The ccool package*

Erwann Rogard[†]

Released 2020/04/15

Abstract

ccool stands for Custom COntent Oriented for IATEX, a concept pioneered by $cool[1]^1$. This is done using a minimalist interface built upon xparse[4]. Specifically, $\Cool<\langle name \rangle>$ begins a series of instructions alternating between 'text' and macro definitions, that themselves optionally expand using predefined or inline rules. For example,

```
\Ccool<Math>[Let~]
i{\mathbb{#1}}{ Nat = N, Real = R }*s{{~\rm{and}~}}
[~denote the natural and real numbers.]{}
```

expands to: "Let $\mathbb N$ and $\mathbb R$ denote the natural and real numbers." As a side effect, $\$ \Nat<\ath>encodes " $\mathbb N$ " (and likewise for \Real). Math being the default for $\langle name \rangle$, <\ath>ath> can be dropped. In conjunction with lamba expressions, this tool allows for encoding the way certain mathematical objects, such as functions, should be formatted. Optionally, the macros can be written to a file, and read, which can be useful for typesetting documents sharing the same notation.

Contents

Ι	Usage	4
0	Convention	4
1	Loading the package	4
2	\Ccool	4
	$2.1 < \langle tl_1 \rangle > \dots$. 4
	2.2 [$\langle tl_2 \rangle$]	
	2.3 $i\{\langle code_1\rangle\}$	
	$2.4 \{\langle kvl_1 \rangle\}$. 5
	2.5 +	
	2.6 *	. 5
	2.7 $s\{\{\langle tl_3\rangle\} \{\langle tl_4\rangle\} \{\langle tl_4\rangle\} \{\langle tl_4\rangle\}\}\{\langle tl_5\rangle\}\}$. 5
	$2.8 o\{\langle code_2\rangle\}$	
	$2.9 \lceil \langle tl_6 \rangle \rceil$. 5

^{*}This file describes version v2.0, last revised 2020/04/15.

 $^{^{\}dagger}$ firstname dot lastname Aus
Tria gmail dot com

¹Whereas cool provided predefined macros, ccool is tool for making macros, hence "custom".

3 Other	5	
4 Do's and dont's	6	
II Listing	7	
<u> </u>		
Listing 1. Version.	7	
Listing 2. Preamble.	7	
Listing 3. Separators.	7	
Listing 4. Hello, world!	7	
Listing 5. Listing 4 read from file.		
Listing 6. Probability space.	8	
Listing 7. Listing 6 read from file.	8	
Listing 8. Mittelwertsatz für n Variable.		
Listing 9. Listing 8 read from file.		
Listing 10. Lambda expression.	9	
Listing 11. Listing 10 read from file.	10	
Listing 12. CUSUM statistic.	10	
Listing 13. Listing 12 read from file.	11	
III Other	12	
1 Acknowledgment	12	
2 Install	12	
3 Issue	12	
4 Support	12	
5 Testing 5.1 Technicality		
References	13	

Cha	ange History	13
Ind	ex	14
IV	Implementation	18
1	aux	18
2	lambda	20
3	log	21
4	make_key	22
5	make_ccool	22
6	msg	24
7	option	24
8	prop	25
9	seq	26
10	sys	27
11	Front-end	28
12	\CcoolClear	28
13	\CcoolHook	28
13	\CcoolLambda	28
13	\CcoolOption 13.2 Expans 13.2 File 13.2 Inner 13.2 Name 13.2 Outer 13.2 Separ 13.3 Write	28 29 29 29 29 30 30 30
14	\CcoolRead	31
15	\CcoolVers	31

16 Misc 31

Part I

Usage

Convention

- 1. Loosely, those of [2] and [4], for example as to the meaning of $\langle token \ list \rangle$.
- 2. If unspecified, the environment in which a macro must be declared is document.

\usepackage

\usepackage{ccool}

Requirement

Default Inner

Example \mathbb{#1}

- 1. ccool.sty is in the path of the LATEX engine. See Part III, section 4.
- 2. Declare it in the *preamble*

```
\cool<\langle tl_1\rangle>
 \Ccool
                     [\langle t1_2 \rangle]
                    \mathtt{i}\{\langle \mathit{code}_1\rangle\}
                    \{\langle kvl_1 \rangle\}
                    \mathtt{s}\{\{\langle \mathtt{t} 1_3 \rangle\} \,|\, \{\langle \mathtt{t} 1_3 \rangle\} \{\langle \mathtt{t} 1_4 \rangle\} \,|\, \{\langle \mathtt{t} 1_3 \rangle\} \{\langle \mathtt{t} 1_4 \rangle\} \{\langle \mathtt{t} 1_5 \rangle\}\}
                    o\{\langle code_2 \rangle\}
                    [\langle t1_6 \rangle]
                    Requirement \langle kvl_1 \rangle is specified (all others optional).
    \langle \mathtt{tl}_1 \rangle
                    Default Name
                    Example Math, ModelA, ModelB
                    Semantics Identifies a group of macros
    \langle \mathtt{tl}_2 \rangle
                    Example Let~
                    Semantics Expands \langle tl_2 \rangle
\langle code_1 \rangle
```

```
Semantics
                      1) \langle val_i \rangle \leftarrow \langle code_1 \rangle applied to \langle val_i \rangle
 \langle kvl_1 \rangle
            Example Elems={\omega_1, \dots, \omega_n}, Sample=\Omega
            Semantics
                      2) \langle \ker_i \rangle \langle \ell l_1 \rangle \rangle \leftarrow \langle val_i \rangle defined in step 1), using Expans for expansion.
                      3) If Write, writes the input used by step 2) to File
            Semantics Appends step 2) and step 3) to \CcoolHook 2
            Semantics
                       4. Expands \langle code_2 \rangle applied to the list created in step 1), using the separator
                           specified by \langle tl_3 \rangle, \langle tl_4 \rangle, \langle tl_5 \rangle.
   \langle \mathtt{tl}_3 \rangle
            Default Separ
            Example {~\in~}
   \langle t1_4 \rangle
            Default Separ
            Example {,~}
   \langle tl_5 \rangle
            Default Separ
            Example {~\&~}
\langle code_2 \rangle
            Default Outer
            Example $\left\{#1\right\}$
```

Semantics $\cool<\langle tl_1\rangle>[\langle tl_6\rangle]$

 $\langle t1_6 \rangle$

²Needed inside a *local group*, for the side effect of \Ccool to persist thereafter.

Other

Continued in Part IV, section 11.

Do's and dont's

```
1.
Don't: \Ccool{ A = a, B = b }[Hello, world!].
Do: \Ccool{ A = a, B = b }[Hello, world!]{}, or
        \Ccool{ A = a, B = b } Hello, world!
2.
Don't: $\\\ \langle key_i \rangle \cdots \rangle
3.
Don't: [a, b)
    Do: {[}a, b{})}
4.
Don't: \Ccool{ F = \cal F }.
    Do: \Ccool{ F = \cal{F} } or \Ccool{ F = \mathcal{F} } }
5.
Don't: [x_0,x]
    Do: \left[x_0,x\right]
6. Also see Part III, section 3
```

Part II

Listing

```
Listing 1.

% \CcoolVers
%

2020/04/15 v2.0 cool — A tool for encoding mathematical notation
```

```
Listing 2. Preamble<sup>a</sup>

"These are the settings to replicate the listings. For exhaustivity, check the documentation section of ccool.dtx.

"\times \usepackage{amsmath, amsthm, commath} \usepackage[T1]{fontenc}\"\char`[
```

```
Listing 3. Separators
        \CcoolOption{
        ^^A% spaces betw. inner and outer brackets matter!->
%
       Separ=\{\{\ \char`@\ \}\{\ \'\%\ \}\{\ \char`@\ \}\}\}
        \ \Ccool<Test>\{ X = x, Y = y \}*[\]
        \{ X = x, Y = y, Z = z \} * [ \setminus  ]
        \{ X = x, Y = y \}*s\{\{\ \\&\ \}\}[\\]
        \{ X = x, Y = y \}*s\{\{ \setminus \& \setminus \}\{, \setminus \}\}[ \setminus ] 
       \{ X = x, Y = y, Z = z \}*s\{\{\setminus \&\setminus \}\}[\setminus\setminus]
%
%
        \{ X = x, Y = y, Z = z \}*s\{\{\ \ \ \ \ \}\{,\ \ \}\}[\ \ ]
        { X = x, Y = y, Z = z }*s{{\ \&\ }{,\ }{\ \&\ }}\\
x @ y
x~\%~y~@~z
x \& y
x \& y
x \& y \& z
x, y \& z
x, y \& z
```

```
Listing 4. Hello, world!<sup>a</sup>

all this looks arcane, it's for the purpose of testing.

% \CcoolOption{ Separ = {{}}{.}}{.}}, Outer = { ###1 } }

% \CcoolOption{ Write = \BooleanTrue }
```

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

```
Listing 5. Listing 4 read from file.
                                      \CcoolRead
%
                                      \verb|\KeyF<Test>\KeyA<Test>\\nobreak|
%
%
                                      \KeyG<Test>\KeyA<Test>\nobreak
%
                                      \KeyH<Test>\KeyA<Test>\nobreak
%
                                      \KeyH<Test>\KeyA<Test>\nobreak
%
                                      {\footnote{\color=0.05cm} {\footnote{\color=0.05cm} {\color=0.05cm} {\color=
%
                                      \KeyM<Test>\KeyR<Test>\nobreak
                                      \Key0<Test>\nobreak
%
                                      \KeyT<Test>\nobreak
%
                                      \KeyL<Test>\nobreak
%
                                      \KeyD<Test>\nobreak
%
                                      \KeyZ<Test>\nobreak
                                      \KeyN<Test>\nobreak
%
\{H\}.\{e\}.\{l\}.\{o\}, [world!]
```

```
Listing 8. Mittelwertsatz für n Variable[3, 17.3]
%
      \CcoolOption{ Write = \BooleanTrue }
%
      \newtheorem{theorem}{Theorem}
      \AfterEndEnvironment{theorem}{\CcoolHook}
%
      \Ccool 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 = { | left[x_0,x|right] } }+
%
         [$n\in N$, ~$\setminus OffMenge\setminus Subseteq\setminus N^n$ eine offene Menge und
    f\in Ci(\Omega_{R}).
%
        Dann gibt es auf jeder Strecke $\Strecke\subset\OffMenge$ einen
    Punkt $\xi\in\Strecke$,~]
        { Steig = { f(x)-f(x_0) }{ x-x_0 } }, Punkt = { xi } }+
%
         [so dass gilt
%
%
         \begin{equation*}
%
           \Steig = \Grad f(\Punkt)^{\perp}
%
         \end{equation*}
%
      \end{theorem}]
%
%
      (Check: N\, \Punkt$)
%
      \CcoolOption{ 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
```

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

(Check: \mathbb{N}, ξ)

```
Listing 9. Listing 8 read from file.
```

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

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

Listing 12. CUSUM statistic[5]

```
%
                \newtheorem{definition}{Definition}
%
                \AfterEndEnvironment{definition}{\CcoolHook}
%
                \CcoolOption{ Write = \BooleanTrue }
%
                \Ccool{ SuchThat = { ;~ }, Time = { t }, Process = { \xi }, StopT = { \x
            \{T\}, EvalAt = \CcoolLambda\{(\#1)\}
%
                [The CUSUM statistic process and the corresponding one-sided CUSUM
           stopping time are defined as follows:
%
                \begin{definition}\label{the CUSUM statistic}. Let~]
                     { Scale = { \lambda }, Real = {\mathcal{R}} }+*s{{~\in~}}[~and~]
%
%
                     { CUSUMthresh = { \langle u \rangle } +*o{$\#1\leq ^{+}$.}
%
                     [~Define the following processes:]
%
                     { LogWald = { u }, CUSUMst = { \StopT_{c} }, CUSUM = { y },
           LogWaldInf = { m } }+
%
                      [\begin{enumerate}
%
                      \int {\int {Scale } = Scale \Pr (Scale)} 
            - \frac{1}{2}\Scale^2\Time$;
%
                          \Lambda = \prod_{N \in \mathbb{N}} EvalAt\{ Scale \} = \inf_{N \in \mathbb{N}} O \le N \in \mathbb{N}
           }\CUSUM_{s} \EvalAt{ \Scale }$.}
%
                     \left( \frac{s\CUSUM_{\Omega}}{Time}\right) =
           \label{logWaldInf_{\Time}\EvalAt{ \Scale } - \LogWald_{\Time}\EvalAt{ }
            \Scale }\geO$, which is the CUSUM statistic process.}
                     \left( \frac{s\c Scale}{\c Scale} \right) = \inf\left( \frac{Time}{\c Scale} \right)
%
            \ge 0 \SuchThat \CUSUM_{\Time}\EvalAt{\Scale} \ge \LogWaldInf
            \right]$, which is the CUSUM stopping time.}
            \end{enumerate}\end{definition}\par]{}
%
%
                (Check: $\Scale$, $\CUSUM$)
%
                \CcoolOption{ Write = \BooleanFalse }
```

%

The CUSUM statistic process and the corresponding one-sided CUSUM stopping time are defined as follows:

Definition 1 . Let $\lambda \in \mathcal{R}$ and $\nu \in \mathcal{R}^+$. Define the following processes:

- 1. $u_t(\lambda) = \lambda \xi_t \frac{1}{2}\lambda^2 t$; $m_t(\lambda) = \inf_{0 \le s \le t} y_s(\lambda)$.
- 2. $y_t(\lambda) = m_t(\lambda) u_t(\lambda) \ge 0$, which is the CUSUM statistic process.
- 3. $T_c(\lambda, m) = \inf[t \ge 0; y_t(\lambda) \ge m]$, which is the CUSUM stopping time.

(Check: λ, y)

Listing 13. Listing 12 read from file.

% \CcoolRead \tab π \Frocess\$ \$\Scale\$ \$\Real\$ \$\CUSUMthresh\$ \$\LogWald\$ \$\CUSUMst\$ \$\CUSUM\$ \$\LogWaldInf\$

%

 $\bar{t} \xi \bar{\lambda} \bar{\mathcal{R}} \nu u \bar{T}_c y m$

Part III

Other

1 Acknowledgment

This work has benefited from Q&A's from the LATEX community[6]. Specific attributions are made throughout this document.

2 Install

Compiling ccool.dtx³ will generate ccool.sty and ccool.pdf

3 Issue

```
1. Don't: Inner={\{###1\}}
   Symptom: \CcoolRead fails
   Do: Inner={\char'{###1\char'}}
```

4 Support

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

5 Testing

5.1 Technicality

Not possible to compile-check the expansion of a certain class of macros against predefined values[8]. Instead, one can visually check Part II, as generated in section 2 on one's own machine, against that of the repository for the same version.

5.2 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

5.3 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)

³Under Unix, \$tex ccool.dtx

5.4 Results

- 1. ccool v1.8 satisfactory on platform i) and engine a)
- 2. ccool v1.8 satisfactory on platform i) and engine b)
- 3. ccool v1.9 satisfactory on platform i) and engines b) and c)
- 4. ccool v1.9 satisfactory on platform i) and engines b), c), and d)

5.5 Other

Check [5] for testing coool with Ilncs

References

- [1] Nick Setzer The cool package, 2005, https://www.ctan.org/pkg/cool
- [2] The LATEX3 Project Team *The LATEX3 interfaces*, 2019, http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3kernel/interface3.pdf
- [3] Thomas F. Sturm *The tcolorbox package*, 2019, http://www.texdoc.net/texmf-dist/doc/latex/tcolorbox/tcolorbox.pdf
- [4] The LATEX3 Project Team *The xparse package*, 2020, http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3packages/xparse.pdf
- [5] Erwann Rogard and Olympia Hadjiliadis *Typesetting a math thesis with ccool*, 2020, https://github.com/rogard/ccool/blob/master/thesis.pdf
- [6] https://tex.stackexchange.com/users/112708/erwann?tab=questions
- [7] @sean-allred's answer to "How to create lambda expressions?", https://tex.stackexchange.com/a/188053/112708
- [8] @joseph-wright's answer to "Checking a function's expansion against a string", https://tex.stackexchange.com/a/534100
- [9] @frougon's answer to "Journaling calls to a function []", https://tex.stackexchange.com/a/536620

Change History

v1.0		Added:\OopsRestore	13
General: Initial version	13	$Added: \verb \logsTest \ldots \ldots \ldots$	13
v1.1		Deleted: Listing 1-5 from v1.0 \dots	13
General: Added: Save	13	Fixed: apparent anomaly in v1.0's	
Added: Listing 1., 2., 3., 4., 6., and		Listing 4, see Listing 3	13
9	13	Replaced:	

$\Omega = \Omega $	v1.6
Replaced:	General: Added: Listing 2 (preamble) 13
$\{\langle kvl_2\rangle\}$ by $\langle kvl_2\rangle$ given that	Renamed: \OopsClear to
option type G not recommended[4] 13	\CcoolClear 13
Replaced: GenericObject by Name 13	Renamed: \OopsDebug to
Replaced: Separators by Separ 13	\CcoolDebug 1
Revamped: much of the	Renamed: \OopsHook to
implementation	\CcoolHook
v1.2	Renamed: \OopsOption to
General: Deleted: \OopsTest 13	\CcoolOption 13
Deleted: $\langle kvl_2 \rangle$ and $\langle code_2 \rangle$ 13	Renamed: \OopsRead to
Deleted: Listing 2-3 from v1.1 13	\CcoolRead
Replaced: \OopsClear{ $\langle tl_1 \rangle$ } by	Renamed: \Oops to \Ccool 13
\text{OopsClear}[$\langle keyval list \rangle$] 13	Renamed: oops to cool (better
Replaced: \Restore by \Read 13	describes the purpose)
Replaced: \Save by \Write 13	v1.7
$\sqrt{1.3}$	General: Added: Legends to listings . 13
· -	Added: Listing 12 (CUSUM) 13
General: Replaced: \OopsNew by \Oops 13	Deleted: \CcoolDebug 13
Replaced: $\{\langle tl_1 \rangle\}$ and $[\langle tl_1 \rangle]$ by	Deleted: Listing 5 from v1.6 13
$\langle \langle tl_1 \rangle \rangle$	v1.8
v1.4	General: Added: \CcoolLambda 1
General: Added: section 4	Added: \CcoolVers 1
Added: \OopsDebug 13	Added: Listing 10, Listing 11 13
Added: \OopsHook 13	Added: Listing 1 13
Added: Expans (for debugging'	v1.9
sake, but)	General: Added: support for LuaTFX 13
Added: Listing 1., 2., and 3. \dots 13	Moved: from Part I to Part IV,
Deleted: Listing 1., and 2 13	what is now that part's section 11 13
Replaced: $s\{\{\langle tl_3\rangle\}\{\langle tl_4\rangle\}\{\langle tl_5\rangle\}\}$	v2.0
by	General: Added: support for XTTFX . 13
$s\{\{\langle tl_3\rangle\} \{\langle tl_3\rangle\}\{\langle tl_4\rangle\} \{\langle tl_3\rangle\}\{\langle tl_4\rangle\}\{\langle tl_4\rangle\}\}$	Deleted: File's dependency on
	texosquery and \pdfcreationdate 1
v1.5	Updated: \RequirePackage,
General: Added: File	\NeedsTeXFormat's second
Deleted: dependence on datetime . 13	argument / TeX Live 2020 1

\mathbf{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.

$\mathbf{Symbols}$	$\langle t1_4 \rangle$ (option)
* (option)	$\langle tl_5 \rangle$ (option)
+ (option)	$\langle tl_6 \rangle$ (option)
$\langle \text{key}_i \rangle$	Expans (option)
$\langle code_1 \rangle$ (option)	File (option)
$\langle code_2 \rangle$ (option)	Inner (option)
$\langle \mathtt{kvl}_1 \rangle$ (option)	Name (option)
$\langle tl_1 \rangle$ (option)	Outer (option)
$\langle t1_2 \rangle$ (option)	Separ (option)
$\langle \mathtt{tl}_3 \rangle$ (option)	Write (option)

	\ccool_make_ccool_exp:nnn
\⊔ 411, 412	\ccool_make_ccool_key:nnn
,	\ccool_make_ccool_sideeffect:nnn
A	171, 192, 201
\AtEndDocument 93	\ccool_make_key:N 143, 169
_	\ccool_make_key:n 138, 145
В	\ccool_make_key:Nn 128, 140
\begin	\g_ccool_option_expans_tl 31, 35, 349
\begingroup 178	\ccool_option_inner:n 217, 221, 358
bool commands:	\g_ccool_option_inner_tl
\bool_gset_false:N 98	
\bool_gset_true:N 105	\ccool_option_name:n <u>225</u> , 372
\bool_if:nTF 120, 174, 197, 414	\g_ccool_option_name_tl
\bool_set_false:N 94	132, 227, 233, 332, 362, 376, 390, 404
\BooleanFalse 419, 420	\ccool_option_outer:n 235, 386
\mathbf{C}	\g_ccool_option_outer_tl
\Ccool 1, 4, 5, 5, 14, 28, 179, 186, 207	237, 241, 365, 379, 393, 407
ccool internal commands:	\ccool_option_separ:n 243, 400
\ccool_aux_inner:n 5, 6, 37, 217	\g_ccool_option_separ_tl
_ccool_aux_inner_set:n $\frac{3}{2}$, $\frac{164}{164}$	245, 249, 364, 378, 392, 406
\ccool_aux_key:N 16, 168	\ccool_prop_append:NN 251, 262
\ccool_aux_key:n 12, 19	_ccool_prop_append:Nn 167, 260
\ccool_aux_key:w	_ccool_prop_append:nn 253, 257
\gccool_aux_key_seq	\ccool_prop_clear_new:n <u>264, 271</u>
	_ccool_prop_clear_new_map:n
\gccool_aux_keyval_seq 165, 166, 168	
\ccool_aux_outer:n 23, 151	\ccool_prop_if_exist:nTF . 161, 273
$\c \sum_{\text{ccool}} aux_{\text{outer}} = 1$	\ccool_prop_item:nn 134, 277
\gccool_aux_prop 25, 28, 45, 167	\ccool_prop_name:n
\ccool_aux_prop:N <u>43</u> , 166	82, 262, 266, 275, 279, <u>281,</u> 284
\cdots _ccool_aux_prop:n $39, 49$	\ccool_prop_new:n 163, 282
\cdots ccool_aux_prop:nn 25	\ccool_seq_from_prop:n 288, 292
\cdots _ccool_aux_prop:w $\underline{31}$, $\underline{41}$	\ccool_seq_from_prop:NNn 82, 286
$\c \c \$	\ccool_seq_use:Nn 154, 298
\ccool_aux_separ:nn $\dots \underline{52}, 77$	\ccool_sys_date: 304, 314
\ccool_aux_val:Nn	\ccool_sys_date_hex: 313, 328
\gccool_aux_val_seq 81, 82, 155	\ccool_sys_filename: . 325, 355, 356
_ccool_erw_seq_use:Nn 294	_ccool_sys_time: 315, 324
\ccool_lambda:nn	\ccool_sys_time_hex: 323, 329
_ccool_lambda_expression 87, 90	\CcoolClear 3, 14, 28, 331
\ccool_log_close: <u>92</u> , 416	\CcoolDebug 14
\ccool_log_entry 179, 180	\CcoolHook 3, 5, 14, 28, 199, 336
\g_ccool_log_file_tl . 100, 103, 353	\CcoolLambda 3, 14, 28, 337
\gccool_log_iow	\CcoolOption
	\CcoolRead
\ccool_log_open: <u>100</u> , 415	\CcoolVers
\gccool_log_open_bool	cs commands:
	\cs_generate_variant:Nn
_ccool_log_read:	6, 30, 74, 112, 127, 137, 142, 259
\(\begin{align*} \cdot \	\cs_gset:Npn 5, 23, 245
\ccool_log_write:n <u>117</u> , 176	\cs_new:Nn 52, 75, 225,
\ccool_make_ccool:nnnn	273, 277, 298, 304, 313, 315, 323, 325
	\cs_new:Npn 281
<u> </u>	

$\cs_new_protected:Nn \dots 3, 12, 16,$	\keys_define:nn 346
21, 26, 39, 43, 79, 95, 101, 107, 113,	\keys_set:nn 344
118, 128, 138, 143, 147, 159, 171,	
217, 235, 243, 260, 264, 268, 282, 286	\mathbf{M}
\cs_new_protected:Npn	\meta 295, 296
$\ldots \ldots 8, 32, 84, 184, 251$	msg commands:
\cs_set:Nn 253	\msg_error:nnn 125, 231
\cs_set_protected:Nn 288	\msg_error:nnnn 68
	\msg_new:nnn 211, 212, 213, 214, 215, 216
D	\msg_warning:nnn 223, 241, 249
\DeclareDocumentCommand 87, 186	
\def 179	${f N}$
TD.	\NewDocumentCommand 331, 336, 341, 422, 429
E 207	
\end	0
\endgroup 180	\0ops 14
\ensuremath 397, 398	\OopsClear
exp commands:	\OopsDebug
\exp_args:NNf 153	\OopsHook
\exp_args:NNx 86, 130, 186	\OopsNew
\exp_args:No 164, 266	\OopsOption
\exp_args:Nx 34	\OopsOptions 14
\exp_last_unbraced:Nf	\OopsRead
359, 373, 387, 401	\OopsRestore 13
\exp_last_unbraced:NNf 300	\OopsTest 13, 14
$\ensuremath{\texttt{exp_not:N}}\ \dots \ 70,\ 223,\ 233,\ 241,\ 249$	options:
\exp_not:n 207	* 5
\expandafter 179	+ 5
\ExplSyntaxOff 432	$\langle code_1 \rangle$
\ExplSyntaxOn 2	$\langle {\it code}_2 angle$
T.	$\langle \mathtt{kvl}_1 \rangle$
F	$\langle t1_1 \rangle$
file commands:	$\langle \mathtt{tl}_2 angle \ldots \qquad \qquad$
\file_input:n 109	$\langle t1_3 \rangle$
${f G}$	$\langle t1_4 \rangle$
\gappto 199	$\langle \mathtt{tl}_5 angle \ldots 5$
\gappto 133	$\langle \mathtt{tl}_6 \rangle$
I	Expans
\IfBooleanT 193	File 29
\IfValueT 191, 205	Inner
\IfValueTF	Name 29
int commands:	Outer
\int_case:nnTF 54	Separ
\int_eval:n 306, 317	Write
\int_to_hex:n 314, 324	
iow commands:	P
\iow_close:N 93, 97	\pdfcreationdate 14
\iow_new:N	prg commands:
\iow_now:Nn	\prg_replicate:nn 57, 88
\iow_open:Nn	prop commands:
\item	\prop_clear_new:N 266
(2002)	\prop_gclear_new:N 45
K	\prop_gput:\nn 28, 255
keys commands:	\prop_if_exist:NTF 275
\lambda keys choice tl 340	\nron item:Nn 255 270 200

$\verb \prop_map_function:NN 257 $	\c_sys_jobname_str 327
\prop_new:N	\c_sys_minute_int 320
\ProvideDocumentCommand 131, 337	\c_sys_month_int 309
${f Q}$	\c_sys_year_int 308
quark commands:	${f T}$
$\ensuremath{\mbox{\sc q_stop}} \dots \ \ \ \ \ \ \ $	tl commands:
R.	\c_empty_tl . 47, 66, 162, 182, 204, 336
\Read 14	\tl_count:n 77
•	\tl_gset:Nn 103, 219, 227, 237, 353
\Restore 14	\tl_gset_eq:NN 349
\mathbf{S}	\tl_log:n 110, 123
\Save	\tl_new:N 31, 100, 117
seq commands:	$\t: 10, 36, 37$
\seq_gclear_new:N 18, 81	
\seq_gput_right: Nn 10, 290	\mathbf{U}
\seq_if_empty:NTF 46	use commands:
\seq_map_function:NN	\use:N
19, 49, 145, 271, 292	\use_i:nn 60, 62
\seq_set_from_clist:Nn 165, 270	\use_ii:nn 61
\seq_use:Nnnn 301	\usepackage 4
sys commands:	
\c_sys_day_int 310	\mathbf{W}
\c_sys_hour_int 319	\Write 14

Part IV

Implementation

```
1 (00=ccool)
                               2 \ExplSyntaxOn
                             1
                                   aux
\__ccool_aux_inner_set:n #1: \langle code \rangle
                               _3 \cs_new_protected:Nn \__ccool_aux_inner_set:n
                                   \cs_gset:Npn \__ccool_aux_inner:n ##1 {#1}
                                   \cs_generate_variant:Nn \__ccool_aux_inner:n { e }
                             (End\ definition\ for\ \_\_ccool\_aux\_inner\_set:n.)
      \__ccool_aux_key:w #1: \langle key \rangle
                             #2 : ⟨ value ⟩
                               8 \cs_new_protected:Npn \__ccool_aux_key:w #1 = #2 \q_stop
                                  \seq_gput_right:Nx \g__ccool_aux_key_seq { \tl_trim_spaces:n{#1} }
                              11 }
                             (End definition for \__ccool_aux_key:w.)
      \__ccool_aux_key:n #1: \langle key = value \rangle
                              12 \cs_new_protected:Nn \__ccool_aux_key:n
                                   \__ccool_aux_key:w #1 \q_stop
                             (End\ definition\ for\ \_\_ccool\_aux\_key:n.)
      \__ccool_aux_key:N #1: \langle seq \rangle
                              {\tt 16} \ \ \verb|\cs_new_protected:Nn \ \\__ccool_aux_key:N \\
                                   \seq_gclear_new:N \g__ccool_aux_key_seq
                                   \seq_map_function:NN #1 \__ccool_aux_key:n
                             (End definition for \__ccool_aux_key:N.)
\__ccool_aux_outer_set:n #1: \langle inline code \rangle
                              21 \cs_new_protected:Nn \__ccool_aux_outer_set:n
                                   \cs_gset:Npn \__ccool_aux_outer:n ##1 {#1}
```

 $(End\ definition\ for\ \verb|__ccool_aux_outer_set:n.|)$

```
\__ccool_aux_prop:nn
                           25 \prop_new:N \g__ccool_aux_prop
                            {\tt ^{26}\ \backslash cs\_new\_protected:Nn\ \backslash\_ccool\_aux\_prop:nn}
                                 \prop_gput:Nnn \g__ccool_aux_prop{#1}{#2}
                           28
                           29 }
                            30 \cs_generate_variant:Nn \__ccool_aux_prop:nn { eo, ee, ex, xo, xe, xx }
                           (End\ definition\ for\ \_\_ccool\_aux\_prop:nn.)
  \__ccool_aux_prop:w #1: \langle key \rangle
                           #2: \langle value \rangle
                            31 \tl_new:N \g__ccool_option_expans_tl
                            32 \cs_new_protected:Npn \__ccool_aux_prop:w #1 = #2 \q_stop
                            33 {
                                 \exp_args:Nx
                            34
                                \use:c{__ccool_aux_prop:\g__ccool_option_expans_tl}
                                { \tl_trim_spaces:n{#1} }
                                 { \__ccool_aux_inner:n{ \tl_trim_spaces:n{#2} } }
                            37
                            38 }
                           (End\ definition\ for\ \verb|\_\_ccool\_aux\_prop:w.|)
  \__ccool_aux_prop:n #1: \langle key = value \rangle
                           39 \cs_new_protected:Nn \__ccool_aux_prop:n
                                 \__ccool_aux_prop:w #1 \q_stop
                           42 }
                           (End definition for \__ccool_aux_prop:n.)
  \__ccool_aux_prop:N #1: \langle keyval \ list \rangle
                            43 \cs_new_protected:Nn \__ccool_aux_prop:N
                                 \prop_gclear_new:N \g__ccool_aux_prop
                                \seq_if_empty:NTF #1
                                 { \c_empty_tl }
                            48
                                   \seq_map_function:NN #1 \__ccool_aux_prop:n
                            49
                                }
                           50
                           51 }
                           (End\ definition\ for\ \verb|\_\_ccool\_aux\_prop:N.)
\__ccool_aux_separ:nn #1: \langle int \rangle
                           #2: \langle tokens \rangle
                            52 \cs_new:Nn \__ccool_aux_separ:nn
                                 \int_case:nnTF {#1}
                            55
                                 {
                                   {1}
                            56
                                   { \prg_replicate:nn{ 3 }{#2} }
                            57
                                   {2}
                            58
                                   {
                            59
```

```
{ \use_i:nn #2 }
                                  { \use_ii:nn #2 }
                         61
                                  { \use_i:nn #2 }
                         62
                         63
                                {3}{#2}
                         64
                             }
                         65
                              { \c_empty_tl }
                         66
                         67
                                \msg_error:nnnn { __ccool }
                                { separ }
                                { \exp_not:N \__ccool_aux_separ:nn }
                         72
                         73 }
                         74 \cs_generate_variant:Nn \__ccool_aux_separ:nn { e }
                        (End definition for \__ccool_aux_separ:nn.)
\__ccool_aux_separ:n #1: \langle tokens \rangle
                         75 \cs_new:Nn \__ccool_aux_separ:n
                              \__ccool_aux_separ:en{ \tl_count:n{#1} }{#1}
                         78 }
                        (End\ definition\ for\ \verb|\__ccool_aux_separ:n.|)
 \__ccool_aux_val:Nn #1: \langle seq \rangle
                        #2: \langle tl \ var \ name \rangle
                         79 \cs_new_protected:Nn \__ccool_aux_val:Nn
                         80 {
                              \seq_gclear_new:N \g__ccool_aux_val_seq
                         81
                              \label{lem:cool_seq_from_prop:NNn } $$ \ccool_aux_val_seq #1 { \ccool_prop_name:n{#2} } $$
                         82
                         83 }
                        (End\ definition\ for\ \verb|\__ccool_aux_val:Nn.|)
                        2
                              lambda
  \__ccool_lambda:nn
                         84 \cs_new_protected:Npn \__ccool_lambda:nn #1 #2
                         85 {
                              \exp_args:NNx
                              \DeclareDocumentCommand \__ccool_lambda_expression
                              { \prg_replicate:nn { #1 } { m } }
                              {#2}
                              91 }
                        (End\ definition\ for\ \_\_ccool\_lambda:nn.)
```

$3 \log$

```
\__ccool_log_close:
                       92 \iow_new:N \g__ccool_log_iow
                       93 \AtEndDocument{\iow_close:N \g__ccool_log_iow}
                       94 \bool_set_false:N \g_ccool_log_open_bool
                       95 \cs_new_protected:Nn \__ccool_log_close:
                            \in \g_ccool_log_iow
                       97
                            \bool_gset_false:N \g__ccool_log_open_bool
                       98
                       99 }
                      (End definition for \__ccool_log_close:.)
  \__ccool_log_open:
                       100 \tl_new:N \g__ccool_log_file_tl
                       101 \cs_new_protected:Nn \__ccool_log_open:
                       102
                            \tl_gset:Nx \g_ccool_log_to_tl{\g_ccool_log_file_tl}
                       103
                            \iow_open:Nn \g__ccool_log_iow {\g__ccool_log_to_tl}
                       104
                            \bool_gset_true:N \g__ccool_log_open_bool
                       105
                       106 }
                      (End\ definition\ for\ \_\_ccool\_log\_open:.)
 \__ccool_log_read:n #1: \langle path \rangle
                       \label{loss_new_protected:Nn } $$ \cs_new_protected:Nn \\ -\ccool_log_read:n $$
                       108 {
                            \file_input:n{#1}
                            \tl_log:n{read~from~#1}
                       110
                       111 }
                       112 \cs_generate_variant:Nn \__ccool_log_read:n { e }
                       (End\ definition\ for\ \_\_ccool\_log\_read:n.)
  \__ccool_log_read:
                       113 \cs_new_protected:Nn \__ccool_log_read:
                            \__ccool_log_read:e{\g__ccool_log_to_tl}
                       115
                       116 }
                      (End\ definition\ for\ \verb|\_\_ccool\_log\_read:.)
\__ccool_log_write:n
                       118 \cs_new_protected:Nn \__ccool_log_write:n
                       119 {
                            \bool_if:nTF{ \g__ccool_log_open_bool }
                       120
                       121
                              \iow_now:Nn \g__ccool_log_iow {#1}
                       122
                              \tl_log:n{ write~to~#1 }
                       123
                       125
                            126 }
                       127 \cs_generate_variant:Nn \__ccool_log_write:n { e }
                      (End definition for \__ccool_log_write:n.)
```

4 make_key

```
\__ccool_make_key:Nn #1: \langle token \rangle
                                #2: \langle key \rangle
                                128 \cs_new_protected:Nn \__ccool_make_key:Nn
                                129 {
                                      \exp_args:NNx
                                130
                                     \ProvideDocumentCommand{#1}
                                131
                                     { D<>{\g_ccool_option_name_tl} }
                                        134
                                     }
                                135
                                136 }
                                137 \cs_generate_variant:Nn \__ccool_make_key:Nn {c}
                                (End definition for \__ccool_make_key:Nn.)
         \__ccool_make_key:n #1: \langle key \rangle
                                138 \cs_new_protected:Nn \__ccool_make_key:n
                                      \cline{1}{make_key:cn{#1}{#1}}
                                141 }
                                142 \cs_generate_variant:Nn \__ccool_make_key:n { e }
                                (End\ definition\ for\ \verb|\__ccool_make_key:n.|)
         \__ccool_make_key:N #1: \langle seq \rangle
                                143 \cs_new_protected:Nn \__ccool_make_key:N
                                144
                                      \seq_map_function:NN #1 \__ccool_make_key:e
                                145
                                146 }
                                (End\ definition\ for\ \_\_ccool\_make\_key:N.)
                                5
                                     make_ccool
\__ccool_make_ccool_exp:nnn
                                147 \cs_new_protected:Nn \__ccool_make_ccool_exp:nnn
                                148 {
                                      \__ccool_aux_val:Nn \g__ccool_aux_key_seq {#1}
                                149
                                      \__ccool_aux_outer_set:n{#3}
                                150
                                      \__ccool_aux_outer:n
                                151
                                152
                                        \exp_args:NNf
                                153
                                        \__ccool_seq_use:Nn
                                154
                                        \g__ccool_aux_val_seq
                                155
                                        {#2}
                                156
                                     }
                                157
                                158 }
                                (End\ definition\ for\ \_\_ccool\_make\_ccool\_exp:nnn.)
```

```
\__ccool_make_ccool_key:nnn
                               159 \cs_new_protected:Nn \__ccool_make_ccool_key:nnn
                               160 {
                                     \__ccool_prop_if_exist:nTF{#1}
                               161
                                     { \c_empty_tl }
                               162
                                     { \__ccool_prop_new:n{#1} }
                                163
                                     \exp_args:No \__ccool_aux_inner_set:n{#2}
                                164
                                     \seq_set_from_clist:Nn \g__ccool_aux_keyval_seq {#3}
                                165
                                     \__ccool_aux_prop:N \g__ccool_aux_keyval_seq
                                     \__ccool_prop_append:Nn \g__ccool_aux_prop {#1}
                                     \__ccool_aux_key:N \g__ccool_aux_keyval_seq
                                     \__ccool_make_key:N \g__ccool_aux_key_seq
                                169
                               170 }
                               (End\ definition\ for\ \verb|\__ccool_make_ccool_key:nnn.|)
                               [9]
     \__ccool_make_ccool_sideeffect:nnn
                               172 {
                                     \cline{1}{make_ccool_key:nnn{#1}{#2}{#3}}
                               173
                                     \bool_if:nTF{ \g__ccool_log_open_bool }
                               174
                                175
                                       \__ccool_log_write:n
                                176
                                177
                                         \begingroup
                                         \label{locality} $$ \left( \cool<\#1>i\{\#2\}\{\#3\} \right) \exp and after $$
                                179
                                180
                                         \endgroup \__ccool_log_entry
                                     }{\c_empty_tl}
                               182
                               183 }
                               (End\ definition\ for\ \_\_ccool\_make\_ccool\_sideeffect:nnn.)
                               #1: \langle token \ list \rangle
     _ccool_make_ccool:nnnn
                               #2:
                                     \langle seq_1 \rangle
                               #3:
                                     \langle seq_2 \rangle
                               #4:
                                     \langle prop \rangle
                                \cs_new_protected:Npn \__ccool_make_ccool:nnnn #1 #2 #3 #4
                                185 {
                                     \exp_args:NNx \DeclareDocumentCommand \Ccool
                                     {%^^A
                                                               4 5 6 7 8
                                               2
                                                    3
                                187
                                       D<>{#1} +o E{ i }{{#2}} m t+ s E{ s o }{{#3}{#4}} +o
                                188
                                     }
                                189
                                     {
                                190
                                       \IfValueT{##2}{##2}
                                191
                                       \__ccool_make_ccool_sideeffect:nnn{##1}{##3}{##4}
                                192
                                       \IfBooleanT{##6}
                                         \__ccool_make_ccool_exp:nnn{##1}{##7}{##8}
                                       }
                                196
                                       \bool_if:nTF{##5}
                                197
                                198
                                         \gappto{\CcoolHook}
                                199
                                200
```

```
202
                                         }
                                203
                                         {\c_empty_tl}
                                204
                                         \IfValueT{##9}
                                205
                                206
                                            \exp_not:n{ \Ccool<##1>[##9] }
                                207
                                         }
                                      }
                                210 }
                                (End definition for \__ccool_make_ccool:nnnn.)
                                6
                                      msg
                                211 \mbox{ } \mbox{msg_new:nnn } {\_ccool}{ generic }{\#1}
                                \label{local_local} $$\min_{n\in\mathbb{N}} {\max_{n\in\mathbb{N}} {\text{iow }}{\#1}^is^closed^can't^write}$$}
                                \label{local_local} $$\max_{new:nnn {\_ccool}{ keyonly }{\#1$-does$-not$-take$-values;$-keyval$-is$-$\#2}$}
                                214 \msg_new:nnn {__ccool}{ keywrong }{#1~does~not~recognize~key~#2}
                                \label{local_local} $$\max_{new:nnn {\_ccool}{ separ }{\#1\sim expects\sim 1\sim to\sim 3\sim items,\sim \#2}$}
                                216 \msg_new:nnn {__ccool}{ unset }{#1~unset}
                                7
                                      option
   \__ccool_aux_inner:n #1: \langle code \rangle
                                217 \cs_new_protected:Nn \__ccool_option_inner:n
                                218 {
                                219
                                      \tl_gset:Nn \g__ccool_option_inner_tl {#1}
                                220 }
                                221 \cool_option_inner:n
                                222 {
                                      \msg_warning:nnn{ __ccool }{ unset }{ \exp_not:N \g__ccool_option_inner_tl }
                                223
                                224 }
                                (End\ definition\ for\ \verb|\__ccool_aux_inner:n.|)
 \cdot = -ccool_option_name:n #1: \langle token \ list \rangle
                                225 \cs_new:Nn \__ccool_option_name:n
                                      \tl_gset:Nn \g__ccool_option_name_tl{#1}
                                227
                                228 }
                                \label{eq:cool_option_name:n} $$ $$ \sum_{n=0}^{\infty} \sum_{n=0}^{\infty} (n^{n}) $$
                                230 {
                                      \msg_error:nnx{ __ccool }
                                231
                                      { generic }
                                      { \exp_not:N\g_ccool_option_name_tl~undefined }
                                234 }
                                (End definition for \__ccool_option_name:n.)
\__ccool_option_outer:n #1: \langle inline code \rangle
                                235 \cs_new_protected:Nn \__ccool_option_outer:n
                                236 {
```

 $\cline{1.5} \cline{1.5} \cli$

201

```
\tl_gset:Nn \g__ccool_option_outer_tl {#1}
                                238 }
                                     __ccool_option_outer:n
                                239
                                240 {
                                      \msg_warning:nnn{ __ccool }{ unset }{ \exp_not:N \g__ccool_option_outer_tl }
                                241
                                242 }
                               (End\ definition\ for\ \verb|\_\_ccool\_option\_outer:n.|)
  \__ccool_option_separ:n #1: \{\langle tl_1 \rangle\}\{\langle tl_2 \rangle\}\{\langle tl_3 \rangle\}
                                243 \cs_new_protected:Nn \__ccool_option_separ:n
                                      \cs_gset:Npn \g__ccool_option_separ_tl {#1}
                                246 }
                                     __ccool_option_separ:n
                                248
                                      \msg_warning:nnn{ __ccool }{ unset }{ \exp_not:N \g__ccool_option_separ_tl }
                                249
                                250 }
                               (End\ definition\ for\ \verb|\_\_ccool\_option\_separ:n.)
                                8
                                     prop
  \__ccool_prop_append:NN
                               #1: \langle prop_1 \rangle
                               #2: \langle prop_2 \rangle
                                251 \cs_new_protected:Npn \__ccool_prop_append:NN #1 #2
                                252 {
                                      \cs_set:Nn \__ccool_prop_append:nn
                                253
                                        \prop_gput:Nnx #1 {##1}{ \prop_item:Nn #2{##1} }
                                      \prop_map_function:NN #2 \__ccool_prop_append:nn
                                257
                                258 }
                                259 \cs_generate_variant:Nn \__ccool_prop_append:NN { cN }
                               (End\ definition\ for\ \_\_ccool\_prop\_append:NN.)
  \__ccool_prop_append:Nn #1: \langle prop \rangle
                               #2: \langle tl \ var \ name \rangle
                                260 \cs_new_protected:Nn \__ccool_prop_append:Nn
                                261 {
                                      \__ccool_prop_append:cN{ \__ccool_prop_name:n {#2} } #1
                                262
                                263 }
                               (End\ definition\ for\ \verb|\__ccool\_prop\_append:Nn.)
\__ccool_prop_clear_new:n #1: \langle tl var name \rangle
                                264 \cs_new_protected:Nn \__ccool_prop_clear_new:n
                                      \exp_args:No \prop_clear_new:c{ \__ccool_prop_name:n {#1} }
                                266
                                267 }
                               (End definition for \__ccool_prop_clear_new:n.)
```

```
\_ccool_prop_clear_new_map:n #1: \langle keyval \ list \rangle
                                   268 \cs_new_protected:Nn \__ccool_prop_clear_new_map:n
                                         \seq_set_from_clist:Nn \g__ccool_aux_key_seq {#1}
                                         \seq_map_function:NN \g__ccool_aux_key_seq \__ccool_prop_clear_new:n
                                   271
                                   272 }
                                   (End\ definition\ for\ \verb|\__ccool_prop_clear_new_map:n.|)
\__ccool_prop_if_exist:nTF #1: \langle tl_1 \rangle
                                   #2: \langle tl_2 \rangle
                                   #3: \langle tl_3 \rangle
                                   273 \cs_new:Nn \__ccool_prop_if_exist:nTF
                                         \prop_if_exist:cTF{ \qrderight \qrderight = ccool_prop_name:n $$\{\#1\} $$} $$
                                   276 }
                                   (End definition for \__ccool_prop_if_exist:nTF.)
      \__ccool_prop_item:nn #1: \langle tl var name \rangle
                                   #2: \langle key \rangle
                                   277 \cs_new:Nn \__ccool_prop_item:nn
                                         \prop_item:cn { \qrupe -ccool_prop_name:n {#1} } {\#2}
                                   279
                                   280 }
                                   (End\ definition\ for\ \verb|\__ccool_prop_item:nn.|)
       \__ccool_prop_name:n #1: \langle tl var name \rangle
                                   \label{local_new:Npn local_prop_name:n #1{ __ccool_#1 }} $$ $$ \cs_new:Npn \ \__ccool_prop_name:n #1{ __ccool_#1 }$
                                   (End\ definition\ for\ \verb|\_\_ccool\_prop\_name:n.)
        \__ccool_prop_new:n #1: \langle tl var name \rangle
                                   282 \cs_new_protected:Nn \__ccool_prop_new:n
                                         \prop_new:c{ \__ccool_prop_name:n {#1} }
                                   284
                                   285 }
                                   (End\ definition\ for\ \verb|\_\_ccool\_prop\_new:n.|)
                                         seq
\__ccool_seq_from_prop:NNn #1: \langle seq_1 \rangle
                                   #2: \langle seq_2 \rangle (keys)
                                   #3: \langle prop \rangle
                                   286 \cs_new_protected:Nn \__ccool_seq_from_prop:NNn
                                         \cs_set_protected:Nn \__ccool_seq_from_prop:n
                                   289
                                         {
                                            290
                                   291
                                         \seq_map_function:NN #2 \__ccool_seq_from_prop:n
                                   292
                                   293 }
```

```
(End\ definition\ for\ \verb|\_\_ccool\_seq\_from\_prop:NNn.)
\__ccool_erw_seq_use:Nn
                             294 %
                                         \begin{arguments}
                             295 %
                                         \item \meta{ seq }
                                         \item \meta{ tokens }
                             296 %
                                         \verb|\end{arguments}|
                             297 %
                             ^{298} \cs_new:\n \__ccool_seq_use:\n
                             299 {
                                   \exp_last_unbraced:NNf
                             300
                                   \seq_use:Nnnn #1
                             301
                                   \__ccool_aux_separ:n{#2}
                             302
                             303 }
                             (End\ definition\ for\ \verb|\__ccool_erw_seq_use:Nn.|)
                             10
                                     \mathbf{sys}
     \__ccool_sys_date:
                             304 \cs_new:Nn \__ccool_sys_date:
                             305 {
                                   \int_eval:n
                             306
                                   {
                             307
                                     \c_sys_year_int * 10000
                                     +\c_sys_month_int * 100
                                     +\c_sys_day_int * 1
                             310
                                  }
                             311
                             312 }
                             (End\ definition\ for\ \verb|\_\_ccool\_sys\_date:.)
 \__ccool_sys_date_hex:
                             313 \cs_new:Nn \__ccool_sys_date_hex:
                             314 {\int_to_hex:n{\__ccool_sys_date:}}
                             (End definition for \__ccool_sys_date_hex:.)
     \__ccool_sys_time:
                             315 \cs_new:Nn \__ccool_sys_time:
                             316 {
                                   \int_eval:n
                             317
                             318
                                     \c_sys_hour_int * 100
                             319
                                     +\c_sys_minute_int * 1
                             321
                             322 }
                             (End definition for \__ccool_sys_time:.)
 \__ccool_sys_time_hex:
                             323 \cs_new:Nn\__ccool_sys_time_hex:
                             324 {\int_to_hex:n{\__ccool_sys_time:}}
                             (End definition for \__ccool_sys_time_hex:.)
```

```
\__ccool_sys_date_hex:--
                                                            328
                                                                               \__ccool_sys_time_hex:
                                                            330 }
                                                           (End\ definition\ for\ \verb|\_\_ccool\_sys\_filename:.)
                                                           11
                                                                                       Front-end
   \CcoolClear
                                                         #1: \langle token \ list \rangle
                                                           Semantics Clears any data created by \cool{decool}{decool}
                                                            331 \NewDocumentCommand{ \CcoolClear }
                                                            _{\rm 332} { D<>{\g_ccool_option_name_t1} }
                                                                                 \__ccool_prop_clear_new_map:n{#1}
                                                            335 }
        \CcoolHook
                                                           Example \AfterEndEnvironment{theorem}{\CcoolHook}
                                                             336 \NewDocumentCommand{\CcoolHook}{}{\c_empty_tl}
\CcoolLambda
                                                           #1: \langle integer \rangle
                                                           #2: \langle code \rangle
                                                          Example \Ccool{ EvalAt = \CcoolLambda{(#1)} }
                                                           Semantics Creates a lambda expression with \langle integer \rangle arguments for \langle code \rangle
                                                            \mbox{\coolLambda} \ProvideDocumentCommand \CcoolLambda { O{1} m }
                                                                               \cline{1.5cm} 
                                                            340 }
```

325 \cs_new:Nn__ccool_sys_filename:

\c_sys_jobname_str--

326 {

__ccool_sys_filename:

```
\CcoolOption
               #1: \langle keyval \ list \rangle
               341 \NewDocumentCommand{ \CcoolOption }
                343 {
                     \keys_set:nn{ __ccool }{#1}
               344
               345 }
               346 \keys_define:nn { __ccool }
               347 {
      Expans
               Value eo|ee|ex|xo|xe|xx
               348 Expans .multichoices:nn = { eo, ee, ex, xo, xe, xx }
                349 { \tl_gset_eq:NN \g__ccool_option_expans_tl \l_keys_choice_tl },
                350 Expans .default:n = { xo },
                351 Expans .initial:n = { xo },
        File
               Value \langle path \rangle
                352 File .code:n = {
               \tl_gset:Nx \g__ccool_log_file_tl{#1}
               354 },
               355 File .default:n = { \__ccool_sys_filename: },
               356 File .initial:n = { \__ccool_sys_filename: },
       Inner
               Value \langle code \rangle, with ####1 as the argument to be replaced
                357 Inner .code:n={
                     \__ccool_option_inner:n{#1}
                     \exp_last_unbraced:Nf
                350
                     \__ccool_make_ccool:nnnn
                360
                361
                       { \g_ccool_option_name_tl }
                362
                       { \g_ccool_option_inner_tl }
                363
                       { \g_ccool_option_separ_tl }
                       { \g_ccool_option_outer_tl }
               367 },
               368 Inner .value_required:n = false,
               369 Inner .default:n = {###1},
               370 Inner .initial:n = {####1},
        Name
               Value \( \text{token list} \)
               371 Name .code:n={
                   \__ccool_option_name:n{#1}
                    \exp_last_unbraced:Nf
                    \__ccool_make_ccool:nnnn
               374
                    {
               375
```

```
{ \g_ccool_option_inner_tl }
        377
                { \g_ccool_option_separ_tl }
        378
                { \g_ccool_option_outer_tl }
        379
        380
        381 },
        382 Name .value_required:n = false,
        383 Name .default:n = { Math },
        384 Name .initial:n = { Math },
Outer
        Value \langle code \rangle, with ####1 as the argument to be replaced
        385 Outer .code:n={
             \__ccool_option_outer:n{#1}
             \exp_last_unbraced:Nf
             \__ccool_make_ccool:nnnn
        388
             {
        389
                { \g__ccool_option_name_tl }
        390
                { \g_ccool_option_inner_tl }
        391
                { \g_ccool_option_separ_tl }
                { \g_ccool_option_outer_tl }
        395 },
        396 Outer .value_required:n = false,
        397 Outer .default:n = { \left\{ \text{ensuremath} \right\} \},
        398 Outer .initial:n = { \ensuremath{####1} },
Separ
        Value That of 'separators' in [2, Section 8 of I3seq]
        399 Separ .code:n={
             \__ccool_option_separ:n{#1}
        400
             \exp_last_unbraced:Nf
        401
             \__ccool_make_ccool:nnnn
        403
                { \g_ccool_option_name_tl }
                { \left\{ \ \ \ \ \ \ \right. }
                { \g_ccool_option_separ_tl }
        406
                { \g_ccool_option_outer_tl }
        407
             }
        408
        409 },
        410 Separ .value_required:n = false,
        411 Separ .default:n = { {\ }and{\ } } { ,{\ } } { ,{\ }and{\ } },
        412 Separ .initial:n = { {\ }and{\ } } { ,{\ } } { ,{\ }and{\ } },
Write
        Value (boolean)
        ^{413} Write .code:n = {
        414 \bool_if:nTF{#1}
             {\__ccool_log_open:}
             {\__ccool_log_close:}
        417 },
```

{ \g_ccool_option_name_tl }

```
418 Write .value_required:n = false,
419 Write .default:n = \BooleanFalse,
420 Write .initial:n = \BooleanFalse
421 }
```

\CcoolRead

#1: $\langle path \rangle$

Semantics

- 1. Reads the definitions in $\langle path \rangle$.
- 2. Writes to ccool.log: 'read from $\langle path \rangle$ '

```
422 \NewDocumentCommand{\CcoolRead}
423 {0}
424 {
425 \IfValueTF{#1}
426 {\__ccool_log_read:e{#1}}
427 {\__ccool_log_read:}
428 }
```

\CcoolVers

Semantics Expands to the package's version

```
429 \NewDocumentCommand{\CcoolVers}
430 {}
431 {\use:c{ver@ccool.sty}}
```

16 Misc

432 \ExplSyntaxOff