The erw-l3 package*

Erwann Rogard†

Released 2020/04/27

Abstract

expl3-like utilities. Some redundant (for sport), some not.

Résumé

Logiciels utilitaires de type expl3. Certains redondants (pour le sport), d'autres pas.

Contents

*(version v0.7, last revised 2020/04/27).

†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 II, section 3.
2. Declare it in the *preamble*

Part II
Other

1 Acknowledgment

This work has benefited from Q&A's from the LaTeXcommunity[2]

2 Install

1) Compile `erw-l3.dtx` (under Unix, `$tex erw-l3.dtx`)

2) Put the generated `erw-l3.sty` in the search path of the LaTeXengine

3 Support

This package is available from `https://www.ctan.org/pkg/erw-l3` and `https://github.com/rogard/erw-l3`.

4 Test

4.1 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
```

4.2 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)`

4.3 Results

1) erw-l3 v0.7 compiles satisfactorily on platform *i)* and engines *b)*, *c)*, and *d)*

4.4 TODO

1. Complete Part I

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 III
Listing

1 basics

Listing 1.

```
f(X)
```

2 int

Listing 2.

```
234512345
```

3 numberdd cs

Listing 3. TODO

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

4 oper

Listing 4.

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

Listing 5.

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

Listing 6.

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

Listing 7. TODO

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

Listing 8.

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

Listing 9.

```
f(X)
g[f(X)]
```

5 sys

Listing 10. sys

```
20200427
1343beb
1817
719
erw-l3–1343beb–719
```

6 tl

Listing 11.

TF

Listing 12.

abracadabracadabracadabra

Listing 13.

a==b==c

Listing 14.

(a)(b)(c)

Listing 15.

(a)(b)(c)

Listing 16.

(a)(b)(c)

Listing 17.

(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 18.

(1+a)(2+b)(3+c)

Part IV
Implementation

1 Opening

```
⟨@@=erw⟩
\ExplSyntaxOn
```

2 basics

```
\cs_set:Npn \__erw_cs_name:N #1
{
  \exp_last_unbraced:Nf \use_i:nnn {\cs_split_function:N #1}
}
\cs_set:Npn \erw_cs_apply:Nnn #1 #2 #3
{
  #1{#2}{#3}
}
\cs_set:Npn \erw_cs_apply:Nnnn #1 #2 #3 #4
{
  #1{#2}{#3}{#4}
}
\cs_set:Npn \erw_cs_apply:Nnnnn #1 #2 #3 #4 #5
{
  #1{#2}{#3}{#4}{#5}
}
\cs_set:Npn \erw_cs_apply:Nn
#1 % fun
#2 % tl
{
  #1{#2}
}
\cs_generate_variant:Nn \erw_cs_apply:Nn {No, Nf, Nx, c}
\cs_set:Npn \erw_cs_set_eq:NN #1 #2
{
  \cs_set:Npn #1 ##1{#2{##1}}
}
\cs_generate_variant:Nn \erw_cs_set_eq:NN {cN}
\cs_set:Npn \erw_cs_gset_eq:NN #1 #2
{
  \cs_gset:Npn #1 ##1{#2{##1}}
}
\cs_generate_variant:Nn \erw_cs_gset_eq:NN {cN}
\cs_set:Npn \erw_cs_set_inline:Nn #1 #2
{
  \cs_set:Npn #1 ##1{#2}
}
\cs_generate_variant:Nn \erw_cs_set_inline:Nn {cn}
\cs_set:Npn \erw_cs_gset_inline:Nn #1 #2
{
  \cs_gset:Npn #1 ##1{#2}
}
```

```
\cs_generate_variant:Nn \erw_cs_gset_inline:Nn {cn}
\cs_set:Npn \erw_identity:n #1{#1}
```

3 int

```
\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_accum:nn{#1}
      {
        \int_eval:n{#2+1}
      }
    }
    {\int_eval:n{#2+1}}
    {#3}
  }
}
\cs_set:Npn \erw_int_range:nn #1 #2
{
  \__erw_int_range:nnn {{#1}}{#1}{#2}
}
\cs_set:Npn \erw_int_range:n #1
{
  \__erw_int_range:nnn {}{0}{#1}
% ^^A Alt to:
% ^^A    \int_step_inline:nn {#1}{##1}
}
```

4 oper

```
\cs_set:Npn \erw_accum:nn #1 #2
{
  {#1{#2}}
}
\cs_set:Npn \erw_compose:NnV
#1 % method
#2 % funs
#3 % var
{
  \erw_cs_set_inline:Nn \__erw_map:n
  {
    #1{##1}#3
  }
  \exp_args:Nf\erw_map:n
  {
```

```
    \tl_reverse:n{#2}
  }
}
\cs_set:Npn \erw_compose:nV #1 #2
{
  \erw_compose:NnV \erw_fold:NV {#1} #2
}
\cs_set:Npn \erw_compose_c:nV #1 #2
{
  \erw_compose:NnV \erw_fold:cV {#1} #2
}
\tl_new:N \g__erw_compose_tl
\cs_set:Npn \erw_compose:nn #1 #2
{
  \tl_set:Nn \g__erw_compose_tl {#2}
  \erw_compose:nV{#1}\g__erw_compose_tl
  \g__erw_compose_tl
}
\cs_set:Npn \erw_compose_c:nn #1 #2
{
  \tl_set:Nn \g__erw_compose_tl {#2}
  \erw_compose_c:nV{#1}\g__erw_compose_tl
  \g__erw_compose_tl
}
\cs_set:Npn \erw_compose_vers:nV #1 #2
{
  \msg_error:nnn{erw}{generic}{erw_compose_vers:nV~yet-to~be~implemented}
}
\cs_set:Npn \erw_compose_vers:nn #1 #2
{
  \erw_cs_no_reset:{}
  \tl_map_function:nN{#1}\erw_cs_no_new:n
  \exp_last_unbraced:Nx
  \erw_compose_c:nn
  {{\erw_cs_no_names_braced:{}}}
  {#2}
}
\cs_set:Npn \erw_compose_seq:nV #1 #2
{
  \erw_compose:NnV \erw_seq_fold:NV {#1} #2
}
\cs_set:Npn \erw_compose_seq_c:nV
#1 % funs
#2 % seq
{
  \erw_compose:NnV \erw_seq_fold:cV {#1} #2
}
\cs_set:Npn \erw_compose_seq_vers:nV #1 #2
{
  \msg_error:nnn{erw}{generic}{erw_compose_seq_vers:nV~yet-to~be~implemented}
}
\tl_new:N \g__erw_seq_fold_item_tl
\cs_set:Npn \erw_seq_fold:NV
#1 % fun
```

```
#2 % seq
{
  \seq_get_right:NN #2 \g__erw_seq_fold_item_tl
  \erw_fold:NV #1 \g__erw_seq_fold_item_tl
  \seq_put_right:No #2 {\g__erw_seq_fold_item_tl}
}
\cs_generate_variant:Nn \erw_seq_fold:NV {cV}
\cs_set:Npn \erw_fold:NV
#1 % fun
#2 % var
{
  \use:c{tl_set:\g__erw_fold_set_par_tl}
  #2
  {\use:c{erw_cs_apply:\g__erw_fold_apply_par_tl}{#1}{#2}}
}
\cs_generate_variant:Nn \erw_fold:NV {cV}
```

5 sys

__erw_sys_date:

```
\cs_new:Nn \__erw_sys_date:
{
  \int_eval:n
  {
    \c_sys_year_int * 10000
    +\c_sys_month_int * 100
    +\c_sys_day_int *  1
  }
}
```

(*End definition for* `\__erw_sys_date:`.)

\erw_sys_date:

```
\cs_new:Nn\erw_sys_date:{\__erw_sys_date:}
```

__erw_sys_date_hex:

```
\cs_new:Nn \__erw_sys_date_hex:
{\int_to_hex:n{\__erw_sys_date:}}
```

(*End definition for* `\__erw_sys_date_hex:`.)

\erw_sys_date_hex:

```
\cs_new:Nn\erw_sys_date_hex:{\__erw_sys_date_hex:}
```

__erw_sys_filename:

```
\cs_new:Nn\__erw_sys_filename:
{
  \c_sys_jobname_str--
  \__erw_sys_date_hex:--
  \__erw_sys_time_hex:
}
```

(*End definition for* `\__erw_sys_filename:`.)

`\erw_sys_filename:`

```
\cs_new:Nn\erw_sys_filename:{\__erw_sys_filename:}
```

`\__erw_sys_time:`

```
\cs_new:Nn \__erw_sys_time:
{
  \int_eval:n
  {
    \c_sys_hour_int * 100
    +\c_sys_minute_int * 1
  }
}
```

(*End definition for* `\__erw_sys_time:`.)

```
\cs_new:Nn\erw_sys_time:{\__erw_sys_time:}
```

`\__erw_sys_time_hex:`

```
\cs_new:Nn\__erw_sys_time_hex:
{\int_to_hex:n{\__erw_sys_time:}}
```

(*End definition for* `\__erw_sys_time_hex:`.)

```
\cs_new:Nn \erw_sys_time_hex:{\__erw_sys_time_hex:}
```

6 tl

```
\prg_set_conditional:Npnn \erw_is_matrix:n #1 { p, TF }
{
  \erw_gset_map_inline:n{==\tl_count:n{##1}}
  \int_compare:nTF
  {
    \exp_args:Nf\tl_count:n{\tl_head:n{#1}}
    \exp_args:Nf \erw_map:n
    {
      \tl_tail:n{#1}
    }
  }
  {\prg_return_true:}
  {\prg_return_false:}
}
\cs_set:Npn \erw_gset_map:N #1
{
  \erw_cs_gset_eq:NN \__erw_map:n #1
}
\cs_set:Npn \erw_gset_map_inline:n #1
{
  \erw_cs_gset_inline:Nn \__erw_map:n {#1}
}
```

```
\cs_set:Npn \erw_last_item:n #1
{
  \exp_args:Nof \tl_item:nn
  {#1}
  {
    \tl_count:n{#1}
  }
}
\cs_set:Npn \erw_map:n #1
{
  \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
}
\cs_set:Npn \__erw_map:nn #1 #2
{
  \quark_if_recursion_tail_stop:n{#1}
  \__erw_map:n{#1}  \__erw_map:nn{#2}
}
\cs_new:Npn \__erw_map:n #1
{
  \msg_error:nnn
  {erw}
  {generic}
  {__erw_map:n~not~set}
}
\cs_set:Npn \erw_map:Nn
#1 % fun
#2 % tl
{
  \erw_cs_set_eq:NN \__erw_map:n #1
  \erw_map:n{#2}
}
\cs_set:Npn \erw_map_inline:nn
#1 % inl
#2 % tl
{
  \erw_cs_set_inline:Nn \__erw_map:n {#1}
  \erw_map:n{#2}
}
\cs_set:Npn \erw_merge:nn #1 #2
{
  {#1#2}
}
\cs_set:Npn \erw_repeat:nn #1 #2
{
  \int_step_inline:nnnn{1}{1}{#1}{#2}
}
\cs_set:Npn \erw_split:nnn #1 #2 #3
{
  \tl_head:n{#1}
  \use:c{exp_args:#3} \tl_map_inline:nn
  {
```

```
    \tl_tail:n
    {
      #1
    }
  }{#2##1}
}
\cs_set:Npn \erw_split:nn #1 #2
{
  \erw_split:nnn{#1}{#2}{Nf}
}
\cs_set:Npn \__erw_map_thread_at:Nnn #1 #2 #3
{
  \erw_cs_apply:Nn #1
  {\exp_args:Nf\tl_item:nn {#3} {#2} }
}
\cs_set:Npn \__erw_map_thread_at:Nnnn #1 #2 #3 #4
{
  \erw_cs_apply:Nnn #1
  {\exp_args:Nf\tl_item:nn {#3} {#2} }
  {\exp_args:Nf\tl_item:nn {#4} {#2} }
}
\cs_set:Npn \__erw_map_thread_at:Nnnnn #1 #2 #3 #4 #5
{
  \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:Npn \__erw_map_thread_at:Nnnnnn #1 #2 #3 #4 #5 #6
{
  \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} }
}
\cs_set:Npn \erw_map_thread_at:Nnn #1 #2 #3
{
  \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_map_thread_at:~count~of~#3~not~withing~1~to~4}
  }
}
\cs_set:Npn \erw_map_thread:Nn #1 #2
{
  % TODO check that #2 is a matrix
  \int_step_inline:nn
  {
    \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
  }
  {
    \erw_map_thread_at:Nnn #1 {##1} {#2}
  }
}
```

7 numbrdcs

```
\int_new:N \g__erw_cs_no_int
\cs_set:Npn \erw_cs_no_name:n #1{__erw_cs_no_\int_to_alph:n{#1}:n}
\cs_set:Npn \erw_cs_no_name_braced:n #1{{\erw_cs_no_name:n{#1}}}
\tl_set:Nn \g__erw_cs_no_name_tl {\erw_cs_no_name:n{\g__erw_cs_no_int}}
\cs_set:Npn \erw_cs_no:nn #1 #2
{
  \erw_cs_apply:cn{__erw_cs_no_\int_to_alph:n{#1}:n}{#2}
}
\cs_new_protected:Npn \erw_cs_no_reset:
{
  \int_zero:N \g__erw_cs_no_int
  \tl_set:Nn \__erw_cs_no_ext_tl{}%^^A remove
}
\cs_new_protected:Npn \erw_cs_no_new:n #1
{
  \int_incr:N \g__erw_cs_no_int
  \erw_cs_set_inline:cn{\g__erw_cs_no_name_tl}
  {
    \token_if_cs:NTF
    {#1}
    {#1{##1}}
    {#1}
  }
}
\cs_new:Npn \erw_cs_no_names:nnn #1 #2 #3
{
  \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_cs_no_name:n
}
\cs_new:Npn \erw_cs_no_names_braced:nnn #1 #2 #3
{
  \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_cs_no_name_braced:n
  % TODO \tl_range_braced:nnn?
```

```
}
\cs_new:Npn \erw_cs_no_names_braced:
{
  \erw_cs_no_names_braced:nnn{1}{1}{\g__erw_cs_no_int}
}
```

8 option

```
\keys_define:nn{__erw}
{
  fold/set_par. tl_set:N = \g__erw_fold_set_par_tl,
  fold/set_par. value_required:n = true,
  fold/set_par. default:n = Nf,
  fold/set_par. initial:n = Nf,
  fold/apply_par. tl_set:N = \g__erw_fold_apply_par_tl,
  fold/apply_par. value_required:n = true,
  fold/apply_par. default:n = Nf,
  fold/apply_par. initial:n = Nf
}
```

9 Closing

```
%      ^^A \ProcessKeysPackageOptions{ erw }
\ExplSyntaxOff
```