

The `erw-l3` package ^{*}

Erwann Rogard[†]

Released 2020/05/18

Abstract

Utilities like `expl3[1]`.

Résumé

Utilitaires de type `expl3[1]`.

Contents

I	Usage	4
1	Loading the package	4
2	basics	4
3	csint	4
4	int	5
5	option	5
6	prop	5
7	seq	5
8	sys	6
9	tl	6
10	<code>\erw_tl_gset_function:n</code>	6
II	Listing	8
1	constants	8
1.	8

^{*}This file describes version v2.2, last revised 2020/05/18.

[†]firstname dot lastname AusTria gmail dot com

2	basics	8
	2.	8
3	csint	8
	3.	8
4	int	9
	4.	9
5	prop	9
	5.	9
	6.	9
6	seq	9
	7.	9
	8.	10
	9.	10
	10.	10
7	sys	11
	11.	11
	12.	11
8	tl	12
	13.	12
	14.	12
	16.	12
	17.	13
	18.	13
	19.	13
	20.	13
	21.	13
	22.	14
III	Other	15
1	Acknowledgment	15
2	Install	15
3	Support	15
	3.1 Platform	15
	3.2 Engine	15
	3.3 Results	15
4	References	15
	Change History	16

Index	17
IV Implementation	21
1 Opening	21
2 basics	21
2.1 backend	21
2.2 frontend	21
3 clist	22
3.1 backend	22
3.2 frontend	22
4 csint	22
4.1 backend	22
4.2 frontend	22
5 int	22
5.1 backend	22
5.2 frontend	23
6 keyval	23
7 msg	23
7.1 backend	23
8 prop	23
8.1 backend	23
8.2 frontend	24
9 oper	24
9.1 backend	24
9.2 frontend	24
10 seq	25
10.1 backend	25
10.2 frontend	25
11 sys	26
11.1 backend	26
12 tl	28
12.1 backend	28
12.2 frontend	30
13 option	32

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^AT_EX engine. See [Part III, section 3](#).
2. Goes in the *preamble*

2 basics

<code>\erw_cs_apply:Nn</code>	<code>\erw_cs_apply:Nn {<cs>}{<token list₁</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{<arg>}</code>
---------------------------------	--

<code>\erw_cs_set_inline:Nn</code>	<code>\erw_cs_set_inline:Nn{<cs>}{<code>}</code>
<code>\erw_cs_set_inline:cn</code>	

3 csint

<code>\erw_csint:nn</code>	<code>\erw_csint:nn{<integer>}{<arg>}</code>
----------------------------	--

<code>\erw_csint_name:n</code>	<code>\erw_csint_name:n{<integer>}</code>
--------------------------------	---

<code>\erw_csint_names:nnn</code>	<code>\erw_csint_names:nnn{<integer>}{<integer>}{<integer>}</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{<integer>}</code>
-------------------------------	--

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

4 int

<code>\erw_int_range:n</code>	<code>\erw_int_range:n{<integer>}</code>
<code>\erw_int_range:nn</code>	

5 option

<code>\erw_option:n</code>	<code>\erw_option:n{<keyval list>}</code>
----------------------------	---

oper / fold_set_par
oper / fold_apply_par
sys / timestamp_delim

6 prop

All functions that modify a $\langle prop \rangle$ check it exists, if not make sure it does.

<code>\erw_prop_put:NN</code>	<code>\erw_prop_put:NN<prop₁><prop₂></code>
-------------------------------	---

<code>\erw_prop_put:Nnn</code>	<code>\erw_prop_put:Nnn<prop>{<key>}{<val>}</code>
--------------------------------	--

<code>\erw_prop_to_clist:Nn</code>	<code>\erw_prop_to_clist:Nn<prop>{<key₁>,...}</code>
------------------------------------	---

7 seq

All functions that modify a $\langle seq \rangle$ check it exists, if not make sure it does.

<code>\erw_seq_compose:nN</code>	<code>\erw_seq_compose:nN{<cs₁>...}<seq></code>
----------------------------------	--

<code>\erw_seq_compose_c:nN</code>	<code>\erw_seq_compose_c:nN{<cs name₁>...}<seq></code>
------------------------------------	---

<code>\erw_seq_compose_vers:nN</code>	<code>\erw_seq_compose:nN{<cs or code₁>...}<seq></code>
---------------------------------------	--

<code>\erw_seq_from_clist:Nn</code>	<code>\erw_seq_from_clist:Nn<seq>{<clist>}</code>
<code>\erw_seq_from_clist:cn</code>	

<code>\erw_seq_from_prop:Nn</code>	<code>\erw_seq_from_prop:NNn⟨seq⟩⟨prop⟩{⟨keyval list⟩}</code>
------------------------------------	---

<code>\erw_seq_put_right:Nn</code>	<code>\erw_seq_put_right:Nn⟨seq⟩{⟨token list⟩}</code>
------------------------------------	---

<code>\erw_seq_use:Nn</code>	<code>\erw_seq_use:Nn⟨seq⟩{⟨items⟩}</code>
------------------------------	--

8 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>	Semantics Timestamp in base 10 or 16

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

9 tl

All functions that modify a $\langle token list \rangle$ check it exists, if not make sure it does.

<code>\erw_tl_append_item:nn</code>	<code>\erw_tl_append_item:nn{⟨arg list⟩}{⟨arg⟩}</code>
-------------------------------------	--

<code>\erw_tl_compose:nN</code>	<code>\erw_tl_compose:nn{cs_1}...{⟨token list⟩}</code>
<code>\erw_tl_compose:nn</code>	

<code>\erw_tl_compose_c:nN</code>	<code>\erw_tl_compose_c:nn{cs name_1}...{⟨token list⟩}</code>
<code>\erw_tl_compose_c:nn</code>	

<code>\erw_tl_compose_vers:nN</code>	<code>\erw_tl_compose_vers:nn{cs or code_1}...{⟨token list⟩}</code>
<code>\erw_tl_compose_vers:nn</code>	

<code>\erw_tl_fold:NN</code>	<code>\erw_tl_fold:NN⟨cs⟩⟨tl var⟩</code>
<code>\erw_tl_fold:cN</code>	

<code>\erw_tl_gset_function:N</code>	<code>\erw_tl_gset_function:n{⟨code⟩}</code>
<code>\erw_tl_gset_function:n</code>	

<hr/>	
<code>\erw_tl_join:nn</code>	<code>\erw_tl_join:nn{<token list₁>}{<token list₂>}</code>
<code>\erw_tl_join:nnn</code>	
<code>\erw_tl_join:nnnn</code>	
<code>\erw_tl_join:nnnnn</code>	
<hr/>	
<code>\erw_tl_last_item:n</code>	<code>\erw_tl_last_item:n{<token list>}</code>
<hr/>	
<code>\erw_tl_map:n</code>	<code>\erw_tl_map:n{<items>}</code>
<code>\erw_tl_map:Nn</code>	Semantics Maps over <code><items></code> using the internal function set by <code>\erw_tl_gset_ function:n</code>
<hr/>	
<code>\erw_tl_map_inline:nn</code>	<code>\erw_tl_map_inline:nn{<code>}{<items>}</code>
<hr/>	
<code>\erw_tl_map_thread:Nn</code>	<code>\erw_tl_math_thread:Nn<cs>{<items>}</code>
<hr/>	
<code>\erw_tl_map_thread_at:Nnn</code>	<code>\erw_tl_math_thread_at:Nnn{<integer>}{<token list>}</code>
<hr/>	
<code>\erw_tl_repeat:nn</code>	<code>\erw_tl_repeat:nn{<integer>}{<token list>}</code>
<hr/>	
<code>\erw_tl_split:nnn</code>	<code>\erw_tl_split:nn{<items>}{<delimiter>}</code>
<code>\erw_tl_split:nn</code>	
<hr/>	
<code>\erw_tl_separators:n</code>	<code>\erw_tl_separators:n{<items>}</code>
<hr/>	

Part II

Listing

1 constants

Listing 1.

```
\ExplSyntaxOn
\seq_const_from_clist:Nn \foo_seq{ A, B, C }
\prop_const_from_keyval:Nn \foo_prop{ A = a, B = b, C = c }
\ExplSyntaxOff
```

2 basics

Listing 2.

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

f(X)

3 csint

Listing 3.

```
\ExplSyntaxOn
\cs_set:Nn \__foo:n{f(#1)}
\cs_set: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)]}

4 int

Listing 4.

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

2345
12345

5 prop

Listing 5.

```
\ExplSyntaxOn
\erw_prop_put:Nnn \baz_prop { D } { d }
\erw_prop_put:NN \baz_prop \foo_prop
\prop_item:Nn \baz_prop{A}
,\prop_item:Nn \baz_prop{B}
,\prop_item:Nn \baz_prop{C}
,\prop_item:Nn \baz_prop{D}
\ExplSyntaxOff
```

a,b,c,d

Listing 6.

```
\ExplSyntaxOn
\erw_prop_to_clist:Nn \foo_prop{ A, B, C }
\ExplSyntaxOff
```

a,b,c

6 seq

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_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 8.

```

\ExplSyntaxOn
\cs_set:Nn \__foo:n {f(#1)}
\cs_set:Nn \__bar:n {g[#1]}
\cs_set:Nn \__baz:n {h\{#1\}}
\erw_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)]}

Listing 9.

```

\ExplSyntaxOn
\erw_seq_from_prop:NNn \bar_seq\foo_prop{ A, B, C }
\seq_use:Nn\bar_seq{,}
\ExplSyntaxOff

```

a,b,c

Listing 10.

```

\ExplSyntaxOn
\erw_seq_use:Nn \foo_seq{{~and~}}\\
\erw_seq_use:Nn \foo_seq{{, \ }{and~}}\\
\erw_seq_use:Nn \foo_seq{{~and~}{, \ }{, \ and~}}
\ExplSyntaxOff

```

A and B and C
A, B, and C
A, B, and C

7 sys

Listing 11.

```
\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
```

20200519-2059
20200519-2059
1343c47%80b
1343c47%80b
erw-l3%1343c47%80b

Listing 12.

```
\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:202005192059
D:20200519205947-04'00'
Hello, world!

8 tl

Listing 13.

```
\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 14.

```
\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 15.

```
\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 16.

```
\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 17.

```

\ExplSyntaxOn
%^^A \erw_seq_use:Nn \foo_seq{{~and~}}
\seq_use:Nn\foo_seq{,}
%^^A \erw_tl_separators:n{{~and~}}\
%^^A \erw_tl_separators:n{{\ ,}{~and~}}\
%^^A \erw_tl_separators:n{{~and~}}{\ ,}{~and~}}
\ExplSyntaxOff

```

A,B,C

Listing 18.

```

\ExplSyntaxOn
\erw_tl_repeat:nn{3}{x}
\ExplSyntaxOff

```

xxx

Listing 19.

```

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

```

a==b==c

Listing 20.

```

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

```

(a)(b)(c)

Listing 21.

```

\ExplSyntaxOn
\cs_set:Nn \__foo:n {(#1)}
\erw_tl_map_thread:Nn \__foo:n
{
  {{a}{b}{c}{d}{e}{f}}
}
\cs_set:Nn \__foo:nn {(#1+#2)}
\erw_tl_map_thread:Nn \__foo:nn
{
  {{a}{b}{c}{d}{e}{f}}
  {{A}{B}{C}{D}{E}{F}}
}
\cs_set:Nn \__foo:nnn {(#1+#2+#3)}
\erw_tl_map_thread:Nn \__foo:nnn
{
  {{a}{b}{c}{d}{e}{f}}
  {{A}{B}{C}{D}{E}{F}}
  {{k}{l}{m}{n}{o}{p}}
}
\cs_set:Nn \__foo:nnnn {(#1+#2+#3+#4)}
\erw_tl_map_thread:Nn \__foo:nnnn
{
  {{a}{b}{c}{d}{e}{f}}
  {{A}{B}{C}{D}{E}{F}}
  {{k}{l}{m}{n}{o}{p}}
  {{K}{L}{M}{N}{O}{P}}
}
\ExplSyntaxOff

```

(a)(b)(c)(d)(e)(f)
(a+A)(b+B)(c+C)(d+D)(e+E)(f+F)
(a+A+k)(b+B+l)(c+C+m)(d+D+n)(e+E+o)(f+F+p)
(a+A+k+K)(b+B+l+L)(c+C+m+M)(d+D+n+N)(e+E+o+O)(f+F+p+P)

Listing 22.

```

\ExplSyntaxOn
\cs_set:Nn \__foo:nn {(#1+#2)}
\erw_tl_map_thread_at:Nnn \__foo:nn{2}
{
  {{a}{b}{c}{d}{e}{f}}
  {{A}{B}{C}{D}{E}{F}}
}
\ExplSyntaxOff

```

(b+B)

Part III

Other

1 Acknowledgment

This work has benefited from Q&A's from the L^AT_EXcommunity[2]

2 Install

- 1) Compile `erw-13.dtx` (under Unix, `$tex timestamp.dtx`)
- 2) Put the generated `erw-13.sty` in the search path of the L^AT_EXengine

3 Support

This package is available from <https://www.ctan.org/pkg/erw-13> and <https://github.com/rogard/erw-13>.

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) erw-13 v2.0 compiles satisfactorily on platform *i)* and engines *b)*, *c)*, and *d)*

References

- [1] The L^AT_EX3 Project Team *The L^AT_EX3 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

v1.1	General: <code>\numbrdcsnew</code> changed to <code>\newnumbrdcs</code> and made 'disambiguable' 15	v1.6	General: Fix: critical bug preventing <code>erw-l3</code> from working without explicit inclusion of <code>expl3</code> 15
	<code>disambig/backend</code> : changes to the key, added <code>\ProcessPackageKeysOption</code> ; . . . 15	v1.7	General: Add: <code>option</code> 15
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ; 15		Add: <code>sys</code> 15
v1.2	General: 15		Move: <code>\erw_fold_apply_par:n</code> . . . 15
	<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) 15		Move: <code>\erw_fold_set_par:n</code> 15
	<code>disambig</code> : pushed the code inside <code>\keys_define:\disambignewcmd</code> no longer takes a token name as arg, rather a token. 15		Rearrange: structure of implementation, e.g. section 9 . . . 15
	Add: <code>\erw_items_to</code> 15		Remove: document level functions, <code>\numbrdcsnew</code> , <code>\numbrdcs</code> 15
	Add: <code>\erw_last_item</code> 15		Replace: listing's implem with that of <code>tocloft</code> 15
	Add: <code>\erw_repeat</code> 15		Replace: vers. numb. from 3 to 2 digits 15
	Add: <code>\erw_split</code> 15	v1.8	General: Add: function for all frontend functions. 15
	Add: <code>\map_thread</code> 15		Remove: <code>\erw_cs_set_eq:NN</code> and variants 15
	Front end cmds no longer generated with module <code>disambig</code> ; Option of the same name deleted; 15		Remove: <code>\erw_is_matrix:n</code> (predicate must be expandable) . . 15
	Re-arrange: the doc to clearly separate frontend from backend . . 15		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> 15
v1.3	General: Replace: versioning, should have been 0.1.2 15		Replace: <code>\@@_map:n</code> by <code>\@@_oper_function:n</code> 15
v1.4	General: Add: <code>\erw_accum</code> 15		Replace: <code>\erw_seq_fold:NN</code> by <code>\erw_oper_fold_seq:NN</code> and likewise for variants 15
	Add: <code>\erw_int_range</code> 15	v1.9	General: Add:
	Add: <code>\erw_is_matrix</code> (to check arg of <code>\erw_tl_map_thread:Nn</code>) 15		<code>\erw_sys_timestamp_delimiter:</code> . 15
	Add: <code>\erw_merge</code> 15		Add: <code>\erw_tl_join:nn</code> and variants . 15
	Add: <code>\erw_set_map_inline</code> 15		Rename: <code>\erw_append_arg:nn</code> to <code>\erw_tl_append_item:nn</code> 15
	Add: <code>\erw_set_map</code> 15		Rename:
	Remove: <code>\erw_items_to</code> (redundant with <code>\tl_range:nnn</code>) . 15		<code>\erw_oper_gset_function:N</code> to <code>\erw_tl_gset_function:N</code> (and variants) 15
v1.5	General: Modify: source repository . . 15	v2.0	General: Add:
	Rearrange: frontend/backend sections 15		<code>\erw_jobnametimestamp:nn</code> and variants 15
	Remove: <code>disambig</code> 15		Remove: <code>\merge:nn</code> (redundant with <code>\erw_join:nn</code>) 15
	Split Section Preliminaries into Conventions and Requirement. . . 15		

\erw_oper_fold_seq:NN	16	\erw_tl_split:nnn	7, 481, 494
\erw_oper_gset_function:N	16	erw internal commands:	
\erw_option:n	5, 531	_erw_cs_name:N	3
\erw_prop_put:NN	5, 17, 130, 138	_erw_csint_ext_tl	74
\erw_prop_put:Nnn	5, 17, 139, 147, 150	\g_erw_csint_int	39, 40, 52, 69, 73
\erw_prop_to_clist:Nn		\g_erw_csint_name_tl	40, 53
.....	5, 17, 116, 128, 189	_erw_function:n	176, 181
\erw_repeat	16	_erw_int_range:nnn	76, 86, 99, 103
\erw_seq_compose:nN	5, 5, 192	_erw_keyval_function:n	118, 123
\erw_seq_compose_c:nN	5, 196	_erw_map:nn	349, 465
\erw_seq_compose_vers:nN	5, 200, 202	_erw_oper_compose:NnN	
\erw_seq_fold:NN	16, 17	151, 194, 198, 409, 419
\erw_seq_from_clist:Nn		\g_erw_oper_fold_apply_par_tl	
.....	5, 17, 204, 208, 210	168, 444
\erw_seq_from_prop:NnN		\g_erw_oper_fold_set_par_tl	164, 442
.....	6, 17, 211, 215, 217	_erw_prop_append:nn	132, 136
\erw_seq_put_right:Nn		_erw_seq_fold:NN	
.....	6, 17, 218, 222, 224	17, 194, 198, 225, 231
\erw_seq_use:Nn	6, 17, 232	\g_erw_seq_fold_item_tl	
\erw_set_map	16	173, 227, 228, 229
\erw_set_map_inline	16	_erw_seq_set_from_clist:Nn	
\erw_split	16	174, 185, 188, 207
\erw_sys_jobnametimestamp:	6, 328	_erw_seq_set_from_prop:Nnn	
\erw_sys_jobnametimestamp:nn	6, 327	186, 191, 214
\erw_sys_timestamp:	6, 302, 337	_erw_sys_date:N	240
\erw_sys_timestamp:nn	6, 296, 333	_erw_sys_date_dec:	240, 282
\erw_sys_timestamp_delimiter:		_erw_sys_date_hex:	240, 283
.....	6, 16, 329	_erw_sys_datetime_base:n	260, 307
\erw_tl_append_item:nn	6, 16, 88, 403	_erw_sys_datetime_dec:	282
\erw_tl_compose:nN	6, 407, 414	_erw_sys_datetime_dec:n	260
\erw_tl_compose:nn	6, 411	_erw_sys_datetime_hex:	283
\erw_tl_compose_c:nN	6, 417, 424	_erw_sys_datetime_hex:n	260
\erw_tl_compose_c:nn	6, 421, 436	_erw_sys_datetime_join:nn	260
\erw_tl_compose_vers:nN	6, 427, 429	_erw_sys_datetime_period:n	260, 307
\erw_tl_compose_vers:nn	6, 431	_erw_sys_jobnametimestamp:	
\erw_tl_fold:NN		290, 298, 328
.....	6, 228, 409, 419, 440, 446	_erw_sys_jobnametimestamp:n	290
\erw_tl_gset_function:N	6, 16, 447	_erw_sys_jobnametimestamp:nn	
\erw_tl_gset_function:n	1, 6, 7, 451	292, 327
\erw_tl_join:nn	7, 16, 35, 286, 294, 300	_erw_sys_jobnametimestamp-	
\erw_tl_join:nnn	7, 36, 270	prefix:	284
\erw_tl_join:nnnn	7, 37	_erw_sys_set_delim:nn	309, 319
\erw_tl_join:nnnnn	7, 38	_erw_sys_time_dec:	251, 282
\erw_tl_last_item:n	7, 455	_erw_sys_time_hex	251
\erw_tl_map:n	7, 157, 463, 470, 475	_erw_sys_time_hex:	259, 283
\erw_tl_map:Nn	7, 467	_erw_sys_timestamp:nn	304, 335, 339
\erw_tl_map_inline:nn	7, 472	\g_erw_sys_timestamp_delim_str	
\erw_tl_map_thread:Nn	7, 16, 517	270, 288, 312, 331
\erw_tl_map_thread_at:Nnn	7, 496, 524	\g_erw_tl_compose_tl	
\erw_tl_math_thread:Nn	7	341, 413, 414, 415, 423, 424, 425
\erw_tl_math_thread_at:Nnn	7	_erw_tl_map_thread_at:Nnn	354, 503
\erw_tl_repeat:nn	7, 477	_erw_tl_map_thread_at:Nnnn	354, 504
\erw_tl_separators:n	7, 17, 236, 527	_erw_tl_map_thread_at:Nnnnn	
\erw_tl_split:nn	7, 492	354, 505

__erw_tl_map_thread_at:Nnnnnn ..	\msg_new:nnn
..... 354, 506 107, 108, 113, 114, 115, 238, 239
__erw_tl_separators:nn 380, 529	
exp commands:	N
\exp_args:Nf	\newnumbrdcs
..... 119, 157, 357, 362, 363, 368, 16
369, 370, 375, 376, 377, 378, 498, 521	\numbrdcs
\exp_args:No 16
306	\numbrdcsnew
\exp_args:Nof 16
457	
\exp_args:Nx	O
88	oper / fold_apply_par (option)
\exp_last_unbraced:Nf	5
5	oper / fold_set_par (option)
\exp_last_unbraced:NNf	5
234	options:
\exp_last_unbraced:No	oper / fold_apply_par
318	5
\exp_last_unbraced:Nx	oper / fold_set_par
435	5
\exp_not:N	sys / timestamp_delim
398	5
\ExplSyntaxOff	
535	P
\ExplSyntaxOn	
2	prg commands:
	\prg_replicate:nn
G	385
g internal commands:	\ProcessPackageKeysOption
\g__erw_tl_function:n	16
..... 153, 342, 352, 449, 453, 469, 474	prop commands:
	\prop_gput:Nnn
I	134
int commands:	\prop_if_exist:NTF
\int_case:nnTF	141
262, 382, 498	\prop_item:Nn
\int_compare:nNnTF	118, 134
78	\prop_map_function:NN
\int_eval:n	136
80, 90, 93, 242, 253	\prop_new:N
\int_incr:N	146
52	\prop_put:Nnn
\int_new:N	143
39	
\int_step_function:nnnN	Q
48, 63	quark commands:
\int_step_inline:nn	\quark_if_recursion_tail_stop:n
105, 519	351
\int_step_inline:nnnn	\q_recursion_stop
479	465
\int_to_alph:n	\q_recursion_tail
43, 45	465
\int_to_hex:n	
249, 250, 259	S
\int_zero:N	
73	seq commands:
	\seq_get_right:NN
K	227
keys commands:	\seq_if_exist:NTF
\keys_define	206, 213, 220
16	\seq_new:N
\keys_define:nn	208, 215, 222
162, 314	\seq_put_right:Nn
\keys_set:nn	178, 221, 229
533	\seq_use:Nnnn
keyval commands:	235
\keyval_parse:NNn	str commands:
122, 180	\str_case:nnTF
	273
M	\subsection
map commands:	325
\map_thread	sys / timestamp_delim (option)
16	5
merge commands:	sys commands:
\merge:nn	\c_sys_day_int
16	246
msg commands:	\c_sys_hour_int
\msg_error:nnn	255
..... 111, 202, 268, 280, 344, 429, 512	\c_sys_jobname_str
\msg_error:nnnn	287
396	\c_sys_minute_int
	256
	\c_sys_month_int
	245
	\c_sys_year_int
	244

T			
tl commands:		\tl_reverse:n 159	
\c_empty_tl	267, 279, 394	\tl_set:Nn	40, 74, 413, 423
\tl_count:n	460, 500, 521, 529	\tl_tail:n	120, 486
\tl_head:n	483, 521	token commands:	
\tl_item:nn	357, 362, 363,	\token_if_cs:NTF	55
368, 369, 370, 375, 376, 377, 378, 457		U	
\tl_map_function:nN	434	use commands:	
\tl_map_inline:nn	484	\use:N	307, 311, 331, 442, 444, 484
\tl_new:N	173, 341	\use_i:nn	389, 390
\tl_range:nnn	16	\use_i:nnn	5
\tl_range_braced:nnn	64	\use_ii:nn	388, 390
		\usepackage	4

Part IV

Implementation

1 Opening

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

2 basics

2.1 backend

```
3 \cs_new: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_new: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_new:Nn \erw_cs_apply:Nnn
13 {
14   #1{#2}{#3}
15 }
16 \cs_new:Nn \erw_cs_apply:Nnnn
17 {
18   #1{#2}{#3}{#4}
19 }
20 \cs_new: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_new: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_new: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 clist

3.1 backend

3.2 frontend

4 csint

4.1 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}}
```

4.2 frontend

```
41 \cs_new:Nn \erw_csint:nn
42 {
43   \erw_cs_apply:cn{__erw_csint_\int_to_alph:n{#1}:n}{#2}
44 }
45 \cs_new: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_new: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 }
```

5 int

5.1 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 }

```

5.2 frontend

```

97 \cs_new:Nn \erw_int_range:nn
98 {
99   \__erw_int_range:nnn {{#1}}{#1}{#2}
100 }
101 \cs_new: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 }

```

6 keyval

```

107 \msg_new:nnn{__erw}{keyval/keyonly}{passed~key~#1~val~#2~where~keyonly}
108 \msg_new:nnn{__erw}{keyval/mandatval}{key~#1~has~no~matching~val}
109 \cs_new:Nn \erw_keyval_keyonly:nn
110 {
111   \msg_error:nnn{__erw}{keyval/keyonly}{#1}{#2}
112 }

```

7 msg

7.1 backend

```

113 \msg_new:nnn{__erw}{generic}{#1}
114 \msg_new:nnn{__erw}{notdecl}{#1~not~declared}
115 \msg_new:nnn{__erw}{notset}{#1~not~set}

```

8 prop

8.1 backend

8.2 frontend

```
116 \cs_new_protected:Nn \erw_prop_to_clist:Nn
117 {
118   \cs_set:Nn \__erw_keyval_function:n {,\prop_item:Nn#1{##1}}
119   \exp_args:Nf
120   \tl_tail:n
121   {
122     \keyval_parse:NNn
123     \__erw_keyval_function:n
124     \erw_keyval_keyonly:nn
125     {#2}
126   }
127 }
128 \cs_generate_variant:Nn \erw_prop_to_clist:Nn { c }
129
130 \cs_new_protected:Nn \erw_prop_put:NN
131 {
132   \cs_set:Nn \__erw_prop_append:nn
133   {
134     \prop_gput:Nnx #1 {##1}{ \prop_item:Nn #2{##1} }
135   }
136   \prop_map_function:NN #2 \__erw_prop_append:nn
137 }
138 \cs_generate_variant:Nn \erw_prop_put:NN { cc }
139 \cs_new_protected:Nn \erw_prop_put:Nnn
140 {
141   \prop_if_exist:NTF#1
142   {
143     \prop_put:Nnn #1 {#2}{#3}
144   }
145   {
146     \prop_new:N #1
147     \erw_prop_put:Nnn #1{#2}{#3}
148   }
149 }
150 \cs_generate_variant:Nn \erw_prop_put:Nnn { c }
```

9 oper

9.1 backend

```
151 \cs_new:Nn \__erw_oper_compose:NnN
152 {
153   \erw_cs_set_inline:Nn \g__erw_tl_function:n
154   {
155     #1{##1}#3
156   }
157   \exp_args:Nf\erw_tl_map:n
158   {
159     \tl_reverse:n{#2}
160   }
161 }
```

9.2 frontend


```

162 \keys_define:nn{__erw}
163 {
164   oper/fold_set_par.tl_gset:N = \g__erw_oper_fold_set_par_tl,
165   oper/fold_set_par.value_required:n = true,
166   oper/fold_set_par.default:n = {Nf},
167   oper/fold_set_par.initial:n = {Nf},
168   oper/fold_apply_par.tl_gset:N = \g__erw_oper_fold_apply_par_tl,
169   oper/fold_apply_par.value_required:n = true,
170   oper/fold_apply_par.default:n = {Nf},
171   oper/fold_apply_par.initial:n = {Nf}
172 }

```

10 seq

10.1 backend

```

173 \tl_new:N \g__erw_seq_fold_item_tl
174 \cs_new_protected:Nn\__erw_seq_set_from_clist:Nn
175 {
176   \cs_set_protected:Nn \__erw_function:n
177   {
178     \seq_put_right:Nn #1{##1}
179   }
180   \keyval_parse:NNn
181   \__erw_function:n
182   \erw_keyval_keyonly:nn
183   {#2}
184 }
185 \cs_generate_variant:Nn \__erw_seq_set_from_clist:Nn { c }
186 \cs_new_protected:Nn\__erw_seq_set_from_prop:NNn
187 {
188   \__erw_seq_set_from_clist:Nn #1
189   {\erw_prop_to_clist:Nn #2 {#3}}
190 }
191 \cs_generate_variant:Nn \__erw_seq_set_from_prop:NNn { cc }

```

10.2 frontend

```

192 \cs_new:Nn \erw_seq_compose:nN
193 {
194   \__erw_oper_compose:NnN \__erw_seq_fold:NN {#1} #2
195 }
196 \cs_new:Nn \erw_seq_compose_c:nN
197 {
198   \__erw_oper_compose:NnN \__erw_seq_fold:cN {#1} #2
199 }
200 \cs_new:Nn \erw_seq_compose_vers:nN
201 {
202   \msg_error:nnn{__erw}{notdecl}{\erw_seq_compose_vers:nN}
203 }
204 \cs_new_protected:Nn\erw_seq_from_clist:Nn
205 {
206   \seq_if_exist:NTF#1
207   {\__erw_seq_set_from_clist:Nn#1{#2}}
208   {\seq_new:N#1\erw_seq_from_clist:Nn#1{#2}}

```

```

209 }
210 \cs_generate_variant:Nn \erw_seq_from_clist:Nn { c }
211 \cs_new_protected:Nn\erw_seq_from_prop:NNn
212 {
213   \seq_if_exist:NTF#1
214   {\__erw_seq_set_from_prop:NNn#1#2{#3}}
215   {\seq_new:N#1\erw_seq_from_prop:NNn#1#2{#3}}
216 }
217 \cs_generate_variant:Nn \erw_seq_from_prop:NNn { cc }
218 \cs_new_protected:Nn\erw_seq_put_right:Nn
219 {
220   \seq_if_exist:NTF#1
221   {\seq_put_right:Nn#1{#2}}
222   {\seq_new:N#1\erw_seq_put_right:Nn #1{#2}}
223 }
224 \cs_generate_variant:Nn\erw_seq_put_right:Nn { c }
225 \cs_new:Nn \__erw_seq_fold:NN
226 {
227   \seq_get_right:NN #2 \g__erw_seq_fold_item_tl
228   \erw_tl_fold:NN #1 \g__erw_seq_fold_item_tl
229   \seq_put_right:No #2 {\g__erw_seq_fold_item_tl}
230 }
231 \cs_generate_variant:Nn \__erw_seq_fold:NN { cN }
232 \cs_new:Nn \erw_seq_use:Nn
233 {
234   \exp_last_unbraced:NNf
235   \seq_use:Nnnn #1
236   \erw_tl_separators:n{#2}
237 }

```

11 sys

11.1 backend

```

238 \msg_new:nnn{__erw}{timestamp / base}{Calling~#1,~arg~must~be~'dec|hex'}
239 \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:
240 \cs_new:Nn \__erw_sys_date_dec:
241 {
242   \int_eval:n
243   {
244     \c_sys_year_int * 10000
245     +\c_sys_month_int * 100
246     +\c_sys_day_int * 1
247   }
248 }
249 \cs_new:Nn \__erw_sys_date:N{\int_to_hex:n{\__erw_sys_date_dec:}}
250 \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
251 \cs_new:Nn \__erw_sys_time_dec:
252 {

```

```

253 \int_eval:n
254 {
255     \c_sys_hour_int * 100
256     +\c_sys_minute_int * 1
257 }
258 }
259 \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
260 \cs_new:Nn\__erw_sys_datetime_base:n
261 {
262     \int_case:nnTF{#1}
263     {
264         {10}{dec}
265         {16}{hex}
266     }
267     {\c_empty_tl}
268     {\msg_error:nnn{\__erw}{timestamp / base}{\__erw_sys_datetime_base:n{#1}}}
269 }
270 \cs_new:Nn\__erw_sys_datetime_join:nn{\erw_tl_join:nnn{#1}{\g__erw_sys_timestamp_delim_str}{#2}}
271 \cs_new:Nn\__erw_sys_datetime_period:n
272 {
273     \str_case:nnTF{#1}
274     {
275         {date}{date}
276         {time}{time}
277         {datetime}{datetime}
278     }
279     {\c_empty_tl}
280     {\msg_error:nnn{\__erw}{timestamp / period}{\__erw_sys_datetime_period:n{#1}}}
281 }
282 \cs_new:Nn\__erw_sys_datetime_dec: {\__erw_sys_datetime_join:nn{\__erw_sys_date_dec:}{\__erw_sys_date_dec:}}
283 \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_jobnametimestamp_prefix:

```

284 \cs_new:Nn\__erw_sys_jobnametimestamp_prefix:
285 {
286     \erw_tl_join:nn
287     {\c_sys_jobname_str}
288     {\g__erw_sys_timestamp_delim_str}
289 }
290 % \begin{macro}{\__erw_sys_jobnametimestamp:n, \__erw_sys_jobnametimestamp:}
291 %     \begin{macrocode}
292 \cs_new:Nn\__erw_sys_jobnametimestamp:nn
293 {
294     \erw_tl_join:nn
295     {\__erw_sys_jobnametimestamp_prefix:}
296     {\erw_sys_timestamp:nn{#1}{#2}}
297 }
298 \cs_new:Nn\__erw_sys_jobnametimestamp:
299 {

```

```

300 \erw_tl_join:nn
301 {\__erw_sys_jobnametimestamp_prefix:}
302 {\erw_sys_timestamp:}
303 }

(End definition for \__erw_sys_jobnametimestamp_prefix:.)

```

__erw_sys_timestamp:nn

```

304 \cs_new:Nn\__erw_sys_timestamp:nn
305 {
306   \exp_args:No
307   \use:c{\__erw_sys\___erw_sys_datetime_period:n{#1}\__erw_sys_datetime_base:n{#2}:}
308 }
309 \cs_new_protected:Nn \__erw_sys_set_delim:nn
310 {
311   \use:c{tl_gset:N#1}
312   \g__erw_sys_timestamp_delim_str{#2}
313 }

(End definition for \__erw_sys_timestamp:nn.)

314 \keys_define:nn{\__erw}
315 {
316   sys / timestamp_delim .code:n =
317   {
318     \exp_last_unbraced:No
319     \__erw_sys_set_delim:nn{n}{#1}
320   },
321   sys / timestamp_delim .value_required:n = true,
322   sys / timestamp_delim .default:n = {-},
323   sys / timestamp_delim .initial:n = {-}
324 }
325 % \subsection{frontend}
326 % \begin{macrocode}
327 \cs_new:Nn\erw_sys_jobnametimestamp:nn{\__erw_sys_jobnametimestamp:nn{#1}{#2}}
328 \cs_new:Nn\erw_sys_jobnametimestamp:{\__erw_sys_jobnametimestamp:}
329 \cs_new:Nn\erw_sys_timestamp_delimiter:
330 {
331   \use:N \g__erw_sys_timestamp_delim_str
332 }
333 \cs_new:Nn\erw_sys_timestamp:nn
334 {
335   \__erw_sys_timestamp:nn{#1}{#2}
336 }
337 \cs_new:Nn\erw_sys_timestamp:
338 {
339   \__erw_sys_timestamp:nn{datetime}{16}
340 }

```

12 tl

12.1 backend

```

341 \tl_new:N \g__erw_tl_compose_tl

```

```

\g__erw_tl_function:n
342 \cs_new_protected:Nn \g__erw_tl_function:n
343 {
344   \msg_error:nnn
345   {erw}
346   {notset}
347   {\g__erw_tl_function:n}
348 }

(End definition for \g__erw_tl_function:n.)

```

```

\__erw_map:nn
349 \cs_set_protected:Nn \__erw_map:nn
350 {
351   \quark_if_recursion_tail_stop:n{#1}
352   \g__erw_tl_function:n{#1} \__erw_map:nn{#2}
353 }

(End definition for \__erw_map:nn.)

```

```

\__erw_tl_map_thread_at:Nnn
\__erw_tl_map_thread_at:Nnnn
  \__erw_tl_map_thread_at:Nnnnn
    \__erw_tl_map_thread_at:Nnnnnn
354 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnn
355 {
356   \erw_cs_apply:Nn #1
357   {\exp_args:Nf\tl_item:nn {#3} {#2} }
358 }
359 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnn
360 {
361   \erw_cs_apply:Nnn #1
362   {\exp_args:Nf\tl_item:nn {#3} {#2} }
363   {\exp_args:Nf\tl_item:nn {#4} {#2} }
364 }
365 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnnn
366 {
367   \erw_cs_apply:Nnnn #1
368   {\exp_args:Nf\tl_item:nn {#3} {#2} }
369   {\exp_args:Nf\tl_item:nn {#4} {#2} }
370   {\exp_args:Nf\tl_item:nn {#5} {#2} }
371 }
372 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnnnn
373 {
374   \erw_cs_apply:Nnnnn #1
375   {\exp_args:Nf\tl_item:nn {#3} {#2} }
376   {\exp_args:Nf\tl_item:nn {#4} {#2} }
377   {\exp_args:Nf\tl_item:nn {#5} {#2} }
378   {\exp_args:Nf\tl_item:nn {#6} {#2} }
379 }

(End definition for \__erw_tl_map_thread_at:Nnn and others.)

```

```

\__erw_tl_separators:nn #1 : < int >

```

```

#2 : < items >

380 \cs_new:Nn \__erw_tl_separators:nn
381 {
382   \int_case:nnTF {#1}
383   {
384     {1}
385     { \prg_replicate:nn{ 3 }{#2} }
386     {2}
387     {
388       { \use_ii:nn #2 }
389       { \use_i:nn #2 }
390       { \use_i:nn #2\use_ii:nn #2 }
391     }
392     {3}{#2}
393   }
394   { \c_empty_tl }
395   {
396     \msg_error:nnnn { __erw }
397     { separ }
398     { \exp_not:N \__erw_tl_separators:nn }
399     {#2}
400   }
401 }
402 \cs_generate_variant:Nn \__erw_tl_separators:nn { e }

(End definition for \__erw_tl_separators:nn.)

```

12.2 frontend

```

403 \cs_new:Nn \erw_tl_append_item:nn
404 {
405   {#1{#2}}
406 }
407 \cs_new:Nn \erw_tl_compose:nN
408 {
409   \__erw_oper_compose:NnN \erw_tl_fold:NN {#1} #2
410 }
411 \cs_new:Nn \erw_tl_compose:nn
412 {
413   \tl_set:Nn \g__erw_tl_compose_tl {#2}
414   \erw_tl_compose:nN{#1}\g__erw_tl_compose_tl
415   \g__erw_tl_compose_tl
416 }
417 \cs_new:Nn \erw_tl_compose_c:nN
418 {
419   \__erw_oper_compose:NnN \erw_tl_fold:cN {#1} #2
420 }
421 \cs_new:Nn \erw_tl_compose_c:nn
422 {
423   \tl_set:Nn \g__erw_tl_compose_tl {#2}
424   \erw_tl_compose_c:nN{#1}\g__erw_tl_compose_tl
425   \g__erw_tl_compose_tl
426 }
427 \cs_new:Nn \erw_tl_compose_vers:nN

```

```

428 {
429   \msg_error:nnn{__erw}{notdecl}{\erw_tl_compose_vers:nN}
430 }
431 \cs_new:Nn \erw_tl_compose_vers:nn
432 {
433   \erw_csint_reset:{}
434   \tl_map_function:nN{#1}\erw_csint_new:n
435   \exp_last_unbraced:Nx
436   \erw_tl_compose_c:nn
437   {{\erw_csint_names_braced:{}}}
438   {#2}
439 }
440 \cs_new:Nn \erw_tl_fold:NN
441 {
442   \use:c{tl_set:\g__erw_oper_fold_set_par_tl}
443   #2
444   {\use:c{erw_cs_apply:\g__erw_oper_fold_apply_par_tl}{#1}{#2}}
445 }
446 \cs_generate_variant:Nn \erw_tl_fold:NN {cN}
447 \cs_new:Nn \erw_tl_gset_function:N
448 {
449   \erw_cs_gset_eq:NN \g__erw_tl_function:n #1
450 }
451 \cs_new:Nn \erw_tl_gset_function:n
452 {
453   \erw_cs_gset_inline:Nn \g__erw_tl_function:n {#1}
454 }
455 \cs_new:Nn \erw_tl_last_item:n
456 {
457   \exp_args:Nof \tl_item:nn
458   {#1}
459   {
460     \tl_count:n{#1}
461   }
462 }
463 \cs_new_protected:Nn \erw_tl_map:n
464 {
465   \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
466 }
467 \cs_new_protected:Nn \erw_tl_map:Nn
468 {
469   \cs_set_eq:NN \g__erw_tl_function:n #1
470   \erw_tl_map:n{#2}
471 }
472 \cs_new_protected:Nn \erw_tl_map_inline:nn
473 {
474   \erw_cs_set_inline:Nn \g__erw_tl_function:n {#1}
475   \erw_tl_map:n{#2}
476 }
477 \cs_new:Nn \erw_tl_repeat:nn
478 {
479   \int_step_inline:nnnn{1}{1}{#1}{#2}
480 }
481 \cs_new:Nn \erw_tl_split:nnn

```

```

482 {
483   \tl_head:n{#1}
484   \use:c{exp_args:#3} \tl_map_inline:nn
485   {
486     \tl_tail:n
487     {
488       #1
489     }
490   }{#2##1}
491 }
492 \cs_new:Nn \erw_tl_split:nn
493 {
494   \erw_tl_split:nnn{#1}{#2}{Nf}
495 }
496 \cs_new_protected:Nn \erw_tl_map_thread_at:Nnn
497 {
498   \exp_args:Nf\int_case:nnTF
499   {
500     \tl_count:n{#3}
501   }
502   {
503     {1}{ \__erw_tl_map_thread_at:Nnn #1{#2}#3 }
504     {2}{ \__erw_tl_map_thread_at:Nnnn #1{#2}#3 }
505     {3}{ \__erw_tl_map_thread_at:Nnnnn #1{#2}#3 }
506     {4}{ \__erw_tl_map_thread_at:Nnnnnn #1{#2}#3 }
507   }
508   {
509     % Do nothing
510   }
511   {
512     \msg_error:nnn{__erw}
513     {generic}
514     {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
515   }
516 }
517 \cs_new_protected:Nn \erw_tl_map_thread:Nn
518 {
519   \int_step_inline:nn
520   {
521     \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
522   }
523   {
524     \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
525   }
526 }
527 \cs_new:Nn \erw_tl_separators:n
528 {
529   \__erw_tl_separators:en{ \tl_count:n{#1} }{#1}
530 }

```

13 option

```

531 \cs_new_protected:Nn\erw_option:n
532 {

```



```
533 \keys_set:nn{__erw}{#1}  
534 }
```

14 Closing

```
535 \ExplSyntaxOff
```