

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

Erwann Rogard<sup>†</sup>

Released 2020/05/01

## Abstract

Utilities based on `expl3`[1].

## Résumé

Utilitaires de type `expl3`[1].

## Contents

<b>I</b>	<b>Usage</b>	<b>4</b>
1	Loading the package	4
2	basics	4
3	csint	4
4	int	5
5	prop	5
6	seq	5
7	sys	6
8	tl	6
10	option	7
<b>II</b>	<b>Listing</b>	<b>8</b>
1	basics	8
1.	.....	8

---

<sup>\*</sup>This file describes version v1.9, last revised 2020/05/01.

<sup>†</sup>firstname dot lastname AusTria gmail dot com

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

<b>IV</b>	<b>Implementation</b>	<b>20</b>
<b>1</b>	<b>Opening</b>	<b>20</b>
<b>2</b>	<b>basics</b>	<b>20</b>
	2.1 backend . . . . .	20
	2.2 frontend . . . . .	20
<b>3</b>	<b>clist</b>	<b>21</b>
	3.1 backend . . . . .	21
	3.2 frontend . . . . .	21
<b>4</b>	<b>csint</b>	<b>21</b>
	4.1 backend . . . . .	21
	4.2 frontend . . . . .	21
<b>5</b>	<b>int</b>	<b>21</b>
	5.1 backend . . . . .	21
	5.2 frontend . . . . .	22
<b>6</b>	<b>keyval</b>	<b>22</b>
<b>7</b>	<b>msg</b>	<b>22</b>
	7.1 backend . . . . .	22
<b>8</b>	<b>prop</b>	<b>22</b>
	8.1 backend . . . . .	22
	8.2 frontend . . . . .	23
<b>9</b>	<b>oper</b>	<b>23</b>
	9.1 backend . . . . .	23
	9.2 frontend . . . . .	23
<b>10</b>	<b>seq</b>	<b>24</b>
	10.1 backend . . . . .	24
	10.2 frontend . . . . .	24
<b>11</b>	<b>sys</b>	<b>25</b>
	11.1 backend . . . . .	25
<b>12</b>	<b>tl</b>	<b>27</b>
	12.1 backend . . . . .	27
	12.2 frontend . . . . .	28
<b>13</b>	<b>option</b>	<b>31</b>

## Part I

# Usage

---

<code>\usepackage</code>	<code>\usepackage{erw-l3}</code>
--------------------------	----------------------------------

---

### Requirement

1. `erw-l3.sty` and its dependencies are in the path of the L<sup>A</sup>T<sub>E</sub>X engine. See [Part III, section 3](#).
2. Goes in the *preamble*

## 2 basics

---

<code>\erw_cs_apply:Nn</code>	<code>\erw_cs_apply:Nn {&lt;cs&gt;} {&lt;token list<sub>1</sub>&gt;}</code>
<code>\erw_cs_apply:(No Nf Nx cn)</code>	
<code>\erw_cs_apply:Nnn</code>	
<code>\erw_cs_apply:Nnnn</code>	
<code>\erw_cs_apply:Nnnnn</code>	

---



---

<code>\erw_cs_identity:n</code>	<code>\erw_cs_identity:n{&lt;arg&gt;}</code>
---------------------------------	--

---



---

<code>\erw_cs_set_inline:Nn</code>	<code>\erw_cs_set_inline:Nn{&lt;cs&gt;} {&lt;code&gt;}</code>
<code>\erw_cs_set_inline:cn</code>	

---

## 3 csint

---

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

---



---

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

---



---

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

---



---

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

---

---

<code>\erw_csint_new:n</code>	<code>\erw_csint_new:n{⟨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 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:NN⟨prop⟩⟨key⟩⟨val⟩</code>
--------------------------------	---

---



---

<code>\erw_prop_to_clist:Nn</code>	<code>\erw_prop_put:NN⟨prop⟩{⟨key₁⟩,...}</code>
------------------------------------	---

---

## 6 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{⟨{⟨csname₁⟩}...⟩}⟨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:NNn</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>
------------------------------------	---

---

## 7 sys

---

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

---



---

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

---



---

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

---

## 8 tl

All functions that modify a *token list* 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{csname_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>	

---



---

<code>\erw_tl_join:nn</code>	<code>\erw_tl_join:nn{token list<sub>1</sub>}{token list<sub>2</sub>}</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>
----------------------------------	--

---

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

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

<u><code>\erw_tl_map_inline:nn</code></u>	<code>\erw_tl_map_inline:nn{&lt;code&gt;}{&lt;items&gt;}</code>
---	---

<u><code>\erw_tl_map_thread:Nn</code></u>	<code>\erw_tl_math_thread:Nn{&lt;cs&gt;}{&lt;items&gt;}</code>
---	--

<u><code>\erw_tl_map_thread_at:Nnn</code></u>	<code>\erw_tl_math_thread_at:Nnn{&lt;integer&gt;}{&lt;token list&gt;}</code>
---	--

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

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

## 10 option

<u><code>\erw_option:n</code></u>	<code>\erw_option:n{&lt;keyval list&gt;}</code>
-----------------------------------	---

oper / fold\_set\_par  
oper / fold\_apply\_par  
sys / timestamp\_delim

## 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_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)]}

### 3 int

Listing 3.

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

---

2345  
12345



## 4 prop

Listing 4. Global

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

---

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

## 5 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_clist:Nn \foo_seq{ A, B, C }
\seq_use:Nn\foo_seq{,}
\ExplSyntaxOff

```

---

A,B,C

#### Listing 10.

```

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

```

---

a,b,c

## 6 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
```

```
20200509-245
20200509-245
1343c3d%f5
1343c3d%f5
erw-l3%1343c3d%f5
```

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:20200509245
D:20200509024536-04'00'
Hello, world!
```

## 7 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<sup>A</sup>T<sub>E</sub>Xcommunity[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<sup>A</sup>T<sub>E</sub>Xengine

### 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<sup>A</sup>T<sub>E</sub>X3 Project Team *The L<sup>A</sup>T<sub>E</sub>X3 interfaces*, 2019, <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf>
- [2] <https://tex.stackexchange.com/users/112708/erwann?tab=questions>

# Change History

v1.0	General: Initial version . . . . .	15	Split Section Preliminaries into Conventions and Requirement. . .	15
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> . . . . .
	<code>disambig/backend</code> : changes to the key, added <code>\ProcessPackageKeysOption</code> ; . . .	15	v1.7	General: Add: <b>Closing</b> . . . . .
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ; . . .	15		Add: <b>sys</b> . . . . .
v1.2	General: . . . . .	15		Move: <code>\erw_fold_apply_par:n</code> . .
	<code>\erw_compose</code> reversed order in which the functions are composed, such that it now conforms to the mathematical cInvention ( $g \circ f$ means $f$ comes before $g$ ) . . . . .	15		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. . . . .	15		Rearrange: structure of implementation, e.g. <b>section 9</b> . .
	Add: <code>\erw_items_to</code> . . . . .	15		Remove: document level functions, <code>\numbrdcsnew</code> , <code>\numbrdcs</code> . . . . .
	Add: <code>\erw_last_item</code> . . . . .	15		Replace: listing's implem with that of <code>tocloft</code> . . . . .
	Add: <code>\erw_repeat</code> . . . . .	15		Replace: vers. numb. from 3 to 2 digits . . . . .
	Add: <code>\erw_split</code> . . . . .	15	v1.8	General: Add: function for all frontend functions. . . . .
	Add: <code>\map_thread</code> . . . . .	15		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; . . . . .	15		Remove: <code>\erw_is_matrix:n</code> (predicate must be expandable) . .
	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> . . . . .
v1.3	General: Replace: versioning, should have been 0.1.2 . . . . .	15		Replace: <code>@@_map:n</code> by <code>@@_oper_function:n</code> . . . . .
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 . . . . .
	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> .
	Add: <code>\erw_merge</code> . . . . .	15		Add: <code>\erw_tl_join:nn</code> and variants
	Add: <code>\erw_set_map_inline</code> . . . .	15		Rename: <code>\erw_append_arg:nn</code> to <code>\erw_tl_append_item:nn</code> . . . . .
	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) . . . . .
v1.5	General: Modify: source repository . .	15	v2.0	General: Add:
	Rearrange: frontend/backend sections . . . . .	15		<code>\erw_jobnametimestamp:nn</code> and variants . . . . .
	Remove: <code>disambig</code> . . . . .	15		



Remove: <code>\merge:nn</code> (redundant with <code>\erw_join:nn</code> ) . . . . .	15
Rename: v0.0 to v1.0, etc. . . . .	15
v2.1	
General: Add:	
<code>\erw_prop_to_clist:Nn</code> ,	
<code>\erw_prop_put:NN</code> , and	
<code>\erw_prop_put:Nnn</code> . . . . .	15
Add: <code>\erw_seq_from_clist:Nn</code> ,	
<code>\erw_seq_from_prop:NNn</code> , and	
<code>\erw_seq_put_right:Nn</code> . . . . .	15
Move: all functions under <a href="#">section 9</a> to <a href="#">section 12</a> or <a href="#">section 10</a> , except	
<code>\@@_oper_compose:NnN</code> . . . . .	15
Replace: <code>\erw_seq_fold:NN</code> by	
<code>\_erw_seq_fold:NN</code> . . . . .	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	
<b>@@ commands:</b>	
<code>\@@_map:n</code> . . . . .	16
<code>\@@_oper_compose:NnN</code> . . . . .	17
<code>\@@_oper_function:n</code> . . . . .	16
<b>B</b>	
<code>\begin</code> . . . . .	284, 285, 320
<b>C</b>	
<b>cs commands:</b>	
<code>\cs_generate_variant:Nn</code> . . . . .	11, 29, 34, 128, 138, 150, 185, 191, 210, 217, 224, 231, 417
<code>\cs_gset:Npn</code> . . . . .	32
<code>\cs_new:Nn</code> . . . . .	3, 7, 12, 16, 20, 25, 30, 35, 36, 37, 38, 41, 45, 46, 61, 66, 67, 97, 101, 109, 151, 192, 196, 200, 225, 234, 243, 244, 245, 253, 254, 264, 265, 276, 277, 278, 286, 292, 298, 321, 322, 323, 327, 331, 374, 378, 382, 388, 392, 398, 402, 411, 418, 422, 426, 448, 452, 463
<code>\cs_new_protected:Nn</code> . . . . .	50, 71, 116, 130, 139, 174, 186, 204, 211, 218, 303, 336, 434, 438, 443, 467, 488, 498
<code>\cs_set:Nn</code> . . . . .	118, 132
<code>\cs_set:Npn</code> . . . . .	24, 27, 76
<code>\cs_set_eq:NN</code> . . . . .	440
<code>\cs_set_protected:Nn</code> . . . . .	176, 343, 348, 353, 359, 366
<code>\cs_split_function:N</code> . . . . .	5
<b>D</b>	
<code>\disambignewcmd</code> . . . . .	16
<b>E</b>	
<b>erw commands:</b>	
<code>\erw_accum</code> . . . . .	16
<code>\erw_append_arg:nn</code> . . . . .	16
<code>\erw_compose</code> . . . . .	16
<code>\erw_cs_apply:Nn</code> . . . . .	4, 7, 11, 43, 350
<code>\erw_cs_apply:Nnn</code> . . . . .	4, 12, 355
<code>\erw_cs_apply:Nnnn</code> . . . . .	4, 16, 361
<code>\erw_cs_apply:Nnnnn</code> . . . . .	4, 20, 368
<code>\erw_cs_gset_eq:NN</code> . . . . .	420
<code>\erw_cs_gset_inline:Nn</code> . . . . .	30, 34, 424
<code>\erw_cs_identity:n</code> . . . . .	4, 16, 24
<code>\erw_cs_set_eq:NN</code> . . . . .	16
<code>\erw_cs_set_inline:Nn</code> . . . . .	4, 25, 29, 53, 153, 445
<code>\erw_csint:nn</code> . . . . .	4, 41
<code>\erw_csint_name:n</code> . . . . .	4, 40, 45, 48, 66
<code>\erw_csint_names:nnn</code> . . . . .	4, 46
<code>\erw_csint_names_braced:</code> . . . . .	4, 67, 408
<code>\erw_csint_names_braced:n</code> . . . . .	4, 63, 66
<code>\erw_csint_names_braced:nnn</code> . . . . .	4, 61, 69
<code>\erw_csint_new:n</code> . . . . .	5, 50, 405
<code>\erw_csint_reset:</code> . . . . .	5, 71, 404
<code>\erw_fold_apply_par:n</code> . . . . .	16
<code>\erw_fold_set_par:n</code> . . . . .	16
<code>\erw_identity:n</code> . . . . .	16
<code>\erw_int_range</code> . . . . .	16
<code>\erw_int_range:n</code> . . . . .	5, 101
<code>\erw_int_range:nn</code> . . . . .	5, 97
<code>\erw_is_matrix</code> . . . . .	16
<code>\erw_is_matrix:n</code> . . . . .	16
<code>\erw_items_to</code> . . . . .	16
<code>\erw_jobnametimestamp:nn</code> . . . . .	16
<code>\erw_join:nn</code> . . . . .	17
<code>\erw_keyval_keyonly:nn</code> . . . . .	109, 124, 182
<code>\erw_last_item</code> . . . . .	16
<code>\erw_merge</code> . . . . .	16
<code>\erw_oper_fold_seq:NN</code> . . . . .	16
<code>\erw_oper_gset_function:N</code> . . . . .	16
<code>\erw_option:n</code> . . . . .	7, 498
<code>\erw_prop_put:NN</code> . . . . .	5, 5, 5, 17, 130, 138

\erw_prop_put:Nnn .	5, 17, 139, 147, 150	\g__erw_csint_name_tl . . . . .	40, 53
\erw_prop_to_clist:Nn . . . . .		\__erw_function:n . . . . .	176, 181
. . . . .	5, 17, 116, 128, 189	\__erw_int_range:nnn .	76, 86, 99, 103
\erw_repeat . . . . .	16	\__erw_keyval_function:n . . .	118, 123
\erw_seq_compose:nN . . . . .	5, 5, 192	\__erw_map:nn . . . . .	343, 436
\erw_seq_compose_c:nN . . . . .	5, 196	\__erw_oper_compose:NnN . . . . .	
\erw_seq_compose_vers:nN .	5, 200, 202	. . . . .	151, 194, 198, 380, 390
\erw_seq_fold:NN . . . . .	16, 17	\g__erw_oper_fold_apply_par_tl . .	
\erw_seq_from_clist:Nn . . . . .		. . . . .	168, 415
. . . . .	5, 17, 204, 208, 210	\g__erw_oper_fold_set_par_tl	164, 413
\erw_seq_from_prop:NNn . . . . .		\__erw_prop_append:nn . . . . .	132, 136
. . . . .	5, 17, 211, 215, 217	\__erw_seq_fold:NN . . . . .	
\erw_seq_put_right:Nn . . . . .		. . . . .	17, 194, 198, 225, 231
. . . . .	5, 17, 218, 222, 224	\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, 322	\__erw_seq_set_from_prop:NNn . . .	
\erw_sys_jobnametimestamp:nn	6, 321	. . . . .	186, 191, 214
\erw_sys_timestamp: . . . . .	6, 296, 331	\__erw_sys_date:N . . . . .	234
\erw_sys_timestamp:nn . . .	6, 290, 327	\__erw_sys_date_dec: . . . . .	234, 276
\erw_sys_timestamp_delimiter: . . .		\__erw_sys_date_hex: . . . . .	234, 277
. . . . .	6, 16, 323	\__erw_sys_datetime_base:n .	254, 301
\erw_tl_append_item:nn .	6, 16, 88, 374	\__erw_sys_datetime_dec: . . . . .	276
\erw_tl_compose:nN . . . . .	6, 378, 385	\__erw_sys_datetime_dec:n . . . . .	254
\erw_tl_compose:nn . . . . .	6, 382	\__erw_sys_datetime_hex: . . . . .	277
\erw_tl_compose_c:nN . . . . .	6, 388, 395	\__erw_sys_datetime_hex:n . . . . .	254
\erw_tl_compose_c:nn . . . . .	6, 392, 407	\__erw_sys_datetime_join:nn . . .	254
\erw_tl_compose_vers:nN . .	6, 398, 400	\__erw_sys_datetime_period:n	254, 301
\erw_tl_compose_vers:nn . . . . .	6, 402	\__erw_sys_jobnametimestamp: . . .	
\erw_tl_fold:NN . . . . .		. . . . .	284, 292, 322
. . . . .	6, 228, 380, 390, 411, 417	\__erw_sys_jobnametimestamp:n . .	284
\erw_tl_gset_function:N . . .	6, 16, 418	\__erw_sys_jobnametimestamp:nn . .	
\erw_tl_gset_function:n . . .	6, 7, 422	. . . . .	286, 321
\erw_tl_join:nn	6, 16, 35, 280, 288, 294	\__erw_sys_jobnametimestamp_-	
\erw_tl_join:nnn . . . . .	6, 36, 264	prefix: . . . . .	278
\erw_tl_join:nnnn . . . . .	6, 37	\__erw_sys_set_delim:nn . . . . .	303, 313
\erw_tl_join:nnnnn . . . . .	6, 38	\__erw_sys_time_dec: . . . . .	245, 276
\erw_tl_last_item:n . . . . .	6, 426	\__erw_sys_time_hex . . . . .	245
\erw_tl_last_time:n . . . . .	6	\__erw_sys_time_hex: . . . . .	253, 277
\erw_tl_map:n . . .	7, 157, 434, 441, 446	\__erw_sys_timestamp:nn	298, 329, 333
\erw_tl_map:Nn . . . . .	7, 438	\g__erw_sys_timestamp_delim_str .	
\erw_tl_map_inline:nn . . . . .	7, 443	. . . . .	264, 282, 306, 325
\erw_tl_map_thread:Nn . . . . .	7, 16, 488	\g__erw_tl_compose_tl . . . . .	
\erw_tl_map_thread_at:Nnn	7, 467, 495	. . . . .	335, 384, 385, 386, 394, 395, 396
\erw_tl_math_thread:Nn . . . . .	7	\__erw_tl_map_thread_at:Nnn	348, 474
\erw_tl_math_thread_at:Nnn . . . . .	7	\__erw_tl_map_thread_at:Nnnn	348, 475
\erw_tl_repeat:nn . . . . .	7, 448	\__erw_tl_map_thread_at:Nnnnn . . .	
\erw_tl_split:nn . . . . .	7, 463	. . . . .	348, 476
\erw_tl_split:nnn . . . . .	7, 452, 465	\__erw_tl_map_thread_at:Nnnnnn . .	
erw internal commands:		. . . . .	348, 477
\__erw_cs_name:N . . . . .	3		
\__erw_csint_ext_tl . . . . .	74		
\g__erw_csint_int . .	39, 40, 52, 69, 73		

exp commands:	
\exp_args:Nf . . . . .	119, 157, 351, 356, 357, 362, 363, 364, 369, 370, 371, 372, 469, 492
\exp_args:No . . . . .	300
\exp_args:Nof . . . . .	428
\exp_args:Nx . . . . .	88
\exp_last_unbraced:Nf . . . . .	5
\exp_last_unbraced:No . . . . .	312
\exp_last_unbraced:Nx . . . . .	406
\ExplSyntaxOff . . . . .	502
\ExplSyntaxOn . . . . .	2
<b>G</b>	
g internal commands:	
\g_erw_tl_function:n . . . . .	153, 336, 346, 420, 424, 440, 445
<b>I</b>	
int commands:	
\int_case:nnTF . . . . .	256, 469
\int_compare:nNnTF . . . . .	78
\int_eval:n . . . . .	80, 90, 93, 236, 247
\int_incr:N . . . . .	52
\int_new:N . . . . .	39
\int_step_function:nnnN . . . . .	48, 63
\int_step_inline:nn . . . . .	105, 490
\int_step_inline:nnnn . . . . .	450
\int_to_alph:n . . . . .	43, 45
\int_to_hex:n . . . . .	243, 244, 253
\int_zero:N . . . . .	73
<b>K</b>	
keys commands:	
\keys_define . . . . .	16
\keys_define:nn . . . . .	162, 308
\keys_set:nn . . . . .	500
keyval commands:	
\keyval_parse:NNn . . . . .	122, 180
<b>M</b>	
map commands:	
\map_thread . . . . .	16
merge commands:	
\merge:nn . . . . .	17
msg commands:	
\msg_error:nnn . . . . .	111, 202, 262, 274, 338, 400, 483
\msg_new:nnn . . . . .	107, 108, 113, 114, 115, 232, 233
<b>N</b>	
\newnumbrdcs . . . . .	16
\numbrdcs . . . . .	16
\numbrdcsnew . . . . .	16
<b>O</b>	
oper / fold_apply_par (option) . . . . .	7
oper / fold_set_par (option) . . . . .	7
options:	
oper / fold_apply_par . . . . .	7
oper / fold_set_par . . . . .	7
sys / timestamp_delim . . . . .	7
<b>P</b>	
\ProcessPackageKeysOption . . . . .	16
prop commands:	
\prop_gput:Nnn . . . . .	134
\prop_if_exist:NTF . . . . .	141
\prop_item:Nn . . . . .	118, 134
\prop_map_function:NN . . . . .	136
\prop_new:N . . . . .	146
\prop_put:Nnn . . . . .	143
<b>Q</b>	
quark commands:	
\quark_if_recursion_tail_stop:n . . . . .	345
\q_recursion_stop . . . . .	436
\q_recursion_tail . . . . .	436
<b>S</b>	
seq commands:	
\seq_get_right:NN . . . . .	227
\seq_if_exist:NTF . . . . .	206, 213, 220
\seq_new:N . . . . .	208, 215, 222
\seq_put_right:Nn . . . . .	178, 221, 229
str commands:	
\str_case:nnTF . . . . .	267
\subsection . . . . .	319
sys / timestamp_delim (option) . . . . .	7
sys commands:	
\c_sys_day_int . . . . .	240
\c_sys_hour_int . . . . .	249
\c_sys_jobname_str . . . . .	281
\c_sys_minute_int . . . . .	250
\c_sys_month_int . . . . .	239
\c_sys_year_int . . . . .	238
<b>T</b>	
tl commands:	
\c_empty_tl . . . . .	261, 273
\tl_count:n . . . . .	431, 471, 492
\tl_head:n . . . . .	454, 492
\tl_item:nn . . . . .	351, 356, 357, 362, 363, 364, 369, 370, 371, 372, 428
\tl_map_function:nN . . . . .	405
\tl_map_inline:nn . . . . .	455
\tl_new:N . . . . .	173, 335
\tl_range:nnn . . . . .	16
\tl_range_braced:nnn . . . . .	64
\tl_reverse:n . . . . .	159



## 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}

```

## 11 sys

### 11.1 backend

```

232 \msg_new:nnn{__erw}{timestamp / base}{Calling~#1,~arg~must~be~'dec|hex'}
233 \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:
234 \cs_new:Nn \__erw_sys_date_dec:
235 {
236   \int_eval:n
237   {
238     \c_sys_year_int * 10000
239     +\c_sys_month_int * 100
240     +\c_sys_day_int * 1
241   }
242 }
243 \cs_new:Nn \__erw_sys_date:N{\int_to_hex:n{\__erw_sys_date_dec:}}
244 \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:
245 \cs_new:Nn \__erw_sys_time_dec:
246 {
247   \int_eval:n
248   {
249     \c_sys_hour_int * 100
250     +\c_sys_minute_int * 1
251   }
252 }

```

```
253 \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
```

```
254 \cs_new:Nn\__erw_sys_datetime_base:n
```

```
\__erw_sys_datetime_join:nn
```

```
255 {
```

```
\__erw_sys_datetime_hex:n
```

```
256 \int_case:nnTF{#1}
```

```
\__erw_sys_datetime_period:n
```

```
257 {
```

```
258 {10}{dec}
```

```
259 {16}{hex}
```

```
260 }
```

```
261 {\c_empty_tl}
```

```
262 {\msg_error:nnn{\__erw}{timestamp / base}{\__erw_sys_datetime_base:n{#1}}}
```

```
263 }
```

```
264 \cs_new:Nn\__erw_sys_datetime_join:nn{\erw_tl_join:nnn{#1}{\g__erw_sys_timestamp_delim_str}{#2}}
```

```
265 \cs_new:Nn\__erw_sys_datetime_period:n
```

```
266 {
```

```
267 \str_case:nnTF{#1}
```

```
268 {
```

```
269 {date}{date}
```

```
270 {time}{time}
```

```
271 {datetime}{datetime}
```

```
272 }
```

```
273 {\c_empty_tl}
```

```
274 {\msg_error:nnn{\__erw}{timestamp / period}{\__erw_sys_datetime_period:n{#1}}}
```

```
275 }
```

```
276 \cs_new:Nn\__erw_sys_datetime_dec: {\__erw_sys_datetime_join:nn{\__erw_sys_date_dec:}{\__erw_sys_date_hex:}}
```

```
277 \cs_new:Nn\__erw_sys_datetime_hex: {\__erw_sys_datetime_join:nn{\__erw_sys_date_hex:}{\__erw_sys_date_dec:}}
```

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

```
\__erw_sys_jobnametimestamp_prefix:
```

```
278 \cs_new:Nn\__erw_sys_jobnametimestamp_prefix:
```

```
279 {
```

```
280 \erw_tl_join:nn
```

```
281 {\c_sys_jobname_str}
```

```
282 {\g__erw_sys_timestamp_delim_str}
```

```
283 }
```

```
284 % \begin{macro}{\__erw_sys_jobnametimestamp:n, \__erw_sys_jobnametimestamp:}
```

```
285 % \begin{macrocode}
```

```
286 \cs_new:Nn\__erw_sys_jobnametimestamp:nn
```

```
287 {
```

```
288 \erw_tl_join:nn
```

```
289 {\__erw_sys_jobnametimestamp_prefix:}
```

```
290 {\erw_sys_timestamp:nn{#1}{#2}}
```

```
291 }
```

```
292 \cs_new:Nn\__erw_sys_jobnametimestamp:
```

```
293 {
```

```
294 \erw_tl_join:nn
```

```
295 {\__erw_sys_jobnametimestamp_prefix:}
```

```
296 {\erw_sys_timestamp:}
```

```
297 }
```

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

\\_\_erw\_sys\_timestamp:nn

```
298 \cs_new:Nn\__erw_sys_timestamp:nn
299 {
300   \exp_args:No
301   \use:c{__erw_sys\___erw_sys_datetime_period:n{#1}\__erw_sys_datetime_base:n{#2}:}
302 }
303 \cs_new_protected:Nn \__erw_sys_set_delim:nn
304 {
305   \use:c{tl_gset:N#1}
306   \g__erw_sys_timestamp_delim_str{#2}
307 }

(End definition for \__erw_sys_timestamp:nn.)

308 \keys_define:nn{__erw}
309 {
310   sys / timestamp_delim .code:n =
311   {
312     \exp_last_unbraced:No
313     \__erw_sys_set_delim:nn{n}{#1}
314   },
315   sys / timestamp_delim .value_required:n = true,
316   sys / timestamp_delim .default:n = {-},
317   sys / timestamp_delim .initial:n = {-}
318 }
319 % \subsection{frontend}
320 % \begin{macrocode}
321 \cs_new:Nn\erw_sys_jobnametimestamp:nn{\__erw_sys_jobnametimestamp:nn{#1}{#2}}
322 \cs_new:Nn\erw_sys_jobnametimestamp:{\__erw_sys_jobnametimestamp:}
323 \cs_new:Nn\erw_sys_timestamp_delimiter:
324 {
325   \use:N \g__erw_sys_timestamp_delim_str
326 }
327 \cs_new:Nn\erw_sys_timestamp:nn
328 {
329   \__erw_sys_timestamp:nn{#1}{#2}
330 }
331 \cs_new:Nn\erw_sys_timestamp:
332 {
333   \__erw_sys_timestamp:nn{datetime}{16}
334 }
```

## 12 tl

### 12.1 backend

```
335 \tl_new:N \g__erw_tl_compose_tl
```

\g\_\_erw\_tl\_function:n

```
336 \cs_new_protected:Nn \g__erw_tl_function:n
337 {
338   \msg_error:nnn
339   {erw}
340   {notset}
```

```

341   {\g__erw_tl_function:n}
342 }

(End definition for \g__erw_tl_function:n.)

```

\\_\_erw\_map:nn

```

343 \cs_set_protected:Nn \__erw_map:nn
344 {
345   \quark_if_recursion_tail_stop:n{#1}
346   \g__erw_tl_function:n{#1} \__erw_map:nn{#2}
347 }

(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
348 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnn
349 {
350   \erw_cs_apply:Nn #1
351   {\exp_args:Nf\__erw_tl_map_thread_at:Nnn {#3} {#2} }
352 }
353 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnn
354 {
355   \erw_cs_apply:Nnn #1
356   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnn {#3} {#2} }
357   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnn {#4} {#2} }
358 }
359 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnnn
360 {
361   \erw_cs_apply:Nnnn #1
362   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnn {#3} {#2} }
363   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnn {#4} {#2} }
364   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnn {#5} {#2} }
365 }
366 \cs_set_protected:Nn \__erw_tl_map_thread_at:Nnnnnn
367 {
368   \erw_cs_apply:Nnnnn #1
369   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnnn {#3} {#2} }
370   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnnn {#4} {#2} }
371   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnnn {#5} {#2} }
372   {\exp_args:Nf\__erw_tl_map_thread_at:Nnnnnn {#6} {#2} }
373 }

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

```

## 12.2 frontend

```

374 \cs_new:Nn \erw_tl_append_item:nn
375 {
376   {#1{#2}}
377 }
378 \cs_new:Nn \erw_tl_compose:nN
379 {
380   \__erw_oper_compose:NnN \erw_tl_fold:NN {#1} #2
381 }
382 \cs_new:Nn \erw_tl_compose:nn

```

```

383 {
384   \tl_set:Nn \g__erw_tl_compose_tl {#2}
385   \erw_tl_compose:nN{#1}\g__erw_tl_compose_tl
386   \g__erw_tl_compose_tl
387 }
388 \cs_new:Nn \erw_tl_compose_c:nN
389 {
390   \__erw_oper_compose:NnN \erw_tl_fold:cN {#1} #2
391 }
392 \cs_new:Nn \erw_tl_compose_c:nn
393 {
394   \tl_set:Nn \g__erw_tl_compose_tl {#2}
395   \erw_tl_compose_c:nN{#1}\g__erw_tl_compose_tl
396   \g__erw_tl_compose_tl
397 }
398 \cs_new:Nn \erw_tl_compose_vers:nN
399 {
400   \msg_error:nnn{__erw}{notdecl}{\erw_tl_compose_vers:nN}
401 }
402 \cs_new:Nn \erw_tl_compose_vers:nn
403 {
404   \erw_csint_reset:{}
405   \tl_map_function:nN{#1}\erw_csint_new:n
406   \exp_last_unbraced:Nx
407   \erw_tl_compose_c:nn
408   {{\erw_csint_names_braced:{}}}
409   {#2}
410 }
411 \cs_new:Nn \erw_tl_fold:NN
412 {
413   \use:c{tl_set:\g__erw_oper_fold_set_par_tl}
414   #2
415   {\use:c{erw_cs_apply:\g__erw_oper_fold_apply_par_tl}{#1}{#2}}
416 }
417 \cs_generate_variant:Nn \erw_tl_fold:NN {cN}
418 \cs_new:Nn \erw_tl_gset_function:N
419 {
420   \erw_cs_gset_eq:NN \g__erw_tl_function:n #1
421 }
422 \cs_new:Nn \erw_tl_gset_function:n
423 {
424   \erw_cs_gset_inline:Nn \g__erw_tl_function:n {#1}
425 }
426 \cs_new:Nn \erw_tl_last_item:n
427 {
428   \exp_args:Nof \tl_item:nn
429   {#1}
430   {
431     \tl_count:n{#1}
432   }
433 }
434 \cs_new_protected:Nn \erw_tl_map:n
435 {
436   \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop

```

```

437 }
438 \cs_new_protected:Nn \erw_tl_map:Nn
439 {
440   \cs_set_eq:NN \g__erw_tl_function:n #1
441   \erw_tl_map:n{#2}
442 }
443 \cs_new_protected:Nn \erw_tl_map_inline:nn
444 {
445   \erw_cs_set_inline:Nn \g__erw_tl_function:n {#1}
446   \erw_tl_map:n{#2}
447 }
448 \cs_new:Nn \erw_tl_repeat:nn
449 {
450   \int_step_inline:nnnn{1}{1}{#1}{#2}
451 }
452 \cs_new:Nn \erw_tl_split:nnn
453 {
454   \tl_head:n{#1}
455   \use:c{exp_args:#3} \tl_map_inline:nn
456   {
457     \tl_tail:n
458     {
459       #1
460     }
461   }{#2##1}
462 }
463 \cs_new:Nn \erw_tl_split:nn
464 {
465   \erw_tl_split:nnn{#1}{#2}{Nf}
466 }
467 \cs_new_protected:Nn \erw_tl_map_thread_at:Nnn
468 {
469   \exp_args:Nf\int_case:nnTF
470   {
471     \tl_count:n{#3}
472   }
473   {
474     {1}{ \__erw_tl_map_thread_at:Nnn #1{#2}#3 }
475     {2}{ \__erw_tl_map_thread_at:Nnnn #1{#2}#3 }
476     {3}{ \__erw_tl_map_thread_at:Nnnnn #1{#2}#3 }
477     {4}{ \__erw_tl_map_thread_at:Nnnnnn #1{#2}#3 }
478   }
479   {
480     % Do nothing
481   }
482   {
483     \msg_error:nnn{__erw}
484     {generic}
485     {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
486   }
487 }
488 \cs_new_protected:Nn \erw_tl_map_thread:Nn
489 {
490   \int_step_inline:nn

```

```

491 {
492   \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
493 }
494 {
495   \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
496 }
497 }

```

## 13 option

```

498 \cs_new_protected:Nn\erw_option:n
499 {
500   \keys_set:nn{__erw}{#1}
501 }

```

## 14 Closing

```

502 \ExplSyntaxOff

```