   { \box_wd:N \l__siunitx_table_tmp_box }
574   {
575     \__siunitx_table_fil:
576     \hbox_unpack:N \l__siunitx_table_before_box
577   }
578   \tl_use:N \l__siunitx_table_tmp_tl
579   \siunitx_number_parse:nN {#2} \l__siunitx_table_tmp_tl
580   \group_begin:
581   \bool_if:NT \l__siunitx_table_auto_round_bool
582   {
583     \exp_args:Nx \keys_set:nn { siunitx }
584     {

```

```

585         round-mode      = places ,
586         round-pad       = true   ,
587         round-precision =
588             \exp_after:wN \_siunitx_table_print_format:nnnnnn
589             \l__siunitx_table_format_tl
590     }
591 }
592 \siunitx_number_process:NN \l__siunitx_table_tmp_tl \l__siunitx_table_tmp_tl
593 \exp_args:NNNV \group_end:
594 \tl_set:Nn \l__siunitx_table_tmp_tl \l__siunitx_table_tmp_tl
595 \tl_set:Nx \l__siunitx_table_tmp_tl
596 {
597     \siunitx_number_output:NN \l__siunitx_table_model_tl \q_nil
598     \exp_not:N \q_mark
599     \siunitx_number_output:NN \l__siunitx_table_tmp_tl \q_nil
600 }
601 \exp_after:wN \_siunitx_table_print_format_auxi:w
602 \l__siunitx_table_tmp_tl \q_stop
603 \hbox_set:Nn \l__siunitx_table_tmp_box { \l__siunitx_table_after_model_tl }
604 \hbox_set_to_wd:Nnn \l__siunitx_table_after_box
605 { \box_wd:N \l__siunitx_table_tmp_box + \l__siunitx_table_carry_dim }
606 {
607     \bool_if:NT \l__siunitx_table_align_after_bool
608     { \skip_horizontal:n { \l__siunitx_table_carry_dim } }
609     #3
610     \_siunitx_table_fil:
611 }
612 \use:c { __siunitx_table_align_ \l__siunitx_table_align_number_tl :n }
613 {
614     \box_use_drop:N \l__siunitx_table_before_box
615     \box_use_drop:N \l__siunitx_table_integer_box
616     \box_use_drop:N \l__siunitx_table_decimal_box
617     \box_use_drop:N \l__siunitx_table_after_box
618 }
619 }
620 \cs_new:Npn \_siunitx_table_print_format:nnnnnn #1#2#3#4#5#6#7
621 { 0 #4 }
622 \cs_new_protected:Npn \_siunitx_table_print_format_hfuzz:NNnn #1#2#3#4
623 {
624     \use:x
625     {
626         \tex_hfuzz:D \c_max_dim
627         \exp_not:n { #1 #2 {#3} {#4} }
628         \tex_hfuzz:D \tex_the:D \tex_huzz:D
629     }
630 }

```

The first numerical part to handle is the comparator. Any white space we need to add goes into the text part *if* alignment is not active (*i.e.* we are looking “backwards” to place this filler).

```

631 \cs_new_protected:Npn \_siunitx_table_print_format_auxi:w
632 #1 \q_nil #2 \q_mark #3 \q_nil #4 \q_stop
633 {
634     \_siunitx_table_color_check:w #3 \q_nil \q_stop

```

```

635 \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box {#1}
636 \bool_if:NTF \l__siunitx_table_align_before_bool
637 {
638   \hbox_set_to_wd:Nnn \l__siunitx_table_integer_box
639   { \box_wd:N \l__siunitx_table_tmp_box }
640   {
641     \__siunitx_table_fil:
642     \tl_if_blank:nF {#3}
643     { \siunitx_print_number:n {#3} }
644   }
645 }
646 {
647   \__siunitx_table_print_format_box:Nn \l__siunitx_table_integer_box {#3}
648   \hbox_set_to_wd:Nnn \l__siunitx_table_before_box
649   {
650     \box_wd:N \l__siunitx_table_before_box
651     + \box_wd:N \l__siunitx_table_tmp_box
652     - \box_wd:N \l__siunitx_table_integer_box
653   }
654   {
655     \__siunitx_table_fil:
656     \hbox_unpack:N \l__siunitx_table_before_box
657   }
658 }
659 \__siunitx_table_print_format_auxii:w #2 \q_mark #4 \q_stop
660 }

```

The integer part follows much the same pattern, except now it is control of the comparator alignment that determines where the white space goes. As we already have content in the `integer` box, we need to measure how much *extra* material has been added. To avoid using more boxes or re-setting, we do that by recording sizes before and after the change. (In effect, `\l__siunitx_table_tmp_dim` is here “`\l__@@_comparator_dim`”.)

```

661 \cs_new_protected:Npn \__siunitx_table_print_format_auxii:w
662 #1 \q_nil #2 \q_nil #3 \q_mark #4 \q_nil #5 \q_nil #6 \q_stop
663 {
664   \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box {#1#2}
665   \bool_lazy_and:nnTF
666   { \l__siunitx_table_align_comparator_bool }
667   { \dim_compare_p:nNn { \box_wd:N \l__siunitx_table_integer_box } > { 0pt } }
668   {
669     \hbox_set_to_wd:Nnn \l__siunitx_table_integer_box
670     {
671       \box_wd:N \l__siunitx_table_integer_box
672       + \box_wd:N \l__siunitx_table_tmp_box
673     }
674     {
675       \hbox_unpack:N \l__siunitx_table_integer_box
676       \__siunitx_table_fil:
677       \siunitx_print_number:n {#4#5}
678     }
679   }
680   {
681     \bool_if:NTF \l__siunitx_table_align_before_bool
682     {

```

```

683         \hbox_set_to_wd:Nnn \l__siunitx_table_integer_box
684         {
685             \box_wd:N \l__siunitx_table_integer_box
686             + \box_wd:N \l__siunitx_table_tmp_box
687         }
688         {
689             \__siunitx_table_fil:
690             \hbox_unpack:N \l__siunitx_table_integer_box
691             \siunitx_print_number:n {#4#5}
692         }
693     }
694     {
695         \dim_set:Nn \l__siunitx_table_tmp_dim
696         { \box_wd:N \l__siunitx_table_integer_box }
697         \hbox_set:Nn \l__siunitx_table_integer_box
698         {
699             \hbox_unpack:N \l__siunitx_table_integer_box
700             \siunitx_print_number:n {#4#5}
701         }
702         \__siunitx_table_print_format_hfuzz:NNnn
703         \hbox_set_to_wd:Nnn \l__siunitx_table_before_box
704         {
705             \box_wd:N \l__siunitx_table_before_box
706             + \box_wd:N \l__siunitx_table_tmp_box
707             + \l__siunitx_table_tmp_dim
708             - \box_wd:N \l__siunitx_table_integer_box
709         }
710         {
711             \__siunitx_table_fil:
712             \hbox_unpack:N \l__siunitx_table_before_box
713         }
714     }
715 }
716 \__siunitx_table_print_format_auxiii:w #3 \q_mark #6 \q_stop
717 }

```

We now deal with the decimal part: there is nothing already in the decimal box, so the basics are easy. We need to “carry forward” any white space, as where it gets inserted depends on the options for subsequent parts.

```

718 \cs_new_protected:Npn \__siunitx_table_print_format_auxiii:w
719 #1 \q_nil #2 \q_nil #3 \q_mark #4 \q_nil #5 \q_nil #6 \q_stop
720 {
721     \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box {#1#2}
722     \__siunitx_table_print_format_box:Nn \l__siunitx_table_decimal_box {#4#5}
723     \dim_set:Nn \l__siunitx_table_carry_dim
724     {
725         \box_wd:N \l__siunitx_table_tmp_box
726         - \box_wd:N \l__siunitx_table_decimal_box
727     }
728     \__siunitx_table_print_format_auxiv:w #3 \q_mark #6 \q_stop
729 }

```

Any separated uncertainty is now picked up. That has a number of parts, so the first step is to look for a sign (which will be #1). We then split, either simply tidying up the markers if there is no uncertainty, or setting it.

```

730 \cs_new_protected:Npn \__siunitx_table_print_format_auxiv:w
731   #1 \q_nil #2 \q_mark #3 \q_nil #4 \q_stop
732   {
733     \tl_if_blank:nTF {#1}
734     { \__siunitx_table_print_format_auxv:w }
735     { \__siunitx_table_print_format_auxvi:w }
736     #1#2 \q_mark #3#4 \q_stop
737   }
738 \cs_new_protected:Npn \__siunitx_table_print_format_auxv:w
739   #1 \q_nil #2 \q_nil #3 \q_nil #4 \q_mark
740   #5 \q_nil #6 \q_nil #7 \q_nil #8 \q_stop
741   { \__siunitx_table_print_format_auxvii:w #4 \q_mark #8 \q_stop }

```

Sorting out the placement of the uncertainty requires both the model and real data widths, so we store the former to avoiding needing more boxes. It's then just a case of putting the carry-over white space in the right place.

```

742 \cs_new_protected:Npn \__siunitx_table_print_format_auxvi:w
743   #1 \q_nil #2 \q_nil #3 \q_nil #4 \q_mark
744   #5 \q_nil #6 \q_nil #7 \q_nil #8 \q_stop
745   {
746     \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box { { } #1#2#3 }
747     \dim_set:Nn \l__siunitx_table_tmp_dim { \box_wd:N \l__siunitx_table_tmp_box }
748     \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box { { } #5#6#7 }
749     \__siunitx_table_print_format_after:N \l__siunitx_table_align_uncertainty_bool
750     \__siunitx_table_print_format_auxvii:w #4 \q_mark #8 \q_stop
751   }

```

Finally, we get to the exponent part: the multiplication symbol is #1 and the number itself is #2. The code is almost the same as for uncertainties, which allows a shared auxiliary to be used.

```

752 \cs_new_protected:Npn \__siunitx_table_print_format_auxvii:w
753   #1 \q_nil #2 \q_mark #3 \q_nil #4 \q_stop
754   {
755     \tl_if_blank:nF {#2}
756     {
757       \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box { { } #1#2 }
758       \dim_set:Nn \l__siunitx_table_tmp_dim { \box_wd:N \l__siunitx_table_tmp_box }
759       \__siunitx_table_print_format_box:Nn \l__siunitx_table_tmp_box { { } #3#4 }
760       \__siunitx_table_print_format_after:N \l__siunitx_table_align_exponent_bool
761     }
762   }

```

A simple auxiliary to avoid relatively expensive use of the print routine for empty parts.

```

763 \cs_new_protected:Npn \__siunitx_table_print_format_box:Nn #1#2
764   {
765     \hbox_set:Nn #1
766     {
767       \tl_if_blank:nF {#2}
768       { \siunitx_print_number:n {#2} }
769     }
770   }

```

A common routine for placing material after the decimal marker and “shuffling”.

```

771 \cs_new_protected:Npn \__siunitx_table_print_format_after:N #1
772   {

```

```

773 \bool_if:NTF #1
774 {
775   \hbox_set_to_wd:Nnn \l__siunitx_table_decimal_box
776   {
777     \box_wd:N \l__siunitx_table_decimal_box
778     + \l__siunitx_table_carry_dim
779     + \box_wd:N \l__siunitx_table_tmp_box
780   }
781   {
782     \hbox_unpack:N \l__siunitx_table_decimal_box
783     \l__siunitx_table_fil:
784     \hbox_unpack:N \l__siunitx_table_tmp_box
785   }
786   \dim_set:Nn \l__siunitx_table_carry_dim
787   {
788     \l__siunitx_table_tmp_dim
789     - \box_wd:N \l__siunitx_table_tmp_box
790   }
791 }
792 {
793   \hbox_set:Nn \l__siunitx_table_decimal_box
794   {
795     \hbox_unpack:N \l__siunitx_table_decimal_box
796     \hbox_unpack:N \l__siunitx_table_tmp_box
797   }
798   \dim_add:Nn \l__siunitx_table_carry_dim
799   {
800     \l__siunitx_table_tmp_dim
801     - \box_wd:N \l__siunitx_table_tmp_box
802   }
803 }
804 }

```

With no alignment, everything supplied is treated more-or-less the same as `\num` (but without the `xparse` wrapper).

```

805 \cs_new_protected:Npn \__siunitx_table_print_none:nnn #1#2#3
806 {
807   \use:c { __siunitx_table_align_ \l__siunitx_table_align_number_tl :n }
808   {
809     #1
810     \siunitx_number_format:nN {#2} \l__siunitx_table_tmp_tl
811     \siunitx_print_number:V \l__siunitx_table_tmp_tl
812     #3
813   }
814 }

```

(End definition for `\__siunitx_table_print:nnn` and others.)

## 2.9 Standard settings for module options

Some of these follow naturally from the point of definition (*e.g.* boolean variables are always `false` to begin with), but for clarity everything is set here.

```

815 \keys_set:nn { siunitx }
816 {

```

```

817     table-align-comparator = true   ,
818     table-align-exponent   = true   ,
819     table-align-text-after  = true   ,
820     table-align-text-before = true   ,
821     table-align-uncertainty = true   ,
822     table-alignment         = center ,
823     table-auto-round        = false  ,
824     table-column-width      = 0pt    ,
825     table-fixed-width       = false  ,
826     table-format            = 2.2    ,
827     table-number-alignment  = center ,
828     table-text-alignment    = center ,

```

Out of order as table-format sets this implicitly too.

```

829     table-alignment-mode    = marker
830 }
831 \end{package}

```