

THE CNLTX BUNDLE

Documentation for L^AT_EX 2_ε Packages or Classes

v0.7a 2013/10/04

L^AT_EX tools and documenting facilities the CN way

Clemens 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 *etc.* Not at least it provides a number of programming tools.

Table of Contents

I. About The Bundle	3	5.2. Versioning Commands, Licensing and Related Stuff . . .	8
1. Background	3	5.3. Input Source Code Files	9
2. Bundled Packages, Classes and Files	4	6. Available Environments	10
3. License and Requirements	5	6.1. Description Environments . .	10
		6.2. Source Code Environments . .	11
		7. Usage of the Various Functions	12
II. Details of Available Commands, Environments and Options	6	7.1. Command Descriptions	12
		7.2. Option Descriptions	13
4. Options and Setup	6	7.3. Environment Descriptions . .	15
		7.4. Code Examples	16
5. Available Commands	6	7.5. Example File	18
5.1. Description of Macros, Environments and Options	6	7.6. Additional Functionality Provided by CNLTX-BASE	18
		7.6.1. Looking for Trailing Punctuation	19

7.6.2.	Counter Representation Commands	20	10.2.	listings	36
7.6.3.	Expandable Document Commands	22	10.2.1.	L ^A T _E X Sourcecode	36
7.7.	Additional Functionality Provided by CNLTX-TOOLS	22	10.2.2.	B _B T _E X Entries	38
7.7.1.	Commands for Defining Different Document Macros	22	10.2.3.	makeindex Style Files	39
7.7.2.	Defining Abbreviations	24	11. PDF Strings and hyperref		39
7.7.3.	Predefined Abbreviations	24	12. Predefined Colors and Color-Schemes		40
8. Formatting Possibilities		26	12.1.	Explicitly Defined Colors	40
8.1.	Formatting by Redefining Hooks	27	12.2.	Actual Used Color Names and Color Schemes	40
8.2.	Formatting by Setting Options	27	13. Language Support		42
9. Commands, Options and Further Settings Directly Related to the CNLTX-DOC Class		29	III. Appendix		42
9.1.	Using Class Options	29	A. Internal Helper Commands		42
9.2.	Information on the Described Package or Class	29	A.1.	Defined by CNLTX-BASE	44
9.3.	Building of the Manuals Title Page	30	A.1.1.	Related to the Bundle	44
9.4.	A Quotation Environment	30	A.1.2.	Programming Tools	44
9.5.	Predefined Preamble	32	A.2.	Defined by CNLTX-DOC	46
9.6.	Predefined Indexing	33	A.3.	Defined by CNLTX-EXAMPLE	47
9.7.	Bibliography with biblatex	34	A.4.	Defined by CNLTX-LISTINGS	48
9.7.1.	Bibliography Entry Types package, class and bundle for biblatex	34	A.5.	Defined by CNLTX-TOOLS	49
9.7.2.	Automatic Bibliography	35	B. List of Known L^AT_EX Control Sequences		49
10. Predefined listings and mdframed Styles		36	C. List of Known L^AT_EX Environments		55
10.1.	mdframed	36	D. List of Entries in cnltx.bib		55
			E. Bibliography		55
			F. Index		58

Part I.

About The Bundle

1. Background

The **CNLT_X** bundle contains different packages and classes.¹ I developed it as a successor of my class `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 **CNLT_X-DOC** loads *all* packages of the bundle which makes it more feature-rich than `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 **CNLT_X-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 **CNLT_X-BASE** will get extended in the future.

The bundle provides listings style for L^AT_EX code, bibliography database files and index style files. It provides a biblatex citation and bibliography style closely linked to **CNLT_X-DOC**. It provides a bibliography database file containing many L^AT_EX 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 50 pages long.

The most detailed documentation for the bundle is as always the source code of the `sty` and `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 `docstrip` and don't write `dtx` files but always write the `sty` or `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 `mdframed` package by Marco Daniel [Dan13], loaded with the option `framemethod = tikz`.

1. Well, *one* class for the time being,

2. Bundled Packages, Classes and Files

Besides all this I included some other ideas of mine in this bundle which are all provided by **CNLT_X-TOOLS**. This includes a mechanism for defining clever abbreviations or macros that make it easy to index names the same way bibl_{at}ex does.

2. Bundled Packages, Classes and Files

The **CNLT_X** bundle currently bundles the following packages, classes and files:

- **CNLT_X-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 **CNLT_X-DOC** class. All other packages and classes of the **CNLT_X** bundle load this package. This package can be used stand-alone.
`\usepackage{cnltx-base}`
- **CNLT_X-DOC** – a class for writing package manuals. Loads **CNLT_X-EXAMPLE** and **CNLT_X-TOOLS** and implicitly all other files of the bundle.
`\documentclass{cnltx-doc}`
- **CNLT_X-EXAMPLE** – a package that defines macros and environments for describing control sequences and options and for including source code. Loads **CNLT_X-LISTINGS**. This package can be used stand-alone.
`\usepackage{cnltx-example}`
- **CNLT_X-LISTINGS** – a package that defines the listings language ‘BibTeX’. Also defines a list of highlighted control sequence names and environment names, loaded by **CNLT_X-EXAMPLE**. The additional control sequence and environment names used to be defined in **CNLT_X-CSNAMES**. That package got removed and its contents are now provided by **CNLT_X-LISTINGS**. This package can be used stand-alone.
`\usepackage{cnltx-listings}`
- **CNLT_X-TOOLS** – a package that defines tools used by **CNLT_X-DOC** that are unrelated to L^AT_EX documentation *per se*. This package can be used stand-alone.
`\usepackage{cnltx-tools}`
- `cnltx.ist` – an index style file that is used when the option `add-index` for **CNLT_X-DOC** is activated and the option `index-style` is not used.
- `cnltx.bib` – a bibliography file that contains a small but growing number of package entries. Used by **CNLT_X-DOC** when the `add-bib` is used.
- `cnltx.bbx`, `cnltx.cbx` and `cnltx.dbx` – files related to the bibl_{at}ex style `cnltx`. The bibl_{at}ex style defined in those files is used when the `add-bib` for **CNLT_X-DOC** is used.

Introduced in
version 0.4

Introduced in
version 0.2

Introduced in
version 0.4

Introduced in
version 0.4

3. License and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the L^AT_EX Project Public License (LPPL), version 1.3 or later (<http://www.latex-project.org/lppl.txt>). The software has the status “maintained.”

The **CNLTX-BASE** package loads the following packages: pgfplots² [Wri11], etoolbox³ [Leh11], ltxcmds⁴ [Obe11b], trimspaces⁵ [Rob09] and xcolor⁶ [Kero7].

The **CNLTX-DOC** class loads the packages **CNLTX-BASE**, **CNLTX-EXAMPLE**, ulem⁷ [Ars11], translations [Nie13d], multicol⁸ [Mit11], ragged2e⁹ [Scho9], marginnote¹⁰ [Koh12] and hyperref¹¹ [OR12]. It is a wrapper class for the KOMA-Script class scrartcl¹² [KN12]. The class has the option `load-preamble` which when used will load additional packages, see section 9.5 on page 32 for details.

The **CNLTX-EXAMPLE** package loads the packages: **CNLTX-BASE**, **CNLTX-LISTINGS**, **CNLTX-TOOLS**, translations,¹³ mdframed¹⁴ [Dan13] and idxcmds¹⁵ [Nie13c].

The **CNLTX-LISTINGS** package loads the packages **CNLTX-BASE**, listings¹⁶ [HM13] and catchfile¹⁷ [Obe11a].

The **CNLTX-TOOLS** package loads the packages **CNLTX-BASE** and accsupp¹⁸ [Obe10].

All other packages that are loaded are loaded by the mentioned packages and not directly by any of the packages or classes of the **CNLTX** bundle. Like all of my packages **CNLTX** implicitly relies on an up to date T_EX distribution.

2. on CTAN as pgfplots: <http://mirrors.ctan.org/macros/latex/contrib/pgfplots/>
3. on CTAN as etoolbox: <http://mirrors.ctan.org/macros/latex/contrib/etoolbox/>
4. on CTAN as oberdiek: <http://mirrors.ctan.org/macros/latex/contrib/oberdiek/>
5. on CTAN as trimspaces: <http://mirrors.ctan.org/macros/latex/contrib/trimspaces/>
6. on CTAN as xcolor: <http://mirrors.ctan.org/macros/latex/contrib/xcolor/>
7. on CTAN as ulem: <http://mirrors.ctan.org/macros/latex/contrib/ulem/>
8. on CTAN as multicol: <http://mirrors.ctan.org/macros/latex/required/tools/multicol/>
9. on CTAN as ragged2e: <http://mirrors.ctan.org/macros/latex/contrib/ms/ragged2e/>
10. on CTAN as marginnote: <http://mirrors.ctan.org/macros/latex/contrib/marginnote/>
11. on CTAN as hyperref: <http://mirrors.ctan.org/macros/latex/contrib/hyperref/>
12. on CTAN as koma-script: <http://mirrors.ctan.org/macros/latex/contrib/koma-script/>
13. on CTAN as translations: <http://mirrors.ctan.org/macros/latex/contrib/translations/>
14. on CTAN as mdframed: <http://mirrors.ctan.org/macros/latex/contrib/mdframed/>
15. on CTAN as idxcmds: <http://mirrors.ctan.org/macros/latex/contrib/idxcmds/>
16. on CTAN as listings: <http://mirrors.ctan.org/macros/latex/contrib/listings/>
17. on CTAN as catchfile: <http://mirrors.ctan.org/macros/latex/contrib/catchfile/>
18. on CTAN as accsupp: <http://mirrors.ctan.org/macros/latex/contrib/oberdiek/accsupp/>

Part II.

Details of Available Commands, Environments and Options

4. Options and Setup

The **CNLT**X bundle has a large number of options. The **CNLT**X-**DOC** class only knows a few options (described in section 9.1 on page 29) 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:

`\setcnltx{<options>}`

Setup command for the **CNLT**X bundle. This command is provided by **CNLT**X-**BASE**.

The source code environments defined by the **CNLT**X-**EXAMPLE** package also have optional arguments that can be used to set the options for the environment locally.

5. Available Commands

5.1. Description of Macros, Environments and Options

provided by
CNLTX-**EXAM**-
PLE

The commands described in this section all are provided by the **CNLT**X-**EXAMPLE** package. They all are related to the typesetting of provided macros, options and the like.

`\code{<arg>}`

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

Introduced in
version 0.2

`\verbcode<char><code><char>`

A verbatim command that uses the same formatting as the source code example environments, cf. section 7.4. This is a wrapper for `\lstinline` which loads the corresponding style.

`\cs*{<name>}`

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

`\csidx{<name>}`

Adds an index entry but does not typeset the control sequence `<name>`.

`\env*{<name>}`

Format the environment `<name>`, `\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.

`\envidx{<name>}`

Adds an index entry but does not typeset the environment `<name>`.

5. Available Commands

`\meta{⟨meta⟩}`

Description of an argument, `\meta{meta}`: $\langle meta \rangle$.

`\marg{⟨arg⟩}`

A mandatory argument. $\langle arg \rangle$ is formatted with `\meta` if it is not blank, `\marg{arg}`: $\{\langle arg \rangle\}$.

`\Marg{⟨arg⟩}`

A mandatory argument. $\langle arg \rangle$ is formatted with `\code` if it is not blank, `\Marg{arg}`: $\{arg\}$.

Introduced in
version 0.2

`\oarg{⟨arg⟩}`

An optional argument. $\langle arg \rangle$ is formatted with `\meta` if it is not blank, `\oarg{arg}`: $[\langle arg \rangle]$.

`\Oarg{⟨arg⟩}`

An optional argument. $\langle arg \rangle$ is formatted with `\code` if it is not blank, `\Oarg{arg}`: $[arg]$.

Introduced in
version 0.2

`\darg{⟨arg⟩}`

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

`\Darg{⟨arg⟩}`

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

Introduced in
version 0.2

`\sarg`

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

`\option*{⟨name⟩}`

An option $\langle name \rangle$, `\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 $\langle name \rangle$.

`\module*{⟨name⟩}`

A module $\langle name \rangle$, `\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 $\langle name \rangle$.

`\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}`: $key = \{\langle value \rangle\}$; `\key-{key}{value}`: $key = \langle value \rangle$

`\keyis*{⟨name⟩}{⟨value⟩}`

A key $\langle name \rangle$ set to value $\langle value \rangle$, the optional star prevents an index entry, `\key{keyis}{value}`: $key = value$.

Introduced in
version 0.2

5. Available Commands

`\choices{⟨clist of choices⟩}`

A list of choices, `\choices{one,two,three}`: one|two|three

`\choicekey{⟨name⟩}{⟨clist of choices⟩}`

A key `⟨name⟩` with a list of possible values, `\choicekey{key}{one,two,three}`: `key = one|two|three`

`\boolkey{⟨name⟩}`

A boolean key `⟨name⟩` 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

5.2. Versioning Commands, Licensing and Related Stuff

provided by
CNLTX-DOC

The commands described in this section are provided by the `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.

Introduced in
version 0.0

`\changedversion{⟨version⟩}`

Gives a sidenote like the one on the left.

Changed in
version 0.0

`\newnote*{⟨cs⟩}[⟨num⟩][⟨optional⟩]{⟨definition⟩}`

Defines a note like `\sinceversion`. The syntax of the command is the same as the one of `\newcommand`. `\sinceversion` was defined as follows:

`\newnote*\sinceversion[1]{Introduced in version~#1}`

`\newpackagename{⟨cs⟩}{⟨name⟩}`

Define a command `⟨cs⟩` that prints `⟨name⟩` formatted like `CNLTX`, i. e. in small caps and colored with the color `cnltx` (see section 12.2).

`\lppl`

Typesets “LPPL” and adds a corresponding index entry.

`\LPPL`

Typesets “L^AT_EX 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 L^AT_EX 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`.

Changed in
version 0.2

`\ctan`

Typesets “CTAN” and adds a corresponding index entry.

5. Available Commands

`\CTAN`

Typesets “Comprehensive T_EX Archive Network” and adds the same index entry as `\ctan`.

`\pkg*{⟨package⟩}`

provided by **CNLT_X-EXAMPLE** Format the package name `⟨package⟩` and add an index entry. The starred variant adds nothing to the index.

`\pkgidx{⟨package⟩}`

provided by **CNLT_X-EXAMPLE** Add an index entry for the package `⟨package⟩`.

`\cls*{⟨class⟩}`

provided by **CNLT_X-EXAMPLE** Format the class name `⟨class⟩` and add an index entry. The starred variant adds nothing to the index.

`\clsidx{⟨class⟩}`

provided by **CNLT_X-EXAMPLE** Add an index entry for the class `⟨class⟩`.

`\CTANurl[⟨directory⟩]{⟨name⟩}`

Writes a CTAN link like the ones in section 3 on page 5 in the footnotes. The predefined directory is `macros/latex/contrib`. The link address will be:

`http://mirrors.ctan.org/⟨directory⟩/⟨name⟩/`.

`\needpackage[⟨directory⟩]{⟨name⟩}`

Introduced in version 0.2 A wrapper for `\pkg{#2}\footnote{\CTANurl[#1]{#2}}`

`\needclass[⟨directory⟩]{⟨name⟩}`

Introduced in version 0.2 A wrapper for `\cls{#2}\footnote{\CTANurl[#1]{#2}}`

```
1 \newpackagename{\foothree}{foo-3}%  
2 now \foothree\ looks like \cnltx.
```

now **FOO-3** looks like **CNLT_X**.

5.3. Input Source Code Files

Similar to the environments described in section 6.2 on page 11 **CNLT_X-EXAMPLE** provides a few commands for inputting source code files, formatting and printing the source code and inputting the file directly.

`\inputexample[⟨options⟩]{⟨file name⟩}`

The equivalent of the example environment, see section 6.2 on page 11.

`\inputsidebyside[⟨options⟩]{⟨file name⟩}`

The equivalent of the sidebyside environment, see section 6.2 on page 11.

6. Available Environments

`\inputsourcecode[⟨options⟩]{⟨file name⟩}`

The equivalent of the `sourcecode` environment, see section 6.2 on the following page.

`\implementation[⟨options⟩]{⟨file name⟩}`

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

Introduced in
version 0.5

It is possible to define further commands like this:

`\newinputsourcefilecmd[⟨option⟩]{⟨control sequence⟩}`

Defines `⟨control sequence⟩` as a new source code input command where `⟨options⟩` are preset.

The existing commands have been defined like this:

```
1 \newinputsourcefilecmd\inputexample
2 \newinputsourcefilecmd[side-by-side]\inputsidebyside
3 \newinputsourcefilecmd[code-only]\inputsourcecode
```

6. Available Environments

6.1. Description Environments

CNLTX-DOC defines some description environments used to describe macros, environments or options.

`\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 7.1 on page 12.

`\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 7.2 on page 13.

`\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 7.3 on page 15.

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

`\list-setup = {⟨definitions⟩}`

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

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

6.2. Source Code Environments

CNLTX-EXAMPLE defines the following environments that are used to display source code and possibly the output of the source code, too.

`\begin{example}[\langle options \rangle]`

This environment is a formatted verbatim environment that also inputs the output of the inputted code. This environment is described in section 7.4 on page 16.

`\begin{sidebyside}[\langle options \rangle]`

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 7.4 on page 16.

`\begin{sourcecode}[\langle options \rangle]`

This environment is a formatted verbatim environment. This environment is described in section 7.4 on page 16.

Introduced in
version 0.2

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

`pre-code = {\langle definitions \rangle}`

`\langle definitions \rangle` are placed before the source code is inserted.

`after-code = {\langle definitions \rangle}`

`\langle definitions \rangle` are placed after the source code is inserted.

`pre-source = {\langle definitions \rangle}`

`\langle definitions \rangle` are placed before the output of the source code is inserted.

`after-source = {\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:

```
1 \newsourcecodeenv{example}
2 \newsourcecodeenv[side-by-side]{sidebyside}
3 \newsourcecodeenv[code-only]{sourcecode}
```

7. Usage of the Various Functions

7.1. Command Descriptions

Inside of the environment commands that was introduced in section 6.1 on page 10 items are input via the following command:

`\command*{⟨name⟩}[⟨stuff after⟩]`

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*!{⟨code⟩}`

Changed in
version 0.3

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`

Introduced in
version 0.5

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`

Introduced in
version 0.5

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`

Introduced in
version 0.5

Default: `\textasteriskcentered`

The macro that holds the sign used by `\expandable` and `\unexpandable`.

```

1 \begin{commands}
2   \command{cs}
3   This is about foo bar baz.
4   \command{cs}[\marg{arg}]
5   This one has an argument.
6   \command{cs}[\sarg{oarg}{option}]
7   This has a star variant and an optional argument.
8   \command{cs}\Default{foo bar}
9   This one has the default replacement text \code{foo bar}
10  \expandable\command{cs}
11  This macro is expandable.
12 \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`

Default: `foo bar`

This one has the default replacement text `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:

`expandable-sign = {⟨definition⟩}`

Default: `\textasteriskcentered`

Redefines `\expandablesign` to `⟨definition⟩`.

Introduced in
version 0.5

7.2. 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.

Changed in
version 0.3

`\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

Introduced in
version 0.3

7. Usage of the Various Functions

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:

```
1 \begin{options}
2   \opt{foo}
3     This makes stuff. Let's add a few more words so that the line gets
4     filled and we can see how the output actually looks.
5   \opt*{foo}\Default{bar}
6     This makes stuff. Let's add a few more words so that the line gets
7     filled and we can see how the output actually looks.
8   \opt{foo}\Module{bar}
9     This option belongs to \module*{bar}. Let's add a few more words so
10    that the line gets filled and we can see how the output actually
11    looks.
12  \opt{foo}\Module{bar}\Default{baz}
13    This option belongs to \module*{bar}. Let's add a few more words so
14    that the line gets filled and we can see how the output actually
15    looks.
16  \keyval{foo}{bar}\Default
17    This makes stuff. Let's add a few more words so that the line gets
18    filled and we can see how the output actually looks.
19  \keyval{foo}{bar}\Default!
20    This makes stuff. Let's add a few more words so that the line gets
21    filled and we can see how the output actually looks.
22  \keyval*{foo}{bar}
23    This makes stuff. Let's add a few more words so that the line gets
24    filled and we can see how the output actually looks.
25  \keyval-{foo}{bar}
26    This makes stuff. Let's add a few more words so that the line gets
27    filled and we can see how the output actually looks.
28  \keychoice{foo}{one,two,three}
29    This makes stuff. Let's add a few more words so that the line gets
30    filled and we can see how the output actually looks.
31  \keybool{foo}
32    This makes stuff. Let's add a few more words so that the line gets
33    filled and we can see how the output actually looks.
34 \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.

7. Usage of the Various Functions

`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.

`foo = {⟨bar⟩}` (required)
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 = {⟨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.

`foo = ⟨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.

`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.

`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.

7.3. Environment Descriptions

Environment descriptions are made – unsurprisingly – with the `environments` environment. It knows the command `\environment:`

`\environment*{⟨name⟩}[⟨stuff after⟩]`

This macro prints the environment name and puts a line break after it. The optional argument allows printing things directly after the environment name and can thus be used for adding arguments.

```

1 \begin{environments}
2   \environment*{foobar}[\oarg{options}]
3   This is environment \env*{foobar}. The star prevents it from being
4   added to the index.
5 \end{environments}

```

```

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

```

7.4. 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.

```

1 \begin{example}
2   a \LaTeX\ code example
3 \end{example}

```

This example would give:

```

1 a \LaTeX\ code example

```

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: 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 minipage environments with all consequences that come with them (think of `\parindent`, page breaks ...). Since a minipage cannot be broken across pages the surrounding mdframed frame gets the option `nobreak` = `true`. This option has no effect on the sourcecode environment.

7. Usage of the Various Functions

`code-left = true|false` Default: true

If true and the option `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: `\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.

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

`\LaTeX` code example

a \LaTeX code example

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

a \LaTeX code example

`\LaTeX` code example

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`

Overwrite the settings with new ones.

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 10.2.1 on page 36.

`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.

`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

7. Usage of the Various Functions

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 `cnltx` 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 `cnltx` listings style which will affect all sourcecode environments!*

`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.

Introduced in
version 0.4

`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.

7.5. Example File

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:

```
1 \documentclass[load-preamble]{cnltx-doc}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \setcnltx{
5   package = mypackage ,
6   authors  = John Doe ,
7   email    = john@doe.com ,
8   add-cmds = {mycommand} ,
9   add-envs = {myenv}
10 }
11 \begin{document}
12   ...
13 \end{document}
```

7.6. Additional Functionality Provided by **CNLTX-BASE**

The **CNLTX-BASE** package's main purpose is to provide programming facilities. Most of its macros are listed in section A.1. However, I like to explain some of its features in a bit more detail.

7.6.1. Looking for Trailing Punctuation

The command `\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.

`\cnltx@ifpunctuation*[\langle punctuation marks \rangle]{\langle true \rangle}{\langle false \rangle}\langle trailing punctuation \rangle`

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

`set-trail-punct = {\langle punctuation marks \rangle}`

Default: `,.!?;:`

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

The usage is probably self-explaining:

```
1 \makeatletter
2 \cnltx@ifpunctuation{(test\cnltx@trailpunct)}{(test)}!\par
3 \cnltx@ifpunctuation[.]{(test\cnltx@trailpunct)}{(test)}!\par
4 a punctuation mark \cnltx@ifpunctuation*{follows}{doesn't follow}!\par
5 a full stop \cnltx@ifpunctuation*[.]{follows}{doesn't follow}!
```

(test!)
 (test)!
 a punctuation mark follows!
 a full stop doesn't follow!

If the non-starred variant has gobbled a `\par` the `\par` is placed back:

```
1 \makeatletter
2 \def\test{\cnltx@ifpunctuation{(test\cnltx@trailpunct)}{(test)}}%
3 \makeatother
4 \test
5
6 \test.
7
8 \test{} .
```

(test)
 (test.)
 (test) .

7.6.2. 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:

```
1 \def\arabic#1{\expandafter\@arabic\csname c@#1\endcsname}
2 \def\@arabic#1{\number #1}
```

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>`, 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:

```
1 \def\foo#1{\expandafter\@foo\csname c@#1\endcsname}
2 \def\@foo#1{do something with #1 (where #1 is a count)}
```

This means in order to get a new counter representation command you actually need to define *two* macros.

CNLTX-BASE defines an interface that allows to define both commands at once without having to think about `\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.

New Commands

\DeclareCounterRepresentation{<command>}{<definition>}

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

\newcounterrepresentation{<command>}{<definition>}

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

\providecounterrepresentation{<command>}{<definition>}

Provides a new counter representation command and its internal equivalent. In the `<definition>` `#1` is used to refer to the counter *number*, that is, the value of `\c@<counter>`. This command will

7. Usage of the Various Functions

define the commands only if neither the user command nor the internal command (cf. `\arabic` and `\@arabic`) already exist and will do nothing if either of them exist.

`\renewcounterrepresentation`{*<command>*}{*<definition>*}

Redefines an existing counter representation command and its internal equivalent. In the *<definition>* #1 is used to refer to the counter *number*, that is, the value of `\c@<counter>`. This command will issue an error if either the user command or the internal command (cf. `\arabic` and `\@arabic`) or both already exist.

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

```
1 \makeatletter\ttfamily
2 before:\par
3 \meaning\arabic\par
4 \meaning\@arabic
5
6 \renewcounterrepresentation\arabic{\the\numexpr#1\relax}%
7 after:\par
8 \meaning\arabic\par
9 \meaning\@arabic
```

```
before:
macro:#1->\expandafter \@arabic \csname c@#1\endcsname
macro:#1->\number #1
after:
macro:#1->\expandafter \@arabic \csname c@#1\endcsname
macro:#1->\the \numexpr #1\relax
```

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?


```
1 \newcounterrepresentation\minusone{\the\numexpr#1-1\relax}%
2 \newcounterrepresentation\multiofffourrm{\romannumeral\numexpr(4*#1)-4\relax}%
3 % \newrobustcmd is provided by the 'etoolbox' package
4 \newrobustcmd*\circlenumber[1]{%
5   \tikz[baseline]\node[anchor=base,draw,shape=circle]{\number#1};}%
6 \newcounterrepresentation\circled{\circlenumber{#1}}%
7 \makeatletter
8 \newcounterrepresentation\twodigits{\two@digits{#1}}%
9 \makeatother
10 \newcounter{test}%
11 \setcounter{test}{9}
12
```

7. Usage of the Various Functions

```

13 \minusone{test}\par
14 \multoffourrm{test}\par
15 \circled{test}\par
16 \twodigits{test}

```

8
xxxii

09

7.6.3. Expandable Document Commands

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

`\newexpandablecmd*{<cs>}[<num args>][<default opt>]{<definition>}`

Introduced in
version 0.7

This command has the same syntax as `\newcommand`. The difference is that if `<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 `\empty` or something. There's another drawback: a command `\test` thus defined cannot distinguish between `\test[]` and `\test{[]}` and will misinterpret the second as a present optional argument.

My recommendation is to never use this for defining a user command.¹⁹ 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>}`

Introduced in
version 0.7

The equivalent of `\renewcommand`. See description of `\newexpandablecmd` for further details. is to `\newcommand`.

`\provideexpandablecmd*{<cs>}[<num args>][<default opt>]{<definition>}`

Introduced in
version 0.7

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

7.7. Additional Functionality Provided by **CNLTX-TOOLS**

7.7.1. Commands for Defining Different Document Macros

The **CNLTX-TOOLS** package defines some additional macros which provide useful functionality also in contexts *not* documenting a L^AT_EX package.

`\newname{<cs>}{<first name>}{<second 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.

¹⁹ I can see the contradiction here: if a command is no user command there is no need for an optional argument.

7. Usage of the Various Functions

`\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 `\lppl` and `\ctan`. *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.

`\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⟩}`

Default: `\textsc{\\, #1}`

Used in some predefined abbreviations.

`\cnltxlatin{⟨abbreviation⟩}`

Default: `\textit{#1}`

Used in some localization strings.

`acronym-format = {⟨definition⟩}`

Default: `\scshape`

Formatting of the acronyms as typeset with `\cnltxacronym`.

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

```
1 \newname\carlisle{David}{Carlisle}%
2 \carlisle\ is a well-known member of the \LaTeX\ community. \carlisle* is
3 the author of many packages such as \pkg*{longtable}. Take a look in the
4 index where you'll find \carlisle* mentioned.
5
6 \lppl\ is defined as \cnltxacronym{LPPL}{\lppl}.
```

David Carlisle is a well-known member of the \LaTeX community. Carlisle is the author of many packages such as `longtable`. Take a look in the index where you'll find Carlisle mentioned.

LPPL is defined as LPPL.

7.7.2. Defining Abbreviations

In section 7.7.1 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 **CNLT_X-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 an intra-sentence space and NASA\@. for an inter-sentence space. The dot added by **CNLT_X-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:

```

1 \ttfamily% <= this will amplify the visual effect of \nonfrenchspacing
2 \newabbr\ab{a.b}%
3 \newabbr\AB{A.B}%
4 \newabbr*\cd{c.d.}%
5 \ab\ and some words\par
6 \ab. and some words\par
7 \AB\ and some words\par
8 \AB. and some words\par
9 \cd\ and some words

```

a.b. and some words
a.b. and some words
A.B. and some words
A.B. and some words
c.d. and some words

Beware: **CNLT_X-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.

7.7.3. Predefined Abbreviations

CNLT_X-TOOLS already provides a bunch of abbreviations defined with its `\newabbr` command.

Abbreviations that allow Localization

CNLT_X-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 13.

7. Usage of the Various Functions

`\ie`

Prints “*i. e.*” or “d. h.”

`\eg`

Prints “*e. g.*” or “z. B.”

`\etc`

Prints “*etc.*” or “etc.”

`\cf`

Prints “*cf.*” or “vgl.”

All of these macros add a final dot followed by `\@` *except* if a dot *directly* follows the macro.

<code>1 \eg\ and some following text\par</code>	<i>e. g.</i> and some following text
<code>2 \eg, and some following text\par</code>	<i>e. g.,</i> and some following text
<code>3 \eg. and some following text\par</code>	<i>e. g.</i> and some following text
<code>4 \selectlanguage{ngerman}</code>	z. B. and some following text
<code>5 \eg\ and some following text\par</code>	z. B., and some following text
<code>6 \eg, and some following text\par</code>	z. B. and some following text
<code>7 \eg. and some following text</code>	

German Abbreviations

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

`\dsh`

Prints “d. h.”

`\Dsh`

Prints “D. h.”

`\usf`

Prints “usf.”

`\usw`

Prints “usw.”

`\uswusf`

Prints “usw. usf.”

`\zB`

Prints “z. B.”

`\ZB`

Prints “Z. B.”

`\vgl`

Prints “vgl.”

`\Vgl`

Prints “Vgl.”

These macros behave the same as the ones described in section 7.7.3 on page 24.

1 <code>\dsh\</code> und weiterer Text <code>\par</code>	d. h. und weiterer Text
2 <code>\dsh.</code> und weiterer Text <code>\par</code>	d. h. und weiterer Text
3 <code>\usw\</code> und weiterer Text <code>\par</code>	usw. und weiterer Text
4 <code>\usw.</code> und weiterer Text <code>\par</code>	usw. und weiterer Text
5 <code>\usf\</code> und weiterer Text <code>\par</code>	usf. und weiterer Text
6 <code>\usf.</code> und weiterer Text <code>\par</code>	usf. und weiterer Text
7 <code>\zB\</code> und weiterer Text <code>\par</code>	z. B. und weiterer Text
8 <code>\zB.</code> und weiterer Text	z. B. 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 7.7.1 on page 22.

`\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.

1 She left for work before 6 <code>\AM</code> , but	She left for work before 6 A.M., but did not
2 did not arrive until 12 <code>\PM</code> . The	arrive until 12 P.M. The interval 5 B.C.–
3 interval 5 <code>\BC</code> --5 <code>\AD</code> is one year	5 A.D. is one year shorter than the interval
4 shorter than the interval	95 A.D.–105 A.D.
5 95 <code>\AD</code> --105 <code>\AD</code> .	

8. 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 **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 12 on page 40.

8.1. Formatting by Redefining Hooks

You can change the formatting by redefining the following commands. They're all defined by the **CNLTX-EXAMPLE** package except where indicated differently.

<code>\codefont</code>	Default: <code>\ttfamily</code>
This command is used for all formatting of source code.	
<code>\sourceformat</code>	Default: <code>\codefont\small</code>
Formatting of the listings.	
<code>\exampleformat</code>	(initially empty)
Special formatting of the output of a listing.	
<code>\versionnoteformat</code>	Default: <code>\footnotesize\sffamily\RaggedRight</code>
Formatting of the notes introduced in section 5.2 on page 8.	
<code>\packageformat</code>	Default: <code>\sffamily</code>
The formatting of package names.	
<code>\classformat</code>	Default: <code>\sffamily</code>
The formatting of class names.	
<code>\argumentformat</code>	Default: <code>\normalfont\itshape</code>
The formatting of <code>\meta{⟨meta⟩}</code> .	

provided by
CNLTX-DOC

```
1 \renewcommand*\codefont{\sffamily\bfseries}
2 \code{foo} and \cs*{bar}, option \option{baz}
```

foo and **\bar**, option **baz**

8.2. Formatting by Setting Options

You can change the formatting of by setting the following options. They're all defined by the **CNLTX-EXAMPLE** package except where indicated differently.

<code>title-format = {⟨definition⟩}</code>	Default: <code>\bfseries\scshape</code>
Formatting of the document title.	

Introduced in
version 0.2

8. Formatting Possibilities

Introduced in version 0.6	<code>abstract-width = {⟨dimension⟩}</code>	Default: <code>.75\linewidth</code>
	The width of the <code>\parbox</code> the abstract as set with the <code>abstract</code> option is placed in.	
Introduced in version 0.6	<code>abstract-format = {⟨definition⟩}</code>	Default: <code>\setlength\parskip{.333\baselineskip}</code>
	Code that is placed in the parbox the abstract is placed in <i>before</i> the abstract text.	
	<code>caption-font = {⟨definition⟩}</code>	Default: <code>\normalfont\small\sffamily</code>
	This option only has any effect if you use the option <code>load-preamble</code> , see section 9.5 on page 32 for details on the option.	
	<code>caption-label-font = {⟨definition⟩}</code>	Default: <code>\normalfont\small\sffamily\scshape</code>
	This option only has any effect if you use the option <code>load-preamble</code> , see section 9.5 on page 32 for details on the option.	
	<code>code-font = {⟨definition⟩}</code>	Default: <code>\ttfamily</code>
	Used for all formatting of source code.	
	<code>source-format = {⟨definition⟩}</code>	Default: <code>\codefont\small</code>
	Formatting of the listings.	
	<code>expl-format = {⟨definition⟩}</code>	(initially empty)
	Special formatting of the output of a listing.	
provided by CNLTX-DOC	<code>module-sep = {⟨definition⟩}</code>	Default: <code>\,>>\,</code>
	Change the separator between module name and corresponding option name.	
provided by CNLTX-DOC	<code>version-note-format = {⟨definition⟩}</code>	Default: <code>\footnotesize\sffamily\RaggedRight</code>
	Formatting of the notes introduced in section 5.2 on page 8.	
	<code>pkg-format = {⟨definition⟩}</code>	Default: <code>\sffamily</code>
	The formatting of package names.	
	<code>cls-format = {⟨definition⟩}</code>	Default: <code>\sffamily</code>
	The formatting of class names.	
	<code>arg-format = {⟨definition⟩}</code>	Default: <code>\normalfont\itshape</code>
	The formatting of <code>\meta{⟨meta⟩}</code> .	
Introduced in version 0.2	<code>default-format = {⟨code⟩}</code>	Default: <code>\uline</code>
	The formatting of <code>\default</code> 's argument. <code>⟨code⟩</code> 's last macro should take one argument.	

```

1 \setcnltx{code-font=\sffamily\itshape}
2 \code{foo} and \cs*{bar}, option \option{baz}

```

foo and *\bar*, option *baz*

9. Commands, Options and Further Settings Directly Related to the **CNLTX-DOC** Class

9.1. Using Class Options

The **CNLTX-DOC** class only knows a few options:

load-preamble = `true` | `false` Default: `false`

See section 9.5 on page 32 for details.

load-preamble+ = `true` | `false` Default: `false`

See section 9.6 on page 33 for details.

add-index = `true` | `false` Default: `false`

See section 9.6 on page 33 for details.

babel-options = {`<options>`} Default: `english`

Options given to the babel²⁰ package. This option only has an effect if **load-preamble** = `true`.

scrartcl = {`<options>`} (initially empty)

Options that are passed to the underlying class scrartcl. *All global options you want to use should be given here.*

9.2. 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.

package = {`<package>`}

The name of the package that is described. Either this option or **class** or **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 **CNLTX**.

class = {`<class>`}

The name of the class that is described. Either this option or **package** or **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 **CNLTX**.

name = {`<name>`}

The name of the class/package that is described. Either this option or **package** or **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 **CNLTX**.

authors = {`<author list>`}

Comma separated list of package/class authors. After each author name you can add an email

Changed in
version 0.4

²⁰. on CTAN as babel: <http://mirrors.ctan.org/macros/latex/required/babel/>

9. Commands, Options and Further Settings Directly Related to the *CNLTX-DOC* Class

address by writing it in square brackets: 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. **CNLTX** tries to extract the information from the given **package** or **class**. This option can be used to set it explicitly.

date = {<date>}

Date of the package/class. **CNLTX** tries to extract the information from the given **package** or **class**. This option can be used to set it explicitly.

info = {<package/class info>}

Information about the package/class. **CNLTX** tries to extract the information from the given **package** or **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 `.75\linewidth`. Actually it does not have to be text but could be an image or whatever you like.

9.3. Building of the Manuals Title Page

If either the **package** or **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 `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:

build-title = `true`|`false`

The default state depends on other options given like **package**. However, setting this option to `false` *after* any of the options described in section 9.2 on the preceding page will prevent the building of a title page and allows you to design your own.

9.4. A Quotation Environment

Introduced in
version 0.5

CNLTX-DOC provides a quotation environment:

\begin{cnltxquote}[<author/reference>]

A quotation environment.

9. Commands, Options and Further Settings Directly Related to the *CNLT*X-DOC Class

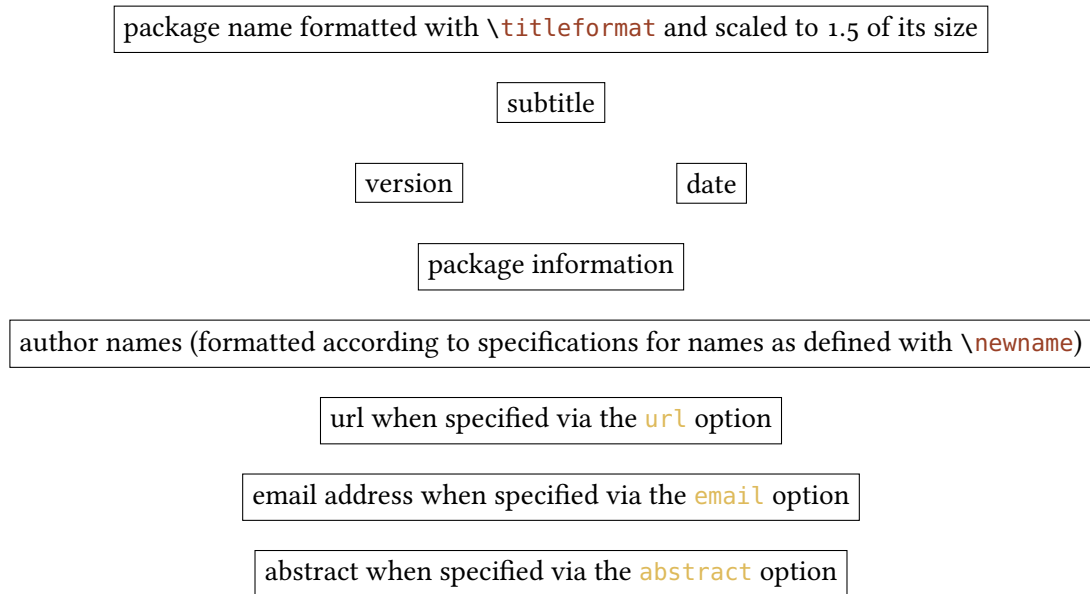


FIGURE 1: Schematic sketch of the title page.

The environment sets the body indented on both sides as it simply uses a quote environment internally. The contents of the optional argument is set flush right after the environment's body. The formatting is controlled by two options:

`quote-format` = $\{\langle definition \rangle\}$

Default: `\small\sffamily`

The formatting of the environment's body.

`quote-author-format` = $\{\langle definition \rangle\}$

Default: `\itshape`

```

1 \begin{cnltxquote}[Douglas Adams, The Restaurant at the End of the Universe]
2   "The first ten million years were the worst," said Marvin, "and the
3   second ten million years, they were the worst too. The third ten million
4   years I didn't enjoy at all. After that I went into a bit of a decline."
5 \end{cnltxquote}

```

"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."

Douglas Adams, The Restaurant at the End of the Universe

9.5. Predefined Preamble

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

load-preamble

Class option that preloads part of my custom preamble.

Using the option will include the following code:

```

1 \RequirePackage{ifxetex,ifluatex}
2 \ifbool{not bool{xetex} and not bool{luatex}}
3   {\RequirePackage[T1]{fontenc}}
4   {\RequirePackage{fontspec}}
5 \RequirePackage[oldstyle]{libertine}
6 % 'libertinehologopatch' is not on CTAN, yet!
7 % you can get it at https://bitbucket.org/cgnieder/libertinehologopatch/
8 \RequirePackage{libertinehologopatch}
9 \RequirePackage[supstfm=libertinesups]{superiors}
10 % libertine does not have superior letters:
11 \def\@makefnmark{%
12   \hbox{%
13     \cnltx@ifisnum{\@thefnmark}
14       {\textsu{\hspace*{\superiors@spaced}\@thefnmark}}
15       {\@textsuperscript{\normalfont\@thefnmark}}}%
16   }%
17 }
18 \RequirePackage{microtype}
19 \ifbool{
20   {
21     test {\ifcsdef{MT@pr@set@romansans}} and
22     test {\ifcsdef{MT@ex@set@romansans}}
23   }
24   {}
25   {
26     \DeclareMicrotypeSet{romansans}{
27       encoding = {*},
28       family   = {rm*,sf*}
29     }
30   }
31 \ifcsdef{MT@tr@set@scshape}
32 {}
33 {
34   \DeclareMicrotypeSet[tracking]{scshape}{
35     encoding = {*} ,
36     shape    = {sc,scit,si}
37   }
38 }
39 \microtypesetup{

```



```

40 tracking    = scshape ,
41 protrusion = romansans ,
42 expansion   = romansans
43 }
44 \ifbool{expr}{not bool{xetex} and not bool{luatex}}
45   {\RequirePackage[scaled=.79]{beramono}}
46   {\setmonofont[Scale=MatchLowercase]{Bitstream Vera Sans Mono}}
47 \RequirePackage{fnpct}
48 \expandafter\RequirePackage\expandafter[\cnltx@babel@options]{babel}
49 \renewcommand*{\othersectionlevelsformat}[3]{%
50   \textcolor{cnltx}{#3\autodot}\enskip}
51 \renewcommand*{\partformat}{%
52   \textcolor{cnltx}{\partname~\thepart\autodot}}
53 \deffootnote[2em]{1em}{\llap{\thefootnotemark. }}%
54 \pagestyle{headings}
55 \setcapindent{1.5em}
56 \setkomafont{caption}{\cnltx@caption@font}
57 \setkomafont{captionlabel}{\cnltx@captionlabel@font}

```

The effect of this preamble is demonstrated by the document you’re reading at this moment.

9.6. Predefined Indexing

*CNLT*X-DOC allows the automated creation of an index. This is done with the help of the *imakeidx* package by Enrico Gregorio [Gre13]. To use this feature you have two class options. They cannot be set with `\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 `load-preamble`, `add-index` and `add-bib`.

Enabling the feature

- loads the *imakeidx*²¹ package,
- uses a given style file for the index that can be specified with the `index-style` option,
- sets a certain setup for the index that can be specified with the `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:

`index-prologue = {⟨text⟩}`

Adds `⟨text⟩` as index prologue between heading and the actual index.

21. on CTAN as *imakeidx*: <http://mirrors.ctan.org/macros/latex/contrib/imakeidx/>

9. Commands, Options and Further Settings Directly Related to the *CNLT*X-DOC Class

`index-space` = { $\langle dimension \rangle$ } Default: 0pt

The vertical space between index prologue and index.

`index-setup` = { $\langle options \rangle$ } Default: `othercode=\footnotesize,level=\addsec`

The options that are passed to imakeidx's `\indexsetup` command.

`makeindex-setup` = { $\langle options \rangle$ } Default: `columns=2,columnsep=1em`

The options that are passed to the `\makeindex` command.

`index-style` = { $\langle style file \rangle$ } Default: `cnltx.ist`

The style file that is used for formatting the index.

The index style file `cnltx.ist` contains the following lines:

```
1 heading_prefix "{\bfseries "  
2 heading_suffix "\\hfil}\\nopagebreak\n"  
3 headings_flag 1  
4 delim_0 "\\dotfill"  
5 delim_1 "\\dotfill"  
6 delim_2 "\\dotfill"  
7 delim_r "\\nohyperpage{\\textendash}"  
8 delim_t ""  
9 suffix_2p "\\nohyperpage{\\,\\GetTranslation{cnltx-f.}\\@}"  
10 suffix_3p "\\nohyperpage{\\,\\GetTranslation{cnltx-ff.}\\@}"
```

The feature is demonstrated by this document which does not contain a single control sequence containing the string `index`!

9.7. Bibliography with biblatex

9.7.1. Bibliography Entry Types package, class and bundle for biblatex

Introduced in
version 0.4

*CNLT*X-DOC defines the bibliography entry types package, class and bundle when `biblatex` [Leh13] is used. This allows specifying L^AT_EX packages in bib files:

```
1 @package{pkg:chngcntr,  
2   title      = {chngcntr} ,  
3   author     = {Peter Wilson} ,  
4   maintainer = {Will Robertson} ,  
5   date       = {2009-09-02} ,  
6   version    = {1.0a} ,  
7   url        = {http://mirror.ctan.org/macros/latex/contrib/chngcntr/}  
8 }  
9 @class{cls:exam,  
10  title      = {exam},  
11  author     = {Philip Hirschhorn},  
12  date       = {2011-05-22},
```

```

13  version      = {2.4},
14  url          = {http://mirror.ctan.org/macros/latex/contrib/exam/}
15 }
16 @bundle{bnd:koma-script,
17  title        = {\KOMAScript} ,
18  sorttitle    = {KOMA-Script} ,
19  indextitle    = {\KOMAScript} ,
20  indexsorttitle = {KOMA-Script} ,
21  author       = {Markus Kohm and Frank Neukahm},
22  date         = {2012-07-29} ,
23  version      = {3.11b} ,
24  url          = {http://mirror.ctan.org/macros/latex/contrib/koma-script/}
25 }

```

As you can see also an entry field maintainer is defined. For this to work you have to use the biblatex bibliography style `cnltx`. This style basically is a clone of the style `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 `cnltx` is provided, again a clone of the `alphabetic` style. The only addition it makes is that indexing of maintainer names is enabled if biblatex's indexing option is used. *The styles can only be used with the **CNLT**X-DOC class as it relies on definitions made by it.*

This document uses the following call of biblatex:

```

1 \usepackage[
2   backend=biber,
3   style=cnltx,
4   sortlocale=en_EN,
5   indexing=cite,
6   useprefix]{biblatex}
7 \addbibresource{cnltx.bib}

```

Actually it let's **CNLT**X-DOC do it, see section 9.7.2 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.

9.7.2. Automatic Bibliography

CNLTX-DOC allows the automated creation of a bibliography.

add-bib = `true|false` Default: `false`
 Enables the automatic creation of a bibliography at the end of the document.

load-preamble+ = `true|false` Default: `false`
 This option has the same effect as adding the options `load-preamble`, `add-index` and `add-bib`.

What this options does is including the following code:

```

1 \RequirePackage[
2   backend=biber,
3   style=cnltx,
4   sortlocale=en_EN,
5   indexing=cite,
6   useprefix]{biblatex}
7 \addbibresource{cnltx.bib}
8 \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.

10. Predefined listings and mdframed Styles

10.1. mdframed

The source code environments (see section 7.4 on page 16) all get a frame with the help of the `mdframed` [Dan13] package. For this a custom style is defined called `cnltx`. The options `frame-options` and `add-frame-options` mentioned in section 7.4 on page 16 manipulate this style. It is predefined with these values:

```

1 \def\cnltx@mdframed@options{
2   backgroundcolor = cnltxbg ,
3   linecolor      = cnltx ,
4   roundcorner    = 5pt
5 }

```

10.2. listings

10.2.1. L^AT_EX Sourcecode

The code of the source code environments (see section 7.4 on page 16) is formatted with the help of the `listings` package [HM13]. A listings style is defined called `cnltx`. The options `add-cmds`, `add-silent-cmds`, `add-envs`, `add-silent-envs`, `listings-options` and `add-listings-options` manipulate this style. It is predefined by `CNLTX-EXAMPLE` as follows:

```

1 \def\cnltx@listings@style{
2   language      = [AllaTeX]TeX,
3   alsolanguage  = [plain]TeX,
4   basicstyle    = {\sourceformat},
5   numbers      = left,
6   numberstyle   = \tiny,

```

```

7  xleftmargin      = 1em,
8  numbersep       = .75em,
9  gobble          = \cnltx@gobble ,
10 columns         = fullflexible,
11 literate         =
12     {ä}{\a}1
13     {ö}{\o}1
14     {ü}{\u}1
15     {Ä}{\A}1
16     {Ö}{\O}1
17     {Ü}{\U}1
18     {ß}{\ss}1 ,
19 breaklines      = true,
20 keepspaces      = true,
21 breakindent     = 1em,
22 commentstyle    = \color{comment},
23 keywordstyle    = \color{cs},
24 deletetexcs     =
25     {
26     a,o,u,A,O,U,
27     begin,
28     center,
29     description,document,
30     end,enumerate,
31     figure,flushleft,flushright,
32     itemize,list,
33     otherlanguage,
34     table,tabu,tabular
35     },
36 deletekeywords  =
37     {
38     a,o,u,A,O,U,
39     begin,
40     center,
41     description,document,
42     end,enumerate,
43     figure,flushleft,flushright,
44     itemize,list,
45     otherlanguage,
46     table,tabu,tabular
47     },
48 % \begin, \end:
49 texcsstyle      = [2]\color{beginend},
50 index           = [2][texcs2],
51 indexstyle      = [2]\@gobble,
52 moretexcs       = [2]{begin,end},
53 % added environments that'll be indexed:
54 texcsstyle      = [3]\color{env},

```

```

55 index           = [3][texcs3],
56 indexstyle      = [3]\envidx,
57 % environments that won't be indexed:
58 texcsstyle      = [4]\color{env},
59 index           = [4][texcs4],
60 indexstyle      = [4]\@gobble,
61 % control sequences that'll be indexed:
62 texcsstyle      = [5]\color{cs},
63 index           = [5][texcs5],
64 indexstyle      = [5]\indexcs,
65 % control sequences that won't be indexed:
66 texcsstyle      = [6]\color{cs},
67 index           = [6][texcs6],
68 indexstyle      = [6]\@gobble
69 }

```

10.2.2. BibTeX Entries

Introduced in
version 0.4

The **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 9.7.1 on page 34. **CNLTX-EXAMPLE** defines a listings style for formatting them called `cnltx-bibtex`:

```

1 \def\cnltx@bibtex@listings@style{
2   language       = BiBTeX,
3   basicstyle     = {\sourceformat},
4   numbers       = left,
5   numberstyle    = \tiny,
6   xleftmargin    = 1em,
7   numbersep      = .5em,
8   gobble         = \cnltx@gobble ,
9   columns       = fullflexible,
10  literate       =
11    {ä}{{\a}}1
12    {ö}{{\o}}1
13    {ü}{{\u}}1
14    {Ä}{{\A}}1
15    {Ö}{{\O}}1
16    {Ü}{{\U}}1
17    {ß}{{\ss}}1 ,
18  breaklines     = true,
19  keepspaces     = true,
20  breakindent    = 1em,
21  commentstyle   = \color{comment},
22  keywordstyle   = \color{bibentry} ,
23  keywordstyle   = [2]\color{bibentryfield}\itshape ,
24  showstringspaces = false ,
25 }

```

10.2.3. makeindex Style Files

Introduced in
version 0.7

CNLTX-LISTINGS defines a listings language makeindex that contains the keywords used in makeindex style files. **CNLTX-EXAMPLE** defines a listings style for formatting them called `cnltx-makeindex`:

```

1 \def\cnltx@makeindex@listings@style{
2   language      = makeindex,
3   basicstyle     = {\sourceformat},
4   numbers       = left,
5   numberstyle    = \tiny,
6   xleftmargin    = 1em,
7   numbersep      = .75em,
8   gobble         = \cnltx@gobble ,
9   columns        = fullflexible,
10  literate       =
11    {ä}{{\a}}1
12    {ö}{{\o}}1
13    {ü}{{\u}}1
14    {Ä}{{\A}}1
15    {Ö}{{\O}}1
16    {Ü}{{\U}}1
17    {ß}{{\ss}}1 ,
18  breaklines     = true,
19  keepspaces     = true,
20  breakindent    = 1em,
21  commentstyle    = \color{comment},
22  keywordstyle    = \color{makeidxkey}\bfseries ,
23  stringstyle     = \color{makeidxstring} ,
24  showstringspaces = false
25 }
```

11. 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 **CNLTX-DOC** loads hyperref this means you should do so, too, when you use **CNLTX-DOC**. This is important for two reasons:

1. Indexing in strings that get written to the table of contents does not 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.

12. Predefined Colors and Color-Schemes

12.1. Explicitly Defined Colors

The **CNLTX-BASE** package defines a number of colors:

cnltxbrown

Per default used for the control sequences.

cnltxblue

Per default used for module names.

cnltxred

Per default used as base color in various places.

cnltxgreen

Unused per default.

cnltxgray

Per default used for formatting comments.

cnltxyellow

Per default used for option names.

cnltxformalblue

Unused per default.

cnltxformalred

Unused per default.

12.2. Actual Used Color Names and Color Schemes

The colors defined in section 12.1 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.

The 'default' color scheme is defined as follows:

Introduced in
version 0.5

12. Predefined Colors and Color-Schemes

```
1 \definecolorscheme{default}{
2   cs          => cnltxbrown , % command sequences
3   option      => cnltxyellow , % options
4   module      => cnltxblue ,  % modules
5   comment     => cnltxgray ,  % comments
6   beginend    => red ,        % \begin and \end
7   env         => black ,      % environment names
8   argument    => black ,      % argument delimiters
9   meta        => black!80 ,   % arguments of \meta
10  cnltx        => cnltxred ,   % base color
11  cnltxbg      => white ,      % source code box background
12  link         => black!90 ,   % hyperlinks
13  versionnote  => black!75    % versioning notes text
14  bibentry     => cnltxgreen , % BibTeX entry types
15  bibentryfield => black ,     % BibTeX entry fields
16  expandable   => red ,        % the color used in \expandable
17  unexpandable => black ,     % the color used in \unexpandable
18  makeidxkey   => cnltxgreen , % used for keywords in the cnltx-makeindex
19                                     % style
20  makeidxstring => black      % used for strings in the cnltx-makeindex
21                                     % style
22 }
```

The ‘blue’ color scheme is defined this way:

```
1 \definecolorscheme{blue}{
2   cs          => cnltxbrown ,
3   option      => cnltxgreen ,
4   module      => cnltxred ,
5   comment     => cnltxgray ,
6   beginend    => red ,
7   env         => black ,
8   argument    => black ,
9   meta        => black!80 ,
10  cnltx        => cnltxblue ,
11  cnltxbg      => yellow!10 ,
12  link         => cnltx ,
13  versionnote  => black!75
14  bibentry     => cnltxyellow ,
15  bibentryfield => black ,
16  expandable   => red ,
17  unexpandable => black ,
18  makeidxkey   => cnltxyellow ,
19  makeidxstring => black
20 }
```

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

```

1 \definecolorsscheme{formal}{
2   cs           => black ,
3   option       => cnltxformalblue ,
4   module       => cnltxblue ,
5   comment      => cnltxgray ,
6   beginend     => red ,
7   env          => black ,
8   argument     => black ,
9   meta         => black!80 ,
10  cnltx         => cnltxformalblue ,
11  cnltxbg       => white ,
12  link          => black!90 ,
13  versionnote  => black!75 ,
14  bibentry      => black ,
15  bibentryfield => black ,
16  expandable    => red ,
17  unexpandable => black ,
18  makeidxkey    => black ,
19  makeidxstring => black
20 }

```

13. Language Support

Introduced in
version 0.2

The **CNLTX-DOC**, the **CNLTX-EXAMPLE** and the **CNLTX-TOOLS** package all rely on the translations package [Nie13d] for providing some document language dependent strings. Currently only translations for English and German are provided. Others can be added and the existing ones changed with the following command provided by the translations package:

\DeclareTranslation{ $\langle language \rangle$ }{ $\langle keyword \rangle$ }{ $\langle translation \rangle$ }

Provide translations for the string identified by the ID $\langle keyword \rangle$.

The defined strings are listed in table 1 on the following page. They are used in indexing strings and in different parts of the document.

Part III.

Appendix

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. Expandable commands are marked with *****.

A. Internal Helper Commands

TABLE 1: Overview over available internationalization key words.

Package/Class	key word	English version	German version
CNLTX-EXAMPLE	<code>cnltx-package</code>	package	Paket
CNLTX-EXAMPLE	<code>cnltx-class</code>	class	Klasse
CNLTX-EXAMPLE	<code>cnltx-bundle</code>	bundle	Bundle
CNLTX-EXAMPLE	<code>cnltx-environment</code>	environment	Umgebung
CNLTX-DOC	<code>cnltx-default</code>	Default	Voreinstellung
CNLTX-DOC	<code>cnltx-empty</code>	initially empty	zunächst leer
CNLTX-DOC	<code>cnltx-required</code>	required	erforderlich
CNLTX-DOC	<code>cnltx-toc</code>	Table of Contents	Inhaltsverzeichnis
CNLTX-DOC	<code>cnltx-license</code>	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	Es ist erlaubt, diese Software unter den Bedingungen der \LaTeX Project Public License (LPPL), Version 1.3 oder später, zu kopieren und zu verteilen (http://www.latex-project.org/lppl.txt). Sie hat den Status
CNLTX-DOC	<code>cnltx-introduced</code>	Introduced in version	Eingeführt in Version
CNLTX-DOC	<code>cnltx-changed</code>	Changed in version	Geändert in Version
CNLTX-DOC	<code>cnltx-f.</code>	f.	f.
CNLTX-DOC	<code>cnltx-ff.</code>	ff.	ff.
CNLTX-DOC	<code>cnltx-maintainer</code>	current maintainer	aktueller Maintainer
CNLTX-DOC	<code>cnltx-maintainers</code>	current maintainers	aktuelle Maintainer
CNLTX-TOOLS	<code>cnltx-i.e.</code>	<i>i. e</i>	d. h
CNLTX-TOOLS	<code>cnltx-e.g.</code>	<i>e. g</i>	z. B
CNLTX-TOOLS	<code>cnltx-cf.</code>	<i>cf</i>	vgl
CNLTX-TOOLS	<code>cnltx-etc.</code>	<i>etc</i>	etc

A.1. Defined by **CNLTX-BASE**

Especially **CNLTX-BASE** defines some useful helper macros that are also used by the other packages and classes.

A.1.1. Related to the Bundle

* **\cnltx@@date**

The creation date of the current version of the bundle.

* **\cnltx@@version**

The version number of the bundle.

* **\cnltx@@info**

The short description of the bundle.

\cnltx@create@bundle@message*{*<module>*}{Error|Warning|WarningNoLine|Info}

Introduced in
version 0.7

Create suiting error and warning messaging commands for the module *<module>* of the **CNLTX** bundle. The starred version creates messages for a class the un-starred version messages for a package.

\cnltx@base@error{*<message>*}

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

\cnltx@base@warning{*<message>*}

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

\cnltx@base@warningnoline{*<message>*}

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

\cnltx@base@info{*<message>*}

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

\cnltx@define@colorscheme{*<name>*}{*<scheme definition>*}

Command that can be used to define a color scheme.

A.1.2. Programming Tools

\cnltx@create@message*{*<prefix>*}{*<package/class name>*}{Error|Warning|WarningNoLine|Info}{*<detailed error message>*}

Changed in
version 0.7

Create error and warning massaging commands **\<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. *<prefix>* will be all lowercase in the generated command.

\cnltx@create@generic@message*{*<prefix>*}{*<package/class name>*}{Error|Warning|WarningNoLine|Info}

Introduced in
version 0.7

Create error and warning massaging commands **\<prefix>@error|warning|warningnoline|**

A. Internal Helper Commands

`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.

`\cnltx@par`

Expands to `\par`. Sometimes you need to smuggle a `\par` in a short macro ...

* `\iftest{⟨test directive⟩}{⟨true⟩}{⟨false⟩}`

Introduced in
version 0.7

Checks if `⟨test directive⟩` is true and either places `⟨true⟩` or `⟨false⟩` in the input stream. `⟨test directive⟩` should be a T_EX test like `\ifx⟨token1⟩⟨token2⟩`.

This is a command in the spirit of etoolbox's `\ifbool` that does the same for a boolean `⟨bool⟩` defined with `\newif\if⟨bool⟩` or `\newbool{⟨bool⟩}`. It corresponds to etoolbox's test directive for its `\ifboolexpr`.

* `\nottest{⟨test directive⟩}{⟨not true⟩}{⟨not false⟩}`

Introduced in
version 0.7

Checks if `⟨test directive⟩` is not true and either places `⟨not true⟩` or `⟨not false⟩` in the input stream. Test directive should be a T_EX test like `\ifx⟨token1⟩⟨token2⟩`.

This is a command in the spirit of etoolbox's `\notbool` that does the same for a boolean `⟨bool⟩` defined with `\newif\if⟨bool⟩` or `\newbool{⟨bool⟩}`.

`\cnltx@ifsym{⟨token⟩}{⟨true⟩}{⟨false⟩}`

A generic version of L^AT_EX's `\ifstar` that checks if `⟨token⟩` follows if the input stream. If yes it is removed and `⟨true⟩` is placed in the input stream else `⟨false⟩`.

`\cnltx@ifdash{⟨true⟩}{⟨false⟩}`

A wrapper for `\cnltx@ifsym{-}`.

`\cnltx@ifbang{⟨true⟩}{⟨false⟩}`

Introduced in
version 0.3

A wrapper for `\cnltx@ifsym{!}`.

* `\cnltx@ifisnum{⟨token list⟩}{⟨true⟩}{⟨false⟩}`

Introduced in
version 0.6

Checks if `⟨token list⟩` is an integer zero or greater and leaves `⟨true⟩` in the input stream if it is an `⟨false⟩` if it isn't.

`\cnltx@expandargs(⟨specs⟩)⟨control sequence⟩`

Introduced in
version 0.7

This is a L^AT_EX 2_ε version of expl3's `\exp_args:N⟨specs⟩`. The command expands the arguments of `⟨control sequence⟩` according to `⟨specs⟩`. In `⟨specs⟩`

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

A. Internal Helper Commands

*\cnltx@stripbs

A shortcut for `\expandafter\@gobble\string`.

`\cnltx@ifin{<tokenlist>}{<search>}{<true>}{<false>}`

Places `<true>` in the input stream if `<search>` is found in `<tokenlist>` and `<false>` if it isn't.

`\cnltx@replace@once{<cs>}{<search>}{<replace>}`

Replaces the first occurrence of `<search>` in the first expansion of `<cs>` with `<replace>`.

`\cnltx@long@replace@once{<cs>}{<search>}{<replace>}`

The same as `\cnltx@replace@once` but `<cs>` will be redefined with `\long`.

Introduced in
version 0.3

`\cnltx@replace@all{<cs>}{<search>}{<replace>}`

Replaces all occurrences of `<search>` in the first expansion of `<cs>` with `<replace>`.

`\cnltx@long@replace@all{<cs>}{<search>}{<replace>}`

The same as `\cnltx@replace@all` but `<cs>` will be redefined with `\long`.

Introduced in
version 0.3

`\cnltx@remove@once{<cs>}{<search>}`

Removes the first occurrence of `<search>` in the first expansion of `<cs>`.

Introduced in
version 0.3

`\cnltx@long@remove@once{<cs>}{<search>}`

The same as `\cnltx@remove@once` but `<cs>` will be redefined with `\long`.

Introduced in
version 0.3

`\cnltx@remove@all{<cs>}{<search>}`

Removes all occurrences of `<search>` in the first expansion of `<cs>`.

Introduced in
version 0.3

`\cnltx@long@remove@all{<cs>}{<search>}`

The same as `\cnltx@remove@all` but `<cs>` will be redefined with `\long`.

Introduced in
version 0.3

A.2. Defined by **CNLTX-DOC**

`\cnltx@doc@error{<message>}`

Issue an error message using `\ClassError{cnltx-doc}`.

`\cnltx@doc@warning{<message>}`

Issue a warning message using `\ClassWarning{cnltx-doc}`.

`\cnltx@doc@warningnoline{<message>}`

Issue a warning message using `\ClassWarningNoLine{cnltx-doc}`.

`\cnltx@doc@info{<message>}`

Issue a message using `\ClassInfo{cnltx-doc}`.

`\cnltx@getfileinfo{<file name>}{<file extension>}`

Extract the date, version and background information for a package or a class and defines `\cnltx@package@date`, `\cnltx@package@version` and `\cnltx@package@info` to contain the extracted data.

`\cnltx@version@note{⟨note⟩}`

Command that is used for the versioning notes internally. Sets `\reversemarginpar` and then writes the note `⟨note⟩` 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 **CNLTX-EXAMPLE**

`\cnltx@example@error{⟨message⟩}`

Issue an error message using `\PackageError{cnltx-example}`.

`\cnltx@example@warning{⟨message⟩}`

Issue a warning message using `\PackageWarning{cnltx-example}`.

`\cnltx@example@warningnoline{⟨message⟩}`

Issue a warning message using `\PackageWarningNoLine{cnltx-example}`.

`\cnltx@example@info{⟨message⟩}`

Issue a message using `\PackageInfo{cnltx-example}`.

`\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 `\cnltxat` in `\cs` and friends. For the sake of convenience you can define a command like `\at` that expands to it.²² In order not to overwrite any such existing macro it is not defined by **CNLTX-EXAMPLE**. This document for example defines `\def\at{\cnltxat}`.

`\cnltxletterat`

An alias of `\cnltxat`.

`\cnltxotherat`

The same as `\cnltxat` but with a ‘@’ with category code 12.

`\cnltxbang`

The same as `\cnltxotherat` except that it contains a ‘!’.

`\cnltxequal`

The same as `\cnltxotherat` except that it contains a ‘=’.

`\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.

22. This is important. If you `\let` it to `\cnltxat` index entries may be sorted differently! Remember: `\cnltxat` is robust.

A. Internal Helper Commands

`\indexcs`

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

`\indexenv`

Introduced in
version 0.7a

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

`\cnltx@treat@lst@index{<new index cs>}{<internal index cs>}`

Introduced in
version 0.7a

This command was used to define `\indexcs` and `\indexenv`:
`\cnltx@treat@lst@index{\indexcs}{\csidx}`

`\newarg[<arg formatting>]{<cs>}{<left delim>}{<right delim>}`

Changed in
version 0.2

Default: `\meta`

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. `\newarg[\code]\Marg{\}{\}`

`\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 `cnltx`.

`\cnltx@listings@style`

Predefined option list for the listings [HM13] style `cnltx`.

A.4. Defined by **CNLT-X-LISTINGS**

`\cnltx@listings@error{<message>}`

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

`\cnltx@listings@warning{<message>}`

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

`\cnltx@listings@warningnoline{<message>}`

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

`\cnltx@listings@info{<message>}`

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

`\cnltx@predefined@control@sequences`

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

B. List of Known L^AT_EX Control Sequences

`\cnltx@predefined@environments`

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

`\listsilentcmds`

Prints all known control sequence names formatted and separated with the separator set with `list-sep`. Requires **CNLTX-EXAMPLE**.

`\listsilentenvs`

Prints all known environment names formatted and separated with the separator set with `list-sep`. Requires **CNLTX-EXAMPLE**.

`\listbibfilekeys{⟨file name⟩}`

Introduced in version 0.7 Prints all cite keys contained in the bibliography file `⟨file name⟩` formatted with `\code` and separated with the separator set with `list-sep`. Requires **CNLTX-EXAMPLE**.

`\listbibfiletypes{⟨file name⟩}`

Introduced in version 0.7 Prints all citation types contained in the bibliography file `⟨file name⟩` formatted with `\code` and separated with the separator set with `list-sep`. Requires **CNLTX-EXAMPLE**.

`\listbibfileentries{⟨file name⟩}`

Introduced in version 0.7 Prints all cite keys contained in the bibliography file `⟨file name⟩` formatted with `\code` and gives their respective entry types, separated with the separator set with `list-sep`. Requires **CNLTX-EXAMPLE**.

`list-sep = {⟨separator⟩}`

Default: `,\space`

Sets the separator for **CNLTX-LISTINGS**’ commands listing the different commands *etc.*

A.5. Defined by **CNLTX-TOOLS**

`\cnltx@tools@error{⟨message⟩}`

Issue an error message using `\PackageError{cnltx-tools}`.

`\cnltx@tools@warning{⟨message⟩}`

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

`\cnltx@tools@warningnoline{⟨message⟩}`

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

`\cnltx@tools@info{⟨message⟩}`

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

`\cnltx@accsupp{⟨actual text⟩}{⟨additional options⟩}{⟨TEX text⟩}`

A wrapper for package `accsupp`’s `\BeginAccSupp{ActualText = ⟨actual text⟩}⟨TEX text⟩\EndAccSupp{}`.

B. List of Known L^AT_EX Control Sequences

Below are listed all *predefined* control sequence names that are treated as “silent” names by **CNLTX**, that is, those defined by **CNLTX-LISTINGS**.

B. List of Known \LaTeX Control Sequences

<code>\-, \@, \@alpha, \@Alpha,</code>	<code>\bar, \baselineskip,</code>	<code>\columnwidth, \cong,</code>
<code>\@arabic, \@ctrerr, \@empty,</code>	<code>\baselinestretch, \batchmode,</code>	<code>\contentsline, \coprod,</code>
<code>\@firstofone, \@firstoftwo,</code>	<code>\BeforeBeginEnvironment,</code>	<code>\copy, \copyright, \cos,</code>
<code>\@gobble, \@ifclassloaded,</code>	<code>\begingroup, \beginsection,</code>	<code>\cosh, \cot, \coth, \count,</code>
<code>\@ifnextchar,</code>	<code>\belowdisplayshortskip,</code>	<code>\countdef, \cr, \crrc,</code>
<code>\@ifpackageloaded, \@ifstar,</code>	<code>\belowdisplayskip, \beta,</code>	<code>\cref, \csdef, \csc, \cslet,</code>
<code>\@makefnmark, \@roman,</code>	<code>\bezier, \bf, \bffam,</code>	<code>\csletcs, \csname, \csuse,</code>
<code>\@Roman, \@secondoftwo,</code>	<code>\bfseries, \bgroup,</code>	<code>\cup, \CurrentOption, \d, \dag,</code>
<code>\@slowromancap,</code>	<code>\bibcite, \bibdata,</code>	<code>\dagger, \dashbox, \dashv,</code>
<code>\@textsuperscript,</code>	<code>\bibitem, \bibliography,</code>	<code>\date, \day, \dblfigrule,</code>
<code>\@thefnmark, \a,</code>	<code>\bibliographystyle,</code>	<code>\dblfloatpagefraction,</code>
<code>\AA, \aa, \above,</code>	<code>\bibstyle, \big, \Big,</code>	<code>\dblfloatsep,</code>
<code>\abovedisplayshortskip,</code>	<code>\bigbreak, \bigcap, \bigcirc,</code>	<code>\dbltextfloatsep,</code>
<code>\abovedisplayskip,</code>	<code>\bigcup, \bigg, \Bigg, \biggl,</code>	<code>\dbltopfraction,</code>
<code>\abovewithdelims,</code>	<code>\Biggl, \biggm, \Biggm, \biggr,</code>	<code>\ddag, \ddagger, \ddot,</code>
<code>\accent, \active, \acute,</code>	<code>\Biggr, \bigl, \Bigl, \bigm,</code>	<code>\ddots, \deadcycles,</code>
<code>\addbibresource,</code>	<code>\Bigm, \bigodot, \bigoplus,</code>	<code>\DeclareCharacterInheritance,</code>
<code>\addcontentsline,</code>	<code>\bigotimes, \bigr, \Bigr,</code>	<code>\DeclareDictTranslation,</code>
<code>\addpenalty, \addtocontents,</code>	<code>\bigskip, \bigskipamount,</code>	<code>\DeclareErrorFont,</code>
<code>\addtocounter, \addtolength,</code>	<code>\bigsqcup, \bigtriangledown,</code>	<code>\DeclareFixedFont,</code>
<code>\addtokomafont,</code>	<code>\bigtriangleup, \biguplus,</code>	<code>\DeclareFontEncoding,</code>
<code>\addtoversion, \addvspace,</code>	<code>\bigvee, \bigwedge,</code>	<code>\DeclareFontEncodingDefaults,</code>
<code>\adjdemerits, \advance,</code>	<code>\binoppenalty, \bmod,</code>	<code>\DeclareFontFamily,</code>
<code>\advancepageno, \ae,</code>	<code>\boldmath, \boolfalse,</code>	<code>\DeclareFontShape,</code>
<code>\AE, \afterassignment,</code>	<code>\booltrue, \bordermatrix,</code>	<code>\DeclareFontSubstitution,</code>
<code>\AfterEndPreamble,</code>	<code>\bot, \botfigrule, \botmark,</code>	<code>\DeclareLanguage,</code>
<code>\AfterEndDocument,</code>	<code>\bottomfraction, \bowtie,</code>	<code>\DeclareLanguageAlias,</code>
<code>\AfterEndEnvironment,</code>	<code>\Box, \box, \boxmaxdepth,</code>	<code>\DeclareLanguageDialect,</code>
<code>\aftergroup, \AfterPreamble,</code>	<code>\brace, \braced, \bracelu,</code>	<code>\DeclareListParser,</code>
<code>\aleph, \allocationnumber,</code>	<code>\bracerd, \braceru,</code>	<code>\DeclareMathAccent,</code>
<code>\allowbreak, \alpha, \Alpha,</code>	<code>\bracevert, \brack, \break,</code>	<code>\DeclareMathAlphabet,</code>
<code>\alpha, \amalg, \and, \angle,</code>	<code>\breve, \brokenpenalty,</code>	<code>\DeclareMathAlphabet,</code>
<code>\approx, \appto, \arabic,</code>	<code>\buildrel, \bullet, \bye,</code>	<code>\DeclareMathDelimiter,</code>
<code>\arccos, \arcsin, \arctan,</code>	<code>\c, \cal, \cap, \caption,</code>	<code>\DeclareMathRadical,</code>
<code>\arg, \array, \arraycolsep,</code>	<code>\cases, \catcode, \cb, \cdot,</code>	<code>\DeclareMathSizes,</code>
<code>\arrayrulewidth,</code>	<code>\cdotp, \cdots, \center,</code>	<code>\DeclareMathSymbol,</code>
<code>\arraystretch, \arrowvert,</code>	<code>\centering, \centerline,</code>	<code>\DeclareMathVersion,</code>
<code>\Arrowvert, \ast, \asympt,</code>	<code>\chapter, \char, \chardef,</code>	<code>\DeclareMicrotypeAlias,</code>
<code>\AtBeginDocument,</code>	<code>\check, \CheckCommand, \chi,</code>	<code>\DeclareMicrotypeBabelHook,</code>
<code>\AtBeginDvi,</code>	<code>\choose, \circ, \circle,</code>	<code>\DeclareMicrotypeSet,</code>
<code>\AtBeginEnvironment,</code>	<code>\citation, \cite, \ClassError,</code>	<code>\DeclareMicrotypeSetDefault,</code>
<code>\AtEndDocument,</code>	<code>\ClassInfo, \ClassWarning,</code>	<code>\DeclareMicrotypeVariants,</code>
<code>\AtEndEnvironment,</code>	<code>\ClassWarningNoLine,</code>	<code>\DeclareOldFontCommand,</code>
<code>\AtEndOfClass,</code>	<code>\cleaders, \cleardoublepage,</code>	<code>\DeclareOption,</code>
<code>\AtEndOfPackage,</code>	<code>\clearpage, \cleartabs,</code>	<code>\DeclarePreloadSizes,</code>
<code>\AtEndPreamble, \atop,</code>	<code>\cline, \closein, \closeout,</code>	<code>\DeclareRobustCommand,</code>
<code>\atopwithdelims, \author,</code>	<code>\clubpenalty, \clubsuit,</code>	<code>\DeclareSizeFunction,</code>
<code>\author, \autodot, \b,</code>	<code>\colon, \color, \columns,</code>	<code>\DeclareSymbolFont,</code>
<code>\backslash, \badness,</code>	<code>\columnsep, \columnseprule,</code>	<code>\DeclareSymbolFontAlphabet,</code>

B. List of Known \LaTeX Control Sequences

<code>\DeclareTextAccent,</code>	<code>\endflushleft,</code>	<code>\frac, \frame, \framebox,</code>
<code>\DeclareTextAccentDefault,</code>	<code>\endflushright, \endgraf,</code>	<code>\frenchspacing, \frown,</code>
<code>\DeclareTextCommand,</code>	<code>\endgroup, \endinput,</code>	<code>\fussy, \futurelet,</code>
<code>\DeclareTextCommandDefault,</code>	<code>\endinsert, \enditemize,</code>	<code>\gamma, \Gamma, \gcd, \ge,</code>
<code>\DeclareTextComposite,</code>	<code>\endline, \endlinechar,</code>	<code>\GenericError, \GenericInfo,</code>
<code>\DeclareTextCompositeCommand,</code>	<code>\endlist, \endlrbox, \endmath,</code>	<code>\GenericWarning, \geq,</code>
<code>\DeclareTextFontCommand,</code>	<code>\endminipage, \endnote,</code>	<code>\gets, \GetTranslation,</code>
<code>\DeclareTextSymbol,</code>	<code>\endpicture, \endsloppypar,</code>	<code>\GetTranslationFor, \gdef,</code>
<code>\DeclareTextSymbolDefault,</code>	<code>\endtabbing, \endtabular,</code>	<code>\gg, \global, \globaldefs,</code>
<code>\DeclareTranslation,</code>	<code>\endtrivlist, \endverbatim,</code>	<code>\glossary, \goodbreak,</code>
<code>\DeclareTranslationFallback,</code>	<code>\enlargethispage, \enskip,</code>	<code>\grave, \H, \halign, \hang,</code>
<code>\def, \defaultthyphenchar,</code>	<code>\enspace, \ensuremath,</code>	<code>\hangafter, \hangindent,</code>
<code>\defaultscriptratio,</code>	<code>\enumerate, \epsilon,</code>	<code>\hat, \hbadness, \hbar, \hbox,</code>
<code>\defaultscriptscriptratio,</code>	<code>\equalign, \equalignno,</code>	<code>\headheight, \headline,</code>
<code>\defaultskewchar,</code>	<code>\eqnarray, \eqno, \equation,</code>	<code>\headsep, \heartsuit,</code>
<code>\deffootnote,</code>	<code>\equiv, \errhelp, \errmessage,</code>	<code>\height, \hfil, \hfill,</code>
<code>\deffootnotemark,</code>	<code>\errorcontextlines,</code>	<code>\hfilneg, \hfuzz, \hglue,</code>
<code>\definecolor, \deg, \delcode,</code>	<code>\errorstopmode, \escapechar,</code>	<code>\hideskip, \hidewidth, \hline,</code>
<code>\delimiter, \delimiterfactor,</code>	<code>\eta, \evensidemargin,</code>	<code>\hoffset, \holdinginserts,</code>
<code>\delimitershortfall,</code>	<code>\everycr, \everydisplay,</code>	<code>\hom, \hookleftarrow,</code>
<code>\delta, \Delta, \depth,</code>	<code>\everyhbox, \everyjob,</code>	<code>\hookrightarrow, \hphantom,</code>
<code>\descriptionlabel,</code>	<code>\everymath, \everypar,</code>	<code>\hrule, \hrulefill, \hsize,</code>
<code>\det, \dh, \DH, \Diamond,</code>	<code>\everyvbox, \ExecuteOptions,</code>	<code>\hskip, \hspace, \hss,</code>
<code>\diamond, \diamondsuit,</code>	<code>\exhyphenpenalty, \exists,</code>	<code>\ht, \huge, \Huge, \hypersetup,</code>
<code>\dim, \dimen, \dimendef,</code>	<code>\exp, \expandafter,</code>	<code>\hyphenation, \hyphenchar,</code>
<code>\dimexpr, \DisableLigatures,</code>	<code>\expandonce, \extracolsep,</code>	<code>\hyphenpenalty, \i, \I,</code>
<code>\discretionary,</code>	<code>\fam, \fbox, \fboxrule,</code>	<code>\ialign, \if, \ifblank,</code>
<code>\displayindent,</code>	<code>\fboxsep, \fi, \filbreak,</code>	<code>\ifbool, \ifboolexpe,</code>
<code>\displaylimits,</code>	<code>\filecontents, \fill,</code>	<code>\ifboolexpr, \ifcase,</code>
<code>\displaylines,</code>	<code>\finalhypendemerits,</code>	<code>\ifcat, \ifcsdef, \ifcsname,</code>
<code>\displaymath, \displaystyle,</code>	<code>\firstmark, \fivern,</code>	<code>\ifdim, \ifdef, \ifeof, \iff,</code>
<code>\displaywidowpenalty,</code>	<code>\fivebf, \fivei, \fivesy,</code>	<code>\iffalse, \IfFileExists,</code>
<code>\displaywidth, \div,</code>	<code>\flat, \floatingpenalty,</code>	<code>\ifhbox, \ifhmode, \ifinlist,</code>
<code>\divide, \dj, \DJ, \do,</code>	<code>\floatpagefraction,</code>	<code>\ifinner, \ifmmode, \ifnum,</code>
<code>\document, \documentclass,</code>	<code>\floatsep, \flushbottom,</code>	<code>\ifodd, \ifstr, \ifstrempty,</code>
<code>\documentstyle, \dospecials,</code>	<code>\flushleft, \flushright,</code>	<code>\ifstrequal, \iftrue,</code>
<code>\dosupereject, \dot,</code>	<code>\fmtname, \fmtversion,</code>	<code>\ifvbox, \ifvmode, \ifvoid,</code>
<code>\doteq, \dotfill, \dots,</code>	<code>\fnsymbol, \folio, \font,</code>	<code>\ifx, \ignorespaces,</code>
<code>\doublehyphendemerits,</code>	<code>\fontdimen, \fontencoding,</code>	<code>\ignorespacesafterend,</code>
<code>\doublerulesep, \downarrow,</code>	<code>\fontfamily, \fontname,</code>	<code>\Im, \imath, \immediate, \in,</code>
<code>\Downarrow, \downbracefill,</code>	<code>\fontseries, \fontshape,</code>	<code>\include, \includeonly,</code>
<code>\dp, \edef, \egroup,</code>	<code>\fontsize, \fontspec,</code>	<code>\indent, \inf, \infty,</code>
<code>\eject, \ell, \else, \em,</code>	<code>\fontsubfuzz, \footins,</code>	<code>\indent, \index, \input,</code>
<code>\emergencystretch,</code>	<code>\footline, \footnote,</code>	<code>\InputIfFileExists,</code>
<code>\emph, \empty, \emptyset,</code>	<code>\footnotemark, \footnoterule,</code>	<code>\inputlineno, \insert,</code>
<code>\endarray, \endcenter,</code>	<code>\footnotesep, \footnotesize,</code>	<code>\insertpenalties, \int,</code>
<code>\endcsname, \enddisplaymath,</code>	<code>\footnotetext, \footskip,</code>	<code>\interdisplaylinepenalty,</code>
<code>\enddocument, \endenumerate,</code>	<code>\forall, \forlistloop,</code>	<code>\interfootnotelinepenalty,</code>
<code>\endeqnarray, \endequation,</code>	<code>\foreignlanguage,</code>	<code>\interlinepenalty,</code>

B. List of Known L^AT_EX Control Sequences

<code>\intertextsep, \intop, \iota,</code> <code>\it, \item. itemindent,</code> <code>\itemitem, \itemize,</code> <code>\itemsep, \iterate, \itfam,</code> <code>\itshape, \j, \jmath, \jobname,</code> <code>\Join, \joinrel, \jot, \k,</code> <code>\kappa, \ker, \kern, \kill,</code> <code>\KOMAOption, \KOMAOptions,</code> <code>\l, \L, \label, \labelsep,</code> <code>\labelwidth, \labelenumi,</code> <code>\labelenumii, \labelenumiii,</code> <code>\labelenumiv, \labelitemi,</code> <code>\labelitemii, \labelitemiii,</code> <code>\labelitemiv, \lambda,</code> <code>\Lambda, \land, \langle,</code> <code>\language, \large, \Large,</code> <code>\LARGE, \lastbox, \lastkern,</code> <code>\lastpenalty, \lastskip,</code> <code>\LaTeX, \LaTeXe, \lbrace,</code> <code>\lbrack, \lccode, \lceil,</code> <code>\ldotp, \ldots, \le, \leaders,</code> <code>\leadsto, \leavevmode, \left,</code> <code>\leftarrow, \Leftarrow,</code> <code>\leftarrowfill, \lefteqn,</code> <code>\leftharpoonupdown,</code> <code>\leftharpoonup,</code> <code>\lefthyphenmin,</code> <code>\leftline, \leftmargin,</code> <code>\leftmargini, \leftmarginii,</code> <code>\leftmarginiii,</code> <code>\leftmarginiv, \leftmarginv,</code> <code>\leftmarginvi, \leftmark,</code> <code>\leftskip, \leq, \rightarrow,</code> <code>\Leftrightarrow, \leq,</code> <code>\leqalignno, \leqno,</code> <code>\let, \lfloor, \limits,</code> <code>\linepenalty, \lineskip,</code> <code>\lineskiplimits, \lg, \lggroup,</code> <code>\lhd, \lhook, \lim, \liminf,</code> <code>\limsup, \line, \linebreak,</code> <code>\linespread, \linethickness,</code> <code>\linewidth, \list, \listadd,</code> <code>\listfiles, \listfiles,</code> <code>\listparindent, \ll, \llap,</code> <code>\lmoustache, \ln, \lnot,</code> <code>\LoadClassWithOptions,</code> <code>\LoadClass, \LoadDictionary,</code> <code>\LoadDictionaryFor,</code> <code>\LoadMicrotypeFile, \log,</code>	<code>\long, \longleftarrow,</code> <code>\Longleftarrow,</code> <code>\longrightarrow,</code> <code>\Longrightarrow,</code> <code>\longmapsto, \longrightarrow,</code> <code>\loop, \looseness, \lor,</code> <code>\lower, \lowercase,</code> <code>\lq, \lslig, \lsstyle,</code> <code>\lstinline, \lstinputlisting,</code> <code>\lrbox, \ltx@ifnextchar,</code> <code>\LuaLaTeX, \LuaTeX, \mag,</code> <code>\magnification, \magstep,</code> <code>\magstephalf, \makeatletter,</code> <code>\makeatother, \makebox,</code> <code>\makefootline, \makeglossary,</code> <code>\makeheadline, \makeindex,</code> <code>\makelabel, \MakeLowercase,</code> <code>\maketitle, \MakeUppercase,</code> <code>\mapsto, \mapstochar,</code> <code>\marginpar, \marginparpush,</code> <code>\marginparsep,</code> <code>\marginparwidth, \mark,</code> <code>\markboth, \markright,</code> <code>\math, \mathaccent,</code> <code>\mathbf, \mathbin,</code> <code>\mathchar, \mathchardef,</code> <code>\mathchoice, \mathclose,</code> <code>\mathcode, \mathellipsis,</code> <code>\mathgroup, \mathhexbox,</code> <code>\mathinner, \mathit,</code> <code>\mathop, \mathopen, \mathord,</code> <code>\mathpalette, \mathparagraph,</code> <code>\mathpunct, \mathrel,</code> <code>\mathrm, \mathsection,</code> <code>\mathsf, \mathsterling,</code> <code>\mathstrut, \mathsurround,</code> <code>\mathhtt, \mathunderscore,</code> <code>\mathversion, \matrix,</code> <code>\max, \maxdeadcycles,</code> <code>\maxdepth, \maxdimen,</code> <code>\mbox, \mdseries, \meaning,</code> <code>\medbreak, \medmuskip,</code> <code>\medskip, \medskipamount,</code> <code>\message, \MessageBreak,</code> <code>\mho, \microtypecontext,</code> <code>\microtypesetup, \mid,</code> <code>\midinsert, \min, \minipage,</code> <code>\mit, \mkern, \models, \month,</code> <code>\moveleft, \moveright,</code>	<code>\mp, \mscount, \mskip, \mu,</code> <code>\multicolumn, \multiply,</code> <code>\multirow, \multispan,</code> <code>\muskip, \muskipdef, \nabla,</code> <code>\narrower, \natural, \ne,</code> <code>\nearrow, \NeedsTeXFormat,</code> <code>\neg, \negthinspace,</code> <code>\neq, \newbox, \newbool,</code> <code>\newcommand, \newcount,</code> <code>\newcounter, \newdimen,</code> <code>\newenvironment, \newfam,</code> <code>\newfont, \newfontfamily,</code> <code>\newhelp, \newif,</code> <code>\newinsert, \newlabel,</code> <code>\newlanguage, \newlength,</code> <code>\newline, \newlinechar,</code> <code>\newmathalphabet, \newmuskip,</code> <code>\newpage, \newread,</code> <code>\newrobustcmd, \newsavebox,</code> <code>\newskip, \newtheorem,</code> <code>\newtoks, \NewTranslation,</code> <code>\newwrite, \next, \ng, \NG,</code> <code>\ni, \noalign, \noboundary,</code> <code>\nobreak, \nobreakspace,</code> <code>\nocite, \nocorr, \nocorrlist,</code> <code>\node, \noexpand, \nofiles,</code> <code>\noindent, \nointerlineskip,</code> <code>\nolimits, \nolinebreak,</code> <code>\nonfrenchspacing,</code> <code>\nonscript, \nonstopmode,</code> <code>\nonumber, \nopagebreak,</code> <code>\nopagenumbers,</code> <code>\normalbaselines,</code> <code>\normalbaselineskip,</code> <code>\normalbottom,</code> <code>\normalcolor, \normalfont,</code> <code>\normalmarginpar,</code> <code>\normallineskip,</code> <code>\normallineskiplimit,</code> <code>\normalsize, \notblank,</code> <code>\nopagebreak, \not,</code> <code>\notin, \nu, \null,</code> <code>\nulldelimiterspace,</code> <code>\nullfont, \number,</code> <code>\numberline, \numexpr,</code> <code>\narrow, \o, \O, \oalign,</code> <code>\obeycr, \obeylines,</code> <code>\obeyspaces, \oddsidemargin,</code> <code>\odot, \oe, \OE, \of,</code>
--	--	---

B. List of Known \LaTeX Control Sequences

<code>\offinterlineskip,</code>	<code>\preto,\pretolerance,</code>	<code>\SaveTranslationFor,</code>
<code>\oint,\ointop,\oldstyle,</code>	<code>\prevdepth,\prevgraf,</code>	<code>\sb,\sbox,\scriptfont,</code>
<code>\oldstylenums,\omega,</code>	<code>\prime,\printacronyms,</code>	<code>\scriptscriptfont,</code>
<code>\Omega,\ominus,\omit,</code>	<code>\printbibliography,</code>	<code>\scriptscriptstyle,</code>
<code>\onecolumn,\oalign,</code>	<code>\printendnotes,\printindex,</code>	<code>\scriptsize,\scriptspace,</code>
<code>\openin,\openout,\openup,</code>	<code>\ProcessPgfoptions,</code>	<code>\scriptstyle,\scrollmode,</code>
<code>\oplus,\OptionNotUsed,</code>	<code>\ProcessOptions,\proclaim,</code>	<code>\scshape,\searrow,</code>
<code>\or,\oslash,\otimes,</code>	<code>\prod,\propto,\protect,</code>	<code>\sec,\secdef,\section,</code>
<code>\othersectionlevelsformat,</code>	<code>\protected,\protecting,</code>	<code>\sectionmark,\selectfont,</code>
<code>\outer,\output,</code>	<code>\providecommand,</code>	<code>\selectlanguage,</code>
<code>\outputpenalty,\oval,\over,</code>	<code>\providerobustcmd,</code>	<code>\setbox,\setcapindent,</code>
<code>\overbrace,\overfullrule,</code>	<code>\ProvidesClass,</code>	<code>\setcounter,\SetExpansion,</code>
<code>\overleftarrow,\overline,</code>	<code>\ProvideDictionaryFor,</code>	<code>\SetExtraKerning,</code>
<code>\overrightarrow,</code>	<code>\ProvidesFile,</code>	<code>\SetExtraSpacing,\setfnpct,</code>
<code>\overwithdelims,\owns,\P,</code>	<code>\ProvidesPackage,</code>	<code>\setkomafont,\setlanguage,</code>
<code>\PackageError,\PackageInfo,</code>	<code>\ProvideTextCommand,</code>	<code>\setlength,\setmainfont,</code>
<code>\PackageWarning,</code>	<code>\psi,\Psi,\pushtabs,</code>	<code>\setmainlanguage,</code>
<code>\PackageWarningNoLine,</code>	<code>\put,\qbezier,\qbeziermax,</code>	<code>\SetMathAlphabet,</code>
<code>\pagebody,\pagebreak,</code>	<code>\qqquad,\quad,\r,\radical,</code>	<code>\setminus,\setmonofont,</code>
<code>\pagecontents,\pagedepth,</code>	<code>\raggedbottom,\raggedleft,</code>	<code>\setotherlanguage,</code>
<code>\pagefilllstretch,</code>	<code>\RaggedLeft,\raggedright,</code>	<code>\setotherlanguages,</code>
<code>\pagefillstretch,</code>	<code>\RaggedRight,\raise,</code>	<code>\SetProtrusion,\setsansfont,</code>
<code>\pagefilstretch,</code>	<code>\raisebox,\rangle,\rbrace,</code>	<code>\SetSymbolFont,\settabs,</code>
<code>\pagegoal,\pageinsert,</code>	<code>\rbrack,\rceil,\Re,\read,</code>	<code>\settodepth,\settoheight,</code>
<code>\pageno,\pagenumbering,</code>	<code>\ref,\refstepcounter,</code>	<code>\settowidth,\SetTracking,</code>
<code>\pageref,\pageshrink,</code>	<code>\relax,\relbar,\Relbar,</code>	<code>\sevenbf,\seveni,</code>
<code>\pagestretch,\pagestyle,</code>	<code>\relpenalty,\removelastskip,</code>	<code>\sevensy,\sevenrm,\sfcode,</code>
<code>\pagetotal,\paperheight,</code>	<code>\renewcommand,</code>	<code>\sffamily,\sharp,\shipout,</code>
<code>\paperwidth,\par,\paragraph,</code>	<code>\renewenvironment,</code>	<code>\shorthandoff,\shorthandon,</code>
<code>\paragraphmark,\parallel,</code>	<code>\renewrobustcmd,</code>	<code>\shortstack,\show,</code>
<code>\parbox,\parfillskip,</code>	<code>\RenewTranslation,</code>	<code>\showbox,\showboxbreadth,</code>
<code>\parindent,\parsep,</code>	<code>\repeat,\RequirePackage,</code>	<code>\showboxdepth,\showhyphens,</code>
<code>\parshape,\parskip,\part,</code>	<code>\RequirePackageWithOptions,</code>	<code>\showlists,\showoutput,</code>
<code>\partformat,\partial,</code>	<code>\restorecr,</code>	<code>\showoverfull,\showthe,</code>
<code>\partname,\partopsep,</code>	<code>\reversemarginpar,\rfloor,</code>	<code>\sidenote,\sigma,\Sigma,</code>
<code>\PassOptionsToClass,</code>	<code>\rgroup,\rhd,\rho,\rhohook,</code>	<code>\sim,\simeq,\sin,\sinh,</code>
<code>\PassOptionsToPackage,</code>	<code>\right,\rightarrow,</code>	<code>\skew,\skewchar,\skip,</code>
<code>\patchcmd,\patterns,</code>	<code>\Rightarrow,\rightarrowfill,</code>	<code>\skipdef,\sl,\slash,</code>
<code>\pausing,\pdfLaTeX,</code>	<code>\rightharpoonupdown,</code>	<code>\slfam,\sloppy,\sloppypar,</code>
<code>\pdfstringdefDisableCommands,</code>	<code>\rightharpoonup,</code>	<code>\slshape,\small,\smallbreak,</code>
<code>\pdfTeX,\penalty,\perp,</code>	<code>\righthyphenmin,</code>	<code>\smallint,\smallskip,</code>
<code>\pgfkeys,\phantom,\phi,</code>	<code>\rightleftharpoons,</code>	<code>\smallskipamount,\smash,</code>
<code>\Phi,\pi,\Pi,\picture,</code>	<code>\rightline,\rightmargin,</code>	<code>\smile,\sp,\space,</code>
<code>\plainoutput,\pm,</code>	<code>\rightmark,\rightskip,</code>	<code>\spacefactor,\spaceskip,</code>
<code>\pmatrix,\pmod,\poptabs,</code>	<code>\rlap,\rm,\rmfamily,</code>	<code>\spadesuit,\span,</code>
<code>\postdisplaypenalty,</code>	<code>\rmoustache,\robustify,</code>	<code>\special,\splitbotmark,</code>
<code>\pounds,\Pr,\prec,\preceq,</code>	<code>\roman,\Roman,\romannumeral,</code>	<code>\splitfirstmark,</code>
<code>\predisdisplaypenalty,</code>	<code>\root,\rootbox,\rq,\rule,\S,</code>	<code>\splitmaxdepth,</code>
<code>\predisplaysize,\preloaded,</code>	<code>\samepage,\SaveTranslation,</code>	<code>\splittopskip,\sqcap,\sqcup,</code>

B. List of Known \LaTeX Control Sequences

\backslash sqrt, \backslash sqsubset, \backslash sqsubseteq, \backslash sqsupset, \backslash sqsupseteq, \backslash ss, \backslash SS, \backslash stackrel, \backslash star, \backslash stepcounter, \backslash stop, \backslash stretch, \backslash string, \backslash strut, \backslash strutbox, \backslash subparagraph, \backslash subparagraphmark, \backslash subsection, \backslash subsectionmark, \backslash subset, \backslash subseteq, \backslash subsubsection, \backslash subsubsectionmark, \backslash succ, \backslash succeq, \backslash sum, \backslash sup, \backslash supereject, \backslash suppressfloats, \backslash supset, \backslash supseteq, \backslash surd, \backslash swarrow, \backslash symbol, \backslash t, \backslash tabalign, \backslash tabbing, \backslash tabbingsep, \backslash tabcolsep, \backslash tableofcontents, \backslash tabskip, \backslash tabular, \backslash tabularnewline, \backslash tabs, \backslash tabsdone, \backslash tabsyet, \backslash tan, \backslash tanh, \backslash tau, \backslash tenbf, \backslash tenex, \backslash teni, \backslash tenit, \backslash tenrm, \backslash tensl, \backslash tensy, \backslash tentt, \backslash TeX, \backslash test, \backslash textasciicircum, \backslash textasciitilde, \backslash textasteriskcentered, \backslash textbackslash, \backslash textbar, \backslash textbf, \backslash textbraceleft, \backslash textbraceright, \backslash textbullet, \backslash textcircled, \backslash textcompwordmark, \backslash textcolor, \backslash textdagger, \backslash textdaggerdbl, \backslash textdollar, \backslash textellipsis, \backslash textemdash, \backslash textemdash, \backslash textenglish, \backslash textexclamdown, \backslash textexclamdown, \backslash textfloatsep, \backslash textfont, \backslash textfraction, \backslash textgreater, \backslash textindent, \backslash textit, \backslash textheight, \backslash textless, \backslash textls, \backslash textmicrotypecontext, \backslash textmd, \backslash textnormal,	\backslash textogonekcentered, \backslash textparagraph, \backslash textperiodcentered, \backslash textquestiondown, \backslash textquotedbl, \backslash textquotedblleft, \backslash textquotedblright, \backslash textquoteleft, \backslash textquoteright, \backslash textregistered, \backslash textrm, \backslash textsc, \backslash textsection, \backslash textsf, \backslash textsl, \backslash textsterling, \backslash textstyle, \backslash textsuperscript, \backslash textsection, \backslash textsubscript, \backslash textsuperscript, \backslash textsterling, \backslash TextSymbolUnavailable, \backslash texttt, \backslash texttrademark, \backslash textunderscore, \backslash textup, \backslash textvisiblespace, \backslash textwidth, \backslash th, \backslash TH, \backslash thanks, \backslash the, \backslash theendnotes, \backslash theenumi, \backslash theenumii, \backslash theenumiii, \backslash theenumiv, \backslash thefootnote, \backslash thefootnotemark, \backslash thempfn, \backslash thempfn, \backslash thempfootnote, \backslash thepage, \backslash thepart, \backslash theta, \backslash Theta, \backslash thicklines, \backslash thickmuskip, \backslash thinlines, \backslash thinmuskip, \backslash thinspace, \backslash thispagestyle, \backslash tikz, \backslash tilde, \backslash time, \backslash times, \backslash tiny, \backslash title, \backslash to, \backslash today toks, \backslash toksdef, \backslash tolerance, \backslash top, \backslash topfigrule, \backslash topfraction, \backslash topglue, \backslash topins, \backslash topinsert, \backslash topmargin, \backslash topmark, \backslash topsep, \backslash topskip, \backslash totalheight, \backslash tracingall, \backslash tracingcommands, \backslash tracingfonts, \backslash tracinglostchars, \backslash tracingmacros,	\backslash tracingonline, \backslash tracingoutput, \backslash tracingpages, \backslash tracingparagraphs, \backslash tracingrestores, \backslash tracingstats, \backslash triangle, \backslash triangleleft, \backslash triangleright, \backslash trivlist, \backslash tt, \backslash ttfam, \backslash ttfamily, \backslash ttraggedright, \backslash two@digits, \backslash twocolumn, \backslash typein, \backslash typeout, \backslash u, \backslash uccode, \backslash uchyph, \backslash unboldmath, \backslash underbar, \backslash underbrace, \backslash underline, \backslash unhbox, \backslash unhcopy, \backslash unitlength, \backslash unkern, \backslash unlhd, \backslash unpenalty, \backslash unrhd, \backslash unskip, \backslash unvbox, \backslash unvcopy, \backslash uparrow, \backslash Uparrow, \backslash upbracefill, \backslash updownarrow, \backslash Updownarrow, \backslash uplus, \backslash uppercase, \backslash upsilon, \backslash Upsilon, \backslash upshape, \backslash usebox, \backslash usecounter, \backslash usefont, \backslash UseMicrotypeSet, \backslash usepackage, \backslash UseTextAccent, \backslash UseTextSymbol, \backslash upshape, \backslash v, \backslash vadjust, \backslash valign, \backslash value, \backslash varepsilon, \backslash varphi, \backslash varpi, \backslash varrho, \backslash varsigma, \backslash vartheta, \backslash vbadness, \backslash vbox, \backslash vcenter, \backslash vdash, \backslash vdots, \backslash vec, \backslash vector, \backslash vee, \backslash verb, \backslash verbatim, \backslash vert, \backslash Vert, \backslash vfil, \backslash vfill, \backslash vfilneg, \backslash vfootnote, \backslash vfuzz, \backslash vglue, \backslash vline, \backslash voffset, \backslash vphantom, \backslash vrule, \backslash vsize, \backslash vskip, \backslash vspace, \backslash vsplit, \backslash vss, \backslash vtop, \backslash wd, \backslash wedge, \backslash widehat, \backslash widetilde, \backslash widowpenalty, \backslash width, \backslash wlog, \backslash wp, \backslash wr, \backslash write, \backslash xdef, \backslash xdefinecolor, \backslash XeLaTeX, \backslash XeTeX xi, \backslash Xi, \backslash xleaders, \backslash xspaceskip, \backslash year, \backslash zeta
--	---	--

C. List of Known L^AT_EX Environments

Below are listed all *predefined* control sequence names that are treated as “silent” names by **CNLT_X**, that is, those defined by **CNLT_X-LISTINGS**.

array, center, description,	itemize, labeling, list,	sloppypar, tabbing, table,
displaymath, document,	longtable, lrbox, math,	tabu, tabular, tabularx,
enumerate, eqnarray, equation,	minipage, otherlanguage,	tabulary, trivlist, verbatim
figure, flushleft, flushright,	picture, quote, quoting,	

D. List of Entries in cnltx.bib

All entries in `cnltx.bib` are entries of the `@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.

pkg:abbrevs (@package),	cls:exam (@class),	pkg:marginnote (@package),
pkg:accsupp (@package),	pkg:examdesign (@package),	pkg:mdframed (@package),
pkg:acro (@package),	pkg:exercise (@package),	pkg:mfirstuc (@package),
pkg:acromake (@package),	pkg:exsol (@package),	pkg:multicol (@package),
pkg:acronym (@package),	pkg:glossaries (@package),	pkg:multienum (@package),
pkg:acroterm (@package),	pkg:hologo (@package),	pkg:nomencl (@package),
pkg:answers (@package),	pkg:hyperref (@package),	pkg:pgf (@package),
pkg:babel (@package),	pkg:idxcmds (@package),	pkg:pgfopts (@package),
pkg:biblatex (@package),	pkg:ifluatex (@package),	pkg:polyglossia (@package),
pkg:catchfile (@package),	pkg:ifxetex (@package),	pkg:probsoln (@package),
pkg:chngcntr (@package),	pkg:imakeidx (@package),	pkg:ragged2e (@package),
bnd:cnltx (@bundle),	bnd:koma-script (@bundle),	pkg:translations (@package),
cls:cnpkgdoc (@class),	bnd:l3experimental (@bundle),	pkg:trimspaces (@package),
pkg:enumitem (@package),	bnd:l3kernel (@bundle),	pkg:ulem (@package),
pkg:environ (@package),	bnd:l3packages (@bundle),	pkg:xcolor (@package),
pkg:eqexam (@package),	pkg:libertine (@package),	pkg:xspace (@package)
pkg:esami (@package),	pkg:listings (@package),	
pkg:etoolbox (@package),	pkg:ltxcmds (@package),	

E. Bibliography

- [Ars11] Donald Arseneau. `ulem`, Mar. 18, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/ulem/>.
- [Dan13] Marco Daniel. `mdframed`. version 1.9b, July 1, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/mdframed/>.
- [Gre13] Enrico Gregorio. `imakeidx`. version 1.3a, July 11, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/imakeidx/>.
- [HM13] Carsten Heinz and Brooks Moses, current maintainer: Jobst Hoffmann.
`listings`. version 1.5b, Aug. 26, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/listings/>.

E. Bibliography

- [Kero7] Uwe Kern. xcolor. version 2.11, Jan. 21, 2007.
URL: <http://mirror.ctan.org/macros/latex/contrib/xcolor/>.
- [KN12] Markus Kohm and Frank Neukahm. KOMA-Script. version 3.11b, July 29, 2012.
URL: <http://mirror.ctan.org/macros/latex/contrib/koma-script/>.
- [Koh12] Markus Kohm. marginnote. version 1.1i, Mar. 29, 2012.
URL: <http://mirror.ctan.org/macros/latex/contrib/marginnote/>.
- [Leh11] Philipp Lehman. etoolbox. version 2.1, Jan. 3, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/etoolbox/>.
- [Leh13] Philipp Lehman, current maintainers: Joseph Wright, Audrey Boruvka, and Philip Kime. biblatex. version 2.7a, July 14, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/biblatex/>.
- [Mit11] Frank Mittelbach. multicol. version 1.7a, June 27, 2011.
URL: <http://mirror.ctan.org/macros/latex/required/tools/multicol/>.
- [Nie13a] Clemens Niederberger. acro. version 1.4b, Sept. 24, 2013.
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/>.
- [Nie13d] Clemens Niederberger. translations. version 1.1a, Sept. 30, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/translations/>.
- [Obe10] Heiko Oberdiek. accsupp. version 0.3, Jan. 16, 2010.
URL: <http://mirror.ctan.org/macros/latex/contrib/oberdiek/>.
- [Obe11a] Heiko Oberdiek. catchfile. version 1.6, Mar. 1, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/oberdiek/>.
- [Obe11b] Heiko Oberdiek. ltxcmds. version 1.22, Nov. 9, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/oberdiek/>.
- [OR12] Heiko Oberdiek and Sebastian Rahtz. hyperref. version 6.83m, Nov. 6, 2012.
URL: <http://mirror.ctan.org/macros/latex/contrib/hyperref/>.
- [Rob09] Will Robertson. trimspaces. version 1.1, Sept. 17, 2009.
URL: <http://mirror.ctan.org/macros/latex/contrib/trimspaces/>.
- [Scho9] Martin Schröder. ragged2e. version 2.1, May 21, 2009.
URL: <http://mirror.ctan.org/macros/latex/contrib/ms/ragged2e/>.
- [Tal13] Nicola L.C. Talbot. glossaries. version 3.05, Apr. 21, 2013.
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/>.

E. Bibliography

- [Wri11] Joseph Wright. pgfopts. version 2.1, June 2, 2011.
URL: <http://mirror.ctan.org/macros/latex/contrib/pgfopts/>.

F. Index

A	
<code>abstract</code>	28, 30 f.
<code>abstract-format</code>	28
<code>abstract-width</code>	28
<code>accsupp</code> (package).....	5, 48 f.
<code>acro</code> (package).....	23
<code>acronym-format</code>	23
<code>\AD</code>	26
<code>add-bib</code>	4, 33, 35
<code>add-cmds</code>	17 f., 36, 48
<code>add-envs</code>	18, 36, 48
<code>add-frame-options</code>	17, 36
<code>add-index</code>	4, 29, 33, 35
<code>add-listings-options</code>	18, 36
<code>add-silent-cmds</code>	17 f., 36
<code>add-silent-envs</code>	18, 36
<code>after-code</code>	11
<code>after-source</code>	11
<code>\AM</code>	26
<code>arg-format</code>	28
<code>\argumentformat</code>	27
Arseneau, Donald.....	5
<code>authors</code>	29
B	
<code>babel</code> (package).....	24, 29
<code>babel-options</code>	29
<code>bar</code>	15
<code>baz</code>	27 f.
<code>\BC</code>	26
<code>biblatex</code> (package).....	3 f., 34 f.
<code>\boolkey</code>	8
Boruvka, Audrey.....	34
<code>build-title</code>	30
C	
<code>caption-font</code>	28
<code>caption-label-font</code>	28
Carlisle, David.....	23
<code>catchfile</code> (package).....	5
<code>\cf</code>	25
<code>\changedversion</code>	8
<code>chngcntr</code> (package).....	35
<code>\choicekey</code>	8
<code>\choices</code>	8
<code>class</code>	29 f.
<code>\classformat</code>	27
<code>\cls</code>	9, 39
<code>cls-format</code>	28
<code>\clsidx</code>	9
<code>\cnltx@date</code>	44
<code>\cnltx@info</code>	44
<code>\cnltx@version</code>	44
<code>\cnltx@accsupp</code>	49
<code>\cnltx@babel@options</code>	33
<code>\cnltx@base@error</code>	44
<code>\cnltx@base@info</code>	44
<code>\cnltx@base@warning</code>	44
<code>\cnltx@base@warningnoline</code>	44
<code>\cnltx@bibtex@listings@style</code>	38
<code>\cnltx@caption@font</code>	33
<code>\cnltx@captionlabel@font</code>	33
<code>\cnltx@copyablespace</code>	48
<code>\cnltx@create@bundle@message</code>	44
<code>\cnltx@create@generic@message</code>	44
<code>\cnltx@create@message</code>	44
<code>\cnltx@define@colorscheme</code>	44
<code>\cnltx@doc@error</code>	46
<code>\cnltx@doc@info</code>	46
<code>\cnltx@doc@warning</code>	46
<code>\cnltx@doc@warningnoline</code>	46
<code>\cnltx@example@error</code>	47
<code>\cnltx@example@info</code>	47
<code>\cnltx@example@warning</code>	47
<code>\cnltx@example@warningnoline</code>	47
<code>\cnltx@expandargs</code>	45
<code>\cnltx@getfileinfo</code>	46
<code>\cnltx@gobble</code>	37 ff.
<code>\cnltx@if@in</code>	46
<code>\cnltx@ifbang</code>	45
<code>\cnltx@ifdash</code>	45
<code>\cnltx@ifisnum</code>	32, 45
<code>\cnltx@ifpunctuation</code>	19
<code>\cnltx@ifsym</code>	45
<code>\cnltx@isvalue</code>	47
<code>\cnltx@listings@error</code>	48
<code>\cnltx@listings@info</code>	48
<code>\cnltx@listings@style</code>	36, 48
<code>\cnltx@listings@warning</code>	48
<code>\cnltx@listings@warningnoline</code>	48
<code>\cnltx@long@remove@all</code>	46
<code>\cnltx@long@remove@once</code>	46
<code>\cnltx@long@replace@all</code>	46
<code>\cnltx@long@replace@once</code>	46
<code>\cnltx@mdframed@options</code>	36, 48
<code>\cnltx@package@date</code>	46
<code>\cnltx@package@info</code>	46
<code>\cnltx@package@version</code>	46
<code>\cnltx@par</code>	45
<code>\cnltx@predefined@control@sequences</code>	48
<code>\cnltx@predefined@environments</code>	49

INDEX

- `\cnltx@remove@all` 46
- `\cnltx@remove@once` 46
- `\cnltx@replace@all` 46
- `\cnltx@replace@once` 46
- `\cnltx@stripbs` 46
- `\cnltx@tools@error` 49
- `\cnltx@tools@info` 49
- `\cnltx@tools@warning` 49
- `\cnltx@tools@warningnoline` 49
- `\cnltx@trailpunct` 19
- `\cnltx@treat@lst@index` 48
- `\cnltx@version@note` 47
- `\cnltxacronym` 23
- `\cnltxat` 47 f.
- `\cnltxbang` 47
- `\cnltxequal` 47
- `\cnltxlatin` 23
- `\cnltxletterat` 47
- `cnltxlist` (environment) 47
- `\cnltxotherat` 47
- `cnltxquote` (environment) 30
- `\cnltxtimeformat` 23, 26
- `cnpkgdoc` (class) 3
- `\code` 6 f., 10 ff., 16 f., 27 f., 48 f.
- `code-font` 28
- `code-left` 17
- `code-only` 16
- `code-sep` 17
- `\codefont` 27 f.
- `\command` 12 f.
- `commands` (environment) 10, 12, 47
- `\cs` 6, 12 f., 27 f., 37 ff., 41 f., 47
- `\csidx` 6, 17, 48
- `\CTAN` 9
- `CTAN` 5, 8 f., 29, 33
- `\ctan` 8 f., 23
- `\CTANurl` 9

- D**
- Daniel, Marco 3, 5, 17, 36, 48
- `\Darg` 7
- `\darg` 7
- `date` 30
- `\DeclareCounterRepresentation` 20
- `\DeclareTranslation` 42
- `\defabbr` 23
- `\Default` 12 ff.
- `\default` 8, 12, 28, 41
- `default-format` 28
- `\definecolorscheme` 40 ff.
- `\Dsh` 25
- `\dsh` 25 f.

- E**
- `\eg` 25
- `email` 30 f.
- `\env` 6, 16, 37 ff., 41 f.
- `\envidx` 6, 38, 48
- `\environment` 15 f.
- `environments` (environment) 10, 15 f., 47
- `\etc` 25
- `etoolbox` (package) 5, 45
- `example` (environment) 4, 9, 11, 16 f., 48
- `\exampleformat` 27
- `\expandable` 12, 41 f.
- `expandable-sign` 13
- `\expandablesign` 12 f.
- `expl-format` 28

- F**
- `foo` 14 f.
- `frame-options` 17, 36

- G**
- `glossaries` (package) 23
- `gobble` 17
- Gregorio, Enrico 33

- H**
- Heinz, Carsten 3, 5, 17 f., 36, 48
- Hoffmann, Jobst 3, 5, 17 f., 36, 48
- `hyperref` (package) 5, 39 f.

- I**
- `idxcmds` (package) 5, 47
- `\ie` 25
- `\iftest` 45
- `imakeidx` (package) 33 f.
- `\implementation` 10
- `index-prologue` 33
- `index-setup` 33 f.
- `index-space` 34
- `index-style` 4, 33 f.
- `\indexcs` 38, 48
- `\indexenv` 48
- `info` 30
- `\inputexample` 9 f.
- `\inputsidebyside` 9 f.
- `\inputsourcecode` 10

- K**
- Kern, Uwe 5
- `key` 7 f.
- `\key` 7, 39
- `\keybool` 13 f.
- `\keychoice` 13 f.
- `\keyis` 7

INDEX

<code>\keyval</code>	13 f.	<code>\newsourcename</code>	11
Kime, Philip	34	Niederberger, Clemens	1, 3, 5, 23, 42, 47
Kohm, Markus	5	<code>\nottest</code>	45
KOMA-Script (bundle)	5		
L		O	
Lehman, Philipp	5, 34	<code>\oarg</code>	7
<code>\license</code>	8	<code>\oarg</code>	7, 12, 16
<code>list-sep</code>	49	Oberdiek, Heiko	5, 39, 48
<code>list-setup</code>	10	<code>\opt</code>	12 ff.
<code>\listbibfileentries</code>	49	<code>\option</code>	7, 12 ff., 27 f., 39, 41 f.
<code>\listbibfilekeys</code>	49	<code>\optionidx</code>	7
<code>\listbibfiletypes</code>	49	options (environment)	10, 13 f., 16, 47
listings (package)	3, 5, 17 f., 36, 38 f., 48		
<code>listings-options</code>	18, 36	P	
<code>\listsilentcmds</code>	49	<code>package</code>	29 f.
<code>\listsilentenvs</code>	49	<code>\packageformat</code>	27
<code>load-preamble</code>	5, 28 f., 32 f., 35	PDF	23, 39 f.
<code>load-preamble+</code>	29, 33, 35	pgfpts (package)	5
<code>\LPPL</code>	8, 23	<code>\pkg</code>	9, 23, 39
LPPL	5, 8, 23, 43	<code>pkg-format</code>	28
<code>\lppl</code>	8, 23	<code>\pkgidx</code>	9
ltxcmds (package)	5	<code>\PM</code>	26
		<code>pre-code</code>	11
M		<code>pre-source</code>	11
<code>makeindex-setup</code>	34	<code>\providecounterrepresentation</code>	20
<code>\MakePercentComment</code>	48	<code>\provideexpandablecmd</code>	22
<code>\Marg</code>	7, 48		
<code>\marg</code>	7, 12, 48	Q	
marginnote (package)	5	<code>quote-author-format</code>	31
mdframed (package)	3, 5, 16 f., 36, 48	<code>quote-format</code>	31
<code>\meta</code>	7, 27 f., 48		
Mittelbach, Frank	5	R	
<code>\Module</code>	13 f.	raggedze (package)	5
<code>\module</code>	7, 14, 41 f.	Rahtz, Sebastian	5, 39
<code>module-sep</code>	28	<code>\renewabbr</code>	23
<code>\moduleidx</code>	7	<code>\renewcounterrepresentation</code>	21
Moses, Brooks	3, 5, 17 f., 36, 48	<code>\renewexpandablecmd</code>	22
multicol (package)	5	Robertson, Will	5, 35
N		S	
<code>name</code>	7	<code>\sarg</code>	7, 12
<code>name</code>	7, 29	Schröder, Martin	5
<code>\needclass</code>	9	<code>scrartcl</code>	29
<code>\needpackage</code>	9	scrartcl (class)	5, 29
Neukahm, Frank	5	<code>set-trail-punct</code>	19
<code>\newabbr</code>	23 f.	<code>\setcnltx</code>	6, 18, 28, 33
<code>\newarg</code>	48	<code>side-by-side</code>	16 f.
<code>\newcounterrepresentation</code>	20 f.	sidebyside (environment)	9, 11, 17
<code>\newexpandablecmd</code>	22	<code>\sinceversion</code>	8
<code>\newinputsourcefilecmd</code>	10	<code>source-format</code>	28
<code>\newname</code>	22 f., 31	sourcecode (environment)	10 f., 16 f., 48
<code>\newnote</code>	8	<code>sourcecode-options</code>	18
<code>\newpackagename</code>	8 f.	<code>\sourceformat</code>	27, 36, 38 f.
		<code>subtitle</code>	30

INDEX

T	
Talbot, Nicola L.C.....	23
<code>title-format</code>	27
<code>\titleformat</code>	31
translations (package).....	5, 42
trimspaces (package).....	5
U	
ulem (package).....	5
<code>\unexpandable</code>	12, 41 f.
<code>url</code>	30 f.
<code>\usf</code>	25 f.
<code>\usw</code>	25 f.
<code>\uswusf</code>	25
V	
<code>\verbcode</code>	6
<code>version</code>	30
<code>version-note-format</code>	28
<code>\versionnoteformat</code>	27
<code>\Vgl</code>	26
<code>\vgl</code>	26
W	
Wilson, Peter.....	35
Wright, Joseph.....	5, 34
X	
xcolor (package).....	5, 40
Z	
<code>\ZB</code>	25
<code>\zB</code>	25 f.