

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

Erwann Rogard<sup>†</sup>

Released 2020/05/01

## 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	sys	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 v1.9, last revised 2020/05/01.

<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>sys</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>Change History</b>	<b>13</b>
	<b>Index</b>	<b>14</b>
<b>IV</b>	<b>Implementation</b>	<b>16</b>
<b>1</b>	<b>Opening</b>	<b>16</b>
<b>2</b>	<b>basics</b>	<b>16</b>
2.1	backend .....	16
2.2	frontend .....	16
<b>3</b>	<b>csint</b>	<b>17</b>
3.1	backend .....	17
3.2	frontend .....	17

<b>4</b>	<b>int</b>	<b>17</b>
4.1	backend	17
4.2	frontend	18
<b>5</b>	<b>msg</b>	<b>18</b>
5.1	backend	18
<b>6</b>	<b>oper</b>	<b>18</b>
6.1	backend	18
6.2	frontend	19
<b>7</b>	<b>sys</b>	<b>20</b>
7.1	backend	20
7.2	frontend	22
<b>8</b>	<b>tl</b>	<b>22</b>
8.1	backend	22
8.2	frontend	23
<b>9</b>	<b>option</b>	<b>24</b>
<b>10</b>	<b>Closing</b>	<b>25</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_cs_apply:Nn</code>	<code>\erw_cs_apply:Nn {\&lt;control sequence&gt;}{\&lt;token list<sub>1</sub>&gt;}</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

---

<code>\erw_csint:nn</code>	<code>\erw_csint:nn{&lt;integer&gt;}{&lt;arg&gt;}</code>
----------------------------	--

---



---

<code>\erw_csint_name:n</code>	<code>\erw_csint_name:n{&lt;integer&gt;}</code>
--------------------------------	---

---



---

<code>\erw_csint_names:nnn</code>	<code>\erw_csint_names:nnn{&lt;integer&gt;}{&lt;integer&gt;}{&lt;integer&gt;}</code>
-----------------------------------	--

---



---

<code>\erw_csint_names_braced:</code>	
<code>\erw_csint_names_braced:n</code>	
<code>\erw_csint_names_braced:nnn</code>	

---



---

<code>\erw_csint_new:n</code>	<code>\erw_csint_new:n{&lt;integer&gt;}</code>
-------------------------------	--

---



---

<code>\erw_csint_reset:</code>	<code>\erw_csint_reset:</code>
--------------------------------	--------------------------------

---

### 4 int

---

<code>\erw_int_range:n</code>	<code>\erw_int_range:n{&lt;integer&gt;}</code>
<code>\erw_int_range:nn</code>	

---

### 5 oper

---

<code>\erw_oper_compose:nN</code>	<code>\erw_oper_compose:nn{&lt;control sequence list&gt;}{&lt;initial value&gt;}</code>
<code>\erw_oper_compose:nn</code>	

---



---

<code>\erw_oper_compose_c:nN</code>	
<code>\erw_oper_compose_c:nn</code>	

---



---

<code>\erw_oper_compose_vers:nN</code>	
<code>\erw_oper_compose_vers:nn</code>	

---

---

```

\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_fold_seq:NN
\erw_seq:cN

```

---

## 6 sys

---

```

\erw_sys_timestamp:nn \erw_sys_timestamp:nn{date|time|datetime}{10|16}
\erw_sys_timestamp:

```

---

**Semantics** Timestamp in base 10 or 16

---

```

\erw_sys_timestamp_delimiter: \erw_sys_timestamp_delimiter:

```

---

## 7 tl

---

```

\erw_tl_append_item:nn \erw_tl_append_item:nn{<arg list>}{<arg>}

```

---



---

```

\erw_tl_gset_function:N
\erw_tl_gset_function:n

```

---



---

```

\erw_tl_join:nn \erw_tl_join:n{<token list1>}{<token list2>}
\erw_tl_join:nnn
\erw_tl_join:nnnn
\erw_tl_join:nnnnn

```

---



---

```

\erw_tl_last_item:n \erw_tl_last_time:n{<token list>}

```

---



---

```

\erw_tl_map:n
\erw_tl_map:Nn

```

---



---

```

\erw_tl_map_inline:nn

```

---

---

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

---

---

`\erw_tl_merge:nn`

---

---

`\erw_tl_repeat:nn`

---

---

`\erw_tl_split:nnn`  
`\erw_tl_split: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 sys

### Listing 10.

```
\ExplSyntaxOn
\noindent\erw_sys_timestamp:nn{date}{10}{-}
\noindent\erw_sys_timestamp:nn{time}{10}{}
\noindent\erw_sys_timestamp:nn{datetime}{10}{}
\erw_sys_timestamp:nn{date}{16}{\%}
\erw_sys_timestamp:nn{time}{16}{}
\erw_option:n{ sys / timestamp_delim = {\%} }
\erw_sys_timestamp:nn{datetime}{16}{}
\ExplSyntaxOff
```

---

```
20200501-1557
20200501-1557
1343c35%615
1343c35%615
```

### Listing 11.

```
\ExplSyntaxOn
\erw_option:n{ sys / timestamp_delim = \c_empty_tl }
\iow_new:N \foo_iow
\tl_set:Nx \foo_dec { \erw_sys_timestamp:nn{datetime}{10} }
\tl_set:Nx \foo_hex { \erw_sys_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:202005011557
D:20200501155741-04'00'
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>

# Change History

v0.0	General: Initial version . . . . .	12	Remove: <code>disambig</code> . . . . .	12
v0.1	General: <code>\numbrdcsnew</code> changed to <code>\newnumbrdcs</code> and made <code>'disambiguable'</code> . . . . .	12	Split Section Preliminaries into Conventions and Requirement. . . . .	12
	<code>disambig/</code> backend: changes to the key, added <code>\ProcessPackageKeysOption;</code> . . . . .	12		
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ; . . . . .	12	v0.6	General: Fix: critical bug preventing <code>erw-l3</code> from working without explicit inclusion of <code>expl3</code> . . . . .
v0.2	General: . . . . .	12	v0.7	General: Add: <code>Closing</code> . . . . .
	<code>\erw_compose</code> reversed order in which the functions are composed, such that it now conforms to the mathematical convention ( $g \circ f$ means $f$ comes before $g$ ) . . . . .	12		Add: <code>sys</code> . . . . .
	<code>disambig</code> : pushed the code inside <code>\keys_define;\disambignewcmd</code> no longer takes a token name as arg, rather a token. . . . .	12		Move: <code>\erw_fold_apply_par:n</code> . . . . .
	Add: <code>\erw_items_to</code> . . . . .	12		Move: <code>\erw_fold_set_par:n</code> . . . . .
	Add: <code>\erw_last_item</code> . . . . .	12		Rearrange: structure of implementation, e.g. <code>section 6</code> . . . . .
	Add: <code>\erw_repeat</code> . . . . .	12		Remove: document level functions, <code>\numbrdcsnew</code> , <code>\numbrdcs</code> . . . . .
	Add: <code>\erw_split</code> . . . . .	12		Replace: listing's implem with that of <code>tocloft</code> . . . . .
	Add: <code>\map_thread</code> . . . . .	12		Replace: vers. numb. from 3 to 2 digits . . . . .
	Front end cmds no longer generated with module <code>disambig</code> ; Option of the same name deleted; . . . . .	12	v0.8	General: Add: function for all frontend functions. . . . .
	Re-arrange: the doc to clearly separate frontend from backend . . . . .	12		Remove: <code>\erw_cs_set_eq:NN</code> and variants . . . . .
v0.3	General: Replace: versioning, should have been 0.1.2 . . . . .	12		Remove: <code>\erw_is_matrix:n</code> (predicate must be expandable) . . . . .
v0.4	General: Add: <code>\erw_accum</code> . . . . .	12		Rename: all cs prefixes to agree with heading under which they come, e.g. <code>\erw_identity:n</code> by <code>\erw_cs_identity:n</code> . . . . .
	Add: <code>\erw_int_range</code> . . . . .	12		Replace: <code>@@_map:n</code> by <code>@@_oper_function:n</code> . . . . .
	Add: <code>\erw_is_matrix</code> (to check arg of <code>\erw_tl_map_thread:Nn</code> ) . . . . .	12		Replace: <code>\erw_seq_fold:NN</code> by <code>\erw_oper_fold_seq:NN</code> and likewise for variants . . . . .
	Add: <code>\erw_merge</code> . . . . .	12	v0.9	General: Add:
	Add: <code>\erw_set_map_inline</code> . . . . .	12		<code>\erw_sys_timestamp_delimiter:</code> . . . . .
	Add: <code>\erw_set_map</code> . . . . .	12		Add: <code>\erw_tl_join:nn</code> and variants . . . . .
	Remove: <code>\erw_items_to</code> (redundant with <code>\tl_range:nnn</code> ) . . . . .	12		Rename: <code>\erw_append_arg:nn</code> to <code>\erw_tl_append_item:nn</code> . . . . .
v0.5	General: Modify: source repository . . . . .	12		Rename:
	Rearrange: frontend/backend sections . . . . .	12		<code>\erw_oper_gset_function:N</code> to <code>\erw_tl_gset_function:N</code> (and variants) . . . . .

# Index

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

## C

cs commands:

\cs\_generate\_variant:Nn .....  
                                   15, 33, 38, 197, 204  
 \cs\_gset:Npn ..... 36  
 \cs\_new:Nn ..... 39, 40, 41, 42, 50,  
                                   65, 71, 207, 216, 217, 218, 226, 227,  
                                   237, 238, 249, 250, 251, 272, 276, 280  
 \cs\_new\_protected:Nn .....  
                                   54, 75, 256, 284, 398  
 \cs\_set:Nn . 3, 7, 11, 16, 20, 24, 29,  
                                   34, 45, 49, 70, 101, 105, 116, 127,  
                                   131, 137, 141, 147, 151, 160, 164,  
                                   168, 172, 176, 191, 198, 322, 344, 348  
 \cs\_set:Npn ..... 28, 31, 80  
 \cs\_set\_eq:NN ..... 336  
 \cs\_set\_protected:Nn .....  
                                   291, 296, 301, 307,  
                                   314, 330, 334, 339, 352, 363, 367, 388  
 \cs\_split\_function:N ..... 5

## E

erw commands:

\erw\_append\_arg:nn ..... 3, 7, 92  
 \erw\_cs\_apply:Nn . 3, 3, 11, 15, 47, 298  
 \erw\_cs\_apply:Nnn ..... 3, 16, 303  
 \erw\_cs\_apply:Nnnn ..... 3, 20, 309  
 \erw\_cs\_apply:Nnnnn ..... 3, 24, 316  
 \erw\_cs\_gset\_eq:NN ..... 174  
 \erw\_cs\_gset\_inline:Nn ... 34, 38, 178  
 \erw\_cs\_identity:n ..... 3, 28  
 \erw\_cs\_set\_inline:Nn .....  
                                   4, 29, 33, 57, 118, 341  
 \erw\_csint:nn ..... 4, 45  
 \erw\_csint\_name:n ... 4, 44, 49, 52, 70  
 \erw\_csint\_names:nnn ..... 4, 50  
 \erw\_csint\_names\_braced: .. 4, 71, 157  
 \erw\_csint\_names\_braced:n .. 4, 67, 70  
 \erw\_csint\_names\_braced:nnn 4, 65, 73  
 \erw\_csint\_new:n ..... 4, 54, 154  
 \erw\_csint\_reset: ..... 4, 75, 153  
 \erw\_int\_range:n ..... 4, 105  
 \erw\_int\_range:nn ..... 4, 101  
 \erw\_oper\_compose:nN .... 4, 127, 134  
 \erw\_oper\_compose:nn ..... 4, 131  
 \erw\_oper\_compose\_c:nN ... 4, 137, 144  
 \erw\_oper\_compose\_c:nn ... 4, 141, 156  
 \erw\_oper\_compose\_seq:nN .... 5, 160

\erw\_oper\_compose\_seq\_c:nN .. 5, 164  
 \erw\_oper\_compose\_seq\_vers:nN ...  
                                   5, 168, 170  
 \erw\_oper\_compose\_vers:nN 4, 147, 149  
 \erw\_oper\_compose\_vers:nn ... 4, 151  
 \erw\_oper\_fold:NN .....  
                                   5, 129, 139, 191, 197, 201  
 \erw\_oper\_fold\_seq:NN .....  
                                   5, 162, 166, 198, 204  
 \erw\_option:n ..... 6, 398  
 \erw\_seq:NN ..... 5  
 \erw\_sys:nn ..... 5  
 \erw\_sys\_timestamp: ..... 5, 280  
 \erw\_sys\_timestamp:nn ..... 5, 276  
 \erw\_sys\_timestamp\_delimiter: 5, 272  
 \erw\_tl\_gset\_function:N ..... 5, 172  
 \erw\_tl\_gset\_function:n ..... 5, 176  
 \erw\_tl\_join:n ..... 5  
 \erw\_tl\_join:nn ..... 5, 39  
 \erw\_tl\_join:nnn ..... 5, 40, 237  
 \erw\_tl\_join:nnnn ..... 5, 41  
 \erw\_tl\_join:nnnnn ..... 5, 42  
 \erw\_tl\_last\_item:n ..... 5, 322  
 \erw\_tl\_last\_time:n ..... 5  
 \erw\_tl\_map:n ... 5, 122, 330, 337, 342  
 \erw\_tl\_map:Nn ..... 5, 334  
 \erw\_tl\_map\_inline:nn ..... 5, 339  
 \erw\_tl\_map\_thread:Nn ..... 6, 388  
 \erw\_tl\_map\_thread\_at:Nnn 6, 367, 395  
 \erw\_tl\_merge:nn ..... 5, 344  
 \erw\_tl\_repeat:nn ..... 6, 348  
 \erw\_tl\_split:nn ..... 6, 363  
 \erw\_tl\_split:nnn ..... 6, 352, 365

erw internal commands:

\g\_\_erw\_compose\_tl .....  
                                   114, 133, 134, 135, 143, 144, 145  
 \\_\_erw\_cs\_name:N ..... 3  
 \\_\_erw\_csint\_ext\_tl ..... 78  
 \g\_\_erw\_csint\_int .. 43, 44, 56, 73, 77  
 \g\_\_erw\_csint\_name\_tl ..... 44, 57  
 \\_\_erw\_int\_range:nnn 80, 90, 103, 107  
 \\_\_erw\_map:nn ..... 291, 332  
 \\_\_erw\_map\_thread\_at:Nnn ... 296, 374  
 \\_\_erw\_map\_thread\_at:Nnnn .. 296, 375  
 \\_\_erw\_map\_thread\_at:Nnnnn . 296, 376  
 \\_\_erw\_map\_thread\_at:Nnnnnn 296, 377  
 \\_\_erw\_oper\_compose:NnN .....  
                                   116, 129, 139, 162, 166



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



## 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_tl_append_item: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}

```

## 6 oper

### 6.1 backend

```

110 \tl_new:N \g__erw_compose_tl
111 \tl_new:N \g__erw_oper_fold_seq_item_tl
112 \cs_set:Nn \__erw_oper_compose:NnN
113 {
114   \erw_cs_set_inline:Nn \__erw_tl_function:n
115   {
116     #1{##1}#3
117   }
118   \exp_args:Nf\erw_tl_map:n
119   {
120     \tl_reverse:n{#2}
121   }
122 }

```

## 6.2 frontend

```

123 \cs_set:Nn \erw_oper_compose:nN
124 {
125   \__erw_oper_compose:NnN \erw_oper_fold:NN {#1} #2
126 }
127 \cs_set:Nn \erw_oper_compose:nn
128 {
129   \tl_set:Nn \g__erw_compose_tl {#2}
130   \erw_oper_compose:nN{#1}\g__erw_compose_tl
131   \g__erw_compose_tl
132 }
133 \cs_set:Nn \erw_oper_compose_c:nN
134 {
135   \__erw_oper_compose:NnN \erw_oper_fold:cN {#1} #2
136 }
137 \cs_set:Nn \erw_oper_compose_c:nn
138 {
139   \tl_set:Nn \g__erw_compose_tl {#2}
140   \erw_oper_compose_c:nN{#1}\g__erw_compose_tl
141   \g__erw_compose_tl
142 }
143 \cs_set:Nn \erw_oper_compose_vers:nN
144 {
145   \msg_error:nnn{__erw}{notdecl}{\erw_oper_compose_vers:nN}
146 }
147 \cs_set:Nn \erw_oper_compose_vers:nn
148 {
149   \erw_csint_reset:{}
150   \tl_map_function:nN{#1}\erw_csint_new:n
151   \exp_last_unbraced:Nx
152   \erw_oper_compose_c:nn
153   {\erw_csint_names_braced:{}}
154   {#2}
155 }
156 \cs_set:Nn \erw_oper_compose_seq:nN
157 {
158   \__erw_oper_compose:NnN \erw_oper_fold_seq:NN {#1} #2
159 }
160 \cs_set:Nn \erw_oper_compose_seq_c:nN
161 {
162   \__erw_oper_compose:NnN \erw_oper_fold_seq:cN {#1} #2
163 }
164 \cs_set:Nn \erw_oper_compose_seq_vers:nN
165 {
166   \msg_error:nnn{__erw}{notdecl}{\erw_oper_compose_seq_vers:nN}
167 }
168 \cs_set:Nn \erw_tl_gset_function:N
169 {
170   \erw_cs_gset_eq:NN \__erw_tl_function:n #1
171 }
172 \cs_set:Nn \erw_tl_gset_function:n
173 {
174   \erw_cs_gset_inline:Nn \__erw_tl_function:n {#1}

```

```

175 }
176 \keys_define:nn{__erw}
177 {
178   oper/fold_set_par.tl_gset:N = \g__erw_oper_fold_set_par_tl,
179   oper/fold_set_par.value_required:n = true,
180   oper/fold_set_par.default:n = {Nf},
181   oper/fold_set_par.initial:n = {Nf},
182   oper/fold_apply_par.tl_gset:N = \g__erw_oper_fold_apply_par_tl,
183   oper/fold_apply_par.value_required:n = true,
184   oper/fold_apply_par.default:n = {Nf},
185   oper/fold_apply_par.initial:n = {Nf}
186 }
187 \cs_set:Nn \erw_oper_fold:NN
188 {
189   \use:c{tl_set:\g__erw_oper_fold_set_par_tl}
190   #2
191   {\use:c{erw_cs_apply:\g__erw_oper_fold_apply_par_tl}{#1}{#2}}
192 }
193 \cs_generate_variant:Nn \erw_oper_fold:NN {cN}
194 \cs_set:Nn \erw_oper_fold_seq:NN
195 {
196   \seq_get_right:NN #2 \g__erw_oper_fold_seq_item_tl
197   \erw_oper_fold:NN #1 \g__erw_oper_fold_seq_item_tl
198   \seq_put_right:No #2 {\g__erw_oper_fold_seq_item_tl}
199 }
200 \cs_generate_variant:Nn \erw_oper_fold_seq:NN {cN}

```

## 7 sys

### 7.1 backend

```

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

```

```

__erw_sys_date:N
__erw_sys_date_dec:
__erw_sys_date_hex:

```

```

203 \cs_new:Nn \__erw_sys_date_dec:
204 {
205   \int_eval:n
206   {
207     \c_sys_year_int * 10000
208     +\c_sys_month_int * 100
209     +\c_sys_day_int * 1
210   }
211 }
212 \cs_new:Nn \__erw_sys_date:N{\int_to_hex:n{\__erw_sys_date_dec:}}
213 \cs_new:Nn \__erw_sys_date_hex:{\int_to_hex:n{\__erw_sys_date_dec:}}

```

(End definition for `\__erw_sys_date:N`, `\__erw_sys_date_dec:`, and `\__erw_sys_date_hex:`.)

```

__erw_sys_time_dec:
__erw_sys_time_hex:

```

```

214 \cs_new:Nn \__erw_sys_time_dec:
215 {
216   \int_eval:n
217   {
218     \c_sys_hour_int * 100

```

```

219     +\c_sys_minute_int * 1
220   }
221 }
222 \cs_new:Nn\__erw_sys_time_hex:{\int_to_hex:n{\__erw_sys_time_dec:}}

```

(End definition for \\_\_erw\_sys\_time\_dec: and \\_\_erw\_sys\_time\_hex.)

```

\__erw_sys_datetime_base:n
\__erw_sys_datetime_dec:n
\__erw_sys_datetime_join:nn
\__erw_sys_datetime_hex:n
\__erw_sys_datetime_period:n
223 \cs_new:Nn\__erw_sys_datetime_base:n
224 {
225   \int_case:nnTF{#1}
226   {
227     {10}{dec}
228     {16}{hex}
229   }
230   {\c_empty_tl}
231   {\msg_error:nnn{\__erw}{timestamp / base}{\__erw_sys_datetime_base:n{#1}}}
232 }
233 \cs_new:Nn\__erw_sys_datetime_join:nn{\erw_tl_join:nnn{#1}{\g__erw_sys_timestamp_delim_str}{#2}}
234 \cs_new:Nn\__erw_sys_datetime_period:n
235 {
236   \str_case:nnTF{#1}
237   {
238     {date}{date}
239     {time}{time}
240     {datetime}{datetime}
241   }
242   {\c_empty_tl}
243   {\msg_error:nnn{\__erw}{timestamp / period}{\__erw_sys_datetime_period:n{#1}}}
244 }
245 \cs_new:Nn\__erw_sys_datetime_dec: {\__erw_sys_datetime_join:nn{\__erw_sys_date_dec:}{\__erw_sys_date_dec:}}
246 \cs_new:Nn\__erw_sys_datetime_hex: {\__erw_sys_datetime_join:nn{\__erw_sys_date_hex:}{\__erw_sys_date_hex:}}

```

(End definition for \\_\_erw\_sys\_datetime\_base:n and others.)

\\_\_erw\_sys\_timestamp:nn

```

247 \cs_new:Nn\__erw_sys_timestamp:nn
248 {
249   \exp_args:No
250   \use:c{\__erw_sys\___erw_sys_datetime_period:n{#1}\__erw_sys_datetime_base:n{#2}:}
251 }
252 \cs_new_protected:Nn \__erw_sys_set_delim:nn
253 {
254   \use:c{tl_gset:N#1}
255   \g__erw_sys_timestamp_delim_str{#2}
256 }

```

(End definition for \\_\_erw\_sys\_timestamp:nn.)

```

257 \keys_define:nn{\__erw}
258 {
259   sys / timestamp_delim .code:n =
260   {
261     \exp_last_unbraced:No
262     \__erw_sys_set_delim:nn{n}{#1}

```

```

263 },
264 sys / timestamp_delim .value_required:n = true,
265 sys / timestamp_delim .default:n = {-},
266 sys / timestamp_delim .initial:n = {-}
267 }

```

## 7.2 frontend

```

268 \cs_new:Nn\erw_sys_timestamp_delimiter:
269 {
270   \use:N \g__erw_sys_timestamp_delim_str
271 }
272 \cs_new:Nn\erw_sys_timestamp:nn
273 {
274   \__erw_sys_timestamp:nn{#1}{#2}
275 }
276 \cs_new:Nn\erw_sys_timestamp:
277 {
278   \__erw_sys_timestamp:nn{datetime}{16}
279 }

```

## 8 tl

### 8.1 backend

\\_\_erw\_tl\_function:n

```

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

```

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

\\_\_erw\_map:nn

```

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

```

*(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

```

292 \cs_set_protected:Nn \__erw_map_thread_at:Nnn
293 {
294   \erw_cs_apply:Nn #1
295   {\exp_args:Nf\tl_item:nn {#3} {#2} }
296 }
297 \cs_set_protected:Nn \__erw_map_thread_at:Nnnn
298 {
299   \erw_cs_apply:Nnn #1

```

```

300   {\exp_args:Nf\tl_item:nn {#3} {#2} }
301   {\exp_args:Nf\tl_item:nn {#4} {#2} }
302 }
303 \cs_set_protected:Nn \__erw_map_thread_at:Nnnnn
304 {
305   \erw_cs_apply:Nnnn #1
306   {\exp_args:Nf\tl_item:nn {#3} {#2} }
307   {\exp_args:Nf\tl_item:nn {#4} {#2} }
308   {\exp_args:Nf\tl_item:nn {#5} {#2} }
309 }
310 \cs_set_protected:Nn \__erw_map_thread_at:Nnnnnn
311 {
312   \erw_cs_apply:Nnnnn #1
313   {\exp_args:Nf\tl_item:nn {#3} {#2} }
314   {\exp_args:Nf\tl_item:nn {#4} {#2} }
315   {\exp_args:Nf\tl_item:nn {#5} {#2} }
316   {\exp_args:Nf\tl_item:nn {#6} {#2} }
317 }

```

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

## 8.2 frontend

```

318 \cs_set:Nn \erw_tl_append_item:nn
319 {
320   {#1{#2}}
321 }
322 \cs_set:Nn \erw_tl_last_item:n
323 {
324   \exp_args:Nof \tl_item:nn
325   {#1}
326   {
327     \tl_count:n{#1}
328   }
329 }
330 \cs_set_protected:Nn \erw_tl_map:n
331 {
332   \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
333 }
334 \cs_set_protected:Nn \erw_tl_map:Nn
335 {
336   \cs_set_eq:NN \__erw_tl_function:n #1
337   \erw_tl_map:n{#2}
338 }
339 \cs_set_protected:Nn \erw_tl_map_inline:nn
340 {
341   \erw_cs_set_inline:Nn \__erw_tl_function:n {#1}
342   \erw_tl_map:n{#2}
343 }
344 \cs_set:Nn \erw_tl_merge:nn
345 {
346   {#1#2}
347 }
348 \cs_set:Nn \erw_tl_repeat:nn

```

```

349 {
350   \int_step_inline:nnnn{1}{1}{#1}{#2}
351 }
352 \cs_set_protected:Nn \erw_tl_split:nnn
353 {
354   \tl_head:n{#1}
355   \use:c{exp_args:#3} \tl_map_inline:nn
356   {
357     \tl_tail:n
358     {
359       #1
360     }
361   }{#2##1}
362 }
363 \cs_set_protected:Nn \erw_tl_split:nn
364 {
365   \erw_tl_split:nnn{#1}{#2}{Nf}
366 }
367 \cs_set_protected:Nn \erw_tl_map_thread_at:Nnn
368 {
369   \exp_args:Nf\int_case:nnTF
370   {
371     \tl_count:n{#3}
372   }
373   {
374     {1}{ \__erw_map_thread_at:Nnn #1{#2}#3 }
375     {2}{ \__erw_map_thread_at:Nnnn #1{#2}#3 }
376     {3}{ \__erw_map_thread_at:Nnnnn #1{#2}#3 }
377     {4}{ \__erw_map_thread_at:Nnnnnn #1{#2}#3 }
378   }
379   {
380     % Do nothing
381   }
382   {
383     \msg_error:nnn{__erw}
384     {generic}
385     {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
386   }
387 }
388 \cs_set_protected:Nn \erw_tl_map_thread:Nn
389 {
390   \int_step_inline:nn
391   {
392     \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
393   }
394   {
395     \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
396   }
397 }

```

## 9 option

```

398 \cs_new_protected:Nn\erw_option:n
399 {

```



```
400 \keys_set:nn{__erw}{#1}  
401 }
```

## 10 Closing

```
402 \ExplSyntaxOff
```