## The ccool package\*

## Erwann Rogard<sup>†</sup>

#### Released 2020/10/10

#### Abstract

The package ccool for  $\LaTeX$  is a key-value interface,  $\char`$ Ccool, on top of xparse's document command parser. Global options control input processing and its expansion. By default, they are set to meet likely requirements, depending on context: the selected language, and which of text and math mode is active. These options can be overriden inline. This versality could find its use, for example, to encode notational conventions (such as  $\Real \to \Relectric Relations)$ ) at the point where they are introduced in the document ("Let  $\R$  denote real numbers"). Polymorphic commands can be generated by parameterizing the keys (for instance, one parameter value for style, another for a property). User input to  $\char`$ Ccool can optionally be serialized. This can useful for typesetting documents sharing the same notation.

#### Résumé

L'extension ccool pour LATEX met à disposition une interface de type clé-valeur, \Ccool, destinée à faciliter la géneration de commandes. Les paramètres globaux contrôlant le traitement de ces clé-valeur sont fixés par défaut pour répondre aux besoins courants, suivant le contexte (langage, mode textuel ou mathématique). Un exemple d'application, est la command-isation des conventions de notation (\Reel  $\rightarrow \rightarrow \mathbb{R}$ ), au point dans le document où elles sont introduites ("Soit  $\mathbb{R}$  les nombres réels."). Des commandes polymorphes peuvent être générées, en associant aux clés un paramètre (par exemple, une valeur pour le style typographique, une autre pour la description du concept associé). En option, les instructions passées à cette interface peuvent être sauvegardées, ce qui peut être utile pour la rédaction de documents faisant appel à des conventions typographiques communes.

#### Contents

Ι	Usage	5
0	Convention	5
1	Loading the package	5

<sup>\*</sup>This file describes version v3.1, last revised 2020/10/10.

<sup>†</sup>firstname dot lastname AusTria gmail dot com

2	\Ccool		5			
	2.1 Core feature		5			
	2.2 Process the val <sub>i</sub> s		6			
	2.3 Append to a hook		6			
	2.4 Expand the $val_i$ s		6			
	2.5 Head		6 6			
	2.6 Tail		6			
	2.8 Write		6			
	2.0 111100	• •	Ü			
3	\CcoolClear		6			
4	\CcoolHook		6			
5	\CcoolLambda		7			
6	\CcoolOption		7			
	6.1 And		7			
	6.2 Expans		7			
	6.3 File		7			
	6.4 Inner		7			
	6.5 Param		7			
	6.6 Outer		7			
	6.7 Separ		8			
	6.8 Write		8			
7	\CcoolRead		8			
8	\CcoolVers		8			
9	Do's and dont's		8			
II	Listing		10			
1.\	CcoolVers		10			
2. This doc's dependencies 10						
3.	Let $\mathbb N$ and $\mathbb R$ denotestart of the tutorial)		10			
4. Equivalent to 3, with \NewDocumentCommand 10						
5. Equivalent to 4, with \Ccool						
6. Equivalent to 5, with expansion						
7. Equivalent to 5, parameterized (end of the tutorial)						
8. Language and mode						
u)	Separators 12					

10.	Hello, world! (for testing's sake)	<b>12</b>
11.	Listing 10 read from file	13
<b>12.</b>	Probability space	13
13.	Listing 12 read from file	13
14.	Mittelwertsatz für nariable.	14
<b>15.</b>	Listing 14 read from file	14
<b>16.</b>	Families of polynomial functions	14
<b>17.</b>	Listing 16 read from file	15
18.	Same as Listing 16, but arbitrary number system	15
<b>17.</b>	Listing 18 read from file	15
20.	Fonction et fonctionelle	16
21.	Listing 20 read from file	16
<b>22</b> .	CUSUM statistic.	16
23.	Listing 22 read from file	17
III	Other	19
1	Acknowledgment	19
2	Genealogy	19
3	Install	19
4	Issue	19
5	Support	19
6	Testing         6.1 Technicality       6.2 Platform         6.3 Engine       6.4 Results         6.5 Other       6.5 Other	19 19 20 20 20 20
7	To do	20
8	References	20

Cha	ange History	21
Ind	ex	23
IV	Implementation	26
1	Opening	26
2	aux	26
3	lang	28
4	log	29
5	make_key	30
6	make_ccool	31
7	msg	32
8	option	33
9	prop	34
10	seq	35
11	Front-end  11.1 \CcoolClear  11.2 \CcoolHook  11.3 \CcoolLambda  11.4 \CcoolOption  11.4.1 And  11.4.2 Expans  11.4.3 File  11.4.4 Inner  11.4.5 Param  11.4.6 Outer  11.4.7 Separ  11.4.8 Write  11.5 \CcoolRead	36 36 36 36 36 36 37 37 37 37 38 38
	11.6 \CcoolVers	38

12 Closing 38

#### Part I

# Usage

#### Convention

- a) Loosely, those of [2], for example as to the meaning of token list, which may be abbreviated tl.
- b) Those of [5], for example [arg] is an 'o'-type argument.
- c)  $\langle X \rangle \leftarrow \mathtt{Y} : \operatorname{set} \langle X \rangle$  to  $\mathtt{Y}$
- d)  $\X \to Y$ :  $\X$  expands to Y
- e) If unspecified, the environment in which a macro is to be used is document.

\usepackage

\usepackage{ccool}

#### Requirement

- 1. ccool.sty is in the path of the LATEX engine. See Part III, section 5.
- 2. Put in the preamble

\Ccool

 $\label{eq:ccool} $$ \c(\tl_1) = \langle tl_2 \rangle \cdot \langle tl_3 \rangle \cdot \langle tl_3 \rangle \cdot \langle tl_4 \rangle \cdot$ 

Semantics See subsection 2.1-2.8.

#### 2.1 Core feature

- 1)  $\langle val_i \rangle \leftarrow \text{ } \text{function} \{\langle val_i \rangle \}$
- 2) define  $\langle \langle key_i \rangle$  such that  $\langle \langle key_i \rangle \rightarrow \langle val_i \rangle$ ,

where \function is encoded in global option Inner. For instance, the side effect of \Ccool{ Real = \mathbb{R}} \forall is \Real  $\rightarrow$  \mathbb{R}. To be sparingly used, global option Expans controls the type of expansion of  $\langle key_i \rangle$  and  $\langle val_i \rangle$ .

See \CcoolLambda to allow command  $\langle key_i \rangle$  to take arguments.

#### 2.2 Process the $val_i$ s

**\Ccool**  $c\{\langle code_1\rangle\}\{\langle keyval\ list_1\rangle\}$  is identical to the Core feature, except it overrides Inner.

In our example, if multiple number systems are defined with  $\cool$  (natural, reals, ...), it is more efficient to omit  $\mbox{mathbb{.}}$  inside  $\langle val_i \rangle$ , and instead use  $\colon \mbox{mathbb{#1}}$ , where #1 means "parameter to be replaced".

#### 2.3 Append to a hook

#### 2.4 Expand the $val_i$ s

**\Ccool**{ $\langle keyval\ list_1 \rangle$ }\* supplements the Core feature with the expansion of the  $\langle val_i \rangle$ 's using typesetting rules encoded in *global option* Separ and Outer. The first are *separators* applied to the  $\langle val_i \rangle$ 's to form a *token list*, and the second a function applied to the latter.

They can be overriden inline by appending further  $s\{\langle separators \rangle\}$  and  $c\{\langle code_2 \rangle\}$ , respectively, to the list of arguments.

#### 2.5 Head

 $(\text{Ccool}[\langle tl_1 \rangle] \{\langle keyval \ list_1 \rangle\} \text{ expands } \langle tl_1 \rangle \text{ and executes the Core feature.}$ There may be situations where it is convenient to pass  $\langle tl_1 \rangle$  as empty.

#### 2.6 Tail

 $\cool{\langle keyval \ list_1 \rangle} [\langle tl_6 \rangle] {\langle keyval \ list_2 \rangle}$  is identical to  $\cool{\langle keyval \ list_1 \rangle}$  followed by  $\cool{\langle tl_6 \rangle} {\langle keyval \ list_2 \rangle}$ .

The combination of Core feature, Head, and Tail allows to integrate typesetting and the creation of commands.

#### 2.7 Parameterize the $key_i$ s

 $\langle cool \langle tl_2 \rangle > \{\langle keyval \ list_1 \rangle\}$  is identical to the Core feature, except  $\langle key_i \rangle$  is replaced by  $\langle key_i \langle tl_2 \rangle \rangle$ . The default value of  $\langle tl_2 \rangle$  is encoded in Param. In our example,  $\langle tl_2 \rangle$  could be Style.

#### 2.8 Write

global option Write is identical to the Core feature, except that if Write is set to \BooleanTrue, the code is written to a file whose path is encoded in global option File.

\CcoolClear

Semantics Clears all  $\langle key_i < tl_2 > \rangle$ 's

```
\CcoolHook
                 \CcoolHook
                 Semantics No side effect or expansion
\CcoolLambda
                 \ \coolLambda[\langle arg\ spec \rangle] \{\langle code \rangle\},\
                 where arg spec is by default an 'o'-type argument.
                 Example \Ccool{ EvalAt = \CcoolLambda{(#1)} }
                 Semantics Returns a command of type \DeclareDocumentCommand[5],
\CcoolOption
                 \verb|\CcoolOption[...\langle key_i\rangle|\langle key_i\rangle=\langle val_i\rangle,...]|
                 where \langle key_i \rangle is either of And, Expans, File, Inner, Param, Outer, Separ, and Write.
                 Semantics Modify the default behavior of \Ccool
          And
                 Also see Part IV And
                 Semantics Sets the translation of and in language \langle key \rangle to \langle val \rangle
                 Syntax \( \lambda keyval \ list \rangle \)
       Expans
                 Also see Core feature and Part IV Expans
                 Syntax eo|ee|ex|xo|xe|xx
         File
                 Also see Part I Write and Part IV File
                 Syntax \langle path \rangle
        Inner
                 Also see Process the valis and Part IV Inner
                 Syntax \langle code \rangle, with ####1 as the placeholder
        Param
                 Also see Parameterize the key_is, and Part IV Param
                 Syntax \langle token \ list \rangle
        Outer
                 Also see Expand the valis, and Part IV Outer
                 Default \ensuremath{####1}
```

```
Syntax \langle code \rangle, with ####1 as the placeholder
     Separ
              Also see Expand the valis; Listing 8; and Part IV Separ
              Other Default behavior depends on whether babel and amsmath are loaded
              Syntax That of separators in [2, Section 8 of I3seq]
     Write
              Also see Part I Write and Part IV Write
              Syntax \BooleanFalse \BooleanTrue
\CcoolRead
              \verb|\CcoolRead[|\langle path \rangle|]|
              Also see Part IV \CcoolRead
              Semantics
                       1. Reads the definitions in \langle path \rangle.
                       2. Writes to ccool.log: 'read from \langle path \rangle'
\CcoolVers
              \CcoolVers
              \mathbf{Semantics} \, \to \, \mathrm{the} \, \, \mathrm{package's} \, \, \mathrm{version}
              9
                     Do's and dont's
```

```
1)
Don't: Inner=\{####1\}
Symptom: \CcoolRead fails
Do: Inner={\char'{####1\char'}}
2)
Don't: $\langle key_i \rangle \char \text{$\langle}$
Do: $\langle key_i \rangle \langle \char \text{$\langle}$
3)
Don't: [a, b)
Do: {[}a, b{)}
4)
```

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

5)

Don't: \[x_0,x\]
   Do: \left[x_0,x\right]

6)

Don't: Use 'd'-type or 'e'-type arguments[5] for \CcoolLambda
   Do: Use only 'm'-type and 'o'-type arguments

7)

Don't: \usepackage[spanish] {babel}
   Do: \usepackage[spanish.noquoting] {babel}[11]

8) Also see Part III, section 4
```

#### Part II

# Listing

#### NB:

1. Some statements affect only the output of listings that come after that in which they appear. The demarcation is indicated by  $^A-A--->$  and  $^A-A----$ , where applicable

# Listing 1. \CcoolVers \CcoolVers 2020/10/10 v3.1 cool — A key-value document command parser

```
Listing 2. This doc's dependencies

\[
\text{ExplSyntax0n} \\__ccooldoc_usepackage:} \\
\text{ExplSyntax0ff} \]

\[
\usepackage{amsmath,amssymb} \\
\usepackage[french,german,spanish.noquoting,english]{babel} \\
\usepackage{bookmark,ccool,microtype,tabto,tcolorbox} \\
\usepackage[T1]{fontenc}
```

```
Listing 3. "Let \mathbb{N} and \mathbb{R} denote..." (start of the tutorial)

Let-\mathbb{N} and \mathbb{R} denote the natural and real numbers.

Let \mathbb{N} and \mathbb{R} denote the natural and real numbers.
```

```
Listing 4. Equivalent to 3, with \NewDocumentCommand \DeclareDocumentCommand\Nat{}{\mathbb{N}} \DeclareDocumentCommand\Real{}{\mathbb{R}} \Let~$\Nat$ and $\Real$ denote the natural and real numbers. Let \mathbb N and \mathbb R denote the natural and real numbers.
```

```
Listing 5. Equivalent to 4, with \Ccool

% ^^A--->
\Ccool c{\mathbb{#1}}{ Nat = {N}, Real = {R} }

Let~$\Nat$ and $\Real$~denote the natural and real numbers.

% ^^A<---
\CcoolClear</pre>
```

Let  $\mathbb N$  and  $\mathbb R$  denote the natural and real numbers.

```
Listing 6. Equivalent to 5, with expansion

% ^A--->
\Ccool[Let~]
c{\mathbb{#1}}{ Nat = {N}, Real = {R} }*
[-denote the natural and real numbers.]{}
% ^A<---
\CcoolClear

Let N and R denote the natural and real numbers.
```

```
Listing 7. Equivalent to 5, parameterized (end of the tutorial)

% ^^A--->
  \Ccool<Style>c{\mathbb{#1}}{ Nat = {N}, Real = {R} }

[Let $\Nat<Style>$ and $\Real<Style>$ denote the natural and real
    numbers.]{}

% ^^A<---
  \CcoolClear<Style>

Let N and R denote the natural and real numbers.
```

```
Listing 8. Language and mode
  % ^^A--->
  \textbf{\languagename}{:}^{\cool{ X = x, Y = y }*}
  \begin{otherlanguage}{spanish}
    \CcoolOption[ Separ ]\\
    \textbf{\languagename}{:}~\Ccool{ X = x, Y = y }*
  \end{otherlanguage}\\
  \textbf{\languagename}{:}^{\cool{ X = x, Y = y }*}
  \\[1em]
  \CcoolOption[ Outer = ####1 ]
  \textbf{\languagename}{:}~\Ccool{ X = this, Y = that }*
  \begin{otherlanguage}{spanish}
    \CcoolOption[ Separ ]\\
    \textbf{\languagename}{:}~\Ccool{ X = esto, Y = aquello }*
  \end{otherlanguage}\\
  \t X = this, Y = that }*
  \CcoolOption[ Separ ]\\
  % ^^A<---
  \CcoolOption
english: x and y
spanish: x y y
english: x and y
```

english: this and that spanish: esto y aquello english: this and that

Listing 10. Hello, world! (testing)

```
Listing 9. Separators (Note<sup>a</sup>)
            ^a [\mathrm{bug}]: Removing the closing \CcoolOption subsequently causes inconsistent
separators between text and math mode (case replicated in uncommented form in dtx)
          % ^^A--->
          \label{lem:coolOption} $$ \operatorname{Separ}_{{\ \char\ @\ }_{\ \char\ @\ }_{\ \char\ @\ }_{\ \char\ 
          \Ccool{ X = x, Y = y }*[\]
         { X = x, Y = y }*s{{-\&-}}[\]
         { X = x, Y = y }*s{{,~}{~\&~}}[\[1em]]
         { X = x, Y = y, Z = z }*[\]
         { X = x, Y = y, Z = z }*s{{-\&-}}[\\]
         { X = x, Y = y, Z = z }*s{{,~}{\&~}}[\\]
          { X = x, Y = y, Z = z }*s{{^{\},^{\},^{\}}\\
          % ^^A<---
         \CcoolOption
          \CcoolClear
x @ y
x \& y
x \& y
x \% y @ z
x \& y \& z
x, y, \& z
x, y, \& z
```

#### 

```
\CcoolOption
\CcoolClear
{H}.{e}.{l}.{o}, [world!]
```

```
Listing 11. Listing 10 read from file
  % ^^A--->
  \CcoolRead
  \KeyF<Test>\KeyA<Test>\nobreak
  \KeyG<Test>\KeyA<Test>\nobreak
  \KeyH<Test>\KeyA<Test>\nobreak
  \KeyH<Test>\KeyA<Test>\nobreak
  {\<text>}\nobreak\Key0<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
  % ^^A<---
  \CcoolClear
\{H\}.\{e\}.\{l\}.\{o\}, [world!]
```

```
Listing 13. Listing 12 read from file

% ^^A--->
\CcoolRead \tab $\Omega$ $\Field$ $\Meas$
% ^^A<---
\CcoolClear
```

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

#### Listing 14. Mittelwertsatz für n Variable [4, 17.3]

```
\CcoolOption[ Write = \BooleanTrue ]
% ^^A--->
\selectlanguage{german}
\newtheorem{theorem}{Theorem}
\AfterEndEnvironment{theorem}{\CcoolHook}
\Ccool c{\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\mathbb{N}, -\infty] 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 = { f(x)-f(x_0) }{ x-x_0 } }, Punkt = { xi } }+
  [so dass gilt
  \begin{equation*}
    \Steig = \Grad f(\Punkt)^{\top}
  \end{equation*}
\end{theorem}]
{}
(Check: $\N$, $\Punkt$)
% ^^A<---
\CcoolClear
\CcoolOption
```

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:  $\mathbb{N}, \xi$ )

#### Listing 15. Listing 14 read from file

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

#### Listing 16. Families of polynomial functions

```
\CcoolOption[ Write = \BooleanTrue ]
% ^^A--->
\Ccool c{\mathbb{#1}}{ Nat = {N}, Real = {R} }
[Let~]
{ PolyR = \CcoolLambda[o]{\Real\IfValueT{#1}{_#1}[X] } }
[$\PolyR[n]$ and $\PolyR$, denote the families of polynomial functions on $\Real$, of order $n$ et and their union over $n \in \Nat$, respectively. ]
{}
% ^^A<---
\CcoolClear
\CcoolOption</pre>
```

Let  $\mathbb{R}_n[X]$  and  $\mathbb{R}[X]$ , denote the families of polynomial functions on  $\mathbb{R}$ , of order n et and their union over  $n \in \mathbb{N}$ , respectively.

#### Listing 17. Listing 16 read from file

```
% ^^A---> \CcoolRead \tab $\PolyR[n]$ et $\PolyR$ $ % ^^A<--- \CcoolClear \overline{\mathbb{R}_n[X]} et \overline{\mathbb{R}[X]}
```

#### Listing 18. Same as Listing 16, but arbitrary number system

```
\CcoolOption[ Write = \BooleanTrue ]
% ^^A--->
\selectlanguage{french}
\Ccool c{\mathbb{#1}}{ Corps = {K}, Nat = {N}, Reel = {R} }
[Soient~]
{
    Poly = \CcoolLambda[om]{#2\IfValueT{#1}{_#1}[X] },
    PolyR = \CcoolLambda[o]{\Poly[#1]{\Reel}}
}
[$\Poly[n]{\Corps}$ et $\Poly{\Corps}$, les familles de polyn\^omes sur
    $\Corps$, de degr\'e $n$ et leur union pour $n \in \Nat$,
    respectivement. En particulier,
ils sont d\'enot\'es $\PolyR[n]$ et $\PolyR$, pour $\Corps=\Reel$.]
{}
% ^^A<---
\CcoolClear
\CcoolOption</pre>
```

Soient  $\mathbb{K}_n[X]$  et  $\mathbb{K}[X]$ , les familles de polynômes sur  $\mathbb{K}$ , de degré n et leur union pour  $n \in \mathbb{N}$ , respectivement. En particulier, ils sont dénotés  $\mathbb{R}_n[X]$  et  $\mathbb{R}[X]$ , pour  $\mathbb{K} = \mathbb{R}$ .

```
Listing 22. CUSUM statistic[]

\( \text{CcoolOption[ Write = \BooleanTrue ]} \)
\( \text{\chick} \chick^{A--->} \\
\newtheorem{\definition}{\Definition} \\
\AfterEndEnvironment{\definition}{\CcoolHook} \\
\Ccool{
\suchThat = \{ ;^ \},
\suchTime = \{ t \},
\suchTrace = \{ \text{xi \},
\suchTopT = \{ T \},
\suchTopT = \
```

```
Real = {\mathcal{R}}
               }+*s{{~\in~}}
                [~and~]
                { CUSUMthresh = { \nu } }+*c{$#1\in\Real^{+}$.}
                [~Define the following processes:]
                       LogWald = { u },
                       CUSUMst = { \StopT_{c} },
                       CUSUM = \{ y \},
                       LogWaldInf = { m }
                [\begin{enumerate}
                               \Delta_{\tilde{T}}  = \Scale\Process_{\Time} -
                 \frac{1}{2}\Scale^2\Time$;
                               \Lambda = \prod_{N \in \mathbb{N}} \mathbb{E} A = \lim_{N \in \mathbb{N}} \mathbb{E} A = \lim_{N \in \mathbb{N}} \mathbb{E} A = \mathbb{E
                }\CUSUM_{s} \EvalAt{ \Scale }$.
                \item{
                               $\CUSUM_{\Time}\EvalAt{ \Scale } = \LogWaldInf_{\Time}\EvalAt{
                 \Scale } - \LogWald_{\Time}\EvalAt{ \Scale }\ge0$,
                              which is the CUSUM statistic process.
                       }
                \item{
                                $\CUSUMst \EvalAt{ \Scale, \LogWaldInf } = \inf\left[ \Time \ge 0
                 \SuchThat \CUSUM_{\Time}\EvalAt{\Scale} \ge \LogWaldInf \right]$,
                               which is the CUSUM stopping time.
                       }
               \end{enumerate}
        \end{definition}\par]{}
        (Check: $\Scale$, $\CUSUM$)
        % ^^A<---
        \CcoolClear
        \CcoolOption
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 \geq 0; y_t(\lambda) \geq m], which is the CUSUM stopping time.
(Check: \lambda, y)
```

#### Part III

## Other

#### 1 Acknowledgment

This work has benefited from Q&A's from the LATEX community[7][10]. Specific attributions are made throughout this document.

#### 2 Genealogy

"Give commands the ability to contain the mathematical meaning while retaining the typesetting versatility" (cool[1]). The addition of 'c', in ccool, is for *custom*. With hinsdight it is restrictive to describe ccool as a tool for encoding mathematical convention.

#### 3 Install

- 1) Compile ccool.dtx (under Unix, \$pdflatex ccool.dtx)
- 2) Put the generated ccool.sty in the search path of the LATEX engine

#### 4 Issue

Listed under:

a) NOTE or \NB, tagged either bug or fixed, inside coool.dtx

## 5 Support

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

### 6 Testing

#### 6.1 Technicality

Not possible to compile-check the expansion of a certain class of macros against predefined values[8]. Instead, one can

- a) Follow the steps in section 3 on one's own machine to generate ccool.pdf
- b) Visually check Part II, against that of the repository, for the same version.

Also see:

c) Also see: section 7 a)

#### 6.2 Platform

i) Linux laptop 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24  $_{\hookrightarrow}$  06:16:15 UTC 2018 x86\_64 x86\_64 x86\_64 GNU/Linux

#### 6.3 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)

#### 6.4 Results

- 1) ccool v1.8 compiles satisfactorily on platform i) and engine a)
- 2) ccool v1.8 compiles satisfactorily on platform i) and engine b)
- 3) ccool v1.9 compiles satisfactorily on platform i) and engines b) and c)
- 4) ccool v2.0 compiles satisfactorily on platform i) and engines b), c), and d)
- 5) ccool v2.1 compiles satisfactorily on platform i) and engines b), c), and d)
- 6) ccool v2.3 compiles satisfactorily on platform i) and engines b), c), and d)
- 7) ccool v2.7 compiles satisfactorily on platform i) and engines b), c), and d)
- 8) ccool v2.8 compiles satisfactorily on platform i) and engines b), c), and d)

#### 6.5 Other

Check [6] for testing cool with lines

#### 7 To do

- a) Regression testing using [3, Section 3.2—Specifying expectations].
- Also see:
  - b) NOTE or \NB tagged abandon|done|todo inside ccool.dtx

#### 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/13kernel/interface3.pdf
- [3] The LATEX3 Project Team *The l3build package*, 2020, http://mirror.utexas.edu/ctan/macros/latex/contrib/l3build/l3build.pdf
- [4] Thomas F. Sturm *The tcolorbox package*, 2019, http://www.texdoc.net/texmf-dist/doc/latex/tcolorbox/tcolorbox.pdf
- [5] The LATEX3 Project Team *The xparse package*, 2020, http://ftp.math.purdue.edu/mirrors/ctan.org/macros/latex/contrib/l3packages/xparse.pdf
- [6] Erwann Rogard and Olympia Hadjiliadis *Typesetting a math thesis with ccool*, 2020, https://github.com/rogard/ccool/blob/master/thesis.pdf
- [7] https://tex.stackexchange.com/users/112708/erwann?tab=questions
- [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
- [10] \Ccool, extension à LATeX à vocation mathématique, http://forum.mathematex.net/latex-f6/ccool-extension-latex-a-vocation-mathematique-t17314.
- [11] @Javier Bezos's answer to https://tex.stackexchange.com/a/547018/112708

## Change History

v1.0	$recommended[5] \dots 21$
General: Initial version 2	1 Replace: GenericObject by Name . $21$
v1.1	Replace: Separators by Separ 21
General: Add: Save 2	1  v1.2
Add: Listing 1., 2., 3., 4., 6., and 9. 2	General: Add: optional *to \OopsNew
$\operatorname{Add}: \operatorname{OopsRestore} \ldots 2$	
$Add: \OopsTest \dots 2$	Delete: \OopsTest 21
Delete: Listing 1-5 from v1.0 2	Delete: $\langle kvl_2 \rangle$ and $\langle code_2 \rangle$ 21
Fix: apparent anomaly in v1.0's	Doloto: Listing 2.3 from v1.1
Listing 4, see Listing 10 2	Replace: $OopsClear\{\langle tl_2 \rangle\}$ by
Rearrange: much of the	\OongCloar[/kawal liet\]
implementation $\dots 2$	1 · · · · · · · · · · · · · · · · · · ·
Replace:	Replace: \Restore by \Read 21
\OopsOptions by \OopsOption $2$	Replace: \Save by \Write $\dots$ 21
Replace: $\{\langle kvl_2 \rangle\}$ by $\langle \langle kvl_2 \rangle\rangle$ given	v1.3
that option type G not	General: Replace: \OopsNew by \Oops 21

Replace: $\{\langle tl_2 \rangle\}$ and $[\langle tl_2 \rangle]$ by		v2.1	
$\langle tl_2 \rangle >$	21	General: Add:Listings 3, 4, 5, 6, 7, 8,	
v1.4		and 9 2	21
General: Add: section 9	21	Replace: $\langle tl_2 \rangle$ 's position within	
Add: \OopsDebug	21	\Ccool's argument list, from first	
Add: \OopsHook	21	to second. Greater versatility 2	21
Add: Expans (for debugging' sake,		Replace: \CcoolLambda's optional	
but)	21	integer argument (number of m's)	
Add: Listing 1., 2., and 3	21	by a standard argument list 2	21
Add:optional +to \OopsNew to make		Replace: global option Name by Param 2	
side effects presist beyond local		Replace: as the de-	-1
group	21	fault of Param, Math by Default . 2	)1
Delete: Listing 1., and 2	21	v2.2	31
Replace: $s\{\{\langle tl_3\rangle\}\{\langle tl_4\rangle\}\}\{\langle tl_5\rangle\}\}$ by			٠,
$\mathtt{s}\{\{\langle tl_3\rangle\}\{\langle tl_4\rangle\}\{\langle tl_4\rangle\}\{\langle tl_4\rangle\}\}$	\$ / +1-	\11	21
D [ [ \002/ ] [ \004/ ] [ \004/ ] [ \004/ ]		respices. Part of the abstract 8 with	
v1.5	21	more straighforward descriptions	
General: Add: File	21	based on input from forum	
		participtants 2	21
Delete: dependence on datetime	21	v2.3	
v1.6		General: Add: Listing 17, Listing 18,	
General: Add: Listing showing part of	01	and Listing 19 2	21
the preamble	21	Complete: Listing 16 2	21
Rename: \OopsClear to	0.1	Rearranged: \Ccool's subsections.	
\CcoolClear	21	Previously, by argument. Now, by	
Rename: \OopsDebug to		feature	21
\CcoolDebug	21	Remove: Listing showing part of the	
Rename: \OopsHook to \CcoolHook	21	preamble 2	21
Rename: \OopsOption to			21
\CcoolOption	21	Replace: In step 2), the created	•-
Rename: \OopsRead to \CcoolRead	21	command's implementation, from	
Rename: \Oops to \Ccool	21	\ProvideDocumentCommand to	
Rename: oops to ccool (better		\DeclareDocumentCommand 2	)1
describes the purpose)	21	v2.4	'nΙ
v1.7			
General: Add: Legends to listings	21	General: Fix: minor error in the	
Add: Listing 22 (CUSUM)	21	listings (\Real rather than \Reel,	
Delete: \CcoolDebug	21	hitherto unnoticed) 2	41
Delete: Listing 5 from v1.6	21	Remove: examples from	
v1.8		Part I, \Ccool, as redundant with	
General: Add: \CcoolVers	21	Part II Listing 3-7 2	21
Add: \CcoolLambda	21	v2.5	
Add: Listing 20, Listing 21	21	General: Modify: File, rely on erw-13's	
Add: Listing 1	21	\erw_jobnametimestamp: 2	21
v1.9		Modify: behavior of Part I Expand	
General: Add: support for Lua $T_EX$	21	the $val_i$ s, rely on erw-l3's	
Move: from Part I to Part IV, what		\erw_seq_use:Nn 2	21
is now that part's section 11	21	v2.6	
v2.0		General: Modify: \CcoolLambda, rely	
General: Add: support for X\(\frac{1}{4}\)TEX \cdots	21	on erw-l3's \erw_lambda:nnn 2	21
Delete: File's dependency on	21	v2.7	
texosquery and \pdfcreationdate	21		21
Update: \RequirePackage,	41		21
\NeedsTeXFormat's second		Modify: \CcoolOption's 'm'-type	-1
argument / TeX Live 2020	21	argument by 'o'-type argument 2	)1
argument / ICA LIVE ZUZU	41	argumentov o-type argument 2	-1

Modify: Separ's default rely on	v3.0	
babel and amsmath, if applicable $\cdot$ . 21	General: Miscellaneous	21
$\sqrt{2.8}$	v3.1	
General: Fix: conflict between	General: Replaced: Listing 2's content,	
\usepackage[spanish]{babel} and	from exhaustive dependencies to	
Parameterize the $key_i$ s 21	those explicit with \usepackage	21
$\sqrt{2.9}$		
General: Miscellaneous 21		

# 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	\ccool_aux_inner_set:n 4, 196
\' 103	\ccool_aux_key:N 17, 200
$\langle \text{key}_i \langle \text{tl}_2 \rangle \rangle$	\ccool_aux_key:n 13, 20
$\langle \text{key}_i \rangle$	\ccool_aux_key:w 9, 15
$\langle \ker_i \rangle$	\g_ccool_aux_key_seq
And (option)	
Expans (option)	\g_ccool_aux_keyval_seq 197, 198, 200
File (option)	\ccool_aux_outer:n 24, 183
Inner (option)	\ccool_aux_outer_set:n <u>22</u> , <u>182</u>
Outer (option)	\gccool_aux_prop 26, 29, 46, 199
Param (option)	\ccool_aux_prop:N <u>44</u> , 198
Separ (option)	$\cdots$ _ccool_aux_prop:n $\underline{40}$ , 50
Write (option)	\ccool_aux_prop:nn <u>26</u>
\^ 108	\ccool_aux_prop:w <u>32</u> , 42
	$\c \sum_{\text{ccool}} \text{aux\_val}: \text{Nn} \ldots \underline{53}, 181$
\ 079 074 07F 001 000 009	\gccool_aux_val_seq 55, 56, 187
\_ \	\ccool_lang_and: $68, 273, 275, 281, 283$
A	\ccool_lang_and:n <u>68</u>
\AtEndDocument 125	\gccool_lang_and_prop 58, 62, 71, 73
(Nothabodamono	\cccool_lang_and_tl <u>86</u> , 347, 348
В	\ccool_lang_and_update:n $\underline{59}$ , $\underline{346}$
\begingroup 210	\ccool_log_close: <u>124</u> , 417
bool commands:	\ccool_log_entry 211, 212
\bool_gset_false:N 130	\gccool_log_file_tl . 132, 135, 354
\bool_gset_true:N 137	\gccool_log_iow
\bool_if:nTF 152, 206, 229, 415	124, 125, 129, 136, 154, 157
\bool_set_false:N 126	\ccool_log_open: <u>132</u> , 416
\BooleanFalse 8, 420, 421	\g_ccool_log_open_bool
\BooleanTrue 6, 8	
	\ccool_log_read: <u>145</u> , 428
$\mathbf{C}$	\ccool_log_read:n <u>139</u> , 147, 427
\c 115	\gccool_log_to_tl 135, 136, 147, 149
\Ccool	\ccool_log_write:n <u>149</u> , 208
2, 5, 6, 7, 7, 10, 21, 22, 211, 218, 239	\ccool_make_ccool:nnnn
ccool internal commands:	
\ccooldocpkg: 430	\ccool_make_ccool_exp:nnn
\ccool_aux_inner:n 6, 7, 38	\ccool_make_ccool_key:nnn $\frac{191}{205}$

$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	\cs_set_protected:Nn 323
203, 224, 233	
$\c \sum_{\text{ccool}_make_key:N} \dots 175, 201$	D
$\c \sum_{\text{ccool}_make_key:n} \dots 170, 177$	\DeclareDocumentCommand 7, 22, 163, 218, 337
$\cline{1}$ _ccool_make_key:Nn $\underline{160}$ , $\underline{172}$	\def 211
$\g_{\text{ccool}}\$ option_expans_tl $32, 36, 350$	
$\cdots$ _ccool_option_inner:n $249$ , $359$	${f E}$
\g_ccool_option_inner_tl	\endgroup 212
249, 252, 364, 378, 392, 406	\ensuremath 398, 399
$\cdots$ _ccool_option_outer:n $259$ , 387	erw commands:
\g_ccool_option_outer_tl	\erw_jobnametimestamp: 22
259, 262, 366, 380, 394, 408	\erw_keyval_error:Nn 63
\ccool_option_param:n <u>254</u> , <u>373</u>	\erw_lambda:nnn 22, 337
\g_ccool_option_param_tl	\erw_prop_keyval_parse:NNNn 61
164, 254, 257, 330, 363, 377, 391, 405	\erw_seq_use:Nn
\ccool_option_separ:n $\frac{264}{401}$	\erw_sys_jobnametimestamp: . 356, 357
\c_ccool_option_separ_default	exp commands:
t1 271, 279, 412, 413	\exp_args:NNf 185
\g_ccool_option_separ_tl	\exp_args:NNx 162, 218
264, 267, 269, 365, 379, 393, 407	\exp_args:No 81, 196, 301
\ccool_prop_append:NN <u>286</u> , <u>297</u>	\exp_args:Nx 35
\ccool_prop_append:Nn 199, 295	\exp_last_unbraced:Nf 360, 374, 388, 402
\ccool_prop_append:nn 288, 292	\exp_not:n 239
\_ccool_prop_clear_new:n 299, 306	\expandafter 211
\ccool_prop_clear_new_map:n	\ExplSyntaxOff
303, 332	\ExplSyntaxOn
\_ccool_prop_if_exist:nTF . 193, 308	1 1 1
$\cdot = \cdot = \cdo$	${f F}$
\ccool_prop_item:nn 166, 312 \ccool_prop_name:n	<b>F</b> file commands:
\ccool_prop_name:n	file commands:
\ccool_prop_name:n 56, 297, 301, 310, 314, <u>316, 319</u>	file commands:     \file_input:n 141
\ccool_prop_name:n	file commands:     \file_input:n 141
\ccool_prop_name:n	file commands:     \file_input:n
\ccool_prop_name:n	Tile commands:   \file_input:n
\ccool_prop_name:n	file commands:     \file_input:n
\ccool_prop_name:n	file commands:     \file_input:n
\ccool_prop_name:n	file commands:       \file_input:n       141         \function       5         G         \gappto       231         I         \lifBooleanT       225         \ifcsdef       79, 269         \lifValueT       223, 237         \lifValueTF       426         iow commands:       \iow_close:N       125, 129         \iow_new:N       124         \iow_now:Nn       154         \iow_open:Nn       136         K         keys commands:         \l_keys_choice_tl       350
\ccool_prop_name:n	file commands:       \file_input:n       141         \function       5         G         \gappto       231         I         \lifBooleanT       225         \ifcsdef       79, 269         \lifValueT       223, 237         \lifValueTF       426         iow commands:       \iow_close:N       125, 129         \iow_new:N       124         \iow_now:Nn       154         \iow_open:Nn       136         K         keys commands:         \l_keys_choice_tl       350         \keys_define:nn       344
\ccool_prop_name:n	file commands:       \file_input:n       141         \function       5         G         \gappto       231         I         \lifBooleanT       225         \ifcsdef       79, 269         \lifValueT       223, 237         \lifValueTF       426         iow commands:       \iow_close:N       125, 129         \iow_new:N       124         \iow_now:Nn       154         \iow_open:Nn       136         K         keys commands:         \l_keys_choice_tl       350         \keys_define:nn       344

M	\ProvideDocumentCommand 22, 335
msg commands: \msg_error:nnn	${f Q}$
\msg_new:nnn 243, 246	quark commands:
\msg_warning:nnn 75	\q_stop 9, 15, 33, 42
${f N}$	${f R}$
\NB	\Read 21
\NeedsTeXFormat	\Real 1, 5, 22
NewDocumentCommand	\Reel
2, 10, 329, 334, 339, 423, 430, 431	\RequirePackage         22           \Restore         21
O	(nestore
\Oops 21, 22	${f S}$
\OopsClear 21, 22	\Save 21
\OopsDebug 22	seq commands:
\OopsHook	$\scalebox{seq\_gclear\_new:N} \dots 19,55$
\OopsNew 21, 22	$\qquad \$
\OopsOption 21, 22	\seq_if_empty:NTF 47
\OopsOptions	\seq_map_function:NN
\OopsRead	20, 50, 177, 306, 327
\OopsTest	\seq_set_from_clist:Nn 197, 305
options:	Т
And	\text 273, 274, 275
Expans	tl commands:
File	\c_empty_tl 48, 194, 214, 236, 334
Inner	\tl_const:Nn 86, 271, 279
Outer	$\t1_gset:Nn \dots 135, 252, 257, 262, 354$
Param	\tl_gset_eq:NN 350
Separ	\tl_log:n 142, 155
Write 8, 38	\tl_new:N 32, 132, 149, 249, 254, 259, 264
P	\tl_trim_spaces:n 11, 37, 38
\pdfcreationdate	${f U}$
prop commands:	use commands:
\prop_clear_new:N 301	\use:N
\prop_gclear_new:N 46	\usepackage 5, 23
\prop_gput:Nnn 29, 64, 290	· ·
\prop_if_exist:NTF 310	$\mathbf{W}$
\prop_if_in:NnTF 70	\Write <u>21</u>
\prop_item:Nn 73, 290, 314, 325	V
\prop_map_function:NN 292	X
\prop_new:N 26, 58, 319	\X 5

#### Part IV

# Implementation

## 1 Opening

1 (\*package)
2 (@@=ccool)

25 }

```
3 \ExplSyntaxOn
                                   aux
\__ccool_aux_inner_set:n #1: \langle code \rangle
                              4 \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\ \verb|\__ccool_aux_inner_set:n.|)
      \__ccool_aux_key:w #1: \langle key \rangle
                             #2 : ⟨ value ⟩
                              9 \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} }
                              12 }
                            (End definition for \__ccool_aux_key:w.)
      \__ccool_aux_key:n #1: \langle key = value \rangle
                              13 \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
                              17 \cs_new_protected:Nn \__ccool_aux_key:N
                                   \verb|\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
                              22 \cs_new_protected:Nn \__ccool_aux_outer_set:n
                                   \cs_gset:Npn \__ccool_aux_outer:n ##1 {#1}
```

```
(End\ definition\ for\ \_\_ccool\_aux\_outer\_set:n.)
\__ccool_aux_prop:nn
                          26 \prop_new:N \g__ccool_aux_prop
                          27 \cs_new_protected:Nn \__ccool_aux_prop:nn
                               \prop_gput:Nnn \g__ccool_aux_prop{#1}{#2}
                          30 }
                          31 \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
                          32 \tl_new:N \g__ccool_option_expans_tl
                          33 \cs_new_protected:Npn \__ccool_aux_prop:w #1 = #2 \q_stop
                          34 {
                               \exp_args:Nx
                          35
                               \use:c{__ccool_aux_prop:\g__ccool_option_expans_tl}
                              { \tl_trim_spaces:n{#1} }
                               { \__ccool_aux_inner:n{ \tl_trim_spaces:n{#2} } }
                          39 }
                        (End\ definition\ for\ \verb|\_\_ccool\_aux\_prop:w.|)
 \__ccool_aux_prop:n #1: \langle key = value \rangle
                          40 \cs_new_protected:Nn \__ccool_aux_prop:n
                               \__ccool_aux_prop:w #1 \q_stop
                          43 }
                        (End\ definition\ for\ \verb|\__ccool_aux_prop:n.|)
 \__ccool_aux_prop:N #1: \langle keyval\ list \rangle
                          44 \cs_new_protected:Nn \__ccool_aux_prop:N
                          45 {
                               \prop_gclear_new:N \g__ccool_aux_prop
                          46
                              \seq_if_empty:NTF #1
                               { \c_empty_tl }
                          49
                                 \seq_map_function:NN #1 \__ccool_aux_prop:n
                          50
                          51
                          52 }
                        (End definition for \__ccool_aux_prop:N.)
 \__ccool_aux_val:Nn #1: \langle seq \rangle
                         #2: \langle tl \ var \ name \rangle
                          53 \cs_new_protected:Nn \__ccool_aux_val:Nn
                          54 {
                               \seq_gclear_new:N \g__ccool_aux_val_seq
                          55
                               \__ccool_seq_from_prop:NNn \g__ccool_aux_val_seq #1 { \__ccool_prop_name:n{#2} }
                          56
                        (End\ definition\ for\ \verb|\__ccool_aux_val:Nn.|)
```

#### 3 lang

```
58 \prop_new:N \g__ccool_lang_and_prop
\__ccool_lang_and_update:n
                                59 \cs_new_protected: Nn \__ccool_lang_and_update:n
                                60 {
                                     \erw_prop_keyval_parse:NNNn
                                     \g__ccool_lang_and_prop
                                62
                                    \erw_keyval_error:Nn
                                    \prop_gput:Nnn
                                     { #1 }
                                65
                                66 }
                                67 \cs_generate_variant: Nn \__ccool_lang_and_update:n { e }
                               (End definition for \__ccool_lang_and_update:n.)
        \__ccool_lang_and:n
         \__ccool_lang_and:
                                68 \cs_new:Nn \__ccool_lang_and:n
                                69 {
                                70
                                     \prop_if_in:NnTF
                                    \g__ccool_lang_and_prop
                                    {#1}
                                    \label{lem:nng_cool_lang_and_prop} $$\{\pi: Nn\g_cool_lang_and_prop\{\#1\}\}$$
                                73
                                74
                                       \msg_warning:nnn{__ccool}{lang_and}{#1}
                                75
                                       \_{\rm ccool\_lang\_and:n\{english\}}
                                76
                                77
                                78 }
                                79 \ifcsdef{languagename}
                                80 {
                                     \cs_new:Nn \__ccool_lang_and:{\exp_args:No\__ccool_lang_and:n{\languagename}}
                                82 }
                                83 {
                                     \cs_new:Nn \__ccool_lang_and:{english}
                               (End\ definition\ for\ \verb|\__ccool_lang_and:n|\ and\ \verb|\__ccool_lang_and:.|)
     \c_{\c} c_ccool_lang_and_tl (Note^{1})
                                86 \tl_const:Nn \c__ccool_lang_and_tl
                                87 {
                                88 %^A https://www.overleaf.com/learn/latex/International_language_support
                                    afrikaans=en,
                                90 basque=eta,
                                ga catalan=i,
                                92 croatian=i,
                                gg czech=a,
                                94 danish=og,
                                    dutch=en,
                                    english=and,
                                    esperanto=kaj,
                                    estonian=ja,
                                98
```

 $<sup>^1</sup>$ [todo]: Non latin-alphabet languages

```
finnish=ja,
     french=et,
100
      galician=e,
101
      german=und,
102
      hungarian=\'es,
103
      icelandic=og,
104
      indonesian=dan,
105
      irish=agus,
106
      italian=e,
     kurmanji=\^u,
108
      latin=et,
109
      latvian=un,
110
      lithuanian=ir,
     ngerman=und,
      polish=i,
      portuguese=e,
114
      romanian=\c{s}i,
115
      slovak=a,
116
117
      spanish=y,
      swedish=och,
      swissgerman=und,
119
      turkish=ve,
120
      turkmen=we,
      welsh=a
122
123 }
(End definition for \c_ccool_lang_and_tl.)
```

## $4 \log$

```
\__ccool_log_close:
                       124 \iow_new:N \g__ccool_log_iow
                        125 \AtEndDocument{\iow_close:N \g__ccool_log_iow}
                        \label{local_log_open_bool} $$ \bool_set_false: N \g_ccool_log_open_bool $$
                        127
                          \cs_new_protected: Nn \__ccool_log_close:
                        128 {
                             \iow_close:N \g__ccool_log_iow
                             \bool_gset_false:N \g__ccool_log_open_bool
                        130
                       131 }
                       (End definition for \__ccool_log_close:.)
 \__ccool_log_open:
                       132 \tilde{g}_c \approx 1
                       133 \cs_new_protected:Nn \__ccool_log_open:
                       134 {
                             \tl_gset:Nx \g_ccool_log_to_tl{\g_ccool_log_file_tl}
                        135
                             \iow_open: Nn \g__ccool_log_iow {\g__ccool_log_to_tl}
                       136
                             \bool_gset_true:N \g__ccool_log_open_bool
                       137
                       138 }
                       (End\ definition\ for\ \verb|\__ccool_log_open:.|)
```

```
\__ccool_log_read:n #1: \langle path \rangle
                       139 \cs_new_protected:Nn \__ccool_log_read:n
                       140 {
                            \file_input:n{#1}
                            \tl_log:n{read~from~#1}
                       143 }
                       144 \cs_generate_variant:Nn \__ccool_log_read:n { e }
                      (End\ definition\ for\ \_\_ccool\_log\_read:n.)
  \__ccool_log_read:
                       145 \cs_new_protected:Nn \__ccool_log_read:
                            \__ccool_log_read:e{\g__ccool_log_to_tl}
                       148 }
                      (End definition for \__ccool_log_read:.)
\__ccool_log_write:n
                       149 \tl_new:N \g__ccool_log_to_tl
                       150 \cs_new_protected:Nn \__ccool_log_write:n
                       151 {
                            \bool_if:nTF{ \g__ccool_log_open_bool }
                       152
                       153
                              \iow_now:Nn \g__ccool_log_iow {#1}
                       154
                              \tl_log:n{ write~to~#1 }
                            157
                       158 }
                       159 \cs_generate_variant:Nn \__ccool_log_write:n { e }
                      (End definition for \__ccool_log_write:n.)
                       5
                            make_key
\__ccool_make_key:Nn #1: \langle \ token \ \rangle
                       #2: \langle key \rangle
                       160 \cs_new_protected:Nn \__ccool_make_key:Nn
                       161 {
                            \exp_args:NNx
                       162
                           \DeclareDocumentCommand{#1}
                       163
                            { D<>{\g_ccool_option_param_tl} }
                              \__ccool_prop_item:nn{##1}{#2}
                       167
                            }
                       168 }
                       169 \cs_generate_variant:Nn \__ccool_make_key:Nn {c}
                      (End definition for \__ccool_make_key:Nn.)
```

```
\__ccool_make_key:n #1: \langle key \rangle
                               170 \cs_new_protected:Nn \__ccool_make_key:n
                                     \cline{1}{make_key:cn{#1}{#1}}
                               172
                               174 \cs_generate_variant:Nn \__ccool_make_key:n { e }
                               (End\ definition\ for\ \_\_ccool\_make\_key:n.)
        \__ccool_make_key:N #1: \langle seq \rangle
                               175 \cs_new_protected: Nn \__ccool_make_key: N
                                     \seq_map_function:NN #1 \__ccool_make_key:e
                               177
                               178 }
                               (End\ definition\ for\ \verb|\__ccool_make_key:N.|)
                                     make_ccool
\__ccool_make_ccool_exp:nnn
                               179 \cs_new_protected:Nn \__ccool_make_ccool_exp:nnn
                               180 {
                                     \__ccool_aux_val:Nn \g__ccool_aux_key_seq {#1}
                               181
                                     \__ccool_aux_outer_set:n{#3}
                               182
                                     \__ccool_aux_outer:n
                                185
                                       \exp_args:NNf
                                186
                                       \erw_seq_use:Nn
                                       \g__ccool_aux_val_seq
                                187
                                       {#2}
                               188
                                    }
                               189
                               190 }
                               (End\ definition\ for\ \verb|\__ccool_make_ccool_exp:nnn.|)
\__ccool_make_ccool_key:nnn
                               191 \cs_new_protected:Nn \__ccool_make_ccool_key:nnn
                               192 {
                                    \__ccool_prop_if_exist:nTF{#1}
                               193
                                    { \c_empty_tl }
                               194
                                    { \__ccool_prop_new:n{#1} }
                                    \exp_args:No \__ccool_aux_inner_set:n{#2}
                                    \seq_set_from_clist:Nn \g__ccool_aux_keyval_seq {#3}
                                    \__ccool_aux_prop:N \g__ccool_aux_keyval_seq
                                198
                                    199
                                    \__ccool_aux_key:N \g__ccool_aux_keyval_seq
                               200
                                     \verb|\cool_make_key:N \end{def} $$ \g_ccool_aux_key_seq $$
                               201
                               202 }
                               (End\ definition\ for\ \_\_ccool\_make\_ccool\_key:nnn.)
```

```
\__ccool_make_ccool_sideeffect:nnn [9]
                               {\tt 203 \ \backslash cs\_new\_protected:Nn \ \backslash\_ccool\_make\_ccool\_sideeffect:nnn}
                               204 €
                                     \cline{1}{make_ccool_key:nnn{#1}{#2}{#3}}
                               205
                                     \bool_if:nTF{ \g__ccool_log_open_bool }
                               206
                               207
                                        \__ccool_log_write:n
                                208
                                209
                                          \begingroup
                                          \label{log_entry} $$ \left( \ccool<\#1>c(\#2)\{\#3\} \right) \simeq \mathbb{C}
                                          \endgroup \__ccool_log_entry
                                     }{\c_empty_tl}
                               214
                               215 }
                               (End\ definition\ for\ \_\_ccool\_make\_ccool\_sideeffect:nnn.)
\__ccool_make_ccool:nnnn
                               #1: \langle token \ list \rangle
                               #2: \langle seq_1 \rangle
                               #3: \langle seq_2 \rangle
                               #4: \langle prop \rangle
                               216 \cs_new_protected:Npn \__ccool_make_ccool:nnnn #1 #2 #3 #4
                                     \exp_args:NNx \DeclareDocumentCommand \Ccool
                               218
                                                                  4 5 6
                                                              3
                                                  2
                                219
                                       +o D<>{#1} E{ c }{{#2}} m t+ s E{ s c }{{#3}{#4}} +o
                                220
                                     }
                                     {
                                        \IfValueT{##1}{##1}
                                        \__ccool_make_ccool_sideeffect:nnn{##2}{##3}{##4}
                               224
                                        \IfBooleanT{##6}
                               225
                                          \__ccool_make_ccool_exp:nnn{##2}{##7}{##8}
                                        }
                               228
                                        \bool_if:nTF{##5}
                               229
                               230
                                        {
                                          \gappto{\CcoolHook}
                               231
                               232
                                             \__ccool_make_ccool_sideeffect:nnn{##2}{##3}{##4}
                               234
                               235
                                        {\c_empty_tl}
                                236
                                237
                                        \IfValueT{##9}
                                        {
                                          \exp_not:n{ \Ccool[##9] }
                               239
                                       }
                               240
                                     }
                               241
                               242 }
                               (End\ definition\ for\ \_\_ccool\_make\_ccool:nnnn.)
```

#### $7 \, \text{msg}$

```
243 \msg_new:nnn {__ccool}
                              244 { iow }
                              245 {#1~is~closed~can't~write}
                              246 \msg_new:nnn {__ccool}
                              247 {lang_and}
                              248 {~key~#1~missing~for~global~option~'And';~falling~back~on~'english'}
                              8
                                    option
  \__ccool_option_inner:n #1: \langle code \rangle
                              249 \tl_new:N \g__ccool_option_inner_tl
                              250 \cs_new_protected:Nn \__ccool_option_inner:n
                                    \tl_gset:Nn \g__ccool_option_inner_tl {#1}
                              252
                              253 }
                             (End\ definition\ for\ \verb|\__ccool_option_inner:n.|)
  \__ccool_option_param:n #1: \langle token \ list \rangle
                              254 \tl_new:N \g__ccool_option_param_tl
                              255 \cs_new_protected:Nn \__ccool_option_param:n
                                    \tl_gset:Nn \g__ccool_option_param_tl{#1}
                              257
                              258 }
                             (End\ definition\ for\ \verb|\__ccool_option_param:n.|)
  \__ccool_option_outer:n #1: \langle inline \ code \rangle
                              ^{259} \tl_new:N \g_ccool_option_outer_tl
                              260 \cs_new_protected:Nn \__ccool_option_outer:n
                                    \tl_gset:Nn \g__ccool_option_outer_tl {#1}
                             (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\}
                              264 \tl_new:N \g__ccool_option_separ_tl
                              266 €
                                   \cs_gset:Npn \g_ccool_option_separ_tl {#1}
                              267
                              268 }
                             (End definition for \__ccool_option_separ:n.)
\g__ccool_option_separ_tl
                              269 \ifcsdef{text}
                                   \tl_const:Nn \c_ccool_option_separ_default_tl
                              272
                                      { \text{\text}({\ })_{\cool\_lang\_and:{\ }} }
                              273
                                      { \text{,{\ }} }
                              274
                                      { \text{\text{,}\ }\ }\ 
                              275
                              276
```

```
277 }
                                278 {
                                      \tl_const:Nn \c_ccool_option_separ_default_tl
                                279
                                280
                                        { \{ \ \} \subseteq ccool\_lang\_and: \{ \ \} }
                                281
                                        { ,{\ } }
                                282
                                        { ,\{\ \}\setminus_{\ \ \ }}
                                283
                                284
                                285 }
                               (End\ definition\ for\ \verb|\g_ccool_option_separ_tl|.)
                                9
                                      prop
  \__ccool_prop_append:NN #1: \langle \; prop_1 \; 
angle
                                #2: \langle prop_2 \rangle
                                286 \cs_new_protected:Npn \__ccool_prop_append:NN #1 #2
                                      \cs_set:Nn \__ccool_prop_append:nn
                                289
                                        \prop_gput:Nnx #1 {##1}{ \prop_item:Nn #2{##1} }
                                290
                                291
                                      \prop_map_function:NN #2 \__ccool_prop_append:nn
                                292
                                293 }
                                294 \cs_generate_variant:Nn \__ccool_prop_append:NN { cN }
                               (End\ definition\ for\ \verb|\_\_ccool\_prop\_append:NN.)
  \__ccool_prop_append:Nn #1: \langle prop \rangle
                                #2: \langle tl \ var \ name \rangle
                                295 \cs_new_protected:Nn \__ccool_prop_append:Nn
                                      \__ccool_prop_append:cN{ \__ccool_prop_name:n {#2} } #1
                                298 }
                               (End\ definition\ for\ \verb|\__ccool_prop_append:Nn.|)
\__ccool_prop_clear_new:n #1: \langle tl \ var \ name \rangle
                                299 \cs_new_protected:Nn \__ccool_prop_clear_new:n
                                      \exp_args:No \prop_clear_new:c{ \__ccool_prop_name:n {#1} }
                                301
                               (End\ definition\ for\ \verb|\__ccool_prop_clear_new:n.|)
      \_ccool_prop_clear_new_map:n #1: \langle keyval list \rangle
                                303 \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
                                306
                                307 }
                               (End definition for \__ccool_prop_clear_new_map:n.)
```

```
\__ccool_prop_if_exist:nTF #1: \langle tl_1 
angle
                              #2: \langle tl_2 \rangle
                              #3: \langle tl_3 \rangle
                               308 \cs_new:Nn \__ccool_prop_if_exist:nTF
                                    \prop_if_exist:cTF{ \__ccool_prop_name:n {#1} }{#2}{#3}
                               310
                               311 }
                              (End definition for \__ccool_prop_if_exist:nTF.)
     \__ccool_prop_item:nn #1: \langle tl \ var \ name \rangle
                              #2: \langle key \rangle
                               312 \cs_new:Nn \__ccool_prop_item:nn
                                    \prop_item:cn { \__ccool_prop_name:n {#1} } {#2}
                               314
                              (End\ definition\ for\ \verb|\__ccool_prop_item:nn.|)
      \__ccool_prop_name:n #1: \langle tl var name \rangle
                               316 \cs_new:Npn \__ccool_prop_name:n #1{ __ccool_#1 }
                              (End definition for \__ccool_prop_name:n.)
       \__ccool_prop_new:n #1: \langle tl var name \rangle
                               317 \cs_new_protected:Nn \__ccool_prop_new:n
                                    \prop_new:c{ \__ccool_prop_name:n {#1} }
                               320 }
                              (End definition for \__ccool_prop_new:n.)
                               10
                                      sea
\__ccool_seq_from_prop:NNn #1: \langle seq_1 \rangle
                              #2: \langle seq_2 \rangle (keys)
                              #3 : 〈 prop 〉
                               \cs_set_protected: Nn \__ccool_seq_from_prop:n
                               323
                               324
                                      325
                               326
                               327
                                    \seq_map_function:NN #2 \__ccool_seq_from_prop:n
                               328 }
                              (End\ definition\ for\ \verb|\__ccool_seq_from_prop:NNn.|)
```

#### 11 Front-end

```
\CcoolClear
               329 \NewDocumentCommand{ \CcoolClear }
               330 { D<>{\g_ccool_option_param_tl} }
                     \_{\tt ccool\_prop\_clear\_new\_map:n{#1}}
               333 }
              (End definition for \CcoolClear. This function is documented on page 6.)
  \CcoolHook
               334 \NewDocumentCommand{\CcoolHook}{}{\c_empty_tl}
              (End definition for \CcoolHook. This function is documented on page 7.)
\CcoolLambda (Note^2)
               335 \ProvideDocumentCommand \CcoolLambda { O{m} m }
                    \erw_lambda:nnn \DeclareDocumentCommand { #1 } { #2 }
               338 }
              (End definition for \CcoolLambda. This function is documented on page 7.)
\CcoolOption (Note^3) (Note^4)
               339 \NewDocumentCommand{ \CcoolOption }
               340 { O{ And, Expans, File, Inner, Param, Outer, Separ, Write } }
               341 {
                    \keys_set:nn{ __ccool }{#1}
               343 }
              (End definition for \CoolOption. This function is documented on page 7.)
               344 \keys_define:nn { __ccool }
               345 {
         And
               346 And .code:n = { \_ccool_lang_and_update:e{ #1 } },
               And .default:n = { \c_ccool_lang_and_tl },
               348 And .initial:n = { \c__ccool_lang_and_tl },
     Expans
               349 Expans .multichoices:nn = { eo, ee, ex, xo, xe, xx }
               350 { \tl_gset_eq:NN \g__ccool_option_expans_tl \l_keys_choice_tl },
               351 Expans .default:n = { xo },
               352 Expans .initial:n = { xo },
```

```
File
        353 File .code:n = {
        354 \tl_gset:Nx \g__ccool_log_file_tl{#1}
        355 },
        356 File .default:n = { \erw_sys_jobnametimestamp: },
        357 File .initial:n = { \erw_sys_jobnametimestamp: },
Inner
        358 Inner .code:n={
             \__ccool_option_inner:n{#1}
             \exp_last_unbraced:Nf
        360
             \__ccool_make_ccool:nnnn
        361
        362
               { \g_ccool_option_param_tl }
        363
               { \g_ccool_option_inner_tl }
               { \g_ccool_option_separ_tl }
               { \g_ccool_option_outer_tl }
             }
        367
        368 },
        369 Inner .value_required:n = false,
        370 Inner .default:n = {####1},
        371 Inner .initial:n = {####1},
Param
        372 Param .code:n={
             \__ccool_option_param:n{#1}
             \exp_last_unbraced:Nf
             \__ccool_make_ccool:nnnn
        375
        376
               { \g_ccool_option_param_tl }
        377
               { \g_ccool_option_inner_tl }
               { \g_ccool_option_separ_tl }
               { \g_ccool_option_outer_tl }
             }
        381
        382 },
        383 Param .value_required:n = false,
        384 Param .default:n = { Default },
        385 Param .initial:n = { Default },
Outer
        386 Outer .code:n={
             \__ccool_option_outer:n{#1}
             \exp_last_unbraced:Nf
        388
             \__ccool_make_ccool:nnnn
        389
             {
        390
               { \g_ccool_option_param_tl }
        391
               { \g_ccool_option_inner_tl }
        392
               { \g_ccool_option_separ_tl }
               { \g_ccool_option_outer_tl }
             }
        395
        396 },
          ^2[todo]: allow only m- or o-type arguments
          <sup>3</sup>[todo]: Fix placeholders passed to options requiring code (only one pound sign)
          ^4[abandon]: Requirement: write to file if Write; Update: redundant with \c
       {Ccool}+Write
```

```
397 Outer .value_required:n = false,
            398 Outer .default:n = { \ensuremath{####1} },
            399 Outer .initial:n = { \ensuremath{####1} },
    Separ
            400 Separ .code:n={
                 \__ccool_option_separ:n{#1}
            401
                 \exp_last_unbraced:Nf
                 \__ccool_make_ccool:nnnn
                   { \g_ccool_option_param_tl }
                   { \g_ccool_option_inner_tl }
                   407
                   { \g_ccool_option_outer_tl }
            408
            409
            410 },
            411 Separ .value_required:n = false,
            412 Separ .default:n = { \c__ccool_option_separ_default_tl },
            413 Separ .initial:n = { \c__ccool_option_separ_default_tl },
    Write
            414 Write .code:n = \{
                 \bool_if:nTF{#1}
                 {\__ccool_log_open:}
                 {\__ccool_log_close:}
            417
            418 },
            419 Write .value_required:n = false,
            420 Write .default:n = \BooleanFalse,
            421 Write .initial:n = \BooleanFalse
            422 }
\CcoolRead
            423 \NewDocumentCommand{\CcoolRead}
            424 {o}
            425 {
                 \IfValueTF{#1}
            426
                 {\__ccool_log_read:e{#1}}
            427
                 {\__ccool_log_read:}
           (End definition for \CcoolRead. This function is documented on page 8.)
\CcoolVers
            430 \NewDocumentCommand{\CcoolVers}
            432 {\use:c{ver@ccool.sty}}
            (End definition for \CcoolVers. This function is documented on page 8.)
            12
                   Closing
            433 \ExplSyntaxOff
            434 (/package)
```