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

Erwann Rogard[†]

Released 2020/04/04

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[3]. Specifically, `\OopsNew{`⟨*token list*$_1$⟩`}`, where ⟨*token list*$_1$⟩ identifies an object, begins a series of instructions alternating between 'text' and definitions, that themselves optionally expand using predefined or inline rules. For example,

```
\OopsNew{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 ⟨*token list*$_1$⟩, `$\Space$` also works. Optionally, the definitions can be saved to a file, and restored, which can be useful for typesetting documents sharing the same notational conventions. Altogether, "practical".

Contents

I Usage 3

1 Convention 3

2 Loading the package 3

3 `\OopsClear` 3

4 `\OopsNew` 3

4.1 `{`⟨*token list*$_1$⟩`}` . . . 4
4.2 `[`⟨*token list*$_2$⟩`]` . . . 4
4.3 `i{`⟨*code*$_1$⟩`}` . . . 4
4.4 `s{{`⟨*token list*$_3$⟩`}{`⟨*token list*$_4$⟩`}{`⟨*token list*$_5$⟩`}}` . . . 4
4.5 `o{`⟨*code*$_2$⟩`}` . . . 4
4.6 `{`⟨*keyval list*$_1$⟩`}` . . . 4
4.7 Remainder . . . 5

[*]This file describes version v1.1, last revised 2020/04/04.
[†]firstname dot lastname AusTria gmail dot com

5 \OopsOption **5**
5.8 Inner . . . 5
5.9 Name . . . 5
5.10 Outer . . . 5
5.11 Save . . . 5
5.12 Separ . . . 5

6 \OopsRestore **5**

7 \OopsTest **5**
7.13 A . . . 5
7.14 B . . . 5

II Listing **6**

Listing 1. **6**

Listing 2. **6**

Listing 3. **6**

Listing 4. **7**

Listing 5. **7**

Listing 6. **7**

Listing 7. **7**

Listing 8. **7**

Listing 9. **8**

III Other **9**

1 Acknowledgment **9**

2 Bug **9**

3 Install **9**

4 Support **9**

5 Unit testing **9**

Change History **10**

Index **10**

IV Implementation **12**

1 `aux` **12**

2 `log` **14**

3 `make` **15**

4 `msg` **17**

5 `option` **17**

6 `prop` **18**

7 `seq` **19**

8 `test` **20**

9 Front-end **20**

10 Misc **22**

Part I
Usage

This part describes .

Convention

1. Loosely, those of [2] and [3], for example as to the meaning of ⟨*token list*⟩ and -NoValue-.
2. If unspecified, the environment in which a function must be declared is `document`.

`\usepackage` `\usepackage{oops}`

Environment Preamble

Requirement `oops.sty` is in the path of the LaTeX engine. See Part III, section 4.

`\OopsClear` `\OopsClear[`⟨*token list*$_1$⟩`]`

`\OopsNew`

```
\OopsNew{⟨token list1⟩}
[⟨token list2⟩]
i{⟨code1⟩}
s{{⟨token list3⟩}{⟨token list4⟩}{⟨token list5⟩}}
o{⟨code2⟩}
{⟨keyval list1⟩}
i{⟨code3⟩}
<⟨keyval list2⟩>
[⟨token list6⟩]
```

Requirement ⟨*token list*$_1$⟩ and ⟨*keyval list*$_1$⟩ are mandatory.

⟨*token list*$_1$⟩

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

Semantics Registers a new object, if applicable

⟨*token list*$_2$⟩

Example `Let~`

Semantics Expands to ⟨*token list*$_2$⟩

⟨*code*$_1$⟩

Example `\mathbb{#1}`

⟨*token list*$_3$⟩

Example `{~\&~}`

⟨*token list*$_4$⟩

Example `{,~}`

⟨*token list*$_5$⟩

Example `{~\&~}`

⟨*code*$_2$⟩

Example `\text{#1}`

⟨*keyval list*$_1$⟩

Example `Sample=\Omega`

Semantics

1. Defines `\`⟨*key*$_i$⟩`[`⟨*token list*$_1$⟩`]` as ⟨*code*$_1$⟩ applied to ⟨*val*$_i$⟩.
2. If `Save=\BooleanTrue`,
   writes `\OopsNew{`⟨*token list*$_1$⟩`}i{`⟨*code*$_1$⟩`}{`⟨*keyval list*$_1$⟩`}` to file `oops`⟨*digits*⟩`.tex`,
   where ⟨*digits*⟩=`\pdfdate`
3. If ⟨*token list*$_2$⟩$\neq$ -NoValue-,
   expands to ⟨*code*$_2$⟩ applied to the list created in 1.,
   using `{`⟨*token list*$_3$⟩`}{`⟨*token list*$_4$⟩`}{`⟨*token list*$_5$⟩`}` as separator.

`⟨code₃⟩`, `⟨keyval list₂⟩`, and `⟨token list₆⟩`

Semantics `\OopsNew{⟨token list₁⟩}i{⟨code₃⟩}{⟨keyval list₂⟩}[⟨token list₆⟩]`

`\OopsOption` `\OopsOption{⟨kvl0⟩}`

Semantics Set default options for `\OopsNew`

`Inner`

Semantics Default for `⟨code₁⟩`

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

`Name`

Semantics Default for `⟨token list₁⟩`

`Outer`

Semantics Default for `⟨code₂⟩`

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

`Save`

Syntax ⟨*boolean*⟩

`Separ`

Syntax That of 'separators' in [2, Section 8 of l3seq]

Semantics Default for `{⟨token list₃⟩}{⟨token list₄⟩}{⟨token list₅⟩}`

`\OopsRestore` `\OopsRestore[⟨path⟩]`

Semantics Restores the definitions saved in ⟨*path*⟩ and writes to `oops.log`: 'restore ⟨*path*⟩'

`\OopsTest` `\OopsTest{A|B}{⟨arg₁⟩}`

`A`

Semantics `\OopsClear`{Test}`\OopsNew`{Test}[A]`⟨arg₁⟩{ X = x, Y = y, Z = z }`

`B`

Semantics `\OopsNew`{Test}[B]`{ W = w, X = x }⟨arg₁⟩< Y = y, Z = z >`

Part II
Listing

Warning: To reproduce the listings in a LaTeX document, use the same formatting instructions as those of the documentation portion of `oops.dtx` (such as `\documentclass`, `\usepackage`, and `\newtcblisting`), and remove any `^^A`. Any deviation from the original may require tinkering.[1]

Listing 1.

```
%      \OopsOption{
%      Inner={\{####1\}},
%      ^^A% spaces betw. inner and outer brackets matter!->
%      Separ={{\ \char`@\ }{\%\ }{\ \char`@\ }},
%      Outer={\char`^####1\$}}
%      \OopsTest{A}{} \tab \X[Test]\Y[Test]\Z[Test]\\
%      \OopsTest{A}{ i{(#1)} } \tab \X[Test]\Y[Test]\Z[Test]\\
%      \OopsTest{A}{ s{{\ \&\ }{,\ }{\ \&\ }} }\\
%      \OopsTest{A}{ o{\char`[#1\char`]} }
%
```

```
A: ^{x}% {y} @ {z}$                 {x}{y}{z}
A: ^(x)% (y) @ (z)$                 (x)(y)(z)
A: ^{x}, {y} & {z}$
A: [{x}% {y} @ {z}]
```

Listing 2.

```
%      \OopsOption{ Inner, Separ, Outer }
%      \OopsTest{A}{} \tab\X[Test]\Y[Test]\Z[Test]\\
%      \OopsTest{A}{ i{(#1)} } \tab\X[Test]\Y[Test]\Z[Test]\\
%      \OopsTest{A}{ s{{\ \&\ }{,\ }{\ \&\ }} }\\
%      \OopsTest{A}{ o{\char`[#1\char`]} }
%
```

```
A: x, y, and z                      xyz
A: (x), (y), and (z)                (x)(y)(z)
A: x, y & z
A: [x, y, and z]
```

Listing 3.

```
%      \OopsTest{B}{} \tab\W[Test]\X[Test]\Y[Test]\Z[Test]\\
%      \OopsOption{ Save = \BooleanTrue }
%      \OopsTest{B}{ i{(#1)} } \tab\W[Test]\X[Test]\Y[Test]\Z[Test]
%      \OopsOption{ Save = \BooleanFalse }
```

[1] For instance, in testing v1.1, I realized `\usepackage[T1]{fontenc}` was needed, to work with `\documentclass{article}` in place of `\documentclass[full]{l3doc}`, hence added it to the documentation portion of `oops.dtx`

```
%
```

B: *w and x* wxyz
B: *w and x* wx(y)(z)

Listing 4.

```
%     \OopsRestore \tab\W[Test]\X[Test]\Y[Test]\Z[Test]
%
```

wx(y)(z)

Listing 5.

```
%     \OopsNew{Math}[We call~]{Elems={\omega_1, \dots, \omega_n}}
%     [~the elementary events, and ]{}<Space=\Omega>
%     [\begin{equation*}\Space=(\Elems)\end{equation*}~the sample space.]
%     {}
%     \OopsClear
%
```

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

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

the sample space.

Listing 6.

```
%     \OopsOption{ Save = \BooleanTrue }
%     \OopsNew{Math}[Let ]s{{,}{,}{,}}o{\ensuremath{\{#1\}}}
%     {Space=\Omega, SigmaField=\mathcal{F}, Measure=\mathcal{P}}
%     [~denote the probability space, where $\SigmaField\subset
    2^{\Space}$.]
%     {}
%     \OopsClear
%     \OopsOption{ Save = \BooleanFalse }
%
```

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

Listing 7.

```
%     \OopsRestore \tab $\Omega$ $\SigmaField$ $\Measure$
%     \OopsClear
%
```

Ω $\mathcal{F}$ $\mathcal{P}$

Listing 8.

```
%     \OopsOption{ Save = \BooleanTrue }
%     \newtheorem{theorem}{Theorem}
%     \OopsNew{Math}i{\mathbb{#1}}{ N = { N } , R = { R } }
%     [\begin{theorem}[Mittelwertsatz f\"ur $n$ Variable]Es~sei~]{}
%       <OffeneMenge={D}, Ci={C^{1}}, Strecke={[x_0,x]}>
%       ^^A%      Strecke={\char`[x_0,x\char`]}  % PASS
%       ^^A%      Strecke={\[x_0,x\]}            % FAIL
%       [$n\in\N$,~$\OffeneMenge\subseteq\N^n$ eine offene Menge und
    $f\in\Ci(\OffeneMenge,\R)$. Dann gibt es auf jeder Strecke
    $\Strecke\subset\OffeneMenge$ einen Punkt $\xi\in\Strecke$,~]{}
%       <yDifferenz={f(x)-f(x_0)},
%       xDifferenz={x-x_0},
%       Steigung={\frac{\yDifferenz}{\xDifferenz}}>
%       [so dass gilt
%       \begin{equation*}\Steigung = \operatorname{grad}
    f(\xi)^{\top}\end{equation*}
%     \end{theorem}]{}
%     \OopsClear
%     \OopsOption{ Save = \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 9.

```
%     \OopsRestore \tab $\N$ $\R$ $\OffeneMenge$ $\Ci$ $\Strecke$
%
```

$\mathbb{N}$ $\mathbb{R}$ 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 5 and Listing 6 are from [1]. Listing 8 is from `tcolbox`[4, 17.3].

2 Bug

1. Some characters don't work for ⟨*`keyval list`*$_1$⟩, but there are workarounds, see Listing 8.

3 Install

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

4 Support

This package is available from `https://www.ctan.org/pkg/oops` (on the source, a.k.a `dtx`, file) and `https://github.com/rogard/oops`.

5 Unit testing

It's not possible to check the expansion of a certain class of macros against predefined values[5]. 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] The LaTeX3 Project Team *The LaTeX3 interfaces* `http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf`

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

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

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

Change History

v1.0

General: Initial version 9

v1.1

General: Added: `Save` 9

Added: Listing 1, Listing 2, Listing 3, Listing 4, Listing 7 and Listing 9 9

Added:`\OopsRestore` 9

Added:`\OopsTest` 9

Deleted: Listing 1-5 from v1.0 9

Fixed: apparent anomaly in v1.0's Listing 4, see Listing 1 9

Replaced: `\OopsOptions` by `\OopsOption` . . 9

Replaced: `{`⟨*keyval list*$_2$⟩`}` by `<keyval list`$_2$`>` given that option type `G` not recommended[3] 9

Replaced: `GenericObject` by `Name` . 9

Replaced: `Separators` by `Separ` . . . 9

Revamped: much of the implementation 9

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

⟨*code*$_1$⟩ (option) *4*
⟨*code*$_2$⟩ (option) *4*
⟨*code*$_3$⟩, ⟨*keyval list*$_2$⟩, and ⟨*token list*$_6$⟩ (option) *5*
⟨*keyval list*$_1$⟩ (option) *4*
⟨*token list*$_1$⟩ (option) *4*
⟨*token list*$_2$⟩ (option) *4*
⟨*token list*$_3$⟩ (option) *4*
⟨*token list*$_4$⟩ (option) *4*
⟨*token list*$_5$⟩ (option) *4*
`A` (option) *5*
`B` (option) *5*
`Inner` (option) *5*
`Name` (option) *5*
`Save` (option) *5*
`Separ` (option) *5*

`\␣` 330, 331

A

`\AtEndDocument` 61

B

`\begingroup` 130
bool commands:
  `\bool_gset_false:N` 66
  `\bool_gset_true:N` 72
  `\bool_if:nTF` 87, 125, 312
  `\bool_set_false:N` 62
`\BooleanFalse` 317, 318
`\BooleanTrue` *4*

C

cs commands:
  `\cs_generate_variant:Nn` 32, 79, 91, 101, 106, 219
  `\cs_gset:Npn` 171, 193, 205
  `\cs_new:Nn` 17, 181, 228, 232, 261
  `\cs_new:Npn` 236
  `\cs_new_protected:Nn` 8, 12, 28, 42, 46, 55, 63, 68, 74, 80, 85, 92, 102, 107, 169, 173, 191, 195, 203, 220, 224, 237, 241, 249, 254
  `\cs_new_protected:Npn` 4, 33, 111, 211
  `\cs_set:Nn` 213
  `\cs_set_protected:Nn` 243

D

`\DeclareDocumentCommand` 113
`\def` 130
`\documentclass` *6*

E

`\endgroup` 130
`\ensuremath` 309, 310
exp commands:
  `\exp_args:NNx` 94, 113
  `\exp_args:No` 119

\exp_last_unbraced:Nf . 287, 300, 321
\exp_last_unbraced:NNo 140
\exp_not:N 25, 179, 189, 201, 209
\exp_not:n 150, 153, 159
\expandafter 130
\ExplSyntaxOff 354
\ExplSyntaxOn 3

F

file commands:
\file_input:n 76

I

\IfValueT 133, 157
\IfValueTF 146, 148, 346
iow commands:
\iow_close:N 61, 65
\iow_new:N 60
\iow_now:Nn 88
\iow_open:Nn 71

K

keys commands:
\keys_define:nn 277
\keys_set:nn 341

M

msg commands:
\msg_error:nnn 23, 89, 187
\msg_error:nnnn 271
\msg_new:nnn . . 164, 165, 166, 167, 168
\msg_warning:nnn 179, 201, 209

N

\NeedsTeXFormat 2
\NewDocumentCommand . . . 333, 338, 343, 350
\newtcblisting *6*

O

oops internal commands:
__oops_aux_key:N . . . . <u>12</u>, 123
__oops_aux_key:n . . . . <u>8</u>, 15
__oops_aux_key:w . . . . <u>4</u>, 10
\g__oops_aux_key_seq . 6, 14, 124, 135
\g__oops_aux_keyval_seq 120, 121, 123
__oops_aux_name:n . . . . <u>17</u>, 280
\g__oops_aux_prop . 27, 30, 39, 48, 122
__oops_aux_prop:N . . . . <u>46</u>, 121
__oops_aux_prop:n . . . . <u>42</u>, 52
__oops_aux_prop:nn 28, 32, 35
__oops_aux_prop:w . . . . <u>27</u>, 44
__oops_aux_val 57, 58, 142
__oops_aux_val:Nn . . . . <u>55</u>, 135
__oops_log_close: . . . . <u>60</u>, 314
__oops_log_entry 130
\g__oops_log_iow . 60, 61, 65, 71, 88, 89
__oops_log_open: . . . . <u>68</u>, 313
\g__oops_log_open_bool 62, 66, 72, 87, 125
__oops_log_restore: . . . . <u>80</u>, 348
__oops_log_restore:n . . . <u>74</u>, 82, 347
__oops_log_save:n . . . . <u>84</u>, 128
\g__oops_log_to_tl 70, 71, 82, 84
__oops_make_key:N . . . . <u>107</u>, 124
__oops_make_key:n . . . . <u>102</u>, 109
__oops_make_key:Nn . . . . <u>92</u>, 104
__oops_make_new:nnn <u>111</u>, 288, 301, 322
\g__oops_name_tl 19, 25, 96, 334
__oops_option_inner:n . . . . . 119, <u>169</u>
__oops_option_inner_:n . . 37, 40, 171
__oops_option_inner_default:n 173, 177, 286
\g__oops_option_inner_tl 175, 179, 290, 303, 324
__oops_option_name:n . . . . <u>181</u>
\g__oops_option_name_tl 183, 189
__oops_option_outer:n . . . . . 136, <u>191</u>
__oops_option_outer_:n 138, 193
__oops_option_outer_default:n . . . . <u>191</u>, 299
\g__oops_option_outer_tl 197, 201, 292, 305, 326
__oops_option_separ_default:n . . . . <u>203</u>, 320
\g__oops_option_separ_tl 205, 209, 291, 304, 325
__oops_prop_append:NN . . . . . <u>211</u>, 222
__oops_prop_append:Nn . . . . . 122, <u>220</u>
__oops_prop_append:nn 213, 217
__oops_prop_clear_new:n . . . <u>224</u>, 336
__oops_prop_if_exist:nTF . . 116, <u>228</u>
__oops_prop_item:nn . . . . 98, <u>232</u>
__oops_prop_name:n . . . . 58, 222, 226, 230, 234, <u>236</u>, 239
__oops_prop_new:n . . . . 118, <u>237</u>
__oops_seq_from_prop:n 243, 247
__oops_seq_from_prop:NNn . . . 58, <u>241</u>
__oops_test:n 273
__oops_test:nn . . . . <u>259</u>, 352
\c__oops_test_a 259, 266
__oops_test_a:n . . . . <u>249</u>, 266
\c__oops_test_b 260, 267
__oops_test_b:n . . . . <u>254</u>, 267
\OopsClear . . . . *1*, *3*, *5*, 251, 256, <u>333</u>
\OopsNew *1*, *4*, *4*, *5*, *5*, *5*, *5*, 113, 130, 150, 153, 159, 252, 257
\OopsOption . . . . *2*, *5*, *10*, <u>338</u>
\OopsOptions *10*
\OopsRestore . . . . *2*, *5*, *10*, <u>343</u>

\OopsTest *2*, *5*, *10*, 350
options:
  ⟨*code*$_1$⟩ *4*
  ⟨*code*$_2$⟩ *4*
  ⟨*code*$_3$⟩, ⟨*keyval list*$_2$⟩, and
    ⟨*token list*$_6$⟩ *5*
  ⟨*keyval list*$_1$⟩ *4*
  ⟨*token list*$_1$⟩ *4*
  ⟨*token list*$_2$⟩ *4*
  ⟨*token list*$_3$⟩ *4*
  ⟨*token list*$_4$⟩ *4*
  ⟨*token list*$_5$⟩ *4*
  A *5*
  B *5*
  Inner *5*
  Name *5*
  Save *5*
  Separ *5*
  Outer *5*
Outer (option) *5*

P

\pdfdate 70
prop commands:
  \prop_clear_new:N 226
  \prop_gclear_new:N 48
  \prop_gput:Nnn 30, 39, 215
  \prop_if_exist:NTF 230
  \prop_item:Nn 215, 234, 245
  \prop_map_function:NN 217
  \prop_new:N 27, 239
\ProvideDocumentCommand 95

Q

quark commands:
  \q_stop 4, 10, 33, 44

S

seq commands:
  \seq_gclear_new:N 14, 57
  \seq_gput_right:Nn 6, 245
  \seq_if_empty:NTF 49
  \seq_map_function:NN 15, 52, 109, 247
  \seq_set_from_clist:Nn 120
  \seq_use:Nnnn 141

T

tl commands:
  \c_empty_tl 50, 117, 132, 269
  \tl_case:NnTF 264
  \tl_const:Nn 259, 260
  \tl_gset:Nn 19, 70, 175, 183, 197
  \tl_log:n 77
  \tl_new:N 84
  \tl_set:Nn 263
  \tl_trim_spaces:n 6, 36, 40
  \l_tmpa_tl 263, 264

U

\usepackage *3*, *6*

Part IV
Implementation

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

1 aux

__oops_aux_key:w

#1 : ⟨ *key* ⟩
#2 : ⟨ *value* ⟩

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

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

__oops_aux_key:n

#1 : ⟨ *key* = *value* ⟩

```
\cs_new_protected:Nn \__oops_aux_key:n
```

```
{
  \__oops_aux_key:w #1 \q_stop
}
```

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

`\__oops_aux_key:N` #1 : ⟨ *seq* ⟩

```
\cs_new_protected:Nn \__oops_aux_key:N
{
  \seq_gclear_new:N \g__oops_aux_key_seq
  \seq_map_function:NN #1 \__oops_aux_key:n
}
```

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

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

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

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

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

```
\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 }
\cs_new_protected:Npn \__oops_aux_prop:w #1 = #2 \q_stop
{
  \__oops_aux_prop:eo
  { \tl_trim_spaces:n{ #1 } }
  { \__oops_option_inner_:n{ #2 } }
% ^^A\prop_gput:Noo \g__oops_aux_prop % v1.1, FAIL with N = N (OK with N= N)
% ^^A  { \tl_trim_spaces:n{ #1 } } { \__oops_option_inner_:n{ #2 } }
}
```

(*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_val:Nn #1 : ⟨ *seq* ⟩
#2 : ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__oops_aux_val:Nn
{
  \seq_gclear_new:N \__oops_aux_val
  \__oops_seq_from_prop:NNn \__oops_aux_val #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:

```
\cs_new_protected:Nn \__oops_log_open:
{
  \tl_gset:Nx \g__oops_log_to_tl{oops\pdfdate}
  \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_restore:n #1 : ⟨*path*⟩

```
\cs_new_protected:Nn \__oops_log_restore:n
{
  \file_input:n{#1}
  \tl_log:n{restore~#1}
}
\cs_generate_variant:Nn \__oops_log_restore:n { e }
```

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

`\__oops_log_restore:`

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

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

`\__oops_log_save:n`

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

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

3 make

`\__oops_make_key:Nn`

#1 : ⟨ *token* ⟩
#2 : ⟨ *key* ⟩

```
\cs_new_protected:Nn \__oops_make_key:Nn
{
  \exp_args:NNx
  \ProvideDocumentCommand{ #1 }
  { O{\g__oops_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`.)

`\__oops_make_new:nnn`

#1 : $\langle$ *seq*$_1$ $\rangle$
#2 : $\langle$ *seq*$_2$ $\rangle$
#3 : $\langle$ *prop* $\rangle$

```
\cs_new_protected:Npn \__oops_make_new:nnn #1 #2 #3
{
  \exp_args:NNx \DeclareDocumentCommand \OopsNew
  { m +o E{ i s o }{ { #1 }{ #2 }{ #3 } } m E{ i }{ { #1 } } d<> +o }
  {
    \__oops_prop_if_exist:nTF{ ##1 }
    { \c_empty_tl }
    { \__oops_prop_new:n{ ##1 } }
    \exp_args:No \__oops_option_inner:n{ ##3 }
    \seq_set_from_clist:Nn \g__oops_aux_keyval_seq { ##6 }
    \__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
    \bool_if:nTF{ \g__oops_log_open_bool }
    {
%^^A https://tex.stackexchange.com/questions/536597
\__oops_log_save:n
{
  \begingroup \def \__oops_log_entry { \OopsNew{ ##1 }i{##3}{ ##6 } } \expandafter \endgroup
}
}{\c_empty_tl}
\IfValueT{ ##2 }
{
  \__oops_aux_val:Nn \g__oops_aux_key_seq { ##1 }
  \__oops_option_outer:n{ ##5 }
  ##2
  \__oops_option_outer_:n
  {
    \exp_last_unbraced:NNo
    \seq_use:Nnnn
    \__oops_aux_val
    { ##4 }
  }
}
\IfValueTF{ ##8 }
{
  \IfValueTF{ ##9 }
  {
    \exp_not:n{ \OopsNew{ ##1 }i{ ##7 }{ ##8 }[ ##9 ] }
  }
  {
    \exp_not:n{ \OopsNew{ ##1 }i{ ##7 }{ ##8 } }
  }
}
{
  \IfValueT{##9}
  {
    \exp_not:n{ \OopsNew{ ##1 }[ ##9 ] }
  }
}
}
}
```

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

4 msg

```
\msg_new:nnn {__oops}{ generic }{ #1 }
\msg_new:nnn {__oops}{ iow }{ #1~is~closed~can't~save }
\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}{ unset }{ #1~unset }
```

5 option

`\__oops_option_inner:n`

#1 : ⟨*inlinecode*⟩

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

(*End definition for* `\__oops_option_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`
`\__oops_option_outer_default:n`

#1 : ⟨ *inline code* ⟩

```
\cs_new_protected:Nn \__oops_option_outer:n
{
  \cs_gset:Npn \__oops_option_outer_:n ##1 { #1 }
}
\cs_new_protected:Nn \__oops_option_outer_default:n
```

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

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

`\__oops_option_separ_default:n`

#1 : {⟨ *token list*$_1$ ⟩}{⟨ *token list*$_2$ ⟩}{⟨ *token list*$_3$ ⟩}

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

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

6 prop

`\__oops_prop_append:NN`
`\__oops_prop_append:cN`

#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
{
  \prop_clear_new:c{ \__oops_prop_name:n { #1 } }
}
```

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

`\__oops_prop_if_exist:nTF`

#1 : ⟨*token list*$_1$⟩
#2 : ⟨*token list*$_2$⟩
#3 : ⟨*token list*$_3$⟩

```
\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`.)

7 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`.)

8 test

__oops_test_a:n

```
\cs_new_protected:Nn \__oops_test_a:n
{
  \OopsClear[Test]
  \OopsNew{Test}[A:~]#1{ X = x, Y = y, Z = z }
}
```

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

__oops_test_b:n

```
\cs_new_protected:Nn \__oops_test_b:n
{
  \OopsClear[Test]
  \OopsNew{Test}[B:~]{ W = w, X = x }#1< Y = y, Z = z >
}
```

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

__oops_test:nn

```
\tl_const:Nn \c__oops_test_a { A }
\tl_const:Nn \c__oops_test_b { B }
\cs_new:Nn \__oops_test:nn
{
  \tl_set:Nn \l_tmpa_tl { #1 }
  \tl_case:NnTF \l_tmpa_tl
  {
    \c__oops_test_a { \__oops_test_a:n{#2} }
    \c__oops_test_b { \__oops_test_b:n{#2} }
  }
  { \c_empty_tl }
  {
    \msg_error:nnnn{ __oops }
    { keywrong }
    { \__oops_test:n }
    { #1 }
  }
}
```

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

9 Front-end

```
\keys_define:nn { __oops }
{
  Name .code:n={
    \__oops_aux_name:n{ #1 }
  },
  Name .value_required:n = false,
  Name .default:n = { Math },
  Name .initial:n = { Math },
  Inner .code:n={
    \__oops_option_inner_default:n{ #1 }
    \exp_last_unbraced:Nf
    \__oops_make_new:nnn
    {
      { \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_default:n{ #1 }
    \exp_last_unbraced:Nf
    \__oops_make_new:nnn
    {
      { \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} },
  Save .code:n = {
    \bool_if:nTF{#1}
    {\__oops_log_open:}
    {\__oops_log_close:}
  },
  Save .value_required:n = false,
  Save .default:n = \BooleanFalse,
  Save .initial:n = \BooleanFalse,
  Separ .code:n={
    \__oops_option_separ_default:n{ #1 }
    \exp_last_unbraced:Nf
    \__oops_make_new:nnn
    {
      { \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 }
{ O{\g__oops_name_tl} }
{
  \__oops_prop_clear_new:n{ #1 }
}
```

(*End definition for* \OopsClear. *This function is documented on page 3.*)

\OopsOption

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

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

\OopsRestore

```
\NewDocumentCommand{\OopsRestore}
{o}
{
  \IfValueTF{#1}
  {\__oops_log_restore:e{#1}}
  {\__oops_log_restore:}
}
```

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

\OopsTest

```
\NewDocumentCommand\OopsTest{mm}
{
  \__oops_test:nn{ #1 }{ #2 }
}
```

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

10 Misc

```
\ExplSyntaxOff
```