

erw-l3*

Erwann Rogard[†]

Released 2018/6/21

Abstract

L^AT_EX3 package defining commands built around `expl3`[1]. For example, `\erw_compose` implements the mathematical concept $f_1 \circ f_2 \cdots \circ f_n$.

Contents

1	Preliminaries	2
I	Usage	2
1	backend	2
	1.1 <code>compose</code>	3
	1.2 <code>csutil</code>	3
	1.3 <code>map</code>	4
	1.4 <code>numbrdcs</code>	4
2	frontend	4
	2.1 <code>disambig</code>	5
	2.2 <code>numbrdcs</code>	5
II	Listings	6
1	Backend	6
	1.1 <code>compose</code>	6
	1.2 <code>csutil</code>	7
	1.3 <code>map</code>	8
	1.4 <code>numbrdcs</code>	10
2	Frontend	10
	2.1 <code>disambig</code>	10
	2.2 <code>numbrdcs</code>	11
III	Other	11

*This file describes version v0.1.3, last revised 2018/6/21.

[†]firstname dot lastname AusTria gmail dot com

1	Support	11
2	Acknowledgment	12
IV	Implementation	12
1	Back end	12
	1.1 <code>compose</code>	12
	1.2 <code>csutil</code>	13
	1.3 <code>map</code>	15
	1.4 <code>numbrdcs</code>	17
2	frontend	18
	2.1 <code>disambig</code>	18
	2.2 <code>numbrdcs</code>	19
	Change History	19
	Index	20

1 Preliminaries

See [Part III section 1](#) on how to get this package. To use it, make sure the file `erw-13.sty` is in the path of the \LaTeX engine. In the preamble of your \LaTeX document, put:

```
\usepackage[<options>]{erw-13}
```

Part I

Usage

The naming conventions are (loosely) those of \LaTeX 3. For example, `<cs>` stands for *control sequence*, which is described in [\[1, Part I3basics\]](#).

1 backend

We call ‘backend’ commands that are `expl3`-like.

1.1 compose

`\erw_compose:nV` `\erw_compose:nV{<cs list>}<var>`

`\erw_compose:nn` Implements the mathematical concept $f_1 \circ f_2 \cdots \circ f_n$. See Listing 1

`\erw_compose_c:nV` `\erw_compose_c:nV{<cs names>}<var>`

`\erw_compose_c:nn` See Listing 2

`\erw_compose_seq:nV` `\erw_compose_seq:nV{<cs list>}<seq>`

Same as `\erw_compose:nV`, but saves each intermediary step See Listing 3

`\erw_compose_seq_c:nV` `\erw_compose_seq_c:nV{<cs names>}<seq>`

See Listing 4

`\erw_compose_vers:nV` `\erw_compose_vers:nV{<list of cs or code>}<var>`

`\erw_compose_vers:nn` See Listing 5. Only the nn version is implemented

`\erw_compose_seq_vers:nV` `\erw_compose_seq_vers:nV{<list of cs or code>}<seq>`

`\erw_compose_seq_vers:nn` Not implemented.

1.2 csutil

`\erw_apply:Nn` `\erw_apply:Nn{<cs>}{<arg>}`

`\erw_apply:cn` Expands to `<cs>{<arg>}`

`\erw_cs_set_eq:NN` `\erw_cs_set_eq:NN{<cs1>}<cs2>`

`\erw_cs_set_eq:cN` `<cs1>←<cs2>`

`\erw_cs_set_inline:Nn` `\erw_cs_set_inline:Nn{<cs>}{<code>}`

`\erw_cs_set_inline:cn`

`\erw_identity:N` `\erw_identity:N{<arg>}`

`\erw_identity:c` Expands to `<arg>`

`\erw_fold:NV` `\erw_fold:NV{<cs>}<var>`

`\erw_fold:cV` `<var>←\erw_apply:NV{<cs>}<var>`. See Listing 7.

`\erw_items_to:nn` `\erw_items_to:nn{<int>}{<token list>}`

See Listing 8

`\erw_last_item:nn` `\erw_last_item:nn{<int>}{<token list>}`

See Listing 8

`\erw_repeat:nn` `\erw_repeat:nn{<int>}{<value>}`

See Listing 9

`\erw_split:nn` `\erw_split:nn{<token list>}{<delimiter>}`

See Listing 10

1.3 map

`\erw_map:Nn` `\erw_map:Nn<cs>{<args>}`

See Listing 11. Redundant with `\tl_map_function:nN`

`\erw_map_inline:nn` `\erw_map_inline:nn{<code>}{<args>}`

See Listing 12

`\erw_map_thread:Nn` `\erw_map_thread:Nn<cs>{<matrix of tokens>}`

Threads `<cs>` over the columns, where the arity of `<cs>` is equal to the number of rows.

See Listing 13

`\erw_map_thread_at:Nnn` `\erw_map_thread_at:Nnn<cs>{<matrix of tokens>}`

1.4 numbrdcs

Part of these commands have a frontend counterpart, see [subsection 2.2](#).

`\erw_numbrd_cs_reset:` `\erw_numbrd_cs_reset:{}_`

See Listing 14

`\erw_numbrd_cs_new:n` `\erw_numbrd_cs_new:n {<cs or code>}`

Use it as the first arg to `\tl_function_map:Nn`

`\erw_numbrd_cs:nn` `\erw_numbrd_cs:nn {<cs or code>}`

`\erw_numbrd_cs_names_braced:nnn` `\erw_numbrd_cs_names_braced:nnn{<first>}{<step>}{<last>}`

See Listing 14

2 frontend

We call frontend commands created with `pkgxparse's\NewDocumentCommand[2]`

2.1 disambig

<hr/> <code>\disambignewcmd</code> <hr/> <code>\disambignewcmd*</code> <hr/>	<code>\disambignewcmd{<token>}{<pars>}{<code>}</code> Analogues of <code>\NewDocumentCommand</code> and <code>\RenewDocumentCommand</code> . See Listing 15
<hr/> <code>\disambignewenv</code> <hr/> <code>\disambignewenv*</code> <hr/>	<code>\disambignewenv{<token>}{<pars>}{<code1>}{<code2>}</code> Analogues of <code>\NewDocumentEnvironment</code> and <code>\RenewDocumentEnvironment</code> . See Listing 16
<hr/> <code>\disambigset</code> <hr/>	<code>\disambigset{<prefix>}</code>
<hr/> <code>\disambigunset</code> <hr/>	<code>\disambigunset{}</code>

2.2 numbrdcs

<hr/> <code>\numbrdcsnew</code> <hr/> <code>\numbrdcsnew*</code> <hr/>	<code>\numbrdcsnew{<list of cs or code>}</code> Creates numbered control sequences. The starred version does not reset. See Listing 17
<hr/> <code>\numbrdcs</code> <hr/>	<code>\numbrdcs{<int>}{<arg>}</code> Evaluates control sequence numbered <code><int></code> with argument <code><arg></code> . See Listing 17

Part II

Listings

1 Backend

1.1 compose

Listing 1

```
\cs_set:Npn \__foo #1 {f(#1)}
\cs_set:Npn \__bar #1 {g[#1]}
\cs_set:Npn \__baz #1 {h\{#1\}}
\tl_set:Nn \l_tmpa_tl{X}
\erw_compose:nV{
  {\__baz}{\__bar}{\__foo}}
  \l_tmpa_tl
\l_tmpa_tl h{g[f(X)]}
\tl_set:Nn \l_tmpa_tl{X}
\erw_compose:nn{
  {\__baz}{\__bar}{\__foo}}
  {X} h{g[f(X)]}
```

Listing 2

```
\cs_set:Npn \__foo #1 {f(#1)}
\cs_set:Npn \__bar #1 {g[#1]}
\cs_set:Npn \__baz #1 {h\{#1\}}
\tl_set:Nn \l_tmpa_tl{X}
\erw_compose_c:nV{
  {__baz}{__bar}{__foo}}
  \l_tmpa_tl
\l_tmpa_tl h{g[f(X)]}
\erw_compose_c:nn{
  {__baz}{__bar}{__foo}}
  {X} h{g[f(X)]}
```

Listing 3

```
\cs_set:Npn \__foo #1 {f(#1)}
\cs_set:Npn \__bar #1 {g[#1]}
\cs_set:Npn \__baz #1 {h\{#1\}}
\seq_new:N\l_tmp_seq
\seq_put_right:Nn\l_tmp_seq{X}
\erw_compose_seq:nV{
  \__baz}\__bar}\__foo}}
\l_tmp_seq
\seq_item:Nn\l_tmp_seq{1}      X
\seq_item:Nn\l_tmp_seq{2}      f(X)
\seq_item:Nn\l_tmp_seq{3}      g[f(X)]
\seq_item:Nn\l_tmp_seq{4}      h{g[f(X)]}
```

Listing 4

```
\cs_set:Npn \__foo #1 {f(#1)}
\cs_set:Npn \__bar #1 {g[#1]}
\cs_set:Npn \__baz #1 {h\{#1\}}
\seq_new:N\l_tmp_seq
\seq_put_right:Nn\l_tmp_seq{X}
\erw_compose_seq_c:nV{
  \__baz}\__bar}\__foo}}
\l_tmp_seq
\seq_item:Nn\l_tmp_seq{1}      X
\seq_item:Nn\l_tmp_seq{2}      f(X)
\seq_item:Nn\l_tmp_seq{3}      g[f(X)]
\seq_item:Nn\l_tmp_seq{4}      h{g[f(X)]}
```

Listing 5

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

1.2 csutil

Listing 6

```
\ExplSyntaxOn \cs_set:Npn \__foo #1 {f(#1)}

\erw_apply:Nn\__foo{X}      f(X)
\ExplSyntaxOff
```

Listing 7

```
\ExplSyntaxOn \cs_set:Npn \__foo #1 {f(#1)}
\tl_set:Nn \l_tmpa_tl{X}
\erw_fold_set_par:n{Nf}
\erw_fold_apply_par:n{Nf}
\erw_fold:NV\__foo\l_tmpa_tl
\l_tmpa_tl f(X)
\cs_set:Npn\__bar #1 {g[#1]}
\erw_fold:cV\__bar\l_tmpa_tl
\l_tmpa_tl g[f(X)]
\ExplSyntaxOff
```

Listing 8

```
\ExplSyntaxOn \erw_last_item:n{{a}{b}{c}}
c
\\erw_items_to:nn{1}{{a}{b}{c}{d}} a
\\erw_items_to:nn{2}{{a}{b}{c}{d}} ab
\\erw_items_to:nn{3}{{a}{b}{c}{d}} abc
\\erw_items_to:nn{4}{{a}{b}{c}{d}} abcd
\ExplSyntaxOff
```

Listing 9

```
\ExplSyntaxOn
\erw_repeat:nn{
  {3}{abracad}}abra abracadabracadabracadabra
\ExplSyntaxOff
```

Listing 10

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

1.3 map

Listing 11

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

Listing 12

```
\ExplSyntaxOn \cs_set:Npn \__foo #1 {(#1)}
\erw_map_inline:nn{
  (#1)}{{a}{b}{c}}          (a)(b)(c)
\ExplSyntaxOff
```

Listing 13

```
\cs_set:Npn \__foo:n #1 {(#1)}
\erw_map_thread:Nn \__foo:n
{
  {{a}{b}{c}{d}{e}{f}}
}
(a)(b)(c)(d)(e)(f)
\cs_set:Npn \__foo:nn #1 #2
  {(#1+#2)}
\erw_map_thread:Nn \__foo:nn
{
  {{a}{b}{c}{d}{e}{f}}
  {{A}{B}{C}{D}{E}{F}}
}
(a+A)(b+B)(c+C)(d+D)(e+E)(f+F)
\cs_set:Npn \__foo:nnn
  #1 #2 #3
  {(#1+#2+#3)}
\erw_map_thread:Nn \__foo:nnn
{
  {{a}{b}{c}{d}{e}{f}}
  {{A}{B}{C}{D}{E}{F}}
  {{k}{l}{m}{n}{o}{p}}
}
(a+A+k)(b+B+l)(c+C+m)(d+D+n)(e+E+o)(f+F+p)
\cs_set:Npn \__foo:nnnn
  #1 #2 #3 #4
  {(#1+#2+#3+#4)}
\erw_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}}
}
(a+A+k+K)(b+B+l+L)(c+C+m+M)(d+D+n+N)(e+E+o+O)(f+F+p+P)
```

1.4 numbrdcs

Listing 14

```
\NewDocumentCommand{\myfoo}{m}{f(#1)}
\NewDocumentCommand{\mybar}{m}{g[#1]}
\NewDocumentCommand{\mybaz}{m}{h\{#1\}}
\numbrdcsnew{\mybaz}{g[#1]}\myfoo}
\ExplSyntaxOn
\exp_last_unbraced:Nx
  \erw_compose_c:nn
  {
    {\erw_numbrd_cs_names
      _braced:nnn{1}{1}{3}}
    {X}
  }
\ExplSyntaxOff                                h{g[f(X)]}
```

2 Frontend

2.1 disambig

Listing 15

Input

```
\disambigset{my}
\disambignewcmd{\foo}{m}{#1~world!}
\noindent\myfoo{Hello}
\disambignewcmd*\foo{m}{#1~universe!}
\\myfoo{Hello}
\disambigunset
\disambignewcmd{\foo}{m}{#1~world!}
\\foo{Hello}
```

Output

```
Hello world!
Hello universe!
Hello world!
```

Listing 16

Input

```
\disambigset{my}
\disambignewenv{bar}{}{H}{!}
\\begin{mybar}ello~world\end{mybar}
\disambignewenv*{bar}{}{J}{!}
\\begin{mybar}ello~world\end{mybar}
```

Output

```
Hello world!
Jello world!
```

2.2 numbrdcs

Listing 17

```
\NewDocumentCommand{\thefoo}{m}{f(#1)}
\NewDocumentCommand{\thebar}{m}{g[#1]}
\NewDocumentCommand{\thebaz}{m}{h\{#1\}}
\numbrdcsnew{
  {\thefoo}
  {g[#1]}
  {\thebaz}}
\numbrdcs{1}{X}          f(X)
\numbrdcs{2}{X}          g[X]
\numbrdcs{3}{X}          h{X}
\numbrdcsnew*{
  {\thefoo}
  {g[#1]}
  {\thebaz}}
\numbrdcs{4}{X}          f(X)
\numbrdcs{5}{X}          g[X]
\numbrdcs{6}{X}          h{X}
```

Part III

Other

1 Support

This package is available from <https://www.ctan.org/pkg/erw-13> (release) or <https://github.com/er-cpp/erw-13> (development) where you can report issues.

2 Acknowledgment

I thank those that have answered my questions on forums pertaining to L^AT_EX3. See here: <https://tex.stackexchange.com/users/112708/erwann?tab=questions> and here: <https://latex.org/forum/memberlist.php?mode=viewprofile&u=61329>

References

- [1] The L^AT_EX3 Project Team *The L^AT_EX3 interfaces* <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf>
- [2] The L^AT_EX3 Project Team *The xparse package* <http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3packages/xparse.pdf>

Part IV

Implementation

```
1 \NeedsTeXFormat{LaTeX2e}
2 \RequirePackage{expl3}[2018/06/01]
3 \RequirePackage{xparse}[2018/02/01]
4 \RequirePackage{l3keys2e}
5 \ExplSyntaxOn
6 \msg_new:nnn{erw}{generic}{#1}
```

1 Back end

1.1 compose

```
7 \cs_set:Npn \erw_compose:NnV
8   #1 % method
9   #2 % funs
10  #3 % var
11  {
12   \erw_fold_set_par:n{Nf}
13   \erw_fold_apply_par:n{Nf}
14   \erw_cs_set_inline:Nn \__erw_map:n
15   {
16     #1{##1}#3
17   }
18   \exp_args:Nf\erw_map:n
19   {
20     \tl_reverse:n{#2}
21   }
22 }
23 \cs_set:Npn \erw_compose:nV #1 #2
24 {
25   \erw_compose:NnV \erw_fold:NV {#1} #2
26 }
27 \cs_set:Npn \erw_compose_c:nV #1 #2
28 {
```

```

29 \erw_compose:NnV \erw_fold:cV {#1} #2
30 }
31 \tl_new:N \__erw_compose_tl
32 \cs_set:Npn \erw_compose:nn #1 #2
33 {
34   \tl_set:Nn \__erw_compose_tl {#2}
35   \erw_compose:nV{#1}\__erw_compose_tl
36   \__erw_compose_tl
37 }
38 \cs_set:Npn \erw_compose:c:nn #1 #2
39 {
40   \tl_set:Nn \__erw_compose_tl {#2}
41   \erw_compose_c:nV{#1}\__erw_compose_tl
42   \__erw_compose_tl
43 }
44 \cs_set:Npn \erw_compose_seq:nV #1 #2
45 {
46   \erw_compose:NnV \erw_fold_seq:NV {#1} #2
47 }
48 \cs_set:Npn \erw_compose_seq_c:nV
49   #1 % funs
50   #2 % seq
51 {
52   \erw_compose:NnV \erw_fold_seq:cV {#1} #2
53 }
54 \cs_set:Npn \erw_compose_vers:nV #1 #2
55 {
56   \msg_error:nnn{erw}{generic}{erw_compose_vers:nV~yet-to-be-implemented}
57 }
58 \cs_set:Npn \erw_compose_seq_vers:nV #1 #2
59 {
60   \msg_error:nnn{erw}{generic}{erw_compose_vers:nV~yet-to-be-implemented}
61 }
62 \cs_set:Npn \erw_compose_vers:nn #1 #2
63 {
64   \erw_numbrd_cs_reset:{}
65   \tl_map_function:nN{#1}\erw_numbrd_cs_new:n
66   \exp_last_unbraced:Nx
67   \erw_compose_c:nn
68     {{\erw_numbrd_cs_names_braced:{}}}
69     {#2}
70 }

```

1.2 csutil

```

71 \cs_set:Npn \__erw_cs_name:N #1
72 {
73   \exp_last_unbraced:Nf \use_i:nnn {\cs_split_function:N #1}
74 }
75 \cs_set:Npn \erw_apply:Nn
76   #1 % fun
77   #2 % tl
78 {
79   #1{#2}
80 }

```

```

81 \cs_generate_variant:Nn \erw_apply:Nn {No, Nf, Nx, c}
82 \cs_set:Npn \erw_cs_set_eq:NN #1 #2
83 {
84   \cs_set:Npn #1 ##1{#2{##1}}
85 }
86 \cs_generate_variant:Nn \erw_cs_set_eq:NN {cN}
87 \cs_set:Npn \erw_cs_set_inline:Nn #1 #2
88 {
89   \cs_set:Npn #1 ##1{#2}
90 }
91 \cs_generate_variant:Nn \erw_cs_set_inline:Nn {cn}
92 \tl_set:Nn \__erw_fold_set_par_tl{c_novalue_tl}
93 \tl_set:Nn \__erw_fold_apply_par_tl{c_novalue_tl}
94 \cs_set:Npn \erw_fold_set_par:n #1
95 {
96   \tl_set:Nn \__erw_fold_set_par_tl{#1}
97 }
98 \cs_set:Npn \erw_fold_apply_par:n #1
99 {
100  \tl_set:Nn \__erw_fold_apply_par_tl{#1}
101 }
102 \cs_set:Npn \erw_fold:NV
103   #1 % fun
104   #2 % var
105 {
106   \use:c{tl_set:\__erw_fold_set_par_tl}
107     #2
108     {\use:c{erw_apply:\__erw_fold_apply_par_tl}{#1}{#2}}
109 }
110 \cs_generate_variant:Nn \erw_fold:NV {cV}
111 \tl_new:N \__erw_fold_seq_item_tl
112 \cs_set:Npn \erw_fold_seq:NV
113   #1 % fun
114   #2 % seq
115 {
116   \seq_get_right:NN #2 \__erw_fold_seq_item_tl
117   \erw_fold:NV #1 \__erw_fold_seq_item_tl
118   \seq_put_right:No #2 {\__erw_fold_seq_item_tl}
119 }
120 \cs_generate_variant:Nn \erw_fold_seq:NV {cV}
121 \cs_set:Npn \erw_identity:n #1{#1}
122 \cs_set:Npn \__erw_items_to:nnn #1 #2 #3
123 {
124   \int_compare:nNnTF
125     {#1}>{#2}
126     {
127       \exp_args:Nf \tl_head:n{#3}
128       \__erw_items_to:nnn
129         {#1}
130         {\int_eval:n{#2+1}}
131         {\exp_args:Nf \tl_tail:n{#3}}
132     }
133     {
134       \exp_args:Nf \tl_head:n{#3}

```

```

135     }
136 }
137 \cs_set:Npn \erw_items_to:nn #1 #2
138 {
139     \__erw_items_to:nnn
140     {#1}
141     {1}
142     {#2}
143 }
144 \cs_set:Npn \erw_last_item:n #1
145 {
146     \exp_args:Nof \tl_item:nn
147     {#1}
148     {
149         \tl_count:n{#1}
150     }
151 }
152 \cs_set:Npn \erw_repeat:nn #1 #2
153 {
154     \int_step_inline:nnnn{1}{1}{#1}{#2}
155 }
156 \cs_set:Npn \erw_split:nnn #1 #2 #3
157 {
158     \tl_head:n{#1}
159     \use:c{exp_args:#3} \tl_map_inline:nn
160     {
161         \tl_tail:n
162         {
163             #1
164         }
165     }{#2##1}
166 }
167 \cs_set:Npn \erw_split:nn #1 #2
168 {
169     \erw_split:nnn{#1}{#2}{Nf}
170 }

```

1.3 map

```

171 \cs_set:Npn \erw_map:n #1
172 {
173     \__erw_map:nn#1\q_recursion_tail\q_recursion_stop\q_recursion_tail\q_recursion_stop
174 }
175 \cs_set:Npn \__erw_map:nn #1 #2
176 {
177     \quark_if_recursion_tail_stop:n{#1}
178     \__erw_map:n{#1} \__erw_map:nn{#2}
179 }
180 \cs_new:Npn \__erw_map:n #1
181 {
182     \msg_error:nnn
183     {erw}
184     {generic}
185     {__erw_map:n~not~set}
186 }

```

```

187 \cs_set:Npn \erw_map:Nn
188   #1 % fun
189   #2 % tl
190 {
191   \erw_cs_set_eq:NN \__erw_map:n #1
192   \erw_map:n{#2}
193 }
194 \cs_set:Npn \erw_map_inline:nn
195   #1 % inl
196   #2 % tl
197 {
198   \erw_cs_set_inline:Nn \__erw_map:n {#1}
199   \erw_map:n{#2}
200 }
201 \cs_set:Npn \erw_apply:Nnn #1 #2 #3
202 {
203   #1{#2}{#3}
204 }
205 \cs_set:Npn \erw_apply:Nnnn #1 #2 #3 #4
206 {
207   #1{#2}{#3}{#4}
208 }
209 \cs_set:Npn \erw_apply:Nnnnn #1 #2 #3 #4 #5
210 {
211   #1{#2}{#3}{#4}{#5}
212 }
213
214 \cs_set:Npn \__erw_map_thread_at:Nnn #1 #2 #3
215 {
216   \erw_apply:Nn #1
217   {\exp_args:Nf\tl_item:nn {#3} {#2} }
218 }
219 \cs_set:Npn \__erw_map_thread_at:Nnnn #1 #2 #3 #4
220 {
221   \erw_apply:Nnn #1
222   {\exp_args:Nf\tl_item:nn {#3} {#2} }
223   {\exp_args:Nf\tl_item:nn {#4} {#2} }
224 }
225 \cs_set:Npn \__erw_map_thread_at:Nnnnn #1 #2 #3 #4 #5
226 {
227   \erw_apply:Nnnn #1
228   {\exp_args:Nf\tl_item:nn {#3} {#2} }
229   {\exp_args:Nf\tl_item:nn {#4} {#2} }
230   {\exp_args:Nf\tl_item:nn {#5} {#2} }
231 }
232 \cs_set:Npn \__erw_map_thread_at:Nnnnnn #1 #2 #3 #4 #5 #6
233 {
234   \erw_apply:Nnnnn #1
235   {\exp_args:Nf\tl_item:nn {#3} {#2} }
236   {\exp_args:Nf\tl_item:nn {#4} {#2} }
237   {\exp_args:Nf\tl_item:nn {#5} {#2} }
238   {\exp_args:Nf\tl_item:nn {#6} {#2} }
239 }
240 \cs_set:Npn \erw_map_thread_at:Nnn #1 #2 #3

```



```

241 {
242   \exp_args:Nf\int_case:nnTF
243   {
244     \tl_count:n{#3}
245   }
246   {
247     {1}{ \_erw_map_thread_at:Nnn #1{#2}#3 }
248     {2}{ \_erw_map_thread_at:Nnnn #1{#2}#3 }
249     {3}{ \_erw_map_thread_at:Nnnnn #1{#2}#3 }
250     {4}{ \_erw_map_thread_at:Nnnnnn #1{#2}#3 }
251   }
252   {
253     % Do nothing
254   }
255   {
256     \msg_error:nnn{erw}
257       {generic}
258       {erw_map_thread_at:~count~of~#3~not~withing~1~to~4}
259   }
260 }
261
262 \cs_set:Npn \erw_map_thread:Nn #1 #2
263 {
264   % TODO check that #2 is a matrix
265   \int_step_inline:nn
266   {
267     \exp_args:Nf \tl_count:n{ \tl_head:n{#2} }
268   }
269   {
270     \erw_map_thread_at:Nnn #1 {##1} {#2}
271   }
272 }

```

1.4 numbrdcs

```

273 \int_new:N \_erw_numbrd_cs_int
274 \cs_set:Npn \erw_numbrd_cs_name:n #1{\_erw_numbrd_cs\_int_to_alph:n{#1}:n}
275 \cs_set:Npn \erw_numbrd_cs_name_braced:n #1{{\erw_numbrd_cs_name:n{#1}}}
276 \tl_set:Nn \_erw_numbrd_cs_name_tl {\erw_numbrd_cs_name:n{\_erw_numbrd_cs_int}}
277 \cs_set:Npn \erw_numbrd_cs:nn #1 #2
278 {
279   \erw_apply:cn{\_erw_numbrd_cs\_int_to_alph:n{#1}:n}{#2}
280 }
281 \cs_new_protected:Npn \erw_numbrd_cs_reset:
282 {
283   \int_zero:N \_erw_numbrd_cs_int
284   \tl_set:Nn \_erw_numbrd_cs_ext_tl{}
285 }
286 \cs_new_protected:Npn \erw_numbrd_cs_new:n #1
287 {
288   \int_incr:N \_erw_numbrd_cs_int
289   \erw_cs_set_inline:cn{\_erw_numbrd_cs_name_tl}
290   {
291     \token_if_cs:NTF
292     {#1}

```

```

293         {#1{##1}}
294         {#1}
295     }
296 }
297 \cs_new:Npn \erw_numbrd_cs_names:nnn #1 #2 #3
298 {
299     \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_numbrd_cs_name:n
300 }
301 \cs_new:Npn \erw_numbrd_cs_names_braced:nnn #1 #2 #3
302 {
303     \int_step_function:nnnN { #1 }{ #2 }{ #3 } \erw_numbrd_cs_name_braced:n
304     % TODO \tl_range_braced:nnn?
305 }
306 \cs_new:Npn \erw_numbrd_cs_names_braced:
307 {
308     \erw_numbrd_cs_names_braced:nnn{1}{1}{\__erw_numbrd_cs_int}
309 }

```

2 frontend

2.1 disambig

```

310 \cs_set:Npn \__erw_disambig:NN #1 #2 {#1{#2}}
311 \cs_generate_variant:Nn \__erw_disambig:NN { Nc }
312 \NewDocumentCommand{\disambignewcmd}{ s m m m }
313 {
314     \msg_error:nnn{erw}{generic}{disambignewcmd~undefined}
315 }
316 \NewDocumentCommand{\disambignewenv}{ s m m m m }
317 {
318     \msg_error:nnn{erw}{generic}{disambignewenv~undefined}
319 }
320 \keys_define:nn { erw }
321 {
322     disambig .code:n =
323     {
324         \RenewDocumentCommand{\disambignewcmd}{ s m m m }
325         {
326             \IfBooleanTF{##1}
327                 {\__erw_disambig:Nc{\RenewDocumentCommand}}
328                 {\__erw_disambig:Nc{\NewDocumentCommand}}
329                 {#1 \__erw_cs_name:N ##2}
330                 {##3}
331                 {##4}
332         }
333         \RenewDocumentCommand{\disambignewenv}{ s m m m m }
334         {
335             \IfBooleanTF{##1}
336                 {\RenewDocumentEnvironment}
337                 {\NewDocumentEnvironment}
338                 {#1##2}
339                 {##3}
340                 {##4}
341                 {##5}

```

```

342 }
343 },
344 disambig .initial:n = \c_empty_tl
345 }
346 \NewDocumentCommand{\disambigset}{ m }
347 {
348   \keys_set:nn { erw }
349   {
350     disambig={#1}
351   }
352 }
353 \NewDocumentCommand{\disambigunset}{}
354 {
355   \disambigset{\c_empty_tl}
356 }

```

2.2 numbrdcs

```

357 \NewDocumentCommand{\numbrdcsnew}{ s m }
358 {
359   \IfBooleanTF{#1}
360     {}
361     { \erw_numbrd_cs_reset:{} }
362   \tl_map_function:nN {#2}\erw_numbrd_cs_new:n
363 }
364 \NewDocumentCommand{\numbrdcs}{ m m }
365 {
366   \erw_numbrd_cs:nn{#1}{#2}
367 }
368 % \ProcessKeysPackageOptions{ erw }
369 \ExplSyntaxOff

```

Change History

0.1	General: Initial version	12	mathematical convention ($g \circ f$ means f comes before g)	12
0.1.1	General:	12	<code>disambig</code> : pushed the code inside <code>\keys_define</code> ; <code>\disambignewcmd</code> no longer takes a token name as arg, rather a token.	12
	<code>\numbrdcsnew</code> changed to <code>\newnumbrdcs</code> and made 'disambiguable'	12	Added <code>\erw_items_to</code>	12
	<code>disambig/backend</code> : changes to the key, added <code>\ProcessPackageKeysOption</code> ; . . .	12	Added <code>\erw_last_item</code>	12
	Brought all the modules under one file; renamed <code>l3erw</code> to <code>erw-l3</code> ; . . .	12	Added <code>\erw_repeat</code>	12
			Added <code>\erw_split</code>	12
0.1.2	General:	12	Added <code>\map_thread</code>	12
	<code>\erw_compose</code> reversed order in which the functions are composed, such that it now conforms to the		Front end cmds no longer generated with module <code>disambig</code> ; Option of the same name deleted;	12
			Re-arranged the doc to clearly separate frontend from backend . .	12

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.

- C**
- cs commands:
- `\cs_generate_variant:Nn` 81, 86, 91, 110, 120, 311
 - `\cs_new:Npn` 180, 297, 301, 306
 - `\cs_new_protected:Npn` 281, 286
 - `\cs_set:Npn` 7, 23, 27, 32, 38, 44, 48, 54, 58, 62, 71, 75, 82, 84, 87, 89, 94, 98, 102, 112, 121, 122, 137, 144, 152, 156, 167, 171, 175, 187, 194, 201, 205, 209, 214, 219, 225, 232, 240, 262, 274, 275, 277, 310
 - `\cs_split_function:N` 73
- D**
- `\disambignewcmd` 5, 19, 312, 324
 - `\disambignewcmd*` 5
 - `\disambignewenv` 5, 316, 333
 - `\disambignewenv*` 5
 - `\disambigset` 5, 346, 355
 - `\disambigunset` 5, 353
- E**
- erw commands:
- `\erw_apply:Nn` ... 3, 3, 75, 81, 216, 279
 - `\erw_apply:Nnn` 201, 221
 - `\erw_apply:Nnnn` 205, 227
 - `\erw_apply:Nnnnn` 209, 234
 - `\erw_compose` 1, 19
 - `\erw_compose:nn` 3, 3, 23, 32, 35
 - `\erw_compose:Nnn` 7, 25, 29, 46, 52
 - `\erw_compose_c:nn` ... 3, 27, 38, 41, 67
 - `\erw_compose_seq:nn` 3, 44
 - `\erw_compose_seq_c:nn` 3, 48
 - `\erw_compose_seq_vers:nn` 3, 58
 - `\erw_compose_vers:nn` 3, 54, 62
 - `\erw_cs_set_eq:NN` 3, 82, 86, 191
 - `\erw_cs_set_inline:Nn` 3, 14, 87, 91, 198, 289
 - `\erw_fold:Nn` .. 3, 25, 29, 102, 110, 117
 - `\erw_fold_apply_par:n` 13, 98
 - `\erw_fold_seq:Nn` 46, 52, 112, 120
 - `\erw_fold_set_par:n` 12, 94
 - `\erw_identity:N` 3
 - `\erw_identity:n` 121
 - `\erw_items_to` 19
 - `\erw_items_to:nn` 3, 137
 - `\erw_last_item` 19
 - `\erw_last_item:n` 144
 - `\erw_last_item:nn` 3
 - `\erw_map:n` 18, 171, 192, 199
 - `\erw_map:Nn` 4, 187
 - `\erw_map_inline:nn` 4, 194
 - `\erw_map_thread:Nn` 4, 262
 - `\erw_map_thread_at:Nnn` ... 4, 240, 270
 - `\erw_numbrd_cs:nn` 4, 277, 366
 - `\erw_numbrd_cs_name:n` 274, 275, 276, 299
 - `\erw_numbrd_cs_name_braced:n` 275, 303
 - `\erw_numbrd_cs_names:nnn` 297
 - `\erw_numbrd_cs_names_braced:` 68, 306
 - `\erw_numbrd_cs_names_braced:nnn` 4, 301, 308
 - `\erw_numbrd_cs_new:n` . 4, 65, 286, 362
 - `\erw_numbrd_cs_reset:` 4, 64, 281, 361
 - `\erw_repeat` 19
 - `\erw_repeat:nn` 4, 152
 - `\erw_split` 19
 - `\erw_split:nn` 4, 167
 - `\erw_split:nnn` 156, 169
- erw internal commands:
- `__erw_compose_tl` 31, 34, 35, 36, 40, 41, 42
 - `__erw_cs_name:N` 71, 329
 - `__erw_disambig:NN` . 310, 311, 327, 328
 - `__erw_fold_apply_par_tl` 93, 100, 108
 - `__erw_fold_seq_item_tl` 111, 116, 117, 118
 - `__erw_fold_set_par_tl` ... 92, 96, 106
 - `__erw_items_to:nnn` ... 122, 128, 139
 - `__erw_map:n` ... 14, 178, 180, 191, 198
 - `__erw_map:nn` 173, 175, 178
 - `__erw_map_thread_at:Nnn` ... 214, 247
 - `__erw_map_thread_at:Nnnn` .. 219, 248
 - `__erw_map_thread_at:Nnnnn` . 225, 249
 - `__erw_map_thread_at:Nnnnnn` 232, 250
 - `__erw_numbrd_cs_ext_tl` 284
 - `__erw_numbrd_cs_int` 273, 276, 283, 288, 308
 - `__erw_numbrd_cs_name_tl` ... 276, 289
- exp commands:
- `\exp_args:Nf` 18, 127, 131, 134, 217, 222, 223, 228, 229, 230, 235, 236, 237, 238, 242, 267
 - `\exp_args:Nof` 146
 - `\exp_last_unbraced:Nn` 66, 73

<code>\ExplSyntaxOff</code>	369		
<code>\ExplSyntaxOn</code>	5		
I			
<code>\IfBooleanTF</code>	326, 335, 359		
int commands:			
<code>\int_case:nnTF</code>	242		
<code>\int_compare:nNnTF</code>	124		
<code>\int_eval:n</code>	130		
<code>\int_incr:N</code>	288		
<code>\int_new:N</code>	273		
<code>\int_step_function:nnnN</code>	299, 303		
<code>\int_step_inline:nn</code>	265		
<code>\int_step_inline:nnn</code>	154		
<code>\int_to_alph:n</code>	274, 279		
<code>\int_zero:N</code>	283		
K			
keys commands:			
<code>\keys_define</code>	19		
<code>\keys_define:nn</code>	320		
<code>\keys_set:nn</code>	348		
M			
map commands:			
<code>\map_thread</code>	19		
msg commands:			
<code>\msg_error:nnn</code>	56, 60, 182, 256, 314, 318		
<code>\msg_new:nnn</code>	6		
N			
<code>\NeedsTeXFormat</code>	1		
<code>\NewDocumentCommand</code>	4,		
5, 312, 316, 328, 346, 353, 357, 364			
<code>\NewDocumentEnvironment</code>	5, 337		
<code>\newnumbrdcs</code>	19		
<code>\numbrdcs</code>	5, 364		
<code>\numbrdcsnew</code>	5, 19, 357		
<code>\numbrdcsnew*</code>	5		
P			
<code>\ProcessKeysPackageOptions</code>	368		
<code>\ProcessPackageKeysOption</code>	19		
Q			
quark commands:			
<code>\quark_if_recursion_tail_stop:n</code>	177		
<code>\q_recursion_stop</code>	173		
<code>\q_recursion_tail</code>	173		
R			
<code>\RenewDocumentCommand</code> ..	5, 324, 327, 333		
<code>\RenewDocumentEnvironment</code>	5, 336		
<code>\RequirePackage</code>	2, 3, 4		
S			
seq commands:			
<code>\seq_get_right:NN</code>	116		
<code>\seq_put_right:Nn</code>	118		
T			
tl commands:			
<code>\c_empty_tl</code>	344, 355		
<code>\c_novalue_tl</code>	92, 93		
<code>\tl_count:n</code>	149, 244, 267		
<code>\tl_function_map:Nn</code>	4		
<code>\tl_head:n</code>	127, 134, 158, 267		
<code>\tl_item:nn</code>	146, 217, 222,		
223, 228, 229, 230, 235, 236, 237, 238			
<code>\tl_map_function:nN</code>	4, 65, 362		
<code>\tl_map_inline:nn</code>	159		
<code>\tl_new:N</code>	31, 111		
<code>\tl_range_braced:nnn</code>	304		
<code>\tl_reverse:n</code>	20		
<code>\tl_set:Nn</code>			
.... 34, 40, 92, 93, 96, 100, 276, 284			
<code>\tl_tail:n</code>	131, 161		
token commands:			
<code>\token_if_cs:NTF</code>	291		
U			
use commands:			
<code>\use:N</code>	106, 108, 159		
<code>\use_i:nnn</code>	73		
<code>\usepackage</code>	2		