The coolthms Package

Jonathan Zachhuber* Michael Fütterer†

Version v1.2, 2013/02/04

Abstract

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

Contents

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

*jonathan.zachhuber@gmail.com
†michaelfuetterer@gmail.com

kind of label that internally saves the name and number of a possible `theorem`-like environment enclosing it.

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 3 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

2.1 Main Commands

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

`\theoremmarkup` This command changes the formatting of theorems. It is explained in section 2.3.

`\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 seperately, 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. Its default value is `section`, so the theorems are numbered within sections 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 ⟨*counter2*⟩ argument.

The default numbering of the theorem environment is ⟨*counter2*⟩.⟨*counter1*⟩. See section 3 for concrete examples.

`\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[1]. 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.

2.2 Package Options

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

Option	Default value	Description
indent	0em	The space every theorem's *content* will be indented.
separator	:	The punctuation sign that will be printed after the caption.
proofname	\proofname	The caption for proofs.
proofsymbol	$\Box$	The symbol that will be printed at the end of proofs.
proofcaptionstyle	\it	The font shape in which the caption for proofs (as given in proofname) is printed.
proofindent	indent	The space proofs will be indented.
minskip	0pt	The minimal theorem pre- and post skip amount.
maxskip	6pt	The maximal theorem pre- and post skip amount.
externalchapters	False	Turn on external chapter mode (see below).

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. If no value is supplied for proofindent, proofs are indented the same amount as all other theorems (i.e. the default value is taken from indent).

externalchapters When the (boolean) option externalchapters is given *and* you use a document class that has chapters, a special behaviour is turned on (if there are no chapters, nothing happens). Usually, sections are numbered within chapters by ⟨*chapter number*⟩.⟨*section number*⟩. Consequently, when numbering theorem-like environments within a section (which is the default behaviour), such an environment gets the number ⟨*chapter number*⟩.⟨*section number*⟩.⟨*thm number*⟩.

In the external chapter mode, however, section numbers are *not* preceded by the chapter number, i.e. they are numbered by a *single* (arabic) number (it follows that subsections then have only two numbers instead of three, and so on). To avoid having chapters and sections with the same numbers, chapters are numbered by roman numerals in this mode (if you don't like this, you can change it by redefining \thechapter).

[1] 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 5.

Consequently theorems are then numbered by ⟨*section number*⟩.⟨*thm number*⟩, without any reference to the chapter in their number. Therefore, there can be theorems that have the same number (if they are in the same section in different chapters). To avoid confusion, in external chapter mode, in every reference to anything that is in another chapter as the current one, the number for whatever we are referring to is additionally preceded by the chapter number (except when referencing a chapter). When referring to something from the current chapter, the chapter number is omitted.

2.3 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[`⟨*header font*⟩`][`⟨*body font*⟩`][`⟨*symbol*⟩`][`⟨*indentation*⟩`][`⟨*separator*⟩`][`⟨*numbering*⟩`]`

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

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

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

`\theoremstyle{mitnummern}` `\theoremstyle{keinenummern}` `\theoremstyle{unserbeweis}` 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 5 for details.

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

3 Examples

We define a theorem enviroment `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 3.1: 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 3.1[2]. And indeed:

Fancy Theorem 3.2 (Title). *The `cooltbms` 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 3.2 (a).

This even works for nested `enumerate`s. However, we recommend using the `enumitem` package to improve the appeareance 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}
  \item Second point.
  \end{enumerate}
\end{thm}
```

Theorem 3.3: Test theorem.

[2] This is just `\cref{testthm}`.

(a) First point.

(i) First sub-point.

(ii) Second sub-point.

(b) Second point.

then writing `\cref{testlabel}` correctly yields Theorem 3.3 (a) (ii).

4 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).

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.

5 Implementation

First we load the packages we'll be needing.

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

```
\let\Finv\@undefined
\let\Game\@undefined
\let\beth\@undefined
\let\gimel\@undefined
\let\daleth\@undefined
\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.

```
\RequirePackage{hyperref}
\RequirePackage{etoolbox}
\RequirePackage{scrbase} % for \ifstr string comparison macro
\RequirePackage{letltxmacro}
\RequirePackage{ifthen}
\RequirePackage{xargs}
\RequirePackage{kvoptions}
```

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

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

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

```
\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:

```
\begingroup
\catcode`\#=12
\gdef\ctp@hashchar{#}%
\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". The value of proofindent is set to that of indent if none is specified.

There's a funny aspect of using \proofname here: If your proof environment is named "proof", then we have a problem, because if it's ever called with an *optional* argument, ntheorem defines the *internal* command \proofname to be the optional argument just passed to the environment. This results in *all* proofs after this to have *only* the optional argument of *this* proof as their title!

\ctp@temp To solve this, we simply expand \proofname, revealing the string behind it. The only problem is that if we do so immediately, we might miss any change of language, i.e. a "legal" change of \proofname by polyglossia or babel. Therefore, we define \ctp@temp to be an unexpanded \proofname first and – using \AtBeginDocument – substitute that for an expanded version and do so *every time the language is changed*. Then, at the end of a proof, we simply substitute the "real" \proofname for the one just created by the proof environment.

```
\@ifundefined{proofname}{\newcommand{\proofname}{Proof}}{}
\let\ctp@temp\proofname\relax
\AtBeginDocument{\edef\ctp@temp{\proofname}%
   \apptocmd{\select@language}{\edef\ctp@temp{\proofname}}{}{}}
\AfterEndEnvironment{proof}{\gdef\proofname{\ctp@temp}}
\DeclareStringOption[\ctp@temp]{proofname}
\DeclareStringOption[$\Box$]{proofsymbol}
\DeclareStringOption[\it]{proofcaptionstyle}
\DeclareStringOption[\coolthms@indent]{proofindent}
\DeclareStringOption[:]{separator}
\DeclareStringOption[0em]{indent}
\DeclareStringOption[0pt]{minskip}
\DeclareStringOption[6pt]{maxskip}
\DeclareBoolOption{externalchapters}
\DeclareLocalOptions{separator,indent,minskip,maxskip%
  proofname,proofsymbol,proofcaptionstyle,proofindent,externalchapters}
\ProcessKeyvalOptions*
```

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:

```
\@ifundefined{c@chapter}{%
  \newcounter{chapter}%
  \setcounter{chapter}{1}%
}{}%
```

We now handle external chapters. If these were requested, \thesection should be redefined to contain only a single number.

```
\ifcoolthms@externalchapters
\renewcommand{\thechapter}{\Roman{chapter}}
\renewcommand{\thesection}{\arabic{section}}
```

Next, we overwrite some definitions made by `cleveref`, namely `\refstepcounter@noarg` and `\refstepcounter@optarg` (which together are used by `cleveref` to redefine the `\refstepcounter` macro). These are invoked when a counter is incremented, and create `\cref@currentlabel`. This definition is altered by us to contain a conditional which might print the chapter number. However, we repeat `cleveref`'s original definition of `\cref@currentlabel` to define `\cref@old@currentlabel`, which we will need later to define the `\Label` command. The last `\ifstr` command ensures that no extra chapter number is printed when referring a chapter.

Note that we are still in the external chapters case.

```
\def\refstepcounter@noarg#1{%
  \cref@old@refstepcounter{#1}%
  \cref@constructprefix{#1}{\cref@result}%
  \@ifundefined{cref@#1@alias}%
    {\def\@tempa{#1}}%
    {\def\@tempa{\csname cref@#1@alias\endcsname}}%
  \edef\chapter@current@value{\the\value{chapter}}
  \protected@edef\cref@currentlabel{%
    [\@tempa][\arabic{#1}][\cref@result]%
    \string\ifstr{\string\the\string\value{chapter}}%
    {\chapter@current@value}{\string\relax}{\thechapter.}%
    \csname p@#1\endcsname\csname the#1\endcsname}
  \protected@edef\cref@old@currentlabel{%
    [\@tempa][\arabic{#1}][\cref@result]%
    \csname p@#1\endcsname\csname the#1\endcsname}
 \ifstr{\@tempa}{chapter}{\protected@edef\cref@currentlabel{\cref@old@currentlabel}}{}}
\def\refstepcounter@optarg[#1]#2{%
  \cref@old@refstepcounter{#2}%
  \cref@constructprefix{#2}{\cref@result}%
  \edef\chapter@current@value{\the\value{chapter}}
  \protected@edef\cref@currentlabel{%
    [#1][\arabic{#2}][\cref@result]%
    \string\ifstr{\string\the\string\value{chapter}}%
    {\chapter@current@value}{\string\relax}{\thechapter.}%
    \csname p@#2\endcsname\csname the#2\endcsname}
  \protected@edef\cref@old@currentlabel{%
    [#1][\arabic{#2}][\cref@result]%
    \csname p@#2\endcsname\csname the#2\endcsname}
 \ifstr{#1}{chapter}{\protected@edef\cref@currentlabel{\cref@old@currentlabel}}{}}
```

Next we redefine the `label` command. We are still in the external chapters case.

```
    \def\ctp@label@noarg#1{%
    \letcs{\mycurrentlabel}{@currentlabel}
    \expandafter\def\csname @currentlabel\endcsname{\string\ifstr%
      {\string\the\string\value{chapter}}{\the\value{chapter}}%
      {\string\relax}{\thechpt}\mycurrentlabel}
    \label@noarg{#1}
    \cslet{@currentlabel}{\mycurrentlabel}
    }%
    \def\ctp@label@optarg[#1]#2{%
    \letcs{\mycurrentlabel}{@currentlabel}
```

```
    \expandafter\def\csname @currentlabel\endcsname{\string\ifstr%
      {\string\the\string\value{chapter}}{\the\value{chapter}}%
      {\string\relax}{\thechpt}\mycurrentlabel}
    \label@optarg[#1]{#2}
    \cslet{@currentlabel}{\mycurrentlabel}
    \def\label{\@ifnextchar[\ctp@label@optarg\ctp@label@noarg}%]
    }
```

This is where the external chapters case ends.

```
\else % belongs to \ifcoolthms@externalchapters
\fi
```

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

```
\newtheoremstyle{keinenummern}%
  {\item[\hskip\labelsep\theorem@headerfont ##1\theorem@separator]}%
  {\item[\hskip\labelsep\theorem@headerfont ##1\ %
    {\normalfont(##3)}\theorem@separator]}
\newtheoremstyle{mitnummern}%
  {\item[\hskip\labelsep\theorem@headerfont ##1\ ##2\theorem@separator]}%
  {\item[\hskip\labelsep\theorem@headerfont ##1\ ##2\ %
    {\normalfont(##3)}\theorem@separator]}
\newtheoremstyle{unserbeweis}%
  {\item[{\hskip\labelsep\theorem@headerfont ##1\theorem@separator}]}%
  {\item[\hskip\labelsep{\theorem@headerfont ##3\theorem@separator}]}
```

`\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.

```
\newcommandx*{\definetheorem}[5][1=thmcnt,3=,5=section]{
  \@ifundefined{c@#1}{
    \@ifundefined{c@#5}{
      \newtheorem{#1}{#1}
    }{
      \newtheorem{#1}{#1}[#5]
    }
  }{}
  \theoremstyle{mitnummern}
  \newtheorem{#2}[#1]{#4}
  \theoremstyle{keinenummern}
  \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

[3] 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):

```
\ifblank{#3}{
  \crefname{#2}{#4}{#4}
}{
  \crefname{#2}{#4}{#3}
}
\crefformat{#2}{##2#4~##1##3}
```

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

```
\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.

```
\BeforeBeginEnvironment{#2}{%
   \addtocounter{#1}{1}%
   \edef\ctp@labelcode%
      {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 LaTeX.

```
\ifcoolthms@externalchapters
    \immediate\write\@auxout{\string\crefname{\ctp@labelcode}%
              {#4\noexpand~%
                \string\ifstr{\string\the\string\value{chapter}}%
                {\chapter@current@value}{\string\relax}{\thechapter.}%
                \csname the#1\endcsname}%
              {#3\noexpand~\csname the#1\endcsname}}\relax%
    \immediate\write\@auxout{\string\crefformat{\ctp@labelcode}%
              {\string##2#4\noexpand~%
                \string\ifstr{\string\the\string\value{chapter}}%
                {\chapter@current@value}{\string\relax}{\thechapter.}%
                \csname the#1\endcsname\noexpand~%
                 \ctp@hashchar1\ctp@hashchar3}}\relax%
\else
    \immediate\write\@auxout{\string\crefname{\ctp@labelcode}%
              {#4\noexpand~\csname the#1\endcsname}%
              {#4 plural\noexpand~\csname the#1\endcsname}}\relax%
    \immediate\write\@auxout{\string\crefformat{\ctp@labelcode}%
              {\string##2#4\noexpand~\csname the#1\endcsname\noexpand~%
                 \ctp@hashchar1\ctp@hashchar3}}\relax%
\fi
```

`\Label` We finally (re)define the `\Label` command. Without the `externalchapters` option, all it does is call the classic `\label` command (from `cleveref`) with our unique label name as identifier. With that option, it does exactly what `cref`'s original `\label@optarg` command (which is simply the `label` command with an optional argument, as defined by `cref`) would do if it were called with `\ctp@labelcode` as optional argument, except that it uses `\cref@old@currentlabel` instead of `\cref@currentlabel`.

```
\ifcoolthms@externalchapters
```

```
      \def\Label##1{\cref@old@label{##1}%
        \protected@edef\cref@currentlabel{%
          \expandafter\cref@override@label@type%
          \cref@old@currentlabel\@nil{\ctp@labelcode}}%
        \protected@write\@auxout{}%
        {\string\newlabel{##1@cref}{{\cref@currentlabel}{\thepage}}}}
\else
      \edef\Label##1{\noexpand\label[\ctp@labelcode]{##1}}%
\fi
```

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

```
      \addtocounter{#1}{-1} %
  }% End of \BeforeBeginEnvironment
}% End of \newcommandx*{\definetheorem}
```

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

```
\newcommandx*{\theoremmarkup}[6][1=\bf,2=\normalfont,3=\relax,%
              4=\coolthms@indent,5=\coolthms@separator,6=arabic]{
```

For some reason, `\hspace*{-#4}` lets the theorem start just slightly into the margin (i.e. somewhere in the conversion process about one character space gets lost). Using `\theorem@indent` solves the problem, although it remains unclear exactly why.

```
  \theoremheaderfont{\hspace*{-\theorem@indent}#1}
  \theorembodyfont{#2}
  \theoremsymbol{#3}
  \theoremindent#4\relax
  \theoremseparator{#5}
  \theoremnumbering{#6}
}
```

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

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

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

```
\theorempreskipamount\coolthms@minskip plus \coolthms@maxskip\relax
\theorempostskipamount\coolthms@minskip plus \coolthms@maxskip\relax
```

This is for proofs:

```
\theoremstyle{unserbeweis}
\theoremmarkup[\coolthms@proofcaptionstyle][\normalfont]%
  [\coolthms@proofsymbol][\coolthms@proofindent]
\expandafter\newtheorem{proof}{\coolthms@proofname}
```

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

```
\theoremmarkup
```

Change History

v0.1

General: Started project 11

v1.0

General: First public version 11

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.