

The `erw-l3` package ^{*}

Erwann Rogard[†]

Released 2020/05/20

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	\erw_tl_gset_function:n	6
II	Listing	8
1	constants	8
1.	8

^{*}This file describes version v2.3, last revised 2020/05/20.

[†]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.	13
17.	13
18.	13
19.	13
20.	13
21.	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	20
1	Opening	20
2	basics	20
	2.1 backend	20
	2.2 frontend	20
3	clist	21
	3.1 backend	21
	3.2 frontend	21
4	csint	21
	4.1 backend	21
	4.2 frontend	21
5	int	21
	5.1 backend	21
	5.2 frontend	22
6	keyval	22
7	msg	22
	7.1 backend	22
	7.2 frontend	22
8	prop	23
	8.1 backend	23
	8.2 frontend	23
9	oper	23
	9.1 backend	23
	9.2 frontend	24
10	seq	24
	10.1 backend	24
	10.2 frontend	24
11	sys	25
	11.1 backend	25
12	tl	28
	12.1 backend	28
	12.2 frontend	29
13	option	31

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{⟨cs₁⟩...}{⟨token list⟩}</code>
<code>\erw_tl_compose:nn</code>	

<code>\erw_tl_compose_c:nN</code>	<code>\erw_tl_compose_c:nn{⟨cs name₁⟩...}{⟨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 code₁⟩...}{⟨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
\seq_put_right:Nn\l_tmpa_seq{ A }
\seq_put_right:Nn\l_tmpa_seq{ B }
\erw_seq_use:Nn \l_tmpa_seq{{~and~}}\\
\erw_seq_use:Nn \l_tmpa_seq{{, \ }{~and~}}\\
\erw_seq_use:Nn \l_tmpa_seq{{~and~}{, \ }{,~and~}}\\[1em]
\seq_put_right:Nn\l_tmpa_seq{ C }
\erw_seq_use:Nn \l_tmpa_seq{{~and~}}\\

```

```

\erw_seq_use:Nn \l_tmpa_seq{{,\ }{and-}}\\
\erw_seq_use:Nn \l_tmpa_seq{{~and-}{,\ }{~,~and-}}\\
\ExplSyntaxOff

```

```

A and B
A and B
A and B

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

```

```

20200520-2321
20200520-2321
1343c48%911
1343c48%911
erw-l3%1343c48%911

```

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:202005202321
D:20200520232158-04'00'
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
\erw_tl_repeat:nn{3}{x}
\ExplSyntaxOff
```

xxx

Listing 18.

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

a==b==c

Listing 19.

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

(a)(b)(c)

Listing 20.

```
\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 21.

```

\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	v1.7	General: Add: <code>option</code> 15
	<code>\ProcessPackageKeysOption</code> ; . . . 15		Add: <code>sys</code> 15
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ; 15		Move: <code>\erw_fold_apply_par:n</code> . . 15
v1.2	General: 15		Move: <code>\erw_fold_set_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		Rearrange: structure of implementation, e.g. section 9 . . . 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		Remove: document level functions, <code>\numbrdcsnew</code> , <code>\numbrdcs</code> 15
	Add: <code>\erw_items_to</code> 15		Replace: listing's implem with that of <code>tocloft</code> 15
	Add: <code>\erw_last_item</code> 15		Replace: vers. numb. from 3 to 2 digits 15
	Add: <code>\erw_repeat</code> 15	v1.8	General: Add: function for all frontend functions. 15
	Add: <code>\erw_split</code> 15		Remove: <code>\erw_cs_set_eq:NN</code> and variants 15
	Add: <code>\map_thread</code> 15		Remove: <code>\erw_is_matrix:n</code> (predicate must be expandable) . . 15
	Front end cmds no longer generated with module <code>disambig</code> ; Option of the same name deleted; 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
	Re-arrange: the doc to clearly separate frontend from backend . . 15		Replace: <code>@@_map:n</code> by <code>@@_oper_function:n</code> 15
v1.3	General: Replace: versioning, should have been 0.1.2 15		Replace: <code>\erw_seq_fold:NN</code> by <code>\erw_oper_fold_seq:NN</code> and likewise for variants 15
v1.4	General: Add: <code>\erw_accum</code> 15	v1.9	General: Add:
	Add: <code>\erw_int_range</code> 15		<code>\erw_sys_timestamp_delimiter:</code> . 15
	Add: <code>\erw_is_matrix</code> (to check arg of <code>\erw_tl_map_thread:Nn</code>) 15		Add: <code>\erw_tl_join:nn</code> and variants . 15
	Add: <code>\erw_merge</code> 15		Rename: <code>\erw_append_arg:nn</code> to <code>\erw_tl_append_item:nn</code> 15
	Add: <code>\erw_set_map_inline</code> 15		Rename:
	Add: <code>\erw_set_map</code> 15		<code>\erw_oper_gset_function:N</code> to <code>\erw_tl_gset_function:N</code> (and variants) 15
	Remove: <code>\erw_items_to</code> (redundant with <code>\tl_range:nnn</code>) . 15	v2.0	General: Add:
v1.5	General: Modify: source repository . . 15		<code>\erw_jobnametimestamp:nn</code> and variants 15
	Rearrange: frontend/backend sections 15		Remove: <code>\merge:nn</code> (redundant with <code>\erw_join:nn</code>) 15
	Remove: <code>disambig</code> 15		
	Split Section Preliminaries into Conventions and Requirement. . . 15		

Rename: v0.0 to v1.0, etc.	15	v2.2	General: Add: \erw_seq_use:Nn	15
v2.1			Add: \erw_tl_separators:n	15
General: Add:		v2.3	General: Add: \msg_new:nnn, module	
\erw_prop_to_clist:Nn,			erw, messages: csnsset	15
\erw_prop_put:NN, and			Add: \msg_new:nnn, module erw,	
\erw_prop_put:Nnn	15		messages: keyval/...	15
Add: \erw_seq_from_clist:Nn,			Fix: 'mark as private code' (hiherto	
\erw_seq_from_prop:Nnn, and			unnoticed)	15
\erw_seq_put_right:Nn	15		Modify: behavior of	
Move: all functions under section 9			\erw_seq_use:Nn	15
to section 12 or section 10 , except			Move: all \msg_new:Nnnn	
\@@_oper_compose:NnN	15		statements under same heading ..	15
Replace: \erw_seq_fold:NN by				
__erw_seq_fold:NN	15			

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	D
@@ commands:	\disambignewcmd <i>16</i>
\@@_map:n <i>16</i>	
\@@_oper_compose:NnN <i>17</i>	
\@@_oper_function:n <i>16</i>	
B	E
\begin <i>291, 292, 327</i>	erw commands:
C	\erw_accum <i>16</i>
cs commands:	\erw_append_arg:nn <i>16</i>
\cs_generate_variant:Nn	\erw_compose <i>16</i>
.... <i>12, 30, 35, 131, 141, 153,</i>	\erw_cs_apply:Nn <i>4, 8, 12, 44, 357</i>
<i>188, 194, 213, 220, 227, 234, 403, 447</i>	\erw_cs_apply:Nnn <i>4, 13, 362</i>
\cs_gset:Npn <i>33</i>	\erw_cs_apply:Nnnn <i>4, 17, 368</i>
\cs_new:Nn <i>4, 8, 13, 17, 21, 26, 31, 36,</i>	\erw_cs_apply:Nnnnn <i>4, 21, 375</i>
<i>37, 38, 39, 42, 46, 47, 62, 67, 68, 98,</i>	\erw_cs_gset_eq:NN <i>450</i>
<i>102, 108, 154, 195, 199, 203, 228,</i>	\erw_cs_gset_inline:Nn ... <i>31, 35, 454</i>
<i>235, 241, 250, 251, 252, 260, 261,</i>	\erw_cs_identity:n <i>4, 16, 25</i>
<i>271, 272, 283, 284, 285, 293, 299,</i>	\erw_cs_set_eq:NN <i>16</i>
<i>305, 328, 329, 330, 334, 338, 381,</i>	\erw_cs_set_inline:Nn
<i>404, 408, 412, 418, 422, 428, 432,</i> <i>4, 26, 30, 54, 156, 475</i>
<i>441, 448, 452, 456, 478, 482, 493, 528</i>	\erw_csint:nn <i>4, 42</i>
\cs_new_protected:Nn . <i>51, 72, 119,</i>	\erw_csint_name:n ... <i>4, 41, 46, 49, 67</i>
<i>133, 142, 177, 189, 207, 214, 221,</i>	\erw_csint_names:nnn <i>4, 47</i>
<i>310, 343, 464, 468, 473, 497, 518, 532</i>	\erw_csint_names_braced: .. <i>4, 68, 438</i>
\cs_set:Nn <i>121, 135</i>	\erw_csint_names_braced:n .. <i>4, 64, 67</i>
\cs_set:Npn <i>25, 28, 77</i>	\erw_csint_names_braced:nnn <i>4, 62, 70</i>
\cs_set_eq:NN <i>470</i>	\erw_csint_new:n <i>5, 51, 435</i>
\cs_set_protected:Nn	\erw_csint_reset: <i>5, 72, 434</i>
.... <i>179, 350, 355, 360, 366, 373</i>	\erw_fold_apply_par:n <i>16</i>
\cs_split_function:N <i>6</i>	\erw_fold_set_par:n <i>16</i>
	\erw_identity:n <i>16</i>
	\erw_int_range <i>16</i>
	\erw_int_range:n <i>5, 102</i>
	\erw_int_range:nn <i>5, 98</i>

<code>\erw_is_matrix</code>	16	<code>\erw_tl_map_inline:nn</code>	7, 473
<code>\erw_is_matrix:n</code>	16	<code>\erw_tl_map_thread:Nn</code>	7, 16, 518
<code>\erw_items_to</code>	16	<code>\erw_tl_map_thread_at:Nnn</code>	7, 497, 525
<code>\erw_jobnametimestamp:nn</code>	16	<code>\erw_tl_math_thread:Nn</code>	7
<code>\erw_join:nn</code>	16	<code>\erw_tl_math_thread_at:Nnn</code>	7
<code>\erw_keyval_keyonly:nn</code>	108, 127, 185	<code>\erw_tl_repeat:nn</code>	7, 478
<code>\erw_last_item</code>	16	<code>\erw_tl_separators:n</code>	7, 17, 239, 528
<code>\erw_merge</code>	16	<code>\erw_tl_split:nn</code>	7, 493
<code>\erw_oper_fold_seq:NN</code>	16	<code>\erw_tl_split:nnn</code>	7, 482, 495
<code>\erw_oper_gset_function:N</code>	16	erw internal commands:	
<code>\erw_option:n</code>	5, 532	<code>__erw_cs_name:N</code>	4
<code>\erw_prop_put:NN</code>	5, 17, 133, 141	<code>__erw_csint_ext_tl</code>	75
<code>\erw_prop_put:Nnn</code>	5, 17, 142, 150, 153	<code>\g__erw_csint_int</code>	40, 41, 53, 70, 74
<code>\erw_prop_to_clist:Nn</code>		<code>\g__erw_csint_name_tl</code>	41, 54
	5, 17, 119, 131, 192	<code>__erw_function:n</code>	179, 184
<code>\erw_repeat</code>	16	<code>__erw_int_range:nnn</code>	77, 87, 100, 104
<code>\erw_seq_compose:nN</code>	5, 5, 195	<code>__erw_keyval_function:n</code>	121, 126
<code>\erw_seq_compose_c:nN</code>	5, 199	<code>__erw_map:nn</code>	350, 466
<code>\erw_seq_compose_vers:nN</code>	5, 203, 205	<code>__erw_oper_compose:NnN</code>	
<code>\erw_seq_fold:NN</code>	16, 17		154, 197, 201, 410, 420
<code>\erw_seq_from_clist:Nn</code>		<code>\g__erw_oper_fold_apply_par_tl</code>	
	5, 17, 207, 211, 213		171, 445
<code>\erw_seq_from_prop:Nnn</code>		<code>\g__erw_oper_fold_set_par_tl</code>	167, 443
	6, 17, 214, 218, 220	<code>__erw_prop_append:nn</code>	135, 139
<code>\erw_seq_put_right:Nn</code>		<code>__erw_seq_fold:NN</code>	
	6, 17, 221, 225, 227		17, 197, 201, 228, 234
<code>\erw_seq_use:Nn</code>	6, 17, 235	<code>\g__erw_seq_fold_item_tl</code>	
<code>\erw_set_map</code>	16		176, 230, 231, 232
<code>\erw_set_map_inline</code>	16	<code>__erw_seq_set_from_clist:Nn</code>	
<code>\erw_split</code>	16		177, 188, 191, 210
<code>\erw_sys_jobnametimestamp:</code>	6, 329	<code>__erw_seq_set_from_prop:Nnn</code>	
<code>\erw_sys_jobnametimestamp:nn</code>	6, 328		189, 194, 217
<code>\erw_sys_timestamp:</code>	6, 303, 338	<code>__erw_sys_date:N</code>	241
<code>\erw_sys_timestamp:nn</code>	6, 297, 334	<code>__erw_sys_date_dec:</code>	241, 283
<code>\erw_sys_timestamp_delimiter:</code>		<code>__erw_sys_date_hex:</code>	241, 284
	6, 16, 330	<code>__erw_sys_datetime_base:n</code>	261, 308
<code>\erw_tl_append_item:nn</code>	6, 16, 89, 404	<code>__erw_sys_datetime_dec:</code>	283
<code>\erw_tl_compose:nN</code>	6, 408, 415	<code>__erw_sys_datetime_dec:n</code>	261
<code>\erw_tl_compose:nn</code>	6, 412	<code>__erw_sys_datetime_hex:</code>	284
<code>\erw_tl_compose_c:nN</code>	6, 418, 425	<code>__erw_sys_datetime_hex:n</code>	261
<code>\erw_tl_compose_c:nn</code>	6, 422, 437	<code>__erw_sys_datetime_join:nn</code>	261
<code>\erw_tl_compose_vers:nN</code>	6, 428, 430	<code>__erw_sys_datetime_period:n</code>	261, 308
<code>\erw_tl_compose_vers:nn</code>	6, 432	<code>__erw_sys_jobnametimestamp:</code>	
<code>\erw_tl_fold:NN</code>			291, 299, 329
	6, 231, 410, 420, 441, 447	<code>__erw_sys_jobnametimestamp:n</code>	291
<code>\erw_tl_gset_function:N</code>	6, 16, 448	<code>__erw_sys_jobnametimestamp:nn</code>	
<code>\erw_tl_gset_function:n</code>	1, 6, 7, 452		293, 328
<code>\erw_tl_join:nn</code>	7, 16, 36, 287, 295, 301	<code>__erw_sys_jobnametimestamp_</code>	
<code>\erw_tl_join:nnn</code>	7, 37, 271	<code>prefix:</code>	285
<code>\erw_tl_join:nnnn</code>	7, 38	<code>__erw_sys_set_delim:nn</code>	310, 320
<code>\erw_tl_join:nnnnn</code>	7, 39	<code>__erw_sys_time_dec:</code>	252, 283
<code>\erw_tl_last_item:n</code>	7, 456	<code>__erw_sys_time_hex</code>	252
<code>\erw_tl_map:n</code>	7, 160, 464, 471, 476	<code>__erw_sys_time_hex:</code>	260, 284
<code>\erw_tl_map:Nn</code>	7, 468	<code>__erw_sys_timestamp:nn</code>	305, 336, 340

sys commands:		\tl_new:N	176, 342
\c_sys_day_int	247	\tl_range:nnn	16
\c_sys_hour_int	256	\tl_range_braced:nnn	65
\c_sys_jobname_str	288	\tl_reverse:n	162
\c_sys_minute_int	257	\tl_set:Nn	41, 75, 414, 424
\c_sys_month_int	246	\tl_tail:n	123, 487
\c_sys_year_int	245	token commands:	
		\token_if_cs:NTF	56
T			
tl commands:		U	
\c_empty_tl	268, 280, 395	use commands:	
\tl_count:n	461, 501, 522, 530	\use:N	308, 312, 332, 443, 445, 485
\tl_head:n	484, 522	\use_i:nn	390, 391
\tl_item:nn	358, 363, 364,	\use_i:nnn	6
	369, 370, 371, 376, 377, 378, 379, 458	\use_ii:nn	389, 391
\tl_map_function:nN	435	\usepackage	4
\tl_map_inline:nn	485		

Part IV

Implementation

1 Opening

```

1 <*package>
2 <@@=erw>
3 % \ExplSyntaxOn

```

2 basics

2.1 backend

```

4 \cs_new:Nn \__erw_cs_name:N
5 {
6   \exp_last_unbraced:Nf \use_i:nnn {\cs_split_function:N #1}
7 }

```

2.2 frontend

```

8 \cs_new:Nn \erw_cs_apply:Nn
9 {
10   #1{#2}
11 }
12 \cs_generate_variant:Nn \erw_cs_apply:Nn {No, Nf, Nx, c}
13 \cs_new:Nn \erw_cs_apply:Nnn
14 {
15   #1{#2}{#3}
16 }
17 \cs_new:Nn \erw_cs_apply:Nnnn
18 {
19   #1{#2}{#3}{#4}
20 }
21 \cs_new:Nn \erw_cs_apply:Nnnnn
22 {
23   #1{#2}{#3}{#4}{#5}
24 }
25 \cs_set:Npn \erw_cs_identity:n #1{#1}
26 \cs_new:Nn \erw_cs_set_inline:Nn
27 {
28   \cs_set:Npn #1 ##1{#2}
29 }
30 \cs_generate_variant:Nn \erw_cs_set_inline:Nn {cn}
31 \cs_new:Nn \erw_cs_gset_inline:Nn
32 {
33   \cs_gset:Npn #1 ##1{#2}
34 }
35 \cs_generate_variant:Nn \erw_cs_gset_inline:Nn {cn}
36 \cs_new:Nn \erw_tl_join:nn{#1#2}
37 \cs_new:Nn \erw_tl_join:nnn{#1#2#3}
38 \cs_new:Nn \erw_tl_join:nnnn{#1#2#3#4}
39 \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

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

4.2 frontend

```
42 \cs_new:Nn \erw_csint:nn
43 {
44   \erw_cs_apply:cn{\__erw_csint_\int_to_alph:n{#1}:n}{#2}
45 }
46 \cs_new:Nn \erw_csint_name:n {\__erw_csint_\int_to_alph:n{#1}:n}
47 \cs_new:Nn \erw_csint_names:nnn
48 {
49   \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_csint_name:n
50 }
51 \cs_new_protected:Nn \erw_csint_new:n
52 {
53   \int_incr:N \g__erw_csint_int
54   \erw_cs_set_inline:cn{\g__erw_csint_name_tl}
55   {
56     \token_if_cs:NTF
57     {#1}
58     {#1{##1}}
59     {#1}
60   }
61 }
62 \cs_new:Nn \erw_csint_names_braced:nnn
63 {
64   \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_csint_names_braced:n
65   % TODO \tl_range_braced:nnn?
66 }
67 \cs_new:Nn \erw_csint_names_braced:n {\erw_csint_name:n{#1}}
68 \cs_new:Nn \erw_csint_names_braced:
69 {
70   \erw_csint_names_braced:nnn{1}{1}{\g__erw_csint_int}
71 }
72 \cs_new_protected:Nn \erw_csint_reset:
73 {
74   \int_zero:N \g__erw_csint_int
75   \tl_set:Nn \__erw_csint_ext_tl{}%^^A TODO remove?
76 }
```

5 int

5.1 backend

```

77 \cs_set:Npn \__erw_int_range:nnn #1 #2 #3
78 {
79   \int_compare:nNnTF
80     {
81       \int_eval:n{#2+1}
82     }>{#3}
83     {
84       {#1}
85     }
86     {
87       \__erw_int_range:nnn
88       {
89         \exp_args:Nx\erw_tl_append_item:nn{#1}
90         {
91           \int_eval:n{#2+1}
92         }
93       }
94       {\int_eval:n{#2+1}}
95       {#3}
96     }
97 }

```

5.2 frontend

```

98 \cs_new:Nn \erw_int_range:nn
99 {
100   \__erw_int_range:nnn {{#1}}{#1}{#2}
101 }
102 \cs_new:Nn \erw_int_range:n
103 {
104   \__erw_int_range:nnn {{0}}{#1}
105 % ^^A Alt to:
106 % ^^A   \int_step_inline:nn {#1}{##1}
107 }

```

6 keyval

```

108 \cs_new:Nn \erw_keyval_keyonly:nn
109 {
110   \msg_error:nnn{erw}{keyval/keyonly}{#1}{#2}
111 }

```

7 msg

7.1 backend

```

112 \msg_new:nnn{__erw}{generic}{#1}
113 \msg_new:nnn{__erw}{separ}{#1-expects~1~to~3~items,~#2}
114 \msg_new:nnn{__erw}{timestamp / base}{Calling~#1,~arg~must~be~'dec|hex'}
115 \msg_new:nnn{__erw}{timestamp / period}{Calling~#1,~arg~must~be~'date|time|datetime'}

```

7.2 frontend

```

116 \msg_new:nnn{erw}{csnset}{#1~not~set}
117 \msg_new:nnn{erw}{keyval/keyonly}{passed~key~#1~val~#2~where~keyonly}
118 \msg_new:nnn{erw}{keyval/mandatval}{key~#1~has~no~matching~val}

```

8 prop

8.1 backend

8.2 frontend

```
119 \cs_new_protected:Nn \erw_prop_to_clist:Nn
120 {
121   \cs_set:Nn \__erw_keyval_function:n {,\prop_item:Nn#1{##1}}
122   \exp_args:Nf
123   \tl_tail:n
124   {
125     \keyval_parse:NNn
126     \__erw_keyval_function:n
127     \erw_keyval_keyonly:nn
128     {#2}
129   }
130 }
131 \cs_generate_variant:Nn \erw_prop_to_clist:Nn { c }
132
133 \cs_new_protected:Nn \erw_prop_put:NN
134 {
135   \cs_set:Nn \__erw_prop_append:nn
136   {
137     \prop_gput:Nnx #1 {##1}{ \prop_item:Nn #2{##1} }
138   }
139   \prop_map_function:NN #2 \__erw_prop_append:nn
140 }
141 \cs_generate_variant:Nn \erw_prop_put:NN { cc }
142 \cs_new_protected:Nn \erw_prop_put:Nnn
143 {
144   \prop_if_exist:NTF#1
145   {
146     \prop_put:Nnn #1 {#2}{#3}
147   }
148   {
149     \prop_new:N #1
150     \erw_prop_put:Nnn #1{#2}{#3}
151   }
152 }
153 \cs_generate_variant:Nn \erw_prop_put:Nnn { c }
```

9 oper

9.1 backend

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



```

162     \tl_reverse:n{#2}
163   }
164 }

```

9.2 frontend

```

165 \keys_define:nn{__erw}
166 {
167   oper/fold_set_par.tl_gset:N = \g__erw_oper_fold_set_par_tl,
168   oper/fold_set_par.value_required:n = true,
169   oper/fold_set_par.default:n = {Nf},
170   oper/fold_set_par.initial:n = {Nf},
171   oper/fold_apply_par.tl_gset:N = \g__erw_oper_fold_apply_par_tl,
172   oper/fold_apply_par.value_required:n = true,
173   oper/fold_apply_par.default:n = {Nf},
174   oper/fold_apply_par.initial:n = {Nf}
175 }

```

10 seq

10.1 backend

```

176 \tl_new:N \g__erw_seq_fold_item_tl
177 \cs_new_protected:Nn\__erw_seq_set_from_clist:Nn
178 {
179   \cs_set_protected:Nn \__erw_function:n
180   {
181     \seq_put_right:Nn #1{##1}
182   }
183   \keyval_parse:NNn
184   \__erw_function:n
185   \erw_keyval_keyonly:nn
186   {#2}
187 }
188 \cs_generate_variant:Nn \__erw_seq_set_from_clist:Nn { c }
189 \cs_new_protected:Nn\__erw_seq_set_from_prop:NNn
190 {
191   \__erw_seq_set_from_clist:Nn #1
192   {\erw_prop_to_clist:Nn #2 {#3}}
193 }
194 \cs_generate_variant:Nn \__erw_seq_set_from_prop:NNn { cc }

```

10.2 frontend

```

195 \cs_new:Nn \erw_seq_compose:nN
196 {
197   \__erw_oper_compose:NnN \__erw_seq_fold:NN {#1} #2
198 }
199 \cs_new:Nn \erw_seq_compose_c:nN
200 {
201   \__erw_oper_compose:NnN \__erw_seq_fold:cN {#1} #2
202 }
203 \cs_new:Nn \erw_seq_compose_vers:nN
204 {
205   \msg_error:nnn{__erw}{csnset}{\erw_seq_compose_vers:nN}
206 }

```

```

207 \cs_new_protected:Nn\erw_seq_from_clist:Nn
208 {
209   \seq_if_exist:NTF#1
210   {\__erw_seq_set_from_clist:Nn#1{#2}}
211   {\seq_new:N#1\erw_seq_from_clist:Nn#1{#2}}
212 }
213 \cs_generate_variant:Nn \erw_seq_from_clist:Nn { c }
214 \cs_new_protected:Nn\erw_seq_from_prop:NNn
215 {
216   \seq_if_exist:NTF#1
217   {\__erw_seq_set_from_prop:NNn#1#2{#3}}
218   {\seq_new:N#1\erw_seq_from_prop:NNn#1#2{#3}}
219 }
220 \cs_generate_variant:Nn \erw_seq_from_prop:NNn { cc }
221 \cs_new_protected:Nn\erw_seq_put_right:Nn
222 {
223   \seq_if_exist:NTF#1
224   {\seq_put_right:Nn#1{#2}}
225   {\seq_new:N#1\erw_seq_put_right:Nn #1{#2}}
226 }
227 \cs_generate_variant:Nn\erw_seq_put_right:Nn { c }
228 \cs_new:Nn \__erw_seq_fold:NN
229 {
230   \seq_get_right:NN #2 \g__erw_seq_fold_item_tl
231   \erw_tl_fold:NN #1 \g__erw_seq_fold_item_tl
232   \seq_put_right:No #2 {\g__erw_seq_fold_item_tl}
233 }
234 \cs_generate_variant:Nn \__erw_seq_fold:NN { cN }
235 \cs_new:Nn \erw_seq_use:Nn
236 {
237   \exp_last_unbraced:NNf
238   \seq_use:Nnnn #1
239   \erw_tl_separators:n{#2}
240 }

```

11 sys

11.1 backend

```

\__erw_sys_date:N
\__erw_sys_date_dec: 241 \cs_new:Nn \__erw_sys_date_dec:
\__erw_sys_date_hex: 242 {
243   \int_eval:n
244   {
245     \c_sys_year_int * 10000
246     +\c_sys_month_int * 100
247     +\c_sys_day_int * 1
248   }
249 }
250 \cs_new:Nn \__erw_sys_date:N{\int_to_hex:n{\__erw_sys_date_dec:}}
251 \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
252 \cs_new:Nn \__erw_sys_time_dec:
253 {
254   \int_eval:n
255   {
256     \c_sys_hour_int * 100
257     +\c_sys_minute_int * 1
258   }
259 }
260 \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
261 \cs_new:Nn\__erw_sys_datetime_base:n
262 {
263   \int_case:nnTF{#1}
264   {
265     {10}{dec}
266     {16}{hex}
267   }
268   {\c_empty_tl}
269   {\msg_error:nnn{\__erw}{timestamp / base}{\__erw_sys_datetime_base:n{#1}}}
270 }
271 \cs_new:Nn\__erw_sys_datetime_join:nn{\erw_tl_join:nnn{#1}{\g__erw_sys_timestamp_delim_str}{#2}}
272 \cs_new:Nn\__erw_sys_datetime_period:n
273 {
274   \str_case:nnTF{#1}
275   {
276     {date}{date}
277     {time}{time}
278     {datetime}{datetime}
279   }
280   {\c_empty_tl}
281   {\msg_error:nnn{\__erw}{timestamp / period}{\__erw_sys_datetime_period:n{#1}}}
282 }
283 \cs_new:Nn\__erw_sys_datetime_dec: {\__erw_sys_datetime_join:nn{\__erw_sys_date_dec:}{\__erw_sys_time_dec:}}
284 \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:
285 \cs_new:Nn\__erw_sys_jobnametimestamp_prefix:
286 {
287   \erw_tl_join:nn
288   {\c_sys_jobname_str}
289   {\g__erw_sys_timestamp_delim_str}
290 }
291 % \begin{macro}{\__erw_sys_jobnametimestamp:n, \__erw_sys_jobnametimestamp:}
292 %   \begin{macrocode}
293 \cs_new:Nn\__erw_sys_jobnametimestamp:nn
294 {
295   \erw_tl_join:nn
296   {\__erw_sys_jobnametimestamp_prefix:}

```

```

297   {\erw_sys_timestamp:nn{#1}{#2}}
298 }
299 \cs_new:Nn\__erw_sys_jobnametimestamp:
300 {
301   \erw_tl_join:nn
302   {\__erw_sys_jobnametimestamp_prefix:}
303   {\erw_sys_timestamp:}
304 }

```

(End definition for __erw_sys_jobnametimestamp_prefix:.)

__erw_sys_timestamp:nn

```

305 \cs_new:Nn\__erw_sys_timestamp:nn
306 {
307   \exp_args:No
308   \use:c{\__erw_sys\___erw_sys_datetime_period:n{#1}\__erw_sys_datetime_base:n{#2}:}
309 }
310 \cs_new_protected:Nn \__erw_sys_set_delim:nn
311 {
312   \use:c{tl_gset:N#1}
313   \g__erw_sys_timestamp_delim_str{#2}
314 }

```

(End definition for __erw_sys_timestamp:nn.)

```

315 \keys_define:nn{\__erw}
316 {
317   sys / timestamp_delim .code:n =
318   {
319     \exp_last_unbraced:No
320     \__erw_sys_set_delim:nn{n}{#1}
321   },
322   sys / timestamp_delim .value_required:n = true,
323   sys / timestamp_delim .default:n = {-},
324   sys / timestamp_delim .initial:n = {-}
325 }
326 % \subsection{frontend}
327 % \begin{macrocode}
328 \cs_new:Nn\erw_sys_jobnametimestamp:nn{\__erw_sys_jobnametimestamp:nn{#1}{#2}}
329 \cs_new:Nn\erw_sys_jobnametimestamp:{\__erw_sys_jobnametimestamp:}
330 \cs_new:Nn\erw_sys_timestamp_delimiter:
331 {
332   \use:N \g__erw_sys_timestamp_delim_str
333 }
334 \cs_new:Nn\erw_sys_timestamp:nn
335 {
336   \__erw_sys_timestamp:nn{#1}{#2}
337 }
338 \cs_new:Nn\erw_sys_timestamp:
339 {
340   \__erw_sys_timestamp:nn{datetime}{16}
341 }

```

12 tl

12.1 backend

```
342 \tl_new:N \g__erw_tl_compose_tl
```

```
\g__erw_tl_function:n
```

```
343 \cs_new_protected:Nn \g__erw_tl_function:n
344 {
345   \msg_error:nnn
346   {erw}
347   {csnset}
348   {\g__erw_tl_function:n}
349 }
```

(End definition for \g__erw_tl_function:n.)

```
\__erw_map:nn
```

```
350 \cs_set_protected:Nn \__erw_map:nn
351 {
352   \quark_if_recursion_tail_stop:n{#1}
353   \g__erw_tl_function:n{#1} \__erw_map:nn{#2}
354 }
```

(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
```

```
355 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnn
356 {
357   \erw_cs_apply:Nn #1
358   {\exp_args:Nf\tl_item:nn {#3} {#2} }
359 }
360 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnn
361 {
362   \erw_cs_apply:Nnn #1
363   {\exp_args:Nf\tl_item:nn {#3} {#2} }
364   {\exp_args:Nf\tl_item:nn {#4} {#2} }
365 }
366 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnnn
367 {
368   \erw_cs_apply:Nnnn #1
369   {\exp_args:Nf\tl_item:nn {#3} {#2} }
370   {\exp_args:Nf\tl_item:nn {#4} {#2} }
371   {\exp_args:Nf\tl_item:nn {#5} {#2} }
372 }
373 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnnnn
374 {
375   \erw_cs_apply:Nnnnn #1
376   {\exp_args:Nf\tl_item:nn {#3} {#2} }
377   {\exp_args:Nf\tl_item:nn {#4} {#2} }
378   {\exp_args:Nf\tl_item:nn {#5} {#2} }
379   {\exp_args:Nf\tl_item:nn {#6} {#2} }
380 }
```

(End definition for __erw_tl_map_thread_at:Nnn and others.)

```

\__erw_tl_separators:nn #1: < int >
#2: < items >

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

(End definition for \__erw_tl_separators:nn.)

```

12.2 frontend

```

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

```

```

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

```

```

482 \cs_new:Nn \erw_tl_split:nnn
483 {
484   \tl_head:n{#1}
485   \use:c{exp_args:#3} \tl_map_inline:nn
486   {
487     \tl_tail:n
488     {
489       #1
490     }
491   }{#2##1}
492 }
493 \cs_new:Nn \erw_tl_split:nn
494 {
495   \erw_tl_split:nnn{#1}{#2}{Nf}
496 }
497 \cs_new_protected:Nn \erw_tl_map_thread_at:Nnn
498 {
499   \exp_args:Nf\int_case:nnTF
500   {
501     \tl_count:n{#3}
502   }
503   {
504     {1}{ \__erw_tl_map_thread_at:Nnn #1{#2}#3 }
505     {2}{ \__erw_tl_map_thread_at:Nnnn #1{#2}#3 }
506     {3}{ \__erw_tl_map_thread_at:Nnnnn #1{#2}#3 }
507     {4}{ \__erw_tl_map_thread_at:Nnnnnn #1{#2}#3 }
508   }
509   {
510     % Do nothing
511   }
512   {
513     \msg_error:nnn{__erw}
514     {generic}
515     {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
516   }
517 }
518 \cs_new_protected:Nn \erw_tl_map_thread:Nn
519 {
520   \int_step_inline:nn
521   {
522     \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
523   }
524   {
525     \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
526   }
527 }
528 \cs_new:Nn \erw_tl_separators:n
529 {
530   \__erw_tl_separators:en{ \tl_count:n{#1} }{#1}
531 }

```

13 option

```

532 \cs_new_protected:Nn\erw_option:n

```



```
533 {  
534   \keys_set:nn{__erw}{#1}  
535 }
```

14 Closing

```
536 \ExplSyntaxOff  
537 </package>
```