The erw-l3 package [*]

Erwann Rogard[†]

Released 2020/04/30

Abstract

Utilities based on expl3[1].

Résumé

Utilitaies de type expl3[1].

Contents

[*] This file describes version v1.8, last revised 2020/04/30.

[†] firstname dot lastname AusTria gmail dot com

Part I
Usage

`\usepackage`

`\usepackage{erw-l3}`

Requirement

1. `erw-l3.sty` and its dependencies are in the path of the LaTeX engine. See Part III, section 3.
2. Goes in the *preamble*

2 basics

`\erw_append_arg:nn`

`\erw_cs_apply:Nn{⟨arg list⟩}{⟨arg⟩}`

```
\erw_cs_apply:Nn
\erw_cs_apply:(No|Nf|Nx|cn)
\erw_cs_apply:Nnn
\erw_cs_apply:Nnnn
\erw_cs_apply:Nnnnn
```

`\erw_cs_apply:Nn {⟨control sequence⟩}{⟨token list₁⟩}`

`\erw_cs_identity:n`

`\erw_cs_identity:n{⟨arg⟩}`

```
\erw_cs_set_inline:Nn
\erw_cs_set_inline:cn
```

`\erw_cs_set_inline:Nn{⟨control sequence⟩}{⟨code⟩}`

3 csint

```
\erw_csint:nn          \erw_csint:nn{⟨integer⟩}{⟨arg⟩}
```

```
\erw_csint_name:n      \erw_csint_name:n{⟨integer⟩}
```

```
\erw_csint_names:nnn   \erw_csint_names:nnn{⟨integer⟩}{⟨integer⟩}{⟨integer⟩}
```

```
\erw_csint_new:n       \erw_csint_new:n{⟨integer⟩}
```

```
\erw_csint_names_braced:
\erw_csint_names_braced:n
\erw_csint_names_braced:nnn
```

```
\erw_csint_reset:
```

4 int

```
\erw_int_range:n       \erw_int_range:n{⟨integer⟩}
\erw_int_range:nn
```

5 oper

```
\erw_oper_compose:nN   \erw_oper_compose:nn{⟨control sequence list⟩}{⟨initial value⟩}
\erw_oper_compose:nn
```

```
\erw_oper_compose_c:nN
\erw_oper_compose_c:nn
```

```
\erw_oper_compose_vers:nN
\erw_oper_compose_vers:nn
```

```
\erw_oper_compose_seq:nN
\erw_oper_compose_seq_c:nN
\erw_oper_compose_seq_vers:nN
```

```
\erw_oper_fold:NN
\erw_oper_fold:cN
```

```
\erw_oper_gset_function:N
\erw_oper_gset_function:n
```

```
\erw_oper_fold_seq:NN
\erw_seq:cN
```

6 timestamp

```
\erw_timestamp:nn
\erw_timestamp:
```

```
\erw_timestamp:nn{date|time|datetime}{10|16}
```

Semantics Timestamp in base 10 or 16

7 tl

```
\erw_tl_last_item:n
```

```
\erw_tl_map:n
\erw_tl_map:Nn
```

```
\erw_tl_map_inline:nn
```

```
\erw_tl_merge:nn
```

```
\erw_tl_repeat:nn
```

```
\erw_tl_split:nnn
\erw_tl_split:nn
```

```
\erw_tl_map_thread_at:Nnn
\erw_tl_map_thread:Nn
```

8 option

```
\erw_option:n
```

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_oper_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 oper

Listing 4.

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

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

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

Listing 6.

```
\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_oper_compose_seq: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 7.

```
\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_oper_compose_seq_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 8.

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

h{g[f(X)]}

Listing 9.

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

f(X)
g[f(X)]

5 timestamp

Listing 10.

```
\ExplSyntaxOn
\noindent\erw_timestamp:nn{date}{10}{-}
\noindent\erw_timestamp:nn{time}{10}\\
\noindent\erw_timestamp:nn{datetime}{10}\\
\erw_timestamp:nn{date}{16}{\%}
\erw_timestamp:nn{time}{16}\\
\erw_option:n{ timestamp / delim = {\%} }
\erw_timestamp:nn{datetime}{16}\\
\ExplSyntaxOff
```

20200430-2202
20200430-2202
1343bee%89a
1343bee%89a

Listing 11.

```
\ExplSyntaxOn
\erw_option:n{ timestamp / delim = \c_empty_tl }
\iow_new:N \foo_iow
\tl_set:Nx \foo_dec { \erw_timestamp:nn{datetime}{10} }
\tl_set:Nx \foo_hex { \erw_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:202004302202
D:20200430220212-04’00’
Hello, world!

6 tl

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 LaTeXcommunity[2]

2 Install

1) Compile `timestamp.dtx` (under Unix, `$tex timestamp.dtx`)

2) Put the generated `timestamp.sty` in the search path of the LaTeXengine

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 LaTeX3 Project Team *The LaTeX3 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`

Part IV
Implementation

1 Opening

```
⟨@@=erw⟩
\ExplSyntaxOn
```

2 basics

2.1 backend

```
\cs_set:Nn \__erw_cs_name:N
{
  \exp_last_unbraced:Nf \use_i:nnn {\cs_split_function:N #1}
}
```

2.2 frontend

```
\cs_set:Nn \erw_append_arg:nn
{
  {#1{#2}}
}
\cs_set:Nn \erw_cs_apply:Nn
{
  #1{#2}
}
\cs_generate_variant:Nn \erw_cs_apply:Nn {No, Nf, Nx, c}
\cs_set:Nn \erw_cs_apply:Nnn
{
  #1{#2}{#3}
}
\cs_set:Nn \erw_cs_apply:Nnnn
{
  #1{#2}{#3}{#4}
}
\cs_set:Nn \erw_cs_apply:Nnnnn
{
  #1{#2}{#3}{#4}{#5}
}
\cs_set:Npn \erw_cs_identity:n #1{#1}
\cs_set:Nn \erw_cs_set_inline:Nn
{
  \cs_set:Npn #1 ##1{#2}
}
\cs_generate_variant:Nn \erw_cs_set_inline:Nn {cn}
\cs_set:Nn \erw_cs_gset_inline:Nn
{
  \cs_gset:Npn #1 ##1{#2}
}
\cs_generate_variant:Nn \erw_cs_gset_inline:Nn {cn}
```

3 csint

3.1 backend

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

3.2 frontend

```
\cs_set:Nn \erw_csint:nn
{
  \erw_cs_apply:cn{__erw_csint_\int_to_alph:n{#1}:n}{#2}
}
\cs_set:Nn \erw_csint_name:n {__erw_csint_\int_to_alph:n{#1}:n}
\cs_new:Nn \erw_csint_names:nnn
{
  \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_csint_name:n
}
\cs_new_protected:Nn \erw_csint_new:n
{
  \int_incr:N \g__erw_csint_int
  \erw_cs_set_inline:cn{\g__erw_csint_name_tl}
  {
    \token_if_cs:NTF
    {#1}
    {#1{##1}}
    {#1}
  }
}
\cs_new:Nn \erw_csint_names_braced:nnn
{
  \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_csint_names_braced:n
  % TODO \tl_range_braced:nnn?
}
\cs_set:Nn \erw_csint_names_braced:n {{\erw_csint_name:n{#1}}}
\cs_new:Nn \erw_csint_names_braced:
{
  \erw_csint_names_braced:nnn{1}{1}{\g__erw_csint_int}
}
\cs_new_protected:Nn \erw_csint_reset:
{
  \int_zero:N \g__erw_csint_int
  \tl_set:Nn \__erw_csint_ext_tl{}%^^A TODO remove?
}
```

4 int

4.1 backend

```
\cs_set:Npn \__erw_int_range:nnn #1 #2 #3
{
  \int_compare:nNnTF
  {
    \int_eval:n{#2+1}
  }>{#3}
  {
    {#1}
  }
  {
    \__erw_int_range:nnn
    {
      \exp_args:Nx\erw_append_arg:nn{#1}
      {
        \int_eval:n{#2+1}
      }
    }
    {\int_eval:n{#2+1}}
    {#3}
  }
}
```

4.2 frontend

```
\cs_set:Nn \erw_int_range:nn
{
  \__erw_int_range:nnn {{#1}}{#1}{#2}
}
\cs_set:Nn \erw_int_range:n
{
  \__erw_int_range:nnn {}{0}{#1}
% ^^A Alt to:
% ^^A    \int_step_inline:nn {#1}{##1}
}
```

5 msg

5.1 backend

```
\msg_new:nnn{__erw}{generic}{#1}
\msg_new:nnn{__erw}{notdecl}{#1~not~declared}
\msg_new:nnn{__erw}{notset}{#1~not~set}
%\end{macrocode}
% \section{\textsf{oper}}\label{impl:oper}
% \subsection{backend}
%    \begin{macrocode}
\tl_new:N \g__erw_compose_tl
\tl_new:N \g__erw_oper_fold_seq_item_tl
\cs_set:Nn \__erw_oper_compose:NnN
{
  \erw_cs_set_inline:Nn \__erw_oper_function:n
  {
    #1{##1}#3
  }
  \exp_args:Nf\erw_tl_map:n
  {
    \tl_reverse:n{#2}
  }
}
```

5.2 frontend

```
\cs_set:Nn \erw_oper_compose:nN
{
  \__erw_oper_compose:NnN \erw_oper_fold:NN {#1} #2
}
\cs_set:Nn \erw_oper_compose:nn
{
  \tl_set:Nn \g__erw_compose_tl {#2}
  \erw_oper_compose:nN{#1}\g__erw_compose_tl
  \g__erw_compose_tl
}
\cs_set:Nn \erw_oper_compose_c:nN
{
  \__erw_oper_compose:NnN \erw_oper_fold:cN {#1} #2
}
\cs_set:Nn \erw_oper_compose_c:nn
{
  \tl_set:Nn \g__erw_compose_tl {#2}
  \erw_oper_compose_c:nN{#1}\g__erw_compose_tl
  \g__erw_compose_tl
}
\cs_set:Nn \erw_oper_compose_vers:nN
{
  \msg_error:nnn{__erw}{notdecl}{\erw_oper_compose_vers:nN}
}
\cs_set:Nn \erw_oper_compose_vers:nn
{
  \erw_csint_reset:{}
  \tl_map_function:nN{#1}\erw_csint_new:n
  \exp_last_unbraced:Nx
  \erw_oper_compose_c:nn
  {{\erw_csint_names_braced:{}}}
  {#2}
}
\cs_set:Nn \erw_oper_compose_seq:nN
{
  \__erw_oper_compose:NnN \erw_oper_fold_seq:NN {#1} #2
}
\cs_set:Nn \erw_oper_compose_seq_c:nN
{
  \__erw_oper_compose:NnN \erw_oper_fold_seq:cN {#1} #2
}
\cs_set:Nn \erw_oper_compose_seq_vers:nN
{
  \msg_error:nnn{__erw}{notdecl}{\erw_oper_compose_seq_vers:nN}
}
\cs_set:Nn \erw_oper_gset_function:N
{
  \erw_cs_gset_eq:NN \__erw_oper_function:n #1
}
\cs_set:Nn \erw_oper_gset_function:n
{
  \erw_cs_gset_inline:Nn \__erw_oper_function:n {#1}
}
\keys_define:nn{__erw}
```

```
{
  oper/fold_set_par.tl_gset:N = \g__erw_oper_fold_set_par_tl,
  oper/fold_set_par.value_required:n = true,
  oper/fold_set_par.default:n = {Nf},
  oper/fold_set_par.initial:n = {Nf},
  oper/fold_apply_par.tl_gset:N = \g__erw_oper_fold_apply_par_tl,
  oper/fold_apply_par.value_required:n = true,
  oper/fold_apply_par.default:n = {Nf},
  oper/fold_apply_par.initial:n = {Nf}
}
\cs_set:Nn \erw_oper_fold:NN
{
  \use:c{tl_set:\g__erw_oper_fold_set_par_tl}
  #2
  {\use:c{erw_cs_apply:\g__erw_oper_fold_apply_par_tl}{#1}{#2}}
}
\cs_generate_variant:Nn \erw_oper_fold:NN {cN}
\cs_set:Nn \erw_oper_fold_seq:NN
{
  \seq_get_right:NN #2 \g__erw_oper_fold_seq_item_tl
  \erw_oper_fold:NN #1 \g__erw_oper_fold_seq_item_tl
  \seq_put_right:No #2 {\g__erw_oper_fold_seq_item_tl}
}
\cs_generate_variant:Nn \erw_oper_fold_seq:NN {cN}
```

6 timestamp

6.1 backend

```
\msg_new:nnn{__erw}{timestamp / base}{Calling~#1,~arg~must~be~’dec|hex’}
\msg_new:nnn{__erw}{timestamp / period}{Calling~#1,~arg~must~be~’date|time|datetime’}
```

__erw_timestamp_date:N
__erw_timestamp_date_dec:
__erw_timestamp_date_hex:

```
\cs_new:Nn \__erw_timestamp_date_dec:
{
  \int_eval:n
  {
    \c_sys_year_int * 10000
    +\c_sys_month_int * 100
    +\c_sys_day_int *  1
  }
}
\cs_new:Nn \__erw_timestamp_date:N{\int_to_hex:n{\__erw_timestamp_date_dec:}}
\cs_new:Nn \__erw_timestamp_date_hex:{\int_to_hex:n{\__erw_timestamp_date_dec:}}
```

(*End definition for* `\__erw_timestamp_date:N`, `\__erw_timestamp_date_dec:`, *and* `\__erw_timestamp_date_hex:`.)

__erw_timestamp_time_dec:
__erw_timestamp_time_hex

```
\cs_new:Nn \__erw_timestamp_time_dec:
{
  \int_eval:n
  {
    \c_sys_hour_int * 100
    +\c_sys_minute_int * 1
  }
}
\cs_new:Nn\__erw_timestamp_time_hex:{\int_to_hex:n{\__erw_timestamp_time_dec:}}
```

(*End definition for* `\__erw_timestamp_time_dec:` *and* `\__erw_timestamp_time_hex`*.*)

__erw_timestamp_datetime_base:n
__erw_timestamp_datetime_dec:n
__erw_timestamp_datetime_join:nn
__erw_timestamp_datetime_hex:n
__erw_timestamp_datetime_period:n

```
\cs_new:Nn\__erw_timestamp_datetime_base:n
{
  \int_case:nnTF{#1}
  {
    {10}{dec}
    {16}{hex}
  }
  {\c_empty_tl}
  {\msg_error:nnn{__erw}{timestamp / base}{\__erw_timestamp_datetime_base:n{#1}}}
}
\cs_new:Nn\__erw_timestamp_datetime_join:nnn{#1#2#3}
\cs_new:Nn\__erw_timestamp_datetime_join:nn{\__erw_timestamp_datetime_join:nnn{#1}{\g__erw_ti
\cs_new:Nn\__erw_timestamp_datetime_period:n
{
  \str_case:nnTF{#1}
  {
    {date}{date}
    {time}{time}
    {datetime}{datetime}
  }
  {\c_empty_tl}
  {\msg_error:nnn{__erw}{ timestamp / period }{\__erw_timestamp_datetime_period:n{#1}}}
}
\cs_new:Nn\__erw_timestamp_datetime_dec: {\__erw_timestamp_datetime_join:nn{\__erw_timestamp_
\cs_new:Nn\__erw_timestamp_datetime_hex: {\__erw_timestamp_datetime_join:nn{\__erw_timestamp_
```

(*End definition for* `\__erw_timestamp_datetime_base:n` *and others.*)

__erw_timestamp:nn

```
\cs_new:Nn\__erw_timestamp:nn
{
  \exp_args:No
  \use:c{__erw_timestamp_\__erw_timestamp_datetime_period:n{#1}_\__erw_timestamp_datetime_bas
}
\cs_new_protected:Nn \__erw_timestamp_set_delim:nn
{
  \use:c{tl_gset:N#1}
  \g__erw_timestamp_delim_str{#2}
}
```

(*End definition for* `\__erw_timestamp:nn`*.*)

```
\keys_define:nn{__erw}
{
  timestamp / delim .code:n =
  {
    \exp_last_unbraced:No
    \__erw_timestamp_set_delim:nn{n}{#1}
  },
  timestamp / delim  .value_required:n = true,
  timestamp / delim  .default:n = {-},
  timestamp / delim  .initial:n = {-}
}
```

6.2 frontend

```
\cs_new:Nn\erw_timestamp:nn
{
  \__erw_timestamp:nn{#1}{#2}
}
\cs_new:Nn\erw_timestamp:
{
  \__erw_timestamp:nn{datetime}{16}
}
```

7 tl

7.1 backend

__erw_oper_function:n

```
\cs_new_protected:Nn \__erw_oper_function:n
{
  \msg_error:nnn
  {erw}
  {notset}
  {\__erw_oper_function:n}
}
```

(*End definition for* `\__erw_oper_function:n`*.*)

__erw_map:nn

```
\cs_set_protected:Nn \__erw_map:nn
{
  \quark_if_recursion_tail_stop:n{#1}
  \__erw_oper_function:n{#1}  \__erw_map:nn{#2}
}
```

(*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

```
\cs_set_protected:Nn \__erw_map_thread_at:Nnn
{
  \erw_cs_apply:Nn #1
  {\exp_args:Nf\tl_item:nn {#3} {#2} }
}
\cs_set_protected:Nn \__erw_map_thread_at:Nnnn
{
  \erw_cs_apply:Nnn #1
  {\exp_args:Nf\tl_item:nn {#3} {#2} }
  {\exp_args:Nf\tl_item:nn {#4} {#2} }
}
\cs_set_protected:Nn \__erw_map_thread_at:Nnnnn
```

```
{
  \erw_cs_apply:Nnnn #1
  {\exp_args:Nf\tl_item:nn {#3} {#2} }
  {\exp_args:Nf\tl_item:nn {#4} {#2} }
  {\exp_args:Nf\tl_item:nn {#5} {#2} }
}
\cs_set_protected:Nn \__erw_map_thread_at:Nnnnnn
{
  \erw_cs_apply:Nnnnn #1
  {\exp_args:Nf\tl_item:nn {#3} {#2} }
  {\exp_args:Nf\tl_item:nn {#4} {#2} }
  {\exp_args:Nf\tl_item:nn {#5} {#2} }
  {\exp_args:Nf\tl_item:nn {#6} {#2} }
}
```

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

7.2 frontend

```
\cs_set:Nn \erw_tl_last_item:n
{
  \exp_args:Nof \tl_item:nn
  {#1}
  {
    \tl_count:n{#1}
  }
}
\cs_set_protected:Nn \erw_tl_map:n
{
  \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
}
\cs_set_protected:Nn \erw_tl_map:Nn
{
  \cs_set_eq:NN \__erw_oper_function:n #1
  \erw_tl_map:n{#2}
}
\cs_set_protected:Nn \erw_tl_map_inline:nn
{
  \erw_cs_set_inline:Nn \__erw_oper_function:n {#1}
  \erw_tl_map:n{#2}
}
\cs_set:Nn \erw_tl_merge:nn
{
  {#1#2}
}
\cs_set:Nn \erw_tl_repeat:nn
{
  \int_step_inline:nnnn{1}{1}{#1}{#2}
}
\cs_set_protected:Nn \erw_tl_split:nnn
{
  \tl_head:n{#1}
  \use:c{exp_args:#3} \tl_map_inline:nn
  {
    \tl_tail:n
    {
      #1
    }
  }{#2##1}
}
\cs_set_protected:Nn \erw_tl_split:nn
{
  \erw_tl_split:nnn{#1}{#2}{Nf}
}
\cs_set_protected:Nn \erw_tl_map_thread_at:Nnn
{
  \exp_args:Nf\int_case:nnTF
  {
    \tl_count:n{#3}
  }
  {
    {1}{ \__erw_map_thread_at:Nnn #1{#2}#3 }
    {2}{ \__erw_map_thread_at:Nnnn #1{#2}#3 }
    {3}{ \__erw_map_thread_at:Nnnnn #1{#2}#3 }
    {4}{ \__erw_map_thread_at:Nnnnnn #1{#2}#3 }
  }
  {
    % Do nothing
  }
  {
    \msg_error:nnn{__erw}
    {generic}
    {erw_tl_map_thread_at:~count~of~#3~not~withing~1~to~4}
  }
}
\cs_set_protected:Nn \erw_tl_map_thread:Nn
{
  % TODO check that #2 is a matrix
  \int_step_inline:nn
  {
    \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
  }
  {
    \erw_tl_map_thread_at:Nnn #1 {##1} {#2}
  }
}
```

8 option

```
\cs_new_protected:Nn\erw_option:n
{
  \keys_set:nn{__erw}{#1}
}
```

9 Closing

```
\ExplSyntaxOff
```