THE CNLTX BUNDLE

Documentation for \LaTeX\ Packages or Classes

V0.12    2015/01/11

\LaTeX\ tools and documenting facilities the \textit{cn} way

Clemens \textsc{Niederberger}

https://github.com/cgnieder/cnltx

contact@mychemistry.eu

A versatile bundle of packages and classes for consistent formatting of
control sequences, package options, source code examples, and writing
a package manual (including an index containing the explained control
sequences, options, ...).

The bundle also provides several other small ideas of mine such as a
mechanism for providing abbreviations \textit{etc}. Not at least it provides a
number of programming tools.

Table of Contents

I. About The Bundle 3
   1. Background 3
   2. Bundled Packages, Classes and Files 4
   3. License and Requirements 5
   4. Usage of the Bundle 6

II. Details of Available Commands, Environments and Options 7
   5. Options and Setup 7

6. Available Commands 7
   6.1. Description of Macros, Environments and Options . . . . 7
   6.2. Versioning Commands, Licensing and Related Stuff . . . . . 9
   6.3. Input Source Code Files . . . . . 11

7. Available Environments 11
   7.1. Description Environments . . . . . 11
   7.2. Source Code Environments . . . . . . 12

8. Usage of the Various Functions 13
   8.1. Command Descriptions . . . . . . . . 13
   8.2. Option Descriptions . . . . . . . . . 15
   8.3. Environment Descriptions . . . . . . . 17
   8.4. Code Examples . . . . . . . . . . . . . 18
   8.5. Compile Source Examples . . . . . . . 20
   8.5.1. The Compilation Process 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.5.2. Floating Output . . .</td>
<td>23</td>
</tr>
<tr>
<td>8.5.3. Selective Output . . .</td>
<td>24</td>
</tr>
<tr>
<td>8.6. Example File . . . . . .</td>
<td>26</td>
</tr>
<tr>
<td>8.7. Additional Functionality Provided by \texttt{CNLTX-BASE} . . .</td>
<td>27</td>
</tr>
<tr>
<td>8.7.1. Looking for Trailing Punctuation . . .</td>
<td>27</td>
</tr>
<tr>
<td>8.7.2. Counter Representation Commands . . .</td>
<td>28</td>
</tr>
<tr>
<td>8.7.3. Expandable Document Commands . .</td>
<td>30</td>
</tr>
<tr>
<td>8.8. Additional Functionality Provided by \texttt{CNLTX-TOOLS} . .</td>
<td>31</td>
</tr>
<tr>
<td>8.8.1. Commands for Defining Different Document Macros . . .</td>
<td>31</td>
</tr>
<tr>
<td>8.8.2. Defining Abbreviations</td>
<td>33</td>
</tr>
<tr>
<td>8.8.3. Predefined Abbreviations</td>
<td>33</td>
</tr>
<tr>
<td>9. Formatting Possibilities</td>
<td>36</td>
</tr>
<tr>
<td>9.1. Formatting by Redefining Hooks</td>
<td>36</td>
</tr>
<tr>
<td>9.2. Formatting by Setting Options</td>
<td>37</td>
</tr>
<tr>
<td>10. Commands, Options and Further Settings Directly Related to the \texttt{CNLTX-DOC} Class</td>
<td>38</td>
</tr>
<tr>
<td>10.1. Using Class Options . . .</td>
<td>38</td>
</tr>
<tr>
<td>10.2. Information on the Described Package or Class . . . . .</td>
<td>38</td>
</tr>
<tr>
<td>10.3. Building of the Manuals Title Page . . .</td>
<td>39</td>
</tr>
<tr>
<td>10.4. A Quotation Environment</td>
<td>40</td>
</tr>
<tr>
<td>10.5. Predefined Preamble . . .</td>
<td>41</td>
</tr>
<tr>
<td>10.6. Predefined Indexing . . .</td>
<td>42</td>
</tr>
<tr>
<td>10.7. Bibliography with \texttt{biblatex} . .</td>
<td>43</td>
</tr>
<tr>
<td>10.7.1. Bibliography Entry Types package, class and bundle for \texttt{biblatex}</td>
<td>43</td>
</tr>
<tr>
<td>10.7.2. Automatic Bibliography</td>
<td>45</td>
</tr>
<tr>
<td>11. Predefined listings and \texttt{mdframed} Styles</td>
<td>45</td>
</tr>
<tr>
<td>11.1. \texttt{mdframed} . . .</td>
<td>45</td>
</tr>
<tr>
<td>11.2. \texttt{listings} . . .</td>
<td>46</td>
</tr>
<tr>
<td>11.2.1. \LaTeX Sourcecode</td>
<td>46</td>
</tr>
<tr>
<td>11.2.2. \LaTeX Entries</td>
<td>47</td>
</tr>
<tr>
<td>11.2.3. \texttt{makeindex} Style Files</td>
<td>48</td>
</tr>
<tr>
<td>12. PDF Strings and \texttt{hyperref}</td>
<td>49</td>
</tr>
<tr>
<td>13. Predefined Colors and Color-Schemes</td>
<td>49</td>
</tr>
<tr>
<td>13.1. Explicitly Defined Colors</td>
<td>49</td>
</tr>
<tr>
<td>13.2. Actual Used Color Names and Color Schemes</td>
<td>50</td>
</tr>
<tr>
<td>14. Language Support</td>
<td>52</td>
</tr>
<tr>
<td>III. Appendix</td>
<td>52</td>
</tr>
<tr>
<td>A. Internal Helper Commands</td>
<td>52</td>
</tr>
<tr>
<td>A.1. Defined by \texttt{CNLTX-BASE}</td>
<td>52</td>
</tr>
<tr>
<td>A.1.1. Related to the Bundle</td>
<td>54</td>
</tr>
<tr>
<td>A.1.2. Programming Tools</td>
<td>54</td>
</tr>
<tr>
<td>A.2. Defined by \texttt{CNLTX-DOC}</td>
<td>58</td>
</tr>
<tr>
<td>A.3. Defined by \texttt{CNLTX-EXAMPLE}</td>
<td>59</td>
</tr>
<tr>
<td>A.4. Defined by \texttt{CNLTX-LISTINGS}</td>
<td>60</td>
</tr>
<tr>
<td>A.5. Defined by \texttt{CNLTX-TOOLS}</td>
<td>60</td>
</tr>
<tr>
<td>B. List of Known \LaTeX Control Sequences</td>
<td>61</td>
</tr>
<tr>
<td>C. List of Known \LaTeX Environments</td>
<td>66</td>
</tr>
<tr>
<td>D. List of Entries in \texttt{cnltx.bib}</td>
<td>66</td>
</tr>
<tr>
<td>E. Bibliography</td>
<td>67</td>
</tr>
<tr>
<td>F. Index</td>
<td>70</td>
</tr>
</tbody>
</table>
Part I.
About The Bundle

1. Background

The \texttt{cnltx} bundle contains different packages and classes.\footnote{Well, \texttt{one} class for the time being.} I developed it as a successor of my class \texttt{cnpkgdoc} [Nie13b] that I used until now for writing the documentation of my packages. The intention behind the new bundle is a cleaner interface and less unnecessary ballast, hence the separation into packages and classes. This is actually a bit of a contradiction: the document class \texttt{cnltx-doc} loads \textit{all} packages of the bundle which makes it more feature-rich than \texttt{cnpkgdoc} ever used to be. The bundle provides source code environments that also print the output and defines quite a lot of macros for formatting of control sequence names, package names, package options and so on.

Part of the motivation is also that users have asked me how I created the manuals for my packages. Now I can refer to this bundle.

Another reason for the splitting into separate packages is – besides the advantage of easier maintenance – is that I wanted to add programming tools that I often use into \texttt{cnltx-base} which may allow me (and others) to use them for other packages, too, without having to define them each time. So it is quite likely that \texttt{cnltx-base} will get extended in the future.

The bundle provides listings style for \LaTeX{} code, bibliography database files and index style files. It provides a biblatex citation and bibliography style closely linked to \texttt{cnltx-doc}. It provides a bibliography database file containing many \LaTeX{} packages. It provides... Let's stop here. You see that the bundle provides a lot of different features which explains why this manual is more than 60 pages long.

The most detailed documentation for the bundle is as always the source code of the \texttt{sty} and \texttt{cls} files but I'm trying to provide a documentation as comprehensive as possible. Reading the source files may show how things are implemented but the intended use only becomes clear when you read this manual.

The bundle reflects the fact that I haven't started using literate programming, yet. I don't use \texttt{docstrip} and don't write \texttt{dtx} files but always write the \texttt{sty} or \texttt{cls} files directly. I write the manual always at the same time but as a separate file. While I'm entirely aware of the advantages of literate programming I never could bring myself to start to use it myself. As a consequence I have no idea if this bundle can be used for it or not.

Source code formatting is done with the help of the powerful listings package [HM13] by Carsten Heinz and later Brooks Moses, now maintained by Jobst Hoffmann. The only real drawback I have found with it is recognizing starred and un-starred versions of an environment as different keywords. This does not seem to be possible which is why indexing of such environments will lead to wrong page numbers.

The fancy frames of the source code examples are realized with the \texttt{mdframed} package by Marco Daniel [Dan13], loaded with the option \texttt{framemethod = tikz}. 
Besides all this I included some other ideas of mine in this bundle which are all provided by \texttt{CNLTX-TOOLS}. This includes a mechanism for defining clever abbreviations or macros that make it easy to index names the same way \texttt{biblatex} does.

## 2. Bundled Packages, Classes and Files

The \texttt{CNLTX} bundle currently bundles the following packages, classes and files:

- \texttt{CNLTX} – a wrapper package for usage in documents. It loads one or more of the following packages. See section 4 for details on the usage.
  \begin{verbatim}
\usepackage{cnltx}
\end{verbatim}

- \texttt{CNLTX-BASE} – a package that defines base macros for error-messaging, expansion control, tokenlist manipulation and defining of expandable macros. It also provides color definitions and defines a few color schemes for the \texttt{CNLTX-DOC} class. All other packages and classes of the \texttt{CNLTX} bundle load this package. This package can be used stand-alone.
  \begin{verbatim}
\usepackage{cnltx-base}
\end{verbatim}

  The packages commands are not described in the main part of this documentation but only in section A.1, i.e., in the appendix.

- \texttt{CNLTX-DOC} – a class for writing package manuals. Loads \texttt{CNLTX-EXAMPLE} and \texttt{CNLTX-TOOLS} and implicitly all other files of the bundle.
  \begin{verbatim}
\documentclass{cnltx-doc}
\end{verbatim}

- \texttt{CNLTX-EXAMPLE} – a package that defines macros and environments for describing control sequences and options and for including source code. Loads \texttt{CNLTX-LISTINGS}. This package can be used stand-alone.
  \begin{verbatim}
\usepackage{cnltx-example}
\end{verbatim}

- \texttt{CNLTX-LISTINGS} – a package that defines the listings language ‘BibTeX’. Also defines a list of highlighted control sequence names and environment names, loaded by \texttt{CNLTX-EXAMPLE}. The additional control sequence and environment names used to be defined in \texttt{CNLTX-CSNAMES}. That package got removed and its contents are now provided by \texttt{CNLTX-LISTINGS}. This package can be used stand-alone.
  \begin{verbatim}
\usepackage{cnltx-listings}
\end{verbatim}

- \texttt{CNLTX-TOOLS} – a package that defines tools used by \texttt{CNLTX-DOC} that are unrelated to \LaTeX\ documentation \textit{per se}. This package can be used stand-alone.
  \begin{verbatim}
\usepackage{cnltx-tools}
\end{verbatim}

- \texttt{CNLTX-TRANSLATIONS} – a package that provides translations needed by the other modules. It makes no sense to use this package standalone although it’s possible.

- \texttt{cnltx.ist} – an index style file that is used when the option \texttt{add-index} for \texttt{CNLTX-DOC} is activated and the option \texttt{index-style} is not used.
3. License and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the \LaTeX\ Project Public License (\LPPL), version 1.3 or later (http://www.latex-project.org/lppl.txt).

The software has the status “maintained.”

The \texttt{CNLTX-BASE} package loads the following packages: pgfopt Hockey, etoolbox \textsuperscript{3} [Leh11],
\texttt{ltxcmds} \textsuperscript{4} [Obe11b],\texttt{pdftexcmds} \textsuperscript{5} [Obe11c],\texttt{trimspaces} \textsuperscript{6} [Rob09] and \texttt{xcolor} \textsuperscript{7} [Ker07].

The \texttt{CNLTX-DOC} class loads the packages \texttt{CNLTX-BASE}, \texttt{CNLTX-EXAMPLE}, \texttt{CNLTX-TRANSLATIONS}, \texttt{ulem} \textsuperscript{8} [Ars11], \texttt{multicol} \textsuperscript{9} [Mit11], \texttt{ragged2e} \textsuperscript{10} [Sch09], \texttt{marginnote} \textsuperscript{11} [Koh12] and \texttt{hyperref} \textsuperscript{12} [OR12]. It is a wrapper class for the \texttt{KOMA-Script} class \texttt{scratcel} \textsuperscript{13} [KN12]. The class has the option \texttt{load-preamble} which when used will load additional packages, see section 10.5 on page 41 for details.

The \texttt{CNLTX-EXAMPLE} package loads the packages \texttt{CNLTX-BASE}, \texttt{CNLTX-LISTINGS}, \texttt{CNLTX-TOOLS}, \texttt{CNLTX-TRANSLATIONS}, \texttt{mdframed} \textsuperscript{14} [Dan13], \texttt{textcomp} \textsuperscript{15} [Rah95], \texttt{idxcmds} \textsuperscript{16} [Nie13c], \texttt{ifnotex} \textsuperscript{16} [Rob10], \texttt{adjustbox} \textsuperscript{17} [Sch12].

The \texttt{CNLTX-LISTINGS} package loads the packages \texttt{CNLTX-BASE}, \texttt{CNLTX-TRANSLATIONS} and \texttt{accsupp} \textsuperscript{18} [Obe11a].

The \texttt{CNLTX-TOOLS} package loads the packages \texttt{CNLTX-BASE}, \texttt{CNLTX-TRANSLATIONS} and \texttt{accsupp} \textsuperscript{19} [Obe11a].

\texttt{CNLTX-TRANSLATIONS} loads the translations package \textsuperscript{20} [Nie13d].

All other packages that are loaded are loaded by the mentioned packages and not directly by any of the packages or classes of the \texttt{CNLTX} bundle. Like all of my packages \texttt{CNLTX} explicitly relies on an up to date \TeX\ distribution.

\begin{itemize}
  \item \texttt{cnltx.bib} – a bibliography file that contains a small but growing number of package entries, see section D. Used by \texttt{CNLTX-DOC} when the \texttt{add-bib} is used.
  \item \texttt{cnltx.bb}, \texttt{cnltx.cbx} and \texttt{cnltx.dbx} – files related to the biblatex style \texttt{cnltx}. The biblatex style defined in those files is used when the \texttt{add-bib} for \texttt{CNLTX-DOC} is used.
\end{itemize}

\begin{footnotesize}
\begin{enumerate}
  \item on CTAN as \texttt{pgfopt}: http://mirrors.ctan.org/macros/latex/contrib/pgfopt/pgfopt/\texttt{h}
  \item on CTAN as \texttt{etoolbox}: http://mirrors.ctan.org/macros/latex/contrib/etoolbox/\texttt{h}
  \item on CTAN as \texttt{oberdiek}: http://mirrors.ctan.org/macros/latex/contrib/oberdiek/\texttt{h}
  \item on CTAN as \texttt{trimspaces}: http://mirrors.ctan.org/macros/latex/contrib/trimspaces/\texttt{h}
  \item on CTAN as \texttt{xcolor}: http://mirrors.ctan.org/macros/latex/contrib/xcolor/\texttt{h}
  \item on CTAN as \texttt{ulem}: http://mirrors.ctan.org/macros/latex/contrib/ulem/\texttt{h}
  \item on CTAN as \texttt{multicol}: http://mirrors.ctan.org/macros/latex/contrib/multicol/\texttt{h}
  \item on CTAN as \texttt{ragged2e}: http://mirrors.ctan.org/macros/latex/contrib/multicol/\texttt{h}
  \item on CTAN as \texttt{marginnote}: http://mirrors.ctan.org/macros/latex/contrib/marginnote/\texttt{h}
  \item on CTAN as \texttt{hyperref}: http://mirrors.ctan.org/macros/latex/contrib/hyperref/\texttt{h}
  \item on CTAN as \texttt{koma-script}: http://mirrors.ctan.org/macros/latex/contrib/koma-script/\texttt{h}
  \item on CTAN as \texttt{mdframed}: http://mirrors.ctan.org/macros/latex/contrib/mdframed/\texttt{h}
  \item on CTAN as \texttt{textcomp}: http://mirrors.ctan.org/macros/latex/contrib/textcomp/\texttt{h}
  \item on CTAN as \texttt{idxcmds}: http://mirrors.ctan.org/macros/latex/contrib/idxcmds/\texttt{h}
  \item on CTAN as \texttt{ifnotex}: http://mirrors.ctan.org/macros/latex/contrib/ifnotex/\texttt{h}
  \item on CTAN as \texttt{adjustbox}: http://mirrors.ctan.org/macros/latex/contrib/adjustbox/\texttt{h}
  \item on CTAN as \texttt{listings}: http://mirrors.ctan.org/macros/latex/contrib/listings/\texttt{h}
  \item on CTAN as \texttt{catchfile}: http://mirrors.ctan.org/macros/latex/contrib/catchfile/\texttt{h}
\end{enumerate}
\end{footnotesize}
4. Usage of the Bundle

The intended use of this bundle is three-fold:

- The main use-case is documenting my own \LaTeX packages. This is done with

  \documentclass{cnltx-doc}

  and actually loads most if not all of the bundle.

- The module \texttt{CNLTX-BASE} is also intended as a programming tools package that will be used in other packages eventually. For example it is used by the \texttt{cntformats} package from the \texttt{exsheets} bundle [Nie14].

- In case parts of this bundle prove useful to be used in a document the recommended way is to add

  \usepackage{cnltx}

  to the preamble which will load the \texttt{CNLTX-BASE} module. Other needed modules can be given as package option by using the name part after the dash as option.

  \usepackage[example]{cnltx}

  would load \texttt{CNLTX-EXAMPLE}.

- Parts of the bundle – especially \texttt{CNLTX-BASE} – may prove useful in other packages. The loading the packages directly as indicated in section 3 seems the best way. After loading \texttt{CNLTX-BASE} the other modules can also be loaded with \texttt{\cnltx@load@module}, see section A.1.1 for details.
Part II.
Details of Available Commands, Environments and Options

5. Options and Setup

The \texttt{c nltk} bundle has a large number of options. The \texttt{c nltk-doc} class only knows a few options (described in section 10.1 on page 38) as class options, though. All other options regardless if they’re defined by a package or a class can and should be set with the setup command:

\texttt{\setc nltk\{options\}}

Setup command for the \texttt{c nltk} bundle. This command is provided by \texttt{c nltk-base}.

The source code environments defined by the \texttt{c nltk-example} package also have optional arguments that can be used to set the options for the environment locally.

6. Available Commands

6.1. Description of Macros, Environments and Options

The commands described in this section all are provided by the \texttt{c nltk-example} package. They all are related to the typesetting of provided macros, options and the like.

\texttt{\code\{arg\}}

Formatting of source code. This is no verbatim command. Used internally in the following commands.

\texttt{\verbatimcode\{char\}\{code\}\{char\}}

A verbatim command that uses the same formatting as the source code example environments, \textit{cf.} section 8.4. This is a wrapper for \texttt{\lstinline} which loads the corresponding style.

\texttt{\cs\{name\}}

Format the control sequence \texttt{\langle name\rangle}, \cs\{name\}: \texttt{name}. Adds a corresponding index entry. The starred form does not add an index entry.

\texttt{\csidx\{name\}}

Adds an index entry but does not typeset the control sequence \texttt{\langle name\rangle}.

\texttt{\env\{name\}}

Format the environment \texttt{\langle name\rangle}, \env\{name\}: name. Adds a corresponding index entry with a hint that the entry refers to an environment. The starred form does not add an index entry.
6. Available Commands

\envidx{(name)}
Adds an index entry but does not typeset the environment (name).

\meta{(meta)}
Description of an argument, \meta{meta}: (meta).

\marg{(arg)}
A mandatory argument. (arg) is formatted with \meta if it is not blank, \marg{arg}: \{\{arg\}\}.

\Marg{(arg)}
A mandatory argument. (arg) is formatted with \code if it is not blank, \Marg{arg}: {\{arg\}}.

\oarg{(arg)}
An optional argument. (arg) is formatted with \meta if it is not blank, \oarg{arg}: \{\{arg\}\}.

\Oarg{(arg)}
An optional argument. (arg) is formatted with \code if it is not blank, \Oarg{arg}: \{\{arg\}\}.

\darg{(arg)}
An argument with parentheses as delimiters. (arg) is formatted with \meta if it is not blank, \darg{arg}: \{\{arg\}\}.

\Darg{(arg)}
An argument with parentheses as delimiters. (arg) is formatted with \code if it is not blank, \Darg{arg}: \{\{arg\}\}.

\sarg
An optional star argument, \sarg: *.

\newarg{\{arg formatting\}}{\}{}{\{left delim\}}{\{right delim\}}
Command used to define the argument commands: \newarg\marg{\{\}\{\}}. The optional argument determines how the argument of the new command will be formatted. This is done with \meta per default. \Marg is defined \newarg{\code}\Marg{\{\}\{\}}.

\option*{(name)}
An option (name), \option{name}: name. Adds a corresponding index entry. The starred form does not add an index entry.

\optionidx{(name)}
Adds an index entry but does not typeset the option (name).

\module*{(name)}
A module (name), \module{name}: name. Adds a corresponding index entry. The starred form does not add an index entry. In some of my packages I like to organize options by grouping them in different classes that I call “modules”. This command refers to those modules.

\moduleidx*{(name)}
Adds an index entry but does not typeset the option (name).
6. Available Commands

\key*{(name)}{(value)}
A key \langle name \rangle with value \langle value \rangle, the optional star prevents an index entry, the optional - strips the braces around \langle value \rangle; \key(key){value}: key = {value}; \key{-key}{value}: key = {value}

\keyis*{(name)}{(value)}
A key \langle name \rangle set to value \langle value \rangle, the optional star prevents an index entry, the optional - strips the braces around value; \key(keyis){value}: key = {value}.

\choices{(clist of choices)}
A list of choices, \choices{one,two,three}: one|two|three

\choicekey{(name)}{(clist of choices)}
A key \langle name \rangle with a list of possible values, \choicekey(key){one,two,three}: key = one|two|three

\boolkey{(name)}
A boolean key \langle name \rangle with choices true and false, \boolkey(key): key = true|false

\default{(value)}
Markup for a default choice, \choices{one,\default{two},three}: one|two|three

6.2. Versioning Commands, Licensing and Related Stuff

The commands described in this section are provided by the \texttt{CNLTX} class except where indicated differently. These commands are related to information about the legal stuff of a package and where to find it on the world wide web.

\sinceversion{(version)}
Gives a sidenote like the one on the left.

\changedversion{(version)}
Gives a sidenote like the one on the left.

\newnote*{(cs)}{(num)}{(optional)}{(definition)}
Defines a note like \sinceversion. The syntax of the command is the same as the one of \texttt{newcommand}. \sinceversion was defined as follows:

\newnote*\sinceversion[1]{}{Introduced in version=\#1}
or actually like this:

\newnote*\sinceversion[1]{\GetTranslation{cnltx-introduced}=\#1}

\newpackagename{(cs)}{(name)}
Define a command \langle cs \rangle that prints \langle name \rangle formatted like \texttt{CNLTX}, \textit{i.e.} in small caps and colored with the color \texttt{cnltx} (see section 13.2).

\lppl
Typesets “LPPL” and adds a corresponding index entry.
6. Available Commands

\LPPL
Typesets “\LaTeX{} Project Public License” and adds the same index entry as \lppl.

\license*{(maintenance status)}
Default: maintained
Typesets “Permission is granted to copy, distribute and/or modify this software under the terms of the \LaTeX{} Project Public License (LPPL), version 1.3 or later (http://www.latex-project.org/lppl.txt). The software has the status “maintained.”. The un-starred variant adds a \par.

\ctan
Typesets “\textsc{CTAN}” and adds a corresponding index entry.

\CTAN
Typesets “Comprehensive \TeX{} Archive Network” and adds the same index entry as \ctan.

\pkg*{(package)}
Provided by \textsc{CTAN}. Format the package name \texttt{(package)} and add an index entry. The starred variant adds nothing to the index.

\pkgidx{(package)}
Provided by \textsc{CTAN}. Add an index entry for the package \texttt{(package)}.

\cls*{(class)}
Provided by \textsc{CTAN}. Format the class name \texttt{(class)} and add an index entry. The starred variant adds nothing to the index.

\clsidx{(class)}
Provided by \textsc{CTAN}. Add an index entry for the class \texttt{(class)}.

\CTANurl{(directory)}{(name)}
Writes a \textsc{CTAN} link like the ones in section 3 on page 5 in the footnotes. The predefined directory is \texttt{macros/latex/contrib}. The link address will be:
http://mirrors.ctan.org/\texttt{(directory)}/\texttt{(name)}/.

\email{(email address)}
Introduced in version 0.11. A wrapper for \href{mailto:#1}{#1}.

\website{(web address)}
Introduced in version 0.11. A wrapper for \href{http://#1}{#1}.

\securewebsite{(web address)}
Introduced in version 0.11. A wrapper for \href{https://#1}{#1}.

\needpackage{(directory)}{(name)}
Introduced in version 0.11. A wrapper for \pkg{\#1}\footnote{\CTANurl{\#1}{\#2}}

\needclass{(directory)}{(name)}
Introduced in version 0.2. A wrapper for \cls{\#1}\footnote{\CTANurl{\#1}{\#2}}
7. Available Environments

6.3. Input Source Code Files

Similar to the environments described in section 7.2 on the next page \texttt{Cnltx-Example} provides a few commands for inputting source code files, formatting and printing the source code and inputting the file directly.

\texttt{\inputexample\{\langle options\rangle\}\{\langle file name\rangle\}}

The equivalent of the \texttt{example} environment, see section 7.2 on the following page.

\texttt{\inputsidebyside\{\langle options\rangle\}\{\langle file name\rangle\}}

The equivalent of the \texttt{sidebyside} environment, see section 7.2 on the next page.

\texttt{\inputsourcecode\{\langle options\rangle\}\{\langle file name\rangle\}}

The equivalent of the \texttt{sourcecode} environment, see section 7.2 on the following page.

\texttt{\implementation\{\langle options\rangle\}\{\langle file name\rangle\}}

A wrapper for \texttt{\lstinputlisting[style=cnltx,#1]{#2}}

It is possible to define further commands like this:

\texttt{\newinputsourcefilecmd\{\langle option\rangle\}\{\langle control sequence\rangle\}}

Defines \texttt{\langle control sequence\rangle} as a new source code input command where \texttt{\langle options\rangle} are preset.

The existing commands have been defined like this:

\texttt{\newinputsourcefilecmd\inputexample}
\texttt{\newinputsourcefilecmd\{side-by-side\}\inputsidebyside}
\texttt{\newinputsourcefilecmd\{code-only\}\inputsourcecode}

7. Available Environments

7.1. Description Environments

\texttt{Cnltx-Doc} defines some description environments used to describe macros, environments or options.
Available Environments

\begin{commands}
A description-like environment for describing commands. While this environment is a list internally and thus recognizes \item own commands are used to describe macros. They are explained in section 8.1 on the next page.
\end{commands}

\begin{options}
A description-like environment for describing options. While this environment is a list internally and thus recognizes \item own commands are used to describe options. They are explained in section 8.2 on page 15.
\end{options}

\begin{environments}
A description-like environment for describing environments. While this environment is a list internally and thus recognizes \item own commands are used to describe environments. They are explained in section 8.3 on page 17.

These environments are lists all using the same internal \list. The setup of this list can be changed via an option:

\begin{verbatim}
list-setup = \{\{definitions\}\}
\end{verbatim}

Default: \leftmargin=0pt \labelwidth=2em \labelsep=0pt \itemindent=-1em

The setup of the \list used by the commands, options and environments environments.

7.2. Source Code Environments

\begin{example}\{\{options\}\}
This environment is a formatted verbatim environment that also inputs the output of the inputted code. This environment is described in section 8.4 on page 18.
\end{example}

\begin{sidebyside}\{\{options\}\}
This environment is a formatted verbatim environment that also inputs the output of the inputted code. Source and output are printed side-by-side. This environment is described in section 8.4 on page 18.
\end{sidebyside}

\begin{sourcecode}\{\{options\}\}
This environment is a formatted verbatim environment. This environment is described in section 8.4 on page 18.
\end{sourcecode}

In each of these environments certain hooks are provided that can be used to add definitions you like:

\begin{verbatim}
pre-code = \{\{definitions\}\}
\end{verbatim}
\begin{verbatim}
\{definitions\} are placed before the source code is inserted.
\end{verbatim}

\begin{verbatim}
after-code = \{\{definitions\}\}
\end{verbatim}
\begin{verbatim}
\{definitions\} are placed after the source code is inserted.
\end{verbatim}
8. Usage of the Various Functions

Pre-output = \{\langle definitions\rangle\}
\langle definitions\rangle are placed before the output of the source code is inserted.

After-output = \{\langle definitions\rangle\}
\langle definitions\rangle are placed after the output of the source code is inserted.

It is possible to define further environments like this:

\newsourcecodeenv\{\langle option\rangle\}\{\langle name\rangle\}

Defines \langle name\rangle as a new source code environment where \langle options\rangle are preset.

The existing environments have been defined like this:

\begin{verbatim}
  \newsourcecodeenv{example}
  \newsourcecodeenv[side-by-side]{sidebyside}
  \newsourcecodeenv[code-only]{sourcecode}
\end{verbatim}

8. Usage of the Various Functions

8.1. Command Descriptions

Inside of the environment commands that was introduced in section 7.1 on page 11 items are input via the following command:

\command*\{\langle name\rangle\}\{\langle stuff after\rangle\}

This macro formats a control sequence with \cs and puts a line break after it. The optional argument allows printing things directly after the command name and can thus be used for adding arguments. The star prevents the creation of an index entry.

\Default*!\{\langle code\rangle\}

This command can be placed after \command or \opt in order to give a default definition of a macro or a default value of an option. The definition will then be placed on the same line flush right. The star prevents the insertion of \newline after it. The optional bang adds the information that an option is mandatory, i.e. has to be set.

\expandable

Adds the symbol ∗ to the left of a command in the margin to indicate that the command is expandable. This command should be used immediately before \command.

\unexpandable

Adds the symbol ∗ to the left of a command in the margin to indicate that the command is not expandable. This command should be used immediately before \command.

\expandablesign

The macro that holds the sign used by \expandable and \unexpandable.
8. Usage of the Various Functions

\expandablesymbol

The symbol ∗, i.e., \expandablesign formatted with the color \expandable.

\unexpandablesymbol

The symbol ∗, i.e., \expandablesign formatted with the color \unexpandable.

\begin{commands}
\begin{verbatim}
\cs
This is about foo bar baz.
\cs{⟨arg⟩}
This one has an argument.
\cs*{⟨option⟩}
This has a star variant and an optional argument.
\cs
This one has the default replacement text \code{foo bar}
\expandable\cs
This macro is expandable.
\end{verbatim}
\end{commands}

\cs
This is about foo bar baz.
\cs{⟨arg⟩}
This one has an argument.
\cs*{⟨option⟩}
This has a star variant and an optional argument.

\cs
This one has the default replacement text \code{foo bar}
\cs
This macro is expandable.

The \expandablesign can of course be redefined to something else you like better. For the sake of completeness there is an option that does exactly this:

\begin{verbatim}
expandable-sign = ⟨⟨definition⟩⟩
\end{verbatim}

Redefines \expandablesign to ⟨⟨definition⟩⟩.
8. Usage of the Various Functions

8.2. Option Descriptions

The options environment knows a few more commands to meet all the different kinds of
options.

\opt*
An option. The star prevents an index entry.

\keyval*{⟨key⟩}{⟨value⟩}
A key/value option. The optional star prevents an index entry. The optional - strips the braces
around ⟨value⟩, see the example below.

\keychoice*{⟨key⟩}{⟨list of choices⟩}
A key/value option where the value is one of a list of choices. The star prevents an index entry.

\keybool*{⟨name⟩}
A boolean key, that is a choice key with choices true and false. The star prevents an index
entry.

\Default*!{⟨code⟩}
This command can be placed after \command or \opt (or any of the other commands for adding
an option to the options list) in order to give a default definition of a macro or a default value
of an option. The definition will then be placed on the same line flush right. The star prevents
the insertion of \newline after it. The optional bang adds the information that an option is
mandatory, i.e., it has to be set.

\Module*!{⟨name⟩}
This command can be placed after \option but before \Default in order to determine the
module the option belongs to. It will be written in the left margin next to the option name. The
star prevents the insertion of \newline after it. The optional bang adds an index entry for the
module. This is somehow inconsistent with many of the other commands where an optional
star prevents an index entry but it fits to the functionality of \Default which is why this syntax
was chosen.

The following demonstrates how the commands would be used to create option descriptions:

\begin{options}
  \opt{foo}
  This makes stuff. Let's add a few more words so that the line gets
  filled and we can see how the output actually looks.
  \opt{foo}\Default{bar}
  This makes stuff. Let's add a few more words so that the line gets
  filled and we can see how the output actually looks.
  \opt{foo}\Module{bar}
  This option belongs to \module*{bar}. Let's add a few more words so
  that the line gets filled and we can see how the output actually
  looks.
\end{options}
The code above gives the following output:

**foo**
This makes stuff. Let's add a few more words so that the line gets filled and we can see how the output actually looks.

**foo**
Default: **bar**
This makes stuff. Let's add a few more words so that the line gets filled and we can see how the output actually looks.

**bar** ➔ **foo**
This option belongs to the module **bar**. Let's add a few more words so that the line gets filled and we can see how the output actually looks.

**bar** ➔ **foo**
Default: **baz**
This option belongs to the module **bar**. Let's add a few more words so that the line gets filled and we can see how the output actually looks.

**foo** = **{{bar}}**
(initially empty)
This makes stuff. Let's add a few more words so that the line gets filled and we can see how the output actually looks.
8. Usage of the Various Functions

\texttt{foo = \{\langle bar \rangle \}}
This makes stuff. Let’s add a few more words so that the line gets filled and we can see how the output actually looks.

\texttt{foo = \{\langle bar \rangle \}}
This makes stuff. Let’s add a few more words so that the line gets filled and we can see how the output actually looks.

\texttt{foo = \langle bar \rangle}
This makes stuff. Let’s add a few more words so that the line gets filled and we can see how the output actually looks.

\texttt{foo = one|two|three}
This makes stuff. Let’s add a few more words so that the line gets filled and we can see how the output actually looks.

\texttt{foo = true|false}
This makes stuff. Let’s add a few more words so that the line gets filled and we can see how the output actually looks.

8.3. Environment Descriptions

Environment descriptions are made—unsurprisingly—with the \texttt{environments} environment. It knows the command \texttt{environment}:

\texttt{\begin{environments}\environment*{\langle name \rangle}\[\oarg{options}\]}
This is environment \texttt{\env*{foobar}}. The star prevents it from being added to the index.

\texttt{\end{environments}}

\begin{foobar}\[\langle options \rangle\]
This is environment \texttt{foobar}. The star prevents it from being added to the index.

\begin{foobar}\[\langle options \rangle\]
This is environment \texttt{foobar}. The star prevents it from being added to the index.
8. Usage of the Various Functions

8.4. Code Examples

Code examples can be included through the example environment or the sourcecode environment. The sourcecode only shows the piece of \LaTeX\ code while the example environment also shows the output of the \LaTeX\ code.

\begin{example}
\begin{verbatim}
a \LaTeX\ code example
\end{verbatim}
\end{example}

This example would give:

\begin{verbatim}
a \LaTeX\ code example
\end{verbatim}

a \LaTeX\ code example

Both environments can be influenced by options:

- **code-only = true|false**  Default: false
  Only typeset the code as code but don’t include it afterwards. The code box above is an example for the usage of this option. This option has no effect on the sourcecode environment: it is already set for this environment.

- **side-by-side = true|false**  Default: false
  Typeset source and output side by side. The code is input on the left and the output on the right. Side by side examples are typeset in \texttt{minipage} environments with all consequences that come with them (think of \texttt{parindent}, page breaks...). Since a \texttt{minipage} cannot be broken across pages the surrounding \texttt{mdframed} frame gets the option \texttt{nobreak = true}. This option has no effect on the sourcecode environment.

- **code-left = true|false**  Default: true
  If \texttt{true} and the option \texttt{side-by-side} is chosen the source code is printed on the right side else on the left. This option has no effect on the sourcecode environment.

- **code-sep = \{⟨definition⟩\}**  Default: \texttt{\hrulefill}
  Code that is inserted between a source code and the corresponding output when printed below each other. This option has no effect on the sourcecode environment.

- **outside = true|false**  Default: false
  If \texttt{true} the output of an example is put outside of the frame in the input stream. This can be useful if the example code contains a floating environment for example.
8. Usage of the Various Functions

The same example again, this time using `side-by-side` (which is the same as using the `sidebyside` environment):

```latex
\begin{sidebyside}
\begin{minipage}{0.4\textwidth}
\LaTeX\ code example
\end{minipage}
\begin{minipage}{0.4\textwidth}
\TeX\ code example
\end{minipage}
\end{sidebyside}
```

`side-by-side` and `code-left = false`:

```latex
\begin{sidebyside}
\begin{minipage}{0.4\textwidth}
\TeX\ code example
\end{minipage}
\begin{minipage}{0.4\textwidth}
\LaTeX\ code example
\end{minipage}
\end{sidebyside}
```

The frame around the examples is done by the `mdframed` package [Dan13]. It is of course possible to customize it:

- `add-frame-options = {⟨mdframed options⟩}` (initially empty)
  
  Add options to the predefined settings.

- `frame-options = {⟨mdframed options⟩}
  
  Default: backgroundcolor=cnltxbg, linecolor=cnltx, roundcorner=5pt

- `add-local-frame = {⟨mdframed options⟩}`
  
  Introduced in version 0.10
  
  Add mdframed options to the environment where the option is used only. This is basically \begin{mdframed}[style=cnltx,⟨options⟩].

- `local-frame = {⟨mdframed options⟩}`
  
  Introduced in version 0.10
  
  Replace the default mdframed options to the environment where the option is used only. This is basically \begin{mdframed}[⟨options⟩].

The source code is formatted using the great listings package [HM13] by Carsten Heinz, Brooks Moses, and Jobst Hoffmann. Similar options exist to adapt listings’ options that are used for formatting the source code. The predefined style has many options that will not be mentioned here. If you’re interested you can find them in cnltx-example.sty or in section 11.2.1 on page 46.

- `gobble = ⟨integer⟩`
  
  Default: 2
  
  The number of initial characters that is gobbled from each line.

- `add-cmds = {⟨list of csnames⟩}` (initially empty)
  
  A list of control sequence names that should be recognized as a command sequence in the source code examples and should be formatted accordingly. The control sequence names in this list will also get an index entry when they’re used in the source example. This is done internally via \csidx. The option should be used to add the new commands that are defined by the package for which you are writing the manual for.
### 8. Usage of the Various Functions

**add-silent-cmds** = \{\{list of csnames\}\}

A list of control sequence names that should be recognized as a command sequence in the source code examples and should be formatted accordingly. The control sequence names in this list will *not* get an index entry when they're used in the source example. There already is quite a large but far from comprehensive list of silent commands but many are still missing. This option allows you to extend the list on a per document basis.

**add-listings-options** = \{\{listings options\}\} (initially empty)

Additional options for the listings [HM13] environments. *This redefines the cmltx listings style which will affect all sourcecode environments!*

**listings-options** = \{\{listings options\}\}

Overwrite existing options with new ones. This can be used to build an own style from scratch. *This redefines the cmltx listings style which will affect all sourcecode environments!*

**add-sourcecode-options** = \{\{listings options\}\}

These options are added to the listings options of the source code environments without redefining the main style. Hence it can be used to locally add options to a source code environment. This is basically `\lstset{style=cmlltx,(options)}`.

**sourcecode-options** = \{\{listings options\}\}

These options are added to the listings options of the source code environments without redefining or using the main style. Hence it can be used to locally add options to a source code environment. This is basically `\lstset{(options)}`.

**add-envs** = \{\{list of environment names\}\} (initially empty)

Like `add-cmds` but for environment names.

**add-silent-envs** = \{\{list of environment names\}\}

Like `add-silent-cmds` but for environment names.

### 8.5. Compile Source Examples

#### 8.5.1. The Compilation Process

When you input an example like

```latex
\begin{example}
\documentclass{article}
\begin{document}
foo
\end{document}
\end{example}
```

---

20
you’ll get an error since the code is input as is and you’ll end up with \documentclass after \begin{document}. There’s a way out, though.

\textbf{\LaTeX-Example} provides the possibility to compile the source code file externally and input the compiled PDF.

This needs shell-escape enabled. The default compilation program is \texttt{pdflatex} which will compile the file two times. The process can be customized with the following options:

\begin{itemize}
\item \texttt{compile = true|false} \hspace{1cm} Default: false
  
  Compile the source code file. Although this option can be set globally it really shouldn’t be! It’s best to give this option explicitly to the source code environment whose body should be compiled. If enabled globally all examples would be compiled and most likely lead to various errors since most examples won’t be complete \LaTeX documents.

\item \texttt{program = pdflatex|luatex|xelatex|arara} \hspace{1cm} Default: \texttt{pdflatex}
  
  The program to compile the source file.

\item \texttt{runs = \{\langle number\rangle\}} \hspace{1cm} Default: 2
  
  The number of compilations.

\item \texttt{exe-with = \{\langle options\rangle\}} \hspace{1cm} (initially empty)
  
  Command line options that can be given to the compilation program chosen with \texttt{program}.

\item \texttt{file-ext = \{\langle extension\rangle\}} \hspace{1cm} Default: pdf
  
  The file extension of the included file of a compiled example.

\item \texttt{add-frame = true|false} \hspace{1cm} Default: true
  
  If true every output page will get a frame.
\end{itemize}

The compiled document will be input with \texttt{\includegraphics}, each page separately. Since the pages of the document are most likely as large as the ones from the main document itself they are scaled down. This is best demonstrated with an example. The following input
8. Usage of the Various Functions

\begin{document}
\maketitle
\section{A Section Title}
\tableofcontents
\section{A Section Title}
\lipsum[1-10]
\end{document}
8. Usage of the Various Functions

The pages get scaled according to two parameters:

\texttt{max-pages = \{\langle number\rangle\}}

Default: 4

The maximum number of pages in a row. The width of the pages is scaled to $\texttt{\textwidth}/n$ where $n$ is either the number of pages $p$ of the compiled document or $\langle\text{number}\rangle$ if $p > \langle\text{number}\rangle$.

\texttt{max-height = \{\langle dimension\rangle\}}

Default: $0.5\texttt{\texttheight}$

The maximum height of a page.

There's another possibility to influence the appearance of the output:

\texttt{graphics = \{\langle options\rangle\}}

(initially empty)

(\langle options\rangle) are passed to \texttt{\includegraphics} for every page that is input.

8.5.2. Floating Output

Since the output can become a quite large figure it might be preferable to have it as a floating figure. This is also possible by using the option \texttt{float}.

\texttt{float = \texttt{true}|\texttt{false}|\{float parameters\}}

Default: false

Choose if the output should be placed in a figure of its own. You can also use this option to specify the floating parameters for the float.

\texttt{float-pos = \{\langle float parameters\rangle\}}

Default: \texttt{tbp}

Set the standard floating parameters that are used if \texttt{float = true}. The default is actually the expansion of \texttt{\fps@figure} and not directly \texttt{tbp}.

\texttt{float-env = \{\langle name\rangle\}}

Default: \texttt{figure}

The floating environment used when the option \texttt{float} is used.

\texttt{caption = \{\langle text\rangle\}}

(initially empty)

(\langle text\rangle) will be used as caption. If left blank no caption will be typeset. If you want to add a \texttt{\label} you can use it in this option. Implicitly sets \texttt{float = true}.

Please note that \texttt{float} only has an effect if \texttt{compile = true} has been set.
8. Usage of the Various Functions

8.5.3. Selective Output

Sometimes it may be preferable not to include all pages of a compiled document but only specific pages. This is possible with the following option.

\texttt{pages = \{\langle specifications\rangle\}}

Select the included pages. \textit{Specification} is a comma-separated list of page numbers and page ranges, e.g., \texttt{1,3,4} or \texttt{1,3-5}. \texttt{1,3-5} is the same as \texttt{1,3,4,5}. If the list includes page numbers larger than the maximum number of pages the PDF has a warning message will be issued and a replacement text will occur in the output where the page would have been.

The input

\begin{verbatim}
\begin{example}\[compile,pages=1\]
\documentclass[a5paper]{scrartcl}
\usepackage{showframe,lipsum}
\author{Clemens Niederberger}
\title{A Test File}
\begin{document}
\maketitle
\tableofcontents
\section{A Section Title}
\lipsum[1-10]
\end{document}
\end{example}
\end{verbatim}

will lead to this output:

\begin{verbatim}
\documentclass[a5paper]{scrartcl}
\usepackage{showframe,lipsum}
\author{Clemens Niederberger}
\title{A Test File}
\begin{document}
\maketitle
\tableofcontents
\section{A Section Title}
\lipsum[1-10]
\end{document}
\end{verbatim}
8. Usage of the Various Functions

Together with the `graphics` option this can be used to output a part of a page. The following source

```latex
\begin{example}[compile,pages=1,graphics={trim={0pt 12cm 0pt 0pt},clip}]
\begin{document}
\maketitle
\tableofcontents
\section{A Section Title}
\lipsum[1-10]
\end{document}
\end{example}
```

will give this output:
8. Usage of the Various Functions

Let’s say you’re documenting a package called `mypackage` that provides the command `\mycommand` and the environment `myenv`. The basic manual setup could then look something like this:
8. Usage of the Various Functions

8.7. Additional Functionality Provided by \texttt{cnltx-base}

The \texttt{cnltx-base} package’s main purpose is to provide programming facilities. Most of its macros are listed in section 8.1. However, I like to explain some of its features in a bit more detail.

8.7.1. Looking for Trailing Punctuation

The command \texttt{\textbackslash cnltx@ifpunctuation} is a conditional that detects if a punctuation mark follows and acts depending on it. What counts as a punctuation mark can be set by the user.

\texttt{\textbackslash cnltx@ifpunctuation*\{⟨punctuation marks⟩\}\{⟨true⟩\}\{⟨false⟩\}\{trailing punctuation\}}

The starred version does not gobble the trailing punctuation while the unstarred does. That’s why in the unstarred version you can also use \texttt{\textbackslash cnltx@trailpunct} to access the gobbled punctuation mark. The optional argument sets the punctuation marks that should be considered for this use only.

\texttt{set-trail-punct = \{⟨punctuation marks⟩\} \quad \text{Default: ,.!?;:}}

Sets the default list of punctuation marks that should be checked if the optional argument of \texttt{\textbackslash cnltx@ifpunctuation} is not used.

The usage is probably self-explaining:

\begin{verbatim}
1 \makeatletter
2 \begin{document}
3 ... 
4 \end{document}
5 \set-trail-punct = \{.,!?,;:\}
6 \begin{document}
7 \texttt{a punctuation mark, \textbackslash cnltx@ifpunctuation\{\textbackslash cnltx@trailpunct\}\{\textbackslash test\}\{\textbackslash test\}}\texttt{!}\texttt{\par}
8 \texttt{a punctuation mark \textbackslash cnltx@ifpunctuation\{-\}\{\textbackslash test\}\{\textbackslash test\}}\texttt{!}\texttt{\par}
9 \texttt{a full stop, \textbackslash cnltx@ifpunctuation\{-\}\{\textbackslash test\}\{\textbackslash test\}}\texttt{!}\texttt{\par}
10 \end{document}
\end{verbatim}

(test!)
(test!)

\begin{verbatim}
(a punctuation mark follows!)
(a full stop doesn’t follow!)
\end{verbatim}
If the non-starred variant has gobbed a `\par` the `\par` is placed back:

\begin{quote}
\begin{alltt}
\makeatletter
\def\test{|(\texttt{\cnltx@ifpunctuation{(test}\cnltx@trailpunct)}){|(test)}%}
\makeatother
\test
\test.
\test{}
\end{alltt}
\end{quote}

\begin{quote}
\begin{tabular}{p{\textwidth}}
(test)
(test.)
(test).
\end{tabular}
\end{quote}

### 8.7.2. Counter Representation Commands

#### Background

A counter representation command like `\arabic{section}` always is a command that calls an associated internal command (`\@arabic` in the case of our example) that acts on the count associated with the counter:

\begin{quote}
\begin{alltt}
\def\arabic#1{|\expandafter\@arabic\csname c@#1\endcsname}
\def\@arabic#1{|\number #1}
\end{alltt}
\end{quote}

The command `\arabic{⟨counter⟩}` builds a command sequence `\c@⟨counter⟩` from its argument ⟨counter⟩. It then calls the internal command `\@arabic` that takes this command sequence as an argument. The command sequence `\c@⟨counter⟩` is the count (in the \TeX{} sense) that is associated with the counter ⟨counter⟩, \emph{i.e.}, it holds the actual number. The command `\arabic` now simply typesets the integer value of the count.

The same holds for every counter representation command. The principle always is as follows:

\begin{quote}
\begin{alltt}
\def\arabic#1{|\expandafter\@arabic\csname c@#1\endcsname}
\def\@arabic#1{|\number #1}
\end{alltt}
\end{quote}

\begin{quote}
\begin{alltt}
\def\foo#1{|\expandafter\@foo\csname c@#1\endcsname}
\def\@foo#1{|do something with #1 (where #1 is a count)}
\end{alltt}
\end{quote}

This means in order to get a new counter representation command you actually need to define \textit{two} macros.
8. Usage of the Various Functions

\textbf{CNLTX-BASE} defines an interface that allows to define both commands at once without having to think about \texttt{\expandafter}, associated counts, internal command names and so on. The only thing left to do is choosing a name for the counter representation and providing a valid definition of what should happen with the (integer) value of the counter.

\textbf{New Commands}

\texttt{\DeclareCounterRepresentation}\{(\texttt{command})\}{\{\texttt{definition}\}}

Declares a new counter representation command and its internal equivalent. In the \texttt{\texttt{definition}} #1 is used to refer to the counter \texttt{number}, that is, the value of \texttt{\c@\texttt{counter}}. This command will silently overwrite any existing definition.

\texttt{\newcounterrepresentation}\{(\texttt{command})\}{\{\texttt{definition}\}}

Defines a new counter representation command and its internal equivalent. In the \texttt{\texttt{definition}} #1 is used to refer to the counter \texttt{number}, that is, the value of \texttt{\c@\texttt{counter}}. This command will issue an error if either the user command or the internal command (cf. \texttt{\arabic} and \texttt{\@arabic}) already exist.

\texttt{\providecounterrepresentation}\{(\texttt{command})\}{\{\texttt{definition}\}}

Provides a new counter representation command and its internal equivalent. In the \texttt{\texttt{definition}} #1 is used to refer to the counter \texttt{number}, that is, the value of \texttt{\c@\texttt{counter}}. This command will define the commands only if neither the user command nor the internal command (cf. \texttt{\arabic} and \texttt{\@arabic}) already exist and will do nothing if either of them exist.

\texttt{\renewcounterrepresentation}\{(\texttt{command})\}{\{\texttt{definition}\}}

Redefines an existing counter representation command and its internal equivalent. In the \texttt{\texttt{definition}} #1 is used to refer to the counter \texttt{number}, that is, the value of \texttt{\c@\texttt{counter}}. This command will issue an error if neither the user command nor the internal command (cf. \texttt{\arabic} and \texttt{\@arabic}) already exist.

Let’s take a look at what is actually defined by these commands:

\begin{verbatim}
\makeatletter\ttfamily
before:\par
\meaning\arabic\par
\meaning\@arabic
\meaning\@\arabic
\renewcounterrepresentation\arabic{\the\numexpr#1\relax}%
\par
\meaning\arabic\par
\meaning\@arabic

before:
macro:#1->\expandafter \@arabic \csname c@\#1\endcsname
macro:#1->\number #1
\end{verbatim}
As you can see nothing bad happens. The commands are only a convenient interface. Let’s take a look at some more realistic examples. The above redefinition was only a demonstration. For example you may want to have a representation which calculates the displayed value from the counter value?

\begin{verbatim}
\newcounterrepresentation\minusone{\the\numexpr#1-1\relax}\
\newcounterrepresentation\multoffourrm{\romannumeral\numexpr(4*#1)-4\relax}\
\newrobustcmd*{\circlenumber}[1]{\tikz[baseline]\node[anchor=base,draw,shape=circle]{\number#1};}\
\newcounterrepresentation\twodigits{\two@digits{#1}}\
\makeatletter\
\newcounter{test}\
\setcounter{test}{9}\
\minusone{test}\par\
\multoffourrm{test}\par\
\circlenumber{test}\par\
\twodigits{test}
\end{verbatim}

\begin{verbatim}
8
xxxii
9
\end{verbatim}

8.7.3. Expandable Document Commands

The commands presented in this section are highly experimental. Use them only if you really have to!

\texttt{\newexpandablecmd*{⟨cs⟩}{⟨num args⟩}{⟨default opt⟩}{⟨definition⟩}}

This command has the same syntax as \texttt{\newcommand}. The difference is that ⟨cs⟩ is defined with an optional argument it is still fully expandable. This comes with a cost: in order to still being able to check for the optional argument it needs to see a following token as argument. If it is used without optional argument and has no mandatory arguments it may be necessary to add a trailing \texttt{\empty} or something. There’s another drawback: a command \texttt{\test} thus
defined cannot distinguish between \text{test}[] and \text{test}{[]} and will misinterpret the second as a present optional argument.

My recommendation is to never use this for defining a user command. Use it in code you can control and only if you have to.

If you define a command without optional argument this command falls back to \newcommand.

\renewexpandablecmd*{⟨cs⟩}{⟨num args⟩}{⟨default opt⟩}{⟨definition⟩}

The equivalent of \renewcommand. See description of \newexpandablecmd for further details.

\provideexpandablecmd*{⟨cs⟩}{⟨num args⟩}{⟨default opt⟩}{⟨definition⟩}

The equivalent of \providecommand. See description of \newexpandablecmd for further details.

8. Usage of the Various Functions

8.8. Additional Functionality Provided by \texttt{CNLTX-TOOLS}


The \texttt{CNLTX-TOOLS} package defines some additional macros which provide useful functionality also in contexts not documenting a \LaTeX{} package.

\newname{⟨cs⟩}{⟨first name⟩}{⟨last name⟩}

Defines ⟨cs⟩ to write out the full name and add an index entry sorted by the last name. Also defines a starred variant of ⟨cs⟩ that only writes the last name but still adds the full index entry.

\name*{⟨first name⟩}{⟨last name⟩}

Typesets a name according to the same specs as the names defined with \newname. Also adds the name to the index. The starred version only writes the name but doesn’t add the name to the index. Index entries either have the form ⟨last name⟩ or ⟨last name⟩, ⟨first name⟩ depending on the usage of the optional argument. It’s safer to define a dedicated macro with \newname to get consistent index entries.

\cnltxacronym{⟨pdf and sort string⟩}{⟨acronym⟩}

Typesets ⟨acronym⟩ with small caps and uses ⟨pdf and sort string⟩ as PDF string and for sorting the index entry that is added. This command was used to define \ppl and \ctan. \textit{This is not intended as a replacement for packages like acro [Nie13a] or glossaries [Tal13]}! In fact it is a “poor man’s” solution that allows me not to require one of those packages.

\newabbr*{⟨control sequence⟩}{⟨definition⟩}

Defines the abbreviation ⟨control sequence⟩ with the definition ⟨definition⟩. The star argument prevents that a dot is added at the end of the definition. An error is raised if ⟨control sequence⟩ already exists.

\renewabbr*{⟨control sequence⟩}{⟨definition⟩}

Redefines the abbreviation ⟨control sequence⟩ with the definition ⟨definition⟩. The star argument prevents that a dot is added at the end of the definition. An error is raised if ⟨control sequence⟩ does not exist already.

\footnote{I can see the contradiction here: if a command is no user command there is no need for an optional argument.}
8. Usage of the Various Functions

\defabbr*{⟨control sequence⟩}{⟨definition⟩}
Defines or overwrites the abbreviation ⟨control sequence⟩ with the definition ⟨definition⟩. The star argument prevents that a dot is added at the end of the definition.

\cnltxtimeformat{⟨abbreviation⟩}
Used in some predefined abbreviations. Default: \textsc{\,#1}

\cnltxlatin{⟨abbreviation⟩}
Used in some localization strings. Default: \textit{#1}

acronym-format = {⟨definition⟩}
Formatting of the acronyms as typeset with \cnltxacronym. Default: \scshape

name-format = {⟨formatting commands⟩}
The formatting of names created with \newname or typeset with \name. Names typeset through the bibliography style \cnltx are also formatted according to this option. ⟨formatting commands⟩ should contain #1 for the actual name. Default: #1

last-name-format = {⟨formatting commands⟩}
The formatting of the last names created with \newname or typeset with \name. Names typeset through the bibliography style \cnltx are also formatted according to this option. ⟨formatting commands⟩ should contain #1 for the actual name. Default: \textsc{#1}

first-name-format = {⟨formatting commands⟩}
The formatting of first names created with \newname or typeset with \name. Names typeset through the bibliography style \cnltx are also formatted according to this option. ⟨formatting commands⟩ should contain #1 for the actual name. Default: #1

A short example of the usage of \newname and \cnltxacronym:

\newname\carlisle{David Carlisle}
\carlisle is a well-known member of the \LaTeX\ community. \carlisle is the author of many packages such as \pkg{longtable}. Take a look in the index where you'll find \carlisle mentioned.
lppl is defined as \cnltxacronym{LPPL}{lppl}.

David CARLISLE is a well-known member of the \LaTeX\ community. David CARLISLE is the author of many packages such as longtable. Take a look in the index where you'll find David CARLISLE mentioned.
lppl is defined as lppl.
8. Usage of the Various Functions

8.8.2. Defining Abbreviations

In section 8.8.1 when describing `\newabbr` and similar commands I said “The star argument prevents that a dot is added at the end of the definition”. We should clarify what that means. Many abbreviations end with a dot. Some don’t which explains the starred form of the commands. But why add a dot automatically in the first place? The reasoning is two-fold:

- Suppose you add the dot explicitly in the definition but forget one or two times that you did – you’ll end up with abbreviations followed by two dots! Macros defined with \texttt{\textbackslash cnltx-tools} recognize a following dot and will not print a second one in those cases.

- In a document where `\nonfrenchspacing` is active the space after a dot in the middle of a sentence should be shorter than the one after the full stop ending a sentence. \TeX{} automatically interprets a dot following a small letter as the end of a sentence and a dot after a capital letter as a dot after an abbreviation inside of a sentence. Usually you solve this by adding `\@` in the appropriate places: e.g. `\@` for a intra-sentence space and `NSA` for a inter-sentence space. The dot added by \texttt{\textbackslash cnltx-tools} always will be followed by an intra-sentence space. If you add a dot explicitly it will be your responsibility. Per default it will then act like a dot after a small letter.

Let’s see some example:

\begin{verbatim}
\ttfamily% <= this will amplify the
visual effect of \nonfrenchspacing
\newabbr\ab{a.b}%% a.b. and some words
\newabbr\AB{A.B}%% A.B. and some words
\newabbr\cd{cd}%% cd and some words
\ab\ and some words\par a.b. and some words
\AB\ and some words\par A.B. and some words
\cd\ and some words\par cd and some words
\end{verbatim}

Beware: \texttt{\textbackslash cnltx-tools} will only leave the dot out if one follows directly in the input! That means that spaces are not ignored. However, of course \TeX{} ignores spaces after macro names so usually this won’t be an issue. If you define an abbreviation with a macro name consisting of one non-letter where spaces are not ignored you have to keep this fact in mind, though.

8.8.3. Predefined Abbreviations

\texttt{\textbackslash cnltx-tools} already provides a bunch of abbreviations defined with its `\newabbr` command.

Abbreviations that allow Localization

\texttt{\textbackslash cnltx-tools} defines a few abbreviations that are sensitive to babel settings. Currently only translations for English and German are provided and the definition falls back to the English
version if you’re using a language other than those. It is possible to add further localization strings quite easily, see section 14.

\textit{\texttt{\textbackslash ie}}

Prints “\textit{i. e}.” or “\textit{d. h}.”

\textit{\texttt{\textbackslash eg}}

Prints “\textit{e. g}.” or “\textit{z. B}.”

\textit{\texttt{\textbackslash etc}}

Prints “\textit{etc}.” or “\textit{etc}.”

\textit{\texttt{\textbackslash cf}}

Prints “\textit{cf}.” or “\textit{vgl}.”

All of these macros add a final dot followed by \texttt{\textbackslash@} except if a dot directly follows the macro.

<table>
<thead>
<tr>
<th>eg \ and some following text \par</th>
<th>e. g. and some following text</th>
</tr>
</thead>
<tbody>
<tr>
<td>eg, and some following text \par</td>
<td>e. g., and some following text</td>
</tr>
<tr>
<td>eg. and some following text \par</td>
<td>e. g. and some following text</td>
</tr>
<tr>
<td>\selectlanguage{ngerman} eg \ and some following text \par</td>
<td>z. B. and some following text</td>
</tr>
<tr>
<td>eg, and some following text \par</td>
<td>z. B., and some following text</td>
</tr>
<tr>
<td>eg. and some following text</td>
<td>z. B. and some following text</td>
</tr>
</tbody>
</table>

\textbf{German Abbreviations}

The following abbreviations are not sensitive to localization and are only of use in a German text. Although they’re defined: please do not use abbreviations at the start of a sentence!

\textit{\texttt{\textbackslash dsh}}

Prints “\textit{d. h}.”

\textit{\texttt{\textbackslash Dsh}}

Prints “\textit{D. h}.”

\textit{\texttt{\textbackslash usf}}

Prints “\textit{usf}.”

\textit{\texttt{\textbackslash usw}}

Prints “\textit{usw}.”

\textit{\texttt{\textbackslash uswusf}}

Prints “\textit{usw. usf}.”

\textit{\texttt{\textbackslash zB}}

Prints “\textit{z. B}.”
8. Usage of the Various Functions

\ZB
Prints “Z. B.”

\vgl
Prints “vgl.”

\Vgl
Prints “Vgl.”

These macros behave the same as the ones described in section 8.8.3 on page 33.

\AM
Prints “A.M.”

\PM
Prints “P.M.”

\AD
Prints “A.D.”

\BC
Prints “B.C.”

In their current definition these abbreviations are meant to be used directly after the time of day or the date, respectively.

\ZB \dsh\ und weiterer Text\par
\vgl \dsh\ und weiterer Text\par
\Vgl \dsh\ und weiterer Text\par
\AD \dsh\ und weiterer Text\par
\AM \dsh\ und weiterer Text\par
\PM \dsh\ und weiterer Text\par
\USW \dsh\ und weiterer Text\par
\USF \dsh\ und weiterer Text\par
z. B. \usw\ und weiterer Text\par
z. B. \usf\ und weiterer Text\par
z. B. \zB\ und weiterer Text\par
z. B. \zB. und weiterer Text

---

Time related Abbreviations
The abbreviations presented in this section differ from the others in that they’re formatted by the command \cnltxtimeformat{}, see section 8.8.1 on page 31.

\AM
Prints “A.M.”

\PM
Prints “P.M.”

\AD
Prints “A.D.”

\BC
Prints “B.C.”

In their current definition these abbreviations are meant to be used directly after the time of day or the date, respectively.

She left for work before 6\AM, but did not arrive until 12\PM. The interval 5\BC--5\AD is one year shorter than the interval 95\AD--105\AD.

She left for work before 6 A.M., but did not arrive until 12 P.M. The interval 5 B.C.–5 A.D. is one year shorter than the interval 95 A.D.–105 A.D.
9. Formatting Possibilities

One of the goals I wanted to achieve with this package is a consistent look and an easy interface for customization. No font choice and no color choice is fixed. In this section ways to change the formatting are shown.

The formatting of the different commands provided by \texttt{CnLtx} and various other properties can be changed in two ways: either by redefining the internal commands that are used for the formatting or by setting a corresponding option. Both variants are described in the next subsections.

How the colors should be changed is described in section 13 on page 49.

9.1. Formatting by Redefining Hooks

You can change the formatting by redefining the following commands. They're all defined by the \texttt{CnLtx-Example} package except where indicated differently.

\texttt{\textbackslash codefont}

This command is used for all formatting of source code.

\texttt{\textbackslash sourceformat}

Formatting of the listings.

\texttt{\textbackslash exampleformat}

Special formatting of the output of a listing.

\texttt{\textbackslash versionnoteformat}

Formatting of the notes introduced in section 6.2 on page 9.

\texttt{\textbackslash packageformat}

The formatting of package names.

\texttt{\textbackslash classformat}

The formatting of class names.

\texttt{\textbackslash argumentformat}

The formatting of \texttt{\textbackslash meta{\langle meta\rangle}}.

\examplefont
\texttt{\textbackslash renewcommand{\textbackslash codefont{\textbackslash sffamily{\bfseries}}}}
\texttt{\textbackslash code{foo} and \texttt{\textbackslash cs*{bar}}, option \texttt{\textbackslash option{baz}}}

\texttt{foo and \textbackslash bar, option baz}
9. Formatting Possibilities

9.2. Formatting by Setting Options

You can change the formatting of by setting the following options. They’re all defined by the `csltx-example` package except where indicated differently.

- **title-format** = \{\langle definition \rangle\}
  Formatting of the document title.
  Default: \bfseries\scshape

- **abstract-width** = \{\langle dimension \rangle\}
  The width of the \texttt{parbox} the abstract as set with the \texttt{abstract} option is placed in.
  Default: \texttt{.75\linewidth}

- **abstract-format** = \{\langle definition \rangle\}
  Code that is placed in the \texttt{parbox} the abstract is placed in before the abstract text.
  Default: \texttt{\setlength\parskip{.333\baselineskip}}

- **caption-font** = \{\langle definition \rangle\}
  This option only has any effect if you use the option \texttt{load-preamble}, see section 10.5 on page 41 for details on the option.
  Default: \texttt{\normalfont\small\sffamily}

- **caption-label-font** = \{\langle definition \rangle\}
  This option only has any effect if you use the option \texttt{load-preamble}, see section 10.5 on page 41 for details on the option.
  Default: \texttt{\normalfont\small\sffamily\scshape}

- **code-font** = \{\langle definition \rangle\}
  Used for all formatting of source code.
  Default: \texttt{\ttfamily}

- **source-format** = \{\langle definition \rangle\}
  Formatting of the listings.
  Default: \texttt{\codefont\small}

- **expl-format** = \{\langle definition \rangle\}
  Special formatting of the output of a listing.
  (initially empty)

- **module-sep** = \{\langle definition \rangle\}
  Change the separator between module name and corresponding option name.
  Default: \texttt{\,\textgreater\textless\,}

- **version-note-format** = \{\langle definition \rangle\}
  Formatting of the notes introduced in section 6.2 on page 9.
  Default: \texttt{\footnotesize\sffamily\RaggedRight}

- **pkg-format** = \{\langle definition \rangle\}
  The formatting of package names.
  Default: \texttt{\sffamily}

- **cls-format** = \{\langle definition \rangle\}
  The formatting of class names.
  Default: \texttt{\sffamily}

- **arg-format** = \{\langle definition \rangle\}
  The formatting of \texttt{\meta\{\langle meta \rangle\}}.
  Default: \texttt{\normalfont\itshape}

- **default-format** = \{\langle code \rangle\}
  The formatting of \texttt{\default}'s argument. \texttt{(code)}'s last macro should take one argument.
  Default: \texttt{\uline}
10. Commands, Options and Further Settings Directly Related to the \texttt{CNLTX-DOC} Class

10.1. Using Class Options

The \texttt{CNLTX-DOC} class only knows a few options:

- \texttt{load-preamble = \texttt{true}|false} \\
  Default: \texttt{false} \\
  See section 10.5 on page 41 for details.

- \texttt{load-preamble+ = \texttt{true}|false} \\
  Default: \texttt{false} \\
  See section 10.6 on page 42 for details.

- \texttt{add-index = \texttt{true}|false} \\
  Default: \texttt{false} \\
  See section 10.6 on page 42 for details.

- \texttt{babel-options = \{\langle options\rangle\}} \\
  Default: \texttt{english} \\
  Options given to the \texttt{babel} package. This option only has an effect if \texttt{load-preamble = true}.

- \texttt{scrartcl = \{\langle options\rangle\}} \\
  (initially empty) \\
  Options that are passed to the underlying class \texttt{scrartcl}. \textit{All global options you want to use should be given here.}

10.2. Information on the Described Package or Class

A manual for a package or a class needs some information on the described package like the package name, the version number, the date and so on. This information is given with the following options. They are used to build the title page of the manual.

- \texttt{package = \{\langle package\rangle\}} \\
  The name of the package that is described. Either this option or \texttt{class} or \texttt{name} should always be given. This command also defines a command sequence from the package name that formats the package name with color and small caps like \texttt{CNLTX}.

- \texttt{class = \{\langle class\rangle\}} \\
  The name of the class that is described. Either this option or \texttt{package} or \texttt{name} should always be given. This command also defines a command sequence from the class name that formats the class name with color and small caps like \texttt{CNLTX}.

\footnote{on \texttt{CTAN} as \texttt{babel}: \url{http://mirrors.ctan.org/macros/latex/required/babel/}}
name = \{(name)\}
The name of the class/package that is described. Either this option or \texttt{package} or \texttt{class} should always be given. This command also defines a command sequence from the class name that formats the class name with color and small caps like \texttt{C\textsc{N}L\textsc{T}X}.

authors = \{(author list)\}
Comma separated list of package/class authors. After each author name you can add an email address by writing it in square brackets: \texttt{Some Name[some@name.com]}. Email addresses specified this way get written as a footnote. At least one author should always be given.

version = \{(version number)\}
Version number of the package/class. \texttt{C\textsc{N}L\textsc{T}X} tries to extract the information from the given \texttt{package} or \texttt{class}. This option can be used to set it explicitly.

date = \{(date)\}
Date of the package/class. \texttt{C\textsc{N}L\textsc{T}X} tries to extract the information from the given \texttt{package} or \texttt{class}. This option can be used to set it explicitly.

info = \{(package/class info)\}
Information about the package/class. \texttt{C\textsc{N}L\textsc{T}X} tries to extract the information from the given \texttt{package} or \texttt{class}. This option can be used to set it explicitly.

subtitle = \{(subtitle)\}
A subtitle, printed below the package/class name.

url = \{(url)\}
The homepage of the package.

email = \{(email)\}
A contact email address.

abstract = \{(abstract)\}
An abstract of the package/class/manual. This is text typeset in a box of $0.75\text{\textwidth}$. Actually it does not have to be text but could be an image or whatever you like.

10.3. Building of the Manuals Title Page

If either the \texttt{package} or \texttt{class} has been given an automatic title page is built using the gathered information. Figure 1 on the next page roughly sketches which informations is used and how the different elements are arranged on the title page. The page style of the title page is \texttt{plain}. Additionally a table of contents is automatically built that is set in two columns. The automatic building of the title page can be prevented by explicitly setting the following option:

\texttt{build-title = true|false}
The default state depends on other options given like \texttt{package}. However, setting this option to \texttt{false} after any of the options described in section 10.2 on the preceding page will prevent the building of a title page and allows you to design your own.
10. Commands, Options and Further Settings Directly Related to the \texttt{CnLtx-Doc} Class

\begin{quote}
\texttt{Douglas Adams, The Restaurant at the End of the Universe}
``The first ten million years were the worst,'' said Marvin, ``and the second ten million years, they were the worst too. The third ten million years I didn't enjoy at all. After that I went into a bit of a decline.''
\end{quote}
10. Commands, Options and Further Settings Directly Related to the \texttt{cnltx-doc} Class

The first ten million years were the worst, said Marvin, and the second ten million years, they were the worst too. The third ten million years I didn’t enjoy at all. After that I went into a bit of a decline.

\textit{Douglas Adams, The Restaurant at the End of the Universe}

10.5. Predefined Preamble

It is possible to load a part of my standard preamble automatically by passing an option as class option.

\texttt{load-preamble}

Class option that preloads part of my custom preamble.

Using the option will include the following code:

\begin{verbatim}
1 \RequirePackage{ifxetex,ifluatex}
2 \ifboolexpr{not bool{xetex} and not bool{luatex}}
3 \{\RequirePackage[T1]{fontenc}}
4 \{\RequirePackage{fontspec}}
5 \RequirePackage[oldstyle]{libertine}
6 \%
7 \% `libertinehologopatch' is not on CTAN, yet!
8 \%
9 \% you can get it at https://bitbucket.org/cgnieder/libertinehologopatch/
10 \RequirePackage{libertinehologopatch}
11 \RequirePackage[superstfm=libertinesups]{superiors}
12 \%
13 \% libertine does not have superior letters:
14 \def\@makefnmark{%
15 \hbox{%
16 \cnltx@ifisnum{@thefnmark}
17 \{\textsu{\hspace*{\superiors@spaced}@thefnmark}}\@textsuperscript{\normalfont@thefnmark}%
18 \}
19 \}
20 \RequirePackage{microtype}
21 \ifboolexpr
22 { test \{\ifcsdef{MT@pr@set@@romansans}} and
23 test \{\ifcsdef{MT@ex@set@@romansans}}
24 }\{}
25 \{\Declaremicrotypeset{romansans}{
26 encoding = {*},
27 family = {rm*,sf*}
28 }\}
29 \ifcsdef{MT@tr@set@@scshape}
\end{verbatim}
The effect of this preamble is demonstrated by the document you’re reading at this moment.

## 10.6. Predefined Indexing

**\textsc{cnltx-doc}** allows the automated creation of an index. This is done with the help of the \texttt{imakeidx} package by Enrico Grech [Gre13]. To use this feature you have two class options. They cannot be set with \texttt{\setcnltx} but must be given as class options.

### add-index = true|false

Default: false

Enables the automatic creation of an index at the end of the document.

### load-preamble+ = true|false

Default: false

This option has the same effect as adding the options \texttt{load-preamble}, \texttt{add-index} and \texttt{add-bib}.

Enabling the feature

- loads the \texttt{imakeidx}\textsuperscript{22} package.

\textsuperscript{22} on CTAN as \texttt{imakeidx}: \url{http://mirrors.ctan.org/macros/latex/contrib/imakeidx/}
10. Commands, Options and Further Settings Directly Related to the \texttt{cnltx-doc} Class

- uses a given style file for the index that can be specified with the \texttt{index-style} option,
- sets a certain setup for the index that can be specified with the \texttt{index-setup} option and
- adds an index at the end of the document.

The following options are available to customize the appearance of the index:

\begin{itemize}
\item \texttt{index-prologue = \{\langle text\rangle\}}
  \begin{itemize}
  \item Adds \langle text\rangle as index prologue between heading and the actual index.
  \end{itemize}
\item \texttt{index-space = \{\langle dimension\rangle\}}
  \begin{itemize}
  \item Default: 0pt
  \item The vertical space between index prologue and index.
  \end{itemize}
\item \texttt{index-setup = \{\langle options\rangle\}}
  \begin{itemize}
  \item Default: othercode=\footnotesize,level=\addsec
  \item The options that are passed to imakeidx's \texttt{indexsetup} command.
  \end{itemize}
\item \texttt{makeindex-setup = \{\langle options\rangle\}}
  \begin{itemize}
  \item Default: columns=2,columnsep=1em
  \item The options that are passed to the \texttt{makeindex} command.
  \end{itemize}
\item \texttt{index-style = \{\langle style file\rangle\}}
  \begin{itemize}
  \item Default: cnltx.ist
  \item The style file that is used for formatting the index.
  \end{itemize}
\end{itemize}

The index style file \texttt{cnltx.ist} contains the following lines:

\begin{verbatim}
heading_prefix "\bfseries "
heading_suffix "\\hfill\\nopagebreak\\n"
headings_flag 1
delim_0 "\\dotfill"
delim_1 "\\dotfill"
delim_2 "\\dotfill"
delim_r "\\nohyperpage{\\textendash}"
delim_t ""
suffix_2p "\\nohyperpage{\\\GetTranslation{cnltx-f.}\\@}"
suffix_3p "\\nohyperpage{\\\GetTranslation{cnltx-ff.}\\@}"
\end{verbatim}

The feature is demonstrated by this document which does not contain a single control sequence containing the string \texttt{index}.

\section*{10.7. Bibliography with biblatex}

\subsection*{10.7.1. Bibliography Entry Types package, class and bundle for biblatex}
\texttt{cnltx-doc} defines the bibliography entry types package, class and bundle when biblatex [Leh13] is used. This allows specifying \LaTeX\ packages in \bib\ files:
As you can see also an entry field maintainer is defined. For this to work you have to use the biblatex bibliography style \texttt{cnltx}. This style basically is a clone of the style \texttt{alphabetic} but defines the necessary additions for the package, class and bundle entry types and the maintainer entry field.

Along with the bibliography style a citation style \texttt{cnltx} is provided, again a clone of the \texttt{alphabetic} style. The only addition it makes is that indexing of maintainer names is enabled if biblatex’s indexing option is used. The styles load \texttt{CNLTX-EXAMPLE} as it relies on definitions made by it.

This document uses the following call of biblatex:
Actually it let’s **cnltx-doc** do it, see section 10.7.2 for details. Just for the sake of the example I am going to cite the chngcntr package now [Wil09] so you can see both the bibliography entry and the indexed names of package, author and maintainer in the appendix.

**10.7.2. Automatic Bibliography**

**Cnltx-doc** allows the automated creation of a bibliography.

\begin{itemize}
\item \texttt{add-bib = true|false} \\
\hspace{2em} Default: false \\
\text{"Enables the automatic creation of a bibliography at the end of the document."}
\item \texttt{load-preamble+ = true|false} \\
\hspace{2em} Default: false \\
\text{"This option has the same effect as adding the options \texttt{load-preamble, add-index} and \texttt{add-bib}."}
\end{itemize}

What this options does is including the following code:

```
\RequirePackage[backend=biber, style=cnltx, sortlocale=en.US, indexing=cite, useprefix]{biblatex}
\addbibresource{cnltx.bib}
\AtEndDocument{\printbibliography}
```

As you can see there’s also a bibliography database file cnltx.bib that provides a yet small but growing number of package entries.

**11. Predefined listings and mdframed Styles**

**11.1. mdframed**

The source code environments (see section 8.4 on page 18) all get a frame with the help of the \texttt{mdframed} [Dan13] package. For this a custom style is defined called \texttt{cnltx}. The options \texttt{frame-options} and \texttt{add-frame-options} mentioned in section 8.4 on page 18 manipulate this style. It is predefined with these values:
11. Predefined listings and mdframed Styles

11.2. listings

11.2.1. \LaTeX Sourcecode

The code of the source code environments (see section 8.4 on page 18) is formatted with the help of the listings package [HM13]. A listings style is defined called \texttt{cnltx}. The options \texttt{add-cmds}, \texttt{add-silent-cmds}, \texttt{add-envs}, \texttt{add-silent-envs}, \texttt{listings-options} and \texttt{add-listings-options} manipulate this style. It is predefined by \texttt{CNLTX-EXAMPLE} as follows:

\begin{verbatim}
def\cnltx@listings@style{
    language = [AlLaTeX]TeX,
    alsolanguage = [plain]TeX,
    basicstyle = {\sourceformat},
    numbers = left,
    numberstyle = \tiny,
    xleftmargin = 1em,
    numbersep = .75em,
    gobble = \cnltx@gobble,
    columns = fullflexible,
    literate =
        {ä}{{"a}}1
        {ö}{{"o}}1
        {ü}{{"u}}1
        {Ä}{{"A}}1
        {Ö}{{"O}}1
        {Ü}{{"U}}1
        {ß}{{\ss}}1,
    breaklines = true,
    keepspaces = true,
    breakindent = 1em,
    commentstyle = \color{comment},
    keywordstyle = \color{cs},
    deletetexcs =
        a,o,u,A,O,U,
        begin,
        center,
        description,document,
}
def\cnltx@mdframed@options{
    backgroundcolor = cnltxbg,
    linecolor = cnltx,
    roundcorner = 5pt
}
\end{verbatim}

46
11. Predefined listings and mdframed Styles

```latex
end, enumerate,
figure, flushleft, flushright,
itemize, list,
otherlanguage,
table, tabu, tabular
},
deletekeywords =
{
  a, o, u, A, O, U,
  begin,
center,
description, document,
end, enumerate,
figure, flushleft, flushright,
itemize, list,
otherlanguage,
table, tabu, tabular
},
%egin, \end:
texcsstyle = \{2\color{beginend},
index = \{2\{texcs2\},
indexstyle = \{2\}@gobble,
moretexcs = \{2\begin, end\},
% added environments that'll be indexed:
texcsstyle = \{3\color{env},
index = \{3\{texcs3\},
indexstyle = \{3\}envidx,
% environments that won't be indexed:
texcsstyle = \{4\color{env},
index = \{4\{texcs4\},
indexstyle = \{4\}@gobble,
% control sequences that'll be indexed:
texcsstyle = \{5\color{cs},
index = \{5\{texcs5\},
indexstyle = \{5\}indexcs,
% control sequences that won't be indexed:
texcsstyle = \{6\color{cs},
index = \{6\{texcs6\},
indexstyle = \{6\}@gobble
%
```}

11.2. BibTeX Entries

The \texttt{cnltx-listings} package defines a listings language BibTeX that contains a huge number of bibentry types and bibentry field types, have a look at section 10.7.1 on page 43. \texttt{cnltx-example} defines a listings style for formatting them called \texttt{cnltx-bibtex}:

---

*Introduced in version 0.4*
11. Predefined listings and mdframed Styles

\begin{verbatim}
def\cnlt@bibtex@listings@style{
  language = \texttt{BibTeX},
  basicstyle = \texttt{\sourceformat},
  numbers = \texttt{left},
  numberstyle = \texttt{\tiny},
  xleftmargin = \texttt{1em},
  numbersep = \texttt{.5em},
  gobble = \texttt{\cnlt@gobble},
  columns = \texttt{fullflexible},
  literate =
    \{ä\}{{"a}}1,
    \{ö\}{{"o}}1,
    \{ü\}{{"u}}1,
    \{Ä\}{{"A}}1,
    \{Ö\}{{"O}}1,
    \{Ü\}{{"U}}1,
    \{ß\}{{\ss}}1,
    \breaklines = \texttt{true},
    \breakindent = \texttt{1em},
    \keepspaces = \texttt{true},
    \commentstyle = \texttt{\color{comment}},
    \keywordstyle = \texttt{\color{bibentry}},
    \keywordstyle = \texttt{[2]\color{bibentryfield}\itshape},
    \showstringspaces = \texttt{false},
}
def\cnlt@makeindex@listings@style{
  language = \texttt{makeindex},
  basicstyle = \texttt{\sourceformat},
  numbers = \texttt{left},
  numberstyle = \texttt{\tiny},
  xleftmargin = \texttt{1em},
  numbersep = \texttt{.75em},
  gobble = \texttt{\cnlt@gobble},
  columns = \texttt{fullflexible},
  literate =
    \{ä\}{{"a}}1,
    \{ö\}{{"o}}1
\end{verbatim}

11.2.3. makeindex Style Files

\texttt{\textsc{Cnltx-listings}} defines a listings language \texttt{makeindex} that contains the keywords used in \texttt{makeindex} style files. \texttt{\textsc{Cnltx-example}} defines a listings style for formatting them called \texttt{cnltx-makeindex}:
12. PDF Strings and hyperref

Since the formatting and indexing commands \cs, \env, \option, \pkg, \cls and \key are robust they are ignored in PDF strings. For this reason you should only use the starred variants in places where PDF bookmarks are built from such as section titles when you use hyperref [OR12]. Since \texttt{CNLTX-DOC} loads hyperref this means you should do so, too, when you use \texttt{CNLTX-DOC}. This is important for two reasons:

1. Indexing in strings that get written to the table of contents does no make much sense, anyway, so the starred versions should be used in section titles even if you don’t use hyperref.

2. When hyperref is loaded the mentioned commands are disabled in PDF strings in a way that expects them to be followed by a star. This means leaving the star out will result in doesn’t match its definition errors.

13. Predefined Colors and Color-Schemes

13.1. Explicitly Defined Colors

The \texttt{CNLTX-BASE} package defines a number of colors:

\texttt{cnltxbrown}

Per default used for the control sequences.

\texttt{cnltxblue}

Per default used for module names.

\texttt{cnltxred}

Per default used as base color in various places.

\texttt{cnltxgreen}

Unused per default.
13. Predefined Colors and Color-Schemes

cnltxgray
   Per default used for formatting comments.

cnltxyellow
   Per default used for option names.

cnltxformalblue
   Unused per default.

cnltxformalred
   Unused per default.

13.2. Actual Used Color Names and Color Schemes

The colors defined in section 13.1 on the previous page are not directly used with those names. Instead colors are used whose names describe their function rather than the color. For this the color names are mapped to actual colors and saved as a coloring scheme. There are currently three predefined color schemes whose definitions are given below. Those definitions also show the actually used color names. They are defined via the following command:

\definecolorscheme{⟨name⟩}{⟨color assignments⟩}

Defines the color scheme ⟨name⟩. When used all assignments will be actually carried out with xcolor’s \colorlet command. How to input ⟨color assignments⟩ will be immediately clear from the examples below.

To activate a color scheme for a document it is simply selected through an option:

color-scheme = {⟨color scheme name⟩}

Default: default
Activate a color scheme previously defined with \definecolorscheme.

The 'default' color scheme is defined as follows:

\definecolorscheme{default}{
  cs => cnltxbrown , % command sequences
  option => cnltxyellow ,% options
  module => cnltxblue , % modules
  comment => cnltxgray , % comments
  beginend => red , % \begin and \end
  env => black , % environment names
  argument => black , % argument delimiters
  meta => black!80 , % arguments of \meta
  cnltx => cnltxred , % base color
  cnltxbg => white , % source code box background
  link => black!90 , % hyperlinks
  versionnote => black!75 % versioning notes text
  bibentry => cnltxgreen , % BibTeX entry types
  bibentryfield => black , % BibTeX entry fields
}
13. Predefined Colors and Color-Schemes

The ‘blue’ color scheme is defined this way:

```
\definecolorscheme{blue}{
  cs => cnltxbrown ,
  option => cnltxgreen ,
  module => cnltxred ,
  comment => cnltxgray ,
  beginend => red ,
  env => black ,
  argument => black ,
  meta => black!80 ,
  cnltx => cnltxblue ,
  cnltxbg => yellow!10 ,
  link => cnltx ,
  versionnote => black!75
}
```

Finally the ‘formal’ color scheme is defined like this:

```
\definecolorscheme{formal}{
  cs => black ,
  option => cnltxformalblue ,
  module => cnltxblue ,
  comment => cnltxgray ,
  beginend => red ,
  env => black ,
  argument => black ,
  meta => black!80 ,
  expandable => black ,
  makeidxkey => cnltxyellow ,
  makeidxstring => black
}
```
14. Language Support

The \texttt{CNLTX-DOC}, the \texttt{CNLTX-EXAMPLE} and the \texttt{CNLTX-TOOLS} package as well as the \texttt{cnltx.ist} index style and the \texttt{cnltx} biblatex style all rely on the translations package \cite{Nie13d} for providing some document language dependent strings.\footnote{Actually they depend on \texttt{CNLTX-TRANSLATIONS} which in turn loads translations.} Currently only translations for English and German are provided. Others can be added and the existing ones changed with the following commands provided by the translations package:

\begin{verbatim}
\DeclareTranslation{⟨language⟩}{⟨keyword⟩}{⟨translation⟩}
\end{verbatim}

Define or redefine translations for the string identified by the ID \texttt{⟨keyword⟩}.

\begin{verbatim}
\RenewTranslation{⟨language⟩}{⟨keyword⟩}{⟨translation⟩}
\end{verbatim}

Renew translations for the string identified by the ID \texttt{⟨keyword⟩}.

The strings defined by \texttt{CNLTX} are listed in table 1 on the following page. They are used in indexing strings and in different parts of the document.

\section*{Part III.}
\section*{Appendix}

\subsection*{A. Internal Helper Commands}

The commands in this section are only described for the sake of completeness. They are not meant to be used in a document. Some of them might be useful in \LaTeX{} programming, though. Expandable commands are marked with $\ast$.

\subsection*{A.1. Defined by \texttt{CNLTX-BASE}}

Especially \texttt{CNLTX-BASE} defines some useful helper macros that are also used by the other packages and classes.
### A. Internal Helper Commands

<table>
<thead>
<tr>
<th>Package/Class</th>
<th>key word</th>
<th>English version</th>
<th>German version</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cnltx-example</strong></td>
<td>cnltx-package</td>
<td>package</td>
<td>Paket</td>
</tr>
<tr>
<td><strong>cnltx-example</strong></td>
<td>cnltx-class</td>
<td>class</td>
<td>Klasse</td>
</tr>
<tr>
<td><strong>cnltx-example</strong></td>
<td>cnltx-bundle</td>
<td>bundle</td>
<td>Bundle</td>
</tr>
<tr>
<td><strong>cnltx-example</strong></td>
<td>cnltx-environment</td>
<td>environment</td>
<td>Umgebung</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-default</td>
<td>Default</td>
<td>Voreinstellung</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-empty</td>
<td>initially empty</td>
<td>zunächst leer</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-required</td>
<td>required</td>
<td>erforderlich</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-toc</td>
<td>Table of Contents</td>
<td>Inhaltsverzeichnis</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-license</td>
<td>Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License (LPPL), version 1.3 or later (<a href="http://www.latex-project.org/lppl.txt">http://www.latex-project.org/lppl.txt</a>). The software has the status Es ist erlaubt, diese Software unter den Bedingungen der LaTeX Project Public License (LPPL), Version 1.3 oder später, zu kopieren und zu verteilen (<a href="http://www.latex-project.org/lppl.txt">http://www.latex-project.org/lppl.txt</a>). Sie hat den Status</td>
<td></td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-introduced</td>
<td>Introduced in version</td>
<td>Eingeführt in Version</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-changed</td>
<td>Changed in version</td>
<td>Geändert in Version</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-f.</td>
<td>f.</td>
<td>f.</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-ff.</td>
<td>ff.</td>
<td>ff.</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-maintainer</td>
<td>current maintainer</td>
<td>aktueller Maintainer</td>
</tr>
<tr>
<td><strong>cnltx-doc</strong></td>
<td>cnltx-maintainers</td>
<td>current maintainers</td>
<td>aktuelle Maintainer</td>
</tr>
<tr>
<td><strong>cnltx-tools</strong></td>
<td>cnltx-i.e.</td>
<td>i.e</td>
<td>d. h</td>
</tr>
<tr>
<td><strong>cnltx-tools</strong></td>
<td>cnltx-e.g.</td>
<td>e.g</td>
<td>z. B</td>
</tr>
<tr>
<td><strong>cnltx-tools</strong></td>
<td>cnltx-cf.</td>
<td>cf</td>
<td>vgl</td>
</tr>
<tr>
<td><strong>cnltx-tools</strong></td>
<td>cnltx-etc.</td>
<td>etc</td>
<td>etc</td>
</tr>
</tbody>
</table>

53
A. Internal Helper Commands

A.1. Related to the Bundle

\texttt{\textbackslash cnltx@date}

The creation date of the current version of the bundle.

\texttt{\textbackslash cnltx@version}

The version number of the bundle.

\texttt{\textbackslash cnltx@info}

The short description of the bundle.

\texttt{\textbackslash cnltx@create@bundle@message*\{⟨module⟩\}\{Error|Warning|WarningNoLine|Info\}}

Create suited error and warning messaging commands for the module \texttt{⟨module⟩} of the \texttt{cnltx} bundle. The starred version creates messages for a class the un-starred version messages for a package.

\texttt{\textbackslash cnltx@base@error\{⟨message⟩\}}

Issue an error message using \texttt{\PackageError{cnltx-base}}.

\texttt{\textbackslash cnltx@base@warning\{⟨message⟩\}}

Issue a warning message using \texttt{\PackageWarning{cnltx-base}}.

\texttt{\textbackslash cnltx@base@warningnoline\{⟨message⟩\}}

Issue a warning message using \texttt{\PackageWarningNoLine{cnltx-base}}.

\texttt{\textbackslash cnltx@base@info\{⟨message⟩\}}

Issue a message using \texttt{\PackageInfo{cnltx-base}}.

\texttt{\textbackslash cnltx@define@colorscheme\{⟨name⟩\}⟨scheme definition⟩}

Command that can be used to define a color scheme.

\texttt{\textbackslash cnltx@load@module\{⟨CNLTX module⟩\}}

Loads the package \texttt{cnltx-⟨CNLTX module⟩.sty}.

\texttt{\textbackslash cnltx@load@modules\{⟨CNLTX modules⟩\}}

Maps the comma separated list \texttt{⟨CNLTX modules⟩} to \texttt{\textbackslash cnltx@load@module}, leading and trailing spaces are trimmed.

A.1.2. Programming Tools

Message Handling

\texttt{\textbackslash cnltx@create@message*\{⟨prefix⟩\}⟨package/class name⟩\{Error|Warning|WarningNoLine|Info\}}

\{(detailed error message)\}

Create error and warning messaging commands \texttt{\textbackslash ⟨prefix⟩@error|warning|warningnoline|info\{⟨message⟩\}. The starred version creates messages for a class the un-starred version messages for a package. All commands have one argument which takes the message. \texttt{⟨prefix⟩} will be all lowercase in the generated command.

54
A. Internal Helper Commands

\texttt{\textbackslash cnltx\textbackslash create\textbackslash generic\textbackslash message}∗\{prefix\}{\{package/class name\}}\{Error|Warning|WarningNoLine|Info\}

Created error and warning messaging commands \{prefix\}@error|warning|warningno\-line info\{message\}. The starred version creates messages for a class the un-starred version messages for a package. All commands have one argument which takes the message except for the error command which gets two arguments, the first for the short version and the second for the detailed message. \{prefix\} will be all lowercase in the generated command.

**Conditionals**

\texttt{\textbackslash iftest}\{test directive\}\{true\}\{false\}

Checks if \{test directive\} is true and either places \{true\} or \{false\} in the input stream. \{test directive\} should be a \TeX{} test like \texttt{\textbackslash if\{token\}\{token2\}}, i.e., demand an \texttt{\else} and \texttt{\fi}. This is a command in the spirit of etoolbox’s \texttt{\textbackslash if\{bool\}} that does the same for a boolean \{bool\} defined with \texttt{\new\textbackslash if\{bool\}} or \texttt{\new\textbackslash bool\{bool\}}. It corresponds to etoolbox’s test directive for its \texttt{\textbackslash ifbool\textbackslash expr}.

\texttt{\textbackslash nottest}\{test directive\}\{not true\}\{not false\}

Checks if \{test directive\} is not true and either places \{not true\} or \{not false\} in the input stream. Test directive should be a \TeX{} test like \texttt{\textbackslash if\{token\}\{token2\}}, i.e., demand an \texttt{\else} and \texttt{\fi}. This is a command in the spirit of etoolbox’s \texttt{\not\textbackslash bool} that does the same for a boolean \{bool\} defined with \texttt{\new\textbackslash if\{bool\}} or \texttt{\new\textbackslash bool\{bool\}}.

\texttt{\textbackslash cnltx\textbackslash if\textbackslash counter}\{\textbackslash{counter}\}\{true\}\{false\}

Checks if \{counter\} is a counter, i.e., if the control sequence names \texttt{\cf\{counter\}}, \texttt{\cl\{counter\}}, \texttt{\p\{counter\}} and \texttt{\the\{counter\}} exist and either leaves \{true\} or \{false\} in the input stream.

\texttt{\textbackslash cnltx\textbackslash ifnextchars\{list of tokens\}\{true\}\{false\}\{trailing token\}

Tests if \{trailing token\} is any of those in \{list of tokens\} and either places \{true\} or \{false\} in the input stream without removing \{trailing token\}.

\texttt{\textbackslash cnltx\textbackslash if\textbackslash sym}\{\textbackslash{token}\}\{true\}\{false\}

A generic version of \TeX{}’s \texttt{\textbackslash if\textbackslash star} that checks if \{token\} follows in the input stream. If yes it is removed and \{true\} is placed in the input stream else \{false\}.

\texttt{\textbackslash cnltx\textbackslash if\textbackslash dash}\{\textbackslash{true}\}\{false\}

A wrapper for \texttt{\textbackslash cnltx\textbackslash if\textbackslash sym\{-\}}.

\texttt{\textbackslash cnltx\textbackslash if\textbackslash bang}\{\textbackslash{true}\}\{false\}

A wrapper for \texttt{\textbackslash cnltx\textbackslash if\textbackslash sym\{!\}}.

\texttt{\textbackslash cnltx\textbackslash if\textbackslash is\textbackslash num}\{\textbackslash{token list}\}\{\textbackslash{true}\}\{false\}

Checks if \{token list\} is an integer zero or greater and leaves \{true\} in the input stream if it is and \{false\} if it isn’t. There is one hopefully extremely unlikely case where the test fails: when \{token list\} starts with “\{integer\}\%” where \% has a category code different than 9 (ignored) or 14 (comment).
A. Internal Helper Commands

\texttt{\textbackslash cnltx@ifshellescape\{}\texttt{\langle true\rangle}\texttt{\{}\texttt{\langle false\rangle}}\texttt{\}}

Checks if shellescape is enabled. It returns true if pdftexcmds' \texttt{\pdf@shellescape} has the value 1. This is a wrapper for \texttt{\iftest{\ifnum\pdf@shellescape=1}}.

\texttt{\textbackslash cnltx@ifin\{}\texttt{\langle tokenlist\rangle}\texttt{\{}\texttt{\langle search\rangle}\texttt{\}{\texttt{\langle true\rangle}}\texttt{\}{\texttt{\langle false\rangle}}\texttt{\}}\texttt{\}}

Places \texttt{\langle true\rangle} in the input stream if \texttt{\langle search\rangle} is found in \texttt{\langle tokenlist\rangle} and \texttt{\langle false\rangle} if it isn’t.

\texttt{\textbackslash cnltx@ifstrequal\{}\texttt{\langle string1\rangle}\texttt{\}{\texttt{\langle string2\rangle}}\texttt{\}{\texttt{\langle true\rangle}}\texttt{\}{\texttt{\langle false\rangle}}\texttt{\}}

Tests if \texttt{\langle string1\rangle} is equal to \texttt{\langle string2\rangle} and either leaves \texttt{\langle true\rangle} or \texttt{\langle false\rangle} in the input stream. This test doesn’t take category codes into account.

\texttt{\textbackslash cnltx@ifinlist\{}\texttt{\langle item\rangle}\texttt{\}{\texttt{\langle listmacro\rangle}}\texttt{\}{\texttt{\langle true\rangle}}\texttt{\}{\texttt{\langle false\rangle}}\texttt{\}}

A conditional for etoolbox lists similar to \texttt{\ifinlist} where braces in items are allowed. This wraps around the proposal in etoolbox’s documentation to redefine \texttt{\do} and loop through the list.

\texttt{\textbackslash cnltx@ifinlistcs\{}\texttt{\langle item\rangle}\texttt{\}{\texttt{\langle listsname\rangle}}\texttt{\}{\texttt{\langle true\rangle}}\texttt{\}{\texttt{\langle false\rangle}}\texttt{\}}

A conditional for etoolbox lists similar to \texttt{\ifinlistcs} where braces in items are allowed. This wraps around the proposal in etoolbox’s documentation to redefine \texttt{\do} and loop through the list.

\textbf{Expansion Tools}

\texttt{\textbackslash expandtwice\{}\texttt{\langle code\rangle}}

Expands \texttt{\langle code\rangle} twice in an \texttt{\edef}-like context. This is a wrapper for \texttt{\unexpanded\expandafter\expandafter\expandafter}.\texttt{\expandafter\expandafter\expandafter}.

\texttt{\textbackslash cnltx@expandargs\{}\texttt{\langle specs\rangle}\texttt{\}{\texttt{\langle control sequence\rangle}}}\texttt{\}}

This is a \LaTeX\ version of expl3’s \texttt{\exp_args:N\langle specs\rangle}. The command expands the arguments of \texttt{\langle control sequence\rangle} according to \texttt{\langle specs\rangle}. In \texttt{\langle specs\rangle}

- \texttt{\textbackslash N} means unexpanded token,
- \texttt{\textbackslash n} means unexpanded braced group,
- \texttt{\textbackslash c} means braced group converted into a control sequence name,
- \texttt{\textbackslash o} means braced group expanded once,
- \texttt{\textbackslash f} means braced group expanded with \texttt{\romannumeral}, and
- \texttt{\textbackslash x} means braced group expanded with \texttt{\edef}.

\textbf{Category Code Stuff}

\texttt{\textbackslash cnltx@save@catcode\{}\texttt{\langle token\rangle}}\texttt{\}}

Saves the current category code of \texttt{\langle token\rangle}.

\texttt{\textbackslash cnltx@restore@catcode\{}\texttt{\langle token\rangle}}\texttt{\}}

Restores the category code of \texttt{\langle token\rangle} as previously saved with \texttt{\cnltx@save@catcode}.
A. Internal Helper Commands

\texttt{\textbackslash cnltx@set@catcode\{\texttt{token}\}\{\texttt{catcode}\}}

Sets the category code of \texttt{\{token\}} to \texttt{\{catcode\}}. This is a wrapper for \texttt{\textbackslash catcode\{\texttt{token}\}=\{\texttt{catcode}\}\textbackslash relax}.

\texttt{\textbackslash cnltx@save@catcodes\{\texttt{tokenlist}\}}

Introduced in version /zero.taboldstyle./one.taboldstyle/one.taboldstyle

Maps \texttt{\textbackslash cnltx@save@catcode} to all tokens in \texttt{\{tokenlist\}}.

\texttt{\textbackslash cnltx@restore@catcodes\{\texttt{tokenlist}\}}

Introduced in version /zero.taboldstyle./one.taboldstyle/one.taboldstyle

Maps \texttt{\textbackslash cnltx@restore@catcode} to all tokens in \texttt{\{tokenlist\}}.

\texttt{\textbackslash cnltx@set@catcodes\{\texttt{tokenlist}\}\{\texttt{catcode}\}}

Introduced in version /zero.taboldstyle./one.taboldstyle/one.taboldstyle

Maps \texttt{\textbackslash cnltx@set@catcode} to all tokens in \texttt{\{tokenlist\}}, i.e., all tokens get category code \texttt{\{catcode\}}.

\texttt{\textbackslash cnltx@make@letter\{\texttt{token}\}}

Introduced in version /zero.taboldstyle./one.taboldstyle/one.taboldstyle

A wrapper for \texttt{\textbackslash cnltx@set@catcode\{\texttt{token}\}\{11\}}.

\texttt{\textbackslash cnltx@make@other\{\texttt{token}\}}

Introduced in version /zero.taboldstyle./one.taboldstyle/one.taboldstyle

A wrapper for \texttt{\textbackslash cnltx@set@catcode\{\texttt{token}\}\{12\}}.

\texttt{\textbackslash cnltx@make@active\{\texttt{token}\}}

Introduced in version /zero.taboldstyle./one.taboldstyle/one.taboldstyle

A wrapper for \texttt{\textbackslash cnltx@set@catcode\{\texttt{token}\}\{13\}}.

Token List Manipulation

\texttt{\textbackslash cnltx@replace@once\{\texttt{cs}\}\{\texttt{search}\}\{\texttt{replace}\}}

Replaces the first occurrence of \texttt{\{search\}} in the first expansion of \texttt{\{cs\}} with \texttt{\{replace\}}.

\texttt{\textbackslash cnltx@greplace@once\{\texttt{cs}\}\{\texttt{search}\}\{\texttt{replace}\}}

Introduced in version /zero.taboldstyle./nine.taboldstyle

The same as \texttt{\textbackslash cnltx@replace@once} but acts globally.

\texttt{\textbackslash cnltx@replace@all\{\texttt{cs}\}\{\texttt{search}\}\{\texttt{replace}\}}

Replaces all occurrences of \texttt{\{search\}} in the first expansion of \texttt{\{cs\}} with \texttt{\{replace\}}.

\texttt{\textbackslash cnltx@greplace@all\{\texttt{cs}\}\{\texttt{search}\}\{\texttt{replace}\}}

 Introduced in version /zero.taboldstyle./nine.taboldstyle

The same as \texttt{\textbackslash cnltx@replace@all} but acts globally.

\texttt{\textbackslash cnltx@remove@once\{\texttt{cs}\}\{\texttt{search}\}}

Removes the first occurrence of \texttt{\{search\}} in the first expansion of \texttt{\{cs\}}.

\texttt{\textbackslash cnltx@gremove@once\{\texttt{cs}\}\{\texttt{search}\}}

Introduced in version /zero.taboldstyle./nine.taboldstyle

The same as \texttt{\textbackslash cnltx@remove@once} but acts globally.

\texttt{\textbackslash cnltx@remove@all\{\texttt{cs}\}\{\texttt{search}\}}

Removes all occurrences of \texttt{\{search\}} in the first expansion of \texttt{\{cs\}}.

\texttt{\textbackslash cnltx@gremove@all\{\texttt{cs}\}\{\texttt{search}\}}

Introduced in version /zero.taboldstyle./nine.taboldstyle

The same as \texttt{\textbackslash cnltx@remove@all} but acts globally.
A. Internal Helper Commands

**Miscellaneous**

* \texttt{\textbackslash cnltx@par}  
  Expands to \texttt{\par}. Sometimes you need to smuggle a \texttt{\par} in a short macro …

* \texttt{\textbackslash cnltx@stripbs}  
  A shortcut for \texttt{\expandafter\@gobble\string}.

\texttt{\textbackslash cnltxat}  
Robust command that typesets ’@’ with category code 11. An @ in command names confuses the indexing of the command names. Either one uses another symbol for makeindex’s “actual” recognition and also tells idxcmds [Nie13c] about it or one uses \texttt{\cnltxat} in \texttt{\cs} and friends. For the sake of convenience you can define a command like \texttt{\at} that expands to it.\textsuperscript{24} In order not to overwrite any such existing macro it is not defined by \texttt{C\textsc{nltx-Example}}. This document for example defines \texttt{\def\at{\cnltxat}}.

\texttt{\textbackslash cnltxletterat}  
An alias of \texttt{\cnltxat}.

\texttt{\textbackslash cnltxotherat}  
The same as \texttt{\cnltxat} but with a ’@’ with category code 12.

\texttt{\textbackslash cnltxbang}  
The same as \texttt{\cnltxotherat} except that it contains a ’!’.  

\texttt{\textbackslash cnltxequal}  
The same as \texttt{\cnltxotherat} except that it contains a ’=’.  

**A.2. Defined by \texttt{C\textsc{nltx-doc}}**

\texttt{\textbackslash cnltx@doc@error{\langle message\rangle}}  
Issue an error message using \texttt{\ClassError{\textsc{cnltx-doc}}}.

\texttt{\textbackslash cnltx@doc@warning{\langle message\rangle}}  
Issue a warning message using \texttt{\ClassWarning{\textsc{cnltx-doc}}}.

\texttt{\textbackslash cnltx@doc@warningnoline{\langle message\rangle}}  
Issue a warning message using \texttt{\ClassWarningNoLine{\textsc{cnltx-doc}}}.

\texttt{\textbackslash cnltx@doc@info{\langle message\rangle}}  
Issue a message using \texttt{\ClassInfo{\textsc{cnltx-doc}}}.

\texttt{\textbackslash cnltx@getfileinfo{\langle file name\rangle}{\langle file extension\rangle}}  
Extract the date, version and background information for a package or a class and defines \texttt{\cnltx@package@date}, \texttt{\cnltx@package@version} and \texttt{\cnltx@package@info} to contain the extracted data.

\textsuperscript{24} This is important. If you \texttt{\let} it to \texttt{\cnltxat} index entries may be sorted differently! Remember: \texttt{\cnltxat} is robust.
A. Internal Helper Commands

\cnltx@version@note
{\langle note \rangle}  
Command that is used for the versioning notes internally. Sets \reversemarginpar and then writes the note \langle note \rangle to the margin with corresponding formatting.

\begin{cnltxlist}
The list environment that is used by the environments commands, options and environments.

A.3. Defined by \texttt{\textbackslash cnltx-example}

\cnltx@example@error\{\langle message \rangle\}
Issue an error message using \texttt{PackageError\{cnltx-example\}}.

\cnltx@example@warning\{\langle message \rangle\}
Issue a warning message using \texttt{PackageWarning\{cnltx-example\}}.

\cnltx@example@warningnoline\{\langle message \rangle\}
Issue a warning message using \texttt{PackageWarningNoLine\{cnltx-example\}}.

\cnltx@example@info\{\langle message \rangle\}
Issue a message using \texttt{PackageInfo\{cnltx-example\}}.

\cnltx@isvalue
Used in definitions of the key/value option typesetting commands. Inserts a = with some stretchable space around and a legal break-point after it.

\indexcs
Version of \texttt{\textbackslash csidx} that takes care of a \texttt{\textbackslash textcompwordmark} inserted by listings. Also replaces all occurrences of @ with category code 11 or 12 with \texttt{\textbackslash cnltxat}. Used to index commands in the sourcecode and example environments that have been added with add-cmds.

\indexenv
Version of \texttt{\textbackslash envidx} that takes care of a \texttt{\textbackslash textcompwordmark} inserted by listings. Also replaces all occurrences of @ with category code 11 or 12 with \texttt{\textbackslash cnltxat}. Used to index environments in the sourcecode and example environments that have been added with add-envs.

\cnltx@treat@lst@index\{\langle new index cs \rangle\}\{\langle internal index cs \rangle\}
This command was used to define \texttt{\indexcs} and \texttt{\indexenv}:
\texttt{\cnltx@treat@lst@index\{\textbackslash indexcs\}\{\textbackslash csidx\}}

\texttt{\textbackslash MakePercentComment}
Sets the category code of % to 14.

\cnltx@copyablespace
Prints a space that is also copyable. Uses the accsupp package by Heiko Oberdiek [Obe10].

\cnltx@mdframed@options
Predefined option list for the mdframed [Dan13] style \texttt{\textbackslash cnltx}.

\cnltx@listings@style
Predefined option list for the listings [HM13] style \texttt{\textbackslash cnltx}.
A. Internal Helper Commands

### A.4. Defined by \texttt{CnLTX-LISTINGS}

\texttt{\textbackslash{}cnltx@listing@error\{⟨message⟩\}}

Issue an error message using \texttt{PackageError\{cnltx-listings\}}.

\texttt{\textbackslash{}cnltx@listing@warning\{⟨message⟩\}}

Issue a warning message using \texttt{PackageWarning\{cnltx-listings\}}.

\texttt{\textbackslash{}cnltx@listing@warningnoline\{⟨message⟩\}}

Issue a warning message using \texttt{PackageWarningNoLine\{cnltx-listings\}}.

\texttt{\textbackslash{}cnltx@listing@info\{⟨message⟩\}}

Issue a message using \texttt{PackageInfo\{cnltx-listings\}}.

\texttt{\textbackslash{}cnltx@predefined@control@sequences}

A comma-separated list of predefined ‘silent’ control sequence names.

\texttt{\textbackslash{}cnltx@predefined@environments}

A comma-separated list of predefined ‘silent’ environment names.

\texttt{\textbackslash{}listsilentcmds}

Prints all known control sequence names formatted and separated with the separator set with \texttt{list-sep}. Requires \texttt{CnLTX-EXAMPLE}.

\texttt{\textbackslash{}listsilentenvs}

Prints all known environment names formatted and separated with the separator set with \texttt{list-sep}. Requires \texttt{CnLTX-EXAMPLE}.

\texttt{\textbackslash{}listbibfilekeys\{⟨file name⟩\}}

Prints all cite keys contained in the bibliography file \texttt{⟨file name⟩} formatted with \texttt{\code} and separated with the separator set with \texttt{list-sep}. Requires \texttt{CnLTX-EXAMPLE}.

\texttt{\textbackslash{}listbibfiletypes\{⟨file name⟩\}}

Prints all citation types contained in the bibliography file \texttt{⟨file name⟩} formatted with \texttt{\code} and separated with the separator set with \texttt{list-sep}. Requires \texttt{CnLTX-EXAMPLE}.

\texttt{\textbackslash{}listbibfileentries\{⟨file name⟩\}}

Prints all cite keys contained in the bibliography file \texttt{⟨file name⟩} formatted with \texttt{\code} and gives their respective entry types, separated with the separator set with \texttt{list-sep}. Requires \texttt{CnLTX-EXAMPLE}.

\texttt{\textbackslash{}list-sep = \{⟨separator⟩\}}

Default: \texttt{, \textbackslash{}space}

Sets the separator for \texttt{CnLTX-LISTINGS’} commands listing the different commands \textit{etc}.

### A.5. Defined by \texttt{CnLTX-TOOLS}

\texttt{\textbackslash{}cnltx@tools@error\{⟨message⟩\}}

Issue an error message using \texttt{PackageError\{cnltx-tools\}}.
\textbf{B. List of Known \LaTeX{} Control Sequences}

\texttt{\textbackslash cnltx@tools\textbackslash warning\{\textit{message}\}}

Issue a warning message using \texttt{\textbackslash PackageWarning\{cnltx-tools\}}.

\texttt{\textbackslash cnltx@tools\textbackslash warningno\textbackslash line\{\textit{message}\}}

Issue a warning message using \texttt{\textbackslash PackageWarningNoLine\{cnltx-tools\}}.

\texttt{\textbackslash cnltx@tools\textbackslash info\{\textit{message}\}}

Issue a message using \texttt{\textbackslash PackageInfo\{cnltx-tools\}}.

\texttt{\textbackslash cnltx@accsupp\{\textit{actual text}\}\{\textit{additional options}\}\{\textit{\LaTeX{} text}\}}

A wrapper for package accsupp's
\texttt{\textbackslash BeginAccSupp\{ActualText = \{actual text\}\}\{\LaTeX{} text\}\EndAccSupp{}}.

\textbf{B. List of Known \LaTeX{} Control Sequences}

Below all \textit{predefined} control sequence names are listed that are treated as “silent” names by \texttt{\textbackslash cnltx}, that is, those defined by \texttt{\textbackslash CNLTX-LISTINGS}.
B. List of Known \TeX Control Sequences
B. List of Known \LaTeX Control Sequences

\fontfamily, \fontname, \fontsize, \fontshape, \fontseries, \fontspec,
\fontsize, \fontspec, \fontsubfuzz, \footins, \footline, \footnote,
\footnotemark, \footnoterule, \footnotetext, \footskip, \forall,
\foralllist, \frown, \fracc, \frame, \framebox,
\framed, \framedsp, \frontheading, \frowndate, \frowndatecommand,
\frowndatetime, \frowndateformat, \frowndigit,
\frowndigitcommand, \frowndigitformat, \frowndigitmacro,
\frownduration, \frowndurationcommand, \frowndurationformat,
\frowndurationmacro, \frowndurationunit, \frowndurationunitcommand,
\frowndurationunitformat, \frowndurationunitmacro,
\frowndurationunitprefix, \frowndurationunitprefixcommand,
\frowndurationunitprefixformat, \frowndurationunitprefixmacro,
\frowndurationunitsuffix, \frowndurationunitsuffixcommand,
\frowndurationunitsuffixformat, \frowndurationunitsuffixmacro,
\frowndurationunitsuffixprefix, \frowndurationunitsuffixprefixcommand,
\frowndurationunitsuffixprefixformat, \frowndurationunitsuffixprefixmacro,
\frowndurationunitsuffixsuffix, \frowndurationunitsuffixsuffixcommand,
\frowndurationunitsuffixsuffixformat, \frowndurationunitsuffixsuffixmacro,
\frowndurationunitsuffixsuffixprefix, \frowndurationunitsuffixsuffixprefixcommand,
\frowndurationunitsuffixsuffixprefixformat, \frowndurationunitsuffixsuffixprefixmacro,
\frowndurationunitsuffixsuffixsuffix, \frowndurationunitsuffixsuffixsuffixcommand,
\frowndurationunitsuffixsuffixsuffixformat, \frowndurationunitsuffixsuffixsuffixmacro,
\frowndurationunitsuffixsuffixsuffixprefix, \frowndurationunitsuffixsuffixsuffixprefixcommand,
\frowndurationunitsuffixsuffixsuffixprefixformat, \frowndurationunitsuffixsuffixsuffixprefixmacro,
\frowndurationunitsuffixsuffixsuffixsuffix, \frowndurationunitsuffixsuffixsuffixsuffixcommand,
\frowndurationunitsuffixsuffixsuffixsuffixformat, \frowndurationunitsuffixsuffixsuffixsuffixmacro,
\frowndurationunitsuffixsuffixsuffixsuffixprefix, \frowndurationunitsuffixsuffixsuffixsuffixprefixcommand,
\frowndurationunitsuffixsuffixsuffixsuffixprefixformat, \frowndurationunitsuffixsuffixsuffixsuffixprefixmacro,
\frowndurationunitsuffixsuffixsuffixsuffixsuffix, \frowndurationunitsuffixsuffixsuffixsuffixsuffixcommand,
\frowndurationunitsuffixsuffixsuffixsuffixsuffixformat, \frowndurationunitsuffixsuffixsuffixsuffixsuffixmacro,
\frowndurationunitsuffixsuffixsuffixsuffixsuffixprefix, \frowndurationunitsuffixsuffixsuffixsuffixsuffixprefixcommand,
\frowndurationunitsuffixsuffixsuffixsuffixsuffixprefixformat, \frowndurationunitsuffixsuffixsuffixsuffixsuffixprefixmacro,
B. List of Known \TeX Control Sequences
C. List of Known \LaTeX\ Environments

Below all predefined environment names are listed that are treated as “silent” names by \texttt{cmltx}, that is, those defined by \texttt{cmltx-listings}.

\begin{itemize}
\item array, center, description, \hfill itemize, labeling, list, \hfill sloppypar, tabbing, table, \hfill sloppypar, tabbing, table, \hfill sloppypar, tabbing, table,
\item displaymath, document, \hfill longtable, \texttt{lrbox}, math, \hfill tabu, tabular, tabularx, \hfill tabular, tabular, tabularx, \hfill tabular, tabular, tabularx,
\item enumerate, \texttt{eqnarray}, equation, \hfill minipage, otherlanguage, \hfill tabularx, \texttt{trivlist}, verbatim \hfill tabularx, \texttt{trivlist}, verbatim,
\item figure, flushleft, flushright, \hfill picture, quote, quoting, \hfill picture, quote, quoting,
\end{itemize}

D. List of Entries in \texttt{cmltx.bib}

Most entries in \texttt{cmltx.bib} are entries of the \texttt{@package} type. The cite keys that the file currently contains are listed below. This list is very likely to be extended significantly in the future.

\begin{itemize}
\item \texttt{pkg:abbrevs (@package)}, \hfill \texttt{pkg:answers (@package)}, \hfill \texttt{pkg:bpchem (@package)},
\item \texttt{pkg:accsupp (@package)}, \hfill \texttt{pkg:array (@package)}, \hfill \texttt{pkg:catchfile (@package)},
\item \texttt{pkg:acro (@package)}, \hfill \texttt{pkg:asymptote (@package)}, \hfill \texttt{pkg:checomponents (@package)},
\item \texttt{pkg:acromake (@package)}, \hfill \texttt{pkg:babel (@package)}, \hfill \texttt{pkg:chemcono (@package)},
\item \texttt{pkg:acronym (@package)}, \hfill \texttt{pkg:bm (@package)}, \hfill \texttt{pkg:chemfig (@package)},
\item \texttt{pkg:acroterm (@package)}, \hfill \texttt{pkg:biblatex (@package)}, \hfill \texttt{pkg:chemformula (@package)},
\item \texttt{pkg:adjustbox (@package)}, \hfill \texttt{pkg:bigfoot (@package)}, \hfill \texttt{pkg:chemgreek (@package)},
\item \texttt{pkg:amsmath (@package)}, \hfill \texttt{pkg:booktabs (@package)}, \hfill \texttt{pkg:chemmacros (@package)},
\end{itemize}
E. Bibliography

url: http://mirror.ctan.org/macros/latex/contrib/ulem/.

url: http://mirror.ctan.org/macros/latex/contrib/mdframed/.

url: http://mirror.ctan.org/macros/latex/contrib/imakeidx/.
E. Bibliography

url: http://mirror.ctan.org/macros/latex/contrib/listings/.

url: http://mirror.ctan.org/macros/latex/contrib/xcolor/.


url: http://mirror.ctan.org/macros/latex/contrib/marginnote/.

url: http://mirror.ctan.org/macros/latex/contrib/etoolbox/.

url: http://mirror.ctan.org/macros/latex/contrib/biblatex/.

url: http://mirror.ctan.org/macros/latex-required/tools/multicol/.

url: http://mirror.ctan.org/macros/latex/contrib/acro/.

[Nie13b] Clemens Niederberger. cnpkgdoc. version 0.3a, July 13, 2013.
url: https://bitbucket.org/cgnieder/cnpkgdoc/.

[Nie13c] Clemens Niederberger. idxcmds. version 0.2b, Aug. 31, 2013.
url: http://mirror.ctan.org/macros/latex/contrib/idxcmds/.

url: http://mirror.ctan.org/macros/latex/contrib/translations/.

url: http://mirror.ctan.org/macros/latex/contrib/exsheets/.

url: http://mirror.ctan.org/macros/latex/contrib/oberdiek/.

url: http://mirror.ctan.org/macros/latex/contrib/oberdiek/.

url: http://mirror.ctan.org/macros/latex/contrib/oberdiek/.

url: http://mirror.ctan.org/macros/latex/contrib/oberdiek/.

url: http://mirror.ctan.org/macros/latex/contrib/hyperref/.

E. Bibliography

        url: http://mirror.ctan.org/macros/latex/contrib/trimspaces/.

        url: http://mirror.ctan.org/macros/latex/contrib/ifxetex/.

        url: http://mirror.ctan.org/macros/latex/contrib/ms/ragged2e/.

        url: http://mirror.ctan.org/macros/latex/contrib/adjustbox/.

        url: http://mirror.ctan.org/macros/latex/contrib/glossaries/.

[Wil09] Peter Wilson, current maintainer: Will Robertson.
        chngcntr. version 1.0a, Sept. 2, 2009.
        url: http://mirror.ctan.org/macros/latex/contrib/chngcntr/.

        url: http://mirror.ctan.org/macros/latex/contrib/pgf.opts/.
### F. Index

| A | abstract. | 37, 39f |
|   | abstract-format | 37 |
|   | abstract-width | 37 |
|   | accsupp (package) | 5, 59, 61 |
|   | acro (package) | 31 |
|   | acronym-format | 32 |
|   | \ad | 35 |
|   | add-bib | 5, 42, 45 |
|   | add-cmds | 19f, 46, 59 |
|   | add-props | 20, 46, 59 |
|   | add-frame | 21 |
|   | add-frame-options | 19, 45 |
|   | add-index | 4, 38, 42, 45 |
|   | add-listings-options | 20, 46 |
|   | add-local-frame | 19 |
|   | add-silent-cmds | 20, 46 |
|   | add-silent-envs | 20, 46 |
|   | add-sourcecode-options | 20 |
|   | adjustbox (package) | 5 |
|   | after-code | 12 |
|   | after-output | 13 |
|   | \am | 35 |
|   | arg-format | 37 |
|   | argumentformat | 36 |
|   | Arsenau, Donald | 5 |
|   | authors | 39 |
| B | babel (package) | 33, 38 |
|   | babel-options | 38 |
|   | bar | 16 |
|   | baz | 36, 38 |
|   | \bc | 35 |
|   | biblatex (package) | 3f, 43f, 52 |
|   | \boolkey | 9 |
|   | Boruyka, Audrey | 43 |
|   | build-title | 39 |
| C | caption | 23 |
|   | caption-font | 37 |
|   | caption-label-font | 37 |
|   | Carlisle, David | 32 |
|   | catchfile (package) | 5 |
|   | \cf | 34 |
|   | \changedversion | 9 |
|   | chngcntn (package) | 45 |
|   | \choicekey | 9 |
|   | \choices | 9 |
|   | class | 38f |

\begin{tabular}{ll}
\textbf{classformat} & \hspace{1cm} 36 \\
\textbf{cls} & \hspace{1cm} 10, 49 \\
\textbf{cls-format} & \hspace{1cm} 37 \\
\textbf{clsidx} & \hspace{1cm} 10 \\
\textbf{\cnltx@date} & \hspace{1cm} 54 \\
\textbf{\cnltx@info} & \hspace{1cm} 54 \\
\textbf{\cnltx@version} & \hspace{1cm} 54 \\
\textbf{\cnltx@accsupp} & \hspace{1cm} 61 \\
\textbf{\cnltx@babeloptions} & \hspace{1cm} 42 \\
\textbf{\cnltx@baseerror} & \hspace{1cm} 54 \\
\textbf{\cnltx@baseinfo} & \hspace{1cm} 54 \\
\textbf{\cnltx@basewarning} & \hspace{1cm} 54 \\
\textbf{\cnltx@basewarninginline} & \hspace{1cm} 54 \\
\textbf{\cnltx@biblatexlistings@style} & \hspace{1cm} 40 \\
\textbf{\cnltx@captionfont} & \hspace{1cm} 42 \\
\textbf{\cnltx@captionlabelfont} & \hspace{1cm} 42 \\
\textbf{\cnltx@copyable@space} & \hspace{1cm} 59 \\
\textbf{\cnltx@create@bundle@message} & \hspace{1cm} 54 \\
\textbf{\cnltx@create@generic@message} & \hspace{1cm} 55 \\
\textbf{\cnltx@create@message} & \hspace{1cm} 54 \\
\textbf{\cnltx@defaultcolorscheme} & \hspace{1cm} 54 \\
\textbf{\cnltx@docerror} & \hspace{1cm} 58 \\
\textbf{\cnltx@docinfo} & \hspace{1cm} 58 \\
\textbf{\cnltx@docwarning} & \hspace{1cm} 58 \\
\textbf{\cnltx@docwarninginline} & \hspace{1cm} 58 \\
\textbf{\cnltx@examplemessage} & \hspace{1cm} 59 \\
\textbf{\cnltx@exampleerror} & \hspace{1cm} 59 \\
\textbf{\cnltx@exampleinfo} & \hspace{1cm} 59 \\
\textbf{\cnltx@examplewarning} & \hspace{1cm} 59 \\
\textbf{\cnltx@examplewarninginline} & \hspace{1cm} 59 \\
\textbf{\cnltx@examplewarninginline} & \hspace{1cm} 59 \\
\textbf{\cnltx@expandargs} & \hspace{1cm} 56 \\
\textbf{\cnltx@getfileinfo} & \hspace{1cm} 58 \\
\textbf{\cnltx@gobble} & \hspace{1cm} 46, 48 \\
\textbf{\cnltx@gremoveall} & \hspace{1cm} 57 \\
\textbf{\cnltx@gremoveonce} & \hspace{1cm} 57 \\
\textbf{\cnltx@replaceall} & \hspace{1cm} 57 \\
\textbf{\cnltx@replaceonce} & \hspace{1cm} 57 \\
\textbf{\cnltx@ifin} & \hspace{1cm} 56 \\
\textbf{\cnltx@ifbang} & \hspace{1cm} 55 \\
\textbf{\cnltx@ifcounter} & \hspace{1cm} 55 \\
\textbf{\cnltx@ifdash} & \hspace{1cm} 55 \\
\textbf{\cnltx@ifinlist} & \hspace{1cm} 56 \\
\textbf{\cnltx@ifinlistcs} & \hspace{1cm} 56 \\
\textbf{\cnltx@ifisnum} & \hspace{1cm} 41, 55 \\
\textbf{\cnltx@ifnextchars} & \hspace{1cm} 55 \\
\textbf{\cnltx@ifpunctuation} & \hspace{1cm} 27f. \\
\textbf{\cnltx@ifshell escalated} & \hspace{1cm} 56 \\
\textbf{\cnltx@ifstrequal} & \hspace{1cm} 56 \\
\textbf{\cnltx@ifsym} & \hspace{1cm} 55 \\
\textbf{\cnltx@isvalue} & \hspace{1cm} 59 \\
\textbf{\cnltx@listings@error} & \hspace{1cm} 60 \\
\end{tabular}
| G | glossaries (package) | 31 |
|   | gobble               | 19 |
|   | graphics             | 23, 25 |
|   | Gregorio, Enrico     | 42 |
| H | Heinz, Carsten       | 3, 5, 19 f., 46, 59 |
|   | Hoffmann, Jobst      | 3, 5, 19 f., 46, 59 |
|   | hyperref (package)   | 5, 49 |
| I | idxcmds (package)    | 5, 58 |
|   | \ie                   | 34 |
|   | \iftest               | 55 f. |
|   | ifxetex (package)     | 5 |
|   | imakeidx (package)    | 42 f. |
|   | \implementation       | 11 |
|   | index-prologue        | 43 |
|   | index-setup           | 43 |
|   | index-space           | 43 |
|   | index-style           | 4.4.3 |
|   | \indexcs              | 47, 59 |
|   | \indexenv             | 59 |
|   | \info                 | 39 |
|   | \inputexample         | 11 |
|   | \inputsidebyside      | 11 |
|   | \inputsourcecode      | 11 |
| K | Kern, Uwe            | 5 |
|   | key                  | 9, 49 |
|   | \key                  | 15 f. |
|   | \keychoice            | 15 f. |
|   | \keyis                | 9 |
|   | \keyval               | 15 f. |
|   | Kime, Philip          | 43 |
|   | Kohm, Markus          | 5 |
|   | KOMA-Script (bundle)  | 5 |
| L | last-name-format      | 32 |
|   | Lehman, Philipp       | 5, 43 |
|   | \license              | 10 |
|   | \listsep              | 60 |
|   | \listsetup            | 12 |
|   | \listbibfileentries   | 60 |
|   | \listbibfilekeys      | 60 |
|   | \listbibliotypes      | 60 |
|   | listings (package)    | 3, 5, 19 f., 46 f., 59 |
|   | listings-options      | 20, 46 |
|   | \listsilentcmds       | 60 |
|   | \allsilentenvs        | 60 |
| M | makeindex-set         | 43 |
|   | \MakePercentComment   | 59 |
|   | \Marg                 | 8 |
|   | \margin               | 8, 14 |
|   | \marginmode           | 5 |
|   | \maxheight            | 23 |
|   | \maxpages             | 23 |
|   | \mdframed (environment)| 19 |
|   | \mdframed (package)   | 3, 5, 18 f., 45, 59 |
|   | \meta                 | 8, 36 f. |
|   | MITTELBACH, Frank      | 5 |
|   | \Module               | 15 f. |
|   | \module               | 8, 15 f., 50 f. |
|   | module-sep            | 37 |
|   | moduleidx             | 8 |
|   | Moses, Brooks          | 3, 5, 19 f., 46, 59 |
|   | multicol (package)     | 5 |
| N | name                  | 8, 38 f. |
|   | \name                 | 31 f. |
|   | name-format            | 32 |
|   | \needclass            | 10 |
|   | \needpackage          | 10 |
|   | Neukamm, Frank         | 5 |
|   | \newabbr              | 31, 33 |
|   | \newarg               | 8 |
|   | \newcountrepresentation | 29 f. |
|   | \newexpandablecmd      | 30 f. |
|   | \newinputsourcefilecmd | 11 |
|   | \newsourcecodeenv     | 9, 31 f., 40 |
|   | \newnote              | 9 |
|   | \newpackagename       | 11 |
|   | \newsourcecodeenv     | 13 |
|   | Niederberger, Clemens  | 3, 5 f., 31, 52, 58 |
|   | \nottest              | 55 |
| O | \aarg                 | 8 |
|   | \aarg                 | 8, 14, 17 |
|   | Oberdiek, Heiko       | 5, 49, 59 |
|   | \opt                  | 13, 15 f. |
|   | \option               | 8, 14 f., 36, 38, 49 f. |
|   | \optionidx            | 8 |
INDEX

options (environment) .................. 12, 15 ff., 59
outside .................................. 18
side-by-side (environment) ............ 11 ff., 19
\sinceversion .......................... 9
source-format .......................... 37
source-code (environment) ............ 11 ff., 18, 59
source-code-options ................... 20
sourceformat .......................... 36, 46, 48
subtitle ................................ 39

P
package ................................ 38 f.
\packageformat .......................... 36
pages .................................. 24
PDF ..................................... 21, 24, 31, 49
pdfLaTeX (program) ..................... 21
pdftexcmds (package) ................. 5, 56
pgopts (package) ....................... 5
\pkg .................................. 10, 32, 49
pkg-format ................................ 37
\pkgidx ................................ 10
PM ...................................... 35
pre-code ................................ 12
pre-output ................................ 13
program .................................. 21
\providecounterrepresentation ........ 29
\provideexpandablecmd ............... 31
Q
quote-author-format .................... 40
quote-format ............................ 40

R
raggedeze (package) .................... 5
Rahtz, Sebastian ...................... 5, 49
\renewabbr ......................... 5, 49
\renewcounterrepresentation .......... 31
\renewexpandablecmd.................. 29
\renewTranslation ................. 31
Robertson, Will ...................... 5, 45
runs .................................. 21
sideby-side ................................ 18 ff.
sideby-side (environment) ............ 11 ff., 19
\sinceversion .......................... 9
source-format .......................... 37
source-code (environment) ............ 11 ff., 18, 59
source-code-options ................... 20
sourceformat .......................... 36, 46, 48
subtitle ................................ 39

T
Talbot, Nicola ......................... 31
textcomp (package) ..................... 5
title-format ........................... 37
\titleformat ........................... 40
translations (package) ............... 5, 52
trimspaces (package) .................. 5

U
ulem (package) ......................... 5
\unexpandable ... .................. 13, 51 ff.
\unexpandablesymbol .................. 14
url ..................................... 39 f.
\usf .................................. 34 f.
\usw .................................. 34 f.
\uswusf ................................ 34

V
\verbcode ................................ 7
version .................................. 39
\version-note-format .................. 37
\versionnoteformat ................... 36
\vgl .................................. 35
\vgl .................................. 35

W
\website ................................ 10
Wilson, Peter .......................... 45
Wright, Joseph ......................... 5, 43

X
xcolor (package) ....................... 5, 50

Z
\ZB .................................... 35
\ZB .................................... 34 f.