

The `erw-l3` package ^{*}

Erwann Rogard[†]

Released 2020/05/01

Abstract

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

Résumé

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

Contents

I	Usage	3
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
II	Listing	7
1	basics	7
	1.	7
2	csint	7
	2.	7

^{*}This file describes version v1.9, last revised 2020/05/01.

[†]firstname dot lastname AusTria gmail dot com

3	int	7
3.	7
4	seq	8
4.	8
5.	8
10.	8
5	sys	9
6.	9
7.	9
6	tl	9
8.	9
9.	10
11.	10
12.	11
13.	11
14.	11
III	Other	12
1	Acknowledgment	12
2	Install	12
3	Support	12
3.1	Platform	12
3.2	Engine	12
3.3	Results	12
	References	12
	Change History	13
	Index	14
IV	Implementation	17
1	Opening	17
2	basics	17
2.1	backend	17
2.2	frontend	17
3	csint	18
3.1	backend	18
3.2	frontend	18

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

Part I

Usage

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

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 {\control sequence}\{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{⟨control sequence⟩}{⟨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 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>	Semantics 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{<arg list>}{<arg>}</code>
-------------------------------------	--

<code>\erw_tl_compose:nN</code>	<code>\erw_tl_compose:nn{<control sequence list>}{<initial value>}</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<control sequence><token></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>	

<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>	

<code>\erw_tl_last_item:n</code>	<code>\erw_tl_last_time:n{<token list>}</code>
----------------------------------	--

<code>\erw_tl_map:n</code>	<code>\erw_tl_map:n{<items>}</code>
<code>\erw_tl_map:Nn</code>	

Semantics Maps over `<items>` 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{<integer>}{<token list>}</code>
--------------------------------	---

<code>\erw_tl_split:nnn</code>	<code>\erw_tl_split:nn{<items>}{<delimiter>}</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
```

```
20200501-1921
20200501-1921
1343c35%781
1343c35%781
erw-l3%1343c35%781
```

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:202005011921
D:20200501192114-04'00'
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^AT_EXcommunity[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^AT_EXengine

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^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

v1.0	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: Closing
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ;	12		Add: sys
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 convention ($g \circ f$ 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. section 6 . . .
	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		Rename: v0.0 to v1.0, etc.

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.

B

`\begin` 253, 254, 289

C

cs commands:

`\cs_generate_variant:Nn`
 11, 29, 34, 193, 200
`\cs_gset:Npn` 32
`\cs_new:Nn` 35, 36, 37,
 38, 46, 61, 67, 203, 212, 213, 214,
 222, 223, 233, 234, 245, 246, 247,
 255, 261, 267, 290, 291, 292, 296, 300
`\cs_new_protected:Nn`
 50, 71, 272, 304, 422
`\cs_set:Nn` . 3, 7, 12, 16, 20, 25, 30,
 41, 45, 66, 97, 101, 112, 123, 127,
 133, 137, 143, 147, 156, 160, 164,
 168, 172, 187, 194, 342, 346, 368, 372
`\cs_set:Npn` 24, 27, 76
`\cs_set_eq:NN` 360
`\cs_set_protected:Nn`
 311, 316, 321, 327,
 334, 354, 358, 363, 376, 387, 391, 412
`\cs_split_function:N` 5

E

erw commands:

`\erw_cs_apply:Nn` 3, 7, 11, 43, 318
`\erw_cs_apply:Nnn` 3, 12, 323
`\erw_cs_apply:Nnnn` 3, 16, 329
`\erw_cs_apply:Nnnnn` 3, 20, 336
`\erw_cs_gset_eq:NN` 170
`\erw_cs_gset_inline:Nn` . . . 30, 34, 174
`\erw_cs_identity:n` 3, 24
`\erw_cs_set_inline:Nn`
 3, 25, 29, 53, 114, 365
`\erw_csint:nn` 4, 41
`\erw_csint_name:n` . . . 4, 40, 45, 48, 66
`\erw_csint_names:nnn` 4, 46
`\erw_csint_names_braced:` . . 4, 67, 153
`\erw_csint_names_braced:n` . . 4, 63, 66
`\erw_csint_names_braced:nnn` 4, 61, 69
`\erw_csint_new:n` 4, 50, 150
`\erw_csint_reset:` 4, 71, 149
`\erw_int_range:n` 4, 101
`\erw_int_range:nn` 4, 97
`\erw_oper_compose:nN` 4, 123, 130
`\erw_oper_compose:nn` 4, 127

`\erw_oper_compose_c:nN` . . . 4, 133, 140
`\erw_oper_compose_c:nn` . . . 4, 137, 152
`\erw_oper_compose_seq:nN` 4, 156
`\erw_oper_compose_seq_c:nN` . . 4, 160
`\erw_oper_compose_seq_vers:nN` . . .
 4, 164, 166
`\erw_oper_compose_vers:nN` 4, 143, 145
`\erw_oper_compose_vers:nn` . . . 4, 147
`\erw_oper_fold:NN`
 5, 125, 135, 187, 193, 197
`\erw_oper_fold_seq:NN`
 5, 158, 162, 194, 200
`\erw_option:n` 6, 422
`\erw_seq:NN` 5
`\erw_sys_jobnametimestamp:` . . 5, 291
`\erw_sys_jobnametimestamp:nn` 5, 290
`\erw_sys_timestamp:` 5, 265, 300
`\erw_sys_timestamp:nn` . . . 5, 259, 296
`\erw_sys_timestamp_delimiter:` 5, 292
`\erw_tl_append_item:nn` . . . 5, 88, 342
`\erw_tl_gset_function:N` 5, 168
`\erw_tl_gset_function:n` 5, 172
`\erw_tl_join:n` 5
`\erw_tl_join:nn` . . 5, 35, 249, 257, 263
`\erw_tl_join:nnn` 5, 36, 233
`\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` . . . 5, 118, 354, 361, 366
`\erw_tl_map:Nn` 5, 358
`\erw_tl_map_inline:nn` 5, 363
`\erw_tl_map_thread:Nn` 6, 412
`\erw_tl_map_thread_at:Nnn` 6, 391, 419
`\erw_tl_merge:nn` 6, 368
`\erw_tl_repeat:nn` 6, 372
`\erw_tl_split:nn` 6, 387
`\erw_tl_split:nnn` 6, 376, 389

erw internal commands:

`\g_erw_compose_tl`
 110, 129, 130, 131, 139, 140, 141
`__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` 311, 356
`__erw_map_thread_at:Nnn` . . . 316, 398

	U	<code>\use_i:nnn</code>	5
use commands:			
	<code>\use:N</code>	189, 191, 270, 274, 294, 379	<code>\usepackage</code> 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 }

```

4.3 frontend

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_datetime_period:n}}
197 \cs_new:Nn \__erw_sys_datetime_hex: {\__erw_sys_datetime_join:nn{\__erw_sys_date_hex:}{\__erw_sys_datetime_period:n}}

```

(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:Nnnnn
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:Nnnnnn
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:Nnnnnnn
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~within~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

```