

nomenc1*

A Package to Create a Nomenclature

Bernd Schandl[†]
schandl@gmx.net

Printed on November 25, 2001

1 Introduction

How often did you try to understand a theorem in a book, but just couldn't figure out what all those strange symbols were all about? The `nomenc1` package should help authors format a nomenclature. It uses the powerful capabilities of the *MakeIndex* program to generate such a list automatically using information provided by the author throughout the text.

2 Usage

2.1 The Basics

The creation of the nomenclature list is very similar to the creation of an index [6, App. A]. You need to:

- Put `\usepackage[options]{nomenc1}` in the preamble of your document.
- `\makeglossary` • Put `\makeglossary` in the preamble of your document.
- `\nomenclature` • Issue the `\nomenclature` command (see Section 2.2) for each symbol you want to have included in the nomenclature list. The best place for this command is immediately after you introduce the symbol for the first time.
- `\printglossary` • Put `\printglossary` at the place you want to have your nomenclature list.

*Package version v3.1a of 2000/12/03.

[†]Up to version v2.2 this package was maintained by Boris Veytsman.

Now put your file through L^AT_EX. The command `\makeglossary` will instruct L^AT_EX to open the glossary file `\langle filename \rangle.glo` corresponding to your L^AT_EX file `\langle filename \rangle.tex` and to write the information from your `\nomenclature` commands to this file.

The next step is to invoke *MakeIndex*. You should instruct *MakeIndex* to use `\langle filename \rangle.glo` as your input file, use `nomencl.ist` as your style file¹, and write output to the file `\langle filename \rangle.gls`. How to do this depends on your implementation of *MakeIndex*. For most UNIX implementations you should write something like

```
makeindex \langle filename \rangle.glo -s nomencl.ist -o \langle filename \rangle.gls
```

Now you have the file `\langle filename \rangle.gls` that contains your nomenclature list properly ordered. The last step is to invoke L^AT_EX on your master file `\langle filename \rangle.tex` once more. It will input your `.gls` file and process it accordingly to the current options. That's all!

2.2 The Main Command

`\nomenclature` The main command of the `nomencl` package has the following syntax:

```
\nomenclature[\langle prefix \rangle]{\langle symbol \rangle}{\langle description \rangle}
```

where `\langle prefix \rangle` is used for fine tuning the sort order, `\langle symbol \rangle` is the symbol you want to describe and `\langle description \rangle` is the actual description. The sortkey will be `\langle prefix \rangle\langle symbol \rangle`, where `\langle prefix \rangle` is either the one from the optional argument or, if no optional argument was given, the default `\langle prefix \rangle` which may be empty. See Section 3 to make sense of this.

Put this command immediately after the equation or text that introduces `\langle symbol \rangle`. Usually it is a good idea to avoid a space or an unquoted newline just in front of the `\nomenclature` command. Put a `%` at the end of the preceding line if necessary. Don't forget to enclose math in `\langle symbol \rangle` in `$` signs.

Let's have a look at a simple example. If your input file looks like the one in Figure 1 then your nomenclature² should look like Figure 2.

Note the necessary quoting of newlines. When the `\nomenclature` macros appear directly after the `equation` environment, quote *all* newlines; when they appear in the middle of a sentence, quote *all but the last* newlines.³

Due to the way `\nomenclature` scans its arguments you don't need to `\protect` any macros, but you also must not have any character in front

¹German users who want to use the shortcut notation `"a` instead of `\"a` have to redefine the quote character in `nomencl.ist` to something other than `"` (and `|`, `@`, `!`), maybe `+` or `&`; see the comment in the source code section and in the file `nomencl.ist`. Furthermore, they should consider using the `-g` switch of *MakeIndex*.

²Note that all the examples are somewhat faked in this document, but they give a good impression of the "real" result.

³I'm not sure how to resolve this more elegantly, but suggestions are welcome.

```

\documentclass{article}
\usepackage{nomencl}
\makeglossary
\begin{document}
\section*{Main equations}
\begin{equation}
a=\frac{N}{A}
\end{equation}%
\nomenclature{ $a$ }{The number of angels per unit area}%
\nomenclature{ $N$ }{The number of angels per needle point}%
\nomenclature{ $A$ }{The area of the needle point}%
The equation  $\sigma = m a$ %
\nomenclature{ $\sigma$ }{The total mass of angels per unit area}%
\nomenclature{ $m$ }{The mass of one angel}
follows easily.
\printglossary
\end{document}

```

Figure 1: Input of a simple example

Nomenclature

- σ The total mass of angels per unit area
- A The area of the needle point
- a The number of angels per unit area
- m The mass of one angel
- N The number of angels per needle point

Figure 2: Output of a simple example

of the first or between the first and the second argument, especially no line break (even with a %). So

```
\nomenclature{ $x$ }%  
  {Description}
```

does *not* work. You can have have line breaks in the argument, but also no %.

2.3 Package Options

The `nomencl` package has the following options:

refeq The phrase “, see equation (*eq*)” is appended to every entry in the nomenclature where *eq* is the number of the last equation in front of the corresponding command `\nomenclature`.

norefeq No equation reference is printed. (default)

refpage The phrase “, page *page*” is appended to every entry in the nomenclature where *page* is the number of the page on which the corresponding command `\nomenclature` appeared.

norefpage No page reference is printed. (default)

prefix Every sort key is preceded by the letter “a” (changeable); see Section 3 to learn why this might make sense. (default)

noprefix No prefix is used.

cfg A configuration file `nomencl.cfg` is loaded, if it exists. (default)

nocfg The configuration file is not loaded.

croatian, danish, english, french, german, italian, polish, russian, spanish, ukrainian The reference texts and the nomenclature title will appear in the corresponding language. Note that in order to use Russian or Ukrainian, you have to have Cyrillic fonts installed and you might need a replacement for *MakeIndex*, e. g. **xindy**. Please help me out with other languages. (default: english)

2.4 Referencing

<code>\refeq</code>	As explained in Section 2.3, you can turn referencing to equations and
<code>\refpage</code>	pages on/off globally using the package options. But sometimes you might
<code>\refeqpage</code>	want to change the referencing behavior for single entries. The following
<code>\norefeq</code>	six macros can be used inside a <code>\nomenclature</code> macro: <code>\refeq</code> , <code>\norefeq</code> ,
<code>\norefpage</code>	<code>\refpage</code> , <code>\norefpage</code> , <code>\refeqpage</code> , <code>\norefpage</code> . The first four work
<code>\norefpage</code>	

similar to the package options, only local to the entry; the last two are shortcuts, but saying `\refeqpage` is equivalent to `\refeq\refpage`.

If we change the relevant parts of the last example as shown in Figure 3 then the nomenclature should look like Figure 4.

```

\begin{equation}
  a=\frac{N}{A}
\end{equation}%
\nomenclature{$a$}{The number of angels per unit area\refeqpage}%
\nomenclature{$N$}{The number of angels per needle point\refeq}%
\nomenclature{$A$}{The area of the needle point\refeq\refpage}%
The equation  $\sigma = m a$ 
\nomenclature{$\sigma$}{The total mass of angels per unit area}%
\nomenclature{$m$}{The mass of one angel\refpage}
follows easily.
\printglossary
\end{document}

```

Figure 3: Input with references

Nomenclature

- σ The total mass of angels per unit area
- A The area of the needle point, see equation (1), page 1
- a The number of angels per unit area, see equation (1), page 1
- m The mass of one angel, page 1
- N The number of angels per needle point, see equation (1)

Figure 4: Output with references

While these macros do not have to be at the end of the entries, it's probably the most sensible place to put them. Note that such local request always supersede the package options.

3 Sort Order of the Entries

The Greek letter σ turned out to be first in the nomenclature list in the examples above because the backslash in `\sigma` precedes any alphabetical character. Sometimes this is not what you want. Then you can use `<prefix>` to fine tune the sort order.

Before we describe the usage of `<prefix>`, we have to explain how *MakeIndex* sorts entries, see [2]. *MakeIndex* distinguishes three kinds of sort keys:

Strings Everything that starts with a alphabetic letter (A...Z, a...z).

Numbers Everything that starts and only contains digits (0...9).

Symbols Everything else.

Each group is sorted separately (and differently), then the groups are sorted in the order symbols, numbers, strings⁴. For the groups the following algorithm⁵ is used:

Strings If two letters are compared, the usual ordering is used (`a<C<q`), but if two words are the same except for the capitalization, then an upper case letter precedes the lower case letter (`Tea<tea`). If a letter is compared with a non-letter (digit, symbol), ASCII code is used (`1<A, A<~`).⁶ If two non-letters are compared (which can not happen at the first position of a string), ASCII code is used (`+<1<:<\`). Additionally there is the issue of word ordering (treat spaces as letter with ASCII code smaller than every printable symbol) and letter ordering (ignore spaces). *MakeIndex* uses word ordering by default, but you can change it with some command line option (`-l` on my UNIX).

Numbers The natural ordering is used (`8<34<111`).

Symbols ASCII code is used (`+<1<:<A<\<a`).

Why did you have to read all this?⁷ Let's consider the following eight nomenclature entries (without the optional argument): `$~Ab$`, `$~aa$`, `\Ab`, `\aa`, `Ab`, `aa`, `Ab`, `aa`. Try to understand the following example with the help of the explanation above and an ASCII table.

⁴With the `-g` switch of *MakeIndex*, they are sorted in the order symbols, strings, numbers.

⁵This is only vaguely described in [2], so I had to figure out special cases by myself. Please correct me if I am wrong

⁶An exception seems to be that the non-letters between upper and lower case letters (code 91–96) are put just before the capital letters (between code 64 and 65) while the non-letters after the lower case letters (code 123–127) are left there. Can someone please enlighten me why?

⁷I hope you did read it ;-)

If you use `nomencl` with its default settings (i. e. “a” is added to every sort key, so every sort key is considered as a string), you will get the sort order $\$aa$, $\$Ab$, $\$aa$, $\$Ab$, $\$~aa$, $\$~Ab$, aa , Ab . Note that aa is in front of Ab in all four pairs; note also the order $\$Ab$, $\$Ab$, $\$~Ab$ which does not agree with the ASCII code.

If you specify the option `noprefix`, then you will get $\$Ab$, $\$Ab$, $\$aa$, $\$aa$, $\$~Ab$, $\$~aa$, aa , Ab . The first six entries are considered as symbols and sorted according to the ASCII code (this time correctly). Note that $\$Ab$ is in front of $\$aa$ because A has the smaller ASCII code. The two strings follow at the end.

Decide for yourself what you prefer. Personally, I like to specify the `noprefix` option and use the optional argument to get exactly the sort order I want. See Section 5 for some special effects.

4 Customization

Besides the things you can customize by using the package options there are a few more commands that you might want to redefine. If you make the same changes in every file, it’s probably easier to put all those in a file `nomencl.cfg` which is automatically read by the `nomencl` package whenever it exists in the search path (unless you specified the `nocfg` option).

4.1 Formatting the Nomenclature

`\printglossary` Probably the most common change to the nomenclature is a different amount of space for the symbols. By default, the nomenclature is formatted as a list with the label width equal to `\nomlabelwidth` which is initialized to 1 cm. You can change this dimension in the `cfg` file or you can use the optional argument of `\printglossary`. If you want to have a little more space for the labels (and you don’t live in a metric world) you can use

```
\printglossary[0.5in]
```

instead of the simple

```
\printglossary
```

`theglossary` If you don’t like the format of the nomenclature at all, you will have to redefine the `theglossary` environment. Maybe a look at the documented code of `nomencl` will help.

`\nomname` In case you don’t like the name of the nomenclature, just redefine the `\nomname` macro, e. g.

```
\renewcommand{\nomname}{List of Symbols}
```

If you are using e. g. the documentclass `book` with page style headings you should also take care of correct headings:

```
\cleardoublepage% or \clearpage
```

```
\markboth{\nomname}{\nomname}% maybe with \MakeUppercase
\printglossary
```

I thought about putting this in the definition of `\printglossary` but decided that it is much easier for the user to add it if he wants than to remove it if he doesn't want it. In case you always need this just define a macro in `nomenc1.cfg` that executes these three lines all at once and can be used instead of `\printglossary`.

`\nomgroup` Usually, *MakeIndex* inserts the macro `\indexspace` between every character group, i. e. between symbols and numbers, numbers and letters and between every two letter groups. The `nomenc1` package inserts the macro `\nomgroup{<arg>}` *instead*, where `<arg>` is either the string “Symbols” or the string “Numbers” or the capital letter of the group that is about to start. You can redefine `\nomgroup` to insert some white space

```
\renewcommand{\nomgroup}[1]{\medskip}
```

or to print a fancy divider

```
\renewcommand{\nomgroup}[1]{%
  \item[]\hspace*{-\leftmargin}%
  \rule[2pt]{0.45\linewidth}{1pt}%
  \hfill #1\hfill
  \rule[2pt]{0.45\linewidth}{1pt}}
```

Note that `\nomgroup` is executed in a list environment, so you need to have an `\item` first and then jump back to the beginning of the line with the `\hspace` command.

`\nompreamble` Maybe you want to explain something just between the title of the
`\nompostamble` nomenclature and the start of the list or at the very end of the list. Just
 redefine the macros `\nompreamble` and `\nompostamble` which do nothing
 by default. Note that they are executed *outside* of the list environment.

`\nomitemsep` The skip between two entries in the nomenclature can be adjusted using
`\nomitemsep`. This should be done in the preamble or the file `nomenc1.cfg`.
 Note that if you want no extra skip between entries you have to use

```
\setlength{\nomitemsep}{-\parsep}
```

`\nomprefix` If you want, you can redefine the default prefix that is used for the
 sortkeys. By default, `\nomprefix` is set to “a”; redefining it supersedes the
 package options `prefix` and `noprefix`.

4.2 Formatting the Entries

`\nomlabel` By default, the labels are just shifted to the left within their allocated
 box. If you want to change this, redefine `\nomlabel` which should get one
 argument, e. g.

```
\renewcommand{\nomlabel}[1]{\hfil #1\hfil}
```

to center the symbols.

`\numentryend` Maybe you would like to have a period at the end of every entry. Just

say
`\renewcommand{\numentryend}{.}`

and there it is. Section 5.2 explains another nice application of this macro.

`\eqdeclaration` If you don't like the text that is used for the references to equations
`\pagedeclaration` and pages, you can define `\eqdeclaration` and `\pagedeclaration`. Both
should accept one argument, namely the equation and page number, re-
spectively. An example is

`\renewcommand{\eqdeclaration}[1]{, first used in eq.~(#1)}`
If you are redefining these macros for a particular language, let me know
and I will add that language to the next release of the `nomenc1` package.

5 Tips and Tricks

In this section, I will gather fancy stuff that people did or might want to do with the `nomenc1` package. Please email any ideas you have.

For most examples, sample configuration files will be generated if you run `LATEX` on the `ins` file. There will for example be a file `sample01.cfg` for the subgroups example in Section 5.1. Rename it to `nomenc1.cfg`, then it will automatically be used by your document. There is no sample file for the `longtable` example in Section 5.3. I am just too lazy right now, maybe I will add it later...

5.1 Subgroups

If you have distinct groups among the identifiers in your nomenclature (e.g. Greek letters for physical constants, Roman letters for variables), you can use the optional argument of `\nomenclature` together with the `\nomgroup` macro to get two groups with separate headings in the nomenclature.

Use something like the following throughout your text

```
\nomenclature[ga ]{ $\alpha$ }{Constant}
\nomenclature[rx ]{ $x$ }{Variable}
```

where “g” and “r” indicate Greek and Roman letters, respectively. Then you include the `ifthen` package and redefine `\nomgroup` e.g. like this.

```
1 (*sample01)
2 \RequirePackage{ifthen}
3 \renewcommand{\nomgroup}[1]{%
4   \ifthenelse{\equal{#1}{R}}{\item[\textbf{Variables}]}{%
5     \ifthenelse{\equal{#1}{G}}{\item[\textbf{Constants}]}{}}
6 \}
```

Note that we have to check for capital letters. All your symbols should have some kind of prefix; maybe you can also use the default prefix “a”. Note that for symbols and numbers you have to check for the strings “Symbols” and “Numbers”.

5.2 Units

Besides the obvious possibility of adding units for symbols in the description string, you can also use `\nomentryend` to shift the unit to the right margin. Something along the lines of a macro

```
7 (*sample02)
8 \newcommand{\nomunit}[1]{%
9   \renewcommand{\nomentryend}{\hspace*{\fill}#1}}
10 (/sample02)
```

should do the job. You can use this macro like this

```
\nomenclature{$l$}{Length\nomunit{m}}
```

Note that the nomenclature will not be a tabular with three columns, but it is pretty close as long as you only have one-line descriptions. Any suggestions for improvements are welcome.

5.3 Using a Long Table instead of a List

The following idea was sent to me by Brian Elmegaard. I have modified it a little bit to make it work with the current version of `nomencl`. Only the basic idea is given, so you have to do some extra thinking (and coding) to get it to work the way you want it.

After loading the `longtable` package in the preamble we first have to modify the macro that writes the entries to the `glo` file (do this is an style file).

```
\def\@@@nomenclature[#1]#2#3{%
\def\@tempa{#2}\def\@tempb{#3}%
\protected@write\@glossaryfile{%
  {\string\glossaryentry{#1\nom@verb\@tempa @{\nom@verb\@tempa}&%
    \begingroup\nom@verb\@tempb\protect\nomeqref{\theequation}%
    |nompageref}{\thepage}}%
\endgroup
\@esphack}
```

Then the nomenclature itself must be changed to start a `longtable` instead of a list. Maybe we could add something for a repeating header on every page.

```
\def\theglossary{%
  \@ifundefined{chapter}{\section*}{\chapter*}{\nomname}%
  \nompreamble
  \begin{longtable}[l]{@{}l@{}}
\def\endtheglossary{%
  \end{longtable}%
  \nompostamble}
```

Finally we add the following two lines at the end of `nomenc1.ist`⁸.

```
item_0 ""
delim_t " \\\n"
```

As I said, this is only the basic idea. An advantage might be the repeating headers on every page, a disadvantage is that there won't be any line breaks in the second column.

5.4 I want it expanded!

The `nomenc1` package tries hard to write the arguments of the `\nomenclature` macro verbatim to the glossary file. This is usually the right thing to do because some macros do not like to be expanded at the wrong moment or give weird results if they are. On the other hand, there are occasions where it is good to have the meaning (or expansion) of a macro in the glossary file instead of its name. There are quite some occasions where you will get in trouble with this expansion, for example, if the expansion of a macro contains `@` (`\mathcal` expands to `\@mathcal`) because `@` is a special character for *MakeIndex* and thus *MakeIndex* will either fail or give unexpected results. You can avoid the expansion on a case by case basis by using `\protect` in front of the macro that should not be expanded.

In order to get macro expansion, the redefinition of the `\@nomenclature` macro within the `\makeglossary` macro has to be changed.

```
11 (*sample04)
12 \def\makeglossary{%
13   \newwrite\@glossaryfile
14   \immediate\openout\@glossaryfile=\jobname.glo
15   \def\@nomenclature{%
16     \@ifnextchar[%
17       {\@@@nomenclature}{\@@@nomenclature[\nomprefix]}}%
18   \typeout{Writing glossary file \jobname.glo}%
19   \let\makeglossary\empty}
```

The new macro to be called by `\@nomenclature` just writes its arguments to the glossary file without further ado, so they will be expanded.

```
20 \def\@@@nomenclature[#1]#2#3{%
21   \protected@write\@glossaryfile{%
22     {\string\glossaryentry{#1#2@[#2]}%
23       \begingroup#3\protect\nomeqref{\theequation}%
24       |nompageref}{\thepage}}}%
25 \end{sample04}
```

As I said above, use these macros with care and look for warnings and errors issued by *MakeIndex*.

⁸Don't forget to rename the file and delete my email address if you want to distribute the file, see the pointer to the LPPL in Section 7.

5.5 Glossary in “Kopka Style”

I was told that the glossary in the L^AT_EX book by Kopka looks roughly like in Figure 5. In order to get a glossary like, there are quite some configurations to do.

Symbol	page number
Explanation.	

Figure 5: Glossary entry in “Kopka Style”

First we have to change the macro `\@@@nomenclature` which takes care of writing the glossary entry to the glossary file. The only difference to the original definition is that we hand over the explanation of a symbol (`#3`) and the equation number to `\nompageref` instead of writing it directly after the symbol (`#2`). This is necessary because the explanation should appear after (actually below) the page number.

```
26 <*sample05>
27 \def\@@@nomenclature[#1]#2#3{%
28   \def\@tempa{#2}\def\@tempb{#3}%
29   \protected@write\@glossaryfile{%
30     {\string\glossaryentry{#1\nom@verb\@tempa @[\@tempa]}%
31       \nompageref{\begingroup\nom@verb\@tempb\protect\nomeqref{\theequation}}}%
32     {\thepage}}%
33   \endgroup
34   \@esphack}
```

Now we change the definition of `\nompageref` so that it accepts two arguments, the explanation (`#1`) and the page number (`#2`). The page number is only printed if required, otherwise `\null` is used to avoid an error because of the following `\linebreak`. Note that it is *not* possible to turn off the page number locally, because the explanation appears after the page number. Does anyone have an idea how to fix this?

```
35 \def\nompageref#1#2{%
36   \if@printpageref\pagedeclaration{#2}\else\null\fi
37   \linebreak#1\numentryend\endgroup}
```

And a few little things. We want dots and a space before the page number appears at the right margin; the explanation should end with a period; and the symbol should be printed in bold face (this only works for regular text, not for formulas).

```
38 \def\pagedeclaration#1{\dotfill\nobreakspace#1}
39 \def\numentryend{.}
40 \def\nomlabel#1{\textbf{#1}\hfil}
41 </sample05>
```

6 Acknowledgements

First and foremost I want to thank Boris Veytsman, who had the idea for the package, maintained it until v2.2 and provided some helpful advice for the new version. I also want to thank Stefan Böhm and Karl Heinz Marbaise who helped testing this package.

For helping out with translations I thank Branka Lončarević (Croatian), Brian Elmegaard (Danish), Denis B. Roegel (French), Sani Egisto (Italian), Artur Gorka (Polish), Pedro Areal (Portuguese), Alejandro Lopez-Valencia (Spanish) and Boris Veytsman (Russian and Ukrainian).

7 Releases and Legal Issues

This package can be redistributed and/or modified under the terms of the L^AT_EX Project Public License distributed from CTAN archives in the directory `macros/latex/base/lpp1.txt`, see e.g. [3]; either version 1.2 of the license, or (at your option) any later version.

The most recent release of the `nomenc1` package can always be found at <http://members.nbc1.com/schand1/nomenc1/>. Usually, the same version is also available at `CTAN/macros/latex/contrib/supported/nomenc1/`.

References

- [1] Braams, Johannes; Carlisle, David; Jeffrey, Alan; Lamport, Leslie; Mittelbach, Frank; Rowley, Chris; Schöpf, Rainer (1996). `ltxidxglo.dtx` – 1996/01/20 v1.1e LaTeX Kernel (Index and Glossary). `CTAN/macros/latex/base/ltxidxglo.dtx`.
- [2] Chen, Pehong; Harrison, Michael A. (1987). Automating Index Preparation. Report UCB/CSD 87/347, Computer Science Division, University of California, Berkeley, CA.
- [3] Comprehensive T_EX Archive Network CTAN. <ftp://ctan.tug.org/tex-archive/>.
- [4] Jones, David M. (1995). A new implementation of L^AT_EX's indexing commands, Version v4.1beta of 1995/09/28. `CTAN/macros/latex/contrib/supported/camel/index.dtx`.
- [5] Knuth, Donald E. (1984). *The T_EXbook*. Addison-Wesley Publishing Company, Reading, MA.
- [6] Lamport, Leslie (1994). *L^AT_EX: A Document Preparation System*. Addison-Wesley Publishing Company, Reading, MA.

- [7] Veytsman, Boris (1996). Package nomencl, Version 2.2. <http://www.plmsc.psu.edu/~boris/nomencl/> (2000/09/15).