

# The coolthms Package

Jonathan Zachhuber\*

Michael Fütterer†

Version v1.0, 2012/03/29

## Abstract

This package makes it possible to directly reference `\items` in theorem-like environments using the `ntheorem` and `cleveref` packages.

## Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>2</b>
<b>3</b>	<b>Formatting Details</b>	<b>3</b>
<b>4</b>	<b>Examples</b>	<b>4</b>
<b>5</b>	<b>Interaction with other packages</b>	<b>5</b>
<b>6</b>	<b>Implementation</b>	<b>6</b>

## 1 Overview

Especially when typesetting lecture notes, one encounters situations such as

```
\begin{theorem}\label{thm1}
\begin{enumerate}
  \item\label{1.1} First point.
  \item\label{1.2} Second point.
\end{enumerate}
\end{theorem}
```

and would subsequently like to write something like `\ref{1.1}` to get something of the form ‘Theorem 1 (a)’.

This, however, is not possible. Of course, one could retreat to writing something like `\ref{thm1}~\ref{1.1}`, but this is no satisfactory solution.

`\Label` The `coolthms` package therefore provides the `\Label` command to create a special kind of label that internally saves the name and number of a possible theorem-like environment enclosing it.

---

\*[jonathan.zachhuber@gmail.com](mailto:jonathan.zachhuber@gmail.com)

†[michaelfuetterer@gmail.com](mailto:michaelfuetterer@gmail.com)

In our above example, one could now write

```
\begin{theorem}\label{thm1}
\begin{enumerate}
\item\Label{1.1} First point.
\item\Label{1.2} Second point.
\end{enumerate}
\end{theorem}
```

and then `\cref{1.1}` provides the desired result. See [section 4](#) for some concrete examples.

`\definetheorem` For this to work, one needs to define the theorem-like environments with the new `\definetheorem` command. This is simply an extension of `ntheorem`'s `\newtheorem` command.

## 2 Usage

The `coolthms` package only defines three useful commands and uses one from the `cleveref` package.

`\definetheorem` The `\definetheorem` command expands `ntheorem`'s `\newtheorem` by saving information later to be used in generating the labels. The syntax is as follows:

```
\definetheorem[⟨counter₁⟩]{⟨env name⟩}[⟨thm plural name⟩]
{⟨thm name⟩}[⟨counter₂⟩]
```

This command defines *two* environments, `⟨env name⟩` and `n⟨env name⟩`. The first one is for numbered theorems, the second one is an unnumbered version. The displayed name of the theorem is `⟨thm name⟩`. You can optionally give the plural form `⟨thm plural name⟩`, which will be used if several theorems of this type are referenced at any one time.

The counter arguments are similar to those of `\newtheorem`. The `⟨counter₁⟩` is the counter that is used for this type of theorem. The package creates a dummy counter, named `thmcnt`, and this is the default value of `⟨counter₁⟩`. Hence, the default setting is for all different types of theorems to be numbered consecutively. If you want to number some type of theorem separately, you should specify a new counter name via the optional argument `⟨counter₁⟩`; if the counter does not exist, it will be created. You can also provide the counter of another theorem environment, to group several types of theorem together.

The `⟨counter₂⟩` is a counter that resets `⟨counter₁⟩` every time it is incremented. It has the default value `section` so the theorems are numbered within a section and `⟨counter₁⟩` is reset to 0 whenever a new section starts. If you want a theorem type to be numbered document-wide without the counter being reset at any time, you should give an empty (i.e. no) `⟨counter₂⟩` argument.

The default numbering of the theorem environment is `⟨counter₂⟩.⟨counter₁⟩`. See [section 4](#) for concrete examples.

`\theoremmarkup` `\theoremmarkup` is a command for formatting theorems. Similar to the corresponding commands from `ntheorem`, it is called *before* defining theorems with `\definetheorem`, affecting the appearance of all theorems defined afterwards until it is called again. Details can be found in [section 3](#).

`\Label` The `\Label` command replaces (or re-implements) the 'conventional' `\label` command and is to be used *only* inside theorem-like environments (which were previously defined with `\definetheorem!`). The `\label` command can (and should) of course still

be used, if no special behaviour is desired<sup>1</sup>. The syntax is exactly the same as for `\label`:

`\Label{\label name}`

`\cref`      The labels thus defined should then be referenced with:

`\cref{\label name}`

This is the `cleveref` ‘version’ of `\ref` and is being used here without being altered in any way.

The package can be called with several options, which are described in the table below.

Option	Default value	Description
<code>indent</code>	0em	The space every theorem’s <i>content</i> will be indented.
<code>separator</code>	:	The punctuation sign that will be printed after the caption.
<code>proofname</code>	<code>\proofname</code>	The caption for proofs.
<code>proofsymbol</code>	<code>\$_Box\$</code>	The symbol that will be printed at the end of proofs.
<code>proofcaptionstyle</code>	<code>\it</code>	The font shape in which the caption for proofs (as given in <code>proofname</code> ) is printed.
<code>proofindent</code>	0em	The space proofs will be indented.
<code>minskip</code>	0pt	The minimal theorem pre- and post skip amount.
<code>maxskip</code>	6pt	The maximal theorem pre- and post skip amount.

Note that the `\proofname` macro is defined by `babel` or `polyglossia` and is a language-specific string containing the proof name. If none of these package is loaded, we define `\proofname` just as “Proof” and use that as a default value.

### 3 Formatting Details

`\theoremmarkup`      The `\theoremmarkup` command is used to describe how your theorems will be formatted. It should be called prior to any `\definetheorem` command. All theorems you define afterwards with `\definetheorem` will be formatted in the way you have set with `\theoremmarkup`, until you invoke `\theoremmarkup` again to change these values. The syntax is:

`\theoremmarkup[\langle header font \rangle][\langle body font \rangle][\langle symbol \rangle]`  
`[\langle indentation \rangle][\langle separator \rangle][\langle numbering \rangle]`

All these parameters are optional and have the following default values:

---

<sup>1</sup>Actually, if you use the `Label` command inside an unnested theorem environment, or for the theorem itself, it will simply display the theorem number twice as the counter is used both in the reference name and, of course, the reference counter. See [section 6](#).

Option	Default value
header font	<code>\bf</code>
body font	<code>\normalfont</code>
symbol	<code>\relax</code>
intendation	the value of the package option <code>indent</code>
separator	<code>:</code>
numbering	arabic

For more detailed information about these parameters, see the documentation of the `ntheorem` package.

`\theoremstyle{mitnummern}` The package also provides (and uses with `\definetheorem`) three new theorem styles. They provide a numbered and an unnumbered theorem style, as well as one intended for proofs. They respect `ntheorem`'s `\theoremheaderfont` but put the optional title addon in parentheses and `\normalfont`. See [section 6](#) for details.

`\theoremstyle{keinenummern}`

`\theoremstyle{unserbeweis}`

`proof` The package also defines a theorem environment intended for proofs, which is called `proof`.

`\sectionname` While the `babel` and `polyglossia` packages define language specific commands for the word “chapter” (`\chaptername`), they include no `\sectionname`. The package therefore defines `\sectionname`, however at the moment it is left blank in all languages but German (“Abschnitt”) and English (“section”), see [section 6](#).

## 4 Examples

We define a theorem environment `thm` with caption “Theorem”:

```
\definetheorem{thm}{Theorem}
```

Now we use this environment to typeset a theorem:

```
\begin{thm}\label{testthm}
  This is a theorem.
\end{thm}
```

**Theorem 4.1:** This is a theorem.

Notice that here we use the conventional `\label` command, as we are not in a nested situation.

Now we’ll define – and then use – a more fancy theorem environment, `fancythm`. Before doing that, we invoke `\theoremmarkup`:

```
\theoremmarkup[\sc][\it][\textleaf][3em][.][Roman]
\definetheorem{fncythm}{Fancy Theorem}
\begin{fncythm}[Title]
  The \ctp{} package is a very useful package for typesetting theorems.
  This theorem is a long one, and we can see that its content is indented.
  We furthermore have an enumerated list of claims.
  \begin{enumerate}
    \item\Label{claim1} Claim 1
    \item\Label{claim2} Claim 2
  \end{enumerate}
\end{fncythm}
```

Thus the header font will be small caps, the body font will be italic, each fancy theorem will be ended by a leaf symbol, its content will be indented 3em, the punctuation sign after its caption will be a period, and the numbering will be Roman. Note that – by default – we use the same numbering as **Theorem 4.1**<sup>2</sup>. And indeed:

FANCY THEOREM 4.2 (Title). *The coolthms package is a very useful package for typesetting theorems. This theorem is a long one, and we can see that its content is indented. We furthermore have an enumerated list of claims.*

(a) Claim 1

(b) Claim 2



Finally, by writing `\cref{claim1}`, we get what we wanted: **Fancy Theorem 4.2 (a)**.

This even works for nested enumerates. However, we recommend using the `enumitem` package to improve the appearance of the reference. For example, when we define

```
\usepackage{enumitem}
\setenumerate[1]{leftmargin=*,labelindent=\parindent,label=(\alph*)}
\setenumerate[2]{leftmargin=*,labelindent=\parindent,label=(\roman*),%
ref=\theenumi~(\roman*)}
```

and we have a theorem

```
\begin{thm}
  Test theorem.
  \begin{enumerate}
    \item First point.
    \begin{enumerate}
      \item First sub-point.
      \item\Label{testlabel} Second sub-point.
    \end{enumerate}
  \end{enumerate}
  \item Second point.
\end{enumerate}
\end{thm}
```

**Theorem 4.3:** Test theorem.

(a) First point.

(i) First sub-point.

(ii) Second sub-point.

(b) Second point.

then writing `\cref{testlabel}` correctly yields **Theorem 4.3 (a) (ii)**.

## 5 Interaction with other packages

To achieve its goals, `coolthms` relies on several other packages, some of which are quite picky. Most notably, we use the `cleveref` package, which likes to be loaded at quite a late point in the document head (please refer to `cleveref`'s package documentation for a detailed account of its interaction with other packages).

---

<sup>2</sup>This is just `\cref{testthm}`.

In general, you will be on the safe side if `coolthms` is the last package you load. If you want to use the language-specific `\sectionname` commands, you should definitely load it *after* `babel` or `polyglossia`, otherwise they will have no effect.

Since we use the `ntheorem` package, `coolthms` *must* be loaded *after* `amsmath`, if this is used. The package works with and without `amsmath`, though.

## 6 Implementation

First we load the packages we'll be needing.

Before loading the `amssymb` package, we need to undefine some commands to avoid errors.

```
1 \let\Finv\@undefined
2 \let\Game\@undefined
3 \let\beth\@undefined
4 \let\gimel\@undefined
5 \let\daleth\@undefined
6 \RequirePackage{amssymb}
```

`hyperref` support is always good when linking stuff, we need lots of little `etoolbox` macros and `xargs` as well as `kvoptions` for our argument and option processing.

```
7 \RequirePackage{hyperref}
8 \RequirePackage{etoolbox}
9 \RequirePackage{ifthen}
10 \RequirePackage{xargs}
11 \RequirePackage{kvoptions}
```

We then check if `amsmath` has been loaded, so we know if to invoke the `amsmath` option in `ntheorem`.

```
12 \@ifpackageloaded{amsmath}
13   {\RequirePackage[thmmarks,amsmath,hyperref]{ntheorem}}
14   {\RequirePackage[thmmarks,hyperref]{ntheorem}}
```

Note that the `cleveref` package must be loaded *after* the `ntheorem` package.

```
15 \RequirePackage{cleveref}
```

`\ctp@hashchar` In order to write a verbatim ‘hash’ sign into our files later, it’s practical to write the `\catcode` change into a small macro:

```
16 \begingroup
17 \catcode`\#=12
18 \gdef\ctp@hashchar{#}%
19 \endgroup
```

`\proofname` We provide options for the proofname, the theorem separator, the proof end symbol and the amount to indent theorem content. The default value of `proofname` should be `\proofname`, which is set by `babel` or `polyglossia`. If the command is not defined, we simply define it as “Proof”.

```
20 \@ifundefined{proofname}{\newcommand{\proofname}{Proof}}{}
21 \DeclareStringOption[\proofname]{proofname}
22 \DeclareStringOption[$\Box$]{proofsymbol}
23 \DeclareStringOption[\it]{proofcaptionstyle}
24 \DeclareStringOption[0em]{proofindent}
25 \DeclareStringOption[:]{separator}
26 \DeclareStringOption[0em]{indent}
```

```

27 \DeclareStringOption[0pt]{minskip}
28 \DeclareStringOption[6pt]{maxskip}
29 \DeclareLocalOptions{separator,indent,minskip,maxskip%
30   proofname,proofsymbol,proofcaptionstyle,proofindent}
31 \ProcessKeyvalOptions*

```

Now we need to define various (an unnumbered, a numbered and a third one for proofs) `theoremstyles`<sup>3</sup> we will be using:

```

32 \newtheoremstyle{keinenummern}%
33   {\item[\hskip\labelsep\theorem@headerfont ##1\theorem@separator]}%
34   {\item[\hskip\labelsep\theorem@headerfont ##1\ %
35     {\normalfont(##3)}\theorem@separator]}
36 \newtheoremstyle{mitnummern}%
37   {\item[\hskip\labelsep\theorem@headerfont ##1\ ##2\theorem@separator]}%
38   {\item[\hskip\labelsep\theorem@headerfont ##1\ ##2\ %
39     {\normalfont(##3)}\theorem@separator]}
40 \newtheoremstyle{unserbeweis}%
41   {\item[\hskip\labelsep\theorem@headerfont ##1\theorem@separator]}%
42   {\item[\hskip\labelsep\theorem@headerfont ##3\theorem@separator]}

```

If there are no chapters (e.g. article class), we should manually create that counter and set it to 1, as we need that in the name of our label:

```

43 \@ifundefined{c@chapter}{%
44   \newcounter{chapter}%
45   \setcounter{chapter}{1}%
46 }{}%

```

`\definetheorem` Now comes the real work: the `\definetheorem` command. `\definetheorem` takes five arguments and passes them to `ntheorem`'s `\newtheorem` in a slightly altered order. We then create a numbered theorem style by name of #2 and an unnumbered style by name of n#2. This is necessary as the starred versions have a different meaning in the `ntheorem` package.

The optional arguments are first checked (i.e. nothing happens if they are not set) and then passed to `\newtheorem` to create the dummy counters that will later be used for numbering the environments.

```

47 \newcommand*{\definetheorem}[5][1=thmcnt,3=,5=section]{
48   \@ifundefined{c@#1}{
49     \@ifundefined{c@#5}{
50       \newtheorem{#1}{#1}
51     }{
52       \newtheorem{#1}{#1}[#5]
53     }
54   }{}
55   \theoremstyle{mitnummern}
56   \newtheorem{#2}[#1]{#4}
57   \theoremstyle{keinenummern}
58   \newtheorem{n#2}[#1]{#4}

```

When simply referring to the environment (i.e. something from a `\label`, not `\Label` command!), we want the reference to consist of '*theorem name* *theorem number*' and all of it should be a hyperlink. `\crefname` takes three arguments: The name of the theorem environment, the singular and plural form of the theorem name. These are

---

<sup>3</sup>Here we use the `ntheorem` package.

stored in #2, #4 and #3, respectively. If no plural form is given, i.e. #3 is undefined, we simply replace it with its singular form (#4):

```
59 \ifblank{#3}{
60   \crefname{#2}{#4}{#4}
61 }{
62   \crefname{#2}{#4}{#3}
63 }
64 \crefformat{#2}{##2#4~##1#3}
```

In the unnumbered version we need to subtract 1 from the counter, as it is nonetheless incremented.

```
65 \BeforeBeginEnvironment{n#2}{\addtocounter{#1}{-1}}
```

`\ctp@labelcode` Now, every time we call our new theorem environment, we want to create a new *unique* label (`\ctp@labelcode`), which we can then use as the label of the nested enumerate environments. However, at this point the counter #1 has not been incremented yet, so we need to do (and later undo) this.

```
66 \BeforeBeginEnvironment{#2}{%
67   \addtocounter{#1}{1}%
68   \edef\ctp@labelcode%
69     {ctp@#2@\roman{chapter}@\roman{section}@\arabic{#1}}\relax%
```

We then write all this information (including the *format* of the label) to the aux file so that it is available at the next run of  $\text{\LaTeX}$ .

```
70   \immediate\write\@auxout{\string\crefname{\ctp@labelcode}%
71     {#4\noexpand~\csname the#1\endcsname}%
72     {#4 plural\noexpand~\csname the#1\endcsname}}\relax%
73   \immediate\write\@auxout{\string\crefformat{\ctp@labelcode}%
74     {\string##2#4\noexpand~\csname the#1\endcsname\noexpand~%
75       \ctp@hashchar1\ctp@hashchar3}}\relax%
```

`\Label` We finally (re)define the `\Label` command. All it does is call the classic `\label` command (from `cleveref`) with our unique label name as identifier:

```
76   \edef\Label##1{\noexpand\label[\ctp@labelcode]{##1}}%
```

Now we're done, all we need to do is correct #1.

```
77   \addtocounter{#1}{-1} %
78 }%
79 }%
```

`\theoremmarkup` Now we define the `\theoremmarkup` command, which is described above.

```
80 \newcommand*{\theoremmarkup}[6][1=\bf,2=\normalfont,3=\relax,%
81   4=\coolthms@indent,5=\coolthms@separator,6=arabic]{
82   \theoremheaderfont{\hspace*{#4}#1}
83   \theorembodyfont{#2}
84   \theoremsymbol{#3}
85   \theoremindent#4\relax
86   \theoremseparator{#5}
87   \theoremnumbering{#6}
88 }
```

And then we want to adjust the format for the other types of references:

```
89 \crefformat{equation}{#2(#1)#3}
90 \crefformat{chapter}{#2\chaptername~#1#3}
```



`\sectionname` Interestingly enough, `\sectionname` is provided neither by `polyglossia` nor `babel`, so we add it in the appropriate languages.

```

91 \@ifundefined{sectionname}{\newcommand{\sectionname}{}{}}{}
92 \gappto\captionsgerman{\renewcommand{\sectionname}{Abschnitt}}%
93 \gappto\captionsenglish{\renewcommand{\sectionname}{section}}
94 \crefformat{section}{#2\sectionname~#1#3}

```

`\theorempreskipamount` We set theorem pre- and post skip amounts.

```

\theorempostskipamount 95 \theorempreskipamount\coolthms@minskip plus \coolthms@maxskip\relax
96 \theorempostskipamount\coolthms@minskip plus \coolthms@maxskip\relax

```

This is for proofs:

```

97 \theoremstyle{unserbeweis}
98 \theoremmarkup[\coolthms@proofcaptionstyle][\normalfont]%
99 [\coolthms@proofsymbol][\coolthms@proofindent]
100 \newtheorem{proof}{\coolthms@proofname}

```

At the end we invoke `\theoremmarkup` to set everything back to the default values.

```
101 \theoremmarkup
```

## Change History

VO.1	V1.0
General: Started project . . . . . 9	General: First public version . . . . . 9

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>Symbols</b>	<code>\begingroup</code> . . . . . 16	<code>\coolthms@proofsymbol</code> . . . . . 99
<code>\#</code> . . . . . 17	<code>\beth</code> . . . . . 3	<code>\coolthms@separator</code> 81
<code>\@auxout</code> . . . . . 70, 73	<code>\bf</code> . . . . . 80	<code>\cref</code> . . . . . 3
<code>\@ifpackageloaded</code> . . 12	<code>\Box</code> . . . . . 22	<code>\crefformat</code> . . . . .
<code>\@ifundefined</code> . . . . . 20, 43, 48, 49, 91	<b>C</b>	. . . . . 64, 73, 89, 90, 94
<code>\@undefined</code> . . 1, 2, 3, 4, 5	<code>\captionsenglish</code> . . . 93	<code>\crefname</code> . . . . . 60, 62, 70
	<code>\captionsgerman</code> . . . . 92	<code>\csname</code> . . . . . 71, 72, 74
	<code>\catcode</code> . . . . . 17	<code>\ctp@hashchar</code> . . . . 16, 75
<code>\_</code> . . . . . 34, 37, 38	<code>\chaptername</code> . . . . . 90	<code>\ctp@labelcode</code> . . . . . 66, 70, 73, 76
<b>A</b>	<code>\coolthms@indent</code> . . . 81	
<code>\addtocounter</code> . . 65, 67, 77	<code>\coolthms@maxskip</code> 95, 96	<b>D</b>
<code>\arabic</code> . . . . . 69	<code>\coolthms@minskip</code> 95, 96	<code>\daleth</code> . . . . . 5
<b>B</b>	<code>\coolthms@proofcaptionstyle</code> . . . . . 98	<code>\DeclareLocalOptions</code> 29
<code>\BeforeBeginEnvironment</code> . . . . . 65, 66	<code>\coolthms@proofindent</code> . . . . . 99	<code>\DeclareStringOption</code> . . . . . 21, 22,
	<code>\coolthms@proofname</code> 100	23, 24, 25, 26, 27, 28

<code>\definetheorem</code> .. 2, 2, <u>47</u>	<code>\labelsep</code> .....	<code>\setcounter</code> .....
	.. 33, 34, 37, 38, 41, 42	<code>\string</code> .....
	<code>\let</code> .....	70, 73, 74
	1, 2, 3, 4, 5	
<b>E</b>	<b>N</b>	<b>T</b>
<code>\edef</code> .....	<code>\newcommand</code> .....	<code>\theorem@headerfont</code>
68, 76	20, 91	.. 33, 34, 37, 38, 41, 42
<code>\endcsname</code> .....	<code>\newcommandx</code> .....	<code>\theorem@separator</code>
71, 72, 74	47, 80	.. 33, 35, 37, 39, 41, 42
<code>\endgroup</code> .....	<code>\newcounter</code> .....	<code>\theorembodyfont</code> ...
19	44	83
environments:	<code>\newtheorem</code> .....	<code>\theoremheaderfont</code> ..
<code>proof</code> .....	.. 50, 52, 56, 58, 100	82
4	<code>\newtheoremstyle</code> ..	<code>\theoremindent</code> .....
<b>F</b>	.. 32, 36, 40	85
<code>\Finv</code> .....	<code>\noexpand</code> ...	<code>\theoremmarkup</code> ....
1	71, 72, 74, 76	.. 2, 3, 80, 98, 101
<b>G</b>	<code>\normalfont</code> ..	<code>\theoremnumbering</code> ..
<code>\Game</code> .....	35, 39, 80, 98	87
2	<b>P</b>	<code>\theorempostskipamount</code>
<code>\gappto</code> .....	<code>\ProcessKeyvalOptions</code>	.....
92, 93	.....	95
<code>\gdef</code> .....	proof (environment) ....	<code>\theorempreskipamount</code>
18	4	.....
<code>\gimel</code> .....	<code>\proofname</code> .....	95
4	20	<code>\theoremseparator</code> ..
<b>H</b>	<b>R</b>	86
<code>\hskip</code> .. 33, 34, 37, 38, 41, 42	<code>\relax</code> .....	<code>\theoremstyle</code> .. 55, 57, 97
<code>\hspace</code> .....	69,	<code>\theoremstyle\{keinenummern\}</code>
82	72, 75, 80, 85, 95, 96	.....
<b>I</b>	<code>\renewcommand</code> ....	4
<code>\ifblank</code> .....	92, 93	<code>\theoremstyle\{mitnummern\}</code>
59	<code>\RequirePackage</code> 6, 7,	.....
<code>\immediate</code> .....	8, 9, 10, 11, 13, 14, 15	4
70, 73	69	<code>\theoremsymbol</code> .....
<code>\it</code> .....		84
23		
<code>\item</code> .. 33, 34, 37, 38, 41, 42		
<b>L</b>	<b>S</b>	<b>W</b>
<code>\Label</code> .....	<code>\sectionname</code> .....	<code>\write</code> .....
1, 2, <u>76</u>	4, <u>91</u>	70, 73
<code>\label</code> .....		
76		