oops, an object oriented practical scribe's package.*

Erwann Rogard†

Released 2020/04/10

Abstract

oops is a package for LaTeX (hence "scribe") for generating macro definitions as the need arises in the document, and to organize them along two dimensions: functions and objects, hence "OO". This is done using a minimalist interface built upon xparse[4]. Specifically, `\Oops<`⟨*object*⟩`>` begins a series of instructions alternating between 'text' and definitions, that themselves optionally expand using predefined or inline rules. For example,

```
\Oops<Math>[Let~]{Space=\Omega}*[~denote the sample space]{}
```

expands to: "Let Ω denote the sample space". As a side effect, `$\Space<Math>$` encodes"Ω". `Math` being the default for ⟨*object*⟩, it can be dropped. Optionally, the definitions can be written to a file, and restored, which can be useful for typesetting documents sharing the same notational conventions. Altogether, "practical".

Contents

I Usage 3

0 Convention 3

1 Loading the package 3

2 \Oops 3
- 2.1 `<`⟨tl_1⟩`>` . . . 4
- 2.2 `[`⟨tl_2⟩`]` . . . 4
- 2.3 `i{`⟨$code_1$⟩`}` . . . 4
- 2.4 `{`⟨kvl_1⟩`}` . . . 4
- 2.5 `+` . . . 4
- 2.6 `*` . . . 4
- 2.7 `s{`⟨tl_3⟩`}|{`⟨tl_3⟩`}{`⟨tl_4⟩`}|{`⟨tl_3⟩`}{`⟨tl_4⟩`}{`⟨tl_5⟩`}` . . . 4
- 2.8 `o{`⟨$code_2$⟩`}` . . . 5
- 2.9 `[`⟨tl_6⟩`]` . . . 5

3 \OopsClear 5

4 \OopsDebug 5

*This file describes version v1.5, last revised 2020/04/10.

†firstname dot lastname AusTria gmail dot com

5 \OopsHook 5

6 \OopsOption 5
6.1 Expans 5
6.2 File 5
6.3 Inner 5
6.4 Name 5
6.5 Outer 5
6.6 Separ 6
6.7 Write 6

7 \OopsRead 6

8 Do's and dont's 6

II Listing 7

Listing 1. 7

Listing 2. 7

Listing 3. 8

Listing 4. 8

Listing 6. 8

Listing 7. 9

Listing 8. 9

III Other 10

1 Acknowledgment 10

2 Also see 10

3 Install 10

4 Issue 10

5 Support 10

6 Testing 10

Change History 12

Index 12

IV Implementation 15
1 aux 15
2 log 17
3 make_key 19
4 make_oops 19
5 msg 21
6 option 21
7 prop 22
8 seq 24
9 Front-end 24
10 Misc 26

Part I
Usage

Convention

1. Loosely, those of [3] and [4], for example as to the meaning of ⟨*token list*⟩.
2. If unspecified, the environment in which a function must be declared is `document`.
3. Where ⟨tl_1⟩ is an optional argument, its default is `Math`.

`\usepackage` `\usepackage{oops}`

Environment *preamble*

Requirement `oops.sty` is in the path of the LaTeX engine. See Part III, section 5.

\Oops

```
\Oops<⟨tl1⟩>
[⟨tl2⟩]
i{⟨code1⟩}
{⟨kvl1⟩}
+
*
s{{⟨tl3⟩}|{⟨tl3⟩}{⟨tl4⟩}|{⟨tl3⟩}{⟨tl4⟩}{⟨tl5⟩}}
o{⟨code2⟩}
[⟨tl6⟩]
```

Requirement ⟨*kvl*$_1$⟩ is specified (all others optional).

⟨*tl*$_1$⟩

Example `Math`, `ModelA`, `ModelB`

Semantics Registers a new object, if applicable

⟨*tl*$_2$⟩

Example `Let~`

Semantics Expands ⟨*tl*$_2$⟩

⟨*code*$_1$⟩

Example `\mathbb{#1}`

Semantics 1. ⟨*val*$_i$⟩ ← ⟨*code*$_1$⟩ applied to ⟨*val*$_i$⟩

⟨*kvl*$_1$⟩

Example `Elems={\omega_1, \dots, \omega_n}, Sample=\Omega`

Semantics 2. `\`⟨*key*$_i$⟩`<`⟨*tl*$_1$⟩`>` ← ⟨*val*$_i$⟩ defined in step 1, using subsection 6.2 for expansion.

3. If `Write=\BooleanTrue`, writes the definitions made in step 2 to file `oops`⟨*digits*⟩`.tex`, where ⟨*digits*⟩=`\pdfcreationdate`

`+`

Other Needed to make `\Oops`'side effect within a *local group* persist thereafter

Semantics Appends step 2 and step 3 to `\OopsHook`

`*`

Semantics 5. Expands ⟨*code*$_2$⟩ applied to the list created in step 1, using the separator specified by subsubsection 2.7.1, subsubsection 2.7.2, subsubsection 2.7.3.

⟨*tl*$_3$⟩

Example `{~\in~}`

⟨*tl*$_4$⟩

Example `{,~}`

⟨*tl*$_5$⟩

Example `{~\&~}`

⟨*code*$_2$⟩

Example `$\left\{#1\right\}$`

⟨*tl*$_6$⟩

Semantics `\Oops<`⟨*tl*$_1$⟩`>[`⟨*tl*$_6$⟩`]`

`\OopsClear` `\OopsClear<`⟨*keyval list*⟩`>`

Semantics Clears any data created by `\Oops{`⟨*tl*$_1$⟩`}`, for all ⟨*tl*$_1$⟩ in ⟨*keyval list*⟩

`\OopsDebug`

Semantics See Part IV

`\OopsHook` `\OopsHook`

Example `\AfterEndEnvironment{theorem}{\OopsHook}`

`\OopsOption` `\OopsOption{`⟨*kvl0*⟩`}`

Semantics Set default options for section 2

`Expans`

Default `xo`

Syntax Either of `eo, ee, ex, xe, xo, xe, xx`

`File`

Syntax *token*

`Inner`

Semantics Default for subsection 2.3

Syntax Use `####1` as the argument to be replaced

`Name`

Semantics Default for subsection 2.1

`Outer`

Semantics Default for subsection 2.8

Syntax Use `####1` as the argument to be replaced

`Separ`

Semantics Default for subsection 2.7

Syntax That of 'separators' in [3, Section 8 of l3seq]

`Write`

Syntax *Boolean*

`\OopsRead` `\OopsRead[`⟨*path*⟩`]`

Other The default for ⟨*path*⟩ is the last write-file (see ⟨kvl_1⟩)

Semantics 1. Reads the definitions in ⟨*path*⟩.
2. Writes to `oops.log`: 'read from ⟨*path*⟩'

Do's and dont's

1.

Don't: `\Oops{ A = a, B = b }[Hello, world!]`.
Do: `\Oops{ A = a, B = b }[Hello, world!]{}`, or
`\Oops{ A = a, B = b } Hello, world!`

2.

Don't: `$\`⟨key_i⟩`<x$`.
Do: `$\`⟨key_i⟩`{<}x$`

3.

Don't: `\Oops[`[a, b)`]`.
Do: `\Oops{{[}a, b{)}}`

4.

Don't: `\Oops{ F = \cal F }`.
Do: `\Oops{ F = \cal{F} }` or `\Oops{ F = \mathcal{F} }`

5. Also see Part III, section 4

Part II
Listing

Tip: To replicate the listings in a LaTeX document, use the same setting as that of the documentation portion of `oops.dtx` (`\documentclass`, `\usepackage`, `\newtcblisting`, ...), and remove any `^^A`.

Listing 1.

```
%     \OopsOption{
%     ^^A% spaces betw. inner and outer brackets matter!->
%     Separ={{\ \char`@\ }{\ \%\ }{\ \char`@\ }}
%     \Oops<Test>{ X = x, Y = y }*[\\]
%     { X = x, Y = y, Z = z }*[\\]
%     { X = x, Y = y }*s{{\ \&\ }}[\\]
%     { X = x, Y = y }*s{{\ \&\ }{,\ }}[\\]
%     { X = x, Y = y, Z = z }*s{{\ \&\ }}[\\]
%     { X = x, Y = y, Z = z }*s{{\ \&\ }{,\ }}[\\]
%     { X = x, Y = y, Z = z }*s{{\ \&\ }{,\ }{\ \&\ }}\\
%
```

$x \mathbin{@} y$
$x \mathbin{\%} y \mathbin{@} z$
$x \mathbin{\&} y$
$x \mathbin{\&} y$
$x \mathbin{\&} y \mathbin{\&} z$
$x, y \mathbin{\&} z$
$x, y \mathbin{\&} z$

Listing 2.

```
%     \OopsOption{ Separ = {{}{.}{.}}, Outer = {####1} }
%     \OopsOption{ Write = \BooleanTrue }
%     \Oops<Test>
%     { KeyA = {.}, KeyB = {!}, KeyC = {\%} }[]
%     { KeyD = {d}, KeyE = {\%} }[]i{\{#1\}}
%     { KeyF = {H}, KeyG = {e}, KeyH = {l} }*[]
%     { KeyI = {\%}, KeyJ = {\%}, KeyK = {\%} }[.\{l\}.\{o\}]
%     { KeyL = {l}, KeyM = {\char`[}, KeyN = {\char`]} }[]
%     { KeyO = {o}, KeyP = {\%}, KeyQ = {\%} }[{,\ }]
%     { KeyR = {w}, KeyS = {o}, KeyT = {r} }*s{{}{}{}}o{{\char`[}#1}[]
%     { KeyU = {\%}, KeyV = {\%}, KeyW = {\%} }[]
%     { KeyX = {\%}, KeyY = {\%}, KeyZ = {\KeyB<Test>} }\nobreak
%     \KeyL<Test>\KeyD<Test>\KeyZ<Test>\KeyN<Test>\\
%     \OopsOption{ Write = \BooleanFalse }
%
```

{H}.{e}.{l}.{l}.{o}, [world!]

Listing 3.

```
%     \OopsRead
%     \KeyF<Test>\KeyA<Test>\nobreak
%     \KeyG<Test>\KeyA<Test>\nobreak
%     \KeyH<Test>\KeyA<Test>\nobreak
%     \KeyH<Test>\KeyA<Test>\nobreak
%     {\{}\nobreak\KeyO<Test>{\}},{\ }\nobreak
%     \KeyM<Test>\KeyR<Test>\nobreak
%     \KeyO<Test>\nobreak
%     \KeyT<Test>\nobreak
%     \KeyL<Test>\nobreak
%     \KeyD<Test>\nobreak
%     \KeyZ<Test>\nobreak
%     \KeyN<Test>\nobreak
%
```

{H}.{e}.{l}.{l}.{o}, [world!]

Listing 4.

```
%     \Oops[We call~]{Elems={{\omega_{1}}, {\dots}, {\omega_{n}}}}*
%     [~the elementary events, and ]{Space=\Omega}
%     [\begin{equation*}\Space=(\Elems)\end{equation*}~the sample space.]
%     {}
%
```

We call $\omega_1, \ldots, \omega_n$ the elementary events, and

$$\Omega = (\omega_1, \ldots, \omega_n)$$

the sample space.

Listing 5.

```
%     \OopsOption{ Write = \BooleanTrue }
%     \Oops[Let~]
%     {Space=\Omega, Field=\mathcal{F}, Meas=\mathcal{P}}
%     *s{{,}}o{$\{#1\}$}
%     [~denote the probability space, where $\Field\subset 2^{\Space}$.]
%     {}
%     \OopsOption{ Write = \BooleanFalse }
%
```

Let $\{\Omega, \mathcal{F}, \mathcal{P}\}$ denote the probability space, where $\mathcal{F} \subset 2^{\Omega}$.

Listing 6.

```
%           \OopsRead \tab $\Omega$ $\Field$ $\Meas$
%
```

$$\Omega\ \mathcal{F}\ \mathcal{P}$$

Listing 7.

```
%       \OopsOption{ Write = \BooleanTrue }
%       \newtheorem{theorem}{Theorem}
%       \AfterEndEnvironment{theorem}{\OopsHook}
%       \Oops i{\mathbb{#1}}
%       { N = { N } , R = { R } }+[]
%       { Grad = { \operatorname{grad} } }+
%       [\begin{theorem}
%         [Mittelwertsatz f\"ur $n$ Variable]Es~sei~]
%         { OffMenge = {D}, Ci = {C^{1}}, Strecke = { [x_0,x] } }+
%         [$n\in\N$,~$\OffMenge\subseteq\N^n$ eine offene Menge und
    $f\in\Ci(\OffMenge,\R)$.
%         Dann gibt es auf jeder Strecke $\Strecke\subset\OffMenge$ einen
    Punkt $\xi\in\Strecke$,~]
%         { Steig = { \frac{ f(x)-f(x_0) }{ x-x_0 } }, Punkt = { \xi } }+
%         [so dass gilt
%         \begin{equation*}
%           \Steig = \Grad f(\Punkt)^{\top}
%         \end{equation*}
%       \end{theorem}]
%       {}
%       \OopsOption{ Write = \BooleanFalse }
%
```

Theorem 1 (Mittelwertsatz für n Variable) *Es sei $n \in \mathbb{N}$, $D \subseteq \mathbb{N}^n$ eine offene Menge und $f \in C^1(D, \mathbb{R})$. Dann gibt es auf jeder Strecke $[x_0, x] \subset D$ einen Punkt $\xi \in [x_0, x]$, so dass gilt*

$$\frac{f(x) - f(x_0)}{x - x_0} = \operatorname{grad} f(\xi)^\top$$

Listing 8.

```
%       \OopsRead \tab $\N$ $\R$ $\OffMenge$ $\Ci$ $\Strecke$
%
```

$$\mathbb{N}\ \mathbb{R}\ D\ C^1\ [x_0, x]$$

Part III
Other

1 Acknowledgment

This work has benefited from Q&A's from the LaTeXcommunity, see here: https://tex.stackexchange.com/users/112708/erwann?tab=questions. Specific references are made in Part IV. Listing 4 and Listing 5 are from [1]. Listing 7 is from tcolbox[5, 17.3].

2 Also see

1. The package cool[2]

3 Install

Compiling `oops.dtx` (under Unix, `$tex oops.dtx`) will generate `oops.sty` and `oops.pdf`

4 Issue

1. **Don't:** `Inner={\{####1\}`

   Symptom: `\OopsRead` fails

   Do: `Inner={\char‘{####1\char‘}}`

5 Support

This package is available from https://www.ctan.org/pkg/oops and https://github.com/rogard/oops.

6 Testing

It's not possible to check the expansion of a certain class of macros against predefined values[6]. Instead, one can check that Part II, as generated in section 3 on one's own machine, agrees with `bench.pdf` available at https://github.com/rogard/oops,

References

[1] A.N. Shiryaev *Probability* Springer, 1995

[2] Nick Setzer *The cool package*, 2005 https://www.ctan.org/pkg/cool

[3] The LaTeX3 Project Team *The LaTeX3 interfaces* http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf

[4] The LaTeX3 Project Team *The xparse package* http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3packages/xparse.pdf

[5] Thomas F. Sturm *The tcolorbox package* http://www.texdoc.net/texmf-dist/doc/latex/tcolorbox/tcolorbox.pdf

[6] https://tex.stackexchange.com/a/534100/112708

Change History

v1.0
General: Initial version 11
v1.1
General: Added: Save 11
Added: Listing 1., 2., 3., 4., 6., and 9. 11
Added:\OopsRestore 11
Added:\OopsTest 11
Deleted: Listing 1-5 from v1.0 . . . 11
Fixed: apparent anomaly in v1.0's Listing 4, see Listing 1 11
Replaced: \OopsOptions by \OopsOption . . 11
Replaced: {⟨kvl2⟩} by <kvl_2> given that option type G not recommended[4] 11
Replaced: GenericObject by Name 11
Replaced: Separators by Separ . . 11
Revamped: much of the implementation 11
v1.2
General: Deleted: \OopsTest 11
Deleted: ⟨kvl2⟩ and ⟨code2⟩ 11
Deleted: Listing 2-3 from v1.1. . . . 11
Replaced: \OopsClear{⟨tl1⟩} by \OopsClear[⟨keyval list⟩] 11
Replaced: \Restore by \Read 11
Replaced: \Save by \Write 11
v1.3
General: Replaced: \OopsNew by \Oops 11
Replaced: {⟨tl1⟩} and [⟨tl1⟩] by <⟨tl1⟩> 11
v1.4
General: Added: section 8 11
Added: \OopsDebug 11
Added: \OopsHook 11
Added: Expans (for debugging' sake, but...) 11
Added: Listing 1., 2., and 3. 11
Deleted: Listing 1., and 2. 11
Replaced: s{{⟨tl3⟩}{⟨tl4⟩}{⟨tl5⟩}} by s{{⟨tl3⟩}|{⟨tl3⟩}{⟨tl4⟩}|{⟨tl3⟩}{⟨tl4⟩}{⟨tl5⟩}} 11
v1.5
General: Added: File 11
Deleted: dependence on datetime . 11

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

* (option) *4*
+ (option) *4*
\⟨key$_i$⟩ *4*, *6*
⟨code$_1$⟩ (option) *4*
⟨code$_2$⟩ (option) *5*
⟨kvl$_1$⟩ (option) *4*
⟨tl$_1$⟩ (option) *4*
⟨tl$_2$⟩ (option) *4*
⟨tl$_3$⟩ (option) *4*
⟨tl$_4$⟩ (option) *4*
⟨tl$_5$⟩ (option) *5*
⟨tl$_6$⟩ (option) *5*
Expans (option) *5*
File (option) *5*
Inner (option) *5*
Name (option) *5*
Separ (option) *6*
Write (option) *6*

\␣ 371, 372

A

\AtEndDocument 87

B

\begin 289
\begingroup 172
bool commands:
\bool_gset_false:N 92
\bool_gset_true:N 99
\bool_if:nTF 114, 168, 192, 352
\bool_set_false:N 88
\BooleanFalse 357, 358
\BooleanTrue *4*

C

cs commands:
\cs_generate_variant:Nn 7, 31, 76, 106, 121, 131, 136, 254
\cs_gset:Npn 6, 24, 240
\cs_new:Nn . . 54, 77, 220, 268, 272, 293
\cs_new:Npn 276
\cs_new_protected:Nn . . 4, 13, 17, 22, 27, 41, 45, 81, 89, 95, 101, 107, 112, 122, 132, 137, 141, 153, 165, 212, 230, 238, 255, 259, 263, 277, 281
\cs_new_protected:Npn 9, 33, 179, 246
\cs_set:Nn 248
\cs_set_protected:Nn 283

D

\DeclareDocumentCommand 181
\def 173, 178
\documentclass *7*

E

\end 292
\endgroup 174
\ensuremath 349, 350
exp commands:
\exp_args:NNf 147
\exp_args:NNx 124, 181
\exp_args:No 158, 261
\exp_args:Nx 35
\exp_last_unbraced:Nf 311, 325, 339, 361
\exp_last_unbraced:NNf 295
\exp_not:N 72, 218, 228, 236, 244
\exp_not:n 202, 306
\expandafter 173
\ExplSyntaxOff 405
\ExplSyntaxOn 3

F

file commands:
\file_input:n 103

G

\gappto 194, 385

I

\IfBooleanT 188
\IfValueT 186, 200
\IfValueTF 401
int commands:
\int_case:nnTF 56
iow commands:
\iow_close:N 87, 91
\iow_new:N 86
\iow_now:Nn 116
\iow_open:Nn 98
\item 290, 291

K

\KeyA 384, 390
keys commands:
\l_keys_choice_tl 303
\keys_define:nn 299
\keys_set:nn 396

M

\meta 290, 291
msg commands:
\msg_error:nnn 119, 226
\msg_error:nnnn 70
\msg_new:nnn 206, 207, 208, 209, 210, 211
\msg_warning:nnn 218, 236, 244

N

\NeedsTeXFormat 2
\NewDocumentCommand . . . 374, 379, 393, 398
\newtcblisting *7*

O

\Oops *1*, *4*, *4*, *5*, *5*, *6*, *12*, 173, 181, 202
oops internal commands:
__oops_aux_inner:n . 6, 7, 38, 39, 212
__oops_aux_inner_set:n 4, 158
__oops_aux_key:N 17, 162
__oops_aux_key:n 13, 20
__oops_aux_key:w 9, 15
\g__oops_aux_key_seq 11, 19, 143, 163, 265, 266
\g__oops_aux_keyval_seq 159, 160, 162
__oops_aux_outer:n 24, 145
__oops_aux_outer_set:n 22, 144
\g__oops_aux_prop 26, 29, 47, 161
__oops_aux_prop:N 45, 160
__oops_aux_prop:n 41, 51
__oops_aux_prop:nn 27, 31
__oops_aux_prop:w 26, 43
__oops_aux_separ:n 77, 297
__oops_aux_separ:nn 54, 79
__oops_aux_val:Nn 81, 143
\g__oops_aux_val_seq 83, 84, 149
__oops_erw_seq_use:Nn 289
__oops_log_close: 86, 354
__oops_log_entry 173, 174
\g__oops_log_file_tl 94, 97, 306
\g__oops_log_iow 86, 87, 91, 98, 116, 119
__oops_log_open: 94, 353
\g__oops_log_open_bool 88, 92, 99, 114, 168
__oops_log_read: 107, 403
__oops_log_read:n 101, 109, 402

\g__oops_log_to_tl .. 97, 98, 109, 111
__oops_log_write:n 111, 170
__oops_make_key:N 137, 163
__oops_make_key:n 132, 139
__oops_make_key:Nn 122, 134, 384, 390
__oops_make_oops:nnnn .....
..... 178, 312, 326, 340, 362
__oops_make_oops_exp:nnn .. 141, 190
__oops_make_oops_key:nnn .. 153, 167
__oops_make_oops_sideeffect:nnn
..... 165, 187, 196
\g__oops_option_expans_tl 32, 36, 303
__oops_option_inner:n . 212, 216, 324
\g__oops_option_inner_tl .....
..... 214, 218, 315, 329, 343, 365
__oops_option_name:n 220, 310
\g__oops_option_name_tl .....
126, 222, 228, 314, 328, 342, 364, 375
__oops_option_outer:n 230, 338
\g__oops_option_outer_tl .....
..... 232, 236, 317, 331, 345, 367
__oops_option_separ:n 238, 360
\g__oops_option_separ_tl .....
..... 240, 244, 316, 330, 344, 366
__oops_prop_append:NN 246, 257
__oops_prop_append:Nn 161, 255
__oops_prop_append:nn 248, 252
__oops_prop_clear_new:n ... 259, 266
__oops_prop_clear_new_map:n 263, 377
__oops_prop_if_exist:nTF .....
..... 155, 268, 381, 387
__oops_prop_item:nn 128, 272
__oops_prop_name:n .....
..... 84, 257, 261, 270, 274, 276, 279
__oops_prop_new:n . 157, 277, 383, 389
__oops_seq_from_prop:n 283, 287
__oops_seq_from_prop:NNn ... 84, 281
__oops_seq_use:Nn 148, 293
\OopsClear *1*, *5*, *12*, 374
\OopsDebug *1*, *5*, *12*, 379
\OopsHook *2*, *4*, *5*, *12*, 178, 194, 385
\OopsNew *12*
\OopsOption *2*, *5*, *12*, 393
\OopsOptions *12*
\OopsRead *2*, *6*, *10*, 398
\OopsRestore *12*
\OopsTest *12*
options:
* *4*
+ *4*
⟨*code*$_1$⟩ *4*
⟨*code*$_2$⟩ *5*
⟨*kvl*$_1$⟩ *4*
⟨*tl*$_1$⟩ *4*
⟨*tl*$_2$⟩ *4*
⟨*tl*$_3$⟩ *4*
⟨*tl*$_4$⟩ *4*
⟨*tl*$_5$⟩ *5*
⟨*tl*$_6$⟩ *5*
Expans *5*
File *5*
Inner *5*
Name *5*
Separ *6*
Write *6*
Outer *5*
Outer (option) *5*

P

\pdfcreationdate 307, 308
prg commands:
\prg_replicate:nn 59
prop commands:
\prop_clear_new:N 261
\prop_gclear_new:N 47
\prop_gput:Nnn 29, 250
\prop_if_exist:NTF 270
\prop_item:Nn 250, 274, 285
\prop_map_function:NN 252
\prop_new:N 26, 279
\ProvideDocumentCommand 125

Q

quark commands:
\q_stop 9, 15, 33, 43

R

\Read *12*
\Restore *12*

S

\Save *12*
seq commands:
\seq_gclear_new:N 19, 83
\seq_gput_right:Nn 11, 285
\seq_if_empty:NTF 48
\seq_map_function:NN .....
..... 20, 51, 139, 266, 287
\seq_set_from_clist:Nn 159, 265
\seq_use:Nnnn 296

T

tl commands:
\c_empty_tl .....
.. 49, 68, 156, 176, 178, 199, 382, 388
\tl_count:n 79
\tl_gset:Nn 97, 214, 222, 232, 306
\tl_gset_eq:NN 303
\tl_log:n 104, 117

\tl_new:N 32, 94, 111
\tl_trim_spaces:n 11, 37, 38, 39
\use_i:nn 62, 64
\use_ii:nn 63
\usepackage 3, 7

U

use commands:
\use:N 36

W

\Write 12

Part IV
Implementation

```
1 ⟨@@=oops⟩
2 \NeedsTeXFormat{LaTeX2e}[2019/10/01]
3 \ExplSyntaxOn
```

1 aux

__oops_aux_inner_set:n #1 : ⟨*code*⟩

```
4 \cs_new_protected:Nn \__oops_aux_inner_set:n
5 {
6   \cs_gset:Npn \__oops_aux_inner:n ##1 {#1}
7   \cs_generate_variant:Nn \__oops_aux_inner:n { e }
8 }
```

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

__oops_aux_key:w #1 : ⟨ *key* ⟩
#2 : ⟨ *value* ⟩

```
9 \cs_new_protected:Npn \__oops_aux_key:w #1 = #2 \q_stop
10 {
11   \seq_gput_right:Nx \g__oops_aux_key_seq { \tl_trim_spaces:n{#1} }
12 }
```

(*End definition for* `\__oops_aux_key:w`.)

__oops_aux_key:n #1 : ⟨ *key* = *value* ⟩

```
13 \cs_new_protected:Nn \__oops_aux_key:n
14 {
15   \__oops_aux_key:w #1 \q_stop
16 }
```

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

__oops_aux_key:N #1 : ⟨ *seq* ⟩

```
17 \cs_new_protected:Nn \__oops_aux_key:N
18 {
19   \seq_gclear_new:N \g__oops_aux_key_seq
20   \seq_map_function:NN #1 \__oops_aux_key:n
21 }
```

(*End definition for* `\__oops_aux_key:N`.)

`\__oops_aux_outer_set:n` `#1` : ⟨ *inline code* ⟩

```
\cs_new_protected:Nn \__oops_aux_outer_set:n
{
  \cs_gset:Npn \__oops_aux_outer:n ##1 {#1}
}
```

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

`\__oops_aux_prop:nn`

```
\prop_new:N \g__oops_aux_prop
\cs_new_protected:Nn \__oops_aux_prop:nn
{
  \prop_gput:Nnn \g__oops_aux_prop{#1}{#2}
}
\cs_generate_variant:Nn \__oops_aux_prop:nn { eo, ee, ex, xo, xe, xx }
```

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

`\__oops_aux_prop:w` `#1` : ⟨ *key* ⟩
`#2` : ⟨ *value* ⟩

```
\tl_new:N \g__oops_option_expans_tl
\cs_new_protected:Npn \__oops_aux_prop:w #1 = #2 \q_stop
{
  \exp_args:Nx
  \use:c{__oops_aux_prop:\g__oops_option_expans_tl}
  { \tl_trim_spaces:n{#1} }
  { \__oops_aux_inner:n{ \tl_trim_spaces:n{#2} } }
%^^A  { \__oops_aux_inner:e{ \tl_trim_spaces:n{#2} } }% DEBUG
}
```

(*End definition for* `\__oops_aux_prop:w`.)

`\__oops_aux_prop:n` `#1` : ⟨ *key* = *value* ⟩

```
\cs_new_protected:Nn \__oops_aux_prop:n
{
  \__oops_aux_prop:w #1 \q_stop
}
```

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

`\__oops_aux_prop:N` `#1` : ⟨*keyval list*⟩

```
\cs_new_protected:Nn \__oops_aux_prop:N
{
  \prop_gclear_new:N \g__oops_aux_prop
  \seq_if_empty:NTF #1
  { \c_empty_tl }
  {
    \seq_map_function:NN #1 \__oops_aux_prop:n
  }
}
```

(*End definition for* `\__oops_aux_prop:N`.)

`\__oops_aux_separ:nn` `#1` : ⟨ *int* ⟩

#2 : ⟨ *tokens* ⟩

```
\cs_new:Nn \__oops_aux_separ:nn
{
  \int_case:nnTF {#1}
  {
    {1}
    { \prg_replicate:nn{ 3 }{#2} }
    {2}
    {
      { \use_i:nn #2 }
      { \use_ii:nn #2 }
      { \use_i:nn #2 }
    }
    {3}{#2}
  }
  { \c_empty_tl }
  {
    \msg_error:nnnn { __erw }
    { separ }
    { \exp_not:N \__oops_aux_separ:nn }
    {#2}
  }
}
\cs_generate_variant:Nn \__oops_aux_separ:nn { e }
```

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

`\__oops_aux_separ:n` #1 : ⟨ *tokens* ⟩

```
\cs_new:Nn \__oops_aux_separ:n
{
  \__oops_aux_separ:en{ \tl_count:n{#1} }{#1}
}
```

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

`\__oops_aux_val:Nn` #1 : ⟨ *seq* ⟩
#2 : ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__oops_aux_val:Nn
{
  \seq_gclear_new:N \g__oops_aux_val_seq
  \__oops_seq_from_prop:NNn \g__oops_aux_val_seq #1 { \__oops_prop_name:n{#2} }
}
```

(*End definition for* `\__oops_aux_val:Nn`.)

2 log

`\__oops_log_close:`

```
\iow_new:N \g__oops_log_iow
\AtEndDocument{\iow_close:N \g__oops_log_iow}
\bool_set_false:N \g__oops_log_open_bool
\cs_new_protected:Nn \__oops_log_close:
{
```

```
  \iow_close:N \g__oops_log_iow
  \bool_gset_false:N \g__oops_log_open_bool
}
```

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

`\__oops_log_open:`

```
\tl_new:N \g__oops_log_file_tl
\cs_new_protected:Nn \__oops_log_open:
{
  \tl_gset:Nx \g__oops_log_to_tl{\g__oops_log_file_tl}
  \iow_open:Nn \g__oops_log_iow {\g__oops_log_to_tl}
  \bool_gset_true:N \g__oops_log_open_bool
}
```

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

`\__oops_log_read:n` `#1` : ⟨*path*⟩

```
\cs_new_protected:Nn \__oops_log_read:n
{
  \file_input:n{#1}
  \tl_log:n{read~from~#1}
}
\cs_generate_variant:Nn \__oops_log_read:n { e }
```

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

`\__oops_log_read:`

```
\cs_new_protected:Nn \__oops_log_read:
{
  \__oops_log_read:e{\g__oops_log_to_tl}
}
```

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

`\__oops_log_write:n`

```
\tl_new:N \g__oops_log_to_tl
\cs_new_protected:Nn \__oops_log_write:n
{
  \bool_if:nTF{ \g__oops_log_open_bool }
  {
    \iow_now:Nn \g__oops_log_iow {#1}
    \tl_log:n{ write~to~#1 }
  }
  { \msg_error:nnn{ __oops }{ iow }{ \g__oops_log_iow }  }
}
\cs_generate_variant:Nn \__oops_log_write:n { e }
```

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

3 make_key

__oops_make_key:Nn
#1 : ⟨ *token* ⟩
#2 : ⟨ *key* ⟩

```
\cs_new_protected:Nn \__oops_make_key:Nn
{
  \exp_args:NNx
  \ProvideDocumentCommand{#1}
  { D<>{\g__oops_option_name_tl} }
  {
    \__oops_prop_item:nn{##1}{#2}
  }
}
\cs_generate_variant:Nn \__oops_make_key:Nn {c}
```

(*End definition for* __oops_make_key:Nn.)

__oops_make_key:n
#1 : ⟨ *key* ⟩

```
\cs_new_protected:Nn \__oops_make_key:n
{
  \__oops_make_key:cn{#1}{#1}
}
\cs_generate_variant:Nn \__oops_make_key:n { e }
```

(*End definition for* __oops_make_key:n.)

__oops_make_key:N
#1 : ⟨ *seq* ⟩

```
\cs_new_protected:Nn \__oops_make_key:N
{
  \seq_map_function:NN #1 \__oops_make_key:e
}
```

(*End definition for* __oops_make_key:N.)

4 make_oops

__oops_make_oops_exp:nnn

```
\cs_new_protected:Nn \__oops_make_oops_exp:nnn
{
  \__oops_aux_val:Nn \g__oops_aux_key_seq {#1}
  \__oops_aux_outer_set:n{#3}
  \__oops_aux_outer:n
  {
    \exp_args:NNf
    \__oops_seq_use:Nn
    \g__oops_aux_val_seq
    {#2}
  }
}
```

(*End definition for* __oops_make_oops_exp:nnn.)

__oops_make_oops_key:nnn

```
\cs_new_protected:Nn \__oops_make_oops_key:nnn
{
  \__oops_prop_if_exist:nTF{#1}
  { \c_empty_tl }
  { \__oops_prop_new:n{#1} }
  \exp_args:No \__oops_aux_inner_set:n{#2}
  \seq_set_from_clist:Nn \g__oops_aux_keyval_seq {#3}
  \__oops_aux_prop:N \g__oops_aux_keyval_seq
  \__oops_prop_append:Nn \g__oops_aux_prop {#1}
  \__oops_aux_key:N \g__oops_aux_keyval_seq
  \__oops_make_key:N \g__oops_aux_key_seq
}
```

(*End definition for* `\__oops_make_oops_key:nnn`.)

__oops_make_oops_sideeffect:nnn

```
\cs_new_protected:Nn \__oops_make_oops_sideeffect:nnn
{
  \__oops_make_oops_key:nnn{#1}{#2}{#3}
  \bool_if:nTF{ \g__oops_log_open_bool }
  {%^^A https://tex.stackexchange.com/questions/536597
    \__oops_log_write:n
    {
      \begingroup
      \def \__oops_log_entry { \Oops<#1>i{#2}{#3} } \expandafter
      \endgroup \__oops_log_entry
    }
  }{\c_empty_tl}
}
```

(*End definition for* `\__oops_make_oops_sideeffect:nnn`.)

__oops_make_oops:nnnn

#1 : ⟨ *token list* ⟩
#2 : ⟨ seq_1 ⟩
#3 : ⟨ seq_2 ⟩
#4 : ⟨ *prop* ⟩

```
\def\OopsHook{\c_empty_tl}
\cs_new_protected:Npn \__oops_make_oops:nnnn #1 #2 #3 #4
{
  \exp_args:NNx \DeclareDocumentCommand \Oops
  {%^^A      2    3         4 5  6    7 8              9
    D<>{#1} +o E{ i }{{#2}} m t+ s E{ s o }{{#3}{#4}} +o
  }
  {
    \IfValueT{##2}{##2}
    \__oops_make_oops_sideeffect:nnn{##1}{##3}{##4}
    \IfBooleanT{##6}
    {
      \__oops_make_oops_exp:nnn{##1}{##7}{##8}
    }
    \bool_if:nTF{##5}
    {
      \gappto{\OopsHook}
```

```
      {
        \__oops_make_oops_sideeffect:nnn{##1}{##3}{##4}
      }
    }
    {\c_empty_tl}
    \IfValueT{##9}
    {
      \exp_not:n{ \Oops<##1>[##9] }
    }
  }
}
```

(*End definition for* `\__oops_make_oops:nnnn`.)

5 msg

```
\msg_new:nnn {__oops}{ generic }{#1}
\msg_new:nnn {__oops}{ iow }{#1~is~closed~can't~write}
\msg_new:nnn {__oops}{ keyonly }{#1~does~not~take~values;~keyval~is~#2}
\msg_new:nnn {__oops}{ keywrong }{#1~does~not~recognize~key~#2}
\msg_new:nnn {__oops}{ separ }{#1~expects~1~to~3~items,~#2}
\msg_new:nnn {__oops}{ unset }{#1~unset}
```

6 option

`\__oops_aux_inner:n` `#1` : ⟨*code*⟩

```
\cs_new_protected:Nn \__oops_option_inner:n
{
  \tl_gset:Nn \g__oops_option_inner_tl {#1}
}
\__oops_option_inner:n
{
  \msg_warning:nnn{ __oops }{ unset }{ \exp_not:N \g__oops_option_inner_tl }
}
```

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

`\__oops_option_name:n` `#1` : ⟨*token list*⟩

```
\cs_new:Nn \__oops_option_name:n
{
  \tl_gset:Nn \g__oops_option_name_tl{#1}
}
\__oops_option_name:n
{
  \msg_error:nnx{ __oops }
  { generic }
  { \exp_not:N\g__oops_option_name_tl~undefined }
}
```

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

`\__oops_option_outer:n` `#1 :` ⟨ *inline code* ⟩

```
\cs_new_protected:Nn \__oops_option_outer:n
{
  \tl_gset:Nn \g__oops_option_outer_tl {#1}
}
\__oops_option_outer:n
{
  \msg_warning:nnn{ __oops }{ unset }{ \exp_not:N \g__oops_option_outer_tl }
}
```

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

`\__oops_option_separ:n` `#1 :` {⟨ tl_1 ⟩}{⟨ tl_2 ⟩}{⟨ tl_3 ⟩}

```
\cs_new_protected:Nn \__oops_option_separ:n
{
  \cs_gset:Npn \g__oops_option_separ_tl {#1}
}
\__oops_option_separ:n
{
  \msg_warning:nnn{ __oops }{ unset }{ \exp_not:N \g__oops_option_separ_tl }
}
```

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

7 prop

`\__oops_prop_append:NN` `#1 :` ⟨ $prop_1$ ⟩
`#2 :` ⟨ $prop_2$ ⟩

```
\cs_new_protected:Npn \__oops_prop_append:NN #1 #2
{
  \cs_set:Nn \__oops_prop_append:nn
  {
    \prop_gput:Nnx #1 {##1}{ \prop_item:Nn #2{##1} }
  }
  \prop_map_function:NN #2 \__oops_prop_append:nn
}
\cs_generate_variant:Nn \__oops_prop_append:NN { cN }
```

(*End definition for* `\__oops_prop_append:NN`.)

`\__oops_prop_append:Nn` `#1 :` ⟨ *prop* ⟩
`#2 :` ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__oops_prop_append:Nn
{
  \__oops_prop_append:cN{ \__oops_prop_name:n {#2} } #1
}
```

(*End definition for* `\__oops_prop_append:Nn`.)

`\__oops_prop_clear_new:n` `#1 :` ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__oops_prop_clear_new:n
{
  \exp_args:No \prop_clear_new:c{ \__oops_prop_name:n {#1} }
}
```

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

`\__oops_prop_clear_new_map:n` `#1` : ⟨ *keyval list* ⟩

```
\cs_new_protected:Nn \__oops_prop_clear_new_map:n
{
  \seq_set_from_clist:Nn \g__oops_aux_key_seq {#1}
  \seq_map_function:NN \g__oops_aux_key_seq \__oops_prop_clear_new:n
}
```

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

`\__oops_prop_if_exist:nTF` `#1` : ⟨tl_1⟩
`#2` : ⟨tl_2⟩
`#3` : ⟨tl_3⟩

```
\cs_new:Nn \__oops_prop_if_exist:nTF
{
  \prop_if_exist:cTF{ \__oops_prop_name:n {#1} }{#2}{#3}
}
```

(*End definition for* `\__oops_prop_if_exist:nTF`.)

`\__oops_prop_item:nn` `#1` : ⟨ *tl var name* ⟩
`#2` : ⟨ *key* ⟩

```
\cs_new:Nn \__oops_prop_item:nn
{
  \prop_item:cn { \__oops_prop_name:n {#1} } {#2}
}
```

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

`\__oops_prop_name:n` `#1` : ⟨ *tl var name* ⟩

```
\cs_new:Npn \__oops_prop_name:n #1{ __oops_#1 }
```

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

`\__oops_prop_new:n` `#1` : ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__oops_prop_new:n
{
  \prop_new:c{ \__oops_prop_name:n {#1} }
}
```

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

8 seq

__oops_seq_from_prop:NNn

#1 : ⟨ *seq*$_1$ ⟩
#2 : ⟨ *seq*$_2$ ⟩ (keys)
#3 : ⟨ *prop* ⟩

```
\cs_new_protected:Nn \__oops_seq_from_prop:NNn
{
  \cs_set_protected:Nn \__oops_seq_from_prop:n
  {
    \seq_gput_right:No #1 { \prop_item:cn{#3}{##1} }
  }
  \seq_map_function:NN #2 \__oops_seq_from_prop:n
}
```

(*End definition for* `\__oops_seq_from_prop:NNn`.)

__oops_erw_seq_use:Nn

```
%      \begin{arguments}
%      \item \meta{ seq }
%      \item \meta{ tokens }
%      \end{arguments}
\cs_new:Nn \__oops_seq_use:Nn
{
  \exp_last_unbraced:NNf
  \seq_use:Nnnn #1
  \__oops_aux_separ:n{#2}
}
```

(*End definition for* `\__oops_erw_seq_use:Nn`.)

9 Front-end

```
\keys_define:nn { __oops }
{
  Expans .multichoices:nn =
  { eo, ee, ex, xo, xe, xx }
  { \tl_gset_eq:NN \g__oops_option_expans_tl \l_keys_choice_tl },
  Expans .default:n = { xo },
  Expans .initial:n = { xo },
  File .code:n = { \tl_gset:Nn  \g__oops_log_file_tl{ \exp_not:n{ #1 } } },
  File .default:n = { oops\pdfcreationdate },
  File .initial:n = { oops\pdfcreationdate },
  Name .code:n={
    \__oops_option_name:n{#1}
    \exp_last_unbraced:Nf
    \__oops_make_oops:nnnn
    {
      { \g__oops_option_name_tl }
      { \g__oops_option_inner_tl }
      { \g__oops_option_separ_tl }
      { \g__oops_option_outer_tl }
    }
  },
```

```
  Name .value_required:n = false,
  Name .default:n = { Math },
  Name .initial:n = { Math },
  Inner .code:n={
    \__oops_option_inner:n{#1}
    \exp_last_unbraced:Nf
    \__oops_make_oops:nnnn
    {
      { \g__oops_option_name_tl }
      { \g__oops_option_inner_tl }
      { \g__oops_option_separ_tl }
      { \g__oops_option_outer_tl }
    }
  },
  Inner .value_required:n = false,
  Inner .default:n = {####1},
  Inner .initial:n = {####1},
  Outer .code:n={
    \__oops_option_outer:n{#1}
    \exp_last_unbraced:Nf
    \__oops_make_oops:nnnn
    {
      { \g__oops_option_name_tl }
      { \g__oops_option_inner_tl }
      { \g__oops_option_separ_tl }
      { \g__oops_option_outer_tl }
    }
  },
  Outer .value_required:n = false,
  Outer .default:n = { \ensuremath{####1} },
  Outer .initial:n = { \ensuremath{####1} },
  Write .code:n = {
    \bool_if:nTF{#1}
    {\__oops_log_open:}
    {\__oops_log_close:}
  },
  Write .value_required:n = false,
  Write .default:n = \BooleanFalse,
  Write .initial:n = \BooleanFalse,
  Separ .code:n={
    \__oops_option_separ:n{#1}
    \exp_last_unbraced:Nf
    \__oops_make_oops:nnnn
    {
      { \g__oops_option_name_tl }
      { \g__oops_option_inner_tl }
      { \g__oops_option_separ_tl }
      { \g__oops_option_outer_tl }
    }
  },
  Separ .value_required:n = false,
  Separ .default:n = { {\ }and{\ } } { ,{\ } } { ,{\ }and{\ } },
  Separ .initial:n = { {\ }and{\ } } { ,{\ } } { ,{\ }and{\ } }
}
```

\OopsClear #1 : ⟨ *tl var name* ⟩

```
\NewDocumentCommand{ \OopsClear }
{ D<>{\g__oops_option_name_tl} }
{
  \__oops_prop_clear_new_map:n{#1}
}
```

(*End definition for* `\OopsClear`. *This function is documented on page* 5.)

\OopsDebug

```
\NewDocumentCommand\OopsDebug{m}
{
  \__oops_prop_if_exist:nTF{#1}
  { \c_empty_tl }
  { \__oops_prop_new:n{#1} }
  \__oops_make_key:Nn \KeyA{KeyA}
  \gappto{\OopsHook}
  {
    \__oops_prop_if_exist:nTF{#1}
    { \c_empty_tl }
    { \__oops_prop_new:n{#1} }
    \__oops_make_key:Nn \KeyA{KeyA}
  }
}
```

(*End definition for* `\OopsDebug`. *This function is documented on page* 5.)

\OopsOption

```
\NewDocumentCommand{ \OopsOption }
{ m }
{
  \keys_set:nn{ __oops }{#1}
}
```

(*End definition for* `\OopsOption`. *This function is documented on page* 5.)

\OopsRead

```
\NewDocumentCommand{\OopsRead}
{o}
{
  \IfValueTF{#1}
  {\__oops_log_read:e{#1}}
  {\__oops_log_read:}
}
```

(*End definition for* `\OopsRead`. *This function is documented on page* 6.)

10 Misc

```
\ExplSyntaxOff
```