

siunitx-print – Printing material with font control*

Joseph Wright[†]

Released 2021-06-22

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 L^AT_EX 2_ε font selection commands are available, in particular `\bfseries`, `\mathrm`, `\mathversion`, `\fontfamily`, `\fontseries` and `\fontshape`, `\familydefault`, `\seriesdefault`, `\shapedefault` and `\selectfont`. It also requires the standard L^AT_EX 2_ε 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 `\mathrm`, the existence of `\symoperators` is assumed; other math font commands are not *required* to exist.

```
\siunitx_print_number:n  
\siunitx_print_number:(V|x)  
\siunitx_print_unit:n  
\siunitx_print_unit:(V|x)
```

```
\siunitx_print_number:n {<material>}  
\siunitx_print_unit:n {<material>}
```

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

*This file describes v3.0.15, last revised 2021-06-22.

[†]E-mail: joseph.wright@morningstar2.co.uk

<hr/> <hr/>	<code>\siunitx_print_match:n</code>	<code>\siunitx_print_match:n {\langle material \rangle}</code>
<hr/>	<code>\siunitx_print_math:n</code>	<code>\siunitx_print_math:n {\langle material \rangle}</code>
<hr/>	<code>\siunitx_print_text:n</code>	<code>\siunitx_print_text:n {\langle material \rangle}</code>

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

1.1 Key-value options

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

<hr/>	<code>color</code>	<code>color = \langle color \rangle</code>
-------	--------------------	--

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

<hr/>	<code>mode</code>	<code>mode = match math text</code>
-------	-------------------	-------------------------------------

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

<hr/> <hr/>	<code>number-color</code>	<code>number-color = \langle color \rangle</code>
-------------	---------------------------	---

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

<hr/>	<code>number-mode</code>	<code>number-mode = match math text</code>
-------	--------------------------	--

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

<hr/> <hr/>	<code>propagate-math-font</code>	<code>propagate-math-font = true false</code>
-------------	----------------------------------	---

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`.

<hr/>	<code>reset-math-version</code>	<code>reset-math-version = true false</code>
-------	---------------------------------	--

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`.

<hr/>	<code>reset-text-family</code>	<code>reset-text-family = true false</code>
-------	--------------------------------	---

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

<hr/> <hr/> reset-text-series	reset-text-series = true false
	Switch to determine whether the active text series is reset to <code>\mdseries</code> when printing in text mode. The standard setting is true .
<hr/> <hr/> reset-text-shape	reset-text-shape = true false
	Switch to determine whether the active text shape is reset to <code>\upshape</code> when printing in text mode. The standard setting is true .
<hr/> <hr/> text-family-to-math	text-family-to-math = true false
	Switch to determine if the family of the current text font should be applied (where possible) to printing in math mode. The standard setting is false .
<hr/> <hr/> text-font-command	text-font-command = $\langle cmd \rangle$
	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.
<hr/> <hr/> text-series-to-math	text-series-to-math = true false
	Switch to determine if the weight of the current text font should be applied (where possible) to printing in math mode. This is achieved by setting the <code>\mathversion</code> , and so will override <code>reset-math-version</code> . The mappings between text and math weight are set . The standard setting is false .
<hr/> <hr/> unit-color	unit-color = $\langle color \rangle$
	Color to apply to units in output: the latter should be a named color defined for use with <code>\textcolor</code> . The standard setting is empty (no color).
<hr/> <hr/> 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 <code>\siunitx-print_...:n</code> is called. The math and text options choose the relevant $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ mode for printing. The standard setting is math .

series-version-mapping

`series-version-mapping / $\langle weight \rangle = \langle version \rangle$`

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`.

2 `siunitx-print` implementation

Start the DocStrip guards.

`1 $\langle *package \rangle$`

Identify the internal prefix (L^AT_EX3 DocStrip convention): only internal material in this *submodule* should be used directly.

`2 $\langle @@=siunitx_print \rangle$`

2.1 Initial set up

The printing routines depend on `amstext` for text mode working.

`3 $\backslash\text{RequirePackage} \{ \text{amstext} \}$`

Color support is always required.

`4 $\backslash\text{RequirePackage} \{ \text{color} \}$`

`$\text{tl_replace_all:N}\text{Nn}$` Required variants.

`5 $\backslash\text{cs_generate_variant:Nn} \text{tl_replace_all:Nnn} \{ \text{NV} \}$`

(End definition for `tl_replace_all:N\text{Nn}`. This function is documented on page ??.)

`$\backslash\text{l_siunitx_print_tmp_tl}$` Scratch space.

`6 $\text{tl_new:N} \text{l_siunitx_print_tmp_tl}$`

(End definition for `l_siunitx_print_tmp_tl`.)

2.2 Printing routines

Options which apply to the main formatting routine, and so are not tied to either symbolic or literal input.

```

\l_siunitx_print_number_color_tl
\l_siunitx_print_number_mode_tl
\l_siunitx_print_unit_color_tl
\l_siunitx_print_unit_mode_tl
\l_siunitx_print_math_font_bool
\l_siunitx_print_math_version_bool
\l_siunitx_print_math_family_bool
\l_siunitx_print_text_font_tl
\l_siunitx_print_math_weight_bool
7 \tl_new:N \l__siunitx_print_number_mode_tl
8 \tl_new:N \l__siunitx_print_unit_mode_tl
9 \keys_define:nn { siunitx }
10 {
11   color .meta:n =
12     { number-color = #1 , unit-color = #1 } ,
13   mode .meta:n =
14     { number-mode = #1 , unit-mode = #1 } ,
15   number-color .tl_set:N =
16     \l__siunitx_print_number_color_tl ,
17   number-mode .choices:nn =
18     { match , math , text }
19     {
20       \tl_set_eq:NN
21         \l__siunitx_print_number_mode_tl \l_keys_choice_tl
22     } ,
23   propagate-math-font .bool_set:N =
24     \l_siunitx_print_math_font_bool ,
25   reset-math-version .bool_set:N =
26     \l_siunitx_print_math_version_bool ,
27   reset-text-family .bool_set:N =
28     \l_siunitx_print_text_family_bool ,
29   reset-text-series .bool_set:N =
30     \l_siunitx_print_text_series_bool ,
31   reset-text-shape .bool_set:N =
32     \l_siunitx_print_text_shape_bool ,
33   text-family-to-math .bool_set:N =
34     \l_siunitx_print_math_family_bool ,
35   text-font-command .tl_set:N =
36     \l_siunitx_print_text_font_tl ,
37   text-series-to-math .bool_set:N =
38     \l_siunitx_print_math_weight_bool ,
39   unit-color .tl_set:N =
40     \l_siunitx_print_unit_color_tl ,
41   unit-mode .choices:nn =
42     { match , math , text }
43     {
44       \tl_set_eq:NN
45         \l__siunitx_print_unit_mode_tl \l_keys_choice_tl
46     }
47 }
```

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

One set of “focussed” options.

```

\l_siunitx_print_version_ul_tl
\l_siunitx_print_version_el_tl
\l_siunitx_print_version_l_tl
\l_siunitx_print_version_sl_tl
\l_siunitx_print_version_m_tl
\l_siunitx_print_version_sb_tl
\l_siunitx_print_version_b_tl
\l_siunitx_print_version_eb_tl
\l_siunitx_print_version_ub_tl
48 \keys_define:nn { siunitx / series-version-mapping }
49 {
50   ul . tl_set:N = \l_siunitx_print_version_ul_tl ,
51   el . tl_set:N = \l_siunitx_print_version_el_tl ,
52   l . tl_set:N = \l_siunitx_print_version_l_tl ,
```

```

53     sl . tl_set:N = \l__siunitx_print_version_sl_tl ,
54     m . tl_set:N = \l__siunitx_print_version_m_tl ,
55     sb . tl_set:N = \l__siunitx_print_version_sb_tl ,
56     b . tl_set:N = \l__siunitx_print_version_b_tl ,
57     eb . tl_set:N = \l__siunitx_print_version_eb_tl ,
58     ub . tl_set:N = \l__siunitx_print_version_ub_tl
59 }

```

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

`\siunitx_print_number:n` The main printing function doesn't actually need to do very much: just set the color and select the correct sub-function.

```

\siunitx_print_number:V
\siunitx_print_number:x
\siunitx_print_unit:n
\siunitx_print_unit:V
\siunitx_print_unit:x
\__siunitx_print_aux:nn
60 \cs_new_protected:Npn \siunitx_print_number:n #1
61 { \__siunitx_print_aux:nn { number } {#1} }
62 \cs_generate_variant:Nn \siunitx_print_number:n { V , x }
63 \cs_new_protected:Npn \siunitx_print_unit:n #1
64 { \__siunitx_print_aux:nn { unit } {#1} }
65 \cs_generate_variant:Nn \siunitx_print_unit:n { V , x }
66 \cs_new_protected:Npn \__siunitx_print_aux:nn #1#2
67 {
68   \tl_if_empty:cTF { l__siunitx_print_ #1 _color_tl }
69   { \use:n }
70   { \exp_args:Nv \textcolor { l__siunitx_print_ #1 _color_tl } }
71   {
72     \use:c
73     {
74       siunitx_print_
75       \tl_use:c { l__siunitx_print_ #1 _mode_tl } :n
76     }
77     {#2}
78   }
79 }

```

(End definition for `\siunitx_print_number:n`, `\siunitx_print_unit:n`, and `__siunitx_print_aux:nn`. These functions are documented on page ??.)

`\siunitx_print_match:n` When the *output* mode should match the input, a simple selection of route can be made.

```

80 \cs_new_protected:Npn \siunitx_print_match:n #1
81 {
82   \mode_if_math:TF
83   { \siunitx_print_math:n {#1} }
84   { \siunitx_print_text:n {#1} }
85 }

```

(End definition for `\siunitx_print_match:n`. This function is documented on page ??.)

`__siunitx_print_replace_font:N` A simple auxiliary for “zapping” the unit font.

```

86 \cs_new_protected:Npn \__siunitx_print_replace_font:N #1
87 {
88   \tl_if_empty:NF \l_siunitx_unit_font_tl
89   {
90     \tl_replace_all:NVn #1
91     \l_siunitx_unit_font_tl
92     { \use:n }
93   }
94 }

```

(End definition for `_siunitx_print_replace_font:N`.)

`\c_siunitx_print_weight_uc_tl` Font widths where the `m` for weight is omitted.

```
95 \clist_map_inline:nn { uc , ec , c , sc , sx , x , ex , ux }
96 { \tl_const:cn { c__siunitx_print_weight_ #1 _tl } { m } }
```

`\c_siunitx_print_weight_sc_tl` (End definition for `\c__siunitx_print_weight_uc_tl` and others.)

`\c_siunitx_print_weight_x_tl` Font widths with one letter.

```
97 \clist_map_inline:nn { l , m , b }
98 { \tl_const:cn { c__siunitx_print_weight_ #1 _tl } { #1 } }
```

(End definition for `\c_siunitx_print_weight_l_tl`, `\c_siunitx_print_weight_m_tl`, and `\c__siunitx_print_weight_b_tl`.)

`\siunitx_print_math:n`

The first step in setting in math mode is to check on the math version. The starting point is the question of whether text series needs to propagate to math mode: if so, check on the mapping, otherwise check on the current math version.

```
\_siunitx_print_extract_series:Nw
\_siunitx_print_convert_series:n
\_siunitx_print_convert_series:v
\_siunitx_print_math_version:nn
\_siunitx_print_math_version:Vn
\_siunitx_print_math_auxi:n
  \_siunitx_print_math_auxii:n
  \_siunitx_print_math_auxiii:n
  \_siunitx_print_math_auxiv:n
\_siunitx_print_math_auxv:n
\_siunitx_print_math_aux:N
\_siunitx_print_math_aux:w
\_siunitx_print_math_aux:Nn
\_siunitx_print_math_aux:cn
\_siunitx_print_math_sub:n
  \_siunitx_print_math_super:n
  \_siunitx_print_math_script:n
\_siunitx_print_math_text:n
99 \cs_new_protected:Npn \siunitx_print_math:n #1
100 {
101   \bool_if:NTF \l__siunitx_print_math_weight_bool
102   {
103     \tl_set:Nx \l__siunitx_print_tmp_tl
104     { \exp_after:wN \_siunitx_print_extract_series:Nw \f@series ? \q_stop }
105     \tl_if_empty:NTF \l__siunitx_print_tmp_tl
106     { \_siunitx_print_math_auxi:n {#1} }
107     { \_siunitx_print_math_version:Vn \l__siunitx_print_tmp_tl {#1} }
108   }
109   { \_siunitx_print_math_auxi:n {#1} }
110 }
```

Look up the math version from the text series. The weight is omitted if it is `m` plus there are either one or two letters, so we have a little work to do. To keep things fast, we use a hash table based lookup rather than a sequence or property list.

```
111 \cs_new:Npn \_siunitx_print_extract_series:Nw #1#2 ? #3 \q_stop
112 {
113   \cs_if_exist:cTF { c__siunitx_print_weight_ #1#2 _tl }
114   { \_siunitx_print_convert_series:v { c__siunitx_print_weight_ #1#2 _tl } }
115   {
116     \cs_if_exist:cTF { c__siunitx_print_weight_ #1 _tl }
117     { \_siunitx_print_convert_series:v { c__siunitx_print_weight_ #1 _tl } }
118     { \_siunitx_print_convert_series:n {#1#2} }
119   }
120 }
121 \cs_new:Npn \_siunitx_print_convert_series:n #1
122 { \tl_use:c { l__siunitx_print_version_ #1 _tl } }
123 \cs_generate_variant:Nn \_siunitx_print_convert_series:n { v }
124 \cs_new_protected:Npn \_siunitx_print_math_auxi:n #1
125 {
126   \bool_if:NTF \l__siunitx_print_math_version_bool
127   { \_siunitx_print_math_version:nn { normal } {#1} }
128   { \_siunitx_print_math_auxii:n {#1} }
129 }
```

Any setting which changes the math version can only be set from text mode (as it applies at the level of a formula). As such, the first test is to see if that needs to be checked if the math version has to be set: if so, switch to text mode, sort it out and switch back. That of course means that in such cases, line breaking will not be possible.

```

130 \cs_new_protected:Npn \__siunitx_print_math_version:nn #1#2
131 {
132   \str_if_eq:VnTF \math@version { #1 }
133   { \__siunitx_print_math_auxii:n {#2} }
134   {
135     \mode_if_math:TF
136     { \text }
137     { \use:n }
138     {
139       \mathversion {#1}
140       \__siunitx_print_math_auxii:n {#2}
141     }
142   }
143 }
144 \cs_generate_variant:Nn \__siunitx_print_math_version:nn { V }

```

At this point, force math mode then start dealing with setting math font based on text family. If the text family is roman, life is slightly different to if it is sanserif or monospaced. In all cases, the outcomes can be handled using the same routines as for normal math mode treatment. The test here is on a string basis as `\f@family` and the `\ldotsdefault` commands have different `\long` status.

```

145 \cs_new_protected:Npn \__siunitx_print_math_auxii:n #1
146 { \ensuremath { \__siunitx_print_math_auxiii:n {#1} } }
147 \cs_new_protected:Npn \__siunitx_print_math_auxiii:n #1
148 {
149   \bool_if:NTF \l__siunitx_print_math_family_bool
150   {
151     \str_case_e:nnF { \f@family }
152     {
153       { \rmdefault } { \__siunitx_print_math_auxv:n }
154       { \sfdefault } { \__siunitx_print_math_aux:Nn \mathsf }
155       { \ttdefault } { \__siunitx_print_math_aux:Nn \mathtt }
156     }
157     { \__siunitx_print_math_auxiv:n }
158   }
159   { \__siunitx_print_math_auxiv:n }
160   {#1}
161 }

```

Now we deal with the font selection in math mode. There are two possible cases. First, we are retaining the current math font, and the active one is `\mathsf` or `\mathtt`: that needs to be applied to the argument. Alternatively, if the current font is not retained, ensure that normal math mode rules are active.

```

162 \cs_new_protected:Npn \__siunitx_print_math_auxiv:n #1
163 {
164   \bool_if:NTF \l__siunitx_print_math_font_bool
165   { \__siunitx_print_math_aux:N \mathsf \mathtt \q_recursion_tail \q_recursion_stop }
166   { \__siunitx_print_math_auxv:n }
167   {#1}
168 }

```



```

169 \cs_new_protected:Npn \__siunitx_print_math_auxv:n #1
170 {
171   \bool_lazy_or:nnTF
172     { \int_compare_p:nNn \fam = { -1 } }
173     { \int_compare_p:nNn \fam = \symoperators }
174     { \use:n }
175     { \mathrm }
176     { #1 }
177 }
178 \cs_new_protected:Npn \__siunitx_print_math_aux:N #1
179 {
180   \quark_if_recursion_tail_stop_do:Nn #1 { \use:n }
181   \exp_after:wN \exp_after:wN \exp_after:wN \__siunitx_print_math_aux:w
182   \cs:w \cs_to_str:N #1 \c_space_tl \cs_end:
183   \use@mathgroup ? { -2 } \q_stop #1
184 }
185 \cs_new_protected:Npn \__siunitx_print_math_aux:w #1 \use@mathgroup #2#3 #4 \q_stop #5
186 {
187   \int_compare:nNnTF \fam = { #3 }
188     { \use_i_delimit_by_q_recursion_stop:nw { \__siunitx_print_math_aux:Nn #5 } }
189     { \__siunitx_print_math_aux:N }
190 }

```

Search-and-replace fun: deal with any font commands in the argument and also inside sub/superscripts.

```

191 \cs_new_protected:Npx \__siunitx_print_math_aux:Nn #1#2
192 {
193   \group_begin:
194     \tl_set:Nn \exp_not:N \l__siunitx_print_tmp_tl { #2 }
195     \__siunitx_print_replace_font:N \exp_not:N \l__siunitx_print_tmp_tl
196     \tl_replace_all:Nnn \exp_not:N \l__siunitx_print_tmp_tl
197       { \char_generate:nn { '\_ } { 8 } }
198       { \exp_not:N \__siunitx_print_math_sub:n }
199     \tl_replace_all:Nnn \exp_not:N \l__siunitx_print_tmp_tl
200       { ^ }
201       { \exp_not:N \__siunitx_print_math_super:n }
202     #1 { \exp_not:N \tl_use:N \exp_not:N \l__siunitx_print_tmp_tl }
203   \group_end:
204 }
205 \cs_generate_variant:Nn \__siunitx_print_math_aux:Nn { c }
206 \cs_new_protected:Npx \__siunitx_print_math_sub:n #1
207 {
208   \char_generate:nn { '\_ } { 8 }
209   { \exp_not:N \__siunitx_print_math_script:n { #1 } }
210 }
211 \cs_new_protected:Npn \__siunitx_print_math_super:n #1
212 { ^ { \__siunitx_print_math_script:n { #1 } } }
213 \cs_new_protected:Npn \__siunitx_print_math_script:n #1
214 {
215   \group_begin:
216     \tl_set:Nn \l__siunitx_print_tmp_tl { #1 }
217     \__siunitx_print_replace_font:N \l__siunitx_print_tmp_tl
218     \tl_use:N \l__siunitx_print_tmp_tl
219   \group_end:

```

220 }

For tex4ht, we need to have category code 12 ^ tokens in math mode. We handle that by intercepting at the first auxiliary that makes sense.

```

221 \AtBeginDocument
222 {
223   \@ifpackageloaded { tex4ht }
224   {
225     \cs_set_protected:Npn \__siunitx_print_math_auxii:n #1
226     {
227       \tl_set:Nn \l__siunitx_print_tmp_tl {#1}
228       \exp_args:NNnx \tl_replace_all:Nnn \l__siunitx_print_tmp_tl
229       { ^ } { \token_to_str:N ^ }
230       \ensuremath { \exp_args:NV \__siunitx_print_math_auxiii:n \l__siunitx_print_tmp_tl }
231     }
232   }
233   { }
234 }

```

(End definition for \siunitx_print_math:n and others. This function is documented on page ??.)

```

\siunitx_print_text:n Typesetting in text mode is easy in font control terms but more tricky in the manipulation
  \__siunitx_print_text_replace:n of the input. The easy part comes first.
  \__siunitx_print_text_replace:N 235 \cs_new_protected:Npn \siunitx_print_text:n #1
  \__siunitx_print_text_replace:NNn 236 {
  \__siunitx_print_text_sub:n 237   \text
  \__siunitx_print_text_super:n 238   {
  \__siunitx_print_text_scripts:NnN 239     \bool_if:NT \l__siunitx_print_text_family_bool
  \__siunitx_print_text_scripts: 240     { \fontfamily { \familydefault } }
  \__siunitx_print_text_scripts_one:NnN 241     \bool_if:NT \l__siunitx_print_text_series_bool
  \__siunitx_print_text_scripts_two:NnNn 242     { \fontseries { \seriesdefault } }
  \__siunitx_print_text_scripts_two:n 243     \bool_if:NT \l__siunitx_print_text_shape_bool
  \__siunitx_print_text_scripts_two:n 244     { \fontshape { \shapedefault } }
  \__siunitx_print_text_fraction:Nnn 245     \bool_lazy_any:nT
  246     {
  247       { \l__siunitx_print_text_family_bool }
  248       { \l__siunitx_print_text_series_bool }
  249       { \l__siunitx_print_text_shape_bool }
  250     }
  251     { \selectfont }
  252     \tl_use:N \l__siunitx_print_text_font_tl
  253     \exp_args:NnV \tl_if_head_eq_meaning:nNTF {#1} \l_siunitx_unit_fraction_tl
  254     { \__siunitx_print_text_fraction:Nnn #1 }
  255     { \__siunitx_print_text_replace:n {#1} }
  256   }
  257 }

```

To get math mode material to print in text mode, various search-and-replace steps are needed.

```

258 \cs_new_protected:Npn \__siunitx_print_text_replace:n #1
259 {
260   \group_begin:
261   \tl_if_head_eq_meaning:nNTF {#1} \mathchoice
262   { \__siunitx_print_text_replace:Nnnnn #1 }
263   {

```

```

264     \_siunitx_print_text_replace:Nnnn {#1}
265     \tl_use:N \l__siunitx_print_tmp_tl
266   }
267   \group_end:
268 }
269 \cs_new_protected:Npx \_siunitx_print_text_replace:N #1
270 {
271   \_siunitx_print_replace_font:N #1
272   \exp_not:N \_siunitx_print_text_replace:NNn #1
273   \exp_not:N \mathord { }
274   \exp_not:N \pm
275   { \exp_not:N \textpm }
276   \exp_not:N \mp
277   { \exp_not:n { \ensuremath { \mp } } } }
278 -
279   { \exp_not:N \textminus }
280   \exp_not:N \times
281   { \exp_not:N \texttimes }
282   \exp_not:N \cdot
283   { \exp_not:N \textperiodcentered }
284   \char_generate:nn { \_ } { 8 }
285   { \exp_not:N \_siunitx_print_text_sub:n }
286   ^
287   { \exp_not:N \_siunitx_print_text_super:n }
288   \exp_not:N \q_recursion_tail
289   { ? }
290   \exp_not:N \q_recursion_stop
291 }
292 \cs_new_protected:Npn \_siunitx_print_text_replace:NNn #1#2#3
293 {
294   \quark_if_recursion_tail_stop:N #2
295   \tl_replace_all:Nnn #1 {#2} {#3}
296   \_siunitx_print_text_replace:NNn #1
297 }
298 \cs_new_protected:Npn \_siunitx_print_text_replace:Nnnnn #1#2#3#4#5
299 {
300   \mathchoice
301     { \text { \_siunitx_print_text_replace:n {#2} } }
302     { \text { \_siunitx_print_text_replace:n {#3} } }
303     { \text { \_siunitx_print_text_replace:n {#4} } }
304     { \text { \_siunitx_print_text_replace:n {#5} } }
305 }

```

When the bidi package is loaded, we need to make sure that `\text` is doing the correct thing.

```

306 \sys_if_engine_xetex:T
307 {
308   \AtBeginDocument
309   {
310     \@ifpackageloaded { bidi }
311     {
312       \cs_set_protected:Npn \_siunitx_print_text_replace:n #1
313       {
314         \group_begin:

```

```

315         \tl_set:Nn \l__siunitx_print_tmp_tl {#1}
316         \__siunitx_print_text_replace:N \l__siunitx_print_tmp_tl
317         \LRE { \tl_use:N \l__siunitx_print_tmp_tl }
318     \group_end:
319 }
320 }
321 { }
322 }
323 }

```

Sub- and superscripts can be in any order in the source. The first step of handling them is therefore to do a look-ahead to work out whether only one or both are present.

```

324 \cs_new_protected:Npn \__siunitx_print_text_sub:n #1
325 {
326     \__siunitx_print_text_scripts:NnN
327     \textsubscript {#1} \__siunitx_print_text_super:n
328 }
329 \cs_new_protected:Npn \__siunitx_print_text_super:n #1
330 {
331     \__siunitx_print_text_scripts:NnN
332     \textsuperscript {#1} \__siunitx_print_text_sub:n
333 }
334 \cs_new_protected:Npn \__siunitx_print_text_scripts:NnN #1#2#3
335 {
336     \cs_set_protected:Npn \__siunitx_print_text_scripts:
337     {
338         \if_meaning:w \l_peek_token #3
339         \exp_after:wN \__siunitx_print_text_scripts_two:NnNn
340         \else:
341         \exp_after:wN \__siunitx_print_text_scripts_one:Nn
342         \fi:
343         #1 {#2}
344     }
345     \peek_after:Nw \__siunitx_print_text_scripts:
346 }
347 \cs_new_protected:Npn \__siunitx_print_text_scripts: { }

```

In the simple case of one script item, we have to do a search-and-replace to deal with anything inside the argument.

```

348 \cs_new_protected:Npn \__siunitx_print_text_scripts_one:Nn #1#2
349 {
350     \group_begin:
351     \tl_set:Nn \l__siunitx_print_tmp_tl {#2}
352     \__siunitx_print_text_replace:N \l__siunitx_print_tmp_tl
353     \exp_args:NNV \group_end:
354     #1 \l__siunitx_print_tmp_tl
355 }

```

For the two scripts case, we cannot use `\textsubscript/\textsuperscript` as they don't stack directly. Instead, we sort out the ordering then use an implementation for both parts that is the same as the kernel text scripts.

```

356 \cs_new_protected:Npn \__siunitx_print_text_scripts_two:NnNn #1#2#3#4
357 {
358     \cs_if_eq:NNTF #1 \textsubscript
359     { \__siunitx_print_text_scripts_two:nn {#4} {#2} }

```

```

360     { \_siunitx_print_text_scripts_two:nn {#2} {#4} }
361   }
362 \cs_new_protected:Npx \_siunitx_print_text_scripts_two:nn #1#2
363 {
364   \group_begin:
365     \exp_not:N \m@th
366     \exp_not:N \ensuremath
367     {
368       ^ { \exp_not:N \_siunitx_print_text_scripts_two:n {#1} }
369       \char_generate:nn { '\_ } { 8 }
370       { \exp_not:N \_siunitx_print_text_scripts_two:n {#2} }
371     }
372   \group_end:
373 }
374 \cs_new_protected:Npn \_siunitx_print_text_scripts_two:n #1
375 {
376   \mbox
377   {
378     \fontsize \sf@size \z@ \selectfont
379     \_siunitx_print_text_scripts_one:Nn \use:n {#1}
380   }
381 }

```

Fraction commands are always math mode, so we have to go back and forth: this is done after general font setting for performance reasons.

```

382 \cs_new_protected:Npn \_siunitx_print_text_fraction:Nnn #1#2#3
383 {
384   \ensuremath
385   {
386     #1
387     { \mbox { \_siunitx_print_text_replace:n {#2} } }
388     { \mbox { \_siunitx_print_text_replace:n {#3} } }
389   }
390 }

```

(End definition for `\siunitx_print_text:n` and others. This function is documented on page ??.)

2.3 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.

```

391 \keys_set:nn { siunitx }
392 {
393   color           = ,
394   mode            = math ,
395   number-color    = ,
396   number-mode     = math ,
397   propagate-math-font = false ,
398   reset-math-version = true ,
399   reset-text-shape = true ,
400   reset-text-series = true ,
401   reset-text-family = true ,
402   text-family-to-math = false ,
403   text-font-command = ,

```

```

404     text-series-to-math = false ,
405     unit-color          =        ,
406     unit-mode           = math
407 }

```

These are separate as they all fall inside the same key.

```

408 \keys_set:nn { siunitx / series-version-mapping }
409 {
410     ul = light ,
411     el = light ,
412     l  = light ,
413     sl = light ,
414     m  = normal ,
415     sb = bold   ,
416     b  = bold   ,
417     eb = bold   ,
418     ub = bold
419 }
420 \end{package}

```