# The ccool package*

Erwann Rogard†

Released 2021-09-20

**Abstract**

This LaTeXpackage provides an interface to define and evaluate key-based replacement rules[3]. It can be used to parse the argument specification of a document command[4].

## Contents



*This file describes version v3.2, last revised 2021-09-20.

†first.lastname at gmail.com

# Part I
# Usage

\Ccool[usage:cs:ccool]

\Ccool `\Ccool[⟨tl1⟩]<⟨tl2⟩>c{⟨code1⟩}{⟨keyval list1⟩}+*s{⟨separators⟩}c{⟨code2⟩}[⟨tl6⟩]`

where ⟨*separators*⟩ is either of: `{⟨tl3⟩}`, `{⟨tl3⟩}{⟨tl4⟩}`, and `{⟨tl3⟩}{⟨tl4⟩}{⟨tl5⟩}`.

**Semantics** See subsection 0.1-0.8.

## 0.1 Core feature

`\Ccool{⟨keyval list1⟩}` executes for each ⟨$key_i$⟩=⟨$val_i$⟩,

*1)* ⟨$val_i$⟩ ← `\function{⟨vali⟩}`

*2)* define `\⟨keyi⟩` such that `\⟨keyi⟩` → ⟨$val_i$⟩,

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

if `\⟨keyi⟩` needs arguments, use `\lambdax` from lambdax on the rhs.

## 0.2 Process the $val_i$'s

`\Ccool c{⟨code1⟩}{⟨keyval list1⟩}` is identical to the Core feature, except it overrides `Inner`.

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

## 0.3 Append to a hook

`\Ccool{⟨keyval list1⟩}+` is identical to the Core feature, except it repeats after `\CcoolHook`. This is useful to make the side effect persist after a *local group* (such as `theorem`).

## 0.4 Expand the $val_i$'s

`\Ccool{⟨keyval list1⟩}*` supplements the Core feature with the expansion of the ⟨$val_i$⟩'s using typesetting rules encoded in *global option* `Separ`and `Outer`. The first are *separators* applied to the ⟨$val_i$⟩'s to form a *token list*, and the second a function applied to the latter.

They can be overriden inline by appending further `s{⟨separators⟩}` and `c{⟨code2⟩}`, respectively, to the list of arguments.

## 0.5 Head

`\Ccool[`⟨*tl*$_1$⟩`]{`⟨*keyval list*$_1$⟩`}` expands ⟨*tl*$_1$⟩ and executes the Core feature.

There may be situations where it is convenient to pass ⟨*tl*$_1$⟩ as empty.

## 0.6 Tail

`\Ccool{`⟨*keyval list*$_1$⟩`}[`⟨*tl*$_6$⟩`]{`⟨*keyval list*$_2$⟩`}` is identical to `\Ccool{`⟨*keyval list*$_1$⟩`}` followed by `\Ccool[`⟨*tl*$_6$⟩`]{`⟨*keyval list*$_2$⟩`}`.

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

## 0.7 Parameterize the *key*$_i$'s

`\Ccool<`⟨*tl*$_2$⟩`>{`⟨*keyval list*$_1$⟩`}` is identical to the Core feature, except ⟨*key*$_i$⟩ is replaced by ⟨*key*$_i$`<`*tl*$_2$`>`⟩. The default value of ⟨*tl*$_2$⟩ is encoded in `Param`. In our example, ⟨*tl*$_2$⟩ could be `Style`.

## 0.8 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`usage:cs:clear

`\CcoolClear` `\CcoolClear<`⟨*tl*$_2$⟩`>{...`⟨*key*$_i$⟩`,...}`

**Semantics** Clears all `\`⟨*key*$_i$`<`*tl*$_2$`>`⟩'s

`\CcoolHook`usage:cs:hook

`\CcoolHook` `\CcoolHook`

**Semantics** No side effect or expansion

`\CcoolOption`usage:cs:option

`\CcoolOption` `\CcoolOption[...`⟨*key*$_i$⟩`|`⟨*key*$_i$⟩`=`⟨*val*$_i$⟩`,...]`

where ⟨*key*$_i$⟩ is either of `And`, `Expans`, `File`, `Inner`, `Param`, `Outer`, `Separ`, and `Write`.

**Semantics** Modify the default behavior of `\Ccool`

`And` `And`usage:opt:an

**Semantics** Sets the translation of *and* in language ⟨*key*⟩ to ⟨*val*⟩

**Syntax** ⟨*keyval list*⟩

`Expans` `Expans`usage:opt:ex

**Syntax** `eo|ee|ex|xo|xe|xx`

`File` `File`usage:opt:fi

**Syntax** ⟨*path*⟩

`Inner` `Inner`usage:opt:in

**Syntax** ⟨*code*⟩, with `####1` as the *placeholder*

`Param` `Param`usage:opt:pa

**Syntax** ⟨*token list*⟩

`Outer` `Outer`usage:opt:ou

**Default** `\ensuremath{####1}`

**Syntax** ⟨*code*⟩, with `####1` as the *placeholder*

`Separ` `Separ`usage:opt:se

**Other** Default behavior depends on whether babel and amsmath are loaded

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

`Write` `Write`usage:opt:wr

**Syntax** `\BooleanFalse`|`\BooleanTrue`

`\CcoolRead`usage:cs:read

---

`\CcoolRead` `\CcoolRead[`⟨*path*⟩`]`

---

**Semantics**

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

`\CcoolVers`usage:cs:vers

---

`\CcoolVers` `\CcoolVers`

---

**Semantics** → the package's version

# Part II
# Other

## 1 Bibliograhy


[1] Olympia Hadjiliadis. "Change–point detection of two–sided alternatives in the Brownian motion model and its connection to the gambler's ruin problem with relative wealth perception". PhD thesis. Columbia University, 2005.

[2] Thomas F. Sturm. *The tcolorbox package*. http://www.texdoc.net/texmf-dist/doc/latex/tcolorbox/tcolorbox.pdf. 2019.

[3] The LaTeX3 Project Team. *The LaTeX3 interfaces*. https://ctan.math.washington.edu/tex-archive/macros/latex/contrib/l3kernel/expl3.pdf. 2019.

[4] The LaTeX3 Project Team. *The xparse package*. https://ctan.math.illinois.edu/macros/latex/contrib/l3packages/xparse.pdf. 2019.

[5] @frougon. *"Journaling calls to a function []"*. https://tex.stackexchange.com/a/536620. 2020.

[6] @Javier Bezos. *When loading babel with spanish, spurious document command parser*. https://tex.stackexchange.com/a/547018/112708. 2020.

## 2 Do's and dont's

1.

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

Symptom: `\CcoolRead` fails

Do: `Inner={\char‘{####1\char‘}}`

2.

Don't: `$⟨key_i⟩<x$`.

Do: `$\⟨key_i⟩{<}x$`

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: `\usepackage[spanish]{babel}`

Do: `\usepackage[spanish.noquoting]{babel}`[6]

## 3 To do

1. Create an environment for `\CoolHook`.

## 4 Support

This package is available from https://github.com/rogard/ccool.

# Part III
# Listing

**Listing 1. "Let $\mathbb{N}$ and $\mathbb{R}$ denote..."**

```
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 2. Same as 1, 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 3. Same as 2, with `\Ccool`**

```
\begingroup
\Ccool c{\mathbb{#1}}{ Nat = {N}, Real = {R} }
Let~$\Nat$ and $\Real$~denote the natural and real numbers.
\endgroup
```

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

**Listing 4. Same as 3, with expansion**

```
\begingroup
  \Ccool[Let~]
  c{\mathbb{#1}}{ Nat = {N}, Real = {R} }*
  [~denote the natural and real numbers.]{}
\endgroup
```

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

**Listing 5. Same as 3, parameterized**

```
\begingroup
  \Ccool<foo>c{\mathbb{#1}}{ Nat = {N}, Real = {R} }
  [Let $\Nat<foo>$ and $\Real<foo>$ denote the natural and real numbers.]{}
\endgroup
```

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

**Listing 6. Mittelwertsatz für $n$ Variable[2, p. 17.3]**

```
\begingroup
\CcoolOption[ Write = \BooleanTrue ]
\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] } }+
  [$n\in\N$,~$\OffMenge\subseteq\N^n$ eine offene Menge und
  $f\in\Ci(\OffMenge,\R)$.
  Dann gibt es auf jeder Strecke $\Strecke\subset\OffMenge$ einen Punkt
  $\xi\in\Strecke$,~]
  { Steig = { \frac{ f(x)-f(x_0) }{ x-x_0 } }, Punkt = { \xi } }+
  [so dass gilt
  \begin{equation*}
    \Steig = \Grad f(\Punkt)^{\top}
  \end{equation*}
\end{theorem}]
{}
(Check: $\N$, $\Punkt$)
  \endgroup
\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 7. Listing 6 read from file**

```
\CcoolRead ~$\N$ $\R$ $\OffMenge$ $\Ci$ $\Strecke$
\CcoolClear
```

---

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

**Listing 8. Probability space**

```
\begingroup
\Ccool[Let~]
{ Space = \Omega, Field = \mathcal{F}, Meas = \mathcal{P} }
*s{{,}}c{$\{#1\}$}
```

```
[~denote the probability space, where~]{ PowerSet = { 2^{\Space} } }
[$\Field\subset \PowerSet$.]
{}
\endgroup
```

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

**Listing 9. Families of polynomial functions**

```
\CcoolOption[ Write = \BooleanTrue ]
\Ccool c{\mathbb{#1}}{ Nat = {N}, Real = {R} }
[Let~]
{ PolyR = \LambdaX[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. ]
{}
\CcoolClear
\CcoolOption
```

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 10. 9 read from file**

```
\begingroup
\CcoolRead $\PolyR[n]$ et $\PolyR$
\endgroup
```

$\mathbb{R}_n[X]$ et $\mathbb{R}[X]$

**Listing 11. Fonction et fonctionelle**

```
\CcoolOption[ Write = \BooleanTrue ]
\selectlanguage{french}
\Ccool<eval>{ fun = \LambdaX{(#1)} }[]<op>{ fun = \LambdaX[mm]{#1[#2]} }
[Supposons une fonction $f\fun<eval>{t}$, et \'etudions le probl\`eme
  o\`u la fonctionnelle $\fun<op>{S}{f}$ est donn\'ee par\dots]{}
\CcoolClear
\CcoolOption
```

Supposons une fonction $f(t)$, et étudions le problème où la fonctionnelle $S[f]$ est donnée par...

**Listing 12. 11 read from file**

```
\CcoolRead $\fun<eval>{t}$, $\fun<op>{S}{f}$
\CcoolClear
```

$(t)$, $S[f]$

**Listing 13. CUSUM statistic[1]**

```
\begingroup
\CcoolOption[ Write = \BooleanTrue ]
\newtheorem{definition}{Definition}
\AfterEndEnvironment{definition}{\CcoolHook}
\Ccool{
  SuchThat = { ;~ },
  Time = { t },
  Process = { \xi },
  StopT = { T },
  EvalAt = \LambdaX{(#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 = { \nu } }+*c{$#1\in\Real^{+}$.}
  [~Define the following processes:]
  {
    LogWald = { u },
    CUSUMst = { \StopT_{c} },
    CUSUM = { y },
    LogWaldInf = { m }
  }+
  [\begin{enumerate}
  \item{
      $\LogWald_{\Time}\EvalAt{ \Scale } = \Scale\Process_{\Time} -
  \frac{1}{2}\Scale^2\Time$;
      $\LogWaldInf_{\Time}\EvalAt{ \Scale } = \inf_{ 0\le s \le \Time
  }\CUSUM_{s} \EvalAt{ \Scale }$.
    }
  \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: $\Scale$, $\CUSUM$)
  \endgroup
  \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. $T_c(\lambda, m) = \inf[t \ge 0;\, y_t(\lambda) \ge m]$, *which is the CUSUM stopping time.*

(Check: $\lambda$, $y$)

**Listing 14. Listing 13 read from file**

```
  \begingroup
  \CcoolRead $\Time$ $\Process$ $\Scale$ $\Real$ $\CUSUMthresh$ $\LogWald$
     $\CUSUMst$ $\CUSUM$ $\LogWaldInf$
  \endgroup
```

$t\ \xi\ \lambda\ \mathcal{R}\ \nu\ u\ T_c\ y\ m$

# Part IV
# Implementation

## 1 Opening

```
⟨*package⟩
⟨@@=ccool⟩
\ExplSyntaxOn
```

## 2 aux

`\__ccool_aux_inner_set:n` `#1` : ⟨*code*⟩

```
\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` : ⟨ *key* ⟩

#2 : ⟨ *value* ⟩

```
\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} }
}
```

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

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

```
\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 : ⟨ *seq* ⟩

```
\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 : ⟨ *inline code* ⟩

```
\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`

```
\prop_new:N \g__ccool_aux_prop
\cs_new_protected:Nn \__ccool_aux_prop:nn
{
  \prop_gput:Nnn \g__ccool_aux_prop{#1}{#2}
}
\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 : ⟨ *key* ⟩
#2 : ⟨ *value* ⟩

```
\tl_new:N \g__ccool_option_expans_tl
\cs_new_protected:Npn \__ccool_aux_prop:w #1 = #2 \q_stop
{
  \exp_args:Nx
  \use:c{__ccool_aux_prop:\g__ccool_option_expans_tl}
  { \tl_trim_spaces:n{#1} }
  { \__ccool_aux_inner:n{ \tl_trim_spaces:n{#2} } }
}
```

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

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

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

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

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

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

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

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

```
\cs_new_protected:Nn \__ccool_aux_val:Nn
{
  \seq_gclear_new:N \g__ccool_aux_val_seq
  \__ccool_seq_from_prop:NNn \g__ccool_aux_val_seq #1 { \__ccool_prop_name:n{#2} }
}
```

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

```
\cs_new:Nn\__ccool_aux_merge:nn{#1#2}
```

# 3 lang

```
\prop_new:N \g__ccool_lang_and_prop
```

`\__ccool_lang_and_update:n`

```
%      \changes{v3.2}
%      {2021/09/20}
%      {Replace~\cs[no-index]{erw_prop_keyval:Nn}~by~\cs[no-index]{prop_set_from_keyval:Nn}}
\cs_new_protected:Nn \__ccool_lang_and_update:n
{
  \prop_set_from_keyval:Nn
  \g__ccool_lang_and_prop
  { #1 }
}
\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:

```
\cs_new:Nn \__ccool_lang_and:n
{
  \prop_if_in:NnTF
  \g__ccool_lang_and_prop
  {#1}
  {\prop_item:Nn\g__ccool_lang_and_prop{#1}}
  {
    \msg_warning:nnn{__ccool}{lang_and}{#1}
    \__ccool_lang_and:n{english}
  }
}
\ifcsdef{languagename}
{
  \cs_new:Nn \__ccool_lang_and:{\exp_args:No\__ccool_lang_and:n{\languagename}}
}
{
  \cs_new:Nn \__ccool_lang_and:{english}
}
```

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

\c__ccool_lang_and_tl (*Note*[1])

```
\tl_const:Nn \c__ccool_lang_and_tl
{
%^^A https://www.overleaf.com/learn/latex/International_language_support
  afrikaans=en,
  basque=eta,
  catalan=i,
  croatian=i,
  czech=a,
  danish=og,
  dutch=en,
  english=and,
  esperanto=kaj,
  estonian=ja,
  finnish=ja,
  french=et,
  galician=e,
  german=und,
  hungarian=\'es,
  icelandic=og,
  indonesian=dan,
  irish=agus,
  italian=e,
  kurmanji=\^u,
  latin=et,
  latvian=un,
  lithuanian=ir,
  ngerman=und,
  polish=i,
  portuguese=e,
```

[1] [todo]: Non latin-alphabet languages

```
  romanian=\c{s}i,
  slovak=a,
  spanish=y,
  swedish=och,
  swissgerman=und,
  turkish=ve,
  turkmen=we,
  welsh=a
}
```

(*End definition for* `\c__ccool_lang_and_tl`.)

# 4 log

`\__ccool_log_close:`

```
\iow_new:N \g__ccool_log_iow
\AtEndDocument{\iow_close:N \g__ccool_log_iow}
\bool_set_false:N \g__ccool_log_open_bool
\cs_new_protected:Nn \__ccool_log_close:
{
  \iow_close:N \g__ccool_log_iow
  \bool_gset_false:N \g__ccool_log_open_bool
}
```

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

`\__ccool_log_open:`

```
\tl_new:N \g__ccool_log_file_tl
\cs_new_protected:Nn \__ccool_log_open:
{
  \tl_gset:Nx \g__ccool_log_to_tl{\g__ccool_log_file_tl}
  \iow_open:Nn \g__ccool_log_iow {\g__ccool_log_to_tl}
  \bool_gset_true:N \g__ccool_log_open_bool
}
```

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

`\__ccool_log_read:n` `#1` : ⟨*path*⟩

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

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

`\__ccool_log_read:`

```
\cs_new_protected:Nn \__ccool_log_read:
{
  \__ccool_log_read:e{\g__ccool_log_to_tl}
}
```

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

__ccool_log_write:n

```
\tl_new:N \g__ccool_log_to_tl
\cs_new_protected:Nn \__ccool_log_write:n
{
  \bool_if:nTF{ \g__ccool_log_open_bool }
  {
    \iow_now:Nn \g__ccool_log_iow {#1}
    \tl_log:n{ write~to~#1 }
  }
  { \msg_error:nnn{ __ccool }{ iow }{ \g__ccool_log_iow }  }
}
\cs_generate_variant:Nn \__ccool_log_write:n { e }
```

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

# 5 make_key

__ccool_make_key:Nn

#1 : ⟨ *token* ⟩
#2 : ⟨ *key* ⟩

```
\cs_new_protected:Nn \__ccool_make_key:Nn
{
  \exp_args:NNx
  \DeclareDocumentCommand{#1}
  { D<>{\g__ccool_option_param_tl} }
  {
    \__ccool_prop_item:nn{##1}{#2}
  }
}
\cs_generate_variant:Nn \__ccool_make_key:Nn {c}
```

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

__ccool_make_key:n

#1 : ⟨ *key* ⟩

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

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

__ccool_make_key:N

#1 : ⟨ *seq* ⟩

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

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

# 6 make_ccool

__ccool_make_ccool_exp:nnn

```
%      ^^A    \erw_seq_use:Nn
\cs_new_protected:Nn \__ccool_make_ccool_exp:nnn
{
  \__ccool_aux_val:Nn \g__ccool_aux_key_seq {#1}
  \__ccool_aux_outer_set:n{#3}
  \__ccool_aux_outer:n
  {
    \exp_args:NNf
    \__ccool_seq_use:Nn
    \g__ccool_aux_val_seq
    {#2}
  }
}
```

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

__ccool_make_ccool_key:nnn

```
\cs_new_protected:Nn \__ccool_make_ccool_key:nnn
{
  \__ccool_prop_if_exist:nTF{#1}
  { \c_empty_tl }
  { \__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
  \__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
}
```

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

__ccool_make_ccool_sideeffect:nnn [5]

```
\cs_new_protected:Nn \__ccool_make_ccool_sideeffect:nnn
{
  \__ccool_make_ccool_key:nnn{#1}{#2}{#3}
  \bool_if:nTF{ \g__ccool_log_open_bool }
  {
    \__ccool_log_write:n
    {
      \begingroup
      \def \__ccool_log_entry { \Ccool<#1>c{#2}{#3} } \expandafter
      \endgroup \__ccool_log_entry
    }
  }{\c_empty_tl}
}
```

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

__ccool_make_ccool:nnnn

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

#4 : ⟨ *prop* ⟩

```
\cs_new_protected:Npn \__ccool_make_ccool:nnnn #1 #2 #3 #4
{
  \exp_args:NNx \DeclareDocumentCommand \Ccool
  {%^^A      2         3    4 5  6           7   8    9
    +o D<>{#1} E{ c }{{#2}} m t+ s E{ s c }{{#3}{#4}} +o
  }
  {
    \IfValueT{##1}{##1}
    \__ccool_make_ccool_sideeffect:nnn{##2}{##3}{##4}
    \IfBooleanT{##6}
    {
      \__ccool_make_ccool_exp:nnn{##2}{##7}{##8}
    }
    \bool_if:nTF{##5}
    {
      \gappto{\CcoolHook}
      {
        \__ccool_make_ccool_sideeffect:nnn{##2}{##3}{##4}
      }
    }
    {\c_empty_tl}
    \IfValueT{##9}
    {
      \exp_not:n{ \Ccool[##9] }
    }
  }
}
```

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

# 7 msg

```
\msg_new:nnn {__ccool}
{ iow }
{#1~is~closed~can't~write}
\msg_new:nnn {__ccool}
{lang_and}
{~key~#1~missing~for~global~option~'And';~falling~back~on~'english'}
```

# 8 option

`\__ccool_option_inner:n` #1 : ⟨*code*⟩

```
\tl_new:N \g__ccool_option_inner_tl
\cs_new_protected:Nn \__ccool_option_inner:n
{
  \tl_gset:Nn \g__ccool_option_inner_tl {#1}
}
```

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

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

```
\tl_new:N \g__ccool_option_param_tl
```

```
\cs_new_protected:Nn \__ccool_option_param:n
{
  \tl_gset:Nn \g__ccool_option_param_tl{#1}
}
```

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

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

```
\tl_new:N \g__ccool_option_outer_tl
\cs_new_protected:Nn \__ccool_option_outer:n
{
  \tl_gset:Nn \g__ccool_option_outer_tl {#1}
}
```

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

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

```
\tl_new:N \g__ccool_option_separ_tl
\cs_new_protected:Nn \__ccool_option_separ:n
{
  \cs_gset:Npn \g__ccool_option_separ_tl {#1}
}
```

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

`\g__ccool_option_separ_tl`

```
\ifcsdef{text}
{
  \tl_const:Nn \c__ccool_option_separ_default_tl
  {
    { \text{{\ }\__ccool_lang_and:{\ }} }
    { \text{,{\ }} }
    { \text{,{\ }\__ccool_lang_and:{\ }} }
  }
}
{
  \tl_const:Nn \c__ccool_option_separ_default_tl
  {
    { {\ }\__ccool_lang_and:{\ } }
    { ,{\ } }
    { ,{\ }\__ccool_lang_and:{\ } }
  }
}
```

(*End definition for* `\g__ccool_option_separ_tl`.)

# 9 prop

`\__ccool_prop_append:NN` `#1` : ⟨ $prop_1$ ⟩

#2 : ⟨ *prop*$_2$ ⟩

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

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

`\__ccool_prop_append:Nn`

#1 : ⟨ *prop* ⟩
#2 : ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__ccool_prop_append:Nn
{
  \__ccool_prop_append:cN{ \__ccool_prop_name:n {#2} } #1
}
```

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

`\__ccool_prop_clear_new:n`

#1 : ⟨ *tl var name* ⟩

```
\cs_new_protected:Nn \__ccool_prop_clear_new:n
{
  \exp_args:No \prop_clear_new:c{ \__ccool_prop_name:n {#1} }
}
```

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

`\__ccool_prop_clear_new_map:n`

#1 : ⟨ *keyval list* ⟩

```
\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
}
```

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

`\__ccool_prop_if_exist:nTF`

#1 : ⟨$tl_1$⟩
#2 : ⟨$tl_2$⟩
#3 : ⟨$tl_3$⟩

```
\cs_new:Nn \__ccool_prop_if_exist:nTF
{
  \prop_if_exist:cTF{ \__ccool_prop_name:n {#1} }{#2}{#3}
}
```

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

`\__ccool_prop_item:nn`

#1 : ⟨ *tl var name* ⟩
#2 : ⟨ *key* ⟩

```
\cs_new:Nn \__ccool_prop_item:nn
{
  \prop_item:cn { \__ccool_prop_name:n {#1} } {#2}
}
```

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

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

```
\cs_new:Npn \__ccool_prop_name:n #1{ __ccool_#1 }
```

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

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

```
\cs_new_protected:Nn \__ccool_prop_new:n
{
  \prop_new:c{ \__ccool_prop_name:n {#1} }
}
```

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

# 10 seq

`\__ccool_seq_from_prop:NNn` `#1` : ⟨ $seq_1$ ⟩
`#2` : ⟨ $seq_2$ ⟩ (keys)
`#3` : ⟨ *prop* ⟩

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

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

# 11 seq_use

`\__ccool_seq_from_prop:NNn`

```
%      \changes{v3.2}
%      {2021/09/20}
%      {Added~\cs[no-index]{__ccool_seq_use:Nn}~in~replacement~of~\cs[no-index][erw_seq_use:N
\msg_new:nnn{__ccool}{separ}{#1~expects~1~to~3~items,~#2}
\cs_new:Nn \__ccool_seq_use:Nn
{
  \exp_last_unbraced:NNf
  \seq_use:Nnnn #1
  \__ccool_tl_separators:n{#2}
}
\cs_new:Nn \__ccool_tl_separators:n
{ \__ccool_tl_separators:en{ \tl_count:n{#1} }{#1} }
\cs_new:Nn \__ccool_tl_separators:nn
{ \int_case:nnTF {#1}
  { {1}
    { \prg_replicate:nn{ 3 }{#2} }
    {2}
    {
```

```
      { \use_ii:nn #2 }
      { \use_i:nn #2 }
      { \use_i:nn #2 \use_ii:nn #2 }
    }
    {3}{#2}
  }
  { \c_empty_tl }
  {
    \msg_error:nnnn { __ccool }
    { separ }
    { \__ccool_tl_separators:nn }
    {#2}
  }
}
\cs_generate_variant:Nn \__ccool_tl_separators:nn { e }
```

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

# 12 Front-end

\CcoolClearimpl:cs:clear

\CcoolClear

```
\NewDocumentCommand{ \CcoolClear }
{ D<>{\g__ccool_option_param_tl} }
{
  \__ccool_prop_clear_new_map:n{#1}
}
```

(*End definition for* `\CcoolClear`. *This function is documented on page* 4.)

\CcoolHookimpl:cs:hook

\CcoolHook

```
\NewDocumentCommand{\CcoolHook}{}{\c_empty_tl}
```

(*End definition for* `\CcoolHook`. *This function is documented on page* 4.)

\CcoolLambdaimpl:cs:lambda

\CcoolLambda (*Note*[2])

```
%      \changes{v3.2}
%      {2021/09/20}{\cs[CcoolLambda]'s~implementation~switched~from~\pkg{erw-l3}~to~\pkg{lamb
\ProvideDocumentCommand \CcoolLambda { O{m} m }
{ \lambdax:nn{#1}{#2} }
```

(*End definition for* `\CcoolLambda`. *This function is documented on page* **??**.)

\CcoolOptionimpl:cs:option

\CcoolOption (*Note*[3]) (*Note*[4])

```
\NewDocumentCommand{ \CcoolOption }
{ O{ And, Expans, File, Inner, Param, Outer, Separ, Write } }
{
```

[2] [todo]: allow only m- or o-type arguments

[3] [todo]: Fix placeholders passed to options requiring code (only one pound sign)

[4] [abandon]: Requirement: write to file if Write; Update: redundant with \cs{Ccool}+Write

```
  \keys_set:nn{ __ccool }{#1}
}
```

(*End definition for* `\CcoolOption`. *This function is documented on page* 4.)

```
\keys_define:nn { __ccool }
{

And .code:n = { \__ccool_lang_and_update:e{ #1 } },
And .default:n = { \c__ccool_lang_and_tl },
And .initial:n = { \c__ccool_lang_and_tl },

Expans .multichoices:nn = { eo, ee, ex, xo, xe, xx }
{ \tl_gset_eq:NN \g__ccool_option_expans_tl \l_keys_choice_tl },
Expans .default:n = { xo },
Expans .initial:n = { xo },

%       \changes{v3.2}
%       {2021/09/20}
%       {Removed~module~key~File's~relianced~on~a~timestamp~(clumsy)}
File .code:n = { \tl_gset:Nx \g__ccool_log_file_tl{#1} } }
\cs_new_protected:Nn
\__ccool_keys_define_file:n
{\keys_define:nn { __ccool }
  {File .code:n = { \tl_gset:Nx \g__ccool_log_file_tl{#1} },
    File .default:n = { #1 },
    File .initial:n = { #1 } } }
\cs_generate_variant:Nn\__ccool_keys_define_file:n{e}
\__ccool_keys_define_file:e
{\exp_args:Ne\__ccool_aux_merge:nn{\c_sys_jobname_str}{_ccool_log}}

\keys_define:nn { __ccool }
{
  Inner .code:n={
    \__ccool_option_inner:n{#1}
    \exp_last_unbraced:Nf
    \__ccool_make_ccool:nnnn
    {
      { \g__ccool_option_param_tl }
      { \g__ccool_option_inner_tl }
      { \g__ccool_option_separ_tl }
      { \g__ccool_option_outer_tl }
    }
  },
Inner .value_required:n = false,
Inner .default:n = {####1},
Inner .initial:n = {####1},

Param .code:n={
  \__ccool_option_param:n{#1}
  \exp_last_unbraced:Nf
  \__ccool_make_ccool:nnnn
  {
    { \g__ccool_option_param_tl }
    { \g__ccool_option_inner_tl }
    { \g__ccool_option_separ_tl }
    { \g__ccool_option_outer_tl }
  }
```

```
},
Param .value_required:n = false,
Param .default:n = { Default },
Param .initial:n = { Default },

Outer .code:n={
  \__ccool_option_outer:n{#1}
  \exp_last_unbraced:Nf
  \__ccool_make_ccool:nnnn
  {
    { \g__ccool_option_param_tl }
    { \g__ccool_option_inner_tl }
    { \g__ccool_option_separ_tl }
    { \g__ccool_option_outer_tl }
  }
},
Outer .value_required:n = false,
Outer .default:n = { \ensuremath{####1} },
Outer .initial:n = { \ensuremath{####1} },

Separ .code:n={
  \__ccool_option_separ:n{#1}
  \exp_last_unbraced:Nf
  \__ccool_make_ccool:nnnn
  {
    { \g__ccool_option_param_tl }
    { \g__ccool_option_inner_tl }
    { \g__ccool_option_separ_tl }
    { \g__ccool_option_outer_tl }
  }
},
Separ .value_required:n = false,
Separ .default:n = { \c__ccool_option_separ_default_tl },
Separ .initial:n = { \c__ccool_option_separ_default_tl },

Write .code:n = {
  \bool_if:nTF{#1}
  {\__ccool_log_open:}
  {\__ccool_log_close:}
},
Write .value_required:n = false,
Write .default:n = \BooleanFalse,
Write .initial:n = \BooleanFalse

}
```

\CcoolRead

```
\NewDocumentCommand{\CcoolRead}
{o}
{
  \IfValueTF{#1}
  {\__ccool_log_read:e{#1}}
  {\__ccool_log_read:}
}
```

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

\CcoolVers

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

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

# 13 Closing

```
\ExplSyntaxOff
⟨/package⟩
```

# Change History

# 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.

## D



## E

### F



### G



### I



### K



### L



### M



### N



### O



### P



### Q



### R



### S



### T