

# The `erw-l3` package <sup>\*</sup>

Erwann Rogard<sup>†</sup>

Released 2020/04/30

## Abstract

Utilities based on `expl3`[\[1\]](#).

## Résumé

Utilitaires de type `expl3`[\[1\]](#).

## Contents

<b>I</b>	<b>Usage</b>	<b>3</b>
1	Loading the package	3
2	basics	3
3	csint	4
4	int	4
5	oper	4
6	timestamp	5
7	tl	5
8	option	6
<b>II</b>	<b>Listing</b>	<b>7</b>
1	basics	7
	1. . . . .	7
2	csint	7
	2. . . . .	7

---

<sup>\*</sup>This file describes version v1.8, last revised 2020/04/30.

<sup>†</sup>firstname dot lastname AusTria gmail dot com

<b>3</b>	<b>int</b>	<b>7</b>
3.	.....	7
<b>4</b>	<b>oper</b>	<b>8</b>
4.	.....	8
5.	.....	8
6.	.....	8
7.	.....	9
8.	.....	9
9.	.....	9
<b>5</b>	<b>timestamp</b>	<b>10</b>
10.	.....	10
11.	.....	10
<b>6</b>	<b>tl</b>	<b>11</b>
12.	.....	11
13.	.....	11
14.	.....	11
<b>III</b>	<b>Other</b>	<b>12</b>
<b>1</b>	<b>Acknowledgment</b>	<b>12</b>
<b>2</b>	<b>Install</b>	<b>12</b>
<b>3</b>	<b>Support</b>	<b>12</b>
3.1	Platform .....	12
3.2	Engine .....	12
3.3	Results .....	12
	<b>References</b>	<b>12</b>
<b>IV</b>	<b>Implementation</b>	<b>13</b>
<b>1</b>	<b>Opening</b>	<b>13</b>
<b>2</b>	<b>basics</b>	<b>13</b>
2.1	backend .....	13
2.2	frontend .....	13
<b>3</b>	<b>csint</b>	<b>14</b>
3.1	backend .....	14
3.2	frontend .....	14
<b>4</b>	<b>int</b>	<b>14</b>
4.1	backend .....	14
4.2	frontend .....	15

<b>5</b>	<b>msg</b>	<b>15</b>
5.1	backend	15
5.2	frontend	15
<b>6</b>	<b>timestamp</b>	<b>17</b>
6.1	backend	17
6.2	frontend	19
<b>7</b>	<b>tl</b>	<b>19</b>
7.1	backend	19
7.2	frontend	20
<b>8</b>	<b>option</b>	<b>21</b>
<b>9</b>	<b>Closing</b>	<b>21</b>

## Part I

# Usage

---

<code>\usepackage</code>	<code>\usepackage{erw-l3}</code>
--------------------------	----------------------------------

---

### Requirement

1. `erw-l3.sty` and its dependencies are in the path of the L<sup>A</sup>T<sub>E</sub>X engine. See [Part III, section 3](#).
2. Goes in the *preamble*

## 2 basics

---

<code>\erw_append_arg:n</code>	<code>\erw_cs_apply:Nn{&lt;arg list&gt;}{&lt;arg&gt;}</code>
--------------------------------	--

---



---

<code>\erw_cs_apply:Nn</code>	<code>\erw_cs_apply:Nn {&lt;control sequence&gt;}{&lt;token list<sub>1</sub></code>
<code>\erw_cs_apply:(No Nf Nx cn)</code>	
<code>\erw_cs_apply:Nnn</code>	
<code>\erw_cs_apply:Nnnn</code>	
<code>\erw_cs_apply:Nnnnn</code>	

---



---

<code>\erw_cs_identity:n</code>	<code>\erw_cs_identity:n{&lt;arg&gt;}</code>
---------------------------------	--

---



---

<code>\erw_cs_set_inline:Nn</code>	<code>\erw_cs_set_inline:Nn{&lt;control sequence&gt;}{&lt;code&gt;}</code>
<code>\erw_cs_set_inline:cn</code>	

---

### 3 csint

---

---

`\erw_csint:nn`    `\erw_csint:nn{integer}{arg}`

---

---

`\erw_csint_name:n`    `\erw_csint_name:n{integer}`

---

---

`\erw_csint_names:nnn`    `\erw_csint_names:nnn{integer}{integer}{integer}`

---

---

`\erw_csint_new:n`    `\erw_csint_new:n{integer}`

---

`\erw_csint_names_braced:`  
`\erw_csint_names_braced:n`  
`\erw_csint_names_braced:nnn`

---

---

---

`\erw_csint_reset:`

### 4 int

---

---

`\erw_int_range:n`    `\erw_int_range:n{integer}`  
`\erw_int_range:nn`

### 5 oper

---

---

`\erw_oper_compose:nN`    `\erw_oper_compose:nn{control sequence list}{initial value}`  
`\erw_oper_compose:nn`

---

---

`\erw_oper_compose_c:nN`  
`\erw_oper_compose_c:nn`

---

---

`\erw_oper_compose_vers:nN`  
`\erw_oper_compose_vers:nn`

---

---

`\erw_oper_compose_seq:nN`  
`\erw_oper_compose_seq_c:nN`  
`\erw_oper_compose_seq_vers:nN`

---

---

```
\erw_oper_fold:NN
\erw_oper_fold:cN
```

---



---

```
\erw_oper_gset_function:N
\erw_oper_gset_function:n
```

---



---

```
\erw_oper_fold_seq:NN
\erw_seq:cN
```

---

## 6 timestamp

---

```
\erw_timestamp:nn \erw_timestamp:nn{date|time|datetime}-{10|16}
\erw_timestamp:
```

---

**Semantics** Timestamp in base 10 or 16

## 7 tl

---

```
\erw_tl_last_item:n
```

---



---

```
\erw_tl_map:n
\erw_tl_map:Nn
```

---



---

```
\erw_tl_map_inline:nn
```

---



---

```
\erw_tl_merge:nn
```

---



---

```
\erw_tl_repeat:nn
```

---



---

```
\erw_tl_split:nnn
\erw_tl_split:nn
```

---



---

```
\erw_tl_map_thread_at:Nnn
\erw_tl_map_thread:Nn
```

---

## 8 option

---

`\erw_option:n`

---

## Part II

# Listing

### 1 basics

Listing 1.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\erw_cs_apply:Nn \__foo:n{X}
\ExplSyntaxOff
```

f(X)

### 2 csint

Listing 2.

```
\ExplSyntaxOn
\cs_new:Nn \__foo:n{f(#1)}
\cs_new:Nn \__baz:n{h\{#1\}}
\tl_map_function:nN {{\__baz:n}{g[#1]}}{\__foo:n}\erw_csint_new:n
\exp_last_unbraced:Nx
\erw_oper_compose_c:nn
{{\erw_csint_names_braced:nnn{1}{1}{3}}
{X}}
\ExplSyntaxOff
```

h{g[f(X)]}

### 3 int

Listing 3.

```
\ExplSyntaxOn
\erw_int_range:nn{2}{5}\
\erw_int_range:n{5}
\ExplSyntaxOff
```

2345  
12345

## 4 oper

Listing 4.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\cs_set:Nn \__bar:n {g[#1]}
\cs_set:Nn \__baz:n {h\{#1\}}
\tl_set:Nn \l_tmpa_tl{X}
\erw_oper_compose:nN{\__baz:n}{\__bar:n}{\__foo:n}\l_tmpa_tl
\l_tmpa_tl\
\tl_set:Nn \l_tmpa_tl{X}
\erw_oper_compose:nn{\__baz:n}{\__bar:n}{\__foo:n}{X}\
\ExplSyntaxOff
```

h{g[f(X)]}  
h{g[f(X)]}

Listing 5.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\cs_set:Nn \__bar:n {g[#1]}
\cs_set:Nn \__baz:n {h\{#1\}}
\tl_set:Nn \l_tmpa_tl{X}
\erw_oper_compose_c:nN{\__baz:n}{\__bar:n}{\__foo:n}\l_tmpa_tl
\l_tmpa_tl\
\erw_oper_compose_c:nn{\__baz:n}{\__bar:n}{\__foo:n}{X}
\ExplSyntaxOff
```

h{g[f(X)]}  
h{g[f(X)]}

Listing 6.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\cs_set:Nn \__bar:n {g[#1]}
\cs_set:Nn \__baz:n {h\{#1\}}
\seq_new:N \l_tmp_seq
\seq_put_right:Nn \l_tmp_seq{X}
\erw_oper_compose_seq:nN{\__baz:n}{\__bar:n}{\__foo:n}\l_tmp_seq
\seq_item:Nn \l_tmp_seq{1}\
\seq_item:Nn \l_tmp_seq{2}\
\seq_item:Nn \l_tmp_seq{3}\
\seq_item:Nn \l_tmp_seq{4}
\ExplSyntaxOff
```



---

X  
f(X)  
g[f(X)]  
h{g[f(X)]}

#### Listing 7.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\cs_set:Nn \__bar:n {g[#1]}
\cs_set:Nn \__baz:n {h\{#1\}}
\seq_put_right:Nn \l_tmp_seq{X}
\erw_oper_compose_seq_c:nN{\__baz:n}{\__bar:n}{\__foo:n}}\l_tmp_seq
\seq_item:Nn \l_tmp_seq{1}\\
\seq_item:Nn \l_tmp_seq{2}\\
\seq_item:Nn \l_tmp_seq{3}\\
\seq_item:Nn \l_tmp_seq{4}
\ExplSyntaxOff
```

---

X  
f(X)  
g[f(X)]  
h{g[f(X)]}

#### Listing 8.

```
\ExplSyntaxOn
\cs_set:Npn \__foo #1 {f(#1)}
\cs_set:Npn \__bar #1 {g[#1]}
\cs_set:Npn \__baz #1 {h\{#1\}}
\erw_oper_compose_vers:nn{\__baz}{g[#1]}{\__foo}}{X}
\ExplSyntaxOff
```

---

h{g[f(X)]}

#### Listing 9.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\tl_set:Nn \l_tmpa_tl{X}
\erw_oper_fold:NN\__foo:n\l_tmpa_tl
\l_tmpa_tl\\
\cs_set:Nn \__bar:n {g[#1]}
\erw_oper_fold:cN {\__bar:n}\l_tmpa_tl
\l_tmpa_tl
\ExplSyntaxOff
```

```
f(X)
g[f(X)]
```

## 5 timestamp

Listing 10.

```
\ExplSyntaxOn
\noindent\erw_timestamp:nn{date}{10}{-}
\noindent\erw_timestamp:nn{time}{10}{\}
\noindent\erw_timestamp:nn{datetime}{10}{\}
\erw_timestamp:nn{date}{16}{\%}
\erw_timestamp:nn{time}{16}{\}
\erw_option:n{ timestamp / delim = {\%} }
\erw_timestamp:nn{datetime}{16}{\}
\ExplSyntaxOff
```

```
20200430-2202
20200430-2202
1343bee%89a
1343bee%89a
```

Listing 11.

```
\ExplSyntaxOn
\erw_option:n{ timestamp / delim = \c_empty_tl }
\iow_new:N \foo_iow
\tl_set:Nx \foo_dec { \erw_timestamp:nn{datetime}{10} }
\tl_set:Nx \foo_hex { \erw_timestamp: }
\iow_open:Nn \foo_iow{\foo_hex}
\iow_now:Nn\foo_iow{Hello,\ world!}
\iow_close:N \foo_iow
D:\foo_dec\
\file_timestamp:n{\foo_hex}\
\file_input:n{\foo_hex}
\ExplSyntaxOff
```

```
D:202004302202
D:20200430220212-04'00'
Hello, world!
```

## 6 tl

Listing 12.

```
\ExplSyntaxOn
\erw_tl_repeat:nn{3}{abracad}abra
\ExplSyntaxOff
```

---

abracadabracadabracadabra

Listing 13.

```
\ExplSyntaxOn
\erw_tl_split:nn{{a}{b}{c}}{==}
\ExplSyntaxOff
\ExplSyntaxOff
```

---

a==b==c

Listing 14.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n {(#1)}
\erw_tl_map:Nn \__foo:n{{a}{b}{c}}
\ExplSyntaxOff
```

---

(a)(b)(c)

## Part III

# Other

### 1 Acknowledgment

This work has benefited from Q&A's from the L<sup>A</sup>T<sub>E</sub>Xcommunity[2]

### 2 Install

- 1) Compile `timestamp.dtx` (under Unix, `$tex timestamp.dtx`)
- 2) Put the generated `timestamp.sty` in the search path of the L<sup>A</sup>T<sub>E</sub>Xengine

### 3 Support

This package is available from <https://www.ctan.org/pkg/timestamp> and <https://github.com/rogard/timestamp>.

#### 3.1 Platform

- i)* Linux laptop 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24  
↪ 06:16:15 UTC 2018 x86\_64 x86\_64 x86\_64 GNU/Linux

#### 3.2 Engine

- a)* pdfTeX 3.14159265-2.6-1.40.20 (TeX Live 2019)
- b)* pdfTeX 3.14159265-2.6-1.40.21 (TeX Live 2020)
- c)* LuaHBTeX, Version 1.12.0 (TeX Live 2020)
- d)* XeTeX 3.14159265-2.6-0.999992 (TeX Live 2020)

#### 3.3 Results

- 1) `timestamp v0.8` compiles satisfactorily on platform *i)* and engines *b)*, *c)*, and *d)*

## References

- [1] The L<sup>A</sup>T<sub>E</sub>X3 Project Team *The L<sup>A</sup>T<sub>E</sub>X3 interfaces*, 2019, <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf>
- [2] <https://tex.stackexchange.com/users/112708/erwann?tab=questions>

## Part IV

# Implementation

## 1 Opening

```
1 <@@=erw>
2 \ExplSyntaxOn
```

## 2 basics

### 2.1 backend

```
3 \cs_set:Nn \__erw_cs_name:N
4 {
5   \exp_last_unbraced:Nf \use_i:nnn {\cs_split_function:N #1}
6 }
```

### 2.2 frontend

```
7 \cs_set:Nn \erw_append_arg:nn
8 {
9   {#1{#2}}
10 }
11 \cs_set:Nn \erw_cs_apply:Nn
12 {
13   #1{#2}
14 }
15 \cs_generate_variant:Nn \erw_cs_apply:Nn {No, Nf, Nx, c}
16 \cs_set:Nn \erw_cs_apply:Nnn
17 {
18   #1{#2}{#3}
19 }
20 \cs_set:Nn \erw_cs_apply:Nnnn
21 {
22   #1{#2}{#3}{#4}
23 }
24 \cs_set:Nn \erw_cs_apply:Nnnnn
25 {
26   #1{#2}{#3}{#4}{#5}
27 }
28 \cs_set:Npn \erw_cs_identity:n #1{#1}
29 \cs_set:Nn \erw_cs_set_inline:Nn
30 {
31   \cs_set:Npn #1 ##1{#2}
32 }
33 \cs_generate_variant:Nn \erw_cs_set_inline:Nn {cn}
34 \cs_set:Nn \erw_cs_gset_inline:Nn
35 {
36   \cs_gset:Npn #1 ##1{#2}
37 }
38 \cs_generate_variant:Nn \erw_cs_gset_inline:Nn {cn}
```

## 3 csint

### 3.1 backend

```
39 \int_new:N \g__erw_csint_int
40 \tl_set:Nn \g__erw_csint_name_tl {\erw_csint_name:n{\g__erw_csint_int}}
```

### 3.2 frontend

```
41 \cs_set:Nn \erw_csint:nn
42 {
43   \erw_cs_apply:cn{__erw_csint_\int_to_alph:n{#1}:n}{#2}
44 }
45 \cs_set:Nn \erw_csint_name:n {__erw_csint_\int_to_alph:n{#1}:n}
46 \cs_new:Nn \erw_csint_names:nnn
47 {
48   \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_csint_name:n
49 }
50 \cs_new_protected:Nn \erw_csint_new:n
51 {
52   \int_incr:N \g__erw_csint_int
53   \erw_cs_set_inline:cn{\g__erw_csint_name_tl}
54   {
55     \token_if_cs:NTF
56     {#1}
57     {#1{##1}}
58     {#1}
59   }
60 }
61 \cs_new:Nn \erw_csint_names_braced:nnn
62 {
63   \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_csint_names_braced:n
64   % TODO \tl_range_braced:nnn?
65 }
66 \cs_set:Nn \erw_csint_names_braced:n {\erw_csint_name:n{#1}}
67 \cs_new:Nn \erw_csint_names_braced:
68 {
69   \erw_csint_names_braced:nnn{1}{1}{\g__erw_csint_int}
70 }
71 \cs_new_protected:Nn \erw_csint_reset:
72 {
73   \int_zero:N \g__erw_csint_int
74   \tl_set:Nn \__erw_csint_ext_tl{}%^^A TODO remove?
75 }
```

## 4 int

### 4.1 backend

```
76 \cs_set:Npn \__erw_int_range:nnn #1 #2 #3
77 {
78   \int_compare:nNnTF
79   {
80     \int_eval:n{#2+1}
81   }>{#3}
```

```

82 {
83   {#1}
84 }
85 {
86   \__erw_int_range:nnn
87   {
88     \exp_args:Nx\erw_append_arg:nn{#1}
89     {
90       \int_eval:n{#2+1}
91     }
92   }
93   {\int_eval:n{#2+1}}
94   {#3}
95 }
96 }

```

## 4.2 frontend

```

97 \cs_set:Nn \erw_int_range:nn
98 {
99   \__erw_int_range:nnn {#1}{#1}{#2}
100 }
101 \cs_set:Nn \erw_int_range:n
102 {
103   \__erw_int_range:nnn {}{0}{#1}
104 % ^^A Alt to:
105 % ^^A \int_step_inline:nn {#1}{##1}
106 }

```

## 5 msg

### 5.1 backend

```

107 \msg_new:nnn{__erw}{generic}{#1}
108 \msg_new:nnn{__erw}{notdecl}{#1~not~declared}
109 \msg_new:nnn{__erw}{notset}{#1~not~set}
110 %\end{macrocode}
111 % \section{\textsf{oper}}\label{impl:oper}
112 % \subsection{backend}
113 % \begin{macrocode}
114 \tl_new:N \g__erw_compose_tl
115 \tl_new:N \g__erw_oper_fold_seq_item_tl
116 \cs_set:Nn \__erw_oper_compose:NnN
117 {
118   \erw_cs_set_inline:Nn \__erw_oper_function:n
119   {
120     #1{##1}#3
121   }
122   \exp_args:Nf\erw_tl_map:n
123   {
124     \tl_reverse:n{#2}
125   }
126 }

```

### 5.2 frontend

```

127 \cs_set:Nn \erw_oper_compose:nN
128 {
129   \__erw_oper_compose:NnN \erw_oper_fold:NN {#1} #2
130 }
131 \cs_set:Nn \erw_oper_compose:nn
132 {
133   \tl_set:Nn \g__erw_compose_tl {#2}
134   \erw_oper_compose:nN{#1}\g__erw_compose_tl
135   \g__erw_compose_tl
136 }
137 \cs_set:Nn \erw_oper_compose_c:nN
138 {
139   \__erw_oper_compose:NnN \erw_oper_fold:cN {#1} #2
140 }
141 \cs_set:Nn \erw_oper_compose_c:nn
142 {
143   \tl_set:Nn \g__erw_compose_tl {#2}
144   \erw_oper_compose_c:nN{#1}\g__erw_compose_tl
145   \g__erw_compose_tl
146 }
147 \cs_set:Nn \erw_oper_compose_vers:nN
148 {
149   \msg_error:nnn{__erw}{notdecl}{\erw_oper_compose_vers:nN}
150 }
151 \cs_set:Nn \erw_oper_compose_vers:nn
152 {
153   \erw_csint_reset:{}
154   \tl_map_function:nN{#1}\erw_csint_new:n
155   \exp_last_unbraced:Nx
156   \erw_oper_compose_c:nn
157   {{\erw_csint_names_braced:{}}}
158   {#2}
159 }
160 \cs_set:Nn \erw_oper_compose_seq:nN
161 {
162   \__erw_oper_compose:NnN \erw_oper_fold_seq:NN {#1} #2
163 }
164 \cs_set:Nn \erw_oper_compose_seq_c:nN
165 {
166   \__erw_oper_compose:NnN \erw_oper_fold_seq:cN {#1} #2
167 }
168 \cs_set:Nn \erw_oper_compose_seq_vers:nN
169 {
170   \msg_error:nnn{__erw}{notdecl}{\erw_oper_compose_seq_vers:nN}
171 }
172 \cs_set:Nn \erw_oper_gset_function:N
173 {
174   \erw_cs_gset_eq:NN \__erw_oper_function:n #1
175 }
176 \cs_set:Nn \erw_oper_gset_function:n
177 {
178   \erw_cs_gset_inline:Nn \__erw_oper_function:n {#1}
179 }
180 \keys_define:nn{__erw}

```



```

181 {
182   oper/fold_set_par.tl_gset:N = \g__erw_oper_fold_set_par_tl,
183   oper/fold_set_par.value_required:n = true,
184   oper/fold_set_par.default:n = {Nf},
185   oper/fold_set_par.initial:n = {Nf},
186   oper/fold_apply_par.tl_gset:N = \g__erw_oper_fold_apply_par_tl,
187   oper/fold_apply_par.value_required:n = true,
188   oper/fold_apply_par.default:n = {Nf},
189   oper/fold_apply_par.initial:n = {Nf}
190 }
191 \cs_set:Nn \erw_oper_fold:NN
192 {
193   \use:c{tl_set:\g__erw_oper_fold_set_par_tl}
194   #2
195   {\use:c{erw_cs_apply:\g__erw_oper_fold_apply_par_tl}{#1}{#2}}
196 }
197 \cs_generate_variant:Nn \erw_oper_fold:NN {cN}
198 \cs_set:Nn \erw_oper_fold_seq:NN
199 {
200   \seq_get_right:NN #2 \g__erw_oper_fold_seq_item_tl
201   \erw_oper_fold:NN #1 \g__erw_oper_fold_seq_item_tl
202   \seq_put_right:No #2 {\g__erw_oper_fold_seq_item_tl}
203 }
204 \cs_generate_variant:Nn \erw_oper_fold_seq:NN {cN}

```

## 6 timestamp

### 6.1 backend

```

205 \msg_new:nnn{__erw}{timestamp / base}{Calling~#1,~arg~must~be~'dec|hex'}
206 \msg_new:nnn{__erw}{timestamp / period}{Calling~#1,~arg~must~be~'date|time|datetime'}

```

```

__erw_timestamp_date:N
__erw_timestamp_date_dec:
__erw_timestamp_date_hex:
207 \cs_new:Nn __erw_timestamp_date_dec:
208 {
209   \int_eval:n
210   {
211     \c_sys_year_int * 10000
212     +\c_sys_month_int * 100
213     +\c_sys_day_int * 1
214   }
215 }
216 \cs_new:Nn __erw_timestamp_date:N{\int_to_hex:n{__erw_timestamp_date_dec:}}
217 \cs_new:Nn __erw_timestamp_date_hex:{\int_to_hex:n{__erw_timestamp_date_dec:}}

(End definition for __erw_timestamp_date:N, __erw_timestamp_date_dec:, and __erw_timestamp_date_hex:.)

```

```

__erw_timestamp_time_dec:
__erw_timestamp_time_hex:
218 \cs_new:Nn __erw_timestamp_time_dec:
219 {
220   \int_eval:n
221   {
222     \c_sys_hour_int * 100
223     +\c_sys_minute_int * 1

```

```

224     }
225 }
226 \cs_new:Nn\__erw_timestamp_time_hex:{\int_to_hex:n{\__erw_timestamp_time_dec:}}

(End definition for \__erw_timestamp_time_dec: and \__erw_timestamp_time_hex.)

\__erw_timestamp_datetime_base:n
\__erw_timestamp_datetime_dec:n
\__erw_timestamp_datetime_join:nn
\__erw_timestamp_datetime_hex:n
\__erw_timestamp_datetime_period:n
227 \cs_new:Nn\__erw_timestamp_datetime_base:n
228 {
229   \int_case:nnTF{#1}
230   {
231     {10}{dec}
232     {16}{hex}
233   }
234   {\c_empty_tl}
235   {\msg_error:nnn{\__erw}{timestamp / base}{\__erw_timestamp_datetime_base:n{#1}}}
236 }
237 \cs_new:Nn\__erw_timestamp_datetime_join:nnn{#1#2#3}
238 \cs_new:Nn\__erw_timestamp_datetime_join:nn{\__erw_timestamp_datetime_join:nnn{#1}{\g__erw_t
239 \cs_new:Nn\__erw_timestamp_datetime_period:n
240 {
241   \str_case:nnTF{#1}
242   {
243     {date}{date}
244     {time}{time}
245     {datetime}{datetime}
246   }
247   {\c_empty_tl}
248   {\msg_error:nnn{\__erw}{timestamp / period}{\__erw_timestamp_datetime_period:n{#1}}}
249 }
250 \cs_new:Nn\__erw_timestamp_datetime_dec: {\__erw_timestamp_datetime_join:nn{\__erw_timestamp
251 \cs_new:Nn\__erw_timestamp_datetime_hex: {\__erw_timestamp_datetime_join:nn{\__erw_timestamp

(End definition for \__erw_timestamp_datetime_base:n and others.)

\__erw_timestamp:nn
252 \cs_new:Nn\__erw_timestamp:nn
253 {
254   \exp_args:No
255   \use:c{\__erw_timestamp\___erw_timestamp_datetime_period:n{#1}\__erw_timestamp_datetime_bas
256 }
257 \cs_new_protected:Nn \__erw_timestamp_set_delim:nn
258 {
259   \use:c{tl_gset:N#1}
260   \g__erw_timestamp_delim_str{#2}
261 }

(End definition for \__erw_timestamp:nn.)

262 \keys_define:nn{\__erw}
263 {
264   timestamp / delim .code:n =
265   {
266     \exp_last_unbraced:No
267     \__erw_timestamp_set_delim:nn{n}{#1}

```

```

268 },
269 timestamp / delim .value_required:n = true,
270 timestamp / delim .default:n = {-},
271 timestamp / delim .initial:n = {-}
272 }

```

## 6.2 frontend

```

273 \cs_new:Nn\erw_timestamp:nn
274 {
275   \__erw_timestamp:nn{#1}{#2}
276 }
277 \cs_new:Nn\erw_timestamp:
278 {
279   \__erw_timestamp:nn{datetime}{16}
280 }

```

## 7 tl

### 7.1 backend

\\_\_erw\_oper\_function:n

```

281 \cs_new_protected:Nn \__erw_oper_function:n
282 {
283   \msg_error:nnn
284   {erw}
285   {notset}
286   {\__erw_oper_function:n}
287 }

```

*(End definition for \\_\_erw\_oper\_function:n.)*

\\_\_erw\_map:nn

```

288 \cs_set_protected:Nn \__erw_map:nn
289 {
290   \quark_if_recursion_tail_stop:n{#1}
291   \__erw_oper_function:n{#1} \__erw_map:nn{#2}
292 }

```

*(End definition for \\_\_erw\_map:nn.)*

\\_\_erw\_map\_thread\_at:Nnn

\\_\_erw\_map\_thread\_at:Nnnn

\\_\_erw\_map\_thread\_at:Nnnnn

\\_\_erw\_map\_thread\_at:Nnnnnn

```

293 \cs_set_protected:Nn \__erw_map_thread_at:Nnn
294 {
295   \erw_cs_apply:Nn #1
296   {\exp_args:Nf\tl_item:nn {#3} {#2} }
297 }
298 \cs_set_protected:Nn \__erw_map_thread_at:Nnnn
299 {
300   \erw_cs_apply:Nnn #1
301   {\exp_args:Nf\tl_item:nn {#3} {#2} }
302   {\exp_args:Nf\tl_item:nn {#4} {#2} }
303 }
304 \cs_set_protected:Nn \__erw_map_thread_at:Nnnnn

```

```

305 {
306   \erw_cs_apply:Nnnn #1
307   {\exp_args:Nf\tl_item:nn {#3} {#2} }
308   {\exp_args:Nf\tl_item:nn {#4} {#2} }
309   {\exp_args:Nf\tl_item:nn {#5} {#2} }
310 }
311 \cs_set_protected:Nn \__erw_map_thread_at:Nnnnnn
312 {
313   \erw_cs_apply:Nnnnn #1
314   {\exp_args:Nf\tl_item:nn {#3} {#2} }
315   {\exp_args:Nf\tl_item:nn {#4} {#2} }
316   {\exp_args:Nf\tl_item:nn {#5} {#2} }
317   {\exp_args:Nf\tl_item:nn {#6} {#2} }
318 }

```

(End definition for \\_\_erw\_map\_thread\_at:Nnn and others.)

## 7.2 frontend

```

319 \cs_set:Nn \erw_tl_last_item:n
320 {
321   \exp_args:Nof \tl_item:nn
322   {#1}
323   {
324     \tl_count:n{#1}
325   }
326 }
327 \cs_set_protected:Nn \erw_tl_map:n
328 {
329   \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
330 }
331 \cs_set_protected:Nn \erw_tl_map:Nn
332 {
333   \cs_set_eq:NN \__erw_oper_function:n #1
334   \erw_tl_map:n{#2}
335 }
336 \cs_set_protected:Nn \erw_tl_map_inline:nn
337 {
338   \erw_cs_set_inline:Nn \__erw_oper_function:n {#1}
339   \erw_tl_map:n{#2}
340 }
341 \cs_set:Nn \erw_tl_merge:nn
342 {
343   {#1#2}
344 }
345 \cs_set:Nn \erw_tl_repeat:nn
346 {
347   \int_step_inline:nnnn{1}{1}{#1}{#2}
348 }
349 \cs_set_protected:Nn \erw_tl_split:nnn
350 {
351   \tl_head:n{#1}
352   \use:c{exp_args:#3} \tl_map_inline:nn
353   {

```

```

354     \tl_tail:n
355     {
356         #1
357     }
358     }{#2##1}
359 }
360 \cs_set_protected:Nn \erw_tl_split:nn
361 {
362     \erw_tl_split:nnn{#1}{#2}{Nf}
363 }
364 \cs_set_protected:Nn \erw_tl_map_thread_at:Nnn
365 {
366     \exp_args:Nf\int_case:nnTF
367     {
368         \tl_count:n{#3}
369     }
370     {
371         {1}{ \__erw_map_thread_at:Nnn #1{#2}#3 }
372         {2}{ \__erw_map_thread_at:Nnnn #1{#2}#3 }
373         {3}{ \__erw_map_thread_at:Nnnnn #1{#2}#3 }
374         {4}{ \__erw_map_thread_at:Nnnnnn #1{#2}#3 }
375     }
376     {
377         % Do nothing
378     }
379     {
380         \msg_error:nnn{__erw}
381         {generic}
382         {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
383     }
384 }
385 \cs_set_protected:Nn \erw_tl_map_thread:Nn
386 {
387     % TODO check that #2 is a matrix
388     \int_step_inline:nn
389     {
390         \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
391     }
392     {
393         \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
394     }
395 }

```

## 8 option

```

396 \cs_new_protected:Nn\erw_option:n
397 {
398     \keys_set:nn{__erw}{#1}
399 }

```

## 9 Closing

```

400 \ExplSyntaxOff

```