

# 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	seq	4
6	sys	5
7	tl	5
9	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>seq</b>	<b>8</b>
4.	.....	8
5.	.....	8
10.	.....	8
<b>5</b>	<b>sys</b>	<b>9</b>
6.	.....	9
7.	.....	9
<b>6</b>	<b>tl</b>	<b>9</b>
8.	.....	9
9.	.....	10
11.	.....	10
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>17</b>
<b>1</b>	<b>Opening</b>	<b>17</b>
<b>2</b>	<b>basics</b>	<b>17</b>
2.1	backend .....	17
2.2	frontend .....	17
<b>3</b>	<b>csint</b>	<b>18</b>
3.1	backend .....	18
3.2	frontend .....	18

<b>4</b>	<b>int</b>	<b>18</b>
4.1	backend	18
4.2	frontend	19
4.3	frontend	19
<b>5</b>	<b>msg</b>	<b>19</b>
5.1	backend	19
<b>6</b>	<b>oper</b>	<b>19</b>
6.1	backend	19
6.2	frontend	20
<b>7</b>	<b>seq</b>	<b>20</b>
7.1	backend	20
7.2	frontend	20
<b>8</b>	<b>sys</b>	<b>20</b>
8.1	backend	20
<b>9</b>	<b>tl</b>	<b>23</b>
9.1	backend	23
9.2	frontend	24
<b>10</b>	<b>option</b>	<b>26</b>
<b>11</b>	<b>Closing</b>	<b>26</b>

## Part I

# Usage

---

```
\usepackage \usepackage{erw-l3}
```

---

### 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

---

```
\erw_cs_apply:Nn \erw_cs_apply:Nn {\control sequence}\{token list_1\}
\erw_cs_apply:(No|Nf|Nx|cn)
\erw_cs_apply:Nnn
\erw_cs_apply:Nnnn
\erw_cs_apply:Nnnnn
```

---

---

<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 seq

---

<code>\erw_seq_compose:nN</code>	
<code>\erw_seq_compose_c:nN</code>	
<code>\erw_seq_compose_vers:nN</code>	

---



---

<code>\erw_seq_fold:NN</code>	
<code>\erw_seq_fold:cN</code>	

---

## 6 sys

---

<code>\erw_sys_jobnametimestamp:nn</code>	<code>\erw_sys_jobnametimestamp:nn{date time datetime}{10 16}</code>
<code>\erw_sys_jobnametimestamp:</code>	

---



---

<code>\erw_sys_timestamp:nn</code>	<code>\erw_sys_timestamp:nn{date time datetime}{10 16}</code>
<code>\erw_sys_timestamp:</code>	<b>Semantics</b> Timestamp in base 10 or 16

---



---

<code>\erw_sys_timestamp_delimiter:</code>	<code>\erw_sys_timestamp_delimiter:</code>
--------------------------------------------	--------------------------------------------

---

## 7 tl

---

<code>\erw_tl_append_item:nn</code>	<code>\erw_tl_append_item:nn{&lt;arg list&gt;}{&lt;arg&gt;}</code>
-------------------------------------	--------------------------------------------------------------------

---



---

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

---



---

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

---



---

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

---



---

<code>\erw_tl_fold:NN</code>	<code>\erw_tl_fold:NN&lt;control sequence&gt;&lt;token&gt;</code>
<code>\erw_tl_fold:cN</code>	

---



---

<code>\erw_tl_gset_function:N</code>	<code>\erw_tl_gset_function:n{&lt;code&gt;}</code>
<code>\erw_tl_gset_function:n</code>	

---



---

<code>\erw_tl_join:nn</code>	<code>\erw_tl_join:nn{&lt;token list<sub>1</sub>&gt;}{&lt;token list<sub>2</sub>&gt;}</code>
<code>\erw_tl_join:nnn</code>	
<code>\erw_tl_join:nnnn</code>	
<code>\erw_tl_join:nnnnn</code>	

---



---

<code>\erw_tl_last_item:n</code>	<code>\erw_tl_last_time:n{&lt;token list&gt;}</code>
----------------------------------	------------------------------------------------------

---

---

<code>\erw_tl_map:n</code>	<code>\erw_tl_map:n{&lt;items&gt;}</code>
<code>\erw_tl_map:Nn</code>	

---

**Semantics** Maps over  $\langle items \rangle$  using the internal function set by `\erw_tl_gset_-function:n`

---

<code>\erw_tl_map_inline:nn</code>
------------------------------------

---



---

<code>\erw_tl_map_thread_at:Nnn</code>
<code>\erw_tl_map_thread:Nn</code>

---



---

<code>\erw_tl_repeat:nn</code>	<code>\erw_tl_repeat:nn{&lt;integer&gt;}{&lt;token list&gt;}</code>
--------------------------------	---------------------------------------------------------------------

---



---

<code>\erw_tl_split:nnn</code>	<code>\erw_tl_split:nn{&lt;items&gt;}{&lt;delimiter&gt;}</code>
<code>\erw_tl_split:nn</code>	

---

## 9 option

---

<code>\erw_option:n</code>
----------------------------

---

## 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_tl_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 seq

Listing 4.

```
\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_seq_compose: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 5.

```
\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_seq_compose_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)]}



## 5 sys

Listing 6.

```
\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}\\
\erw_sys_jobnametimestamp:
\ExplSyntaxOff
```

```
20200502-939
20200502-939
1343c36%3ab
1343c36%3ab
erw-l3%1343c36%3ab
```

Listing 7.

```
\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:20200502939
D:20200502093959-04'00'
Hello, world!
```

## 6 tl

Listing 8.

```
\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_tl_compose:nN{\__baz:n}{\__bar:n}{\__foo:n}}\l_tmpa_tl
\l_tmpa_tl\
\tl_set:Nn \l_tmpa_tl{X}
\erw_tl_compose:nn{\__baz:n}{\__bar:n}{\__foo:n}}{X}\
\ExplSyntaxOff

```

---

```

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

```

#### Listing 9.

```

\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_tl_compose_c:nN{\__baz:n}{\__bar:n}{\__foo:n}}\l_tmpa_tl
\l_tmpa_tl\
\erw_tl_compose_c:nn{\__baz:n}{\__bar:n}{\__foo:n}}{X}
\ExplSyntaxOff

```

---

```

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

```

#### Listing 10.

```

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

```

---

```

h{g[f(X)]}

```

#### Listing 11.

```

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

```

---

f(X)  
g[f(X)]

#### 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

v1.0	General: Initial version . . . . .	12	Split Section Preliminaries into Conventions and Requirement. . .	12
v1.1	General: <code>\numbrdcsnew</code> changed to <code>\newnumbrdcs</code> and made 'disambiguable' . . . . .	12	v1.6	General: Fix: critical bug preventing <code>erw-l3</code> from working without explicit inclusion of <code>expl3</code> . . . . .
	<code>disambig/backend</code> : changes to the key, added <code>\ProcessPackageKeysOption</code> ; . . .	12	v1.7	General: Add: <b>Closing</b> . . . . .
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ; . . .	12		Add: <b>sys</b> . . . . .
v1.2	General: . . . . .	12		Move: <code>\erw_fold_apply_par:n</code> . .
	<code>\erw_compose</code> reversed order in which the functions are composed, such that it now conforms to the mathematical cInvention ( $g \circ f$ means $f$ comes before $g$ ) . . . . .	12		Move: <code>\erw_fold_set_par:n</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		Rearrange: structure of implementation, e.g. <b>section 6</b> . .
	Add: <code>\erw_items_to</code> . . . . .	12		Remove: document level functions, <code>\numbrdcsnew</code> , <code>\numbrdcs</code> . . . . .
	Add: <code>\erw_last_item</code> . . . . .	12		Replace: listing's implem with that of <code>tocloft</code> . . . . .
	Add: <code>\erw_repeat</code> . . . . .	12		Replace: vers. numb. from 3 to 2 digits . . . . .
	Add: <code>\erw_split</code> . . . . .	12	v1.8	General: Add: function for all frontend functions. . . . .
	Add: <code>\map_thread</code> . . . . .	12		Remove: <code>\erw_cs_set_eq:NN</code> and variants . . . . .
	Front end cmds no longer generated with module <code>disambig</code> ; Option of the same name deleted; . . . . .	12		Remove: <code>\erw_is_matrix:n</code> (predicate must be expandable) . .
	Re-arrange: the doc to clearly separate frontend from backend . .	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> . . . . .
v1.3	General: Replace: versioning, should have been 0.1.2 . . . . .	12		Replace: <code>@@_map:n</code> by <code>@@_oper_function:n</code> . . . . .
v1.4	General: Add: <code>\erw_accum</code> . . . . .	12		Replace: <code>\erw_seq_fold:NN</code> by <code>\erw_oper_fold_seq:NN</code> and likewise for variants . . . . .
	Add: <code>\erw_int_range</code> . . . . .	12	v1.9	General: Add:
	Add: <code>\erw_is_matrix</code> (to check arg of <code>\erw_tl_map_thread:Nn</code> ) . . . .	12		<code>\erw_sys_timestamp_delimiter:</code> .
	Add: <code>\erw_merge</code> . . . . .	12		Add: <code>\erw_tl_join:nn</code> and variants
	Add: <code>\erw_set_map_inline</code> . . . .	12		Rename: <code>\erw_append_arg:nn</code> to <code>\erw_tl_append_item:nn</code> . . . . .
	Add: <code>\erw_set_map</code> . . . . .	12		Rename:
	Remove: <code>\erw_items_to</code> (redundant with <code>\tl_range:nnn</code> ) .	12		<code>\erw_oper_gset_function:N</code> to <code>\erw_tl_gset_function:N</code> (and variants) . . . . .
v1.5	General: Modify: source repository . .	12	v2.0	General: Add:
	Rearrange: frontend/backend sections . . . . .	12		<code>\erw_jobnametimestamp:nn</code> and variants . . . . .
	Remove: <code>disambig</code> . . . . .	12		

Move: all functions under <a href="#">section 6</a> to <a href="#">section 9</a> or <a href="#">section 7</a> , except <code>\@@_oper_compose:NnN</code> . . . . .	Remove: <code>\merge:nn</code> (redundant with <code>\erw_join:nn</code> ) . . . . . Rename: v0.0 to v1.0, etc. . . . .
12	12 12

## Index

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

Symbols	
<b>@@ commands:</b>	
<code>\@@_map:n</code> . . . . .	<i>13</i>
<code>\@@_oper_function:n</code> . . . . .	<i>13</i>
<b>B</b>	
<code>\begin</code> . . . . .	204, 205, <i>240</i>
<b>C</b>	
cs commands:	
<code>\cs_generate_variant:Nn</code> . . . . .	11, 29, 34, <i>151</i> , <i>337</i>
<code>\cs_gset:Npn</code> . . . . .	<i>32</i>
<code>\cs_new:Nn</code> . . . . .	35, 36, 37, 38, 46, 61, 67, 154, 163, 164, 165, 173, 174, 184, 185, 196, 197, 198, 206, 212, 218, 241, 242, 243, 247, <i>251</i>
<code>\cs_new_protected:Nn</code> . . . . .	50, 71, 223, 256, <i>418</i>
<code>\cs_set:Nn</code> . . . . .	3, 7, 12, 16, 20, 25, 30, 41, 45, 66, 97, 101, 110, 133, 137, 141, 145, 294, 298, 302, 308, 312, 318, 322, 331, 338, 342, 346, <i>368</i>
<code>\cs_set:Npn</code> . . . . .	24, 27, <i>76</i>
<code>\cs_set_eq:NN</code> . . . . .	<i>360</i>
<code>\cs_set_protected:Nn</code> . . . . .	263, 268, 273, 279, 286, 354, 358, 363, 372, 383, 387, <i>408</i>
<code>\cs_split_function:N</code> . . . . .	<i>5</i>
<b>D</b>	
<code>\disambignewcmd</code> . . . . .	<i>13</i>
<b>E</b>	
erw commands:	
<code>\erw_accum</code> . . . . .	<i>13</i>
<code>\erw_append_arg:nn</code> . . . . .	<i>13</i>
<code>\erw_compose</code> . . . . .	<i>13</i>
<code>\erw_cs_apply:Nn</code> . . . . .	3, 7, 11, 43, <i>270</i>
<code>\erw_cs_apply:Nnn</code> . . . . .	3, 12, <i>275</i>
<code>\erw_cs_apply:Nnnn</code> . . . . .	3, 16, <i>281</i>
<code>\erw_cs_apply:Nnnnn</code> . . . . .	3, 20, <i>288</i>
<code>\erw_cs_gset_eq:NN</code> . . . . .	<i>340</i>
<code>\erw_cs_gset_inline:Nn</code> . . . . .	30, 34, <i>344</i>
<code>\erw_cs_identity:n</code> . . . . .	4, <i>13</i> , <i>24</i>
<code>\erw_cs_set_eq:NN</code> . . . . .	<i>13</i>
<code>\erw_cs_set_inline:Nn</code> . . . . .	4, 25, 29, 53, 112, <i>365</i>
<code>\erw_csint:nn</code> . . . . .	4, <i>41</i>
<code>\erw_csint_name:n</code> . . . . .	4, 40, 45, 48, <i>66</i>
<code>\erw_csint_names:nnn</code> . . . . .	4, <i>46</i>
<code>\erw_csint_names_braced:</code> . . . . .	4, 67, <i>328</i>
<code>\erw_csint_names_braced:n</code> . . . . .	4, 63, <i>66</i>
<code>\erw_csint_names_braced:nnn</code> . . . . .	4, 61, <i>69</i>
<code>\erw_csint_new:n</code> . . . . .	4, 50, <i>325</i>
<code>\erw_csint_reset:</code> . . . . .	4, 71, <i>324</i>
<code>\erw_fold_apply_par:n</code> . . . . .	<i>13</i>
<code>\erw_fold_set_par:n</code> . . . . .	<i>13</i>
<code>\erw_identity:n</code> . . . . .	<i>13</i>
<code>\erw_int_range</code> . . . . .	<i>13</i>
<code>\erw_int_range:n</code> . . . . .	4, <i>101</i>
<code>\erw_int_range:nn</code> . . . . .	4, <i>97</i>
<code>\erw_is_matrix</code> . . . . .	<i>13</i>
<code>\erw_is_matrix:n</code> . . . . .	<i>13</i>
<code>\erw_items_to</code> . . . . .	<i>13</i>
<code>\erw_jobnametimestamp:nn</code> . . . . .	<i>13</i>
<code>\erw_last_item</code> . . . . .	<i>13</i>
<code>\erw_merge</code> . . . . .	<i>13</i>
<code>\erw_oper_fold_seq:NN</code> . . . . .	<i>13</i>
<code>\erw_oper_gset_function:N</code> . . . . .	<i>13</i>
<code>\erw_option:n</code> . . . . .	6, <i>418</i>
<code>\erw_repeat</code> . . . . .	<i>13</i>
<code>\erw_seq_compose:nN</code> . . . . .	4, <i>133</i>
<code>\erw_seq_compose_c:nN</code> . . . . .	4, <i>137</i>
<code>\erw_seq_compose_vers:nN</code> . . . . .	4, <i>141</i> , <i>143</i>
<code>\erw_seq_fold:NN</code> . . . . .	4, <i>13</i> , <i>135</i> , <i>139</i> , <i>145</i> , <i>151</i>
<code>\erw_set_map</code> . . . . .	<i>13</i>
<code>\erw_set_map_inline</code> . . . . .	<i>13</i>
<code>\erw_split</code> . . . . .	<i>13</i>
<code>\erw_sys_jobnametimestamp:</code> . . . . .	5, <i>242</i>
<code>\erw_sys_jobnametimestamp:nn</code> . . . . .	5, <i>241</i>
<code>\erw_sys_timestamp:</code> . . . . .	5, <i>216</i> , <i>251</i>
<code>\erw_sys_timestamp:nn</code> . . . . .	5, <i>210</i> , <i>247</i>
<code>\erw_sys_timestamp_delimiter:</code> . . . . .	5, <i>13</i> , <i>243</i>

\erw_tl_append_item:nn .	5, 13, 88, 294
\erw_tl_compose:nN . . . . .	5, 298, 305
\erw_tl_compose:nn . . . . .	5, 302
\erw_tl_compose_c:nN . . . . .	5, 308, 315
\erw_tl_compose_c:nn . . . . .	5, 312, 327
\erw_tl_compose_vers:nN . . . . .	5, 318, 320
\erw_tl_compose_vers:nn . . . . .	5, 322
\erw_tl_fold:NN . . . . .	5, 148, 300, 310, 331, 337
\erw_tl_gset_function:N . . . . .	5, 13, 338
\erw_tl_gset_function:n . . . . .	5, 6, 342
\erw_tl_join:nn . . . . .	5, 13, 35, 200, 208, 214
\erw_tl_join:nnn . . . . .	5, 36, 184
\erw_tl_join:nnnn . . . . .	5, 37
\erw_tl_join:nnnnn . . . . .	5, 38
\erw_tl_last_item:n . . . . .	5, 346
\erw_tl_last_time:n . . . . .	5
\erw_tl_map:n . . . . .	6, 116, 354, 361, 366
\erw_tl_map:Nn . . . . .	6, 358
\erw_tl_map_inline:nn . . . . .	6, 363
\erw_tl_map_thread:Nn . . . . .	6, 13, 408
\erw_tl_map_thread_at:Nnn . . . . .	6, 387, 415
\erw_tl_repeat:nn . . . . .	6, 368
\erw_tl_split:nn . . . . .	6, 383
\erw_tl_split:nnn . . . . .	6, 372, 385
erw internal commands:	
\__erw_cs_name:N . . . . .	3
\__erw_csint_ext_tl . . . . .	74
\g__erw_csint_int . . . . .	39, 40, 52, 69, 73
\g__erw_csint_name_tl . . . . .	40, 53
\__erw_int_range:nnn . . . . .	76, 86, 99, 103
\__erw_map:nn . . . . .	263, 356
\__erw_map_thread_at:Nnn . . . . .	268, 394
\__erw_map_thread_at:Nnnn . . . . .	268, 395
\__erw_map_thread_at:Nnnnn . . . . .	268, 396
\__erw_map_thread_at:Nnnnnn . . . . .	268, 397
\__erw_oper_compose:NnN . . . . .	110, 135, 139, 300, 310
\g__erw_oper_fold_apply_par_tl . . . . .	127, 335
\g__erw_oper_fold_set_par_tl . . . . .	123, 333
\g__erw_seq_fold_item_tl . . . . .	132, 147, 148, 149
\__erw_sys_date:N . . . . .	154
\__erw_sys_date_dec: . . . . .	154, 196
\__erw_sys_date_hex: . . . . .	154, 197
\__erw_sys_datetime_base:n . . . . .	174, 221
\__erw_sys_datetime_dec: . . . . .	196
\__erw_sys_datetime_dec:n . . . . .	174
\__erw_sys_datetime_hex: . . . . .	197
\__erw_sys_datetime_hex:n . . . . .	174
\__erw_sys_datetime_join:nn . . . . .	174
\__erw_sys_datetime_period:n . . . . .	174, 221
\__erw_sys_jobnametimestamp: . . . . .	204, 212, 242
\__erw_sys_jobnametimestamp:n . . . . .	204
\__erw_sys_jobnametimestamp:nn . . . . .	206, 241
\__erw_sys_jobnametimestamp- prefix: . . . . .	198
\__erw_sys_set_delim:nn . . . . .	223, 233
\__erw_sys_time_dec: . . . . .	165, 196
\__erw_sys_time_hex . . . . .	165
\__erw_sys_time_hex: . . . . .	173, 197
\__erw_sys_timestamp:nn . . . . .	218, 249, 253
\g__erw_sys_timestamp_delim_str . . . . .	184, 202, 226, 245
\g__erw_tl_compose_tl . . . . .	255, 304, 305, 306, 314, 315, 316
\__erw_tl_function:n . . . . .	112, 256, 266, 340, 344, 360, 365
exp commands:	
\exp_args:Nf . . . . .	116, 271, 276, 277, 282, 283, 284, 289, 290, 291, 292, 389, 412
\exp_args:No . . . . .	220
\exp_args:Nof . . . . .	348
\exp_args:Nx . . . . .	88
\exp_last_unbraced:Nf . . . . .	5
\exp_last_unbraced:No . . . . .	232
\exp_last_unbraced:Nx . . . . .	326
\ExplSyntaxOff . . . . .	422
\ExplSyntaxOn . . . . .	2
I	
int commands:	
\int_case:nnTF . . . . .	176, 389
\int_compare:nNnTF . . . . .	78
\int_eval:n . . . . .	80, 90, 93, 156, 167
\int_incr:N . . . . .	52
\int_new:N . . . . .	39
\int_step_function:nnnN . . . . .	48, 63
\int_step_inline:nn . . . . .	105, 410
\int_step_inline:nnnn . . . . .	370
\int_to_alph:n . . . . .	43, 45
\int_to_hex:n . . . . .	163, 164, 173
\int_zero:N . . . . .	73
K	
keys commands:	
\keys_define . . . . .	13
\keys_define:nn . . . . .	121, 228
\keys_set:nn . . . . .	420
M	
map commands:	
\map_thread . . . . .	13

msg commands:		\c_sys_jobname_str . . . . .	201
\msg_error:nnn . . . . .		\c_sys_minute_int . . . . .	170
. . . . .	143, 182, 194, 258, 320, 403	\c_sys_month_int . . . . .	159
\msg_new:nnn . . . . .	107, 108, 109, 152, 153	\c_sys_year_int . . . . .	158
<b>N</b>		<b>T</b>	
\newnumbrdcs . . . . .	13	tl commands:	
\numbrdcs . . . . .	13	\c_empty_tl . . . . .	181, 193
\numbrdcsnew . . . . .	13	\tl_count:n . . . . .	351, 391, 412
<b>P</b>		\tl_head:n . . . . .	374, 412
\ProcessPackageKeysOption . . . . .	13	\tl_item:nn . . . . .	271, 276, 277,
<b>Q</b>		282, 283, 284, 289, 290, 291, 292, 348	
quark commands:		\tl_map_function:nN . . . . .	325
\quark_if_recursion_tail_stop:n	265	\tl_map_inline:nn . . . . .	375
\q_recursion_stop . . . . .	356	\tl_new:N . . . . .	132, 255
\q_recursion_tail . . . . .	356	\tl_range:nnn . . . . .	13
<b>S</b>		\tl_range_braced:nnn . . . . .	64
seq commands:		\tl_reverse:n . . . . .	118
\seq_get_right:NN . . . . .	147	\tl_set:Nn . . . . .	40, 74, 304, 314
\seq_put_right:Nn . . . . .	149	\tl_tail:n . . . . .	377
str commands:		token commands:	
\str_case:nnTF . . . . .	187	\token_if_cs:NTF . . . . .	55
\subsection . . . . .	239	<b>U</b>	
sys commands:		use commands:	
\c_sys_day_int . . . . .	160	\use:N . . . . .	221, 225, 245, 333, 335, 375
\c_sys_hour_int . . . . .	169	\use_i:nnn . . . . .	5
		\usepackage . . . . .	3



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

```

### 6.2 frontend

```

121 \keys_define:nn{__erw}

```

```

122 {
123   oper/fold_set_par.tl_gset:N = \g__erw_oper_fold_set_par_tl,
124   oper/fold_set_par.value_required:n = true,
125   oper/fold_set_par.default:n = {Nf},
126   oper/fold_set_par.initial:n = {Nf},
127   oper/fold_apply_par.tl_gset:N = \g__erw_oper_fold_apply_par_tl,
128   oper/fold_apply_par.value_required:n = true,
129   oper/fold_apply_par.default:n = {Nf},
130   oper/fold_apply_par.initial:n = {Nf}
131 }

```

## 7 seq

### 7.1 backend

```

132 \tl_new:N \g__erw_seq_fold_item_tl

```

### 7.2 frontend

```

133 \cs_set:Nn \erw_seq_compose:nN
134 {
135   \__erw_oper_compose:NnN \erw_seq_fold:NN {#1} #2
136 }
137 \cs_set:Nn \erw_seq_compose_c:nN
138 {
139   \__erw_oper_compose:NnN \erw_seq_fold:cN {#1} #2
140 }
141 \cs_set:Nn \erw_seq_compose_vers:nN
142 {
143   \msg_error:nnn{__erw}{notdecl}{\erw_seq_compose_vers:nN}
144 }
145 \cs_set:Nn \erw_seq_fold:NN
146 {
147   \seq_get_right:NN #2 \g__erw_seq_fold_item_tl
148   \erw_tl_fold:NN #1 \g__erw_seq_fold_item_tl
149   \seq_put_right:No #2 {\g__erw_seq_fold_item_tl}
150 }
151 \cs_generate_variant:Nn \erw_seq_fold:NN {cN}

```

## 8 sys

### 8.1 backend

```

152 \msg_new:nnn{__erw}{timestamp / base}{Calling~#1,~arg~must~be~'dec|hex'}
153 \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:
154 \cs_new:Nn \__erw_sys_date_dec:
155 {
156   \int_eval:n
157   {
158     \c_sys_year_int * 10000
159     +\c_sys_month_int * 100
160     +\c_sys_day_int * 1
161   }

```

```

162 }
163 \cs_new:Nn \__erw_sys_date:N{\int_to_hex:n{\__erw_sys_date_dec:}}
164 \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

```

```

165 \cs_new:Nn \__erw_sys_time_dec:
166 {
167   \int_eval:n
168   {
169     \c_sys_hour_int * 100
170     +\c_sys_minute_int * 1
171   }
172 }
173 \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

```

```

174 \cs_new:Nn \__erw_sys_datetime_base:n
175 {
176   \int_case:nnTF{#1}
177   {
178     {10}{dec}
179     {16}{hex}
180   }
181   {\c_empty_tl}
182   {\msg_error:nnn{\__erw}{timestamp / base}{\__erw_sys_datetime_base:n{#1}}}
183 }
184 \cs_new:Nn \__erw_sys_datetime_join:nn{\erw_tl_join:nnn{#1}{\g__erw_sys_timestamp_delim_str}{#2}}
185 \cs_new:Nn \__erw_sys_datetime_period:n
186 {
187   \str_case:nnTF{#1}
188   {
189     {date}{date}
190     {time}{time}
191     {datetime}{datetime}
192   }
193   {\c_empty_tl}
194   {\msg_error:nnn{\__erw}{timestamp / period}{\__erw_sys_datetime_period:n{#1}}}
195 }
196 \cs_new:Nn \__erw_sys_datetime_dec: {\__erw_sys_datetime_join:nn{\__erw_sys_date_dec:}{\__erw_sys_time_dec:}}
197 \cs_new:Nn \__erw_sys_datetime_hex: {\__erw_sys_datetime_join:nn{\__erw_sys_date_hex:}{\__erw_sys_time_hex:}}

```

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

```

\__erw_sys_jobnametimestamp_prefix:

```

```

198 \cs_new:Nn \__erw_sys_jobnametimestamp_prefix:
199 {
200   \erw_tl_join:nn
201   {\c_sys_jobname_str}
202   {\g__erw_sys_timestamp_delim_str}
203 }
204 % \begin{macro}{\__erw_sys_jobnametimestamp:n, \__erw_sys_jobnametimestamp:}

```

```

205 % \begin{macrocode}
206 \cs_new:Nn\__erw_sys_jobnametimestamp:nn
207 {
208   \erw_tl_join:nn
209   {\__erw_sys_jobnametimestamp_prefix:}
210   {\erw_sys_timestamp:nn{#1}{#2}}
211 }
212 \cs_new:Nn\__erw_sys_jobnametimestamp:
213 {
214   \erw_tl_join:nn
215   {\__erw_sys_jobnametimestamp_prefix:}
216   {\erw_sys_timestamp:}
217 }

```

(End definition for \\_\_erw\_sys\_jobnametimestamp\_prefix:.)

\\_\_erw\_sys\_timestamp:nn

```

218 \cs_new:Nn\__erw_sys_timestamp:nn
219 {
220   \exp_args:No
221   \use:c{\__erw_sys_\__erw_sys_datetime_period:n{#1}_\__erw_sys_datetime_base:n{#2}:}
222 }
223 \cs_new_protected:Nn \__erw_sys_set_delim:nn
224 {
225   \use:c{tl_gset:N#1}
226   \g__erw_sys_timestamp_delim_str{#2}
227 }

```

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

```

228 \keys_define:nn{\__erw}
229 {
230   sys / timestamp_delim .code:n =
231   {
232     \exp_last_unbraced:No
233     \__erw_sys_set_delim:nn{n}{#1}
234   },
235   sys / timestamp_delim .value_required:n = true,
236   sys / timestamp_delim .default:n = {-},
237   sys / timestamp_delim .initial:n = {-}
238 }
239 % \subsection{frontend}
240 % \begin{macrocode}
241 \cs_new:Nn\erw_sys_jobnametimestamp:nn{\__erw_sys_jobnametimestamp:nn{#1}{#2}}
242 \cs_new:Nn\erw_sys_jobnametimestamp:{\__erw_sys_jobnametimestamp:}
243 \cs_new:Nn\erw_sys_timestamp_delimiter:
244 {
245   \use:N \g__erw_sys_timestamp_delim_str
246 }
247 \cs_new:Nn\erw_sys_timestamp:nn
248 {
249   \__erw_sys_timestamp:nn{#1}{#2}
250 }
251 \cs_new:Nn\erw_sys_timestamp:
252 {

```

```

253 \__erw_sys_timestamp:nn{datetime}{16}
254 }

```

## 9 tl

### 9.1 backend

```

255 \tl_new:N \g__erw_tl_compose_tl

\__erw_tl_function:n

256 \cs_new_protected:Nn \__erw_tl_function:n
257 {
258   \msg_error:nnn
259   {erw}
260   {notset}
261   {\__erw_tl_function:n}
262 }

(End definition for \__erw_tl_function:n.)

\__erw_map:nn

263 \cs_set_protected:Nn \__erw_map:nn
264 {
265   \quark_if_recursion_tail_stop:n{#1}
266   \__erw_tl_function:n{#1} \__erw_map:nn{#2}
267 }

(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
268 \cs_set_protected:Nn \__erw_map_thread_at:Nnn
269 {
270   \erw_cs_apply:Nn #1
271   {\exp_args:Nf\tl_item:nn {#3} {#2} }
272 }
273 \cs_set_protected:Nn \__erw_map_thread_at:Nnnn
274 {
275   \erw_cs_apply:Nnn #1
276   {\exp_args:Nf\tl_item:nn {#3} {#2} }
277   {\exp_args:Nf\tl_item:nn {#4} {#2} }
278 }
279 \cs_set_protected:Nn \__erw_map_thread_at:Nnnnn
280 {
281   \erw_cs_apply:Nnnn #1
282   {\exp_args:Nf\tl_item:nn {#3} {#2} }
283   {\exp_args:Nf\tl_item:nn {#4} {#2} }
284   {\exp_args:Nf\tl_item:nn {#5} {#2} }
285 }
286 \cs_set_protected:Nn \__erw_map_thread_at:Nnnnnn
287 {
288   \erw_cs_apply:Nnnnn #1
289   {\exp_args:Nf\tl_item:nn {#3} {#2} }
290   {\exp_args:Nf\tl_item:nn {#4} {#2} }
291   {\exp_args:Nf\tl_item:nn {#5} {#2} }

```

```

292   {\exp_args:Nf\tl_item:nn {#6} {#2} }
293 }

```

(End definition for `\_erw_map_thread_at:Nnn` and others.)

## 9.2 frontend

```

294 \cs_set:Nn \erw_tl_append_item:nn
295 {
296   {#1{#2}}
297 }
298 \cs_set:Nn \erw_tl_compose:nN
299 {
300   \__erw_oper_compose:NnN \erw_tl_fold:NN {#1} #2
301 }
302 \cs_set:Nn \erw_tl_compose:nn
303 {
304   \tl_set:Nn \g__erw_tl_compose_tl {#2}
305   \erw_tl_compose:nN{#1}\g__erw_tl_compose_tl
306   \g__erw_tl_compose_tl
307 }
308 \cs_set:Nn \erw_tl_compose_c:nN
309 {
310   \__erw_oper_compose:NnN \erw_tl_fold:cN {#1} #2
311 }
312 \cs_set:Nn \erw_tl_compose_c:nn
313 {
314   \tl_set:Nn \g__erw_tl_compose_tl {#2}
315   \erw_tl_compose_c:nN{#1}\g__erw_tl_compose_tl
316   \g__erw_tl_compose_tl
317 }
318 \cs_set:Nn \erw_tl_compose_vers:nN
319 {
320   \msg_error:nnn{__erw}{notdecl}{\erw_tl_compose_vers:nN}
321 }
322 \cs_set:Nn \erw_tl_compose_vers:nn
323 {
324   \erw_csint_reset:{}
325   \tl_map_function:nN{#1}\erw_csint_new:n
326   \exp_last_unbraced:Nx
327   \erw_tl_compose_c:nn
328   {{\erw_csint_names_braced:{}}}
329   {#2}
330 }
331 \cs_set:Nn \erw_tl_fold:NN
332 {
333   \use:c{tl_set:\g__erw_oper_fold_set_par_tl}
334   #2
335   {\use:c{erw_cs_apply:\g__erw_oper_fold_apply_par_tl}{#1}{#2}}
336 }
337 \cs_generate_variant:Nn \erw_tl_fold:NN {cN}
338 \cs_set:Nn \erw_tl_gset_function:N
339 {
340   \erw_cs_gset_eq:NN \__erw_tl_function:n #1

```



```

341 }
342 \cs_set:Nn \erw_tl_gset_function:n
343 {
344   \erw_cs_gset_inline:Nn \__erw_tl_function:n {#1}
345 }
346 \cs_set:Nn \erw_tl_last_item:n
347 {
348   \exp_args:Nof \tl_item:nn
349   {#1}
350   {
351     \tl_count:n{#1}
352   }
353 }
354 \cs_set_protected:Nn \erw_tl_map:n
355 {
356   \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
357 }
358 \cs_set_protected:Nn \erw_tl_map:Nn
359 {
360   \cs_set_eq:NN \__erw_tl_function:n #1
361   \erw_tl_map:n{#2}
362 }
363 \cs_set_protected:Nn \erw_tl_map_inline:nn
364 {
365   \erw_cs_set_inline:Nn \__erw_tl_function:n {#1}
366   \erw_tl_map:n{#2}
367 }
368 \cs_set:Nn \erw_tl_repeat:nn
369 {
370   \int_step_inline:nnnn{1}{1}{#1}{#2}
371 }
372 \cs_set_protected:Nn \erw_tl_split:nnn
373 {
374   \tl_head:n{#1}
375   \use:c{exp_args:#3} \tl_map_inline:nn
376   {
377     \tl_tail:n
378     {
379       #1
380     }
381     }{#2##1}
382 }
383 \cs_set_protected:Nn \erw_tl_split:nn
384 {
385   \erw_tl_split:nnn{#1}{#2}{Nf}
386 }
387 \cs_set_protected:Nn \erw_tl_map_thread_at:Nnn
388 {
389   \exp_args:Nf\int_case:nnTF
390   {
391     \tl_count:n{#3}
392   }
393   {
394     {1}{ \__erw_map_thread_at:Nnn #1{#2}#3 }

```

```

395     {2}{ \__erw_map_thread_at:Nnnn #1{#2}#3 }
396     {3}{ \__erw_map_thread_at:Nnnnn #1{#2}#3 }
397     {4}{ \__erw_map_thread_at:Nnnnnn #1{#2}#3 }
398   }
399   {
400     % Do nothing
401   }
402   {
403     \msg_error:nnn{__erw}
404     {generic}
405     {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
406   }
407 }
408 \cs_set_protected:Nn \erw_tl_map_thread:Nn
409 {
410   \int_step_inline:nn
411   {
412     \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
413   }
414   {
415     \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
416   }
417 }

```

## 10 option

```

418 \cs_new_protected:Nn\erw_option:n
419 {
420   \keys_set:nn{__erw}{#1}
421 }

```

## 11 Closing

```

422 \ExplSyntaxOff

```